



CEBU INSTITUTE OF TECHNOLOGY
UNIVERSITY

IT342-G2

SYSTEMS INTEGRATION AND

ARCHITECTURE 1

FUNCTIONAL REQUIREMENTS SPECIFICATION (FRS)

Project Title: Mini App – User Registration & Authentication

Prepared By: Bayonas, Jay Lord C.

Date of Submission: February 06, 2026

Version: 1

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.....	4
2.4.	Assumptions and Dependencies	4
3.	System Features and Functional Requirements	4
3.1.	Feature 1: User Registration.....	4
3.2.	Feature 2: User Login.....	4
3.3	Feature 3: View Profile / Dashboard.....	4
4.	Non-Functional Requirements	4
5.	System Models (Diagrams)	5
5.1.	ERD	5
5.2.	Use Case Diagram	5
5.3.	Activity Diagram	6
5.4.	Class Diagram.....	7
5.5.	Sequence Diagram	8
6.	Appendices	9

1. Introduction

1.1. Purpose

The purpose of this system is to develop a web-based user authentication system that allows users to register, log in, access their profile, and log out securely.

1.2. Scope

The system allows users to:

- Register an account
- Log in using valid credentials
- View a protected profile/dashboard when authenticated
- Log out of the system

1.3. Definitions, Acronyms, and Abbreviations

- User – A person who registers and uses the system
- Authentication – Process of verifying user identity
- Authorization – Process of allowing access to protected pages
- API – Application Programming Interface
- UI – User Interface
- ERD – Entity Relationship Diagram

2. Overall Description

2.1. System Perspective

The system follows a client-server architecture:

- Frontend: React UI for user interaction
- Backend: Spring Boot REST API for authentication logic
- Database: Stores user credentials and profile information

2.2. User Classes and Characteristics

Guest User

- Not logged in
- Can register and log in

Authenticated User

- Logged in
- Can view profile/dashboard
- Can log out

2.3. Operating Environment

- Frontend: React (Web Browser)
- Backend: Spring Boot (Java)
- Database: MySQL
- Diagram Tool: draw.io / diagrams.net

2.4. Assumptions and Dependencies

- Users have internet access
- Backend API is available and running
- Database is properly configured

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 user to create a new account by providing required details.

Functional Requirements:

- The system shall allow users to enter registration details
- The system shall validate user input
- The system shall store encrypted passwords in the database

3.2. Feature 2: User Login

Description: Allows registered users to log in using valid credentials.

Functional Requirements:

- The system shall authenticate user credentials
- The system shall generate a token/session upon successful login
- The system shall deny access if credentials are invalid

3.3 Feature 3: View Profile / Dashboard

Description: Allows authenticated users to access protected pages.

Functional Requirements:

- The system shall restrict access to logged-in users only
- The system shall display user information
- The system shall redirect unauthenticated users to the login page

4. Non-Functional Requirements

- Security: Passwords must be encrypted; protected routes must require authentication
- Performance: Login and registration should respond within acceptable time
- Usability: Simple and clear UI for users
- Reliability: System should consistently handle authentication requests
- Maintainability: Code and diagrams should support easy future updates

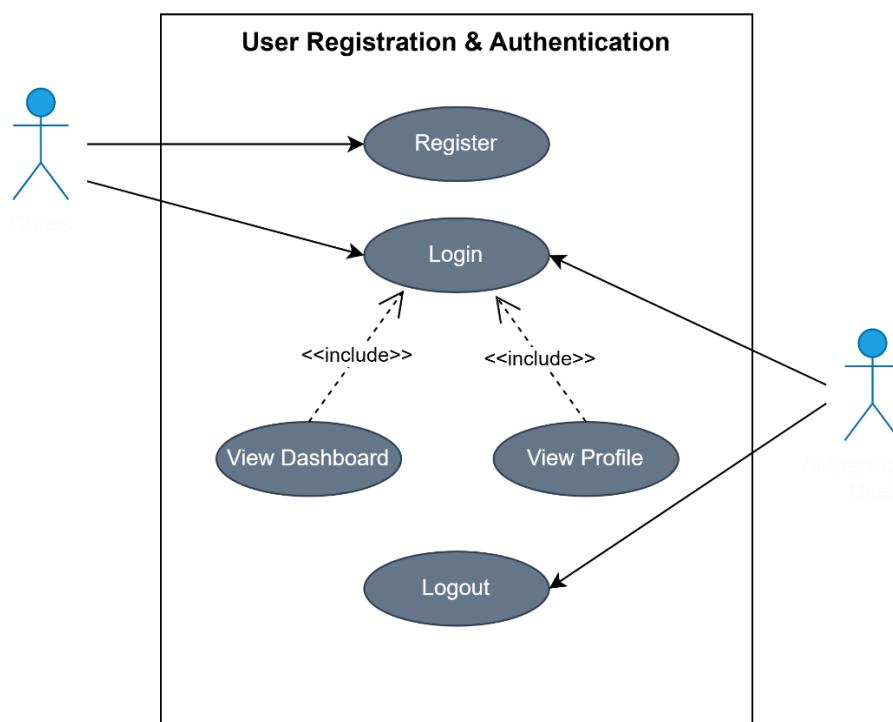
5. System Models (Diagrams)

Insert the necessary diagrams for the system:

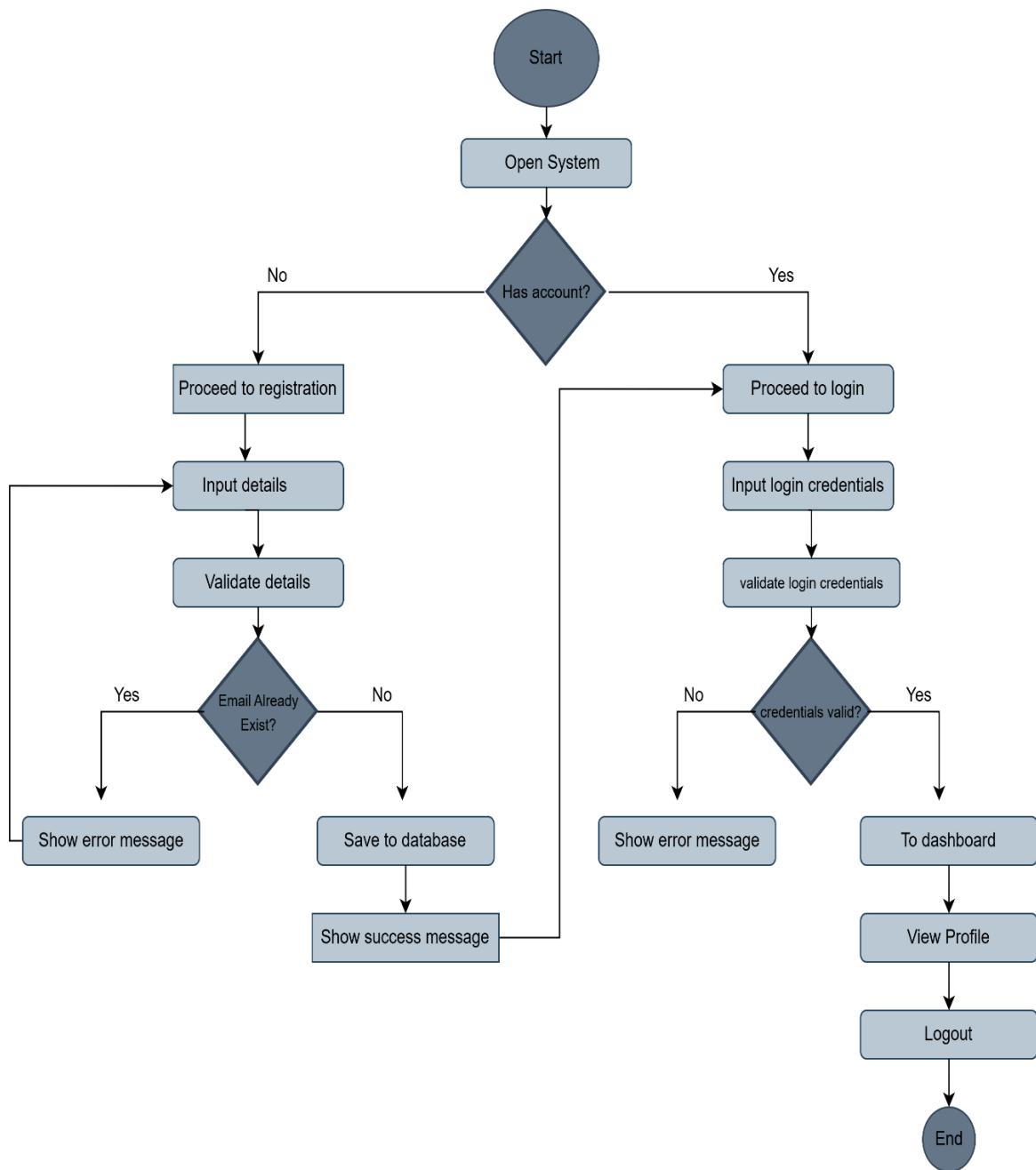
5.1. ERD

User		
PK	<u>UserId</u>	BIGINT
	username	VARCHAR(50)
	email	VARCHAR(100)
	password	VARCHAR(255)
	role	VARCHAR(20)
	created_at	TIMESTAMP
	updated_at	TIMESTAMP

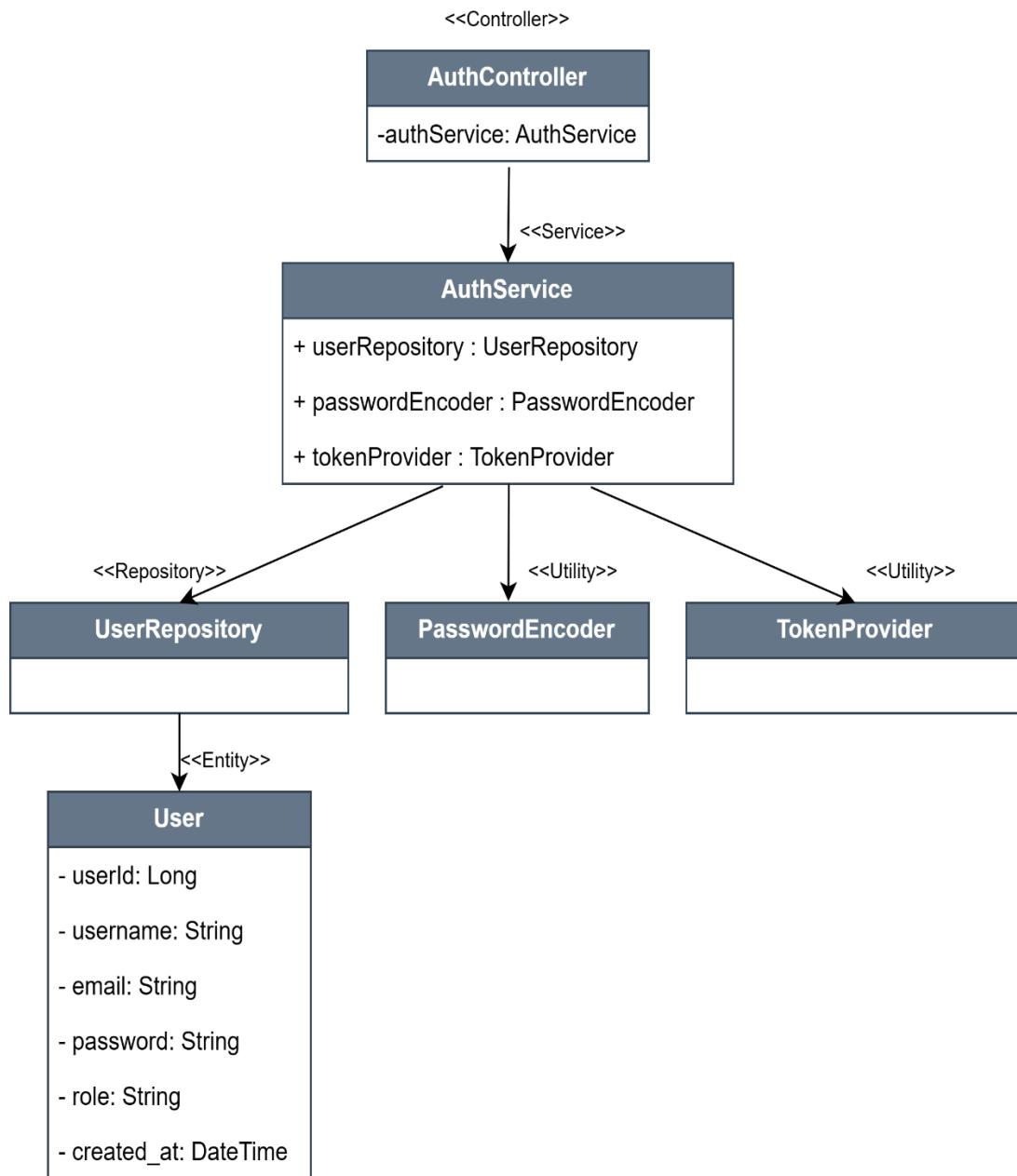
5.2. Use Case Diagram



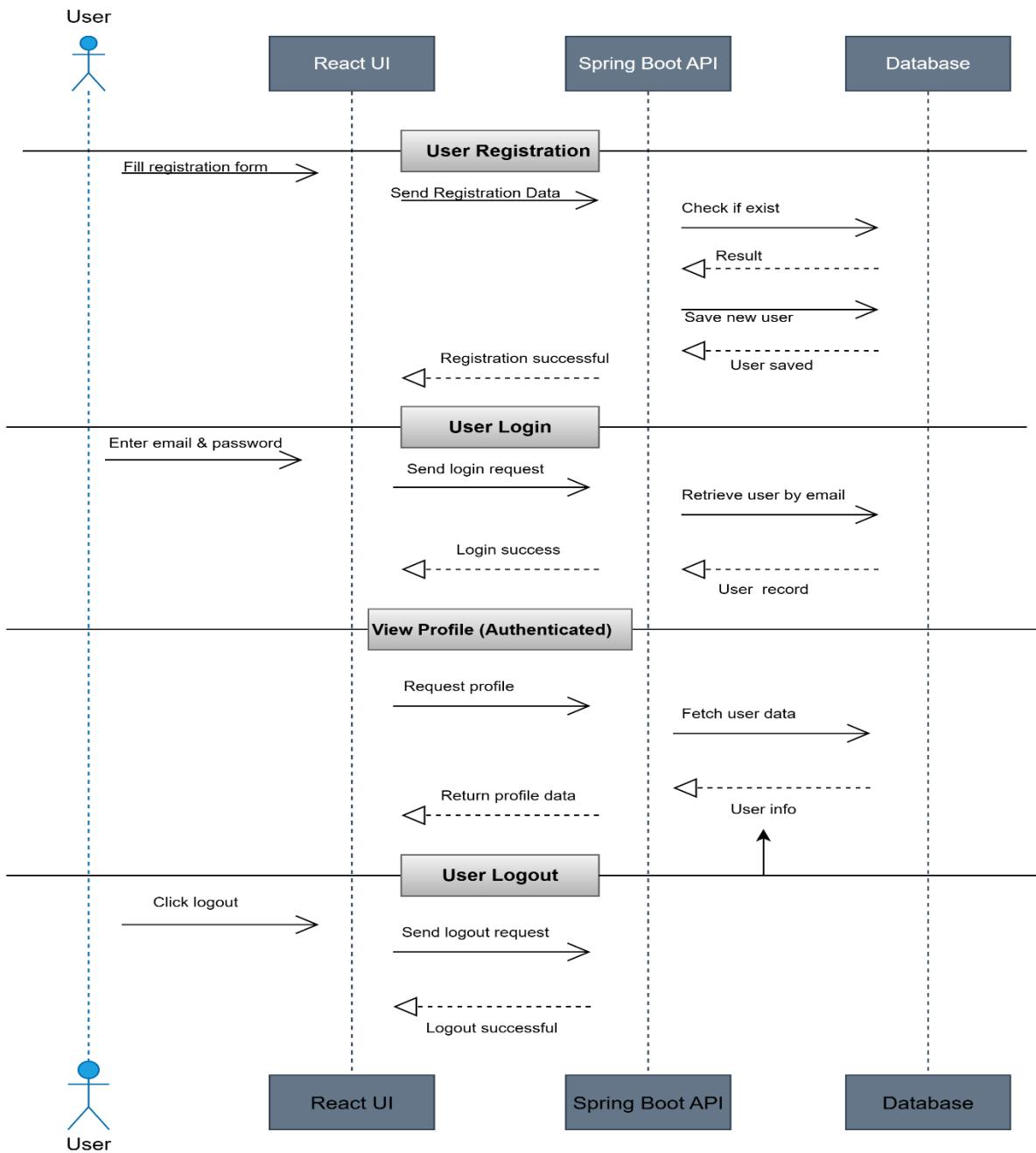
5.3. Activity Diagram



5.4. Class Diagram



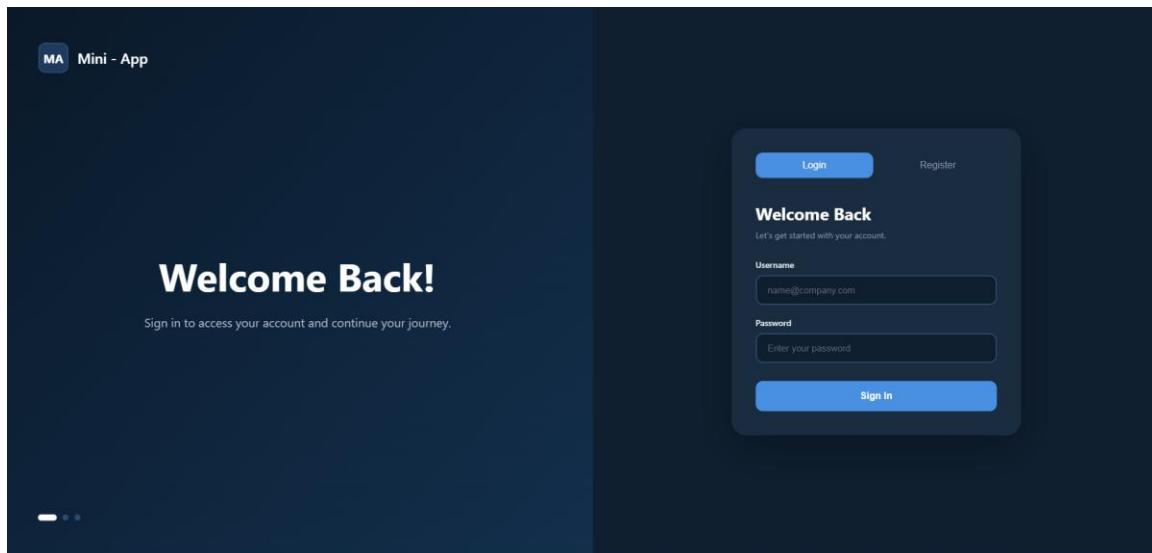
5.5. Sequence Diagram



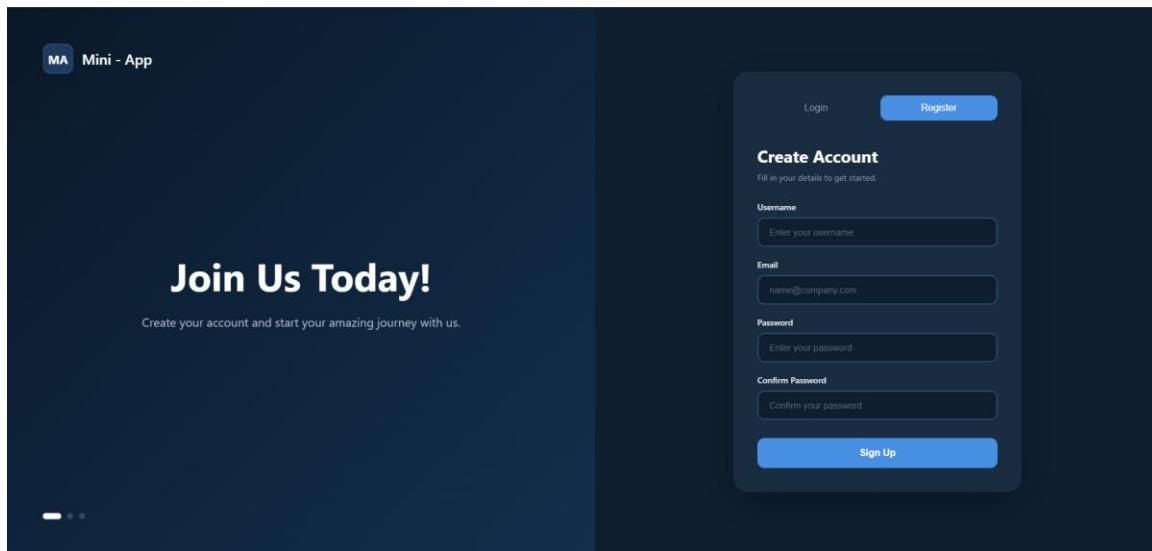
6. Appendices

This appendix includes the tools, technologies, assumptions, and references used to support the User Registration and Authentication system documentation. The system utilizes draw.io/diagrams.net for diagram creation, React for the frontend interface, Spring Boot for the backend API, and a relational database such as MySQL or PostgreSQL for data storage.

Login



Register



Dashboard

The screenshot shows the 'Dashboard' page of a user portal. At the top, there's a header with the text 'OVERVIEW' and the title 'Dashboard'. Below the header, a welcome message says 'Welcome back, king.' A section titled 'Account Status' displays the following information:

Email	king@gmail.com
Username	king
Member Since	2/10/2026

At the bottom left of the dashboard area, there's a 'Logout' button.

View Profile

The screenshot shows the 'Profile Information' page of a user profile. At the top, there's a header with the text 'USER PROFILE' and the title 'Profile Information'. In the center, there's a circular placeholder for a profile picture with the letter 'K' inside. Below the placeholder, the username 'king' and email 'king@gmail.com' are displayed. A summary bar at the bottom provides quick access to account details:

User ID	2
Username	king
Email	king@gmail.com
Account Created	2/10/2026, 12:33:32 AM

At the bottom left of the profile area, there's a 'Logout' button.