Capstone - Stage 1

GitHub Username: VarunBarad

Attendance Tracker

Description

Worried about your attendance like every regular semester?
No need to anymore, Attendance Tracker helps you keep track of attendance in different classes, so that you always know which classes you must attend if detention is not something on your wishlist.

Intended User

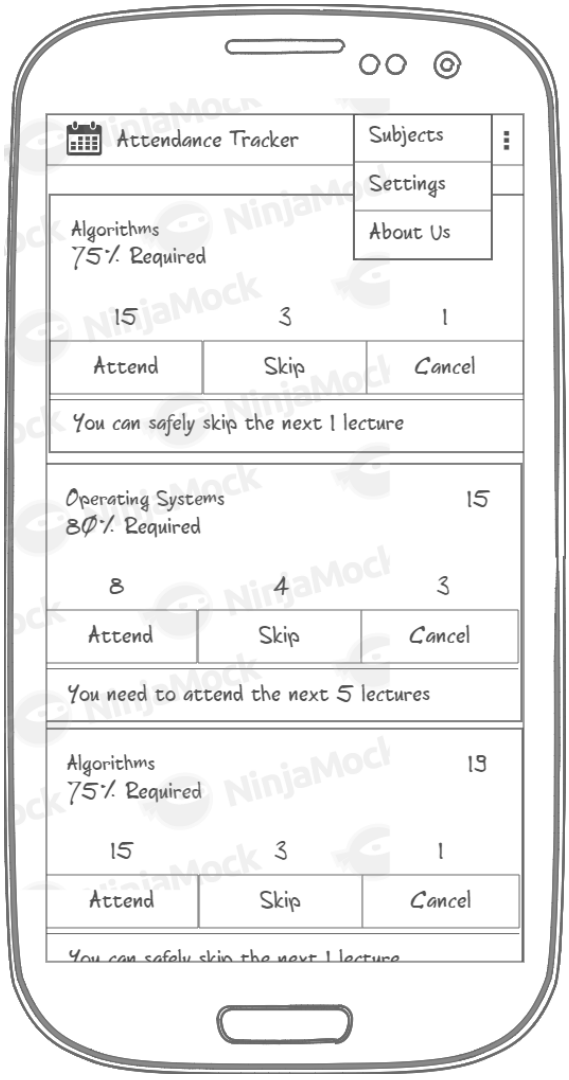
College students who are worried about whether they have maintained their attendance above the minimum required level to avoid detention.

Features

- Easily mark off whether you attended any class on a particular day.
- Forgot to mark someday? No problem, just go to the history and edit it.
- Add a widget to further simplify it.
- Archive past subjects so they don't create clutter.
- Get a bird's-eye view (monthly view) of any subjects attendance.
- Set a custom threshold of required attendance per class.
- Data backup on Google Drive to simplify setup on getting a new device.

User Interface Mocks

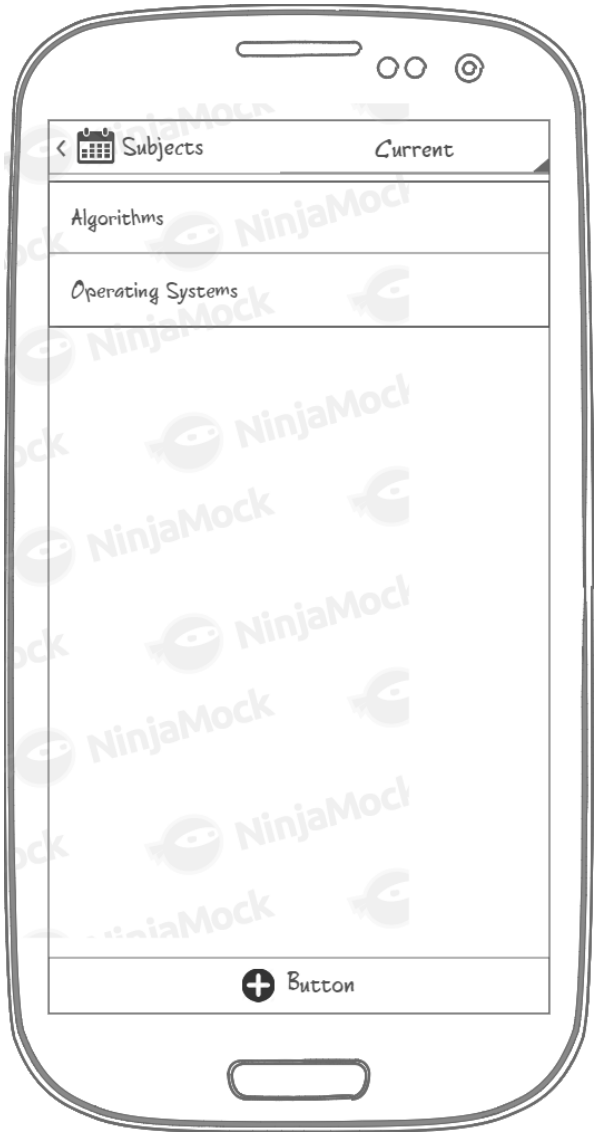
Screen 1 - Current Classes' Attendance



The landing screen of the app that lists all the current classes and provides a quick way to mark whether you attended/skipped any class today.
In the navigation menu, users can choose to go to one of the following screens:

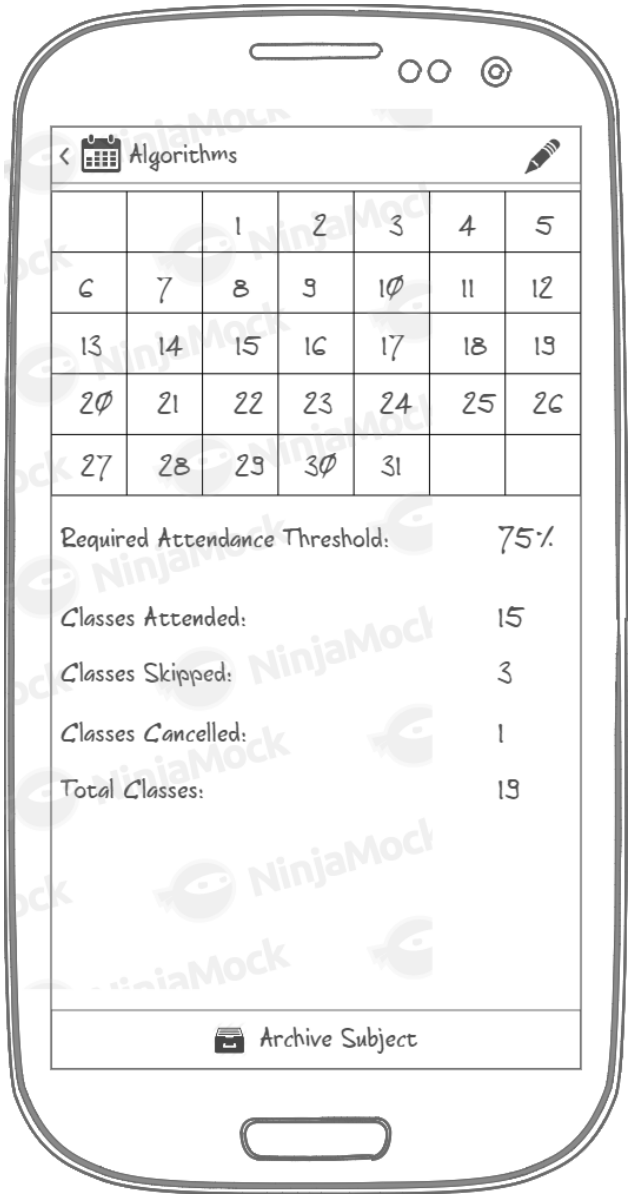
1. Subjects
2. Settings
3. About Developer

Screen 2 - List of Subjects



This displays a list of all the subjects the user has added to the app. Tapping on any subject opens the details view of that particular subject.
Using a spinner on top, the user can select whether they want to see their current subjects or the archived ones.

Screen 3 - Subject Details



This shows a monthly view of user's attendance in a particular subject and a summary of their attendance. The user can tap on any date to edit their attendance for that date.
An action-button at the top allows the user to edit details for that subject.
The button at the bottom allows the user to archive the subject.

Screen 4 - Add a subject

This is the screen the user is presented with when they want to add a new subject. The user can provide a name and the threshold of minimum required attendance for the new subject.

Screen 5 - Edit subject details

This is similar to the screen for adding a subject, but is displayed when the user wants to edit the details of any existing subject.

Screen 6 - Settings

The settings inside our app are of default minimum attendance threshold and a checkbox as to whether or not we are to count cancelled classes as skipped classes.

Screen 7 - About Developer

This is a simple screen giving some information about me.

Screen 8 - Widget

The widget simply contains 3 buttons, pressing any of them will open an activity which shows a list of current subjects, tapping on one of them will register the action from widget to that corresponding subject in that date.

Key Considerations

How will your app handle data persistence?

I will be using the SQLite and some helper methods for data persistence.

Describe any edge or corner cases in the UX.

Some of the UX edge cases are as under:

1. No current subjects added
2. User trying to edit their attendance for a date in the future

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

1. A library for monthly calendar view (not sure which one yet).

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

I will be using Firebase AdMob to show advertisements in the app.

Next Steps: Required Tasks

Task 1: Project Setup

- Create project in Android Studio
- Add required dependencies to buildscript
- Setup project for Firebase AdMob

Task 2: Creating the data model

Deciding how to store all the data regarding the subjects and how to keep track of the daily attendance.

Task 3: Creating UI for the whole app

Create the UI for all the screens in the application. Just making the things appear as they should and handle their wiring in a later stage.

Task 4: Wire the UI with logic

Write logic for connecting the UI with actions that are to be performed with them.

- Link different lists with their adapters
- Launch correct screens on user interaction

Task 5: Setup backup & restore with Google Drive

- Write code to backup data to Google Drive
- Implement a JobDispatcher *Job* to execute a nightly backup
- Implement restore functionality to check for existing backups and restore them

Task 6: Integrate AdMob

- Insert ad-banner in UI at appropriate position(s)
- Link the ad-banner with Firebase AdMob to show actual ads