CEBU INSTITUTE OF TECHNOLOGY
UNIVERSITY

IT342-G2 SYSTEMS INTEGRATION AND ARCHITECTURE 1

FUNCTIONAL REQUIREMENTS SPECIFICATION (FRS)

Project Title: HealthGate

Prepared By: Vein Carmell G. Pangilinan

Date of Submission: 02/16/2026

Version: 4.0

Table of Contents

1. Introduction

1.1. Purpose

This document defines the functional and non-functional requirements of the HealthGate system. It is intended for students, instructors, developers, and system designers who will design, implement, and evaluate the user registration and authentication system.

1.2. Scope

HealthGate is a secure authentication system that allows users to register, log in, view a protected dashboard, and log out. It ensures that only authenticated users can access protected pages. The system focuses on authentication and user management only.

1.3. Definitions, Acronyms, and Abbreviations

List and define important terms used in this document.

2. Overall Description

2.1. System Perspective

HealthGate is a standalone authentication module that can be integrated into larger systems such as healthcare apps, clinic systems, or service platforms. It communicates between a frontend client and backend server through secure APIs.

2.2. User Classes and Characteristics

a. **Guest User** – Unregistered or logged-out user; can register and log in.
b. **Authenticated User** – Logged-in user; can access dashboard and profile.

2.3. Operating Environment

Frontend: React Web Application

Backend: Spring Boot REST API

Database: MySQL / PostgreSQL

Tools: VS Code, Postman, Git, Draw.io (Diagrams.net)

2.4. Assumptions and Dependencies

- Users have internet access.
- The system depends on database availability.
- Secure token authentication (JWT) is implemented.
- Backend services must be running for system operation.

3. System Features and Functional Requirements

3.1. Feature 1: User Registration

Description: Allows new users to create an account.
Functional Requirements:

- The system shall allow users to register using email and password.
- The system shall validate input fields.
- The system shall encrypt passwords before storage.
- The system shall prevent duplicate email registration.

3.2. Feature 2: User Login

Description: Allows registered users to log in.
Functional Requirements:

- The system shall authenticate users using credentials.
- The system shall generate authentication tokens.
- The system shall reject invalid login attempts.
- The system shall create user sessions.

3.3. Feature 3: User Dashboard/Profile

Description: Allows logged-in users to access protected content.
Functional Requirements:

- The system shall restrict access to authenticated users only.
- The system shall display user profile information.
- The system shall verify authentication tokens.

3.4. Feature 4: User Logout

Description: Allows users to securely log out.
Functional Requirements:

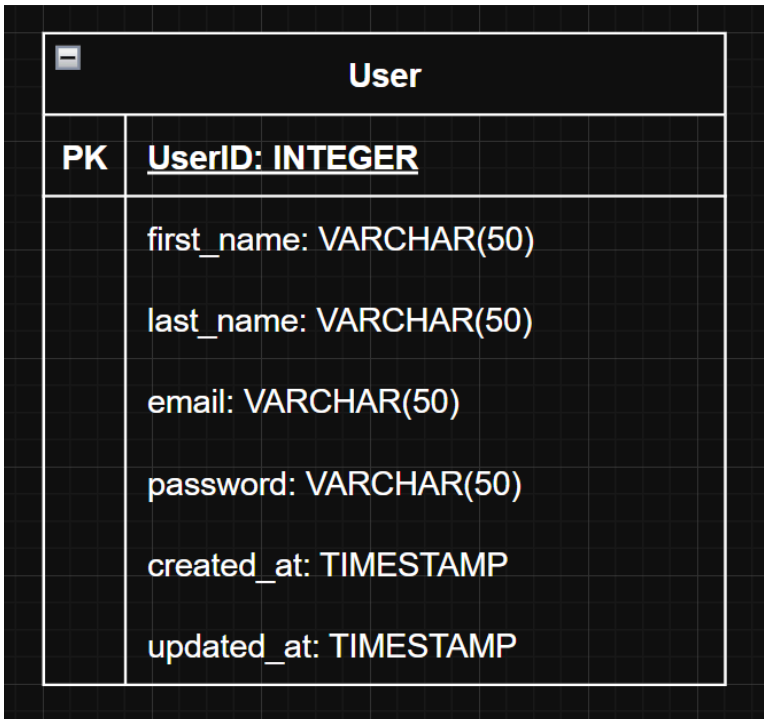
- The system shall invalidate authentication tokens.
- The system shall destroy active sessions.
- The system shall redirect users to login page.

4. Non-Functional Requirements

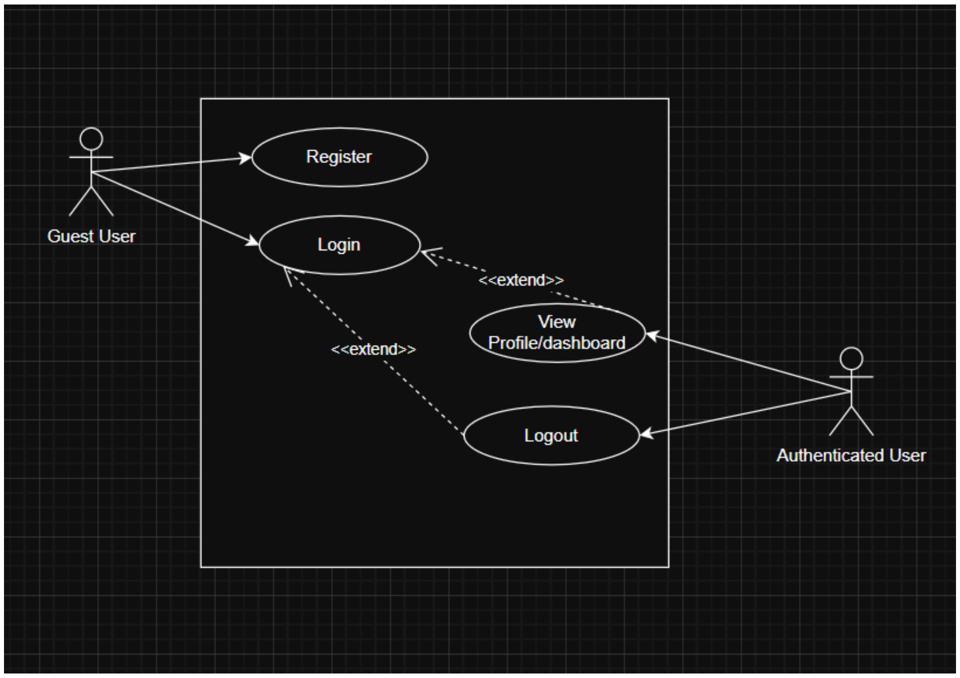
- **Security:** Password hashing, token-based authentication, secure APIs.
- **Performance:** Login and registration responses under 2 seconds.
- **Usability:** Simple UI and clear error messages.
- **Reliability:** System uptime and data consistency.
- **Scalability:** Supports future expansion and more users.

5. System Models (Diagrams)

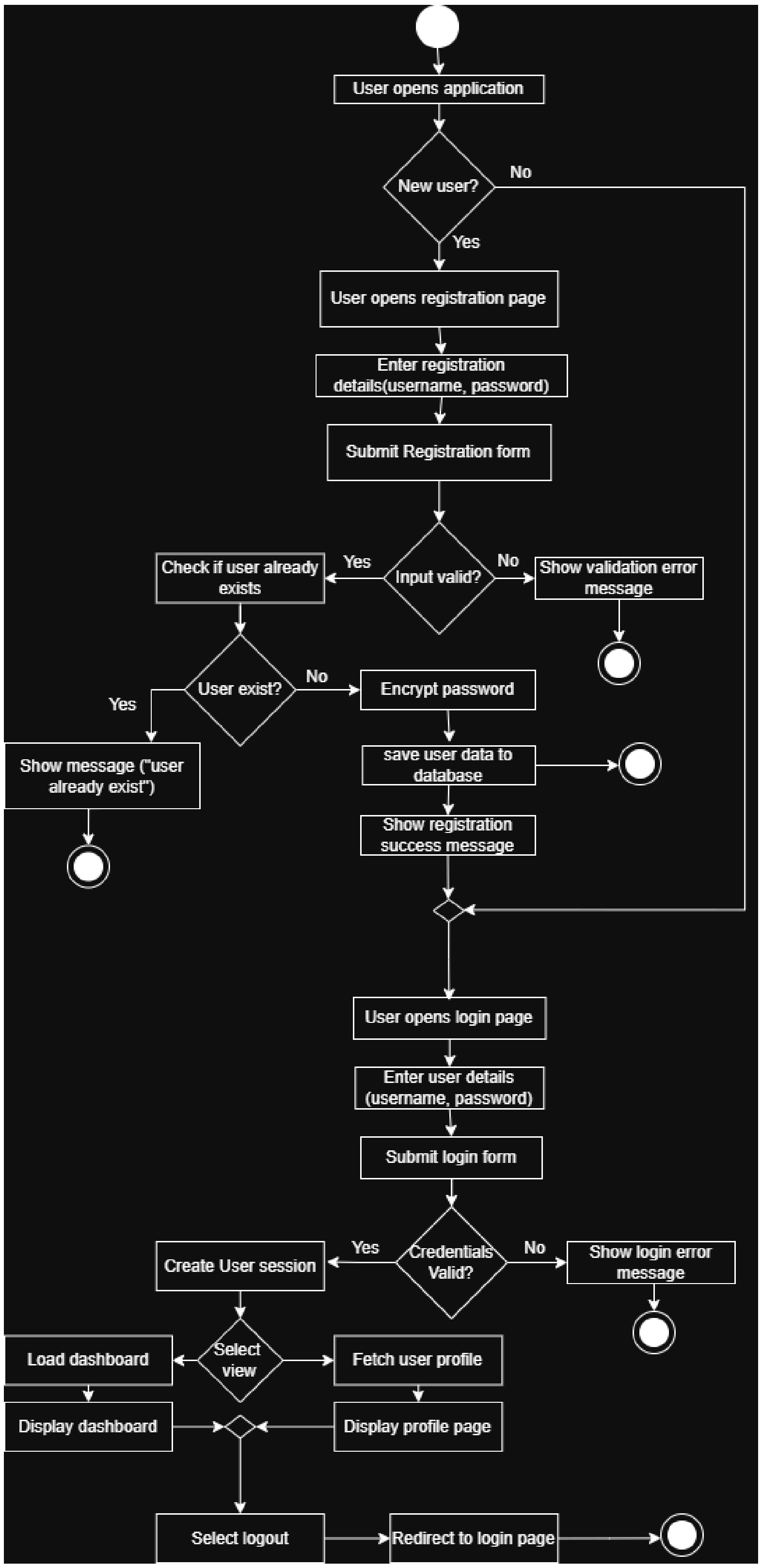
5.1. ERD



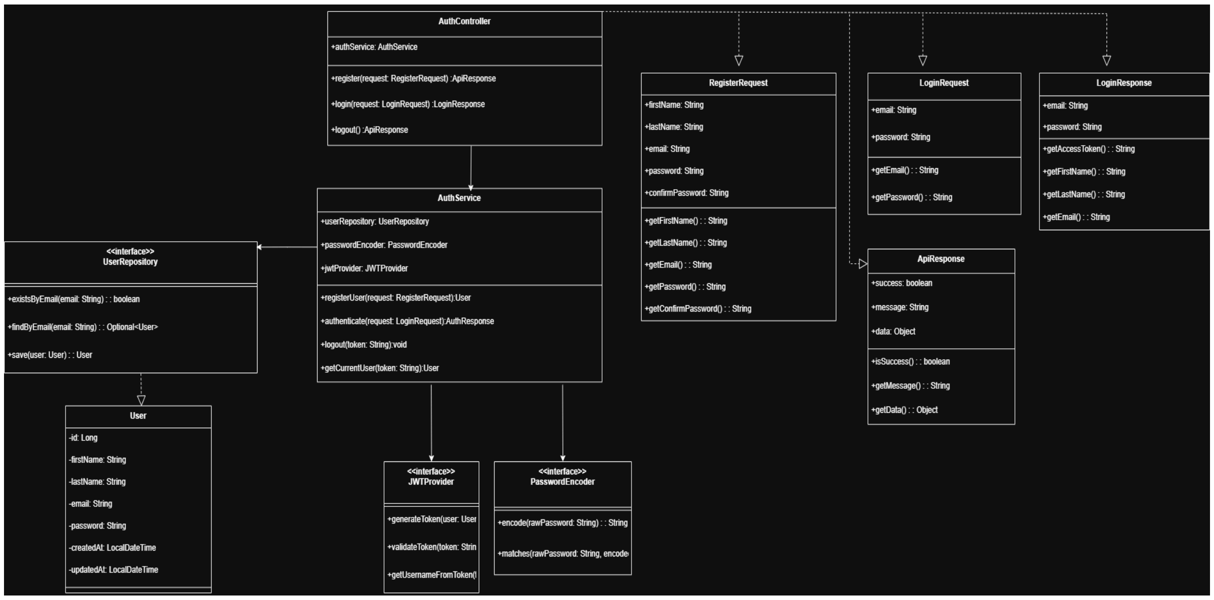
5.2. Use Case Diagram



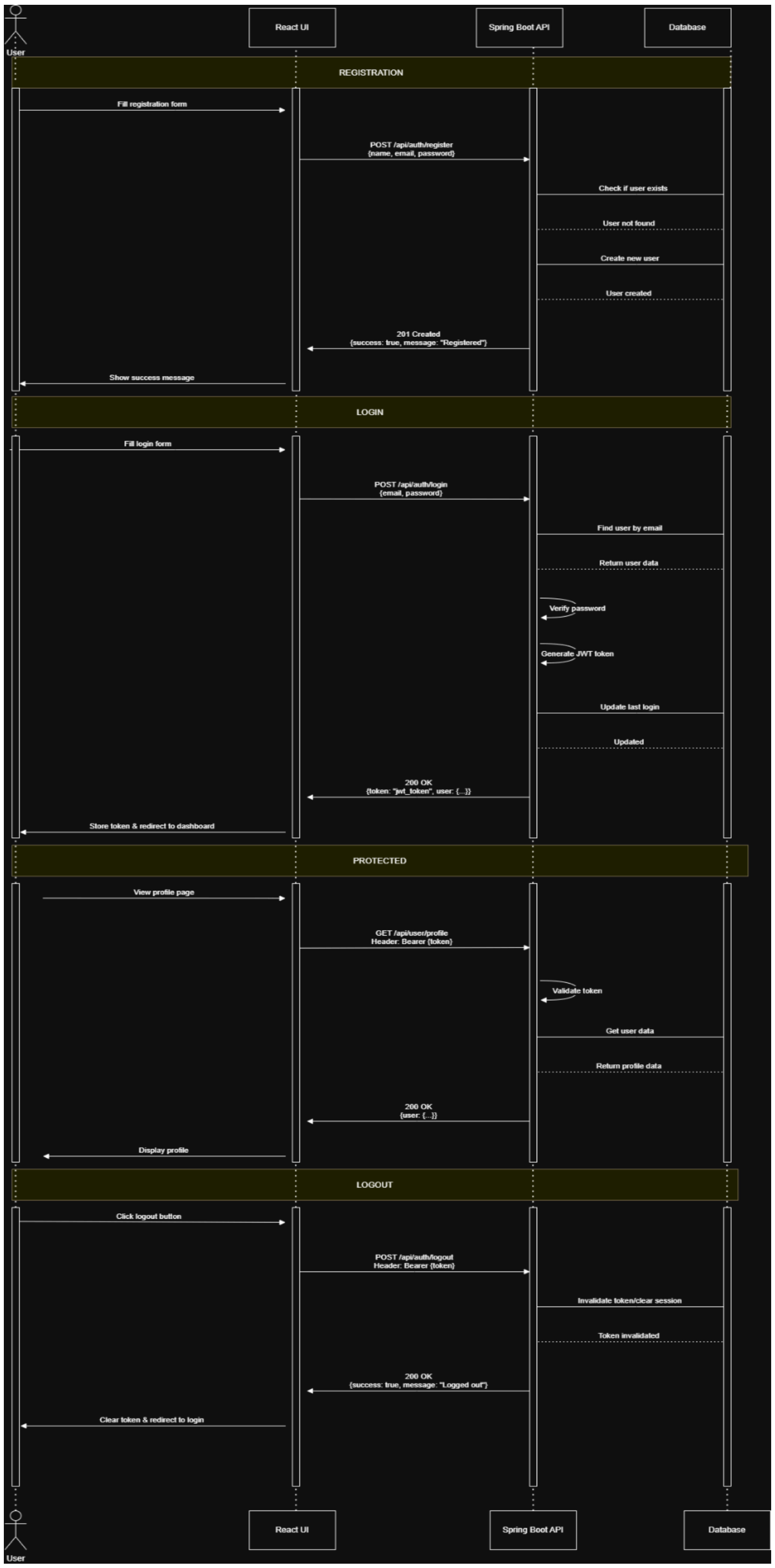
5.3. Activity Diagram



5.4. Class Diagram

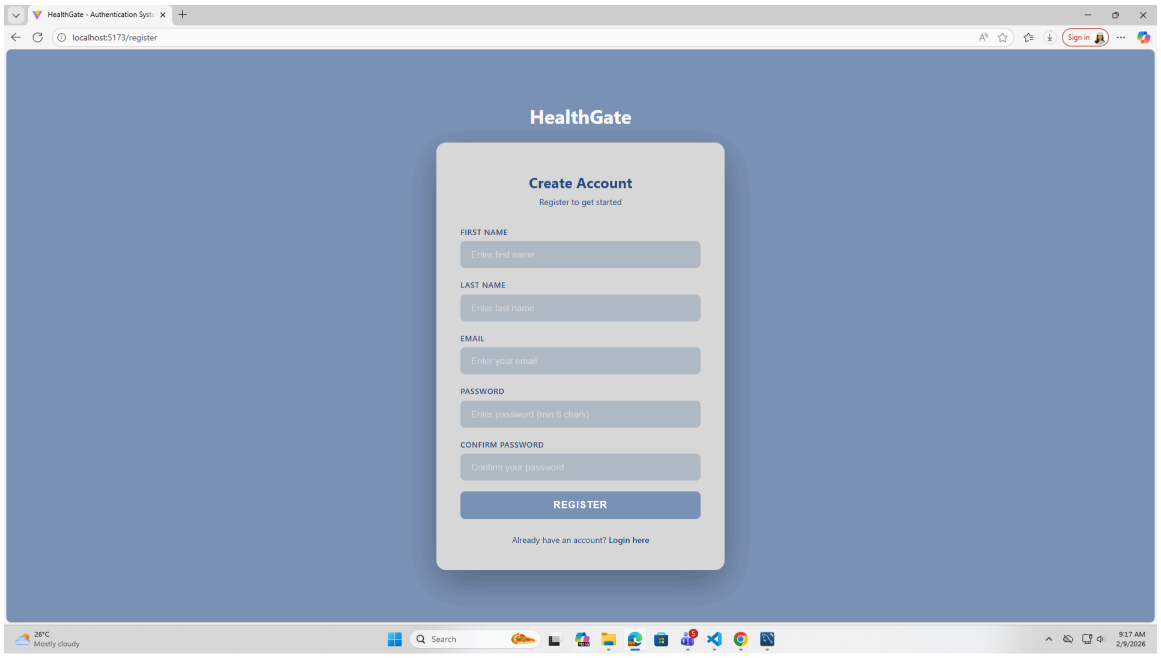

AuthController
+authService: AuthService
+register(request: RegisterRequest) :ApiResponse
+login(request: LoginRequest) :LoginResponse
+logout() :ApiResponse
AuthService
+userRepository: UserRepository
+passwordEncoder: PasswordEncoder
+jwtProvider: JWTProvider
+registerUser(request: RegisterRequest):User
+authenticate(request: LoginRequest):AuthResponse
+logout(token: String):void
+getCurrentUser(token: String):User
<<interface>>
UserRepository
+existsByEmail(email: String) : : boolean
+findByEmail(email: String) : : Optional<User>
+save(user: User) : : User
User
-id: Long
-firstName: String
-lastName: String
-email: String
-password: String
-createdAt: LocalDateTime
-updatedAt: LocalDateTime
<<interface>>
JWTProvider
+generateToken(user: User
+validateToken(token: Strin
+getUsernameFromToken(t
<<interface>>
PasswordEncoder
+encode(rawPassword: String) : : String
+matches(rawPassword: String, encode
RegisterRequest
+firstName: String
+lastName: String
+email: String
+password: String
+confirmPassword: String
+getFirstName() : : String
+getLastName() : : String
+getEmail() : : String
+getPassword() : : String
+getConfirmPassword() : : String
LoginRequest
+email: String
+password: String
+getEmail() : : String
+getPassword() : : String
LoginResponse
+email: String
+password: String
+getAccessToken() : : String
+getFirstName() : : String
+getLastName() : : String
+getEmail() : : String
ApiResponse
+success: boolean
+message: String
+data: Object
+isSuccess() : : boolean
+getMessage() : : String
+getData() : : Object

5.5. Sequence Diagram



6. Screenshots of the Web UI:

6.1. Register



HealthGate

Create Account

Register to get started

FIRST NAME

Enter first name

LAST NAME

Enter last name

EMAIL

Enter your email

PASSWORD

Enter password (min 6 chars)

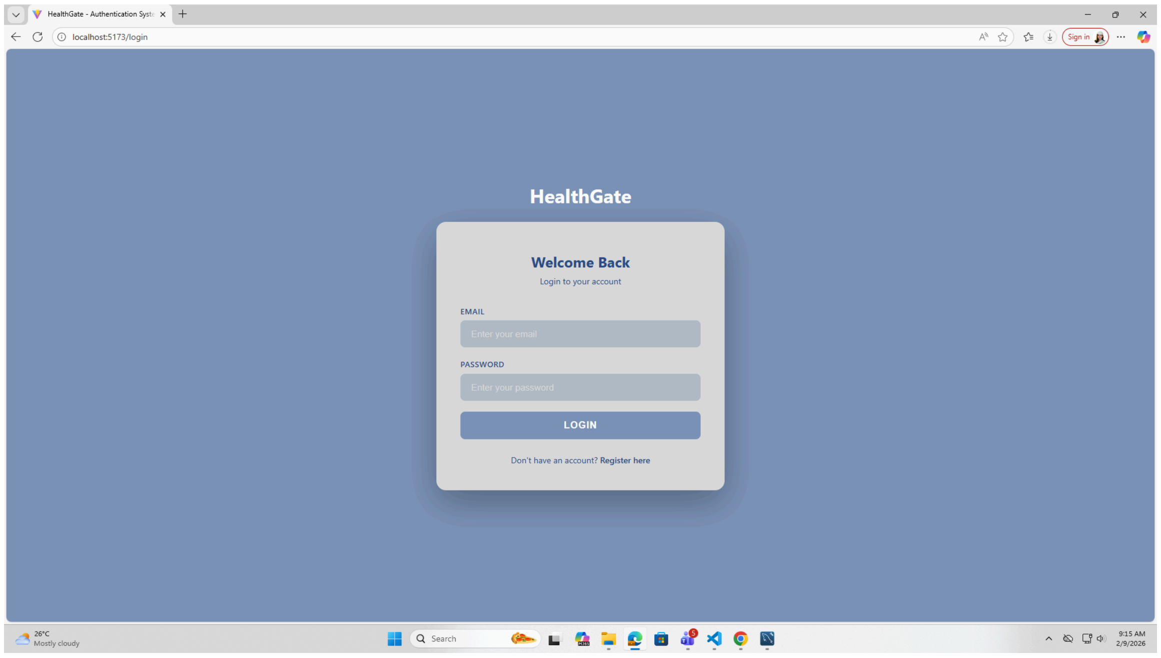
CONFIRM PASSWORD

Confirm your password

REGISTER

Already have an account? Login here

6.2. Login



HealthGate

Welcome Back

Login to your account

EMAIL

Enter your email

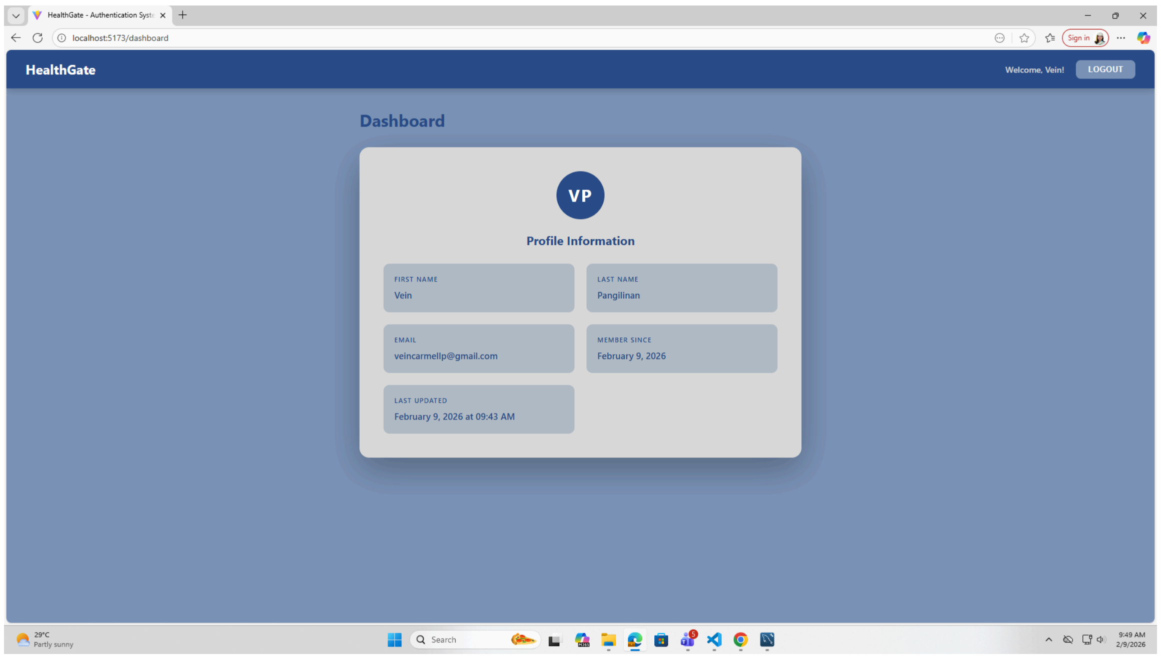
PASSWORD

Enter your password

LOGIN

Don't have an account? Register here

6.3 Dashboard/Profile



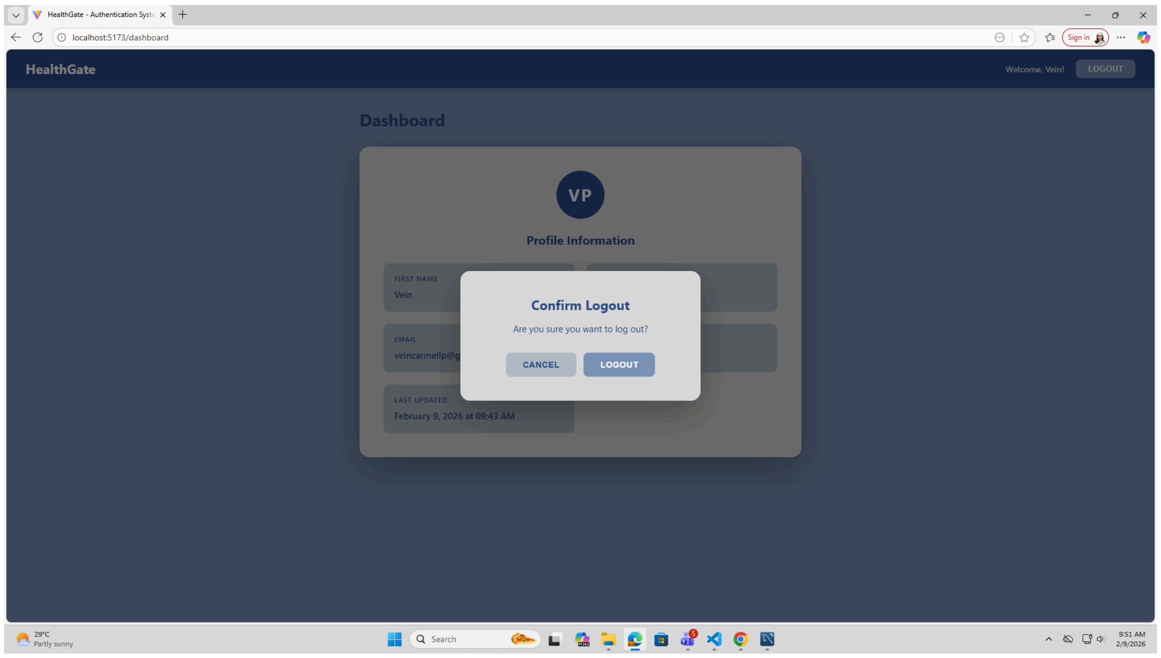
Dashboard

VP

Profile Information

FIRST NAME Vein	LAST NAME Pangilinan
EMAIL veincarmellp@gmail.com	MEMBER SINCE February 9, 2026
LAST UPDATED February 9, 2026 at 09:43 AM	

6.4 Logout



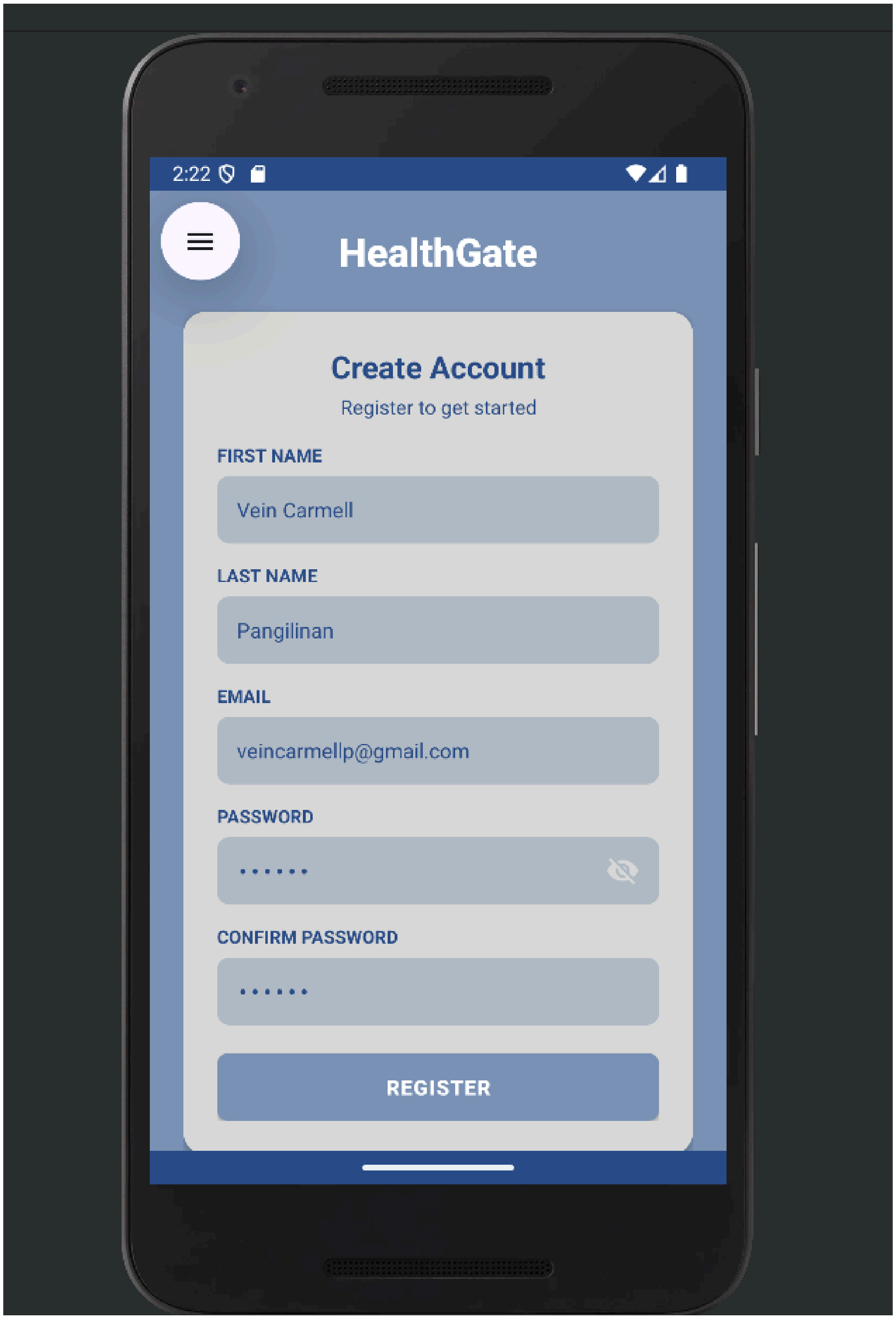
Confirm Logout

Are you sure you want to log out?

CANCEL LOGOUT

7. Screenshots of the Mobile UI:

7.1. Register



2:22

HealthGate

Create Account

Register to get started

FIRST NAME

Vein Carmell

LAST NAME

Pangilinan

EMAIL

veincarmellp@gmail.com

PASSWORD

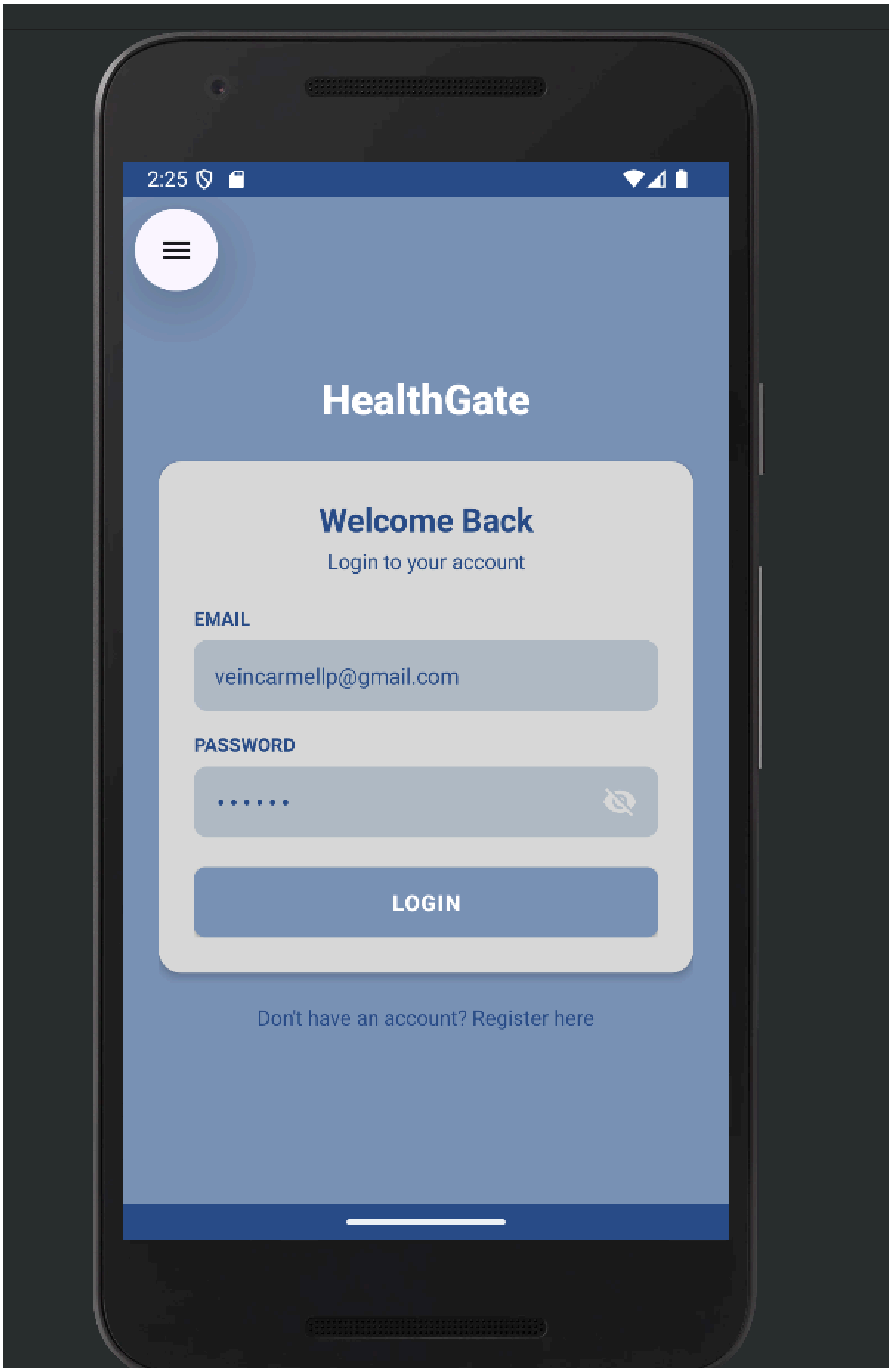
••••••

CONFIRM PASSWORD

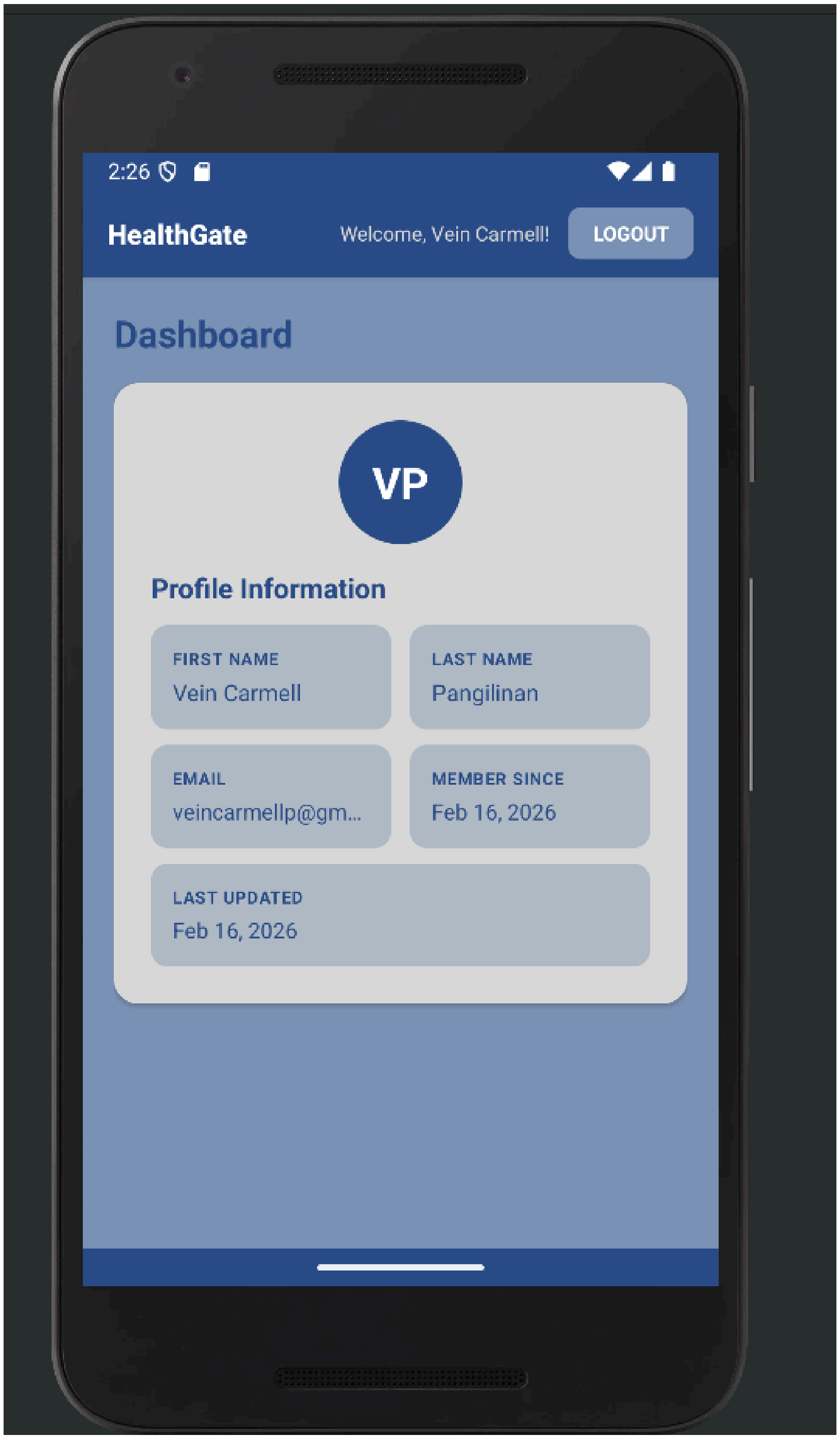
••••••

REGISTER

7.2. Login


2:25
HealthGate
Welcome Back
Login to your account
EMAIL
veincarmellp@gmail.com
PASSWORD
••••••
LOGIN
Don't have an account? Register here

7.3 Dashboard/Profile


2:26

HealthGate

Welcome, Vein Carmell!

LOGOUT

Dashboard

VP

Profile Information

FIRST NAME
Vein Carmell

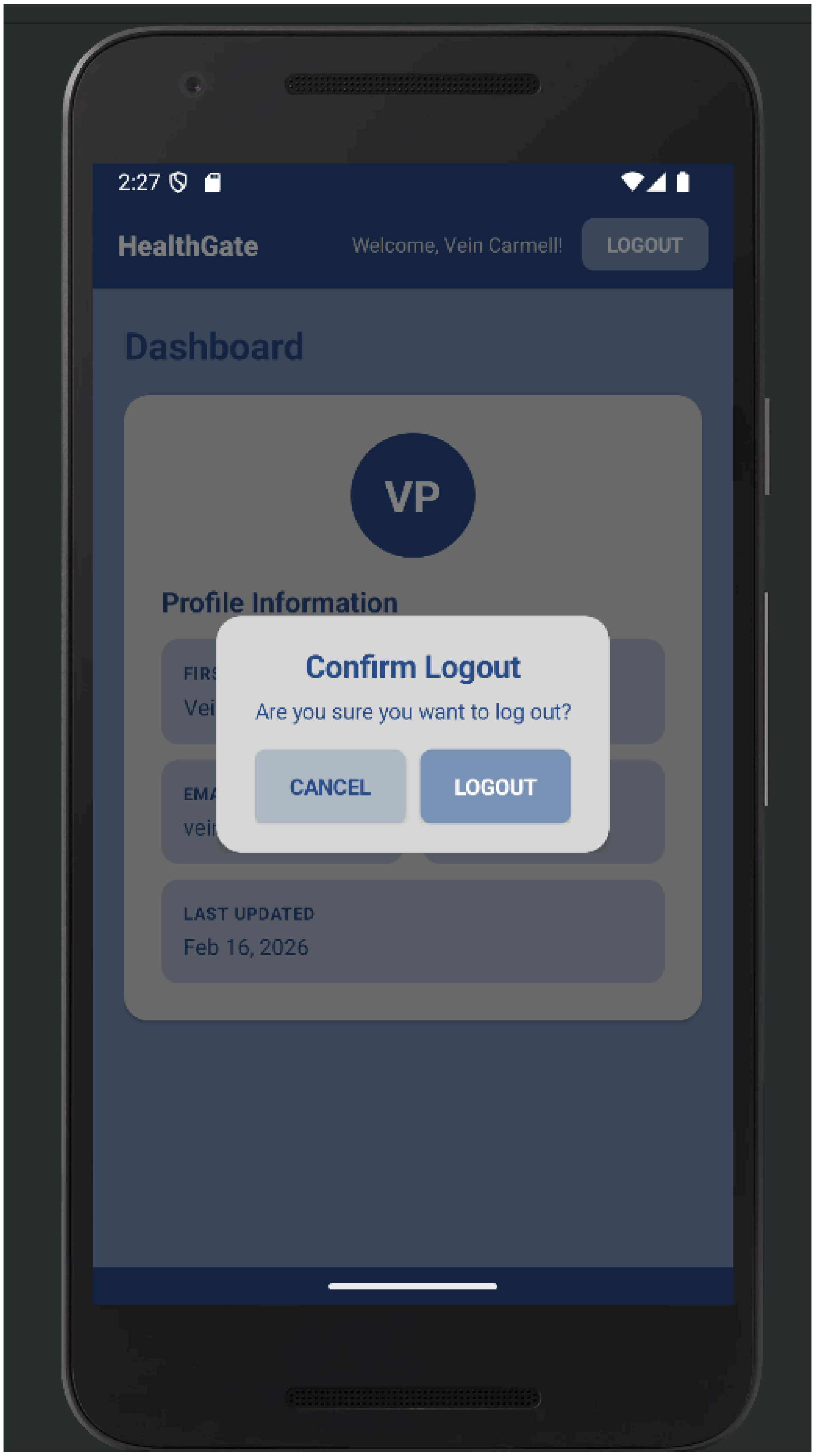
LAST NAME
Pangilinan

EMAIL
veincarmellp@gm...

MEMBER SINCE
Feb 16, 2026

LAST UPDATED
Feb 16, 2026

7.4 Logout



Confirm Logout

Are you sure you want to log out?

LAST UPDATED

Feb 16, 2026

8. Appendices

GeeksforGeeks. (2026, January 21). *Sequence diagrams Unified Modeling Language (UML)*. GeeksforGeeks. https://www.geeksforgeeks.org/system-design/unified-modeling-language-uml-sequence-diagrams/

GeeksforGeeks. (2026a, January 21). *Activity Diagrams Unified Modeling Language (UML)*. GeeksforGeeks. https://www.geeksforgeeks.org/system-design/unified-modeling-language-uml-activity-diagrams/