Adventist University of Central Africa

P.O. Box 2461 Kigali, Rwanda | www.auca.ac.rw | info@auca.ac.rw

Faculty of Information Technology
Mobile Programming (INSY-8414)
MID-SEM Exam
PROJECT TITLE: MyBook Library
Due date: 21 July 2024

Develop a cross-platform mobile application using Flutter or React Native that allows users to manage their personal book library. The app will include features to add, edit, delete, and view books. Users can also rate books and mark them as read or unread. The app will store user preferences such as the sorting order of the book list using Shared Preferences.

Key Features:

Flutter Version:

1. **UI Design:**
   - Use Flutter's Material Design widgets for a cohesive look
   - Implement a home screen with a list of books
   - Create screens for adding/editing book details
   - Design a book detail view
   - Implement a settings screen
2. **Data Persistence (SharedPreferences):**
   - Use shared_preferences package to store user preferences
   - Save sorting preferences (by title, author, rating)
   - Store theme preferences (light/dark mode)
3. **Data Management (Content Provider equivalent):**
   - Use SQLite with sqflite package for local database
   - Implement CRUD operations for books
   - Create a data access layer to manage database operations
4. **State Management:**
   - Use Provider or Riverpod for state management
5. **Key Features:**
   - Add, edit, and delete books
   - Rate books and mark as read/unread
   - Sort books based on different criteria
   - Search functionality
   - Theme switching (light/dark mode)

React Native Version:

1. **UI Design**:
   - Use React Native Paper or Native Base for consistent UI components
   - Create a home screen with a FlatList of books
   - Create screens for adding/editing book details
   - Design a book detail view
   - Create a settings screen
2. **Data Persistence (SharedPreferences equivalent):**
   - Use @react-native-async-storage/async-storage for storing user preferences
   - Save sorting preferences (by title, author, rating)
   - Save sorting preferences and theme settings
3. **Data Management (Content Provider equivalent):**
   - Use react-native-sqlite-storage for local database management
   - Implement CRUD operations for books
   - Create a data service layer to handle database operations
4. **State Management:**
   - Use Redux or Context API for state management
5. **Key Features:**
   - Add, edit, and delete books
   - Rate books and mark as read/unread
   - Sort books based on different criteria
   - Search functionality
   - Theme switching (light/dark mode)

Common Implementation Steps:

1. Set up the project structure
2. Design and implement the UI components
3. Implement data persistence for user preferences
4. Set up the local database and CRUD operations
5. Implement state management
6. Add sorting and filtering functionality
7. Implement search feature
8. Add theme switching capability
9. Test the application on both iOS and Android

-END-