Snap Vision

Software Requirements Specification

Demo 4 – Updated Version

Table of Contents

1. Introduction

Business Need:
Navigating complex environments like university campuses or large office buildings is challenging for new students, visitors, and even staff. Snap Vision aims to simplify wayfinding through a mobile app that overlays digital navigation arrows on the real world using Augmented Reality (AR).

Project Scope:
Snap Vision is a mobile navigation tool that offers both 2D map-based and AR-based indoor/outdoor guidance. The system includes:

- A mobile AR navigation interface
- An admin dashboard for managing locations and paths
- A backend with pathfinding and positioning support

2. User Stories/User Characteristics

2.1 Admin related user stories

Title: Upload Floor Plan	Priority: Medium	Estimate: 2-3 days
User Story: As an admin user, I want to upload a floor plan image for a building so that I can visually define and align POIs, and navigation paths.		
Acceptance Criteria: Admin user will have access to all locations. Floor plan is uploaded and associated with the selected building. Floor plan appears on the admin interface for POI placement. Admin can: - Add room POI - Edit room POI - Delete room POI		

Title: Add Indoor Navigation Paths	Priority: High	Estimate: 1 week
User Story: As an admin, I want to connect rooms and areas using walkable paths so the system can calculate the best route for users.		
Acceptance Criteria: Admin interface allows line/path drawing on floor plan. Each path stores: - Distance - Optional metadata: wheelchair accessible, stairs, etc. Paths are visible on the map and used in route calculations.		

Title: QR Marker Location Assignment	Priority: Medium	Estimate: 1-2 days
User Story: As an admin, I want to associate QR codes with specific locations so that users can scan them to set their current position accurately.		
Acceptance Criteria: Admin can enter QR ID and associate it with a building/room. QR scan sets the user's anchor point.		

Title: Manage Users	Priority: Low	Estimate: 2-3 days
User Story: As a current admin, I want to promote other users to an admin or editor role so that they can help manage the map data and I want to be able to delete users.		
Acceptance Criteria: Admin can search for a specific user. Admin can select a user and assign them the role of admin/editor. Admin can filter users/admin users/editor users. Admin can delete a user.		

Title: Manage Buildings	Priority: High	Estimate: 4-5 days
User Story: As an admin user, I want to be able to add and edit building information.		
Acceptance Criteria: Admin can: - Add POI - Edit existing POI - Delete existing POI		

2.2 Editor related user stories

<table>
<tr><td>Title: Upload Floor Plan</td><td>Priority: Medium</td><td>Estimate: 2-3 days</td></tr>
<tr><td colspan="3">User Story:

As an editor user, I want to upload a floor plan image for my specific locations buildings so that I can visually define and align POIs, and navigation paths.</td></tr>
<tr><td colspan="3">Acceptance Criteria:

Editor will only have access to the location admin gives permission for.
Floor plan is uploaded and associated with the selected building.
Floor plan appears on the admin interface for POI placement.

Editor can:
- Add room POI
- Edit room POI
- Delete room POI</td></tr>
</table>

<table>
<tr><td>Title: Add Indoor Navigation Paths</td><td>Priority: High</td><td>Estimate: 1 week</td></tr>
<tr><td colspan="3">User Story:

As an editor, I want to connect rooms and areas using walkable paths so the system can calculate the best route for users of my specific location.</td></tr>
<tr><td colspan="3">Acceptance Criteria:

Editor interface allows line/path drawing on floor plan.

Each path stores:
- Distance
- Optional metadata: wheelchair accessible, stairs, etc.

Paths are visible on the map and used in route calculations.</td></tr>
</table>

Title: QR Marker Location Assignment	Priority: Low	Estimate: 1-2 days
User Story: As an editor, I want to associate QR codes with specific locations for my assigned location so that users can scan them to set their current position accurately.		
Acceptance Criteria: Editor can enter QR ID and associate it with a building/room within their assigned location. QR scan sets the user's anchor point.		

Title: Manage Buildings	Priority: High	Estimate: 4-5 days
User Story: As an editor user, I want to be able to add and edit building information for my specific location.		
Acceptance Criteria: Editor can: - Add POI - Edit existing POI - Delete existing POI		

2.3 Viewer related user stories

2.3.1 Basic Features

Title: Remember Me	Priority: Small	Estimate: 1 day
User Story: As a user, I want quick access to the app, without having to login each time.		
Acceptance Criteria: A user has to create their account, and then they will be logged in permanently.		

Title: Recently Visited	Priority: Small	Estimate: 1 day
User Story: As a user, I want quick access to the locations I have just visited.		
Acceptance Criteria: A user will be able to see the recently visited locations on their home page.		

Title: Help Menu	Priority: Medium	Estimate: 1-2 days
User Story: As a user, I want to access a help menu in the app so that I can easily find information and guidance about how to use the app.		
Acceptance Criteria: User can view answers to Frequently Asked Questions. User can find an email address to contact Support. User can view a tutorial for important features of the app.		

Title: Notifications	Priority: Medium	Estimate: 1-2 days
User Story: As a user, I want to be able to receive notifications for any updates in the app.		
Acceptance Criteria: User will receive notifications when - App has an updated feature - A new badge is available - A user has not visited the app in a week		

Title: Privacy Settings	Priority: Medium	Estimate: 1-2 days
User Story: As a user, I want to be able to control my privacy settings.		
Acceptance Criteria: Users need to grant the app permissions such as location, camera and notifications in order to use certain aspects of the app.		

Title: Dark Mode	Priority: Small	Estimate: 1-2 days
User Story: As a user, I want to be able to customize my theme based on my own preference.		
Acceptance Criteria: User can choose between light mode and dark mode.		

Title: Support Information	Priority: Medium	Estimate: 1-2 days
User Story: As a user, I want a detailed tutorial on how to use the app, to be able to view frequently asked questions and be able to ask support any questions I may have.		
Acceptance Criteria: User can view - Frequently Asked Questions - Snap Vision Contact Details - Tutorial on how to use the app		

Title: Accessibility Settings	Priority: Medium	Estimate: 1-2 days
User Story: As a user, I want to edit accessibility options on the app.		
Acceptance Criteria: User can toggle haptic feedback. User can enable accessibility mode that prioritises using elevators instead of stairs for indoor navigation.		

2.3.2 Outdoor Navigation

Title: 2D Outdoor Navigation	Priority: High	Estimate: 1-2 weeks
User Story: As a user, I want to select a destination and receive a path to get there so I can navigate around campus easily.		
Acceptance Criteria: Users can search for or tap a destination on the map. Users can use real time text to speech updates for navigation A path is generated and shown on a 2D map. Location updates as the user moves.		

Title: Toggle Navigation Mode	Priority: Medium	Estimate: 2-3 days
User Story: As a user, I want to switch between AR view and 2D map view so I can navigate in different ways based on my preference.		
Acceptance Criteria: A button can switch between views. State is preserved when toggling.		

Title: AR Outdoor Navigation	Priority: High	Estimate: 1-2 weeks
User Story: As a user, I want to select a destination and see an AR arrow that guides me to my destination.		
Acceptance Criteria: Users can search for or tap a destination on the map. Users can use real time text to speech updates for navigation. An AR arrow is generated that guides the user to their destination.		

<table>
<tr><td>Title: Enable Voice and Haptic Navigation</td><td>Priority: Medium</td><td>Estimate: 2-3 days</td></tr>
<tr><td colspan="3">User Story:
As a visually impaired user, I want to get spoken directions and haptic feedback so I can navigate independently.</td></tr>
<tr><td colspan="3">Acceptance Criteria:
Accessibility Mode toggle is available in settings.
When enabled
- App reads out the next direction.
- Subtle vibration is triggered on direction changes (e.g., turn left).</td></tr>
</table>

<table>
<tr><td>Title: Share Navigation Link</td><td>Priority: Medium</td><td>Estimate: 1 week</td></tr>
<tr><td colspan="3">User Story:
As a user, I want to share my current or destination location with a friend so they can find me easily.</td></tr>
<tr><td colspan="3">Acceptance Criteria:
A share location button generates a secure shareable link.
The link will allow another user to view the current user’s location.</td></tr>
</table>

<table>
<tr><td>Title: Navigate to Shared Location</td><td>Priority: Medium</td><td>Estimate: 2-3 days</td></tr>
<tr><td colspan="3">User Story:
As a user, I want to open a shared navigation link so I can find a friend or meet-up spot.</td></tr>
<tr><td colspan="3">Acceptance Criteria:
Tapping a shared link will open the app with a pre-set destination.
App will show the route from the current location.
If the user does not have the app, it takes them to the playstore</td></tr>
</table>

Title: Report Crowd Density on Map	Priority: Medium	Estimate: 2-3 days
User Story: As a user, I want to be able to report crowded areas as well as view crowded areas on the map.		
Acceptance Criteria: User can report crowds in buildings, which will be visible to other users. Users can view crowd markers on the map. Users can update previous crowd reports with new information.		

2.3.3 Indoor Navigation

Title: Manual Indoor Navigation	Priority: High	Estimate: 1-2 weeks
User Story: As a user, I want to select a destination in a building and receive a path to get there so I can navigate around campus easily.		
Acceptance Criteria: Users can scan a QR code to load their position. Users will receive step by step directions that they can follow. User needs to manually mark a step as done as they follow the generated path.		

Title: QR Code Positioning	Priority: Medium	Estimate: 4-5 days
User Story: As a user, I want to scan a QR code at my location so the app knows exactly where I am.		
Acceptance Criteria: QR scanner opens in-app. Scanning sets the user's location on the map.		

2.3.4 Additional Features

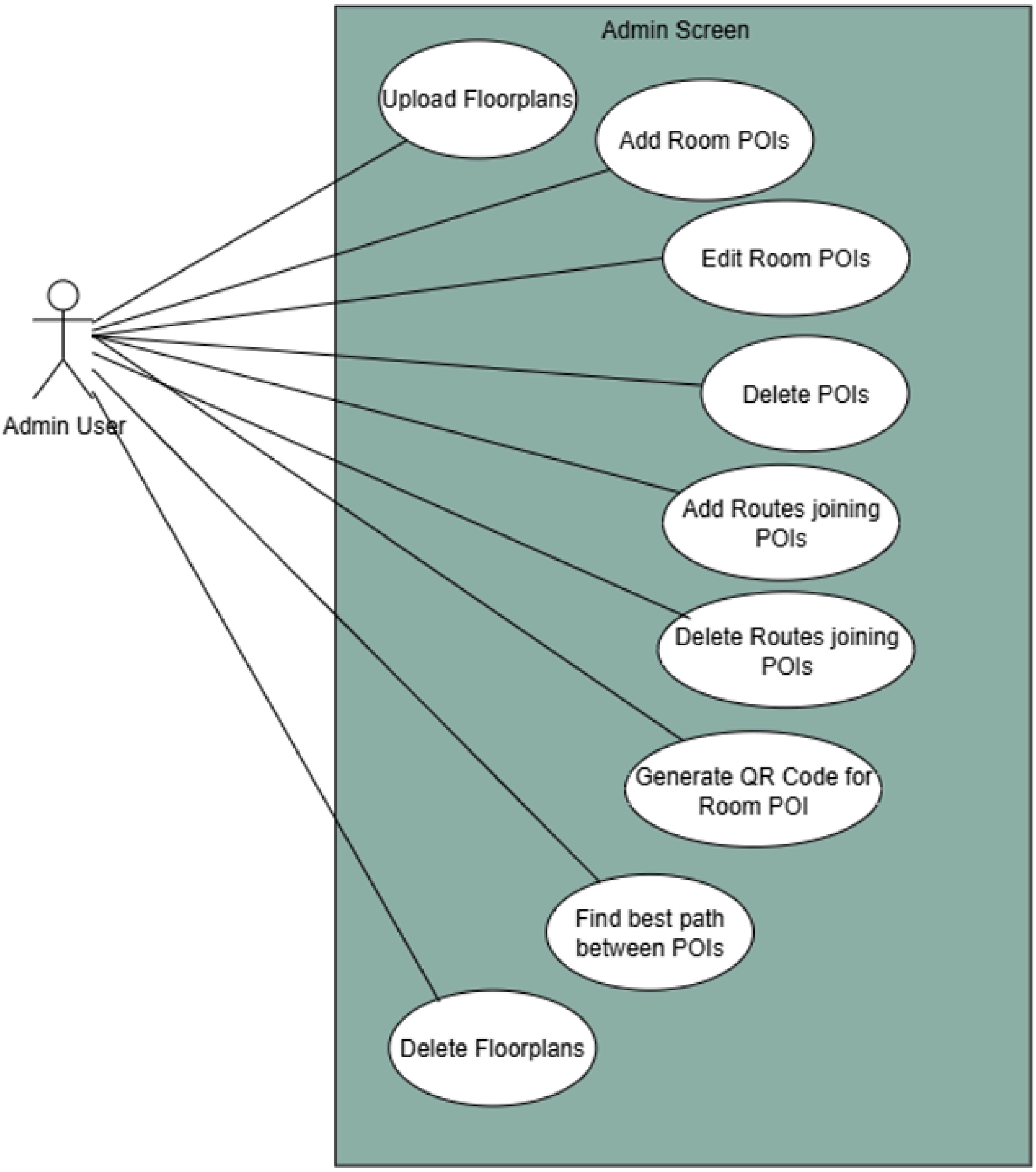
Title: Gamification	Priority: Medium	Estimate: 4-5 days
User Story: As a user, I want to earn badges/points for completing tasks.		
Acceptance Criteria: Badges can be completed to earn points. Points can be used to characterise the app.		

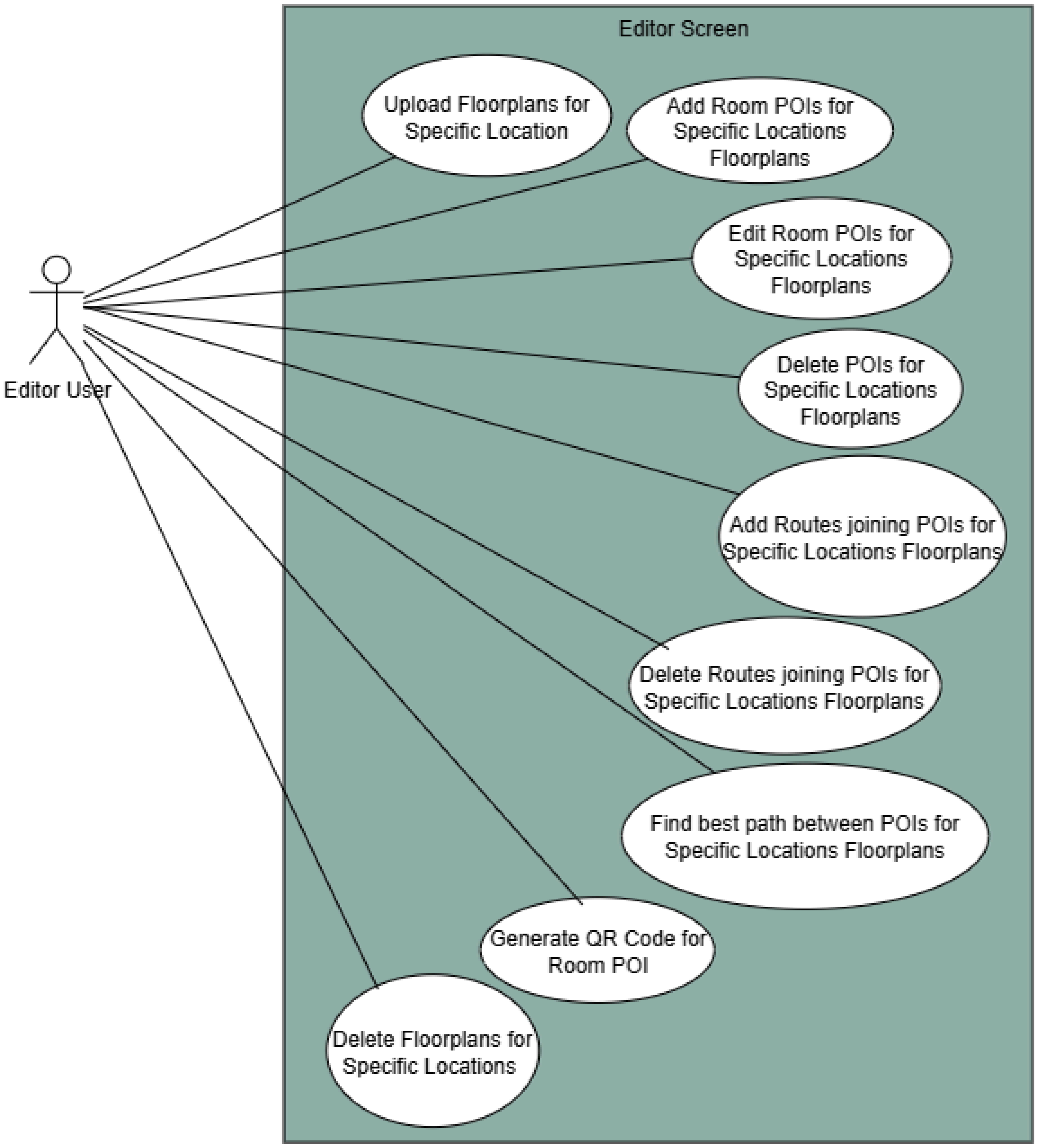
Title: Shop	Priority: Medium	Estimate: 4-5 days
User Story: As a user, I want to spend my points on app customisation.		
Acceptance Criteria: Points can be used to purchase app customisations such as custom icons and themes.		

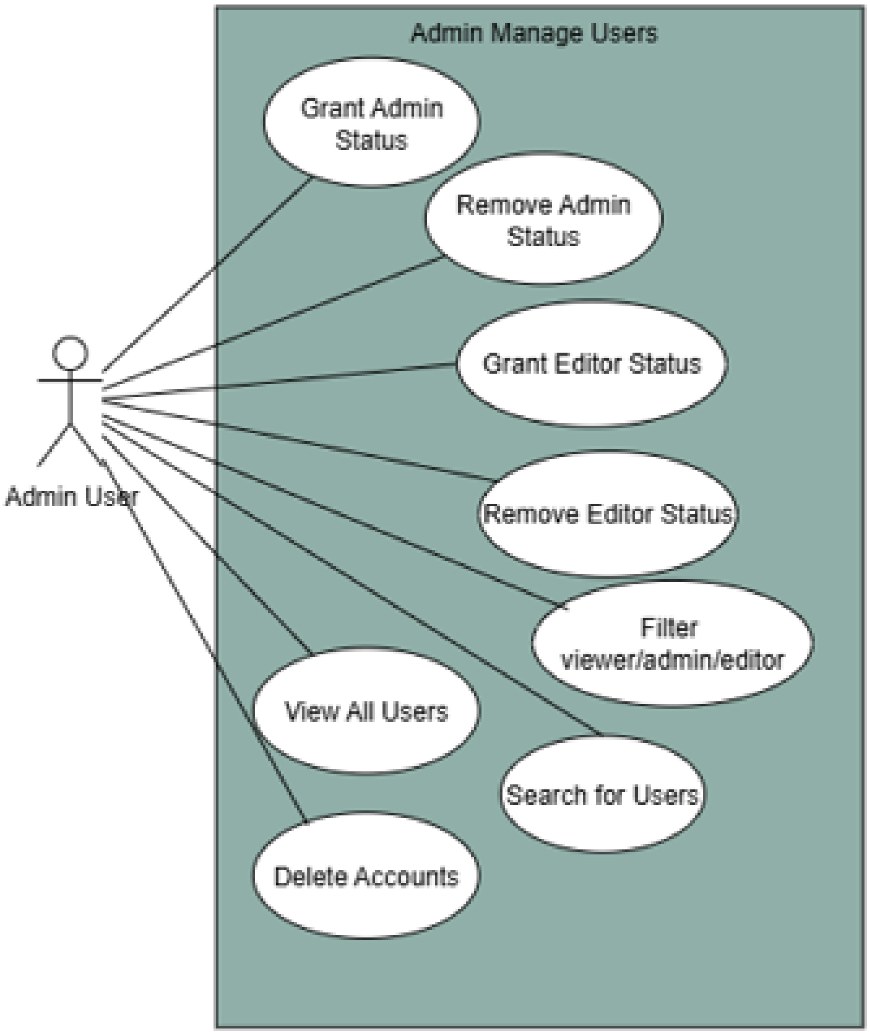
Title: Scheduling	Priority: Medium	Estimate: 4-5 days
User Story: As a user, I want to create a timetable for my classes and be notified when a class is about to start.		
Acceptance Criteria: A timetable UI is available for users to create their own schedule. User receives a notification 10 minutes before a class starts. User can enter the app and choose to navigate to their class.		

3. Use Case Diagrams

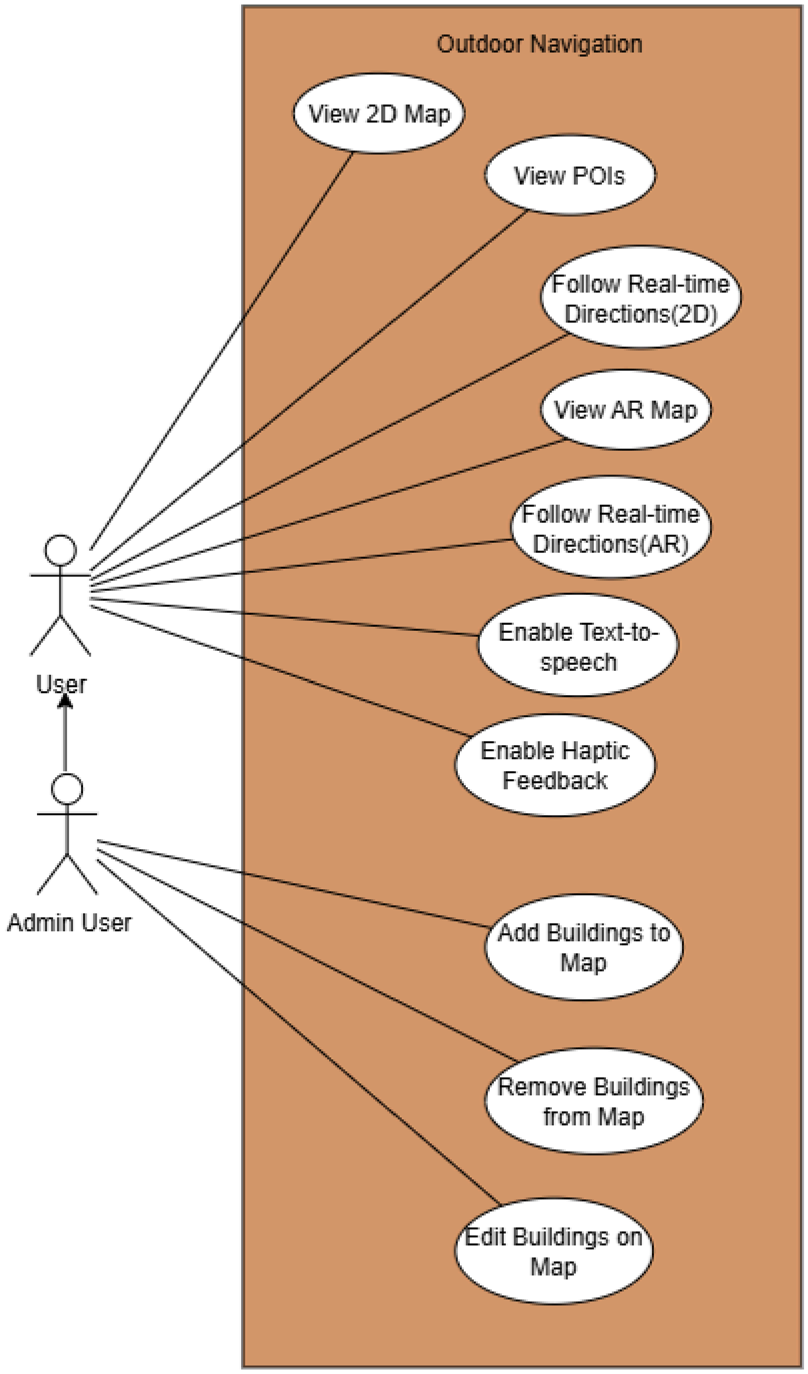
3.1 Admin/Editor Use Cases

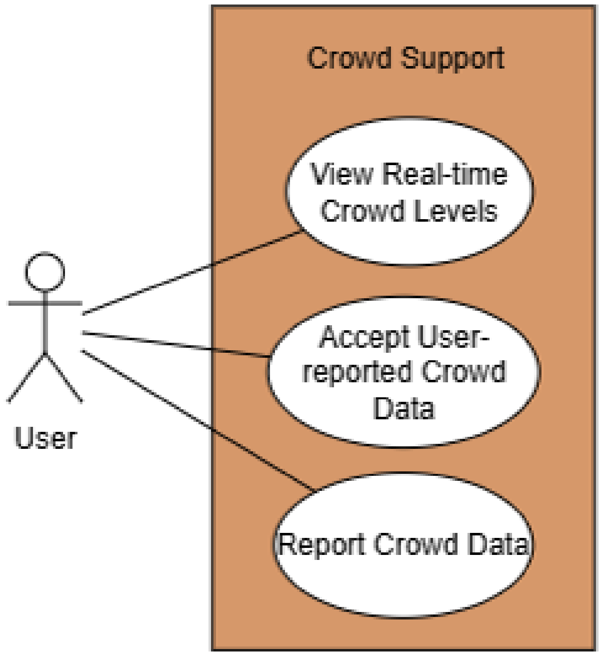


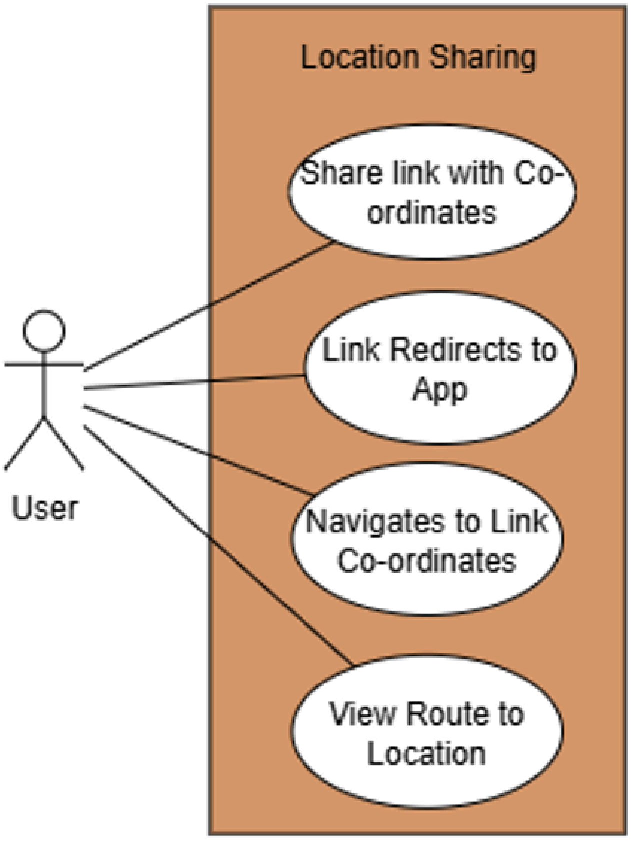

Editor Screen
Editor User
Upload Floorplans for Specific Location
Add Room POIs for Specific Locations Floorplans
Edit Room POIs for Specific Locations Floorplans
Delete POIs for Specific Locations Floorplans
Add Routes joining POIs for Specific Locations Floorplans
Delete Routes joining POIs for Specific Locations Floorplans
Find best path between POIs for Specific Locations Floorplans
Generate QR Code for Room POI
Delete Floorplans for Specific Locations


Admin Manage Users
Grant Admin Status
Remove Admin Status
Grant Editor Status
Remove Editor Status
Filter viewer/admin/editor
View All Users
Search for Users
Delete Accounts
Admin User

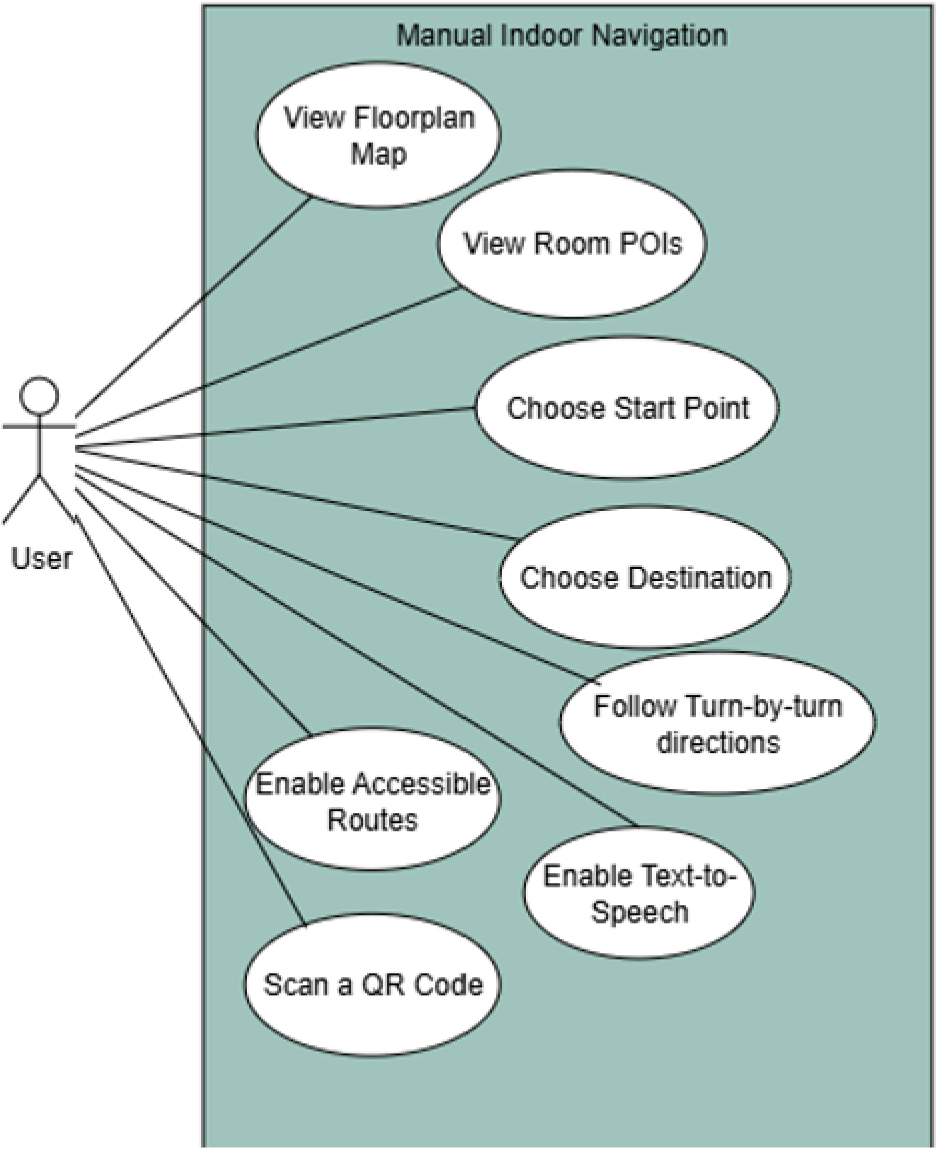
3.2 Outdoor Navigation Use Cases



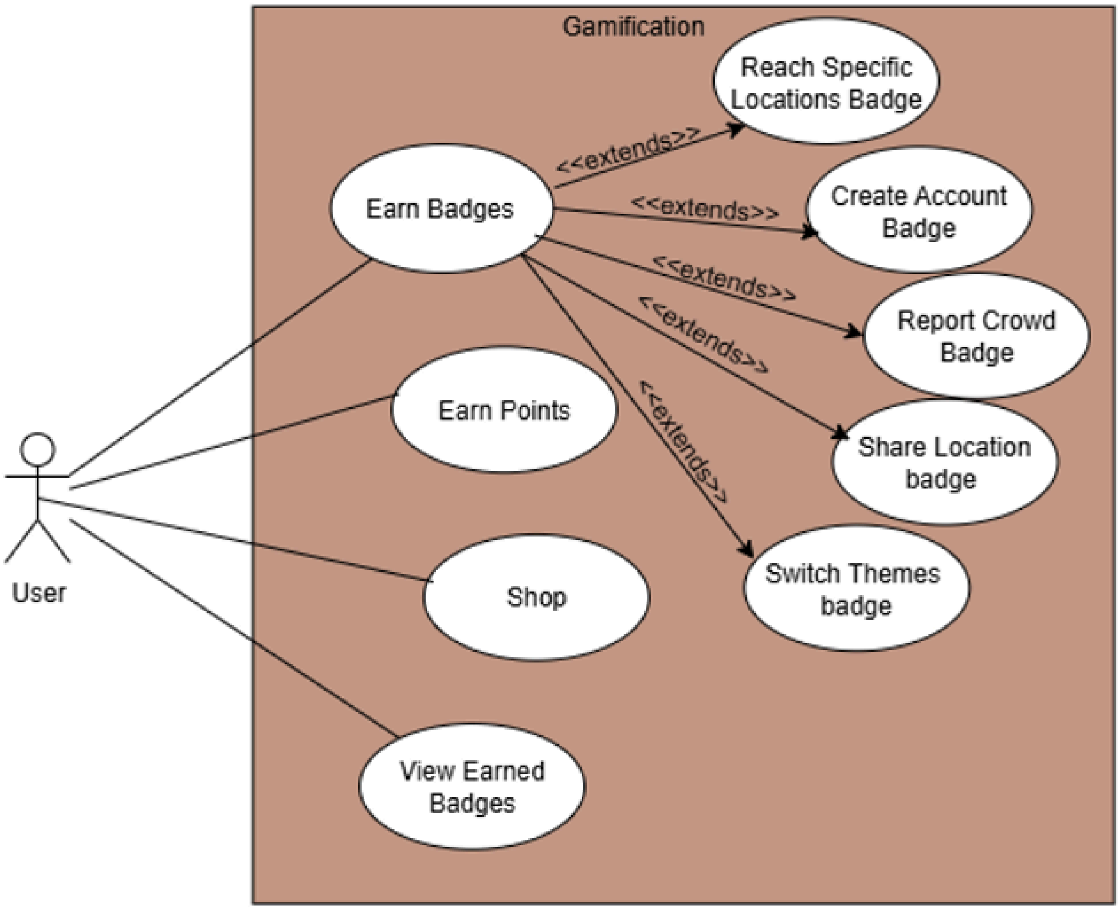

Crowd Support
View Real-time Crowd Levels
Accept User-reported Crowd Data
Report Crowd Data
User

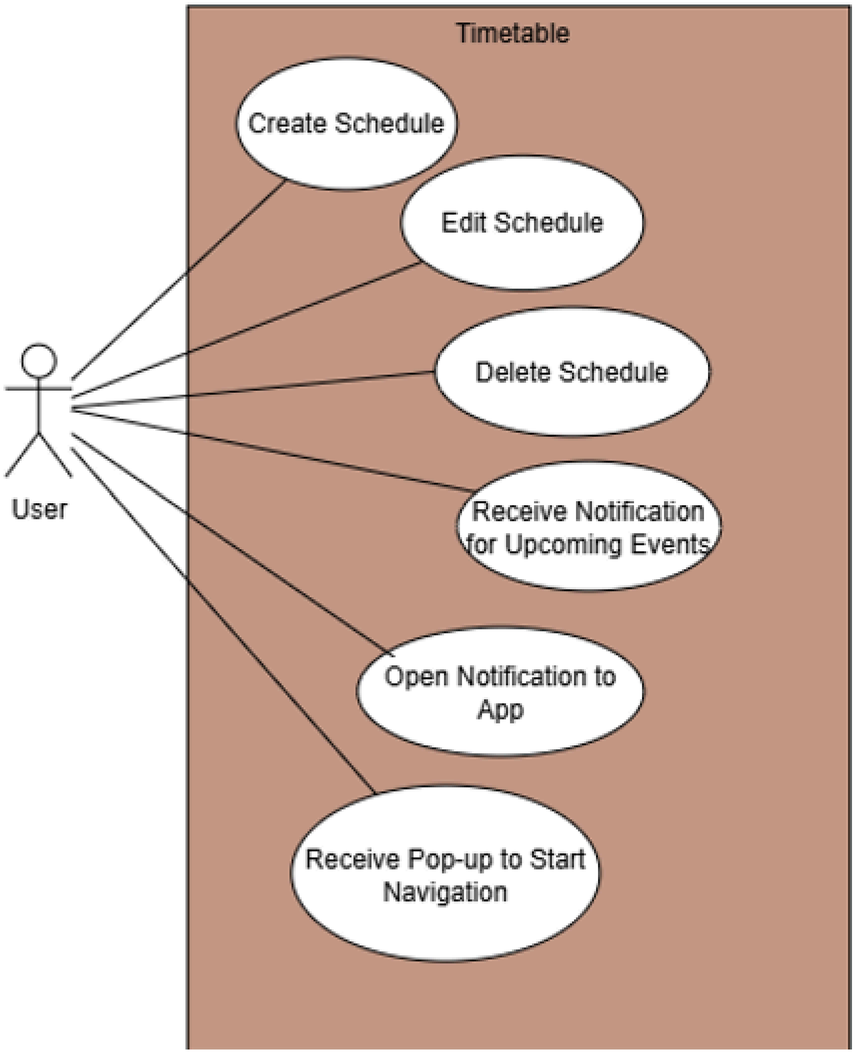

Location Sharing
Share link with Co-ordinates
Link Redirects to App
Navigates to Link Co-ordinates
View Route to Location
User

3.3 Indoor Navigation Use Cases



3.4 Additional Use Cases




Timetable
Create Schedule
Edit Schedule
Delete Schedule
Receive Notification for Upcoming Events
Open Notification to App
Receive Pop-up to Start Navigation
User

5. Functional Requirements

FR1: A user must be able to register on the app

1.1. Users must register with a valid email address and secure password (minimum 8 characters, containing uppercase, lowercase, and numbers)

1.2. System must verify email addresses before account activation

1.3. System must prevent duplicate email registrations

FR2: A user must be able to login with valid credentials

2.1. Users must authenticate using valid email and password credentials

2.2. Admin and editor users will be hardcoded and will have access to exclusive features

2.3. System must provide "Remember Me" functionality

2.4. System must implement password reset functionality via email

FR3: System must support three user roles: Regular User, Editor, and Administrator

3.1. Editors must have location-specific administrative privileges

3.2. Administrators must have full system access across all locations

3.3. System must restrict feature access based on user roles

FR4: System must detect and verify user location for navigation

4.1. System must request and verify location permissions before proceeding

4.2. System must support GPS positioning for outdoor navigation

4.3. System must provide QR code scanning as location fallback method

4.4. System must maintain location accuracy within 2 meters for navigation

FR5: System must support multi-environment navigation

5.1. The user must be able to switch between indoor or outdoor navigation

5.2. System must seamlessly transition between indoor and outdoor navigation modes

5.3. System must support multi-floor building navigation

5.4 System must support multiple location/campus configurations

FR6: A user must be able to search for places to get directions

6.1. Users must be able to search for buildings

6.2. System must provide autocomplete suggestions during search

6.3. System must display search results with destination details and route

FR7: A user must be able to discover points of interest

7.1. System must display categorized POIs on the map interface

7.2. System must show real-time POI status (crowd levels)

FR8: System must calculate optimal routes for users

8.1. System must calculate optimal routes considering distance

8.2. System must recalculate routes automatically when users deviate from planned path

FR9: A user must be able to follow turn-by-turn navigation

9.1. System must provide step-by-step navigation instructions

9.2. System must display remaining distance

9.3. System must provide voice-guided navigation instructions

9.4. System must provide haptic feedback during navigation

FR10: A user must be able to view maps of their area

10.1. Users must access 2D map view with real-time position tracking

10.2. Users must access AR overlay navigation with directional indicators

10.3. System must provide schematic indoor navigation views

10.4. Users must be able to switch between navigation modes seamlessly

10.5. System must display mini-map in AR mode for context

FR11: A user must be able to follow AR navigation directions

11.1. System must overlay directional arrows and instructions in camera view

11.2. System must calibrate AR display based on device orientation and movement

11.3. System must support AR navigation in outdoor environments

11.5. System must handle AR camera permissions and device compatibility

FR12: A user must be able to share their current location

12.1. Users must be able to share current location via shareable links

12.2. Shared links must open destination in app for existing users

12.3. System must validate co-ordinates in shared location links before processing

FR13: A user must be able to view and report crowding levels

13.1. Users must be able to report crowd density levels at locations

13.2. System must display real-time crowd information on map interface

13.3. System must expire crowd reports after 1 hour

FR14: A user must have access to gamified features

14.1. Users must earn badges for completing navigation routes

14.2. Users must earn achievements for exploring new locations

14.3. Users must earn points for completing challenges and reaching destinations

14.4. System must track user statistics (distances traveled, locations visited)

14.5. Users must be able to view achievement progress and unlock conditions

FR15: A user must be able to customize their app experience

15.1. Users must be able to purchase app customizations with earned points

15.2. System must track user inventory

FR16: An admin user must be able to manage building content

16.1. Administrators must be able to upload and manage building floorplans

16.2. Administrators must be able to add, edit, and remove POIs

16.3. Administrators must be able to set up room markers and waypoints

16.4. Administrators must be able to create and modify indoor navigation paths

FR17: An admin user must be able to manage user accounts

17.1. Administrators must be able to view and search user accounts

17.2. Administrators must be able to grant/revoke user privileges

17.3. Administrators must be able to manage location-specific editor permissions

17.4. Administrators must be able to handle user account deletion requests

FR18: An admin user must be able to configure indoor positioning

18.1. Administrators must be able to set up QR codes

FR19: A user must be able to access accessibility features

19.1. Users must have access to text-to-speech navigation instructions

19.2. Users must have access to haptic feedback for navigation guidance

19.3. Users must have access to wheelchair-friendly indoor routes

19.4. System must comply with WCAG accessibility guidelines

19.5. Users must be able to configure accessibility preferences

FR20: A user must be able to schedule a trip to a certain location

20.1. Users must be able to create a timetable by setting the day, time and location

20.2. Users must be notified 10 minutes before the scheduled event

20.3. Users must have the option to navigate to their destination by opening the notification and going to the map screen

6. Service Contracts

6.1. Authentication Service

Description: Handles user registration, login, and role-based access control (e.g., admin/user). Roles are stored in Firestore under userInformation.

Inputs:

- email (string) - User’s registered email address.
- password (string) - User’s password.

Outputs:

- access_token (string) - Token used for authenticated requests.
- user_role (string) - The role assigned to the user (admin, editor, user).

Usage/Interaction Rules:

Use @react-native-firebase/auth:

- signInWithEmailAndPassword(email, password).
- onAuthStateChanged(callback) to track session.
- getIdToken(forceRefresh) for secure calls to backend.

6.2. User Role and RBAC Service

Description: Resolves the caller's role and allowed locations to enforce client-side and Firestore rule–based authorization.

Inputs:

- uid (string) – Current authenticated user ID.

Outputs:

- role (enum) - Effective role.
- editorLocations (string[]) - Location IDs the editor can manage.

Usage/Interaction Rules:

- Read Firestore at userInformation/{uid}
- Enforce UI guards: admins have full access, editors are restricted to editorLocations and users are read-only.

6.3. Floorplan Upload Service

Description: Uploads a floorplan image to Firebase Storage and writes floorplan metadata to Firestore.

Inputs:

- locationId (string) - Location scope.
- buildingID (string) - Target building.
- floorNumber (number) - Floor identifier.
- file (blob) - Floorplan image.
- uploadedBy (string) - UID performing the upload.

Outputs:

- downloadURL (string) - Public/authorized URL for the floorplan.
- floorplanDoc (object) - Firestore document snapshot (metadata).

Usage/Interaction Rules:

- Storage path: floorplans/{locationId}/{buildingId}/{floorNumber}.png.
- Firestore path: locations/{locationId}/{buildingId}/floorplans/{floorNumber} with fields buildingId, floorLabel, downloadURL, timestamp, uploadedBy.
- Validate image size before upload.

6.4. POI Management Service

Description: Creates/updates/deleted POIs within a location and (optionally) a building.

Inputs:

- locationId (string) - Location scope.
- type (enum: building | room) - POI category.
- payload (object) - { name, centroid, tags }.
- poiId (string) - For updates and deletes.

Outputs:

- Poi_document (object) - Created/updated POI doc.
- status (string) - Error or success.

Usage/Interaction Rules:

- Buildings path: locations/{locationId}/buildingPOIs/{buildingId}
- Rooms path: locations/{locationId}/roomPOIs/{roomId}
- Reads are public to authenticated users and are used for navigation.
- Admins have access to all locations and are able to create POIs on the map.
- Editors have access to certain locations and cannot add POIs to the map.

6.5. PathPOI Service

Description: Manages indoor path graph edges connecting rooms for routing.

Inputs:

- locationId (string) - Location scope.
- edge (object) - {fromRoomId, toRoomId, weight, level, isAccessible}.
- pathId (string) - For updates/deletes.

Outputs:

- path_document (object) - Created/updated path edge.
- status (string) - Error or success.

Usage/Interaction Rules:

- Firestore path: locations/{locationId}/pathPOIs/{pathId}.
- Validate fromRoomId/toRoomId exist in roomPOIs.
- Admins have access to all locations.
- Editors have access to certain locations.
- Keep graph consistent: prevent duplicate edges and enforce bidirectional edges.

6.6. Live Location Service

Description: Lets a user share their current location with chosen recipients.

Inputs:

- ownerUid (string) - User sharing their location.
- recipients (string[]) - UIDs.
- location (object) – { lat, lng } or { x, y, buildingId, floorNumber }.
- expiresAt (datetime).

Outputs:

- shareID (string) - Related to the shareable link.
- status (string) - Error or success.

Usage/Interaction Rules:

- Firestore: deep links are used to generate a shareable link.
- Privacy: only ownerUid and recipients can access the link.

6.7. Recently Visited Service

Description: Tracks and surfaces the user’s last visited POIs for quick access and personalization.

Inputs:

- Uid (string) - Current user.
- poiId (string) - The POI that was visited recently.

Outputs:

- recentList (array) - Ordered recent POIs with { poiId,name, lastVisited }.

Usage/Interaction Rules:

- Firestore path: recentlyVisited/{uuid}.
- Client write is capped at a certain number to keep information fresh.
- Any authenticated user may read their own list only.

6.8. Outdoor Route Generation

Description: Generates outdoor walking routes via a Node/Express backend that calls MapTiler/OSM/ORS.

Inputs:

- origin (object) - { lat, long }.
- destination (object) - {lat, long}.
- idToken (string) - Firebase Auth token for authenticated access.

Outputs:

- route (GeoJSON LineString) - Full path polyline.
- steps (array) - Turn-by-turn instructions.

Usage/Interaction Rules:

- Endpoint: get/api/directions (from Open Route Services).
- Backend verifies tokens and queries the API to generate a route.

6.9. Timetable Service

Description: Lets a user create a timetable on the app and schedules notifications 10 minutes before a class starts

Inputs:

- uid (string) - current user
- course (object) - {course, venue, startTime, endTime, day, buildingId, buildingName, locationId)
- autoNavigationEnabled (boolean) - user preference for scheduled notifications
- entries (array) - array of timetable entries for the authenticated user
- idToken (string) - Firebase Auth token for authenticated access.

Outputs:

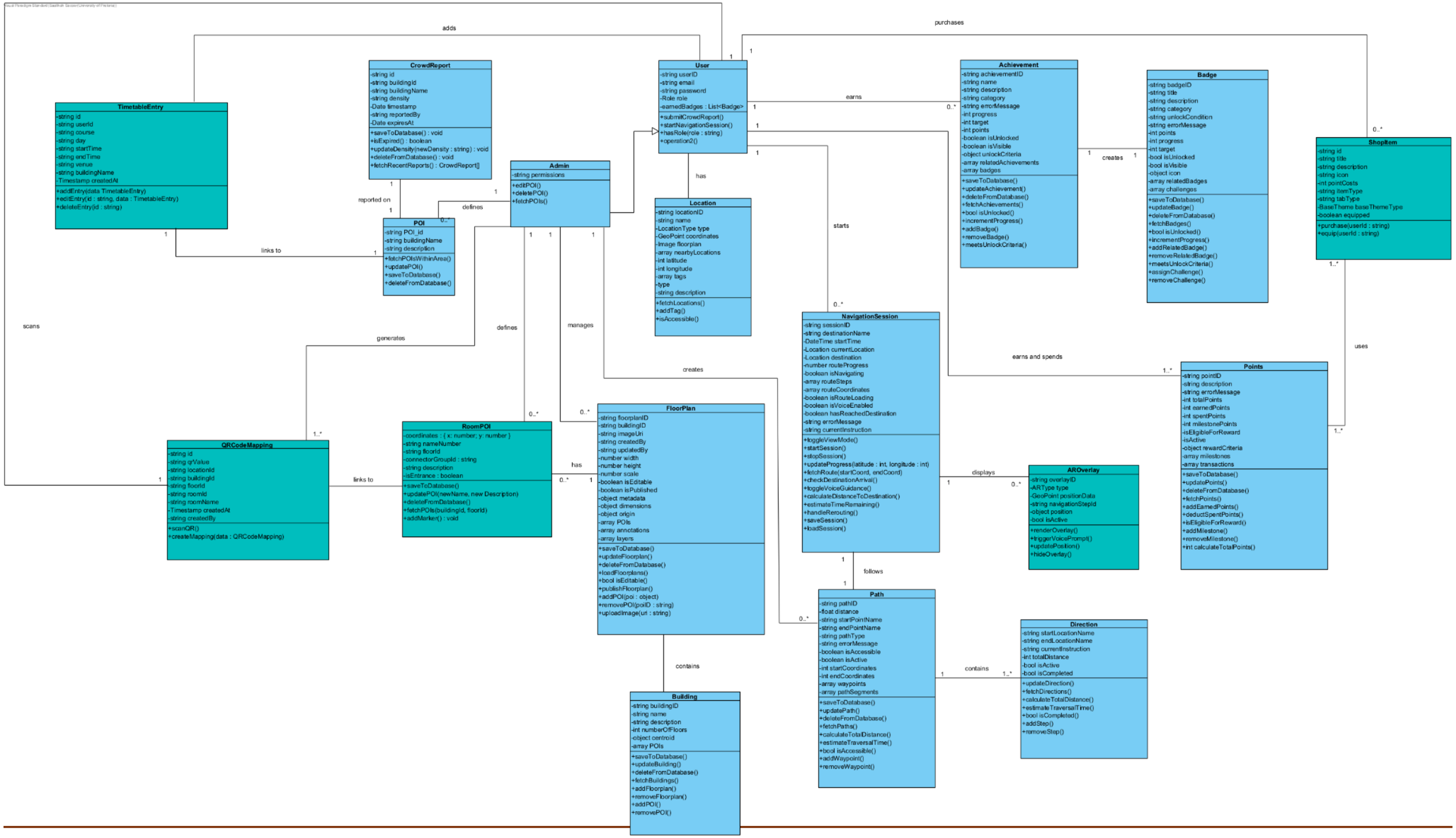
- entryId (string) - unique identifier for the created/updated entry
- notificationIds (array) - array of notification IDs for tracking
- scheduledTimes (array) - array of scheduled notification timestamps
- navigationData (object) - contains course, venue, startTime, buildingInformation

Usage/Interaction Rules:

- Firebase path for timetable entries: /timetables/{documentId}
- Firebase path for notifications: /userFCMTokens/{userId}
- Any authenticated user can only see their own timetable

7. Domain Model

Link to Domain Model PDF



8. Deployment Model

Component	Technology	Hosting
Mobile App	React Native + ARCore	Distributed via Google Play Store
Backend	Firebase App Hosting	Deployed via Firebase
Authentication	Firebase Authentication	Firebase Authentication Service
Data Storage (POIs, Maps)	Firebase Storage	Firebase
CI/CD Pipeline	GitHub Actions	Will auto-deploy to Firebase on main branch

9. Live Deployed System

The Snap Vision system will be deployed using a serverless architecture and will be accessible on Android devices via an APK. The backend and data services are hosted entirely in the cloud to allow for real-time navigation and remote content updates.