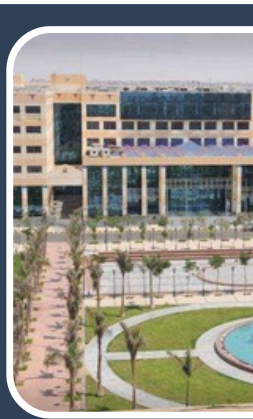
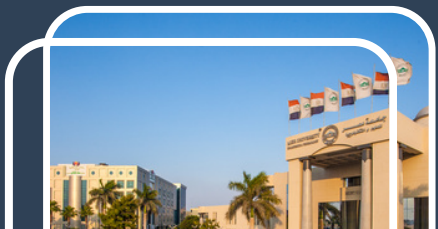
DEPI

U SEARCH

Rowan Nour
Sohaila Said
Ahmed Kabary
Menna Ossama





supervisor:

Esraa Fathy

Documentation

Project Overview

U Search is a mobile application built using Flutter and Firebase that helps prospective university students search for top-ranking universities, view university details, and mark favorites. The app uses Firebase for user authentication and an in-house REST API to fetch information about universities. It provides a user-friendly interface for exploring university options and will be expanded with more features in the future.

Technologies Used

1. Flutter

Flutter is a powerful framework that allows for fast development of cross-platform mobile applications. It provides:

- **Widgets**: Highly customizable UI components.
- **Hot Reload**: Instantly preview UI changes during development.
- **State Management**: Efficiently manages app state through tools like Provider and Bloc.

2. Firebase

Firebase provides backend services and tools to manage user authentication, data storage, and more. Specifically:

- **Firebase Authentication**: Allows users to log in, sign up, and manage profiles.
- **Firebase Firestore (planned)**: To store and retrieve user-specific data like favorites and custom lists.

3. REST API

We implemented a REST API to serve university data in JSON format. The API returns detailed information about universities, including their ranking, fees, location, and descriptions. This API is a core part of how the app fetches university information dynamically.

App Structure

1. Screens

- **TopRankingDetails**: Displays detailed information about a selected university, including its description, ranking, location, and fees. It allows users to add or remove universities from their favorites list.

- **Favorite**: Shows a list of universities that the user has favorited, enabling quick access to the universities they are most interested in.
- **UniMatch**: A screen where users can filter universities based on criteria such as location and fees, using drop-down menus to narrow down results.
- **Profile**: Displays user profile details, allowing them to view and edit their personal information. The screen also features a log-out button to sign out from the app.

2. Widgets

Widgets are essential components in the app for building UI elements:

- **BigText**: A reusable widget used to display large, prominent text throughout the app. It customizes size and color for different contexts.
- **SmallText**: A widget for rendering smaller text, often used for descriptions and subtler labels.
- **AppIcon**: A reusable icon widget that ensures a consistent style across the app for icons like "favorite" or "back."
- **CustomButton**: A customizable button widget used across various screens for actions like submitting or logging out.

3. Utils

Utility classes and functions enhance code reusability and maintainability:

- **Colors (AppColors)**: A utility class that defines the color palette for the app, making it easier to maintain a consistent visual design.
- **Dimensions**: A utility class that defines standard sizes and padding for widgets, ensuring consistent spacing and layout across the application.

Backend Services

Firebase Authentication

- **User Management**: Firebase is used to handle user login, signup, and session management. Users are authenticated using their email and password, allowing them to save preferences like favorited universities.

REST API for University Information

- **API Structure**: The API provides data about universities in JSON format. Below is an example of a response from the API:

json

Copy code

```
{
 "total_size": 5,
 "type_id": 2,
 "offset": 0,
 "universities": [
  {
   "id": 1,
   "name": "Egypt-Japan University of Science and Technology (E-JUST)",
   "description": "E-JUST is a public university in Alexandria, Egypt...",
   "fees": 7500,
   "img": "assets/images/img_4.png",
   "location": "Alexandria, Egypt"
  },
  ...
 ]
}
```


This API is integrated into the app using Flutter's http package, which allows seamless fetching and updating of university data.

State Management

1. Provider

We used the **Provider** package for managing state across the app, particularly for managing the list of favorite universities. FavoritesProvider is responsible for adding and removing universities from the user's favorites list.

2. Bloc (UniCubit)

For managing the logic behind fetching and filtering university data, we use the **Bloc (Business Logic Component)** pattern, specifically using a **Cubit** for a simplified approach. The UniCubit fetches universities from the API and provides functionality for filtering results based on user input, like fees and location.

Core Features

1. **Top-Ranking Universities**:
    - Displays a list of the top-ranked universities with details like their ranking, location, and fees.
    - Users can view detailed descriptions, including images and educational offerings.
2. **Favorites List**:
    - Users can add universities to their favorites list for easy access later.
    - The list is stored locally using the state management system (and Firebase Firestore in future plans).
3. **Profile Management**:
    - Users can view and edit their profiles, including changing their names, email addresses, and other personal information.
    - Users can also log out of their accounts via Firebase Authentication.
4. **Uni Match Feature**:
    - Users can filter universities by location and tuition fees, using a simple form interface that interacts with the REST API.

Future Plans

1. **Expanded Dataset**: We plan to expand the dataset to include more universities worldwide, giving users access to an even broader selection of institutions.
2. **In-app Communities**: We aim to create communities for each university, allowing students to interact, share experiences, and support one another within the app.
3. **Improved Personalization**: By storing user preferences and data in Firestore, we will provide a more personalized experience where users can track their interests and updates on universities.

Screens

Home Screen

Purpose:

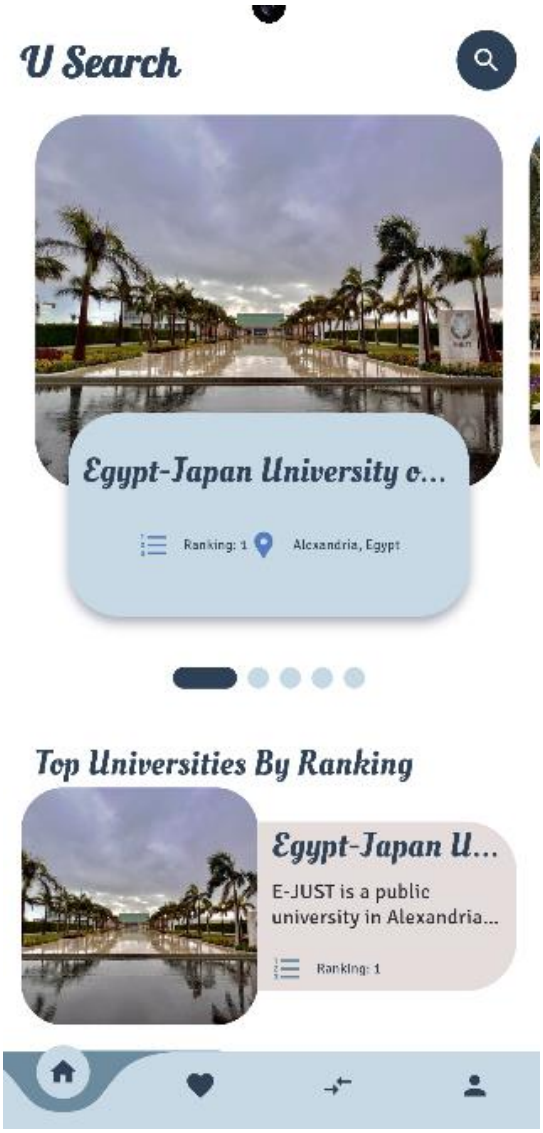
The Home Screen is the landing page of U Search. It displays a list of top-ranking universities for users to explore. It also provides navigation to other key features, such as Uni Match and Favorites.

Features:

- **Top Universities List**: Displays a list of universities fetched from the API, including their name, location, and fees.
- **Navigation**: Users can navigate to the "Top Ranking Details" screen by tapping on a university or explore other sections like "Favorites" and "Uni Match."
- **Search Functionality**: A search bar allows users to quickly search for specific universities by name.

Widgets Used:

- BigText: Displays the heading of the screen.
- SmallText: Used for brief descriptions.
- ListView.builder: Dynamically generates the list of universities.
- AppIcon: Customizes icons for navigation.



TopRankingDetails Screen

Purpose:

The TopRankingDetails screen provides detailed information about a selected university. Users can read a university's description, view its location, fees, and mark it as a favorite.

Features:

- **University Details**: Displays university name, location, fees, and a detailed description.
- **Favorite Feature**: Users can mark the university as a favorite using a heart icon.
- **Dynamic Data**: Data is fetched from the REST API and displayed dynamically.

Widgets Used:

- BigText and SmallText: For heading and description display.
- IconButton: For marking a university as a favorite.
- Image.asset: To display university images fetched from the API.

Favorites Screen

Purpose:

The Favorites screen is where users can view the universities they have marked as favorites. This feature allows them to easily revisit their preferred universities.

Features:

- **Favorite Universities List**: Displays all universities that the user has added to their favorites.
- **Remove from Favorites**: Users can remove universities from the list if they change their mind.
- **Persistent State**: Favorites are managed using Provider, ensuring that the state remains consistent even as users navigate away from the app.

Widgets Used:

- ListView.builder: To generate the list of favorite universities.
- IconButton: To remove a university from favorites.

- AlertDialog: Confirms the removal of universities from favorites.

UniMatch Screen

Purpose:

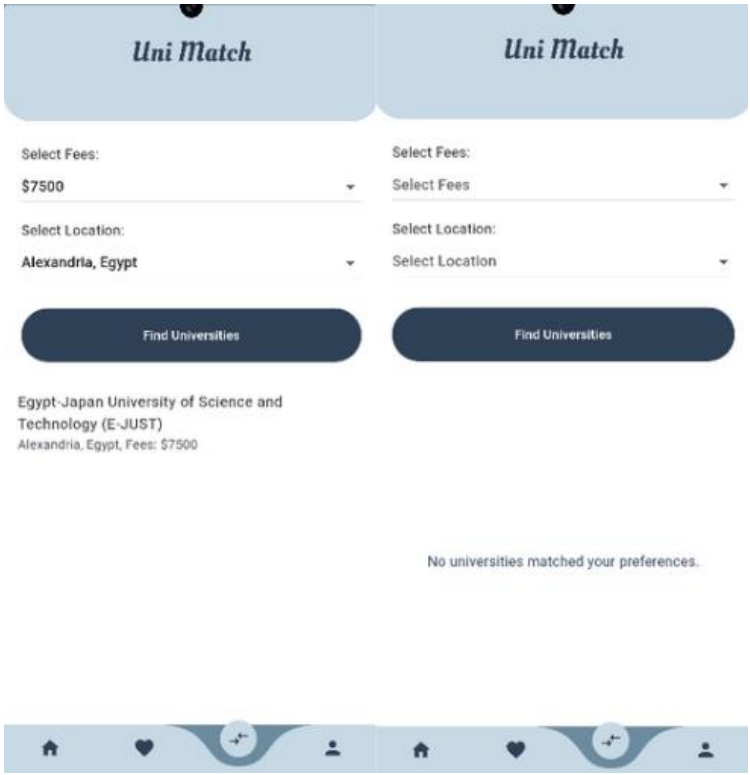
The UniMatch screen allows users to filter universities based on their preferences, such as fees and location, and find the best match for their needs.

Features:

- **Filters**: Users can select from dropdown menus to filter universities by fees and location.
- **Dynamic Search**: The list of universities updates dynamically based on the user's selections.
- **Results Display**: Displays filtered results, showing only universities that match the selected criteria.

Widgets Used:

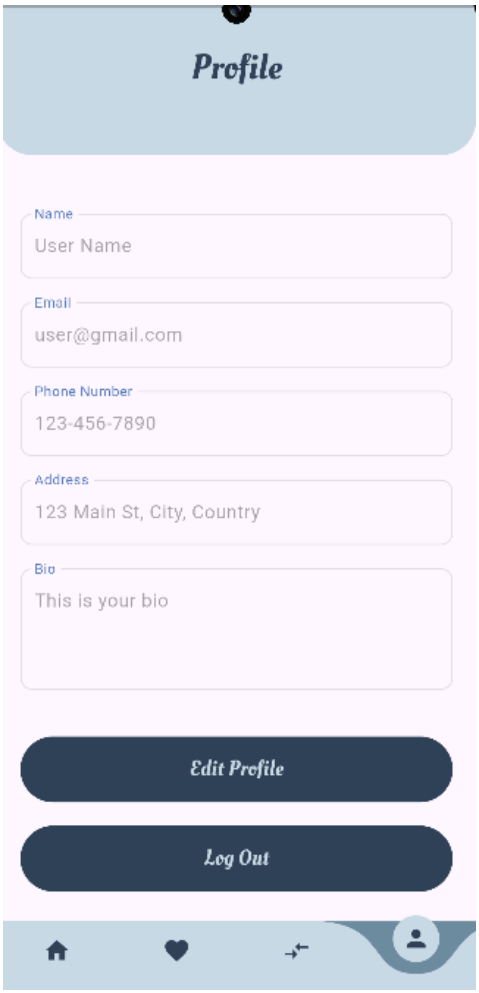
- DropdownButton: For selecting fees and location.
- BlocBuilder: To manage state and display the list of universities based on user filters.
- ListView.builder: To display filtered results.



Profile Screen

Purpose:

The Profile screen displays and allows users to edit their personal information, such as name, email, phone number, address, and bio. Users can also log out from the profile screen.



Features:

- **Edit Profile**: Users can toggle between viewing and editing their profile information.
- **Profile Fields**: Fields include name, email, phone number, address, and bio.
- **Logout**: A button allows users to log out securely.
- **Dynamic Data**: The user's information is fetched from Firebase Authentication and displayed in the relevant fields.

Widgets Used:

- TextField: For editing user details.
- CustomButton: For saving profile changes and logging out.
- FirebaseAuth: Integrated to handle authentication and logout functionality.

Log In/Sign Up Screen (Authentication)

Purpose:

This screen allows users to either log in to their existing account or sign up for a new account. It's integrated with Firebase Authentication for secure user management.

Features:

- **Login/Sign Up Form**: A form that accepts email and password input.
- **Firebase Authentication**: Authenticates users and stores their session.
- **Error Handling**: Displays appropriate error messages if login or sign-up fails.

Widgets Used:

- TextField: For capturing email and password input.
- ElevatedButton: To trigger login or sign-up actions.
- FirebaseAuth: For handling user authentication.
- SnackBar: For displaying error or success messages.

- 

Conclusion

U Search offers a streamlined and interactive way for students to explore universities and manage their preferences. By leveraging modern technologies like Flutter, Firebase, and a custom API, we built an app that can grow over time, providing even more features and a richer experience.