BionicLoop IDE Software Packet Review Binder

BionicLoop IDE Software Packet Review Binder

Single-file review copy of the current submission-facing software packet.

Packet type
: IDE software review binder

Generated
: 2026-04-07 14:42 EDT

Prepared by
: BionicLoop engineering

Source folder
: /Users/jcostik/BionicLoop/Docs/Quality/IDE_Software_Packet

Internal Review Copy
Scope note: This binder is a single-file review copy of the current submission-facing software packet. It is not the final IDE submission assembly.

Included Documents

1. **IDE Software Baseline Overview**
   Docs/Quality/IDE_Software_Packet/IDE_Software_Baseline_Overview.md
2. **IDE Software Scope And Deferred Items**
   Docs/Quality/IDE_Software_Packet/IDE_Software_Scope_and_Deferred_Items.md
3. **IDE Software Risk And Cybersecurity Summary**
   Docs/Quality/IDE_Software_Packet/IDE_Software_Risk_and_Cybersecurity_Summary.md
4. **IDE Software Prior Testing Summary**
   Docs/Quality/IDE_Software_Packet/IDE_Software_Prior_Testing_Summary.md
5. **IDE Software Verification Summary**
   Docs/Quality/IDE_Software_Packet/IDE_Software_Verification_Summary.md
6. **Risk Analysis (Quality)**
   Docs/Quality/RiskAnalysis.md
7. **Software Requirements Specification (SRS)**
   Docs/Quality/SoftwareRequirementsSpecification.md
8. **Software Design Description (SDD)**
   Docs/Quality/SoftwareDesignDescription.md
9. **Software Verification and Validation Plan (SVVP)**
   Docs/Quality/SoftwareVerificationAndValidationPlan.md
10. **Requirements Traceability Matrix (RTM)**
    Docs/Quality/TraceabilityMatrix.md
11. **Cybersecurity Plan (Investigational Build)**
    Docs/Quality/CybersecurityPlan.md
12. **BionicLoop Instructions for Use (IFU)**
    Docs/Quality/IFU/BionicLoop_IFU_v1.4.md

Table of Contents

- Table of Contents
- IDE Software Baseline Overview
  - Purpose
  - Current Baseline
  - Software Under Review
  - Intended Investigational Operating Model
  - High-Level Functional Areas In Scope
  - Explicit Baseline Boundaries
  - Primary Support References
- IDE Software Scope And Deferred Items
  - Purpose
  - In Scope For The Current Packet
  - Deferred From The Current Packet
  - Accepted Current-Baseline Decisions
  - Not Engineering-Owned In This Pass
  - Practical Review Rule
  - Primary Support References
- IDE Software Risk And Cybersecurity Summary
  - Purpose
  - Main Software Risk Themes
  - Current Risk-Control Position
  - Cybersecurity Boundary

    - Current Cybersecurity Recommendation
    - Conditions Required For The Investigational Baseline
    - Open Cybersecurity Items That Still Matter
    - Primary Support References
- IDE Software Prior Testing Summary
    - Purpose
    - Current Evidence Shape
    - What This Means For IDE
    - Areas With Strong Existing Development Coverage
    - What Still Needs Freeze-Time Promotion Or Audit
    - Reviewer Guidance
    - Primary Support References
- IDE Software Verification Summary
    - Purpose
    - Current Verification Posture
    - Verification Rule
    - Current Blocking Verification Items
    - Primary Support References
- Risk Analysis (Quality)
    - Revision History
    - Method
    - Risk Register
    - Residual Risk Acceptance Notes
- Software Requirements Specification (SRS)
    - Revision History
    - Requirement Format
    - Runtime and Cadence
    - CGM Input Handling
    - Fingerstick BG Input Handling
    - Pump Handling
    - Meal Announce
    - State Persistence and Recovery
    - Algorithm Verification Controls
    - Telemetry and Traceability
    - UI Safety and Status
    - Clinical Settings and Pregnancy Configuration
    - User Alerts and Escalation
    - Security and Privacy
    - Traceability Source Mapping
- Software Design Description (SDD)
    - Revision History
    - 1. Design Intent
    - 2. Major Components
    - 3. Runtime Sequence (Nominal)
    - 4. Key Design Policies
    - 5. Data Persistence
    - 6. Requirement Allocation
    - 7. Design Constraints
    - 8. BG Design and Pending Decisions
    - 9. Simulation Strategy (Medium now, High later)
- Software Verification and Validation Plan (SVVP)
    - Revision History
    - 1. Test Document Acronyms
    - 2. Verification Strategy
    - 3. Test Environments
    - 4. Entry and Exit Criteria
    - 5. Seed Test Inventory
    - 6. Evidence
    - 7. Deferred/Planned Validation
    - 8. Xcode Automated UI Testing Strategy
    - 9. UI Automation Verification Mapping
    - 10. Manual Screenshot UI Review Protocol
- Requirements Traceability Matrix (RTM)
    - Revision History
    - Matrix
    - Usage
    - Notes
- Cybersecurity Plan (Investigational Build)
    - Revision History
    - 1. Scope
    - 2. Security Objectives
    - 3. System Trust Boundary Summary
    - 4. Supplier / Inherited Control Matrix
    - 5. BionicLoop-Owned Local Security Controls

- 6. Security Baseline Controls
- 7. Threat Scenarios (Current Focus)
- 8. Planned Evidence Artifacts
- 9. Current Control and Evidence Matrix
- 10. Identity and Access Planning (Deferred from Current Software Handoff Package)
- 11. Current Missing Artifacts / Open Cyber Gaps
- 12. Deferred Auth / Cloud Scope Note
- 13. Incident Handling (Seed)
- 14. Regulatory Alignment
- Revision History
- Investigational Use Statement
- 1. About This Guide
- 2. System Overview
- 3. Before First Use
- 4. Sign In and Account Access
- 5. Home Screen Orientation
- 6. Settings, Clinical Configuration, and Dose Review
- 7. Dexcom G7 Workflow
- 8. Pod Workflow
- 9. Routine Daily Workflows
- 9.1 Manual BG Entry
- 9.2 Meal Announcement
- 9.5 Alert Review
- 10. Alerts and User Response
- 11. Troubleshooting and Recovery
- 12. Documentation and Escalation
- Appendix A. Figure Index
- Appendix B. Abbreviations and Acronyms
- Appendix C. Daily Use Quick Checklist

The table of contents is generated automatically from binder headings and links to each section in the PDF.
Included source: Docs/Quality/IDE_Software_Packet/IDE_Software_Baseline_Overview.md

IDE Software Baseline Overview

Status: Reviewer summary Owner: BionicLoop engineering Last updated: 2026-04-07 12:32 EDT

Purpose

Describe the current investigational software baseline in concise reviewer-facing terms.

This summary is not the full architecture or design history. It is the short-form software/device description for the current IDE software packet.

Current Baseline

- Working baseline commit for this packet: `b302dd3`
- Freeze-time baseline SHA is still to be filled at actual package freeze

Software Under Review

The investigational controller is an iPhone application composed of:

- `BionicLoop` app UI and orchestration layer
- `BionicLoopCore` runtime, domain, persistence, and algorithm-host logic
- embedded `LoopKit` support code
- embedded `G7SensorKit` integration for Dexcom G7 data intake
- embedded `OmniBLE` integration for Omnipod DASH pump communication
- the bridged dosing algorithm and supporting runtime telemetry/persistence surfaces

Intended Investigational Operating Model

The accepted baseline is a phone-controlled automated insulin-delivery workflow using:

- Dexcom G7 CGM data as the live glucose input
- Omnipod DASH as the insulin-delivery device
- BionicLoop as the controller/runtime/alerting UI

Current operating assumptions and boundaries:

- closed-loop guardrail remains algorithm-driven dosing only; there is no manual test-dose path

- meal announcement and manual BG entry are participant-facing workflow inputs in the current baseline
- Dexcom remains the source of truth for primary CGM alarming behavior
- the current package is focused on local device/app behavior, not full device-to-cloud or production auth closure

High-Level Functional Areas In Scope

- runtime step scheduling and wake-cause handling
- CGM intake and availability handling
- DASH pump status, delivery, and alert normalization
- algorithm execution and guarded command application
- Home, meal, BG, pump, and CGM operational workflows
- alert presentation, acknowledgement, and review behavior relevant to study use
- software-facing IFU and operational screenshot package

Explicit Baseline Boundaries

This packet does not establish closure for:

- full cloud/device-to-cloud verification
- Part 11 closure
- broader authentication/authorization/provider-policy closure
- broader cybersecurity hardening beyond the documented investigational baseline

Primary Support References

- IDE_Software_Handoff_Index.md
- SoftwareDesignDescription.md
- Requirements.md

Included source: Docs/Quality/IDE_Software_Packet/IDE_Software_Scope_and_Deferred_Items.md

IDE Software Scope And Deferred Items

Status: Reviewer summary Owner: BionicLoop engineering Last updated: 2026-04-07 12:32 EDT

Purpose

State clearly what engineering is claiming in the current IDE software packet, what is deferred, and what belongs to the receiving quality/submission team.

In Scope For The Current Packet

Topic	Current Position
Local app/runtime/algorithm/device software docs	In scope
Software verification structure and STP ownership mapping	In scope
Software-facing IFU package and screenshots	In scope
Study-relevant software risk summary	In scope
Proportionate cybersecurity summary for local software posture	In scope

Deferred From The Current Packet

Topic	Current Position	Why Deferred
Cloud / device-to-cloud verification closure	Deferred	Current handoff is focused on local controller software readiness, not cloud-system closure.
Part 11 readiness package	Deferred	Requires broader system and quality-process decisions beyond this software handoff slice.
`SRS-SEC-003..009` auth/provider/session-continuity closure	Deferred	These rows still involve broader provider-policy, auth-model, and cloud/compliance scope decisions.
`SRS-BG-008` / `TV-BG-007` step-0 BG rescue	Deferred	Current accepted baseline keeps step `0` CGM-only.

Accepted Current-Baseline Decisions

Item	Accepted Decision
SRS-MEAL-004	Late meal submit executes on the current due step.
SRS-CLIN-002	Current clinician gate remains the investigational passcode control with explicit transition required before production release.
TV-MEAL-003	Remains in scope because it reflects accepted current meal behavior.

Not Engineering-Owned In This Pass

Topic	Current Position
Formal review and approval signatures	Receiving quality/submission team
Final IDE submission assembly and release authorization	Receiving quality/submission team
Residual-risk acceptance signoff	Receiving quality/submission team
Supplier/manufacturing/training/non-software QMS artifacts	Outside engineering-owned packet

Practical Review Rule

If a reviewer asks whether a topic must block this software packet, default answer should be:

- `yes` only if it is part of the accepted local software baseline being claimed for study use
- `no` if it is deferred from the current packet or outside the engineering-owned software handoff scope

Primary Support References

- IDE_Software_Handoff_Disposition_Log.md
- IDE_Submission_Readiness_Report.md
- TraceabilityMatrix.md

Included source: Docs/Quality/IDE_Software_Packet/IDE_Software_Risk_and_Cybersecurity_Summary.md

IDE Software Risk And Cybersecurity Summary

Status: Reviewer summary Owner: BionicLoop engineering Last updated: 2026-04-07 12:32 EDT

Purpose

Provide a proportionate study-facing summary of the main software risk and cybersecurity points for the current IDE software baseline.

Main Software Risk Themes

The software risk story for the current baseline is concentrated in four areas:

1. incorrect or mistimed insulin-delivery decisions
2. stale, missing, or degraded CGM/pump state leading to poor control decisions
3. alerting/acknowledgement failures that could hide significant device/runtime conditions
4. local telemetry/export handling and operator access to stored investigational data

Current Risk-Control Position

Key controls already represented in the codebase and quality chain include:

- guarded runtime execution and command application when pump state is unavailable
- normalized alert taxonomy with explicit severity and acknowledgement behavior
- clinician-gated critical settings
- explicit accepted/deferred requirement dispositions rather than silent ambiguity
- unit/integration/UI verification structure across algorithm, runtime, alerting, and operator workflow surfaces

The full risk register and trace mapping remain in:

- RiskAnalysis.md
- TraceabilityMatrix.md

Cybersecurity Boundary

The current software baseline relies on a split boundary:

- manufacturer/device-side controls for Omnipod DASH and Dexcom G7 are inherited and documented as relied-upon controls

- BionicLoop owns the local controller-app behaviors, including local storage, export, logging, and package provenance statements

Current cyber scope for this software baseline is limited to:

- local file/export posture
- permissions/background-mode posture
- secret/logging review posture
- provenance/process notes for embedded packages
- explicit statement of what cloud/auth/security closure is deferred

Current Cybersecurity Recommendation

Engineering's current recommendation is:

- acceptable with explicit conditions for the current investigational software handoff baseline
- not positioned as a broader release baseline without additional hardening

Reason:

- the main local issue is real and explicit: step telemetry CSV is automatically exported into Documents
- file sharing and open-in-place are currently enabled
- reviewed auth/network logging does not directly expose bearer tokens or passwords in the reviewed paths
- the package does not over-claim ownership of DASH/G7/Dexcom-app security behavior

Conditions Required For The Investigational Baseline

The current investigational recommendation assumes:

1. the local CSV/export posture is explicitly accepted as part of the study baseline
2. `TV-SEC-001` is promoted into formal evidence at freeze
3. the current file-sharing/open-in-place posture is explicitly dispositioned in the submission record
4. the software baseline continues to avoid broader release claims for deferred auth/cloud/security areas

Open Cybersecurity Items That Still Matter

- external supplier/FDA artifact linkage for inherited DASH/G7/Dexcom-app claims
- freeze-time `TV-SEC-001` execution
- freeze-time SBOM/advisory artifact execution
- documented acceptability decision on the investigational export/file-sharing posture

Primary Support References

- CybersecurityPlan.md
- Cybersecurity_Handoff_Register.md
- Cybersecurity_Baseline_Acceptability_Recommendation.md
- RiskAnalysis.md

Included source: Docs/Quality/IDE_Software_Packet/IDE_Software_Prior_Testing_Summary.md

IDE Software Prior Testing Summary

Status: Reviewer summary Owner: BionicLoop engineering Last updated: 2026-04-07 12:32 EDT

Purpose

Summarize the current body of prior software testing that supports the investigational baseline, without requiring reviewers to start from the full RTM or detailed STP/STR library.

Current Evidence Shape

The current software package already has broad engineering evidence across:

- unit tests for core algorithm-host, runtime policy, alerting, persistence, and safety behavior
- app-side unit tests for Home, meal, BG, CGM, pump, and alert workflows
- targeted UI tests for critical operator flows
- STP ownership documents for algorithm, automation, simulation, hardware, and alert drill lanes
- controlled trace chain linking risk, requirements, design, verification rows, and review evidence

What This Means For IDE

For this packet, the useful claim is:

- there is already a substantial prior engineering test body supporting the accepted local software baseline
- the remaining work is not to invent new verification structure, but to trim and promote the right subset into the formal frozen IDE evidence set

Areas With Strong Existing Development Coverage

- runtime wake-cause and degraded-mode behavior
- meal announcement and manual BG workflow handling
- alert normalization, acknowledgement, and notification behavior
- clinical settings gating and applied-configuration behavior
- major Home/pump/CGM workflow rendering and state transitions

What Still Needs Freeze-Time Promotion Or Audit

- RTM rows still marked `In progress`, `Partial`, or `Pending`
- final audit of which rows are truly claimed in the IDE packet
- formal promotion/rerun of specific high-risk or in-scope rows, including `TV-SEC-001`
- final baseline SHA capture for the frozen evidence set

Reviewer Guidance

This packet should be read as:

- evidence that the software package has meaningful prior engineering verification depth already
- not yet a claim that every detailed RTM row is formally closed for the frozen IDE baseline

The freeze-time task is to carry forward only the evidence needed for the accepted packet scope.

Primary Support References

- SoftwareVerificationAndValidationPlan.md
- TraceabilityMatrix.md
- STP/README.md
- IDE_Submission_Readiness_Report.md

Included source: Docs/Quality/IDE_Software_Packet/IDE_Software_Verification_Summary.md

IDE Software Verification Summary

Status: Reviewer summary Owner: BionicLoop engineering Last updated: 2026-04-07 12:32 EDT

Purpose

Show the current verification posture for the IDE software packet at a level suitable for review planning.

Current Verification Posture

Area	Current Posture	Freeze-Time Need
Core runtime and algorithm-host behavior	Structured unit/integration coverage exists and is already mapped into the controlled quality chain.	Keep only the rows actually claimed for IDE and promote their evidence into the formal freeze set.
Device integration and alerting behavior	App-side tests and supporting protocol ownership exist for CGM, DASH, and normalized alert behavior.	Audit claimed rows and rerun/promote only the accepted IDE-scope evidence.
Critical operator workflows	Targeted UI and app-level regression coverage exists for Home, meal, BG, settings, alerts, and related workflows.	Carry forward only the workflows that materially support study use.
Cybersecurity export/file handling	Review notes, checklist, and engineering recommendation exist.	Execute and promote `TV-SEC-001` at freeze.
Detailed RTM closure	Structure is in place, but many rows are still marked `In progress` or reference working evidence.	Complete the packet-scope RTM audit and stop claiming working-lane evidence as closure evidence.

Verification Rule

For this packet, verification should be considered sufficient when:

- accepted IDE-scope behaviors are clearly identified
- each claimed behavior has a credible verification path
- freeze-time evidence promotion is limited to the claimed subset

This packet does **not** require:

- formal trace closure for deferred scope
- formal promotion of verification lanes not being claimed for this packet

Current Blocking Verification Items

1. RTM evidence audit into `formal-ready`, `rerun-needed`, and `deferred`
2. final formal execution/promotion for the claimed rows still lacking freeze-ready evidence
3. replacement of working-lane references in any row still being claimed

Primary Support References

- TraceabilityMatrix.md
- SoftwareVerificationAndValidationPlan.md
- IDE_Submission_Closure_Checklist.md
- IDE_Submission_Readiness_Report.md

Included source: Docs/Quality/RiskAnalysis.md

Risk Analysis (Quality)

Status: Submission-candidate risk analysis (approval and freeze metadata pending) Version: 0.91 Owner: BionicLoop engineering
Prepared by: BionicLoop engineering Reviewer: _____
Approver: _____
Decision date: _____
Effective date: _____
Baseline freeze SHA: _____
Scope: BionicLoop investigational closed-loop app (Dexcom G7 + OmniPod DASH + Algo2015) Last updated: 2026-04-07 14:17 EDT

Revision History

Version	Date	Author	Summary of Changes
0.1	2026-03-25	Engineering	Initial controlled draft baseline
0.9	2026-04-06	BionicLoop engineering	Added handoff-ready metadata and refined software-only cybersecurity boundary language for `RA-009`
0.91	2026-04-07	BionicLoop engineering	Tightened residual-risk wording for local telemetry exposure, investigational clinical gating, and reconnect-fallback behavior to match the current baseline and remaining freeze-time evidence posture

Method

Each hazard is tracked with:

- `RA-ID`
- hazard statement
- initial severity/probability
- controls
- linked requirements (`SRS-*`)
- linked verification (`TV-*` / manual protocol IDs)
- residual-risk note

Severity and probability are currently qualitative (`Low`, `Medium`, `High`) and can be replaced with numeric scoring later.

Risk Register

RA-ID	Hazard	Initial Risk	Key Controls	SRS Links	Verification Links	Residual Risk
RA-001	Loop step not executed on expected cadence due to iOS wake variability	High	CGM-triggered doWork, bounded wake-cause policy, cadence anchor, skip logic, stale-state visibility	SRS-RUN-001, SRS-RUN-002,	TV-RUN-001, TV-RUN-002, TV-RUN-003	Medium

RA-ID	Hazard	Initial Risk	Key Controls	SRS Links	Verification Links	Residual Risk
				SRS-RUN-003, SRS-UI-001		
RA-002	Invalid CGM values used as real glucose input	High	Out-of-range sanitize to unavailable (`-1`), step-0 gate, degraded-step safety policy, explicit blocked-state messaging	SRS-CGM-001, SRS-CGM-002, SRS-CGM-003, SRS-CGM-004	TV-CGM-001, TV-CGM-002, TV-CGM-003, TV-CGM-004	Medium
RA-003	Pump unavailable causes incorrect delivery behavior	High	Degraded algorithm input with unavailable pump fields, command block on unavailable pump, auto-refresh delivery-state convergence, and closed-loop-only command exposure (no manual bolus path)	SRS-PUMP-001, SRS-PUMP-002, SRS-PUMP-005	TV-PUMP-001, TV-PUMP-002, TV-PUMP-005, TV-PUMP-006	Medium
RA-004	Meal announce accepted while pump not safe/ready	High	Meal announce pump-ready gating, unavailable-reason messaging, retry-timing feedback, and borrow-window enforcement	SRS-MEAL-001, SRS-MEAL-002, SRS-MEAL-003, SRS-MEAL-004, SRS-MEAL-005	TV-MEAL-001, TV-MEAL-002, TV-MEAL-003, TV-MEAL-004, TV-MEAL-005, TV-MEAL-006	Medium
RA-005	Dose quantization mismatch with DASH minimum deliverable dose	Medium	Algorithm constants + command quantization validation + telemetry reconciliation	SRS-PUMP-003, SRS-LOG-001	TV-PUMP-003, TV-LOG-001	Low/Medium
RA-006	App relaunch/reset causes state mismatch and unsafe cadence	High	Persisted runtime/algorithm/device-manager state and explicit full reset semantics	SRS-STATE-001, SRS-STATE-002, SRS-STATE-003	TV-STATE-001, TV-STATE-002, TV-STATE-003	Medium
RA-007	Home pod status stale despite reconnect	Medium	Connection-driven refresh path, observer wiring	SRS-PUMP-004	TV-PUMP-004	Low
RA-008	Missing auditability of algorithm inputs/outputs/commands	High	Per-step telemetry capture and traceability mapping	SRS-LOG-001, SRS-LOG-002	TV-LOG-001, TV-LOG-002	Medium
RA-009	Investigational telemetry artifacts stored or exported locally are accessed outside intended site/operator controls	High	Current baseline automatically writes detailed step telemetry to a Documents-based CSV export path; risk is bounded only by site/device physical control, operator handling procedures, explicit local file-access review, and receiving-team freeze disposition of the current `UIFileSharingEnabled` /	SRS-SEC-001, SRS-SEC-002	TV-SEC-001	High (accepted only if the investigational local export/file-sharing posture is explicitly

RA-ID	Hazard	Initial Risk	Key Controls	SRS Links	Verification Links	Residual Risk
			`LSSupportsOpeningDocumentsInPlace` posture. Cloud/auth/provider controls are documented for system context but are deferred from this IDE software baseline and are not claimed as active mitigations for this row.			approved at freeze)
RA-010	User workflow errors (profile, meal UI, status interpretation, stale/unreliable CGM value interpretation, incorrect device date/time leading to misleading recency context)	Medium	Explicit UI states, startup-cancel paths, background-auto-cancel for meal composer, stale/unreliable-CGM display masking (`--` and no trend arrow after 11 minutes or when `hasReliableGlucose == false`), boundary CGM rendering (`LOW`/`HIGH`), stepped CGM chart scaling (`300/350/400`), UTC clock-drift checks with actionable warning (`>10 min` skew, 24-hour warning rate limit), SOP-driven usability checks	SRS-UI-001, SRS-UI-002, SRS-UI-003, SRS-UI-004, SRS-UI-005, SRS-UI-006, SRS-UI-007, SRS-UI-008, SRS-VAL-001, SRS-LOG-006	TV-UI-001, TV-UI-002, TV-UI-003, TV-UI-004, TV-UI-007, TV-UI-008, TV-UI-009, TV-UI-010, TV-LOG-006	Medium
RA-011	Critical device/runtime alerts are missed, delayed, or masked by lower-severity messages	High	Multi-source alert normalization, severity precedence, debounce/coalescing rules, explicit ack/clear policy, protocol-aligned wording	SRS-ALERT-001, SRS-ALERT-002, SRS-ALERT-003, SRS-ALERT-004, SRS-ALERT-005, SRS-ALERT-006	TV-ALERT-001, TV-ALERT-002, TV-ALERT-003, TV-ALERT-004, TV-ALERT-005	Medium
RA-012	Manual BG entry is stale, invalid, duplicated, misapplied to wrong step, or executed while loop is off, causing incorrect algorithm input or user confusion	High	Dedicated `bgCheck` policy with single pending candidate, immediate-next-step-only validity, replace-on-new submit semantics, explicit stale/duplicate guards, loop-armed gating, accepted manual BG range `20...600 mg/dL`, source-tagged telemetry, and explicit user feedback for rejected BG actions	SRS-BG-001, SRS-BG-002, SRS-BG-003, SRS-BG-004, SRS-BG-005, SRS-BG-006, SRS-BG-007, SRS-BG-008, SRS-BG-009, SRS-BG-010, SRS-BG-011, SRS-BG-012	TV-BG-001, TV-BG-002, TV-BG-003, TV-BG-004, TV-BG-005, TV-BG-006, TV-BG-007, TV-BG-008, TV-BG-009, TV-BG-010, TV-BG-011, TV-BG-012	Medium
RA-013	Clinical settings misconfiguration or unauthorized access leads to unsafe algorithm/session configuration	High	Dedicated clinician-gated settings surface, investigational clinician passcode gate, strict selector bounds/step validation, lbs->kg normalization, persisted defaults/migration,	SRS-CLIN-001, SRS-	TV-CLIN-001, TV-CLIN-002, TV-CLIN-	Medium/High (accepted investigational residual;

RA-ID	Hazard	Initial Risk	Key Controls	SRS Links	Verification Links	Residual Risk
	(target/upfront/TMAX/subject/weight/start-reset controls)		clinician-controlled participant target-access profile (`Pregnancy` / `Standard`), required approval capture before participant target changes, profile-bounded clinician target options with nearest-allowed normalization on profile switch, and unchanged start/reset behavior semantics under relocated controls. The passcode gate is an investigational shared operational control only and is not authenticated role-based access.	CLIN-002, SRS-CLIN-003, SRS-CLIN-004, SRS-CLIN-005, SRS-CLIN-006, SRS-CLIN-007, SRS-CLIN-008, SRS-CLIN-009, SRS-CLIN-010, SRS-CLIN-011, SRS-CLIN-012, SRS-VAL-001, SRS-LOG-008	003, TV-CLIN-004, TV-CLIN-005, TV-CLIN-006, TV-CLIN-007, TV-CLIN-008, TV-CLIN-009, TV-CLIN-010, TV-CLIN-011, TV-CLIN-012, TV-CLIN-013, TV-LOG-008	replace passcode-only gating before production release)
RA-014	Algorithm logic regression or latent C++ branch defects remain undetected due to insufficient structural/deterministic test coverage, leading to incorrect dosing recommendations under specific input sequences	High	Dedicated Algo2015 structural-coverage campaign, deterministic golden-vector replay suite, bridge-contract edge-case testing, persistence/restart continuity testing, mandatory static-analysis lane on formal runs, and risk-based conditional MISRA closure (report+deviations or explicit N/A rationale) with IDE-traceable STR evidence packaging	SRS-ALG-001, SRS-ALG-002, SRS-ALG-003, SRS-ALG-004, SRS-ALG-005, SRS-ALG-006, SRS-ALG-007	TV-ALG-001, TV-ALG-002, TV-ALG-003, TV-ALG-004, TV-ALG-005, TV-ALG-006, TV-ALG-007, TV-ALG-008, TV-ALG-009, TV-ALG-010, TV-ALG-011	Medium
RA-015	The armed loop stops achieving timely successful steps and scheduled dosing recommendations may remain unserved without actionable user escalation	High	Current baseline monitors successful-step continuity rather than CGM receipt alone: while armed, the app computes a `15`-minute interruption deadline from the most recent successful step (or session start / first-step wait state if none exists yet), raises `ALERT-ALGORITHM-STEPPING-INTERRUPTED` with local notification when that deadline is missed, preserves Home status visibility, and clears interruption state on the next successful step or loop disarm/reset. The alert remains distinct from stronger CGM unavailable/failed and pump-native alerts and may include blocker-specific detail (`No CGM`, `No Pod`, signal loss, other runtime gates). Runtime now also permits	SRS-CGM-005, SRS-RUN-004, SRS-RUN-005, SRS-ALERT-013, SRS-ALERT-014,	TV-CGM-005, TV-RUN-004, TV-RUN-005, TV-RUN-006, TV-RUN-007, TV-ALERT-012, TV-ALERT-013, TV-UI-001	Medium

RA-ID	Hazard	Initial Risk	Key Controls	SRS Links	Verification Links	Residual Risk
			guarded reconnect-based fallback execution after the first anchored step exists using the current due step only without re-anchoring or backlog replay. Current implementation executes this path when accepted CGM receipt age exceeds the approved freshness limit and also when CGM freshness is unavailable; freeze closure shall either narrow the implementation to the stale-CGM-only gate or explicitly accept and verify the broader behavior. Remaining closure work is real-device validation of reconnect/background behavior.	SRS-UI-001, SRS-UI-002		
RA-016	Meal announce is duplicated, silently lost, or ambiguously applied because submit/outcome state is not durable across relaunch, comms interruption, or competing triggers	High	Implemented baseline now persists pending meal-request state plus correlated meal `flow_id` across relaunch, blocks duplicate meal entry while the intended execution step remains unresolved, defers Home success messaging/telemetry until runtime/coordinator result is known, blocks repeat meal entry with explicit operator guidance when command outcome is uncertain until reconciliation or session reset/disarm, rejects competing-trigger slot conflicts with explicit retry guidance instead of silently reassigning meal intent to a later borrowed step, and emits `accepted` / `resolved` telemetry closure when the request is accepted, reconciles after uncertainty, or is cleared by session reset. Loop command telemetry preserves uncertain-vs-blocked semantics for cloud reconstruction.	SRS-MEAL-007, SRS-MEAL-008, SRS-MEAL-009, SRS-MEAL-010, SRS-LOG-007, SRS-UI-002	TV-MEAL-008, TV-MEAL-009, TV-MEAL-010, TV-MEAL-011, TV-LOG-007	Medium

Residual Risk Acceptance Notes

- `RA-009`: The current local export/file-sharing posture requires explicit receiving-team acceptance for the investigational baseline and is not suitable for a broader production claim without hardening.
- `RA-013`: Passcode-only Clinical Settings gating is accepted only as an investigational operational control and must be replaced by authenticated role-based access before production release.
- `RA-014`, `RA-015`, and `RA-016`: Residual-risk closure depends on formal freeze-lane evidence promotion; current documented residuals remain provisional until those artifacts are executed and linked.

Included source: Docs/Quality/SoftwareRequirementsSpecification.md

Software Requirements Specification (SRS)

Status: Final draft prepared for handoff (pending review)
Version: 0.9
Owner: BionicLoop engineering
Prepared by: BionicLoop engineering Reviewer: _____
Approver: _____
Decision date: _____
Effective date: _____
Baseline freeze SHA: _____
Scope: Closed-loop investigational runtime, device integration, and safety-critical UI behavior Last updated: 2026-04-06 15:05 EDT

Revision History

Version	Date	Author	Summary of Changes
0.1	2026-03-25	Engineering	Initial controlled draft baseline
0.9	2026-04-06	BionicLoop engineering	Added handoff-ready document-control metadata and software-handoff disposition language

Requirement Format

Each requirement uses a stable ID and a testable `shall` statement.

Runtime and Cadence

- SRS-RUN-001: The runtime shall maintain 5-minute algorithm cadence anchored to first successful step execution time.
- SRS-RUN-002: The runtime shall compute expected step index from anchored schedule and skip duplicate step execution for the same index.
- SRS-RUN-003: The runtime shall execute loop work only on configured wake causes (current policy: `cgmUpdate`, `bgCheck`, `mealAnnounce`, and guarded `pumpReconnect` fallback paths).
- SRS-RUN-004: After the first successful anchored step exists, the runtime shall permit a guarded pump-reconnect fallback wake to execute the current due step when no accepted CGM receipt newer than the approved CGM freshness limit exists and normal execution safety gates pass.
- SRS-RUN-005: Pump-reconnect fallback shall not execute step `0`, shall not re-anchor cadence, and shall not replay multiple missed slots; each reconnect-triggered execution may service at most the current due step.

Team review notes:

- SRS-RUN-003 reflects the currently implemented wake-cause policy.
- Guarded reconnect fallback is implemented as a secondary wake path from pump-status recovery, not raw BLE connect.
- Current reconnect-fallback gate is `accepted CGM receipt age > 5 minutes`, matching the present CGM freshness limit for algorithm-use eligibility.

CGM Input Handling

- SRS-CGM-001: The runtime shall map CGM values outside `39...401 mg/dL` to unavailable input (`-1`) before algorithm invocation.
- SRS-CGM-002: Step `0` shall require a fresh in-range CGM input.
- SRS-CGM-003: Step `>0` shall be allowed to run with unavailable CGM input (`-1`) when fresh in-range CGM is not available.
- SRS-CGM-004: The UI shall show loop reason/state when step execution is blocked for missing fresh CGM at step `0`.
- SRS-CGM-005: When the loop is armed and no successful algorithm step has completed for longer than the approved interruption threshold, the app shall detect that loop stepping is stalled and expose that condition to alert/status logic.

Team review notes:

- SRS-CGM-005 is implemented with a `15`-minute threshold based on the most recent successful step time; before the first successful step exists, the threshold is measured from session start / first-step wait state.
- The interruption condition is recomputed on foreground from persisted runtime/session state and is distinct from the protocol Dexcom Bluetooth interruption alert (`> 2 hours`).

Fingerstick BG Input Handling

- SRS-BG-001: The app shall provide a manual BG entry flow with explicit `Submit` and `Cancel` actions, enforce accepted input range `20...600 mg/dL`, and capture a submission timestamp at user confirmation time.
- SRS-BG-002: Manual BG submit shall trigger a dedicated runtime wake cause (`bgCheck`) with no future-slot borrowing and shall create/update a single pending BG candidate for the next eligible algorithm step.
- SRS-BG-003: If submit occurs after the current due step has already executed, the pending BG candidate shall roll forward to the immediate next step (not discarded immediately).
- SRS-BG-004: During `bgCheck`, manual BG shall map to algorithm BG input (`BGval`) while CGM input mapping policy remains independent (`CGM` may be unavailable `-1`).
- SRS-BG-005: BG submission shall not override pump command-application safety gates; when pump status is unavailable/unknown, runtime may execute algorithm step but shall block command application.
- SRS-BG-006: Runtime shall reject stale manual BG submissions using a defined freshness window and provide an explicit user-visible reason.
- SRS-BG-007: Telemetry shall record BG source (`manualBG`), submitted value, submit timestamp, and execution result for each `bgCheck` attempt.
- SRS-BG-008: Step-0 BG rescue is deferred from the current software handoff baseline. If a future approved baseline enables step `0` BG rescue, it shall permit execution only when fresh in-range CGM is unavailable and a valid fresh manual BG is present.
- SRS-BG-009: Pending BG candidate shall be valid for one step only and must be discarded if not consumed on that immediate next step.
- SRS-BG-010: If a new manual BG is submitted while a pending BG candidate exists and has not yet been consumed, the newer submission shall replace the older candidate.
- SRS-BG-011: Manual BG submit shall not execute `bgCheck` when loop runtime is disarmed (`Algorithm Off` state).
- SRS-BG-012: Until step-0 BG rescue is explicitly approved, manual BG submit shall be rejected before the first successful anchored step exists (`firstSuccessfulStepAt == nil`).

Team review notes:

- SRS-BG-008 is deferred from the current software handoff baseline; current accepted behavior remains CGM-only at step `0` via SRS-BG-012.
- Manual BG accepted range is currently set to `20...600 mg/dL`.
- Manual BG freshness duration remains a required configuration value that must be explicitly approved and versioned.
- SRS-BG-011 is implemented in app runtime to prevent loop-off execution from manual BG UI.
- SRS-BG-012 is currently enforced in runtime; step-0 BG rescue remains disabled by default.

Pump Handling

- SRS-PUMP-001: On pump-status refresh failure or unknown state, runtime shall continue algorithm execution with unavailable pump input fields and block command application.
- SRS-PUMP-002: In closed-loop mode, manual bolus command paths shall not be exposed.
- SRS-PUMP-003: Dosing command handling shall respect DASH delivery constraints (including minimum practical delivery quantum).
- SRS-PUMP-004: Home pod status tile shall update to reflect actual connect/disconnect state transitions without requiring user entry into pump settings.

- SRS-PUMP-005: While pump delivery is in progress, Home and meal-availability pump state shall auto-refresh without requiring navigation into pump settings.

Meal Announce

- SRS-MEAL-001: Meal announce shall only use borrow execution when request time is before the next scheduled slot and within borrow window (<= 2 future 5-minute slots).
- SRS-MEAL-002: Meal announce shall be blocked unless an active pod is present, pump status is known, and pump is not delivering. Unavailable-reason precedence shall report noPump when no active pod is present, and reserve signalLoss for active-pod unknown-status conditions.
- SRS-MEAL-003: Meal announce UI shall provide explicit unavailable reason and next retry time when applicable.
- SRS-MEAL-004: If meal announce is requested after the next scheduled slot is already due/missed, runtime shall execute meal input on the current due step (expectedStep) instead of rejecting solely due to late borrow timing.
- SRS-MEAL-005: Meal announce shall be blocked until the loop establishes the first successful anchored step (firstSuccessfulStepAt), even when pump and CGM are otherwise available.
- SRS-MEAL-006: If the meal announce composer remains visible while runtime availability changes, the app shall revalidate meal availability on foreground refresh and immediately before submit; unavailable state shall block dispatch and present current blocked messaging instead of using stale availability.
- SRS-MEAL-007: Meal announce submit shall not be presented or logged as success until runtime/coordinator result is known. Executed success shall only be shown after runtime completion, and blocked/rejected outcomes shall produce explicit user-visible result messaging. Uncertain command outcomes shall produce explicit blocked guidance rather than optimistic success.
- SRS-MEAL-008: Once a meal announce request is accepted for execution, the app shall persist pending meal-request state across relaunch until the request is explicitly cleared, reconciled against executed step history, or reset by session reset/disarm. Persisted state shall include the concrete target step accepted by runtime and the correlated meal flow identifier needed for lifecycle telemetry closure.
- SRS-MEAL-009: While a persisted prior meal request remains pending, additional meal announces shall be blocked after relaunch and the user shall be informed that a meal request is already in progress. If the prior request has an uncertain command outcome, duplicate meal entry shall remain blocked until reconciliation or session reset/disarm, and the user shall be informed that the prior meal delivery outcome is uncertain.
- SRS-MEAL-010: If a competing trigger or slot conflict prevents meal execution on the originally observed slot, runtime shall not silently lose or reinterpret the meal request. The app shall block the submit with explicit slot-conflict/retry guidance rather than silently reassigning the meal to a different borrowed step.
- SRS-MEAL-011: If meal announce is requested while pump bolus delivery is already in progress, the app shall present a destructive cancel-delivery path on Home that can cancel the current bolus, report the actual insulin delivered before cancellation in Home's standard alert-summary region, and preserve that delivered amount for subsequent algorithm-step accounting when the operator later reopens meal announce.

Team review note:

- SRS-MEAL-004 is accepted for the current software handoff baseline as the implemented late-submit meal behavior.
- Meal tooLate retry messaging shall point to the immediate next due-step time boundary (not an added fixed 5-minute delay).
- SRS-MEAL-007 through SRS-MEAL-011 are now implemented in the app baseline through runtime-result-gated meal success handling, blocked/rejected submit messaging, persisted pending meal-request state + flow correlation, uncertain-outcome duplicate blocking, relaunch reconciliation against executed-step/pump-delivery evidence, explicit slot-conflict blocking when another step advances after the meal flow was opened, a destructive cancel-delivery flow when meal entry is attempted during active bolus delivery, and correlated accepted/resolved lifecycle telemetry closure.

State Persistence and Recovery

- SRS-STATE-001: The app shall persist algorithm state and runtime cadence state across relaunch.
- SRS-STATE-002: Reset Algo shall clear persisted algorithm state, persisted cadence state, and step timeline/session history to start a fresh session.
- SRS-STATE-003: Pump and CGM manager state shall persist across relaunch to avoid forced re-pairing.

Algorithm Verification Controls

- SRS-ALG-001: The Algo2015 implementation and bridge path shall maintain a deterministic golden-vector regression suite with fixed inputs and expected outputs under version control.
- SRS-ALG-002: Algo2015 verification shall include structural coverage reporting for C bridge and C++ algorithm sources, with predefined coverage targets and documented rationale for uncovered regions.
- SRS-ALG-003: Bridge contract behavior for null inputs, state reset edge conditions, sentinel mappings, and subject-id boundary handling shall be explicitly verified.
- SRS-ALG-004: Algorithm state continuity shall be verified across multi-step execution, persistence/reload, and reset-to-fresh session boundaries.
- SRS-ALG-005: Any change to pregnancy configuration inputs (target, meal upfront, TMAX) shall include differential algorithm-output evidence against baseline replay vectors.
- SRS-ALG-006: Formal Algo2015 verification runs shall include a static-analysis lane for algorithm/bridge sources with logged findings status and run linkage metadata.
- SRS-ALG-007: MISRA assessment for the host-side investigational Algo2015 lane shall be risk-based and conditional: each formal run shall include either linked MISRA evidence with signed deviation handling (if applicable) or an explicit not-applicable decision rationale with compensating controls.

Telemetry and Traceability

- SRS-LOG-001: For each executed step, telemetry shall include an explicit step execution timestamp (step_executed_at), algorithm input snapshot, output snapshot, and pump command result.

- SRS-LOG-002: Telemetry storage shall support export with stable column definitions.
- SRS-LOG-003: Telemetry persistence shall not block UI responsiveness (no heavy synchronous file generation on main actor).
- SRS-LOG-004: Debug builds shall provide a local cloud-log upload threshold control with levels `Error`, `Warning`, `Info`, `Debug`; default threshold shall remain `Error`, and threshold evaluation shall be inclusive (`selected level` and higher severities).
- SRS-LOG-005: Clinical settings save flow shall emit `ui.critical` telemetry for `state_viewed`, `submit`, `cancel`, and `blocked` outcomes with stable `screen_id`/`element_id` values and old/new field details for changed clinical parameters.
- SRS-LOG-006: `app.lifecycle.launched` and `app.lifecycle.foregrounded` telemetry payloads shall include timezone and clock-check context fields: `device_timezone_id`, `device_utc_offset_seconds`, `clock_check_result`, and when available `clock_check_skew_seconds`, `clock_check_rtt_ms`, `clock_check_at_utc`.
- SRS-LOG-007: Meal-announce telemetry shall expose a correlated request lifecycle so cloud reconstruction does not rely on optimistic submit success alone. Implemented baseline includes `submitted`, `accepted`, `success`, `blocked`, `uncertain`, and `resolved` transitions keyed by meal `flow_id`, with replay state persisted until emission so `accepted` and `resolved` are not lost across terminate/relaunch windows; `resolved` is emitted for immediate completion, relaunch/pump-evidence reconciliation after uncertainty, or session-reset clear. Loop command telemetry shall preserve command outcome semantics, including uncertain delivery, rather than collapsing all non-success outcomes into generic blocked state.
- SRS-LOG-008: Target-selection telemetry shall capture clinician-controlled regular-settings target profile changes and participant target-change approval-capture events with stable `ui.critical` event identifiers and required detail fields (`target_range_profile`, requested/applied target, approval metadata, and blocked/cancelled reasons where applicable).

UI Safety and Status

- SRS-UI-001: Home loop-status card shall use deterministic state precedence: `No Session` > `No Pod` > `No CGM` > `Ready` > age-based states, where age-based states are derived from cadence due-time (`firstSuccessfulStepAt + (lastExecutedStep + 1) * 300s`) rather than raw `lastSuccessfulRunAt`.
- SRS-UI-002: Home availability and status messaging shall align with runtime behavior (no false available state for blocked actions).
- SRS-UI-003: Initial CGM and Pod startup/setup modals shall provide an explicit `Cancel` action that dismisses the modal without forcing entry into settings.
- SRS-UI-004: If the meal announcement composer is visible and the app transitions to background, the composer shall auto-dismiss/cancel so meal intent does not persist across lifecycle interruption.
- SRS-UI-005: If the latest CGM reading is older than `11` minutes, or if the latest CGM reading is present but not reliable (`hasReliableGlucose == false`), CGM display surfaces shall mask the reading as `--` and shall not display a trend arrow.
- SRS-UI-006: The app shall perform UTC drift checks on launch, on timezone/significant-time-change events, and on foreground when the last successful check is older than 24 hours. If absolute clock skew exceeds 600 seconds, the app shall present a non-blocking actionable warning no more than once per 24 hours; unavailable UTC checks shall not show warning spam.
- SRS-UI-007: CGM display surfaces shall render boundary readings as textual values `LOW` (`<=39 mg/dL`) and `HIGH` (`>=401 mg/dL`) instead of numeric mg/dL values.
- SRS-UI-008: Home CGM chart y-axis scaling shall use bounded stepped maxima (`300`, `350`, `400 mg/dL`) based on displayed data peak, and chart/scrub rendering shall keep boundary-value interpretation consistent with `SRS-UI-007`.

Clinical Settings and Pregnancy Configuration

- SRS-CLIN-001: The app shall provide a dedicated `Clinical Settings` area for clinician-only configuration and control actions.
- SRS-CLIN-002: Entry to `Clinical Settings` shall be gated by an investigational passcode control. The current software handoff baseline uses configured passcode value `020508`; production release requires replacement with authenticated role-based access.
- SRS-CLIN-003: `Subject ID`, `Weight`, `Start Algo`, and `Reset Algo` controls shall be hosted in `Clinical Settings` and not exposed in non-clinician settings sections.
- SRS-CLIN-004: Target selection shall support `90`, `100`, `110`, `120`, and `130 mg/dL` only.
- SRS-CLIN-005: Meal upfront percentage selection shall support exactly `75%` and `90%`.
- SRS-CLIN-006: `TMAX` selection shall support `40...70` minutes inclusive, in increments of `5`.
- SRS-CLIN-007: Clinical settings values shall be persisted with versioned defaults/migration so runtime behavior is deterministic across relaunch and app update.
- SRS-CLIN-008: Relocating `Start Algo` and `Reset Algo` into `Clinical Settings` shall not change their runtime behavior semantics.
- SRS-CLIN-009: Clinical Settings shall provide a clinician-controlled target-access profile for participant-facing settings with exactly two values: `Pregnancy` and `Standard`.
- SRS-CLIN-010: Participant-facing settings and the clinician target selector shall expose only the target options enabled by the current target-access profile: `Pregnancy -> 90, 100, 110 mg/dL`; `Standard -> 110, 120, 130 mg/dL`.
- SRS-CLIN-011: Participant target changes in regular settings shall require approval capture before apply. The app shall block the change unless the operator records the approving clinical staff member name and an approximate approval time.
- SRS-CLIN-012: If the clinician changes the target-access profile while the draft target is outside the newly selected profile subset, the app shall normalize the draft target to the nearest allowed value before save rather than preserving an out-of-profile target.
- SRS-VAL-001: Weight entry shall be pounds on UI, integer-only, stored as kilograms for runtime/algorithm use; in current UX scope this entry is clinician-gated.

Team review notes:

- SRS-CLIN-002 is accepted for the current software handoff baseline as an investigational control and must be replaced by authenticated, role-based gating before production release.
- Pregnancy configuration control set is sourced from `Docs/Requirements/AlgorithmChanges_PregnancyConfig.md`.
- Implemented baseline now includes a clinician-controlled participant target-access profile (`Pregnancy` / `Standard`), participant target-change approval capture in regular settings, and a clinician target selector that is constrained to the currently selected profile subset.

User Alerts and Escalation

- SRS-ALERT-001: The app shall normalize alerts from pump (OmniBLE), CGM (G7SensorKit), algorithm/runtime, and app safety policies into a single alert domain model.
- SRS-ALERT-002: Each alert shall include source, severity, timestamp, user-facing message, and recommended action metadata.
- SRS-ALERT-003: Alert presentation shall apply severity-based channels and deterministic precedence so critical alerts are never hidden by lower-severity alerts.
- SRS-ALERT-004: The app shall debounce or coalesce transient reconnect/noise events to reduce false-alarm UX while preserving true fault visibility.
- SRS-ALERT-005: Alert life-cycle behavior shall define clear/acknowledge rules per alert type (auto-clear vs explicit acknowledge).
- SRS-ALERT-006: Protocol-required alert conditions and response wording shall be represented in app behavior and traceable to verification evidence.
- SRS-ALERT-007: For high-priority non-CGM alerts (actionable/safetyCritical), the app shall issue background local notifications with de-duplication and per-severity cooldown to avoid alert spam.
- SRS-ALERT-008: The app shall provide an in-app Alert Center showing both active alerts and recently cleared alerts.
- SRS-ALERT-009: Pump and CGM alert issue/retract lifecycle state shall persist across app relaunch, including lookup of unretracted alerts and restoration of active alert visibility.
- SRS-ALERT-010: Alerts with time-sensitive countdown wording shall refresh displayed remaining time at least once per minute while active so user-facing timing remains current.
- SRS-ALERT-011: Home shall present active alerts as a prioritized vertical carousel (severity first, then recency) with explicit multiplicity indication when more than one alert is active.
- SRS-ALERT-012: Safety-critical pump alerts (ALERT-PUMP-FAULT, ALERT-PUMP-INCOMPATIBLE) shall not auto-clear solely due to `hasActivePod == false`; they shall clear only on explicit lifecycle retraction/recovery or explicit acknowledge workflow.
- SRS-ALERT-013: When armed-loop execution is stalled beyond the approved interruption threshold, the app shall issue an actionable interruption alert distinct from G7 unavailable/failed/expired alerts and clear it once successful step execution resumes or the loop is disarmed.
- SRS-ALERT-014: The interruption alert shall use broader `Algorithm Stepping Interrupted` semantics, triggered when the armed loop has not achieved a successful algorithm step for longer than the approved interruption threshold, regardless of whether the immediate blocker is missing CGM, missing pod, pump signal loss, or another runtime execution gate.
- SRS-ALERT-015: BionicLoop CGM availability/failure alerts shall be in-app informational status only on Home / Alert Center, shall not require explicit acknowledge, and shall not schedule background local notifications. The FDA-cleared Dexcom application shall be treated as the source of truth for CGM alarming and shall be configured accordingly in study use.
- SRS-ALERT-016: When a trustworthy G7 reading is `<55 mg/dL`, the app shall issue an in-app-only urgent-low review alert on Home / Alert Center, shall allow clinician acknowledge/review capture, shall not schedule background local notifications, and shall auto-clear the active alert state on a later trustworthy G7 reading `>=55 mg/dL`. This alert shall be app-derived from CGM data and shall not claim Dexcom-app acknowledgement.

Alert evolution notes:

- Current implemented baseline is ALERT-ALGORITHM-STEPPING-INTERRUPTED.
- The alert triggers on absent successful step execution rather than CGM-specific receipt loss alone.
- `Algorithm Stepping Interrupted` preserves root-cause visibility so stronger underlying pump alerts remain available and prioritized while CGM state remains visible as separate informational context.

Alert source inventory and normalized mapping baseline:

- Alert Inventory and Mapping

Security and Privacy

- SRS-SEC-001: Long-term telemetry architecture shall support secure cloud upload as primary transport, with local CSV retained only as a development-only path.
- SRS-SEC-002: Telemetry export and storage paths shall be controlled to minimize PHI/PII exposure.
- SRS-SEC-003: Protected cloud API access is deferred from the current software handoff package. If cloud-backed protected APIs are included in a future accepted baseline, they shall require authenticated user identity managed by Cognito before access is granted.
- SRS-SEC-004: Multi-provider onboarding policy is deferred from the current software handoff package. If included in a future accepted baseline, onboarding shall define the allowed `Sign in with Apple`, `Google`, and `Email` paths explicitly.
- SRS-SEC-005: Cloud authorization-role enforcement is deferred from the current software handoff package. If included in a future accepted baseline, authorization shall enforce least-privilege role and scope checks for telemetry and dashboard access.
- SRS-SEC-006: Protected-cloud authentication failure handling is deferred from the current software handoff package. If included in a future accepted baseline, authentication and session failures shall fail closed for protected cloud operations while preserving safe local app behavior.
- SRS-SEC-007: Cognito password-recovery workflow is deferred from the current software handoff package. If included in a future accepted baseline, email/password onboarding shall provide reset-code request and confirmation with explicit success/failure feedback.
- SRS-SEC-008: Launch session-restore behavior is deferred from the current software handoff package. If included in a future accepted baseline, authenticated persisted sessions shall attempt secure restore/refresh without unnecessary manual re-login when restore succeeds.
- SRS-SEC-009: Auth-failure Home-bypass continuity behavior is deferred from the current software handoff package. If included in a future accepted baseline, the app shall support explicit local Home bypass with persistent remote-monitoring login-recovery messaging when unauthenticated local loop state remains active.

Team review notes:

- SRS-SEC-003..009 are deferred from the current software handoff package and require explicit scope re-entry before closure is claimed.
- Existing development implementation may cover portions of SRS-SEC-007..009, but that implementation is not being claimed as closed software-handoff scope in this package.
- Role model (clinical/engineering/admin) and token lifecycle policy are pending.

Traceability Source Mapping

Primary upstream references:

- `Docs/Requirements/Requirements.md`
- `Docs/Requirements/RiskAnalysis.md`
- `Docs/Analysis/Marjorie_AlgorithmIO_GapAnalysis.md`
- `Docs/Architecture/Architecture.md`
- `Docs/Requirements/AlgorithmChanges_PregnancyConfig.md`

Included source: Docs/Quality/SoftwareDesignDescription.md

Software Design Description (SDD)

Status: Final draft prepared for handoff (pending review)
Version: 0.9
Owner: BionicLoop engineering Prepared by: BionicLoop engineering Reviewer: _____
Approver: _____
Decision date: _____
Effective date: _____
Baseline freeze SHA: _____
Last updated: 2026-04-06 16:35 EDT

Revision History

Version	Date	Author	Summary of Changes
0.1	2026-04-05	Engineering	Initial controlled design draft
0.9	2026-04-06	BionicLoop engineering	Added handoff-ready document-control metadata and software-handoff disposition language

1. Design Intent

This design implements a CGM-triggered closed-loop runtime with:

- deterministic cadence state
- degraded execution behavior when inputs are unavailable
- explicit gating for pump command application
- traceable per-step telemetry

2. Major Components

- `SDD-APP-001`: `LoopRuntimeEngine` (`BionicLoop/Runtime/LoopRuntimeEngine.swift`) handles wake handling, availability logic, session control, and telemetry orchestration.
- `SDD-APP-002`: `LoopSessionStore` owns app-layer runtime persistence reads/writes for loop armed state and runtime state snapshots.
- `SDD-APP-003`: `LoopWorkScheduler` owns CGM-timestamp trigger dedupe and arm/reset semantics for runtime wake dispatch.
- `SDD-APP-004`: `LoopAlertMediator` owns pump signal-loss tracking and report gating in app-layer state.
- `SDD-APP-005`: `LoopTelemetryWriter` owns app-layer telemetry-store writes and reconciliation calls.
- `SDD-APP-006`: `LoopRuntimeWorkExecutor` owns coordinator doWork execution snapshot sequencing (latest reading capture, operation execution, completion timestamp, state snapshot).
- `SDD-APP-007`: `DeviceClockSyncMonitor` (`BionicLoop/App/AuthSessionNetworking.swift`) owns UTC midpoint drift checks (`GET /v1/time/utc`), retry policy, 24-hour foreground check gating, timezone/significant-time-change trigger handling, and lifecycle telemetry context projection.
- `SDD-AUTH-001`: App auth-session boundary is split into `AppAuthSessionManager` (secure token persistence, restore, refresh, sign-out clear) and `AuthenticatedAPIClient` (bearer-auth request composition + one-time refresh retry on unauthorized response), implemented in `BionicLoop/App/AuthSessionNetworking.swift`. Recovered/refreshed token sets are validated for telemetry-ingest scope before persistence.
- `SDD-AUTH-002`: Email/password recovery boundary is implemented in `CognitoPasswordRecoveryService` (`ForgotPassword`, `ConfirmForgotPassword`) plus unauthenticated route/state wiring (`.forgotPassword`) and `NoAuthForgotPasswordView` for reset-code request, confirmation, resend, and user-feedback messaging.
- `SDD-CORE-001`: `LoopRuntimeCoordinator` (`BionicLoopCore/.../Runtime/LoopRuntimeCoordinator.swift`) handles due-step logic, input assembly, algorithm invocation, and command application.
- `SDD-ALG-001`: `RealBUDosingAlgorithm` + `Algo2015Bridge` map runtime input into C bridge structs and receive output/state.
- `SDD-PUMP-001`: `PumpServiceAdapter` + `PumpStatusObserver` handle status refresh, command execution, delivery reconciliation, home-status projection, and auto-polling while `deliveryState == delivering`.
- `SDD-CGM-001`: `G7ViewModel` + CGM subsystem adapters provide CGM state ingestion and wake trigger source; telemetry `cgm.state.changed` is emitted from refreshed/current callback state to avoid stale `has_sensor` transitions, and `cgm.connection.changed` derives `status_text` from refreshed lifecycle state before emit.
- `SDD-BG-001`: Manual BG entry path spans Home/UI entry, runtime `bgCheck` wake dispatch, single-pending-candidate BG state, next-eligible-step consumption logic, and telemetry source tagging (`manualBG`).
- `SDD-CLIN-001`: Clinical settings boundary spans passcode-gated settings presentation, clinician-only control hosting (`Subject ID`, `Weight`, `Start Algo`, `Reset Algo`), runtime-facing pregnancy configuration projection (`target`, `meal upfront`, `TMAX`), a clinician-controlled participant

target-access profile (`Pregnancy` / `Standard`), and the participant-facing target-change approval-capture sheet hosted in general settings.
- `SDD-LOG-001`: `LoopTelemetryStore` stores per-step records for UI + export, and `CloudTelemetryReporter` emits authenticated telemetry envelopes with correlation metadata (`event_id`, `session_id`, `subject_id`) plus `auth_user_sub` derived from Cognito ID-token `sub` (fallback `UNSET` only when unavailable); loop-step payloads carry explicit execution timestamp `step_executed_at` (execution semantics) in addition to envelope `created_at` (ingest creation time). Loop command telemetry mapping is schema-stable: `loop.command.blocked` is emitted only when a recommendation existed but was blocked, while `pump.command.result` remains pump-result schema only and emits on delivery-result deltas only (no unchanged refresh re-emission). Outbox flushing is loss-avoidant: if flush is requested while a flush is active, a deferred follow-up flush is queued and executed immediately after the active pass completes (with full-flush escalation when any deferred request requires non-high-priority delivery). Client cloud-log threshold resolution uses precedence `remote override (active, TTL-bound) -> local threshold -> default error`; local threshold is exposed in debug builds via `HomeSettingsView` and persisted through `CloudLogUploadPolicy`.
- `SDD-SIM-001`: Deterministic simulation harness (medium-fidelity) uses protocol-conformant mock CGM/pump services plus a virtual clock/scenario runner to exercise runtime logic without BLE transport dependency.
- `SDD-ALERT-001`: Alert normalization/presentation layer is implemented as an app-level normalized model (`AppAlert*` types + `AppAlertCenter`) with deterministic precedence (`severity`, then recency), dedupe keys, and debounced signal-loss handling. Current live UI presentation includes a Home vertical active-alert carousel (prioritized and multiplicity-aware), plus a Home bell entry to Alert Center (active + recently cleared), with secondary access from Settings. G1 source inventory/mapping baseline is documented in Alert Inventory and Mapping.
- `SDD-QA-001`: Algo2015 verification infrastructure uses deterministic replay vectors plus instrumented coverage collection, mandatory static-analysis lane execution on formal runs, and conditional MISRA policy linkage (report+deviations or explicit N/A rationale) to generate reproducible `TV-ALG-*` evidence artifacts for IDE submission, as specified in Algo2015 Verification Plan.

3. Runtime Sequence (Nominal)

1. New CGM timestamp arrives.
2. `LoopRuntimeEngine` triggers coordinator `doWork`.
3. Coordinator computes expected step from anchor + interval.
4. Coordinator builds algorithm input:
    - CGM (or unavailable `-1` by policy).
    - Pump fields (or unavailable values if pump not available).
5. Algorithm returns recommendation and updated internal state blob.
6. If pump status is available and recommendation passes gates, command is attempted.
7. Telemetry record is persisted with input/output/command fields.
8. Pump status refresh reconciles delivered/requested units.

4. Key Design Policies

- `SDD-POL-001` (Cadence): anchored 5-minute slot model.
- `SDD-POL-002` (CGM): step `0` strict gate, step `>0` degraded allowed.
- `SDD-POL-003` (Pump): unavailable pump permits algorithm step but blocks command application.
- `SDD-POL-004` (Meal announce): explicit pump-ready gating (`active pod present`, `known`, and `not delivering`), established first-step cadence anchor requirement, borrow-window gating for pre-due execution, and current-due-step execution when request arrives after slot is due/missed. Unavailable-reason precedence reports `noPump` before `signalLoss` when there is no active pod. Home/meal presentation logic revalidates meal availability when the app refreshes to `active` while the composer is visible and again immediately before submit, closing stale-composer paths by replacing them with the current unavailable presentation instead of dispatching a now-invalid meal request. This is the accepted current software handoff baseline for late-submit meal behavior.
- `SDD-POL-005` (Reset): full session reset clears persisted runtime + algorithm + timeline session data.
- `SDD-POL-006` (Home status): state precedence is deterministic and runtime-derived; age-based states use cadence phase (`nextDueAt` from anchor + step index) so borrowed meal steps do not falsely age status before the next due boundary. Current thresholds: `Active` through `nextDueAt + 2m`, `Aging` through `nextDueAt + 15m`, then `Stale`.
- `SDD-POL-007` (Modal setup cancellation): startup/setup modals for CGM and Pod provide direct dismiss behavior via explicit `Cancel`, including persisted-manager/no-active-pod Pod setup.
- `SDD-POL-009` (Meal composer lifecycle safety): if app scene transitions to background while meal composer is presented, the composer is auto-cancelled/dismissed.
- `SDD-POL-010` (BG check): manual BG submit triggers `bgCheck` with no borrowing, uses a single pending BG candidate (replace-on-new submit), applies candidate to the next eligible step, and expires candidate if not consumed on that immediate next step.
- `SDD-POL-011` (BG safety gate): manual BG submit path checks loop-armed state in `LoopRuntimeEngine` and shall not dispatch `bgCheck` while disarmed.
- `SDD-POL-012` (BG step-0 guard): until step-0 BG rescue is approved, manual BG submit is rejected if `firstSuccessfulStepAt` is not established.
- `SDD-POL-013` (Clinical settings policy): clinician configuration access is passcode-gated using the current investigational baseline value `020508`, control values are validated to allowed ranges (`target 90...130 step 10`, `meal upfront 75/90`, `TMAX 40...70 step 5`), and mapped into algorithm/runtime configuration without changing existing start/reset semantics. `HomeSettingsView` persists a target-access profile (`Pregnancy`, `Standard`) alongside the clinical config; participant-facing settings derive their available target buttons from that profile (`Pregnancy -> 90/100/110`, `Standard -> 110/120/130`). The clinician target selector now uses that same profile subset and normalizes any inherited out-of-profile draft target to the nearest allowed value when the profile changes. When a previously conflicting subject identifier is corrected and the save/claim flow succeeds, `HomeSettingsView` explicitly retracts the app-level `ALERT-SUBJECT-ID-CONFLICT` alert so operator-facing alert state matches the resolved persisted configuration. Separately, Home now performs a throttled silent re-claim check for the currently persisted subject identifier when the conflict alert is still active and the operator launches, foregrounds, or reopens settings; if the backend claim now succeeds, Home retracts the stale conflict alert without requiring a settings edit/resave round-trip. This investigational control remains subject to replacement by authenticated role-based access before production release.
- `SDD-POL-014` (CGM display masking for stale/unreliable data): `G7ViewModel` applies a display-staleness cutoff of `11` minutes and checks `hasReliableGlucose` on the latest G7 reading. If stale or unreliable, display accessors emit masked glucose text (`-- mg/dL` / `--`), suppress trend-arrow rendering, and do not append unreliable points into chart-history persistence.

- SDD-POL-015 (Auth/runtime decoupling): authentication/session state shall not stop or reset local loop runtime state. Auth gates cloud/protected operations and onboarding surfaces, while therapy runtime continuity is preserved unless user explicitly performs runtime reset/start-stop actions.
- SDD-POL-016 (Auth re-entry with continuity fallback): app launch attempts silent token-session restore/refresh when persisted auth state is authenticated; on restore success, authenticated UX remains uninterrupted. If auth state changes to unauthenticated while restore is in flight (for example explicit user sign-out), launch reconciliation preserves the current unauthenticated state and does not auto-reauthenticate. When restore fails (or user is unauthenticated) and local runtime indicates active algorithm state, login UX exposes explicit Home bypass. In bypass mode, Home shows a persistent auth-recovery alert (ALERT-AUTH-LOGIN-REQUIRED) with direct Log In action until re-authentication completes.
- SDD-POL-017 (Clinical/settings telemetry): Clinical Settings save-review flow emits ui.critical telemetry with stable event/action mapping: state_viewed on review presentation, submit on Save, cancel on review dismissal, and blocked on invalid/locked/no-change save attempts. Save telemetry includes changed-field list and old/new clinical values (subject, weight_lbs, target_mgdl, target_range_profile, meal_upfront_percent, tmax_minutes). Participant target-change approval capture emits its own stable ui.critical lifecycle (state_viewed, submit, cancel, blocked) with requested target, current target, target-access profile, approving staff name, and approval timestamp.
- SDD-POL-018 (Clock-sync safety telemetry): App lifecycle telemetry (app.lifecycle.launched, app.lifecycle.foregrounded) includes timezone + UTC-check context fields (device_timezone_id, device_utc_offset_seconds, clock_check_result, optional skew/rtt/timestamp). UTC checks run on launch, timezone/significant-time-change notifications, and foreground only when last successful check is older than 24 hours. If absolute skew exceeds 600 seconds, app raises actionable ALERT-APP-CLOCK-SKEW warning no more than once per 24 hours; unavailable checks do not show user warnings.
- SDD-POL-019 (CGM boundary display + chart scaling): CGM value rendering maps <=39 mg/dL to LOW and >=401 mg/dL to HIGH across Home/G7 primary value surfaces and scrub value presentation; Home inline CGM chart uses bounded stepped maxima (300, 350, 400 mg/dL) based on observed peak values so out-of-range excursions remain visible without unbounded axis drift.
- SDD-POL-020 (algorithm stepping interruption monitoring): LoopRuntimeEngine monitors for stalled loop execution while armed using persisted runtime/session state rather than CGM-receipt time alone. The interruption deadline is computed from lastSuccessfulRunAt + 15 minutes; if no successful step exists yet, the baseline is session start / first-step wait state. AppAlertCenter clears any prior pending interruption notification and schedules a replacement local-notification deadline at the current interruption deadline; if the deadline passes while still current, it raises actionable in-app alert state. Monitoring is recomputed on app foreground and refreshed after each successful step. The alert clears when successful stepping resumes or when the loop is disarmed/reset.
- SDD-POL-021 (reconnect fallback execution): current cadence remains anchored to firstSuccessfulStepAt with the existing early execution lead (27 seconds). Pump reconnect is treated as a secondary wake source rather than a cadence source: app runtime listens for pump-status recovery (not raw BLE connect), only after an anchored session exists, only for step >0, only when the accepted CGM receipt is older than the approved freshness limit (>5 minutes), and only when the current due slot has not already executed. The reconnect path may execute only the current due step, shall not replay missed slots, shall not re-anchor the schedule, and shall remain suppressed once fresh CGM receipts resume so CGM retains priority as the primary loop trigger.
- SDD-POL-022 (algorithm stepping interrupted alert policy): the normalized runtime interruption alert is ALERT-ALGORITHM-STEPPING-INTERRUPTED, keyed to lack of successful algorithm execution rather than lack of CGM receipt alone. Trigger basis is loop armed plus no successful step for >15 minutes, measured from the most recent successful executed step or, if none exists yet, from session arm time / first-step wait state. The alert issues even when the immediate blocker is not CGM-specific (for example no active pod, pump signal loss, or another execution gate), clears on the next successful step or loop disarm/reset, and exposes blocker/root-cause detail without masking stronger source-native CGM/pump alerts.
- SDD-POL-023 (meal-request durability baseline): LoopRuntimeState persists pending meal-request state across relaunch using submit time, the concrete execution step accepted by LoopRuntimeCoordinator, a pending-state discriminator (awaitingExecution, uncertainCommandOutcome), and the correlated meal flow identifier. On startup/foreground availability checks, runtime reconciles that state against persisted step history (lastExecutedStep) and, for uncertain outcomes, against pump-delivery evidence (lastDelivery.requestTimeStep) after pump attachment. While the pending request remains unresolved, mealAnnouncementAvailability() returns requestInProgress or uncertainPreviousMeal and duplicate meal entry is blocked until reconciliation or session reset/disarm.
- SDD-POL-024 (meal outcome handling): Home/UI and runtime telemetry split meal flow into pre-submit interaction plus outcome-driven submitted, accepted, success, blocked, uncertain, and resolved paths. HomeView keeps the composer in a submitting state while LoopRuntimeEngine.announceMeal() awaits coordinator completion, dismisses only on executed success, presents explicit blocked messaging when runtime returns a blocked/rejected outcome, and presents explicit uncertain-delivery guidance when pump command outcome is unresolved. Home captures the lastExecutedStep observed when the meal composer opens and passes that snapshot with submit; LoopRuntimeCoordinator compares it against current cadence state and returns mealSlotConflict if another step has already advanced, so runtime maps the result to explicit slot-conflict retry guidance instead of silently reassigning the meal into a new borrowed step. Runtime now emits correlated accepted telemetry after coordinator acceptance and resolved telemetry when the request clears immediately, reconciles after uncertainty, or is cleared by session reset, using persisted replay state for the meal flow identifier + target step until those lifecycle events are emitted so cloud review can close the lifecycle across relaunch/terminate windows. Loop-command telemetry carries explicit command_outcome semantics so uncertain delivery is not flattened into generic blocked state.
- SDD-POL-025 (participant target-change approval flow): The general settings target control is a constrained participant-facing action, not a direct free-form clinical config editor. HomeRegularTargetChangeApprovalPolicy first validates that the requested target belongs to the clinician-selected profile subset and that the request would change the applied target. Valid requests open an approval-capture sheet that blocks dismissal until the operator either cancels or records both approving staff name and approximate approval time. Only after successful validation does the app persist the updated target into the shared clinical config store and update the live runtime-facing target binding. Invalid approval-capture attempts emit blocked telemetry and keep the prior target unchanged.
- SDD-POL-026 (CGM urgent-low review alert): AppCGMManagerDelegate.syncG7Alerts(for:) derives ALERT-CGM-URGENT-LOW from current live G7 data when latestReading.hasReliableGlucose == true, the reading timestamp is within the trusted freshness window, and glucose < 55 mg/dL. The alert is app-derived for clinician review and cloud telemetry, not Dexcom-app alarm acknowledgement. AppAlertCenter keeps the alert active after acknowledge by stamping acknowledgedAt rather than clearing it immediately, so Home / Alert Center can display reviewed state while the urgent-low condition remains active. The active urgent-low alert auto-clears only when a later trustworthy reading is >=55 mg/dL; stale, unreliable, or missing readings leave the active episode in place. Repeated live-state upserts preserve existing acknowledgedAt for the active episode, and a later new <55 mg/dL episode reissues as unacknowledged after recovery clears the prior episode. CGMAlertPersistenceStore persists acknowledged dedupe state so reviewed active urgent-low episodes survive app reset / delegate reattach until glucose recovery clears the episode. As with all CGM-derived alerts, this path never schedules OS local notifications.

- SDD-POL-027 (meal cancel-delivery path): When the operator opens meal announce while pump bolus delivery is already in progress, HomeActionCoordinator routes the flow into a dedicated cancel-delivery presentation on Home instead of exposing a second meal submit path. HomeView reveals a destructive slider-backed Cancel Delivery action beneath the Manual BG / Let's Eat controls, and that action can now remain visible automatically while an active meal-announcement bolus is still in progress, even without a second tap on Let's Eat, by deriving the active cancelable state from current pump delivery state plus the matching meal step telemetry. When PumpStatus.lastDelivery is unavailable during reconnect/status-thin windows, Home falls back to the latest still-delivering meal step record so the cancel affordance is not hidden solely because detailed delivery accounting has not yet arrived. The cancel action calls PumpStatusObserver.cancelBolusDelivery(), which delegates to PumpService.cancelBolus() and converts the canceled dose into actual requested/delivered units. On success, Home shows an orange partial-delivery summary in the normal alert-summary region above the chart so the operator can see how much insulin was already delivered before cancellation before later reopening meal announce. That summary now carries cancel time plus meal type/size context derived from the matching meal step telemetry, so the operator can distinguish which meal was interrupted without reopening the composer. The summary is retained until both at least one later algorithm step has executed and at least 5 minutes have elapsed, preventing the warning from disappearing immediately after the next step. The Home insulin chart derives interrupted-delivery styling directly from existing step telemetry (requestedUnits vs deliveredUnits), but active in-progress meal delivery remains rendered in the normal meal-dose color while the pump still reports .delivering; LoopTelemetryStore now marks a successfully issued bolus as delivering immediately until the first pump refresh arrives, so a just-started meal bolus does not flash yellow while waiting for hardware status. Caution color is reserved for actual interrupted/partial delivery once delivery has been stopped or otherwise ended short. When cancel succeeds, PumpStatusObserver explicitly reconciles the shared LoopTelemetryStore step record with the cancel-returned delivered amount and forces the corresponding local delivery state back to idle, so the chart bar height reflects actual delivered insulin even if the immediate post-cancel status refresh or lagging LoopKit event history still overstates the completed amount. Later pump-status observer refreshes continue reconciling the shared LoopTelemetryStore record for the same step until delivery completes, so an in-progress partial snapshot does not remain yellow on the chart after a later refresh proves full delivery completed. The chart render compactor preserves .delivering when adjacent bars collapse onto one display pixel so compaction cannot strip active-delivery styling. The pump adapter preserves the delivered amount in PumpStatus.lastDelivery so the next algorithm step consumes actual partial-delivery evidence rather than assuming the full original bolus completed, and when LoopKit event history lags a later pod status refresh it now floors delivered insulin to the pod-reported requested - bolusNotDelivered amount instead of regressing a completed bolus back into partial-delivery state.
- SDD-POL-008 (Alert policy, partial): multi-source alert normalization model drives pump and CGM alert mapping plus pump signal-loss debounce (5 minutes), no-active-pod debounce (5 minutes), dedupe, deterministic alert precedence, and auto-clear behavior. Home active-alert presentation uses a vertical carousel ordered by severity then recency, with multiplicity indicator and step-through controls when multiple alerts are active. High-priority non-CGM alerts additionally route to background local notifications through AppAlertCenter with per-alert dedupe and severity cooldown (actionable: 5 minutes, safetyCritical: 60 seconds), and resolved alerts clear matching pending/delivered OS notifications. CGM availability/failure alerts are informational in-app status only and never schedule background local notifications; the FDA-cleared Dexcom application remains the source of truth for CGM alarming. Notification-tap routing maps alert category to app context (pump -> Pod modal, cgm -> CGM modal, non-device -> Home); the cgm route remains defined for in-app navigation/preview support even though production CGM alerts do not schedule OS notifications. Safety-critical alerts expose explicit acknowledge action on Home banner/carousel and in Alert Center (ackState == requiresAcknowledge); current production required-ack alerts are pump fault, pump incompatible, and app-derived ALERT-CGM-URGENT-LOW. OS-notification clear calls are idempotent on every retract attempt (including already-cleared keys) to cover async schedule/retract races; alert lifecycle telemetry still emits alert.notification.cleared only when an active alert transitions to removed. App-level active/recently-cleared alert state is persisted and restored across relaunch; pump/cgm delegate PersistedAlertStore hooks are implemented for issued/unretracted/retracted lifecycle lookup. On no-active-pod restore/sync cleanup, only non-critical pod-tied alerts are auto-retracted; ALERT-PUMP-FAULT and ALERT-PUMP-INCOMPATIBLE are retained until explicit lifecycle retraction or acknowledge closure. Time-sensitive alert countdown text (ALERT-PUMP-EXPIRING and ALERT-PUMP-EXPIRED) is carried as countdown metadata and refreshed by AppAlertCenter on a minute interval while active so displayed remaining time stays current. Pod expiration/expired alerting is sourced from both delegate issue/retract callbacks and state-driven synthesis from pump expiresAt on status refresh/startup to avoid relaunch misses and misclassification. For CGM, failed/expired classification is state-driven (lifecycleState/algorithmState) while delegate fallback keyword mapping prioritizes unavailable-temporary states and does not escalate failed/expired from message-only payload text; the only app-derived CGM required-ack path is the urgent-low review alert defined in SDD-POL-026. Runtime-derived ALERT-ALGORITHM-STEPPING-INTERRUPTED is step-based rather than CGM-receipt-based, carries blocker detail, routes notification-tap navigation to Home, and remains distinct from pump-native alerts while CGM state remains visible as informational context. Remaining work is expanded severity channels/escalation surfaces and real-device/background validation. Mapping baseline and source inventory are maintained in Alert Inventory and Mapping.

5. Data Persistence

- SDD-DATA-001: runtime cadence state is persisted in UserDefaultsLoopRuntimeStateStore.
- SDD-DATA-002: algorithm state blob is persisted in UserDefaultsAlgoStateStore.
- SDD-DATA-003: step telemetry is persisted in LoopTelemetryStore.
- SDD-DATA-004: device managers are persisted via app delegates for reconnect behavior.
- SDD-DATA-005: alert lifecycle state is persisted in both AppAlertCenter (active + recently-cleared timeline, capped at 100 entries) and delegate-level LoopKit PersistedAlertStore implementations (issued/retracted records for pump/cgm sources).
- SDD-DATA-006: clinical settings are persisted in a versioned model with deterministic defaults and migration behavior; weight is normalized lbs (UI) -> kg (runtime input). The persisted clinical config now also carries the participant target-access profile, with migration default/inference rules so legacy saved target values load into either Pregnancy or Standard deterministically.

6. Requirement Allocation

SRS ID	Design Element(s)
SRS-RUN-001..005	SDD-APP-001, SDD-CORE-001, SDD-POL-001, SDD-POL-021
SRS-ALG-001..007	SDD-ALG-001, SDD-QA-001
SRS-CGM-001..005	SDD-CORE-001, SDD-POL-002, SDD-POL-020, SDD-CGM-001, SDD-APP-003, SDD-SIM-001
SRS-BG-001..012	SDD-APP-001, SDD-CORE-001, SDD-BG-001, SDD-POL-010, SDD-POL-011, SDD-POL-012, SDD-LOG-001

SRS ID	Design Element(s)
SRS-CLIN-001..012	SDD-CLIN-001, SDD-POL-013, SDD-POL-017, SDD-POL-025, SDD-DATA-006
SRS-PUMP-001..005	SDD-PUMP-001, SDD-POL-003, SDD-SIM-001
SRS-MEAL-001..006	SDD-APP-001, SDD-CORE-001, SDD-POL-004
SRS-MEAL-007..011	SDD-APP-001, SDD-PUMP-001, SDD-LOG-001, SDD-POL-023, SDD-POL-024, SDD-POL-027
SRS-STATE-001..003	SDD-DATA-001..005, SDD-POL-005
SRS-LOG-001..008	SDD-LOG-001, SDD-SIM-001, SDD-POL-017, SDD-POL-018, SDD-POL-024, SDD-POL-025, SDD-APP-007
SRS-UI-001..008	SDD-APP-001, SDD-POL-006, SDD-POL-007, SDD-POL-009, SDD-POL-014, SDD-POL-018, SDD-POL-019, SDD-CGM-001
SRS-VAL-001	SDD-CLIN-001, SDD-DATA-006
SRS-ALERT-001..015	SDD-ALERT-001, SDD-POL-008, SDD-POL-020, SDD-POL-022, SDD-DATA-005
SRS-SEC-001..002	SDD-LOG-001 + `Docs/Quality/CybersecurityPlan.md` (current software handoff scope)
SRS-SEC-003..004, SRS-SEC-006..009	SDD-AUTH-001, SDD-AUTH-002, SDD-POL-015, SDD-POL-016 + `Docs/Quality/CybersecurityPlan.md` (documented implementation; deferred from current software handoff package)

7. Design Constraints

- iOS background execution is opportunistic; runtime requires robust skip/degraded behavior.
- External device SDK behavior (`OmniBLE`, `G7SensorKit`) constrains event cadence and status timing.
- Closed-loop safety policy requires avoiding manual bolus path exposure.

Security/auth scope note:

- `SDD-AUTH-001`, `SDD-AUTH-002`, `SDD-POL-015`, and `SDD-POL-016` document current implementation boundaries because the code exists in the repository.
- For the current engineering software handoff package, those auth/provider/session-continuity design elements are documented implementation context only.
- Closure claims for `SRS-SEC-003..009` remain deferred and are not being asserted through the current handoff package.

8. BG Design and Pending Decisions

8.1 Functional flow

1. User opens BG entry flow and submits BG with explicit confirmation.
2. Runtime validates session state and computes `expectedStep`.
   - Submitted BG must be within `20...600 mg/dL`.
3. Runtime creates/updates one pending BG candidate:
   - if due step has not executed yet, target that due step;
   - if due step already executed, target immediate next step.
   - if a candidate already exists, replace with newer BG submission.
4. Runtime dispatches `doWork(cause: bgCheck)` without borrowing future slots.
5. On execution, coordinator consumes pending BG candidate only when step index matches candidate target and candidate freshness is valid.
6. Coordinator builds algorithm input with:
   - `BGval` from submitted manual BG.
   - CGM input from normal CGM mapping policy (may remain `-1`).
7. Algorithm executes once for due step.
8. Command application follows normal pump safety policy.
9. Telemetry records `manualBG` source, value, timestamps, and execution outcome.
10. If candidate target step is missed without consumption, runtime discards candidate as too old.

8.2 Guards and invariants

- No future-slot borrowing is allowed for `bgCheck`.
- Exactly one pending BG candidate is kept at a time.
- New BG submit replaces existing pending candidate before consumption.
- Pending BG candidate is single-step scoped and expires if not consumed on immediate next target step.
- Manual BG freshness is required at execution time.
- Pump-unavailable behavior remains unchanged from degraded-mode policy.
- Before first successful anchored step exists, manual BG submit is rejected (step-0 BG rescue disabled by current policy).

8.3 Pending decisions before implementation lock

- Final manual BG freshness window.
- Step-0 BG-rescue policy (`SRS-BG-008`) is deferred; current implementation remains CGM-only at step 0.

9. Simulation Strategy (Medium now, High later)

9.1 Medium-fidelity harness (in-scope)

- Architecture:
  - Scenario runner emits deterministic event timelines into runtime ports.
  - Mock services conform to production protocols (`CGMService`, `PumpService`) so runtime/coordinator logic remains unmodified.
  - Virtual clock controls step-slot progression (`300s` cadence, early window, skip/catch-up timing).
- Scenario event classes:
  - CGM readings (value, timestamp, reliability, availability transitions)
  - pump status transitions (`idle`, `delivering`, `unknown`, disconnect/reconnect)
  - command-result events (success/failure/reconciliation data)
  - normalized alert issue/retract events
- Required outputs:
  - executed step index and skip reason
  - algorithm input/output snapshot
  - command-application decision
  - alert center active/recent lifecycle state

9.2 High-fidelity emulator (future)

- Target:
  - BLE/session-level emulation for Omni/G7 transport behavior (timing jitter, ACK patterns, session-restore edge cases).
- Positioning:
  - not a near-term gating dependency; added after medium-fidelity harness is stable.

9.3 Safety/testing contribution

- Improves hazard-path reproducibility for race conditions that are hard to force on demand with real devices.
- Expands deterministic pre-merge coverage for gating failures (step timing, degraded inputs, command-block behavior, alert escalation).
- Produces traceable verification artifacts that can be mapped to RA/SRS/TV with lower variance than ad hoc manual runs.
- Complements, but does not replace, hardware-in-the-loop verification for true transport and physical-delivery behavior.

Included source: Docs/Quality/SoftwareVerificationAndValidationPlan.md

Software Verification and Validation Plan (SVVP)

Status: Final draft prepared for handoff (pending review)
Version: 0.9
Owner: BionicLoop engineering Prepared by: BionicLoop engineering Reviewer: _____
Approver: _____
Decision date: _____
Effective date: _____
Baseline freeze SHA: _____
Last updated: 2026-04-06 17:20 EDT

Revision History

Version	Date	Author	Summary of Changes
0.1	2026-04-05	Engineering	Initial controlled verification draft
0.9	2026-04-06	BionicLoop engineering	Added handoff-ready metadata, software-handoff disposition language, and clarified the in-scope local security verification row

1. Test Document Acronyms

Common structure used here:

- SVVP: Software Verification and Validation Plan
- STP: Software Test Protocol (test procedures and expected results)
- STR: Software Test Report (actual execution evidence)

2. Verification Strategy

Verification is split into:

- Unit tests (core logic, algorithm mapping, policy gates)
- Integration tests (runtime + adapters + persistence)
- System/manual tests (real-device behavior, BLE reconnection, onboarding flows)

Initial STP draft set:

- STP-ALG-001

- STP-AUTO-001
- STP-SIM-001
- STP-HW-001
- STP-ALERT-001
- STP-TV-Ownership-Map
- STR-Execution-and-Reporting-Guide

Submission-scope note:

- Device-to-cloud / BionicScout verification is not included in the current submission-scope STP draft set and should be treated as deferred/out-of-scope unless submission scope is explicitly revised.
- For the current engineering software handoff package, `TV-SEC-001` remains the only in-scope security verification row; `TV-SEC-002..008` are deferred from claimed closure in this pass.

3. Test Environments

- iOS Simulator for deterministic unit/integration tests.
- Physical iPhone + Dexcom G7 + OmniPod DASH for connection/cadence and delivery behavior.

4. Entry and Exit Criteria

Entry:

- SRS and SDD IDs updated for proposed change.
- Risk impacts reviewed for affected paths.

Exit:

- All planned `TV-*` tests pass or deviations documented.
- Traceability matrix updated with evidence links (`STR-*` artifacts, logs, screenshots).
- No unresolved `High` severity regressions.

5. Seed Test Inventory

Test ID	Level	Purpose	SRS Link
TV-RUN-001	Unit	Expected step math anchored to first successful run	SRS-RUN-001, SRS-RUN-002
TV-RUN-002	Integration	Duplicate step prevention (`stepNotDue`)	SRS-RUN-002
TV-RUN-003	Unit/Integration	Runtime doWork dispatch is constrained to allowed wake causes (`cgmUpdate`, `bgCheck`, `mealAnnounce`, guarded `pumpReconnect`)	SRS-RUN-003
TV-RUN-004	Unit/Integration	Reconnect fallback executes only after an anchored session exists and only when accepted CGM receipt age exceeds the approved fallback freshness limit	SRS-RUN-004, SRS-CGM-005
TV-RUN-005	Unit/Integration	Reconnect fallback does not execute step `0`, does not re-anchor cadence, and does not replay multiple missed slots	SRS-RUN-001, SRS-RUN-002, SRS-RUN-005
TV-RUN-006	Unit/Integration	Fresh accepted CGM receipt suppresses reconnect fallback and same-slot CGM/reconnect triggers coalesce to one execution	SRS-RUN-002, SRS-RUN-004, SRS-RUN-005
TV-RUN-007	System/Hardware	Real-device reconnect fallback validates current-due-step execution after CGM interruption without duplicate command application	SRS-RUN-004, SRS-RUN-005, SRS-CGM-005
TV-ALG-001	Unit (Bridge)	Bridge null-pointer guards and edge-state reset behavior	SRS-ALG-003
TV-ALG-002	Unit (Bridge)	Input mapping and sentinel behavior (`requestTime`, pump availability, subject-id boundaries)	SRS-ALG-003
TV-ALG-003	Unit (Bridge)	Output/state handoff and step increment continuity at bridge boundary	SRS-ALG-003, SRS-ALG-004
TV-ALG-004	Unit (Algorithm)	Deterministic nominal golden-vector replay	SRS-ALG-001

Test ID	Level	Purpose	SRS Link
TV-ALG-005	Unit (Algorithm)	Degraded/unavailable-input golden-vector replay	SRS-ALG-001, SRS-ALG-003
TV-ALG-006	Unit (Algorithm)	Meal/BG/intervention golden-vector replay	SRS-ALG-001, SRS-ALG-005
TV-ALG-007	Unit/Integration	Stateful continuity across persistence/reload/reset boundaries	SRS-ALG-004
TV-ALG-008	Unit (Algorithm)	Boundary/sentinel cases (CGM, BG, pump) remain deterministic and safe	SRS-ALG-003, SRS-ALG-004
TV-ALG-009	Differential	Pregnancy config differential replay (`target`, `upfront`, `TMAX`) vs baseline	SRS-ALG-005
TV-ALG-010	Coverage	Structural coverage report generation and threshold compliance for Algo2015 + bridge	SRS-ALG-002
TV-ALG-011	Toolchain/Process	Static-analysis quality lane execution, and MISRA policy evidence closure as either linked report+deviations or explicit not-applicable decision rationale	SRS-ALG-006, SRS-ALG-007
TV-CGM-001	Unit	Out-of-range CGM -> unavailable (`-1`) mapping	SRS-CGM-001
TV-CGM-002	Unit	Step-0 fresh/in-range gating	SRS-CGM-002
TV-CGM-003	Unit	Step>0 degraded run with unavailable CGM	SRS-CGM-003
TV-CGM-004	Unit/UI	Step-0 blocked-for-CGM path exposes explicit reason/state messaging to user surfaces	SRS-CGM-004, SRS-UI-002
TV-CGM-005	Integration/System	Armed-loop absence of successful step execution beyond the approved interruption threshold is detected as a stalled-step condition using last-success/session-start timing	SRS-CGM-005
TV-BG-001	Unit/Integration	`bgCheck` creates/uses a single pending BG candidate without borrowing future slots beyond immediate next-step policy	SRS-BG-002
TV-BG-002	Unit/Integration	Submit after due-step execution rolls BG candidate to immediate next step and uses it there	SRS-BG-003
TV-BG-003	Unit/Integration	BG value maps to algorithm `BGval` while CGM mapping remains independent	SRS-BG-004
TV-BG-004	Unit/Integration	Pump unavailable during `bgCheck` blocks command application without overriding degraded policy	SRS-BG-005, SRS-PUMP-001
TV-BG-005	Unit/UI	Stale manual BG is rejected with explicit user-visible reason	SRS-BG-006
TV-BG-006	Unit/Integration	Telemetry records `manualBG` source, value, timestamps, and execution outcome	SRS-BG-007, SRS-LOG-001
TV-BG-007	Unit/Integration	Deferred from current software handoff baseline. If step-0 BG rescue is enabled in a future accepted baseline, verify it executes only when policy gates pass.	SRS-BG-008
TV-BG-008	Unit/UI	Manual BG entry rejects values outside `20...600 mg/dL` with explicit validation messaging	SRS-BG-001
TV-BG-009	Unit/Integration	Pending BG candidate expires if not consumed on the immediate next target step	SRS-BG-009
TV-BG-010	Unit/Integration	New BG submission replaces existing pending candidate before execution	SRS-BG-010
TV-BG-011	Unit/Integration	Manual BG submit while loop is disarmed does not dispatch runtime execution (`bgCheck`)	SRS-BG-011
TV-BG-012	Unit/Integration	Manual BG submit before first successful anchored step is rejected and does not create pending BG state	SRS-BG-012
TV-CLIN-001	UI/Integration	Clinical settings access is gated by passcode prompt; incorrect passcode blocks entry; correct passcode unlocks settings	SRS-CLIN-001, SRS-CLIN-002
TV-CLIN-002	UI/Smoke	`Subject ID`, `Weight`, `Start Algo`, and `Reset Algo` are presented in `Clinical Settings` and not in participant-facing settings sections	SRS-CLIN-003

Test ID	Level	Purpose	SRS Link
TV-CLIN-003	Unit/UI	Target selector enforces allowed values (`90, 100, 110, 120, 130 mg/dL`) and rejects out-of-set values	SRS-CLIN-004
TV-CLIN-004	Unit/UI	Meal upfront selector enforces two-option set (`75%, 90%`) and maps selected value into runtime config	SRS-CLIN-005
TV-CLIN-005	Unit/UI	`TMAX` selector enforces `40...70` inclusive with `5`-minute increments	SRS-CLIN-006
TV-CLIN-006	Unit/Integration	Clinical settings persistence restores values across relaunch with deterministic default/migration behavior	SRS-CLIN-007
TV-CLIN-007	Unit/Integration	`Start Algo` and `Reset Algo` behavior remains unchanged after relocation into `Clinical Settings`	SRS-CLIN-008
TV-CLIN-008	Unit	Weight conversion and validation path stores kg from integer lbs UI input	SRS-VAL-001, SRS-CLIN-003
TV-CLIN-009	Unit/Integration	Clinical save-review semantics hold: no persisted/runtime config mutation before `Save+OK`, cancel keeps prior applied config, and persisted update appears in next step telemetry snapshot	SRS-CLIN-007, SRS-CLIN-008, SRS-LOG-001
TV-CLIN-010	Unit/UI	Participant-facing settings and the clinician target selector expose only the target set enabled by the clinician-selected target-access profile (`Pregnancy` vs `Standard`)	SRS-CLIN-009, SRS-CLIN-010
TV-CLIN-011	Unit/UI	Participant target change requires approval capture and blocks apply until approving staff name and approximate approval time are both recorded	SRS-CLIN-011
TV-CLIN-012	Unit/UI	Clinical Settings normalizes the draft target to an allowed profile value when the clinician changes the target-access profile	SRS-CLIN-012, SRS-CLIN-010
TV-CLIN-013	Unit/Integration	Target-access profile persists across save/relaunch/migration and is reflected consistently in both participant and clinician settings views	SRS-CLIN-007, SRS-CLIN-009, SRS-CLIN-010
TV-PUMP-001	Unit	Pump unavailable -> run step, block command application	SRS-PUMP-001
TV-PUMP-002	Integration	Signal-loss policy persistence and clear behavior	SRS-PUMP-001, SRS-UI-002
TV-PUMP-003	Integration	Delivery reconciliation and min-dose quantization behavior	SRS-PUMP-003
TV-PUMP-004	System	Home pod card updates on connect/disconnect without entering settings	SRS-PUMP-004
TV-PUMP-005	Integration/System	Delivery-state clears from `delivering` via auto-refresh without opening Pump settings	SRS-PUMP-005
TV-PUMP-006	UI/Integration	Closed-loop surfaces do not expose manual bolus command paths	SRS-PUMP-002
TV-MEAL-001	Unit	Meal announce borrow-window gating	SRS-MEAL-001
TV-MEAL-002	Unit	Meal announce blocked when pump delivering/unknown	SRS-MEAL-002
TV-MEAL-003	Unit/Integration	Meal announce executes on current due step when slot is already due/missed	SRS-MEAL-004
TV-MEAL-004	Unit	Meal announce rejected before first successful anchored step	SRS-MEAL-005
TV-MEAL-005	Unit	Meal unavailable reason precedence reports `noPump` before `signalLoss` when no active pod is present	SRS-MEAL-002, SRS-UI-002

Test ID	Level	Purpose	SRS Link
TV-MEAL-006	Unit/UI	Meal unavailable messaging includes explicit actionable reason and retry timing when blocked	SRS-MEAL-003, SRS-UI-002
TV-MEAL-007	Unit/UI	Meal composer revalidates availability on foreground refresh and immediately before submit so stale available state cannot dispatch an invalid meal request	SRS-MEAL-006, SRS-UI-002
TV-MEAL-008	Unit/UI/Integration	Meal submit does not present success until runtime result is known; blocked/rejected and uncertain outcomes surface explicit user-facing recovery messaging	SRS-MEAL-007, SRS-UI-002
TV-MEAL-009	Integration	Pending or uncertain meal request state, including correlated flow ID, persists across relaunch and prevents duplicate meal entry until resolved	SRS-MEAL-008, SRS-MEAL-009, SRS-STATE-001
TV-MEAL-010	Integration/System	Command-outcome uncertainty (timeout/error with unresolved delivery state) blocks repeat meal announce and surfaces explicit operator guidance until reconciliation; immediate-success and reconciled meal lifecycle closure remain replayable across terminate/relaunch windows until `resolved` telemetry is emitted	SRS-MEAL-008, SRS-MEAL-009, SRS-PUMP-001
TV-MEAL-011	Integration/System	Competing-trigger slot conflict does not silently lose or reinterpret meal intent; user receives explicit slot-conflict blocked/retry feedback and no hidden reassignment to a different borrowed step	SRS-MEAL-010, SRS-RUN-002, SRS-UI-002
TV-MEAL-012	Unit/UI/Integration	When meal entry is opened during active bolus delivery, the app presents a destructive Home inline cancel-delivery flow, keeps that flow visible while active meal delivery remains in progress, reports actual delivered insulin after cancellation in the Home summary region, retains that summary until both the next later algorithm step and a 5-minute minimum display window have passed, renders active in-progress meal delivery in the normal meal-dose color while reserving caution color for actual interrupted delivery, and preserves the delivered amount for subsequent algorithm-step accounting	SRS-MEAL-011, SRS-PUMP-003, SRS-UI-002
TV-STATE-001	Integration	Relaunch restores cadence and algorithm state	SRS-STATE-001
TV-STATE-002	Integration	Reset clears all session state and starts fresh	SRS-STATE-002
TV-STATE-003	Integration/System	Pump and CGM manager state persistence supports reconnect without forced re-pairing on relaunch	SRS-STATE-003
TV-LOG-001	Unit	Step telemetry contains explicit `step_executed_at` plus input/output/command fields	SRS-LOG-001
TV-LOG-002	Integration	CSV export schema and row append behavior	SRS-LOG-002
TV-LOG-003	Unit/Integration	Async export avoids main-actor blocking	SRS-LOG-003
TV-LOG-004	Unit/UI	Debug-only cloud-log threshold control persists selected level and upload filter remains inclusive (`selected level` and higher severities) with default fallback to `Error`	SRS-LOG-004
TV-LOG-005	Unit	Clinical Settings save flow emits deterministic `ui.critical` telemetry (`state_viewed`, `submit`, `cancel`, `blocked`) with stable element IDs and old/new value details	SRS-LOG-005
TV-LOG-006	Unit/Integration	App lifecycle telemetry includes timezone and clock-check context fields with correct trigger semantics (`launch`, `foreground` >24h gate, `timezone_or_time_changed`)	SRS-LOG-006
TV-LOG-007	Unit/Integration	Meal-request telemetry exposes the implemented lifecycle transitions (`submitted`, `accepted`, `success`, `blocked`, `uncertain`, `resolved`) without optimistic-success duplication, with replay durability across terminate/relaunch windows, and loop-command telemetry preserves explicit command outcome semantics (`applied`, `blocked`, `uncertain`)	SRS-LOG-007, SRS-MEAL-007
TV-LOG-008	Unit/UI	Target-access-profile and participant approval-capture telemetry emit stable `ui.critical` events with required detail fields (`target_range_profile`, requested/applied target, approval metadata, and blocked/cancelled reason)	SRS-LOG-008
TV-UI-001	UI/System	Home loop-state precedence rendering and cadence-phase age classification (`nextDueAt`-based `Active/Aging/Stale`)	SRS-UI-001
TV-UI-002	UI/System	Availability messaging matches runtime outcomes	SRS-UI-002
TV-UI-003	UI/System	CGM/Pod setup modal `Cancel` dismisses directly and does not force settings on no-active-pod startup	SRS-UI-003
TV-UI-004	Unit/UI	Meal announcement composer auto-cancels on app background transition	SRS-UI-004

Test ID	Level	Purpose	SRS Link
TV-UI-005	UI/Smoke	Home primary controls are present and actionable in deterministic launch mode (`settings`, `manual BG`, `Let's Eat`)	SRS-UI-002
TV-UI-006	UI/Smoke	Home settings and manual-BG sheets can be opened and dismissed without dead-end navigation	SRS-UI-002, SRS-BG-001
TV-UI-007	Unit/UI	CGM display masks stale (`>11m`) or unreliable (`hasReliableGlucose == false`) readings as `--` and hides trend arrow	SRS-UI-005
TV-UI-008	Unit/Integration	UTC clock-drift warning behavior: `>600s` skew emits non-blocking actionable warning with 24h rate limit, `<=600s` shows no warning, and unavailable checks do not spam warnings	SRS-UI-006
TV-UI-009	Unit/UI	CGM value formatting maps boundaries to textual `LOW`/`HIGH` across display surfaces and suppresses unit suffix for those states	SRS-UI-007
TV-UI-010	Unit/UI	Home CGM chart uses bounded dynamic y-axis maxima (`300/350/400`) based on displayed peak values	SRS-UI-008
TV-ALERT-001	Unit	Alert normalization maps Omni/G7/runtime events to canonical model fields	SRS-ALERT-001, SRS-ALERT-002
TV-ALERT-002	Unit/Integration	Alert precedence keeps critical alert visible when lower-severity alerts coexist	SRS-ALERT-003
TV-ALERT-003	Integration	Transient reconnect events are debounced/coalesced without suppressing persistent faults	SRS-ALERT-004
TV-ALERT-004	Integration/System	Alert clear/ack rules behave per alert type and update UI state correctly	SRS-ALERT-005
TV-ALERT-005	System/Manual	Protocol-required alerts and wording are present and actionable in app flows	SRS-ALERT-006
TV-ALERT-006	Unit/Integration	High-priority non-CGM alerts emit background local notifications with dedupe/cooldown, while CGM alerts and informational alerts do not	SRS-ALERT-007
TV-ALERT-007	Unit/UI	Alert Center shows active and recently-cleared alerts with deterministic sorting and acknowledge path for required-ack alerts	SRS-ALERT-008, SRS-ALERT-005
TV-ALERT-008	Integration	Pump/CGM persisted-alert lifecycle hooks preserve issued/unretracted/retracted state across relaunch and restore active alert visibility	SRS-ALERT-009
TV-ALERT-009	Unit/Integration	Time-sensitive alert countdown text refreshes at minute cadence while active without notification spam	SRS-ALERT-010, SRS-ALERT-007
TV-ALERT-010	Unit/UI	Home active-alert vertical carousel preserves severity/recency ordering, shows multiplicity, and allows deterministic navigation through active alerts	SRS-ALERT-003, SRS-ALERT-011
TV-ALERT-011	Unit/Integration	No-active-pod cleanup retracts only non-critical pod-tied alerts while retaining `ALERT-PUMP-FAULT` and `ALERT-PUMP-INCOMPATIBLE` until explicit closure	SRS-ALERT-005, SRS-ALERT-012
TV-ALERT-012	Integration/System	`Algorithm Stepping Interrupted` issues an actionable alert, clears on resumed successful stepping or loop disarm, and remains distinct from informational G7 unavailable/failed status surfaces	SRS-ALERT-013, SRS-ALERT-003, SRS-ALERT-004, SRS-ALERT-005
TV-ALERT-013	Unit/Integration/System	`Algorithm Stepping Interrupted` issues after `>15 minutes` without successful step execution while armed, carries blocker/root-cause detail, clears on next successful step or loop disarm, and preserves stronger pump/source-native alert precedence while leaving CGM state as informational context	SRS-ALERT-014, SRS-ALERT-003, SRS-ALERT-004, SRS-ALERT-005, SRS-UI-002
TV-ALERT-014	Unit/Integration	CGM availability/failure normalized alerts remain informational in-app status only, do not expose required-ack behavior, and never schedule background local notifications	SRS-ALERT-015
TV-ALERT-015	Unit/Integration	App-derived CGM urgent-low review alert issues only for trustworthy G7 readings `<55 mg/dL`, preserves reviewed state while active, auto-clears on trustworthy recovery `>=55 mg/dL`, persists acknowledged active state across reset/reattach, and never schedules background local notifications	SRS-ALERT-016, SRS-ALERT-005, SRS-ALERT-007
TV-SEC-001	Integration	Local export controls and file handling behavior, including development-only CSV export and the current file-sharing / open-in-place surface	SRS-SEC-002

Test ID	Level	Purpose	SRS Link
TV-SEC-002	Integration/System	Deferred from current software handoff package. If secure cloud upload primary-path closure is re-entered into scope, verify cloud telemetry upload control behavior and failure handling.	SRS-SEC-001
TV-SEC-003	Integration/System	Deferred from current software handoff package. If protected cloud API access is re-entered into scope, verify it requires valid authenticated session.	SRS-SEC-003, SRS-SEC-006
TV-SEC-004	UI/Integration	Deferred from current software handoff package. If multi-provider onboarding is re-entered into scope, verify allowed sign-in entry points and failure states.	SRS-SEC-004, SRS-SEC-006
TV-SEC-005	Integration/System	Deferred from current software handoff package. If authorization-role enforcement is re-entered into scope, verify unauthorized telemetry/dashboard actions are denied.	SRS-SEC-005, SRS-SEC-006
TV-SEC-006	Unit/Integration	Deferred from current software handoff package. If password-recovery workflow is re-entered into scope, verify reset-code request and confirm-reset success/failure handling.	SRS-SEC-007, SRS-SEC-006
TV-SEC-007	Unit/Integration	Deferred from current software handoff package. If launch session restore is re-entered into scope, verify authenticated UX is preserved when token recovery succeeds.	SRS-SEC-008, SRS-SEC-006
TV-SEC-008	Unit/UI	Deferred from current software handoff package. If auth-failure Home-bypass continuity is re-entered into scope, verify the login-required alert and recovery action.	SRS-SEC-009, SRS-SEC-006

5.0 Algo2015 Structural-Coverage Campaign

The detailed campaign definition, thresholds, and required STR artifact set are maintained in Algo2015 Verification Plan. Execution progress and phase-level closure tracking are maintained in Algo2015 Execution Roadmap.

5.1 Proposed Simulation Campaign (Workstream H)

This campaign adds deterministic scenario replay (medium-fidelity mocks) as a required verification layer for runtime safety logic. It complements hardware-in-the-loop testing and does not replace real-device validation.

Test ID	Level	Purpose	SRS Link
TV-SIM-001	Integration (deterministic sim)	Reproduce anchored cadence across reconnect/relaunch windows and assert step index continuity (`expected`, `executed`, `skipReason`)	SRS-RUN-001, SRS-RUN-002, SRS-STATE-001
TV-SIM-002	Integration (deterministic sim)	Validate step-0 hard gate and step>0 degraded CGM execution (`-1`) across stale/out-of-range/noisy sensor sequences	SRS-CGM-001, SRS-CGM-002, SRS-CGM-003
TV-SIM-003	Integration (deterministic sim)	Validate pump-unknown/unavailable execution with command-block and no false delivery application	SRS-PUMP-001, SRS-PUMP-005
TV-SIM-004	Integration (deterministic sim)	Validate meal and BG trigger interplay under missed-step, reconnect, and degraded-input conditions	SRS-MEAL-001, SRS-MEAL-002, SRS-BG-002, SRS-BG-003
TV-SIM-005	Integration (deterministic sim)	Validate alert lifecycle, countdown refresh progression, dedupe, and clear behavior during state churn	SRS-ALERT-003, SRS-ALERT-004, SRS-ALERT-010

Planned evidence:

- `STR-SIM-*` scenario reports with script file, expected output snapshot, actual output snapshot, and pass/fail deltas.
- Script baseline: `/Users/jcostik/BionicLoop/Scripts/run_sim_harness_verification.sh` (emits `run-context`, `results`, `trace-map`, and suite logs).
- Merge-gate helper: `/Users/jcostik/BionicLoop/Scripts/check_sim_merge_gate.sh` (runs `TV-SIM-*` only when high-risk runtime paths are touched).

Future extension (high-fidelity):

- After medium-fidelity stability, add BLE/session-level emulation cases for hardware-specific transport faults and timing jitter that mock services cannot represent.

Current implemented deterministic simulation coverage:

- `testTVSIM001_AnchoredCadenceAcrossReconnectAndRelaunch` (`TV-SIM-001`)
- `testTVSIM002_StepZeroGateAndStepGreaterThanZeroDegradedCGMExecution` (`TV-SIM-002`)
- `testTVSIM003_PumpUnavailableAndUnknownStatesBlockCommandApplication` (`TV-SIM-003`)
- `testTVSIM004_MealAndBGInterplayAcrossMissedStepsAndReconnectChurn` (`TV-SIM-004`)
- `testTVSIM005_AlertLifecycleChurnCountdownDedupeAndClearTransitions` (`TV-SIM-005`)

Current implemented alert-test coverage:

- `testTopAlertPrefersHigherSeverityThenMostRecent` and `testSortedAlertsOrdersBySeverityRecencyAndStableDedupeKey` cover deterministic alert ordering precedence (`TV-ALERT-002` subset).
- `testHomeAlertCarouselNavigatorClampsAndWrapsIndexes` covers Home vertical-carousel paging invariants (clamp + wrap) used for deterministic multi-alert navigation (`TV-ALERT-010` subset).
- `testNoActivePodDebounceAddsAndClearsAlert` and `testHomeAlertSyncEvaluatorReflectsCombinedPumpConditions` cover no-active-pod alert path and suppression of competing signal-loss state when no pod is present (`TV-ALERT-003`, `TV-ALERT-004` subset).
- `testSignalLossDebounceAddsAndClearsAlert` covers debounce + auto-clear behavior, actionable background notification cooldown/dedupe, and clear-on-retract notification cleanup (`TV-ALERT-003`, `TV-ALERT-004`, `TV-ALERT-006` subset).
- `testSignalLossDebounceSuppressesTransientCondition` covers transient suppression and notification authorization priming dedupe (`TV-ALERT-003`, `TV-ALERT-006` subset).

- `testShowPreviewAlertsSupportsMultipleTypesAndPrecedence` covers severity-filtered background notification routing (critical not informational), alert-category route mapping, and safety-critical acknowledge behavior (`TV-ALERT-002`, `TV-ALERT-004`, `TV-ALERT-006` subset).
- `testCloudTelemetryReporterSurfacesSubjectIDConflictAndStopsRetryFor409Conflict`, `testHomeSettingsViewClearsResolvedSubjectIDConflictAlert`, `testSubjectIDConflictAutoResolutionPolicyRequiresActiveAlertNonEmptySubjectAndNoInFlightCheck`, and `testSubjectIDConflictAutoResolutionPolicyThrottlesSameSubjectAndAllowsChangedSubject` cover the app-policy subject-ID conflict alert lifecycle: issue on permanent cloud claim conflict, explicit retract after successful corrected Clinical Settings save, and throttled Home auto-revalidation of the currently persisted subject ID when a stale conflict alert remains active (`TV-ALERT-005` subset).
- `testCGMAlertsNeverScheduleBackgroundNotifications`, `testCGMAlertMapperFailedFromSensorFailedState`, `testCGMAlertMapperUnavailableFromWarmupState`, and `testCGMFailedAlertRestoresFromLiveStateAcrossAlertCenterResetUntilRecovery` cover the CGM availability/failure policy: informational in-app status only, no required-ack path, and no background local notifications (`TV-ALERT-014`, `TV-ALERT-006` subset).
- `testCGMUrgentLowAlertMapperIssuesForReliableReadingBelow55`, `testCGMUrgentLowAlertMapperClearsAt55OrAbove`, `testCGMUrgentLowAlertMapperSkipsUnreliableReading`, `testCGMUrgentLowAlertMapperSkipsStaleReading`, `testUrgentLowAcknowledgeMarksAlertReviewedWithoutClearingActiveState`, `testCGMUrgentLowAcknowledgePersistsAcrossAlertCenterResetUntilRecovery`, and `testCGMAlertsNeverScheduleBackgroundNotifications` cover the app-derived urgent-low review alert trigger, trustworthy-data gate, reviewed-state retention, reset/reattach persistence, recovery auto-clear, and no-OS-notification policy (`TV-ALERT-015`, `TV-ALERT-006` subset).
- `testAlertCenterTracksRecentlyClearedAlerts` and `testAlertCenterRestoresPersistedActiveAndClearedAlerts` cover in-app Alert Center active/recent behavior and persistence restore path (`TV-ALERT-007`, `TV-ALERT-008` subset).
- `testPumpAlertMapperExpiringIncludesCountdownDeadline`, `testPumpAlertMapperExpiredForPodExpiringAlert`, and `testTimeSensitivePumpExpiringAlertRefreshesMessageWithoutReschedulingNotification` cover pod-expiration countdown mapping (expiring and expired paths) plus minute-refresh text updates without extra background notification scheduling (`TV-ALERT-009`, `TV-ALERT-006` subset).
- UI automation now covers Home-to-Alert-Center routing, acknowledge-to-recent flow, and relaunch persistence visibility (`testUI007_HomeAlertCenterButtonOpensAlertCenter`, `testUI008_AlertCenterAcknowledgeMovesAlertToRecentlyCleared`, `testUI009_AlertCenterPersistsAcrossRelaunch`) (`TV-ALERT-007`, `TV-ALERT-008` subset).
- `testPumpPersistedAlertStoreReturnsIssuedAndRetractedAlerts` and `testCGMPersistedAlertStoreReturnsIssuedAndRetractedAlerts` cover delegate `PersistedAlertStore` issue/retract lookup behavior (`TV-ALERT-008` subset).
- `testPumpExpirationAlertSyncPlannerReturnsRetractsWhenNoExpirationAlertsApply` covers no-active-pod retract-set safety boundary by excluding critical fault/incompatible alerts from auto-retract cleanup (`TV-ALERT-011` subset).
- `testCGMAlertMapperPrioritizesUnavailableOverFailedKeywordCollision` and `testCGMAlertMapperDoesNotClassifyMessageOnlyFailedAsSensorFailure` verify CGM fallback keyword mapping cannot escalate transient/message-only text into `ALERT-CGM-FAILED-OR-EXPIRED` (`TV-ALERT-001` subset).

Current implemented runtime-refactor regression coverage:

- `testMealPumpUnavailableReasonMapping` verifies meal-unavailable reason precedence (`noPump` over `signalLoss` when no active pod exists) (`TV-MEAL-005` subset).
- `testMealAnnouncementSheetLifecycleRevalidatesOnlyOnForeground` and `testHomeRuntimeActionCoordinatorMealComposerContinuationDecision` verify the foreground revalidation gate and stale-composer availability remapping used before meal submit dispatch (`TV-MEAL-007` subset).
- `testMealAnnouncementAvailabilityBlocksPersistedPendingMealRequestAcrossRelaunch`, `testMealAnnouncementAvailabilityReconcilesResolvedPendingMealRequestOnLaunch`, `testMealAnnouncementAvailabilityConsumesPersistedResolvedTelemetryReplayStateOnLaunch`, `testReconciledPendingMealAnnouncementStateClearsWhenTargetStepAlreadyExecuted`, `testMealAnnouncementResolutionEventUsesPersistedFlowIDForResolvedPendingState`, and `testMealAnnouncementResolvedEventUsesPersistedResolvedTelemetryReplayState` verify persisted pending meal-request durability, relaunch duplicate blocking, replay-token consumption, target-step reconciliation, and correlated flow-ID closure for resolved lifecycle telemetry (`TV-MEAL-009` subset, `TV-LOG-007` subset).
- `LoopRuntimeCoordinatorMealAnnouncementTests.testMealAnnouncePersistsPendingMealOnlyAfterExecutionStepAccepted` and `LoopRuntimeCoordinatorMealAnnouncementTests.testMealAnnounceRejectedBeforeAcceptanceDoesNotPersistPendingMealState` verify that pending meal state is written only after the coordinator has accepted a concrete execution step and is not left behind for rejected meal attempts (`TV-MEAL-009` subset).
- `testAnnounceMealReturnsBlockedWhenLoopIsOff`, `testAnnounceMealReturnsBlockedWhenPersistedPendingMealExists`, `testReconciledUncertainPendingMealAnnouncementStateClearsWhenPumpDeliveryMatchesTargetStep`, `testMealAnnouncementResolutionEventUsesReconciledAfterUncertainForUncertainClear`, `testMealAnnouncePersistsPendingMealOnlyAfterExecutionStepAccepted`, `testMealAnnounceRejectedBeforeAcceptanceDoesNotPersistPendingMealState`, `testMealAnnounceUncertainDeliveryRetainsPendingMealState`, and `testHomeMealAnnouncementSubmitPolicyEventsAndBlockedContent` verify that meal submit no longer reports optimistic success, that blocked runtime outcomes map to explicit blocked results, and that Home/runtime expose deterministic `submitted`/`accepted`/`success`/`uncertain`/`resolved` telemetry closure with explicit uncertain reconciliation semantics (`TV-MEAL-008`, `TV-MEAL-010`, `TV-LOG-007` subset).
- `testHomeRuntimeActionCoordinatorRoutesPumpDeliveringToCancelDeliveryFlow`, `testMealAnnouncementCancelledDeliverySummaryUsesPartialDeliveryCopy`, `testMealAnnouncementCancelledDeliverySummaryHandlesNoDeliveredInsulin`, `testMealAnnouncementCancelledDeliverySummaryIncludesCancelDetails`, `testMealAnnouncementCancelledDeliveryPolicyRequiresFiveMinutesAndNextStep`, `testMealAnnouncementCancelledDeliveryPolicyUsesNextStepThreshold`, `testMealAnnouncementDisplaySupportMapsMealContext`, `testPumpServiceAdapterCancellationDeliveryStatusUsesRequestedAndDeliveredUnits`, `testPumpServiceAdapterResolvedBolusDeliveredUnitsPrefersPodCompletionWhenEventHistoryLags`, `testPumpServiceAdapterResolvedBolusDeliveredUnitsUsesBestAvailableProgressWhileBolusing`,

`testPumpServiceAdapterAuthoritativeCompletedDeliveryPrefersCanceledUnitsWhenIdle`, `testRecordDoWorkResultMarksSuccessfulBolusAsDeliveringBeforePumpRefresh`, `testReconcilePumpStatusUpdatesInterruptedDeliveryToCompletedAfterLaterRefresh`, `testReconcileCanceledDeliveryUsesDeliveredUnitsForInterruptedMealBar`, `testPumpStatusObserverRefreshReconcilesSharedTelemetryStoreUntilDeliveryCompletes`, `testPumpStatusObserverApplyCanceledBolusDeliveryReconcilesSharedTelemetry`, `testInsulinChartPointFlagsInterruptedDeliveryWhenDeliveredLessThanRequested`, `testInsulinChartPointDoesNotFlagActiveDeliveryAsInterrupted`, `testInlineInsulinChartStylingUsesCautionColorForInterruptedDelivery`, `testInlineInsulinPointCompactorPreservesDeliveringStateWhenCollapsingPoints`, `testHomeViewStateBuilderActiveMealDeliveryCancellationContextUsesOnlyDeliveringMealStep`, and `testUI002b_MealCancelDeliveryFlowShowsPartialDeliverySummaryAndComposer` verify the meal cancel-delivery path: active-delivery routing into a destructive Home inline cancel flow, automatic visibility of the cancel control while a meal bolus is still actively delivering, requested/delivered-unit reporting after cancellation, orange partial-delivery context in Home's alert-summary region above the chart, cancel-time plus meal-context summary detail, optimistic active-delivery chart state immediately after a successful bolus command, explicit canceled-delivery reconciliation into shared step telemetry so interrupted bar height matches actual delivered insulin, normal meal-color chart rendering while delivery is still active, compactor preservation of delivering state when bars visually collapse, caution-color rendering only for actual interrupted delivery derived from requested-vs-delivered telemetry, later pump-refresh reconciliation back to completed delivery when the bolus finishes normally, pod-status flooring when event-history delivery lags, and preservation of delivered insulin accounting for the next algorithm step when the operator later reopens meal announce (`TV-MEAL-012` subset, `TV-PUMP-003` supporting coverage).
- `testLoopRuntimeEngineResetAlgorithmSessionKeepsClinicalSettings` also confirms session reset clears runtime carry-over while preserving unrelated clinical settings; pending meal-request fields are included in that cleared runtime state (`TV-MEAL-009` supporting coverage).
- `testDoWorkFeedsBackRequestedAndDeliveredWhenBelowDashMinimumQuantum` verifies delivery reconciliation preserves requested-vs-delivered values across steps when request is below DASH minimum deliverable quantum (`TV-PUMP-003`).
- `testLoopRuntimeWorkExecutorRecordsLatestReadingBeforeOperation`, `testLoopRuntimeWorkExecutorSkipsRecordReadingWhenNoLatestReading`, and `testLoopRuntimeWorkExecutorReturnsOperationResultWithoutMutation` verify behavior-preserving extraction for doWork execution snapshot sequencing.
- `testLoopSessionStorePersistsAlgorithmArmedAndRuntimeState` and `testLoopSessionStoreClearRuntimeStateReturnsEmptyState` verify session persistence boundaries.
- `testLoopWorkSchedulerOnlyTriggersForNewTimestampWhileArmed` verifies CGM timestamp dedupe/arm/reset behavior.
- `testLoopAlertMediatorReportsSignalLossUntilKnownRefresh` and `testLoopAlertMediatorKeepsSignalLossForUnknownRefresh` verify signal-loss policy mediation behavior.

Current implemented clock-sync telemetry safety coverage:

- `testDeviceClockSyncMonitorFlagsSkewAndPublishesWarningAtThresholdBreach` verifies midpoint skew calculation and warning emission when absolute skew exceeds 600 seconds (`TV-UI-008`, `TV-LOG-006` subset).
- `testDeviceClockSyncMonitorWithinThresholdReportsOKWithoutWarning` verifies `<=600s` skew reports `ok` and does not emit warning alerts (`TV-UI-008` subset).
- `testDeviceClockSyncMonitorForegroundCheckUses24HourSuccessfulCheckGate` verifies foreground checks are gated by 24-hour successful-check interval (`TV-LOG-006` subset).
- `testDeviceClockSyncMonitorTimezoneChangeForcesFreshCheckInsideForegroundGate` verifies timezone/time-change trigger bypasses the foreground gate and performs a fresh UTC check (`TV-LOG-006`, `TV-UI-008` subset).
- `testDeviceClockSyncMonitorRetriesAndReturnsUnavailableWithoutWarningOnNetworkFailures` and `testDeviceClockSyncMonitorLimitsSkewWarningsToOncePer24Hours` verify retry/unavailable behavior and warning cooldown control (`TV-UI-008` subset).

Current implemented CGM UI stale-display safety coverage:

- `testG7ViewModelMasksStaleReadingAndHidesTrendWhenTimestampOlderThanElevenMinutes` verifies stale CGM masking to `--` and hidden trend arrow when reading age exceeds 11 minutes (`TV-UI-007`).
- `testG7ViewModelMasksUnreliableCurrentReadingAndDoesNotFallbackToHistoryValue` verifies unreliable current CGM readings are masked to `--`, trend is hidden, and UI does not fallback-display historical value while current state is unreliable (`TV-UI-007`).
- `testG7ViewModelMasksUnreliableCurrentReadingWithoutTimestampAndDoesNotFallback` verifies unreliable current reading masking remains enforced when `latestReadingTimestamp` is missing (restore/partial-state edge), preventing fallback numeric display (`TV-UI-007`).
- `testG7ViewModelMasksStalePersistedHistoryWhenNoLiveReadingExists` verifies stale persisted-history fallback is also masked to `--` (`TV-UI-007`).
- `testG7ViewModelUsesFreshPersistedHistoryWhenLatestReadingIsUnavailable` verifies non-stale persisted-history fallback still displays glucose value (control case for `TV-UI-007` boundary behavior).
- `testG7ViewModelDisplayFormattingMapsExtremeValuesToHighLow` verifies boundary formatting (`<=39` -> `LOW`, `>=401` -> `HIGH`) and unit-label suppression semantics for boundary text (`TV-UI-009`).
- `testInlineCGMChartDerivationDynamicYAxisMaximumAndValues` verifies stepped CGM y-axis scaling behavior (`300/350/400`) and corresponding tick derivation (`TV-UI-010`).

Current implemented Algo2015 verification coverage:

- `Algo2015BridgeContractTests` methods cover initial bridge contract behavior for null-guard paths, state-reset edge handling (`stateData == nil && timeStep > 0`), subject-id nil/long boundary handling, and state handoff continuity (`TV-ALG-001`, `TV-ALG-002`, `TV-ALG-003` baseline subset).
- `Algo2015GoldenVectorTests.testNominalCGMSequenceMatchesGoldenOutputs` locks a deterministic nominal replay vector for drift detection (`TV-ALG-004` baseline subset).
- `Algo2015GoldenVectorTests.testUnavailableCGMSequenceProducesFiniteDeterministicOutputs` adds degraded/unavailable-CGM replay coverage (`TV-ALG-005` baseline subset).

- `Algo2015GoldenVectorTests.testMealAndManualBGInputsProduceDeterministicMealPathSignals` adds meal/manual-BG intervention replay coverage (`TV-ALG-006` baseline subset).
- `Algo2015GoldenVectorTests.testPersistedStateReloadMatchesContinuousExecution` and `Algo2015GoldenVectorTests.testResetToFreshStateProducesDeterministicStepZeroOutput` add persistence/reload/reset continuity verification (`TV-ALG-007`).
- `Algo2015GoldenVectorTests.testCGMBoundaryValuesRemainFiniteAndBounded` adds CGM boundary/sentinel replay coverage (`TV-ALG-008` baseline subset).
- `Algo2015GoldenVectorTests.testHigherTargetProducesLessInsulinForSameHyperglycemicSequence` adds differential target-behavior verification (`TV-ALG-009` baseline subset).
- `Algo2015OracleSupport` now provides a reusable oracle framework for deterministic replay, snapshot assertions, and continuity checks across Algo2015 test suites (`TV-ALG-004`, `TV-ALG-005`, `TV-ALG-006`, `TV-ALG-007`, `TV-ALG-008`).
- `Algo2015MetamorphicTests` adds property/metamorphic checks for deterministic replay identity and monotonic sensitivity to target/CGM transforms (`TV-ALG-004`, `TV-ALG-009` supporting evidence).
- `Algo2015DifferentialReplayTests` adds staged differential replay with JSON report output (`differential-report.json`) and now asserts all pregnancy parameters are consumed (`targetMgDL`, `mealUpfrontPercent`, `tmaxMinutes`) with deterministic checks for target monotonicity, applied meal-upfront profile, and TMAX-driven output variation (`TV-ALG-009`).
- `Algo2015DifferentialReplayTests.testPregnancyDifferentialReplayProducesDeterministicReport` now additionally asserts that `90%` upfront meal profile front-loads more meal insulin than `75%` at meal step and in immediate post-meal cumulative window (`TV-ALG-009`).
- Evidence artifact path: `Docs/Quality/Evidence/STR-ALG-001/2026-02-18-tv-alg-001-004/`.
- Additional evidence artifact path: `Docs/Quality/Evidence/STR-ALG-001/2026-02-18-tv-alg-005-006-008/`.
- Continuity evidence artifact path: `Docs/Quality/Evidence/STR-ALG-001/2026-02-18-tv-alg-007/`.
- `Scripts/run_algo2015_coverage.sh` now generates `llvm-profdata`/`llvm-cov` artifacts for `Algo2015/Algorithm_2015_10_13.cpp` and bridge sources (`TV-ALG-010` baseline subset), with focused branch-closure scenarios for `Adapt_MB_Rs`, `Meal_Bolus`, `Highs_Lows`, `Set_Target`, `SaveData`, `Trim_Arrays`, `Pumps_CGM_UI_Fields`, `Extract_CGM_Adapt`, and MB history save/load loops.
- Coverage script now supports explicit exception-package signoff metadata (`reviewerName`, `reviewerRole`, `decisionDate`, `decisionStatus`, `decisionNotes`) via CLI flags or environment variables for formal STR runs.
- Coverage packaging now includes hardened branch exception artifacts:
  - `branch-exception-package.md` with reviewer sign-off section
  - `branch-exception-package.json` with machine-readable sign-off fields and per-symbol rationale/safety/mitigation/disposition records
- `Scripts/run_algo2015_verification.sh` provides staged deterministic orchestration (`prepare`, `coverage`, `run`, `evaluate`, `package`, `all`) with immutable run context + manifest packaging for STR reproducibility.
- `InputFields` automated suite (`TV-ALG-001`, `TV-ALG-002`, `TV-ALG-003`, `TV-ALG-008`, `TV-ALG-009` subset) now runs as part of staged execution and emits structured assertions (`results.json`) plus observations (`inputfields-observations.tsv`).
- `CoreReqs` requirement-tagged suite now runs as part of staged execution and maps assertion outcomes directly to `SRS-ALG-001...005` with structured results (`suites/core-reqs/results.json`).
- `Differential` requirement-tagged suite now runs as part of staged execution and emits structured assertions + JSON report (`suites/differential/results.json`, `suites/differential/differential-report.json`).
- `ToolVerification` boundary-transfer suite now runs as part of staged execution and verifies bridge-to-core parity for deterministic boundary cases (`suites/tool-verification/results.json`).
- `StaticAnalysis` suite now runs as part of staged execution and verifies clang build/analyze lane execution, CodeReviewLog run-SHA linkage, and MISRA-policy linkage metadata (`suites/static-analysis/results.json`).
- MISRA is treated as a risk-based conditional quality lane for this host-side investigational path: formal evidence must close the lane either with linked MISRA report/deviation artifacts (when applicable) or with explicit not-applicable decision rationale captured in the STR decision package.
- Package manifest includes quality-lane linkage fields (`qualityLanes.codeReviewLinkage`, `qualityLanes.misraLinkage`) for STR audit traceability.
- Submission-grade packaging outputs now include:
  - `str-template-check.json` (required artifact completeness check)
  - `suite-assertion-trace-map.{json,md}` (assertion-level `TV-ALG-*` + `SRS-ALG-*` mapping)
  - `reproducibility-recipe.md` (single-command rerun + checksum verification recipe)
- Coverage artifact paths:
  - `Docs/Quality/Evidence/STR-ALG-001/2026-02-18-tv-alg-010-coverage/`
  - `Docs/Quality/Evidence/STR-ALG-001/2026-02-18-tv-alg-010-coverage-clean-01/`
  - `Docs/Quality/Evidence/STR-ALG-001/2026-02-18-tv-alg-011-verification-rerun/`
- `Docs/Quality/Evidence/STR-ALG-001/2026-02-18-tv-alg-012-verification-b2-b3-final/`
- Current coverage snapshot (`2026-02-18`, latest run `tv-alg-012-verification-b2-b3-final`):
  - `Algorithm_2015_10_13.cpp`: function `100.00%`, line `95.13%`, branch `88.02%`
  - `Algo2015Bridge.c`: function `100.00%`, line `100.00%`, branch `100.00%`
- Latest local working snapshot (`2026-02-19`, non-formal run):
  - `Algorithm_2015_10_13.cpp`: function `100.00%`, line `97.33%`, branch `90.58%`
  - `Algo2015Bridge.c`: function `100.00%`, line `100.00%`, branch `100.00%`
- Branch-rationale artifact path:
  - `Docs/Quality/Evidence/STR-ALG-001/2026-02-18-tv-alg-012-verification-b2-b3-final/suites/coverage/uncovered-branch-gap-map.md`
  - Exception package (legacy baseline): `Docs/Quality/Evidence/STR-ALG-001/2026-02-18-tv-alg-010-coverage-clean-01/branch-exception-package.md`
- Staged run summary artifacts:
- `Docs/Quality/Evidence/STR-ALG-001/2026-02-18-tv-alg-012-verification-b2-b3-final/evaluation-summary.json`
- `Docs/Quality/Evidence/STR-ALG-001/2026-02-18-tv-alg-012-verification-b2-b3-final/manifest.json`
- `Docs/Quality/Evidence/STR-ALG-001/2026-02-18-tv-alg-012-verification-b2-b3-final/suites/inputfields/results.json`
- `Docs/Quality/Evidence/STR-ALG-001/2026-02-18-tv-alg-012-verification-b2-b3-final/suites/core-reqs/results.json`
- `Docs/Quality/Evidence/STR-ALG-001/2026-02-18-tv-alg-012-verification-b2-b3-final/suites/differential/results.json`
- `Docs/Quality/Evidence/STR-ALG-001/2026-02-18-tv-alg-012-verification-b2-b3-final/suites/differential/differential-report.json`

- `Docs/Quality/Evidence/STR-ALG-001/2026-02-18-tv-alg-012-verification-b2-b3-final/suites/tool-verification/results.json`

6. Evidence

Expected evidence package per change:

- test command output (`xcodebuild`, `swift test`)
- failing/passing test IDs
- device test logs where applicable
- screenshots for UI safety behavior
- link to changed requirement and risk IDs

7. Deferred/Planned Validation

- Extended overnight cadence reliability runs.
- Real hardware fault-injection scenarios (disconnects, stale CGM, unavailable pump).
- Real hardware no-new-CGM-data interruption runs to verify threshold breach, alert timing, and clear-on-recovery behavior distinct from G7 failed/expired states.
- Planned reconnect-fallback hardware runs after policy implementation to verify `>5 minute` accepted-CGM-receipt gating, current-due-step-only execution, and restored CGM priority after data flow resumes.
- `Algorithm Stepping Interrupted` unit/integration validation is now implemented for step-based timing, no-alert-when-disarmed behavior, and clear-on-success/disarm behavior. Remaining planned validation is real-device/background confirmation for non-CGM blocker coverage and rendered root-cause messaging under live pump/CGM conditions.

Current automated coverage for CGM interruption behavior:

- `BionicLoopAlertTests.testAlgorithmSteppingInterruptionMonitoringSchedulesFutureNotificationAndRaisesAlertAtDeadline`
- `BionicLoopAlertTests.testAlgorithmSteppingInterruptionMonitoringClearsActiveAlertWhenSteppingResumes`
- `BionicLoopInfrastructureTests.testLoopRuntimeEngineArmedSessionSchedulesStepInterruptionMonitoringAndResetClearsIt`
- `BionicLoopInfrastructureTests.testLoopRuntimeEngineForegroundRefreshShowsStepInterruptionWhenThresholdExceededBeforeFirs`
- `BionicLoopInfrastructureTests.testLoopRuntimeEngineForegroundRefreshUsesLastSuccessfulRunDeadlineWhenAvailable`
- `BionicLoopInfrastructureTests.testLoopRuntimeEngineForegroundRefreshDoesNotShowStepInterruptionWhenDisarmed`
- `BionicLoopInfrastructureTests.testLoopRuntimeEngineForegroundRefreshDoesNotShowCGMInterruptionWhenDisarmed`
- Formal usability/human-factors sessions for meal announcement and safety messaging.

8. Xcode Automated UI Testing Strategy

Purpose:

- Use `XCTest` UI automation as repeatable verification evidence for deterministic UI behavior and requirement conformance.

Best leverage areas:

- Navigation and modal routing correctness.
- Presence/enabled-state of safety-critical controls.
- State-to-message rendering for known inputs.
- Regression checks for setup flows and dismiss paths.
- Non-hardware-dependent interaction logic (for example meal sheet presentation/cancel behavior).

Not a primary tool for:

- BLE transport reliability and reconnect behavior.
- Background wake cadence and overnight timing reliability.
- Real pump delivery confirmation and physical device alert timing.

Execution model:

- Run UI tests on Simulator with deterministic launch fixtures.
- Use app launch arguments/environment to force reproducible runtime states.
- Use stable accessibility identifiers for controls, labels, and state badges.
- Keep one fast smoke suite as release gate; keep extended suite for nightly runs.

9. UI Automation Verification Mapping

- Automated UI evidence is acceptable for `[SRS-UI](SoftwareRequirementsSpecification.md#srs-ui)-*` and portions of `[SRS-MEAL](SoftwareRequirementsSpecification.md#srs-meal)-*` and `[SRS-ALERT](SoftwareRequirementsSpecification.md#srs-alert)-*` where behavior is deterministic and fixture-driven.
- Hardware-coupled requirements still require integration/system evidence from real-device runs.
- Preferred command:
  - `xcodebuild -scheme BionicLoop -destination 'platform=iOS Simulator,name=iPhone 17' -only-testing:BionicLoopUITests test`
- Evidence artifacts:

- test logs, pass/fail results, captured screenshots/attachments, and linked `TV-*` IDs in `RTM`.

Current Automated UI Suite Mapping (F5)

XCTest Method	TV-ID Link	Requirement Link	Notes
`testUI001_HomeShowsPrimaryControls`	TV-UI-005	SRS-UI-002	Smoke check for Home control availability using deterministic fixtures.
`testUI002_MealUnavailableWhenLoopOff`	TV-UI-002	SRS-UI-002	Verifies unavailable-state messaging path and dismissal UX.
`testUI003_SettingsSheetCanDismiss`	TV-UI-006	SRS-UI-002	Guards against modal navigation traps in settings entry path.
`testUI004_ManualBGSheetCanOpenAndCancel`	TV-UI-006	SRS-BG-001	Verifies explicit cancel path for manual BG entry UX.
`testUI005_HomeShowsAlertBannerPreview`	TV-ALERT-002	SRS-ALERT-003	Verifies deterministic top-alert preview rendering on Home.
`testUI006_HomeShowsCriticalAlertPreview`	TV-ALERT-002	SRS-ALERT-003	Verifies critical alert preview path and title rendering.
`testUI007_HomeAlertCenterButtonOpensAlertCenter`	TV-ALERT-007	SRS-ALERT-008	Verifies Home alert-center bell entry and active-alert visibility in Alert Center.
`testUI008_AlertCenterAcknowledgeMovesAlertToRecentlyCleared`	TV-ALERT-007	SRS-ALERT-005, SRS-ALERT-008	Verifies acknowledge transition from active alert state to recently-cleared timeline.
`testUI009_AlertCenterPersistsAcrossRelaunch`	TV-ALERT-008	SRS-ALERT-009	Verifies persisted active alert visibility after relaunch (`UI_TEST_PRESERVE_DEFAULTS`).
`testUI010_ClinicalSettingsNavigatesAndAutoLocksOnExit`	TV-CLIN-001	SRS-CLIN-001, SRS-CLIN-002	Verifies passcode-gated entry, unlock success, and auto-lock on exit/re-entry.
`testUI011_ClinicalSettingsSaveDismissesSettingsSheet`	TV-CLIN-009	SRS-CLIN-007, SRS-CLIN-008	Verifies `Save+OK` closes settings flow after review-confirmation path.
`testUI012_ClinicalSettingsInvalidPasscodeBlocksUnlock`	TV-CLIN-001	SRS-CLIN-001, SRS-CLIN-002	Verifies invalid passcode path shows explicit error and keeps clinician controls hidden.
`testUI013_ClinicalControlsVisibleOnlyInsideUnlockedClinicalSettings`	TV-CLIN-002	SRS-CLIN-003	Verifies relocated Start/Reset controls are absent in general settings and present only in unlocked Clinical Settings.
`testUI014_RegularTargetChangeRequiresApprovalCaptureAndPersists`	TV-CLIN-011, TV-CLIN-013	SRS-CLIN-011, SRS-CLIN-007	Verifies regular-settings target changes block until approval fields are completed, then persist into clinician-visible applied target state.
`testUI015_ClinicalTargetPickerFollowsSelectedProfileRange`	TV-CLIN-010	SRS-CLIN-009, SRS-CLIN-010	Verifies the clinician target picker only exposes the targets enabled by the selected Pregnancy/Standard profile.
`testUI016_ClinicalProfileChangeNormalizesTargetAndPersists`	TV-CLIN-012, TV-CLIN-013	SRS-CLIN-012, SRS-CLIN-009, SRS-CLIN-010	Verifies changing the clinician-selected profile snaps an inherited out-of-range draft target to the nearest allowed value and persists the normalized result.

Evidence reference:

- STR-UI-AUTO-001 / 2026-02-12-f5-ui-smoke

Current Clinical Unit Mapping (K1/K2 baseline)

XCTest Method	TV-ID Link	Requirement Link	Notes
`testClinicalSettingsPolicyPasscodeValidation`	TV-CLIN-001	SRS-CLIN-001, SRS-CLIN-002	Verifies the current investigational clinical passcode gate accepts only configured value (`020508`).

XCTest Method	TV-ID Link	Requirement Link	Notes
`testClinicalSettingsPolicyNormalizationAndDefaults`	TV-CLIN-003, TV-CLIN-004, TV-CLIN-005	SRS-CLIN-004, SRS-CLIN-005, SRS-CLIN-006	Verifies allowed-option enforcement and deterministic fallback defaults for target/upfront/TMAX selectors.
`testClinicalSettingsSavePolicyPrepareSaveReviewBlockedStates`	TV-CLIN-001	SRS-CLIN-001, SRS-CLIN-002	Verifies locked/invalid/no-change save attempts are blocked with deterministic reasons/messages.
`testClinicalSettingsSavePolicyPrepareSaveReviewBuildsChangedFieldList`	TV-CLIN-009	SRS-CLIN-007	Verifies review model includes complete changed-field set for old/new clinical config diff.
`testClinicalSettingsSavePolicySaveApplySemantics`	TV-CLIN-009	SRS-CLIN-007, SRS-CLIN-008, SRS-LOG-001	Verifies no persisted change before save confirmation, cancel preserves applied config, and saved config appears in next-step telemetry snapshot fields.
`testClinicalSettingsSavePolicyUICriticalEvents`	TV-LOG-005	SRS-LOG-005	Verifies deterministic `ui.critical` event mapping and detail payload for state_viewed/submit/cancel/blocked paths.
`testClinicalSettingsPolicyTargetRangeProfiles`	TV-CLIN-010, TV-CLIN-012	SRS-CLIN-009, SRS-CLIN-010, SRS-CLIN-012	Verifies Pregnancy/Standard profile subsets and nearest-allowed normalization behavior when the active profile changes.
`testRegularTargetChangeApprovalPolicyPrepareAndValidate`	TV-CLIN-011	SRS-CLIN-011	Verifies participant target changes require approver name and approval timestamp before apply.
`testRegularTargetChangeApprovalPolicyBlocksNoChangeAndOutOfProfileSelection`	TV-CLIN-010, TV-CLIN-011	SRS-CLIN-010, SRS-CLIN-011	Verifies participant target-change flow rejects no-op requests and targets outside the clinician-selected profile.
`testRegularTargetChangeApprovalTelemetryEvents`	TV-LOG-008	SRS-LOG-008	Verifies participant approval-capture telemetry includes target profile, requested/current target, approver name, and approval timestamp.

Current regression command used in development for this slice:

- `xcodebuild -scheme BionicLoop -project BionicLoop.xcodeproj -destination 'platform=iOS Simulator,name=iPhone 17' -only-testing:BionicLoopTests test`

UI execution note for this slice:

- `BionicLoopUITests` are wired into the current scheme and targeted UI cases can be launched with `xcodebuild ... -only-testing:BionicLoopUITests/... test`.
- Focused UI verification for `testUI014_RegularTargetChangeRequiresApprovalCaptureAndPersists`, `testUI015_ClinicalTargetPickerFollowsSelectedProfileRange`, and `testUI016_ClinicalProfileChangeNormalizesTargetAndPersists` passed on 2026-03-25 against simulator device `21A8EB79-294B-4DB2-8AB5-9166F5B375A8` (`Test-BionicLoop-2026.03.25_11-55-08--0400.xcresult`).
- Local simulator/xctrunner instability may still require rerunning the focused UI lane in future environments, but this slice now has a captured green UI pass.

10. Manual Screenshot UI Review Protocol

Scope:

- Required for all user-facing changes, especially safety-state messaging, alert presentation, and clinical controls.

Capture set:

- Light mode and dark mode screenshots.
- Changed screen in: baseline state, interactive state, blocked/error state, and post-action state.
- If applicable, include one large-text (Dynamic Type) capture for key screens.

Review rubric:

- Typography and text integrity:
  - no clipping, truncation, overlap, or ambiguous wording.
  - units/values formatting is consistent (mg/dL, U, %, min, timestamps).
- Spacing and alignment:
  - consistent spacing rhythm and card/control alignment.
  - safe-area compliance; no accidental edge clipping.
- Visual hierarchy:
  - critical safety states and primary actions are immediately distinguishable.
  - secondary text does not compete with critical signals.
- Accessibility and contrast:
  - sufficient contrast in both themes.
  - color is supplemented by text/icon/position cues.
  - tappable controls remain legible and touch-accessible.
- Motion and transitions:
  - state transitions are smooth and non-jarring.
  - no stale labels/icons during animated or async state changes.

Evidence and traceability:

- Save screenshots and review notes under the applicable `STR-*` evidence path.
- Link that path in:
  - `Docs/Quality/TraceabilityMatrix.md`
  - `Docs/Quality/CodeReviewLog.md` entry for the commit
  - any related bug entry in `Docs/Quality/Bugs/BugTracker.md`.

Included source: Docs/Quality/TraceabilityMatrix.md

Requirements Traceability Matrix (RTM)

Status: Submission-candidate trace matrix (formal evidence promotion and freeze metadata pending) Version: 0.91 Owner: BionicLoop engineering Prepared by: BionicLoop engineering Reviewer: _____ Approver: _____ Decision date: _____ Effective date: _____ Baseline freeze SHA: _____ Last updated: 2026-04-07 14:17 EDT

Revision History

Version	Date	Author	Summary of Changes
0.1	2026-04-05	Engineering	Initial controlled RTM draft
0.9	2026-04-06	BionicLoop engineering	Added handoff-ready metadata and refined `RA-009` cybersecurity trace mapping for the software-only handoff package
0.91	2026-04-07	BionicLoop engineering	Narrowed `RA-009` to the current local-security claim set, aligned `RA-013` and `RA-015` evidence notes with the implemented baseline, and changed high-risk freeze blockers to `Rerun needed` status

This matrix links risk hazards, requirements, design elements, and verification artifacts.

Matrix

RA-ID	SRS ID	SDD ID	Verification (TV-ID)	Evidence (STR/Logs)
RA-001	SRS-RUN-001, SRS-RUN-002, SRS-RUN-003	SDD-POL-001, SDD-APP-003	TV-RUN-001, TV-RUN-002, TV-RUN-003, TV-SIM-001	Partial (`Docs/Quality/Evidence/Working/STR-SIM-001/2026-02-19-h5-smoke/`)
RA-002	SRS-CGM-001,	SDD-POL-002, SDD-CGM-001	TV-CGM-001, TV-CGM-002,	Partial (`Docs/Quality/Evidence/Working/STR-SIM-001/2026-02-19-h5-smoke/`)

RA-ID	SRS ID	SDD ID	Verification (TV-ID)	Evidence (STR/Logs)
	SRS-CGM-002, SRS-CGM-003, SRS-CGM-004		TV-CGM-003, TV-CGM-004, TV-SIM-002	
RA-003	SRS-PUMP-001, SRS-PUMP-002, SRS-PUMP-005	SDD-POL-003, SDD-PUMP-001	TV-PUMP-001, TV-PUMP-002, TV-PUMP-005, TV-PUMP-006, TV-SIM-003	Partial (`Docs/Quality/Evidence/Working/STR-SIM-001/2026-02-19-h5-smoke/`)
RA-004	SRS-MEAL-001, SRS-MEAL-002, SRS-MEAL-003, SRS-MEAL-004, SRS-MEAL-005, SRS-MEAL-006	SDD-POL-004, SDD-APP-001	TV-MEAL-001, TV-MEAL-002, TV-MEAL-003, TV-MEAL-004, TV-MEAL-005, TV-MEAL-006, TV-MEAL-007, TV-SIM-004	Partial (BUG-001 real-device closure evidence 2026-02-11: `Docs/Quality/Evidence/` simulation: `Docs/Quality/Evidence/Working/STR-SIM-001/2026-02-19-h5-smoke/`
RA-005	SRS-PUMP-003, SRS-LOG-001	SDD-PUMP-001, SDD-LOG-001	TV-PUMP-003, TV-LOG-001	Partial (`LoopRuntimeCoordinatorPumpExecutionTests.testDoWorkFeedsBackReque`
RA-006	SRS-STATE-001, SRS-STATE-002, SRS-STATE-003	SDD-DATA-001, SDD-DATA-002, SDD-DATA-003, SDD-DATA-004, SDD-POL-005	TV-STATE-001, TV-STATE-002, TV-STATE-003	Partial
RA-007	SRS-PUMP-004, SRS-PUMP-005	SDD-PUMP-001	TV-PUMP-004, TV-PUMP-005	Pending
RA-008	SRS-LOG-001, SRS-LOG-002, SRS-LOG-003, SRS-LOG-004, SRS-	SDD-LOG-001, SDD-POL-017, SDD-POL-018, SDD-POL-024, SDD-POL-025, SDD-APP-007	TV-LOG-001, TV-LOG-002, TV-LOG-003, TV-LOG-004, TV-LOG-005, TV-LOG-006, TV-LOG-007, TV-LOG-008	Partial (implemented baseline: authenticated cloud telemetry envelope + persistent outl cap drop policy + non-blocking upload + expanded runtime/CGM/pump/alert emitters `auth_user_sub` from ID-token `sub` with `UNSET` fallback. Lifecycle telemetry now inclu (`device_timezone_id`, `device_utc_offset_seconds`, `clock_check_result`, optional launch/foreground/time-change trigger semantics. Meal announce telemetry now reco `blocked`, `uncertain`, and `resolved` lifecycle transitions without optimistic-success du preserved across relaunch/session-reset closure. Clinical target telemetry now captures capture details with stable `ui.critical` event contracts. Evidence: `BionicLoopInfrastructureTests.testCloudTelemetryReporterSendsRequiredEn` `BionicLoopInfrastructureTests.testCloudTelemetryReporterFallsBackToUnse` `BionicLoopInfrastructureTests.testCloudTelemetryReporterDerivesAuthUser` `BionicLoopInfrastructureTests.testCloudTelemetryReporterReturnsNilForMa` `BionicLoopInfrastructureTests.testCloudTelemetryReporterNormalizesEnvir`

RA-ID	SRS ID	SDD ID	Verification (TV-ID)	Evidence (STR/Logs)
	LOG-005, SRS-LOG-006, SRS-LOG-007, SRS-LOG-008			`BionicLoopInfrastructureTests.testCloudTelemetryOutboxRestoresInflightE` `BionicLoopInfrastructureTests.testCloudTelemetryOutboxDropsOldestNonHig` `BionicLoopInfrastructureTests.testCloudTelemetryReporterRetriesOnTransi` `BionicLoopInfrastructureTests.testCloudTelemetryReporterMarksPermanentF` `BionicLoopInfrastructureTests.testCloudLogUploadPolicyUsesRemoteOverrid` `BionicLoopInfrastructureTests.testCloudLogUploadLoggerUploadsOnlyAtOrAb` `BionicLoopInfrastructureTests.testCloudLogUploadPolicyLocalThresholdDef` `BionicLoopInfrastructureTests.testCloudLogUploadPolicyPersistsAndEvalua` `BionicLoopInfrastructureTests.testCloudTelemetryReporterQueuesFollowUpF` `BionicLoopInfrastructureTests.testG7ConnectionTelemetryPayloadUsesLifec` `BionicLoopInfrastructureTests.testDeviceClockSyncMonitorForegroundCheck` `BionicLoopInfrastructureTests.testDeviceClockSyncMonitorTimezoneChangeF` `BionicLoopRuntimeTests.testClinicalSettingsSavePolicyUICriticalEvents`, `BionicLoopRuntimeTests.testRegularTargetChangeApprovalTelemetryEvents`, `BionicLoopRuntimeTests.testMealAnnouncementResolutionEventUsesPersisted` `BionicLoopRuntimeTests.testMealAnnouncementResolvedEventUsesPersistedRe` `BionicLoopRuntimeTests.testMealAnnouncementResolutionEventUsesReconcile` `BionicLoopRuntimeTests.testMealAnnouncementAvailabilityConsumesPersiste` `BionicLoopHomeStateTests.testHomeMealAnnouncementSubmitPolicyEventsAndB`
RA-009	SRS-SEC-001, SRS-SEC-002	SDD-LOG-001, SDD-POL-015, CybersecurityPlan.md, Cybersecurity_Handoff_Register.md	TV-SEC-001	Support: Cybersecurity_Local_File_and_Permission_Review.md, Cybersecurity_Basel Cybersecurity_Handoff_Register.md. Formal: `TV-SEC-001` / `STR-SEC-001` required for `SRS-SEC-003..009`.
RA-010	SRS-UI-001, SRS-UI-002, SRS-UI-003, SRS-UI-004, SRS-UI-005, SRS-UI-006, SRS-UI-007, SRS-UI-008, SRS-VAL-001, SRS-BG-001	SDD-POL-006, SDD-POL-007, SDD-POL-009, SDD-POL-014, SDD-POL-018, SDD-POL-019	TV-UI-001, TV-UI-002, TV-UI-003, TV-UI-004, TV-UI-005, TV-UI-006, TV-UI-007, TV-UI-008, TV-UI-009, TV-UI-010	Partial (`Docs/Quality/Evidence/STR-UI-AUTO-001/2026-02-12-f5-ui-smoke/`, plu `Docs/Quality/Evidence/STR-BUG-001/2026-02-11-relaunch-meal/`; clock-sync co `BionicLoopInfrastructureTests.testDeviceClockSyncMonitorFlagsSkewAndPub` `BionicLoopInfrastructureTests.testDeviceClockSyncMonitorWithinThreshold` `BionicLoopInfrastructureTests.testDeviceClockSyncMonitorRetriesAndRetur` `BionicLoopInfrastructureTests.testDeviceClockSyncMonitorLimitsSkewWarni` `BionicLoopInfrastructureTests.testG7ViewModelDisplayFormattingMapsExtre` `BionicLoopHomeStateTests.testInlineCGMChartDerivationDynamicYAxisMaximu`
RA-011	SRS-ALERT-001, SRS-ALERT-002, SRS-ALERT-003, SRS-ALERT-004, SRS-ALERT-005, SRS-ALERT-006, SRS-ALERT-007, SRS-ALERT-008, SRS-	SDD-ALERT-001, SDD-POL-008, SDD-POL-026, SDD-DATA-005	TV-ALERT-001, TV-ALERT-002, TV-ALERT-003, TV-ALERT-004, TV-ALERT-005, TV-ALERT-006, TV-ALERT-007, TV-ALERT-008, TV-ALERT-009, TV-ALERT-010, TV-ALERT-011, TV-ALERT-014, TV-	Partial (implemented: `AppAlertCenter` + Home alert carousel + Home bell + Settings pump/cgm normalized mapping + delegate persisted-alert lifecycle hooks + app-level a notification channel for non-CGM alerts with dedupe/cooldown + minute-refresh time- availability/failure alerts remain informational in-app only and do not schedule OS noti issues only from trustworthy live G7 `<55 mg/dL`, preserves reviewed state while active reattach, and auto-clears on trustworthy recovery `>=55 mg/dL`; evidence: `testTopAler` `testSortedAlertsOrdersBySeverityRecencyAndStableDedupeKey`, `testHomeAlert` `testNoActivePodDebounceAddsAndClearsAlert`, `testHomeAlertSyncEvaluatorRef` `testSignalLossDebounceAddsAndClearsAlert`, `testSignalLossDebounceSuppress` `testShowPreviewAlertsSupportsMultipleTypesAndPrecedence`, `testCGMAlertsNe` `testUrgentLowAcknowledgeMarksAlertReviewedWithoutClearingActiveState`, `testCGMUrgentLowAcknowledgePersistsAcrossAlertCenterResetUntilRecovery`, `testCGMUrgentLowAlertMapperIssuesForReliableReadingBelow55`, `testCGMUrgen` `testAlertCenterTracksRecentlyClearedAlerts`, `testAlertCenterRestoresPersi` `testPumpPersistedAlertStoreReturnsIssuedAndRetractedAlerts`, `testCGMPersistedAlertStoreReturnsIssuedAndRetractedAlerts`, `testPumpExpirationAlertSyncPlannerReturnsRetractsWhenNoExpirationAlerts` `testPumpAlertMapperExpiringIncludesCountdownDeadline`, `testTimeSensitivePumpExpiringAlertRefreshesMessageWithoutReschedulingNo` `testIssueAndRetractPumpAlertUpdatesAppAlertCenter`, `testIssueAndRetractIn` `testIssueAndRetractCGMAlertUpdatesAppAlertCenter`, `testUI007_HomeAlertCen` `testUI008_AlertCenterAcknowledgeMovesAlertToRecentlyCleared`, `testUI009_A` `testAlertCenterClearsNotificationsWhenRetractingAbsentAlert`, `testRetractingAbsentAlertStillClearsNotificationRequests`; simulation evide `001/2026-02-19-h5-smoke/`; source mapping baseline: `Docs/Quality/AlertInventc`

RA-ID	SRS ID	SDD ID	Verification (TV-ID)	Evidence (STR/Logs)
	ALERT-009, SRS-ALERT-010, SRS-ALERT-011, SRS-ALERT-012, SRS-ALERT-015, SRS-ALERT-016		ALERT-015, TV-SIM-005	
RA-012	SRS-BG-001, SRS-BG-002, SRS-BG-003, SRS-BG-004, SRS-BG-005, SRS-BG-006, SRS-BG-007, SRS-BG-008, SRS-BG-009, SRS-BG-010, SRS-BG-011, SRS-BG-012	SDD-BG-001, SDD-POL-010, SDD-POL-011, SDD-POL-012, SDD-LOG-001	TV-BG-001, TV-BG-002, TV-BG-003, TV-BG-004, TV-BG-005, TV-BG-006, TV-BG-007, TV-BG-008, TV-BG-009, TV-BG-010, TV-BG-011, TV-BG-012	Partial
RA-013	SRS-CLIN-001, SRS-CLIN-002, SRS-CLIN-003, SRS-CLIN-004, SRS-CLIN-005, SRS-CLIN-006, SRS-	SDD-CLIN-001, SDD-POL-013, SDD-POL-017, SDD-POL-025, SDD-DATA-006	TV-CLIN-001, TV-CLIN-002, TV-CLIN-003, TV-CLIN-004, TV-CLIN-005, TV-CLIN-006, TV-CLIN-007, TV-CLIN-008, TV-CLIN-009, TV-CLIN-010, TV-CLIN-011, TV-CLIN-012, TV-CLIN-	Support evidence demonstrates selector bounds, profile gating, approval capture, and tl role-based access is not claimed in the current baseline. Formal clinical-settings eviden

RA-ID	SRS ID	SDD ID	Verification (TV-ID)	Evidence (STR/Logs)
	CLIN-007, SRS-CLIN-008, SRS-CLIN-009, SRS-CLIN-010, SRS-CLIN-011, SRS-CLIN-012, SRS-VAL-001, SRS-LOG-008		013, TV-LOG-008	
RA-014	SRS-ALG-001, SRS-ALG-002, SRS-ALG-003, SRS-ALG-004, SRS-ALG-005, SRS-ALG-006, SRS-ALG-007	SDD-ALG-001, SDD-QA-001	TV-ALG-001, TV-ALG-002, TV-ALG-003, TV-ALG-004, TV-ALG-005, TV-ALG-006, TV-ALG-007, TV-ALG-008, TV-ALG-009, TV-ALG-010, TV-ALG-011	Support package exists in `Docs/Quality/Evidence/Working/STR-ALG-001/...`; form required for freeze.
RA-015	SRS-CGM-005, SRS-RUN-004, SRS-RUN-005, SRS-ALERT-013, SRS-ALERT-014, SRS-UI-001, SRS-UI-002	SDD-POL-020, SDD-POL-021, SDD-POL-022, SDD-POL-008, SDD-APP-003, SDD-CGM-001	TV-CGM-005, TV-RUN-004, TV-RUN-005, TV-RUN-006, TV-RUN-007, TV-ALERT-012, TV-ALERT-013, TV-UI-001	Support evidence exists for interruption deadline, alerting, and fresh-CGM suppression fallback when CGM freshness is unavailable. Formal runtime evidence promotion and required for freeze.
RA-016	SRS-MEAL-007, SRS-MEAL-008, SRS-MEAL-	SDD-POL-023, SDD-POL-024, SDD-POL-027, SDD-APP-001, SDD-LOG-001	TV-MEAL-008, TV-MEAL-009, TV-MEAL-010, TV-MEAL-011, TV-MEAL-012, TV-	Support automated evidence exists for pending/uncertain meal state, duplicate blocking Formal meal-lifecycle evidence promotion is still required for freeze.

RA-ID	SRS ID	SDD ID	Verification (TV-ID)	Evidence (STR/Logs)
	009, SRS-MEAL-010, SRS-MEAL-011, SRS-LOG-007, SRS-UI-002		LOG-007, TV-PUMP-003	

Usage

For each PR or change batch:

1. Add/update impacted `SRS-*`.
2. Update `SDD-*` references.
3. Add/update `TV-*` and run tests.
4. Attach evidence references in this table.

Notes

- `Status` values: `Planned`, `In progress`, `Rerun needed`, `Complete`, `Blocked`, `Deferred (current software handoff package)`, and `Deferred (partial scope)` when only a subset of a mapped hazard/row is intentionally claimed in the current package.
- Evidence can reference CI run IDs, local test logs, or manual protocol records.

Included source: Docs/Quality/CybersecurityPlan.md

Cybersecurity Plan (Investigational Build)

Status: Final draft prepared for handoff (pending review)
Version: 0.9
Owner: BionicLoop engineering
Prepared by: BionicLoop engineering Reviewer: _____
Approver: _____
Decision date: _____
Effective date: _____
Baseline freeze SHA: _____
Last updated: 2026-04-06 12:37 EDT

Revision History

Version	Date	Author	Summary of Changes
0.1	2026-03-27	Engineering	Initial cybersecurity draft baseline
0.9	2026-04-06	BionicLoop engineering	Added handoff-ready metadata and expanded the software handoff cyber package with trust boundaries, inherited-control mapping, local-control mapping, and explicit artifact gaps

1. Scope

This plan covers cybersecurity controls for:

- mobile app runtime
- connected-device communications (CGM + pump)
- telemetry data handling (local and future cloud path)

Current software-handoff boundary:

- In current engineering handoff scope:
  - local/device-resident telemetry handling and export controls
  - software design statements for secure telemetry architecture
  - documentation of implemented auth/session code paths where they affect software understanding
- Deferred from the current software handoff package:
  - formal closure of cloud-upload verification

- protected API access, onboarding-provider, authorization-role, password-recovery, session-restore, and auth-failure continuity claims represented by `SRS-SEC-003..009`

Execution tracking for the current software handoff package is maintained in Cybersecurity_Handoff_Register.md.

2. Security Objectives

- Preserve safety-critical command integrity.
- Protect confidentiality of participant data.
- Maintain availability of safety-relevant status and logs.

3. System Trust Boundary Summary

Primary trust-boundary elements for the current software handoff package:

- iPhone-resident BionicLoop application
- Dexcom G7 sensor communication path and supporting official Dexcom application
- Omnipod DASH pod communication path
- local persistence (`UserDefaults`, app documents, keychain)
- local alerting and notification surfaces
- future cloud telemetry path

Current boundary statement:

- BionicLoop owns the controller-app software behavior, local persistence choices, local export behavior, alerting behavior, and cloud-client code it ships.
- Dexcom and Insulet own the cleared device firmware, proprietary radio/device behavior, and primary CGM/pump safety/security controls of their respective products.
- Local package code (`OmniBLE`, `G7SensorKit`, `LoopKit`) is treated as software of unknown provenance / inherited open-source software that still requires version, provenance, and delta tracking by BionicLoop.

4. Supplier / Inherited Control Matrix

Component	Security Property Relied Upon	Current Evidence Available in Repo	Additional Artifact Needed for Handoff-Grade Closure	BionicLoop Ownership Boundary
Omnipod DASH pod / radio session	Device-side command/session security and encrypted command transport	Local OmniBLE implementation uses LTK-backed encrypted DASH transport with nonce sequencing and AES-CCM handling in `OmniBLE/OmniBLE/Bluetooth/EnDecrypt/EnDecrypt.swift`, `OmniBLE/OmniBLE/PumpManager/MessageTransport.swift`, and `OmniBLE/OmniBLE/PumpManager/PodComms.swift`.	Insulet / FDA-cleared product cybersecurity and interoperability artifact or approved supplier summary identifying the relied-upon control.	BionicLoop may rely on the pod/session security model but does not claim original ownership of DASH firmware/radio security.
Dexcom G7 sensor / radio session	Device-side authenticated CGM session behavior	Local G7 integration exposes a dedicated authentication characteristic and authenticated service boundary in `G7SensorKit/G7SensorKit/BluetoothServices.swift`; the plugin README also requires use of the official G7 app in `G7SensorKit/README.md`.	Dexcom / FDA-cleared product cybersecurity and interoperability artifact or approved supplier summary identifying the relied-upon control.	BionicLoop may rely on the G7 session model and official Dexcom app requirement but does not claim original ownership of Dexcom device/app security.
Official Dexcom application	Primary CGM alarm and safety alerting	Current BionicLoop requirements and IFU explicitly state Dexcom remains the source of truth for CGM alarming.	Formal citation package for Dexcom alarm/security behavior used by the study / submission team.	BionicLoop owns only its supplemental in-app CGM status/review presentation.
`OmniBLE` local package	Pump communication implementation	Local source and tests are present in repo; README identifies it as Loop-associated Omnipod Bluetooth PumpManager code; the current provenance review now also records the identified likely public upstream repo URL, likely import commit basis, and shipped local delta summary, and the curated delta review now classifies the reviewed local transport changes as connection/pairing recovery behavior rather than cryptographic weakening.	Any available upstream tag/release mapping for the identified import basis.	BionicLoop owns the chosen version and any local modifications it ships.

Component	Security Property Relied Upon	Current Evidence Available in Repo	Additional Artifact Needed for Handoff-Grade Closure	BionicLoop Ownership Boundary
`G7SensorKit` local package	CGM communication implementation	Local source is present in repo; README documents official G7 app dependency; the current provenance review now also records the identified likely public upstream repo URL, likely initial import and later sync commit basis, and shipped local delta summary, and the curated delta review now records no security-semantic local delta in the reviewed BLE/auth/logging surfaces.	Any available upstream tag/release mapping for the identified import/sync basis.	BionicLoop owns the chosen version and any local modifications it ships.
`LoopKit` local package	Shared data storage / platform helper library behavior	Local source and license are present in repo; the current provenance review now also records the identified likely public upstream repo URL, likely import commit basis, and shipped local delta summary, and the curated delta review now records no security-semantic local delta in the reviewed auth/keychain/logging surfaces.	Any available upstream tag/release mapping for the identified import basis.	BionicLoop owns the chosen version and any local modifications it ships.

5. BionicLoop-Owned Local Security Controls

Control Area	Current Implementation / Observation	Evidence in Repo	Curre Limitat Residua
Token and credential storage	Auth tokens and stored credentials are persisted in iOS Keychain with `kSecAttrAccessibleAfterFirstUnlockThisDeviceOnly`.	`BionicLoop/App/AuthSessionNetworking.swift`	Applies o auth/sessi data, not local tele artifacts.
Authenticated API requests	Protected requests carry `Authorization: Bearer <token>` and retry once after `401` using refreshed token.	`BionicLoop/App/AuthenticatedAPIClient.swift`, `BionicLoop/App/AuthSessionNetworking.swift`	Formal cl for protec cloud/AP scope ren deferred i handoff package.
Local telemetry outbox persistence	Telemetry outbox is persisted locally with bounded queue behavior and retry/permanent-failure handling.	`BionicLoop/App/CloudTelemetryReporter.swift`, `BionicLoop/App/CloudTelemetryOutbox.swift`	Outbox currently persists ir `UserDefa` this shoul treated as app-level implemer detail, no protected term reco store.
Export minimization boundary	Step CSV export is documented as development-only.	`BionicLoop/Runtime/LoopTelemetryStore.swift`, `Docs/Quality/SoftwareRequirementsSpecification.md`	CSV curr writes pla UTF-8 to app Docu directory; explicit fi protectio setting is applied ir code.
File-access surface	App currently enables `UIFileSharingEnabled` and `LSSupportsOpeningDocumentsInPlace`.	`BionicLoop/Resources/Info.plist`	This incre the impor of documen operator/s controls a exported and shoul reviewed before productio release.
Logging discipline	Debug/API/telemetry logging redacts request query strings and avoids dumping bearer tokens directly.	`BionicLoop/App/AuthenticatedAPIClient.swift`, `BionicLoop/App/CloudTelemetryReporter.swift`,	Reviewed the curre handoff

Control Area	Current Implementation / Observation	Evidence in Repo	Curre Limitat Residua
		`BionicLoop/App/AuthSessionNetworking.swift`, `Docs/Quality/Cybersecurity_Logging_and_Secret_Review.md`	package, debug-response-snippet behavior console observabi remain a residual developm risk considera auth/clou scope is l reopened
Platform permissions	Current app declares Bluetooth-central background use and Bluetooth usage descriptions.	`BionicLoop/Resources/Info.plist`	No separ entitleme file is pre an explici least-priv permissio review re is still ne for hando evidence.

6. Security Baseline Controls

- Transport security for all cloud telemetry endpoints (future primary path).
- Minimize persistent local sensitive data; local CSV/export remains development-only.
- Principle of least privilege for iOS permissions, background modes, and file-sharing exposure.
- Dependency and package version tracking (SBOM-ready inventory).
- Logging controls: avoid secrets and unnecessary identifiers in clear text.
- Explicit failure handling for unavailable/unknown device states.
- Identity and access control implementation exists in development builds, but formal closure for protected auth/provider/role flows is deferred from the current software handoff package.

7. Threat Scenarios (Current Focus)

- BLE disruption or spoofed/disrupted communication causing stale status.
- Unauthorized access to exported local telemetry files.
- Data exfiltration via insecure cloud upload path (future risk).
- Dependency vulnerability introducing runtime compromise.
- Account takeover or unauthorized API access via weak/misconfigured auth flows.
- Privilege escalation from incorrect role/scope mapping.
- Improper access to app Documents content because file sharing / open-in-place is enabled while development exports are present.
- Security-relevant behavior drift between local forks of `OmniBLE` / `G7SensorKit` / `LoopKit` and their upstream source without explicit provenance tracking.

8. Planned Evidence Artifacts

- threat model worksheet (data flow + trust boundaries)
- SBOM/dependency inventory snapshots
- static analysis/dependency scan results
- penetration/resilience test notes for network upload path (when implemented)
- incident-response drill notes
- identity-provider configuration review notes (Cognito, Apple, Google, email flow)
- auth negative-test evidence (expired/revoked token, role denial, session hijack resistance)
- supplier / inherited-control matrix with upstream provenance and local delta tracking for `OmniBLE`, `G7SensorKit`, and `LoopKit`
- local export/file-sharing review note covering `UIFileSharingEnabled`, `LSSupportsOpeningDocumentsInPlace`, and development-only CSV behavior
- current execution register: `Docs/Quality/Cybersecurity_Handoff_Register.md`
- current provenance review note: `Docs/Quality/Cybersecurity_SOUP_Provenance_Review.md`
- current embedded-package delta review note: `Docs/Quality/Cybersecurity_Embedded_Package_Delta_Review.md`
- current local file/permission review note: `Docs/Quality/Cybersecurity_Local_File_and_Permission_Review.md`
- current dependency inventory note: `Docs/Quality/Cybersecurity_Dependency_Inventory.md`
- current SBOM/advisory process note: `Docs/Quality/Cybersecurity_SBOM_and_Advisory_Process.md`
- current logging/secret review note: `Docs/Quality/Cybersecurity_Logging_and_Secret_Review.md`

- current supplier artifact request list: `Docs/Quality/Cybersecurity_Supplier_Artifact_Request_List.md`
- current baseline acceptability recommendation: `Docs/Quality/Cybersecurity_Baseline_Acceptability_Recommendation.md`
- current `TV-SEC-001` freeze execution checklist: `Docs/Quality/Cybersecurity_TV_SEC_001_Freeze_Execution_Checklist.md`

9. Current Control and Evidence Matrix

Security Area	Related IDs	Current Handoff Scope	Current Control / Design Statement	Verification / Evidence Position
Inherited DASH and G7 link security	`RA-003`, `RA-009`	Supporting inherited control context	Current code shows encrypted DASH transport and G7 authentication surfaces; BionicLoop relies on supplier/device security rather than re-implementing device firmware controls.	Supplier/FDA artifact linkage still needed before handoff-grade closure can reference inherited controls cleanly.
Local telemetry export and file handling	`SRS-SEC-002`, `TV-SEC-001`, `RA-009`	In scope	Local export is controlled, data minimization remains required, and development-only local export path is explicitly documented. File-sharing/open-in-place surface is explicitly recognized as part of the threat boundary.	Needs formal evidence rerun/promotion at freeze; currently tracked as software-handoff work, not deferred.
Secure cloud upload as primary transport architecture	`SRS-SEC-001`, `TV-SEC-002`, `RA-009`	Design statement documented; formal verification deferred	Architecture intent remains documented: secure cloud upload is the long-term primary transport and local CSV is development-only.	Formal cloud-upload control verification is deferred from the current software handoff package.
Protected API access, onboarding providers, role authorization, password recovery, session restore, auth-failure continuity	`SRS-SEC-003..009`, `TV-SEC-003..008`, `RA-009`	Deferred from current software handoff package	Existing implementation may exist in development builds and remains documented for software understanding.	No closure is claimed in this handoff package; re-open only when auth/cloud scope is intentionally brought back into package scope.
SOUP / local-package provenance	`RA-009` supporting control	In scope as documentation/process requirement	Local packages are embedded in the repo and now have a current repo-observable provenance review note, curated local-delta review note, dependency inventory note, and SBOM/advisory process note for the current baseline.	Current process documents exist; remaining work is any available tag/release mapping plus freeze-time SBOM/advisory execution.

10. Identity and Access Planning (Deferred from Current Software Handoff Package)

This section is intentionally marked as pending team discussion before implementation lock.

- Proposed onboarding providers:
  - `Sign in with Apple`
  - `Google`
  - `Email` (exact mode pending decision: magic link vs password)
- Proposed AWS identity platform:
  - Cognito `User Pool` as primary identity provider
  - optional `Identity Pool` only if temporary AWS credentials are required
- Required decisions before implementation:
  - per-user identity requirement vs limited unauthenticated investigational mode
  - role model and least-privilege scope design
  - token/session lifecycle and revocation handling
  - consent/legal onboarding content
  - deprovisioning and emergency-access process

11. Current Missing Artifacts / Open Cyber Gaps

- Approved supplier or FDA-cited cybersecurity/interoperability artifacts for the exact Dexcom G7 and Omnipod DASH security properties being relied upon
- Any available upstream tag/release mapping for the identified `OmniBLE`, `G7SensorKit`, and `LoopKit` import/sync baselines
- Freeze-time execution of the formal SBOM/advisory artifact defined by `Cybersecurity_SBOM_and_Advisory_Process.md`
- Receiving-team freeze decision on whether to accept the documented investigational conditions for the current Documents-directory export, file-sharing, permission surface, and residual debug-logging posture, or require hardening
- Formal execution and promotion of in-scope `TV-SEC-001` using `Cybersecurity_TV_SEC_001_Freeze_Execution_Checklist.md`
- Clear decision on whether any portion of cloud-security verification should be re-entered into the current handoff package

12. Deferred Auth / Cloud Scope Note

For the current engineering software handoff package:

- `SRS-SEC-003..009` are not claimed as closed software scope

- `TV-SEC-003..008` are not handoff-ready evidence obligations in this pass
- `SDD-AUTH-001`, `SDD-AUTH-002`, `SDD-POL-015`, and `SDD-POL-016` remain documented because the implementation exists, but they should be treated as documented implementation context rather than accepted closure claims for this package

13. Incident Handling (Seed)

- Define severity levels (`Critical`, `High`, `Medium`, `Low`).
- Create response SLAs by severity.
- For `Critical`/`High` findings:
  - triage immediately
  - patch and verify
  - update risk register + traceability matrix
  - communicate impact to study/clinical stakeholders

14. Regulatory Alignment

See `Docs/Quality/RegulatoryReferences.md` for FDA cybersecurity guidance links and applicability notes.

Included source: Docs/Quality/IFU/BionicLoop_IFU_v1.4.md

BionicLoop Instructions for Use (IFU)

Document ID
: IFU-BL-001

Version
: 1.4 (Draft)

Date
: 2026-04-05

Prepared by
: BionicLoop engineering

Reviewed by
: _____

Investigational Use Only
Clinical Use Limitation: BionicLoop is investigational software and is not for independent clinical treatment use.

This document is intended for study-team review and supervised use preparation. The app screens shown are representative simulator captures from the current software baseline.

Revision History

Version	Date	Author	Summary of Changes
1.0	2026-03-05	Team	Initial IFU draft package
1.1	2026-03-05	Team	Expanded operational narrative and alerts
1.2	2026-03-05	Team	Added simulator screenshots and workflow cleanup
1.3	2026-03-05	BionicLoop engineering	Added controlled title page, revision history, generated TOC, and print packaging
1.4	2026-04-05	BionicLoop engineering	Rebuilt as a workflow-driven user guide with refreshed screen set, operational checklists, alert-response guidance, and stronger task coverage

Investigational Use Statement

BionicLoop is investigational software and is **not approved for independent clinical treatment use**.

- Use only under approved study protocol and supervision.
- Use only after protocol-specific training on Dexcom G7, Omnipod DASH, and BionicLoop workflows.
- Do not rely on BionicLoop as the sole basis for treatment decisions outside approved investigational procedures.
- Follow site escalation procedures for therapy-impacting alarms, workflow failures, device faults, and documentation.

1. About This Guide

This IFU describes how trained study users should:

- sign in and recover app access,

- review Home before acting,
- verify clinical settings and start the algorithm session,
- set up CGM and Pod workflows,
- enter manual BG values,
- announce meals,
- stop an in-progress meal-announcement delivery if needed,
- review alerts and recent dose steps,
- document and escalate unresolved issues.

Intended Users

This guide is written for:

- clinicians,
- study staff,
- trained supervised operators.

What This Guide Does Not Replace

This guide does not replace:

- Dexcom G7 app instructions,
- Omnipod DASH setup and safety instructions,
- study protocol,
- site-specific escalation procedures.

2. System Overview

BionicLoop is a research closed-loop app that uses Dexcom G7 glucose data, Omnipod DASH pump connectivity, and a 5-minute algorithm step cadence to present current status and algorithm-directed dosing context.

System Components

Component	Role in Workflow
iPhone running BionicLoop	Home view, settings, meal/BG entry, alert presentation, telemetry
Dexcom G7 sensor and Dexcom ecosystem	Source of glucose data and primary CGM alarm handling
Omnipod DASH Pod	Insulin delivery path
BionicLoop cloud/telemetry path	Remote monitoring and quality review support when configured

What BionicLoop Does

- displays current CGM, Pod, and loop status on Home,
- supports clinical configuration and algorithm session control,
- accepts manual BG entry,
- accepts meal announcement input,
- records recent algorithm/dosing steps for review,
- presents in-app alerts and recent alert history.

What BionicLoop Does Not Do

- BionicLoop does **not** replace Dexcom G7 app alarming.
- BionicLoop does **not** send OS notifications for CGM alerts.
- BionicLoop CGM alerts are informational on Home / Alert Center.
- Dexcom G7 app remains the **source of truth for CGM alarms** and should be configured per protocol.

CGM alerting policy: Keep the FDA-cleared Dexcom application configured correctly for urgent low, low, high, signal, and other sensor alarms. Review any BionicLoop CGM messages as supplemental app context, not as the primary alarm source.

3. Before First Use

Before supervised use, confirm all of the following.

3.1 Training and Readiness Checklist

1. The user has been trained on the study workflow and escalation rules.
2. The iPhone is charged and Bluetooth is enabled.
3. The user can sign in or site login support is available.
4. Dexcom G7 app access is available for pairing and sensor management.
5. Pod supplies and site procedures are available.

6. Subject ID, weight, and approved clinical settings are available.
7. The user understands that unresolved therapy-impacting issues require escalation, not repeated blind retries.

3.2 Daily Start-of-Shift Checks

Before initiating therapy-impacting actions:

1. Launch BionicLoop and confirm the correct subject context.
2. Confirm CGM value freshness and trend availability.
3. Confirm Pod presence and connection state.
4. Review active alerts before announcing a meal or entering BG.
5. If using remote monitoring, confirm login/session continuity.

4. Sign In and Account Access

Use the authentication flow when the app opens to login-required state.

1. Launch the app.
2. Enter the required account identifier.
3. Select `Secure Sign In`.
4. Use `Create account` or `Forgot password?` only when authorized for the current workflow.
5. If authentication cannot be completed, stop and escalate per site process.


Figure 1. Login entry screen.

4.1 If Login Is Lost During Active Therapy

If the app indicates login is required while therapy remains active:

- treat this as a monitoring/session continuity issue,
- reauthenticate as soon as permitted,
- continue to follow protocol-defined local safety procedures,
- document any remote-monitoring gap.

5. Home Screen Orientation

Home is the operational landing surface and should be reviewed before any action.


Figure 2. Home with current CGM, Pod status, insulin chart, and action controls.

5.1 What to Review on Home

Home Area	What to Verify Before Acting
CGM card	Current value is present and appears plausible
Pod card	Pod is connected and not faulted/expired
Loop status card	Algorithm session appears active and not obviously blocked
Alert region above chart	Any active or recent issue that changes what you should do next
Chart	Recent glucose direction and recent insulin delivery pattern
Action row	`Manual BG` and `Let's Eat` are available when expected

5.2 If Home Does Not Look Ready

Do not proceed directly to meal announcement or therapy-impacting action if:

- CGM is missing or stale,
- Pod is not available,
- a pump fault/incompatible/expired condition is active,
- an interruption or blocked-state alert is active and unresolved.

6. Settings, Clinical Configuration, and Dose Review

Use Settings to review the user/session state, device entry points, recent dose history, and protected clinical settings.


Figure 3. Bionic Loop Settings sheet.

6.1 Open Settings

1. Tap the settings gear on Home.
2. Review the current target, subject ID, weight, and algorithm-input summary.
3. Open device settings from this sheet when CGM or Pod setup/review is needed.

6.2 Review Recent Dose Steps

Use `Recent Dose Steps` when you need to confirm what the algorithm recently recommended and delivered.

1. Open Settings.
2. Tap `Recent Dose Steps`.
3. Review step number, delivered dose, CGM value, and timestamps.
4. Use this screen when documenting workflow review or reconciling recent activity.


Recent Dose Steps screen
Figure 4. Recent Dose Steps review screen.

6.3 Clinical Settings

Clinical Settings are passcode-protected and should only be changed by authorized users.

1. Open Settings.
2. Open `Clinical Settings`.
3. Unlock using the approved passcode.
4. Review or update:
   - Subject ID
   - Weight (lb)
   - target profile / glucose target
   - meal split
   - TMAX
5. Save only after confirming the values match the approved subject plan.


Clinical Settings screen
Figure 5. Clinical Settings with algorithm input controls.

6.4 Start Algorithm Session

Use `Start Algo` only from Clinical Settings, and only when no active algorithm session is already running.

Before starting:

1. Confirm subject ID and weight.
2. Confirm the approved target profile and input settings.
3. Confirm CGM and Pod workflows are ready enough to support the first anchored loop step.

7. Dexcom G7 Workflow

Use the CGM flow to onboard or review Dexcom G7 status inside BionicLoop.

7.1 First-Time CGM Setup

1. Open Settings.
2. Select `Open CGM Settings`.
3. Follow the guided Dexcom G7 setup path.
4. Continue pairing/calibration/sensor-management tasks in the Dexcom app as required.


CGM setup screen
Figure 6. Dexcom G7 setup entry view.

7.2 Review CGM Status

Use the CGM settings view to confirm sensor details and recent reading status.

Check:

- sensor expiration timing,
- last reading value and timestamp,
- trend information,
- Bluetooth freshness/connection recency.


CGM settings screen
Figure 7. Dexcom G7 operational/settings view.

7.3 CGM Alarm Handling

If a low, urgent low, high, or other sensor alarm is clinically relevant:

1. Use the Dexcom app as the primary alarm source.
2. Respond per protocol.
3. Use BionicLoop only as supplemental review context on Home / Alert Center.

8. Pod Workflow

Use the Pod flow to begin DASH onboarding or to return to pump-related setup when a new Pod is needed.

1. Open Settings.
2. Select `Open Pod Settings`.
3. Follow the Pod setup instructions for reminder configuration, fill, pairing, and placement.
4. Complete all physical device steps per Omnipod instructions.


Figure 8. Pod setup entry view.

8.1 Before Proceeding With Pod-Dependent Actions

Confirm:

- the Pod is active,
- signal is present,
- no pump fault/incompatible/setup-incomplete alert is active,
- the Pod is not beyond its usable service window.

9. Routine Daily Workflows

9.1 Manual BG Entry

Use manual BG only when needed per protocol or when prompted by the workflow.

1. On Home, tap the droplet button.
2. Select the BG value.
3. Confirm the entry using `Use BG`.
4. Cancel if the value is not ready to submit.


Figure 9. Manual BG entry sheet.

9.2 Meal Announcement

Use meal announcement only when Home is ready and the workflow is available.

1. On Home, tap `Let's Eat`.
2. Select meal type.
3. Select the relative carb content for that meal.
4. Review the enabled delivery control.
5. Slide to deliver when the meal announcement is correct.


Figure 10. Meal announcement composer.


Figure 11. Meal announcement after meal selection, ready for delivery.

9.3 Cancel In-Progress Meal Delivery

If meal-announcement delivery is already in progress, Home presents a dedicated red `Cancel Delivery` slider beneath the main action row.

1. Verify that the cancel action is for the active meal-announcement delivery.
2. Slide `Cancel Delivery`.
3. Wait for the Home summary to report what insulin was actually delivered.
4. If another meal announcement is still needed, reopen the meal flow and re-enter it deliberately.


Figure 12. Home with meal-announcement delivery in progress and cancel slider available.

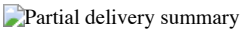
Figure 13. Orange partial-delivery summary shown above the chart after delivery is interrupted.

9.4 How to Interpret the Partial Delivery Summary

If the Home banner states that a partial amount was delivered:

- the algorithm requested more insulin than was actually delivered,
- the displayed delivered amount is the amount that reached the pump before cancellation,
- the partial-delivery summary remains visible long enough for review and then clears automatically after a later step and minimum display time,
- the delivered amount remains part of the recorded therapy history.

9.5 Alert Review

Use Alert Center to review active alerts and recently cleared alerts.

1. Tap the bell on Home.
2. Review all active alerts before proceeding with therapy-impacting actions.
3. Acknowledge or resolve alerts only as permitted by workflow and protocol.
4. Use the recent section to confirm a condition has cleared.


Figure 14. Alert Center showing an active pump fault alert.

10. Alerts and User Response

Alert handling should be based on the alert source and the risk to ongoing therapy.

10.1 Pump / Therapy-Impacting Alerts

Examples include:

- pump fault,
- incompatible Pod,
- no active Pod,
- signal loss,
- Pod expired / stopping dosing soon,
- setup incomplete,
- algorithm stepping interrupted when delivery cannot proceed normally.

User action:

1. Read the alert fully.
2. Resolve the underlying device or workflow issue.
3. Do not assume therapy is normal until the Home state and alerts agree that the condition is cleared.

10.2 CGM Alerts Inside BionicLoop

CGM alerts in BionicLoop are informational app context only.

- BionicLoop does not send OS notifications for CGM alerts.
- Dexcom app remains the source of truth for urgent low and other CGM alarm behavior.
- A BionicLoop urgent-low review alert may appear in-app for trustworthy glucose below the urgent-low threshold; this is a review aid and does not replace Dexcom alarming.

10.3 Alert-Response Principles

Use the following response sequence:

1. Check whether the alert changes therapy readiness.
2. Confirm the device state directly when applicable.
3. Review Home and recent dose context.
4. Escalate unresolved therapy-impacting conditions instead of repeatedly retrying actions.

11. Troubleshooting and Recovery

Situation	What to Do
No current CGM value shown	Check Dexcom app, sensor state, and reading freshness before relying on BionicLoop glucose context
Meal announcement unavailable	Read the blocking message and confirm pump, CGM freshness, timing window, and session status
Pod unavailable / no active Pod	Reconnect or replace the Pod per protocol before proceeding

Situation	What to Do
Pump fault / incompatible alert	Treat as therapy-impacting; replace/recover per protocol and verify Home clears appropriately
Pod expired or stopping soon	Plan Pod replacement before continued therapy reliance
Partial delivery recorded	Review the delivered amount, confirm whether further meal announcement is still needed, and document if clinically relevant
Login required for remote monitoring	Reauthenticate when allowed; document the monitoring gap if required

11.1 If the Screen State and Device State Do Not Match

If BionicLoop and the physical device/app state do not appear to agree:

1. Stop and reassess.
2. Review active alerts.
3. Verify Dexcom state in the Dexcom app and Pod state in the Pod workflow.
4. Capture screenshots if needed.
5. Escalate per site process.

12. Documentation and Escalation

For unresolved or therapy-impacting conditions:

1. Stabilize the situation per study protocol.
2. Record the time and sequence of events.
3. Capture the relevant Home / Alert Center / device screen.
4. Note any partial delivery amount or relevant recent step context.
5. Escalate to the supervising clinical/study contact.

Appendix A. Figure Index

Figure	File	Section Use
Figure 1	`auth-login-start.png`	Sign in
Figure 2	`home-default.png`	Home overview
Figure 3	`home-settings.png`	Settings
Figure 4	`recent-dose-steps.png`	Recent dose review
Figure 5	`home-clinical-settings.png`	Clinical settings
Figure 6	`cgm-setup.png`	CGM setup
Figure 7	`cgm-settings.png`	CGM settings
Figure 8	`pump-setup.png`	Pod setup
Figure 9	`home-manual-bg.png`	Manual BG
Figure 10	`home-meal-announcement.png`	Meal composer
Figure 11	`meal-ready-to-deliver.png`	Meal ready-to-deliver state
Figure 12	`home-meal-cancel-delivery.png`	Cancel-delivery control
Figure 13	`home-meal-cancel-delivery-partial.png`	Partial-delivery summary
Figure 14	`home-alert-center.png`	Alert Center

Appendix B. Abbreviations and Acronyms

Abbreviation	Meaning
IFU	Instructions for Use
CGM	Continuous Glucose Monitor
BG	Blood Glucose
TMAX	Time to maximum insulin effect parameter
DASH	Omnipod DASH pump system

Appendix C. Daily Use Quick Checklist

1. Confirm the correct subject and active session context.
2. Confirm CGM freshness.
3. Confirm Pod readiness.
4. Review alerts before acting.
5. Use `Manual BG` or `Let's Eat` only when Home is ready.
6. Review any partial delivery or interruption summary before proceeding.
7. Document and escalate any therapy-impacting mismatch or unresolved alert.