

## CHAPTER 1

# INTRODUCTION

### 1.1 INTRODUCTION TO ANDROID

Android Inc. was founded in Palo Alto, California, in October 2003 by Andy Rubin, Rich Miner, Nick Sears, and Chris White. Rubin described the Android project as having "tremendous potential in developing smarter mobile devices that are more aware of its owner's location and preferences". The early intentions of the company were to develop an advanced operating system for digital cameras, and this was the basis of its pitch to investors in April 2004. The company then decided that the market for cameras was not large enough for its goals, and five months later it had diverted its efforts and was pitching Android as a handset operating system that would rival Symbian and Microsoft Windows Mobile.

### 1.1 Introduction to Application

The Android healthcare management project is a user-friendly mobile application that offers a wide range of healthcare services. It allows users to conveniently schedule lab tests, purchase medicines, find doctors, access informative health articles, and manage their order details. With a focus on enhancing accessibility and convenience, this app aims to empower users in making informed healthcare decisions and efficiently manage their health-related needs.

### 1.2 OVERVIEW OF THE PROJECT

The Android healthcare management project is a user-friendly mobile application that offers a wide range of healthcare services. It allows users to conveniently schedule lab tests, purchase medicines, find doctors, access informative health articles, and manage their order details. With a focus on enhancing accessibility and convenience, this app aims to empower users in making informed healthcare decisions and efficiently manage their health-related needs.

## LITRERATURE SURVEY

1. **Paper name:** Android application for healthcare(IEEE).

**Author name:** Samyuktha Challa, G. Geethakumari,

January, 2021.

- 2. Paper name:** Development of an android based doctor's tele-information system.

**Author name:** Himansu Rout, 2020.

## EXISTING SYSTEM

Manually performed healthcare systems refer to healthcare processes and procedures that are carried out without the aid of advanced technology or automation. In such systems, healthcare professionals rely heavily on manual labor, human judgment, and basic tools to diagnose, treat, and monitor patients. While these systems have been prevalent in the past, the advent of modern technology has significantly transformed healthcare practices.

Manual healthcare systems often involve time-consuming and labor-intensive tasks, such as paper-based medical records, manual data entry, and manual medication administration. This can result in inefficiencies, increased risk of errors, and delays in patient care. Additionally, the reliance on human judgment alone can introduce variability in diagnoses and treatment plans .

## PROPOSED SYSTEM

- Improve access to quality healthcare
- Enhance efficiency in healthcare delivery
- Utilize technology for better patient care
- Emphasize preventive measures
- Ensure affordability and equitable distribution
- Strengthen healthcare infrastructure
- Foster collaboration between healthcare providers
- Prioritize patient privacy and data security
- Promote evidence-based practices

- Encourage interdisciplinary research

## **AIM**

Aim for the healthcare system is to develop innovative technologies and solutions that enhance patient care, improve efficiency, and increase accessibility to healthcare services for all.

## CHAPTER 2

# REQUIREMENTS SPECIFICATION

## 2.1 SOFTWARE REQUIREMENTS

- Windows 8/10/11 or any Linux operating system for download of android studio.
- An android studio and Java Virtual Machine is required for compiling the source code to make the executable file which can then be directly executed.
- Emulator is required for virtual android mobile
- Emulator: To perform and display the functionality of the project.
- Android studio: To create, design, test, debug and run the android project.
- Development environment required is android SDK

## 2.2 HARDWARE REQUIREMENTS

Considering our project, strictly defining hardware requirements is not wise. By the time as our number of users increase, our hardware requirements will change.

- RAM: 8GB minimum or above
- Processor: Intel Core or AMD with 64 bit processor

## CHAPTER 3

# METHODOLOGY

A methodology is the combination of logically related methods and step by techniques for successful planning, control and delivery of the project. It is scientifically-proven, systematic and disciplined approach to project development and implementation.

The project is developed in Android Studio Integrated Development Environment (IDE) using the Java programming language. Java is high-level, class-based, object-oriented, programming language that is designed on write once, run anywhere (WORA). The application layout is designed in XML. It is a markup language and file format for storing, transmitting, and reconstructing arbitrary data. It defines a set of rules for encoding documents in a format that is both human-readable and machine-readable.

### 3.1 Flowchart of the Application

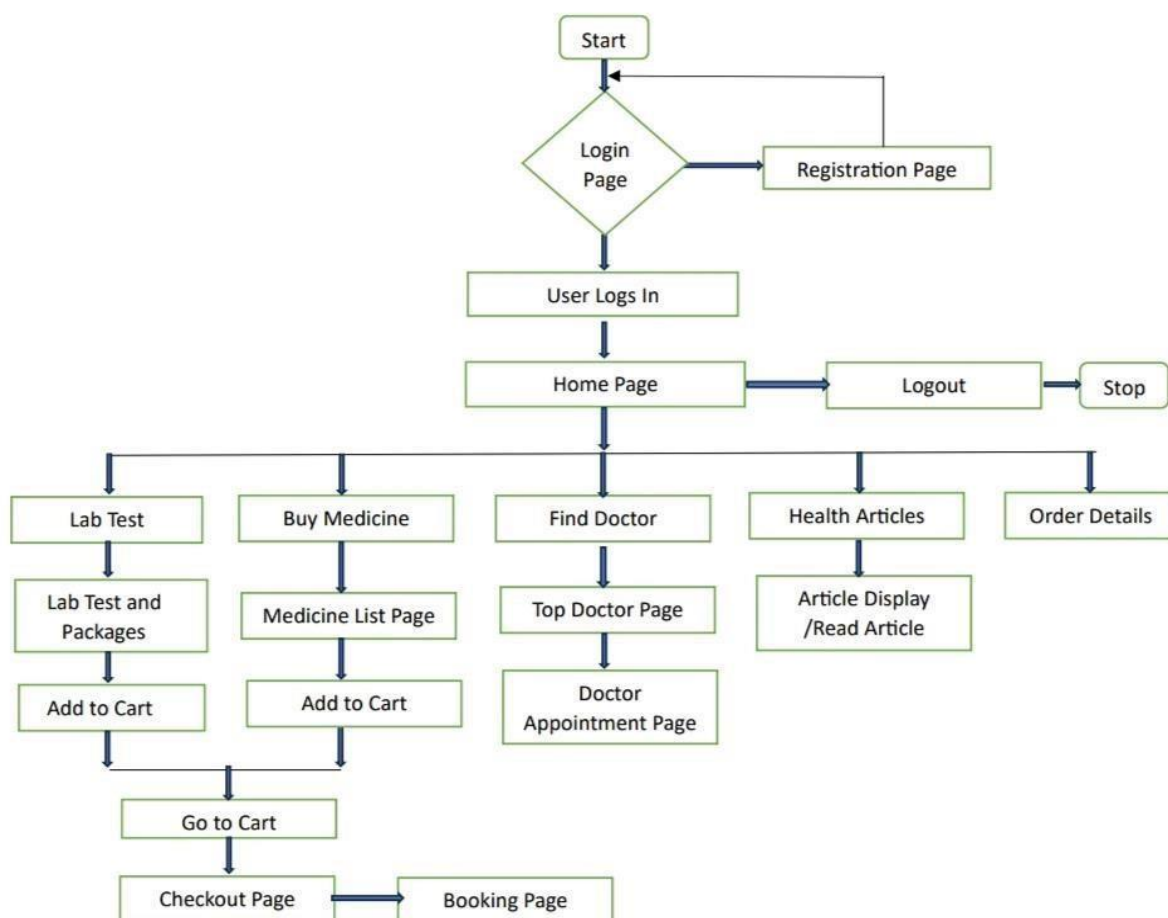


Fig 3.1 Flowchart of the Application

The figure 3.1 represents the flow of user interactions in the healthcare services system, showcasing the sequence of actions from login and registration to accessing different services and features. It's important to note that this is a simplified representation, and the actual implementation may involve additional components, data management, and security measures based on the requirements of healthcare services system. The login page provides a secure and user-friendly interface for users to access their accounts.

The registration option in a healthcare system serves as the initial step for users to create their accounts and gain access to the system's features and functionalities. This process helps in maintaining the privacy and security of user information.

An overview of the project's key features:

- Lab Test
- Buy Medicine
- Find Doctor
- Health Articles
- Logout

## CHAPTER 4

# IMPLEMENTATION

Java provides a vast set of packages and methods inside it that can be used to develop the applications in a faster way. Android Studio IDE provides Intelligent Code Completion through which it provides rapid development of the code.

### 4.1 Methods

**1.Intent** public class Intent extends Object implements Parcelable, Cloneable An intent is an abstract description of an operation to be performed. It can be used with startActivity to launch an Activity, broadcastIntent to send it to any interested BroadcastReceiver components, and Context.startService(Intent) or Context.bindService(Intent, ServiceConnection, int) to communicate with a background Service.

```
Intent i = new Intent(Activity.this, NewActivity.class); startActivity(i);
```

An Intent provides a facility for performing late runtime binding between the code in different applications. Its most significant use is in the launching of activities, where it can be thought of as the glue between activities. It is basically a passive data structure holding an abstract description of an action to be performed.

### 2.Bundle

```
public final class Bundle  
extends BaseBundle implements Cloneable, Parcelable
```

A mapping from String keys to various Parcelable values

```
Bundle b = new Bundle();
```

### 3.SQLiteOpenHelper

#### 3.1 Save data using SQLite

Saving data to a database is ideal for repeating or structured data, such as contact information. This page assumes that you are familiar with SQL databases in general and helps you get started with SQLite databases on Android. The APIs you'll need to use a database on Android are available in the android.database.sqlite package.

```
public final class Sample {
    // To prevent someone from accidentally instantiating the contract class,
    // make the constructor private.

    private

    Sample() {}

    /* Inner class that defines the table contents */ public
    static class FeedEntry implements
        BaseColumns { public static final String TABLE_NAME = "entry";
        public static final String COLUMN_NAME_TITLE = "title"; public
        static final String COLUMN_NAME_SUBTITLE = "subtitle";
        }
    }
```

- The SQLiteOpenHelper class contains a useful set of APIs for managing your database. When you use this class to obtain references to your database, the system performs the potentially long-running operations of creating and updating the database only when needed and not during app startup. All you need to do is call `getWritableDatabase()` or `getReadableDatabase()`.
- To use SQLiteOpenHelper, create a subclass that overrides the `onCreate()` and `onUpgrade()` callback methods. You may also want to implement the `onDowngrade()` or `onOpen()` methods, but they are not required.
- Insert data into the database by passing a `ContentValues` object to the `insert()` method.

- `SQLiteDatabase db = dbHelper.getWritableDatabase();`

// Create a new map of values, where column names are the keys

`ContentValues values = new ContentValues();`

`values.put(FeedEntry.COLUMN_NAME_TITLE, title);`

`values.put(FeedEntry.COLUMN_NAME_SUBTITLE, subtitle);`

// Insert the new row, returning the primary key value of the new row long `newRowId =`

`db.insert(FeedEntry.TABLE_NAME, null, values);`

## 1.2Read information from a database



To read from a database, use the query() method, passing it your selection criteria and desired columns. The method combines elements of insert() and update(), except the column list defines the data you want to fetch (the "projection"), rather than the data to insert. The results of the query are returned to you in a Cursor object.

```
SQLiteDatabase db = dbHelper.getReadableDatabase();

// Define a projection that specifies which columns from the database //
you will actually use after this query.

String[] projection = {
    BaseColumns._ID,
    FeedEntry.COLUMN_NAME_TITLE,
    FeedEntry.COLUMN_NAME_SUBTITLE
};

// Filter results WHERE "title" = 'My Title'
String selection = FeedEntry.COLUMN_NAME_TITLE + " = ?";
String[] selectionArgs = { "My Title" }

// How you want the results sorted in the resulting Cursor
String sortOrder=FeedEntry.COLUMN_NAME_SUBTITLE
+ " DESC";

Cursor cursor = db.query(
    FeedEntry.TABLE_NAME, // The table to query
    projection,            // The array of columns to return (pass null to
                           // get all) selection,          // The columns for the WHERE clause
    selectionArgs,         // The values for the WHERE clause
    null,                  // don't group the rows
    null,                  // don't filter by
    row groups sortOrder   // The sort
    order );
```

### 1.3Close the connection

Since getWritableDatabase() and getReadableDatabase() are expensive to call when the database is closed, you should leave your database connection open for as long as you possibly need to access it. dbHelper.close()

## CHAPTER 5

### RESULT

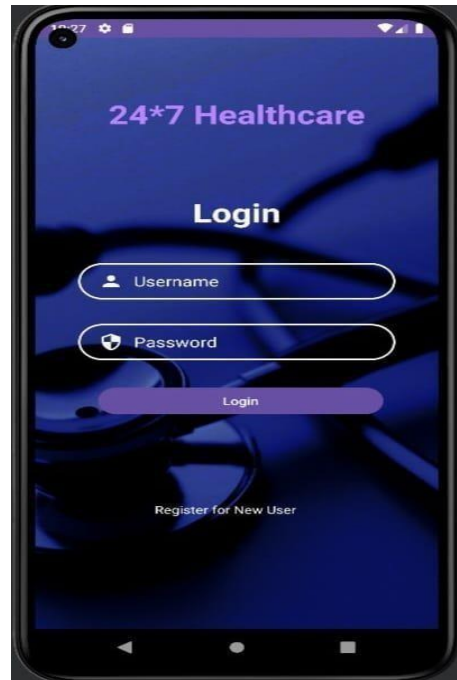


Figure 5.1: User Setup



Figure 5.2: Registration page



Figure 5.3: Home Page

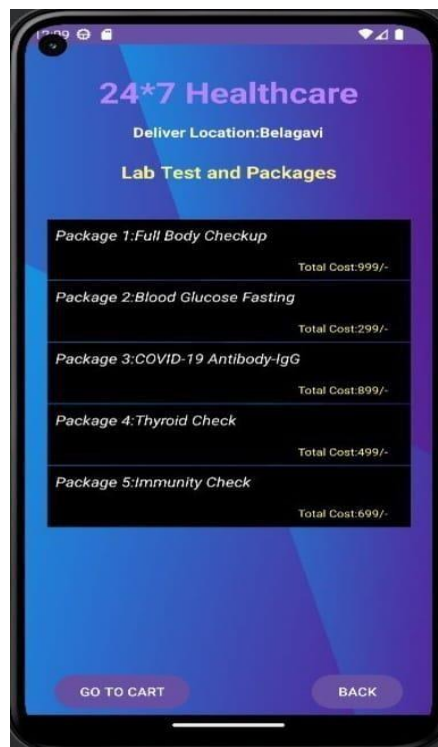


Figure 5.4: Lab Test and Packages

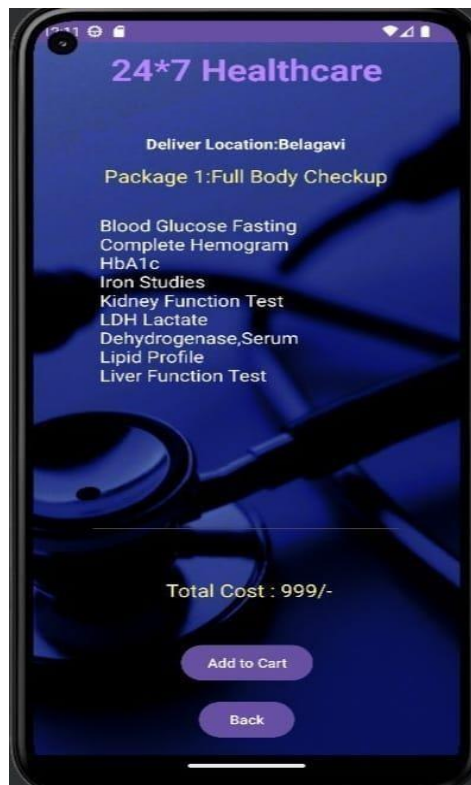


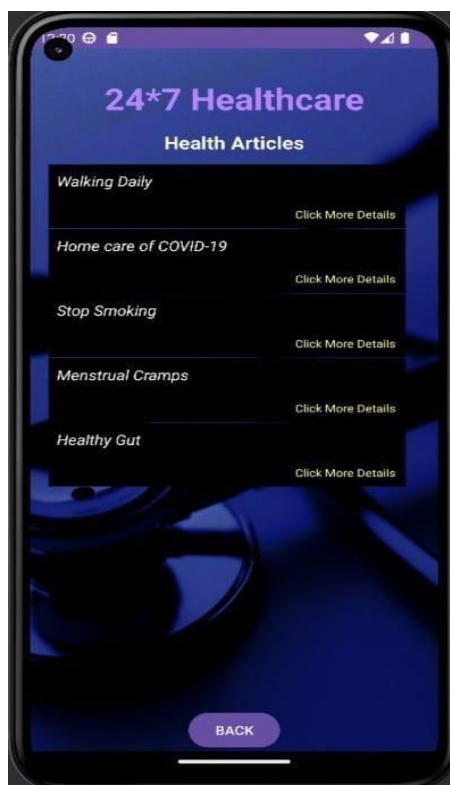
Figure 5.5: Package details



Figure 5.6: Buy Medicine



**Figure 5.7: Top Doctors**



**Figure 5.8: Health Articles**

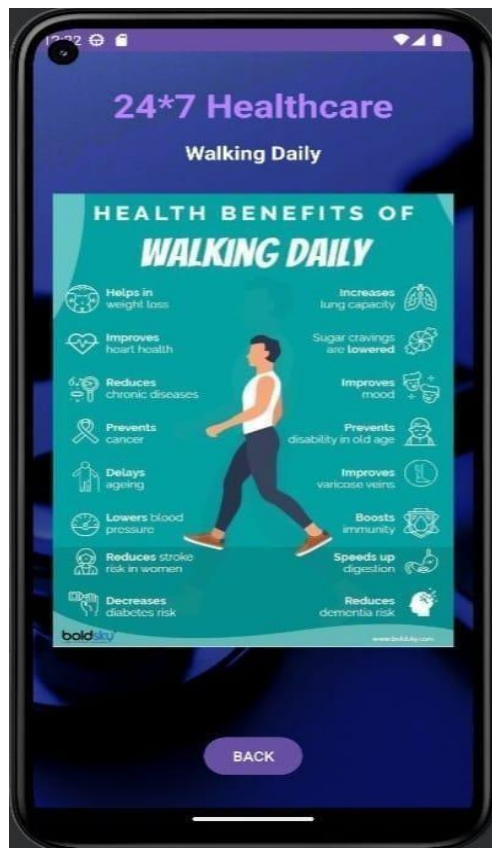


Figure 5.9: Article Detail



Figure 5.10: Cart Details



**Figure 5.11: Order Details**

## CHAPTER 6

### CONCLUSION

In conclusion, the development of our healthcare services project has been a significant endeavor, aiming to provide users with a comprehensive platform for managing their healthcare needs. By integrating various features such as lab test management, medicine tracking, doctor discovery, health articles, and order details, we have strived to create a user-friendly and efficient system.

Through extensive research, analysis, and feedback from users and healthcare professionals, we have designed a platform that addresses several key challenges in the healthcare industry. Our project offers a streamlined approach to accessing and managing healthcare services, empowering individuals to take control of their well-being.

Lastly, our project incorporates an order details feature, which provides users with an overview of their healthcare-related transactions, such as appointments, lab tests, and medication orders. This feature enhances transparency and facilitates efficient record-keeping for users and healthcare providers alike.

Overall, our healthcare services project aims to enhance the accessibility, efficiency, and convenience of healthcare management for users. By leveraging technology and incorporating various essential features, we believe our project contributes to improving the overall healthcare experience. However, continuous testing, user feedback, and iterative improvements will be crucial to refining and optimizing the platform further..



## REFERENCES

1. mHealth: From Smartphones to Smart Systems" edited by Robert Istepanian, Swamy Laxminarayan, and Constantinos S. Pattichis (2016).
2. Paper name: Android application for healthcare(IEEE).  
Author name: Samyuktha Challa, G. Geethakumari, 26 January 2021.