Software Requirements Specification

for

EagleNav

Version 1.6

Prepared by Jian Mejia, Lcndr Aquino, George Barrone,

Abigail Diaz, Isidoro Flores, Santosh Kumar,

David Lopez, Reda Masri, Athena Ruiz

OSD (Office for Students with Disabilities)

California State University, Los Angeles

May 06, 2026

1. Introduction **5**
1.1 Purpose 5
1.2 Intended Audience and Reading Suggestions 5
1.3 Product Scope 5
1.3.1 Geography (Where it works) 5
1.3.2 Users (Who it is for) 5
1.4 Definitions, Acronyms, and Abbreviations 6
1.5 References 6
2. Overall Description **7**
2.1 System Analysis 7
2.2 Product Perspective 7
2.3 Product Functions 7
2.4 User Classes and Characteristics 8
2.5 Operating Environment 8
2.6 Design and Implementation Constraints 9
2.7 User Documentation 9
2.8 Assumptions and Dependencies 9
2.9 Additional Considerations from User Interviews 9
2.10 Apportioning of Requirements (future implementations) 10
3. External Interface Requirements **11**
3.1 User Interfaces 11
3.2 Hardware Interfaces 11
3.3 Software Interfaces 11
3.4 Communications Interfaces 12
4. Requirements Specification **13**
4.1 Functional Requirements 13
4.1.0 General Application Behavior 13
4.1.1 Destination Selection 13
4.1.2 Destination Confirmation 13
4.1.3 Core Navigation 14
4.1.4 Accessibility – Mobility Support 14
4.1.5 Accessibility – Hearing Support 15
4.1.6 Accessibility – Vision Support 15

4.1.7 Accessibility – Cognitive Support 15
4.1.8 Safety & Security 15
4.1.9 Personalization & Scheduling 15
4.1.10 Notifications & Feedback 16
4.1.11 Search & Discovery 16
4.1.12 Accessibility – Color Blind Support 16
4.1.13 Speech Feedback (Text-to-Speech System) 16
4.1.14 Navigation Enhancements 17
4.1.15 Orientation and Environmental Awareness 19
4.1.16 Implementation Status Summary 20
4.2 External Interface Requirements 17
4.3 Logical Database Requirements 17
5. Other Nonfunctional Requirements 19
5.1 Performance Requirements 19
5.2 Safety Requirements 19
5.3 Security Requirements 19
5.4 Software Quality Attributes 19
5.5 Business Rules 19
Appendix A: Glossary 20
Appendix B: Analysis Models 20
B.1 Data Flow Diagram (High-Level) 20
B.2 Entity-Relationship Diagram (Simplified) 21
B.3 State-Transition Diagram (Navigation Session Example) 21
Appendix C: To Be Determined List 21

Revision History

Name	Date	Reason For Changes	Version
Jian Mejia	09/08/2025	First draft for Eagle Nav SRS	1.0
Jian Mejia	9/30/2025	Expanded functional requirements, added accessibility support (vision/hearing/mobility) and UI requirements	1.2
Abigail Diaz	9/30/2025	Added navigation requirements	1.3
Jian Mejia	10/17/2025	Reformatted	1.4
Jian Mejia	4/28/2026	Refined functional requirements by fixing duplicate and inconsistent requirement IDs. Removed unnecessary general feedback features while retaining obstacle reporting functionality. Incorporated user testing findings to improve usability-focused requirements. Updated section structure, wording, and formatting for clarity and consistency.	1.5
Abigail Diaz	05/06/2026	Added FR-14 (Navigation Enhancements) and FR-15 (Orientation and Environmental Awareness) to capture requirements identified during development and field testing. Added Implementation Status Summary (4.1.16). Updated glossary with navigation-specific terms. Version aligned with current system architecture.	1.6

1. Introduction

1.1 Purpose

This document specifies the software requirements for *Eagle Nav*, a mobile application designed to improve accessibility and navigation on the CSULA campus, with a strong focus on students supported by the Office for Students with Disabilities (OSD). This is the **initial version** of the SRS and represents the baseline requirements for the project's first release.

1.2 Intended Audience and Reading Suggestions

- **Developers**: Use Section 4 for functional and interface requirements.
- **Testers/QA**: Focus on Section 4 and Section 5 for measurable test cases.
- **Project Managers**: Review Sections 2 and 5 for goals, scope, and constraints.
- **OSD Staff & Students**: Refer to Section 2 and Section 3 for user-facing features and accessibility considerations.
- **Documentation Writers**: Use Sections 2.7 and 3.1 for guides and onboarding documentation.

1.3 Product Scope

1.3.1 Geography (Where it works)

- Campus coverage: California State University, Los Angeles main campus only (academic buildings, student services, housing perimeter paths, and all parking structures on the main campus).

- Indoor coverage (v1): Major public corridors, elevators, ramps, and primary entrances/exits of high-traffic buildings. Detailed room-to-room routing is limited to public areas and classrooms listed in the campus map data.

- Off-campus: Not supported (no routing beyond the campus boundary).

1.3.2 Users (Who it is for)

- Primary: CSULA students (including OSD-supported users), staff, and faculty.

- Secondary: Campus visitors.

1.4 Definitions, Acronyms, and Abbreviations

- **OSD** – Office for Students with Disabilities
- **SRS** – Software Requirements Specification
- **API** – Application Programming Interface
- **AR** – Augmented Reality
- **SIS** – Student Information System
- **POI** – Point of Interest

1.5 References

- IEEE Std 830-1998, IEEE Recommended Practice for SRS
- CSULA Campus Map: https://www.calstatela.edu/map

2. Overall Description

2.1 System Analysis

- **Problem:** Students with disabilities face difficulty navigating campus due to incomplete accessibility data (stairs, unclear signage). This largely affects students with visual impairments (our primary user group).
- **Solution:** Provide a mobile app that delivers accessibility-aware navigation, real-time alerts, and safety features without having to rely on inside knowledge about campus.
- **Goals:** Promote inclusion and independence for all students navigating campus.

2.2 Product Perspective

Eagle Nav is a new, standalone mobile application developed specifically for **California State University, Los Angeles** to assist students, faculty, staff, and visitors with **campus navigation and accessibility awareness**.

The app serves as part of the university's ongoing commitment to inclusion and equal access. It is designed to help individuals locate buildings with accessibility features such as ramps, elevators, and other key areas across campus while promoting safe and efficient movement for everyone—especially those supported by the **Office for Students with Disabilities (OSD)**.

EagleNav complements existing campus resources such as printed maps, signage, and facility guides by providing a **modern, interactive tool** that unifies this information in one place. It does not replace existing systems but enhances them by making navigation more intuitive, personalized, and accessible.

The app is intended to be a **campus companion**, helping users orient themselves, plan routes, and stay informed about accessibility-related updates or changes on campus. Its purpose is to create a more inclusive and confident experience for all members of the CSULA community.

2.3 Product Functions

- **Campus Navigation:**
  Helps users locate outdoor buildings and their accessible entrances.

- **Accessible Route Planning:**
  Offers pathways that prioritize accessibility—avoiding stairs and blocked areas while guiding users toward ramps, elevators, and wide walkways.

- **Guided Directions:**
  Provides clear step-by-step directions using both visual and audio cues.

- **Favorites and Scheduling:**
  Allows users to save frequently visited places or plan routes based on their daily class or work schedules.

- **Campus Updates and Alerts:**
  Shares real-time information about construction, closures, or emergency situations that might affect campus travel.

- **Safety Assistance:**
  Includes a quick way to contact campus security or request help in case of an emergency.

- **Inclusive Design:**
  Uses clear visuals, readable text, and color-blind-friendly options to ensure that everyone, regardless of ability, can comfortably use the app.

2.4 User Classes and Characteristics

- **Primary**: Students with disabilities (wheelchair, visually impaired/low vision, hearing impairments).
- **Secondary**: General student body, OSD staff, facilities management, campus security.

2.5 Operating Environment

- Flutter (Dart SDK).
- Android 10+ / iOS 14+.
- Requires Wifi or Cellular Connectivity, GPS, Bluetooth, push notifications, Camera, Haptics.

2.6 Design and Implementation Constraints

- AR via ARCore (Android) and ARKit (iOS).
- No Real-time data integration with campus systems.

2.7 User Documentation

- In-app onboarding tutorial.
- PDF guides for OSD staff.

2.8 Assumptions and Dependencies

- Campus Wi-Fi or cellular connectivity is generally available.
- OSD will maintain accessibility data.
- Users will have Camera/GPS/Bluetooth-enabled smartphones.

2.9 Additional Considerations from User Interviews

2.9.1 Initial Findings from User Interviews

- Both indoor and outdoor navigation present challenges. Outdoor routes are often more difficult due to wide spaces, obstacles, and inconsistent app accuracy. Indoor routes are confusing in buildings such as King Hall, which has C-shaped layouts and inconsistent numbering.
- Scope must be carefully managed. Outdoor navigation may be easier to implement first, while indoor navigation requires more detailed mapping of rooms, wings, elevators, and subsections.
- Users emphasized that requirements are evolving and more interviews are needed to decide priorities between indoor vs. outdoor features. Parallel prototyping may be required to test feasibility.

2.9.2 Findings from User Testing (Latest Iteration)

User testing revealed several important non-technical improvements that would enhance usability and overall effectiveness of the application.

- **Navigation clarity is critical.** Users expect turn instructions and arrival notifications to be accurate and timed appropriately. Early or delayed instructions can cause confusion and reduce trust in the system.

- **Continuous guidance is preferred over step-counting.** Users indicated that counting steps (e.g., “57 steps”) is unreliable, and the system should instead provide ongoing updates such as distance remaining or upcoming turns.
- **Detection feedback must be clear and consistent.** Users expect obstacle information to include direction (left, right, ahead) and to reflect the real environment accurately. Missing or incorrect detections can lead to confusion.
- **People awareness is an important feature.** Users expressed interest in being notified of nearby individuals with distance-based feedback (e.g., “person 5 feet ahead”), especially in crowded areas.
- **Landmark-based guidance is highly valuable.** Users found it helpful when the system provides context about surroundings (e.g., buildings or major campus landmarks) to improve orientation.
- **Independent voice guidance is preferred.** Users appreciated that the application provides its own voice instead of requiring system VoiceOver, allowing flexibility for different accessibility preferences.
- **User interaction should be simple and intuitive.** Users should clearly understand how to start navigation and select destinations without confusion or extra steps.
- **Real-time updates during navigation would improve usability.** Users suggested periodic updates (e.g., distance remaining or confirmation of direction) instead of relying solely on initial instructions.

2.10 Apportioning of Requirements (future implementations)

- Video tutorials (captioned).

3. External Interface Requirements

3.1 User Interfaces

- Clean/ Minimal UI
    - Makes app easy to navigate and find what the user needs
- Large touch targets
    - For users with low vision- makes it easier to find and interact with UI
- Optional High Contrast UI
    - Makes UI easier to differentiate and navigate around
- Customizable font sizes.
    - Increases readability
- Voice Prompt Integration
    - Minimizes the need for typing as it simplifies functionality
- Screen Reader Support
    - Required for those with limited or no visibility

3.2 Hardware Interfaces

- GPS/location services.
- Wifi (for map data)
- Camera (for AR overlays).
- Microphone (voice input).

3.3 Software Interfaces

Eagle Nav connects with a few general data sources to provide accurate, up-to-date navigation across campus.

- **Public Map Data:**
  The app uses general, publicly available map information to display the campus layout and help users navigate between buildings and landmarks.

- **Campus Information Sources:**
  Eagle Nav references publicly accessible university web pages and facilities notices to stay current on updates such as building closures, maintenance work, or accessibility changes.

3.4 Communications Interfaces

- HTTPS/TLS encrypted connections.
- Push notifications
- Location sharing for emergency services.

4. Requirements Specification

This section specifies all requirements for *Eagle Nav*. Each requirement is uniquely identified and structured for traceability, testability, and clarity.

4.1 Functional Requirements

4.1.0 General Application Behavior

- **FR-0.1** The system shall display the campus map as the default view when the application is opened.
- **FR-0.2** The system shall provide access to core functions including Home, Favorites, Alerts, Profile, and Emergency Assistance.
- **FR-0.3** Upon launch, the system shall automatically focus on the search bar and use Text-to-Speech to announce: “Search for a destination,” allowing the user to begin speaking or typing their query.

4.1.1 Destination Selection

- **FR-1.1** The system shall allow the user to search for campus buildings and favorite landmarks by typing or speaking a name.
- **FR-1.2** The system shall visually display a list of matching search results that updates dynamically as the user types or speaks.
- **FR-1.3** The system shall allow the user to select a destination from the search results or to tap a building or favorite landmark on the map.
- FR-1.4 The system shall allow the user to select a new destination, even when one selection has been made, either by:

  - Searching for another destination in the search bar.

  - Directly tapping a destination on the map.

- FR-1.5 The system shall allow the user to clear the typed input and in doing so, clear the search result list from the map view.
- FR-1.6 The system shall provide a “Start Navigation” button once a destination selection has been made and retract the list of results to clear the map view.

4.1.2 Destination Confirmation

- **FR-2.1** The system shall display the generated route from the user's current location to the selected destination. The route should be color coded so the user may know navigation has not started.
- **FR-2.2** The system shall visually mark the selected building or landmark on the map.
- **FR-2.3** The system shall display the route summary, including total distance, estimated travel time, and entrance name, so that the user is aware of the selected destination before navigation begins.

- **FR-2.4** The system shall allow the user to modify their choice by clearing the current destination or selecting an alternate one prior to starting navigation.
- **FR-2.5** After presenting the route summary for the selected destination, the system shall display a "Start Navigation" button to allow the user to begin navigation.

  (All corresponding announcements provided through Text-to-Speech—see Section 4.1.11 Speech Feedback.)

4.1.3 Core Navigation

- **FR-3.1** The system shall use a custom pedestrian path network for on-campus navigation, providing walkable routes from any user location on campus to the selected destination building, even where external map providers lack pedestrian paths.
- **FR-3.2** Upon starting navigation, the system shall:
  1. Provide turn-by-turn navigation to the selected on-campus destination using the device's GPS to determine the user's current location.
  2. Announce the initial walking direction (straight, left, or right).
- **FR-3.3** The system shall alert the user of obstacles in their path during navigation. (should separate since it deals with warnings)
- **FR-3.4** The system shall automatically recalculate the navigation route if the user deviates from the planned path, updating the route in real time.
- **FR-3.5** The system shall provide vibration feedback to notify the user when they are off the planned route or approaching a turn during navigation.

4.1.4 Accessibility – Mobility Support

- **FR-4.1** The system shall present large, high-contrast buttons with touch targets.
- **FR-4.2** The system shall support voice commands for entering destinations and initiating navigation.
- **FR-4.3** The system shall provide one-tap shortcuts to pre-programmed destinations such as the library, cafeteria, and OSD office.
- **FR-4.4** The system shall provide text-to-speech guidance for turn-by-turn navigation.
- **FR-4.5** The system shall detect obstacles (stairs, blockages, construction) and suggest alternative accessible routes.
- **FR-4.6** The system shall prioritize accessible features such as ramps, elevators, and automatic doors when generating routes.
- **FR-4.7** The system shall provide an emergency button to contact campus security or OSD, transmitting the user's current location.
- **FR-4.8** The system shall allow users to report accessibility issues such as broken elevators or blocked paths.
- **FR-4.9** The system shall support user profiles that adjust routing preferences (wheelchair, cane, temporary injury).

4.1.5 Accessibility – Hearing Support

- **FR-5.1** The system shall provide distinct haptic vibration patterns for left turn, right turn, continue straight, and wrong turn.
- **FR-5.2** Textual Navigation Guidance
    - The system shall provide a mode that allows users to view turn-by-turn navigation instructions as readable text.
- **FR-5.3** The system shall allow customization of vibration intensity, distance alerts, and color schemes.
- **FR-5.4** The system shall support customizable haptic vibration patterns for steering feedback (left, right, straight, wrong turn).

4.1.6 Accessibility – Vision Support

- **FR-6.1** The system shall be compatible with screen readers (VoiceOver, TalkBack).
- **FR-6.2** The system shall provide spoken descriptions of menus, buttons, and navigation steps.
- **FR-6.3** The system shall allow voice-based input for destinations.
- **FR-6.4** The system shall allow adjustment of font sizes and zoom levels.
- **FR-6.5** The system should provide conversational AI-based navigation assistance in future versions.

4.1.7 Accessibility – Cognitive Support

- **FR-7.1** The system shall provide simple, sequential navigation instructions.
- **FR-7.2** The system shall support optional confirmation after each instruction.
- **FR-7.3** The system shall support landmark-based guidance (e.g., “Turn right at the fountain”).
- **FR-7.4** The system shall allow selection of routes that avoid high-traffic or noisy areas.

4.1.8 Safety & Security

- **FR-8.1** The system shall provide a one-touch or voice-activated connection to campus security.
- **FR-8.2** The system shall provide a “Safe Walk” feature that shares real-time location with trusted contacts or campus security until deactivated.
- **FR-8.3** The system shall categorize notifications into academics, events, and emergencies.

4.1.9 Personalization & Scheduling

- **FR-9.1** The system shall allow users to bookmark frequently visited locations.
- **FR-9.2** The system shall provide quick-access commands for bookmarked locations.
- **FR-9.3** The system shall allow users to manually enter their class schedule.
- **FR-9.4** The system shall provide next-class routing suggestions based on schedule and time.

- **FR-9.5** The system should integrate with the university SIS in future versions to automatically import schedules.
- **FR-9.6** The system shall allow customization of voice guidance speed, pitch, and volume.

4.1.10 Notifications & Feedback

- **FR-10.1** The system shall deliver real-time push notifications about construction, blockages, and emergencies.
- **FR-10.2** The system shall update routes within 5 seconds after receiving a facilities alert.

4.1.11 Search & Discovery

- **FR-11.1** The system shall provide a search tool for classrooms, departments, restrooms, elevators, and accessible parking.
- **FR-11.2** The system shall display relevant accessibility attributes for search results (e.g., nearest ramp or elevator).
- **FR-11.3** The system shall support landmark-based search queries (e.g., “eagle statue,” “fountain,” “cafeteria entrance”).
- **FR-11.4** The system shall provide context for indoor search results, including building wing, floor level, and room numbering sequence (ascending/descending, odd/even).
- **FR-11.5** The system shall display relevant points of interest (POIs) such as cafeterias, restrooms, and accessible entrances.
- **FR-11.6** The system shall reflect temporary obstacles or events in search results (e.g., “Main entrance crowded, rerouting to side ramp”).
- **FR-11.7** The system shall allow users to drop virtual beacons and later search for or navigate back to them.

4.1.12 Accessibility – Color Blind Support

- **FR-12.1** The system shall provide at least three **color-blind friendly UI themes** (protanopia, deuteranopia, tritanopia).
- **FR-12.2** The system shall provide **high-contrast color palettes** for maps, routes, and alerts.
- **FR-12.3** The system shall use **icons, shapes, and text labels** in addition to color to convey directions and alerts.
- **FR-12.4** The system shall apply **redundant signals** (color + vibration + text) for all critical notifications.
- **FR-12.5** The system should allow users to customize colors manually if preset modes are insufficient.

4.1.13 Speech Feedback (Text-to-Speech System)

- **FR-13.1** The system shall provide spoken confirmation at the start of navigation: “Navigating to [name of destination].”
- **FR-13.2** The system shall provide spoken descriptions for major navigation stages, including destination selection, route confirmation, navigation start, and arrival.

- **FR-13.3** The system shall vocalize important contextual information such as total distance, estimated travel time, and entrance name.
- **FR-13.4** The system shall announce the activation of key buttons (e.g., “Start Navigation”) when they appear on-screen.
- **FR-13.5** The system shall deliver spoken alerts for rerouting, deviations, or detected obstacles during navigation.

4.1.14 Navigation Enhancements

The following requirements were identified during development and field testing. They describe capabilities that exist in the current implementation but were not captured in the original SRS.

- **FR-14.1** The system shall route to the closest accessible entrance of the selected building rather than to the building centroid. Entrance coordinates shall be resolved from a local data file (buildings.json) bundled with the application.
- **FR-14.2** The system shall express all navigation directions relative to the user’s current body orientation (e.g., “slightly to your right,“ “behind you“) rather than in cardinal or map-relative terms (e.g., “head north“).
- **FR-14.3** The system shall use heading-aware thresholds when determining whether the user has deviated from the planned route. When the user’s compass heading is aligned with the route direction, a generous deviation threshold shall apply. When the user’s heading is opposed to the route direction, a tight threshold shall apply.
- **FR-14.4** The system shall implement a two-phase navigation start. Phase 1 shall orient and guide the user from their current position to the nearest point on the routable path network. Phase 2 shall begin standard turn-by-turn guidance once the user is on the path.
- **FR-14.5** The system shall prioritize urgent spoken cues (turn-now, arrival) over non-urgent ones (countdown alerts, status updates). An urgent cue shall interrupt any non-urgent announcement currently in progress.
- **FR-14.6** The system shall base reroute decisions on a count of consecutive GPS ticks exceeding the deviation threshold rather than on elapsed time, to account for varying GPS update rates across devices.
- **FR-14.7** The system shall convert metric distances returned by the routing engine into approximate step counts using a configurable stride length and shall announce distances in steps rather than meters.
- **FR-14.8** The system shall detect arrival when the user is within 8 meters of the final step endpoint and shall announce the direction of the entrance relative to the user’s current heading.
- **FR-14.9** The system shall use the Valhalla routing engine’s pedestrian costing model with custom parameters that prefer sidewalks and walkways, penalize alleys and tunnels, and exclude unpaved surfaces.

4.1.15 Orientation and Environmental Awareness

The following requirements describe user-facing features that provide spatial awareness and orientation support beyond turn-by-turn navigation.

- **FR-15.1** The system shall provide a Compass Service that smooths raw magnetometer readings, validates them for sudden spikes, and exposes a stable heading value (currentHeading) to all components that require body-relative direction computation.
- **FR-15.2** The system shall provide an Orientation Coach that runs at the start of every navigation session. The coach shall compare the user's compass heading to the bearing of the route's first segment and produce a body-relative coaching phrase (e.g., "turn slightly to your right to align") until the user's heading is within 15 degrees of the route direction.
- **FR-15.3** Once the Orientation Coach determines that the user is aligned with the route, it shall lock in and cease firing for the remainder of the navigation session to avoid interfering with turn-by-turn announcements.
- **FR-15.4** The system shall provide a Landmark Service that, on user request, identifies campus buildings within 50 meters of the user's current position and announces their name, body-relative direction, and approximate distance in steps.
- **FR-15.5** The Landmark Service shall operate independently of whether navigation is active. The user may request a landmark status update from any application state.
- **FR-15.6** The system shall provide its own independent voice channel for spoken guidance, separate from the device's system screen reader (e.g., iOS VoiceOver or Android TalkBack), so that users who rely on other accessibility tools are not interrupted.
- **FR-15.7** The system shall not use gestures that conflict with the device's accessibility layer. Any gesture reserved by VoiceOver or TalkBack (e.g., double-tap for selection) shall not be overridden by the application.

4.1.16 Implementation Status Summary

The following table summarizes the implementation status of each functional requirement section as of version 1.6. Sections marked as Implemented are present in the current build. Sections marked as Partial have some requirements implemented and others deferred. Sections marked as Future Work describe features that were specified but not implemented in the current release.

FR-0 General Application Behavior: Implemented.

FR-1 Destination Selection: Implemented.

FR-2 Destination Confirmation: Implemented.

FR-3 Core Navigation: Implemented.

FR-4 Accessibility, Mobility Support: Partial. FR-4.1, FR-4.4, FR-4.5 implemented. FR-4.2 (voice commands), FR-4.3 (one-tap shortcuts), FR-4.6 (elevator/ramp preference in routing), FR-4.7 (emergency button), FR-4.8 (report issues), FR-4.9 (user profiles) deferred to future work.

FR-5 Accessibility, Hearing Support: Partial. FR-5.1 (haptic turn patterns) implemented. FR-5.3 (custom vibration settings), FR-5.4 (customizable patterns) deferred to future work.

FR-6 Accessibility, Vision Support: Partial. FR-6.1 (screen reader compatibility), FR-6.2 (spoken descriptions), FR-6.4 (font/zoom) implemented. FR-6.3 (voice input), FR-6.5 (conversational AI) deferred to future work.

FR-7 Accessibility, Cognitive Support: Partial. FR-7.1 (simple sequential instructions), FR-7.3 (landmark-based guidance) implemented. FR-7.2 (optional confirmation), FR-7.4 (avoid high-traffic routes) deferred to future work.

FR-8 Safety and Security: Not modified in this update.

FR-9 Personalization and Scheduling: Future Work. All requirements (FR-9.1 through FR-9.6) deferred.

FR-10 Notifications and Feedback: Not modified in this update.

FR-11 Search and Discovery: Partial. FR-11.1 (search tool), FR-11.3 (landmark queries) implemented. FR-11.4 (indoor results), FR-11.6 (temporary obstacles), FR-11.7 (virtual beacons) deferred to future work.

FR-12 Accessibility, Color Blind Support: Future Work. All requirements (FR-12.1 through FR-12.5) deferred.

FR-13 Speech Feedback: Implemented.

FR-14 Navigation Enhancements: Implemented. New in v1.6.

FR-15 Orientation and Environmental Awareness: Implemented. New in v1.6.

4.2 External Interface Requirements

- Input Sources: GPS data, microphone (voice commands), user schedule input
- Outputs: On-screen maps, AR overlays, spoken navigation, haptic patterns, push notifications, emergency alerts.
- Interfaces:
    - Google Maps Flutter plugin (map data).
    - ARCore/ARKit via Flutter bindings.

4.3 Logical Database Requirements

Data Entities:

- User Profiles (accessibility preferences, bookmarks, schedule).
- Campus POI Data (buildings, accessibility attributes, entrances).
- User Created Landmark Data
- Notifications (alerts, emergency updates).

Constraints:

- Must encrypt all user data at rest and in transit.

4.4 Design Constraints

- Must be compatible with Android 10+ and iOS 14+.
- Designed to support users with visual impairments
- Must support secure storage for sensitive data.

5. Other Nonfunctional Requirements

5.1 Performance Requirements

- 95% of routes generated in <3 seconds
- Alerts delivered within 30 seconds of issuance.

5.2 Safety Requirements

- The emergency button must work offline (SMS fallback).
- Safe Walk must continuously transmit location until canceled.

5.3 Security Requirements

- End-to-end encryption for location data.

5.4 Software Quality Attributes

- Usability prioritized over feature complexity.
- Reliability: 99% uptime of core functions.
- Maintainability via modular Flutter architecture.

5.5 Business Rules

- Emergency features restricted to on-campus use.

Appendix A: Glossary

This glossary defines terms, acronyms, and abbreviations used throughout this SRS.

- **API** – Application Programming Interface; software interface that allows communication between different applications or services.
- **AR (Augmented Reality)** – Technology that overlays digital navigation markers and directions on top of the physical world via the device's camera.
- **Beacon (Bluetooth Beacon)** – A small, low-power device that transmits signals to nearby smartphones to aid indoor navigation.
- **Bookmark** – A user-saved location for quick access (e.g., classroom, library).
- **Flutter** – A cross-platform mobile development framework using the Dart programming language, used to build *Eagle Nav*.
- **GPS (Global Positioning System)** – Satellite-based system used for determining a user's outdoor location.
- **Navigation Profile** – A set of user preferences (e.g., wheelchair user, cane user) that customize navigation suggestions.
- **OSD** – Office for Students with Disabilities at CSULA; primary stakeholder for accessibility.
- **Push Notification** – A real-time alert sent to a user's device, often for emergencies or updates.
- **SIS (Student Information System)** – The university system that manages student schedules and course enrollments.
- **SSO (Single Sign-On)** – Authentication system allowing users to log in once and access multiple applications securely.
- **TTS (Text-to-Speech)** – A feature that converts text into spoken voice guidance for visually impaired users.
- **POI** - Point of Interest

- **Body-relative direction** – A direction expressed relative to the user's current facing (e.g., "to your front right") rather than a cardinal direction (e.g., "northeast").
- **buildings.json** – Local data file bundled with the application containing campus building names and their accessible entrance coordinates.
- **Closest-entrance routing** – The technique of routing to the nearest accessible entrance of a building rather than to its geographic center.
- **CompassService** – On-device service that smooths and validates raw magnetometer readings, exposing a stable heading value (currentHeading).
- **currentHeading** – Smoothed compass heading in degrees (0 = magnetic north, clockwise). Used by every body-relative feature in the application.

- **Heading-aware threshold** – A deviation threshold that varies based on whether the user's compass heading aligns with the route direction (generous threshold) or opposes it (tight threshold).
- **Landmark Service** – User-initiated feature that identifies nearby campus buildings and announces their name, body-relative direction, and approximate step distance.
- **Orientation Coach** – One-shot service that runs at navigation start, aligning the user with the route's first segment via compass-based coaching phrases.
- **Pedestrian costing** – Valhalla routing engine configuration that assigns preference weights to different path types (sidewalks preferred, alleys penalized, unpaved excluded).
- **renum_file.osm.pbf** – The campus OpenStreetMap extract in Protocol Buffer Binary Format, hosted on Cloudflare R2 and consumed by the Valhalla routing engine.
- **Two-phase navigation start** – Navigation model where Phase 1 guides the user from their current position onto the routable path network, and Phase 2 begins standard turn-by-turn guidance.
- **Valhalla** – Open-source routing engine developed by Mapzen/Interline, hosted on Google Cloud Platform for EagleNav. Computes pedestrian routes from OpenStreetMap data with configurable costing parameters.

Appendix B: Analysis Models

This appendix contains high-level analysis models to illustrate *Eagle Nav* functionality.

B.1 Data Flow Diagram (High-Level)

1. **Input**: GPS/Bluetooth → Navigation Engine
2. **Input**: User profile/preferences → Navigation Engine
3. **Input**: Facilities management API (alerts) → Notification System
4. **Process**: Navigation Engine generates route → AR Display / Map / Voice Guidance
5. **Output**: User receives visual, audio, or haptic directions
6. **Output**: Emergency button triggers location share → Campus Security

B.2 Entity-Relationship Diagram (Simplified)

- **User** — has → **Profile**
- **Profile** — stores → **Preferences** (accessibility, schedule, bookmarks)
- **Facilities System** — provides → **Alerts**
- **Alerts** — update → **Routes**

B.3 State-Transition Diagram (Navigation Session Example)

States: Idle → Destination Input → Route Calculation → Navigation Active → Route Complete
Events: Voice/Manual Input, Facility Alert, Wrong Turn, Emergency Button

Appendix C: To Be Determined List

===== LATER ADDITIONS (Post-Version 0.2) =====

FR-2.10 The system shall provide a “Cane Mode” and a “Guide Dog Mode” in user profiles, adjusting the level of detail in navigation instructions.

- Cane Mode: Provide detailed obstacle and tactile cue descriptions.

- Guide Dog Mode: Provide higher-level auditory/haptic reinforcement without excessive micro-obstacle details.