CEBU INSTITUTE OF TECHNOLOGY
UNIVERSITY

IT342-Section SYSTEMS INTEGRATION AND ARCHITECTURE 1

FUNCTIONAL REQUIREMENTS SPECIFICATION (FRS)

Project Title:

Prepared By: Ryan Noay

Date of Submission:

Version: 1.0

Table of Contents

1. Introduction 3
   1.1. Purpose 3
   1.2. Scope 3
   1.3. Definitions, Acronyms, and Abbreviations 3
2. Overall Description 3
   2.1. System Perspective 3
   2.2. User Classes and Characteristics 3
   2.3. Operating Environment 3
   2.4. Assumptions and Dependencies 3
3. System Features and Functional Requirements 3
   3.1. Feature 1: 3
   3.2. Feature 2: 3
4. Non-Functional Requirements 3
5. System Models (Diagrams) 4
   5.1. ERD 4
   5.2. Use Case Diagram 4
   5.3. Activity Diagram 4
   5.4. Class Diagram 4
   5.5. Sequence Diagram 4
6. Appendices 4

- ## Introduction

  - ### Purpose

    BusPay is a smart ticketing system designed to modernize the fare collection process of Public Utility Jeepneys (PUJs) and other Public Utility Vehicles (PUVs). BusPay aims to provide a hassle-free commuting experience for passengers while reducing the workload and stress of conductors through a digital, efficient, and secure payment solution.

  - ### Scope

    BusPay is a mobile-based smart ticketing application that allows passengers to pay fares electronically using a tap-to-pay system integrated with QR code or NFC technology.

The scope of the system includes:

2.1 Passenger Features

- User registration and secure login
- Account profile management
- Cashless fare payment via QR code or NFC tap
- Real-time balance checking
- Digital transaction history
- In-app wallet with cash-in functionality
- Fare calculation based on route and distance (if applicable)

2.2 Conductor/Operator Features

- Fare verification system
- Real-time payment confirmation
- Daily transaction monitoring
- Route and trip management
- Passenger payment validation

2.3 Administrative Features

- User account management
- Fare rate configuration
- Transaction monitoring and reporting
- System analytics and performance tracking
- Security and fraud detection monitoring

2.4 System Boundaries

BusPay covers the digital payment and fare management system only. It does not include:

- Physical vehicle management
- Maintenance scheduling

- Fleet ownership management (unless integrated in future versions)
- Non-digital payment systems (e.g., traditional cash-only systems)

The system is intended for deployment on Android and iOS mobile platforms and may integrate with external payment gateways and financial services for wallet funding.

○ Definitions, Acronyms, and Abbreviations

Definitions

- **E-commerce:** The buying and selling of goods or services over the internet.
- **Electronic Payment (E-payment):** A method of paying for goods or services electronically through payment gateways or online banking.
- **Client-Server Architecture:** A system design where the client (frontend) sends requests to a server (backend), which processes and returns the results.
- **RESTful API:** A web service that allows communication between frontend and backend systems using HTTP requests and responses.
- **Product Catalog:** A collection of product information including names, descriptions, images, and prices displayed to customers.
- **Shopping Cart:** A feature that allows users to select and temporarily store products before completing a purchase.
- **Transaction Processing:** The execution and recording of a purchase or payment in the system.
- **User Authentication:** A security process that verifies the identity of a user before granting access to the system.

Acronyms

- **API:** Application Programming Interface
- **UI:** User Interface
- **UX:** User Experience

- **DBMS:** Database Management System
- **HTTPS:** HyperText Transfer Protocol Secure

Abbreviations

- **Admin:** Administrator – manages system operations and monitors users and transactions.
- **Seller / Vendor:** Local business or individual selling products in the system.
- **Customer / Buyer:** End-user purchasing products through the system.
- **NFR:** Non-Functional Requirements

Overall Description

System Perspective

BusPay is a standalone mobile smart ticketing application designed to modernize the fare payment system of Public Utility Jeepneys (PUJs) and other Public Utility Vehicles (PUVs). The system replaces traditional cash-based fare collection with a digital tap-to-pay solution using QR codes or NFC technology.

The application operates within a larger transportation ecosystem that includes passengers, conductors, transport operators, and financial service providers. BusPay integrates with:

- Mobile devices (smartphones of passengers and conductors)
- A centralized cloud-based server for transaction processing
- External payment gateways or e-wallet services for cash-in functionality
- Internet connectivity for real-time transaction validation

The system consists of:

- A **Passenger Mobile Application**
- A **Conductor/Operator Application**
- A **Web-based Admin Dashboard**
- A centralized database and backend server

BusPay interacts with external systems such as banking APIs, e-wallet services, and network providers to ensure secure and real-time payment processing.

- ### User Classes and Characteristics

Passengers

Description: Commuters who use PUJs/PUVs and want a faster, cashless way to pay fares.
Characteristics:

- Basic to intermediate smartphone knowledge
- Owns an Android or iOS device
- May or may not be familiar with digital wallets
- Requires simple and easy-to-use interface

Conductors/Drivers

Description: Personnel responsible for verifying and managing passenger fares.
Characteristics:

- Basic smartphone knowledge
- Needs fast payment verification
- Requires real-time transaction confirmation
- May operate in areas with unstable internet connection

Transport Operators

Description: Fleet owners or managers monitoring daily operations.
Characteristics:

- Requires access to transaction reports
- Needs summary of revenue per vehicle or route
- Moderate familiarity with digital systems

System Administrator

Description: Personnel responsible for managing and maintaining the system.
Characteristics:

- Advanced technical knowledge
- Manages user accounts, fare configurations, and reports
- Monitors system security and fraud detection

- ### Operating Environment

Hardware Requirements

- Smartphone (Android 8.0+ or iOS 13+) for passengers
- Smartphone or tablet device for conductors
- Computer or laptop for admin dashboard access
- Cloud server for backend processing
- Optional NFC-enabled devices (if NFC is implemented)

Software Requirements

- Android or iOS mobile operating systems
- Web browser (Google Chrome, Microsoft Edge, Safari) for admin dashboard
- Backend server environment (e.g., Node.js, Java Spring Boot, or similar)
- Database management system (e.g., MySQL, PostgreSQL, or MongoDB)

Network Requirements

- Stable internet connection (4G/5G/WiFi)
- Secure HTTPS communication protocol
- API integration with payment gateways or e-wallet services

○ Assumptions and Dependencies

Assumptions

- Users own compatible smartphones.
- Passengers are willing to adopt cashless payment methods.
- Transport operators approve digital fare integration.
- Conductors are trained to use the application.
- Internet connectivity is available in most operating areas.

Dependencies

- Integration with third-party payment gateways or e-wallet providers.
- Availability of cloud hosting services.
- Government regulations regarding digital payments and public transportation.
- Mobile device hardware capability (camera for QR scanning, NFC support if used).
- Compliance with data privacy laws (e.g., local Data Privacy Act).

● System Features and Functional Requirements

○ Feature 1: User Account Management

Description:

This feature allows passengers, conductors, and administrators to create, access, and manage their accounts securely within the BusPay system. It ensures proper authentication and profile management for all users.

Functional Requirements:

- **FR-1.1:** The system shall allow passengers to register using a mobile number and/or email address.

- **FR-1.2:** The system shall require users to create a secure password during registration.
- **FR-1.3:** The system shall send a verification code (OTP) for account confirmation.
- **FR-1.4:** The system shall allow users to log in using registered credentials.
- **FR-1.5:** The system shall allow users to reset their password in case of forgotten credentials.
- **FR-1.6:** The system shall allow users to update their profile information.
- **FR-1.7:** The system shall securely store user data in the centralized database.
- **FR-1.8:** The system shall log out users after a period of inactivity.

○ Feature 2: Digital Fare Payment (Tap-to-Pay System)

Description:

This feature enables passengers to pay fares digitally using QR code scanning or NFC tap technology. It ensures fast, secure, and real-time transaction processing between passengers and conductors.

Functional Requirements:

- **FR-2.1:** The system shall generate a unique QR code for each passenger account.
- **FR-2.2:** The system shall allow conductors to scan passenger QR codes to process payments.
- **FR-2.3:** The system shall support NFC-based tap payment if supported by the device.
- **FR-2.4:** The system shall automatically deduct the correct fare amount from the passenger's wallet balance.
- **FR-2.5:** The system shall display real-time payment confirmation to both passenger and conductor.
- **FR-2.6:** The system shall prevent transactions if the passenger's wallet balance is insufficient.
- **FR-2.7:** The system shall record each completed transaction in the database.
- **FR-2.8:** The system shall generate a digital receipt after successful payment.

○ Feature 3: In-App Wallet and Cash-In System

Description:

This feature allows passengers to store digital funds within the BusPay application and add money through integrated payment gateways or partner services.

Functional Requirements:

- **FR-3.1:** The system shall provide a digital wallet for each passenger account.
- **FR-3.2:** The system shall allow passengers to view their current wallet balance in real time.

- **FR-3.3:** The system shall allow passengers to cash in funds through integrated e-wallet or banking services.
- **FR-3.4:** The system shall update wallet balance immediately after successful cash-in.
- **FR-3.5:** The system shall record all wallet transactions (cash-in and deductions).
- **FR-3.6:** The system shall notify users of successful or failed wallet transactions.

Feature 4: Transaction History and Reporting

Description:

This feature allows users and administrators to view transaction records for monitoring, tracking, and auditing purposes.

Functional Requirements:

- **FR-4.1:** The system shall allow passengers to view their transaction history.
- **FR-4.2:** The system shall display transaction details including date, time, route, and fare amount.
- **FR-4.3:** The system shall allow conductors to view daily collected fare summaries.
- **FR-4.4:** The system shall allow administrators to generate system-wide transaction reports.
- **FR-4.5:** The system shall allow filtering of reports by date, route, or vehicle.
- **FR-4.6:** The system shall store transaction records securely for audit purposes.

Non-Functional Requirements

This section defines the quality attributes and operational standards that BusPay must meet to ensure system efficiency, security, and user satisfaction.

Performance Requirements

- **NFR-1.1:** The system shall process fare payment transactions within 3 seconds under normal network conditions.
- **NFR-1.2:** The system shall support at least 10,000 concurrent users without performance degradation.
- **NFR-1.3:** The system shall update wallet balances in real time after each transaction.
- **NFR-1.4:** The system shall load main application screens within 2 seconds.
- **NFR-1.5:** The system shall maintain stable performance during peak commuting hours.

Security Requirements

- **NFR-2.1:** The system shall use secure authentication mechanisms (e.g., encrypted passwords).
- **NFR-2.2:** The system shall encrypt all sensitive data during transmission using HTTPS/SSL protocols.

- **NFR-2.3:** The system shall securely store user credentials and financial data in encrypted format.
- **NFR-2.4:** The system shall implement role-based access control (Passenger, Conductor, Admin).
- **NFR-2.5:** The system shall automatically log out users after a defined period of inactivity.
- **NFR-2.6:** The system shall comply with applicable data privacy laws and regulations.
- **NFR-2.7:** The system shall detect and prevent duplicate or fraudulent transactions.

Usability Requirements

- **NFR-3.1:** The system shall provide a simple and user-friendly interface.
- **NFR-3.2:** The system shall support clear navigation with minimal steps to complete a transaction.
- **NFR-3.3:** The system shall provide clear error messages and guidance when transactions fail.
- **NFR-3.4:** The system shall support both English and local language (if required).
- **NFR-3.5:** The system shall provide accessibility features such as readable fonts and clear icons.

Reliability Requirements

- **NFR-4.1:** The system shall maintain 99% system availability during operational hours.
- **NFR-4.2:** The system shall automatically recover from minor failures without data loss.
- **NFR-4.3:** The system shall maintain accurate transaction records without duplication.
- **NFR-4.4:** The system shall perform regular automated backups of the database.

Compatibility Requirements

- **NFR-5.1:** The system shall operate on Android (version 8.0 and above).
- **NFR-5.2:** The system shall operate on iOS (version 13 and above).
- **NFR-5.3:** The admin dashboard shall be compatible with major web browsers (Chrome, Edge, Safari).
- **NFR-5.4:** The system shall support devices with QR camera functionality and optional NFC capability.

Scalability Requirements

- **NFR-6.1:** The system shall support expansion to additional routes and transport operators.
- **NFR-6.2:** The system shall allow integration with additional payment gateways in the future.
- **NFR-6.3:** The system architecture shall support increased user growth without major redesign.

Maintainability Requirements

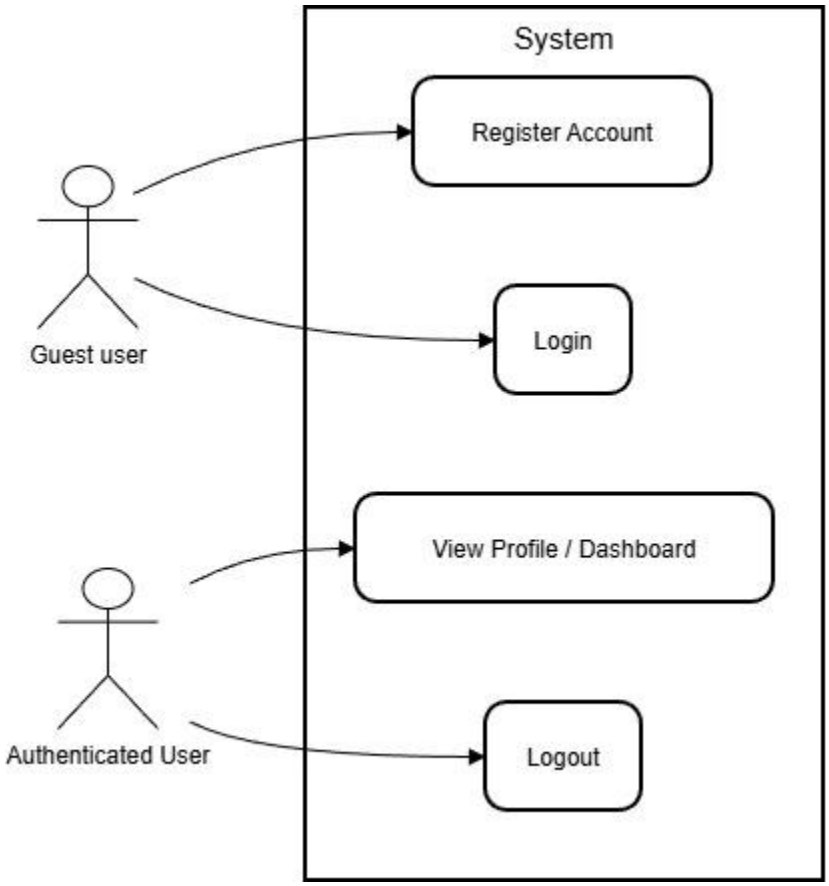
- **NFR-7.1:** The system shall be modular to allow easier updates and feature enhancements.
- **NFR-7.2:** The system shall log system errors for debugging and monitoring.
- **NFR-7.3:** The system shall allow updates without affecting stored transaction data.

System Models (Diagrams)

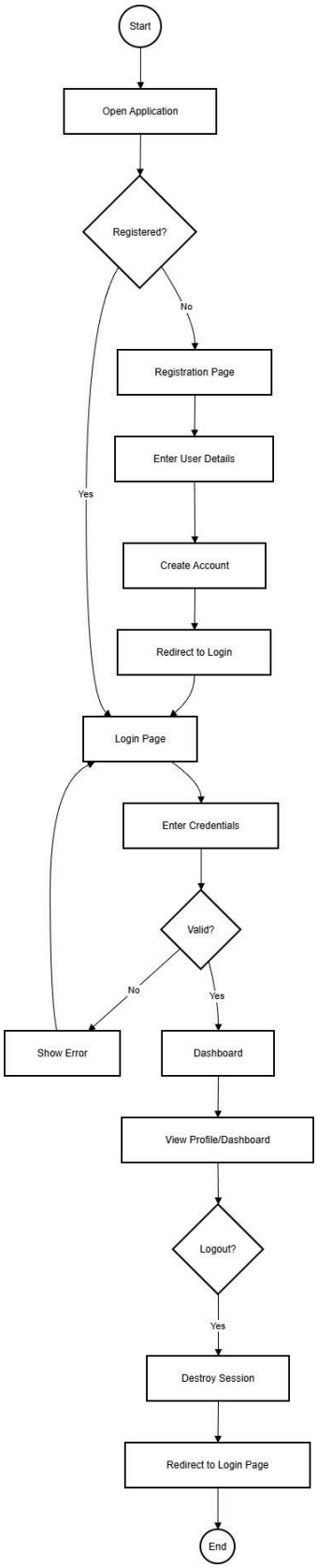
ERD

USER		
int	user_id	PK
varchar	username	
varchar	email	
varchar	password_hash	
datetime	created_at	
datetime	last_login	
enum	status	

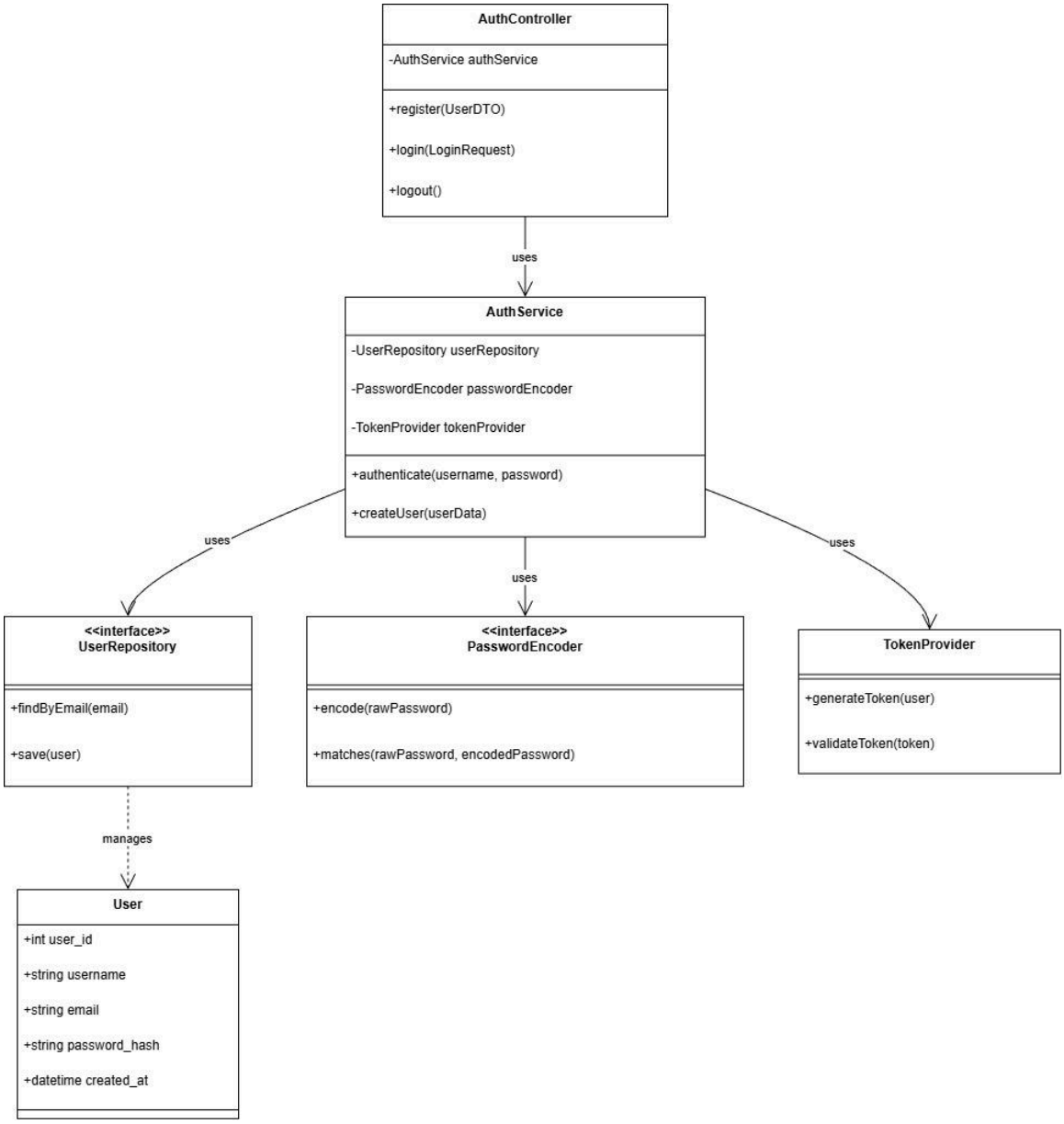
- **Use Case Diagram**



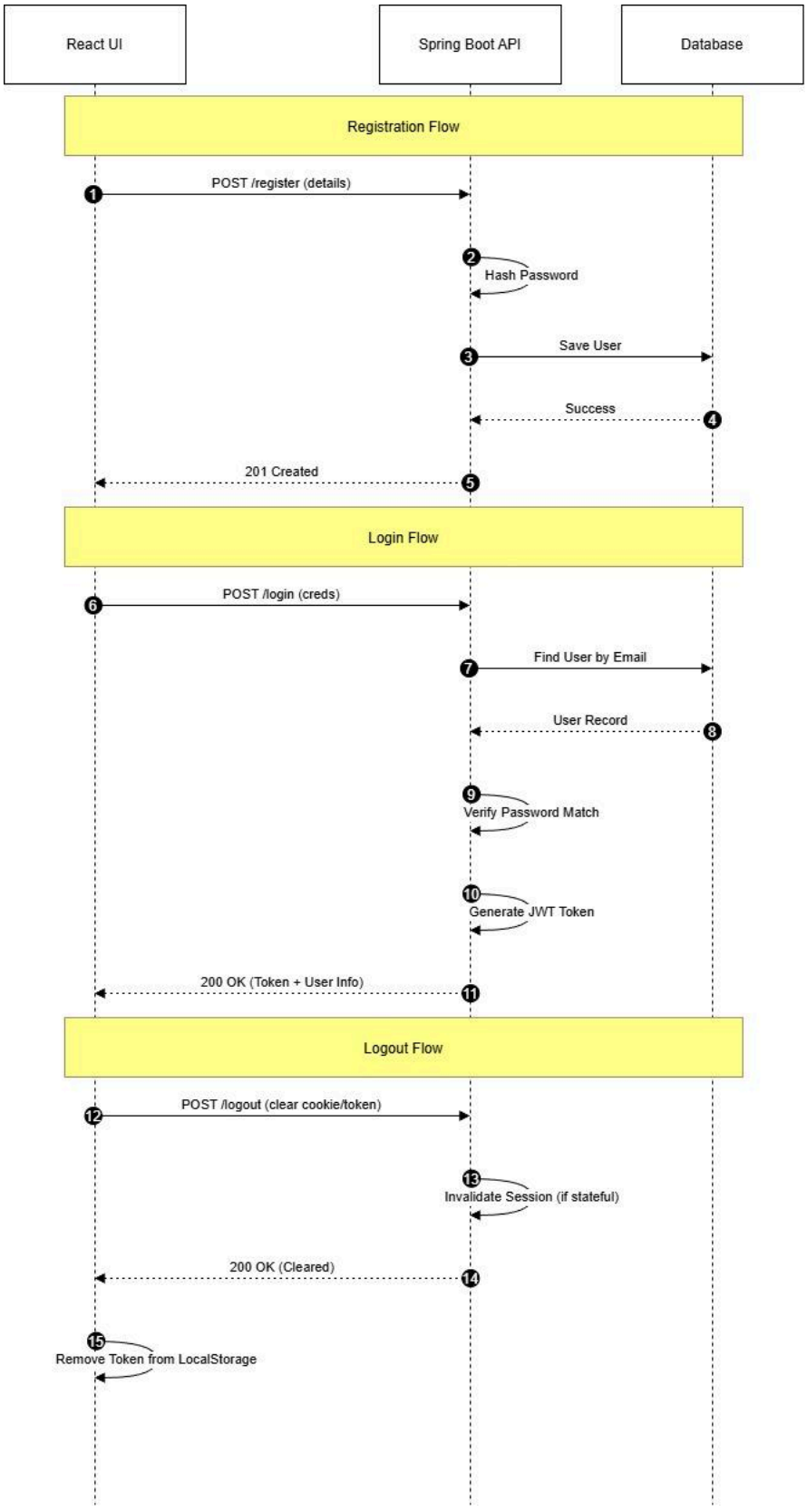
- ## Activity Diagram



- **Class Diagram**



- ## Sequence Diagram



- **Appendices**
  Include any additional information, references, or support materials.