CS 491 - Senior Design Project 1

Bilkent University

Fall 2025

Analysis and Requirements Report

Group T2525

Ege Ertem 22202433

Can Kütükoğlu 22202619

Sami Bora Akoğuz 22202184

Mehmet Rodi Aydoğdu 22201856

Cem Tarkan Tekcan 22201590

1. Introduction

This report serves as the formal Analysis and Requirement Report for the Senior Design Project titled **"Odyssey"** (Group T2525). Odyssey is a cross-platform mobile application designed to redefine urban exploration by merging the physical world with digital creativity. The application gamifies tourism, converting ordinary city guides into interactive experiences. Unlike traditional tour apps that offer static content, Odyssey utilizes Large Language Models (LLMs) to generate dynamic narratives and puzzles, creating a unique hybrid of digital geocaching and guided tours.

2. Current System

The current market for interactive tourism is populated by applications that attempt to blend sightseeing with gamification. Platforms like CityFans, Explorial, and Questo allow users to solve puzzles and unlock stories while touring landmarks, while Geocaching has demonstrated a massive demand for location-based puzzle adventures [1,2,3]. Aside from these digital solutions, many users still rely on traditional, non-interactive methods such as physical guidebooks, static maps, or audio tours to navigate new cities.

However, these applications only support manually generated content or static solutions. Applications like Questo depend only on a community of content creators to generate and decide on new routes, which limits scalability and consistency [2]. Moreover, traditional guides also stick with static content which lacks dynamic interaction or personalization. Briefly, no widely renown solution exists to automatically generate dynamic

and fresh content without static human intervention, which is the exact need that Odyssey is aimed to fulfill.

3. Proposed System

3.1. Overview

Odyssey is a cross-platform mobile application designed to gamify tourism by merging physical exploration with digital storytelling. The system transforms standard city tours into interactive experiences using three primary modes: Puzzle Tours, which challenge users with trivia, gyroscope mechanics, and Augmented Reality (AR); Story Tours, which provide relaxed, narrative-driven routes generated by AI or the community; and Hybrid Tours, which combine both storytelling and interactive challenges.

The system architecture is divided into a mobile Client Layer and a Backend Layer. The Client Layer is a mobile application that manages local storage, GPS tracking to facilitate real-time user interaction and other required sensors/interfaces we will require throughout the project. The Backend Layer, built on Django, hosts essential services including User Authentication, Social Services (managing profiles, adding friends and such), and the Tour Manager. A critical component of the proposed system is the AI Story Creation Module, which utilizes a structured precreated prompt to communicate with Large Language Models (LLMs), allowing the system to generate dynamic narratives and puzzles for users in real-time.

Access to the system is managed through four distinct user roles: Normal Users (free trial access), Premium Users (subscription-based access to exclusive features), Content

Creators, and Admins. By integrating these components, Odyssey aims to create a safe, community-driven platform that redefines travel through digital creativity.

3.2. Functional Requirements

3.2.1. User Management & Access Control

- The system shall support user registration (e.g., email/password, social login) and login.
- The users shall be able to login with different permissions:
  - **Normal User:** Base level, can play tours, use social features.
  - **Premium User:** A paid role with access to exclusive features (e.g., ad-free hints, premium tours).
  - **Content Creator:** A role with privileges to create and publish tours.
  - **Super Admin:** A role with full system-level access for management and moderation.
- The users shall be able to pay to upgrade to the “Premium User” role.
- The users shall be able to gain XP to increase their user “level”.

3.2.2. Social & Profile Features

- Each user shall be able to have a customized profile.
- The profile shall display:
  - Badges earned
  - A total count of completed tours.
  - Last few completed tour locations
  - Following/follower count
- Users shall be able to search for other users.
- Users shall be able to follow and be followed by other users.

3.2.3. Map & Navigation

- Users shall use a map interface as the primary navigation.

- The system shall obtain and display the user's real-time location using the device's GPS.
- The system shall fetch and display Points of Interest (POIs) from a third-party API.

3.2.4. Tour Creation & Management (Content Creator Role)

- Users with the "Content Creator" role shall be able to create a new tour.
- Content creators shall be able to see the analytics of their tours.
- Creators shall define the tour type:
  - **Story Mode:** A narrative-driven tour.
  - **Puzzle Mode:** A challenge-based tour.
  - **Hybrid Mode:** A tour combining both story and puzzle elements.
- Creators shall be able to add and define categories for their tours (e.g. Secure, Paid Tolls).

3.2.5. Tour Gameplay Mechanics (Normal User Role)

- Users shall be able to browse, select, and start a generated tour.
- The system shall save a history of the user's previously generated tours.
- **Story Mode:** Upon arriving at a designated location, the next part of the story shall automatically unlock. The user may see the full route in advance.
- **Puzzle Mode:** The route is not fully known. Users must solve a puzzle at one location to unlock the next POI.
- The system must verify the user's physical presence at a POI before unlocking content.
- The system shall provide puzzle mechanics, including:
  - **Trivia:** Text-based questions with multiple choice and open-ended questions.
  - **AR:** Augmented Reality challenges.
  - **Gyroscope:** Puzzles based on device orientation.

- There shall be “Premium” Tours, which are only accessible by “Premium” users.

3.2.6. Monetization & Engagement

- The system shall provide "hint" and "skip" options for puzzles activated by spending in-game currencies.
- Hints/skips shall be accessible by watching an advertisement or by being a Premium User as well.
- The system shall send push notifications to users for engagement.
- The app shall have tokens that are purchasable with real-world currency and obtainable with ads. Users shall also gain free tokens daily.
- Tours shall be unlockable by spending tokens

3.2.7. Community & Moderation

- Users shall be able to leave a review for a completed tour.
- Users shall be able to rate a tour.
- Users shall be able to report inappropriate routes or users.
- Users shall be able to add the tours to their saved tours.

3.2.8. Admin

- Admin(s) shall be able to get reports of monetization reports
- Admin(s) shall be able to get reports of selected tour usage and be able to delete inappropriate tours
- Admin(s) shall be able to delete certain users

3.3. Non-functional Requirements

3.3.1. Usability

- **Learnability:** First time users must be able to find and start their first tour within 60 seconds of logging in without external instructions.
- **Efficiency:** An experienced "Content Creator" user shall be able to create a new "Story Mode" tour with 5 locations (with pre-prepared text) in under 15 minutes.

- **Error Prevention:** The system must validate all user inputs during tour creation. If a user tries to save a tour with incomplete or wrong information, the system shall provide a clear and understandable error message.
- **Localization:** The application interface text must be apart from the code to support future translation. The system must initially support English.

3.3.2. Reliability

- **Availability:** The main core of the backend services (user login, tour browsing, payment processing) shall have an uptime of 99%.
- **Offline Mode:** The application must have an “offline mode." If a user loses internet during an ongoing tour, the app shall cache the current step. Then the app shall continue to track the location of the user. The app shall allow the user to complete the current step and also display a clear indicator for the offline mode.
- **Data Integrity:** All user-facing write operations must be transactional. A user's "tours completed count" must never be out of sync with their actual tour history.

3.3.3. Performance

- **Application Load Time:** The application must cold-start under 3 seconds on a mid-range smartphone on a 4.5G connection.
- **API Response Time:** All backend API calls must have a server-side processing time of 200ms or less.
- **Resource Utilization (Client):** The application's background location tracking must not consume more than 15% of the device's battery per hour of active use.
- **Resource Utilization (AR):** The Augmented Reality (AR) camera view must render at a consistent minimum of 24 frames per second on all supported devices to avoid jitter.

3.3.4. Supportability

- **Monitoring & Logging:** The backend system shall log all critical errors and API performance to a centralized dashboard. The client side app must report all uncaught exceptions and crashes to this centralized dashboard.
- **Testability:** The system architecture shall allow mocked GPS data. This will create the environment for automated testing of the "location check" feature without requiring physical movement.

- **Deployment:** The development team must be able to deploy a non-critical backend hotfix to production with minimal downtime.
- **Modularity:** The React Native frontend and backend must be decoupled. A developer must be able to work on the user profile UI without needing to run the tour creation service.
- **Documentation:** All API endpoints must be documented using a standard like OpenAPI (Swagger).

3.3.5. Scalability

- **Data Scalability:** The database schema must be designed to handle 100.000+ user-generated tours and 1.000.000+ reviews. Queries must still execute in under 1 second.

3.4. Pseudo Requirements

The following requirements represent the implementation constraints and specific design choices imposed by the development team to ensure system quality, maintainability, and compatibility with available resources.

- **Mobile Development Framework:** The client-side application shall be developed using React Native and TypeScript to ensure cross-platform compatibility (iOS and Android) within a single codebase.
- **Backend Framework:** The server-side application shall be implemented using the Django web framework, which is in Python, to leverage its many built-in features and administration panel.
- **Database Management System:** The system shall utilize PostgreSQL as the primary relational database for storing user profiles, tour data, and transaction logs.
- **AI Integration Protocol:** The system shall work by constructing a structured prompt that requires valid JSON output containing real-world locations and narrative content

based on the user's selected city and theme. The system then sends this prompt to LLM, parses the returned JSON string, and immediately persists the valid data into the database.

- **Coding Standards:**
    - Python code shall adhere to the **PEP 8** style guide for consistent formatting.
    - Frontend code shall utilize **TypeScript Strict Mode** to prevent type-related runtime errors.
- **Version Control:** The development team shall use Git / GitHub for version control and source code management.
- **Documentation Standards:** API endpoints shall be documented using the OpenAPI / Swagger specification to facilitate frontend-backend integration.
- **Deployment Environment:** The backend infrastructure shall be containerized using Docker

3.5. System Models

3.5.1. Scenarios

3.5.1.1. Create and run a tour from scratch

Goal: A user creates a new tour and starts navigating it.

Primary actor: Content Creator

Preconditions: User is logged in; location permission granted; internet available.

Trigger: User taps Create Tour.

Main flow:

1. User taps Create Tour.
2. System asks for tour title, description, category/tags, and cover image.
3. User adds 3–8 stops by selecting points on the map or searching for places.
4. For each stop, the user adds a short note + optional photo/audio.
5. User taps Publish.
6. System validates required fields (title, at least 1 stop, location info).
7. System saves the tour and marks it as Public.
8. User taps Start Tour.
9. System opens navigation and shows the next stop + distance

Postconditions: A public tour exists; navigation session begins.

3.5.1.2. Report unsafe/misleading content in a tour

Goal: User reports a tour/stop that is dangerous or incorrect.

Primary actor: User

Preconditions: User is viewing a tour.

Trigger: User taps Report on tour/stop.

Main flow:

1. User opens a tour and notices unsafe instructions / wrong location.
2. User taps Report.
3. System asks for report type (e.g., “Safety issue”, “Wrong location”, “Harassment”, “Spam”) and a comment.
4. User submits report.
5. System acknowledges receipt and stores the report with tour/stop ID, timestamp, reporter ID.
6. System flags the tour for moderator review.
7. If multiple reports exceed threshold, system temporarily hides the tour from search .

Postconditions**:** Report stored; moderation queue updated; possible visibility change.

3.5.1.3. User registration and login

Goal: A new user creates an account and logs into the system.

Primary actor: Guest User

Preconditions: App installed; internet connection available.

Trigger: User taps Sign Up / Login on welcome screen.

Main success flow (Registration):

1. Guest user opens the app.
2. System displays Welcome screen with options: Login and Register.
3. User taps Register.
4. System asks for required information (username, email, password).
5. User fills the form and taps Create Account.
6. System validates input (email format, password rules, uniqueness).
7. System creates the user account and stores encrypted credentials.

8. System automatically logs in the user.
9. User is redirected to the Profile screen.

3.5.2. Use Case

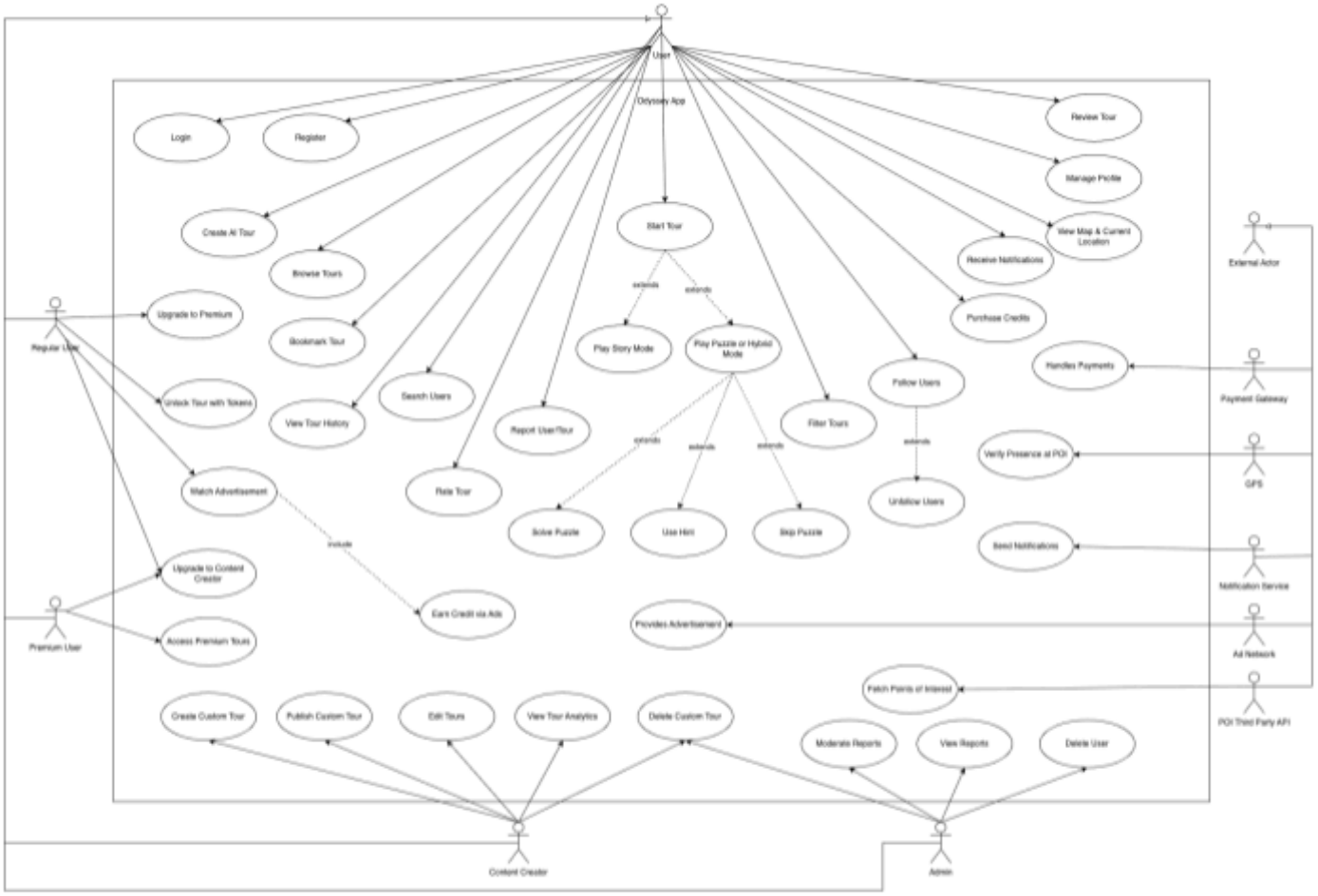

Fig 1. Use Case Diagram

3.5.3. Class Diagram

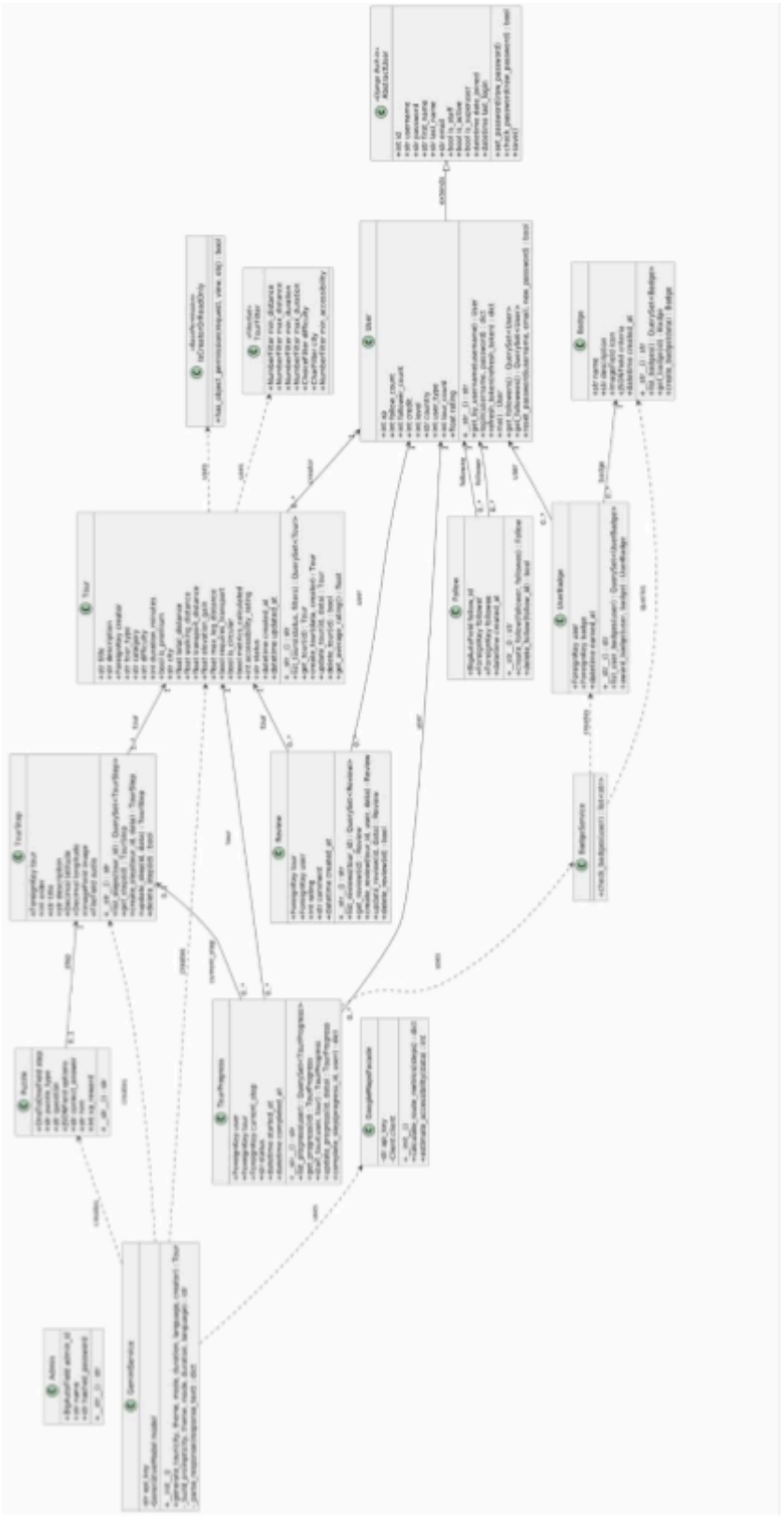

Fig. 2 Class diagram of the django backend

3.5.4. E/R Diagram

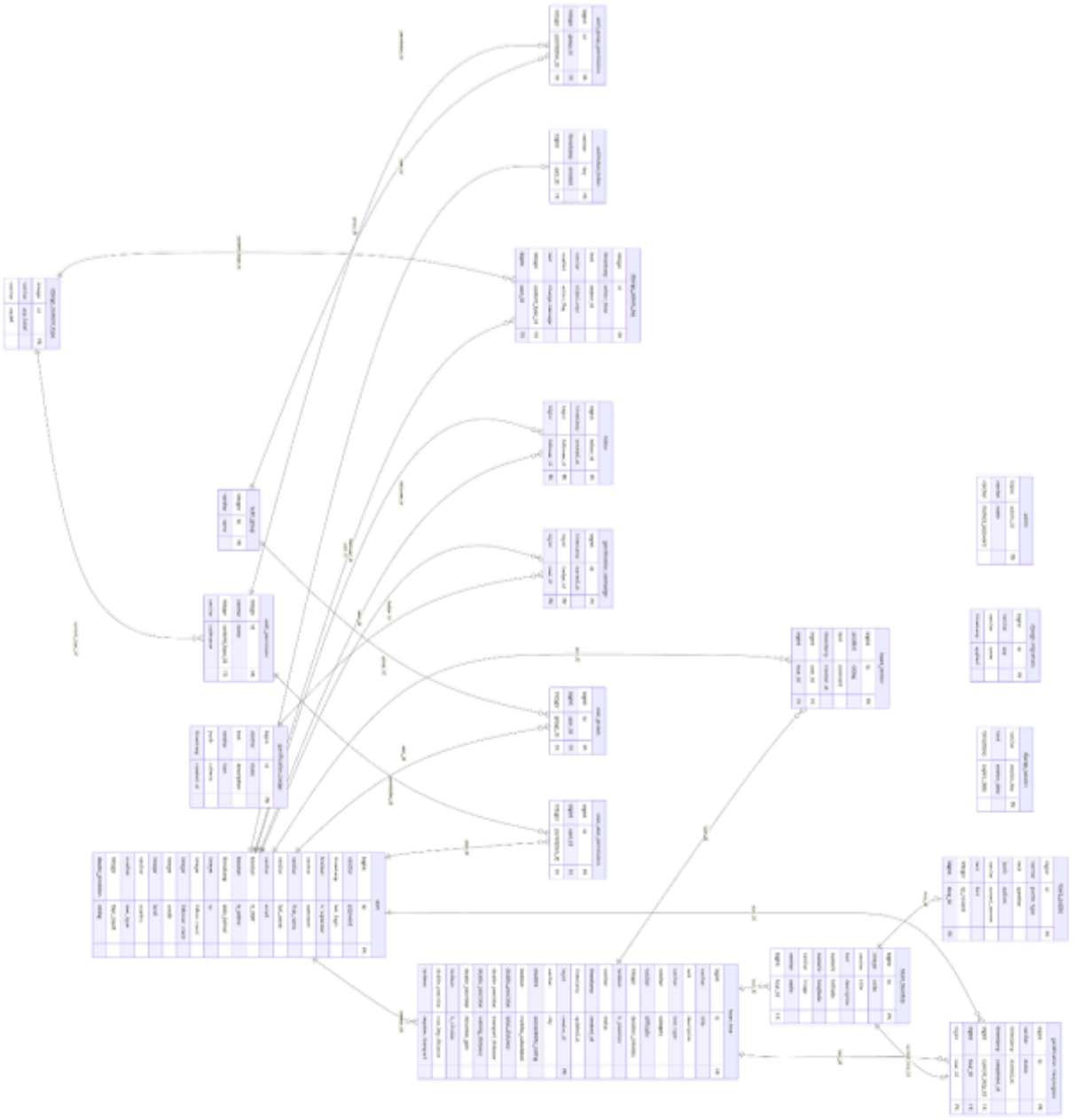

Fig. 3 E/R Diagram

3.5.5. Dynamic Models

3.5.5.1. State Diagram

Fig. 4 State diagram of tour and tour progress

3.5.5.2. Sequence Diagram

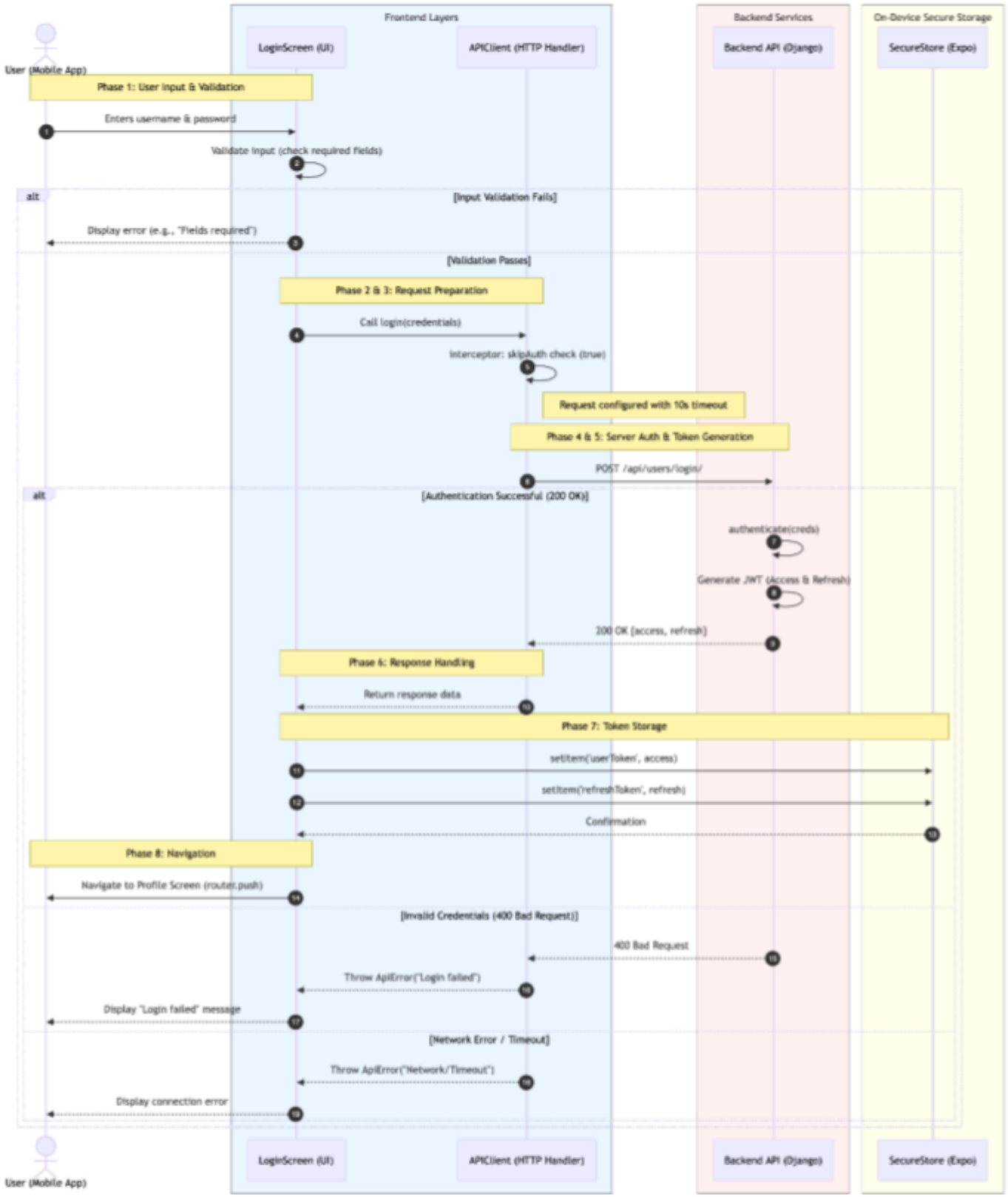

Fig. 5 Sequence diagram of user login process

3.5.6. User Interface

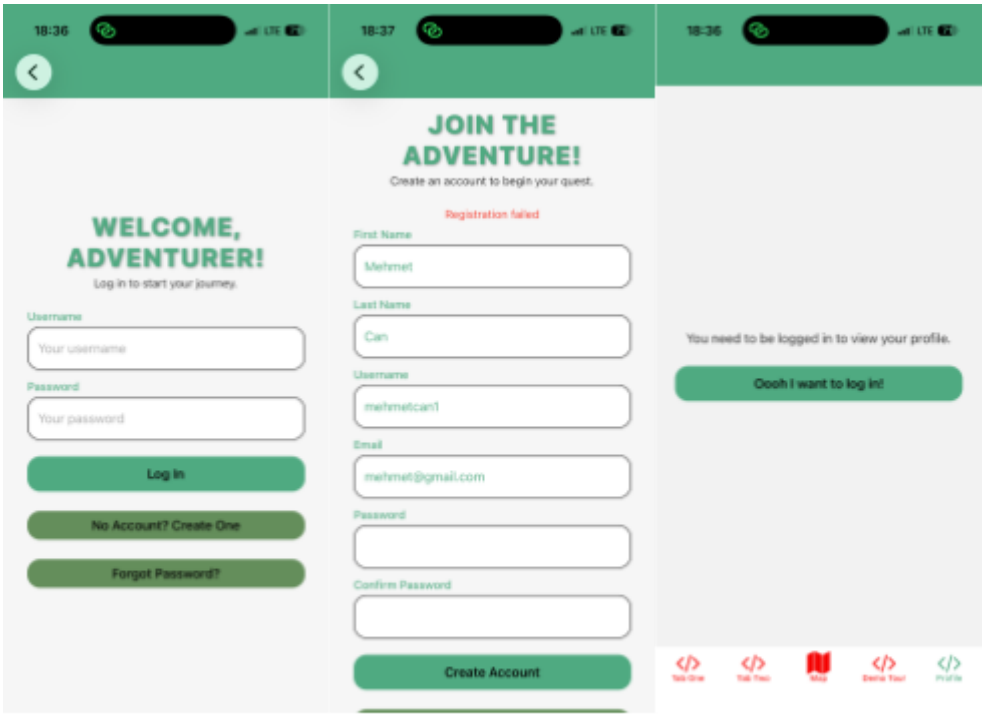

Fig. 6 User login mockups

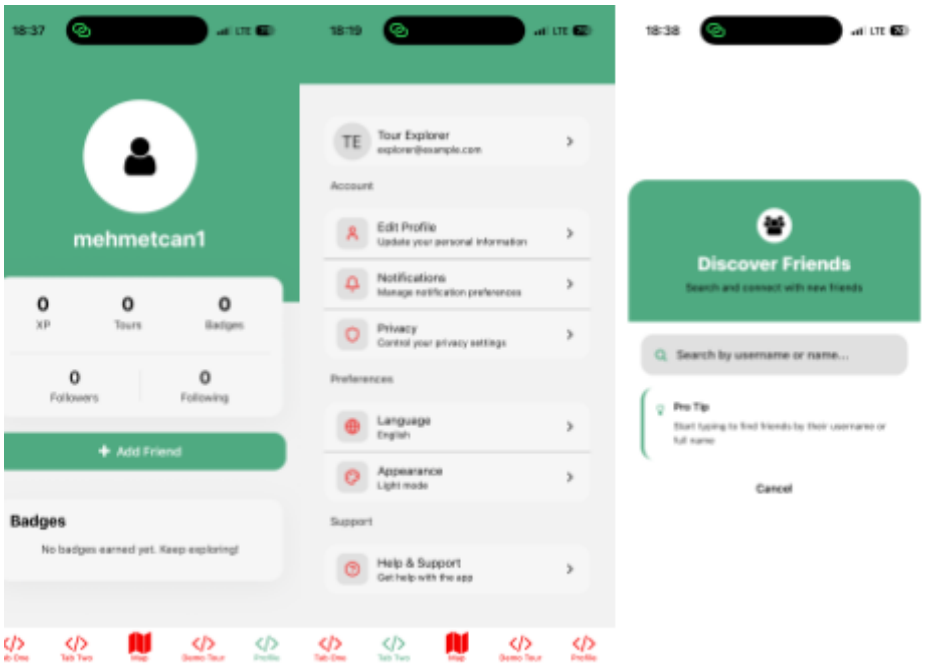

Fig. 7 Profile page mockups

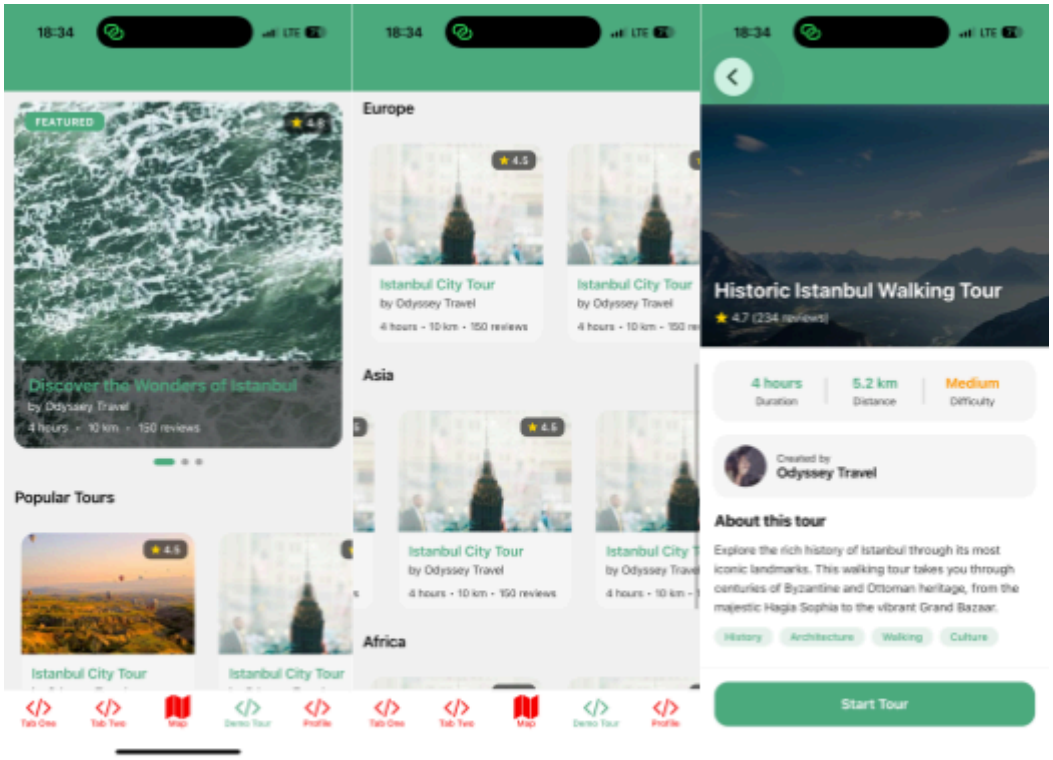

Fig 8. Tour filtering and selecting

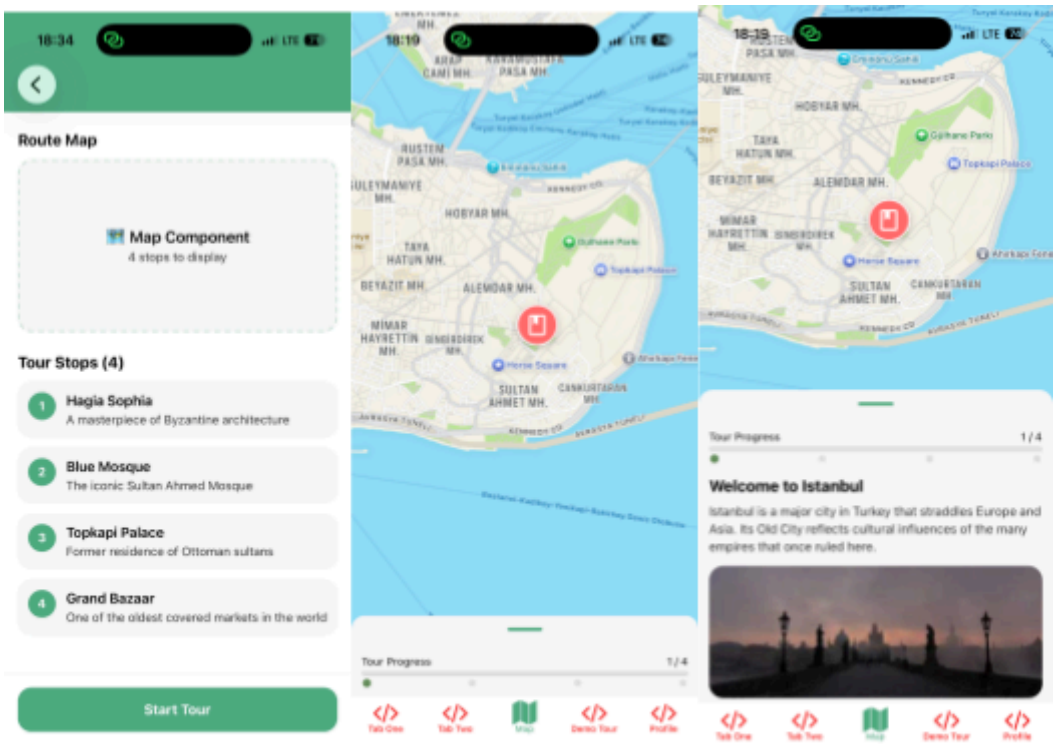

Fig. 9 Tour mockups

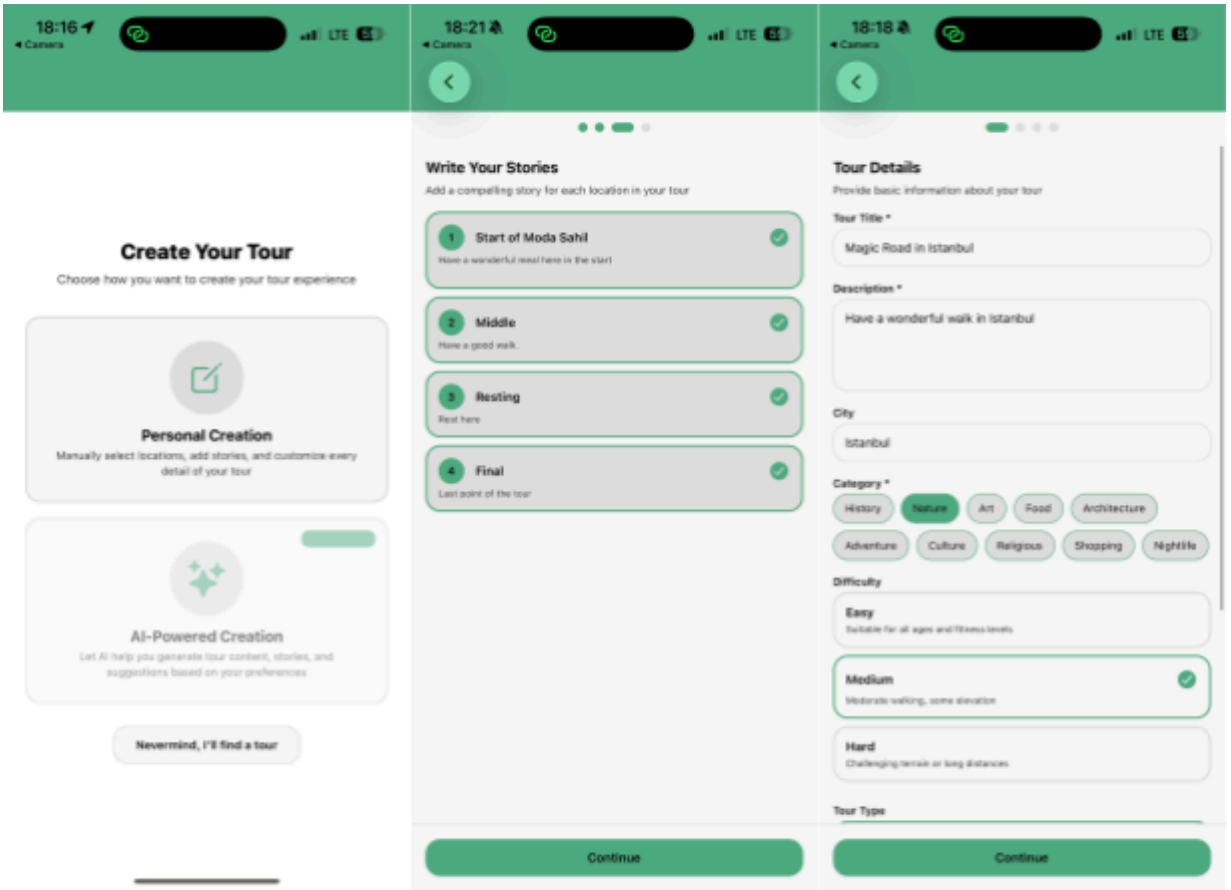

Fig. 10 Creating tour

Fig. 11 Editing tour steps

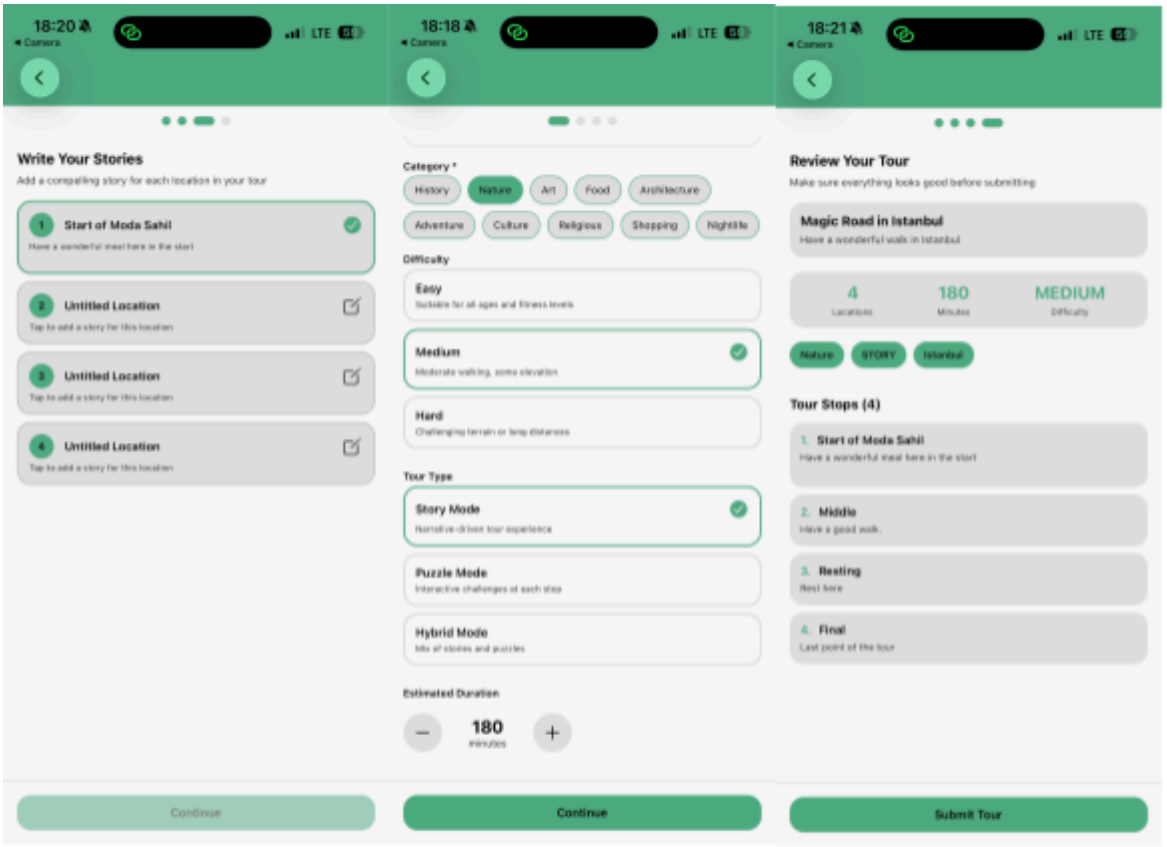

Fig 12. Create Tour Cnt.

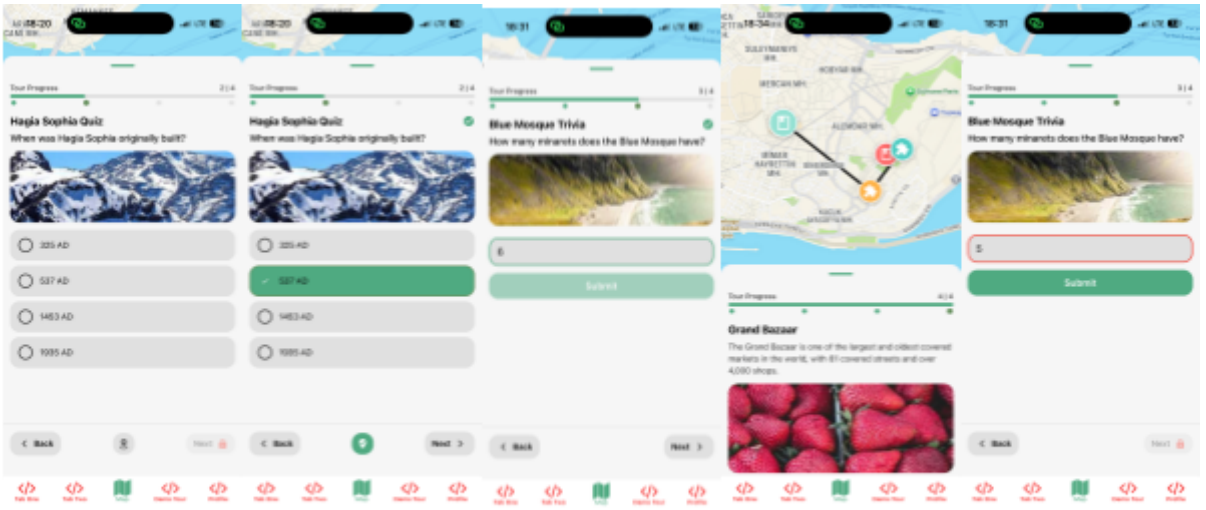

Fig. 13 Tour puzzle mechanics

4. Other Analysis Elements

4.1. Consideration of Various Factors in Engineering Design

4.1.1. Constraints

4.1.1.1. Economic Constraints

- The cost of Generative AI (LLM) tokens must be monitored and limited to prevent exceeding the free trial limits or a fixed monthly budget.
- Development and testing are restricted to the personal computers and mobile devices currently owned by the project team. No new hardware will be purchased.
- The backend infrastructure must be deployed on platforms offering free student tiers during the initial phase.
- The economic model is constrained by the standard 15-30% platform fee deducted by app stores from all user transactions.

4.1.1.2. Technical Constraints

- The application will be developed strictly for Cross-Platform using React Native.
- The core “AI Storytelling” feature requires an active internet connection (4G/5G/Wi-Fi) to function. This functionality should not be allowed in offline mode.

- The user experience is constrained by the hardware limitations of mobile GPS sensors, limited to 5-10 meters of accuracy.
- The application must not consume more than 20-25% of the device battery per hour of active gameplay.
- The AI generation process must return a story segment within 60 seconds to maintain game immersion and ensure seamless gameplay.

4.1.1.3. Legal and Ethical Constraints

- The system must comply with GDPR and KVKK regulations regarding the storage of user location data. Location history must be anonymized or deleted after a set period.
- The system is constrained to filter out user-generated content that contains hate speech, explicit content, or illegal references.
- The application must include disclaimers stating that historical "thriller" elements generated by AI may not be factually accurate.
- The project is constrained by the intellectual property rights of map providers. Their logos and attribution must be visible.

4.1.1.4. Environmental and Safety Constraints

- The game logic is constrained to prevent generating clues or waypoints in hazardous areas.
- The User Interface (UI) must use high-contrast colors to ensure readability under bright outdoor sunlight.
- Outdoor urban environments can be too noisy for audio-only guidance. The application must provide text captions for all audio narrations.
- The app must limit user interaction while the GPS detects movement speeds above walking pace (e.g., driving) to prevent accidents.

4.1.1.5. Implementation Constraints

- The project is limited to using specific languages that are compatible with the chosen frameworks.
- The scope of the project features is limited by the development capacity of a 5-person team.
- The project is constrained by the token context window of the chosen LLM, requiring summarization techniques for long stories.

4.1.1.6. Monetization and Policy Constraints

- Since the application sells digital goods (Premium User status, Game Credits), it is constrained to use proprietary payment gateways (Google Play Billing / Apple StoreKit) rather than generic credit card processors, in compliance with App Store Review Guidelines.

4.1.1.7. Hardware and Sensor Constraints

- The "Augmented Reality" puzzle features are constrained to devices supporting ARCore (Android 7.0+) or ARKit (iOS 11+). The system must gracefully degrade to a 2D interface for devices that are not supported.
- "Gyroscope Puzzles" are designed for devices equipped with a physical gyroscope sensor. The application must perform a hardware capability check during installation or startup.

4.1.1.8. Security and Anti-Cheat Constraints

- To protect the "Token Economy" from location faking, the application is designed to detect and block execution on rooted or jailbroken devices where GPS faking is possible.
- All calculations regarding XP gain, Token distribution, and Tour Completion must occur on the backend server. The client-side application is constrained to acting as an interface for these values to protect the integrity.

4.1.2. Standards

4.1.2.1. Documentation and Modeling Standards

- **IEEE 830-1998:** This structure is used to define all the functional and non-functional requirements in a clear way.
- **UML 2.5.1:** The "Unified Modeling Language" standard is used for the structural diagrams in the reports.

4.1.2.2. Programming and Code Style Standards

- **PEP 8:** Backend will use PEP 8 style for naming, indentation, and code structure to ensure better readability and structure.
- **TypeScript Strict Mode:** Frontend development will integrate TypeScript's strict type checking to prevent runtime errors across the React Native codebase.

4.1.2.3. Data and Communication Standards

- **HTTPS (TLS 1.3):** Data transmission between the client and the server will be encrypted using TLS to ensure data privacy.

4.1.2.4. Regulatory and Ethical Standards

- **GDPR (General Data Protection Regulation):** The application will strictly apply to Regulation 2016/679. This includes implementing a request for the user to share location, ensuring explicit user consent for location tracking, and providing mechanisms for users to request data deletion.
- **OAuth 2.0:** This open authentication standard will be used for secure social login to decrease the handling of user credentials

4.2. Risks and Alternatives

Table 1: Factors that can affect analysis and design.

	Effect level	Effect
Public health	6	Needs accessibility considerations, walking-duration safety guidance, and general health disclaimers.
Public safety	9	Needs accessibility considerations, walking-duration safety guidance, and general health disclaimers.
Public welfare	5	Inclusive design (low-end devices), offline/low-data support, fair access.
Global factors	6	Inclusive design (low-end devices), offline/low-data support, fair access
Cultural factors	6	Respectful content, avoid sensitive locations/topics, localized tone and imagery.
Social factors	7	Respectful content, avoid sensitive locations/topics, localized tone and imagery.
Environmental factors	5	Respectful content, avoid sensitive locations/topics, localized tone and imagery.
Economic factors	8	Maps/AI API costs, hosting costs, pricing/premium decisions, cost-driven performance

Table 2: Risks

	Likelihood	Effect on the project	B Plan Summary
Risk 1	8	Maps/AI API costs, hosting costs, pricing/premium decisions, cost-driven performance	Larger acceptance radius, manual “confirm location,” map pin + directions, offline hints
Risk 2	6	Features fail or become too expensive to run	Background queueing, changing LLM providers/models, temporarily disabling AI generation
Risk 3	5	Trust loss, legal/compliance risk	Store minimal location, encryption, strict permissions, logging/monitoring, security review checklist

4.3. Project Plan

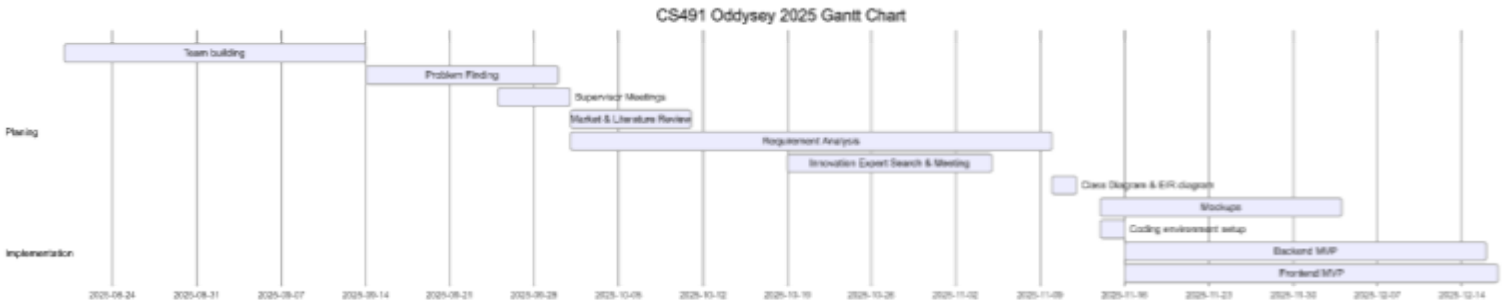

Fig 14. Gantt Chart for the Project up to 18 Dec. 2025

WP 1: *Project Initiation & Problem Definition*			
Start date: *2025-08-20* ***End date:*** *2025-10-01*			
Leader:	*Ege Ertem*	**Members involved:**	*Can Kütükoğlu,* *Sami Bora Akoğuz,* *Mehmet Rodi Aydoğdu,* *Cem Tarkan Tekcan*
Objectives: To establish the team structure, identify a viable problem domain in the tourism sector, and validate the initial project concept with the project supervisor.			
Tasks: ***Task 1.1 Team Building:*** *Forming the group and establishing communication channels (Discord/WhatsApp) and meeting schedules.* ***Task 1.2 Problem Finding:*** *Brainstorming potential project ideas and analyzing current gaps in interactive tourism.* ***Task 1.3 Supervisor Meetings:*** *Presenting initial ideas to the supervisor for feedback and obtaining approval for the "Odyssey" concept.*			
Deliverables ***D1.1:*** *Project Proposal Draft* ***D1.2:*** *Approved Project Topic*			

WP 2: *Analysis, Design & Requirements*			
Start date: *2025-10-01* ***End date:*** *2025-12-04*			
Leader:	*Can Kütükoğlu*	**Members involved:**	*Ege Ertem,* *Sami Bora Akoğuz,* *Mehmet Rodi Aydoğdu,* *Cem Tarkan Tekcan*
Objectives: To analyze the market and literature, define detailed system requirements, consult with innovation experts, and create the architectural and visual designs (UML & UI) for the application.			
Tasks: ***Task 2.1 Market & Literature Review:*** *Researching competitors (CityFans, Questo) and academic papers on AI in tourism to support the project thesis.* ***Task 2.2 Innovation Expert Search & Meeting:*** *Consulting with domain experts to refine the "AI Storytelling" innovation aspect.* ***Task 2.3 Requirement Analysis:*** *Defining Functional and Non-Functional requirements and writing the Project Specification Report.* ***Task 2.4 System Modeling:*** *Designing the Class Diagram and Entity-Relationship (E/R) diagrams for the backend.* ***Task 2.5 UI Mockups:*** *creating high-fidelity interface designs and wireframes for the mobile application.*			
Deliverables ***D2.1:*** *Project Specification Report* ***D2.2:*** *Analysis and Requirement Report* ***D2.3:*** *System Models (UML Diagrams)* ***D2.4:*** *UI/UX Mockups (Figma/Canva exports)*			

WP 3: Initial Implementation (MVP)			
Start date: *2025-11-14* ***End date:*** *2025-12-17*			
Leader:	*Sami Bora Akoğuz*	**Members involved:**	*Ege Ertem,* *Can Kütükoğlu,* *Mehmet Rodi Aydoğdu,* *Cem Tarkan Tekcan*
Objectives: To set up the development environment and produce a Minimum Viable Product (MVP) containing the core backend structure and basic frontend navigation features.			
Tasks: ***Task 3.1 Coding Environment Setup:*** *Configuring the Git repository, Docker containers, and CI/CD pipelines.* ***Task 3.2 Backend MVP:*** *Implementing the Django framework, database connections, and basic User Authentication APIs.* ***Task 3.3 Frontend MVP:*** *Initializing the React Native project and implementing the Map interface with basic location tracking.*			
Deliverables ***D3.1:*** *GitHub Repository (Initial Commit)* ***D3.2:*** *Working MVP Demo (Login, tour & Map View)*			

In the following weeks, we are planning to implement our ideas aside from the minimum viable product (MVP). These include more immersive puzzles, such as AR and gyro mechanics to solve a step, a more inclusive tour sharing social platform, real transaction handling, implementing ads and other features explained in our diagrams and functional requirements above.

4.4. Ensuring Proper Teamwork

To ensure efficient collaboration and successful project delivery, the team adopted an Agile-inspired methodology, dividing the workforce into specialized sub-teams while maintaining synchronized integration efforts.

Communication & Management The team utilized Discord as the primary platform for daily communication and voice meetings, while WhatsApp was used for urgent updates. Jira was employed for task tracking and backlog management, allowing the team to visualize progress

across different work packages. Regular synchronization meetings were held weekly to align the Backend and Frontend progress. The development team was split into two primary clusters to parallelize the implementation phase. The backend team was responsible for the Django framework setup, database schema design (PostgreSQL), and the implementation of core logic. The frontend team was responsible for the React Native mobile application, UI/UX design implementation, and the AR/Map modules.

To prevent "integration hell" at the end of the cycle, we assigned specific members (Cem Tarkan Tekcan and Can Kütükoğlu) to oversee the API integration process. This ensured that the JSON data contracts between the Django backend and the Mobile client were strictly followed and debugged in real-time.

4.4.1. Github Contributions

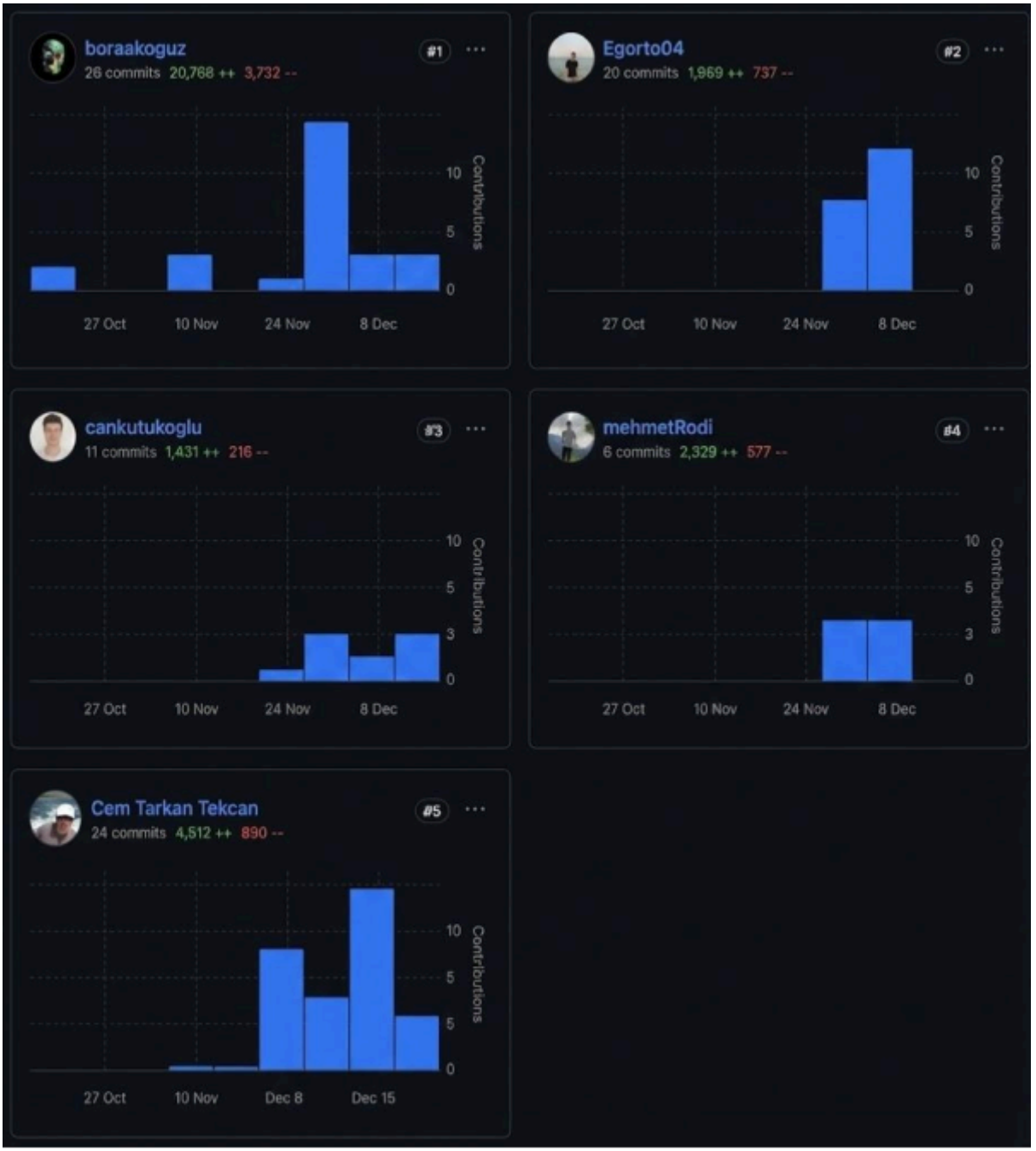

Fig 15. Github Contributions

4.4.2. Jira tickets

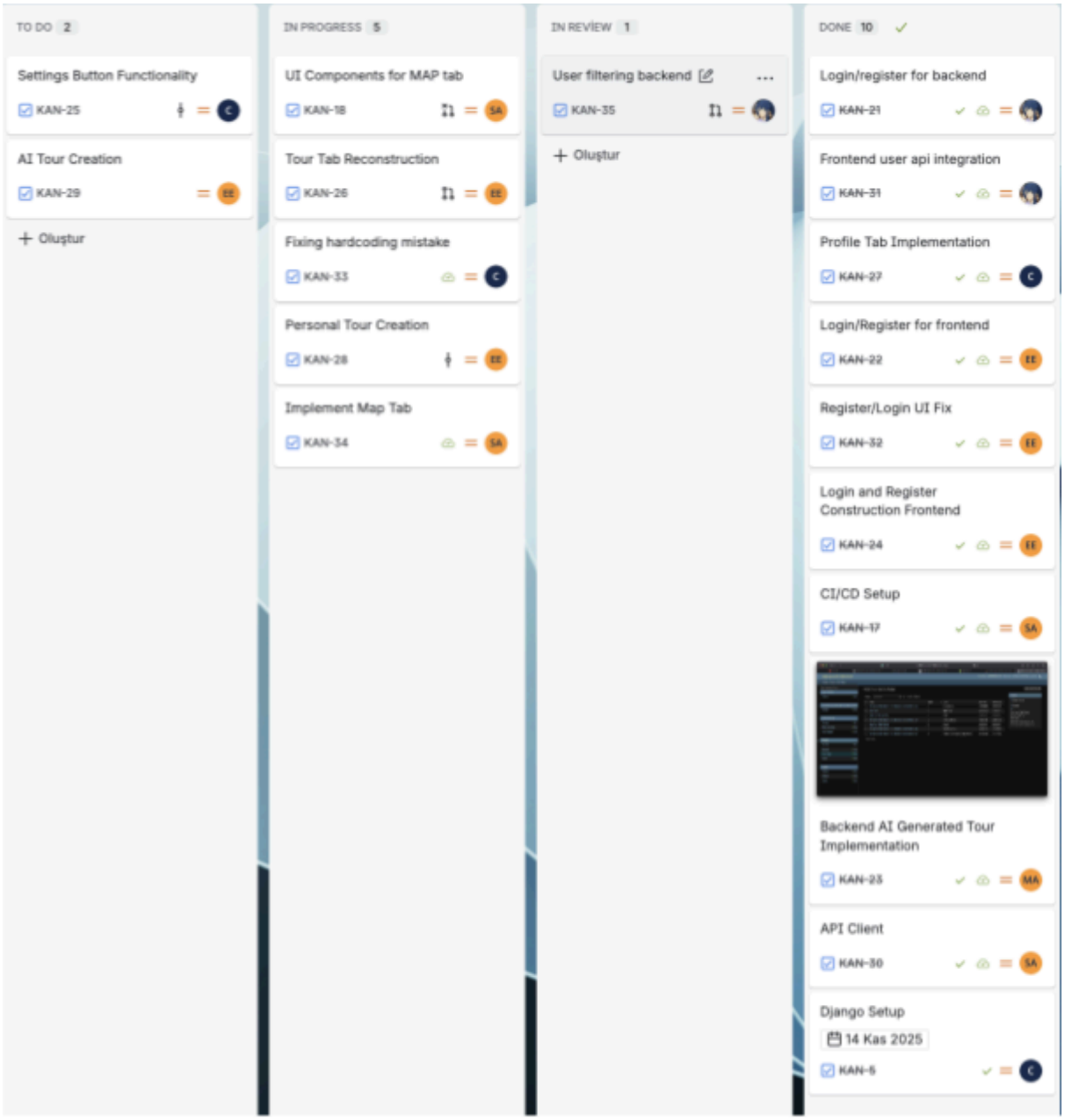

Fig 16. Jira Tickets of Odyssey up to 18 Dec. 2025

4.5. Ethics and Professional Responsibilities

4.5.1. Public Safety and Physical Liability

Since Odyssey relies on real-time location tracking via GPS and Augmented Reality (AR) mechanics, the most critical ethical concern is user safety during physical movement. This app may create a risk of “distracted walking”, which could lead to accidents while walking or in traffic. To solve this professional liability, the application shall implement a “physical awareness” warning upon each launch.

4.5.2. Data Privacy and Location Ethics

The system architecture requires the real-time location collection of the user and also creates a database that will store the detailed user information. This usage of user data could lead to a significant privacy risk if mishandled. To deal with this problem, the app will store minimal data only for the purpose of navigation and gameplay verification. Moreover, to comply with professional standards such as “KVKK”, all location history and personal information will be encrypted and access controlled with secure social login mechanisms.

4.5.3. AI Integrity and Misinformation

Odyssey’s core innovation involves using AI to generate narratives and historical storytelling. This could lead to a risk of AI misinformation. AI misinformation can decrease the trust of the user to the app and the software. There will be pre-determined structured prompts for the usage of LLM to ensure narratives stay only with factual data.

4.5.4. Accessibility and Inclusivity

This app aims to create a game out of tourism, with the reliance on physical movement, GPS navigation, and AR challenges inherently creates accessibility barriers for individuals with mobility or visual impairments. To address this ethically, the application will clearly categorize and tag tours based on their physical requirements.

4.6. Planning for New Knowledge and Learning Strategies

In order to gain new knowledge about unfamiliar topics, we have done literature research and brainstormed with each other. In addition, we have consulted to large language models for support. We have conducted meetings every week to catch up with each other and exchange information. We have also consulted our expert friends from other project groups to get their knowledge on special

fields. For instance, our choice of React Native for mobile development stemmed from our chat with a former Google intern friend.

5. Glossary

- **API (Application Programming Interface):** A set of protocols and tools that allows different software applications to communicate with each other. In this project, it refers to backend services and third-party services (like maps).
- **AR (Augmented Reality):** A technology that superimposes a computer-generated image on a user's view of the real world, utilized in the project for puzzle mechanics.
- **Blue-Green Deployment:** A technique that reduces downtime and risk by running two identical production environments (Blue and Green). Only one serves live traffic while the other is updated.
- **Cold-Start:** The time it takes for the application to load from a completely closed state to a fully interactive state.
- **Cross-Platform:** Software developed to work on multiple mobile operating systems (iOS and Android) using a single code base (in this case, React Native).
- **Datadog / Sentry:** Monitoring and observability platforms used to track application errors, performance metrics, and logs.
- **GDPR (General Data Protection Regulation):** A legal framework that sets guidelines for the collection and processing of personal information from individuals who live in the European Union.
- **Generative AI:** A type of artificial intelligence technology capable of generating text, images, or other media in response to prompts. Used here to create "Story Mode" narratives.
- **Geocaching:** An outdoor recreational activity, in which participants use a Global Positioning System (GPS) receiver or mobile device and other navigational techniques to hide and seek containers.
- **GPS (Global Positioning System):** A satellite-based navigation system used to determine the ground position of an object.
- **Hotfix:** A small piece of code developed to correct a specific bug in a software product that is already live in production, usually requiring immediate attention.

- **JSON (JavaScript Object Notation):** A lightweight data-interchange format that is easy for humans to read and write and easy for machines to parse and generate. Used here for structuring AI puzzle outputs.
- **KVKK (Kişisel Verilerin Korunması Kanunu):** The Turkish Law on the Protection of Personal Data, roughly the Turkish equivalent of GDPR.
- **LLM (Large Language Model):** A deep learning algorithm that can recognize, summarize, translate, predict, and generate text and other content based on knowledge gained from massive datasets.
- **Localization:** The process of adapting the application interface and content to a specific locale or market (e.g., translating text from English to other languages).
- **Mocked Data:** Artificial data used during testing to simulate real-world conditions (e.g., Mocked GPS data allows testing location features without walking around physically).
- **Model Context Protocol (MCP):** An open standard that provides a universal interface for AI models to securely discover and access external data, tools, and resources.
- **OpenAPI (Swagger):** A specification for building, documenting, and consuming RESTful web services.
- **POI (Point of Interest):** A specific point location that someone may find useful or interesting, such as a landmark, museum, or puzzle location.
- **React Native:** An open-source UI software framework created by Meta Platforms, used to develop applications for Android and iOS by enabling developers to use React along with native platform capabilities.
- **Rolling Deployment:** A deployment strategy where the new version of an application replaces the old version across the server fleet one by one or in batches, ensuring zero downtime.
- **Token (AI):** The basic unit of text (part of a word) that an LLM processes. Usage costs are calculated based on the number of tokens processed.
- **Token (Game Currency):** A virtual currency within the application used to unlock tours or buy hints, distinct from AI tokens.
- **XP (Experience Points):** A unit of measurement used in the game to quantify a user's progression and level advancement.

6. References

[1] *Cityfans*. CityFans [Online]. Available: https://cityfans.com/en. [Accessed: 16-Nov-2025]

[2] *Questo*. Questo, 12-Apr-2025. [Online]. Available: https://questoapp.com/. [Accessed: 16-Nov-2025]

[3] "FAQ", *Explorial*. Explorial. [Online]. Available: https://explorial.com/ [Accessed: 16-Nov-2025]