CEBU INSTITUTE OF TECHNOLOGY
UNIVERSITY

IT342-G4
SYSTEMS INTEGRATION AND ARCHITECTURE 1

FUNCTIONAL REQUIREMENTS SPECIFICATION (FRS)

Project Title: User Registration and Authentication

Prepared By: Ramirez, Ruther Gerard L.

Date of Submission: 02/03/2026

Version: 1.0

Table of Contents

1. Introduction

1.1. Purpose

The purpose of this document is to define the software architecture and design for the **User Authentication Module**. It details the system flow for user registration, login, profile viewing, and logout. This document serves as the primary reference for the implementation phase using React (Frontend) and Spring Boot (Backend).

1.2. Scope

The system is a web-based application authentication layer.

- **In Scope:** User account creation, secure login using credentials, accessing a protected dashboard/profile, and terminating the session via logout.
- **Out of Scope:** Password recovery (forgot password), multi-factor authentication, and role-based access control (RBAC) beyond simple user roles.

1.3. Definitions, Acronyms, and Abbreviations

- **SRS:** Software Requirements Specification
- **API:** Application Programming Interface
- **JWT:** JSON Web Token (implied for stateless authentication)
- **UI:** User Interface (React)
- **DTO:** Data Transfer Object

2. Overall Description

2.1. System Perspective

This module functions as the gateway for the larger application. It acts as the security barrier, ensuring that only authenticated identities can access the protected resources (Profile/Dashboard) of the system.

2.2. User Classes and Characteristics

- **Guest User:** An unregistered or unauthenticated visitor who can only access the Landing, Login, and Registration pages.
- **Authenticated User:** A registered user who has successfully logged in. They have access to the Dashboard and Profile pages.

2.3. Operating Environment

- **Client:** Modern Web Browser (Chrome, Firefox, Edge).
- **Frontend:** React.js.
- **Backend:** Java Spring Boot.
- **Database:** SQL Database (e.g., MySQL or H2).
- **Diagramming:** draw.io / diagrams.net.

2.4. Assumptions and Dependencies

- The database is operational and reachable by the Spring Boot API.
- Network connectivity exists between the React Client and the Spring Boot API.

3. System Features and Functional Requirements

Describe each major feature of the system and its functional requirements.

3.1. Feature 1: User Registration

Description: Allows a guest to create a new account by providing necessary credentials.
Functional Requirements:
- The system shall validate that the email is unique.
- The system shall encrypt the user's password before saving it to the database.
- The system shall provide feedback upon successful or failed registration.

3.2. Feature 2: User Login

Description: Allows a registered user to authenticate and gain access to the system.
Functional Requirements:
- The system shall verify the username/email and password against the database.
- The system shall issue an authentication token (or session) upon successful login.
- The system must prevent access to protected pages (Dashboard) after logout.

3.3 Feature 3: Logout

Description: Allows an authenticated user to terminate their session securely.
Functional Requirements:
- The system shall invalidate the user's session or remove the stored authentication token from the client.
- The system shall redirect the user to the Login page after logout.
- The system must prevent access to protected pages (Dashboard) after logout.

4. Non-Functional Requirements

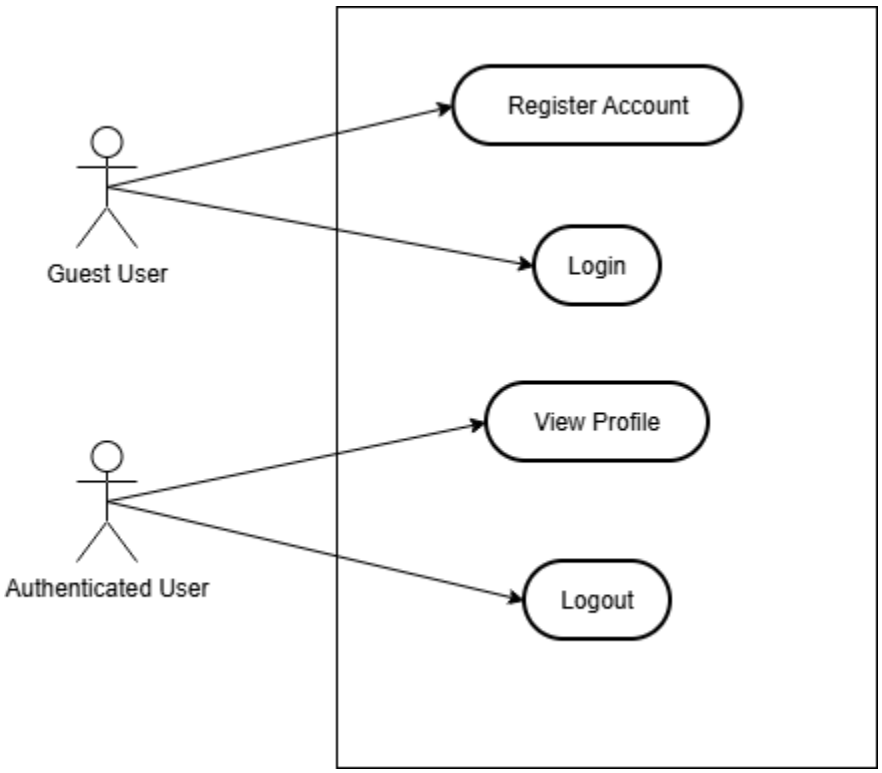
- **Security:** Passwords must never be stored in plain text. APIs must be protected against unauthorized access.
- **Usability:** Forms must provide clear validation error messages.
- **Reliability:** The authentication service should be available 99.9% of the time.

5. System Models (Diagrams)

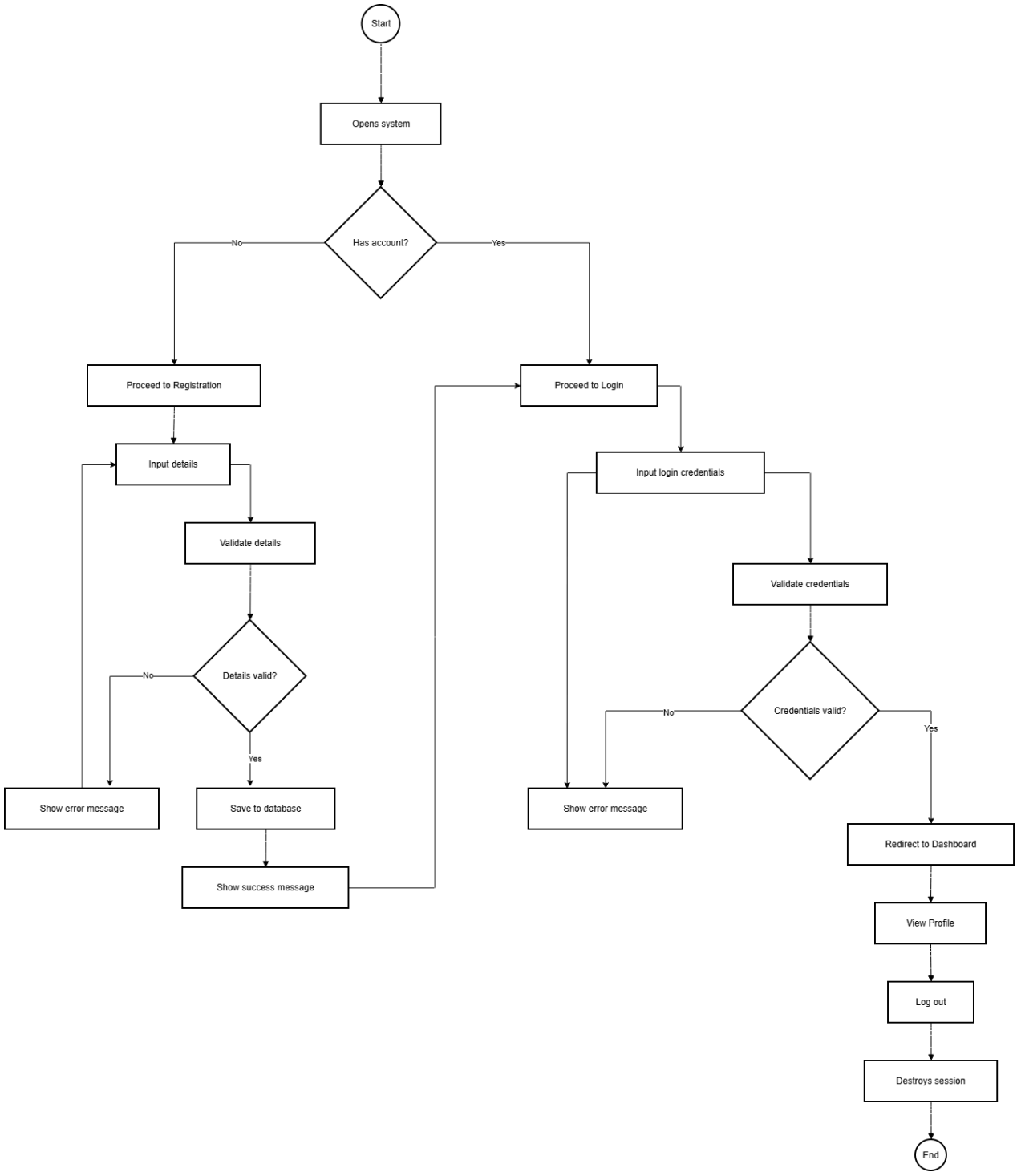
5.1. ERD

Users	
PK	**user_id**
	first_name
	last_name
	email
	password
	created_at
	updated_at
	status

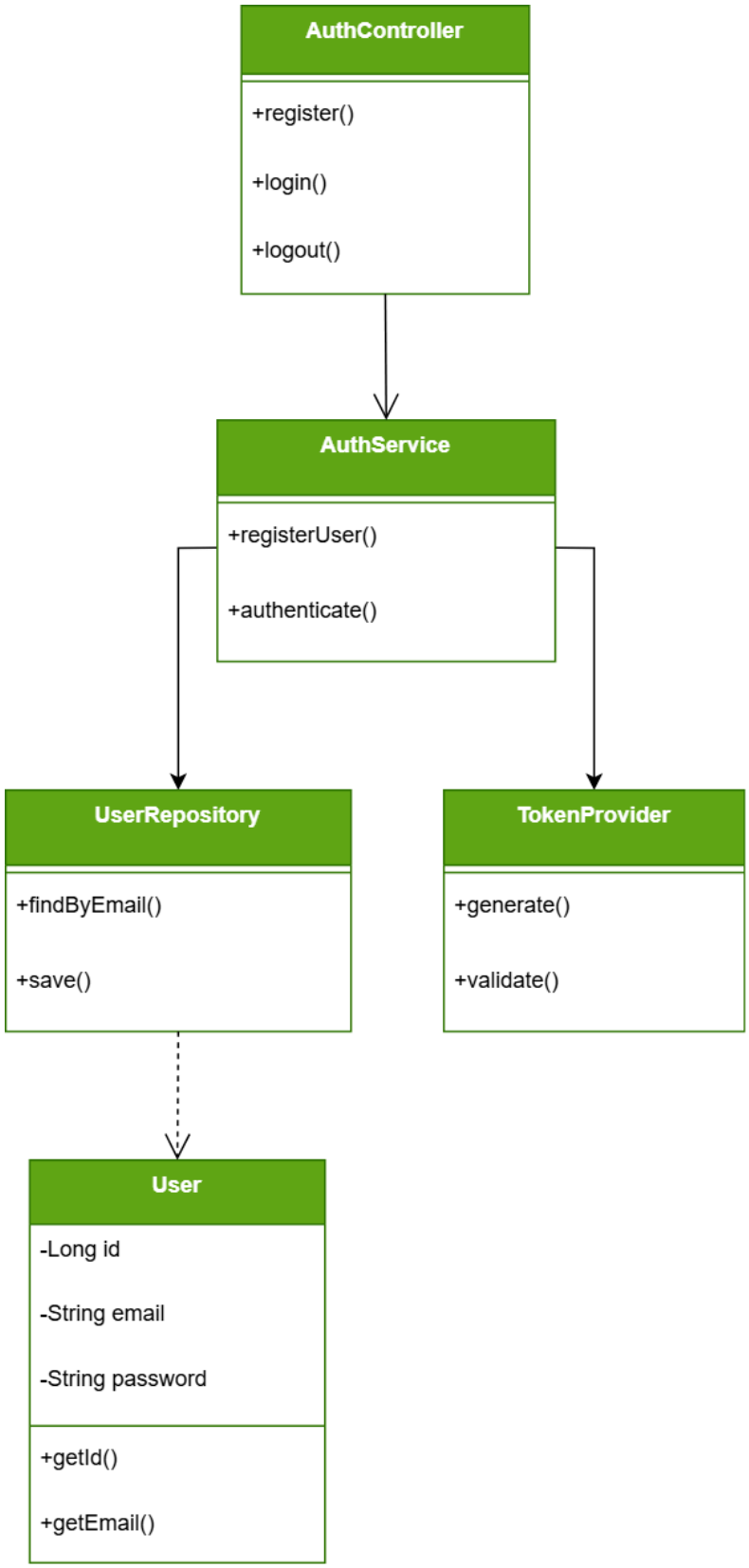
5.2. Use Case Diagram



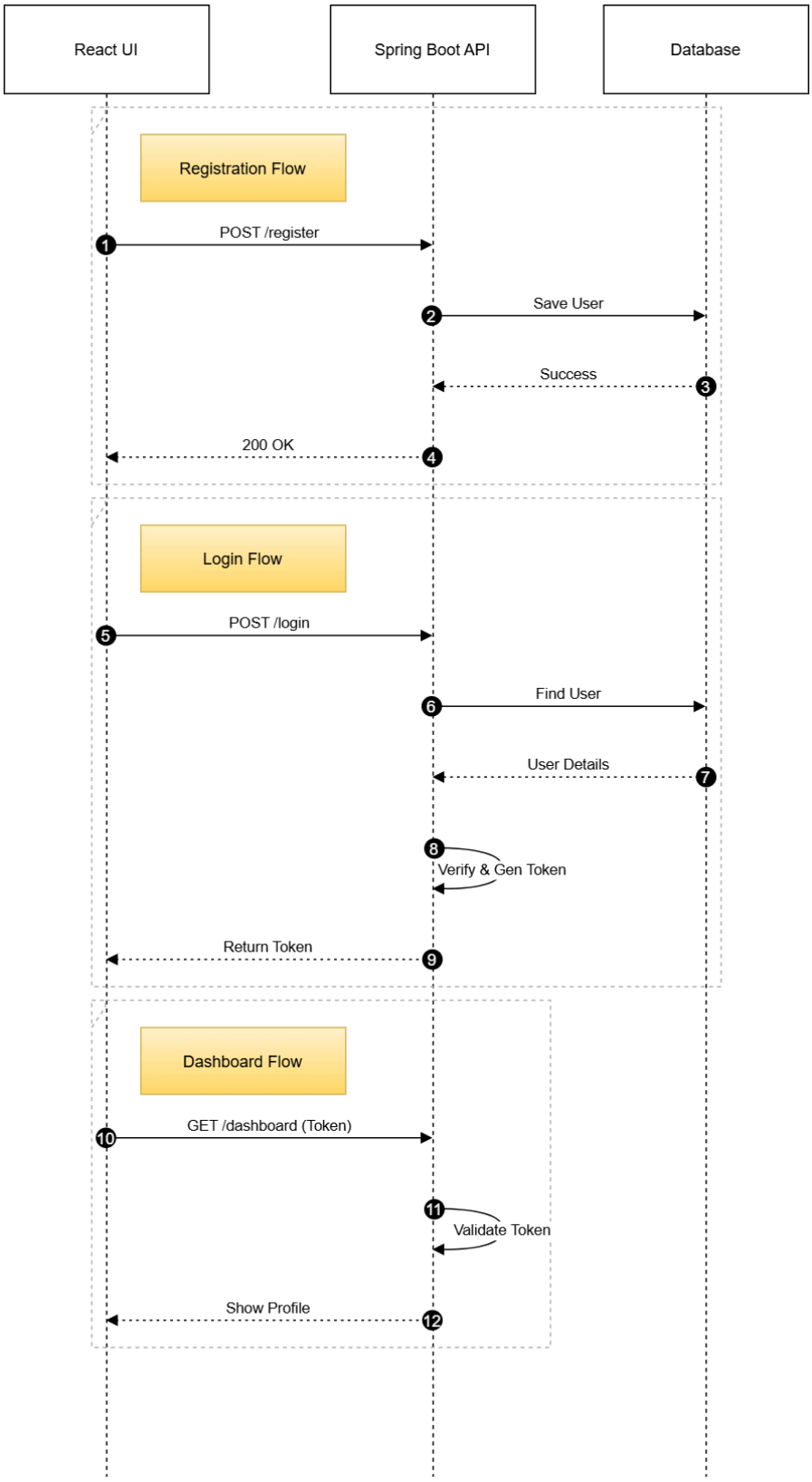
5.3. Activity Diagram



5.4. Class Diagram



5.5. Sequence Diagram



6. Appendices

References

[1] Spring.io, "Securing a Web Application," Spring Guides, 2024.
https://spring.io/guides/gs/securing-web/

[2] React.dev, "React: Interactivity and State," React Documentation, 2024.
https://react.dev/learn/adding-interactivity

Support Material:

- Diagramming Tool: Diagrams were created using Draw.io (diagrams.net)