PROJECT REPORT ON HEALTH CARE APPLICATION

Submitted in partial fulfilment of the requirements for the award of the degree of

Bachelor of Technology

(Computer Science & Engineering)

UNDER THE SUPERVISION OF

DR. SANJEEV DHAWAN

SUBMITTED BY

Amit Yadav – 251902081 (CSE-A, 8th sem.)

Session: 2019-2023

University Institute of Engineering and Technology

Kurukshetra University, Kurukshetra - 136119

Acknowledgement

We express our sincere regards and indebtedness to my project guide **Mr. SANJEEV**, **Professor UIET, KUK** for her valuable time, guidance, encouragement, support, and cooperation throughout the duration of our project. We would sincerely like to thank the UIET Department for giving us the opportunity to work on enhancing our technical skills while undergoing this project. This project helped in understanding the various parameters which are involved in the development of a **Health Care Application** and the working and integration of the Software. We would like to thank **MR. SANJEEV DHAWAN**, **Professor UIET, KUK** and the whole department for their constant support.

~ AMIT

Declaration

I, **Health Care Application** developer student of Bachelor of Technology (Computer Science and Engineering), University Institute of Engineering and Technology, Kurukshetra University, Kurukshetra, under class Roll No **251902081**, for the session 2019 – 2023, hereby declare that the project entitled **"Health Care Application"** has been completed by me in last semester is an original work and has not been submitted earlier for award of any degree or diploma to the best of our knowledge and belief. The project contains no such material that may be illegal and offensive.

~AMIT

Certificate

It is certified that Amit Yadav student of Bachelor of Technology Computer Science and Engineering), under class Roll No’s. 251902081 for the session 2019-2023, has completed the project entitled “**HEALTH CARE**” under my supervision. The project report is, in our opinion, worthy for consideration in accordance with the rules and regulations of University Institute of Engineering and Technology, Kurukshetra University, Kurukshetra. I wish him all success in his endeavors.

MR. SANJEEV DHAWAN

Table of Contents Page No.

Problem Statement & Motivation

Problem Statement:

The healthcare industry has always been a critical sector that requires constant attention and improvements. There is a constant need for better healthcare systems that can provide efficient and quality care to patients. In many countries, healthcare systems still face challenges such as long wait times, inefficient processes, and limited access to medical records. Additionally, with the ongoing COVID-19 pandemic, there is a need for contactless medical services and telemedicine

Motivation:

The motivation behind building a healthcare system Android app is to address the challenges faced by the healthcare industry and improve patient outcomes. The app will provide a platform for patients to easily access medical services, manage their health records, and communicate with healthcare providers. By integrating telemedicine features, patients can receive medical advice and treatment from the comfort of their homes, reducing the need for in-person visits.

The app will also help healthcare providers to streamline their processes, reduce wait times, and provide better care to patients. By digitizing medical records, healthcare providers can easily access patient information and make informed decisions. Additionally, the app can help reduce the spread of infectious diseases by offering contactless medical services.

Overall, the healthcare system Android app will be a valuable tool for both patients and healthcare providers, providing a platform for efficient and quality healthcare services.

INTRODUCTION

In recent years, mobile applications have revolutionized various industries, including the healthcare sector. With the increasing adoption of smartphones and mobile devices, healthcare providers are leveraging mobile technology to provide better care to patients. The healthcare system Android app is a prime example of how mobile technology can improve healthcare services.

The healthcare system Android app is a comprehensive platform that enables patients to easily access medical services, manage their health records, and communicate with healthcare providers. The app is designed to be user-friendly and intuitive, allowing patients to access medical services with just a few taps on their mobile devices.

The app is equipped with various features such as appointment scheduling, prescription management, and telemedicine services that enable patients to receive medical care from anywhere, at any time. Patients can also view their medical records, including diagnoses, treatments, and test results, in a secure and confidential manner.

Healthcare providers can also benefit from the app by managing their schedules, communicating with patients, and accessing medical records from a centralized platform. With the app's telemedicine feature, healthcare providers can offer remote consultations, reducing the need for in-person visits and improving the efficiency of their services.

In summary, the healthcare system Android app is a powerful tool that can enhance the quality and efficiency of healthcare services. The app offers patients and healthcare providers a comprehensive platform for managing medical services, promoting better health outcomes, and increasing patient satisfaction

Overview

A healthcare system Android app is a mobile application designed to provide healthcare services to patients and healthcare providers. The app serves as a platform for patients to easily access medical services, manage their health records, and communicate with healthcare providers. Healthcare providers can use the app to manage their schedules, communicate with patients, and access medical records.

The app is equipped with various features such as appointment scheduling, prescription management, and telemedicine services that enable patients to receive medical care from anywhere, at any time. Patients can also view their medical records, including diagnoses, treatments, and test results, in a secure and confidential manner.

The app also offers healthcare providers a platform for managing patient information, tracking patient history, and coordinating patient care. With the app's telemedicine feature, healthcare providers can offer remote consultations, reducing the need for in-person visits and improving the efficiency of their services.

The healthcare system Android app is designed to improve the quality and efficiency of healthcare services. By providing patients and healthcare providers with a comprehensive platform for managing medical services, the app can promote better health outcomes, increase patient satisfaction, and enhance the overall healthcare

Objective

The main objective of a healthcare system Android app is to provide efficient and quality healthcare services to patients by leveraging mobile technology. The app aims to achieve the following objectives:

1. Easy Access to Medical Services: The app allows patients to easily access medical services such as appointment scheduling, prescription management, and telemedicine services, from their mobile devices.

2. Improved Communication: The app promotes improved communication between patients and healthcare providers by offering a platform for secure and confidential messaging and remote consultations.

3. Efficient Medical Records Management: The app enables healthcare providers to efficiently manage patient information, track patient history, and coordinate patient care by digitizing medical records and making them easily accessible.

4. Enhanced Patient Care: By providing patients with access to medical services and enabling healthcare providers to offer remote consultations, the app aims to enhance patient care and promote better health outcomes.

5. Reduced Administrative Burden: The app aims to reduce the administrative burden on healthcare providers by automating tasks such as appointment scheduling, prescription management, and medical record management.

Overall, the objective of a healthcare system Android app is to provide a comprehensive platform for managing medical services, promoting better health outcomes, and enhancing the overall healthcare experience for both patients and healthcare providers.

Materials

- **Hardware**
  - A laptop or Computer
  - Android Device

- **Software**
  - Android Studio
  - Java JDK
  - Android OS (5.0 or Higher)

TOOLS & TECHNOLOGIES

Android SDK

The Android SDK is a software development kit that includes a comprehensive set of development tools. These include a debugger, libraries, a handset emulator based on QEMU, documentation, sample code, and tutorials. Currently supported development platforms include computers running Linux (any modern desktop Linux distribution), Mac OS X 10.5.8 or later, and Windows 7 or later. As of March 2015, the SDK is not available on Android itself, but software development is possible by using specialized Android applications.

Until around the end of 2014, the officially-supported integrated development environment (IDE) was Eclipse using the Android Development Tools (ADT) Plugin. As of 2015, Android Studio[8] is the official IDE; however, developers are free to use others, but Google made it clear that ADT was officially deprecated since the end of 2015 to focus on Android Studio as the official Android IDE. Additionally, developers may use any text editor to edit Java and XML files, then use command line tools (Java Development Kit and Apache Ant are required) to create, build and debug Android applications as well as control attached Android devices (e.g., triggering a reboot, installing software package(s) remotely).

Enhancements to Android's SDK go hand-in-hand with the overall Android platform development. The SDK also supports older versions of the Android platform in case developers wish to target their applications at older devices. Development tools are downloadable components, so after one has downloaded the latest version and platform, older platforms and tools can also be downloaded for compatibility testing.

Android applications are packaged in .apk format and stored under /data/app folder on the Android OS (the folder is accessible only to the root user for security reasons). APK package contains .dex files (compiled byte code files called Dalvik executables), resource files, etc.

JDK: Java Development Kit

As a major programming language throughout the past few decades, Java has been integral to computer science, and the Java JDK is the essential vehicle for establishing a development environment.

People new to Java may be confused about whether to use the JRE or the JDK. To run Java applications and applets, simply download the JRE. However, to develop Java applications and applets as well as run them, the JDK is needed. One way to think of it is that JRE is "just for executables" or limited to that scope, where the JDK is the development toolkit, as mentioned.

Java developers are initially presented with two JDK tools, java and javac. Both are run from the command prompt. Java source files are simple text files saved with an extension of .java. After writing and saving Java source code, the javac compiler is invoked to create .class files. Once the .class files are created, the 'java' command can be used to run the java program.

For developers who wish to work in an integrated development environment (IDE), a JDK bundled with Netbeans can be downloaded from the Oracle website. Such IDEs speed up the development process by introducing point-and-click and drag-and-drop features for creating an application.

There are different JDKs for various platforms. The supported platforms include Windows, Linux and Solaris. Mac users need a different software development kit, which includes adaptations of some tools found in the JDK.

JDK is an acronym for Java Development Kit. The Java Development Kit (JDK) is a software development environment which is used to develop java applications and applets. It physically exists. It contains JRE + development tools.

The JDK contains a private Java Virtual Machine (JVM) and a few other resources such as an interpreter/loader (Java), a compiler (javac), an archiver (jar), a documentation generator (Javadoc) etc. to complete the development of a Java Application.

Android Emulator

The Android Emulator simulates Android devices on your computer so that you can test your application on a variety of devices and Android API levels without needing to have each physical device. The emulator offers these advantages:

Flexibility: In addition to being able to simulate a variety of devices and Android API levels, the emulator comes with predefined configurations for various Android phone, tablet, Wear OS, and Android TV devices.

High fidelity: The emulator provides almost all of the capabilities of a real Android device. You can simulate incoming phone calls and text messages, specify the location of the device, simulate different network speeds, simulate rotation and other hardware sensors, access the Google Play Store, and much more.

Speed: Testing your app on the emulator is in some ways faster and easier than doing so on a physical device. For example, you can transfer data faster to the emulator than to a device connected over USB.

In most cases, the emulator is the best options for your testing needs. This page covers the core emulator functionalities and how to get started with it.

Android Device

An Android device is a device that runs on the Android operating system. Android is an array of software intended for mobile devices that features an operating system, core applications and middleware. An Android device may be a smartphone, tablet PC, e-book reader or any type of mobile device that requires an OS .

Android is developed by the Open Handset Alliance, which is led by Google. Some of the well-known Android device manufacturers include Acer, HTC, Samsung, LG, Sony Ericsson and Motorola.

Explanation of Android

Android device developers and programmers can find information at the Android website, which offers an Android software development kit.

Within a short period, the Android platform became so popular that it surpassed Windows Mobile and Symbian for a number of applications. Various mobile device manufacturers embraced the Android platform due to its overwhelming popularity. The reasons behind this success are as follows:

- Cutting-edge technology offered by Google
- Extremely user-friendly platform
- Can be used in smartphones as well as tablets
- Any user can do modifications to the platform as the Android SDK is open to users
- Availability of huge volume of applications

SQLite Database

SQLite is a software library that provides a relational database management system (RDBMS). It is embedded within the Android operating system and is used to store and manage application data.

SQLite is a lightweight and compact database system that is well suited for mobile devices because of its small footprint and low overhead. It is an open-source and self-contained database engine that requires no separate server process and no configuration. It uses a single file for storing data, which makes it easy to deploy and manage.

In Android, SQLite is used extensively to store and manage application data, such as settings, preferences, user data, and other application-specific information. Android provides a set of APIs for developers to interact with SQLite databases and perform operations such as creating tables, inserting, updating, and querying data.

Integrate SQLite Database in android

To integrate SQLite in an Android application, follow these steps:

1. Create a new Android project in Android Studio.

2. In the project structure, create a new folder named "assets" and create a new file named "database_name.db" (replace "database_name" with the name of your database).

3. Open the Terminal or Command Prompt and navigate to the project directory. Type the following command to create a Java class that will manage the database:

```
 android create dbadapter -p <package_name> -n
<class_name> -d <database_name> -t
<table_name> -c <column_name> -i <id_column_name>
```

This will create a Java class that contains methods to create, upgrade, and manage the database.

4. In the Java class, implement methods to create and upgrade the database, as well as methods to insert, update, delete, and query data. Here is an example of a method to create a table in the database:

```
public void createTable(SQLiteDatabase db) {
db.execSQL("CREATE TABLE " + TABLE_NAME + " (" +
COLUMN_ID + " INTEGER PRIMARY KEY AUTOINCREMENT," +
COLUMN_NAME + " TEXT," +
COLUMN_AGE + " INTEGER)");          }
```

This method creates a table with three columns: an auto-incrementing ID column, a name column, and an age column.

5. In your application code, create an instance of the database adapter class and use its methods to perform database operations. Here is an example of how to insert data into the database:

```
DBAdapter dbAdapter = new DBAdapter(context);
dbAdapter.open();
dbAdapter.insertData("John Doe", 25);
dbAdapter.close();
```

This code creates an instance of the database adapter class, opens the database, inserts a row with the name "John Doe" and age 25, and then closes the database.

These are the basic steps to integrate SQLite in an Android application. There are many resources available online that provide more detailed information and examples of how to use SQLite in Android.

How does it work?

SQLite is a file-based database system that stores data in a single file on the file system. When an application uses SQLite, it creates a connection to the database file and can then read or modify data in the database by executing SQL commands.

The SQLite database engine is responsible for managing the database file and performing database operations. When a database connection is established, the engine reads the database schema, which describes the structure of the database, including tables, columns, and relationships.

When a SQL command is executed, the engine parses the command and checks its syntax and semantics. If the command is valid, the engine performs the requested operation, such as inserting, updating, or querying data. The engine also handles transactions, which ensure that a group of related database operations are performed atomically (all-or-nothing) and consistently.

SQLite provides a number of storage classes for different data types, including integers, floating-point numbers, text strings, and binary data. The engine also supports indexing, which can speed up data retrieval by creating a data structure that maps values in a column to their corresponding rows.

One of the key features of SQLite is its ability to handle concurrent access to the database file by multiple connections. SQLite uses a locking mechanism to ensure that only one connection can modify the database at a time, while allowing multiple connections to read from the database simultaneously.

Overall, SQLite is a fast and efficient database system that is well suited for embedded and mobile applications due to its small footprint and low overhead.

Key capabilities of SQLite Database

SQLite is a powerful and versatile database system that has a number of key capabilities, including:

1. ACID compliance: SQLite supports ACID (Atomicity, Consistency, Isolation, Durability) transactions, which ensure that database operations are performed atomically (all-or-nothing), consistently, and durably.
2. Cross-platform compatibility: SQLite is a cross-platform database system that runs on a wide range of platforms, including desktop operating systems (such as Windows, macOS, and Linux) and mobile operating systems (such as Android and iOS).
3. Small footprint: SQLite is a lightweight database system that has a small memory and disk footprint, making it well-suited for embedded and mobile applications.
4. Self-contained: SQLite is a self-contained database engine that requires no separate server process and no configuration, which makes it easy to deploy and manage.
5. High performance: SQLite is a fast database system that is capable of handling a high volume of read and write operations. Its design and architecture make it highly optimized for efficiency and speed.
6. Extensibility: SQLite provides a number of extension mechanisms that allow developers to add custom functionality to the database system, such as user-defined functions, collations, and virtual tables.
7. Support for SQL: SQLite supports the SQL (Structured Query Language) syntax, which is a standard language for managing and querying relational databases.

Overall, SQLite is a versatile and capable database system that offers a wide range of features and capabilities for managing and querying data. Its small footprint, high performance, and cross-platform compatibility make it an ideal choice for many embedded and mobile applications.

XML

XML (eXtensible Markup Language) is a markup language that defines a set of rules for encoding documents in a format that is both human-readable and machine-readable. It allows for the creation of custom tags and attributes, allowing for greater flexibility and customization than HTML. XML is commonly used for data storage and data interchange, as well as for the \ creation of web services and APIs.

In Android, XML is commonly used to define the layout of an app's user interface, as well as to store data in shared preferences and other persistent storage mechanisms. XML files can be easily edited and manipulated using a variety of tools, making it a popular choice for app developers.

XML is mainly used in Android studio to create the Front end of the application or user Interface of the application. File format of XML is .XML.

In android studio the UI is created with 2 methods.

1. By writing the Code
2. Or by drag and drop of the elements

Working of Video Call Application:

The working of a healthcare system Android app involves several steps:

1. User Registration: Users (i.e. patients and healthcare providers) need to register on the app by providing their basic information, such as name, email, and phone number.

2. User Authentication: Once registered, users need to authenticate themselves using their login credentials, such as username and password.

3. Appointments: Patients can schedule appointments with healthcare providers using the app. The app sends a notification to the healthcare provider to confirm the appointment, and the appointment is added to the healthcare provider's schedule.

4. Medical Records: Patients can view their medical records, including diagnoses, treatments, and test results, in a secure and confidential manner. Healthcare providers can also access medical records from a centralized platform.

5. Prescription Management: Patients can manage their prescriptions using the app, including ordering refills and receiving reminders for medication schedules.

6. Telemedicine: The app offers a telemedicine feature that allows healthcare providers to offer remote consultations to patients. This reduces the need for in-person visits and improves the efficiency of healthcare services.

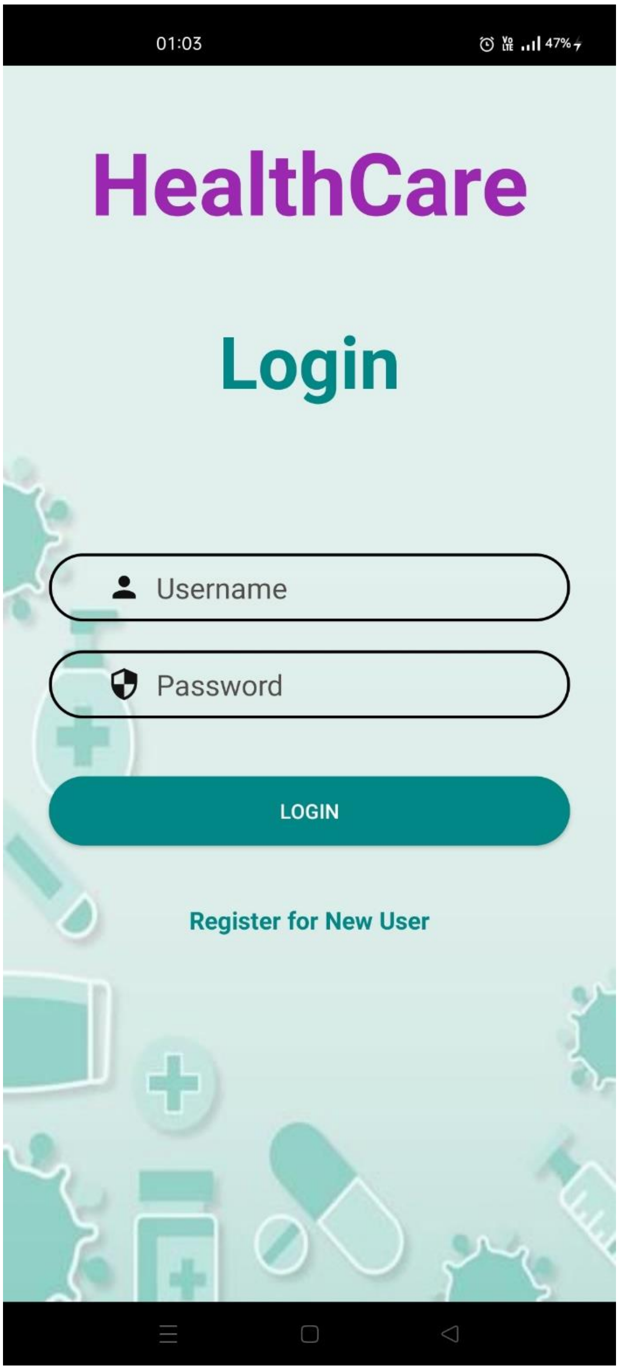
7. Billing and Payments: Patients can make payments for medical services directly through the app, and healthcare providers can manage their billing and payments through the app as well.

The working of a healthcare system Android app involves a secure and efficient platform for managing medical services, promoting better health outcomes, and enhancing the overall healthcare experience for both patients and healthcare providers.

Working with Screenshots

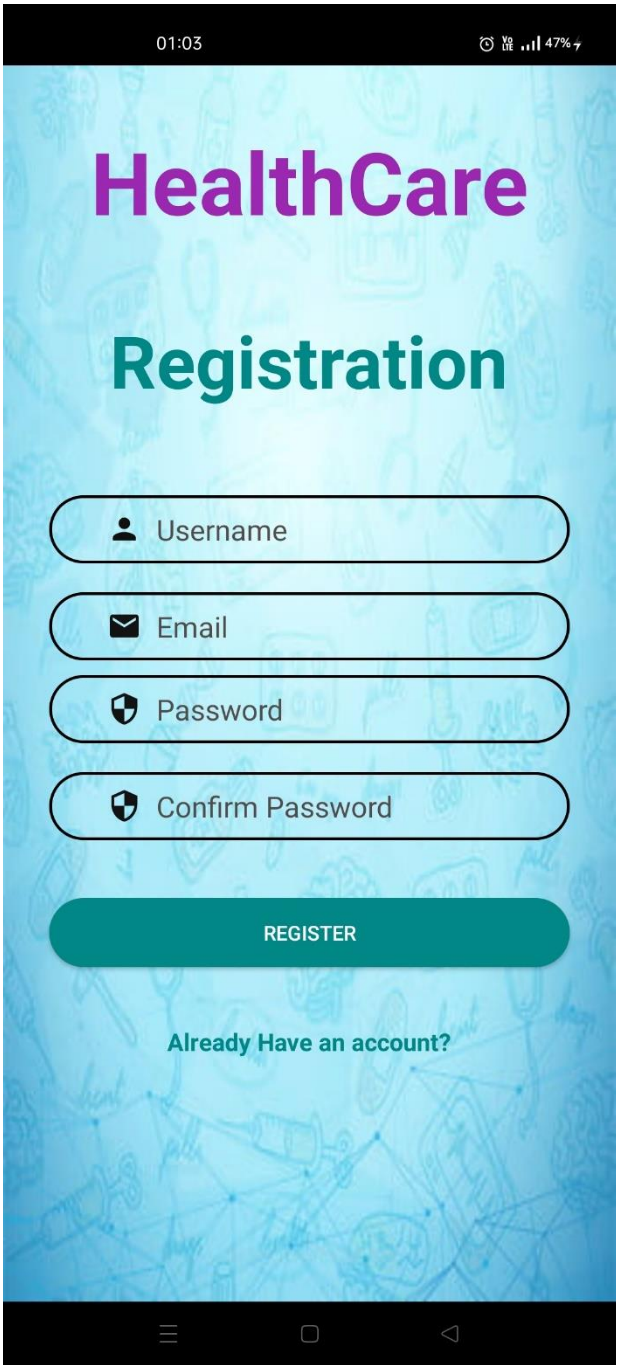
Login

This page provides the user to enter username and password so that user can login to the main dashboard screen and if the user is new to application, then the user can also Register by clicking on the “Register for New User” button.



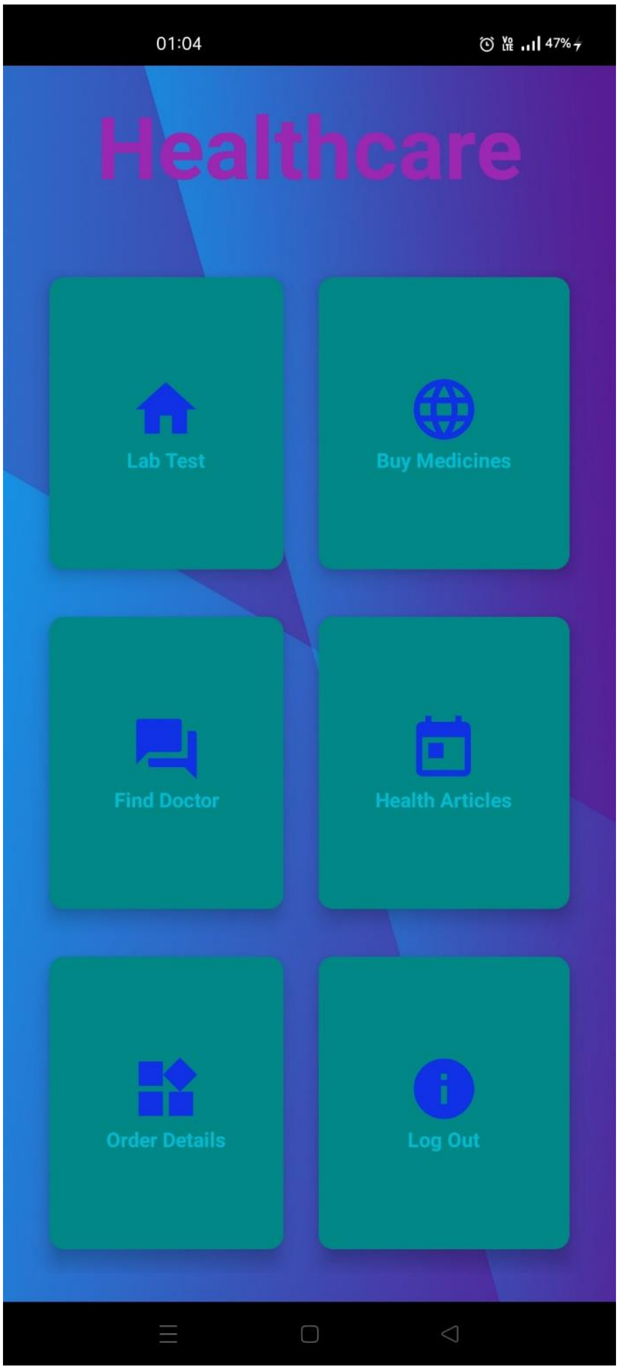
Registration

This page provides the user to enter their Information like “Username”, “Email”, “password”, so that the new user can create an account.



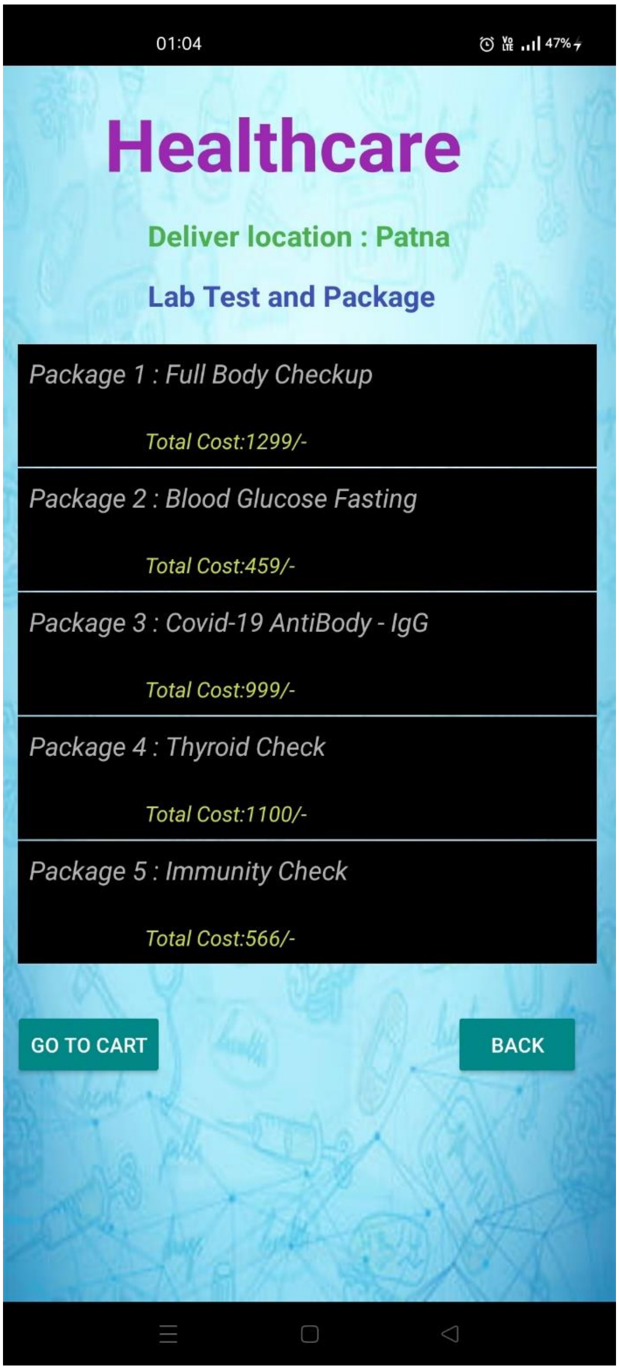
Dashboard Page

Dashboard Page contains the Lab test in which you can book different kind of lab test available. You can buy different kind of medicines read articles which are related to make your body healthy. You can book appointment from doctor.



Features

These are some screenshots of the Healthcare application which shows the features of the application, like we can book lab test , buy medicines , find doctor and read health articles.



01:04 47%

Healthcare

Deliver location : Patna

Lab Test and Package

Package 1 : Full Body Checkup

Total Cost:1299/-

Package 2 : Blood Glucose Fasting

Total Cost:459/-

Package 3 : Covid-19 AntiBody - IgG

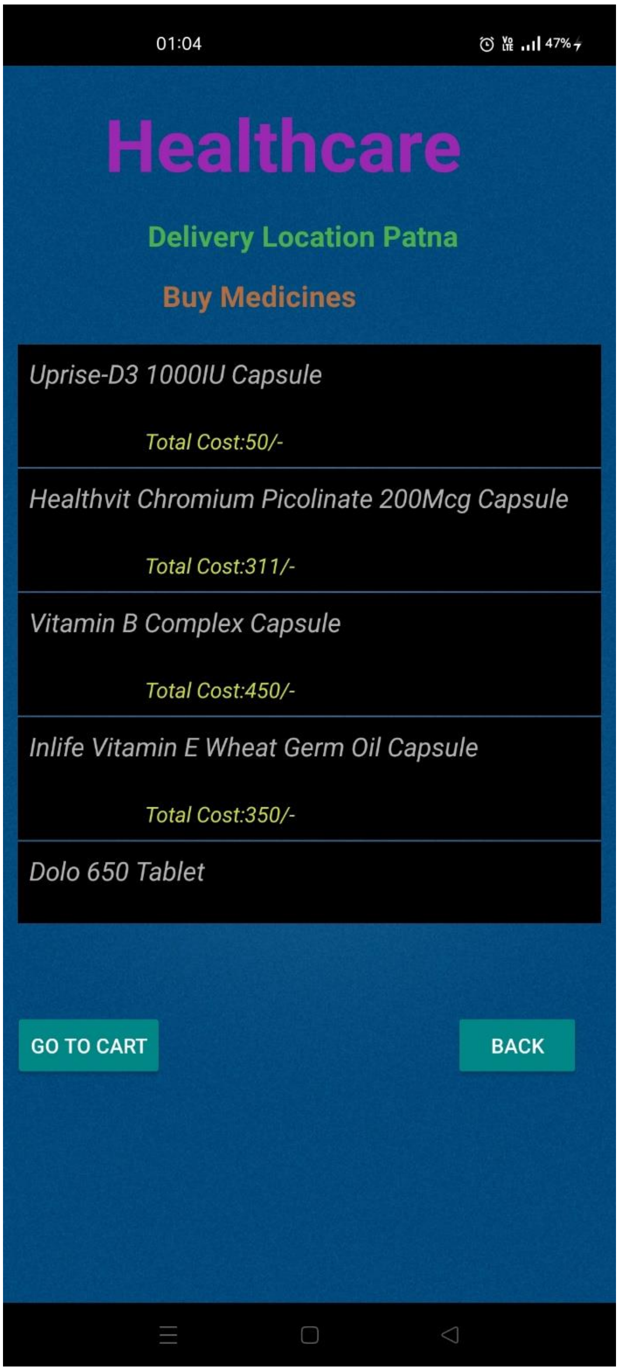
Total Cost:999/-

Package 4 : Thyroid Check

Total Cost:1100/-

Package 5 : Immunity Check

Total Cost:566/-



Healthcare

Delivery Location Patna

Buy Medicines

Uprise-D3 1000IU Capsule

Total Cost:50/-

Healthvit Chromium Picolinate 200Mcg Capsule

Total Cost:311/-

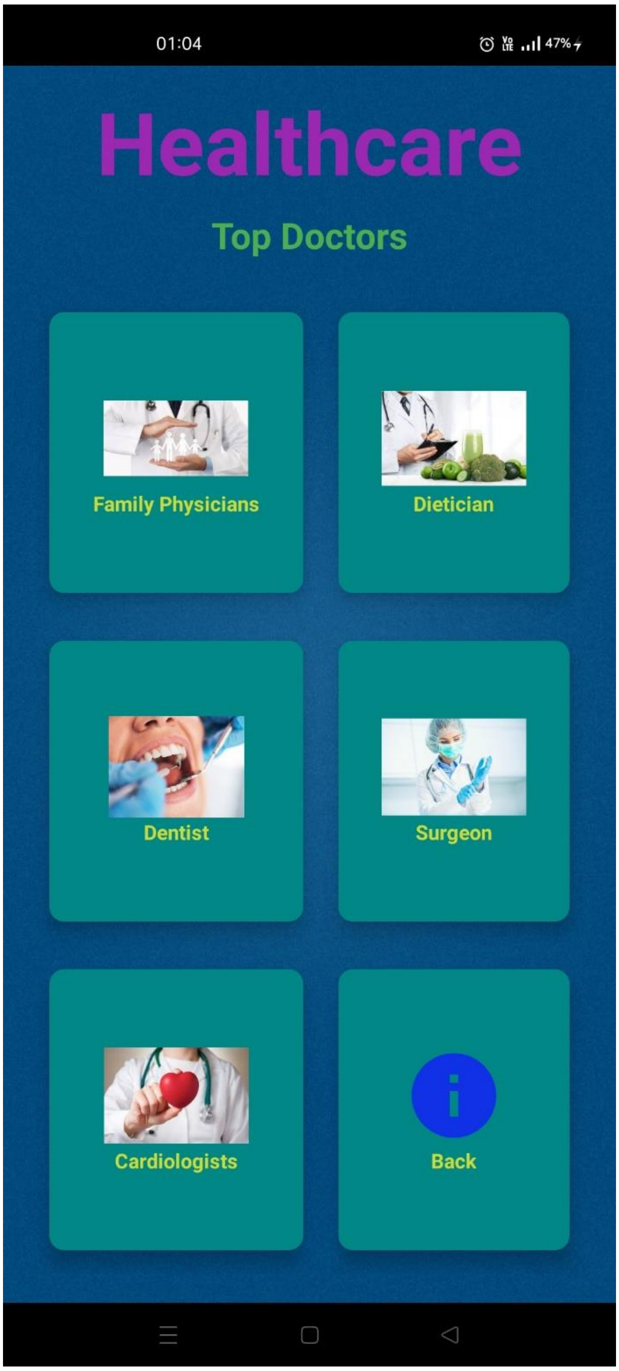
Vitamin B Complex Capsule

Total Cost:450/-

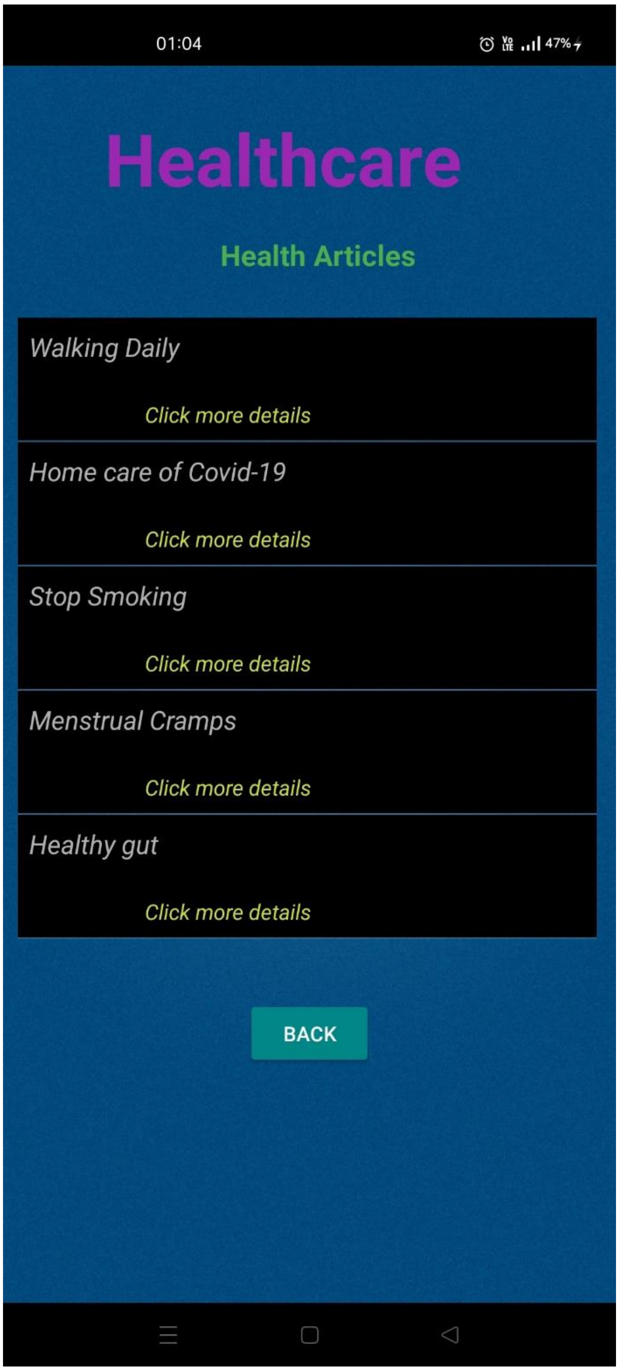
Inlife Vitamin E Wheat Germ Oil Capsule

Total Cost:350/-

Dolo 650 Tablet



01:04
47%
Healthcare
Top Doctors
Family Physicians
Dietician
Dentist
Surgeon
Cardiologists
Back



Healthcare

Health Articles

Walking Daily

Click more details

Home care of Covid-19

Click more details

Stop Smoking

Click more details

Menstrual Cramps

Click more details

Healthy gut

Click more details

Conclusion

In conclusion, a healthcare system Android app has the potential to revolutionize the way healthcare is delivered to patients. With features such as remote patient monitoring, personalized medicine, telemedicine, health education, wearable device integration, electronic health record integration, and medication management, healthcare providers can deliver more effective and efficient care to their patients.

By utilizing technology to improve patient engagement and communication, healthcare system Android apps can empower patients to take control of their health, make informed decisions, and ultimately lead healthier lives. Moreover, with the ongoing COVID-19 pandemic, such apps can be invaluable in providing remote healthcare and reducing the risk of infection transmission. Therefore, the development and implementation of healthcare system Android apps should be encouraged and supported to enhance the overall healthcare experience and improve health outcomes for patients.

Future Works

There are several potential areas for future work for a healthcare system Android app. Some possible directions could include:

1. Remote patient monitoring: Develop features that allow patients to track their vital signs and symptoms from home, and share them with their healthcare providers for monitoring and intervention when necessary.

2. Personalized medicine: Incorporate algorithms and machine learning techniques to analyze patient data, and provide personalized treatment recommendations and plans based on their medical history, genetic makeup, and other relevant factors.

3. Telemedicine: Enhance the app's video consultation capabilities to allow patients to consult with healthcare providers remotely, reducing the need for in-person visits.

4. Health education: Add features that provide patients with reliable and accessible health information, including articles, videos, and other resources that can help them better understand their conditions and treatment options.

5. Wearable device integration: Allow patients to connect their wearable devices, such as fitness trackers and smartwatches, to the app to monitor their health and receive real-time feedback and insights.

6. Electronic health record integration: Integrate with electronic health record systems to provide healthcare providers with a complete view of patients' medical histories, diagnoses, and treatment plans.

7. Medication management: Develop features to help patients manage their medications, including reminders to take pills, refill prescriptions, and track side effects.

Overall, there is significant potential for healthcare system Android apps to improve patient care, streamline communication between patients and providers, and enhance the overall healthcare experience.

Reference

1. Google

2. YouTube

3. SQLite Database docs

4. Geeksforgeeks

5. Stack over flow