

AUTOMATED MGNREGA ANDROID SYSTEM

Main Project Report

Submitted by

ANEESHA MARTIN

Reg No : FIT20MCA-2022

*Submitted in partial fulfillment of the requirements for the award of
the degree of*

*Master of Computer Applications
Of*

A P J Abdul Kalam Technological University



FEDERAL INSTITUTE OF SCIENCE AND TECHNOLOGY (FISAT)®

ANGAMALY-683577, ERNAKULAM(DIST)

JULY 2022

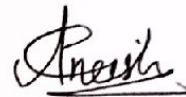
DECLARATION

I, **Aneesha Martin** hereby declare that the report of this project work, submitted to the Department of Computer Applications, Federal Institute of Science and Technology (**FISAT**), Angamaly in partial fulfillment of the award of the degree of Master of Computer Application is an authentic record of my original work.

The report has not been submitted for the award of any degree of this university or any other university.

Date : 12/7/2022

Place: Angamaly



ANEESHA MARTIN

FEDERAL INSTITUTE OF SCIENCE AND TECHNOLOGY (FISAT)

(An ISO 9001:2015 Certified, NAAC ('A' Grade) Accredited Institution with NBA Accredited Programmes
(Approved by AICTE - Affiliated to APJ Abdul Kalam Technological University, Kerala)

Owned and Managed by Federal Bank Officers' Association Educational Society

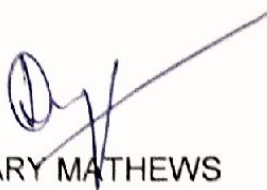
HORMIS NAGAR, MOOKKANNOOR P.O., ANGAMALY - 683 577, ERNAKULAM DT., KERALA, S. INDIA.

Tel: (0) 0484-2725272 Fax: 0484 - 2725250 E-mail: mail@fisat.ac.in Website: www.fisat.ac.in

11TH July 2022

TO WHOMSOEVER IT MAY CONCERN

This is to certify that Mr./Ms. ANEESHA MARTIN (Reg. No. FIT20MCA-2022) has successfully completed his/her Main Project with the title "Automated MGNREGA android System", in the Department of Computer Applications, FISAT, during the period from 30th March 2022 to 11th July 2022.



Dr DEEPA MARY MATHEWS

HEAD OF THE DEPARTMENT



FEDERAL INSTITUTE OF SCIENCE AND TECHNOLOGY (FISAT)®
ANGAMALY, ERNAKULAM-683577

DEPARTMENT OF COMPUTER APPLICATIONS



CERTIFICATE

This is to certify that the project report titled "AUTOMATED MGN-REGA ANDROID SYSTEM" submitted by Aneesha Martin [Reg No: FIT20MCA-2022] towards partial fulfillment of the requirements for the award of the degree of Master of Computer Applications is a record of bonafide work carried out by her during the year 2022.

Project Guide

Ms. Anju L

A handwritten signature in blue ink, appearing to be 'Anju L', is written over the name 'Ms. Anju L'.



Head of the Department

Dr. Deepa Mary Mathews

A handwritten signature in blue ink, appearing to be 'Deepa', is written over the name 'Dr. Deepa Mary Mathews'.

Submitted for the viva-voce held on at

Examiner:

ACKNOWLEDGEMENT

I am deeply grateful to **Dr. Manoj George** , Principal, FISAT, Angamaly for providing me with adequate facilities, way and means by which I was able to complete our mini project work and I express my sincere thanks to **Dr. C Sheela** ,Vice Principal FISAT, Angamaly.

My sincere thanks to **Dr. Deepa Mary Mathews**, Head,Dept. of Computer Applications, FISAT, who had been a source of inspiration. I express heartiest thanks to **Ms. Anju L** my project guide for the encouragement and valuable suggestion . I express my heartiest gratitude to my scrum master **Dr.Shidha M.V** and the faculty members in my department for their constant encouragement and never ending support throughout the project.I would also like to express my sincere gratitude to the lab faculty members for their guidance

Finally I express my thanks to all my friends who gave me wealth of suggestions for successful completion of this project.

ABSTRACT

The Mahatma Gandhi National Rural Employment Guarantee Act, 2005 notified on September 7, 2005 mandate to provide at least 100 days of guaranteed wage employment in a financial year to every rural household whose adult members volunteer to do unskilled manual work. The programme is demand driven. National Rural Employment Guarantee Act provides job to the worker in rural areas as well as their locality by Job card. job card to check new job as well as payment details. Check job card list and know your job card status. Job card number search and get to know the update. You can easily search your job card by providing some basic details or by job card number. This app help and guide you how check job card list, job card number search and many more related to job card. It makes sure that there is no malpractice in the process of registration, distribution of job cards, etc. and also maintains all related registers at the Gram Panchayat level. all the technical sanctions are issued, recorded and monitored by them. This department is responsible for supervising worksites regularly. They also capture daily attendance in muster roll, maintain measurement books at the work sites and update entries in the job card.

Contents

1	INTRODUCTION	8
2	PROOF OF CONCEPT	9
2.1	EXISTING SYSTEM	10
2.2	DRAWBACKS OF EXISTING SYSTEM	11
2.3	PROPOSED SYSTEM	11
2.4	OBJECTIVES	11
3	SOFTWARE DESCRIPTION	12
3.0.1	DEVOLPING PLATFORM	12
3.1	FRONT END	14
3.2	BACK END	14
3.2.1	SYSTEM DESIGN	15
4	PROJECT DESCRIPTION	17
4.1	MODULES	17
4.1.1	REGISTRATION	17
4.1.2	ATTENDANCE	17
4.1.3	PAYMENT	18
4.1.4	JOB MANAGEMENT	18
4.2	USERS	18

5	TESTING AND VALIDATION	19
5.1	UNIT TESTING	20
5.2	INTEGRATION TESTING	22
5.3	UNIT TESTING	22
5.4	USER ACCEPTANCE TESTING	22
5.5	OUTPUT TESTING	23
5.6	VALIDATION CHECK	23
6	CONCLUSION AND FUTURE SCOPE	25
6.1	CONCLUSION	25
6.2	FUTURE SCOPE	27
7	APPENDIX	28
7.1	SOURCE CODE	28
7.1.1	LOGIN ACTIVITY	28
7.1.2	MAIN ACTIVITY	31
7.1.3	SIGNUP ACTIVITY	33
7.1.4	LOGIN ACTIVITY	36
7.2	SCREENSHOT	37
7.3	TABLES	38
7.3.1	WORK	38
7.3.2	USER	39
7.3.3	WORK_LABOUR	40
8	REFERENCES	49

Chapter 1

INTRODUCTION

Mahatma Gandhi National Rural Employment Guarantee Act Has Inherent Provisions For Proactive Disclosure Of Information To Its Citizens In Reference To Implementation Of The Mgnregs. Citizen Awareness Is A Key To Efficient, Effective And Transparent Execution Of The Scheme, While The Programme Rests On Participation Principles. Paradigm Shift From An Employment-Oriented To A Sustainable And Productive Livelihood Support Programme Warrants Greater Public Engagement. Mobile Phone Acts As A Mode Of Communication With People At Farther Places, With Greater Ease And Lesser Time

This application can be used by the authorized personnel and the general public to get and give updates about the status of various works that has been implemented under the Mahatma Gandhi National Rural Employment Gurantee Act.

The app can be used by anyone to report the status of a particular work. This app is also used by the authorized personnel of the Rural Development and Panchayat Raj Department to take photographs of the works and to enter the e-Measurements and create e-Estimates.

Chapter 2

PROOF OF CONCEPT

Now it's easier that you can know about all the services provided by the MGNREGS in our smartphone. This application aims in providing a more user-friendly through which more number of common people can achieve their needs. It also focuses on providing successful completion of requirements within short period of time. It incorporates various functionalities such as Labour job card, registration, job management and so on which will in turn make the system more worth. The MGNREGS was initiated with the objective of enhancing livelihood security in rural areas by providing at least 100 days of guaranteed wage employment in a financial year, to every household whose adult members volunteer to do unskilled manual work. This project is very useful for the people. This system is very safe and secure than the old system. It saves money of the people as they do not need to travel to know many things about their job card, attendance, salary etc

As the part of the project, to know more about the services, I visited Manjapra Panchyat. I met MGNREGA sir and he told us about the different services they are providing. The objective of the MGNREGA is to enhance livelihood security in rural areas by providing at least 100 days of guaranteed wage employment in a financial year to every household whose adult members volunteer to do unskilled manual work. I came know about the services. I learned about the pension , payment, attendance, job card etc. They can view the job card, attendance and

their work.

MGNREGA gives legal guarantee of wage employment to the adult members of rural households who are willing to do unskilled manual labour subject to a maximum of 100 days per household. Every rural household has the right to register under MGNREGA. Job cards are issued to every household registered under MGNREGA within 15 days from the date of receipt of the application for the job card registration.

The registered job cardholders can seek employment by giving a group / individual application. A minimum of 10 job seekers shall apply to sanction a new work under MGNREGA. This project will be very useful for the users.

The old system is pen and paper and is still using in the MGNREGS and is not safe to store all the details in pen and paper. So new system is the android application Automated MGNREGA

2.1 EXISTING SYSTEM

In existing system, they don't have an application for the Automated MGNREGA Android System. Now they are using pen and paper to save all the information related to the MGNREGA. This system is not safe enough to follow in today's world. Even when the flood came, we know that many important documents were collapsed during those days. In existing system, the farmers are not aware of the services provided by the MGNREGA on a timely basis. They have a job card, in which they can know how much work is completed and how much work is there to complete. They can also know their salary details of their work done.

2.2 DRAWBACKS OF EXISTING SYSTEM

- It is not safe.
- Use of book and paper.
- There is no application.
- It is very expensive.

2.3 PROPOSED SYSTEM

In proposed system, we have developed an application for the MGNREGA. All the details will be saved in the application itself. As it is a world of technology, we all use smartphones. All the data will be updated on a timely basis and farmers can be aware of the services on time. The job card is set into the application so that it won't be lost any time. With a single click we can see the job card. We can save so much of time if the job card is there in the smart phone itself

2.4 OBJECTIVES

- Main aim is to make people aware of the services provided by the MGNREGA on a timely basis
- Social protection for the most vulnerable people living in rural India by guaranteeing wage employment opportunities.
- Enhance livelihood security of the rural poor through generation of wage employment opportunities in works leading to creation of durable assets.

Chapter 3

SOFTWARE DESCRIPTION

3.0.1 DEVOLPING PLATFORM

Android software development is the process by which new applications are created for the Android operating system. Applications are usually developed in the Java programming language using the Android software Development Kit (SDK), but other development environments are also available. The Android software development kit (SDK) 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 OSX10.5.8 or later, and Windows XP or later. The SDK is not currently available on Android; however software can be developed 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) plug-in, through IntelliJ IDEA IDE (all editions) fully supports Android Development out of the box, and NetBeans IDE also supports Android Development via a plug-in. As of 2015, Android Studio, made by Google and powered by intellij, is the officials IDE;

However, developers are free to use others. Additionally, develop-

ers 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(eg: triggering a reboot, installing software packages(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 packed 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).

3.1 FRONT END

ANDROID

All user interface elements in an Android app are built using View and View Group objects. A View is an object that draws something on the screen that the user can interact with. A View Group is an object that holds other View (and View Group) objects in order to define the layout of the interface. Android provides a collection of both View and View Group subclasses that offer you common input controls (such as buttons and text fields) and various layout models (such as a linear or relative layout)

JAVA

Java was conceived by James Gosling, Patrick Naughton, at Sun Microsystems in 1991. This language was initially called ‘Oak’ but was renamed ‘java’ in 1995. It is a Language grounded in the need and experience of the people who device it. This reason why Java becomes so important is described in its buzzards, Simple, Object-oriented, Secure, portable, Robust, Multithreaded, Architectural-neutral, Interpreted.

3.2 BACK END

MYSQL MySQL “My S-Q-L” is an open source relational database management system (RDBMS). MySQL is free and open-source software under the terms of the GNU General Public License, and is also available under a variety of proprietary licenses. MySQL was owned and sponsored by the Swedish company MySQL AB, which was bought by Sun Microsystems (now Oracle Corporation).In 2010, when Oracle acquired Sun, Wideners forked the opensource MySQL project to create MariaDB. MySQL is a component of the LAMP web application software stack (and others), which is an acronym for Linux, Apache, MySQL, Perl/PHP/Python. MySQL is used by many database-driven web applications, including WordPress, Drupal, and phpBB.

3.2.1 SYSTEM DESIGN

USE CASE DIAGRAM

A use case diagram is used to represent the dynamic behavior of a system. It encapsulates the system's functionality by incorporating use cases, actors, and their relationships. It models the tasks, services, and functions required by a system/subsystem of an application. It depicts the high-level functionality of a system and also tells how the user handles a system.

The main purpose of a use case diagram is to portray the dynamic aspect of a system. It accumulates the system's requirement, which includes both internal as well as external influences. It invokes persons, use cases, and several things that invoke the actors and elements accountable for the implementation of use case diagrams. It represents how an entity from the external environment can interact with a part of the system. It is essential to analyze the whole system before starting with drawing a use case diagram, and then the system's functionalities are found. And once every single functionality is identified, they are then transformed into the use cases to be used in the use case diagram.

After that, we will enlist the actors that will interact with the system. The actors are the person or a thing that invokes the functionality of a system. It may be a system or a private entity, such that it requires an entity to be pertinent to the functionalities of the system to which it is going to interact. Once both the actors and use cases are enlisted, the relation between the actor and use case/ system is inspected. It identifies the no of times an actor communicates with the system.

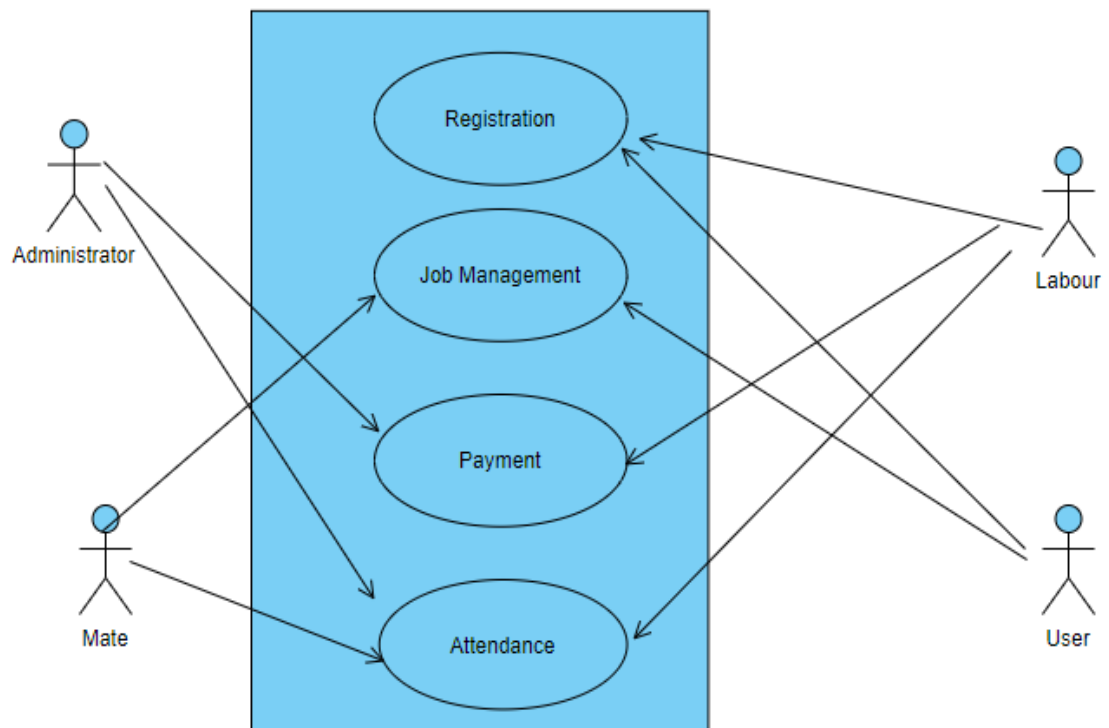


Figure 3.1: AUTOMATED MGNREGA ANDROID SYSTEM

Chapter 4

PROJECT DESCRIPTION

4.1 MODULES

A module is a separate unit of software or hardware. Typical characteristics of modular components include portability, which allows them to be used in a variety of systems, and interoperability, which allows them to function with the components of other systems. The term was first used in architecture.

4.1.1 REGISTRATION

In registration module, it contains details of registration and login. New user must register into the app. Registered users can login into the app and use the app properly. To use this app registration is an important step

4.1.2 ATTENDANCE

In attendance module, it contains the details related to attendance. Attendance is an important concept in life. So in this app the mates can mark the attendance of the labour. They can also view thier daily attendance. All the data's are directly saved to the database. We can view the data anytime.

4.1.3 PAYMENT

In payment module, it contains all the details related to payment. Admin has all the control over the payment. Admin updates all the information related to payment. Labours are the persons who do the work, so they can view their salary.

4.1.4 JOB MANAGEMENT

This module is related to job management. Job management means we can manage the job. If we want some people to clean our surroundings, then we can initiate the people to do the job. Mates can also initiate the work to be done. All the works done will be saved in the database

4.2 USERS

Administrator: It makes sure that there is no malpractice in the process of registration, distribution of job cards, etc. and also maintains all related registers at the Gram Panchayat level. all the technical sanctions are issued, recorded and monitored by them.

Mates: This department is responsible for supervising worksites regularly. They also capture daily attendance in muster roll, maintain measurement books at the work sites and update entries in the job card.

Labours: They are the workers of this scheme. They can know about their registrations and the hours of works they have done. They can view their job card.

Panchayath members (Users): If we need someone to clean our surroundings, we can initiate it. We can choose how many persons we want.

Chapter 5

TESTING AND VALIDATION

Testing is an important stage in the software development life cycle. System testing is a critical element of a software quality assurance and represents the ultimate review of specification, design and coding. Testing is one-way developers can validate the quality of a software product and verify that it fully meets the specification. During testing, the system is tested with a set of cases and checked whether the input of the program is performing as it is expected. The system tested and reviewed to ensure that the entire user requirement has being satisfied. Testing was done throughout the system development at various stages since it is always a good practice to test the system at many different levels at various intervals that is sub systems, program modules as work progress and finally the system as a whole. If this is not done, then the poorly tested system can fail after installation. Testing is a very tedious and time-consuming job. For a test to be successful the tester should try and make the program file. Each test is designed with the intention of finding errors in the way system will process it. Though testing of a program doesn't guarantee the reliability of the system, it is done to assure that the system runs errors free. The Testing process begins by developing a comprehensive plan to test the general function- laity and special features on a variety of platform combinations. Strict quality control procedures are used. The Process verifies that the application meets the requirements specified in the sys-

tem requirements document and is bug free. At the End of each testing day, the summary of completed and failed tests is prepared. And the Application is redeveloped and retested until every item is resolved. All the changes and retesting are tracked through spread sheets. Applications are not allowed to launch until all identified problem are fixed. Finally, a report is prepared at the end of testing to show exactly what was tested and to list the final outcomes. The software testing methodology is applied in four distinct phases:

- Unit Testing
- Integration Testing
- User Acceptance Testing
- Output Testing

ADVANTAGE OF SYSTEM TESTING

The testers do not require more knowledge of programming to carry out this testing. It will test the entire product or software so that we will easily detect the errors or defects which cannot be identified during the unit testing and integration testing. The testing environment is similar to that of the real time production or business environment. It checks the entire functionality of the system with different test scripts and also it covers the technical and business requirements of clients. After this testing, the product will almost cover all the possible bugs or errors and hence the development team will confidently go ahead with acceptance testing.

5.1 UNIT TESTING

The first level of testing is unit testing. This is also known as “Module Testing”. To check whether each module in the software is proper and it gives desired outputs to the given inputs. All validations and conditions are tested in the module in the first unit. The goal is to test the internal logic of the module. In unit testing

each unit are tested during the programming stage itself. Developers typically do unit testing in order to trace out bugs in each module of the code. Unit testing is done in parallel with coding. It includes testing each function and procedures. In module testing each module are tested for any possible logical error. They are also tested for specification to see if they are working as per the program should do and they are tested under various conditions. Each module is being tested thoroughly in order to discover pitfalls. Specification testing examines the specification what the program should do and how it should perform under various conditions. The testing will be done by entering data into different tables using forms. The data with less validation will be tested first. Whenever an error is encountered, an informative error message will be displayed which informs user about the type of error. Testing and Validation testing the program will be tested. The unit testing is done to identify:

- The insertion of values in the database The updating of values in to the database.
- The deletion of values from the database.
- Response of the system to inputs
- All the needed forms are checked for validation.
- The correctness of the phone number format (10-digit number).
- Whether the entries are in correct format.
- Email validation (email address must contain at least @ sign and a dot(.))
- Characters or numbers are within the limit given (proper working of all fields, response of the system is delaying or not etc).

5.2 INTEGRATION TESTING

The modules are integrated to form complete software package. It addresses the issues associated with given problem of verification and program construction. Test that part of the system at some level work together correctly the purpose of integration is to verify functional, performance and reliability requirements placed on major design items. This testing is conducted on the basis of modules. The integration testing is performed to detect design errors by focusing on testing the interconnection between modules. The objective is to take the unit tested modules are combined and tested as a whole. Integration testing is any type of software testing that seeks to verify the interfaces between components against a software design. Software components may be integrated in an iterative way or all together ("big bang"). Normally the former is considered a better practice since it allows interface issues to be located more quickly and fixed. Integration testing works to expose defects in the interfaces and interaction between integrated. Testing and Validation components (modules). The task of the integration test is to check that components or software applications, interacts without error. Therefore, testing the data flow between 2 modules is integration testing

5.3 UNIT TESTING

Unit testing, a testing technique using which individual modules are tested to determine if there are any issues by the developer himself. It is concerned with functional correctness of the standalone modules. The main aim is to isolate each unit of the system to identify, analyse and fix the defects.

5.4 USER ACCEPTANCE TESTING

This is the key factor for the success of any system. User acceptance is done by constantly keeping in touch with prospective system at the time of development

and making changes whenever required. This is done with regard to the input and output screen designs. User acceptance testing is done by the user to check whether the project has met the requirement that has been mentioned at the beginning of the project.

5.5 OUTPUT TESTING

After performing the validation testing, the next step is output testing of the proposed system since no system could be useful if it does not produce the required output in the specific format. The output generated or displayed by the system under consideration is tested asking the users about the format required by them. Here, the output is considered in two ways, one is on the screen and other is printed format. The output format on the screen is found to be correct as the format designed according to the user needs. For the hard copy also, the output comes out as specified by the user. Hence output testing does not result in any connection in the system.

5.6 VALIDATION CHECK

Validation Testing is the process of evaluating software during the development process or at the end of the development process to determine whether it satisfies specified business requirements. Validation Testing ensures that the product actually meets the client's needs. Validation testing can be defined in many ways, but a simple definition is that validation succeeds when the software functions in a manner that can reasonable expected by the user. That means the input we are entering to the system will be checked. Testing and Validation the entered input is valid or not. The validation phase reveals the failures and the buds in the developed system. It will become to known about the practical difficulties the system faces when the operated in the true environment. Validation is the process of ensuring that user input is clean, correct, and useful. Validation can be defined by many

different methods, and deployed in many different ways

Chapter 6

CONCLUSION AND FUTURE SCOPE

6.1 CONCLUSION

Mahatma Gandhi National Rural Employment Guarantee Act (MGN-REGA) has inherent provisions for proactive disclosure of information to its citizens in reference to implementation of the MGNREGA. Citizen awareness is a key to efficient, effective and transparent execution of the scheme, while the programme rests on participation principles. Paradigm shift from an employment-oriented to a sustainable and productive livelihood support programme warrants greater public engagement.

Mahatma Gandhi National Rural Employment Guarantee Act (MGN-REGA) is an android application which is developed for Manjapra panchayath. By using this app, MGNREGA will be aware of the services of a timely basis. This app will be very useful for the users, mates and labours to know about all the services like daily attendance, set wages and work management. It saves time and effort of people. Thus it brings an innovative idea by which the Labours of Manjapra panchayath can become aware of all the services provided by the

MGNREGA to people. They provide different services to the people for supporting. This application will be very useful for the people as admin updates all the information and registered users can view and apply for their different services.

6.2 FUTURE SCOPE

The technology changes at a fast pace and the people should also change. In existing system, there is no application for the people of this panchayath and all the details are saved in paper and pen, so its not that much safe. So this application will be very useful for the people as admin updates the information and users can view the details on correct time using their own smartphones. Pension module can be updated by adding the pension payment details. This application will be very useful for the people as they get the updated details and news on time. Different services can be known by using a single application. The main problem is the lack of technology, but by developing this application the problem will be solved.

Chapter 7

APPENDIX

7.1 SOURCE CODE

7.1.1 LOGIN ACTIVITY

```
package com.dzniox.e_panchayath.ui.activities;
import androidx.appcompat.app.AppCompatActivity;
import android.content.Intent;
import android.os.Bundle;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;
import com.dzniox.e_panchayath.R;
import com.dzniox.e_panchayath.network.APIClient;
import com.dzniox.e_panchayath.network.APIInterface;
import com.dzniox.e_panchayath.sharedPref.PrefKeys;
import com.dzniox.e_panchayath.sharedPref.PrefUtils;
import org.json.JSONObject;
import retrofit2.Call;
```

```
import retrofit2.Callback;
import retrofit2.Response;
public class LoginActivity extends AppCompatActivity
private Button Login,SignUp;
private EditText Email,Password;
APIInterface apiInterface;
PrefUtils preferences;
@Override
protected void onCreate(Bundle savedInstanceState)
super.onCreate(savedInstanceState);
setContentView(R.layout.activity_login);
apiInterface = APIClient.getClient().create(APIInterface.class);
preferences = PrefUtils.getInstance(this);
Login = findViewById(R.id.login);
SignUp = findViewById(R.id.signup);
Email = findViewById(R.id.email);
Password = findViewById(R.id.password);
if (preferences.getBooleanValue(PrefKeys.IS_LOGGED_IN,false))
GoToMainActivity();
else
SignUp.setOnClickListener(view ->
Intent intent = new Intent(this, SignupActivity.class);
startActivity(intent);
finish(); );
Login.setOnClickListener(view ->
if (Email.getText().toString().trim().equals(""))
Password.getText().toString().trim().equals(""))
Toast.makeText(this,"Check
Credentials",Toast.LENGTH_LONG).show();
else
```

```
DoLogin();

);
private void DoLogin()
try
Call<String> call = apiInterface.loginUser(
Email.getText().toString().trim(),
Password.getText().toString()); call.enqueue(new Callback<String>()
@Override
public void onResponse(Call<String> call,
Response<String> response)
if (response.code() == 200)
try
JSONObject data = new JSONObject(response.body());
if (data.getJSONArray("data").length() < 0)
preferences.setValue(PrefKeys.IS_LOGGED_IN, true);
preferences.setValue("name",data.getJSONArray("data").getJSONObject(0)
.getString("name"));
preferences.setValue("email",data.getJSONArray("data").getJSONObject(0)
.getString("email"));
preferences.setValue("id",data.getJSONArray("data").getJSONObject(0).getString("id"));
preferences.setValue("age",data.getJSONArray("data").getJSONObject(0).getString("age"));
preferences.setValue("phone",data.getJSONArray("data").getJSONObject(0)
.getString("phone"));
preferences.setValue("user_type",data.getJSONArray("data").getJSONObject(0).getString("user_type"));
preferences.setValue("dob",data.getJSONArray("data").getJSONObject(0).getString("dob"));
preferences.setValue("panchayath",data.getJSONArray("data").getJSONObject(0).getString("panchayath"));
preferences.setValue("ward",data.getJSONArray("data").getJSONObject(0).getString("ward"));
preferences.setValue("aadhar",data.getJSONArray("data").getJSONObject(0).getString("aadhar"));
GoToMainActivity();
```

```
catch (Exception e)
@Override
public void onFailure(Call<String> call, Throwable t)

String a = t.toString();
catch (Exception e)
Toast.makeText(this,e.toString(),Toast.LENGTH_SHORT).show();

private void GoToMainActivity()
Intent intent = new Intent(this, MainActivity.class);
startActivity(intent);
finish();
```

7.1.2 MAIN ACTIVITY

```
Package com.dzniox.e_panchayath.ui.activities;
import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;
import androidx.fragment.app.Fragment;
import androidx.fragment.app.FragmentTransaction;
import android.content.Intent;
import android.os.Bundle;
import android.view.MenuItem;
import android.view.View;
import com.dzniox.e_panchayath.R;
import com.dzniox.e_panchayath.sharedPref.PrefUtils;
```



```
import com.dzniox.e_panchayath.ui.fragments.ProfileFragment;
import com.dzniox.e_panchayath.ui.fragments.WorkFragment;
import com.google.android.material.bottomnavigation.BottomNavigationView;
public class MainActivity extends AppCompatActivity
    PrefUtils preferences;
    private BottomNavigationView nav_view;
    private WorkFragment workFragment;
    private ProfileFragment profileFragment;
    @Override
    protected void onCreate(Bundle savedInstanceState)
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main); preferences = PrefUtils.getInstance(this);
    workFragment = new WorkFragment();
    profileFragment = new ProfileFragment();
    nav_view = findViewById(R.id.nav_view);
    LoadFragment(workFragment);
    nav_view.setOnNavigationItemSelectedListener(new BottomNavigationView.OnNavigationItemSelectedListener() {
        @Override
        public boolean onNavigationItemSelected(@NonNull MenuItem item)

        switch (item.getItemId())
        case R.id.navigation_home:
            LoadFragment(workFragment);
            break;
        case R.id.navigation_profile:
            LoadFragment(profileFragment);
            break;

        return true;
```

```
);

public void GoToback(View view)
public void LoadFragment(Fragment fragment)
final FragmentTransaction transaction = getSupportFragmentManager().beginTransaction();
transaction.replace(R.id.container, fragment);
transaction.addToBackStack(null);
transaction.commit();

public void LogOut(View view)
preferences.clear();
Intent intent = new Intent(this, SplashScreenActivity.class);
startActivity(intent);
finish();

}
```

7.1.3 SIGNUP ACTIVITY

```
package com.dzniox.e_panchayath.ui.activities;
import androidx.appcompat.app.AppCompatActivity;
import android.content.Intent;
import android.os.Bundle;
import android.widget.Button;
import android.widget.CheckBox;
import android.widget.EditText;
import android.widget.TextView;
import android.widget.Toast;
import com.dzniox.e_panchayath.R;
import com.dzniox.e_panchayath.network.APIClient;
```

```
import com.dzniox.e_panchayath.network.APIInterface;
import retrofit2.Call;
import retrofit2.Callback;
import retrofit2.Response;
public class SignupActivity extends AppCompatActivity
private Button SignUp;
private EditText
Name,Email,Password,Age,DOB,Panchayath,Ward,Aadhar,Phone; private TextView
Login;
private CheckBox user_type;
APIInterface apiInterface;
private String UserType = "4";
@Override
protected void onCreate(Bundle savedInstanceState)
super.onCreate(savedInstanceState);
setContentView(R.layout.activity_signup);
apiInterface = APIClient.getClient().create(APIInterface.class);

SignUp = findViewById(R.id.signup);
Name = findViewById(R.id.name);
Email = findViewById(R.id.email);
Age = findViewById(R.id.age);
DOB = findViewById(R.id.dob);
Phone = findViewById(R.id.phone);
Panchayath = findViewById(R.id.panchayath);
Ward = findViewById(R.id.wardno); Aadhar = findViewById(R.id.aadhar);
Password = findViewById(R.id.password);
Login = findViewById(R.id.login);
user_type = findViewById(R.id.usertype);
Login.setOnClickListener(view ->
```

```
Intent intent = new Intent(this, LoginActivity.class);
startActivity(intent);
finish();
);
SignUp.setOnClickListener(view -> {
    if (usertype.isChecked())
        UserType = "3";
    else
        UserType = "4";
```

```
DoSignUp(Name.getText().toString().trim(),
Email.getText().toString().trim(),
Password.getText().toString().trim(),
Age.getText().toString().trim(),
DOB.getText().toString().trim(),
Panchayath.getText().toString().trim(),
Ward.getText().toString().trim(),
Aadhar.getText().toString().trim(),
Phone.getText().toString().trim());
);
```

```
private void DoSignUp(String name, String email, String
password, String age, String dob, String panchayath, String
ward, String aadhar, String phone) {
    if (name.equals("") || email.equals("") ||
password.equals("") || age.equals("") ||
dob.equals("") || panchayath.equals("") ||
ward.equals("") || aadhar.equals(""))
        Toast.makeText(this, "Check Fields", Toast.LENGTH_LONG).show();
    else
        Call<String> call = apiInterface.signUpUser(email, password, name, age, dob, panchayath, ward, aadhar,
```

```
call.enqueue(new Callback<String>()
@Override
public void onResponse(Call<String> call, Response<String> response)
if (response.code() == 200)
GoToLoginActivity();

@Override
public void onFailure(Call<String> call, Throwable t );
private void GoToLoginActivity()
Toast.makeText(this,"SignUp
Completed",Toast.LENGTH_LONG).show();
Intent intent = new Intent(this, LoginActivity.class);
startActivity(intent);
finish();
```

7.1.4 LOGIN ACTIVITY

```
package com.dzniox.e_panchayath.ui.activities;
import androidx.appcompat.app.AppCompatActivity;
import android.content.Intent;
import android.os.Bundle;
import android.os.Handler;
import com.dzniox.e_panchayath.databinding.ActivitySplashScreenBinding;
public class SplashScreenActivity extends AppCompatActivity

private ActivitySplashScreenBinding binding;
final Handler handler = new Handler();
@Override
protected void onCreate(Bundle savedInstanceState)
```

```
super.onCreate(savedInstanceState);
binding = ActivitySplashScreenBinding.inflate(getLayoutInflater());
setContentView(binding.getRoot());
handler.postDelayed(new Runnable()
@Override
public void run()
// Do something after 5s = 5000ms
Intent intent = new Intent(SplashScreenActivity.this,
LoginActivity.class);
startActivity(intent);
, 1000);
```

7.2 SCREENSHOT

Here I add some sample screenshots of the proposed system which includes,

- splash page
- Login page
- Labour Registration
- User registration
- attendance
- Set Wage
- Job Management
- Labour Job card

7.3 TABLES

7.3.1 WORK

This is work table. This table is to know attendance of labors, registration of users.

Name	Type	Description
id	int	Work id
User_id	Text	User id
title	Text	Job title
desc	Text	Description
Labour_id	Text	Labour id
First_half	Text	First half attendance
Second_half	Text	Second Half attendance
amount	Text	Labour amount
paid	Text	Paid Salary
address	Text	address
area	Text	Area
ward	Text	Labour Ward
category	Text	Job Category

7.3.2 USER

This table contains the details regarding registration of labours.

Name	Type	Description
id	int	Work id
email	Text	User id
password	Text	Job title
name	Text	Name
dob	Text	Date of Birth
age	Text	Age
phone	Text	Phone number
Panchayath	Text	Panchayat
ward	Text	ward
adhar	Text	Adhar number
User_type	int	User type

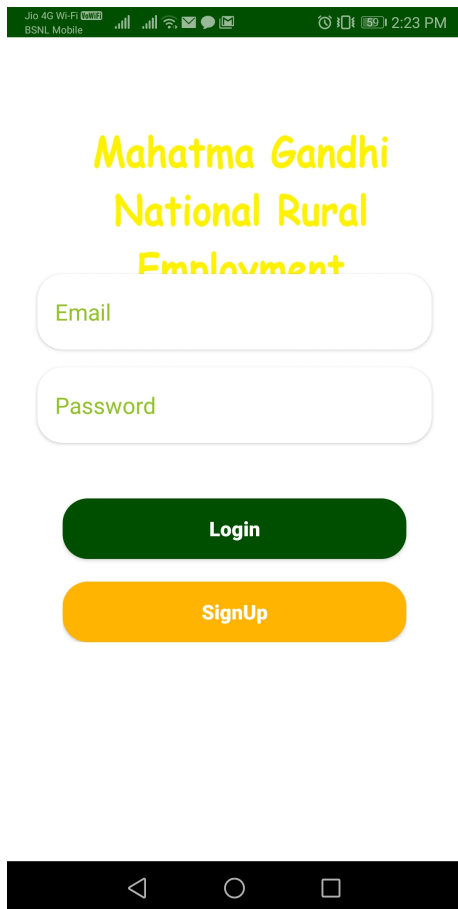
7.3.3 WORK LABOUR

This table contains details of the work done. We can view the details of the labour.

Name	Type	Description
id	int	id
Labour_id	int	Labour id
name	text	name
Work_id	int	Work id




Figure 7.1: splash Page



The screenshot shows the login interface of the Mahatma Gandhi National Rural Employment system on an Android phone. At the top, the status bar displays 'Jio 4G Wi-Fi', 'BSNL Mobile', signal strength, battery level at 99%, and the time 2:23 PM. The app title 'Mahatma Gandhi National Rural Employment' is centered in yellow. Below it are two white input fields with green placeholder text: 'Email' and 'Password'. Under the 'Email' field is a green 'Login' button, and under the 'Password' field is an orange 'SignUp' button. At the bottom, the Android navigation bar shows the back, home, and recent apps icons.

Figure 7.2: Login Page

Sign Up



Ammu

ammu@gmail.com

.....

3690852174

28

1/08/1990

Manjapra

.....

3690852174

28

1/08/1990

Manjapra

123456

Signup as Labour

Login

SIGNUP

Figure 7.3: Labour Registration

The image displays two side-by-side screenshots of an Android application's 'Sign Up' screen. Both screenshots show a registration form with the following fields: Name (filled with 'Nimmy'), Email (filled with 'nimmy@gmail.com'), Password (masked with '.....'), Aadhaar Number (filled with '3690852147'), Age (filled with '29'), Date of Birth (filled with '01/2/1990'), and Location (filled with 'Manjapra'). The left screenshot shows the form with a 'Login' link and a green 'SIGNUP' button at the bottom. The right screenshot shows the same form with a 'Login' link and a green 'SIGNUP' button at the bottom.

Figure 7.4: User Registration

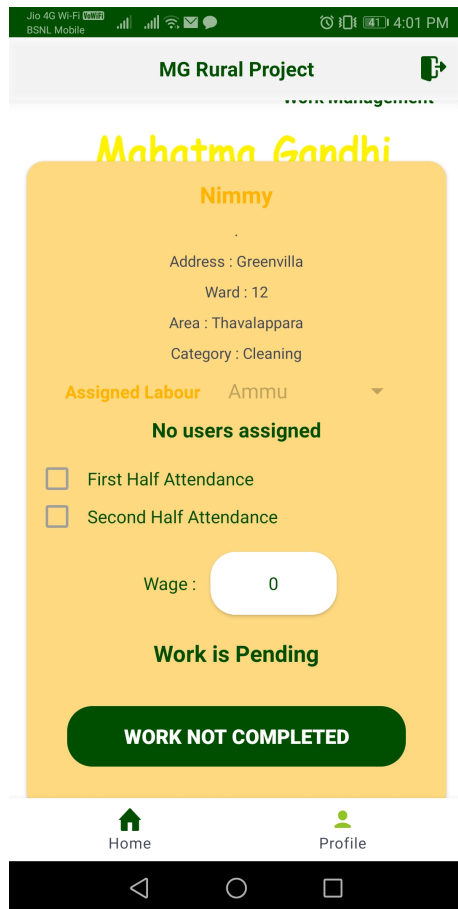


Figure 7.5: attendance

The screenshot shows the 'MG Rural Project' app interface. At the top, the status bar displays 'Jio 4G Wi-Fi', 'BSNL Mobile', and the time '4:01 PM'. Below the status bar, the app title 'MG Rural Project' is visible. The main content area is titled 'Mahatma Gandhi' and features a yellow card for a user named 'Nimmy'. The card displays the following information: Address : Greenvilla, Ward : 12, Area : Thavalappara, and Category : Cleaning. Below this, there is a section for 'Assigned Labour' with a dropdown menu showing 'Ammu'. A message 'No users assigned' is displayed. There are two checkboxes for 'First Half Attendance' and 'Second Half Attendance', both of which are unchecked. A 'Wage' field is shown with a value of '0'. Below the wage field, it says 'Work is Pending'. At the bottom of the card, there is a green button labeled 'WORK NOT COMPLETED'. The bottom navigation bar has two icons: 'Home' and 'Profile'. The Android system bar is visible at the very bottom.

Figure 7.6: Set Wage

The screenshot displays the 'MG Rural Project' app interface. At the top, a green status bar shows 'Jio 4G', 'BSNL Mobile', and the time '1:21 PM'. Below this, a grey header bar contains the text 'MG Rural Project' and a green location pin icon. The main content area is a white rounded rectangle containing a form with the following fields: 'Aneesha', '27/7/2022', 'Greenvilla', 'Thavalappara', '12', and 'Cultivation'. A large green 'SUBMIT' button is positioned at the bottom of the form. Below the form, there are two icons: a green house icon labeled 'Home' and a green person icon labeled 'Profile'. At the very bottom, a black Android navigation bar features three white icons: a back arrow, a circle, and a square.

Figure 7.7: Job Management



Figure 7.8: Labour Job card

Chapter 8

REFERENCES

- [1] Wei-Meng Lee, Beginning Android 4 Application Development, Wiley publications, 2012
- [2] Roger. S Pressman, Software Engineering- A Practitioner's Approach, McGraw Hill International Edition, 7th Edition, 2010
- [3] <https://nregs.kerala.gov.in/en/home/>
- [4] Tutorials on Android Programming. Retrieved from <http://tutorialspoint.com/android/>