Time-Table Application

Prof: Bo.Sheng

Team:- 1)Banodaya Cheerla

2) Praveen Rengasamy Thirumeni Palanivelu

Course:- Mobile Application(CS443)

Unlike a paper planner or school diary, this application integrates all areas of our academic life-see homework due and overdue for classes, classes that conflict with your exams, easy to use application. In this world of chaos, we wanted to make everyone's life simple by creating an Android application for setting up peoples time table in a user-friendly manner and set up their goals for the week, and be productive in their day-to-day life. Class Time table is the perfect companion for school, college, or university. Keep track of classes and add events to your week's schedule with ease.

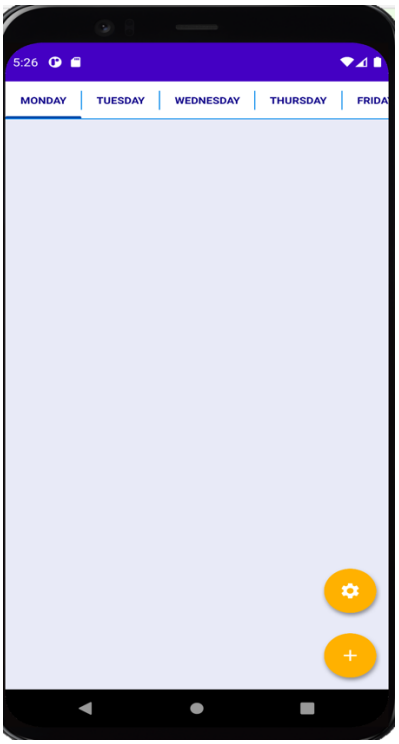
Project Statement:-

We are developing an android timetable application for users to set their timetable in one week in advance to be organized and make life easy for a user. The basic idea of this application is to keep eye on our day-to-day activities in a decent manner. For instance, a user can set his daily activities in beforehand itself and if he/she wants to repeat for the rest of the week user is allowed to do that. Not the only user can repeat the same timetable every week, the user also has so many preferences like he/she can set a different timetable every day or every week depends on him/her. Rounding up to all, this application has various features and as said this application is user-friendly. As being a student we know-how its onerous to manage our schedule with a lot of other principal things in this routine life. As we know student life is unpredictable we feel somehow we can manage a user's daily activities with this application to make a trouble-free schedule for the rest of his/her day. From this application, we can say that a student and an employee will be more benefited from this application, and also it is useful for setting up meetings with employers in any field regardless of their work. There are a various number of this application out there, this application was developed catching bits and pieces from a few apps out there in the android market.

Application design:-

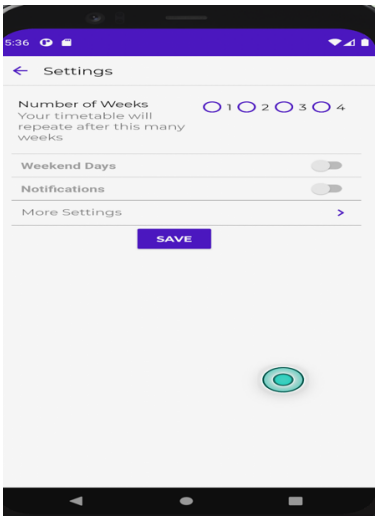
This section first describes the high-level design of this application and later on describes the modules which are used to design this android application. So now let's see the design of this application and the explanation along with it.

i)Home Page:-



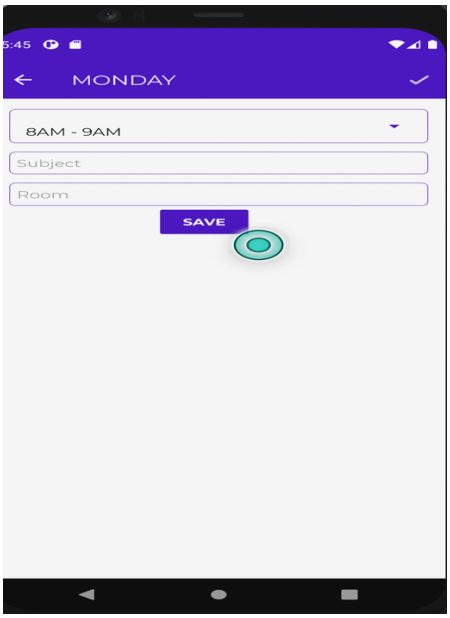
This is the Home page, when a user opens this app this page is shown initially. Where in this page you can see all days of the week , setting module, an editor module. Using that editor module user can add his/her classes or events eventually in this application. And setting module has other preferences which will be discussed in the following sections.

ii) Setting Module:-



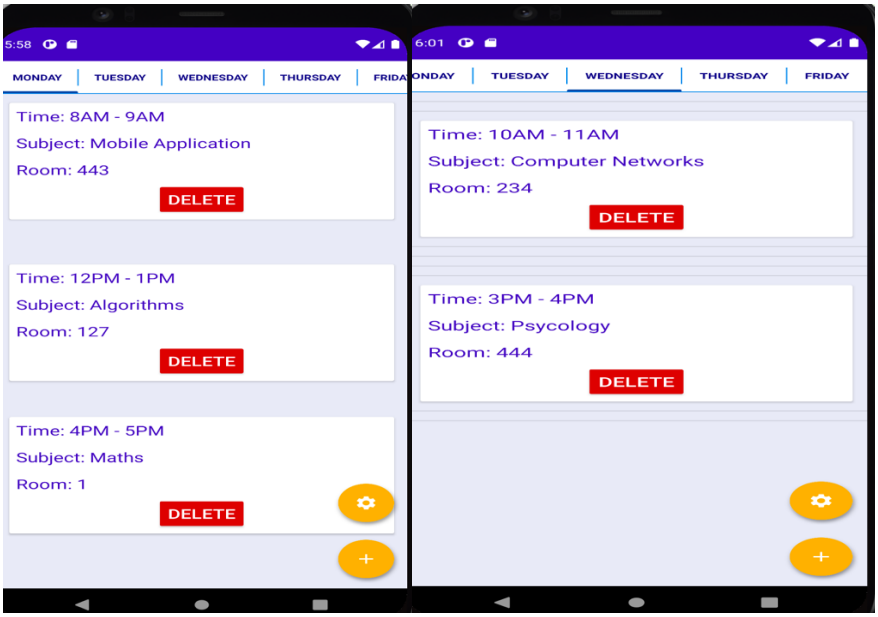
This page shows up when the user clicks on the setting widget. As you can see there is number of weeks where your timetable will repeat after this many weeks. And also push buttons for weekend days and notifications.

iii)Editor Module:-



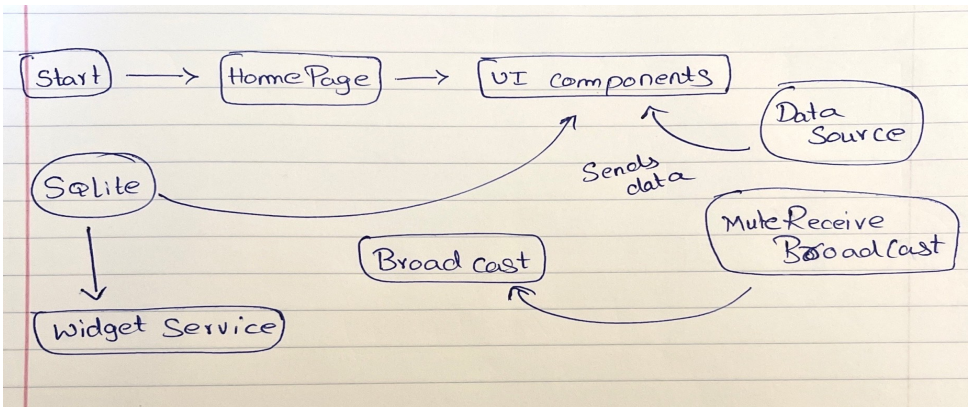
This page is where the user can add his timetable for class depending on timing and selecting subjects and room for that particular task.

iv)Page after adding classes:-



User can add his classes like this way and also user can delete that course if he doesn't need that course anymore. Not only classes an employer can set up his meeting also depending on what time he chooses.

So in this application, we have been using different modules such as activity, adaptor, and fragment. As we can see in the activity section each module has its own set of rules, along with that this application is using SQLite database to store, update, delete and retrieve the information of the user anytime. In this application, the user can also delete the courses, update them whenever the user needs them. In android, we use an adaptor as a bridge between components of the UI and data-driven components that help to fill the data in the UI component. And that information holds the data and sends the data to the adaptor and then takes the data and shows it at the frontmost application. Coming to fragments, for each day module in this application we used fragments in the same activity to update and add classes if necessary.



In this application, we are using a service, broadcast receiver, and a content provider to monitor the workload of this application when the user is online. As we know Broadcast receiver sends or receives messages from the android system. These broadcast messages are sent when an event has occurred. So in this application, we have used MuteReceiverBroadcast which extends Broadcast for alerting the user when it is due. Talking about service, we Used Intent service as it is an extension of service that handles asynchronous requests on demand of the user. Basically, this Intent service will receive the Intents, launch a worker thread, and stop the service as appropriate, and are handled on a single work theard. Last but not least, we are also using a Content provider for this application to manage access to the data stored by users and the way they are shared between the source. Finally, in Activity MainActivity is extend to fragment activity to organize all the components in a proper well-mannered application.

Taking Activity, service, Content provider, and Broadcast receiver into consideration tells how much work does this application doing and keeps track of all the processes of this application. All these components here in this app are helpful to one other are grouped in such a way that without activity there are no fragments involved and without service, we cannot handle the asynchronous requests by a user in this app. Beyond the Content provider, we cannot manage the data transmission, and leaving out the Broadcast receiver, we cannot listen to system-wide cast events. One more thing to add up is about Database, we used SQLite database for handling all those data-oriented stuff (as we known android room Is the advanced version of SQLite). These all are

handled in such a way that everything is accessible to each function and avoiding any bug in the future.

Application Implementation and Evaluation:-

UI:-

- MainActivity:- The name itself says as the main activity, which is the first screen to appear in this application where it creates drawers and fragments.
- The remaining activities in this app are extended to AppCompactActivity
- Editor Activity is a widget where the user has the option to click on it and add his classes to the timetable.
- Setting Activity is also a widget where the user has the freedom to filter out more options in his/her app.
- In projects activity, this class takes the information from the user and. saves the data and creates a table as well in the database.
- ProjectShow Activity is a class where the user can see what he updated in this application.
- Attendance Activity is a class that tells you how many classes you attended and OnsaveToDatabse() stores this data in the database.

Fragments:-

- We have created fragments for each weekday separately because all these functions might be overlapped.
- We are creating variable sqlArray[] to store the data of the user where he/she adds classes/meetings in their timetable.
- We are listening to the listview() by calling setOnItemLongClickListener().
- We are using the onCreateLoader() function to load the data when user clicks on any fragment.
- We are repeating this method in every fragment so that the user can set his timetable in a friendly manner.

Adapter:-

As we know why we use adapters because it helps us to fill the data in the UI components.

- MyAdapter is a class where you can get the view from getview() method call and listen from setOnClickListener() and updates the query in the database.
- MyattAdapter is a platform between UI components and data sources.

Other Classes:-

- MuteReceiveBroadCast is a class that extends BroadCastRecevier where we are using this for the audio manager to manage alarms.
- Timetable Provider is a class where it manages the data and store it in the database and have access to what user types.
- WidgetService is a class where it handles all the asynchronous requests which are made by the user.
- Widget class is extended to AppWidgetProvider where this widget gets periodic updates about the applications.
- Utility class is just another helper class that provides methods that are common across the application where we do DateTime manipulation.
- This Helper class extends SQLiteOpenHelper that manages database creation and also manages the version control.
- Contract class helps applications to interact with URI's.

Experience and Thoughts:-

While working on this project we felt team building is very important and how good we put our ideas in this and implement them so that we don't do any mistakes. Like every developer faces problems we also underwent some issues like we were not able to update or delete the information in a database, so we somehow managed to store the information correctly in the database after crossing so many hurdles. The most challenging phase was SQLite(background experience using SQLite was null), again learning from basic to implementing this database in this application was something we are proud of. Not only SQLite we have learned so many functionalities like provider and broadcast receiver which we are new. All in all this project gave us confidence that we are capable of learning things and keep developing those newly acquired skills.