How to Use this Template

1. Create a new document, and copy and paste the text from this template into your new document [Select All → Copy → Paste into new document]
2. Name your document file: "**Capstone_Stage1**"
3. Replace the text in green

GitHub Username: https://github.com/JasperEssien2

School Finder

Description

School finder is an app that focuses on school owners and the general public. School owners can make use of it to broaden the awareness of their school to the mass/public. In today's world were parents are looking for the best school to send their wards / children but that hope is squashed by lack of funds, this app might be useful to them. Or a situation where school owners have standard schools, but is hidden due to other great schools overshadowing it. This app goal is the solve the problems listed above. To minimize a situation of fraud or a situation where non standard schools or un-approved school are being uploaded to deceive the public, a provision is made in the app that must be abided by school owners, they must upload a certificate(s) / an image showing the school is approved by the government. To ensure this goal is met, to the

benefit of the school, provisions are made for the school to stand out, by uploading images of the structure, achievements, uploading the subjects, classes/departments and their lecturers/teachers profile handling those subject/classes, displaying their biography which includes but not limited to their qualification and image if they choose too. To prevent situations where normal users spam applications to the school, a provision is made where forms and exam questions for each subjects can be uploaded, to be used for a timed exam when a user applies for a particular class. Parents or students can search for schools near them, in a certain Country/State, or by the type of school (Kindergarten, Universities, High School etc). They can apply for a certain class and depending on the school requirement, if an exam is to be done before applying, the parent can call the child to take a timed exam / test which activates the camera to avoid cheating, results and the filled form are then sent to the school in question for further processing, and the user will be notified when the processing is done.

Intended User

This app is intended for Families, Students, Schools.

Features

List the main features of your app. For example:

- Searches for schools
- Applying for a school
- Taking exams

- This application will be written solely in the Java Programming Language.
- The app will set content description on images to support accessibility, text will be outlined in the res/strings.xml file.
- Resources will be stored in the res dir. And will follow naming conventions specific to the app.
- AsyncTask will be used to get data from google sheet db.

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 Google Drawings, www.ninjamock.com, Paper by 53, Photoshop or Balsamiq.

School Owners flavor prototype

https://www.figma.com/proto/WPPQu7WjRw5p7KTfbAcVqrCV/School-Owner-App-page?node-id=1%3A2&scaling=scale-down

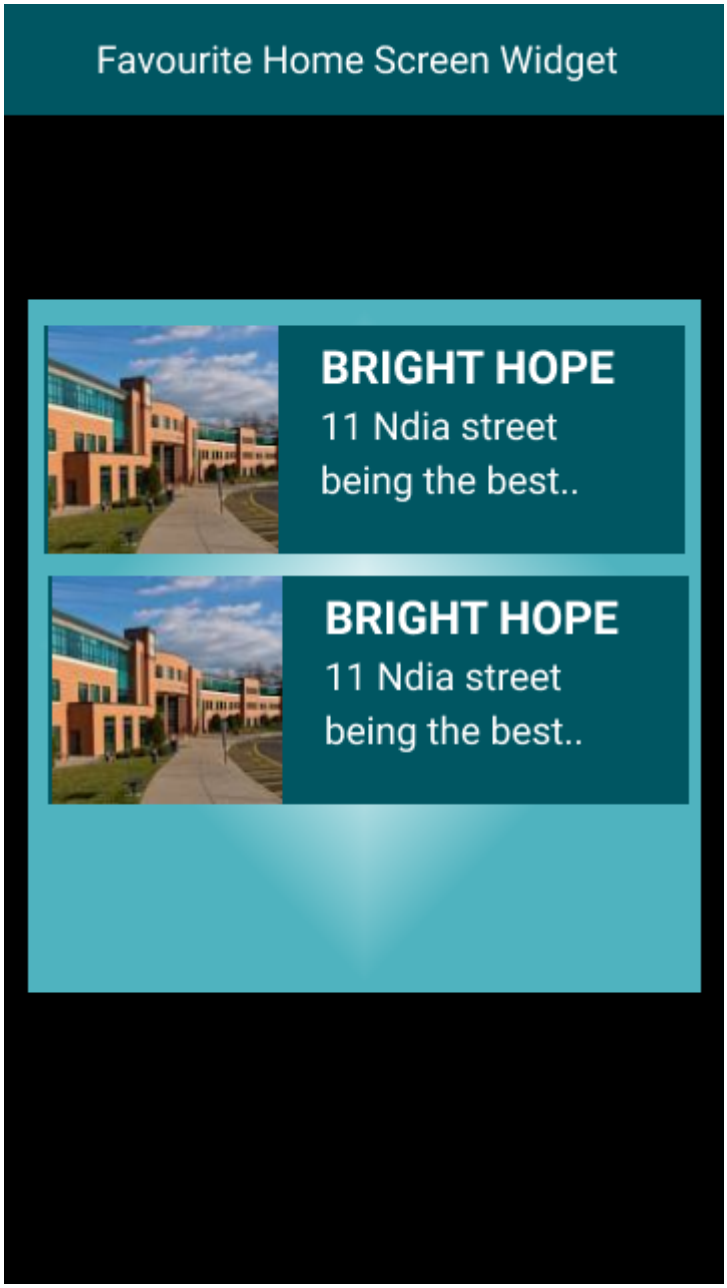
I used figma to design and the link to the prototype is above.

Normal Users prototype

https://www.figma.com/proto/CXByRmQFVGndadNZlwoBr6/Normal-Users-App-Flavor?node-id=1%3A2&scaling=scale-down

Link to normal users flavor prototype, figma

Add as many screens as you need to portray your app's UI flow.



Key Considerations

How will your app handle data persistence?

Using Firebase Real time database to store users details, schools details, etc.
Firebase Storage, for storing images.
Room db: when a school is selected as favourites, will use room db to store it.
Shared Preference: Used in normal users flavor for saving result filter options.

Google sheet db: Used in school owners flavor for storing exams/test questions.

Describe any edge or corner cases in the UX.

Will use befitting transitions

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

- Picasso to handle the loading and caching of images - version: .
- Stacked cardview library will use it for displaying stacked card view as search result in the normal users app flavor - version:
- Country_state_picker for selecting countries and states - version:

Version names for all libraries, Gradle, and Android Studio

- Picasso version: 2.71828
- Com.yuyakaido.android:card-stack-view:2.2.2
- de.hdodenhof.circleimageview:3.0.0
- com.makeramen:roundedimageview:2.3.0
- Gradle version: 4.10.1
- Android Studio version: 3.3

Describe how you will implement Google Play Services or other external services.

Google Sheet db: Used in school owners flavor, for storing application text/exams question.
Google places near me api: for searching school close by in the normal users flavor.
Firebase push notifications: for pushing notifications to the app.

Next Steps: Required Tasks

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

Task 1: Project Setup

To setup and/or configure this project. I will:

- Configure libraries
- Configure gradle to use two flavors (school owner flavor and normal users flavor) of the app
- Adding all necessary dependencies in the gradle, making sure using a stable version
- Configure gradle to use databinding feature
- Add all necessary dependencies for MVVM android architecture pattern

Task 2: Implement UI for Each Activity and Fragment

In this task i'll create each layouts and link them to their activities or fragments

- Build UI for Each activity and fragment
- Create a "dialog box" module for each flavors of the app
- Build UI for all dialog box required
- Create classes and implement methods that will inflate the dialog boxes taking in the context

Task 3: Create required POJO files

In this task i will create pojo models that is required in the app

Task 4: Create a Module for Dummy Data

Using the pojo files i created, i will create a java module that contains java classes to populate dummy data to be used for test purposes.

Task 5: Architecture the app using MVVM architecture

In this task i'll use MVVM pattern to architecture the app

- Create view models to be used
- Using livedata
- Link the generated dummy data to the app.

Task 6: Structure Firebase Realtime Database

Structuring the firebase realtime database in relation to my app needs

- Connect the app to the firebase project
- Add dummy data for test purposes

Task 7: Configure the app to use Google sheet db api

Google sheet db api will be used for storing questions

- To edit or add questions, when question settings button is clicked in the school owners flavor, a webview is opened opening a google db sheet page where they can edit the file
- When apply button is clicked in normal users flavor, if an exam is required, get the google sheet from the specific school firebase node, then using the ap, fetch required questions

Task 8: Furnish the app

This task is to furnish the app, focusing on the design and making it more appealing

- Create different layout variants (Tablet, landscape etc)
- Add transitions animations to activities or fragments that requires it

Task 9: Add widget

In this task i'll add a widget to enable users of the normal flavor to access schools saved as favourites.

Task 10: Test the app

This task is to test the app using espresso testing and probably unit testing

Add as many tasks as you need to complete your app.

Submission Instructions

- After you've completed all the sections, download this document as a PDF [File → Download as PDF]
    - Make sure the PDF is named "**Capstone_Stage1.pdf**"
- Submit the PDF as a zip or in a GitHub project repo using the project submission portal

If using GitHub:

- Create a new GitHub repo for the capstone. Name it "**Capstone Project**"
- Add this document to your repo. Make sure it's named "**Capstone_Stage1.pdf**"