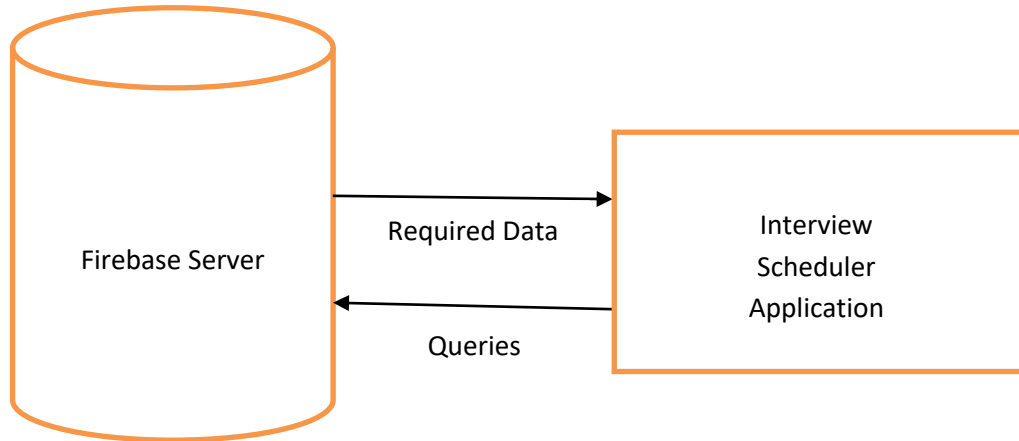
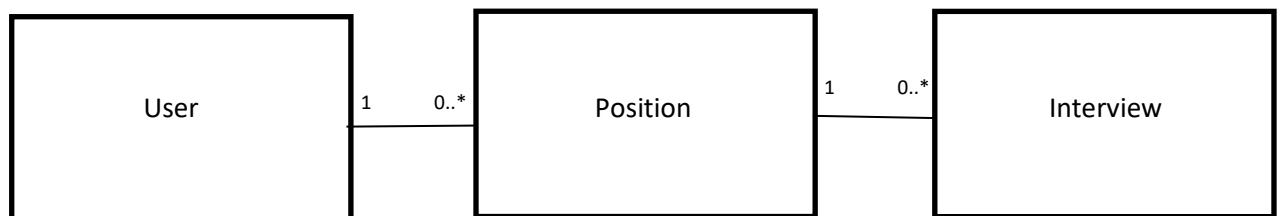


Criterion B: Design

High level architecture overview: -

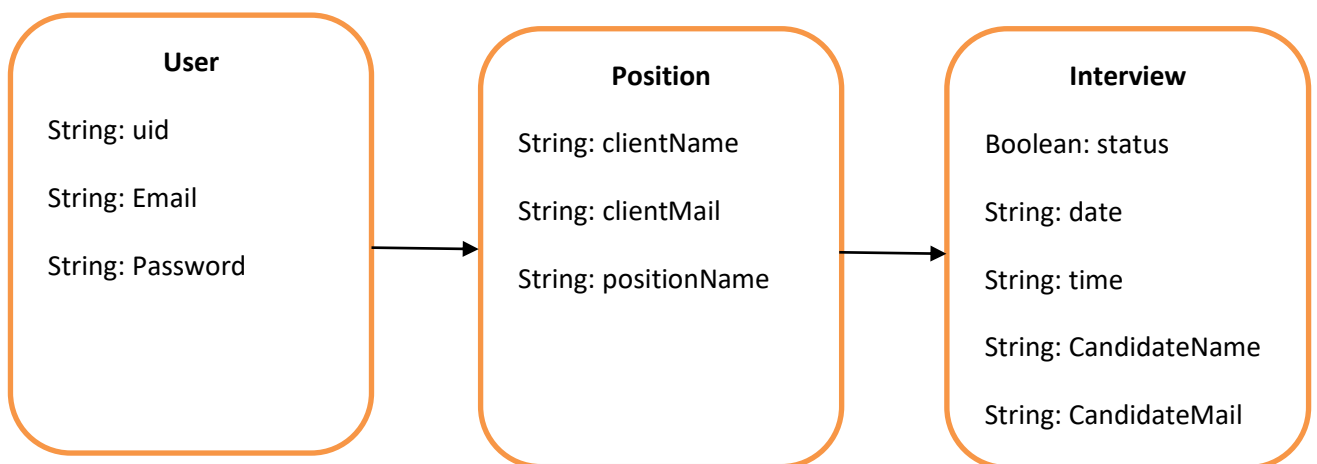


Communication between the two components happens over the internet.



Class Diagram

UML Diagram:



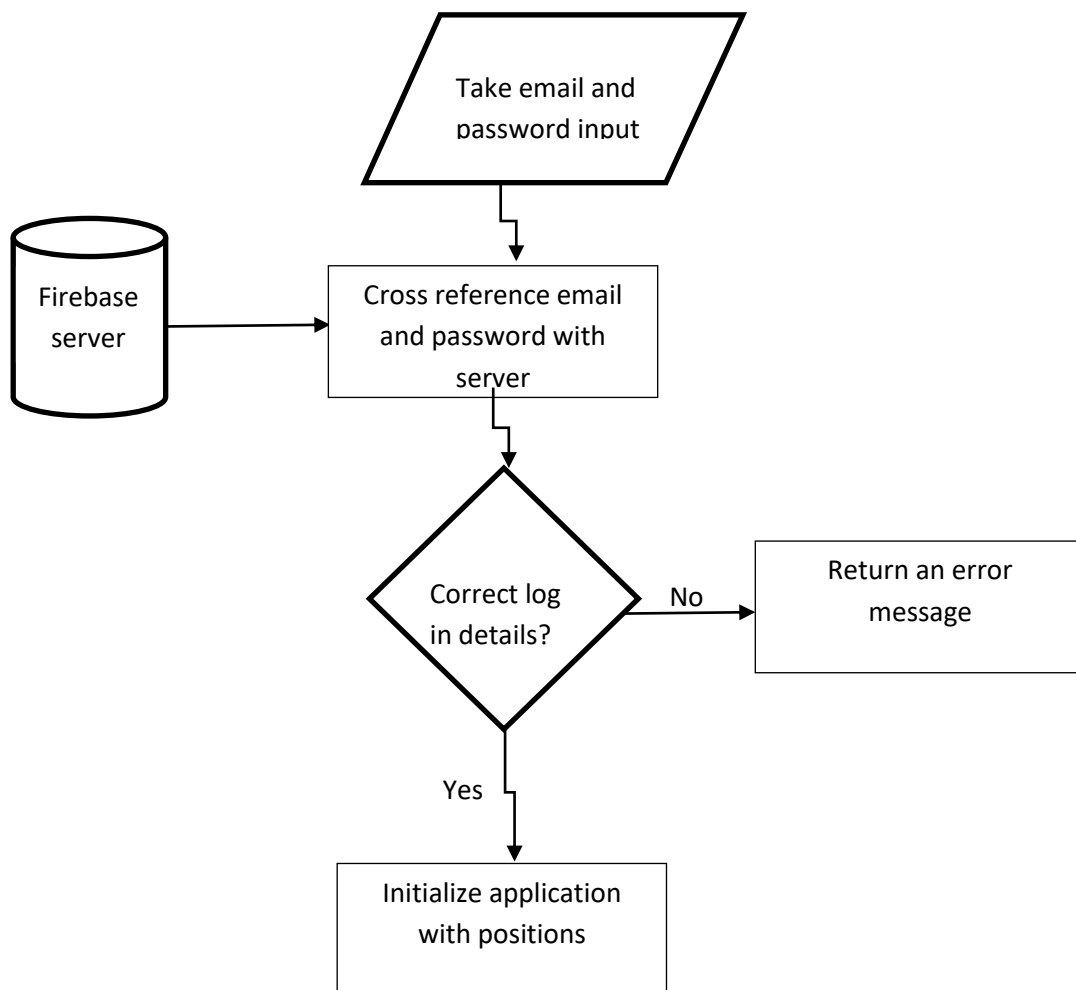
Function of the different classes: -

User: As the name suggests it'll be the client using the software. It has an email and password that allows for authentication and password reset functionality.

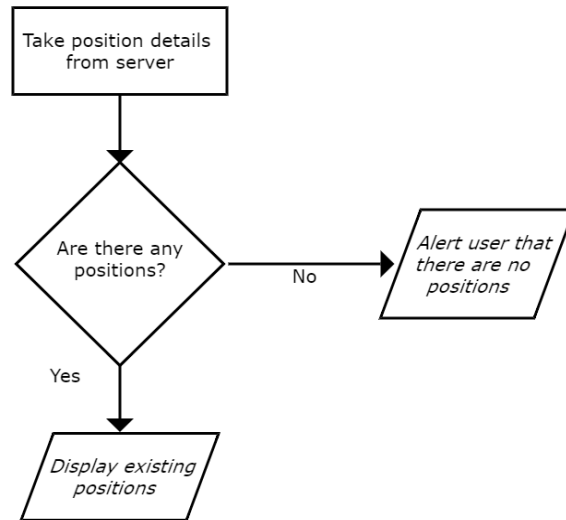
Position: an abstraction of the real world positions the user is scheduling interviews for.

Interview: consists of a time, date and candidate details of the interviews the user schedules. Flow chart for authentication

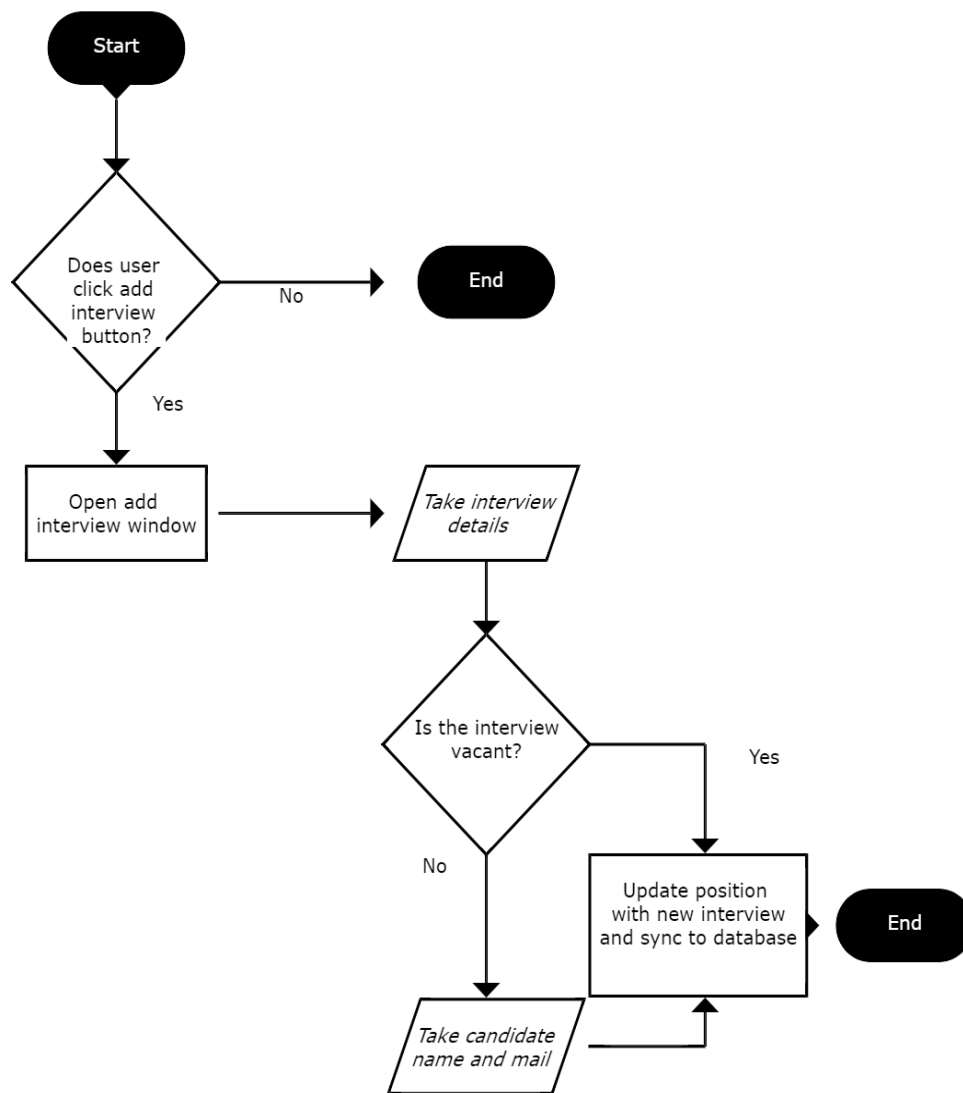
Flowchart for authentication:



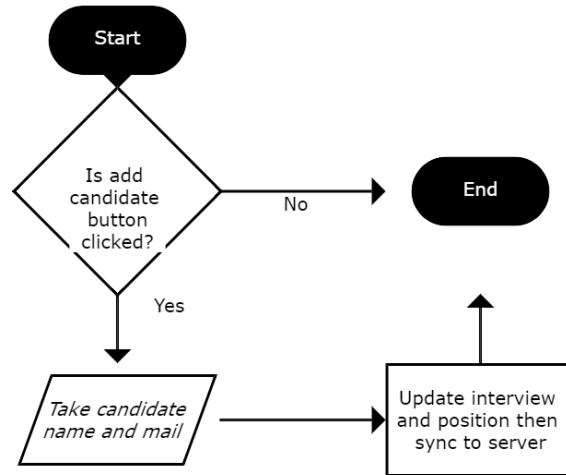
Flowchart to initialize home page



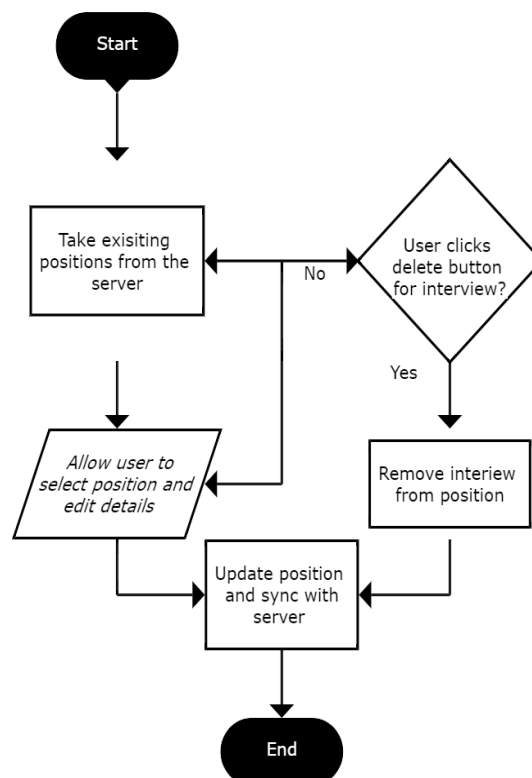
Flowchart to add interviews



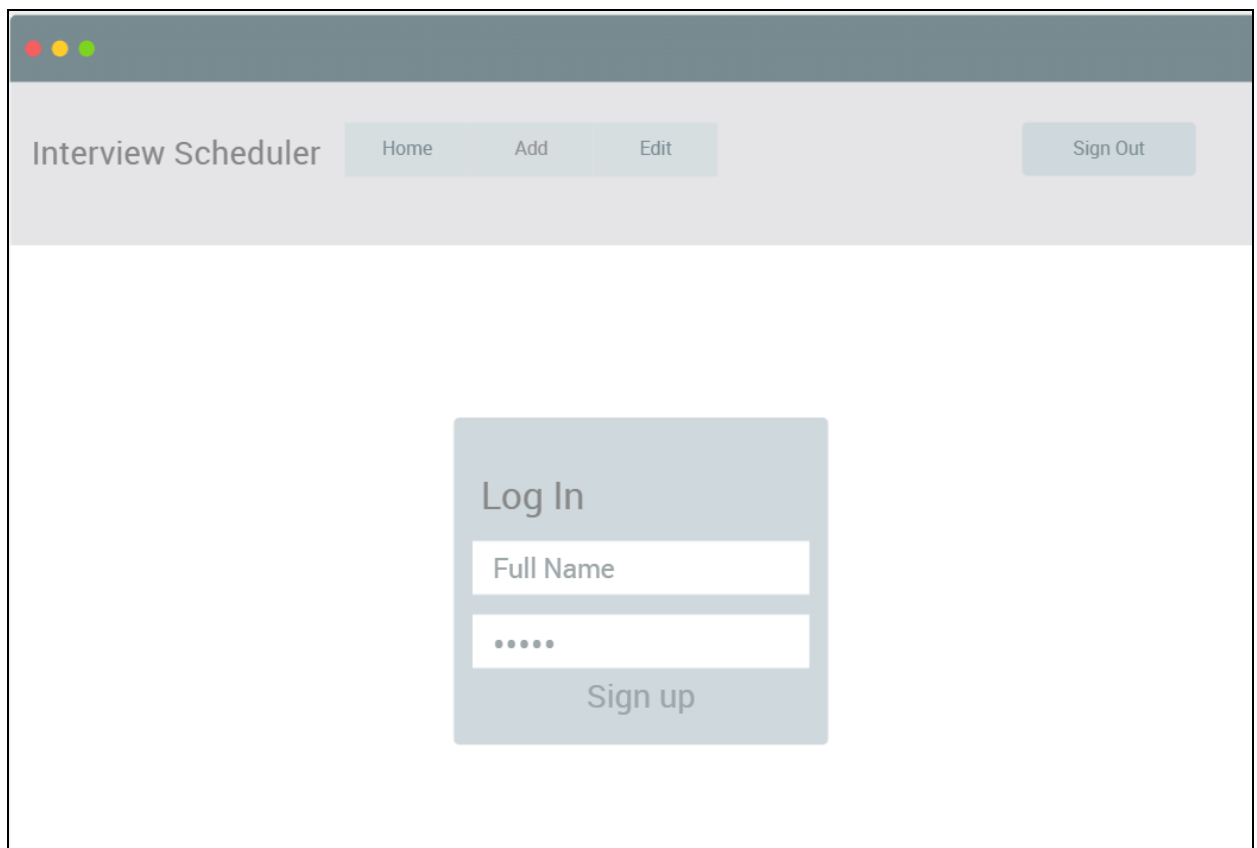
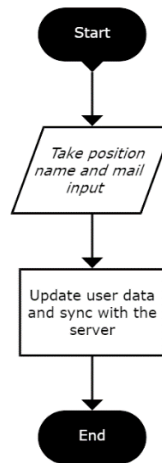
Flowchart to add candidate



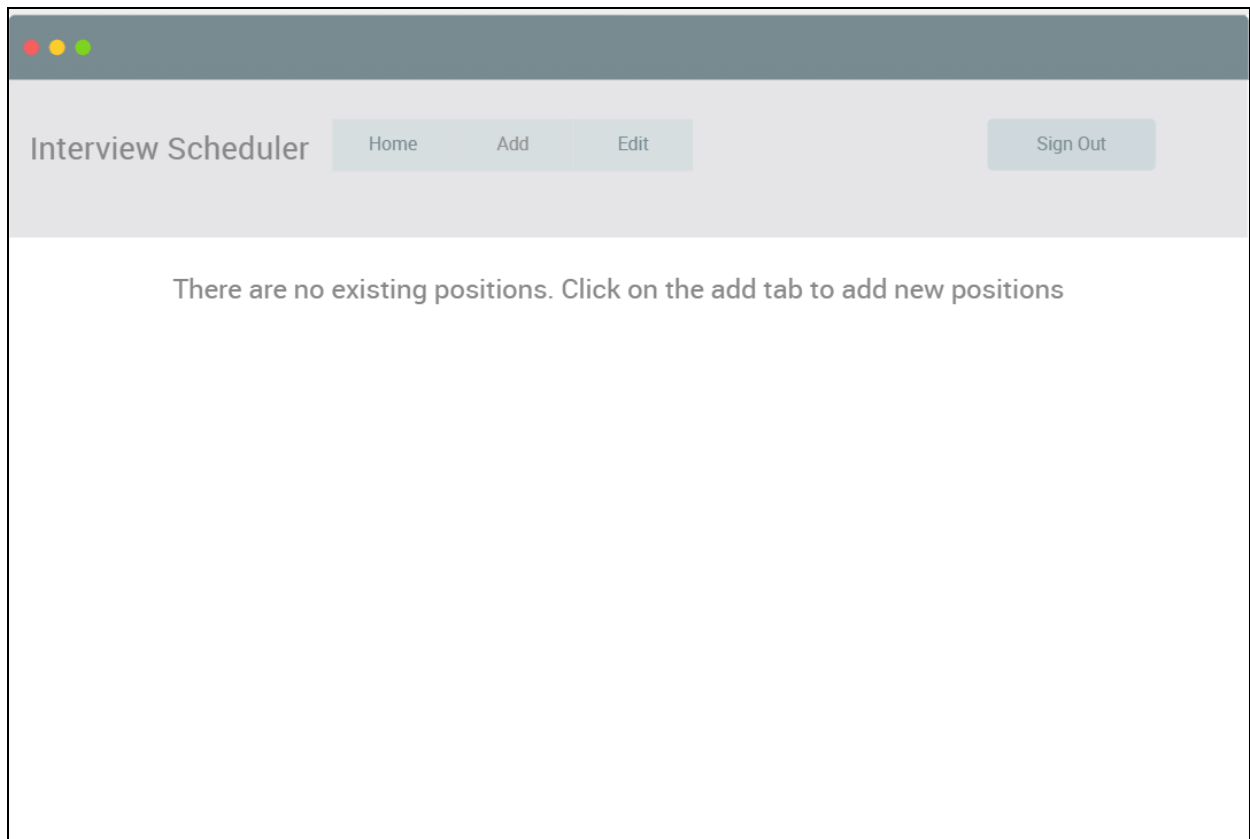
Flowchart to edit position



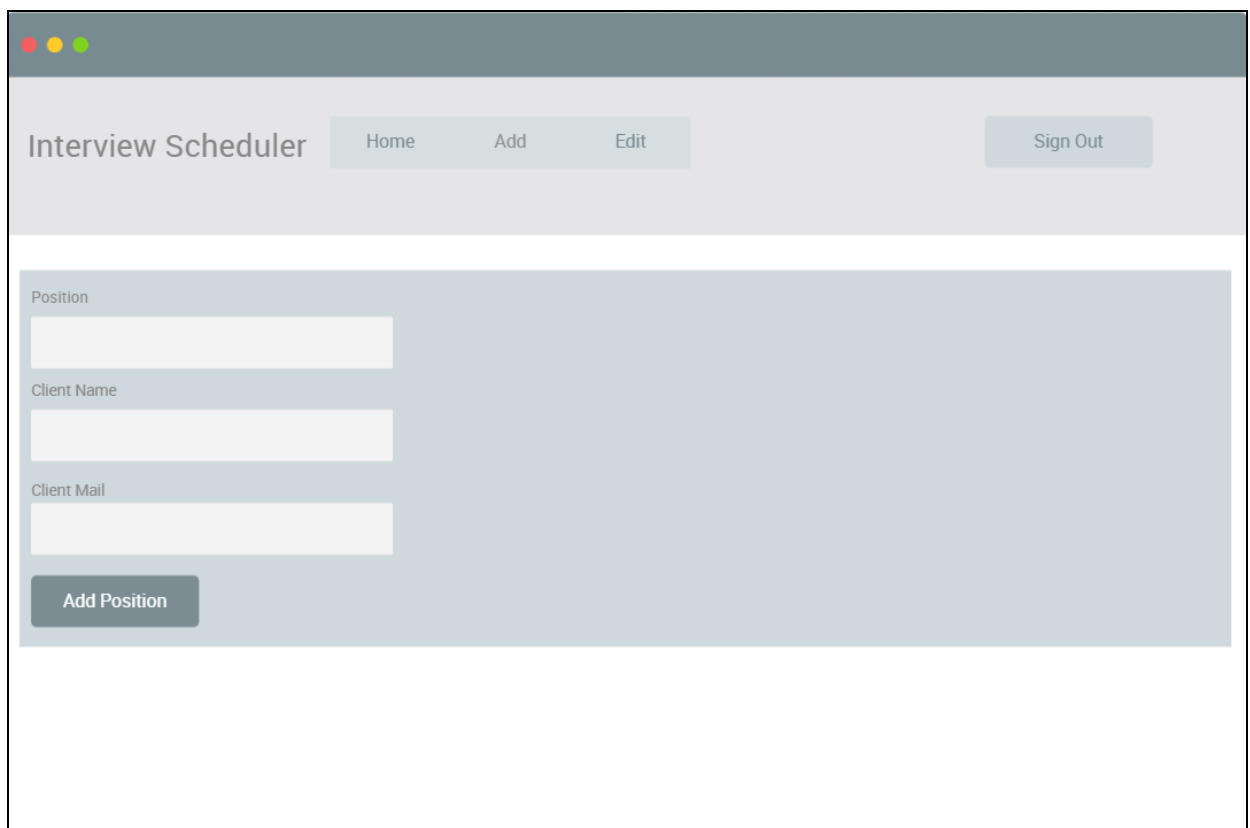
Flowchart for add position page



Prototype 1: - Login Page



Prototype 2: - No positions page



Prototype 3: - Add Page

Interview Scheduler

Home

Add

Edit

Sign Out

Select Position

▼

Position

Client Name

Client Mail

Update Position

Prototype 4: - Edit Page

Interview Scheduler

HomeAddEdit

Sign Out

XYZ

Position

XYZ

Client Name

Name of XYZ

Client Mail

Mail of XYZ

Update Position

Date of interview

Filled: Vacant

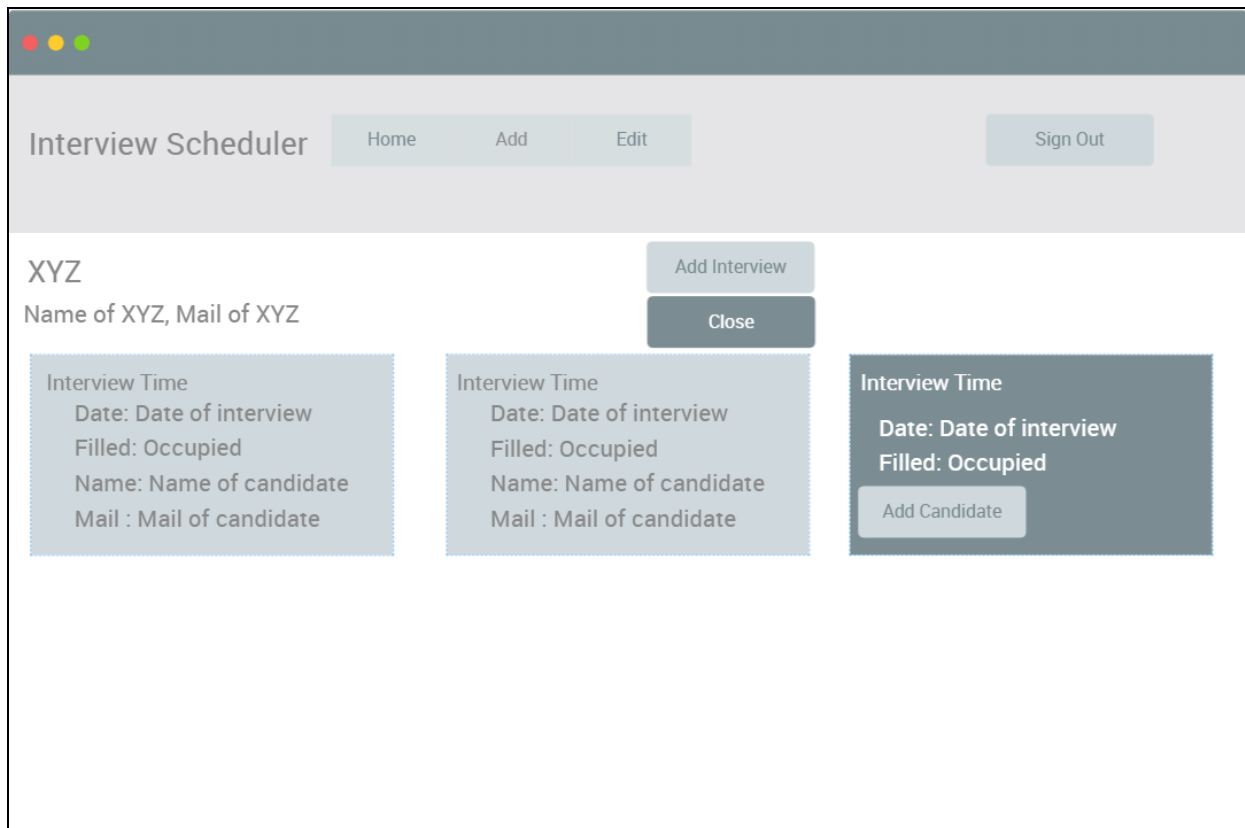
Delete

Date of interview

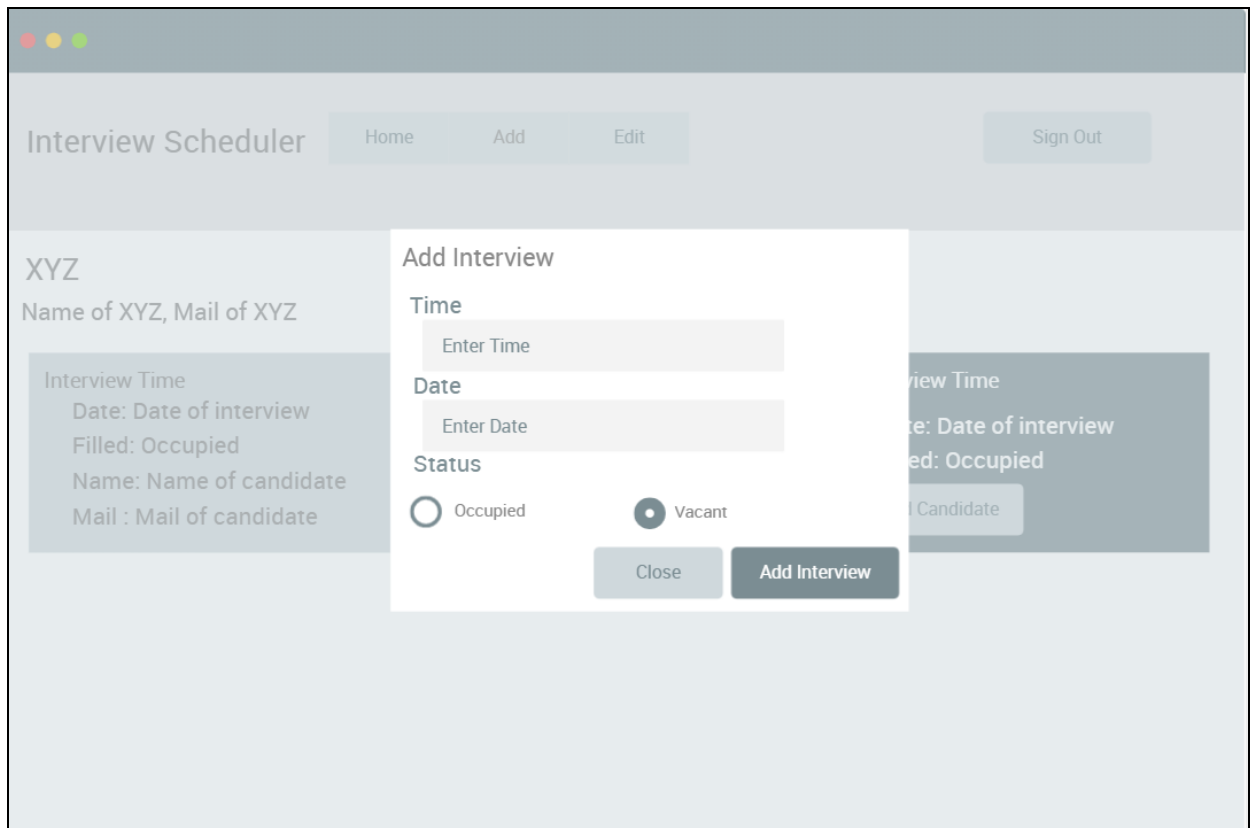
Filled: Occupied

Delete

Prototype 5: - Editing selected position



Prototype 6: - Home page of application



Prototype 7: - Vacant Add interview

Interview Scheduler

Home

Add

Edit

Sign Out

XYZ

Name of XYZ, Mail of XYZ

Interview Time

Date: Date of interview

Filled: Occupied

Name: Name of candidate

Mail : Mail of candidate

Add Interview

Time

Enter Time

Date

Enter Date

Status

☒ Occupied ☐ Vacant

Candidate Name

Enter Name

Candidate Mail

Enter Mail

Close

Add Interview

Interview Time

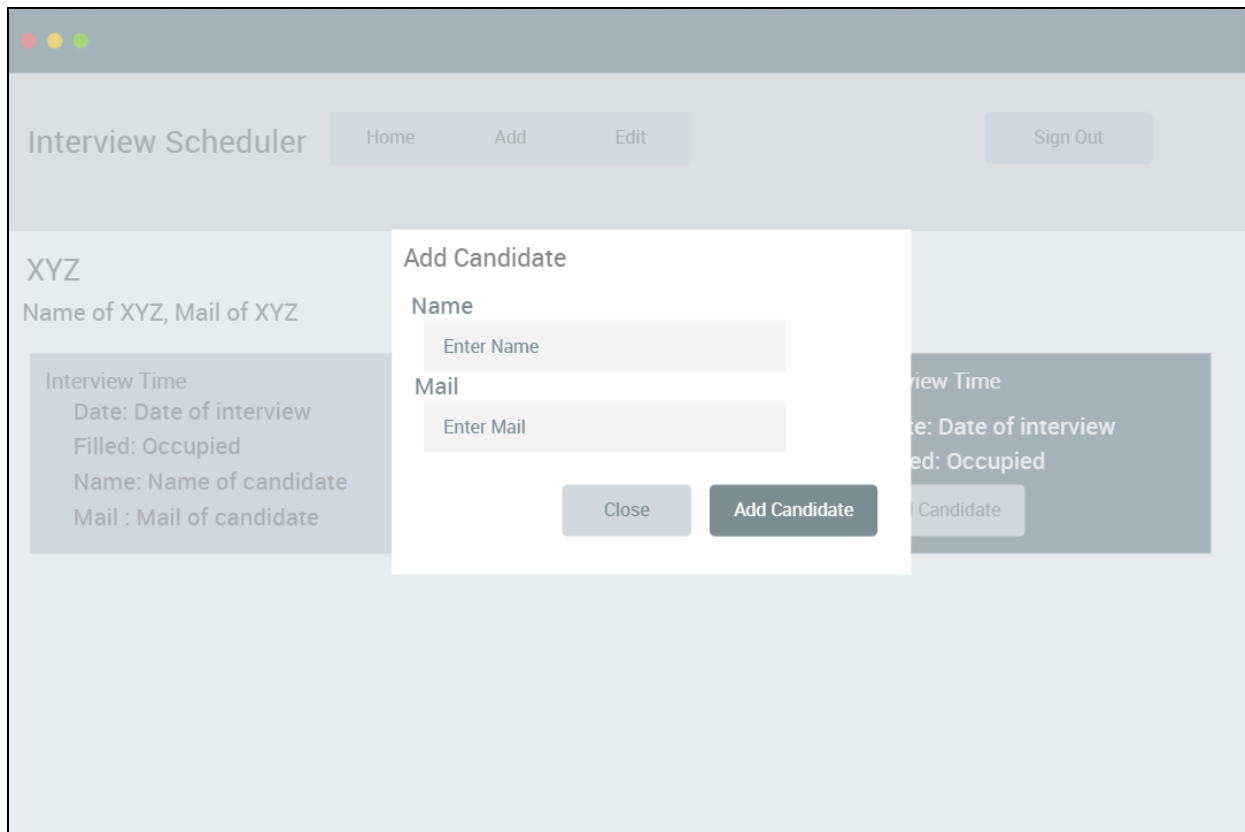
Date: Date of interview

Filled: Occupied

Name: Name of candidate

Mail : Mail of candidate

Prototype 8: - Occupied add interview



Prototype 9: - Add candidate for vacant interview

Testing plan:

Success criteria	Testing plan
Ability to create users	Creating two separate users using firebase
Ability to authenticate users to separate interview data	Logging in from two different accounts to ensure that each has different data
Ability to remain logged into the application	Reloading page, closing window and closing browser to check if user is still logged in
Ability to create positions that are stored in online database	Creating positions and checking for them in home page and on the server
Ability to add new interviews for each position	Adding new interviews and checking for them in both the server and the application
Ability to change vacancy of an interview if a new candidate wishes to join a vacant interview	Adding candidates to vacant interviews and checking if this reflects in the server and home page
Ability to edit position details	Editing positions and ensuring the changes reflect on the home page and server
Ability to delete interviews	Deleting interviews and checking for the changes
Ability to close positions	Closing positions and checking for the changes
Ability to log out of application	Logging out of application and ensuring there is no way to access data until logged back in

Word count: 242