Tutor Me

By: Caps On

RETRO RABBIT

Welcome to our DEMO 2 !

High Level Definition

The Tutor Me project is an idea that aims to provide a platform where students that need help academically can easily find a tutor that can help them with specific modules. This project will be useful especially now that the covid-19 pandemic has made it much difficult to have face-to-face conversations, ask for help with their studies from fellow students, and interact with lecturers for additional assistance. As with the rest of the world, every matter related to students is moving towards being digital and virtual.and connect it to your audience.

Architectural design strategy

We will be making use of the Decomposition design strategy which has to do with breaking down a complex system into smaller and much more manageable chunks.

Architectural Development

Architectural design strategy

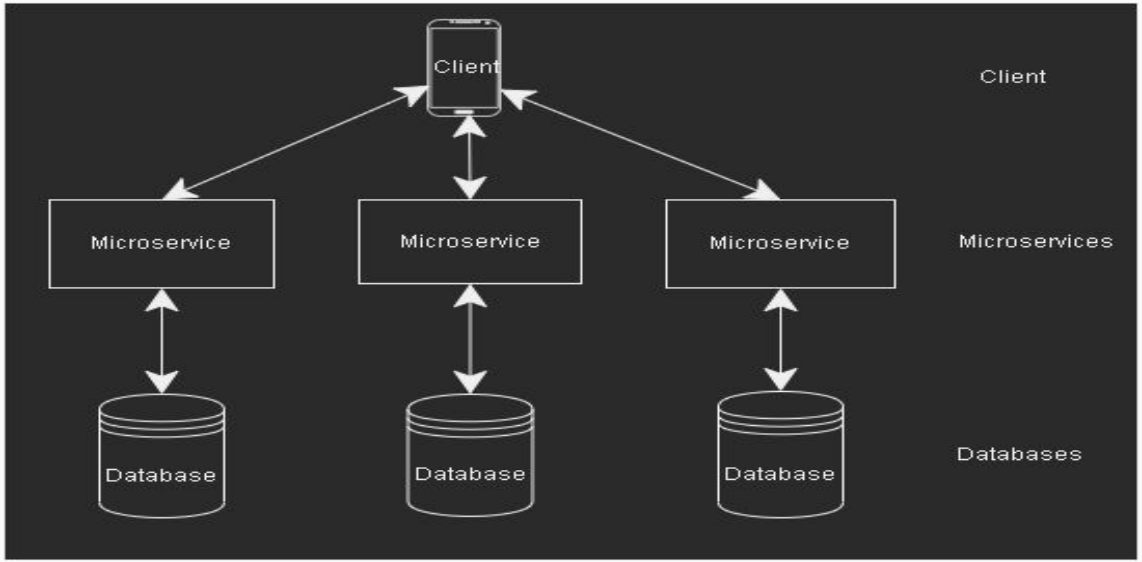
- **WHY?** System is well-defined and it is tightly-coupled which will allow us to break it down into granular subsystems, develop them and then integrate the various subsystems back into the whole system.
- Namely: Tutee and Tutor Subsystems

Architectural design strategy

- **then further broke down each of those into**
  - **video call,**
  - **chat functionality,**
  - **profile management and**
  - **connecting Tutors with Tutees .**

Architectural styles

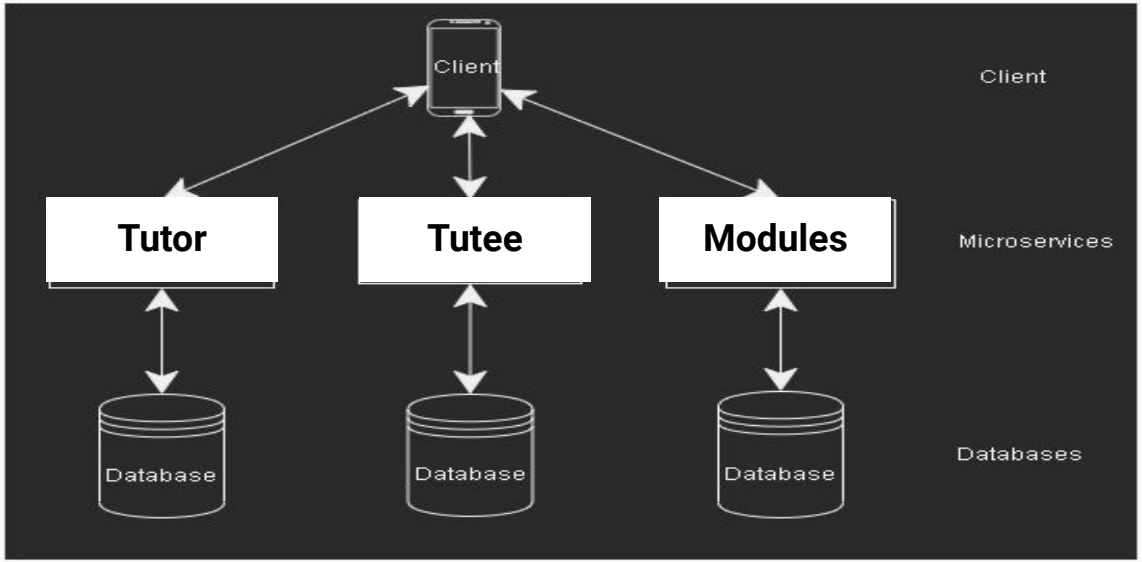
We will be making use of a Microservices Architecture.



Architectural styles

We will be making use of a Microservices Architecture.

Institution



Architectural quality requirements

- Availability - 99% of the time
- Security - All passwords ,All credentials kept safe
- Maintainability - Microservices, structured code
- Scalability - All enrolled students should be catered for.
- Serviceability - Code can easily be improved or fixed

Architectural constraints

- The app should be able to work on all android devices running Android 4.1 and newer. Iphone users will need to update to iOS 12. The technologies that we use should be compatible with these devices (which they are).
- The app should be as fault-tolerant as possible. The nature of the microservice architecture helps in this regard by making each service as independent to the others as possible.
- The system should make use of open source solutions for accomplishing most (if not all) tasks.

Technology choices

- Client - Flutter and Dart
- Microservices - ASP.NET Core
- Microservices - SignalR
- Microservices - VideoSDK
- Database - Microsoft SQL Server

(Pros Cons and Justifications in PDF doc)

Use Cases :

1. **Search**
2. **Profile Update and document uploading**
3. **Filter**
4. **Selection and updating of Modules**
5. **Status view: Online/Offline**

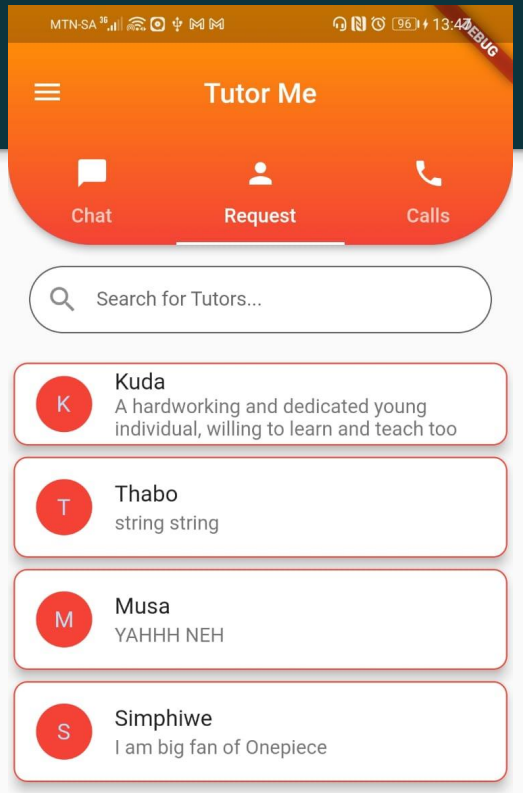
Use Case 1: Search

- Users want to be able

  To search from available lists

  Such as search for specific tutors

  By name in the suggested Tutors tab



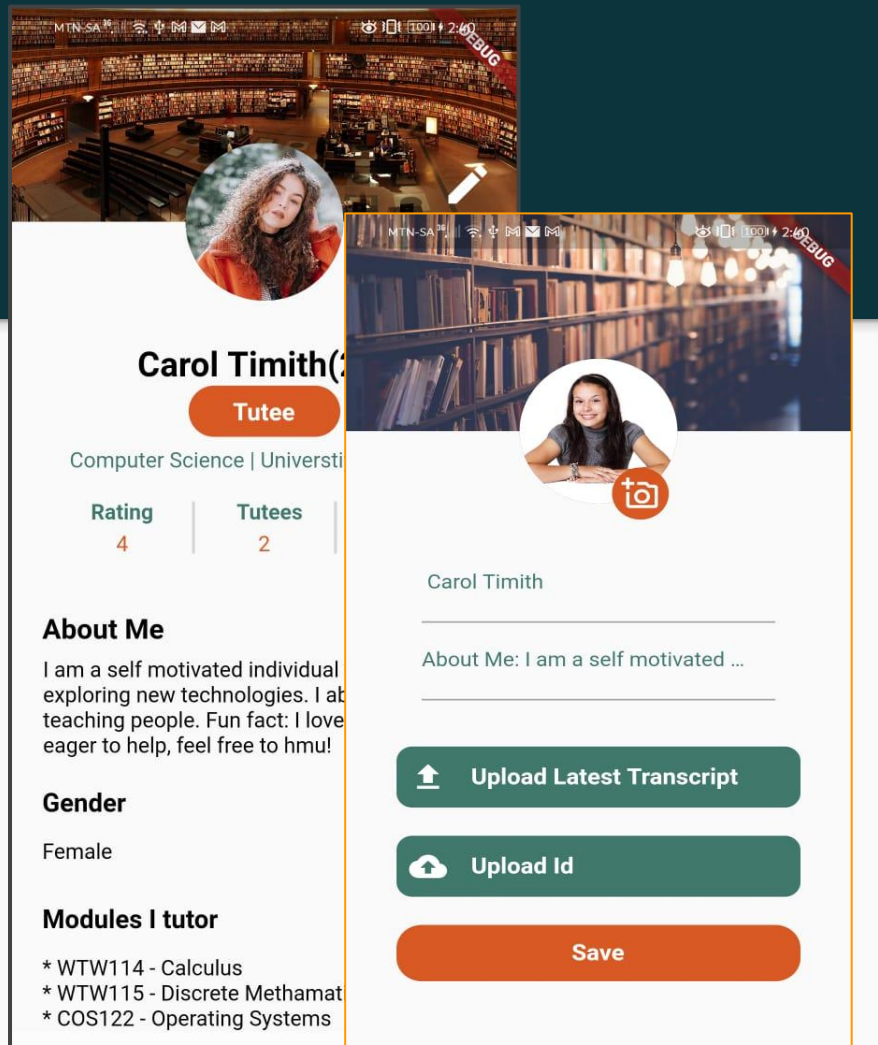
Use Case 2: Profile Update

- Users want to be able

  To update their public profiles

  This includes profile image,

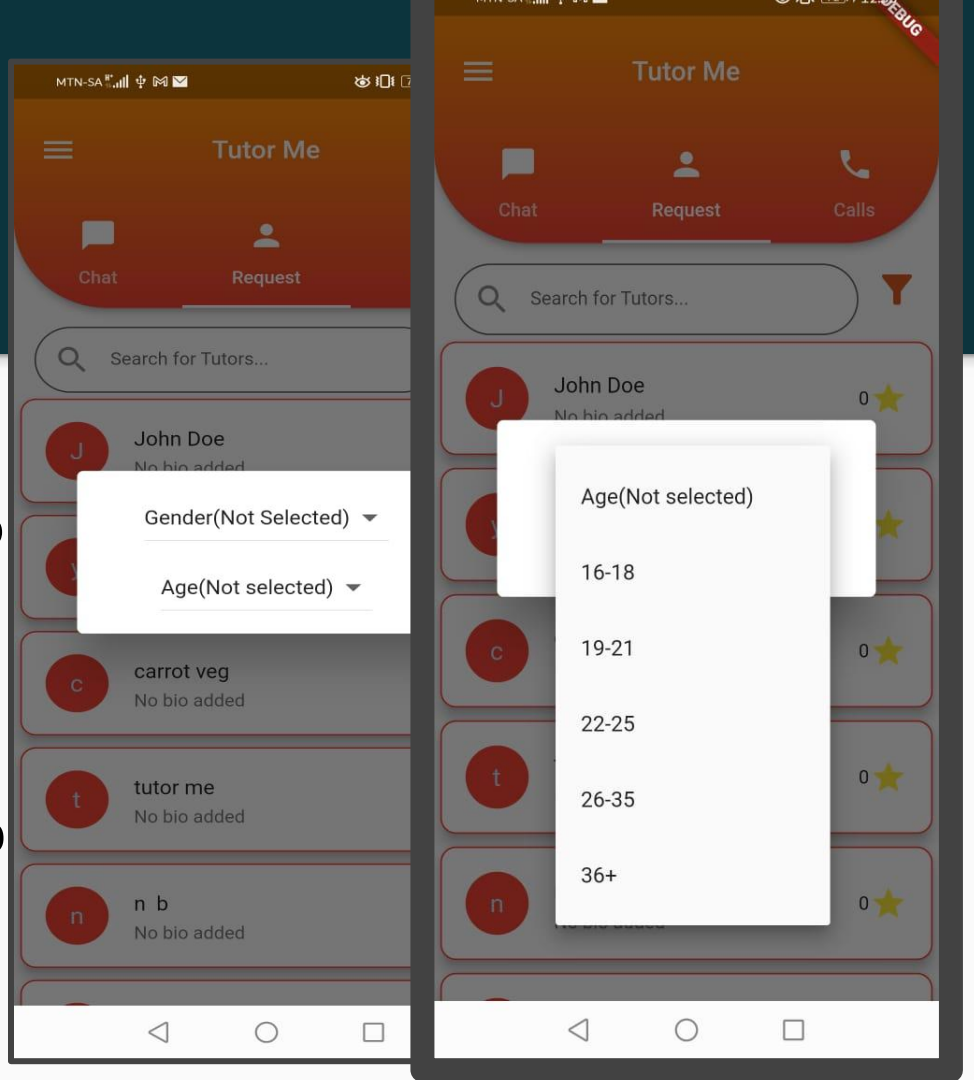
  Name, And Bio.



Use Case 3: Filter

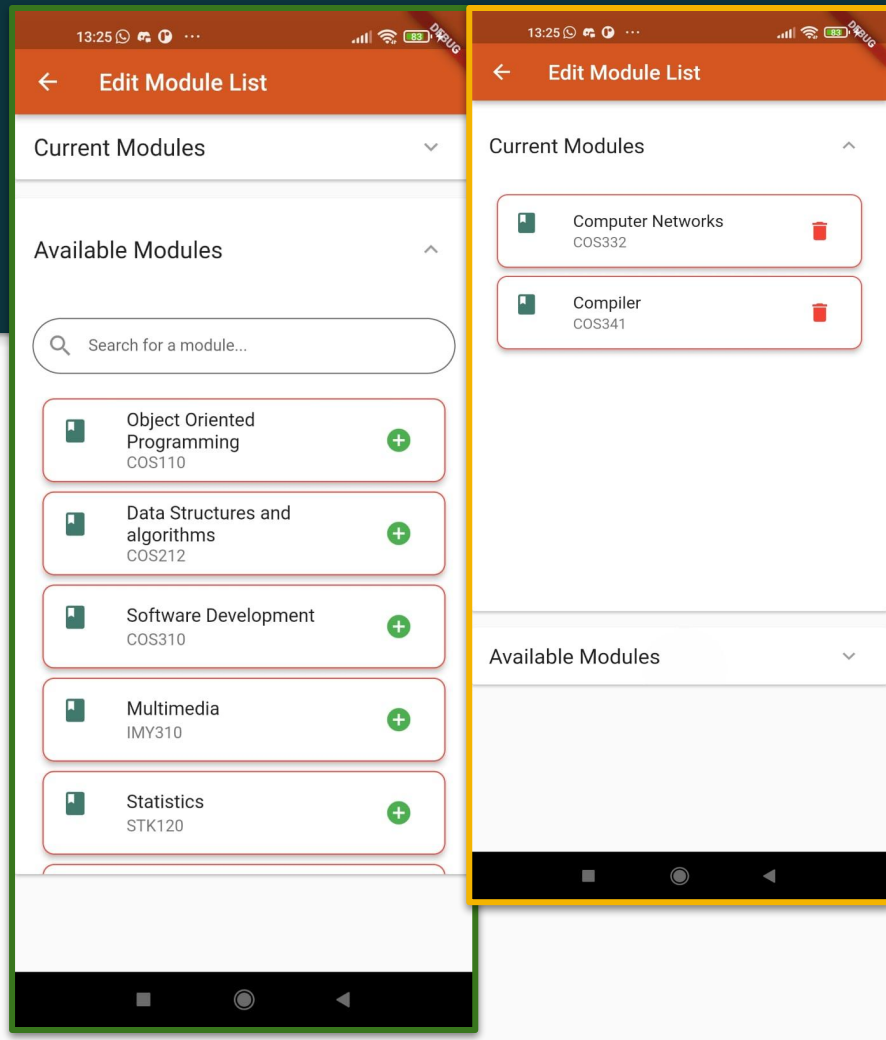
- Users want to be able To filter out the listed or suggested tuto by age and gender.

- This is to allow for a comfortab learning environment



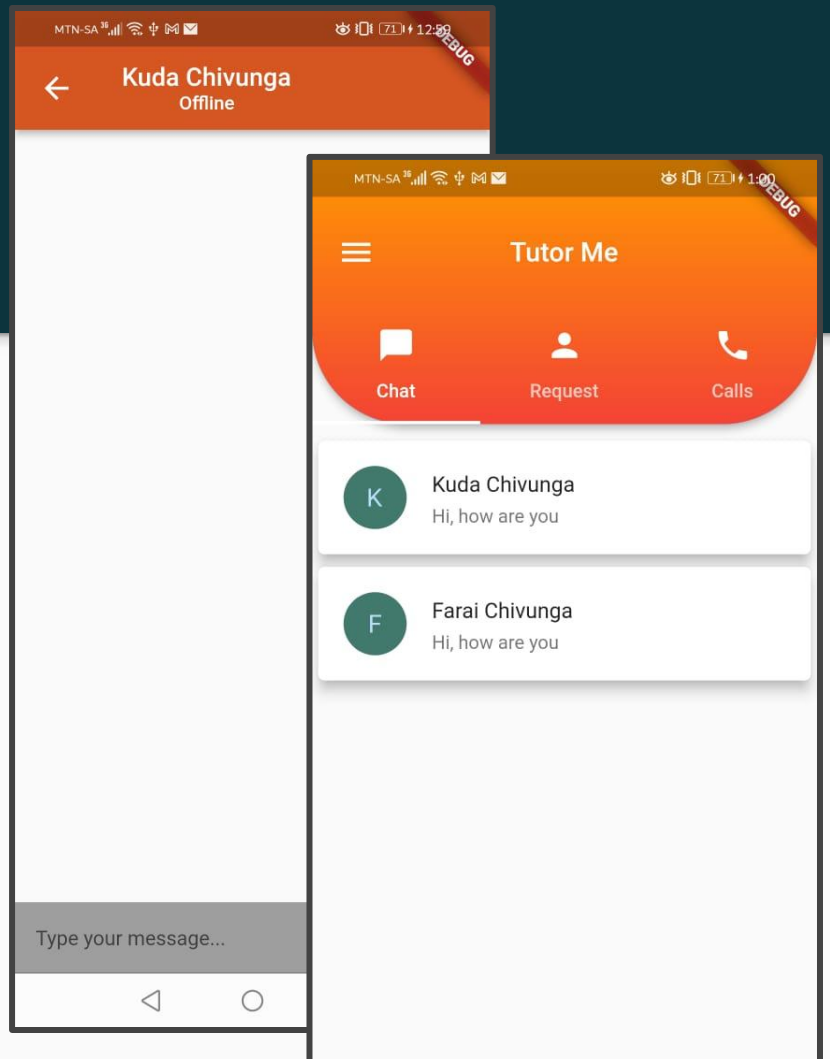
Use Case 4: Selection And Updating of modules

- Users want to be able To select modules they want tutoring for as well as modules they want to tutor for
- Users want to be able to change their modules selection by either adding more modules or deleting modules



Use Case 5: Status view: Online/Offline

- Users want to be able To see their contact's statuses, whether they are online or offline
- This use case will aid in completing a bigger use case, which is the ability to send and receive messages.



Video demo consisting of a minimum of 5 working use cases : **Live Demo**

- Search Functionality
- Filter
- Tutor Profile Updating
- Select Modules to be Tutored
- Online/Offline

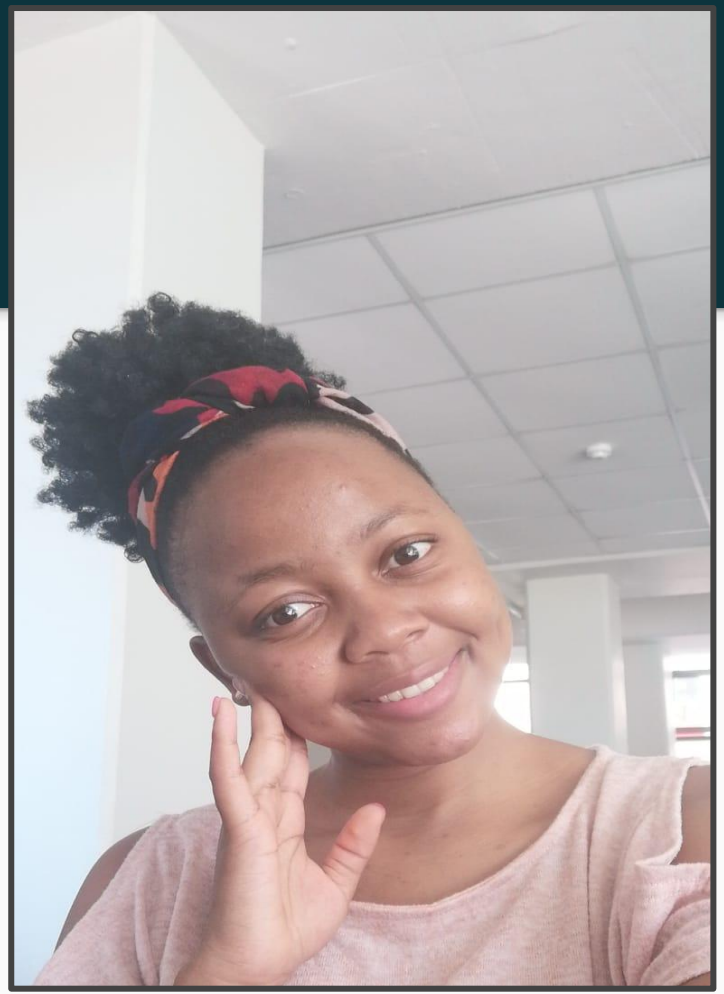
Unit Testing : Demo

Integration testing and test percentage coverage

Kudakwashe Chivunga

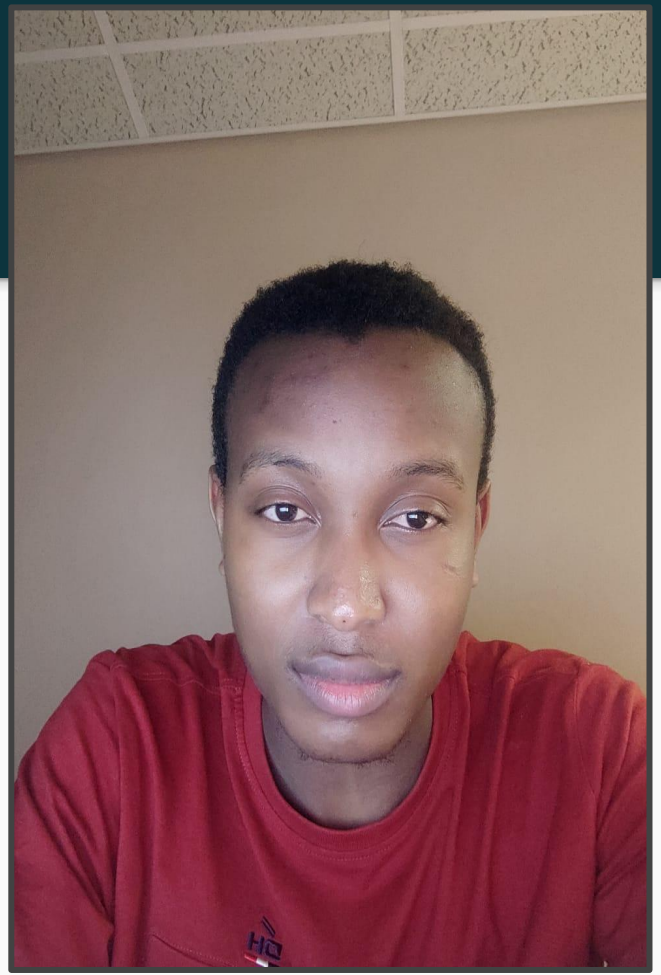
- Team Lead
  (GitHub Issues, GitHub Project board,
  scheduling Of meetings)

- Front-end Developer
  (figma, Flutter(Dart))



Musa Mabasa

- Front-end Developer (figma, Flutter(Dart))
- Backend to Front-end Integration



Farai Chivunga

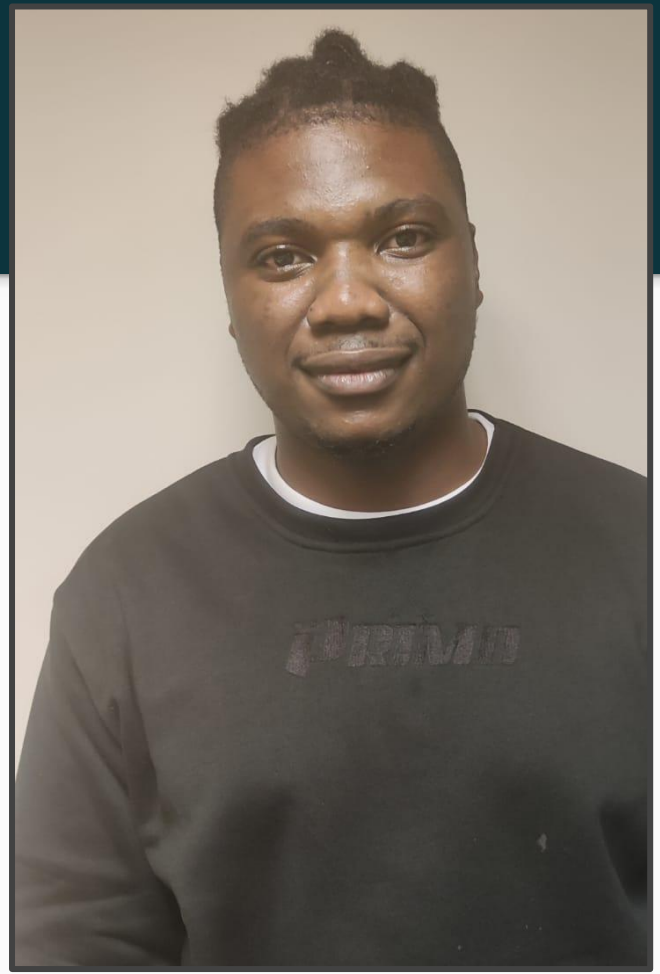
- Documentation
- Developer Operations
- Assisted in the Hosting of the Database and API

Thabo Maduna

- Lead Backend Engineer
- Backend functionality
- Construct API
- Assisted in the Hosting of the Database and API

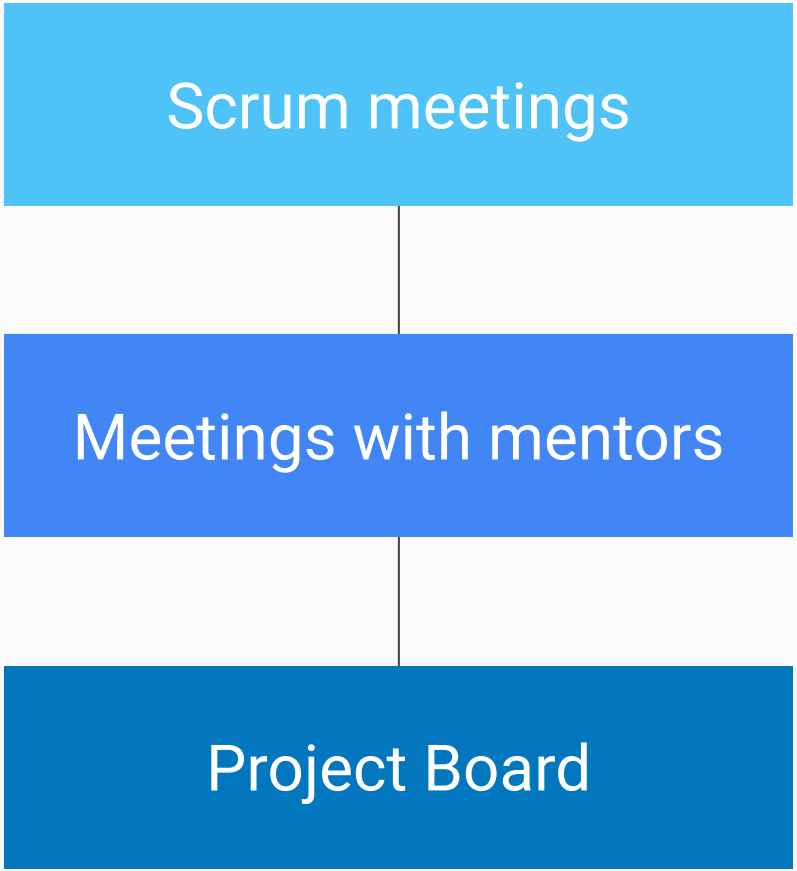
Simphiwe Ndlovu

- Tester
- Assisted in the Hosting of the Database and API



Team Collaboration

Here is what we did to complete and prepare for Demo 2



Documentation?

- SRS document - Functional requirements
- Project board
- Sprint planning
- User manual
- Architectural Document
- GitHub