Project Proposal

notes.ai

Faculty of Science, Ontario Tech University

CSCI 4100: Mobile Devices

Michael Miljanovic

Oct 20, 2023

Project Description

Our project consists of an AI-powered note-taking mobile application that allows users to quickly write down their thoughts and save them on their phone, which will sync across all devices that have the app.

The application will interface with a server to provide AI-powered features such as autocomplete, OCR and automated sorting. We are also leveraging the power of AI to introduce features that will make the application more accessible for users with impairments and or disabilities.

A cloud-based database will be available for users to load and sync their notes across all devices. In order to sync the local SQLite database to the online database, users must sign in or create a new profile.

To develop such a sophisticated application that is also reliable, notes.ai will leverage services and platforms such as Google Firebase and Microsoft Azure to ensure the security of user data and the reliability of services from the server, respectively.

The goal of our project is to build on typical features of note-taking applications by improving on areas that are not only conventional to a note-taking app but also include new features, specifically with the transformative power of AI. Overall, the team aims to deliver a note-taking experience that is intuitive and seamless.

Name	Responsibility
Aaveg Shangari	Aaveg will be working on the development and implementation of AI features and general backend functionality.
David Houle-Tymeczko	David will be responsible for overseeing the entire spectrum of notes.ai's technological stack, with a special emphasis on the front-end and back-end.
Royce Mathew	Royce will be responsible for overseeing the entire spectrum of notes.ai's technological stack, with a special emphasis on databases.
Sheida Ebrahimi Siaghi	Sheida will be the principal front-end developer and will oversee general front-end functionality.
Srivathsan Sivakumar	Sri will be responsible for overseeing the entire spectrum of notes.ai's technological stack, with a special emphasis on AI.

Features and Functionalities

The features and functionalities of notes.ai are primarily divided into five sections based on what each suite of technology aims to achieve:

1. **Artificial Intelligence-based Augmentations**
    a. **Autocomplete Text**
    b. **Categorization**
    c. **Optical Character Recognition**
2. **Text Formatting and Input**
    a. **Markdown Support**
    b. **Writing/Drawing Support**
3. **Data Storage and Synchronization**
    a. **Cloud Storage**
    b. **Local Storage**
4. **Accessibility:**
    a. **Text to Speech**
    b. **Speech to Text**
5. **Algorithms**
    a. **Search**

1.a Autocomplete Text: The app will feature dynamic text completion. This allows users to write notes faster and spend less time writing notes character by character.

1.b Categorization: The content of the note is read by a fine-tuned AI model, which then recommends the most apt category to put the notes in. If the category does not exist already, a new one is created.

1.c Optical Character Recognition: Our application provides users with the capability to upload images, enabling it to perform text extraction through Optical Character Recognition (OCR).

2.a Markdown Support: The app offers robust support for Markdown formatting, enabling users to effortlessly apply various formatting elements to their notes using plain text. With this feature, users can easily incorporate bullet points, lists, headers, and more, all while maintaining a simple and intuitive editing experience.

2.b Writing / Drawing Support: The app allows users to draw on notes using a stylus, finger, or mouse. This can be useful because it allows users to port their notes to a drawing tablet.

3.a Cloud Storage: The app allows users to sync notes to the cloud and across multiple devices and access them from anywhere. Users will be notified when the cloud is synced.

3.b Local Storage: By employing an SQLite database on the user's device, the app maintains persistent local storage of the user's notes.

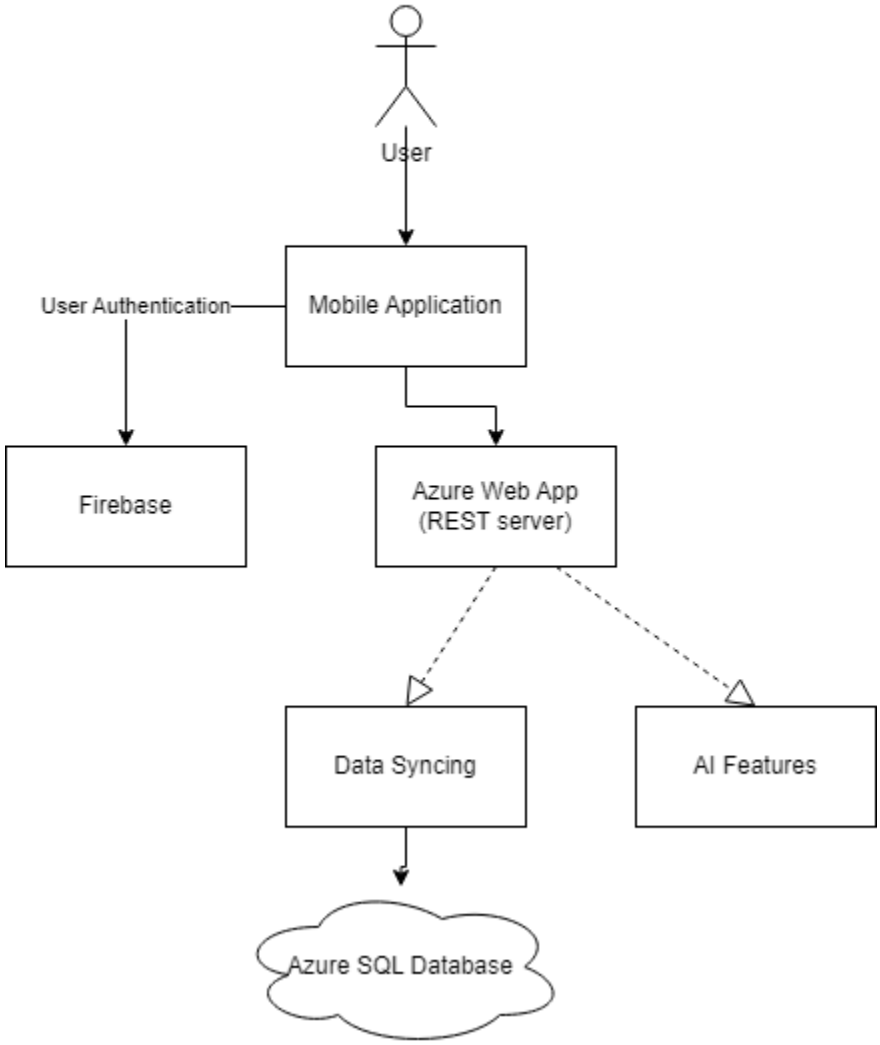
4.a Text to Speech: Users will be able to listen to the content of their notes using this functionality, allowing greater accessibility and convenience

4.b Speech to Text: Users will be able to take notes using speech-to-text, allowing greater accessibility and convenience.

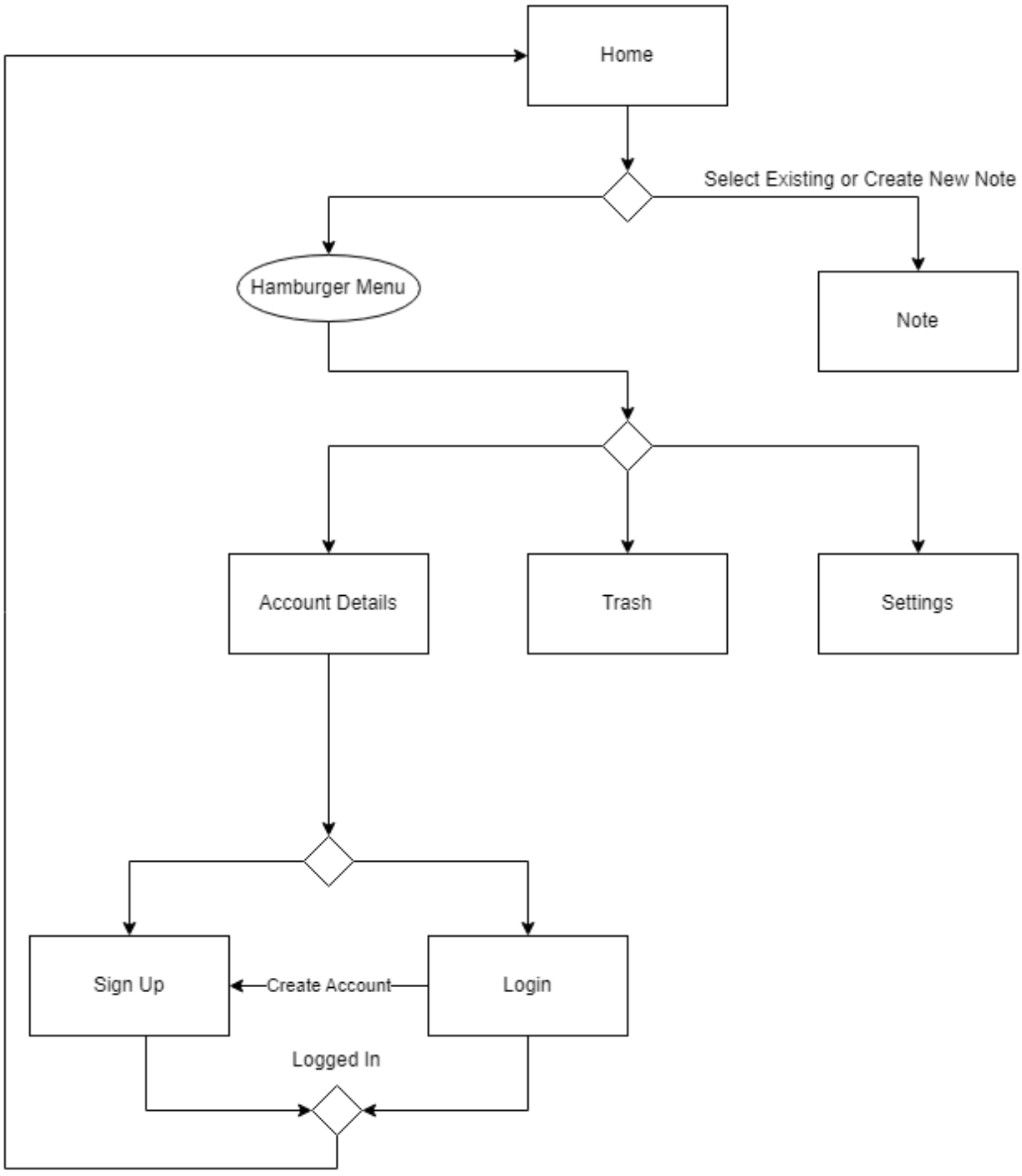
5.a Search: Users can search for their notes by entering keywords found in the titles or within the note contents. Additionally, users can tag their notes with relevant labels, making it easier to categorize and sort their content. With the search function, users can apply tag-based filters, making the search easier.

Code Design

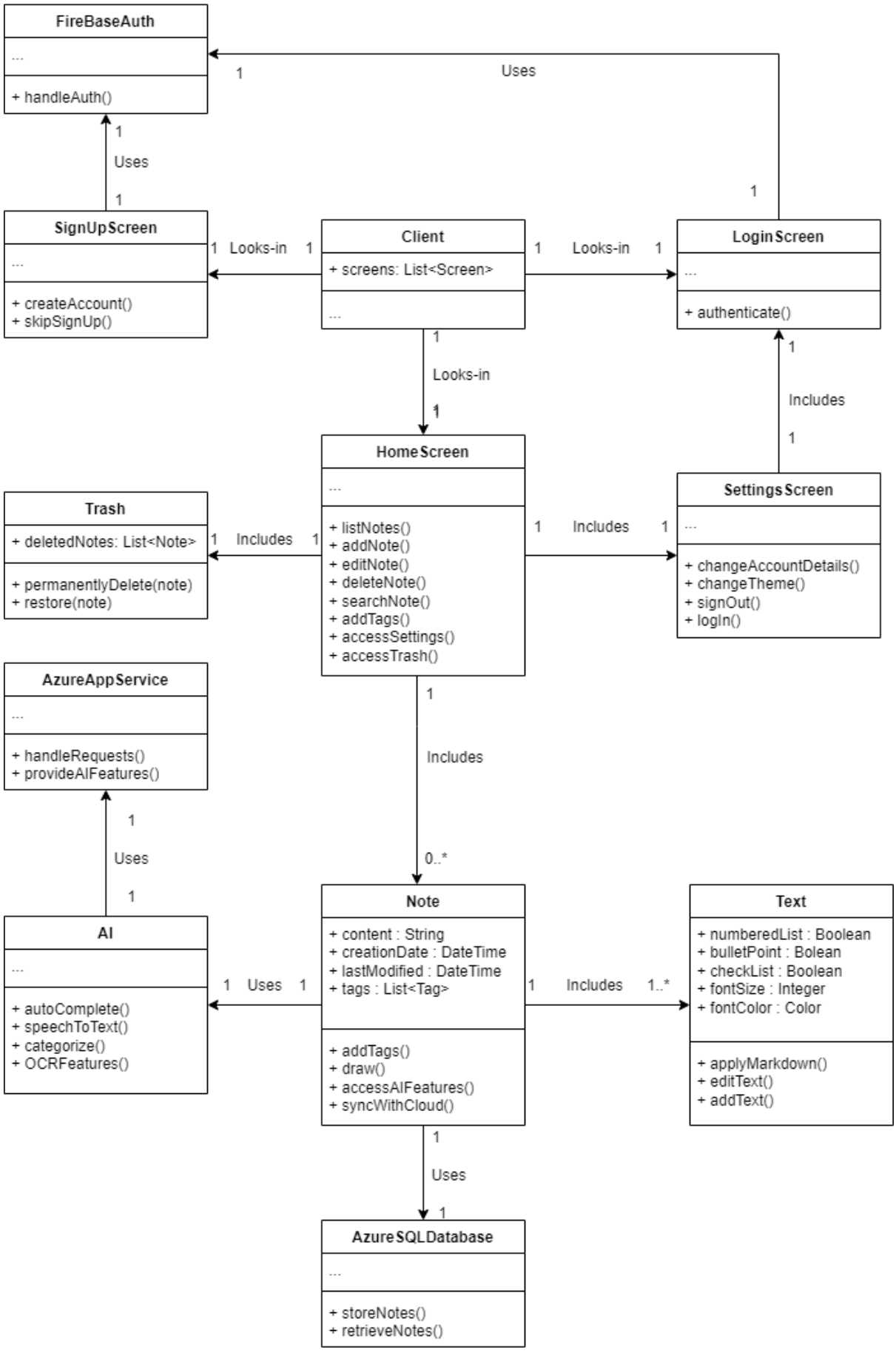
High-level Overview



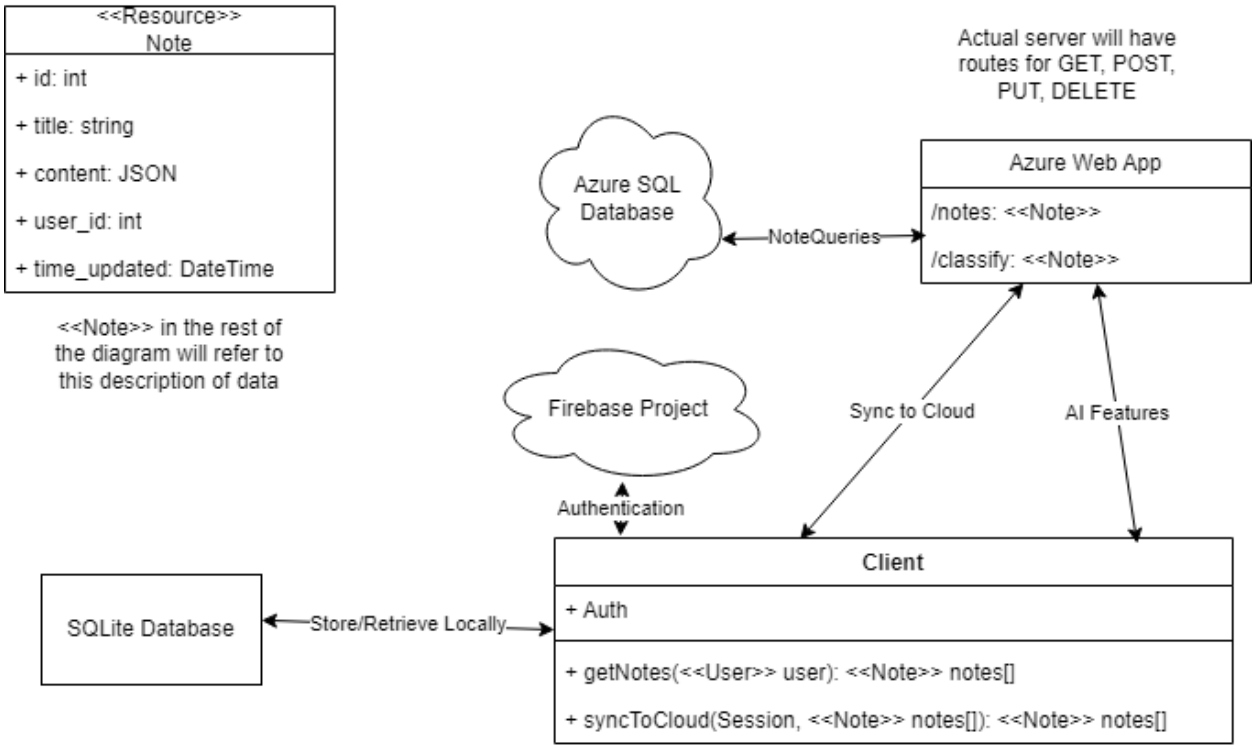
Screens Navigation

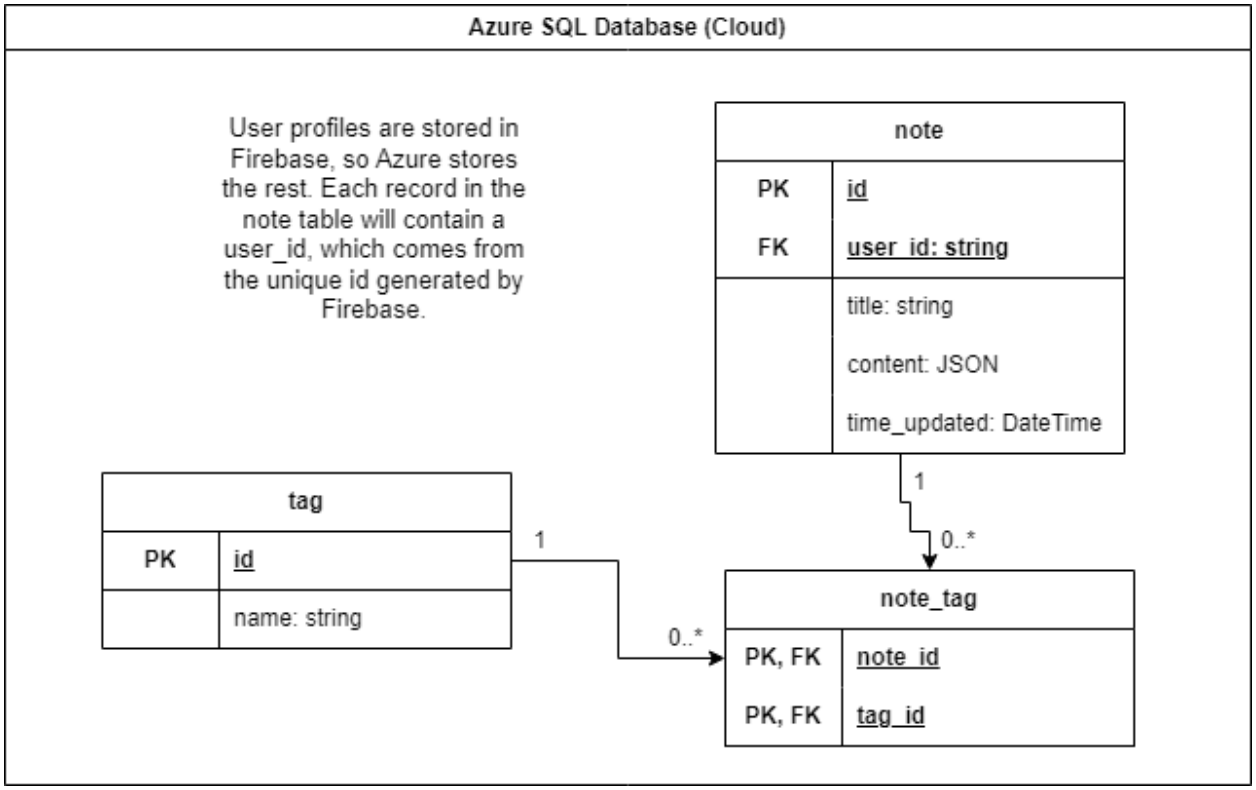


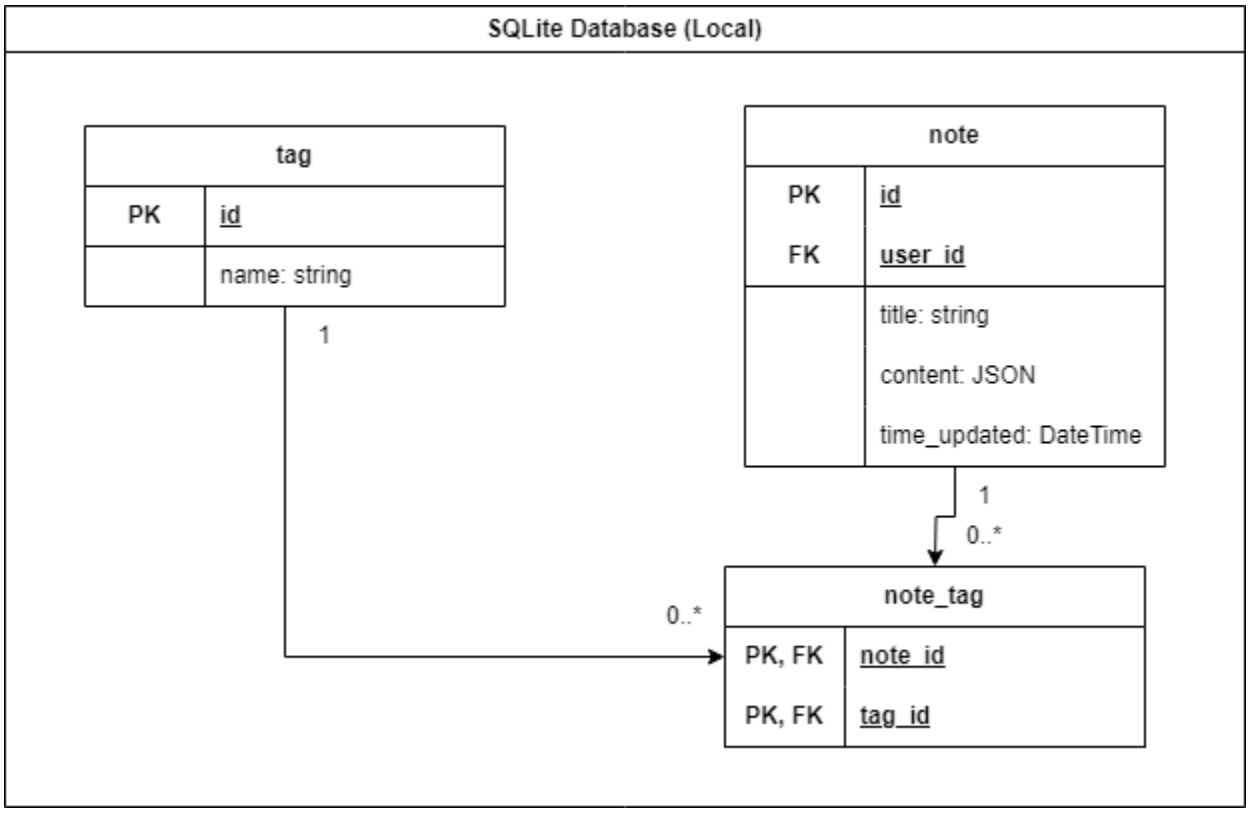
Design Class Diagram



Server Design

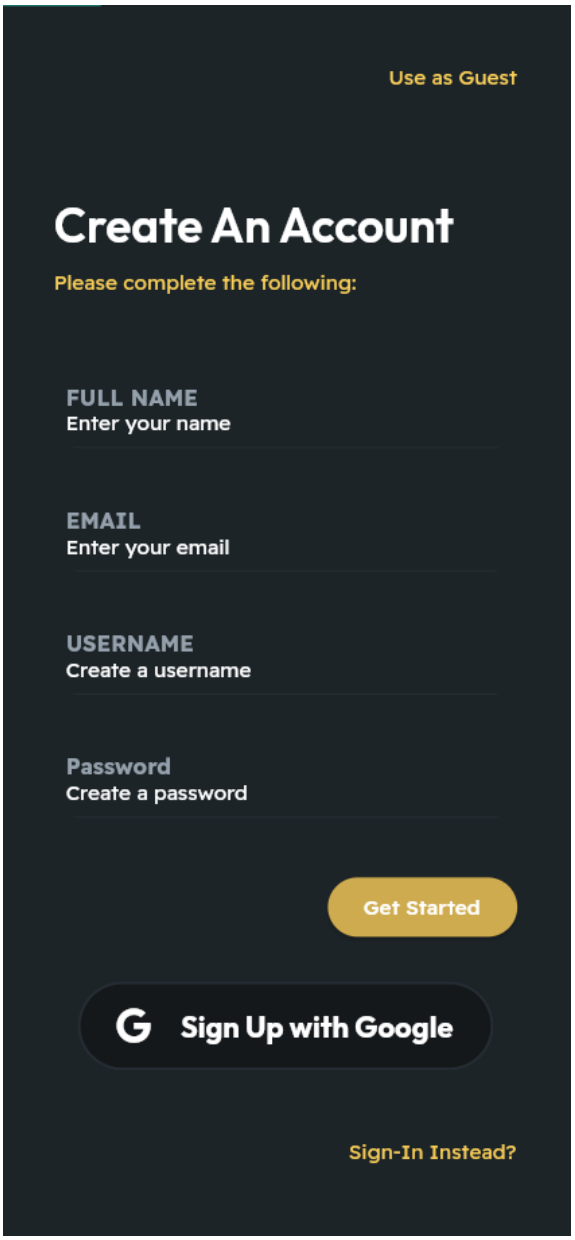



Azure SQL Database (Cloud)
User profiles are stored in Firebase, so Azure stores the rest. Each record in the note table will contain a user_id, which comes from the unique id generated by Firebase.
note
PK
id
FK
user_id: string
title: string
content: JSON
time_updated: DateTime
1
0..*
tag
PK
id
name: string
1
0..*
note_tag
PK, FK
note_id
PK, FK
tag_id

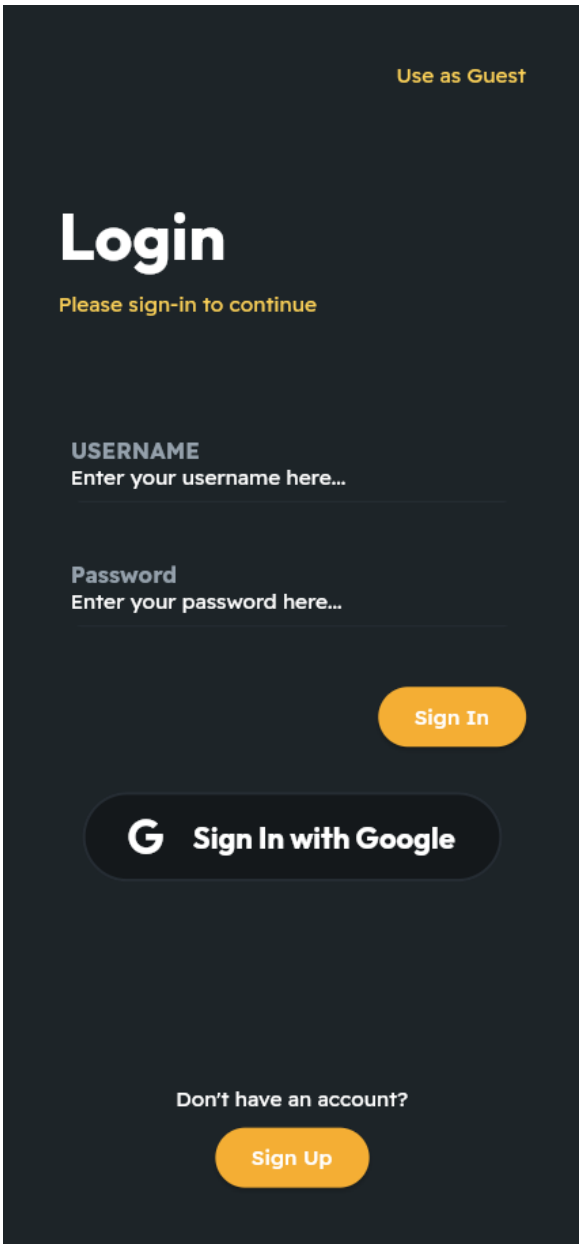

SQLite Database (Local)
tag
PK
id
name: string
1
note
PK
id
FK
user_id
title: string
content: JSON
time_updated: DateTime
1
0..*
0..*
note_tag
PK, FK
note_id
PK, FK
tag_id

Mock-Up of User Interface

Sign Up Page

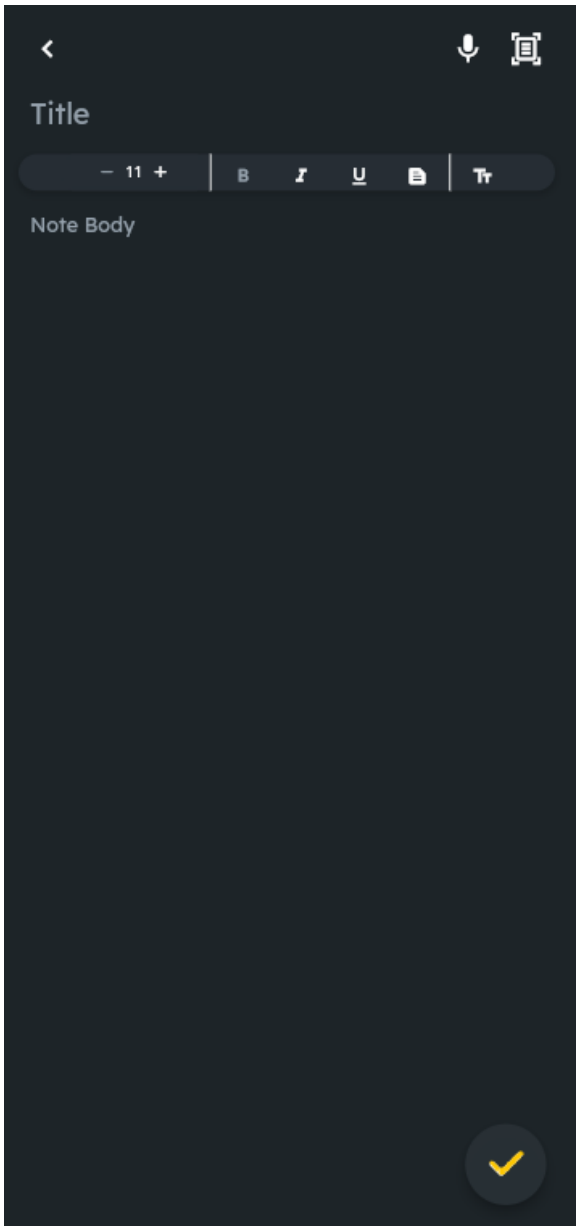


Sign In Page



Home Page

Create Note



Settings Page