Functional Specification Contents

- Project title: DAA Scheduling Application
- Student Name: Lucas Harper
- Student ID: 21331096
- Supervisor: Boualem Benatallah

Table of contents

1. Introduction

1.1 Overview

The app I am developing is a tool to help coordinators (A role in the airport for people to manage breaks) keep track of who's on duty, who's on break, and which positions the officers are placed in. It is designed to replace the old-school methods of paper schedules and printed out excel sheets, which can be a hassle to update and keep organized. By going digital, the app makes it easy for coordinators to view, assign, and adjust staff roles on the fly, making sure everyone is where they need to be, when they need to be there, etc.

1.2 Business Context

The Dublin Airport Authority (DAA) is the organization sponsoring the development of this app. This app is being developed with a focus on addressing challenges faced by

the Airport Security Unit (ASU). Coordinating breaks and availability in real time is crucial to ensuring that the operation runs smoothly and that passengers experience minimal delays. By replacing outdated, paper-based scheduling systems with a modern, digital solution, this tool perfectly fits with the DAA's commitment to a fluid operation and enhanced passenger satisfaction.

1.3 Glossary

ASU (Airport Security Unit): The team responsible for maintaining security at the airport, including managing passenger screening, access control, and other security tasks.

React: A JavaScript library for building user interfaces. React allows developers to create reusable UI components and enables efficient updates to the user interface when data changes. Used in this project to build the front-end application.

Django: A high-level Python web framework that enables rapid development of secure and maintainable web applications. Used in this project to build the back-end, including APIs for data handling and database management.

Axios: A JavaScript library used to make HTTP requests from the React front end to the Django backend to fetch or update data.

2. General Description of the Project

2.1 Product / System Functions

The system is designed to simplify and optimize staff scheduling and management for the Dublin Airport Authority (DAA). It provides several functionalities aimed at streamlining operations and improving efficiency:

Staff Management

- **Drag-and-Drop Assignment:** Staff members can be easily assigned to tasks or positions (e.g., on-duty roles, breaks) using a drag-and-drop interface.
- **Categorizating Staff:** The system categorizes staff into distinct groups such as "On Duty," "On Break," and "Finished" making it easier to monitor and adjust resource allocation in real time.

Scheduling and Break Management

- **Custom Break Schedules:** Coordinators can set dynamic break schedules with timers tailored to operational needs. The system ensures compliance with labor regulations such as the first break has to be before 4 and a half hours and avoids overlapping breaks that could disrupt operations. (Link to labor laws in appendices)

- **Visual Indicators:** Staff in the "On Break" section display are easy to see and the user is notified when certain staff should be coming back from there break. The UI is also user friendly and easy to use.

Rollcall Feature

- **Live Updates:** Automatically syncs with the database to show real-time staff availability for new assignments based on who is working eg: 3:50am staff will appear at 3:40am and 5:00am staff will appear at 4:50am.
- **Quick Deployment:** Coordinators can quickly assign staff members from the rollcall pool to active positions as needs arise.

Seamless Integration

- **Backend Integration:** Built with Django, the system efficiently handles data storage and retrieval for all scheduling operations. APIs ensure secure communication between the front end and back end through the implementation of industry-standard security practices, including HTTPS for encrypted data transmission, token-based authentication (such as OAuth 2.0 or JSON Web Tokens), rate limiting to prevent abuse, and robust input validation to mitigate injection attacks. Sensitive information is encrypted at rest and in transit, ensuring end-to-end security.
- **Real-Time Data Sync:** React components dynamically update to reflect changes in staff roles or schedules, ensuring all users see the latest data without manual refresh. This data is imported daily at 4am from an excel sheet which is made up outside of this project.

Reporting and Monitoring

- **Shift Duration Display:** The app calculates and displays the duration of assigned shifts for better time management.
- **Overtime Alerts:** Flags staff who exceed their scheduled break or shift time, helping coordinators stay notified.

Usability and Accessibility

- **Mobile-Friendly Interface:** Designed to work on tablets or mobile devices for on-the-go tasks.
- **Customizable Views:** Coordinators can toggle between views to focus on specific tasks, such as rollcall or break management. It is extremely accessible and can is tailored to different Coordinators needs

2.2 User Characteristics and Objectives

The primary users of this system are supervisors, coordinators, and duty managers within the Dublin Airport Authority (DAA).

Supervisors and Coordinators:

- **Technical Proficiency:** Most Supervisors and Coordinators have moderate technical skills and are accustomed to using traditional scheduling tools like Excel or paper-based systems. While some may not be highly experienced with software, they are familiar with basic drag-and-drop and form-based interfaces.
- **Domain Expertise:** Supervisors and Coordinators have a deep understanding of staff requirements, operational priorities, and regulatory guidelines for breaks and shift durations.
- **Time-Sensitive Needs:** Supervisors and Coordinators responsibilities are time-critical, requiring fast, accurate updates to staff assignments in a high-pressure environment.

Duty Managers:

- **Broader Scope:** Managers have a higher-level view of operations and require insights into resource allocation across multiple teams. They are less involved in day-to-day scheduling but rely on reporting and monitoring tools for strategic decision-making.
- **Experience with Systems:** While comfortable with workplace tools, they may require simplified interfaces and reporting dashboards that are clear and require quick access to critical information.

Improve Efficiency:

- Reduce the time spent updating schedules.
- Simplify the process of reassigning staff roles during busy periods or emergencies.

Enhance Accuracy:

- Minimize errors in break timing or staff allocation.
- Provide real-time updates to ensure all users have consistent information.

Ensure Compliance:

- Automatically enforce regulatory requirements for break durations and shift lengths.
- Alert users to potential compliance issues, such as missed or overdue breaks.

Enable Better Monitoring:

- Provide clear visibility of who is on duty, on break, or who is finished.

- Display countdown timers and shift durations for effective time management.

Support Quick Decision-Making:

- Allow supervisors to respond to changes in real-time without delays caused by outdated systems or manual processes.

User Wish List Features

- **Mobile Compatibility:** A version of the system that works on tablets and smartphones for supervisors on the move.
- **Real-Time Notifications:** Alerts for changes in staff assignments, overdue breaks, or new rollcall needs.
- **Customizable Views:** The ability to tailor the interface based on the specific needs of the user, such as a break-focused view or a rollcall-focused view.

While the "wish list" features are desirable, the following solutions are more feasible in the immediate term:

- A browser-based platform optimised for both desktop and tablet use.
- Real-time data syncing to ensure updates are reflected instantly without manual refresh.
- Automated alerts for overdue breaks or shifts within the existing interface.
- A user-friendly drag-and-drop interface for staff management.

2.3 Operational Scenarios

Scenario 1: Managing Daily Staff Assignments

Actors: Coordinator
Context: At the start of a shift, the Coordinator is tasked with assigning roles to staff members based on operational needs.
Steps:

1. The Coordinator logs into the system and navigates to the "On Duty" section.
2. They review the list of available staff members in the "Rollcall" section.
3. Using the drag-and-drop functionality, they assign staff members to the "On Duty" section based on their roles and shift timings.
4. Once the assignments are complete, the system automatically updates and notifies coordinators about the new assignments.

Expected Outcome:

- Staff are assigned to their roles efficiently.
- Real-time updates ensure coordinators and other stakeholders are informed instantly.

Scenario 2: Responding to Peak Periods

Actors: Coordinator, Supervisor
Context: A sudden surge in passenger volume requires additional security staff at specific positions
Steps:

1. The coordinator identifies available staff in the "On Duty" section who are back from break and are not assigned to critical roles.
2. They reassign these staff members to the required points using the drag-and-drop interface.
3. If additional staff are needed, they pull personnel from the "Staff (SED)" section as an example ensuring compliance with shift rules.
4. The supervisor would then check over this ensuring the operation is going smoothly.

Expected Outcome:

- The increased demand is addressed quickly with minimal disruption.
- Compliance with break schedules and shift durations is maintained.

Scenario 3: Managing Break Schedules

Actors: Coordinator
Context: Staff members need to be assigned breaks at appropriate intervals to comply with labor regulations.
Steps:

1. The coordinator selects a staff member from the "On Duty" section and drags them into the "On Break" section.
2. The system prompts the coordinator to set a break duration and start time.
3. Once the break is confirmed, the system starts a countdown timer and alerts the coordinator when the break is about to end.
4. The staff member is moved back to the "On Duty" section when the break is completed.

Expected Outcome:

- Break schedules are managed seamlessly without manual tracking.
- Staff compliance with regulatory requirements is ensured.

Scenario 4: Rollcall at Shift Start

Actors: Coordinator
Context: A new shift begins, and staff need to be accounted for and assigned roles.
Steps:

1. The Coordinator opens the "Rollcall" section and reviews the list of available staff.
2. Each staff member is assigned to their respective roles in the "On Duty" section.
3. The system automatically updates the list, removing assigned staff from the "Rollcall" section.

Expected Outcome:

- The shift begins on time with all staff assigned to their roles.
- The process is streamlined and avoids errors.

2.4 Constraints

Performance Constraints

- **Real-Time Updates:** The system must process and display staff assignments, break schedules, and rollcall updates in real time to ensure operational efficiency and timely decision-making.
- **Scalability:** The platform should handle a growing number of staff members and operations without degradation in performance, particularly during peak travel periods.

Compliance Constraints

- **Labor Regulations:** The system must adhere to labor laws and DAA-specific protocols, such as tracking break durations, shift limits, and role assignments, to avoid compliance violations.
- **Data Security:** Sensitive staff information, such as schedules and assignments, must be securely stored and transmitted to comply with GDPR and other applicable data protection regulations.

Integration Constraints

- **Existing Systems:** The system must integrate seamlessly with existing DAA systems to ensure data consistency.

Usability Constraints

- **Device Compatibility:** The platform should be accessible across various hardware, including desktop computers, tablets, and mobile devices, used by supervisors, coordinators, and duty managers.
- **User Expertise:** The system must cater to users with varying levels of technical expertise, ensuring ease of use through a simple and intuitive interface.

Development Constraints

- **Technology Stack:** The system must be developed using React for the frontend and Django for the backend.
- **Timeline:** The project must be completed within a specified timeframe to allow for a vast amount of testing with operational goals in mind, such as seasonal travel demands or regulatory deadlines.

Operational Constraints

- **24/7 Availability:** As an operational tool for DAA staff, the system must remain functional 24/7 to accommodate around-the-clock operations, particularly during peak hours.
- **Offline Support:** The system should include limited offline capabilities, such as data caching, to ensure functionality in case of temporary network outages.

3. Functional Requirements

Dynamic Staff Assignment

- **Description:** The system must allow for real-time assignment and reassignment of staff to roles and locations based on operational needs.
- **Criticality:** Essential – Central to the system's primary purpose.
- **Technical Issues:** Requires integration with a database to manage staff profiles and operational roles effectively. Real-time updates necessitate efficient data handling and low-latency performance.
- **Dependencies with Other Requirements:** Dependent on the user interface for drag-and-drop functionality and real-time data synchronization.

Break Scheduling

- **Description:** The system must facilitate the scheduling of staff breaks while ensuring compliance with labor laws and minimizing disruption to operations.
- **Criticality:** Essential – Prevents compliance violations and ensures smooth operations.
- **Technical Issues:** Implementation of timers and alerts to track break durations and manage staggered scheduling. Integration with regulatory compliance rules is required.
- **Dependencies with Other Requirements:** Relies on staff profile data and the drag-and-drop interface for scheduling.

Rollcall Management

- **Description:** The system must maintain a real-time rollcall of available staff, enabling supervisors to quickly identify and assign staff to tasks.
- **Criticality:** High – Critical for efficient management during high-demand periods.
- **Technical Issues:** Real-time data updates and reliable database queries are needed to manage rollcall functionality.
- **Dependencies with Other Requirements:** Requires integration with the staff assignment and break scheduling modules.

Drag-and-Drop Interface

- **Description:** The system must support an intuitive drag-and-drop interface for assigning staff, managing schedules, and updating roles.
- **Criticality:** High – Enhances usability and efficiency for coordinators and supervisors.
- **Technical Issues:** Requires React-based front-end development with a robust state management system to ensure smooth interactions.
- **Dependencies with Other Requirements:** Integral to dynamic staff assignment and break scheduling features.

User Access Management

- **Description:** The system must provide role-based access controls, ensuring that only authorized personnel can manage assignments or view sensitive data.
- **Criticality:** Medium – Important for maintaining system integrity and security.
- **Technical Issues:** Requires integration with Django's authentication system and configuration of user roles.
- **Dependencies with Other Requirements:** Operates independently but influences data visibility and actions for other features.

4. System Architecture

Front-End (React):

- **Description:** The user interface is built using React to ensure a responsive, interactive, experience for users. It implements drag-and-drop functionality for staff assignment and schedule updates.
- **Functions:**
    - Displays real-time staff data, schedules, and roles.
    - Provides a drag-and-drop interface for task assignments and break scheduling.
    - Offers role-based access views (e.g., supervisors vs coordinators).
- **3rd Party Libraries Used:**
    - React-DND for drag-and-drop capabilities.
    - Axios for API communication with the back-end.
    - CSS modules for styling.

Back-End (Django):

- **Description:** The back-end is built using Django, providing the system's core logic, data management, and API endpoints.
- **Functions:**
    - Handles user authentication and role-based access control.
    - Manages real-time updates to staff schedules, breaks, and roles.
    - Implements compliance monitoring and alert systems.
- **3rd Party Components:**
    - Django REST Framework (DRF) for API creation.
    - Django's built-in user authentication for secure access management.

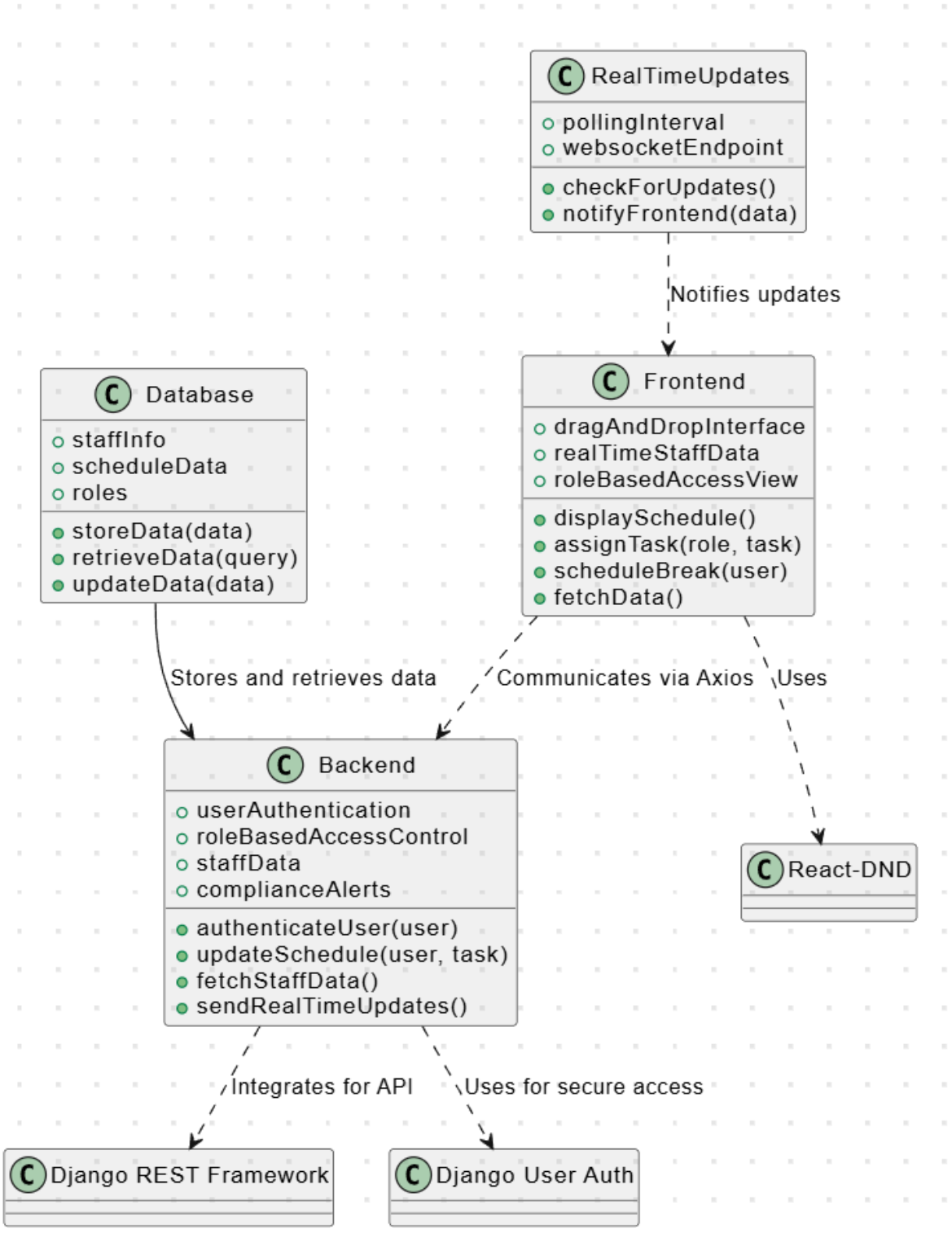
Data Flow:

- User actions on the front-end (e.g., assigning a staff member to a role) trigger API calls to the back-end.
- The back-end processes these requests, updates the database, and returns responses to the front-end.
- The front-end updates the UI dynamically based on the server response.

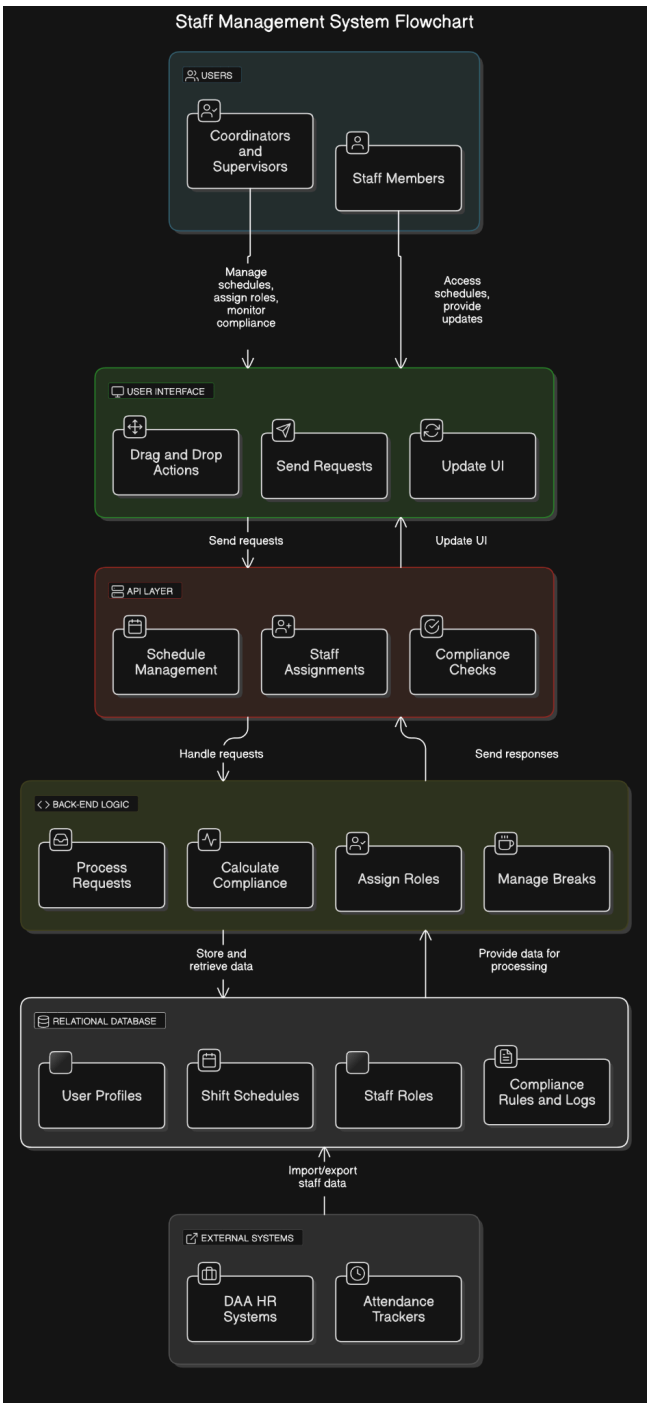
Real-Time Updates:

- The system periodically checks for updates (via polling or WebSocket-based notifications) to ensure data remains current across devices and users.

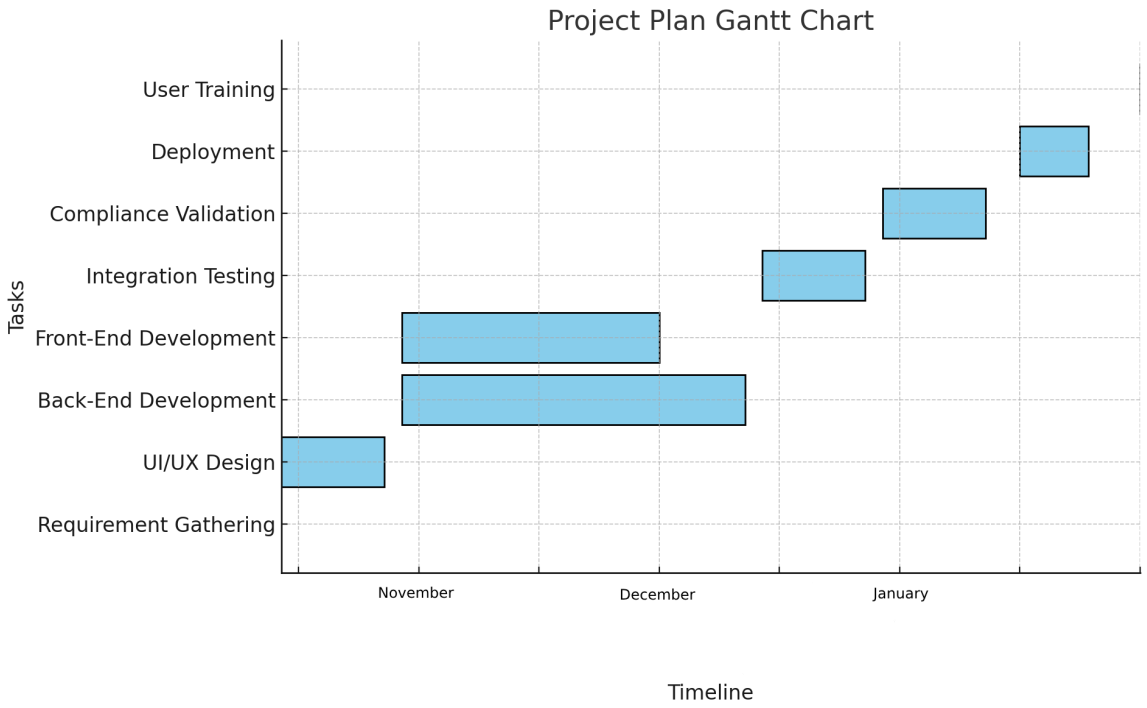
  Class Diagram of the System Architecture:



5. High-Level Design



6. Preliminary Schedule



7. Appendices

React: https://react.dev/

Django: https://www.djangoproject.com/

DAA Code of conduct: https://www.daa.ie/our-company/code-of-conduct/

Labor Laws:
https://www.citizensinformation.ie/en/employment/employment-rights-and-conditions/hours-of-work/work-breaks-and-rest-periods/