

MOBILE APPLICATION PROJECT PRESENTATION

DEVELOPED USING FLUTTER

Software Development

DLBSEPPSD01_E

Ibe, Christopher Obinna

92107689

Tutor

Remfert, Christian

GitHub Repository

<https://github.com/Etitii27/iu-software-project-flutter.git>

Date

6th September, 2024

INTRODUCTION

Project Overview

A Flutter-based mobile application that combines three key functionalities

Chat App

Weather App

Todo App

PROBLEM STATEMENT

Users often need multiple apps to handle different daily tasks such as communication, weather checking, and task management. The goal was to create a single, unified platform that caters to these needs efficiently.

PROJECT GOALS

Chat App

Provide a real-time messaging platform for users to communicate

Weather App

Offer current weather data and forecasts to users.

Todo App

Enable users to manage and organise their daily tasks.

PLANNING AND ORGANIZATION

Approach

Used an Agile Scrum methodology to manage the development process, allowing for iterative development and continuous feedback.

Justification for Scrum

Scrum was chosen for its flexibility and ability to handle changing requirements. The iterative approach facilitated regular sprint reviews and adjustments.

PROJECT PHASES

- Planning: Define project goals, user needs, and requirements.
- Design: Create wireframes and architectural models.
- Development: Implement ChatApp, Weather App, and Todo App functionalities.
- Testing: Conduct unit, widget, and integration tests.
- Evaluation and Refinement: Evaluate the app and make necessary improvements.

STAKEHOLDERS

- Users: End-users who will use the app for chatting, checking weather, and managing tasks.
- Development Team: Responsible for the design, implementation, and testing of the application.

KEY RESPONSIBILITIES

- **Phase Planning:** Sprint planning, daily stand-ups, sprint reviews, and retrospectives
- **Work Breakdown Structure:**
 - **Sprint 1:** User Authentication and ChatApp functionality.
 - **Sprint 2:** Weather App integration and API connections.
 - **Sprint 3:** Todo App development and local storage management.
 - **Sprint 4:** Integration, UI/UX refinement, and testing.

SOFTWARE ARCHITECTURE OVERVIEW

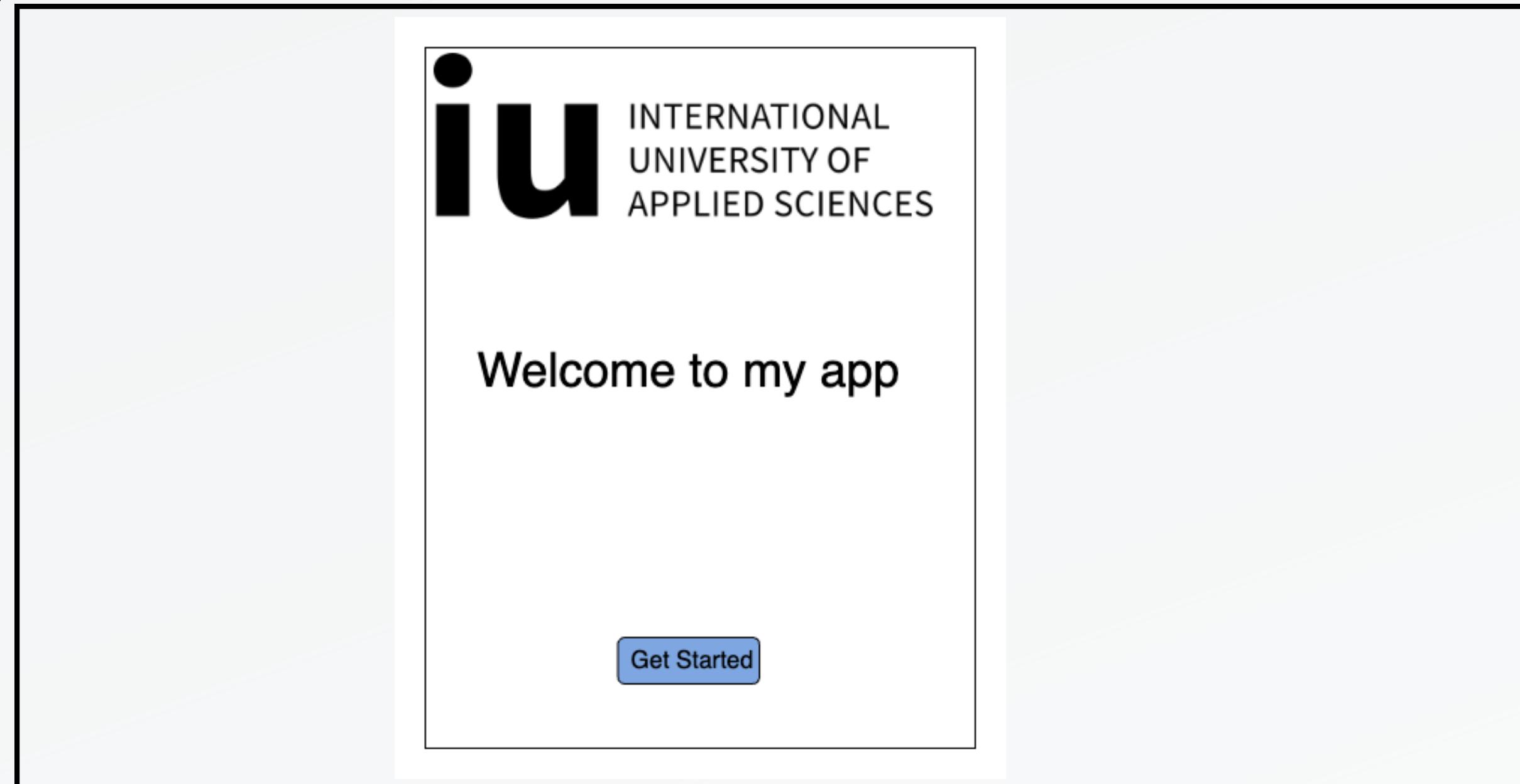
Architecture Style

MVVM (Model-View-ViewModel) architecture to ensure a modular, testable, and maintainable codebase.

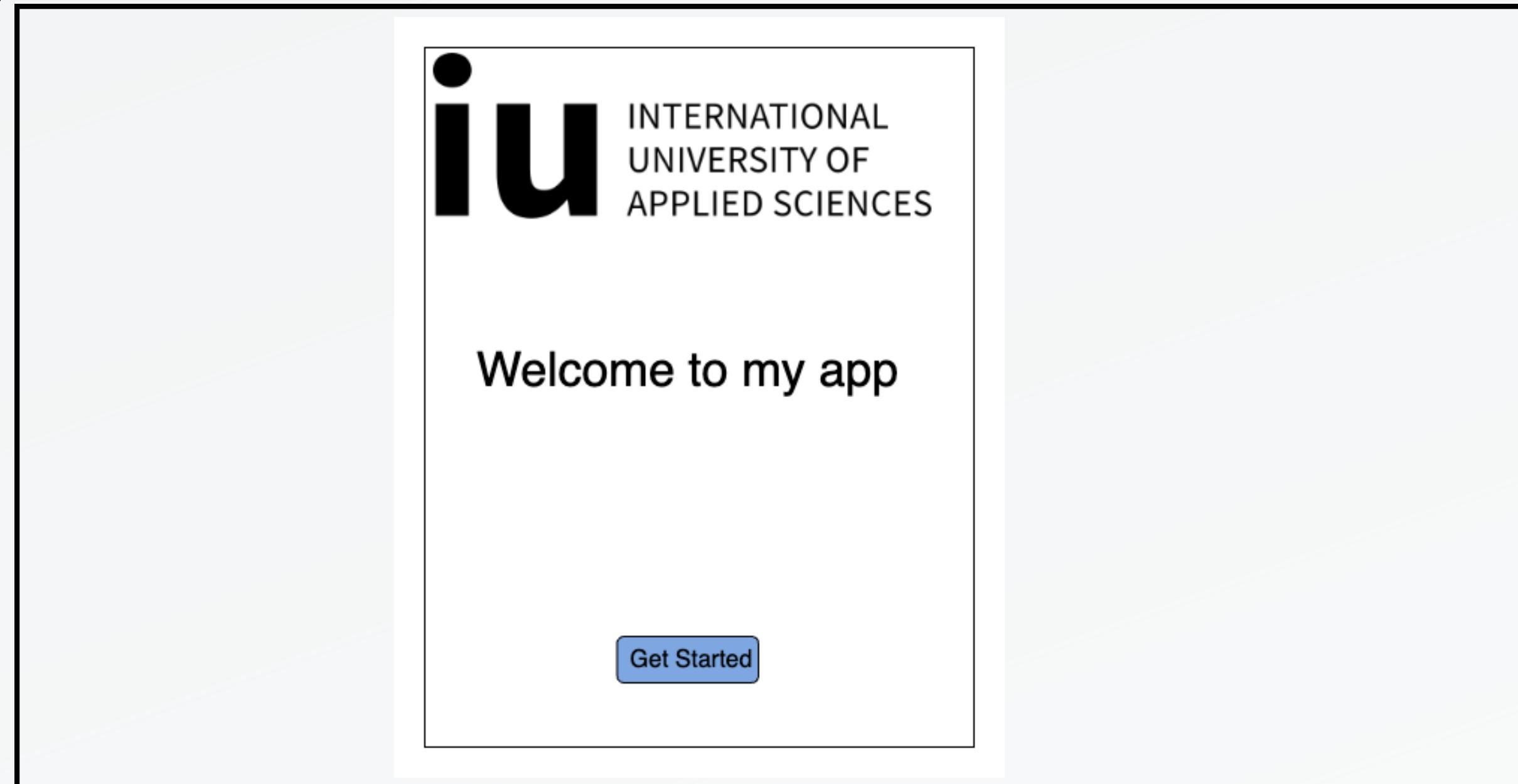
Tools and Frameworks

Flutter (Dart), Firebase,

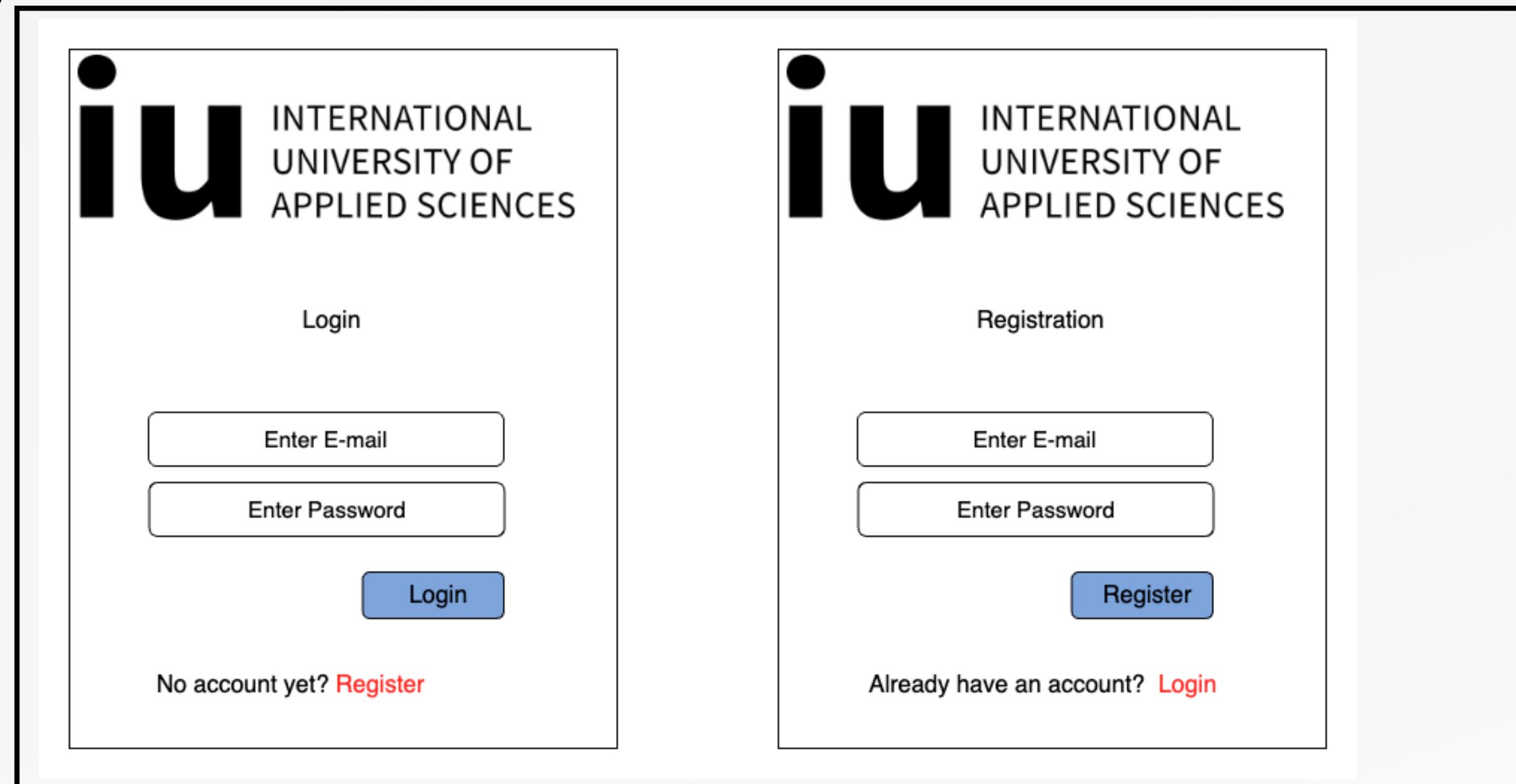
WIREFRAMES AND USER INTERFACES



WIREFRAMES AND USER INTERFACES



WIREFRAMES AND USER INTERFACES



The wireframe illustrates two user interface screens for the International University of Applied Sciences (iu).

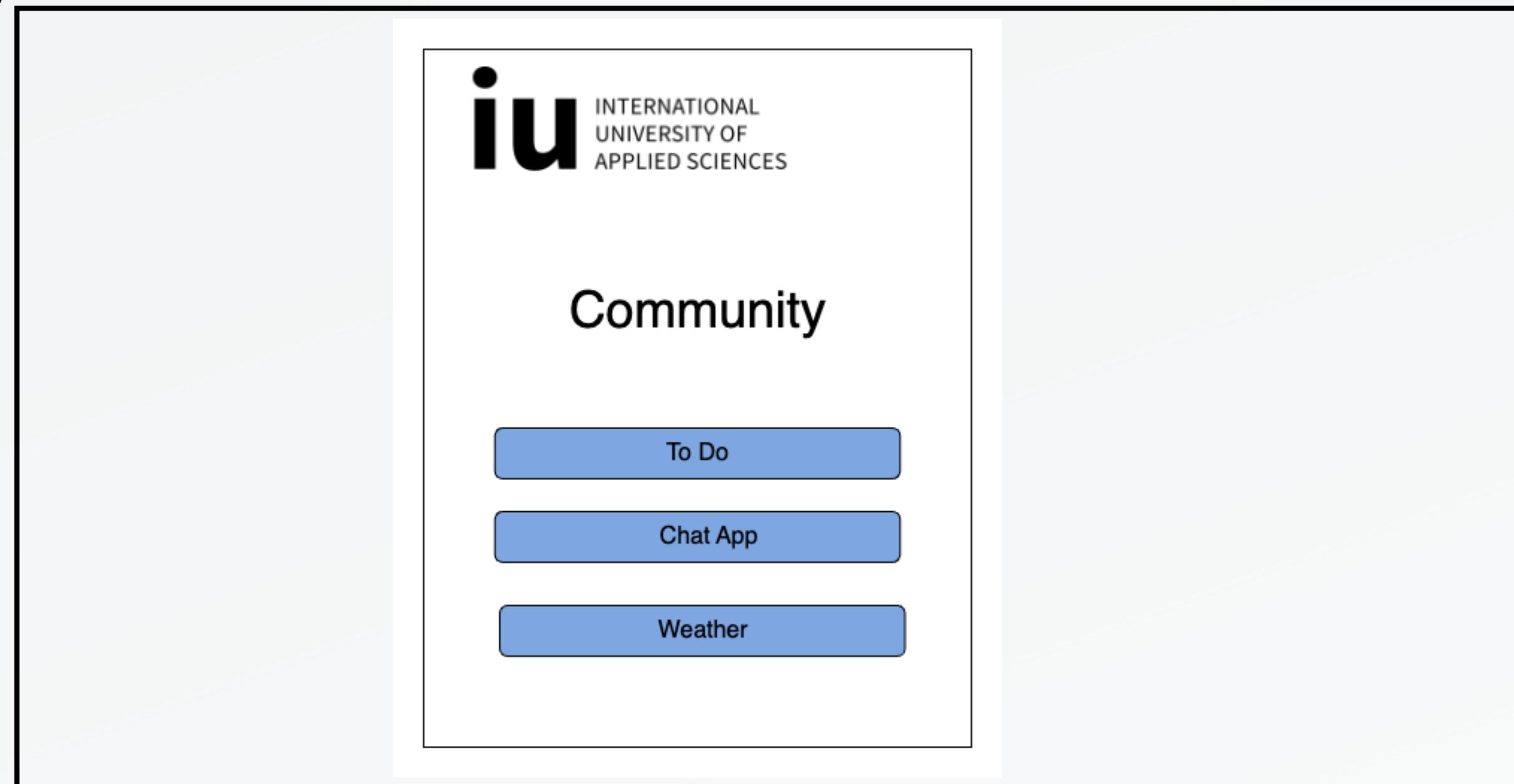
Login Screen:

- Header: iu INTERNATIONAL UNIVERSITY OF APPLIED SCIENCES
- Title: Login
- Input fields: Enter E-mail, Enter Password
- Action button: Login (blue)
- Text at bottom: No account yet? [Register](#)

Registration Screen:

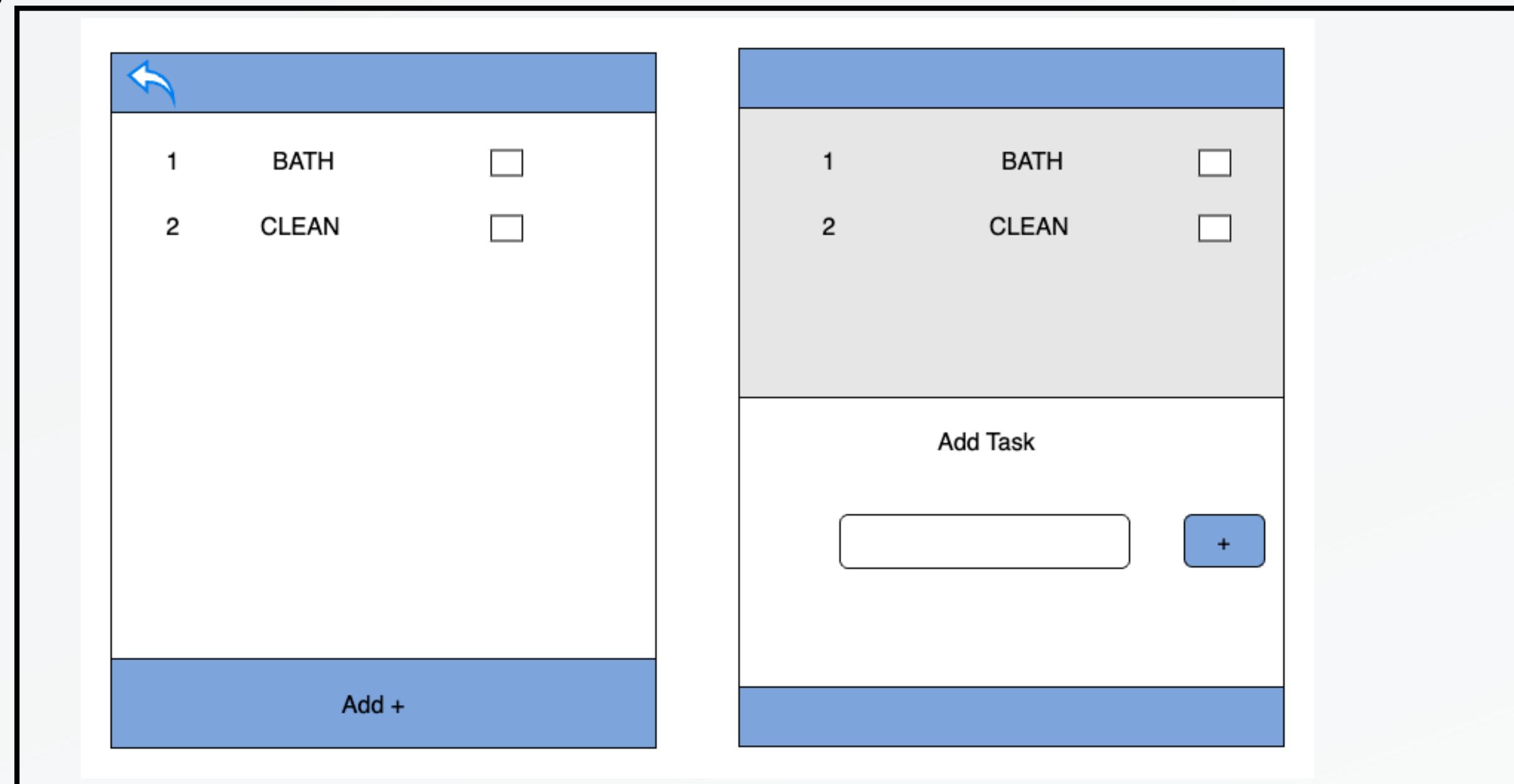
- Header: iu INTERNATIONAL UNIVERSITY OF APPLIED SCIENCES
- Title: Registration
- Input fields: Enter E-mail, Enter Password
- Action button: Register (blue)
- Text at bottom: Already have an account? [Login](#)

WIREFRAMES AND USER INTERFACES

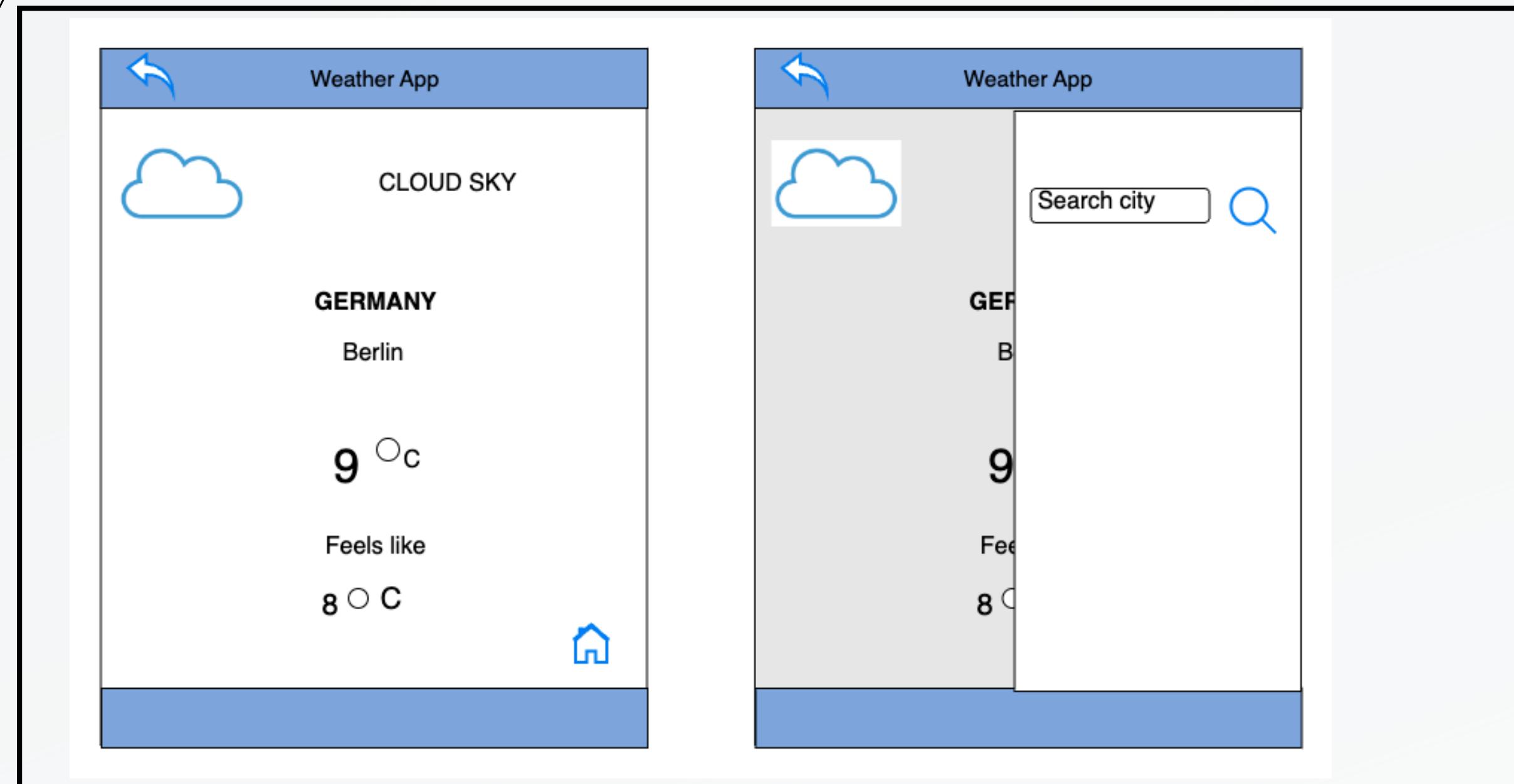


The image shows a wireframe of a user interface. At the top left is the 'iu' logo with the text 'INTERNATIONAL UNIVERSITY OF APPLIED SCIENCES'. Below it is the word 'Community'. Underneath 'Community' are three blue rectangular buttons with white text: 'To Do', 'Chat App', and 'Weather'.

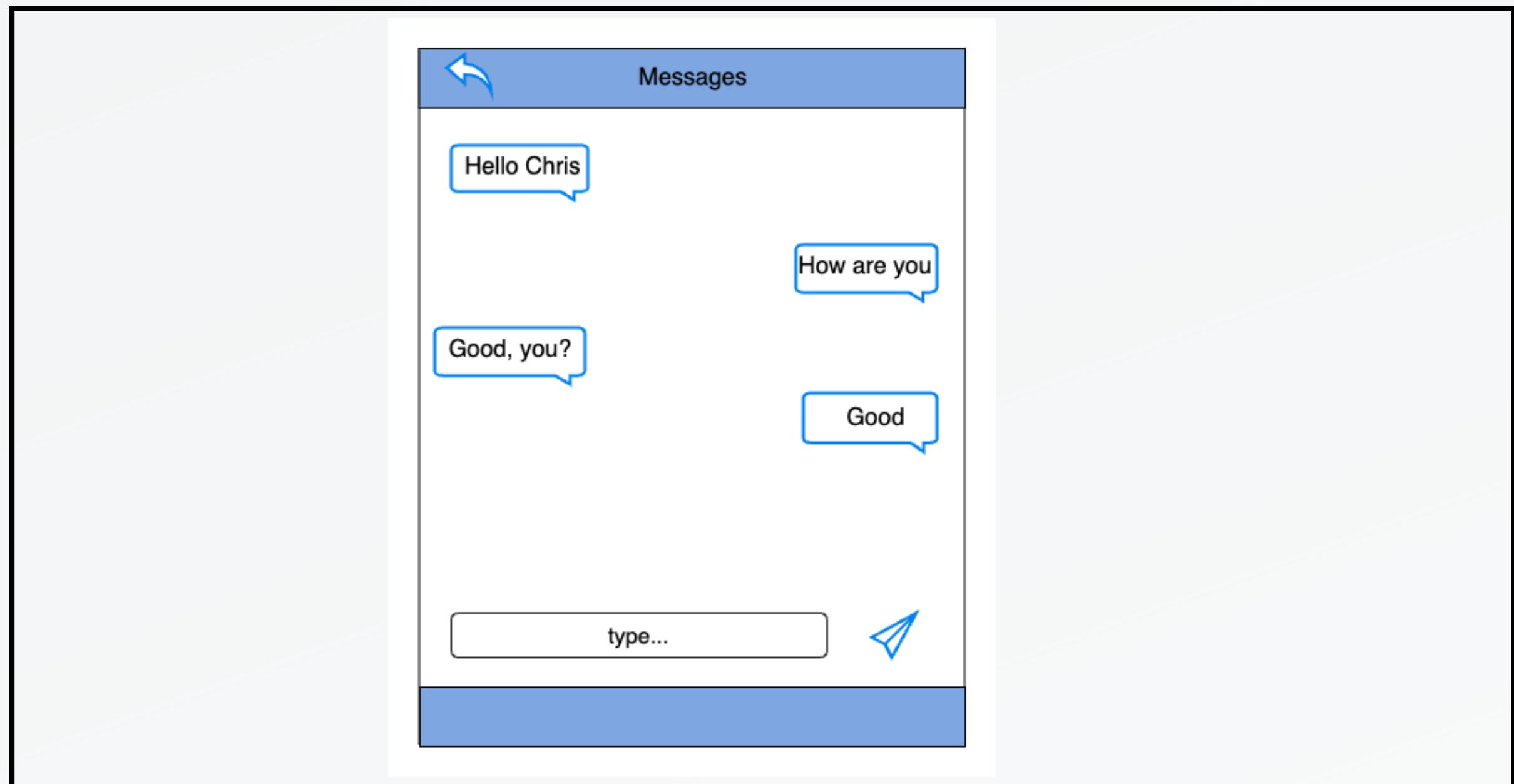
WIREFRAMES AND USER INTERFACES



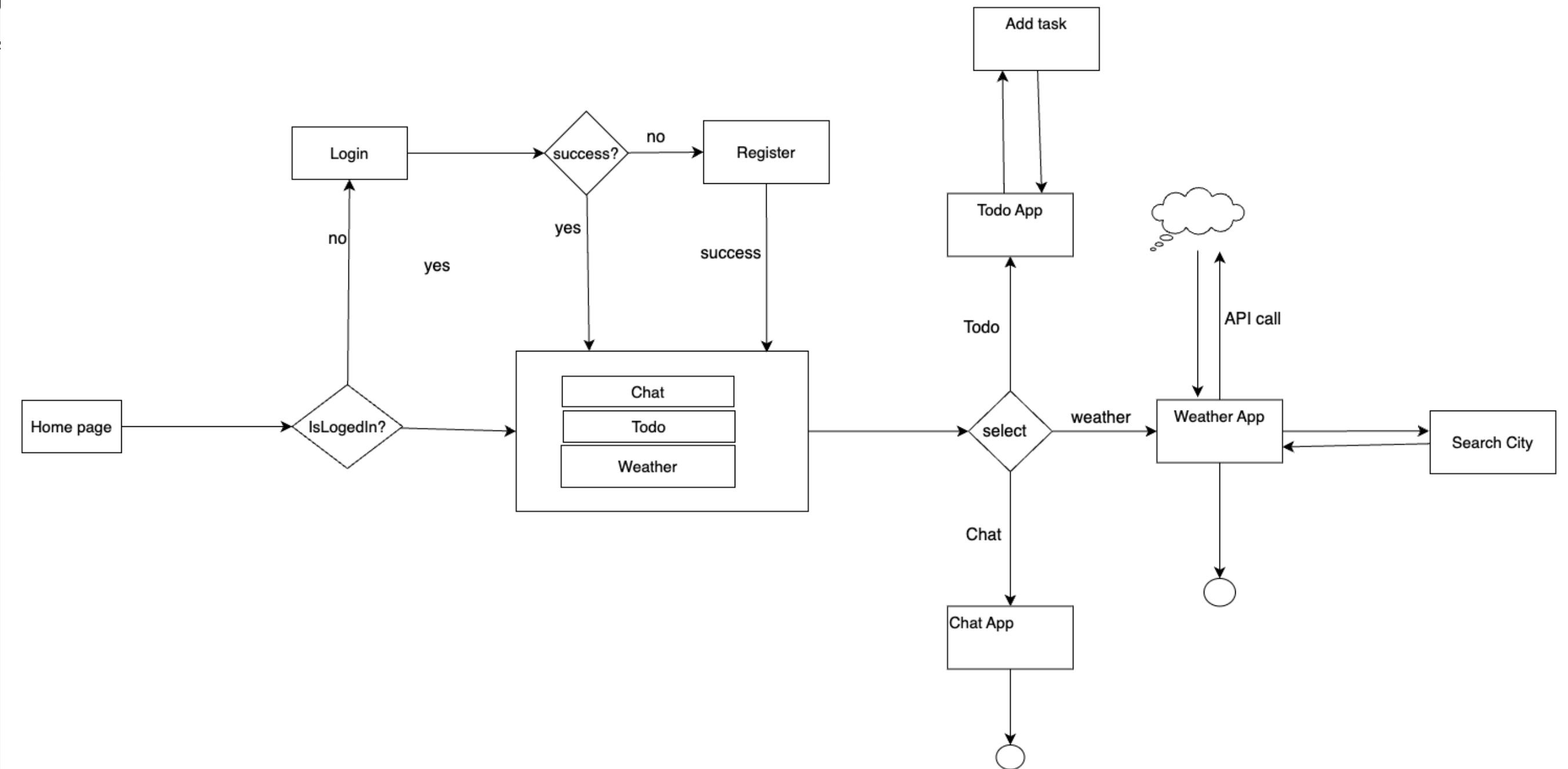
WIREFRAMES AND USER INTERFACES



WIREFRAMES AND USER INTERFACES



ACTIVITY DIAGRAM



KEY IMPLEMENTATION DETAILS

- **User Authentication:** SignInWithEmailAndPassword using Firebase.
- **ChatApp Real-Time Updates:** Using StreamBuilder to listen to Firestore updates.
- **Weather Data Fetching:** Handling HTTP requests to an external API using the http package.
- **Todo List Management:** Using Provider for state management and Firestore for data persistence.

TESTING STRATEGY

- **Unit Tests:** For utility functions and business logic.
- **Widget Tests:** For testing UI components like buttons, forms, and lists.
- **Integration Tests:** For complete user flows (login, chat, add/edit/delete tasks).

TESTING STRATEGY

Testing Frameworks

`flutter_test, integration_test, mockito, firebase_auth_mocks.`

- **Unit Tests:** For utility functions and business logic.
- **Widget Tests:** For testing UI components like buttons, forms, and lists.
- **Integration Tests:** For complete user flows (login, chat, add/edit/delete tasks).

GOAL ACHIEVEMENT

- **The app meets the core requirements:** Real-time chat, weather updates, and task management.
- The app complies with Google's Material Design guidelines.

CONCLUSION

- The project successfully developed a multi-functional mobile application that integrates ChatApp, Weather App, and Todo App functionalities, meeting most of the original goals.
- The use of Flutter and Firebase proved to be an effective combination for rapid and reliable development.
- The Agile Scrum methodology facilitated continuous improvement and successful project delivery.

AREA OF IMPROVEMENT

- **Enhanced User Experience:** Add more interactive elements, animations, and dark mode.
- In the todoApp, Strike-through done task instead of deleting them
- In Chat App, add the functionality to add picture and video.
- Add more weather and timezones to the weather app.

LESSONS LEARNED AND PERSONAL DEVELOPMENT

- Effective use of Scrum methodology in a mobile development project.
- Gained experience with Firebase integration and Flutter development.
- Learned to write more testable and maintainable code using MVVM architecture.
- Improved skills in mobile app development, backend integration, and state management.
- Enhanced problem-solving skills through iterative development and debugging.

REFERENCES

- Flutter. (n.d.). Flutter documentation. Flutter.dev. Retrieved September 5, 2024, from <https://docs.flutter.dev/>
- Dart. (n.d.). Dart documentation. Dart.dev. Retrieved September 5, 2024, <https://dart.dev/guides>
- Google. (n.d.). Firebase documentation. Firebase.google.com. Retrieved September 5, 2024, <https://firebase.google.com/docs>
- Google. (n.d.). Firebase Authentication. Firebase.google.com. Retrieved September 5, 2024, <https://firebase.google.com/docs/auth>
- Google. (n.d.). Cloud Firestore documentation. Firebase.google.com. Retrieved September 5, 2024, <https://firebase.google.com/docs/firestore>
- Google. (n.d.). Material Design guidelines. Material.io. Retrieved September 5, 2024, from <https://m3.material.io/>
- OpenWeather. (n.d.). OpenWeatherMap API documentation. Openweathermap.org. Retrieved September 5, 2024, <https://openweathermap.org/api>