



**VIT<sup>®</sup>**  
**Vellore Institute of Technology**  
(Deemed to be University under section 3 of UGC Act, 1956)

**SCHOOL OF COMPUTER SCIENCE ENGINEERING AND INFORMATION SYSTEMS**

**Course Code: SWE2008**

**Course Name: Android Programming**

**Faculty Name: Dr. P. Shunmuga Perumal**

**Final Review Document**

**Slot: B1+TB1**

**Project Title: Patient Monitoring - Healthcare Application**

<b><u>Team Details:</u></b>	<b><u>Reg No</u></b>	<b><u>Name</u></b>
	1) 22MIS0626	1) Vignesh S
	2) 22MIS0600	2) R Rochana
	3) 22MIS0325	3) K Vasanth Kumar

## **Abstract:**

Patient monitoring technologies have developed as critical instruments in modern healthcare, transforming how doctors track and manage patient well-being in clinical settings. Patient monitoring systems have become increasingly important in recent years, as healthcare facilities and medical professionals seek more effective ways to monitor and manage patient health during treatment and recovery. Implementing patient monitoring systems allows healthcare providers to improve clinical results, manage patients with complex medical conditions, and continuously monitor patient health status. These systems are especially useful for healthcare providers who want to remotely monitor hospitalised patients, intensive care unit patients, and those with chronic diseases, minimising the requirement for continual bedside attendance while yet providing complete treatment. Patient monitoring systems have several key features, including continuous vital sign tracking, real-time monitoring of physiological parameters, automated recording of patient data patterns, and the provision of critical alerts and clinical decision support. These systems can be linked to electronic health records (EHR) and hospital information systems, giving healthcare practitioners a comprehensive view of patient health data and medical history. The solutions allow for early diagnosis of patient deterioration, assist timely medical interventions, and promote evidence-based clinical decision-making.

**Keywords:** Patient monitoring system, real-time vital signs tracking, healthcare application, clinical decision support, electronic health records integration, predictive analytics, remote patient monitoring, alert systems, data visualization dashboard, patient safety, chronic disease management, cloud-based platform, mobile accessibility.

## **Applications of the Project:**

### **1. Continuous Health Tracking:**

The project allows doctors and nurses to continuously monitor vital signs and patient conditions without physical checks.

### **2. Early Warning System:**

The alert mechanism helps identify abnormal vitals early, preventing severe medical issues.

### **3. Virtual Medical Assistance:**

Through the video call feature, patients can receive medical guidance remotely without visiting the hospital.

### **4. Digital Patient Records:**

All patient details are stored electronically, making information easier to access, update, and manage.

### **5. Improved Hospital Efficiency:**

By organizing vitals, reports, and patient profiles, the system reduces manual workload and speeds up hospital operations.

### **6. Supports Home Monitoring:**

Suitable for monitoring elderly or long-term patients at home who need regular health supervision.

### **7. Smarter Clinical Decisions:**

Doctors can make informed decisions using the recorded data and trends of patient vitals.

### **8. Faster Emergency Handling:**

Critical health alerts help medical staff act quickly during emergency situations.

## Software and hardware details:

### Software Requirements:

- 1. Android Studio IDE:** The main development tool used to build, debug, and test the application.
- 2. Java/Kotlin Languages:** Used to handle app logic, UI actions, database operations, and backend processes.
- 3. SQLite Local Database:** Stores patient details, vital readings, and doctor information within the device.
- 4. XML Layouts:** Helps to design the user interface elements for all app screens.
- 5. Android SDK (21 or higher):** Provides essential libraries and APIs required for developing Android applications.
- 6. Material UI Libraries:** Offers ready-made components like cards, switches, and navigation bars for modern UI design.

### Hardware Requirements:

- 1. Android Device:** The app can run on any smartphone/tablet with Android 5.0 or above.
- 2. Camera Module:** Needed for video streaming during online doctor consultations.
- 3. Audio Hardware:** The microphone and speaker allow two-way communication between doctor and patient.
- 4. Internet Access:** Required for live video calls and future cloud-based features.
- 5. Minimum 2GB RAM:** Ensures smooth functioning of the application and video modules.

**6. Compatible Processor:** A basic quad-core processor is enough for efficient app performance.

## **App layouts:**

### **1. Dashboard Layout (fragment\_dashboard.xml):**

**Displays a summary of the application, quick access information such as:**

Patient overview

Quick shortcuts

Basic instructions

### **2. Patients Layout (fragment\_patients.xml):**

**This layout contains:**

A scrollable view

A dynamic vertical list of patient cards

Each card shows:

Name

Age

Condition

Blood Pressure

Heart Rate

Edit & Delete buttons

A Floating/Add button to enter new patient information

### **3. Add/Edit Patient Dialog (dialog\_add\_patient.xml):**

**A popup dialog used for:**

Entering patient name

Age

Condition

Blood pressure

Heart rate

Save / Cancel buttons

### **4. Vitals Layout (fragment\_vitals.xml):**

**Shows:**

Patient health vitals

Latest heart rate

BP values

Status indicators (Normal / High / Low)

Card-based UI for readability

### **5. Alerts Layout (fragment\_alerts.xml):**

**Displays:**

List of alert cards

**Alerts for:**

High BP

Low BP

Abnormal heart rate

**Cards are color-coded:**

RED → High risk

ORANGE → Warning

GREEN → Normal vitals

**6. TeleConsult Layout (fragment\_teleconsult.xml):**

**Contains:**

Doctor photo

Name & specialization

Description

Video preview window

Start Video Call button

**7. Settings Layout (fragment\_settings.xml):**

**Provides:**

Toggle for Notifications

Dark Mode switch

About App section

App Version

Logout button

## **8. Reports Layout (fragment\_reports.xml):**

### **Shows:**

Average BP

Total alerts generated

Live statistics of vitals

High/Low alerts count

Graph-based results (optional extension)

## **9. Activity Main Layout (activity\_main.xml):**

### **Includes:**

FrameLayout (fragment container)

Bottom Navigation Bar

“More” menu handling for Alerts, Settings, TeleConsult

### **App Zip File Link:**

<https://www.dropbox.com/scl/fi/a7qp57w1xnh2qidm6jt8n/PatientMonitoring8.zip?rlkey=2326aza7d64mnerq8m1ey9vsf&st=1xirrid7&dl=0>



## Code Samples:

### activity\_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:orientation="vertical"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">

    <!-- ✔ Fragment container for swapping screens -->
    <FrameLayout
        android:id="@+id/fragment_container"
        android:layout_width="match_parent"
        android:layout_height="odp"
        android:layout_weight="1" />

    <!-- ✔ Bottom Navigation -->
    <com.google.android.material.bottomnavigation.BottomNavigationView
        android:id="@+id/bottom_navigation"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        app:menu="@menu/bottom_nav_menu"
        app:itemIconTint="@color/nav_item_color"
        app:itemTextColor="@color/nav_item_color"
        android:background="?android:attr/windowBackground" />

</LinearLayout>
```

## MainActivity.java

```
package com.example.patientmonitoring;

import android.os.Bundle;
import android.view.MenuInflater;
import android.view.View;
import androidx.appcompat.app.AppCompatActivity;
import androidx.appcompat.widget.PopupMenu;
import androidx.fragment.app.Fragment;
import
com.google.android.material.bottomnavigation.BottomNavigationView;

public class MainActivity extends AppCompatActivity {

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

        // ✓ Initialize Bottom Navigation
        BottomNavigationView bottomNav
        findViewById(R.id.bottom_navigation);

        bottomNav.setOnItemSelectedListener(item -> {
            Fragment selectedFragment = null;
            int id = item.getItemId();

            // ✓ Replaced switch with if-else
            if (id == R.id.navigation_dashboard) {
                selectedFragment = new DashboardFragment();
            } else if (id == R.id.navigation_vitals) {
                selectedFragment = new VitalsFragment();
            } else if (id == R.id.navigation_patients) {
                selectedFragment = new PatientsFragment();
            } else if (id == R.id.navigation_reports) {
                selectedFragment = new ReportsFragment();
            } else if (id == R.id.navigation_more) {
```

```

        showMoreMenu(bottomNav);
        return false; // Do not replace fragment
    }

    if (selectedFragment != null) {
        getSupportFragmentManager().beginTransaction()
            .replace(R.id.fragment_container, selectedFragment)
            .commit();
        return true;
    }
    return false;
});

// ✓ Load default fragment (Dashboard)
if (savedInstanceState == null) {
    getSupportFragmentManager().beginTransaction()
        .replace(R.id.fragment_container, new DashboardFragment())
        .commit();
}
}

// ✓ "More" Popup Menu Handling
private void showMoreMenu(View anchor) {
    PopupMenu popup = new PopupMenu(this, anchor);
    MenuInflater inflater = popup.getMenuInflater();
    inflater.inflate(R.menu.menu_more, popup.getMenu());

    popup.setOnMenuItemClickListener(item -> {
        Fragment selectedFragment = null;
        int id = item.getItemId();

        if (id == R.id.nav_alerts) {
            selectedFragment = new AlertsFragment();
        } else if (id == R.id.nav_settings) {
            selectedFragment = new SettingsFragment();
        } else if (id == R.id.nav_teleconsult) {
            selectedFragment = new TeleConsultFragment();
        }
    });
}

```

```

        if (selectedFragment != null) {
            getSupportFragmentManager().beginTransaction()
                .replace(R.id.fragment_container, selectedFragment)
                .commit();
            return true;
        }
        return false;
    });

    popup.show();
}
}

```

### **AndroidManifest.xml**

```

<manifest
xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.patientmonitoring">

    <!-- Permissions for video call & internet -->
    <uses-permission android:name="android.permission.INTERNET" />
    <uses-permission android:name="android.permission.CAMERA" />
    <uses-permission
android:name="android.permission.RECORD_AUDIO" />

    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:supportsRtl="true"
        android:theme="@style/Theme.PatientMonitoring">

        <activity
            android:name=".MainActivity"
            android:exported="true">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

```

```
        <category android:name="android.intent.category.LAUNCHER"
/>
    </intent-filter>
</activity>

</application>
</manifest>
```

### **bottom\_nav\_menu.xml**

```
<?xml version="1.0" encoding="utf-8"?>
<menu xmlns:android="http://schemas.android.com/apk/res/android">

    <item
        android:id="@+id/navigation_dashboard"
        android:icon="@drawable/ic_dashboard"
        android:title="Dashboard" />

    <item
        android:id="@+id/navigation_vitals"
        android:icon="@drawable/ic_vitals"
        android:title="Vitals" />

    <item
        android:id="@+id/navigation_patients"
        android:icon="@drawable/ic_patients"
        android:title="Patients" />

    <item
        android:id="@+id/navigation_reports"
        android:icon="@drawable/ic_reports"
        android:title="Reports" />

    <item
        android:id="@+id/navigation_more"
        android:icon="@drawable/ic_more"
        android:title="More" />
```

</menu>

### **menu\_more.xml**

```
<?xml version="1.0" encoding="utf-8"?>
```

```
<menu xmlns:android="http://schemas.android.com/apk/res/android">
```

```
  <item
```

```
    android:id="@+id/nav_alerts"
```

```
    android:icon="@drawable/ic_alert"
```

```
    android:title="Alerts" />
```

```
  <item
```

```
    android:id="@+id/nav_settings"
```

```
    android:icon="@drawable/ic_settings"
```

```
    android:title="Settings" />
```

```
  <item
```

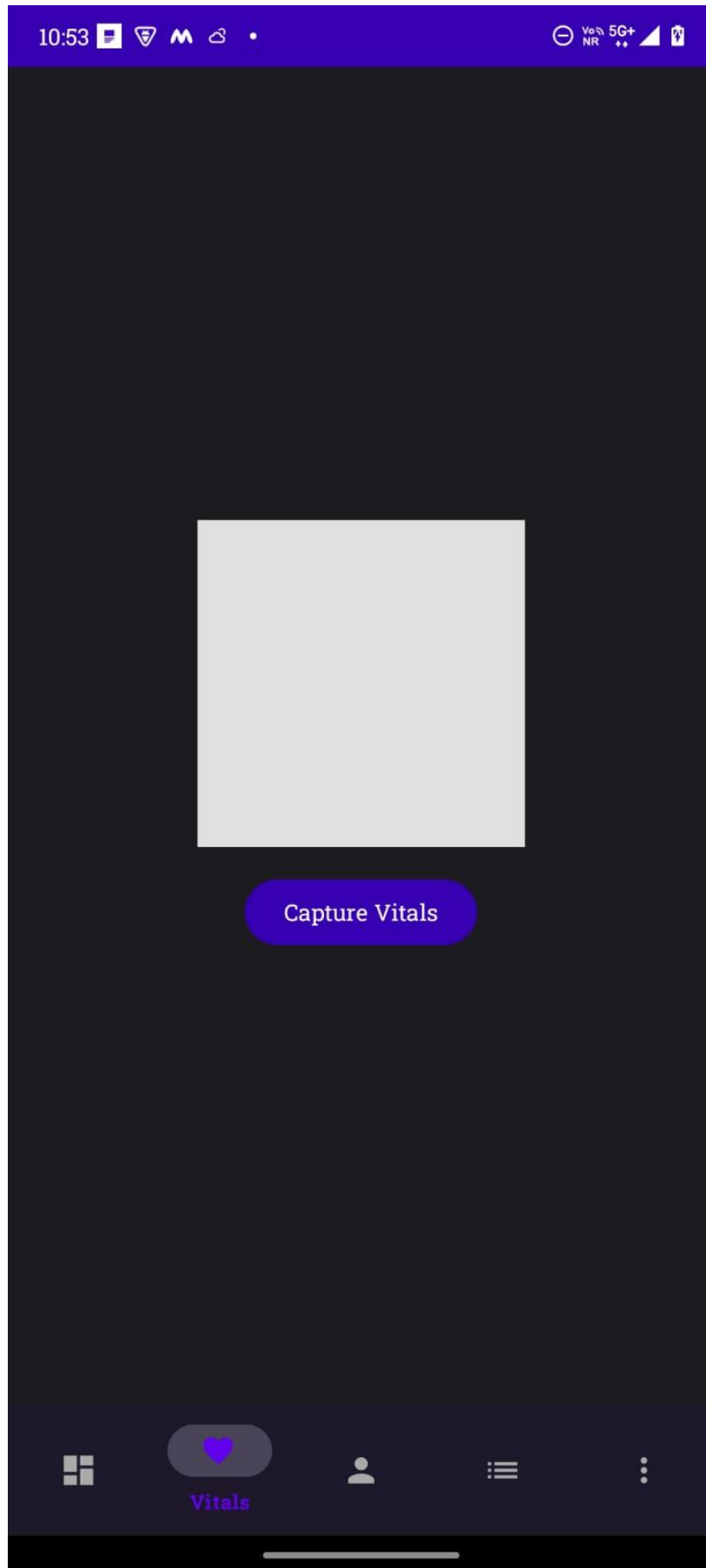
```
    android:id="@+id/nav_teleconsult"
```

```
    android:icon="@drawable/ic_teleconsult"
```

```
    android:title="TeleConsult" />
```

```
</menu>
```

## App Screenshots:



[+ Add Patient](#) **Rochana**

Age: 21

Condition: Mild Fever

Blood Pressure: 118/80

Heart Rate: 78 bpm

 [Edit](#)  [Delete](#) **vasanth**

Age: 21

Condition: Migraine

Blood Pressure: 125/80

Heart Rate: 76 bpm

 [Edit](#)  [Delete](#) **vignesh**

Age: 21

Condition: cold

Blood Pressure: 112/80

Heart Rate: 79 bpm

 [Edit](#)  [Delete](#) **vignesh s**

Age: 35

Condition: heart problem

Blood Pressure: 146/80

Heart Rate: 108 bpm

 [Edit](#)  [Delete](#)

Patients





 **Rochana**

 Rochana – All vitals are normal.

 **vasanth**

 vasanth – All vitals are normal.

 **vignesh**

 vignesh – All vitals are normal.

 **vignesh s**

 Elevated BP (146/80)

 High Heart Rate (108 bpm)



Reports

Reports





**Dr. Priya Sharma**

Cardiologist

Contact: priya@hospital.com

Start Video Call



Dashboard



10:54



VoNR 5G+  
NR

× Join meeting

Patient Monitorin...



Enter your name

Join Meeting

Join In Low Bandwidth Mode

## Settings

Enable Notifications



Dark Mode



About App

Version 1.0.0

**Log Out**



## Conclusion:

- ✧ The Patient Monitoring System successfully provides an efficient platform for tracking patient information, vital signs, alerts, and teleconsultation services.
- ✧ By integrating features such as dynamic patient management, real-time alert generation, and doctor interaction through video calling, the application enhances the overall healthcare monitoring process.
- ✧ The system ensures that patient data is stored securely and can be updated or reviewed whenever necessary.
- ✧ Features like alerts for abnormal vitals, editable patient records, and an intuitive dashboard make the application practical for both medical staff and caretakers.
- ✧ Overall, this project demonstrates how mobile technology can support healthcare workflows by offering quick access to patient health data, improving clinical decision-making, and enabling remote monitoring.
- ✧ The app can be further improved with modules like cloud synchronization, AI-based health predictions, and advanced reporting tools.