7009ICT

AR Application Design Document

Griffith GO: *A Gamified AR Campus Navigation Experience*

Author:

Name	Student Number	Role in Team
Jeeyoung Park	S5378253	Project Manager
Josh Yu	S5401891	Member
Yeongjoo Kim	S5413681	Member

Workshop: Wednesday, 10AM~12PM, Shamus Smith

Revision History

Date	Description	Author
20/Mar	Updated part 2 section (Application Focus and Scope)	Josh Yu
21/Mar	Filled 1. Introduction	Jeeyoung Park
21/Mar	Updated part 3 section	Yeongjoo Kim
29/Mar	Updated part 4 section (Technology and issues)	Josh Yu
30/Mar	Revised 3 and 4. added conditions, re-organise, tone polishing created 5.3. Interface Design	Jeeyoung Park
2/April	Updated part 6 draft	Yeongjoo Kim
2/Apr	Added 5.1. Required Device Environment	Jeeyoung Park
9/Apr	Revised 3, 5.2	Yeongjoo Kim
9/Apr	Revised part 2&4 section + references Updated summary and reflection on meeting records Adjust line spacing of the document	Josh Yu
10/Apr	Revised section 1, 2, 5 Integrated/re-organized storyboards. Overall tone polishing, quality check. Created index + Organized References.	Jeeyoung Park

INDEX

1 Introduction

1.1 Purpose of this document

Overview of the Document

This document covers the design and development of an augmented reality (AR) application to support wayfinding and enhance user engagement on the Griffith campus. It serves as a key reference document for the AR application's features, user experience, and technical specifications throughout the project. Additionally, by detailing the project's technical/non-technical requirements, design principles, and implementation plan (including storyboards), it provides a comprehensive blueprint for the actual application development.

Objectives and Goals

This document aims to clearly define the design of the AR application and establish the foundation for its implementation. It outlines core objectives, such as enhancing the wayfinding experience through an immersive AR environment and fostering engagement through gamification.

Enhance Wayfinding Efficiency:

According to research, AR-integrated navigation systems significantly reduce both the time taken to reach a destination and users' mental workload compared to other navigational approaches (Qiu et al., 2023). Leveraging this advantage, the Nathan Campus AR Wayfinder will provide temporal and emotional support for busy university students and staff with hectic academic schedules.

Boost Engagement via Gamification:

The application incorporates AR badges (e.g., "Building Master") and an XP system to transform wayfinding into an enjoyable challenge. Additionally, inspired by Pokémon GO—which achieved over 60% 3-day retention—the app includes a wildlife animal collection feature to sustain user interest and encourage active participation (Pokémon GO Statistics Report 2016 EDITION, n.d.).

Furthermore, this document will serve as a reference for evaluating the project′s success in achieving both technical goals and a positive user experience.

1.2 Scope of this document

Covered Topics

This document will address the key elements involved in the project, including an introduction to AR technology and its potential users, along with their backgrounds. It will also examine the technology stack, detailing the hardware and software tools used for AR development, including anticipated challenges. Additionally, the document will provide an overview of the system architecture, outlining project requirements, reused components, and their interactions. Finally, it will cover the application design, explaining requirements, system architecture (including reused components), user interfaces, and storyboards in detail.

Out of Scope

This document will not explore aspects beyond the direct scope of the project, such as detailed marketing strategies or external business considerations unrelated to the AR application's functionality. It also excludes specific platform or design choices that fall outside the defined hardware and tools, such as desktop applications, iOS-specific features, or head-mounted devices.

Limitations

Several constraints may affect the project, including time and resource limitations, which could restrict the depth of features developed and tested. Technical limitations tied to specific tools and devices may also impact functionality or user experience. Furthermore, due to limited optimization time, performance issues related to stability or functionality may not be fully resolved within the project's scope.

2 AR Application Overview

Application Focus

This AR campus navigation app focuses on helping students and visitors at Griffith campus easily and enjoyably find their way to destination buildings. The app utilizes augmented reality (AR) to create an intuitive and interactive wayfinding experience by displaying virtual arrow objects. Users can pin their current location using GPS or scan a QR code to set their starting point, then select desired destinations such as the library, campus heart, or specific buildings. The system displays two route options along with estimated travel time and automatically suggests alternative routes if users take a wrong turn. Upon reaching the destination, the system notifies users of navigation completion through intuitive visual effects and notification sounds.

Moreover, to make the app more enjoyable and user-engaging, it adopted a badge collection system and interactive virtual animals. While using the navigation, users may see animals like koalas or possums in AR, with the spatial environment of Griffith Nathan campus.

With the wildlife animals, the badge system also increases user retention. For example, visiting the N79 building ten times will unlock a special badge. If a badge is still locked, the app gives a hint to help the user understand how to unlock it. This increases user engagement and motivation.

Scope of Design

Inclusions

The core AR wayfinding system utilizes a live camera feed to display interactive 3D path arrows and collectible building badges, incorporating gamification elements to increase engagement. To ensure accurate positioning, the app implements QR-based location anchoring as a supplement to GPS, particularly useful in dense campus areas where GPS signals may be unreliable. Users can customize their experience through various options, including route preferences that allow selection between scenic or fastest paths, and the ability to toggle AR effects, voice guidance, and notifications to optimize battery life and personal preferences. The application integrates with a comprehensive campus map database containing building coordinates and notable landmarks, while a Firebase backend supports user profile management, including synchronization of earned badges and XP. Additionally, the system provides immediate feedback through visual and audio cues when users deviate from their designated paths, ensuring timely correction and maintaining navigation efficiency.

Exclusions

The system does not support detailed indoor navigation, such as locating specific rooms inside buildings. It also excludes public transport integration, meaning it only offers walking navigation within the campus and does not provide bus or train schedules. Furthermore, the system lacks real-time crowd detection, so it cannot track campus congestion levels or crowd density.

3 People

Target User Demographics

The primary users of the AR wayfinding application are expected to be university students (aged 18–30), both domestic and international, who form the majority of on-campus traffic. Secondary users include staff, researchers, and on-campus workers (aged 26–50), as well as visiting professors and contractors (50+). The app caters to all genders without demographic skew. Geographically, it targets Nathan Campus students and staff, along with new visitors (e.g., conference attendees) unfamiliar with the campus layout. Given Griffith University's highly diverse population, the user base spans various cultural backgrounds, including international students from Asia, Europe, and the Americas, as well as local Australians.

3.1 People: Background

Skill/Technology Level

The app is designed for beginner to intermediate users, balancing accessibility and functionality. While younger students are generally tech-savvy, older staff or contractors may have limited AR experience, necessitating an intuitive interface. To accommodate this, the app will include introductory tutorials for AR newcomers, while offering optional advanced features (e.g., timetable synchronization) for power users. This tiered approach ensures broad usability without overwhelming less experienced users.

Interest & Preferences

User needs vary by role: Students prioritize gamified elements (e.g., badges, wildlife collection) and social features (leaderboards), alongside time-efficient routes to classes. Staff and workers focus on practical navigation (shortest paths) and accessibility settings (adjustable text size/VFX toggles). Meanwhile, contractors require real-time safety updates, such as alerts for construction zones. These preferences inform tailored design solutions for each group.

Accessibility and Localization

User needs vary by role: Students prioritize gamified elements (e.g., badges, wildlife collection) and social features (leaderboards), alongside time-efficient routes to classes. Staff and workers focus on practical navigation (shortest paths) and accessibility settings (adjustable text size/VFX toggles). Meanwhile, contractors require real-time safety updates, such as alerts for construction zones. These preferences inform tailored design solutions for each group.

Concurrent Users

During peak times (e.g., class changes), the app anticipates 50–200+ active users simultaneously. To handle this load, the infrastructure will leverage cloud-based AR anchoring to prevent overlapping virtual guides and include an offline mode for areas with poor connectivity (e.g., construction sites). This ensures reliability across diverse campus environments.

4 Technology

The AR campus navigation application will be developed using the following technologies, each with distinct advantages and limitations.

Unity and C#

The application employs Unity (version 6000.0.40f1) for constructing 3D scenes and implementing AR functionalities. C# serves as the primary programming language for developing core application logic, including navigation arrows, dynamic route updates, and user interaction systems. The synergy between Unity and C# offers robust development capabilities while maintaining accessibility. Furthermore, Unity's cross-platform compatibility (supporting both Android and iOS) ensures scalability for future expansions. Extensive online documentation and community resources facilitate efficient troubleshooting during development (Unity AR+GPS Location Docs, 2020).

However, certain limitations exist. Some advanced Unity features require third-party plugins, necessitating additional learning efforts. Additionally, AR functionality cannot be fully tested within the Unity Editor alone; real-device testing is essential for validating camera tracking accuracy and AR object placement. This constraint introduces inefficiencies in the testing phase.

AR Foundation

The application integrates AR Foundation, a Unity framework enabling digital object placement in real-world environments. Utilizing a Plane Manager, the system detects horizontal surfaces (e.g., walkways) to overlay navigational cues, such as directional arrows, onto physical paths. A key advantage of AR Foundation is its dual-platform support (Android and iOS), broadening accessibility.

Nevertheless, hardware limitations pose challenges. As noted by *Google for Developers (2024)*, ARCore (the underlying Android AR platform) is incompatible with older or low-performance devices, potentially leading to suboptimal tracking or reduced frame rates.

Geospatial Positioning: GPS and QR Codes

The application relies on GPS for initial user localization, though signal inaccuracies may occur in urban canyons or densely populated areas. To mitigate this, the Geospatial API

(Saraswathi et al., 2023) could be implemented, fusing GPS data with visual-inertial odometry for enhanced precision in complex environments.

QR codes provide an auxiliary localization method. By scanning campus-located codes, users manually calibrate their starting position, improving navigation reliability when GPS signals are unreliable. While this hybrid approach enhances accuracy, it requires user intervention to locate and scan QR codes.

Android Device

Due to resource constraints (e.g., lack of Apple development tools like Xcode or macOS hardware), the current implementation targets Android exclusively. This decision aligns with the development team' s existing Android device inventory, streamlining testing procedures.

However, Android' s fragmented ecosystem introduces variability in performance. Disparities in camera quality, display resolution, and computational power across devices may affect AR rendering consistency. Rigorous testing across multiple device specifications is planned to ensure broad compatibility.

System architecture

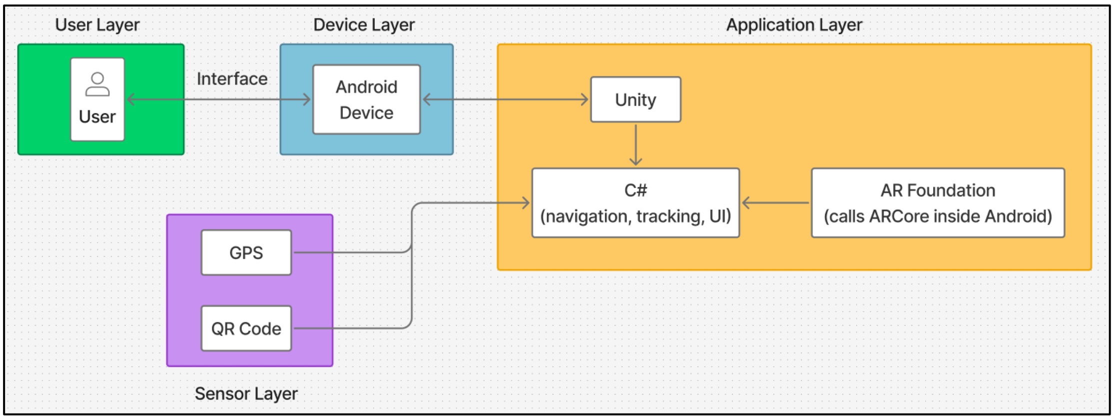

Figure 1. System Architecture of the AR Campus Navigation Application

The system adopts a layered architecture:

The User Layer (green) represents the user interacting with the system by tapping the screen or scanning a QR code. The user provides inputs such as location or destination and receives visual feedback (e.g. AR arrows or animal objects) from the application through the device screen.

The Device Layer (blue) represents the Android device, which handles both user interactions and sensor input. It communicates with the application layer to send input data and receive rendered AR content for display.

The Application Layer (orange) contains the main logic developed in Unity. It uses C# scripts to process user actions, handle navigation, and manage AR rendering through AR Foundation. AR Foundation, in turn, communicates with ARCore, which runs on the Android system.

The Sensor Layer (purple) consists of sensors such as GPS, the phone's camera, and QR code scanning. These sensors provide essential location and environmental data, which is passed to the device and then to the application layer for processing.

4.1 Issues

GPS Signal Instability

Although the application is designed for outdoor use, GPS signal reliability may be compromised in specific areas, such as between buildings or under dense tree coverage. Signal instability can lead to positional inaccuracies and erroneous navigation guidance, adversely affecting both system precision and user experience.

Environmental Dependencies of AR

AR Foundation's performance is highly dependent on optimal environmental conditions, including sufficient lighting and textured surfaces. Even in outdoor settings, suboptimal conditions—such as overcast weather, low ambient light, or flat/reflective ground surfaces—may impair plane detection. Consequently, AR elements (e.g., directional arrows) may exhibit instability, including floating, jittering, or misalignment.

Furthermore, empirical studies indicate that device orientation significantly impacts AR navigation accuracy. Maintaining a smartphone tilt angle between 90° and 100° (near-vertical) yields superior tracking performance compared to horizontal positioning (Fajrianti et al., 2023).

Device Fragmentation

Not all mobile devices support ARCore, Google's AR platform. Even among compatible devices, performance limitations on older or lower-end hardware—including thermal throttling, processing delays, or application crashes—may degrade stability. Establishing minimum device specifications and conducting rigorous cross-device testing are essential to ensure consistent functionality.

5 Design

5.1 Requirements

Required Device Environment

Software Requirements

This application requires Android version 8.0 (Oreo) or later to support ARCore functionality for augmented reality features. Additionally, the latest version of Google Play Services is required for Firebase integration. For optimal performance, Android 11 or higher is recommended (Google for Developers, 2024).

Hardware Requirements

The minimum hardware specifications include a Snapdragon 670 (64-bit ARM) processor, at least 4GB of RAM for smooth AR rendering, and a minimum of 2GB storage space for the app and cached data. Required sensors include an accelerometer, gyroscope, and magnetometer for AR tracking, along with GPS for location-based navigation. A rear-facing camera with autofocus capability is also required, with a minimum resolution of 720p to support AR object recognition (Google for Developers, 2024).

Network Requirements

Stable Wi-Fi or 4G LTE connection is required for online features including map loading, account synchronization, and environment recognition.

ARCore-Specific Constraints

Devices with outdated GPU drivers are incompatible with ARCore and cannot run this application. Additionally, environmental conditions affect AR performance - adequate ambient lighting is required for stable tracking. Highly reflective or shiny surfaces should be avoided as they may interfere with AR surface detection.

Testing Recommendations:

Real-world testing is necessary to ensure application reliability. For instance, GPS accuracy should be verified in various campus locations (e.g., between buildings), and AR recognition stability should be validated under different lighting conditions (sunny vs. cloudy days). Additionally, tests should monitor battery consumption and device overheating during extended AR application usage exceeding 10 minutes.

System Requirements - Functional

1. Onboarding & Setup

The application shall request and manage user permissions, including location and camera access, upon first launch. Users will set a starting point through manual map selection, QR code scanning at designated campus locations, or automatic GPS detection (default). Additionally, the app will provide a searchable map of campus buildings for destination selection. Once a destination is chosen, the system will display route options (shortest or scenic) along with estimated travel time.

2. Navigation

The app shall render AR arrows aligned with the user's real-world path using AR Foundation. Users will earn XP for following correct routes and collecting in-path arrows. If a deviation from the route is detected, the app will trigger visual/sound alerts (e.g., humorous character popup) and automatically recalculate and display alternative paths. Upon arrival at the destination, the system will award a building-specific badge and notify the user.

3. Exploration & Rewards

Users will be able to discover and scan AR wild animals (e.g., koalas) to unlock encyclopedia entries and earn Animal Badges. The app will maintain a user profile displaying collected badges, XP, optional leaderboard rankings, and wildlife encyclopedia progress. Additionally, users can toggle AR effects (visual/sound) and calibrate AR alignment in the Settings menu.

4. Backend & Integration

Firebase will be used to sync user achievements (badges and XP) across devices.

System Requirements – Non-Functional

1. Usability

First-time users must complete onboarding within 90 seconds or less. AR arrows must align with real-world paths with less than 1 meter of positional error.

2. Performance

AR rendering (arrows) must maintain at least 30 FPS on supported Android devices. Route recalculation must occur within 2 seconds or less.

3. Compatibility

The app will support ARCore-enabled Android devices.

4. Accessibility

The application must comply with WCAG 2.1 AA standards, including sound guidance and high-contrast UI (W3C, 2023). Text elements must support English and three additional major languages (Mandarin, Hindi, Spanish).

5. Security & Privacy

User location data must be anonymized. QR codes must contain only non-sensitive campus location IDs.

6. Scalability

The backend must support at least 200 concurrent users during peak hours.

5.2 System Architecture

The AR campus navigation application is built on a modular system architecture that supports location-based guidance, real-time AR rendering, and gamified interaction. The application is developed using Unity and C# for Android devices and leverages ARCore through Unity's AR Foundation framework.

Core Components

1. User Interface Layer

This layer manages all user interactions including navigation setup, AR view, and profile display. It connects directly to gamified features such as XP tracking, badge collections, and the AR animal encyclopedia.

2. Navigation Module

This module is responsible for calculating optimal walking routes and detecting user deviation. It integrates GPS data and QR code input to define the starting point and continuously tracks movement to guide users along the correct path. It also triggers feedback when the user strays off route, supporting features such as wrong-path alerts (structural gamification).

3. AR Visualization Module

This module renders real-time AR arrows or collectible items in the user's environment using Unity AR Foundation. This module displays bonus AR arrows (content-based gamification) and handles environmental cues, including feedback animations and sound effects.

4. Gamification Module

This module manages user progress, experience points, badge unlocking, and animal discovery. It ensures structural elements like XP and achievements are tracked and stored. It also enables content-driven interaction by placing AR animals in specific locations and updating the in-app encyclopedia upon discovery.

5. Sensor and Input Layer

This layer collects data from the device's GPS, motion sensors, and camera. QR code scanning is supported for precise location initialization, improving the accuracy of both AR overlays and navigation paths.

6. Backend Services (Optional)

Backend Services support user login, save progress data, and synchronization of gamification data (e.g., XP, badges). This ensures persistent user profiles and future scalability for multi-device use.

This architecture supports both traditional AR functionality and advanced gamified features. Each module contributes to either structural game mechanics (e.g., XP, badges) or content-based engagement (e.g., exploration, animal discovery), resulting in a cohesive, interactive campus navigation experience.

5.3 Interface Design

Design System

1. Color Palette

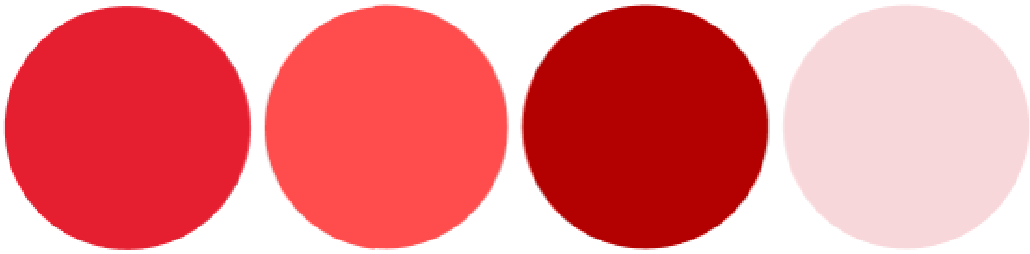

Figure 2. Color Palette for Primary and Secondary Color Variations

Role	HEX	Usage
Primary	#E51F30	Main buttons, badges, key actions
Primary Light	#FF4D4D	Interactive states (hover/active)
Primary Dark	#B30000	Pressed states, accents
Secondary	#F8D7DA	Backgrounds, subtle highlights
Accent	#5CB85C	Success states, correct path indicators
Neutral	#333333	Text, icons
Neutral Light	#F5F5F5	App background, card surfaces

Table 1. Role and HEX Values Related to Color Variations

2. Typography

Type	Size	Weight	Usage
Heading 1	24px	Bold	Screen titles
Heading 2	20px	Semi-bold	Section headers
Body	16px	Regular	Primary text
Caption	14px	Light	Secondary text, hints
Button	16px	Medium	All buttons

Table 2. Proposed Typography for UI Design

Font Family:

- **Primary: Roboto** (Android system font)
- Fallback: Sans-serif

Key Screens Definition

The application interface consists of five key screens designed to facilitate intuitive navigation and user engagement.

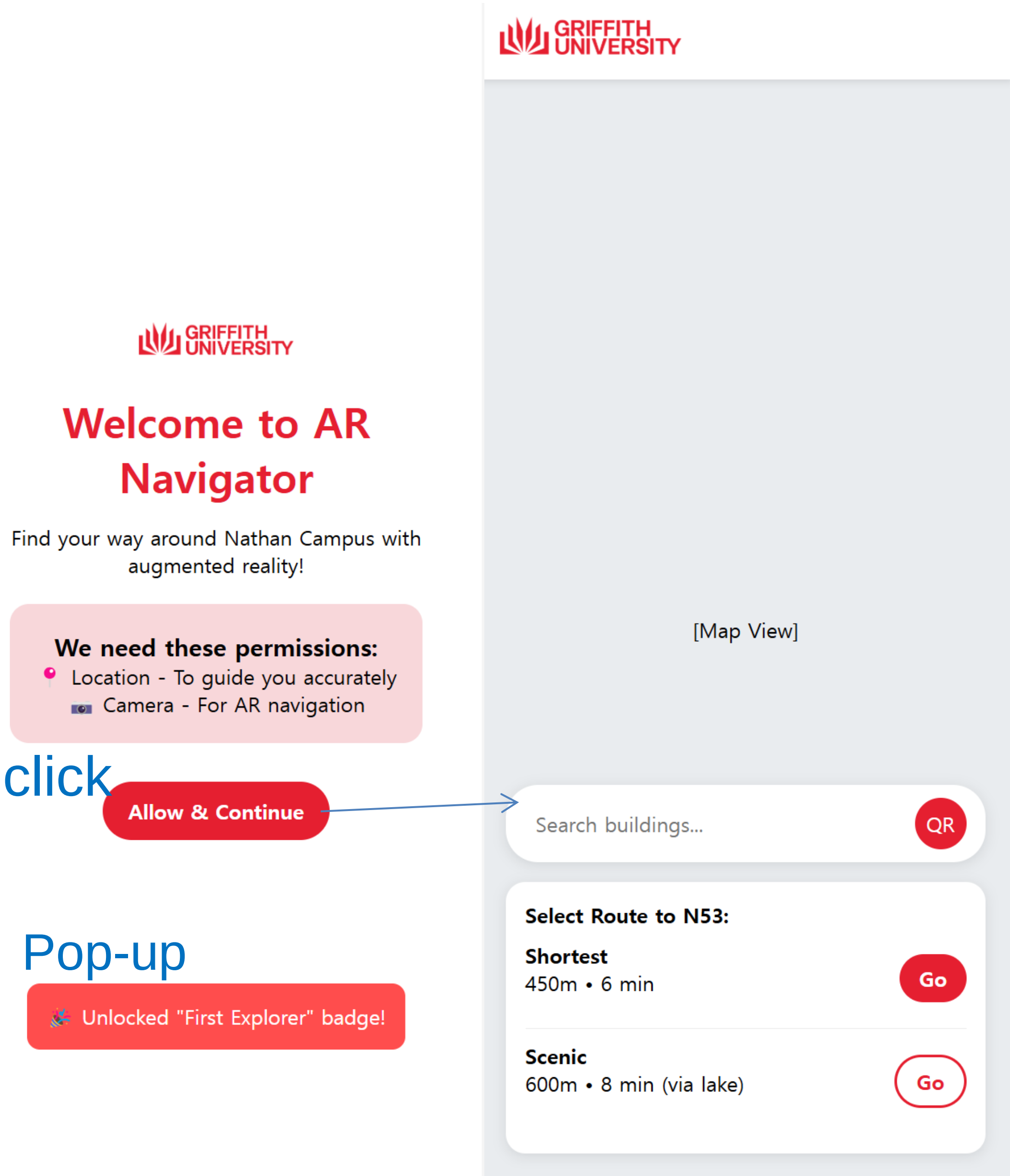

Figure 3. Permission and Onboarding to location Search page.

The **Permission & Onboarding** screen features a popup requesting camera and location access, followed by an animated "First Explorer" badge unlock. Users interact by tapping "Allow" to proceed to setup.

The **Start/Destination Selection** screen displays a map view with the user's current GPS location, a search bar for building names or QR scanning, and route options (shortest/scenic) with estimated travel times. Upon selecting a destination, the AR view automatically launches.

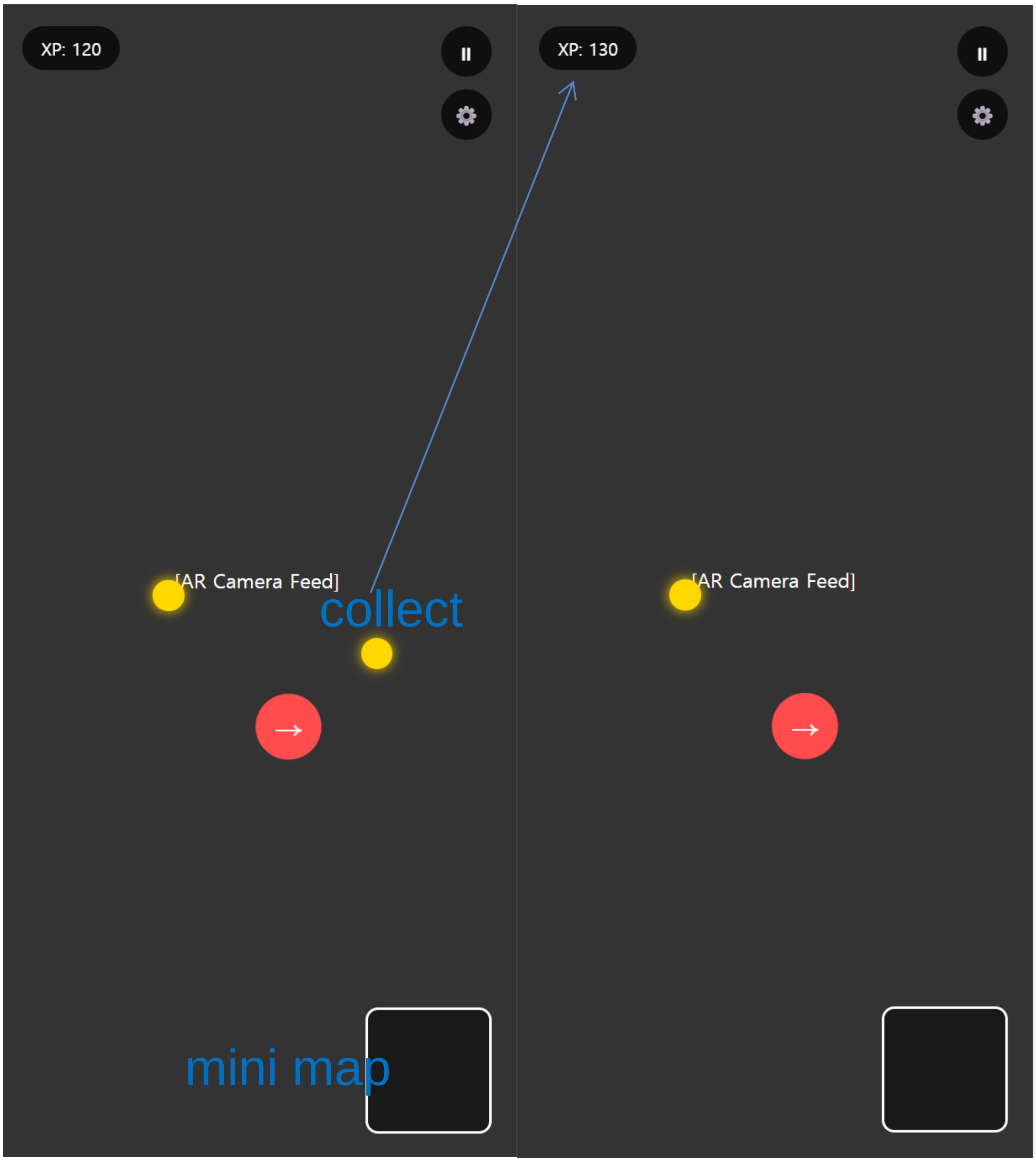

Figure 4. AR Navigation Camera View.

The **AR Navigation View** overlays live camera footage with directional AR arrows and collectible coins. A mini-map in the bottom corner and an XP counter track progress, while a "Pause/Settings" button in the top-right provides controls. Users follow the arrows and collect coins to earn XP.

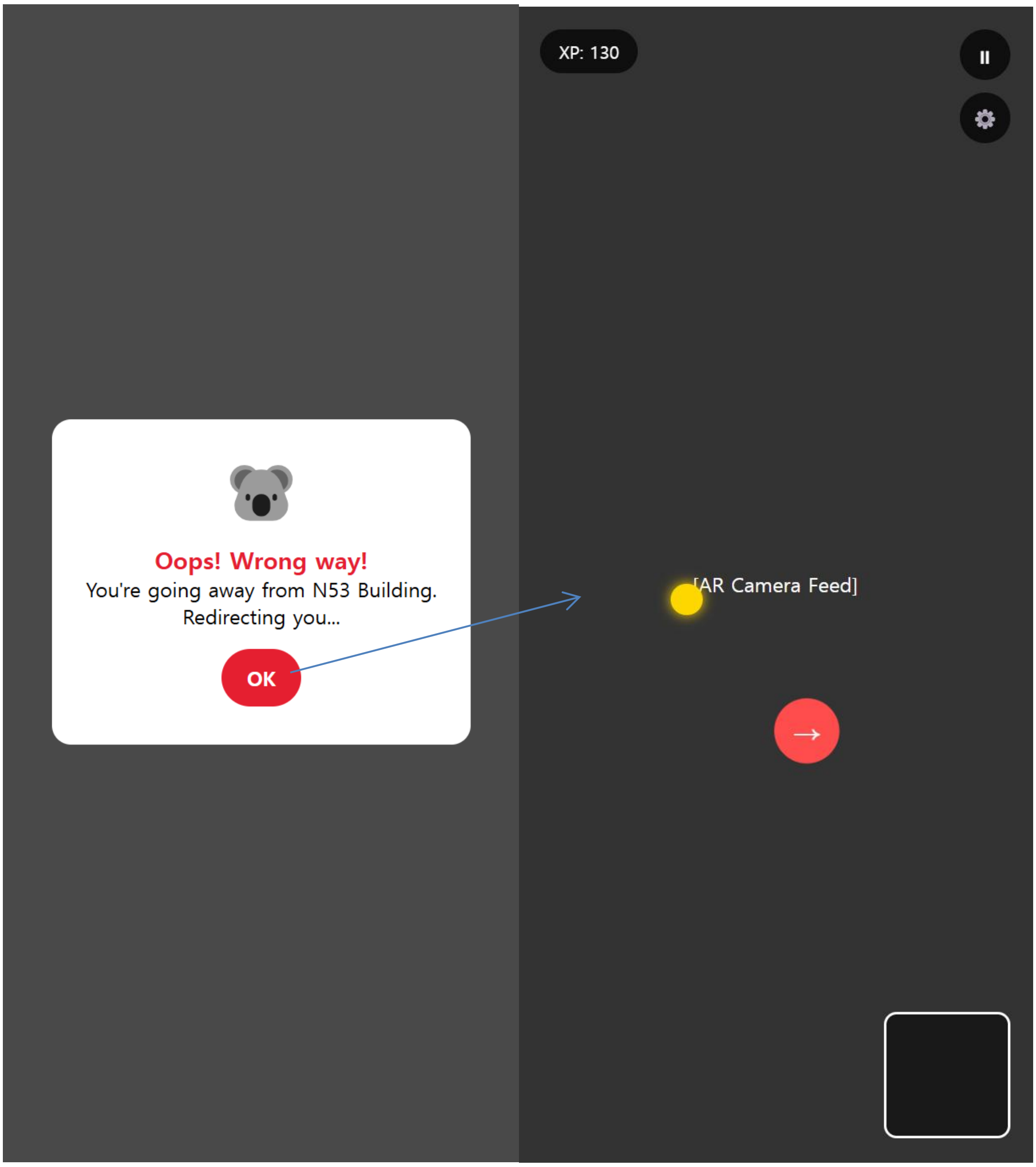

Figure 5. Wrong Path Notification.

If a user strays from the path, the **Wrong Path Alert** screen triggers playful visual effects (e.g., a bouncing koala emoji) with the message "Wrong way! Redirecting..." while automatically recalculating the route. The AR path updates once the user corrects their direction.

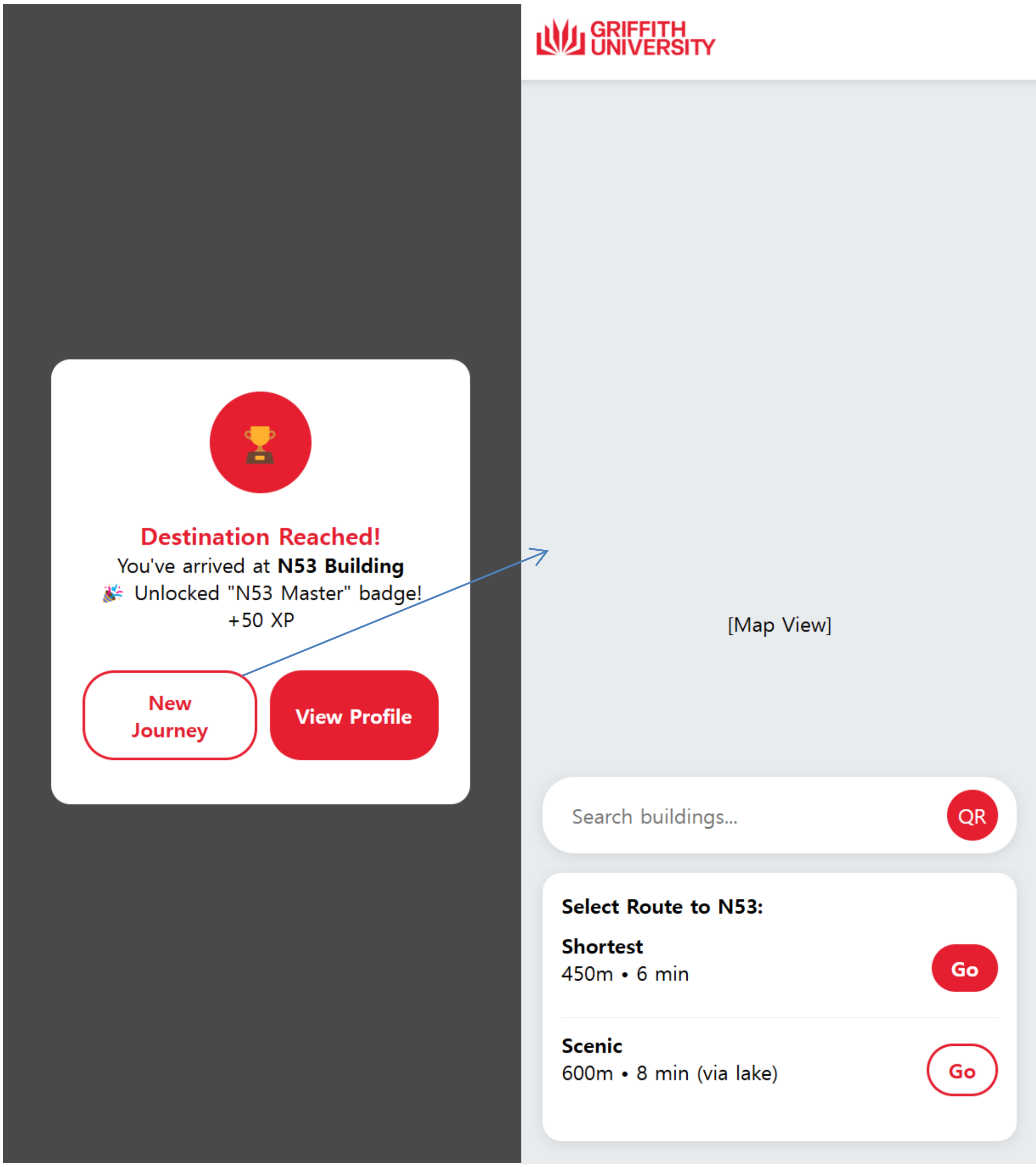

Figure 6. Destination Arrival Notification.

Upon arrival, the **Destination Reached** screen congratulates the user with a "Building Master" badge popup and presents two options: "View Profile" or "New Journey." Tapping either button exits AR mode.

Executable Prototype via Chrome

ui_prototype_Jeeyoung.html

5.4 Storyboards

Onboarding and Starting Point Setup

The user journey begins with app installation and permission granting. Upon first launch, users will grant location and camera access, which automatically unlocks the "First Explorer" badge. The next step involves setting a starting point, where users can either search on the map or use their current location. For destination selection, users will search for campus buildings, after which the system displays route options (shortest or scenic) along with estimated travel time. Key technical components supporting this flow include the permission UI and building database integration.

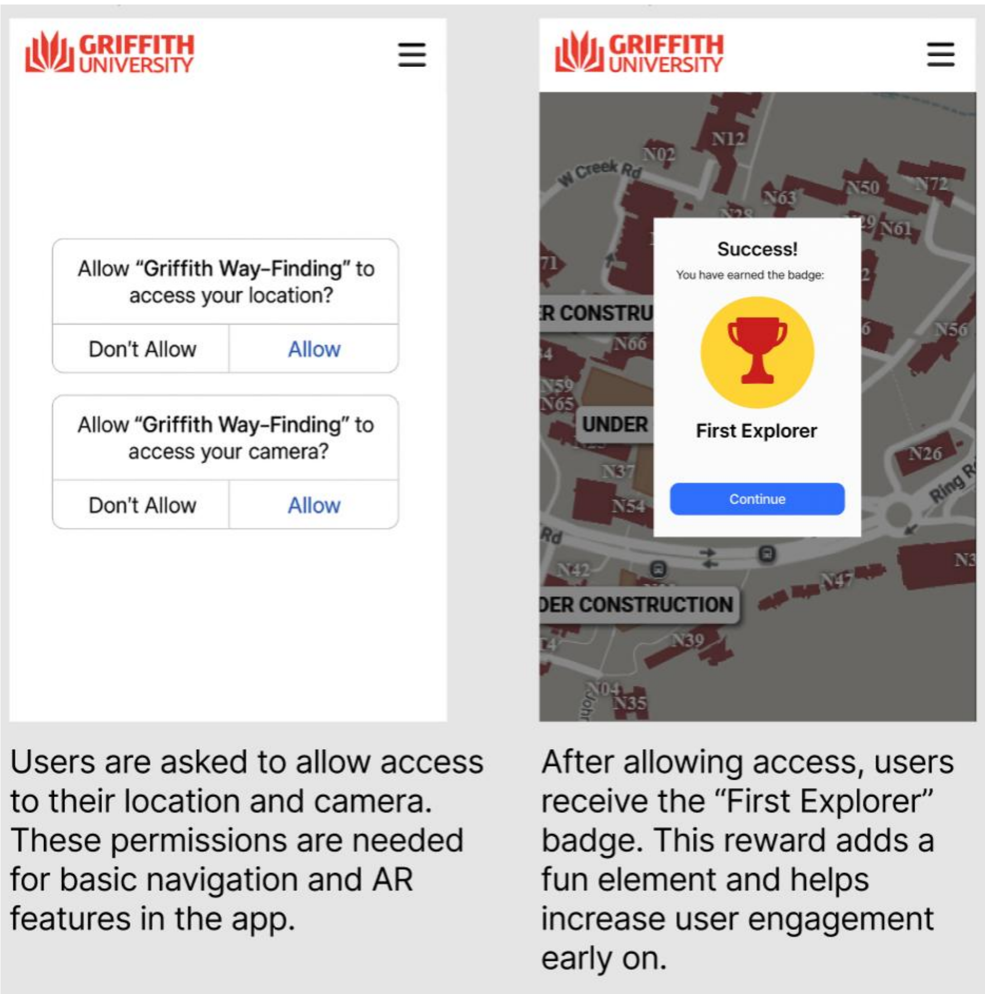

Figure 7. App Install and Permissions.

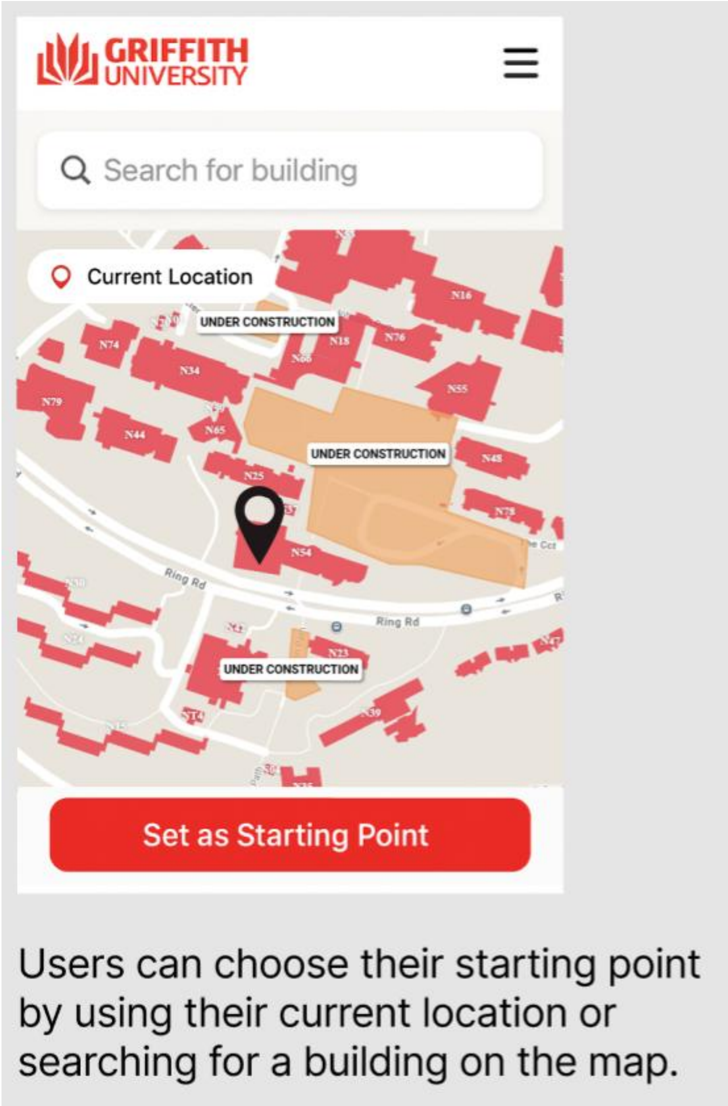

Figure 8. Starting Point Setup.

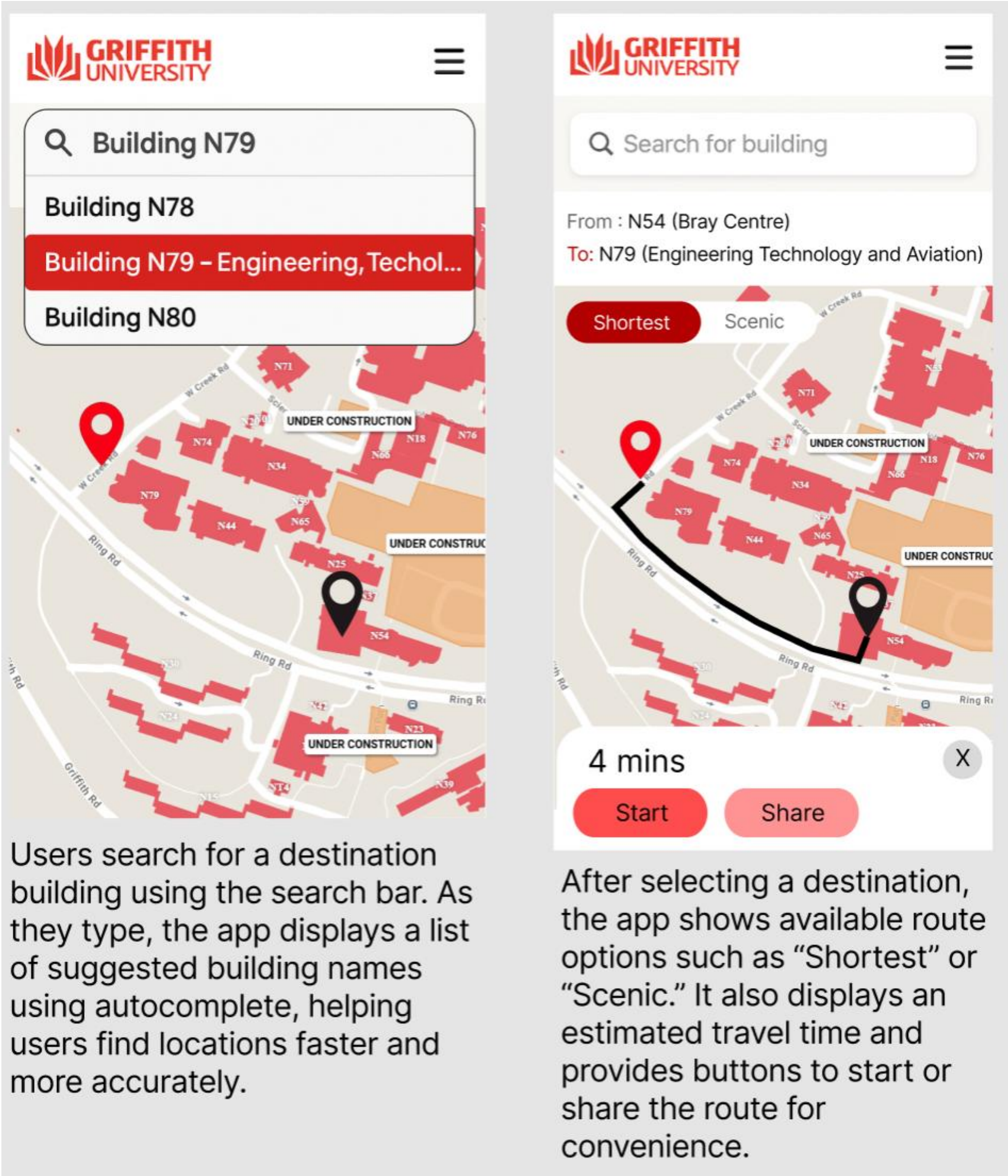

Figure 9. Destination Selection.

Navigation

During the navigation phase, users follow AR path guidance featuring collectible arrows and coins. Correct path following earns them XP. The system incorporates wrong path detection, triggering visual effects (VFX) and sounds (such as character popups) when users deviate, followed by automatic rerouting. Upon reaching the destination, users receive a building-specific badge and completion alert. This functionality is enabled by AR arrow object rendering algorithms, off-route detection systems, and reward trigger mechanisms.

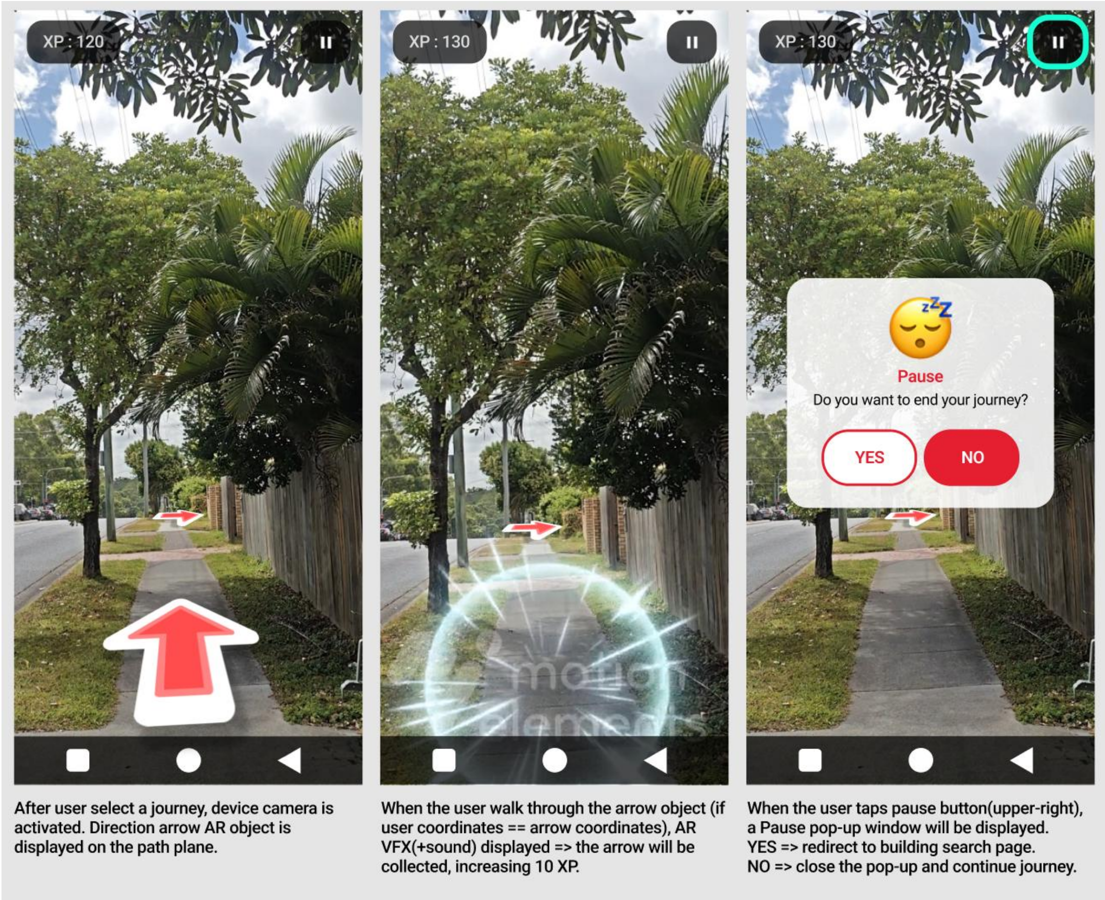

Figure 10. AR Path Navigation Process.

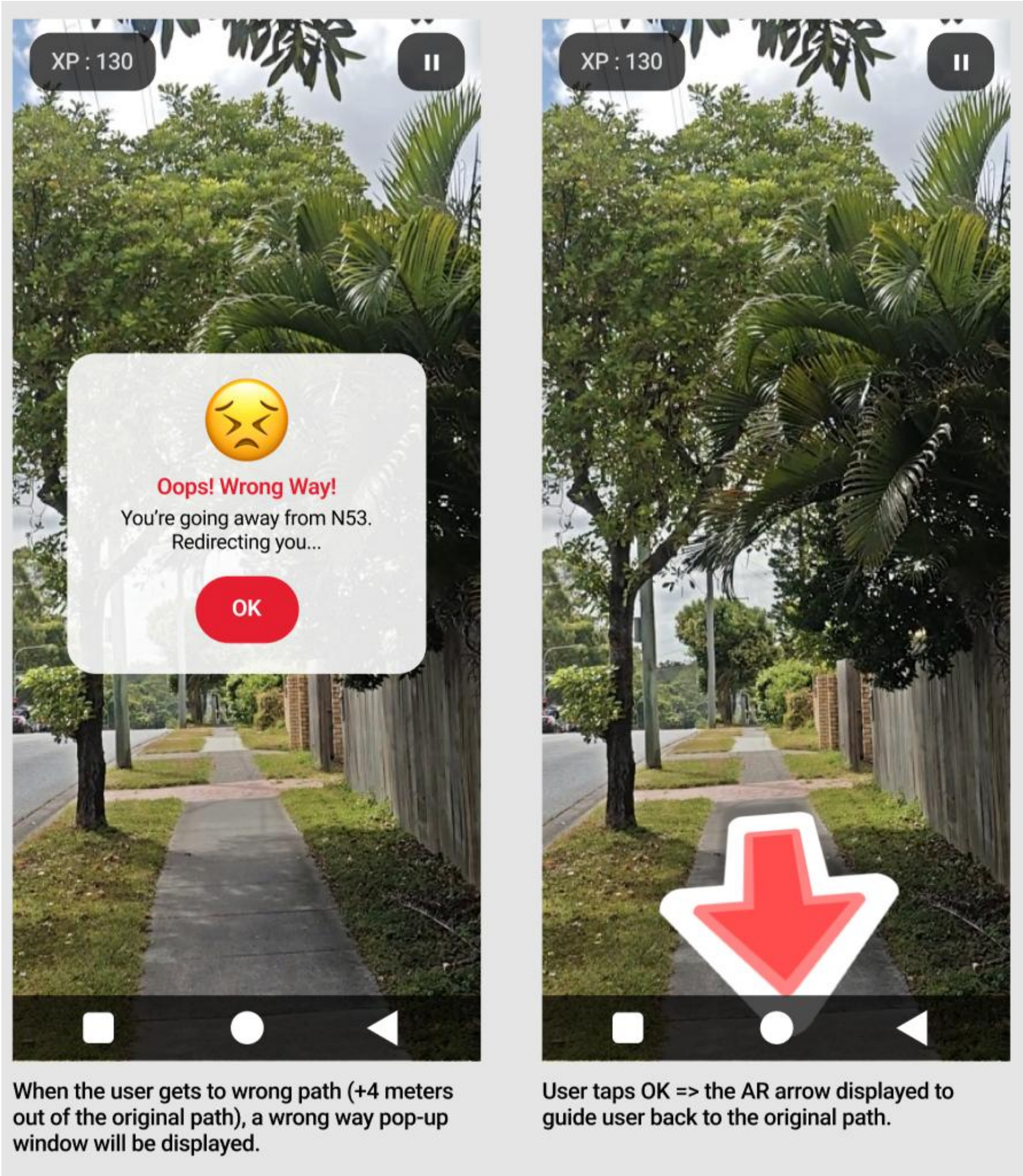

Figure 11. Wrong Path Detection and Re-direction.

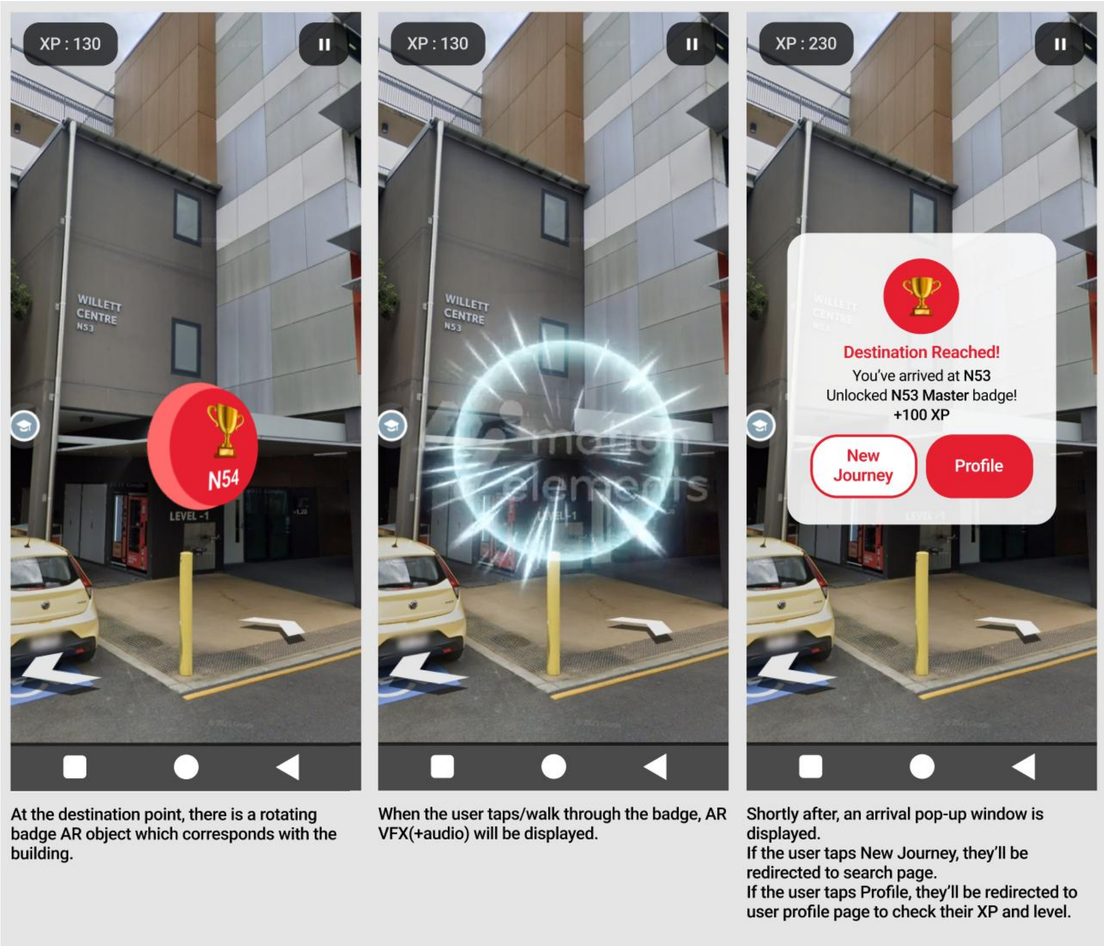

Figure 12. Building Badge Collection and Destination Arrival.

Exploration, Profile and Rewards

The exploration component allows users to discover and interact with AR wildlife animals. By greeting AR animals, users unlock encyclopedia entries and earn Animal Badges. The profile/achievements section displays collected badges, XP, leaderboard rankings, and wildlife encyclopedia progress. Settings include options to toggle VFX/sounds and calibrate AR alignment. Core technical features supporting these interactions include AR animal placement, encyclopedia UI design, and Firebase integration for achievement synchronization across devices.

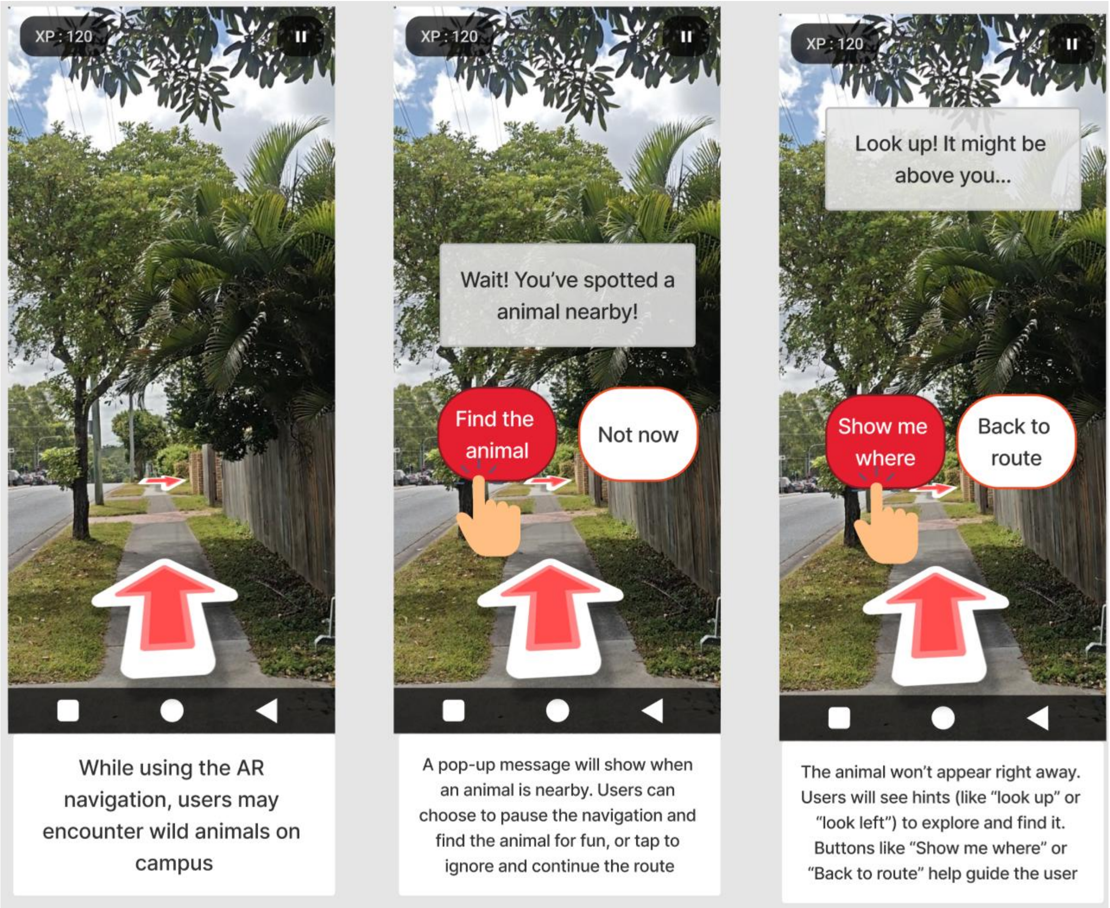

Figure 13. Encountering Process of Wildlife Animal.

Figure 14. Animal Badge Collection and Continue Journey.

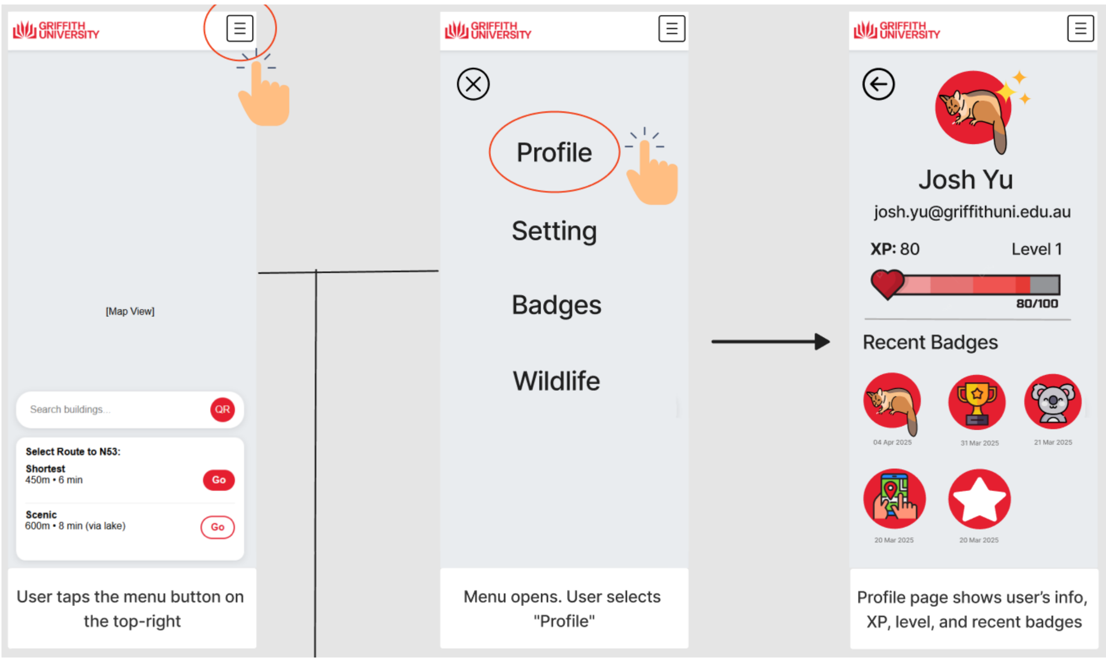

Figure 15. Navigation Bar Containing Profile, Setting, Badges and Wildlife Tab.

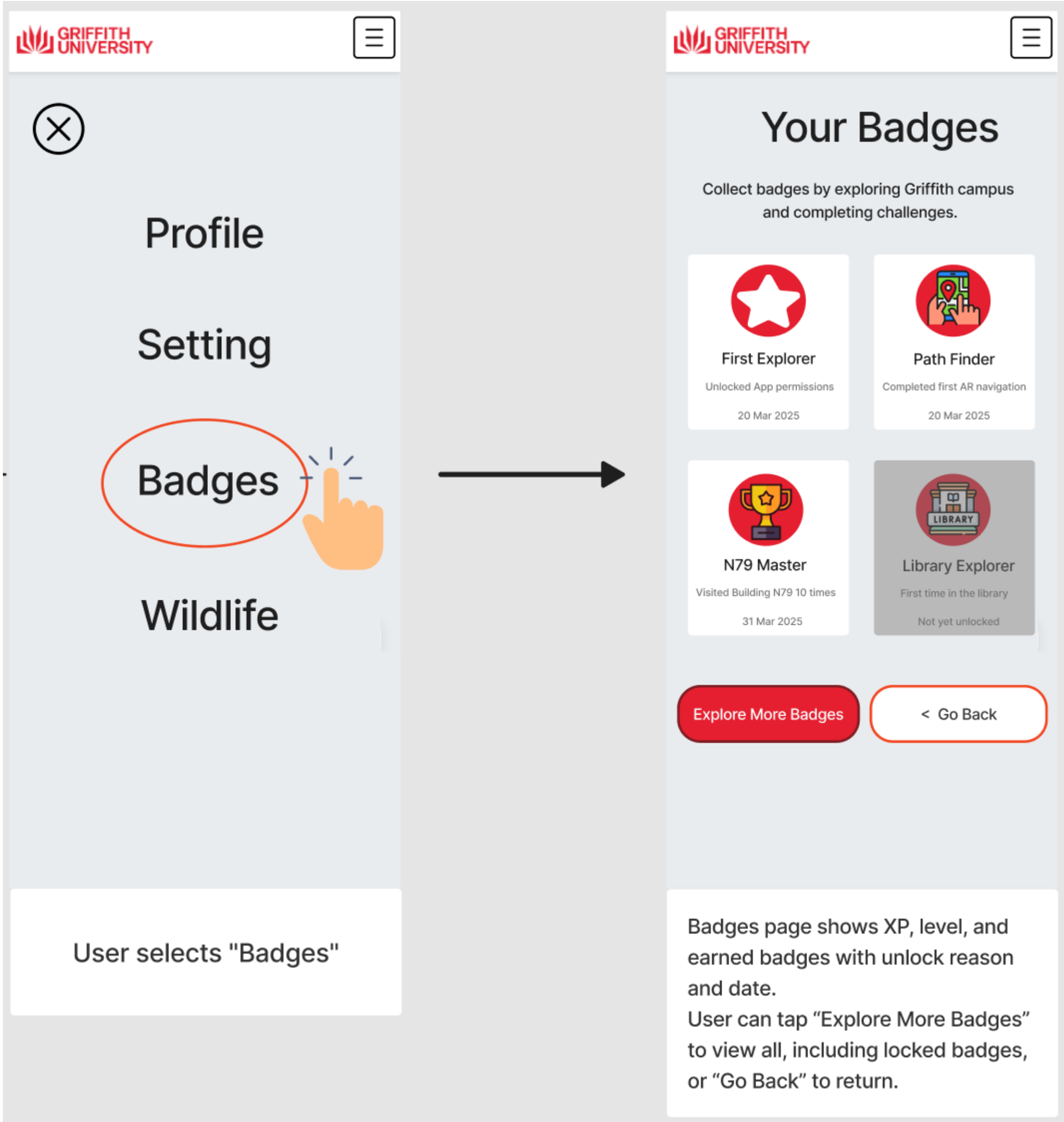

Figure 16. Navigate to Badges Page.

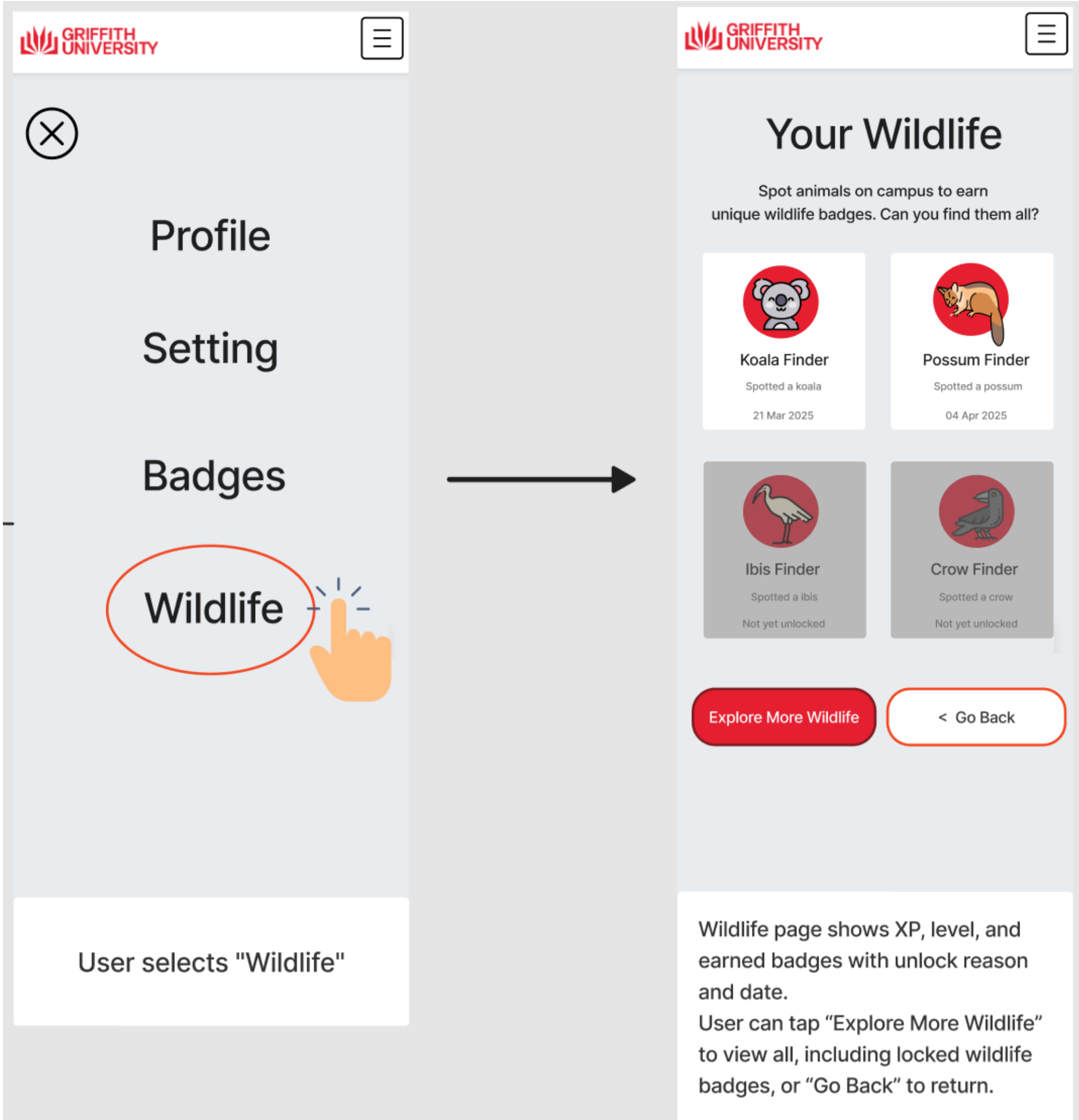

Figure 17. Navigate to Wildlife Page.

Team Collaboration Figma Page

[Shared Figma Link]

6 Meeting Summaries

Meeting 1 – Wednesday, 19 March 2025

Attendees: Jeeyoung, Josh, Yeongjoo

Agenda:

- Review Sections 1–3.
- Assign writing responsibilities.
- Set internal deadline for first drafts.

Key Discussion Points:

- The team decided to begin the report by drafting Sections 1, 2, and 3.
- Each section would focus on a different aspect of the project: purpose and scope, system overview, and target users.
- Members selected sections based on topic familiarity and preference.
- A shared draft deadline was set to allow time for feedback before starting later sections.

Action List:

- Jeeyoung (Section 1 – Introduction):
  - Write about the project's purpose, objectives, and scope.
  - Clearly state any limitations or excluded areas.

- Josh (Section 2 – AR Application Overview):
  - Describe the app's main features and navigation flow.
  - Identify features that are not included in the scope (e.g., indoor navigation).

- Yeongjoo (Section 3 – People):
  - Define the main user groups.
  - Describe their backgrounds, needs, and how the app design will support them.

Internal deadline: 23 March 2025

Reflection:
During the session, all members participated actively and shared their ideas about the project direction. The task allocation process went smoothly, and each person chose their section. And we successfully set up an internal deadline.

Meeting 2 – Wednesday, 26 March 2025

Attendees: Jeeyoung, Josh, Yeongjoo

Agenda:

- Plan content development for Sections 4 and 5.
- Clarify technical scope and user interaction flow.
- Allocate writing tasks for the upcoming sections.

Key Discussion Points:

- Technologies used in the application should be clearly explained with their strengths and limitations.
- Functional and non-functional requirements must include gamification and accessibility considerations.
- The system architecture should present a modular structure in a layered format.
- Interface design will be created in Figma and reflect a consistent visual style.

Action List:

- Josh (Section 4 – Technology):
  - Describe core technologies (Unity, C#, AR Foundation, GPS, QR code, Android).
  - Explain their benefits, drawbacks, and key implementation challenges.

- Yeongjoo (Section 5.1 – Requirements, 5.2 – System Architecture):
  - Write functional and non-functional requirements, incorporating gamification and accessibility.
  - Define system modules and present a layered architecture based on user interaction flow.

- Jeeyoung (Section 5.3 – Interface Design):
  - Use HTML/CSS/JS to create executable interface prototype.
  - Create design Systems including colour scheme, layout, and typography.
  - Ensure UI aligns with prior requirements and user needs.

Internal deadline: 30 March 2025

Reflection:
Regarding the tasks assigned at previous meeting, all members completed their assigned sections by the internal deadline. Project Manager had integrated all written drafts. The team showed strong engagement, and everyone contributed actively with ideas. Communication remained clear and smooth.

Meeting 3 – Wednesday, 2 April 2025

Attendees: Jeeyoung, Josh, Yeongjoo

Agenda:

- Develop the application storyboard based on Sections 1–5.3.
- Finalise user flow and screen transitions.
- Assign tasks for storyboard refinement.

Key Discussion Points:

- The team agreed to create the storyboard using any tools/programs, based on the content completed in Sections 1 to 5.3.
- The storyboard would be divided into three parts, each representing a different stage of user interaction.
- Tasks were divided among members to work on each part independently while maintaining visual consistency.

Action List:

- Yeongjoo (Part 1 – Onboarding and Setup):
    - Create storyboard screens for permission requests, location setup (via QR/GPS), and destination selection.
    - Use Figma to visualise the user experience during app onboarding.

- Jeeyoung (Part 2 – Navigation):
    - Design screens for AR guidance, collectible rewards, wrong-path detection, and arrival feedback.
    - Ensure alignment with the features and logic defined in previous sections.

- Josh (Part 3 – Exploration and Rewards):

    - Develop screens for AR animal encounters, badge collection, and profile view.
    - Include settings and motivational feedback elements in the design.

Internal deadline: 7 April 2025

Reflection:
In Meeting 2, we assigned tasks related to the technical sections of the report. Josh was responsible for Section 4 (Technology), Yeongjoo worked on Sections 5.1 (Requirements) and 5.2 (System Architecture), and Jeeyoung handled Section 5.3 (Interface Design). All members completed their parts by the internal deadline. Project Manager had integrated all written drafts.

Meeting 4 – Wednesday, 9 April 2025

Attendees: Jeeyoung, Josh, Yeongjoo

Agenda:

- The Project Manager reviewed whole document and shared feedback based on marking rubrics.
- Identify and assign revision tasks for Sections 1, 2, 4, 5.3, and 6
- Decide on formatting changes for the final report

Key Discussion Points:

- Academic or real-world references are needed in Sections 1 and 2 to support claims (e.g., AR adoption rates, user surveys, or competitor analysis).
- Section 1 requires clearer and more specific objectives.
- Section 4 needs deeper technical descriptions such as plane detection and anchor persistence, including named tools (e.g., Unity 2022.3 LTS, AR Foundation 5.0, Geospatial API).
- A comparison and justification of chosen technologies (e.g., Unity vs. Unreal Engine) should be included.
- Section 5.3 must include reasoning for design decisions with supporting sources (e.g., Roboto font readability backed by Material Design).
- Section 6 needs to explicitly track the status of previous tasks, noting what was completed or revised.

- The team agreed to shift from bullet-point style to a formal report format for all written sections to improve readability and meet assessment standards.

Action List:

- Jeeyoung:
  - Revision based on feedback (Section 1 – Introduction, Section 5.3 – Interface Design).
  - Final quality check and overall tone polishing.

- Josh:
  - Revision based on feedback (Section 2 – AR App Overview, Section 4 – Technology).

- Yeongjoo:
  - Revision based on feedback (Section 6 – Project Tracking).

Internal deadline: 10 April 2025

Reflection:
In Meeting 3, we planned to create the storyboard in Figma, split into three parts. Yeongjoo was responsible for the onboarding and setup screens, Jeeyoung worked on the navigation phase, and Josh handled the exploration and rewards. All three members completed their parts by the agreed date. We also gave each other feedback to keep the visual style consistent. Teamwork was positive and productive.

7 References

Fajrianti, E. D., Funabiki, N., Sukaridhoto, S., Panduman, Y. Y. E., Kong, D., Shihao, F., & Pradhana, A. A. S. (2023, June 22). *INSUS: Indoor navigation system using Unity and smartphone for user ambulation assistance*. *Information, 14*(7), 359. https://doi.org/10.3390/info14070359

Google for Developers. (2024). *ARCore supported devices*. https://developers.google.com/ar/devices

Pokémon Go Statistics Report 2016 Edition. (n.d.). https://artillry.co/wp-content/uploads/2018/05/Pokemon-Go-Statistics.pdf

Qiu, Z., Mostafavi, A., & Kalantari, S. (2023). *Use of augmented reality in human wayfinding: A systematic review*. ArXiv. https://doi.org/10.48550/arxiv.2311.11923

Saraswathi, K. L. V. R., Laakshini, G., Komma, S., & Subhani, S. (2023, October). *ARCore Geospatial API*. *International Journal of Engineering Research and Applications, 13*(10), 8–13. https://www.researchgate.net/publication/374547629_ARCore_Geospatial_API

Unity AR+GPS Location Docs (v3.6.0). (2020). *Routes and navigation*. Unity. https://docs.unity-ar-gps-location.com/routes/#quickstart

Unity Technologies. (n.d.). *AR Foundation 6.1 documentation*. Unity. https://docs.unity3d.com/Packages/com.unity.xr.arfoundation@6.1/manual/index.html

W3C. (2023, September 21). *Web content accessibility guidelines (WCAG) 2.1*. https://www.w3.org/TR/WCAG21/