FORGOTTEN PASSWORD FEATURE

REALIZED BY : EDGAR BENITO MARTIN (ER1791) AND
ROSARIO PAVONE(ER1799)

Summarization

1.Introduction and component description

This project we will present and show all the documentation about the user registration and the login Modules. We are going to import all the software documentation about this module that we think is convenient.

The User Registration and Login Module provides secure registration and authentication functionalities to enable user access control within the application. It ensures that only authenticated users can access restricted content, maintain data integrity, and uphold security standards. This component will manage user accounts, passwords, and sessions.

2. Component Requirements

2.1 Functional Requirements:

1. **User Registration:**

The User Registration process will require users to sign up with a valid email address and a secure password that meets defined criteria, including a minimum length of nine characters and a mix of uppercase, lowercase, and numerical characters and also have to repeat the password for ensure there is no mistakes in the password.

2. **User Login:**

For User Login, users will access their accounts using their registered email and password and account lockouts triggered after multiple failed login attempts, providing robust protection against unauthorized access.

3. **Password Management:**

Will include options for password recovery and updates. Users can reset their password through an email-based link, helping them regain access if they forget their credentials. Additionally, password update functionality will be available in the user settings, allowing for easy, secure updates to maintain account security.

4. **Session Management:**

Session Management will ensure that user sessions are securely maintained, providing options to log out as needed. Automatic session timeouts will be enforced after a period of inactivity, balancing usability with security by preventing unauthorized access to unattended sessions.

2.2 Non-Functional Requirements

1. **Security:**

In terms of security, the system will implement password encryption using strong hashing algorithms, to ensure that stored user credentials are securely protected.

Additionally, measures will be in place to prevent common security vulnerabilities, including SQL injection, cross-site scripting (XSS), and cross-site request forgery (CSRF) attacks, creating a robust security foundation for user interactions.

2. **Performance:**

Performance will be optimized so that both authentication and registration processes complete quickly for a smooth user experience. The system will also be designed to manage concurrent user requests efficiently, allowing multiple users to interact with the system without delays or slowdowns.

3. **Usability:**

Usability will be a key focus, ensuring that the interface is intuitive, making user interactions straightforward. Any errors encountered during registration or login will be communicated through clear and helpful error messages, guiding users in resolving issues with minimal frustration.

4. **Scalability:**

scalability will be built into the design, allowing the system to accommodate a growing user base and increased activity levels without loss of performance. This design will also facilitate future expansions, such as multi-factor authentication, making it adaptable to evolving security needs and new features.

5. **Layout:**

We want to build an easy and understandable layout for the modules in can be used for different kind of persons without problems and also with a clean and modern appearance.

3.Component architecture

3.1 Architecture and Tech Stack:

The architecture follows a three-tier model comprising the Frontend, Backend, and Database.

The application will follow a client-server model, the front-end will consist of a web, desktop or mobile application that communicates with a back-end hosted on a server. This server will manage user data, handle authentication, and interface with database.

3.2 Key technologies include:

Backend: Java, Jakarta Servlet API, JDBC

Frontend: JSP for user interfaces and HTML or CSS.

Database: MySQL or PostgreSQL for user data storage because we need a relational database

Communication: HTTP for the request from frontend to backend and will check to use

Encryption: Secure hashing algorithms like Bcrypt

3.3 Session Management

The sessions will be tracked using HTTP sessions, allowing the application to manage user states efficiently and securely. And we will improve may be implementing JWT (JSON Web Token).

3.4 API Descriptions

We are going to have three API Endpoint, one for Login, one for Register and the one for Reset

We are going to use an API for sending the email when the user has to reset the password. We will use an API provide from Sidemail that integrate a professional password reset email and also this API allow you to customize the email's appearance, so we can match this email with our branding.

4.Database structure

	Data Type	Constraints	Description
id	integer	PRIMARY KEY, AUTO_INCREMENT	Unique identifier for each user.
username	varchar (50)	UNIQUE, NOT NULL	User's unique username.
email	varchar (100)	UNIQUE, NOT NULL	User's email address for authentication.
password_hash	varchar(255)	NOT NULL	Hashed password for secure storage.
Create_date	Timestamp()	NOT NULL	The date that the user was created
Last_modified	Timestamp()	NOT NULL	The last date user profile was updated
Status	tinyint(1)	NOT NULL, DEFAULT=0	The status define if the user is active
Token	varchar(255)	DEFAULT= NULL	Used for sending emails and activate or reset password
Token_type	enum('activation', 'reset')	DEFAULT= NULL	Define if token is for reset password or for activate an account
Token_expires_at	datetime	DEFAULT= NULL	Define which day expire the token if the user maintain status 0

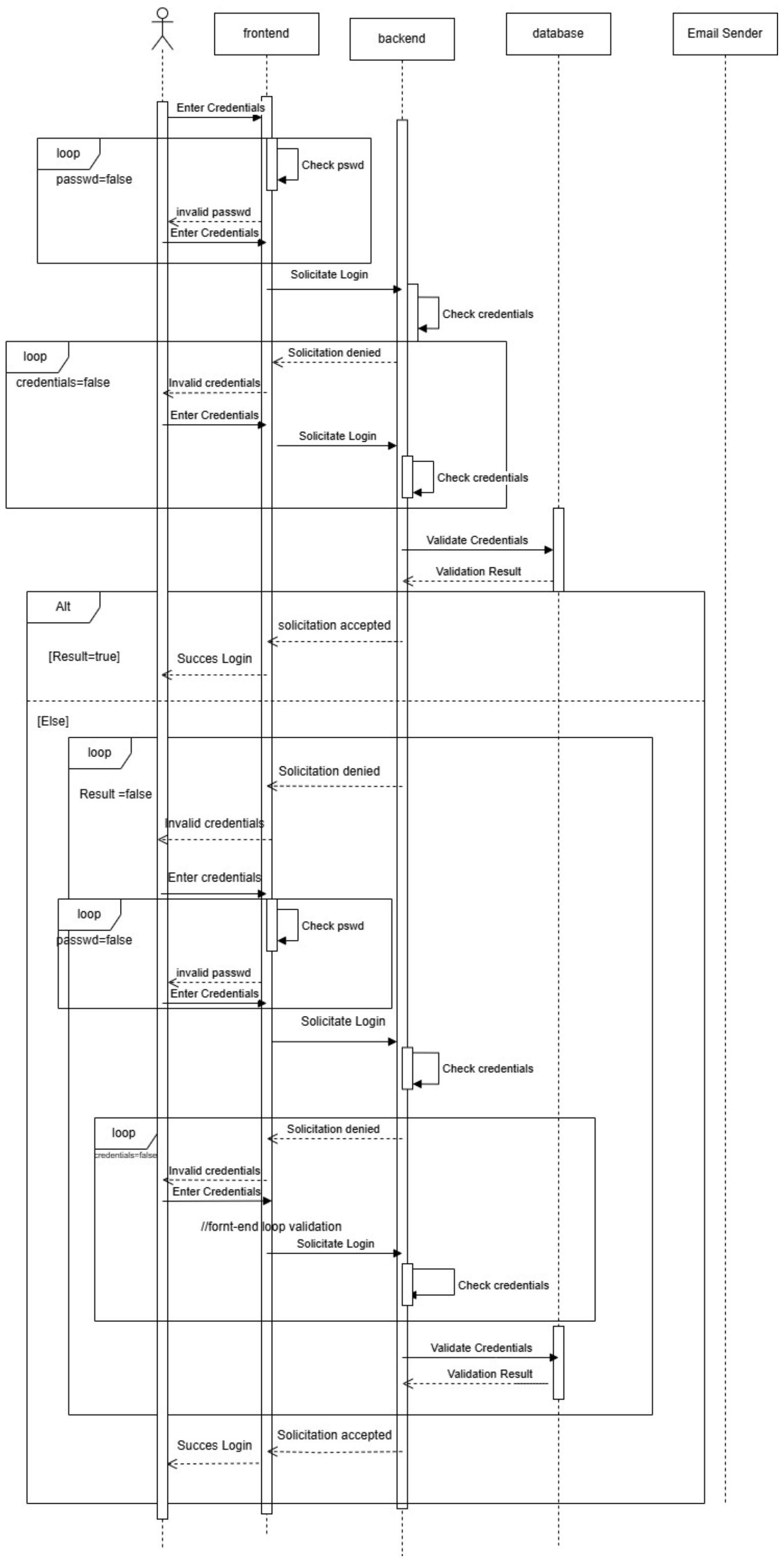
5.UML diagrams

5.1 Login Diagram

Diagram Structure:

- **Actors**: User

login user



Alternative Paths:

- incorrect credentials

5.2 Register User Diagram

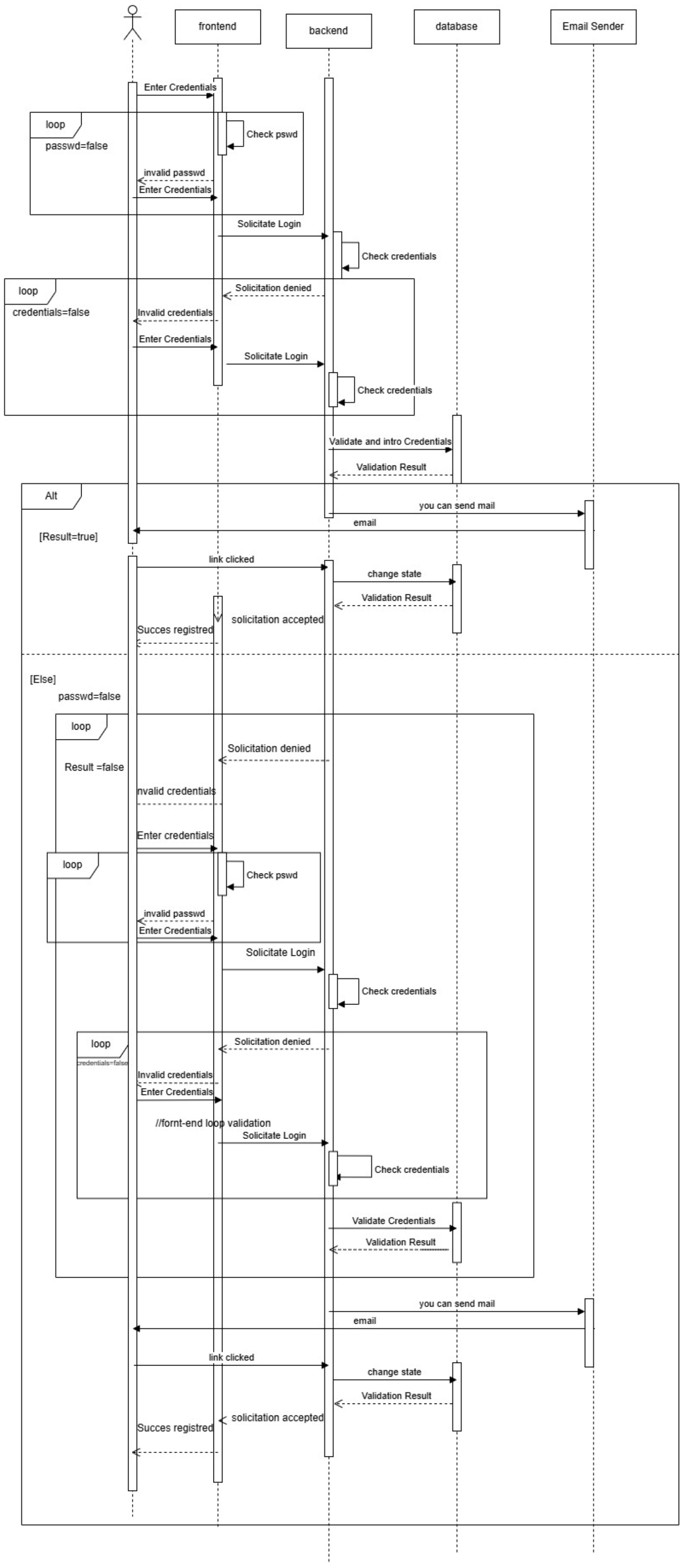
Diagram Structure:

- **Actors**: User

Alternative Paths:

- Invalid credentials.

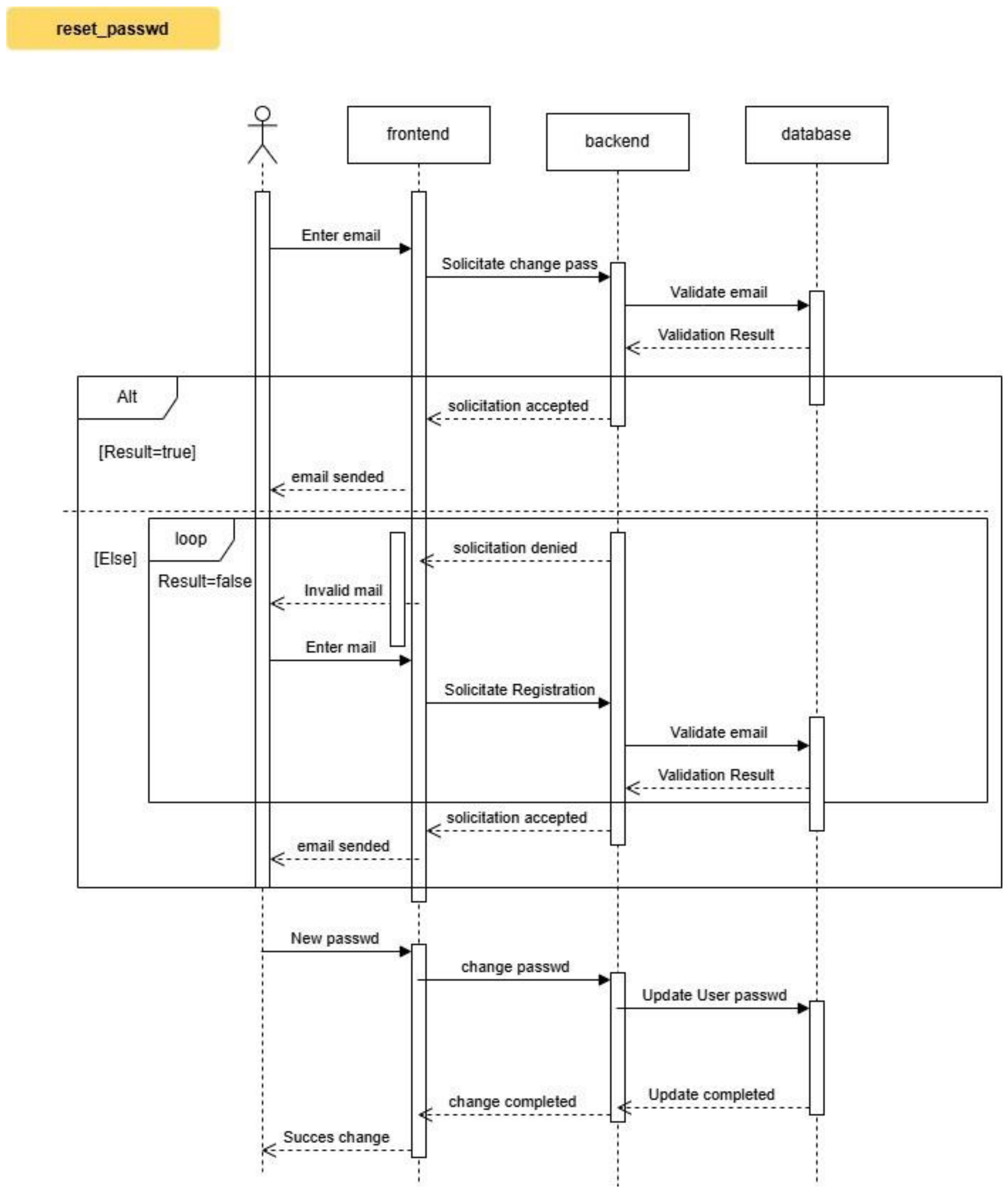
register user



5.3 Reset Password Diagram

Diagram Structure:

- **Actors**: User



Alternative Paths:

- Messages for email not found.