

Submission Instructions

1. After you've completed all the sections, download this document as a PDF [File → Download as PDF]
 2. Create a new GitHub repo for the capstone. Name it "**Capstone Project**"
 3. Add this document to your repo. Make sure it's named "**Capstone_Stage1.pdf**"
-

[Description](#)

[Intended User](#)

[Features](#)

[User Interface Mocks](#)

[Login Screen](#)

[Candidates list screen](#)

[Candidate details screen](#)

[Candidate interview screen](#)

[Open question screen](#)

[Multiple choice question screen](#)

[Questions tablet screen](#)

[Key Considerations](#)

[How will your app handle data persistence?](#)

[Describe any libraries you'll be using and share your reasoning for including them.](#)

[Describe how you will implement Google Play Services.](#)

[Next Steps: Required Tasks](#)

[Task 1: Project Setup](#)

[Task 2: Implement UI for Each Activity and Fragment](#)

[Task 3: Firebase Authentication](#)

[Task 4: Retrieve data from Firebase](#)

[Task 5: Save questions answers/results](#)

[Task 5: Connect Google Drive](#)

GitHub Username: marcinbak

Hirefy

Description

The app helps to facilitate process of hiring new employees, starting with creating candidate profile, screening, and conducting interviews.

It is a step by step guide that makes sure you don't miss anything. Additionally provides full interviewing module where you can design your own questions and see previous interviews results.

Intended User

Companies conducting recruitment processes.

Features

- Display candidates list with scheduled interview times (pre-screen, interview, etc)
- Conduct interview with different type of questions: multiple choice, open questions, etc
- Select candidate, start interview, view results, comments (podio)
- Connection to Podio to get candidates.

Interview: a question per screen

Question types: multiple choice, multiple entry, open question, text page

User Interface Mocks

These can be created by hand (take a photo of your drawings and insert them in this flow), or using a program like Photoshop or Balsamiq.

Login Screen



Login screen with app's (company's) logo and login with Google+ button to login.

Candidates list screen



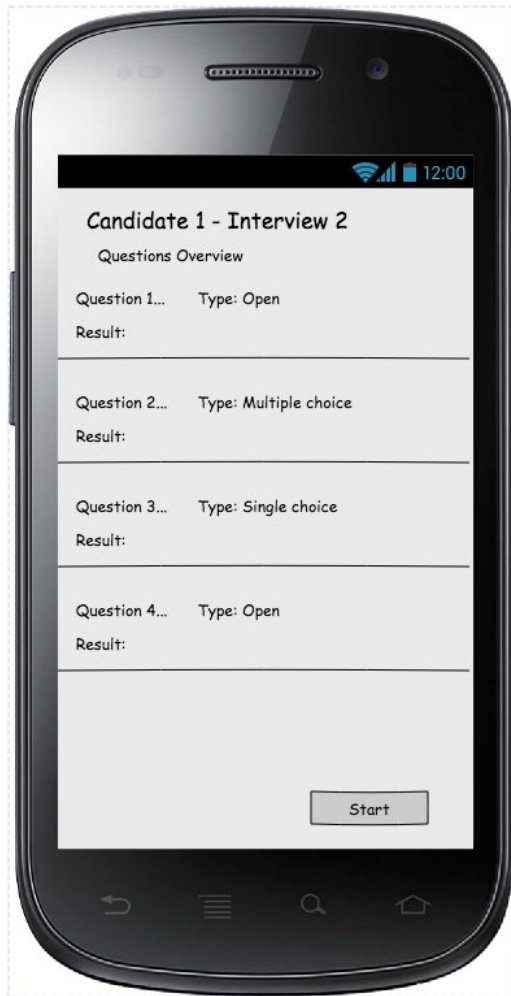
List of candidates with details for each candidate. Add new candidate FAB.

Candidate details screen



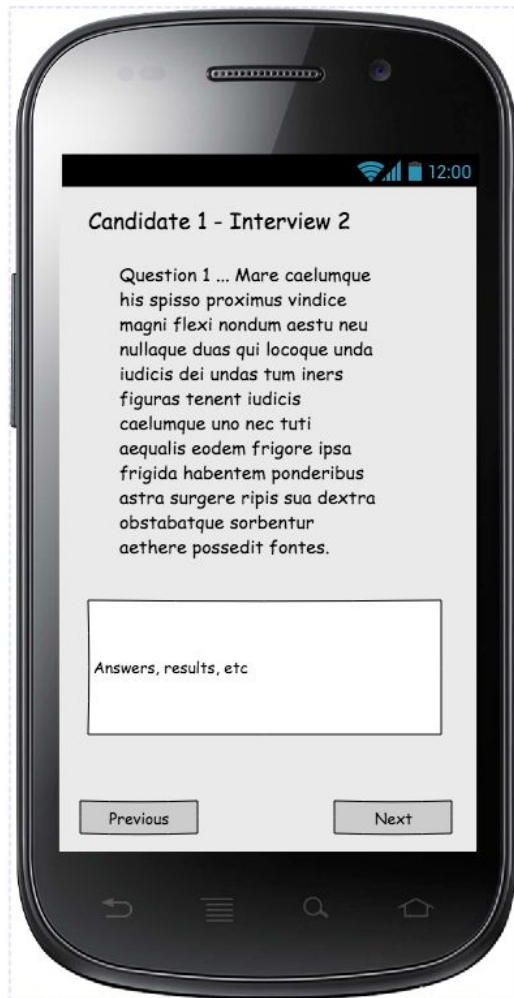
Candidate details. Open/attach resume from google drive. List of past and planned interviews with the candidate. List of comments regarding the candidate profile, posting new comment.

Candidate interview screen

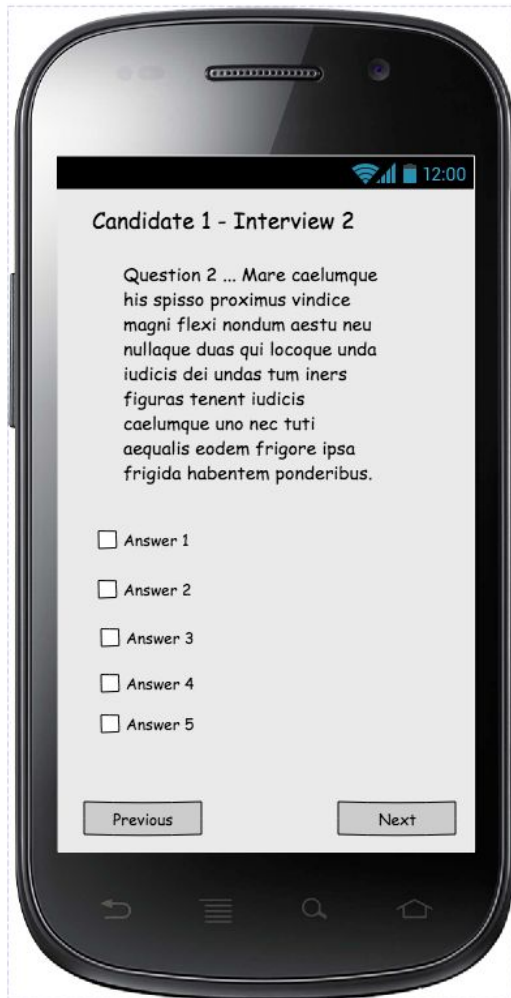


List of question within specific interview. Start button to start interview.

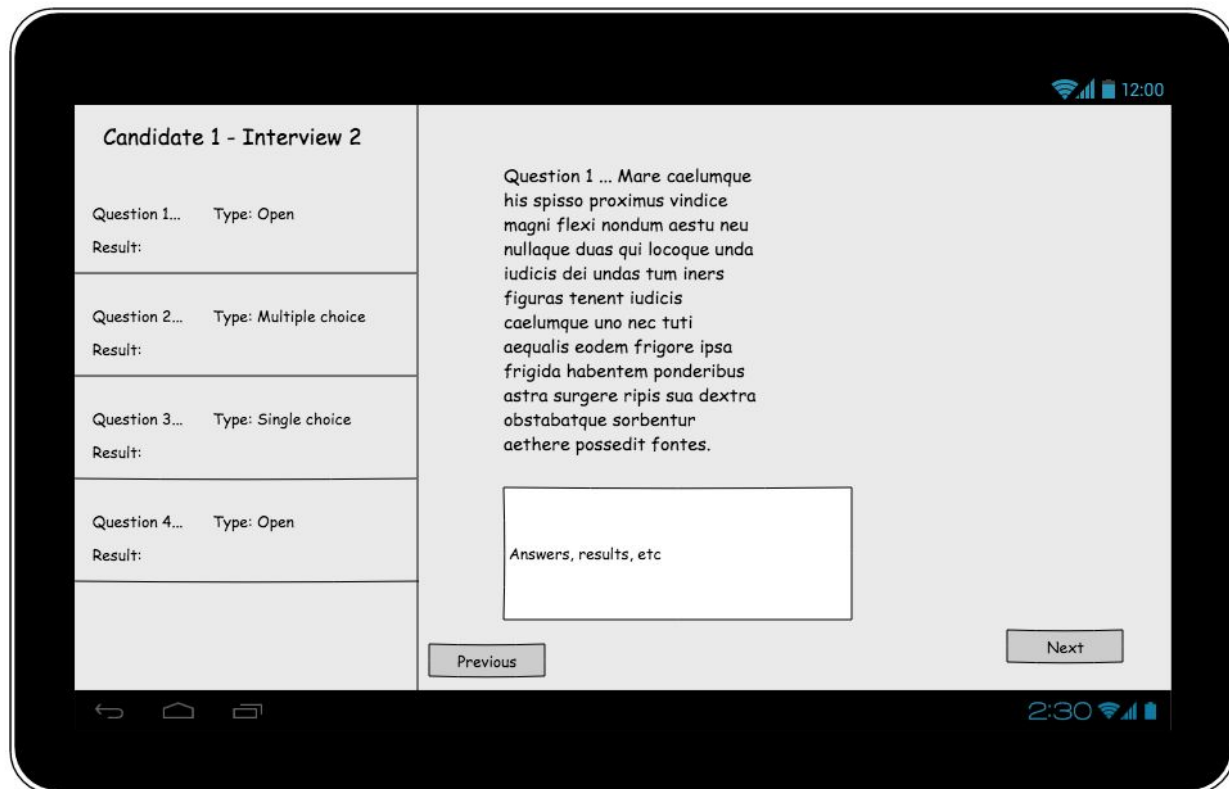
Open question screen



Multiple choice question screen

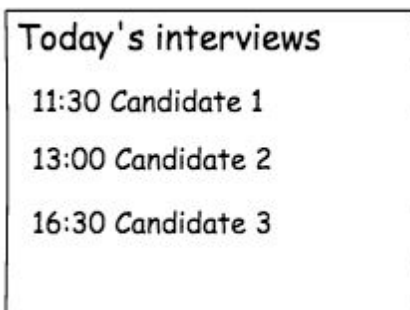


Questions tablet screen



App Widget

App widget will display upcoming interviews for today.



Key Considerations

How will your app handle data persistence?

The data will be stored in Firebase Realtime Database.

The app will periodically poll data for use by App Widget. The scheduling of the poll will happen using Firebase JobDispatcher.

Managing asynchronous calls?

Thanks to Firebase Database there is no need to handle async calls in the app, however App's Widget requires fresh data even when the app is not used. For this I will use Firebase JobDispatcher to fetch fresh data periodically.

Describe any libraries you'll be using and share your reasoning for including them.

1. Dagger2 for dependency management - better code structure and readability.
2. Firebase libraries for authentication, realtime database, crash reporting.

Describe how you will implement Google Play Services.

- Firebase Authentication - user authentication
- Google Account Login - Google+ login
- Google Drive - to access candidate's CV

Next Steps: Required Tasks

This is the section where you can take the main features of your app (declared above) and decompose them into tangible technical tasks that you can complete incrementally until you have a finished app.

Task 1: Project Setup

- Prepare repository
- Configure libraries
- Setup Firebase project
- Create empty app

Task 2: Implement UI for Each Activity and Fragment

List the subtasks.

- Build UI for Login screen
- Build UI for candidates list screen
- Build UI for candidate details screen
- Build UI for Questions list screen
- Build UI for Question scree
- Tablet UI for Questions screen

Task 3: Firebase Authentication

Implement Firebase Authentication

List the subtasks:

- Configure Firebase
- Connect Google+ login action to login button
- User authentication

Task 4: Retrieve data from Firebase

List the subtasks:

- Retrieve list of candidates
- Retrieve candidate details
- Retrieve list of interviews for candidate
- Retrieve interview questions

Task 5: Save questions answers/results

List the subtasks:

- Store answers to different types of questions in Firebase Database
- Add comments to candidate profile

Task 5: Connect Google Drive

List the subtasks:

- Configure google drive access
- Attach Google Drive document to profile as Resume
- Open Resume from Google Drive

Submission Instructions

1. After you've completed all the sections, download this document as a PDF [File → Download as PDF]
2. Create a new GitHub repo for the capstone. Name it "**Capstone Project**"
3. Add this document to your repo. Make sure it's named "**Capstone_Stage1.pdf**"

