[Description](#_sm4ra97uwo11)

[Intended User](#_aws88pzfmqca)

[Features](#_zheq5430xrpq)

[User Interface Mocks](#_giquerrw6g46)

[Screen 1](#_a4jdupabry3k)

[Screen 2](#_dpcbbkx5yry)

[Key Considerations](#_gvcvmae8jn8u)

[How will your app handle data persistence?](#_v8my7nhtvz0m)

[Describe any corner cases in the UX.](#_gw69vjn1ico0)

[Describe any libraries you’ll be using and share your reasoning for including them.](#_6yqqubmw5bs)

[Describe how you will implement Google Play Services.](#_qrxg682nywe6)

[Next Steps: Required Tasks](#_v518bncmggeg)

[Task 1: Project Setup](#_8oe8zpk3qsmp)

[Task 2: Implement UI for Each Activity and Fragment](#_rzllsk6uqztx)

[Task 3: Your Next Task](#_fdmohs7hes)

[Task 4: Your Next Task](#_umfwsvmx7tpn)

[Task 5: Your Next Task](#_kjidlkq4xm3u)

**GitHub Username**: Sneha010

MonoMoM

# Description

With MonoMoM write your notes and share across the meeting participants. It uses Google's smart Nearby Message solution that merges meeting notes captured by all the members in the meeting. It is smart digital alternative over manual process. It is easy to use and allows cloud syncing that lets you access your meeting notes anytime anywhere. Hence, help your meetings to become more productive and effective

Now no worries of losing the important notes taken by you just because you lose your notepad.

Have to leave off in middle of meeting? Don't worry you will not miss anything, read full version of MOM later with MonoMoM.

Save paper, Go digital with MonoMoM!

# Intended User

Anyone can use this app, who wants to make their meetings or discussions fruitful.

# Features

List the main features of your app. For example:

* Saves information
* Takes pictures
* Other features like that

# 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



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



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.

# Key Considerations

### How will your app handle data persistence?

Describe how your app with handle data. (For example, will you build a Content Provider or connect to an existing one?)

### Describe any corner cases in the UX.

For example, how does the user return to a Now Playing screen in a media player if they hit the back button?

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

For example, Picasso or Glide to handle the loading and caching of images.

### Describe how you will implement Google Play Services.

Describe which Google Play Services you will use and how.

# 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

Write out the steps you will take to setup and/or configure this project. See previous implementation guides for an example.

You may want to list the subtasks. For example:

* Configure libraries
* Something else

If it helps, imagine you are describing these tasks to a friend who wants to follow along and build this app with you.

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

List the subtasks. For example:

* Build UI for MainActivity
* Build UI for something else

## Task 3: Your Next Task

Describe the next task. For example, “Implement Google Play Services,” or “Handle Error Cases,” or “Create Build Variant.”

Describe the next task. List the subtasks. For example:

* Create layout
* Something else

## Task 4: Your Next Task

Describe the next task. List the subtasks. For example:

* Create layout
* Something else

## Task 5: Your Next Task

Describe the next task. List the subtasks. For example:

* Create layout
* Something else

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**”