

Student Companion

Description

Student Companion is an app to track attendance of the students with ease, and show some useful things (Timetable, Holidays etc.) at one place. With reminders to mark attendance at the particular time make it one less thing to remember. Also with the smart card at the dashboard which tells if attendance is coming lower if the user continuous to miss lecture he/she cannot cross minimum threshold of attendance

Intended User

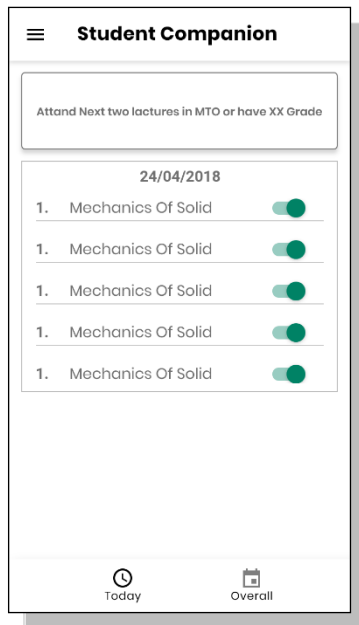
Students who wants to track their daily attendance or for those who has a minimum attendance threshold in their school/high school/collage

With some modifications it can also be targeted to see regularity of the employee of the company

Features

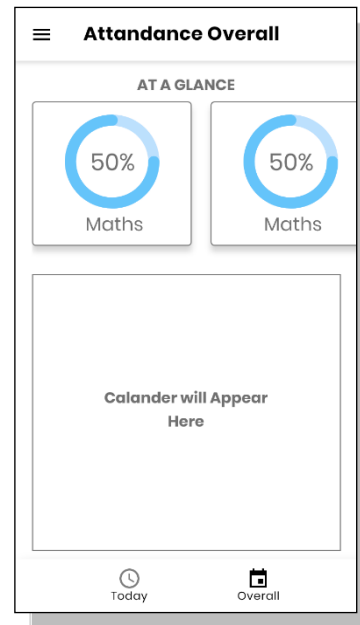
- ✓ App is written completely in Java language
- ✓ App supports RTL Layout switching to support accessibility for RTL language
- ✓ App has all the hardcoded string in strings.xml for easy translation and multiple language support in future
- ✓ Saves attendance and set reminder every day for time to mark attendance
- ✓ Shows time table and holidays
- ✓ Shows summery of attendance as graph
- ✓ Shows smart card whenever necessary (If attendance is coming near threshold it shows how many days more you have attend in order to make up for the days missed in threshold)

User Interface Mocks



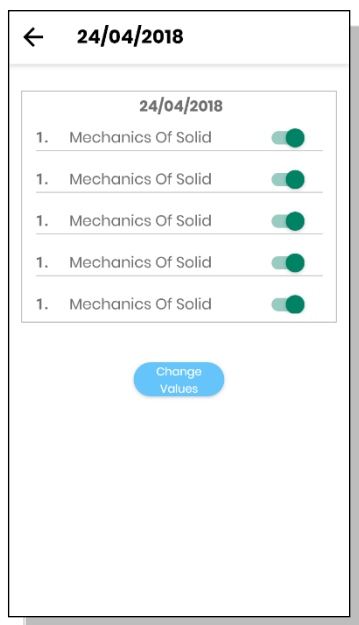
Dashboard/Main Screen

Home Screen layout for showing Smart cards and attendance of today's attendance



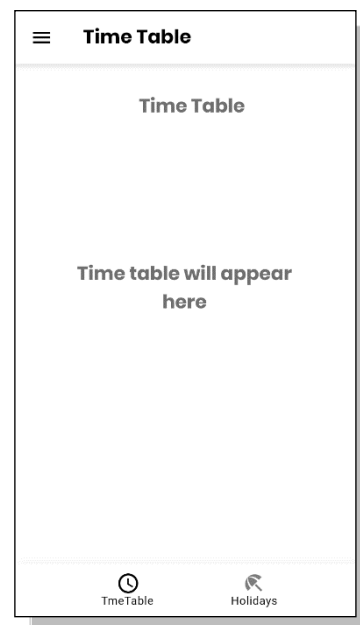
(1) Overall attendance Screen

Overall Screen to show the attendance of each subject and also show the calendar for selecting the dates show the attendance of the particular date's attendance while touching on the calendar takes to the next screen



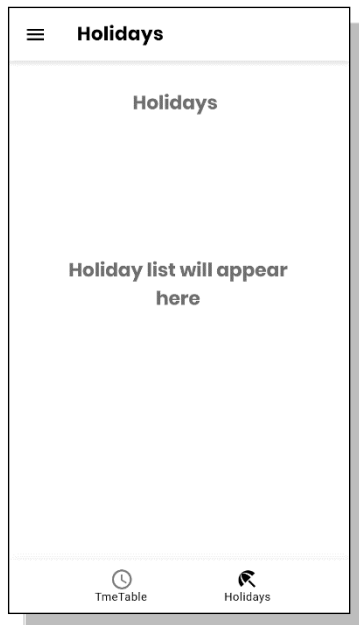
Individual attendance Screen

Shows Individual attendance of particular date and also gives option to change values of that date



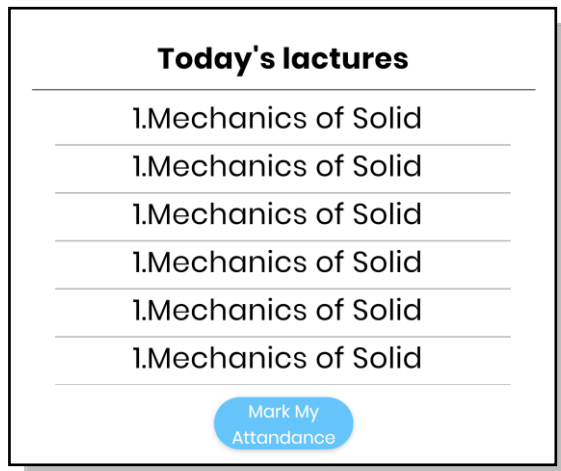
Timetable Screen

Time table screen to show time table for each day's lectures and labs schedule



Holidays Screen

Holidays screen to show holidays in the current year



Widget

Widget to show today's lectures and clicking on button will open Dashboard

For full prototype: <https://xd.adobe.com/view/05ea31c8-2108-40a7-710e-2225f71b37a5-73db/>

Key Considerations

- **How will your app handle data persistence?**
 - App will use Room data persistence library for storing data offline and to execute query in SQLite database
- **Describe any edge or corner cases in the UX.**
 - If navigation drawer is opened in any stage drawer will close first while pressing back button
 - When IndividualAttendanceActivity is opened than user will be redirected to the previous step when pressing back button
- **Describe any libraries you'll be using and share your reasoning for including them.**

Library	Usage	Version
Butterknife	To bind Android views and callbacks to fields and methods	9.0.0-rc1
Room Persistence Library	Create and manage Databases	1.1.1
FitChart	To show attendance graph	1.0
Firebase database	To fetch data from firebase database	16.0.3
Firebase Auth	To authenticate user	16.0.4

- **Describe how you will implement Google Play Services or other external services.**
- Use Firebase Database, Firebase Auth in order to fetch additional data(Timetable, Holidays) from internet

Next Steps: Required Tasks

Task 1: Project Setup

- Create a new project in android studio and add dependencies for the library that are being used in this project.
- Create Empty Activities (MainActivity, IndividualAttendanceActivity/Fragment, OverallAttendanceActivity/Fragment, TimeTableActivitiy/Fragment, HolidaysActivity/Fragment, SettingsActivity/Fragment) along with empty layout files
- Add Navigation Drawer and configure it
- Create/Collect as many as possible drawables, shapes, etc. files that might be used during the project

Task 2: Implement UI for Each Activity and Fragment

- Build UI for MainActivity
- Build UI for IndividualAttendanceActivity
- Build UI for OverallAttendanceActivity
- Build UI for TimeTableActivitiy
- Build UI for HolidaysActivity
- Build UI for SettingsActivity

Task 3: Add Firebase SDK and connect app with Firebase

- Add Firebase SDK
- Connect app with firebase Database/Firestore
- Write code to handle firebase auth and logging in user in order to be able to retrieve data from database

Task 4: Crete and Implement Databases

- Create new Database and table with the help of Room library and create needed components for Room (DAO, Entity etc.).

Task 5: Create and implement logic for showing attendance

- Create logic for showing attendance by getting data from database and calculating needed things and apply it in code
- Feed test data inside database to test the results and verify with actual values

Task 6: Create and implement logic for attendance interactions in app

- Create logic for various action done by user (Added one new attendance Changed attendance from previous days)

- Write a code to respond and refresh the data in both database and on UI while user actions occurred

Task 7: Implement logic to fetch data from Firebase

- Write logic for getting data from firebase for Timetable and Holidays data
- (Optional: Write code to save data locally by getting data from firebase/Firestore and saving it in Room database or offline saving)

Task 8: Create preference of Settings reminder

- In Settings Activity create preference for setting the reminder and implement the logic required for that
- Apply firebase job dispatcher to schedule the job to execute daily at the given time
- Write code to remove current FirebaseJobDispatcher and add new job dispatcher whenever onSharedPreferencesChanged occurs