

How to Use this Template

1. Make a copy [File → Make a copy...]
2. Rename this file: “**Capstone_Stage1**”
3. Replace the text in green

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](#)

[Screen 1](#)

[Screen 2](#)

[Key Considerations](#)

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

[Describe any corner cases in the UX.](#)

[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: Your Next Task](#)

[Task 4: Your Next Task](#)

[Task 5: Your Next Task](#)

GitHub Username: Mahendran

Teacher’s pet

Description

The app provides a way for teachers to seamlessly access academic information of their students (both individually and as a class). The app also acts as a portal for academic interaction between the students and teachers by the usage of a Question and Answers wall.

Intended User

Teachers and Students

Features

The main features of the app are:

- Saves information
- Provides a medium for interaction

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.

Screen 1

Login Screen for Teacher



Replace the above image with your own mock [click on the above image, then navigate to

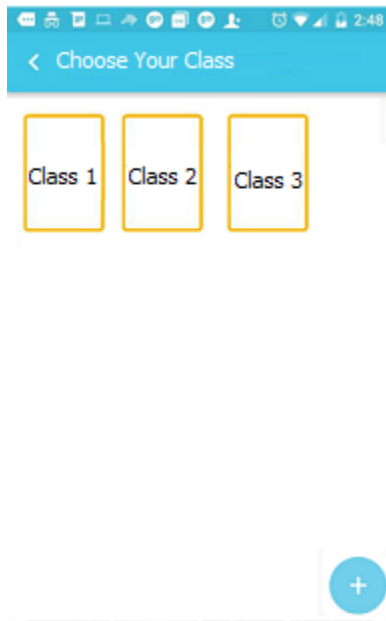
Insert → Image...]

Provide descriptive text for each screen

Screen 2

On Login: Class select screen.

The fab button provides a text bar with which the teacher can add classes.



Replace the above image with your own mock [click on the above image, then navigate to

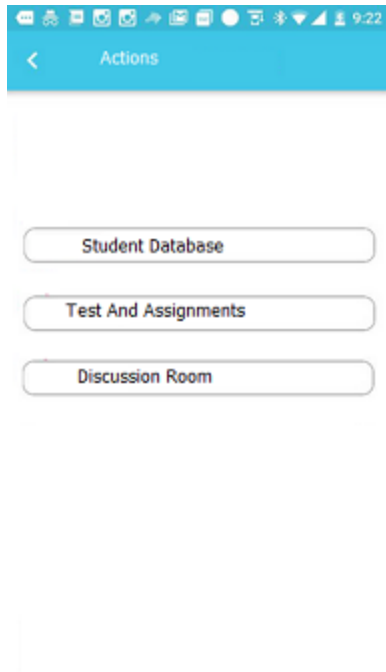
Insert → Image...]

Provide descriptive text for each screen

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

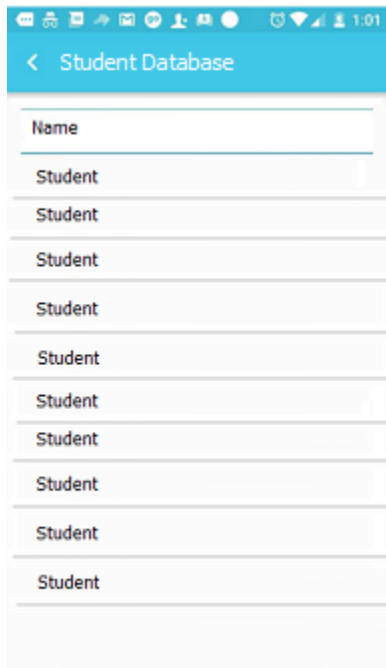
Screen 3:

On Selecting a class the actions screen comes up.



Screen 4:

Selecting Student Database shows the number of students in the particular class.



Screen 5:

On clicking each individual student this details screen comes up. None of them are editable.

Profile

Photo

Name

Phone Number

Parent's Phone No

Address

Performance

CGPA

Rank

74 82

Test 1 Test 1

Screen 6:

On clicking Test and Assignments the below screen comes up. Can view existing tests and create new ones. The average, highest score and lowest score are calculated and displayed automatically.

Tests & Assignment

Test 1	
Credit :2	Syllabus:x
Average:70	Highest Score:90
Lowest Score:40	

+

Screen 7:

On clicking the add button the student database automatically comes up to add marks and remarks of each student.

Test Name:

Credit :

Syllabus :

Name	Mark	Remarks
Student	65	+
Student	65	+
Student	65	+
Student	65	+
Student	65	+

Save

Screen 8:

On clicking the discussion board the below screen pops up. The teacher can only answer the questions of students and can make a class announcement by clicking the fab button.

Question

Answer

Question

Answer

Question

Answer

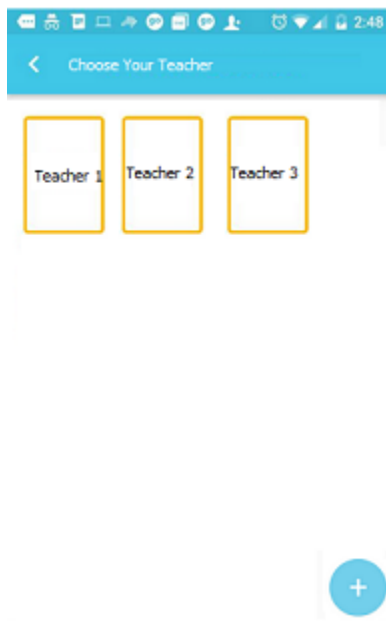
Save

Screen 9:

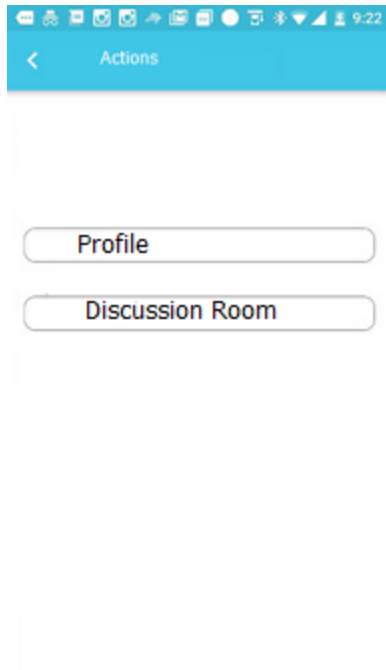
Student Login Screen.



Screen 10:
Teacher selection.

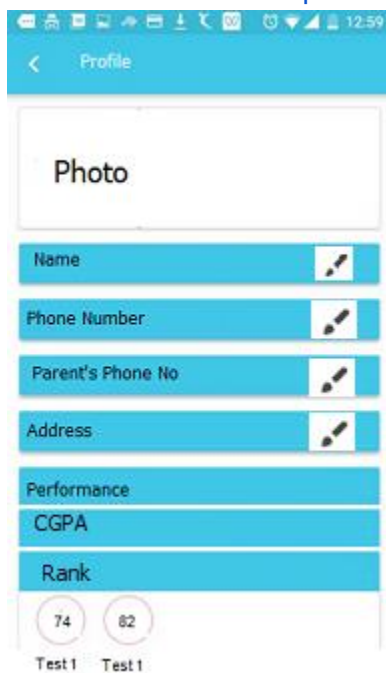


Screen 11:
Action Screen for student



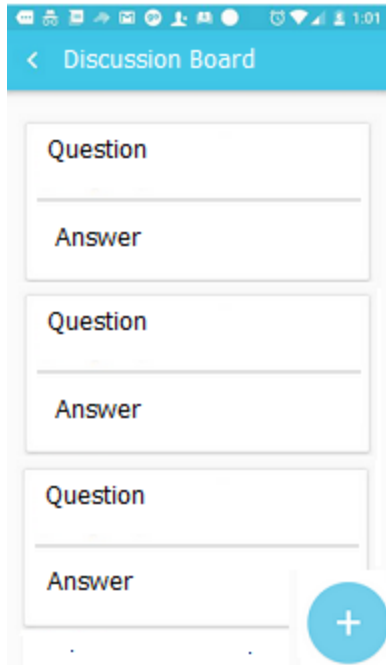
Screen 12:

Student profile. The name, phone numbers and address are editable by the student and will reflect in the teacher's profile.



Screen 13:

Discussion board screen. Students can only ask questions by using the fab button.



Key Considerations

How will your app handle data persistence?

Data Persistence would be handled by the usage of content providers and firebase.

Describe any corner cases in the UX.

Every screen can be reloaded by using the back button.

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

Picasso for loading the photos and butterknife for easy initialization of screens.

Describe how you will implement Google Play Services.

Would be [com.google.firebase](https://firebase.google.com/docs/database/) for database functions and would be using [com.google.firebase.auth](https://firebase.google.com/docs/auth/) to authorize the logging of students and teachers.

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

- Configure libraries

Task 2: Implement UI for Each Activity and Fragment

- Build UI for MainActivity
- Implement main Activity
- Build UI for Actions Screen
- Implement actions screen differentiating student and teacher.
- For teacher create UI for Student Database and implement it.
- Build UI for Student Profile.
- Build UI for Tests And Assignments Screen
- Build UI for Adding Assignments
- Build UI for Discussion Board
- Implement discussion board as per student or teacher.

Task 3: Database

- Implement firebase
- Implement content providers

Task 4: Authentication

- Implement Firebase Auth

Task 5: Making the App Production Ready

- Create Build Variants
- Check for errors

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

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**"