

[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: [trikhatarun](#)

Memories-What Happened Here Reminder

Description

This is a location based reminder app which will remind user of a certain event/memory when he/she is in/near the area.

Intended User

People who forget events', occasions', memories' dates, this app will remind them of the time they had spent at that location when they are nearby.

Features

- Location based reminder
- Stores data on firebase database

- Caches the current month events in memory locally
- Firebase login to store data with google calendar api

User Interface Mocks

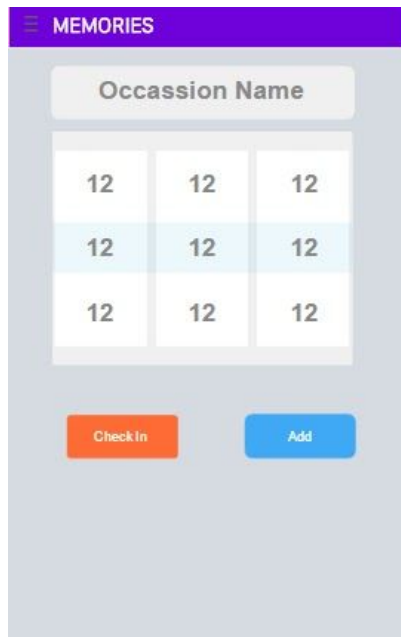
These can be created by hand (take a photo of your drawings and insert them in this flow), or using a program like Photoshop or Balsamiq.

Screen 1



Main screen which user encounters when they first open the app, it contains what happened last year on the current date.

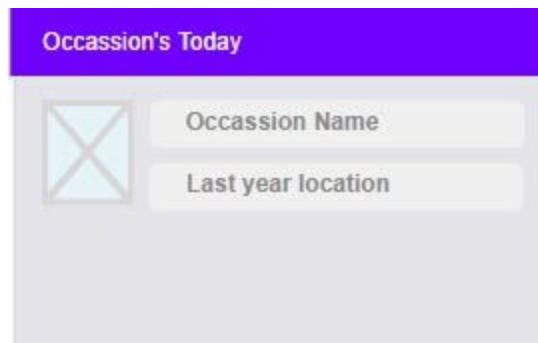
Screen 2



Screen 2 is titled "MEMORIES" in a purple header bar. Below the header is a text input field labeled "Occassion Name". Underneath the input field is a 3x3 grid of input fields, each containing the number "12". The middle row of the grid is highlighted in light blue. At the bottom of the screen are two buttons: an orange "Check In" button and a blue "Add" button.

Add event screen which opens on clicking fab button.

Screen 3: Widget



Screen 3 is titled "Occassion's Today" in a purple header bar. Below the header is a square icon with a light blue background and a grey 'X' inside. To the right of the icon are two text input fields: "Occassion Name" and "Last year location".

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

Key Considerations

How will your app handle data persistence?

Data will be persistent throughout, irrespective of changing devices as all the data will be stored on google calendar and firebase storage.

Describe any corner cases in the UX.

Internet Connectivity: This corner case occurs when there is no internet connectivity. It will be handled by displaying previously cached data and displaying no internet connectivity toast but if there is no previously cached data, no internet connectivity error will be displayed to the user without crashing the app.

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

Picasso: To load images

ButterKnife: To bind views

Design library: To use design tools provided by google

Describe how you will implement Google Play Services.

Google/Firebase Auth: To login user and store their details to their relevant.

Google Calendar: To store user events in their calendar

Firebase database: To store user related data.

Firebase/Google Analytics: To analyse the user activity to improve the app.

Admob: To display ads

Firebase Job dispatcher: to fetch data from calendar api every month

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

Adding relevant libraries and testing them out with small tasks to get familiar with their functioning.

Task 2: Implement UI for Each Activity and Fragment

- Building UI for main activity
- Building UI for add event activity
- Building UI for ViewPagerFragments for today and month view

Task 3: Implement Firebase Login

- Implement firebase login to gain access to user's calendar, only google login.

Task 4: Implement google calendar api

- Implement google calendar api.
- Implement add event functionality to the app with calendar.
- Implement event fetching from calendar.

Task 5: Implement Firebase storage

- Store the data related to event in Firebase database, photos taken on that day.

Task 6: Implement admob

Task 7: Implement Firebase JobDispatcher to fetch events from google calendar once a month for complete month.