1. Introduction

This document outlines the fully implemented and unit/integration tested user stories for our StudentHUB project to date. It details each story that includes GUI functionality, identifies the main developers responsible for each story, and shows the distribution of testing responsibilities among team members. Additionally, the document includes bug reports and code review findings related to the user stories.

2. Updated User Story Assignments & Descriptions

2.1 Course Calendar

Assigned To: Abdulkadir Ahmed
Status: Complete
Expected ITR3 Enhancements:

- **UI/UX Improvements:** Better spacing and a refined 24-hour calendar display.
- **Backend Enhancements:** Smoother database interactions for course management.
- **Bug Fixes & Testing:** Continued testing to ensure reliability and responsiveness.

2.2 User Accounts (New)

Assigned To: Abdulkadir Ahmed
Expected Completion: March 25
Scope and Features:

Registration:

- The user provides a username, password, year of study, and program.
- Frontend validations enforce that:
    - **Username:** Must be more than 6 characters and less than 14 characters.
    - **Password:** Must be exactly 6 characters.
    - **Year of Study:** Must be a number between 1 and 20.
- The backend verifies that usernames are unique and stores the user's details in our Render-hosted PostgreSQL database.

Login:

- Users can log in with their username and password.
- Upon successful login, the app uses AsyncStorage for auto-login, displaying the user's account details.

Account Management:

- The logged-in view displays user details (username, year of study, and program) and provides options to "Logout" or "Delete Account."
- "Delete Account" sends a DELETE request to remove the user from the database and logs them out.

Auto-Login:

- The system retains the login state across app restarts, so users remain logged in until they explicitly log out or delete their account.

2.3 Group Chat (Without User Accounts)

Assigned To: Simmonly & Digvijay
Notes:

- A basic messaging backend has been implemented.

2.4 Group Chat (With User Accounts)

Assigned To: Simmonly & Digvijay
Status: Not Started
Expected Completion: March 28
Notes:

- Once the User Accounts feature is finalized, integration into the Group Chat will be implemented to enable personalized messaging.

2.5 AI Advising Chatbot

Assigned To: Fardad Rashidian
Status: Ongoing
Expected Completion: March 31
Scope:

- **Backend:** Processes user PDFs and summarizes content using GPT-4.

- **Frontend Integration:** The chatbot is being integrated into the UI with ongoing improvements to the user experience.

---

3. Testing Assignment & End-to-End Manual Test Cases

Each team member is assigned user stories that they were not heavily involved in implementing to ensure independent testing. Below are sample test cases for the **User Accounts** story (note that the Course Calendar also has a comprehensive set of frontend tests, as documented in its test suite).

User Accounts – End-to-End Test Cases

Test Case 1: Username Validation

- **Objective:** Verify that the username field only accepts input between 7 and 14 characters.
- **Steps:**
    1. Open the registration form.
    2. Enter a username with 6 characters (e.g., “user1a”).
    3. Fill in a valid password, year of study, and program.
    4. Tap “Submit Registration.”
    5. **Expected Result:** An error message “Username must be more than 6 characters” is displayed.
    6. Repeat with a username of 15 characters and expect an error indicating the username is too long.

Test Case 2: Password Validation

- **Objective:** Ensure that the password field only accepts exactly 6 characters.
- **Steps:**
    1. Enter a valid username and other details.
    2. Enter a password with 5 or 7 characters.

3. Submit the form.

4. **Expected Result:** An error message “Password must be exactly 6 characters” appears.

Test Case 3: Year of Study Validation

- **Objective:** Confirm that the Year of Study field accepts only numbers between 1 and 20.

- **Steps:**

    1. Enter a valid username, password, and program.

    2. Enter an invalid number (e.g., “0” or “21”) in Year of Study.

    3. Submit the registration.

    4. **Expected Result:** An error message “Year of Study must be a number between 1 and 20” is displayed.

Test Case 4: Successful Registration & Auto-Login

- **Objective:** Validate that a user with all valid inputs is registered and auto-logged in.

- **Steps:**

    1. Enter a valid username (7–14 characters), a 6-character password, a valid year (e.g., “3”), and a program.

    2. Tap “Submit Registration.”

    3. **Expected Result:** The logged-in view is displayed with the correct user details.

    4. Close and reopen the app to ensure that auto-login persists.

Test Case 5: Logout Functionality

- **Objective:** Verify that the “Logout” button logs the user out and clears stored data.

- **Steps:**

    1. While logged in, tap “Logout.”

2. **Expected Result:** The app returns to the login screen and all form fields are cleared.

Test Case 6: Delete Account Functionality

- **Objective:** Ensure that the “Delete Account” button removes the user from the database and logs them out.
- **Steps:**
    1. While logged in, tap “Delete Account.”
    2. Confirm any prompts.
    3. **Expected Result:** The account is deleted, an alert confirms deletion, and the app returns to the login screen.
    4. Attempt to log in again using the deleted credentials; the system should reject the login.

Note: The Course Calendar feature also has an extensive set of frontend tests (using Jest and Testing Library), ensuring that all UI components, validations, and course scheduling functionalities are verified.

---

4. Bug Reporting & Code Review Summary

Bug Reporting

- **Bug 1:** Duplicate username error message appears only briefly; recommendation to improve the visibility and duration of error alerts.
- **Bug 2:** On some devices, the auto-login persistence does not refresh immediately upon app restart.
  (All bugs are reported on the team’s GitHub repository with steps to reproduce, observed behavior, and expected behavior.)

Code Review & Code Smells

- **Security Concern:** Storing passwords as plain text is a major risk.
    - *Recommendation:* Implement password hashing (e.g., using bcrypt).

- **Repeated Validation Logic:** Registration validations appear in both the frontend and backend.
    - *Recommendation:* Abstract the validation logic into reusable functions to reduce duplication.
- **UI Feedback:** Error messages in the frontend are sometimes unclear or too brief.
    - *Recommendation:* Improve error messaging for better user guidance.
- **General Design:** Some functions, especially in the registration endpoint, are lengthy and could be refactored for clarity.
    - *Recommendation:* Refactor lengthy methods and enhance code modularity.

---

5. Conclusion

This document serves as the comprehensive assignment deliverable for our team. It outlines our current user stories with clear implementation details and testing assignments, alongside documented bugs and code review findings. All team members are responsible for updating this document on our team's GitHub repository as further issues and improvements are identified.