

---

[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:** [shabkal5a6](#)

## Your Wall

### Description

This application is a note app, but these notes will have an icon each and they will be stucked in a wall. In this wall, in addition to sticking notes on it, you can draw or write anything comes in your mind. You will be able to have more than one wall, so that you can separate your notes to different categories. The data of the app will be saved on the device local memory only using sqLite.

### Intended User

Students mostly, but anyone may use it.

### Features

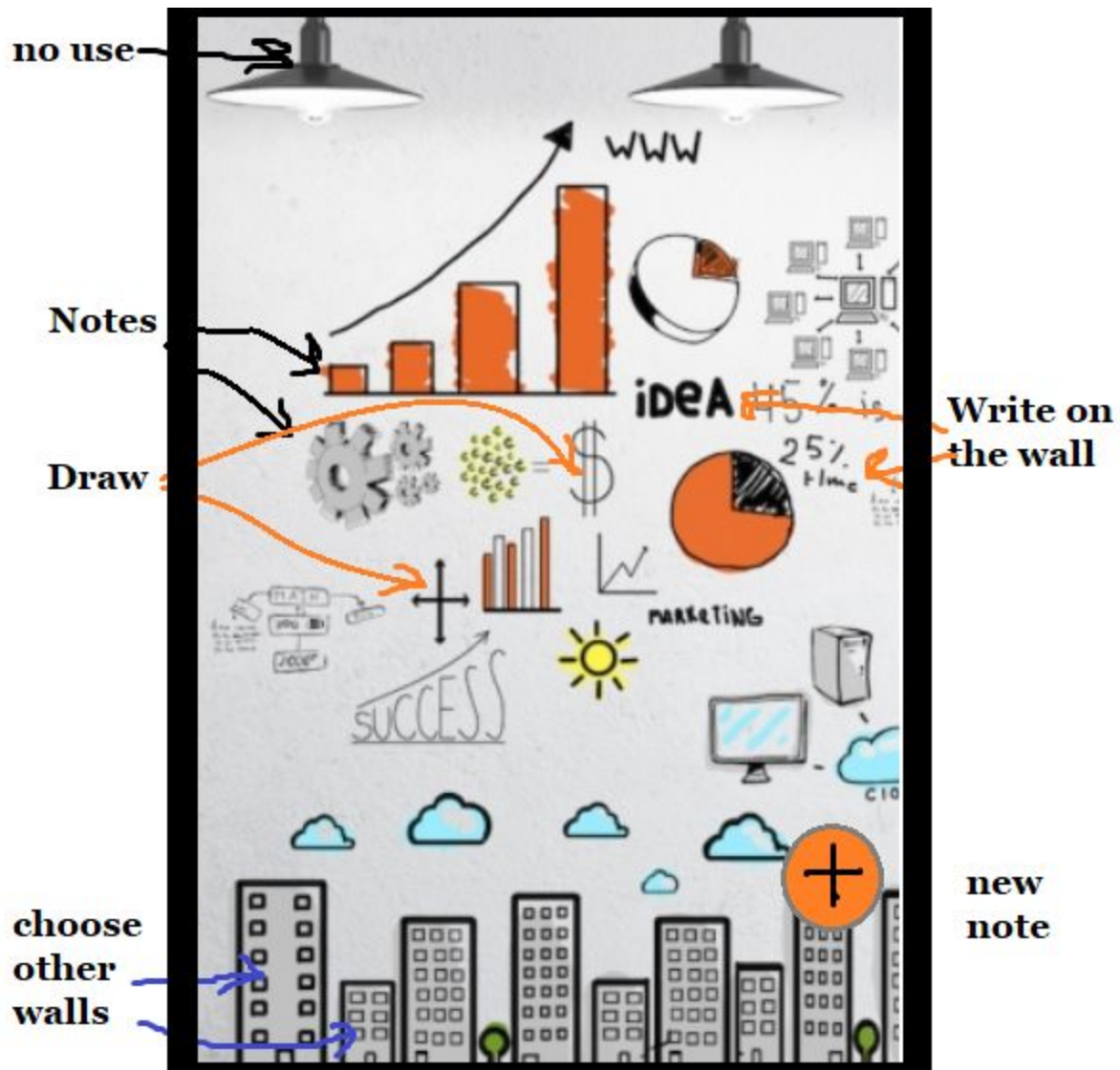
List the main features of your app. For example:

- Saves Notes

- Write notes about notes on the wall
- Draw simple drawing on the wall

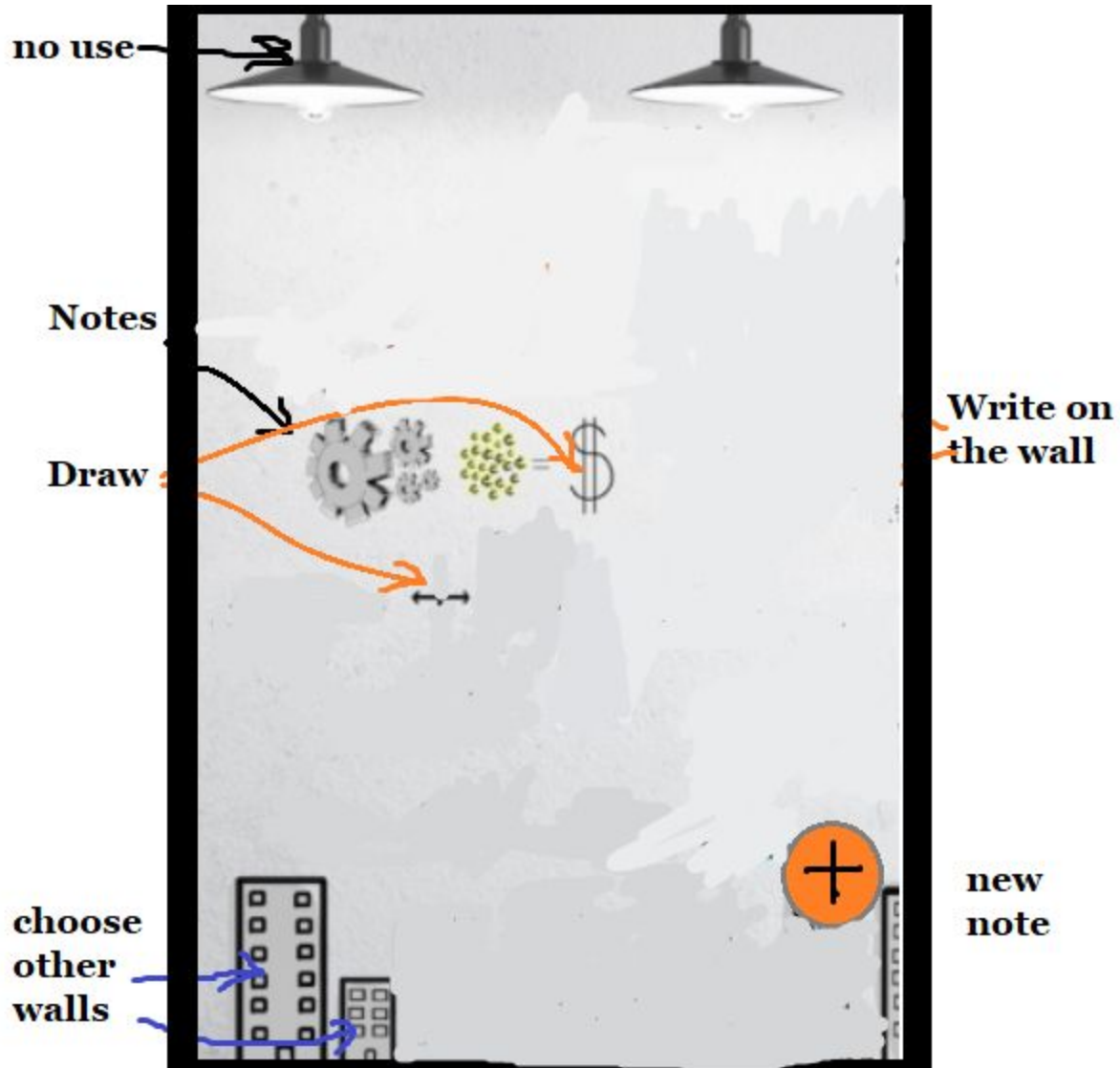
## User Interface Mocks

### Screen 1 : Main UI



As shown in the image above, you can make notes by double click on the location you like or by the new note button, and you can relocate it by long pressing. Inside the + button you'll find button for drawing, writing on the wall and making a new wall, in addition to new note button.

## Screen 2 : Main UI



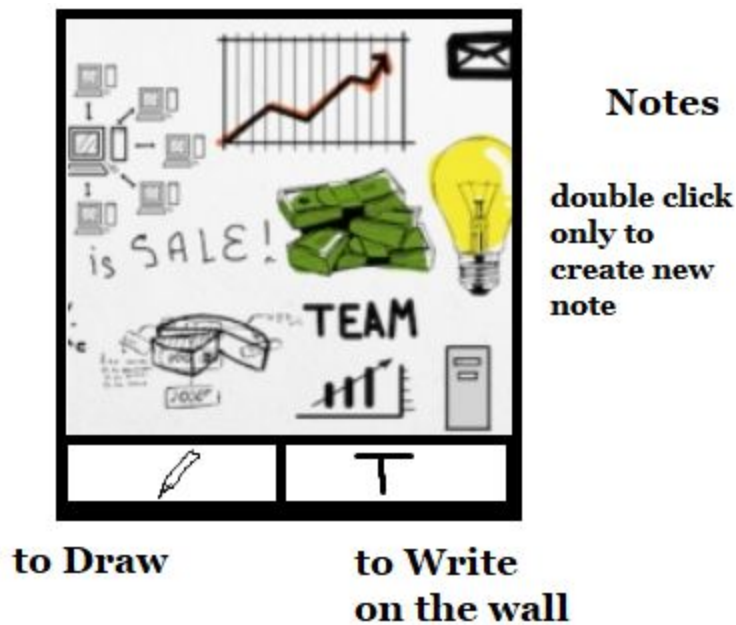
Less notes are created with less walls shown in image 2.

### Screen 3 : New Note UI

The image shows a mobile app screen for creating a new note. It has a light gray background with a dark gray border. At the top, there is a white rectangular input field labeled "Title". Below it is a smaller white rectangular button labeled "Select Icon". In the center is a large white rectangular text area labeled "Body". At the bottom is a yellow rectangular button labeled "Save".

The image above shows New note layout, which will contain title editText and Body editText. Select Icon is a button shows a list of icon, the user will have to select one otherwise default image will be setted. After writing the note, the user have to save the note by clicking the Save button at the bottom.

## Screen 4 : Widget UI



Last image shows the widget ui. It will contain a customView, which allow double click. Button to allow the user to draw and text button to write directly on the view.

## Key Considerations

How will your app handle data persistence?

A content provider will be built to store data locally using sqlite.  
And using AsyncTaskLoader the data will appear on the main ui.

Describe any edge or corner cases in the UX.

No videos.  
no internet connection is required.

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

Picasso

ButterKnife  
AsyncTaskLoader

DualButton library - for edit note ui, will contain (Edit, Save)  
EasyFonts - provide different type of fonts

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

- AdMob : will add advertisement to the app
- Cloud Firestore (Firebase service) : all data of the user will be stored in the cloud.

## 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

- Download Picasso Library
- Add the Firebase Admin SDK to the app.

### Task 2: Implement UI for Each Activity and Fragment

List the subtasks. For example:

- Build UI for MainActivity
- Build a CustomView to contain the notes in the main\_layout
- Build Notes Adapter to draw the notes in the CustomView
- Add icons for images
- Build UI for NewNote activity
- Build a UI for EditNote activity

### Task 3: Build Database and connect to Cloud Firestore

- Database should contain two tables, one for notes and the other for walls
- Notes table should have a foreign key that referenced to the wall id
- Sync data with the cloud.

#### Task 4: Implement onTouchView in the MainActivity

- Double click, makes new note on that location

#### Task 5: How to draw on the CustomView:

- Since there is no library that can do the same, i will go through some steps to make my own class as mentioned in this source:

<https://stackoverflow.com/questions/16650419/draw-in-canvas-by-finger-android>

- CustomView will extends view
- Draw all notes icons on canvas.
- On the canvas, you can draw by finger - when the drawing button clicked -.

#### Task 5: fixing bugs, improving ui

---

#### 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**"