



Adventist University of Central Africa

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

ID: 25969

Name: Duhimbazimana Johns

CheckMe – Todo App

Project Overview

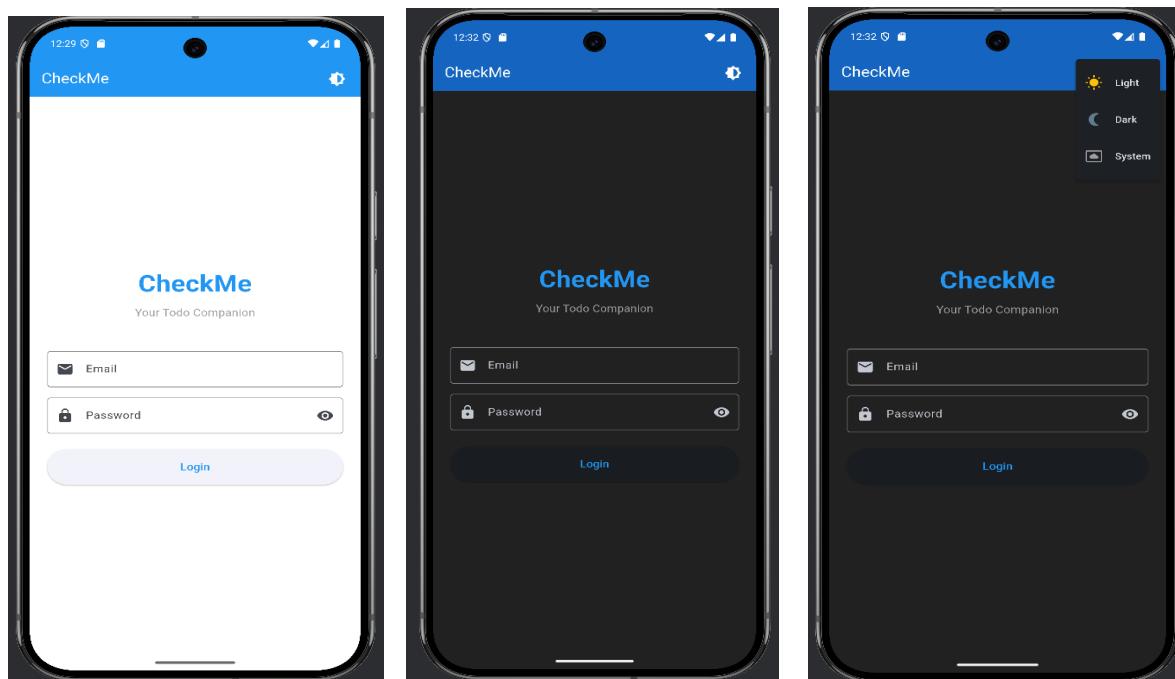
CheckMe – Todo App is a mobile app developed for users to be able to complete their daily tasks effectively. The app provides users with the ability to log in, view their dashboard, add new todos, mark a task as completed, delete an item, view a task's info, mark an item as a certain category, and search in their list of todos. The app also supports theme mode switching (light/dark/system) and integrates RivePad for interactive animations. The project is a practice of implementing skills in state management, UI designing, and multi-screen navigation using Flutter.

Implemented Core Features

1. Login Screen

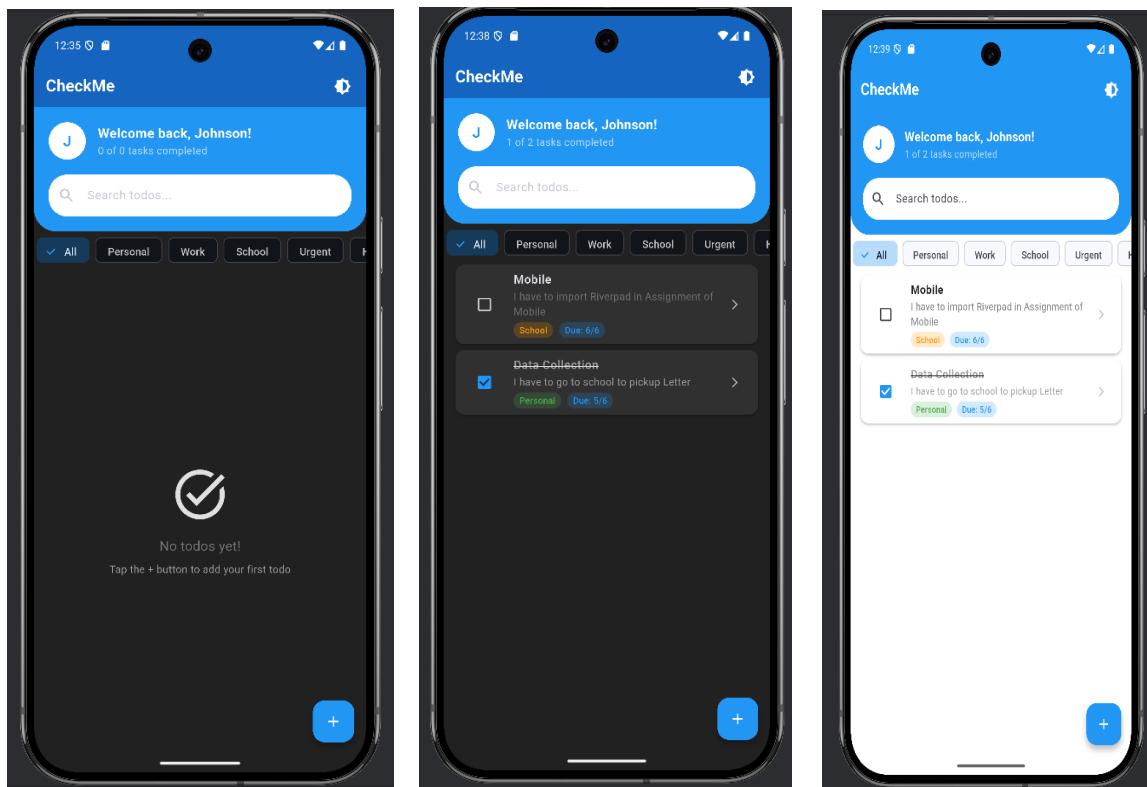
The application starts with a login screen that supports password and email input fields. Validation rules have been included to ask if the email address is in the correct format and if the password contains a minimum of eight characters. Upon receiving appropriate credentials, the user is navigated to the Home Screen of the app. And also supports theme mode switching (light/dark/system) and integrates RivePad

Login (Theme Mode)



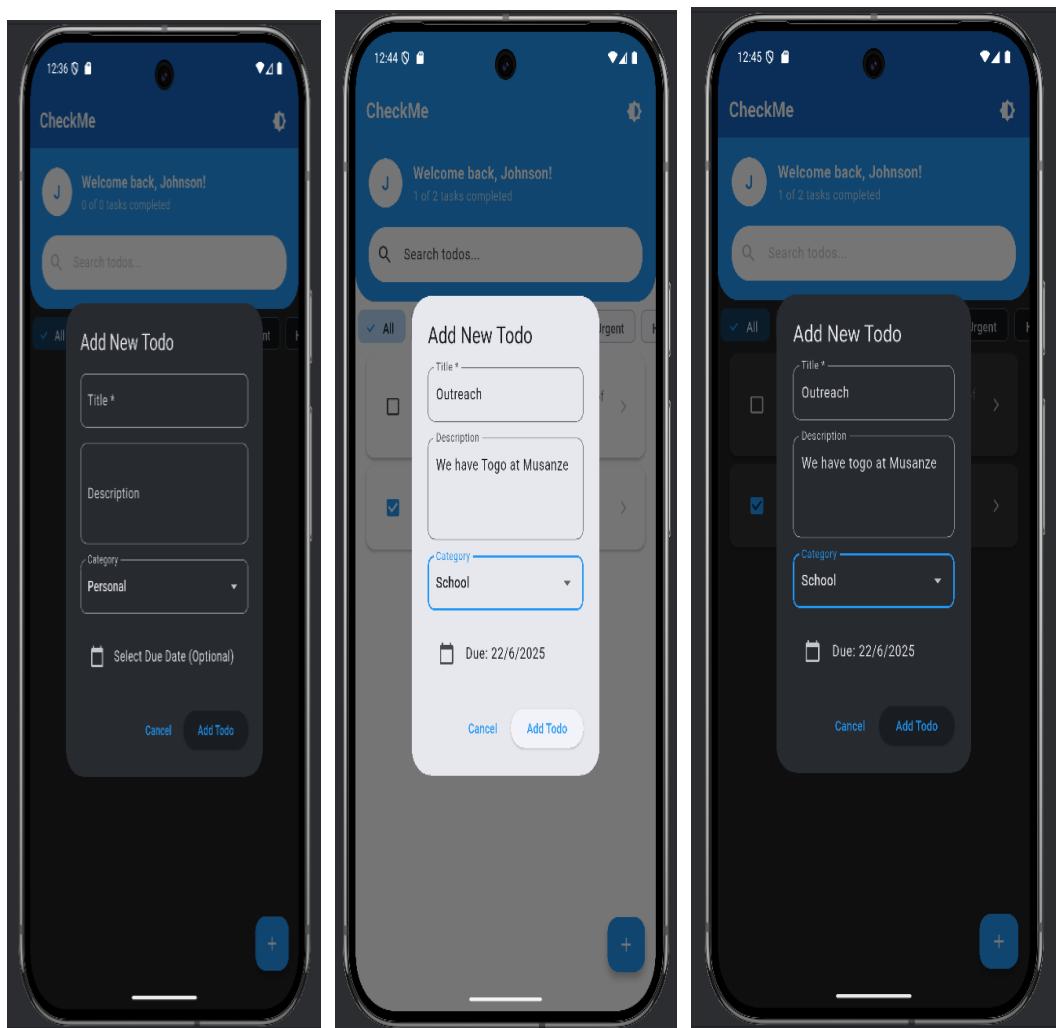
2. Home Screen

Upon login, the user is greeted by a greeting message displaying the user's name and avatar. The Home Screen displays a dashboard that displays a scrollable list of todos. The todo item displays the title and a checkbox, and the list is dynamically built using the ListView widget. Any change in the list, such as adding or editing a todo, is implemented using the setState function in order to display updates immediately in the user interface. And also supports theme mode switching (light/dark/system) and integrates RivePad



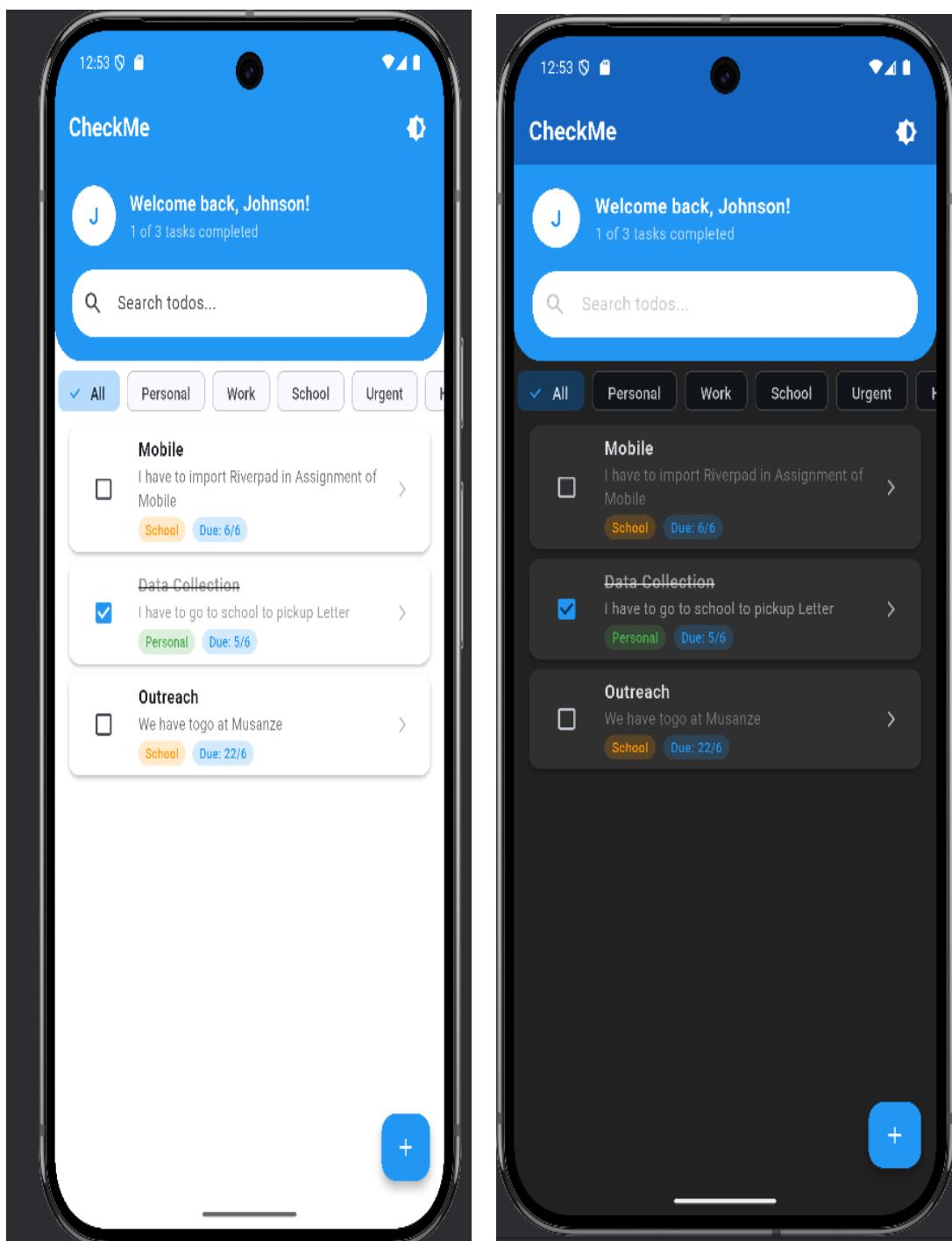
3. Add Todo Feature

User may add a new todo by tapping the floating action button, which reveals a new form. The form prompts the user to input a title for the todo and includes an optional description. The form may include fields for due date and category selection. When the user taps the save button, the new todo is inserted into the list, and the UI is refreshed through setState. The users are able to assign each todo a specific category, say "School", "Personal", or "Urgent". The application also has filtering tools by which the users can view todos belonging to a specific category. When a new todo is being added, the user can set a due date. If it has already expired and the todo is not yet done, the app indicates it with an "Overdue" label, typically red colored to be noticed. And also supports theme mode switching (light/dark/system) and integrates RivePad



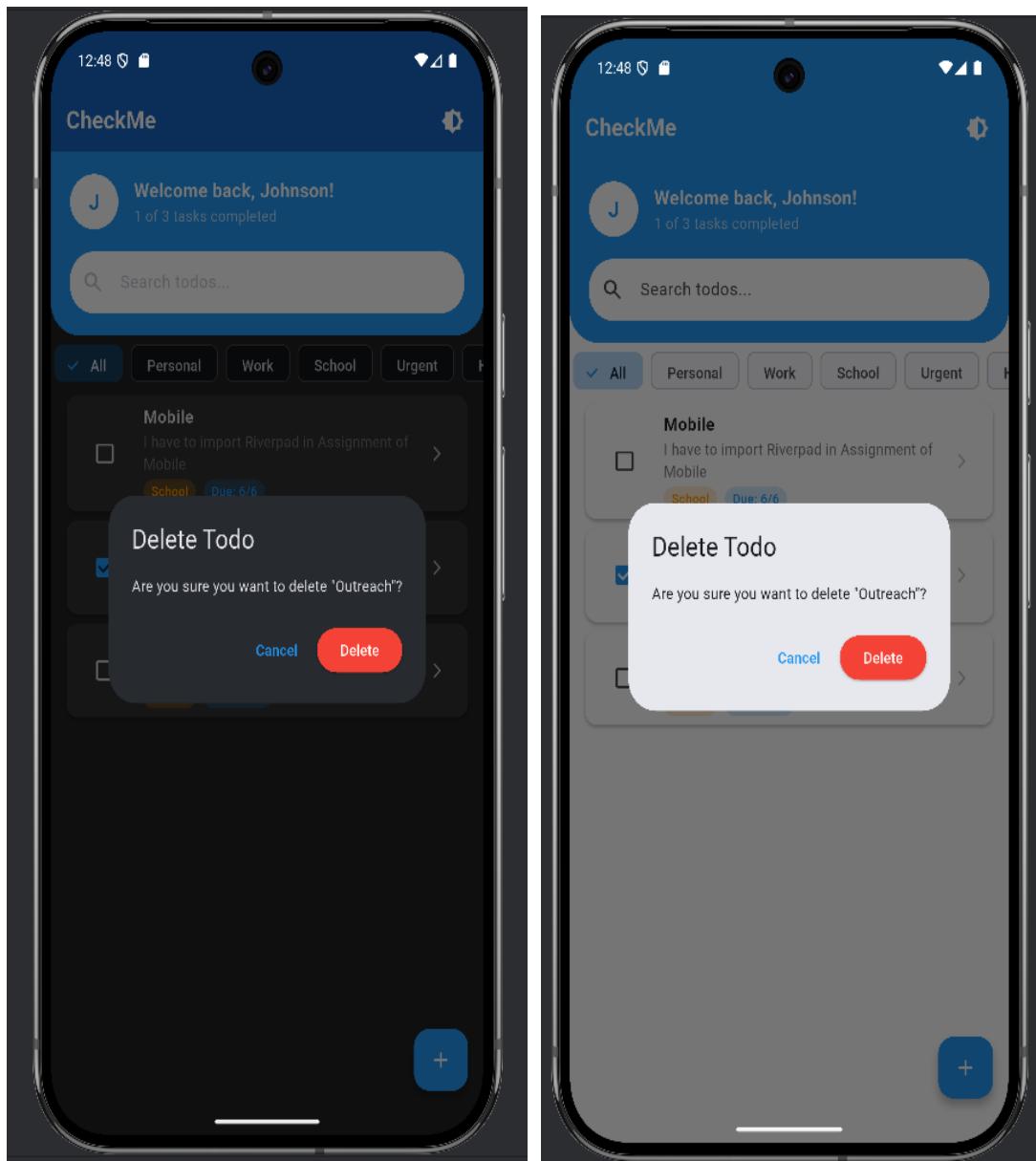
4. Mark as Done

Each todo item contains a checkbox where the user can mark the task done. When a todo is marked as done, it is rendered visually with a strikethrough or faded text to indicate that it has already been completed.



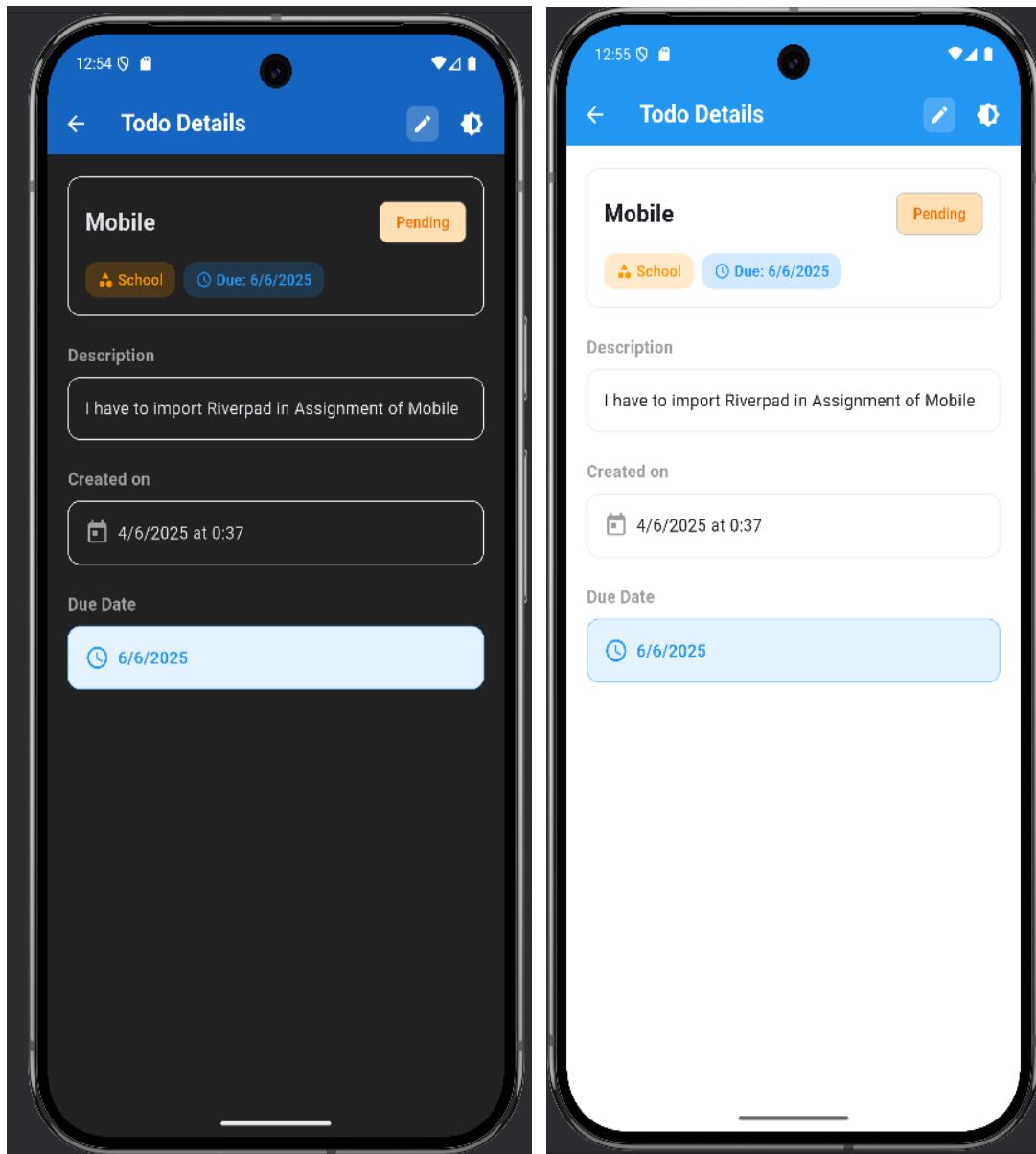
5. Delete Todo

Users can delete a todo from the list by swiping the item to left or long pressing on the item. Such an interaction will remove the selected todo from the list and refresh the UI.

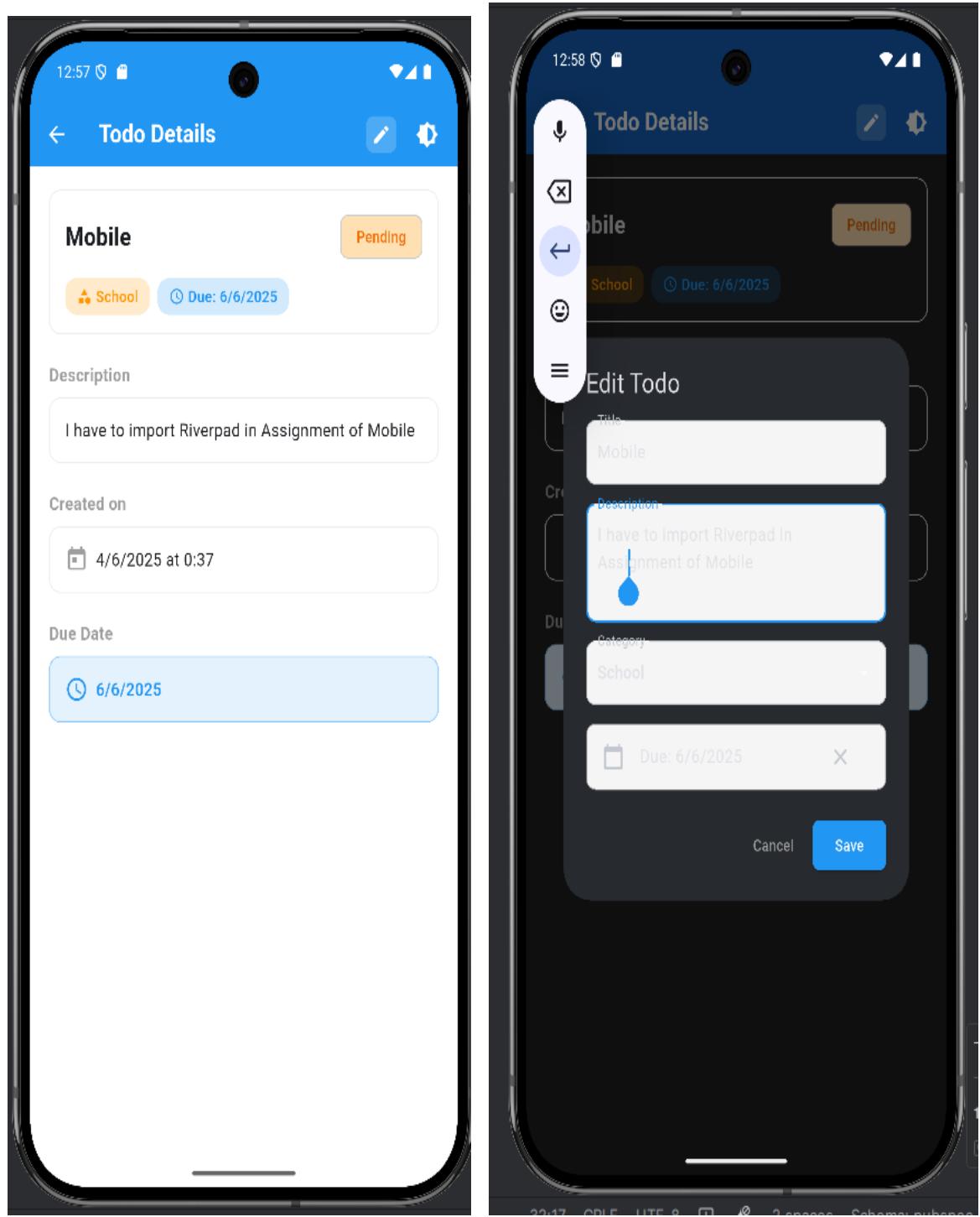


6. Todo Details Page

When a specific todo item is clicked by the user, a new screen is displayed to provide complete information about the task. This includes the title of the todo, description, creation date, and the ability to edit information. This aspect adds depth and significance to the todo items and enhances user experience.

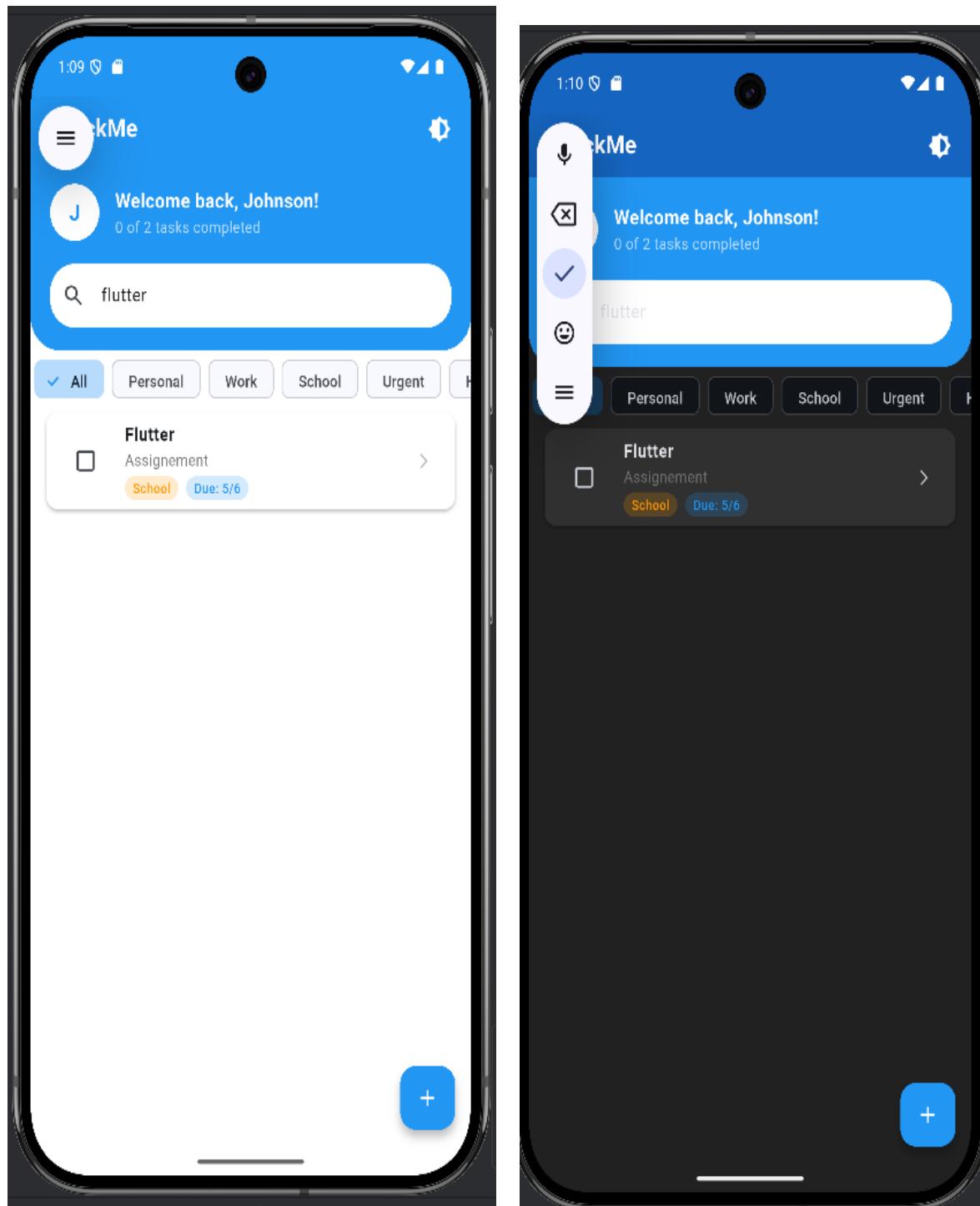


Edit Screen



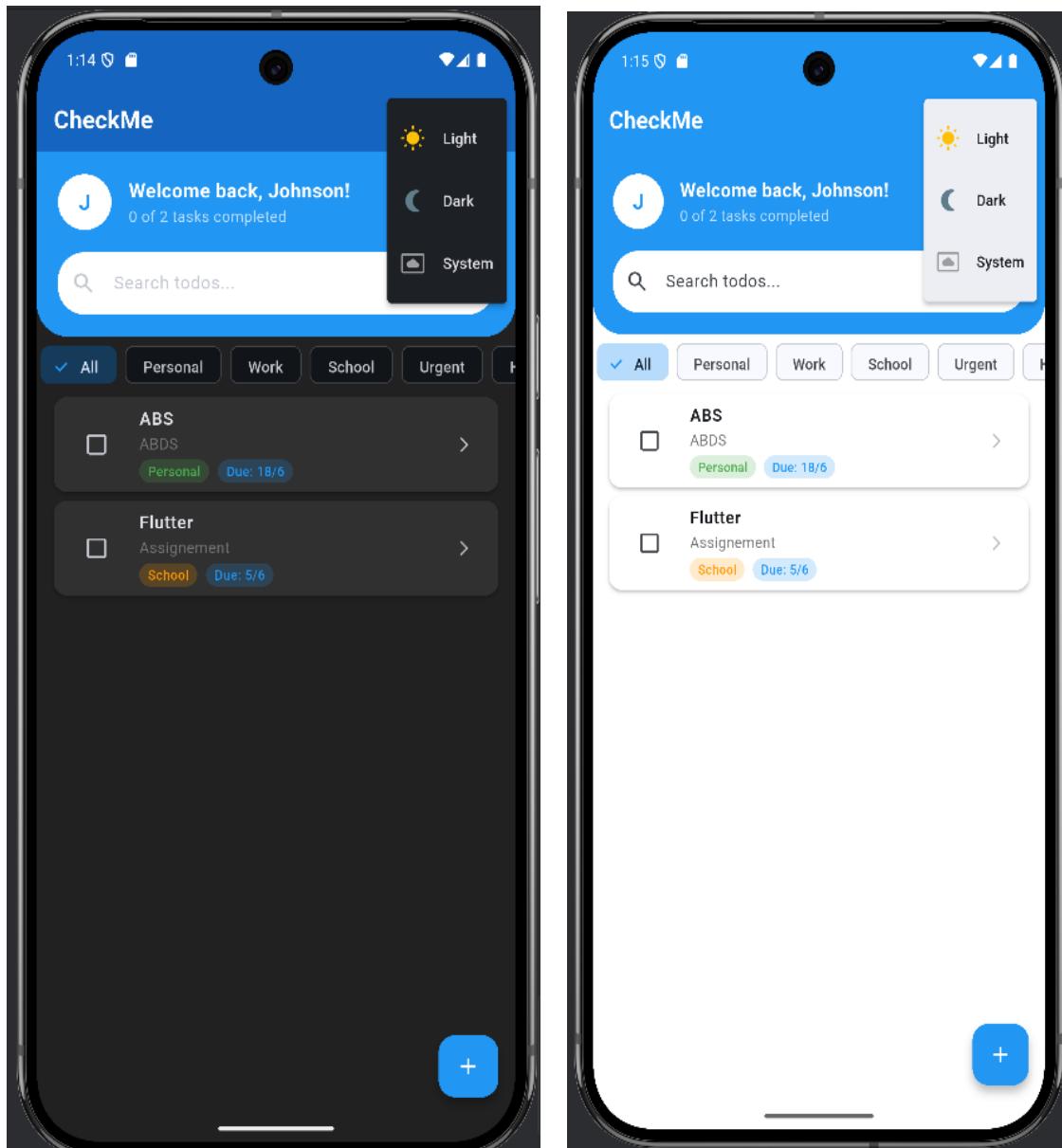
8. Search Todos

The application features a search bar which allows users to easily locate specific todos. When the user enters a query, the list dynamically filters out and displays only the relevant todos that match the query, either by title or description. This aspect greatly enhances the usability of the application for users with a lot of tasks to handle.



8. Change Theme Feature

Switch between **Light Mode**, **Dark Mode**, or follow **System Theme**.



❖ Technologies Used

This project was developed with Flutter for the UI and Dart for the programming language. The application uses the `setState` method for local state management and `Navigator` to navigate between the screens. Theme mode support was imported to allow switching between light and dark themes, and `RivePad` was integrated to add interactive animations.