Brijeshkumar Himmatbhai Dholakiya (127201)

Mohammad Chehreghani Montazer (125531)

Requirement Remarks + Status Report

Ambiguity

1. **Multi-User Capability and Authentication Method**
   - **Issue:** The original document did not specify if the application supports multiple users or details about the authentication method.
   - **Resolution:** The client group clarified that the application would support multiple users and authentication will be implemented to manage user access. The method of authentication is explained in detail in the implementation phase of this document.
   - **Added Requirement:** The application shall support multiple users. Each user will have a unique username and email. Users must authenticate themselves to access their events and settings.
2. **Task Management vs. Event Management**
   - **Issue:** There was confusion whether "task management" and "event management" are separate functionalities.
   - **Resolution:** Client clarified that "task" was meant to refer to "event," so the application will only manage events.
   - **Updated Requirement:** The application will allow users to create, edit, and delete events, and to read their created events in a tabular format. Each event can have attributes like title, date, time, and description.
3. **Different Views Clarification**
   - **Issue:** It was unclear what "different views" meant.
   - **Resolution:** It's clarified that different views include a list view of events and a calendar view.
   - **Updated Requirement:** The application will provide only a list view for users. The calendar view would be an addition for future implementations.

Incompleteness

1. **Database Requirement**
    - **Issue:** It was not explicitly mentioned that a database is required for storing events and user information.
    - **Resolution:** Acknowledged that a database is implied by the nature of the application and should be explicitly included in the requirements.
    - **Added Requirement:** A database shall be used to persistently store user information and events. This will allow events to be saved and retrieved across different sessions.
2. **User Interface Simplicity**
    - **Issue:** The requirements mentioned a simple UI, but specifics were lacking.
    - **Resolution:** The aim is to have a simple, non-crowded UI without unnecessary graphics.
    - **Updated Requirement:** The user interface shall be intuitive and straightforward, avoiding unnecessary graphical elements. It will include essential elements like input fields, buttons, event list, and a calendar view.
3. **Security and Credential Handling**
    - **Issue:** Initial requirements skipped over security measures for credentials due to perceived complexity.
    - **Resolution:** While complete security measures like hashing were deemed too complex for beginners, a note about appreciation for potential implementation would suffice.
    - **Updated Requirement:** Basic security will be implemented where feasible. For example, storing passwords in plain text should be avoided; however, if hashing is implemented, it will enhance security.

Imprecision

1. **Clear Error Messages and User Guidance**
    - **Issue:** The term "clear error messages" was vague.
    - **Resolution:** Error messages should inform the user of the exact nature of the error and provide hints for correction.
    - **Updated Requirement:** The application will provide clear, descriptive error messages that help the user identify and correct mistakes. For instance, if a user enters an incorrect password, the application will specifically indicate that the password is incorrect rather than a generic error message.

Implementation General Technical Overview

- Programming Language – Java
- Additional Tools & Dependencies: JavaFX
- Database – SQL
- Build Management – Gradle
- IDEs Used (compatibility): VScode and Eclipse
- Version Control: Git + Gitlab

Implementation Status

1. Database Implementation: Yes
    - For storing of user credentials and events created by each user

2. User Interface: Yes
    - Clear and uncrowded UI with specific window frames for various features including the first interaction, register, login, home menu, event list, and more.

3. User Creation: Yes
    - UI implemented specific window frame.

4. User Authentication and Constraints: Yes
   Login and Sign-up buttons
    - Validation for 'Register' in the front-end:
        - Username accepts only alphanumeric values.
        - Emails should contain '@' and '.com' ending.
        - Passwords of at least 6 characters.
        - Password confirmation field should match with the input in password field.
    - Validation for 'Register' in the backend:
        - Username and password must both be unique.
    - Validation for 'Login' in the front-end:
        - Username input accepts either one of username or email address.
        - Both 'username' and 'password' fields must have given input values.

5. Event Creation: Yes
    - API implementation for creating an event.
    - Detail input fields including title, date, time, and description.

6. Event Listing: Yes
    - UI window frame is implemented with a method that lists the events created for each user.

7. Task Management: Partially
    - Edition and deletion of the CRUD operations are pending.

8. Navigation: Yes
    - From the home menu, users can navigate through different tabs including event list, calendar view, etc.

9. Reliability: Yes
    - Error handling is done for errors occurred, such as:
        -

10. Performance: Pending
    - There would be a need for test cases for this requirement.

11. Constraints: Yes
    - The application is implemented by following the technical requirements such including the use of java language for implementation, building based on Gradle, and avoidance of external library usage.

Future Implementation:

In addition to the remaining of the requirements below requirements are to be considered as the next phase of implementation.

1. Forgot password option which happens by a temporary password provided with a link in an email sent to the email address of the user that is valid for a short limit time, for example 1 hour. By pressing on the link, user is taken to a window frame where they can enter the temporary password, new password, and the confirmation to the new password.

2. Edition of the profile, where users can edit their username and password, as well as addition of a profile photo.
3. PDF export of the event list of the user, sent to them via an email and possible to save on their system.
4. Tags for events which could provide them categorization of their events by classifying events of the same tag into separated event lists.
5. Multiple language support for every input field of the program other than email address and password inputs.
6. Time zone selection, for each event with having CEST as the default time zone.
7. Admin panel for handling the database, which naturally is followed by having a role management system defined for user and administrator, and the UI implementations necessary to the administrator.
8. CLI implementation for the project.