GiveBack Tech Design

# Overview

Mock interviews currently being conducted are managed on Google forms and spreadsheets. The problems arising due to forms and spreadsheets are that they are not scheduled automatically and cancellations are high. The interviewees are not able to determine when the interview will be scheduled. Even the interviewers are busy at times and are not able to conduct interviews at a scheduled time. So rescheduling the interviews is a headache.

# Solution

The prospective interviewers should register to the app as normal users and ask the admin to become an interviewer. The admin shall be able to do a background check on LinkedIn and other social platforms before onboarding the interviewer.

The interviewers should be able to provide their schedules for specific types of interviews (Java, JS, DSA, System design, UX, PM, etc). Eg: on 21/10/2022 from 4 pm IST to 5 pm IST on DSA.

The candidate shall log in to the app as an interviewee and shall be selecting the type of interview they are interested in and should be able to see the available dates and times. Once they select a date and time their interview should be scheduled and they must receive a calendar invitation as well as a reminder.

### Problems

**What if the candidate registers for all interviews present on the screen?**

We need to validate that the candidate **does not have a future interview**, if they have then we do not allow them to schedule another interview. A candidate may only schedule another interview **after a month** of their last interview date.

**Who can reschedule an interview? (Not required as part of MVP)**

A candidate cannot reschedule an interview because the ratio of candidates is much more than the interviewers and there could be a higher chance of candidates rescheduling the interviews for any difficulties. If they are not able to attend the interview, they should simply cancel and book another one.

An interviewer shall be able to reschedule an interview and propose a new time in case they are not able to attend the interview. The Interviewee may accept or decline.

**Cancellations.**

Both parties should be able to cancel the interview stating the reason for cancellation. The reason is to determine if there needs to be a change in the process.

**No Shows**

In case the candidate or the interviewer does not show up, the other party should cancel the interview and mark the reason as “No Show”.

In case none of the parties shows up, the entry could be marked as canceled later stating the reason.

# User Journeys

## Actors

1. **God**: They shall have access to the entire app and every user schedule. They shall be able to perform every action on behalf of the other actors.
2. **Interviewer**
3. **Interviewee/Candidate**

## Login

Any user coming to the app does not need to register, they can directly login to the app via Google OAuth. Once they have logged in, they need to provide information about themselves like:

1. An email address if not present - shall be used to send calendar notifications.
2. LinkedIn profile - for the interviewer to connect with the interviewee if required.
3. Platform Handles: like GitHub, hackerrank, leetcode.
4. Link for resume: for the interviewer to refer before the interview if required.

Users wanting to take interviews shall send a request to update their profile as interviewers. The profile must contain a LinkedIn link and resume link for consideration. And shall be approved by **God.**

Even Gods shall have to follow the same procedure.

## Upload Categories

God should upload different categories in which interviews may happen. These categories shall be further selected by the interviewers and interviewees.

## Upload Schedules

An interviewer shall be taken to the screen where they shall be selecting a category and a date when they are available for the interview. Then they select different time slots when they can conduct those interviews and upload them.

They should be able to see their upcoming interviews and existing uploaded schedules as well.

## Cancel Schedules

In case the interviewer is not available on an uploaded schedule, they should be able to cancel it. In case there is an interview booked against the schedule, the interview shall be canceled as well and the interviewer must provide a reason for cancellation.

## Booking an Interview

For booking an interview, the candidate must have provided a LinkedIn profile link. Once they land on the page, they should select the category in which they want to be interviewed. Upon category selection, they shall be shown the available dates and respective times when the interviews can be scheduled for the category.

The interview should be booked 24 hours before the start time.

We need to validate that the candidate **does not have a future interview**, if they have then we do not allow them to schedule another interview. A candidate may only schedule another interview **after a month** of their last interview date.

## Canceling an interview

A candidate/Interviewer may cancel an interview at any time and should provide a reason for the same. If the start time of the schedule is more than 24 hours in the future, then the schedule becomes available again else it remains booked.

1. What Firebase services would be need
2. Schemas-- the structure of the database
3. Sevices Signatures
4. Local Caches and functions for handling data.
5. Calendar Integration
6. What Firebase services do we need?

1. Authentication - Google oAuth [link](https://firebase.google.com/docs/auth/web/google-signin)

2. Realtime Database

3. Functions

**Realtime Database**

2. Schemas

Login interviewer and interviewee

Google oAuth

Interviewer Table

Interviewee table

| **Variables** | **EXAMPLE** |
| --- | --- |
| Interview Name | “Interview Review” |
| Interview Type | “Programming” |
| Date | “27 may 2010” |
| Start time | “03:20 pm GMT” |

3. Services Signature

4. Local Caches and functions for handling data.

5.CALENDAR INTEGRATION

[link](https://medium.com/flutter-community/flutter-use-google-calendar-api-adding-the-events-to-calendar-3d8fcb008493)