Medicine Tracker and Notification System

A mini-project report submitted in partial fulfilment of the Academic requirements for the award of the Degree of

BACHELOR OF ENGINEERING
IN
INFORMATION TECHNOLOGY

By

Arikela Srija	**(2451-20-737-039)**
Sattavaram Praneeth Kumar	**(2451-20-737-045)**
Gargula Ajay Kumar	**(2451-20-737-049)**

Under the guidance of
P. Sita Sowjanya
Professor,Dept of IT

DEPARTMENT OF INFORMATION TECHNOLOGY

MATURI VENKATA SUBBA RAO (MVSR)

ENGINEERING COLLEGE

(An Autonomous Institution)

(Affiliated to Osmania University, Hyderabad. Recognized by AICTE)
Nadergul, Saroornagar Mandal, Hyderabad-501510

2022-2023

MATURI VENKATA SUBBA RAO (MVSR)
ENGINEERING COLLEGE
(An Autonomous Institution)
(Affiliated to Osmania University, Hyderabad. Recognized by AICTE)
Nadergul, Saroornagar Mandal, Hyderabad-501510

DEPARTMENT OF INFORMATION TECHNOLOGY

CERTIFICATE

This is to certify that the mini project work entitled "**Medicine Tracker and Notification System**" is a bonafide work carried out by **Ms. A. Srija (2451-20-737-039), Mr. S. Praneeth Kumar (2451-20-737-045), Mr. G. Ajay Kumar (2451-20-737-049)** in partial fulfilment of the requirements for the award of degree of **Bachelor of Engineering** in **Information Technology** from **Maturi Venkata Subba Rao (M.V.S.R.) Engineering College,** affiliated to OSMANIA UNIVERSITY, Hyderabad, during the Academic Year 2021-22. Under our guidance and supervision.

The results embodied in this report have not been submitted to any other university or institute for the award of any degree or diploma.

Signature of Project Coordinator   Signature of Guide

Signature of Head, ITD   Signature of External Examiner

DECLARATION

This is to certify that the work reported in the present mini-project entitled "Medicine Tracker and Notification System" is a record of bonafide work done by us in the Department of Information Technology, M.V.S.R. Engineering College, and Osmania University. This report is based on the project work done entirely by us and not copied from any other source.

The results embodied in this project report have not been submitted to any other University or Institute for the award of any degree or diploma to the best of our knowledge and belief.

Roll Number	**Student Name**	**Signature**
2451-20-737-039	Arikela Srija	
2451-20-737-045	Sattavaram Praneeth Reddy	
2451-20-737-049	Gargula Ajay Kumar	

ACKNOWLEDGEMENT

We with extreme jubilance and deepest gratitude, would like to thank our guide, **P. Sita Sowjanya, AssistantvProfessor**, Department of Information Technology, Maturi Venkata Subba Rao (MVSR) Engineering College, for her constant encouragement to us to complete our work in time.

With immense pleasure, we record our deep sense of gratitude to our beloved Head of the department **Dr.K.VenuGopal Rao** Dean-Academics & HOD, Department of Information Technology, Maturi Venkata Subba Rao Engineering College, for permitting and providing facilities to carry out this project.

We would like to extend our gratitude to **D.Muninder** and **K.Devaki** Mini-Project-I coordinators, Department of Information Technology, Maturi Venkata Subba Rao Engineering College, for their valuable suggestions and timely help during the course of the project.

Finally, we express, from the bottom of our heart and deepest gratitude to the entire faculty, my parents and family for the support, dedication, comprehension and love.

A.Srija (2451-20-737-039)
S.Praneeth Reddy (2451-20-737-045)
G.Ajay Kumar (2451-20-737-049)

MVSR Engineering College

Department of Information Technology

COURSE NAME: MINI PROJECT I

COURSE CODE: PW 654 IT

VISION

To impart technical education to produce competent and socially responsible engineers in the field of Information Technology.

MISSION

M1. To make the teaching-learning process effective and stimulating.

M2. To provide adequate fundamental knowledge of sciences and Information Technology with a positive attitude.

M3. To create an environment that enhances skills and technologies required for industry.

M4. To encourage creativity and innovation for solving real world problems.

M5. To cultivate professional ethics in students and inculcate a sense of responsibility towards society.

PROGRAM EDUCATIONAL OBJECTIVES (PEOs)

The Bachelor's program in Information Technology is aimed at preparing graduates who will:

I. Apply knowledge of mathematics and Information Technology to analyze, design and implement solutions for real world problems in core or in multidisciplinary areas.

II. Communicate effectively, work in a team, practice professional ethics and apply knowledge of computing technologies for societal development.

III. Engage in Professional development or postgraduate education to be a life-long learner.

PROGRAM OUTCOMES (POs)

1. Engineering knowledge: Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.

2. Problem analysis: Identity, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using the first principles of mathematics, natural sciences, and engineering sciences.

3. Design/development of solutions: Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.

4. Conduct investigations of complex problems: Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.

5. Modern tool usage: Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.

6. The engineer and society: Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.

7. Environment and sustainability: Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.

8. Ethics: Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.

9. Individual and team work: Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.

10. Communication: Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.

11. Project management and finance: Demonstrate knowledge and understanding of the engineering and management principles and apply these to one’s own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.

12. Life-long learning: Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

PROGRAM SPECIFIC OUTCOMES (PSOS):

(1)Hardware design: An ability to analyze, design, simulate and implement computer hardware/software and use basic analog/digital circuits, VLSI design for various computing and communication system applications.

(2) Software design: An ability to analyze a problem, design an algorithm, identify and define the computing requirements appropriate to its solution and implement the same.

COURSE OBJECTIVES:

1. To enhance practical &amp; Professional skills.

2.To familiarize the tools and techniques of symmetric literature survey and documentation.

3. To expose students to industry practices and teamwork.

4. To encourage students to work with innovative and entrepreneurial ideas.

COURSE OUTCOMES:

On successful completion of this course students will be able to:

1. Define a problem of the recent advancements with applications towards society.

2. Outline requirements and perform requirement analysis for solving the problem.

3. Design and develop a software and/or hardware-based solution within the scope of project using contemporary technologies and tools.

4. Test and deploy the applications for use.

5. Develop the Project as a team and demonstrate the application, with effective written and oral communications.

ABSTRACT

Maintaining good health is crucial in our busy and stressful world. However, medication adherence has emerged as a serious issue, as non-adherence can negatively impact patients and increase medical costs. To help with this, our Medicine Tracker and Notification System allows users to easily track their medication intake and receive timely reminders. After registering, users can add and update their medication details and timing, as well as maintain a timetable for tracking their intake. The system also provides information on the amount of medication purchased and consumed. Notifications are sent when the medication quantity is low or when it's time to take the next dose. With this system, users can improve their medication adherence and ultimately support their overall health and well-being. With the ability to update medication timing and view their consumption, users can easily maintain a timetable and stay on track with their medication schedule. Ultimately, our system is a valuable tool for promoting good health and well-being.

LIST OF FIGURES

LIST OF TABLES

LIST OF SYMBOLS

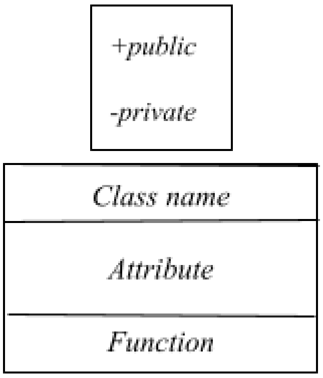
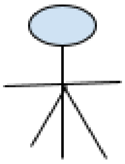
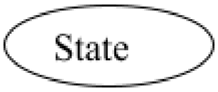
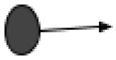
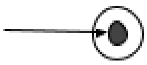
S.no.	Notation name	Notation	Description
1	Class		It aggregates several classes into a single class.
2	Actor		It aggregates several classes into a single class.
3	State		State of the processes.
4	Initial State		Initial state of the object.
5	Final state		Final state of the object.

TABLE OF CONTENTS

CHAPTER 1
INTRODUCTION

- In today's fast-paced and stressful world, maintaining good health is more important than ever.

- However, the challenge of medication adherence is a serious concern that impacts patients and increases medical costs.

- Our Medicine Tracker and Notification System offers a user-friendly platform that allows you to effortlessly track your medication intake and receive timely reminders.

- With our system, you can easily add, update, and view your medication details and timing, making it simple to maintain a well-organized timetable.

- With our Medicine Tracker and Notification System, you'll experience improved medication adherence, leading to better health outcomes and reduced medical costs.

1.1 PROBLEM STATEMENT

File based system to store information about patients and the drug prescriptions they are supposed to receive and at what time. This has led to inaccurate and inefficient delivery of drugs to patients at the right time leading to high mortality rate in the hospital.

1.2 OBJECTIVES

To develop a system that keeps track of and guides patients to effectively provide drugs to patients as prescribed and remind them at the time of taking the dozes.

1.3 MOTIVATION

The remarkable problem is that patients forget to take the proper medicines in the proper proportion and at the proper time. Medication adherence, which refers to the degree or extent to which a patient takes the right medication at the right time according to a doctor’s prescription, has recently emerged as a serious issue because many studies have reported that non adherence may critically affect the patient, thereby raising medical costs. So, the motivation of our project is to reduce medication non-adherence.

1.4 EXISTING SYSTEM

Multiple Medication Reminder Systems have been developed on different platforms. A lot of these systems require special hardware devices to remind the patient about the medicine in-take timings. Purchasing new hardware devices becomes costly and more time and money consuming.

DRAWBACKS OF EXISTING SYSTEM

- User forgetfulness
- Complexity and usability
- Cost

1.5 PROPOSED SYSTEM

The proposed medication reminder system that utilizes existing smartphone technology. The application will serve as the central hub for medication reminders and management. Users will create a profile and input their medication details into the app. This includes the medication name, dosage, frequency, and preferred reminder times. The mobile app will send push notifications to the user's smartphone at the designated reminder times. The reminders will include the medication name, dosage, and any additional instructions. Designing the application with a user-friendly interface that is easy to navigate, making it accessible to users of all age groups and technical backgrounds.

1.6 SCOPE

The scope for this project is quite promising, with the advancement of technology and the growing interest in healthcare and wellness. Many countries are experiencing an aging population, which leads to an increased demand for healthcare services and medication management hence the need for medication reminders to ensure adherence to prescribed treatments will also increase.

The main aim or purpose of medicine reminders is to improve medication adherence. Medication adherence refers to the extent to which patients take their medications as prescribed by healthcare professionals. It is a crucial aspect of healthcare, as poor adherence can lead to ineffective treatments, worsening of health conditions, increased healthcare costs, and a higher risk of hospitalization.

Medicine reminders play a vital role in helping individuals remember to take their medications at the right time and in the correct dosage.

CHAPTER 2

LITERATURE SURVEY

Name	Year	Authors Name	Algorithms/ Techniques used	Advantages	Limitations
Android Based Application to Ensure Medical Adherence	June,2021	Anushka Prashant Amte, Bryan Francis, Elisha Halona Edwin Gras, Sini Anna Reji.	The software stack used for application development was the Flutter SDK and the Dart programming language. For the backend, Firebase and its services such as Realtime Database, Storage and Authentication are used.	All medications and their images along with the time of them to be taken are stored and registered in a database. Through the mobile phone camera, the application can click photos of each medication to help the users to differentiate between different medications by its shape or color.	It does not provide real-life images of the medicine nor the instructions as to how the medicine is to be consumed.
Mobile Application for Medicine Reminder	2021	Ms. S. A. Patil , Ms. Monika Bhanuse , Ms. Snehal Mali , Ms. Vishaka Swami	The algorithms used are Markov chains for statistical pattern notification, Natural language generation and Rule based systems for for remainders and analyze structured data	With doctors overseeing medication intake, precision is guaranteed and potential dangers are preemptively addressed based on patients' health conditions. This proactive approach ensures accurate treatment and minimizes risks.	Patients lack awareness of medication intake; SMS and emails may be ignored. Monitoring multiple patients' data poses challenges for doctors.
Med - X, Medicine Reminder App	2021	Prafful Naikode, Poojal Pithwa, Lakshitha Landge	Threshold alarm algorithm sets off alerts when a certain value exceeds or falls below a predefined threshold, ensuring timely notifications for critical events or conditions. It helps maintain safety and prompt action in real-time monitoring systems.	This System has both Alarm and Notification. Doctor prescription is stored in MongoDB	Special hardware device is used as an alarm system.
Med-Tracker: Android app for medicine tracking	2020	Shrutika Deokar, Sourabh Khandake, Shubam Ture, Deepti Lawand	This framework utilizes NLP algorithm(Key Extraction) and OCR(Optical Character Recognition) is an algorithm used which converts images of text into machine readable text format.	Firebase is a Backend-as-a-Service, which permits users with all types of authentication. OCR makes it easier to store data directly from text irrespective of platforms.	If at all the text is not clear in the prescription image, the OCR may recognize and store the wrong data.
Medical Care Reminder for Infants Using Android Application	June,2017	Abdulrahman Alkandari, Dana Aladem, Somalia Asaad, Samer Moein	Reminder Prediction Algorithm(RPA)	Any extra information or instructions which the user wants to add can be written in the note section. This application targets the children, the vaccination dates, names and process can be stored and encourages parents to learn how to deal with side effects of vaccination. The instruction service is dedicated to give specific awareness to the users.	Alarm should be set by the user itself. If they forget to set the alarms they will not receive the reminders.

CHAPTER 3

SYSTEM REQUIREMENTS SPECIFICATIONS

3.1 HARDWARE REQUIREMENTS

● Processor	:	3.0 GHz Intel(R) Pentium (R) processor
● Hard Disk	:	80GB
● RAM	:	4.0GB
●Flash Disk	:	4.0GB

3.2 SOFTWARE REQUIREMENTS

● Operating System	:	Windows 7/8/10, Android
● IDE	:	Android Studio
● Technology	:	Java

CHAPTER 4

SYSTEM DESIGN

4.1 SYSTEM ARCHITECTURE

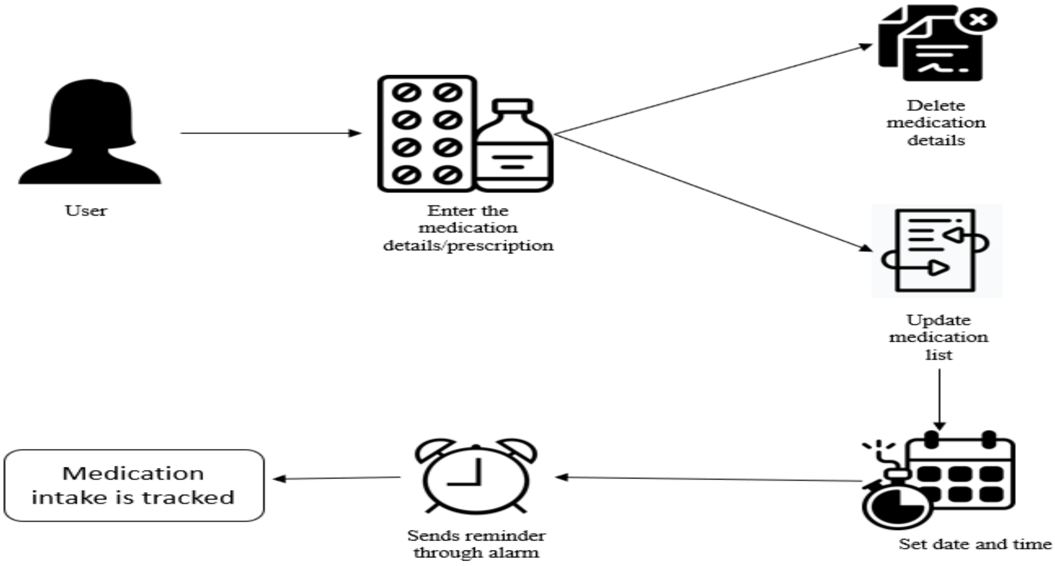

User
Enter the medication details/prescription
Delete medication details
Update medication list
Set date and time
Sends reminder through alarm
Medication intake is tracked

Fig 4.1- System Architecture

4.1.1 ARCHITECTURE DESCRIPTION

- During the initial step, the user provides medication details or prescription information, which encompasses the medicine name, type of medicine, and the prescribed number of doses.

- The user has the ability to modify or delete the entered prescription if deemed necessary.

- The date and time for the medicine intake are established.

- The scheduled reminder is signaled through an alarm at the designated time.

- The user's medication history will be monitored based on the input provided by the user during the reminder notification.

4.1.2 FLOW OF THE PROJECT

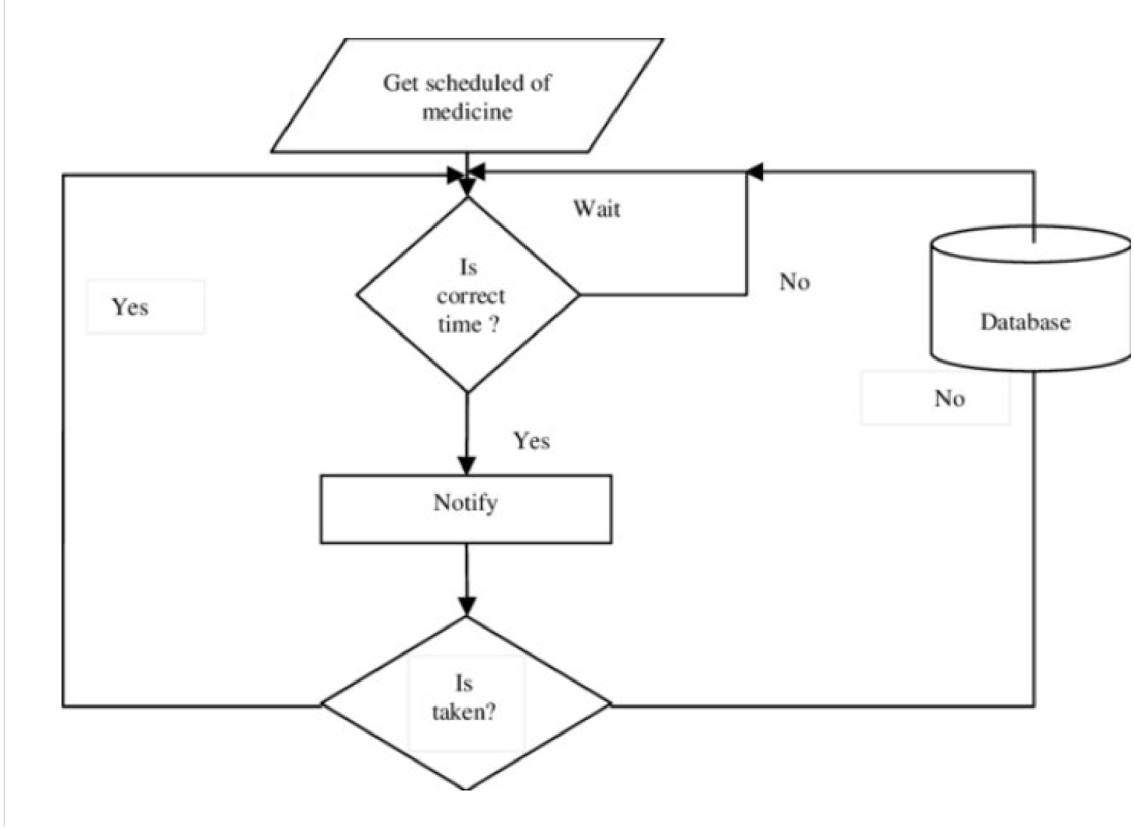


Fig-4.1.2 FLOW OF THE PROJECT

4.2 UML DIAGRAMS

UML stands for Unified Modeling Language. UML is a standardized general-purpose modeling language in the field of object-oriented software engineering. The standard is managed, and was created by, the Object Management Group.

The goal is for UML to become a common language for creating models of object-oriented computer software. In its current form UML comprises two major components: A Meta-model and a notation. In the future, some form of method or process may also be added to; or associated with, UML.

The Unified Modeling Language is a standard language for specifying, Visualization, Constructing and documenting the artifacts of software systems, as well as for business modeling and other non-software systems.
The UML represents a collection of best engineering practices that have proven successful in the modeling of large and complex systems.

The UML is a very important part of developing objects-oriented software and the software development process. The UML uses mostly graphical notations to express the design of software projects.

4.2.1 USE CASE DIAGRAM

A use case diagram in the Unified Modeling Language (UML) is a type of behavioral diagram defined by and created from a Use-case analysis. Its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals (represented as use cases), and any dependencies between those use cases.

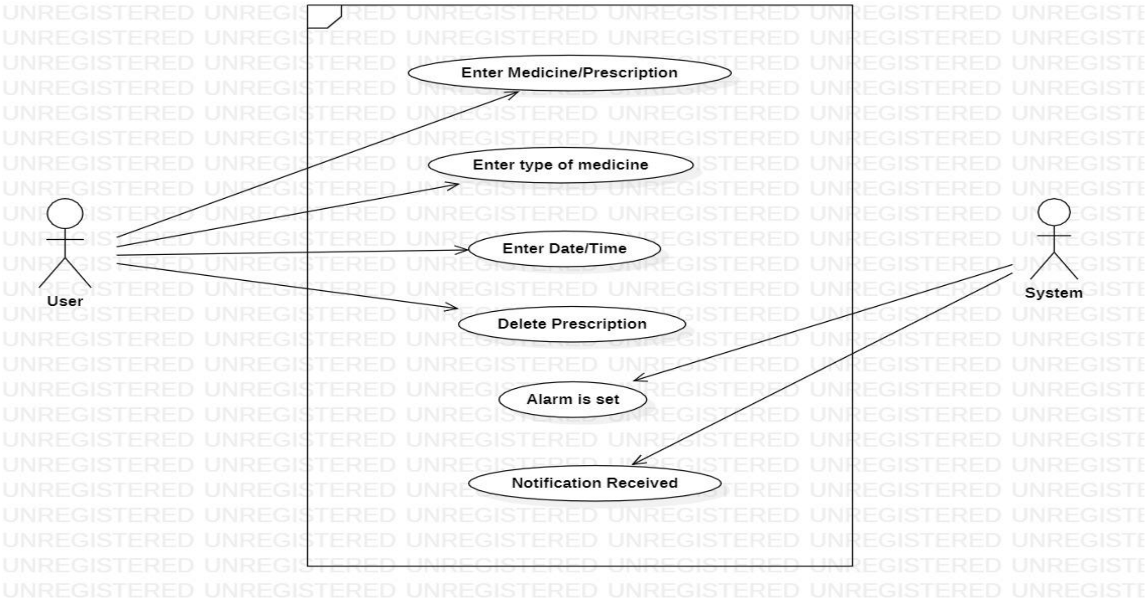


Fig-4.2.1 Use case Diagram

4.2.2 CLASS DIAGRAM

In software engineering, a class diagram in the Unified Modeling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the systems classes, their attributes, operations (or methods), and the relationships among the classes. It explains which class contains which information.

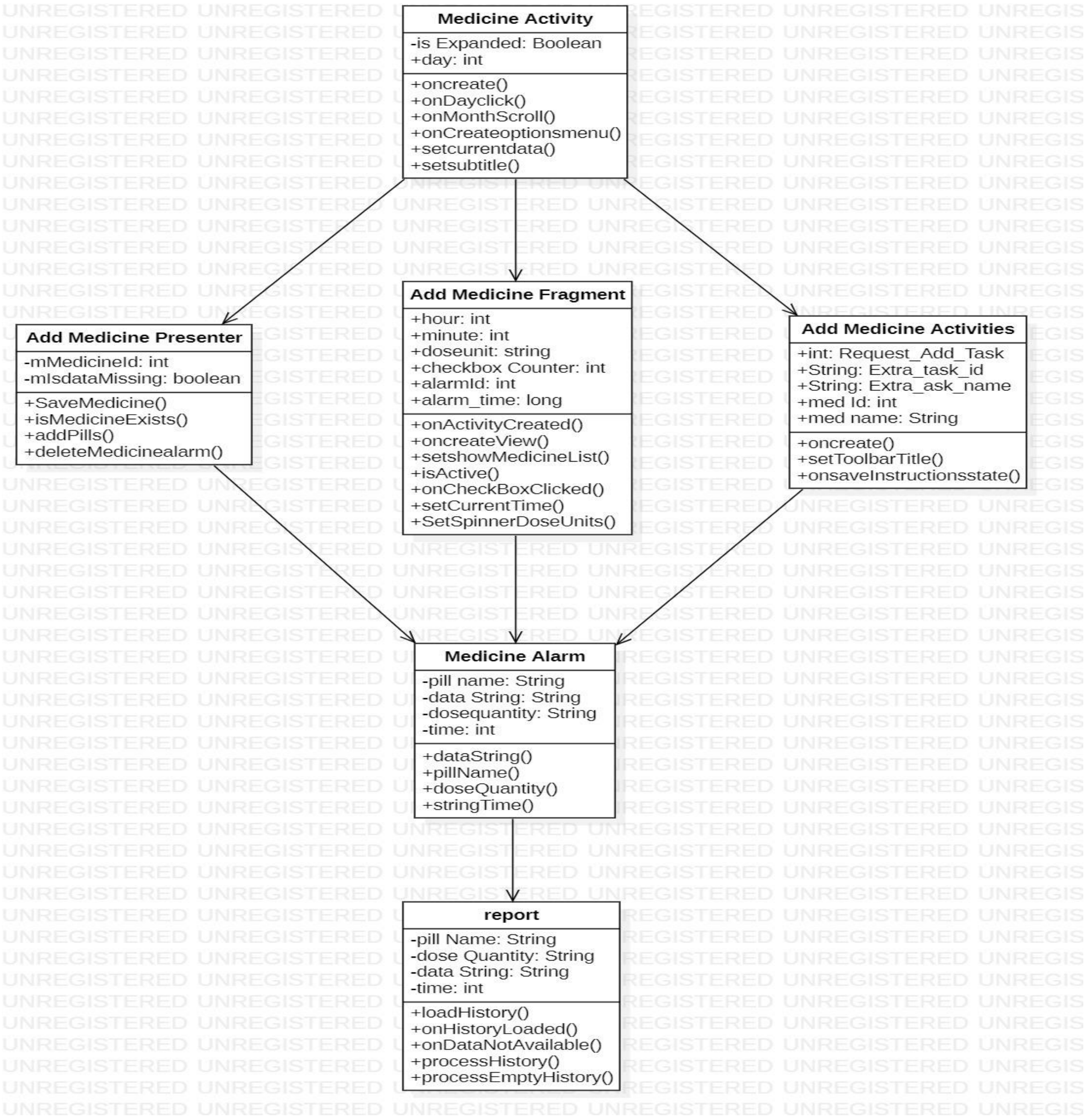


Fig-4.2.2 Class diagram

4.2.3 ACTIVITY DIAGRAM

Activity diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency. In the Unified Modelling Language, activity diagrams can be used to describe the business and operational step-by-step workflows of components in a system. An activity in the diagram shows the overall flow of control.

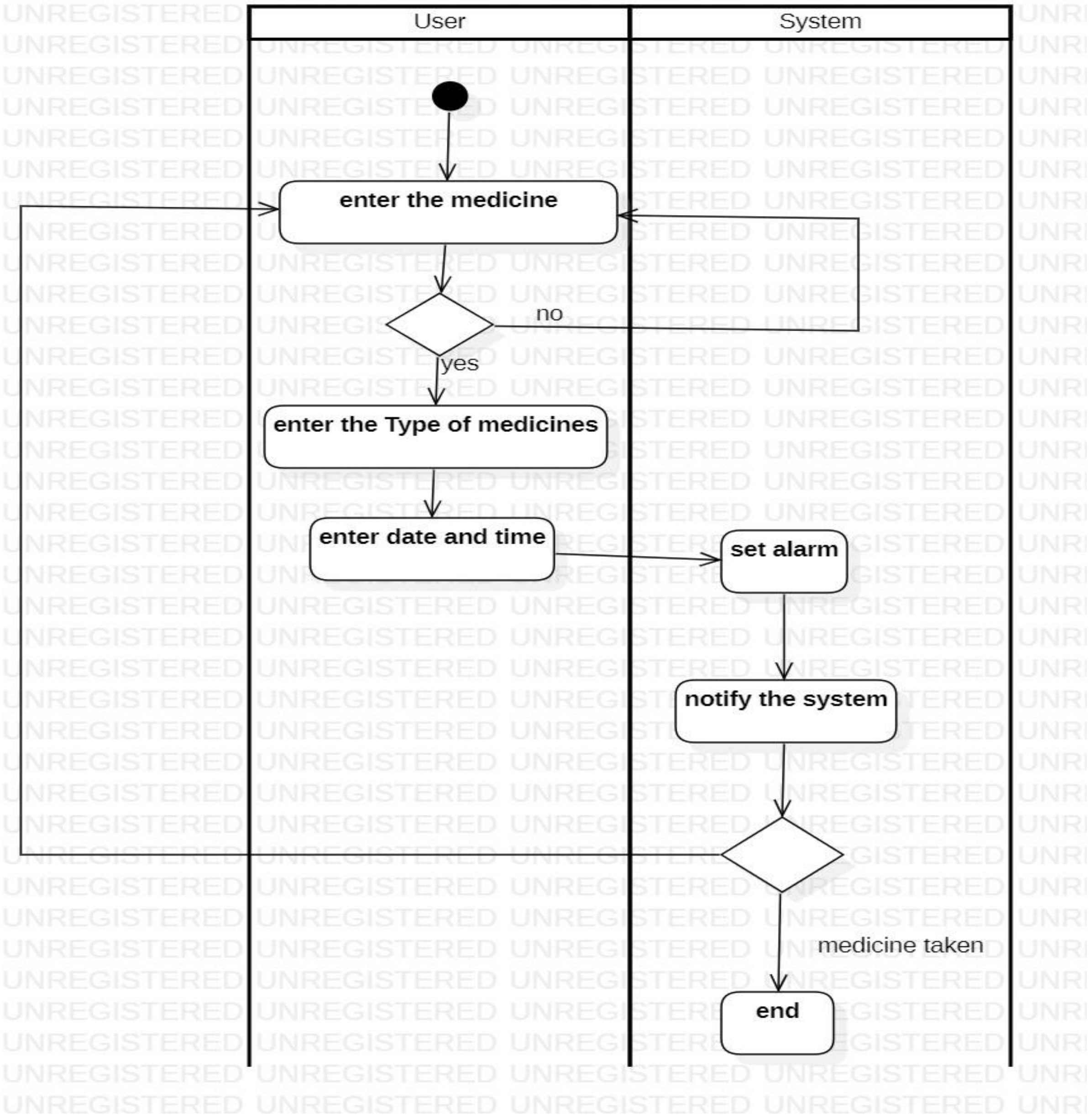


Fig-4.2.3 Activity diagram

CHAPTER 5

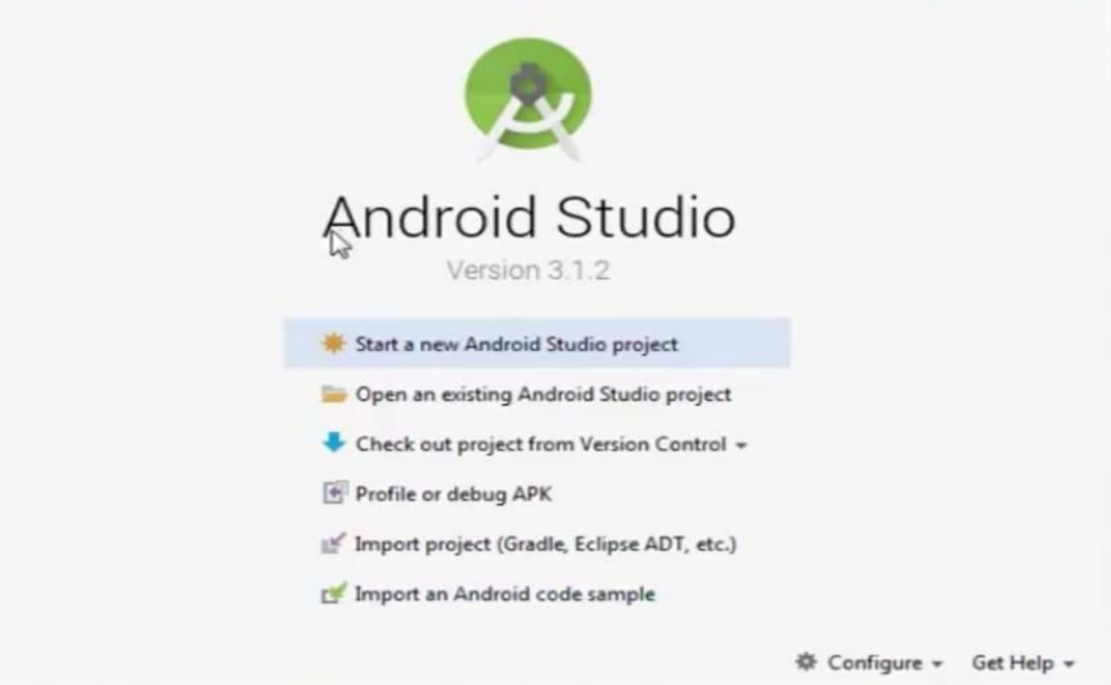
IMPLEMENTATION

5.1 ENVIRONMENTAL SETUP

Installing Android Studio:

1. Download the Android Studio:
   - Visit the official Android Studio website: https://developer.android.com/studio
   - Click on the "Download Android Studio" button.
   - Choose the appropriate version for your operating system (Windows, macOS, or Linux).
2. Install Android Studio:
   - Once the download is complete,locate the installer file and run it.
   - Follow the installation wizard's instructions to install Android Studio on your computer.
   - You may be prompted to choose a location for installation and configure some settings during the process.
3. SDK Components:
   - After the installation is complete, launch Android Studio.
   - On the first launch, Android Studio will prompt you to install the necessary Android SDK components. These components are essential for developing Android applications.
   - Follow the on-screen instructions to download and install the required SDK components. This process may take some time, depending on your internet speed and the components you choose to install.
4. Configure Android Virtual Device (AVD)- Optional:
   - If you want to test your apps on virtual devices you can set up an Android Virtual Device (AVD) by opening the AVD Manager within Android Studio.
   - Navigate to "Tools" > "AVD Manager," and then click " Create Virtual Device."
   - Follow the instructions to select a device definition, choose a system image, and configure the virtual device.

5. Complete Setup:
    - Once the SDK components are installed, and any optional configurations are done, Android Studio is ready to use.



Android Studio

Version 3.1.2

Start a new Android Studio project

Open an existing Android Studio project

Check out project from Version Control ▾

Profile or debug APK

Import project (Gradle, Eclipse ADT, etc.)

Import an Android code sample

Configure ▾ Get Help ▾

Fig-5.1 Android Studio

5.2 WORKING METHODOLOGY

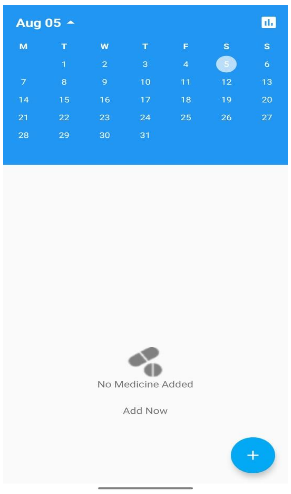
- Project setup: Install Android Studio and download SDK version 30.02.4.1, gradle version 7.4.1.
- Design the application: Create a visually appealing design for the application using Java and XML. Consider incorporating a medicine theme into the layout,colors and images.
- Homepage: Create a homepage that consists of a calendar and medication adding option.
- After adding medicines or prescription with medicine name,medicine days, reminder time, dose quantity, type of medicine. The entered data is tracked.
- When it’s time for medicine intake the alarm is triggered.
- When the alarm is triggered a message pops up with a check box.

- If the medicine is taken yes check box is ticked or else no.
- Based on the input medicine history is tracked.

CHAPTER 6

RESULTS

6.1 Home Page



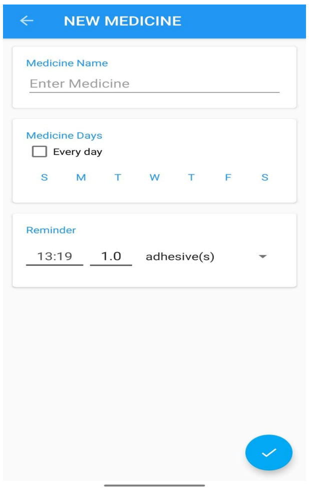
Aug 05

M	T	W	T	F	S	S
	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30	31			

No Medicine Added

Fig-6.1 Home Page

6.2 Medication Adding page



Medicine Name: Enter Medicine

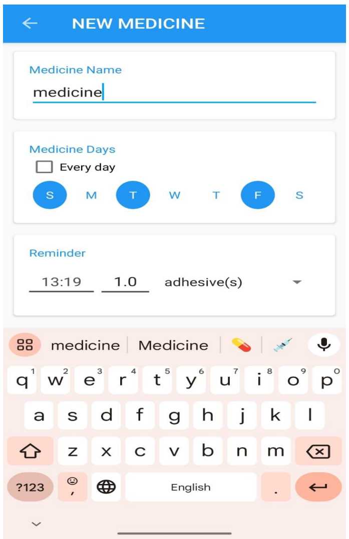
Medicine Days

□ Every day

S M T W T F S

Reminder: 13:19 1.0 adhesive(s)

Fig-6.2.1 Medication Adding Page



NEW MEDICINE

Medicine Name

medicine

Medicine Days

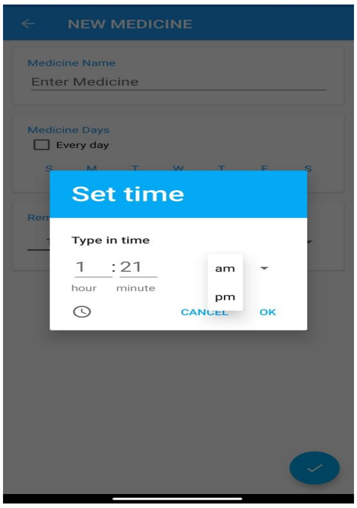
□ Every day

S M T W T F S

Reminder

13:19 1.0 adhesive(s)

Fig-6.2.2 Entering medicine name, day on which medicine should be taken



NEW MEDICINE

Medicine Name

Enter Medicine

Medicine Days

□ Every day

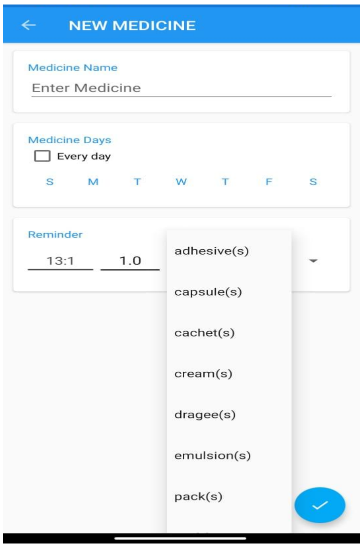
Set time

Type in time

1 : 21

hour minute

Fig-6.2.3 Setting the time for medicine intake



NEW MEDICINE

Medicine Name

Enter Medicine

Medicine Days

□ Every day

S M T W T F S

Reminder

13:1 1.0

adhesive(s)

capsule(s)

cachet(s)

cream(s)

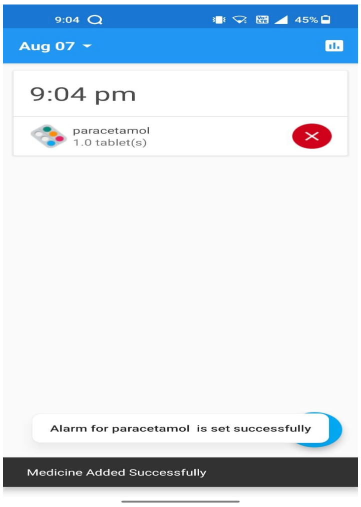
dragee(s)

emulsion(s)

pack(s)

Fig-6.2.4 Selecting type of medicine

6.3 Successful setting of alarm page



9:04 45%

Aug 07

9:04 pm

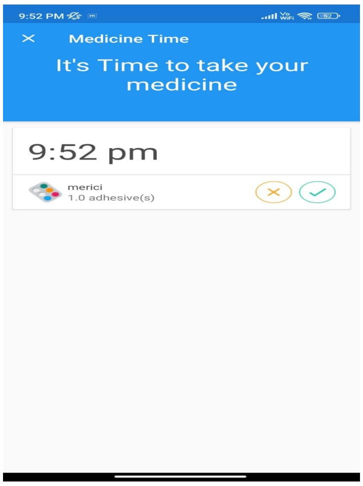
paracetamol
1.0 tablet(s)

Alarm for paracetamol is set successfully

Medicine Added Successfully

Fig-6.3 Successful setting of alarm

6.4 Alarm receiving Page



9:52 PM

Medicine Time

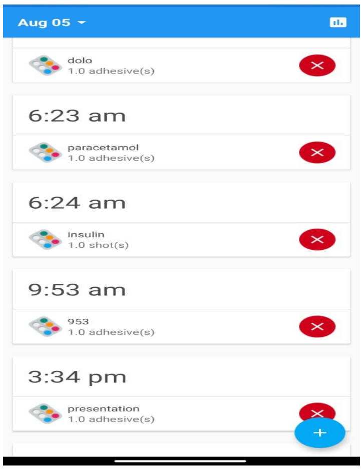
It's Time to take your medicine

9:52 pm

merici
1.0 adhesive(s)

Fig-6.4 Alarm receiving Page

6.5 Medication intake History Page



Aug 05

dolo
1.0 adhesive(s)

6:23 am

paracetamol
1.0 adhesive(s)

6:24 am

insulin
1.0 shot(s)

9:53 am

953
1.0 adhesive(s)

3:34 pm

presentation
1.0 adhesive(s)

Fig-6.5 History Page

CHAPTER 7

TESTS

7.1 TESTING

7.1.1 Unit Testing

Unit testing is usually conducted as part of a combined code and unit test phase of the software lifecycle, although it is not uncommon for coding and unit testing to be conducted as two distinct phases.

Test strategy and approach

Field testing will be performed manually and functional tests will be written in detail.

Test objectives

- All field entries must work properly.
- Pages must be activated from the identified link.
- The entry screen, messages and responses must not be delayed.

Features to be tested

- Verify that the entries are of the correct format.
- All links should take the user to the correct page.

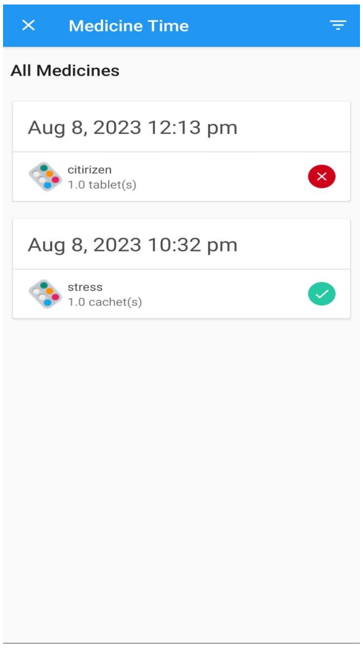
7.1.2 Integration Testing

Software integration testing is the incremental integration testing of two or more integrated software components on a single platform to produce failures caused by interface defects.

The task of the integration test is to check that components or software applications, e.g. components in a software system or – one step up – software applications at the company level – interact without error.

Test Results: All the test cases mentioned above passed successfully. No defects encountered.

7.2 TEST CASES



All Medicines

Aug 8, 2023 12:13 pm

citirizen
1.0 tablet(s)

Aug 8, 2023 10:32 pm

stress
1.0 cachet(s)

Fig-7.2 Test cases

- Upon receiving the alarm reminder, a pop-up appears, prompting the user to confirm their medication intake. If the user takes the medicine, they can select the tick checkbox; otherwise, they can select the cross checkbox.

CHAPTER 8

CONCLUSION AND FUTURE ENHANCEMENTS

The successful development of the “Medicine Tracker and Notification System” represents a significant leap forward in addressing the critical issue of medication non-adherence. By providing users with a user-friendly and efficient platform to monitor and manage their medication intake, this system has the potential to greatly enhance an individual's overall health and well-being. With its seamless ability to track medication details, update timings, and deliver timely reminders, users can now take charge of their medication schedule and make substantial strides towards improved adherence. By continuously refining and expanding the features of the Medicine Tracker and Notification System, we are poised to create a comprehensive solution that not only addresses medication adherence but also serves as a valuable tool in the larger landscape of modern healthcare. In this era of heightened technological integration, we can look forward to a future where individuals have the tools they need to effortlessly manage their medications, prioritize their health, and enjoy improved quality of life.

- get a reminder for medicine intake automatically.
- It can include medicine intake history for future references.

REFERENCES

- Alfredo V. Atienza, Lorissa Joana E. Buenas, Froilan G. Destreza, "Towards Personalized Medication Management: A Novel Approach to Medication Reminder Using Wearable Sensors and IoT", 2023 IEEE International Conference on Automatic Control and Intelligent Systems (I2CACIS), pp.167-172, 2023.

- Pradnya Mhatre, Rasika Patil, "IoT Based Pill Reminder System", International Journal of Advanced Research in Science, Communication and Technology, pp.536, 2022.

- M. Neela Harish, "Patient Medicine Reminder System", Advances in Materials Research, vol.5, pp.73, 2021.

- Dan O. Oketch, Topside E. Mathonsi, "Smart Task Reminder Model for People with Mild Dementia at Home", 2021 International Conference on Computational Science and Computational Intelligence (CSCI), pp.1236-1240, 2021.

- Abhishek Madankar, Akhilesh Agrawal, Vedant Yede, "IoT based Advance Pill Reminder System for Distinct Patients", 2021 Fifth International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC), pp.161-165, 2021.

- S. B. Kumar, W. W. Goh and S. Balakrishnan, "Smart Medicine Reminder Device For The Elderly", 2018 Fourth International Conference on Advances in Computing Communication & Automation (ICACCA), pp. 1-6, 2018.

APPENDIX

Source code:

```
package com.delaroystudios.alarmreminder;

import android.app.LoaderManager;
import android.app.ProgressDialog;
import android.content.ContentUris;
import android.content.CursorLoader;
import android.content.Intent;
import android.content.Loader;
import android.database.Cursor;
import android.net.Uri;
import android.support.design.widget.FloatingActionButton;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.support.v7.widget.Toolbar;
import android.view.View;
import android.widget.AdapterView;
import android.widget.ListView;

import com.delaroystudios.alarmreminder.data.AlarmReminderContract;
import com.delaroystudios.alarmreminder.data.AlarmReminderDbHelper;

public class MainActivity extends AppCompatActivity implements
LoaderManager.LoaderCallbacks<Cursor> {

    private FloatingActionButton mAddReminderButton;
    private Toolbar mToolbar;
    AlarmCursorAdapter mCursorAdapter;
    AlarmReminderDbHelper alarmReminderDbHelper = new
AlarmReminderDbHelper(this);
    ListView reminderListView;
    ProgressDialog prgDialog;

    private static final int VEHICLE_LOADER = 0;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        mToolbar = (Toolbar) findViewById(R.id.toolbar);
        setSupportActionBar(mToolbar);
        mToolbar.setTitle(R.string.app_name);
        reminderListView = (ListView) findViewById(R.id.list);
        View emptyView = findViewById(R.id.empty_view);
        reminderListView.setEmptyView(emptyView);

        mCursorAdapter = new AlarmCursorAdapter(this, null);
        reminderListView.setAdapter(mCursorAdapter);

        reminderListView.setOnItemClickListener(new AdapterView.OnItemClickListener() {
          @Override
          public void onItemClick(AdapterView<?> adapterView, View view, int position, long
id) {

        Intent intent = new Intent(MainActivity.this, AddReminderActivity.class);

        Uri currentVehicleUri =
ContentUris.withAppendedId(AlarmReminderContract.AlarmReminderEntry.CONTENT_U
RI, id);

        // Set the URI on the data field of the intent
        intent.setData(currentVehicleUri);

        startActivity(intent);

          }
        });


        mAddReminderButton = (FloatingActionButton) findViewById(R.id.fab);

        mAddReminderButton.setOnClickListener(new View.OnClickListener() {
          @Override
          public void onClick(View v) {
        Intent intent = new Intent(v.getContext(), AddReminderActivity.class);
        startActivity(intent);
          }
        });

        getLoaderManager().initLoader(VEHICLE_LOADER, null, this);


    }

    @Override
    public Loader<Cursor> onCreateLoader(int i, Bundle bundle) {
        String[] projection = {
        AlarmReminderContract.AlarmReminderEntry._ID,
        AlarmReminderContract.AlarmReminderEntry.KEY_TITLE,
        AlarmReminderContract.AlarmReminderEntry.KEY_DATE,
        AlarmReminderContract.AlarmReminderEntry.KEY_TIME,
        AlarmReminderContract.AlarmReminderEntry.KEY_REPEAT,
                AlarmReminderContract.AlarmReminderEntry.KEY_REPEAT_NO,
                AlarmReminderContract.AlarmReminderEntry.KEY_REPEAT_TYPE,
                AlarmReminderContract.AlarmReminderEntry.KEY_ACTIVE

        };

        return new CursorLoader(this,   // Parent activity context
                AlarmReminderContract.AlarmReminderEntry.CONTENT_URI,   // Provider
content URI to query
                projection,             // Columns to include in the resulting Cursor
                null,                   // No selection clause
                null,                   // No selection arguments
                null);                  // Default sort order

    }

    @Override
    public void onLoadFinished(Loader<Cursor> loader, Cursor cursor) {
        mCursorAdapter.swapCursor(cursor);

    }

    @Override
    public void onLoaderReset(Loader<Cursor> loader) {
        mCursorAdapter.swapCursor(null);

    }
}
```