

MINI PROJECT

(2021-22)

“Time Tracker”

Project Report



Institute of Engineering & Technology

Submitted By -

Vasu Agarwal (191500900)

Prateek Gupta (191500587)

Nikhil Gupta (191500498)

Under the Supervision of

Mr. Piyush Vashistha

Technical Trainer

Department of Computer Engineering & Applications



Department of Computer Engineering and Applications
GLA University, 17 km. Stone NH#2, Mathura-Delhi Road,
Chaumuha, Mathura – 281406 U.P (India)

Declaration

I the undersigned solemnly declare that the project report **TIME TRACKER** is based on my team work carried out during the course of our study under the supervision of **Mr. Piyush Vashishtha, Technical Trainer, Department of Computer Science, GLA UNIVERSITY, MATHURA**. I assert the statements made and conclusions drawn are an outcome of my team research work. I further certify that the work contained in the report is original and has been done by me and my team under the general supervision of my supervisor. We have followed the guidelines provided by the university in writing the report. Whenever we have used materials like data, text from other sources, we have given due credit to them in the text of the report and giving their details in the references. The work has not been submitted to any other Institution for any other degree in this university or any other University of India or abroad. The contents of this project report, in full or in parts, have not been submitted to any other Institute or University for the award of any degree.

Sign: *VasuAgarwal*

Name of Candidate: Vasu Agarwal

University Roll No.:191500900

Sign: *PrateekGupta*

Name of Candidate: Prateek Gupta

University Roll No.:191500587

Sign: *NikhilGupta*

Name of Candidate: Nikhil Gupta

University Roll No.:191500498



Department of Computer Engineering and Applications
GLA University, 17 km. Stone NH#2, Mathura-Delhi Road,
Chaumuha, Mathura – 281406 U.P (India)

Certificate

This is to certify that the project entitled “**Time Tracker**”, carried out in Mini Project is a group work by Vasu Agarwal, Prateek Gupta and Nikhil Gupta is submitted in partial fulfillment of the requirements for the award of the degree Bachelor of Technology (Computer Science & Engineering).

Signature of Supervisor:

Name of Supervisor: Mr. Piyush Vashistha

Date:

Training Certificates

- **Vasu Agarwal**

Certificate in Android with Core Java

www.netcamp.in

NETCAMP®



NSPL/24/21/639

This is to certify that

Vasu Agarwal

has demonstrated an understanding of the solutions and technologies covered in the Certificate in Android with Core Java course and has successfully completed the training programme held during 15th June, 2021 - 31st August, 2021

A handwritten signature in black ink, appearing to read "SANTU", with a stylized flourish at the end.

Santu Purkait
Director
Netcamp Solutions Private Limited

- **Prateek Gupta**



- **Nikhil Gupta**

Certificate in Android with Core Java

www.netcamp.in

NETCAMP®



NSPL/24/21/911

This is to certify that

Nikhil Gupta

has demonstrated an understanding of the solutions and technologies covered in the Certificate in Android with Core Java course and has successfully completed the training programme held during 15th June, 2021 - 31st August, 2021

A handwritten signature in black ink, appearing to read "SANTU".

Santu Purkait
Director
Netcamp Solutions Private Limited



Department of Computer Engineering and Applications
GLA University, 17 km. Stone NH#2, Mathura-Delhi Road,
Chaumuha, Mathura – 281406 U.P (India)

ACKNOWLEDGEMENT

We would like to express our sincere thanks to Mr. Piyush Vashistha for his valuable guidance and support in completing our project. Your valuable guidance and suggestions helped us in various phases of the completion of this project. We will always be thankful to you in this regard.

He has been helping us since Day 1 in this project. He provided us with the roadmap, the basic guidelines explaining on how to work on the project. He has been conducting regular meeting to check the progress of the project and providing us with the resources related to the project. Without his help, we wouldn't have been able to complete this project.

Finally, we would like thank our parents and friends, without them this project would not have been completed.

Thanking You

Sign: *VasuAgarwal*

Name of Candidate: Vasu Agarwal

University Roll No.:191500900

Sign: *PrateekGupta*

Name of Candidate: Prateek Gupta

University Roll No.:191500587

Sign: *NikhilGupta*

Name of Candidate: Nikhil Gupta

University Roll No.:191500498

ABSTRACT

In this project, we are creating an android application, which we named ‘Time Tracker’ which basically manages time for each task user enters in it. The purpose of our app ‘Time Tracker’ is to manage time for different tasks to help students to schedule their day. User can see all the tasks he/she had entered on the front page in the form of recycler view which fetches data from database. For Database,

We have used Room Database in our application. Room Database is used to store data in user’s phone internal storage. Start time which user entered at that time alarm will ring with the help of Broadcast receiver class.

And with the help of Pending Intent the alarm will ring on the exact time that the user had set. Our Project will provide user a better UI experience.

Android App ecosystem is diverse and is changing people’s life all over the world. Android users are expected to increase because of the advance changes of the operating system and the way it deals with issues and compatibility with other mobile devices.

CONTENTS

Cover Page.....	i
Declaration	ii
Certificate	iii
Training Certificate.....	iv
Acknowledgement	vii
Abstract	viii
Contents.....	ix
List Of figures.....	xi
List Of tables	xii
Chapter 1Introduction	1
• 1.1 Context.....	1
• 1.2 Motivation	1
• 1.3 Objective.....	1
• 1.4 Sources	2
Chapter 2 Software Requirement Analysis	3
• 2.1 Impact of Online Learning in Daily Life	3
• 2.2 Problem Statement... ..	3
• 2.3 Hardware and Software Requirements	3
• 2.4 Modules and Functionalities	5

• 2.5 Time Tracker on Android Application.....,,,,,	7
Chapter 3 Software Design... ..	8
• 3.2 Data Flow Diagram.....	9
Chapter 4 Technology Used... ..	9
• 4.1 Android.....	10
• 4.2 Version of Android	11
• 4.3 Tools and Languages	12
• 4.4 Basic Terminology.....	13
Chapter 5 Implementation.....	15
• 5.1 Implementation of Time Tracker.....	15
Chapter 6 Testing.....	17
Chapter 7 Conclusion.....	18
References	19

LIST OF FIGURES

1. Data Flow Diagram.....	9
2. Android Kit-Kat.....	11

LIST OF TABLES

1. Version of Android... ..	11
2. Testing of Time Tracker.....	17

CHAPTER 1

INTRODUCTION

1.1 CONTEXT

This Android Application “Time Tracker” is the name of our mini-project for the fulfilment of the degree of Bachelor of Technology in Computer Science and Engineering at GLA University, Mathura supervised by Mr. Piyush Vashistha. This project has been completed in approximately two months and has been executed in modules, meetings are done in order to check progress and solve issues.

1.2 MOTIVATION

Using a time management app is one of the easiest ways to get more of something we wish we all had: productivity. From to-do lists and calendar reminders to shopping lists and meetings, it is easy to get overwhelmed with the number of places we keep reminders and tasks. Thanks to time management apps, you can easily keep track of everything in one place, not only helping you reach goals faster, but also saving you time in the long run.

While time management apps aren’t new, some of the technology and accessibility are, giving new meaning to the phrase, “There’s an app for that.” Whether you are a manager hoping to assign employees tasks or a mom who manages the calendar for the entire family, there is a time management app out there with functionality built with you in mind. Here we rounded up the best time management apps available, all with the goal of making you more productive, so that you spend less time planning and more time doing (especially if that means more time for yourself).

1.3 OBJECTIVE

As all of us know that time is money so to save time and use it effectively is the Main Objective of this project. This app manages time so that the student may not need to waste his/her time useless work. A lot of people believe they cannot reach their dreams, travel to exotic locations, land their dream jobs, finish their projects before the deadline, get enough sleep every night, and spend enough time with loved ones because they don’t have enough time. It’s not limited time that’s the problem, it’s **bad time management**. Users want better UI and app that looks attractive, so to provide users a good experience in studying and not get bored while seeing UI, Improve your performance, Produce better work, Deliver work on time, Reduce your stress, Boost your confidence. By providing attractive UI users will get attracted towards it.

1.4 SOURCES

The Source Code of our mini-project will be available on the following link
https://github.com/Vasu-Agarwal-2509/Time_Tracker

CHAPTER -2

SOFTWARE REQUIREMENT ANALYSIS

2.1 IMPACT OF TIME MANAGEMENT ON DAILY LIFE

Time management application helps you prioritize your tasks so that you ensure you have enough time available to complete every project. The quality of your work increases when you're not rushing to complete it ahead of a fast approaching deadline. This app improves the users day and user can use its time more efficiently than previous.

This application is used where user is self motivated in managing its time and searching for time management application. It can be used where user passes a lot of time in unnecessary tasks.

2.2 PROBLEM STATEMENT

Time management plays a vital role in improving student's academic performance and achievements. Some observed problems in UUM in the academic activities of students may owe their causes largely to time management. This is when the students do not manage their time properly. Commonly, they may not come or may delay in coming to school. There may be delay in planning for academic activities postponement of time to do assignment and reading for tests and examination. They may be facing interruptions during the lectures hours like receiving lengthy calls, pingings, social networks and mingle with friends. Besides, students have join the extra activities such as event activities, co-curriculum activities, club activities and others. This may be one of the reason that cause student poor in time management.

This project is implemented on Android to facilitate easier access on a popular medium. The project uses JAVA language to provide the app its functionality. Room Database is used as storage where all data is stored. The GUI components are developed using XML.

Time Tracker is an inexpensive, efficient and comfortable way for students to easily manage their time.

2.3 HARDWARE AND SOFTWARE REQUIREMENTS

Hardware Requirement

- Processor: Intel i5
- Operating System: Any Operating System
- RAM: 4 GB (or higher)
- Hard disk: 256GB

Software Requirement

- Software used: Android Studio
- Language used: Java, XML
- Database: Room Database
- User Interface Design: Android Application

2.4 MODULES AND FUNCTIONALITIES

1. First module contains activity that contains a list of all tasks and its starting and ending time and also a floating button to add task.
2. Second module contains a dialog box which will be seen after clicking on floating button. There are three views 2 textview and 1 edittext. It also contains add button by clicking on which data gets added to database.
3. On clicking on textview it opens time picker from which we can pick time.
4. When we click on add button alarm is set on start time that we have entered which is received by broadcast receiver class.

2.5 TIME TRACKER ON ANDROID APPLICATION

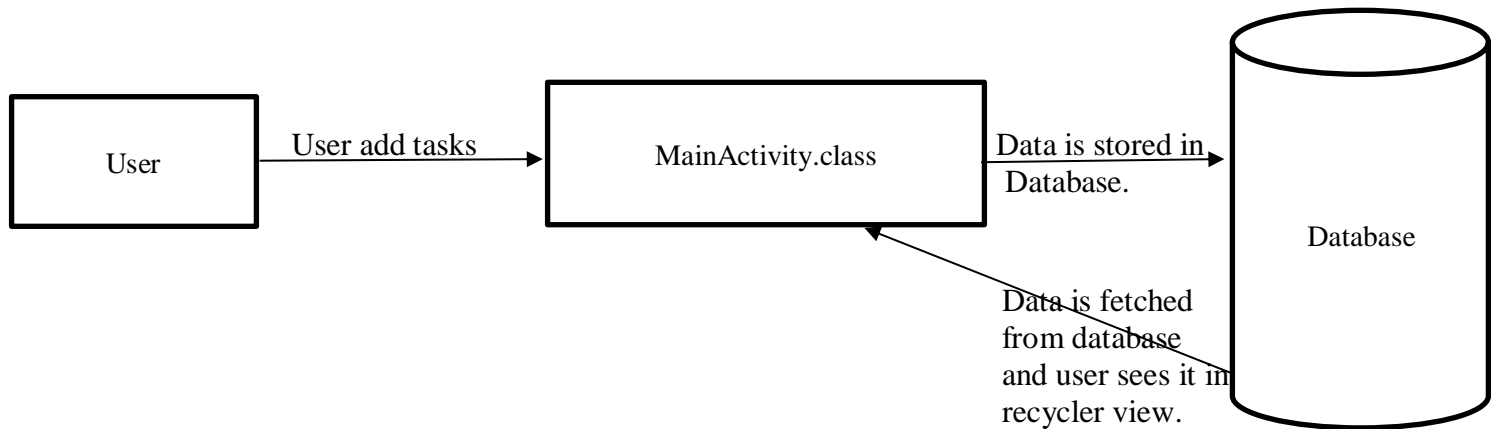
Project is on time management application. The purpose of our app 'Time Tracker' is to manage time for different tasks to help students to schedule their day. User can see all the tasks he/she had entered on the front page in the form of recycler view which fetches data from database. For Database,

We have used Room Database in our application. Room Database is used to store data in user's phone internal storage. Start time which user entered at that time alarm will ring with the help of Broadcast receiver class.

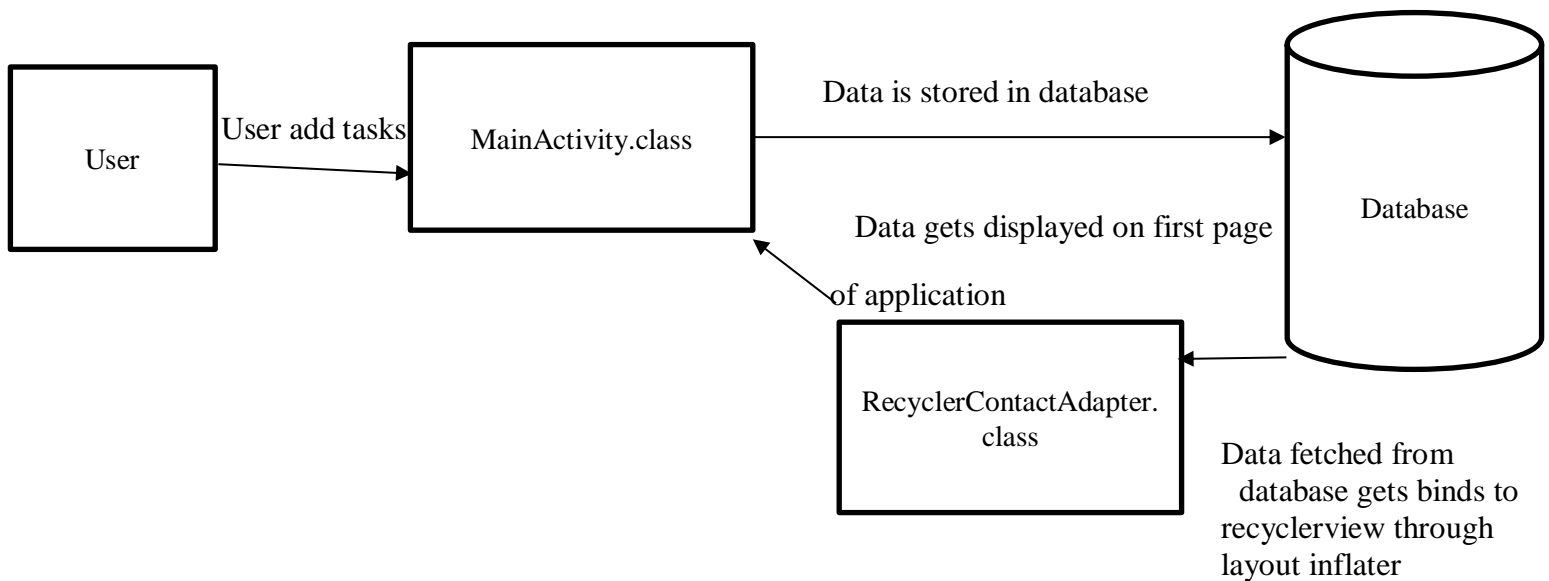
And with the help of Pending Intent the alarm will ring on the exact time that the user had set. Our Project will provide user a better UI experience.

CHAPTER-3 SOFTWARE DESIGN

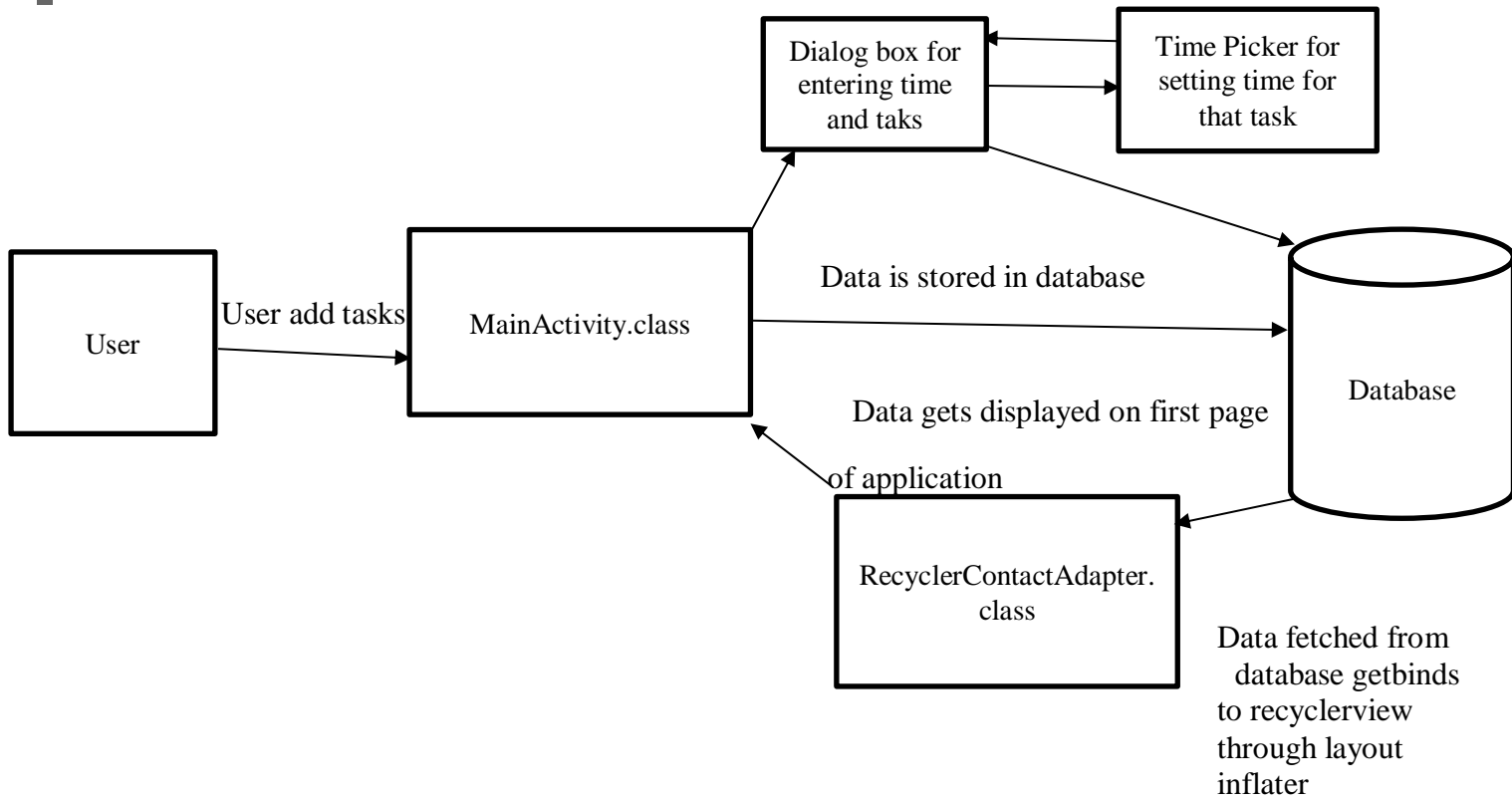
0 Level DFD: -



1 level DFD: -



2 level DFD



CHAPTER-4

TECHNOLOGY USED

ANDROID

Android is a Linux-based operating system designed primarily for touch screen devices such as smart phone tablets and computers. Released in 2008, is now owned by Google. So android is a operating system like Windows, Ubuntu and Mac OS and a lot number of devices use Android these days like mobile phones, watches, laptop and television. So, we also created an android application “Time Tracker”, a store of courses. Play Store is a market place for all the Android Apps. So we need to know what basically an android app is. An Android app is software running on a Android Platform. So this can be concluded that like all the software it is a combination of Backend and Frontend. Backend to design the logical parts of the app, for the functionality whereas Front End to develop the User Interface. And to implement the various parts of the android app, we require a number of tools and technologies which will come into picture. But first it would be great to see the three different types of Android Apps: -

4.1 VERSION OF ANDROID

Each year Android releases a new version with better features, better security and better User Interface experience and a new symbol. Here is the table of list of versions



Figure-5: Android KitKat

Code name	Version number	Initial release date
(No codename)	1.0	September 23, 2008
Petit Four	1.1	February 9, 2009
Cupcake	1.5	April 27, 2009
Donut	1.6	September 15, 2009
Eclair	2.0 - 2.1	October 26, 2009
Froyo	2.2 - 2.2.3	May 20, 2010
Gingerbread	2.3 - 2.3.7	December 6, 2010
Honeycomb	3.0 - 3.2.6	February 22, 2011
Ice Cream Sandwich	4.0 - 4.0.4	October 18, 2011
Jelly Bean	4.1 - 4.3.1	July 9, 2012
KitKat	4.4 - 4.4.4	October 31, 2013
Lollipop	5.0 - 5.1.1	November 12, 2014
Marshmallow	6.0 - 6.0.1	October 5, 2015
Nougat	7.0 - 7.1.2	August 22, 2016
Oreo	8.0 - 8.1	August 21, 2017
Pie	9.0	August 6, 2018

Table -1: Versions of Android

4.2 TOOLS AND LANGUAGES

Tools used to build the Android App are: -

- **Android Studio:** Android Studio is an environment that help us create and edit Android applications. It is the official IDE for Android App Development. It has IntelliJ's powerful code editor and developer tools and various features that enhance productivity while developing apps.
- **Software Development Kit (SDK):** Android Studio requires a collection of libraries and data which are there SDK.

Languages used in building an Android Application are classified as per the Front End and Back End. For designing the Front End of an application, we have used XML and for the Back End we have used JAVA.

- **XML:** An XML file is an extensible markup language file, and it is used to structure data for storage and transport. In an XML file, there are both tags and text. The tags provide the structure to the data. XML is a markup language, which means it is a computer language that uses tags to describe components in a file.
- **JAVA:** Java is a programming language and a platform. Java is a high level, robust, object-oriented and secure programming language.

Java was developed by *Sun Microsystems* (which is now the subsidiary of Oracle) in the year 1995. *James Gosling* is known as the father of Java. Before Java, its name was *Oak*. Since Oak was already a registered company, so James Gosling and his team changed the name from Oak to Java.

4.3 BASIC TERMINOLOGY

- **Layout:** Layout is the parent of view. It arranges all the views in a proper manner on the screen.

- **Activity**: An activity can be referred as your device's screen which you see. User can place UI elements in any order in the created window of user's choice.
- **View**: A view is an UI which occupies rectangular area on the screen to draw and handle user events.
- **Emulator**: An emulator is an Android virtual device through which you can select the target Android version or platform to run and test your developed application.
- **Manifest file**: Manifest file acts as a metadata for every application. This file contains all the essential information about the application like app icon, app name, launcher activity, and required permissions etc.
- **API**: Short for Application Programming Interface. APIs are functions that developers can call on to access specific features by calling upon programs, code, and services that others have written. For example, if a developer wants to draw a button on the screen, she can insert a small bit of code that says "draw this kind of button, with this color and size and style, at this location" instead of dozens of lines of code that tells the graphics processor, in detail, exactly how to draw a button. If the application wants your location, it can use the location API to "get the device's location" and let Google's code handle the rest, instead of requiring the developer to build an entire location service from scratch just for her own app. There are thousands of APIs in Android, covering everything from drawing interface elements, to the cameras, to location access, to accessing storage, to 3D graphics (see: OpenGL ES) and much more.
- **Intent**: Intents are an essential part of the Android ecosystem. They are used to express an action to be performed. Intents allow you to interact with components from the same applications as well as with components contributed by other applications. It can be classified into implicit and explicit intents.

- **Implicit intent:** It does not name a specific component, but instead declare a general action to perform, which allows a component from another app to handle it.
- **Explicit Intent:** It specifies the component to start by name. You'll typically use an explicit intent to start a component in your own app, because you know the class name of the activity or service you want to start.
- **APK:** Short for "Android application package." The extension used in Android app installation files (e.g., app.apk). Similar in nature to an EXE file on Windows.
- **SDK:** Short for "Software Development Kit." As it pertains to Android, the SDK is a set of tools such as code libraries, a debugger, and a handset emulator that can be run on Windows, Mac, or Linux to facilitate the creation of Android apps by developers. While the SDK is generally intended for use by developers, end users can install the software on their home computer to execute ADB and Fast boot commands.
- **Action Bar:** The action bar is an important design element, usually at the top of each screen in an app that provides a consistent familiar look between Android apps. It is used to provide better user interaction and experience by supporting easy navigation through tabs and drop-down lists.
- **Navigation bar:** Android Navigation Drawer is a sliding left menu that is used to display the important links in the application. Navigation drawer makes it easy to navigate to and fro between those links. It's not visible by default and it needs to be opened either by sliding from left or clicking its icon in the Action Bar.
- **Fragment:** A Fragment represents a behavior or a portion of user interface in a Fragment Activity. You can combine multiple fragments in a single activity to build a multi-pane UI and reuse a fragment in multiple activities.
- **Firebase** is a Backend-as-a-Service (Baas). It provides developers with a variety of tools and services to help them develop quality apps, grow their user base, and earn profit. It is built on Google's infrastructure. Firebase is categorized as a NoSQL database program,

which stores data in JSON-like documents. Firebase has three core services: a real-time database, user authentication and hosting. With the Firebase iOS SDK, you can use these services to create apps without writing any server code.

JSON stands for JavaScript Object Notation. It is an independent data exchange format and is the best alternative for XML. JSON is used for data interchange (posting and retrieving) from the server. Hence knowing the syntax and its usability is important. JSON is the best alternative for XML and its more readable by human

CHAPTER -5

IMPLEMENTATION

By making an app conceptual design with functional flow diagrams is the best way to communicate your vision to the mobile app developer. Making your concept clear to the developer is probably the most important factor in successful android app development.

5.1 Implementation of the Time Tracker:

Implementation of Time Tracker is taken place in various modules. First, we have presented a task viewer page to the user then where user can see all tasks.

Step to be followed to develop the app:

1. First module contains activity that contains a list of all tasks and its starting and ending time and also a floating button to add task.
2. Second module contains a dialog box which will be seen after clicking on floating button. There are three views 2 textview and 1 edittext. It also contains add button by clicking on which data gets added to database.
3. On clicking on textview it opens time picker from which we can pick time.
4. When we click on add button alarm is set on start time that we have entered which is received by broadcast receiver class.

CHAPTER - 6

TESTING

When source code is generated, it is necessary to test the application that we are going to present to user. Testing is an important phase of development.

The Android framework includes an integrated testing framework that helps you test all aspects of your application and the SDK tools include tools for setting up and running test applications. Whether you are working in Eclipse with ADT or working from the command line, the SDK tools help you set up and run your tests within an emulator or the device you are targeting.

CHAPTER -7

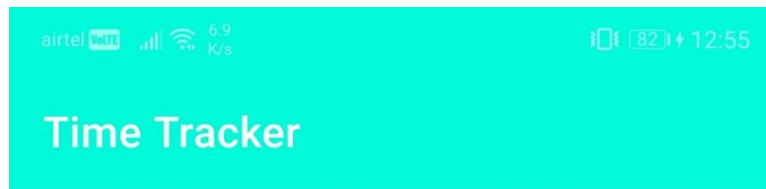
CONCLUSION

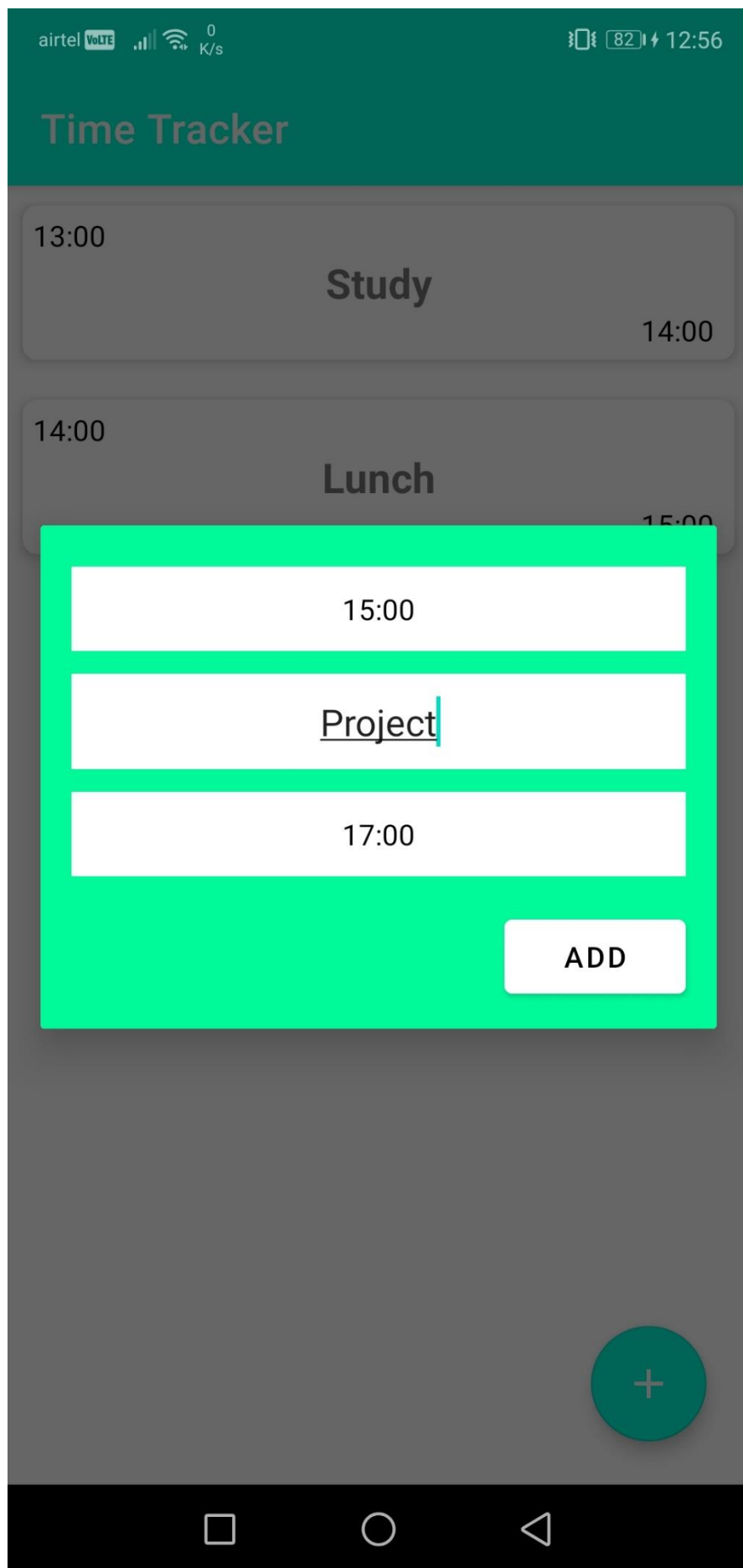
This project has brought a positive impact in the lives of students and working professionals. It helped user in managing their time and effectively use their day. Users are more relaxed and stress free. After using this app users are productive in their task and complete everything on time without any delay.

It has become easy for students to study for their and academics and chill out with their friends as they utilize their time in right manner. This project also helped users in reminding of tasks that they forget as it contains alarm service which rings at the start time of that task.

CHAPTER-8

IMPLEMENTATION AND USER-INTERFACE





The screenshot shows an Android application titled "Time Tracker". The app has a dark green header bar with the title. Below the header is a grey background. A red rectangular box highlights a form with three input fields: "Start Time", "Enter Task", and "End Time". To the right of these fields is a white button with the text "ADD". At the bottom right of the screen, there is a circular green button with a white plus sign. The status bar at the top shows "airtel VoLTE", signal strength, Wi-Fi, 6.2 K/s, battery level at 82%, and the time 12:55. The bottom navigation bar shows the standard Android icons: a square, a circle, and a triangle.

airtel VoLTE 6.2 K/s 82% 12:55

Time Tracker

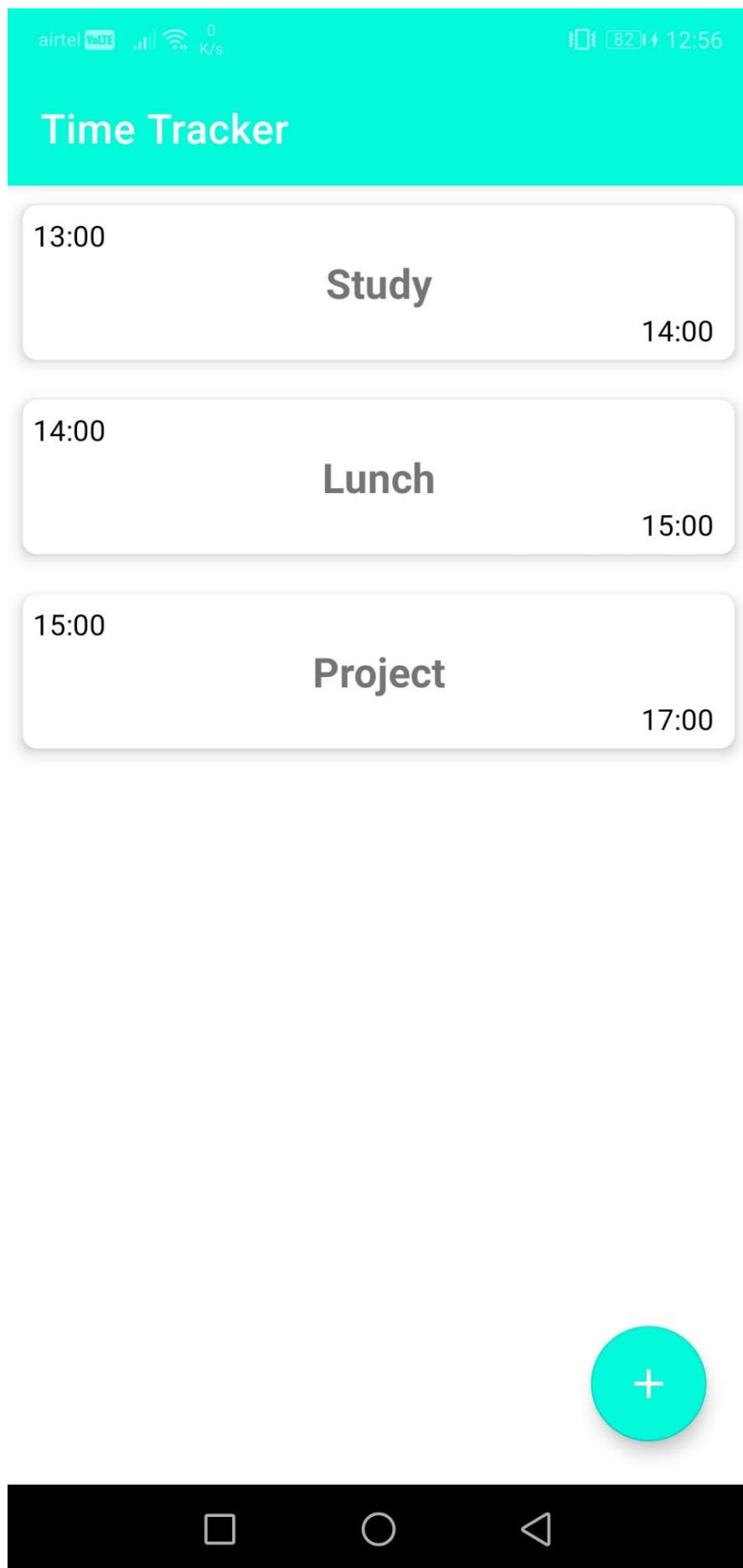
Start Time

Enter Task

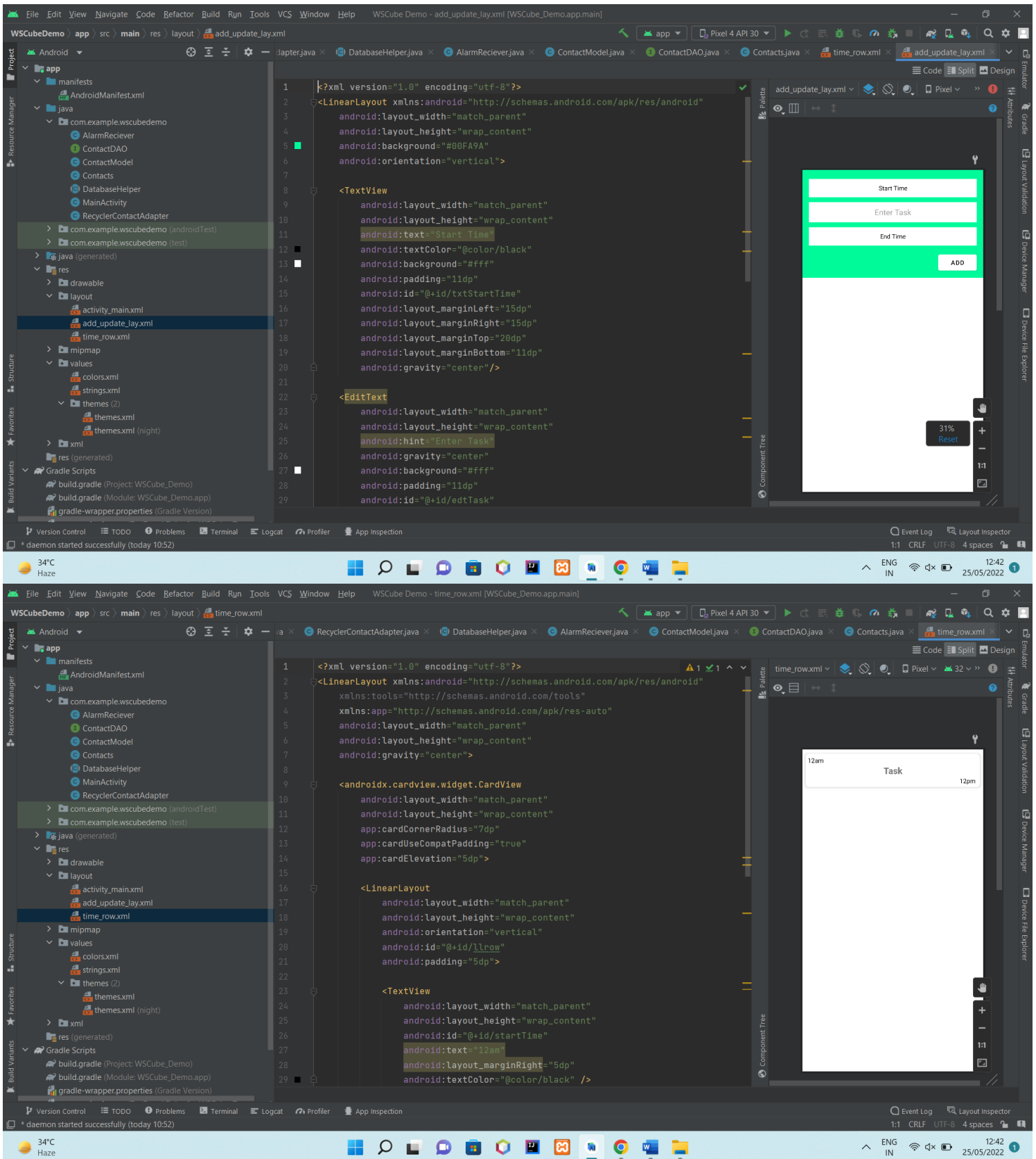
End Time

ADD

+



Android Application – Time Tracker



Android Application – Time Tracker

The image displays two screenshots of the Android Studio IDE, showing the development of a Time Tracker application.

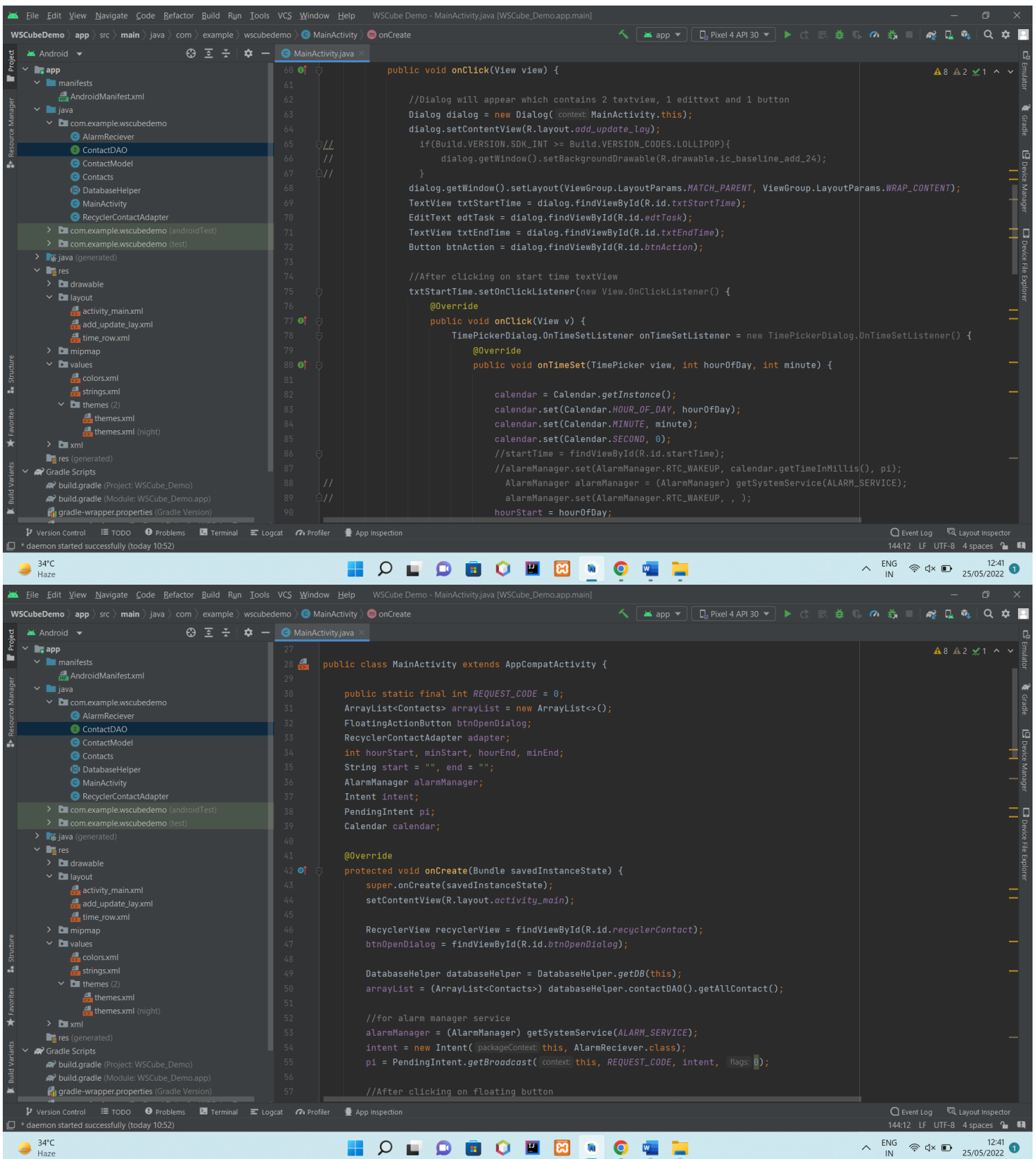
Top Screenshot: The IDE shows the `Contacts.java` file. The code defines an `@Entity` with a table name `"contact_table"` and a primary key `id`. It includes columns `start`, `task`, and `end`. The `Contacts` class implements getters and setters for these fields.

```
7 @Entity(tableName = "contact_table")
8 public class Contacts {
9
10     @PrimaryKey(autoGenerate = true)
11     protected int id;
12
13     @ColumnInfo(name = "start")
14     protected String start;
15
16     @ColumnInfo(name = "task")
17     protected String task;
18
19     @ColumnInfo(name = "end")
20     protected String end;
21
22     Contacts(String start, String task, String end){
23         this.start = start;
24         this.task = task;
25         this.end = end;
26     }
27
28     public String getStart() { return start; }
29
30     public void setStart(String start) { this.start = start; }
31
32     public String getTask() { return task; }
33
34     public void setTask(String task) { this.task = task; }
35
36     public String getEnd() { return end; }
```

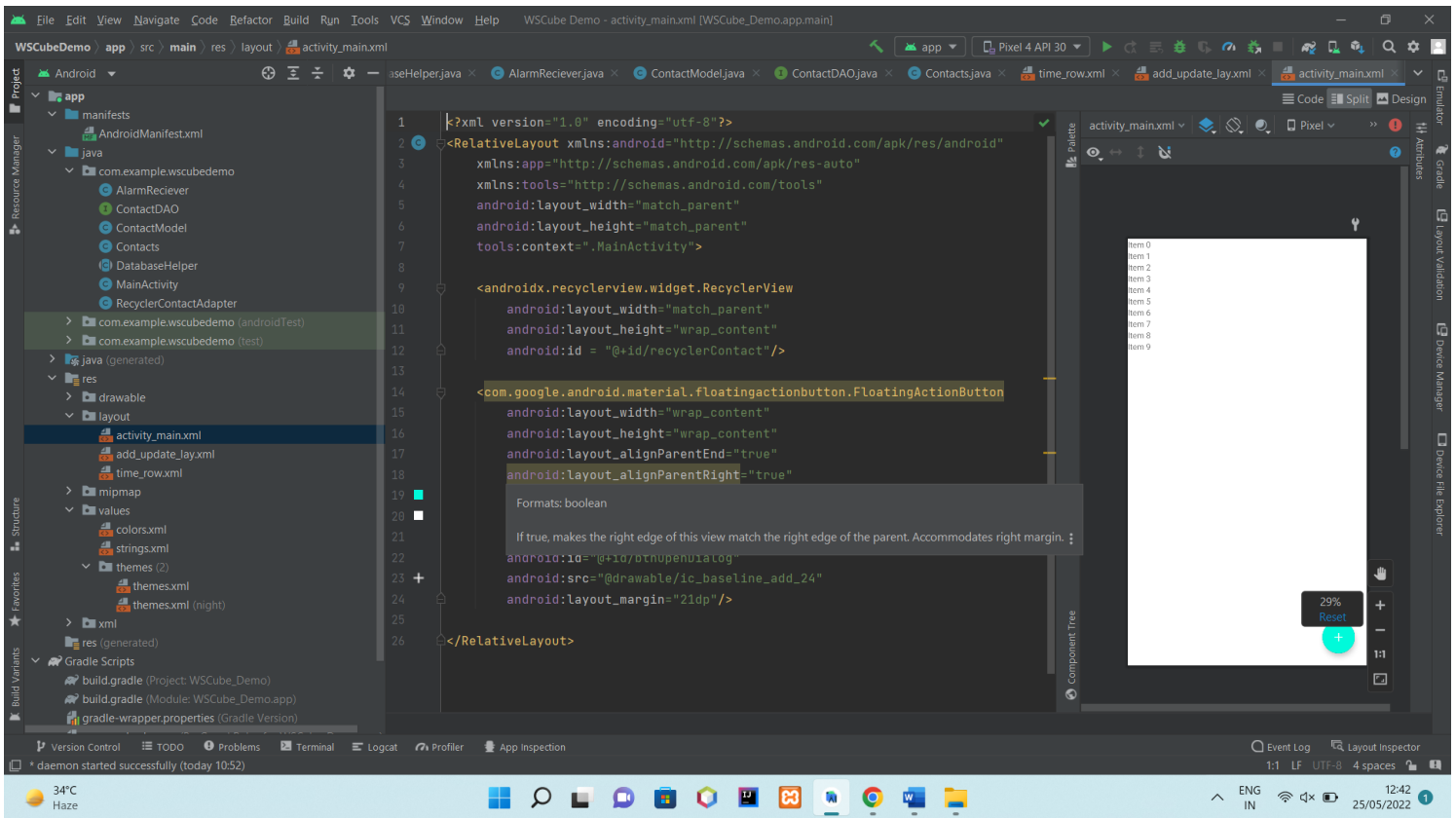
Bottom Screenshot: The IDE shows the `RecyclerViewAdapter.java` file. The code implements the `onClickListener` for the RecyclerView, which displays a dialog for editing contact details. It also includes logic to cancel an alarm.

```
84 //after clicking on each row
85 holder.llrow.setOnClickListener(new View.OnClickListener() {
86
87     @Override
88     public void onClick(View view) {
89         Log.d("tag: \"tester\", msg: \"onRowClick\");
90
91         Dialog dialog = new Dialog(context);
92         dialog.setContentView(R.layout.add_update_layout);
93         dialog.getWindow().setLayout(ViewGroup.LayoutParams.MATCH_PARENT, ViewGroup.LayoutParams.WRAP_CONTENT);
94
95         TextView txtStartTime = dialog.findViewById(R.id.txtStartTime);
96         EditText edtTask = dialog.findViewById(R.id.edtTask);
97         TextView txtEndTime = dialog.findViewById(R.id.txtEndTime);
98         Button btnAction = dialog.findViewById(R.id.btnAction);
99         btnAction.setText("Update");
100         txtStartTime.setText(arrayList.get(position).start);
101         edtTask.setText(arrayList.get(position).task);
102         txtEndTime.setText(arrayList.get(position).end);
103
104         //AlarmReceiver receiver = new AlarmReceiver(true);
105         //AlarmManager alarmManager = (AlarmManager) context.getSystemService(ALARM_SERVICE);
106         //Intent intent = new Intent(context, AlarmReceiver.class);
107         //PendingIntent pendingIntent = PendingIntent.getBroadcast(context, REQUEST_CODE, intent, 0);
108         //Log.d("tester", "cancel alarm");
109         //alarmManager.cancel(pendingIntent);
110
111         //After clicking on startTime textview
112         txtStartTime.setOnClickListener(new View.OnClickListener() {
113
114             @Override
115             public void onClick(View v) {
```


Android Application – Time Tracker



Android Application – Time Tracker



REFERENCES

1. Introduction to Android:

<http://developer.android.com/guide/index.html>.

2. *Android API:*

<http://developer.android.com/reference/packages.html>

3. *Android User Interfaces:*

<http://developer.android.com/guide/topics/ui/index.html>

4. *Layout:*

<http://developer.android.com/guide/topics/ui/declaring-layout.html>

5. *Android Training:*

<http://developer.android.com/training/index.html>.