

[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: SreeshasKS

Attendance Tracker

Description

All in one Attendance tracker and Manager. Provides a dashboard for a custom generated time table and shows you the attendance statistics. Stores a history of all your attended classes and semesters. Provides importing data from .xsl format files. Exporting to .xsl format files. Provides an admin console for mentors to keep track of attendance during the semester classes.

Intended User

School / College Students and Mentors/Teachers.

Features

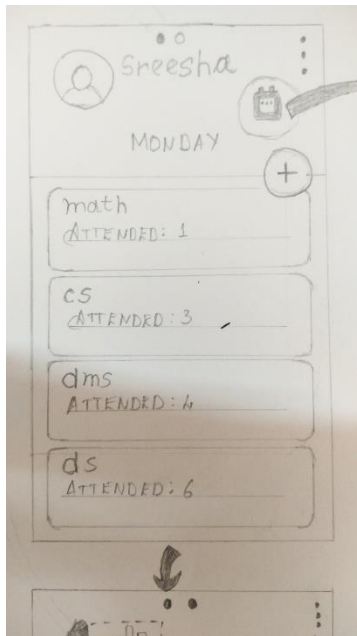
- Provides a dashboard to view classes attended for the entire week.
- Provides an Admin feature to host classes.

- Provides an Admin console to mark attendance of students enrolled into their classes.
- Provides statistics on classes attended and attending.
- Predicts attendance based on the time table and the number of classes missed.

User Interface Mocks

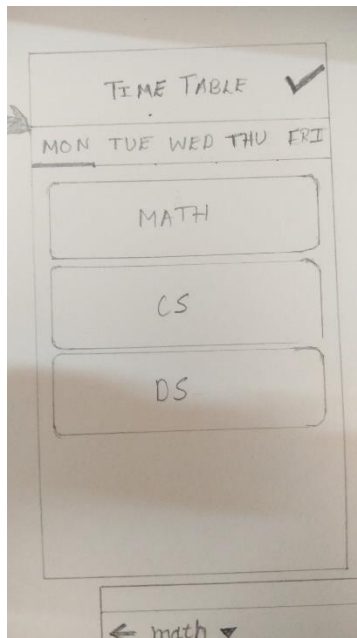
These can be created by hand (take a photo of your drawings and insert them in this flow), or using a program like Photoshop or Balsamiq.

Screen 1



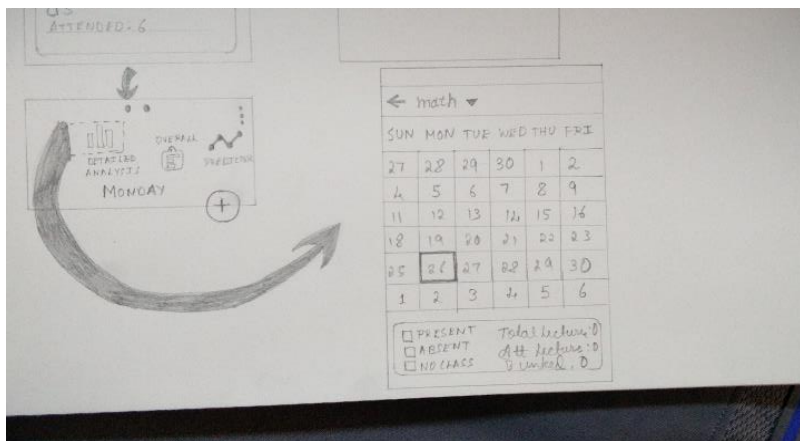
The screens above are clear and concise as to what the app will do. I shall add features soon.

Screen 2



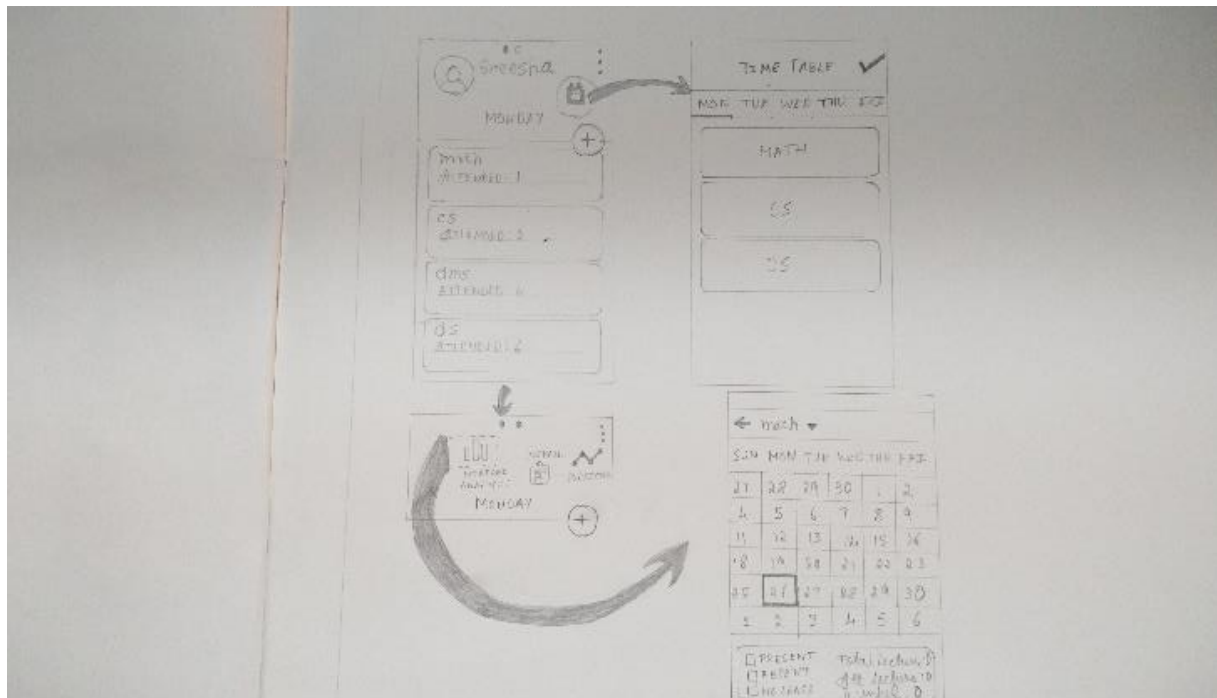
Time Table Activity to view Daily Time Table.

Screen 3



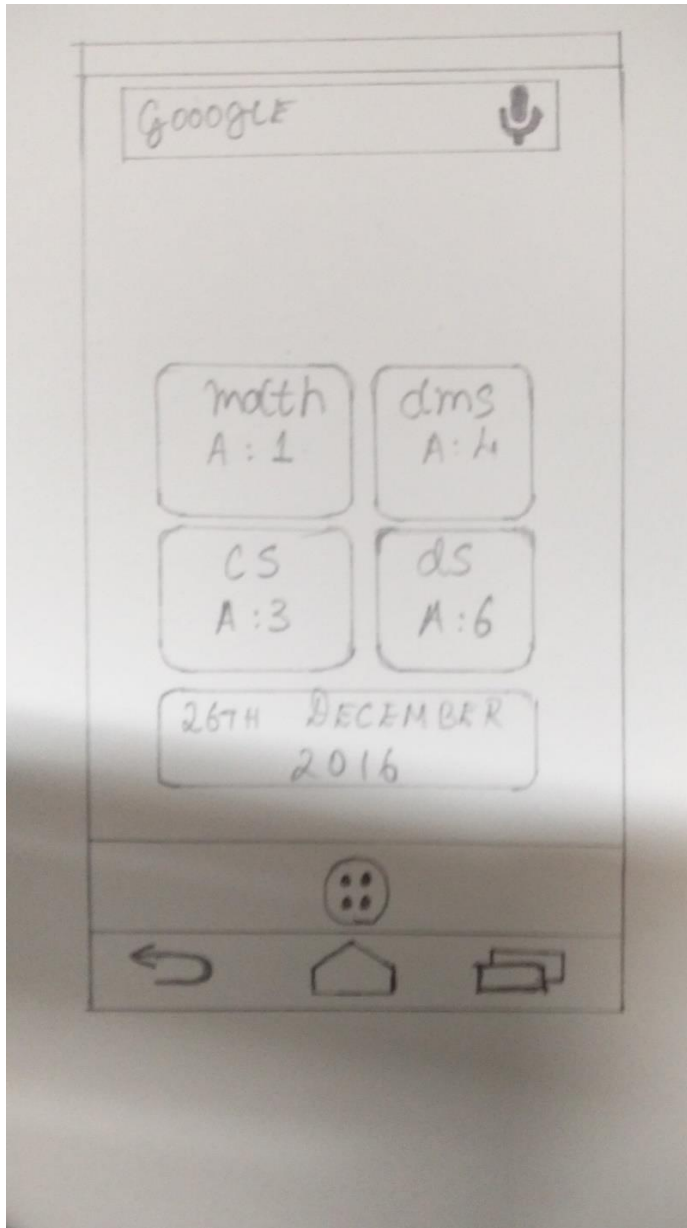
View Detailed Analysis of user attendance data.

Screen 4



Over all activity transition and activity UI mocks.

Screen 5



Widget Mock.

Key Considerations

How will your app handle data persistence?

SQLite for handling persistent data and content provider for abstraction of the data layer from the UI layer.

Describe any corner cases in the UX.

App shall use a Navigation Drawer to navigate to the key features of the app. And a back button, other than the system button, shall be provided to navigate to the root activity of the app.

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

1) **com.afollestad.material-dialogs:core:0.8.6.2 : For Material Dialogs**

Describe how you will implement Google Play Services.

- 1) Fire Base : For user authentication.
- 2) Fire base Messaging: For Real time Messaging.

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

- Setup the Login Activity for User Authentication Using Fire Base.
- Setup the Async Task to download User Data and register User data for a new registration
- Setup the Main Activity which shows a dashboard of the time table for the user.

Task 2: Implement UI for Each Activity and Fragment

- Build UI for Time Table Activity
- Build UI for Detailed Analysis Activity
- Build UI for Overall Attendance Activity
- Build UI for Predictor
- Build UI for Admin Console for teachers
- Build UI for Registered Students Activity

Task 3: Fire Base real time Database

- Implement the Data Model Using No SQL – Fire Base Data Base
- Implement SQLite data base to mirror the No SQL data base on Fire Base in the local app.

Task 4: Implement Server Data Base Schema

- Implement Server side data base to store user data and run analytics
- Implement Async Task to query the server for data.
- Implement Scheduler to sync data between the app and the server

Task 5: Link Multiple Activities

- Link All the Activities , Validate and Parse JSON data from server and Fire Base.
- Implement Data Wrangling using Parcellable for passing data between activities.

Task 6: Implement Real Time Notifications

- Implement Fire Base Cloud Messaging for real time alerts of attendance changes made by the admin.
- Implement Attendance sharing among peers , using notifications for subscribed peers.

Task 7: Import and Export Attendance Data using Excel files

- Implement the server side code for importing Excel Data and Exporting Excel Data
- Use JSON Generated from Excel for User Authentication and integration
- Use Excel Data for Admin Registration.
- Export Student Attendance to an Excel Sheet which can used as a hard copy !

