## GOVERNMENT POLYTECHNIC, PUNE

(An Autonomous Institute of Government of Maharashtra)



DEPARTMENT OF COMPUTER ENGINEERING

ACADEMIC YEAR 2021-22

**PROJECT REPORT ON**

# “GramSwaraj: Digital Gram Panchayat”

**SUBMITTED BY:**

|  |  |
| --- | --- |
| Omkar Dayanand Sapkal | 1906112 |
| Isha Narendra Patil | 1906099 |
| Hindavi Vinodrao Lande | 1906077 |
| Vaishnavi Pramod Pardeshi | 1906093 |

**UNDER THE GUIDANCE OF**

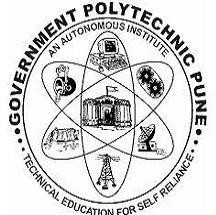
**Prof. G. B. Garud**

**(COMPUTER ENGINEERING DEPARTMENT)**

**GOVERNMENT POLYTECHNIC PUNE**

(An Autonomous Institute Of Government of Maharashtra)

**Department Of Computer Engineering**

ACADEMIC YEAR: 2021-22

CERTIFICATE

This is to certified that

|  |  |
| --- | --- |
| **Omkar Dayananda Sapkal** | **1906112** |
| **Isha Narendra Patil** | **1906099** |
| **Hindavi Vinod Lande** | **1906077** |
| **Vaishnavi Pramod Pardeshi** | **1906093** |

of class Third Year (2021-22) have successfully completed project on **“GramSwaraj : Digital Grampanchayat”** under the guidance of **“Prof. G. B. Garud”** in parallel fulfillment of requirement for the award of Diploma in Computer Engineering from Government Polytechnic, Pune.

…..………………………. …….…….…………  **Dr. Shankar B. Nikam Dr. Vitthal Bandal**  Head of the Department Principal

…..………………………. …….…….…………  **Prof. Gauri B. Garud External Examiner**  Project Guide

**ACKNOWLEDGEMENT**

It is my proud privilege and duty to acknowledge the kind of help and guidance received from several people in the preparation of this report. It would not have been possible to prepare this report in this form without their valuable help, cooperation and guidance.

First and foremost, I wish to record my sincere gratitude to the Management of this college and to our Respected Principal Dr. V. S. Bandal, for his constant support and encouragement in the preparation of this report and for the availability of library and laboratory facilities needed to prepare this report.

My sincere thanks to Dr. Shankar B. Nikam, Head of department, Computer Department, Government Polytechnic, Pune for her valuable suggestions and guidance throughout the preparation of this report.

I express my sincere gratitude to my Guide, Mrs. Gauri Garud for guiding me in investigations of this case study and in carrying out experimental work. Our numerous discussions were extremely helpful. I hold him in esteem for guidance, encouragement and inspiration received from him.

We wish to thank our parents for financing our studies and helping us throughout our life for achieving perfection and excellence. Their personal help in making this report and project worth presentation is gratefully acknowledged. Last but not the least we thank the Almighty for continuous strength we were bestowed for completion of this report.

|  |  |
| --- | --- |
| **Omkar Dayananda Sapkal** | **[1906112]** |
| **Isha Narendra Patil** | **[1906099]** |
| **Hindavi Vinod Lande** | **[1906077]** |
| **Vaishnavi Pramod Pardeshi** | **[1906093]** |

**TABLE OF CONTENTS**

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Contents** | | **Page No.** |
| 1 | Introduction | | 1 |
| 2 | Project Plan | | 2 |
|  | 2.1 | Software Model | 2 |
|  | 2.2 | Approach in project development | 2 |
|  | 2.3 | Roles and Responsibilities | 2 |
|  | 2.4 | Project Schedule | 3 |
| 3 | Requirement Analysis | | 4 |
|  | 3.1 | Hardware Requirement | 4 |
|  | 3.2 | Software Requirement | 4 |
|  | 3.3 | Performance Requirement | 4 |
|  | 3.4 | Safety Requirement | 4 |
|  | 3.5 | Security Requirement | 4 |
| 4 | Project Design | | 6 |
|  | 4.1 | Feasibility of Project | 6 |
|  | 4.2 | Technologies Used | 6 |
|  | 4.3 | Tools Used | 8 |
|  | 4.4 | Diagrams | 10 |
|  | 4.5 | Project Output | 17 |
| 5 | System Testing | | 29 |
|  | 5.1 | Testing Types | 29 |
|  | 5.2 | Test Cases | 30 |
| 6 | Future Scope | | 32 |
| 7 | Conclusion | | 33 |
| 8 | References | | 34 |

**LIST OF FIGURES**

|  |  |  |
| --- | --- | --- |
| **Figure No.** | **Figure Name** | **Page No.** |
| 1 | Level 0 DFD | 10 |
| 2 | Level 1 DFD | 11 |
| 3 | Level 2 DFD (Admin) | 11 |
| 4 | Level 2 DFD (Public) | 12 |
| 5 | ER Diagram | 12 |
| 6 | Class diagram | 13 |
| 7 | Sequence diagram | 13 |
| 8 | Use case diagram | 14 |
| 9 | Activity diagram | 15 |
| 10 | Component diagram | 15 |
| 11 | Deployment Diagram | 16 |
| 12 | Object Diagram | 16 |

**LIST OF TABLES**

|  |  |  |
| --- | --- | --- |
| **Table No** | **Table Name** | **Page No** |
| 1 | Roles and Responsibilities | 2 |
| 2 | Project Schedule | 3 |
| 3 | Testing types | 29 |
| 4 | Test cases | 30 |

**ABSTRACT**

Over the past two decades, India’s ability to create, select, adapt, use and profit from knowledge has become increasingly important to its sustainable economic growth and improvement of living standard. The growing role of Information and Communication Technology (ICT) tools has strengthened the human intellectual capacity and the formation of modern lifestyles.

With this background we have initiated and rolled forward the concept of ‘GramSwaraj: Digital Panchayat’ Application. Digital Panchayat means – an application based dynamic digital interface created for each and every Panchayat in Maharashtra, giving information about particular panchayat in a two-way flow of content. It is user friendly application, as it is developed in android programming language, with back-end technology of ‘Google’s Firebase Cloud service’ also mainly platform independent Java Programming language. The objective is to facilitate and improve Panchayat functioning on day-to-day basis, through two-way flow of information and content. The main objective of this project is to reach government service in a digital way. With the help of this project better communication may be developed between local village governing authority and the public and, in this way, it is helpful in information distribution. Panchayat staff will accept this application and verify and send this application to head office from where application will be approved. At present most of the services like birth/ death registration, income certificate, property / land related documents, etc. are done offline or in paper medium in that the villagers have to go for all the information in the Panchayat office and they face many difficulties.

This paper provides the application for reducing paper work at Gram panchayat and e-services will give the simple servicing. It also encloses the fact of reducing the labor of paper work.

**CHAPTER 1: INTRODUCTION**

The main objective of this project is to reach government service in a digital way. In this online service, the public can see the list of government services in their cell phone and they are able to get information about call register documents and may apply for online services. With the help of this project better communication may be developed between government service providers and the public and in this way, it is helpful in information distribution.

Panchayat staff will accept this application and verify and send this application to head office from where application will be approved. And the users can see the information of all the processes in the option of application status. At present all the services are done offline or in paper medium in that the villagers have to go for all the information in the Panchayat office and they face many difficulties. Enough line method the users may not know the application status in a proper way. They face many difficulties and they are unable to know where their application is posted in a proper way or not.

With the help of digital Gram Panchayat services, the users may fill a form in an online way so they can see all the application status in their mobile in an easy way. That is all the information about whether the application is approved or rejected. The users can get easily information and they don’t have to go Panchayat office many times. And panchayat offices also do their work in an easy way and they also have no loads of file or paper. Every detail of all the schemes will be on one click of user, in particular the user may fill the form.

**CHAPTER 2: PROJECT PLAN**

**2.1 Software Model:**

* For this project, we use iterative model.
* Iterative process starts with a simple implementation of a subset of the software requirements and iteratively enhance the evolving version until the full system is implemented.
* At each iteration design modification are made and new functional capabilities are added. The basic idea behind this method is to develop a system through is to develop a system through repeated cycles and in smaller portions at a time

**2.2 Approach in project development**

The project is developed following the SMART approach

* Specific
* Measurable
* Achievable
* Realistic
* Time-bound

**2.3 Roles and responsibilities**

|  |  |
| --- | --- |
| **Roles** | **Responsibilities** |
| Omkar Dayanand Sapkal (1906112) | Requirement gathering, Backend Development, Diagrams, Frontend development |
| Isha Narendra Patil (1906099) | Requirement gathering, Backend Development, Diagrams, Frontend development |
| Hindavi Vinodrao Lande (1906077) | Requirement gathering, Diagrams front end development |
| Vaishnavi Pramod Pardeshi (1906093) | Requirement gathering, Diagrams, frontend development |

Table No. 1 : Roles and Responsibilities

**2.4 Project Schedule**

|  |  |  |
| --- | --- | --- |
| **Weekly Schedule** | **Number of Days** | **Work Done** |
| Week 1 - 28th March | 7 Days | General Discussion of Topic and finalizing the topic from the mentor. |
| Week 2 - 4th April | 7 Days | Requirement analysis and Gathering |
| Week 3 - 11th April | 7 Days | Design the architecture of the project and designed the UML Diagrams. |
| Week 4 - 18th April | 7 Days | Design the Front End for Login and Registration Page for GramSwaraj Application. |
| Week 5 – 2  5th April | 7 Days | Designing Home Page for Digital Gram panchayat app. |
| Week 6 - 2nd May | 7 Days | Implementing proper validation for the Login and Register form and attaching firebase real time database to it. |
| Week 7 - 9th May | 7 Days | Designing the About us Page using android programming, also created the contact us, forms under gram panchayat has been designed. |
| Week 8 - 16th May | 7 Days | Designed navigation drawer for application. And updating the Application as per the guide’s instructions. |
| Week 9 - 23th May | 7 Days | In this period of time GramSwaraj application’s Galary, Yojana’s, Feedback, Contact us modules has been designed by different group members. |
| Week 10 - 30th May | 7 Days | Connecting all the projects with database and collaborate it with other group members. |
| Week 11 - 6th June | 7 Days | Connected all the projects and modules together with the help of intent and GUI finishing of the application has been completed. And the overall complete project is established. |
| Week 12 - 13th June | 5 Days | Testing of code and Report Writing. |

Table No . 2 : Project Schedule

**CHAPTER 3: REQUIREMENT ANALYSIS**

**3.1 Hardware Specification:**

* Hard Disk – 5 GB
* A computer with at least 8GB of RAM(Random Access Memory)
* Processor – i3
* Monitor – Standard LED Monitor
* Input Device -Keyboard

**3.2 Software Requirements:**

* Operating System – Windows 7/8/10
* Coding Software – Android Studio
* Languages- Java
* Database – Firebase
* Code Collaboration- GitHub

**3.3 Performance Requirement:**

Performance of the functions and every module must be well. The overall performance of the software will enable the users to work efficiently. Performance of encryption of data should be fast. Performance of the providing virtual environment should be fast.

**3.4 Safety Requirement:**

The application is designed in modules where errors can be detected and fixed easily. This makes it easier to install and update new functionality if required.

**3.5 Security Requirement:**

All data will be encrypted using strong encryption algorithm and according to location encryption is done. Software quality attributes: Our software has many quality attribute that are given below:-

**• Adaptability:** This software is adaptable by all teachers and students.

**• Availability:** This software is freely available to all techers and students. The availability of the software is easy for everyone.

**• Maintainability:** After the deployment of the project if any error occurs then it can be easily maintained by the software developer

**. • Reliability:** The performance of the software is better which will increase the reliability of the Software.

**• User Friendliness:** Since, the software is a GUI application; the output generated is much user friendly in its behavior.

**• Integrity:** Integrity refers to the extent to which access to software or data by unauthorized persons can be controlled.

**• Security:** Users are authenticated using many security phases so reliable security is provided.

**• Testability**: The software will be tested considering all the aspects

**CHAPTER 4: PROJECT DESIGN**

**4.1 Feasibility of the Project:**

Feasibility study is an important phase of the System Development Life Cycle where the proposed system is viewed to determine the suit ability and feasibility of the system i.e. to determine whether or not the system is worth doing. An important outcome of the preliminary investigation phase is that the package required is feasible and meets the efficiency criteria.

**4.2 Technologies Used:**

* **Android Programming language:**

Android software development is the process by which applications are created for devices running the Android operating system. Google states that "Android apps can be written using Kotlin, Java, and C++ languages" using the Android software development kit (SDK), while using other languages is also possible. All non-Java virtual machine (JVM) languages, such as Go, JavaScript, C, C++ or assembly, need the help of JVM language code, that may be supplied by tools, likely with restricted API support. Some programming languages and tools allow cross-platform app support (i.e. for both Android and iOS). Third party tools, development environments, and language support have also continued to evolve and expand since the initial SDK was released in 2008. The official Android app distribution mechanism to end users is Google Play; it also allows staged gradual app release, as well as distribution of pre-release app versions to testers.

The Android software development kit (SDK) includes a comprehensive set of development tools. The Android SDK Platform Tools are a separately downloadable subset of the full SDK, consisting of command-line tools such as adb and fast boot. The Android Debug Bridge (ADB) is a tool to run commands on a connected Android device. Fast boot is a protocol used for flashing file systems. Code written in C/C++ can be compiled to ARM, or x86 native code (or their 64-bit variants) using the Android Native Development Kit (NDK).

*Android Open Accessory Development Kit*

The Android 3.1 platform (also back ported to Android 2.3.4) introduces Android Open Accessory support, which allows external USB hardware (an Android USB accessory) to interact with an Android-powered device in a special "accessory" mode. When an Android-powered device is in accessory mode, the connected accessory acts as the USB host (powers the bus and enumerates devices) and the Android-powered device acts as the USB device. Android USB accessories are specifically designed to attach to Android-powered devices and adhere to a simple protocol (Android accessory protocol) that allows them to detect Android-powered devices that support accessory mode.

* **Java Programming Language:**

Java is a high-level, class-based, object-oriented programming language that is designed to have as few implementation dependencies as possible. It is a general-purpose programming language intended to let programmers write once, run anywhere (WORA),[17] meaning that compiled Java code can run on all platforms that support Java without the need to recompile.[18] Java applications are typically compiled to byte code that can run on any Java virtual machine (JVM) regardless of the underlying computer architecture. The syntax of Java is similar to C and C++, but has fewer low-level facilities than either of them. The Java runtime provides dynamic capabilities (such as reflection and runtime code modification) that are typically not available in traditional compiled languages. As of 2019, Java was one of the most popular programming languages in use according to GitHub, particularly for client–server web applications, with a reported 9 million developers.[21]

Java was originally developed by James Gosling at Sun Microsystems and released in May 1995 as a core component of Sun Microsystems' Java platform. The original and reference implementation Java compilers, virtual machines, and class libraries were originally released by Sun under proprietary licenses. As of May 2007, in compliance with the specifications of the Java Community Process, Sun had relicensed most of its Java technologies under the GPL-2.0-only license. Oracle offers its own HotSpot Java Virtual Machine, however the official reference implementation is the OpenJDK JVM which is free open-source software and used by most developers and is the default JVM for almost all Linux distributions.

As of March 2022, Java 18 is the latest version, while Java 17, 11 and 8 are the current long-term support (LTS) versions. Oracle released the last zero-cost public update for the legacy version Java 8 LTS in January 2019 for commercial use, although it will otherwise still support Java 8 with public updates for personal use indefinitely. Other vendors have begun to offer zero-cost builds of OpenJDK 8 and 11 that are still receiving security and other upgrades.

* **XML:**

Extensible Markup Language (XML) 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. The World Wide Web Consortium's XML 1.0 Specification of 1998 and several other related specifications—all of them free open standards—define XML.

The design goals of XML emphasize simplicity, generality, and usability across the Internet. It is a textual data format with strong support via Unicode for different human languages. Although the design of XML focuses on documents, the language is widely used for the representation of arbitrary data structures, such as those used in web services.

Several schema systems exist to aid in the definition of XML-based languages, while programmers have developed many application programming interfaces (APIs) to aid the processing of XML data.

* **Firebase:**

****Firebase is a platform developed by Google for creating mobile and web applications. It was originally an independent company founded in 2011. In 2014, Google acquired the platform and it is now their flagship offering for app development.

Firebase evolved from Evolve, a prior startup founded by James Tamplin and Andrew Lee in 2011. Envolve provided developers an API that enables the integration of online chat functionality into their websites. After releasing the chat service, Tamplin and Lee found that it was being used to pass application data that were not chat messages. Developers were using Envolve to sync application data such as game state in real time across their users. Tamplin and Lee decided to separate the chat system and the real-time architecture that powered it. They founded Firebase as a separate company in 2011 and it launched to the public in April 2012.

Firebase's first product was the Firebase Realtime Database, an API that synchronizes application data across iOS, Android, and Web devices, and stores it on Firebase's cloud. The product assists software developers in building real-time, collaborative applications.

**4.3 Tools Used:**

**Android Studio:**

****Android Studio is the official integrated development environment (IDE) for Google's Android operating system, built on JetBrains' IntelliJ IDEA software and designed specifically for Android development. It is available for download on Windows, macOS and Linux based operating systems or as a subscription-based service in 2020.It is a replacement for the Eclipse Android Development Tools (E-ADT) as the primary IDE for native Android application development.

Android Studio was announced on May 16, 2013, at the Google I/O conference. It was in early access preview stage starting from version 0.1 in May 2013, then entered beta stage starting from version 0.8 which was released in June 2014.The first stable build was released in December 2014, starting from version 1.0.On May 7, 2019, Kotlin replaced Java as Google's preferred language for Android app development. Java is still supported, as is C++.

**Features**

The following features are provided in the current stable version:

* Gradle-based build support
* Android-specific refactoring and quick fixes
* Lint tools to catch performance, usability, version compatibility and other problems
* ProGuard integration and app-signing capabilities
* Template-based wizards to create common Android designs and components
* A rich layout editor that allows users to drag-and-drop UI components, option to preview layouts on multiple screen configurations
* Support for building Android Wear apps
* Built-in support for Google Cloud Platform, enabling integration with Firebase Cloud Messaging (Earlier 'Google Cloud Messaging') and Google App Engine.Android Virtual Device (Emulator) to run and debug apps in the Android studio.
* Android Studio supports all the same programming languages of IntelliJ (and CLion) e.g. Java, C++, and more with extensions, such as Go; and Android Studio 3.0 or later supports Kotlin and "all Java 7 language features and a subset of Java 8 language features that vary by platform version."

External projects backport some Java 9 features. While IntelliJ states that Android Studio supports all released Java versions, and Java 12, it's not clear to what level Android Studio supports Java versions up to Java 12 (the documentation mentions partial Java 8 support). At least some new language features up to Java 12 are usable in Android.

**4.4 Diagrams**

**UNIFIED MODELING DIAGRAM**

**Data Flow Diagram( 0 level)**

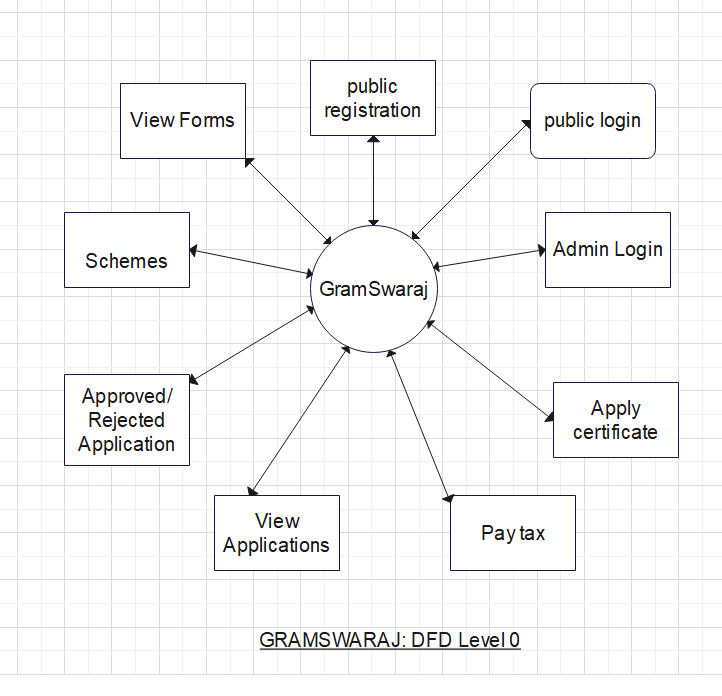


Figure1:Data flow diagram (0level)

**Data Flow Diagram(1 level)**

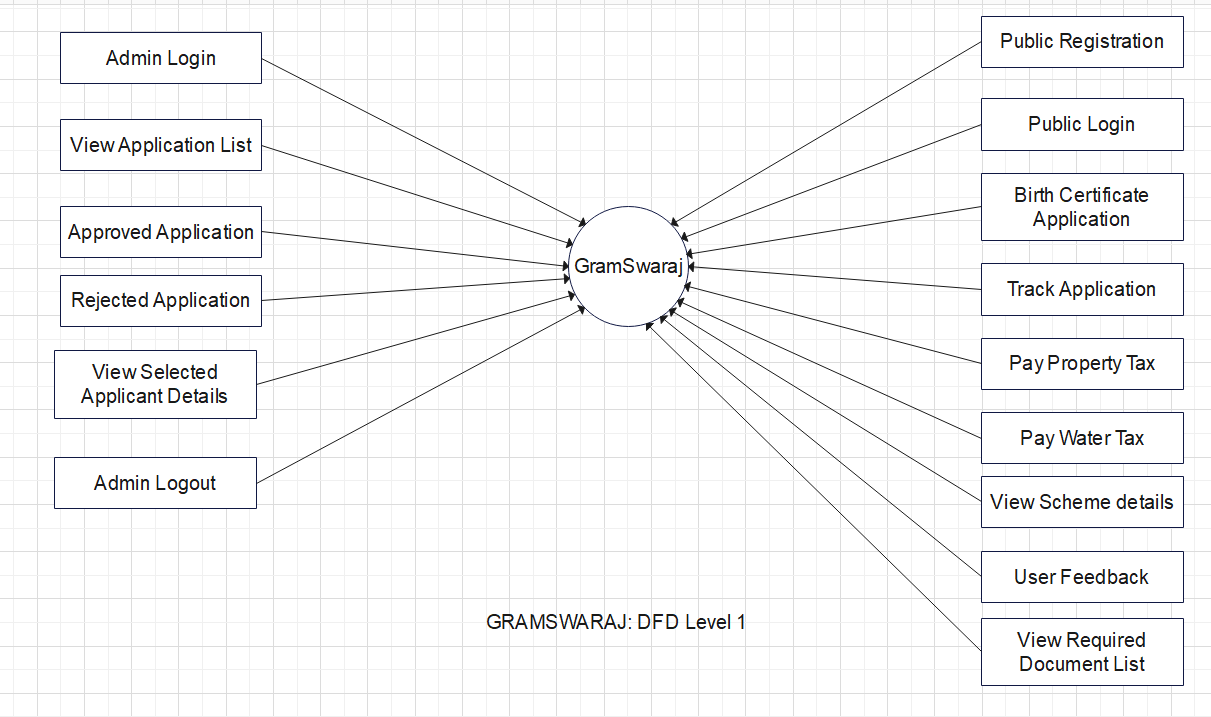
****

Figure 2: Data flow diagram (1level)

**Data Flow Diagram(2 level)**

**Admin:**

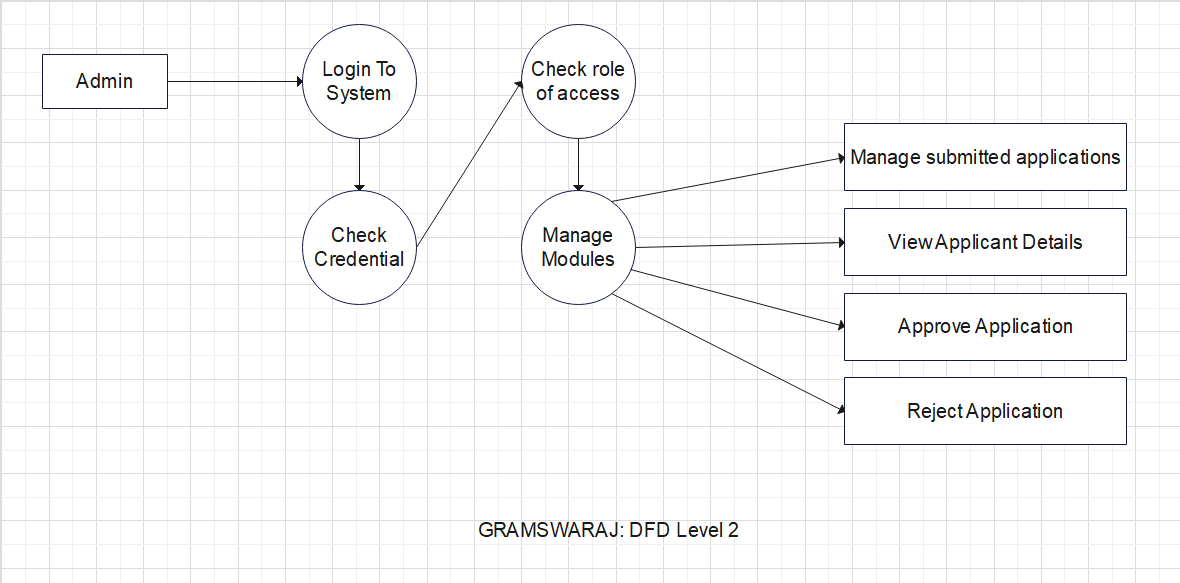
****

Figure 3: Data flow diagram (2 level)

**Public:**

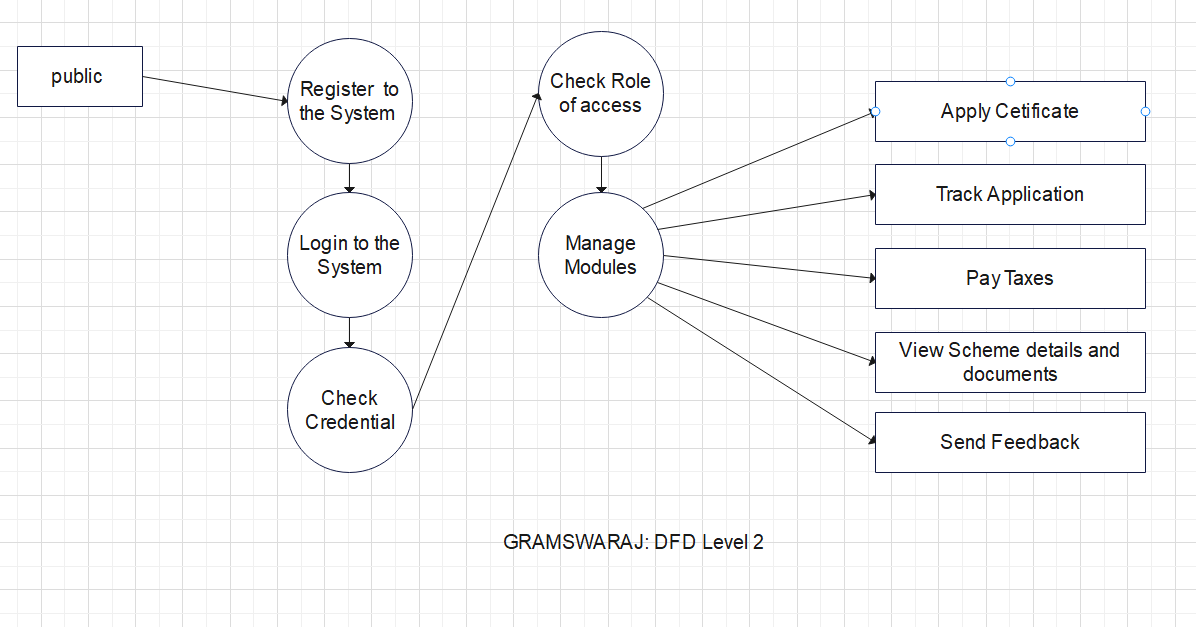
****

Figure 4: Data flow diagram (2 level)

**ER Diagram**

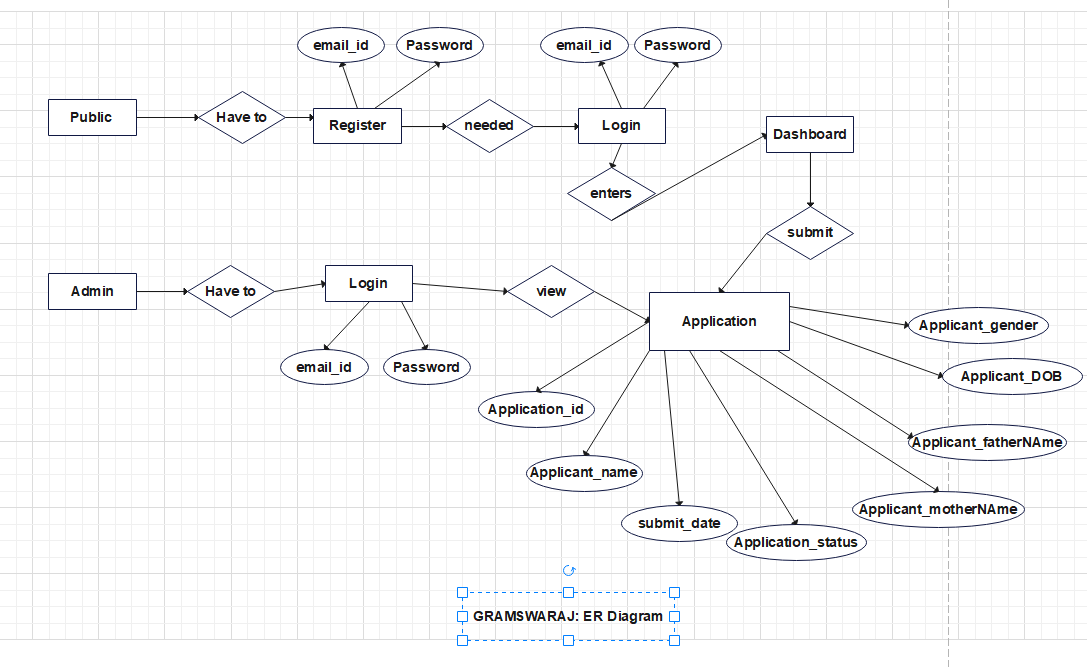
****

Figure 5: ER Diagram

**Class Diagram**

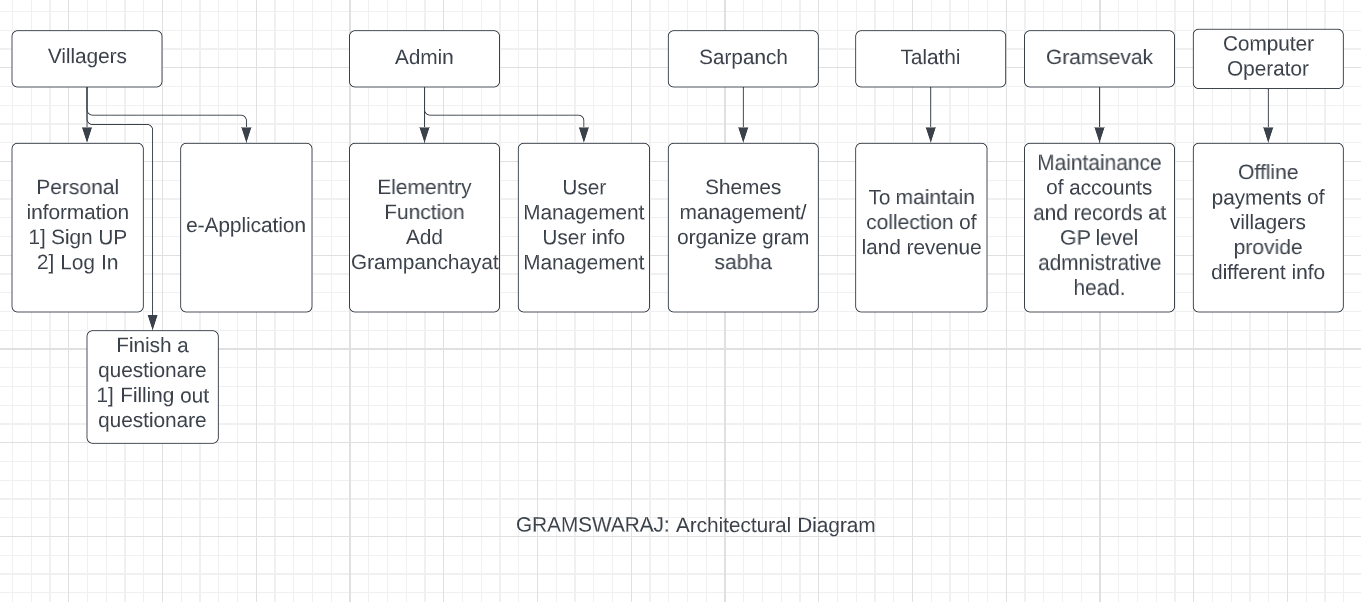
****

Figure 6: Class diagram

**Sequence Diagram**

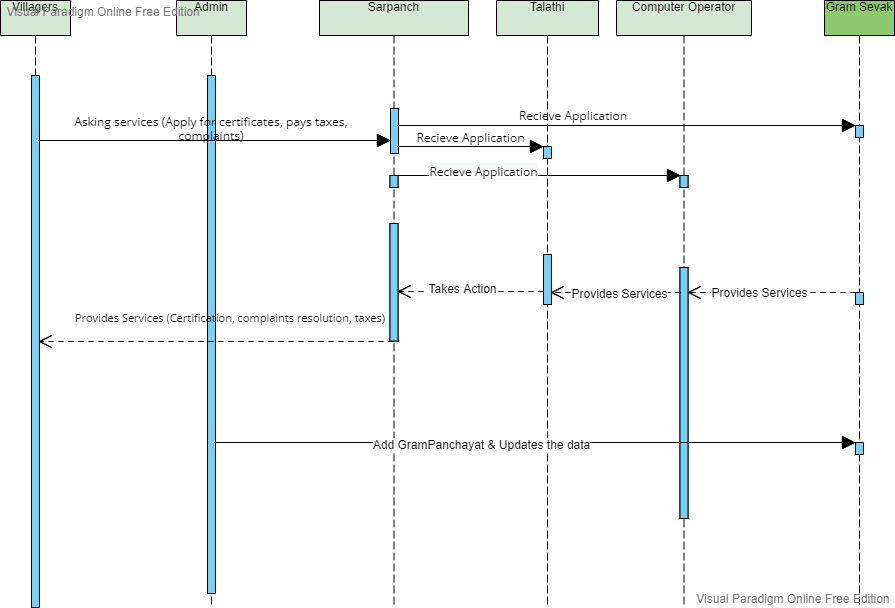
****

Figure 7: Sequence Diagram

**Use Case Diagram**

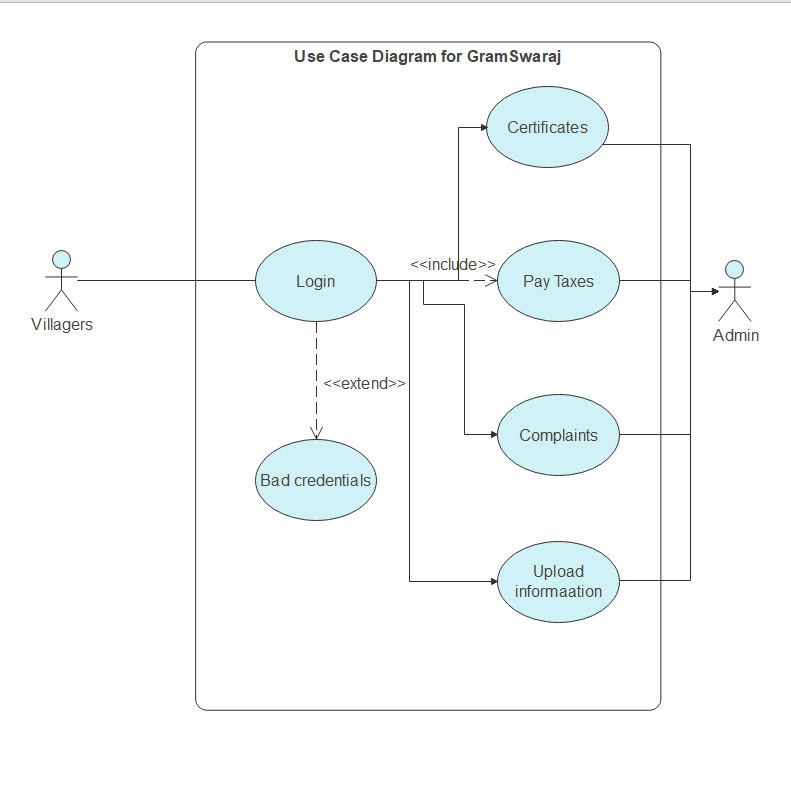
****

Figure 8: Use case diagram

**Actitvity Diagram**

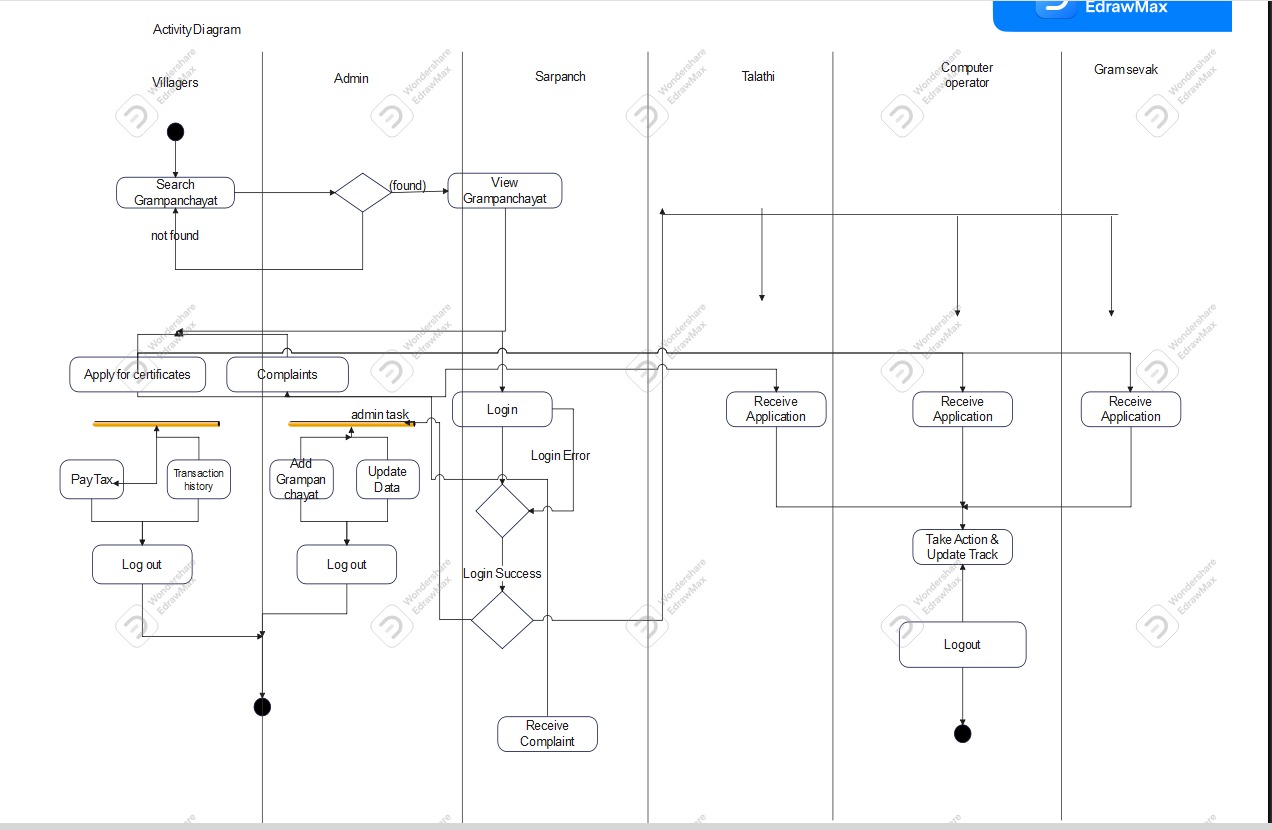
****

Figure 9: Activity diagram

**Component Diagram**

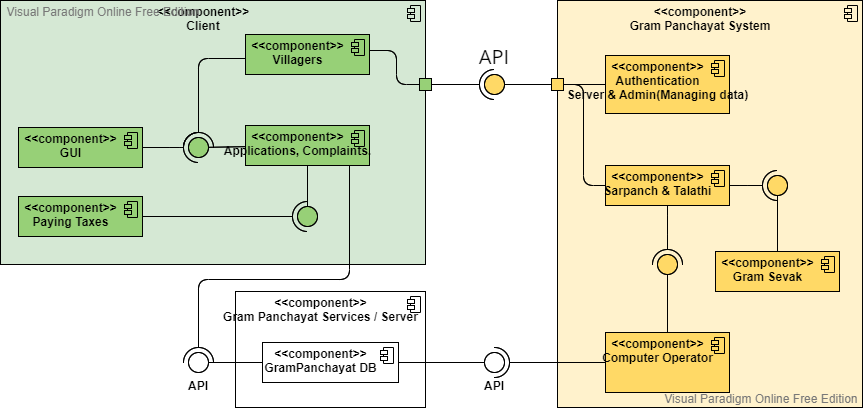
****

Figure 10: Component diagram

**Deployment Diagram**

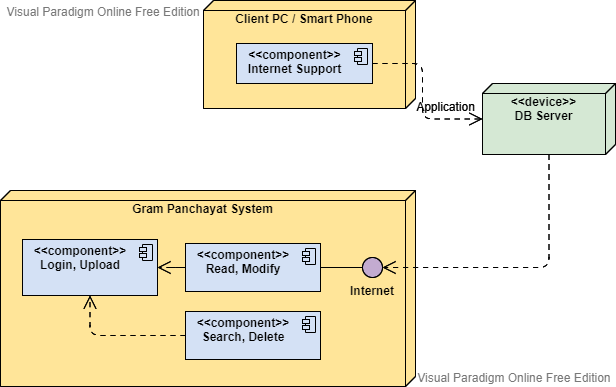
****

Figure 11: Deployment diagram

**Object Diagram**

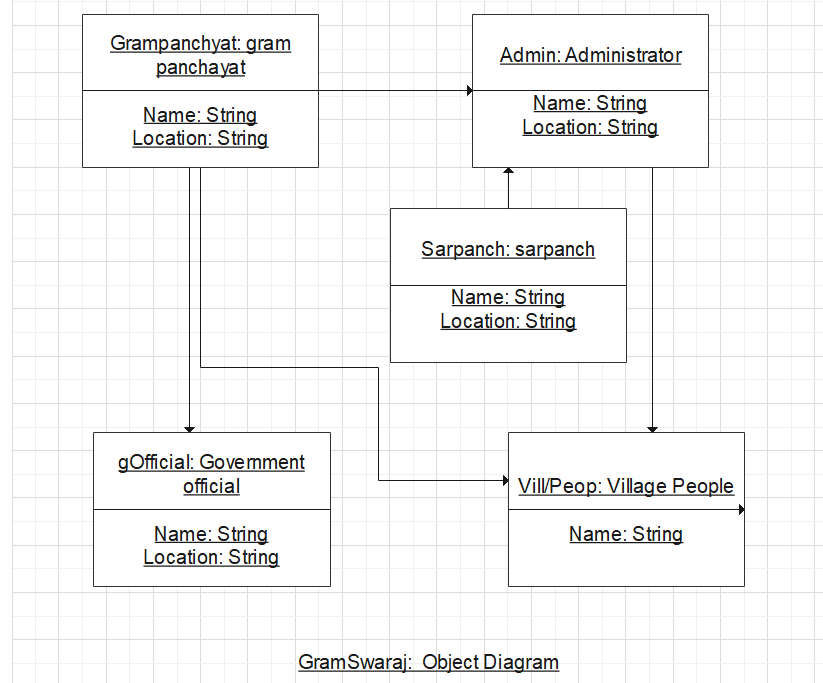
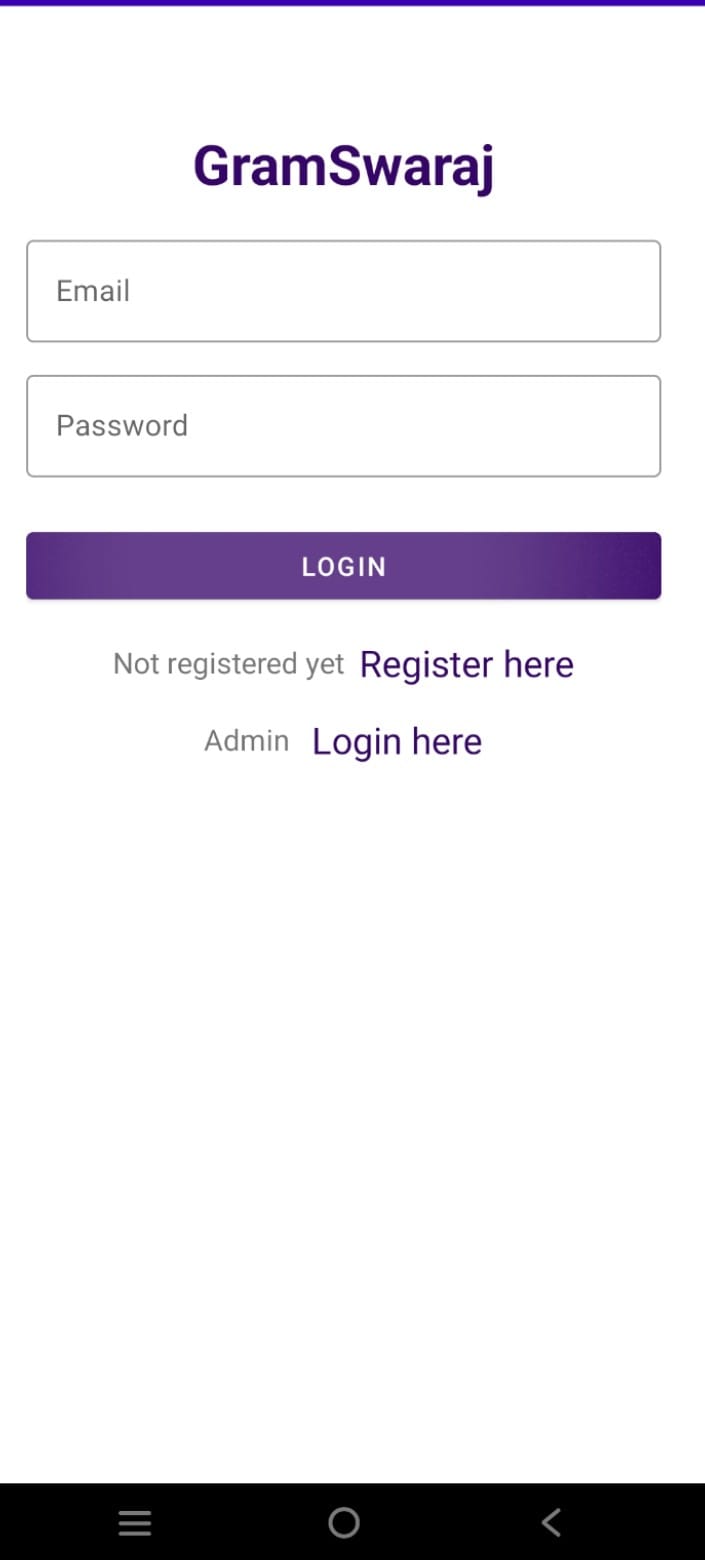
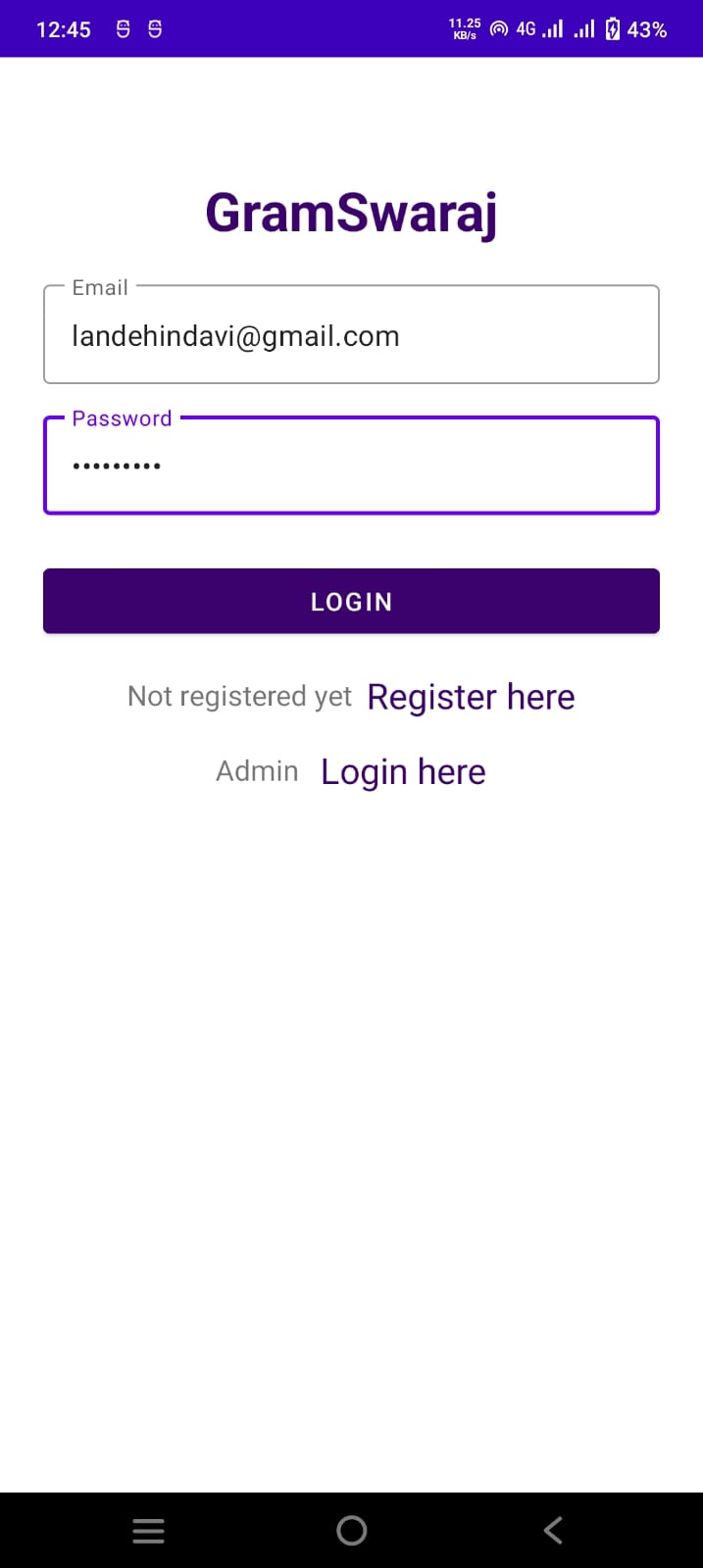
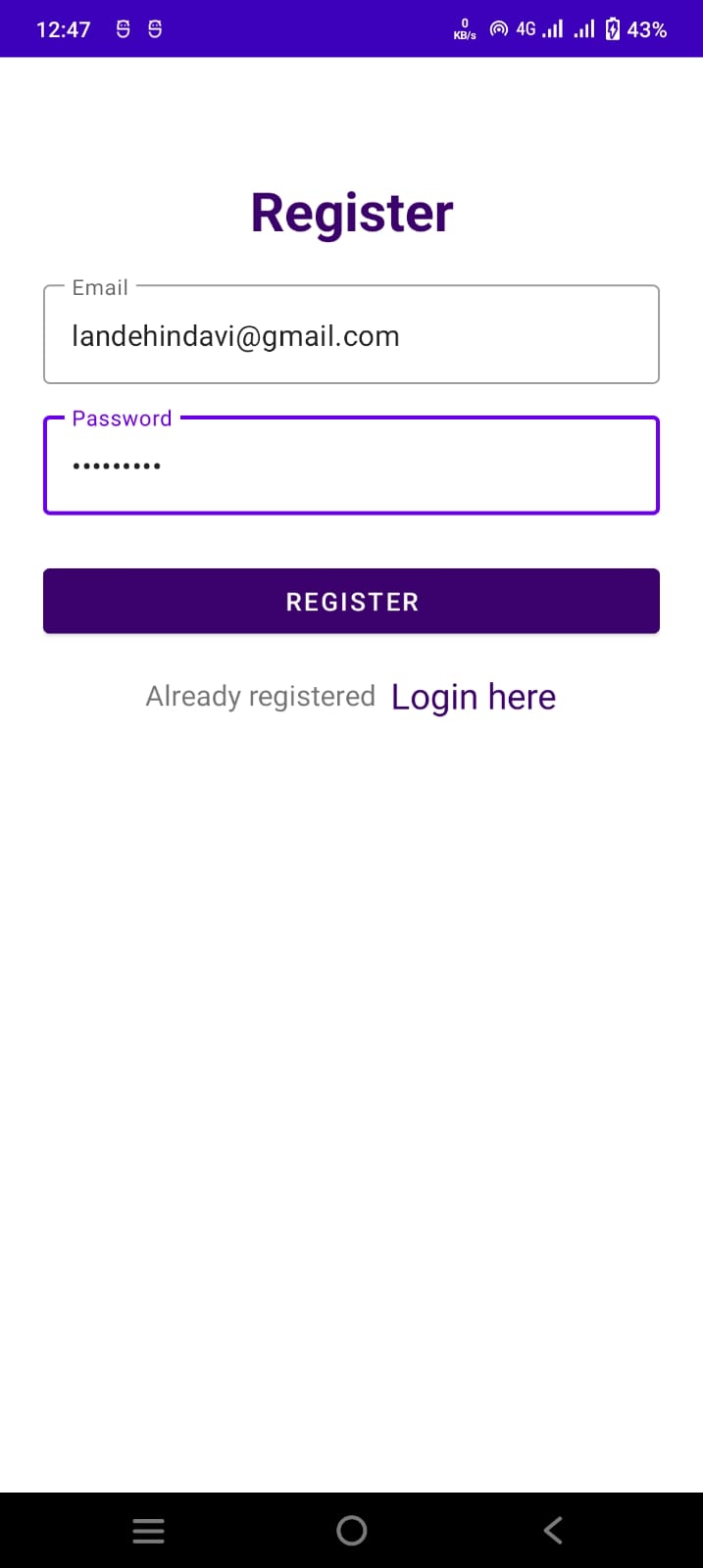
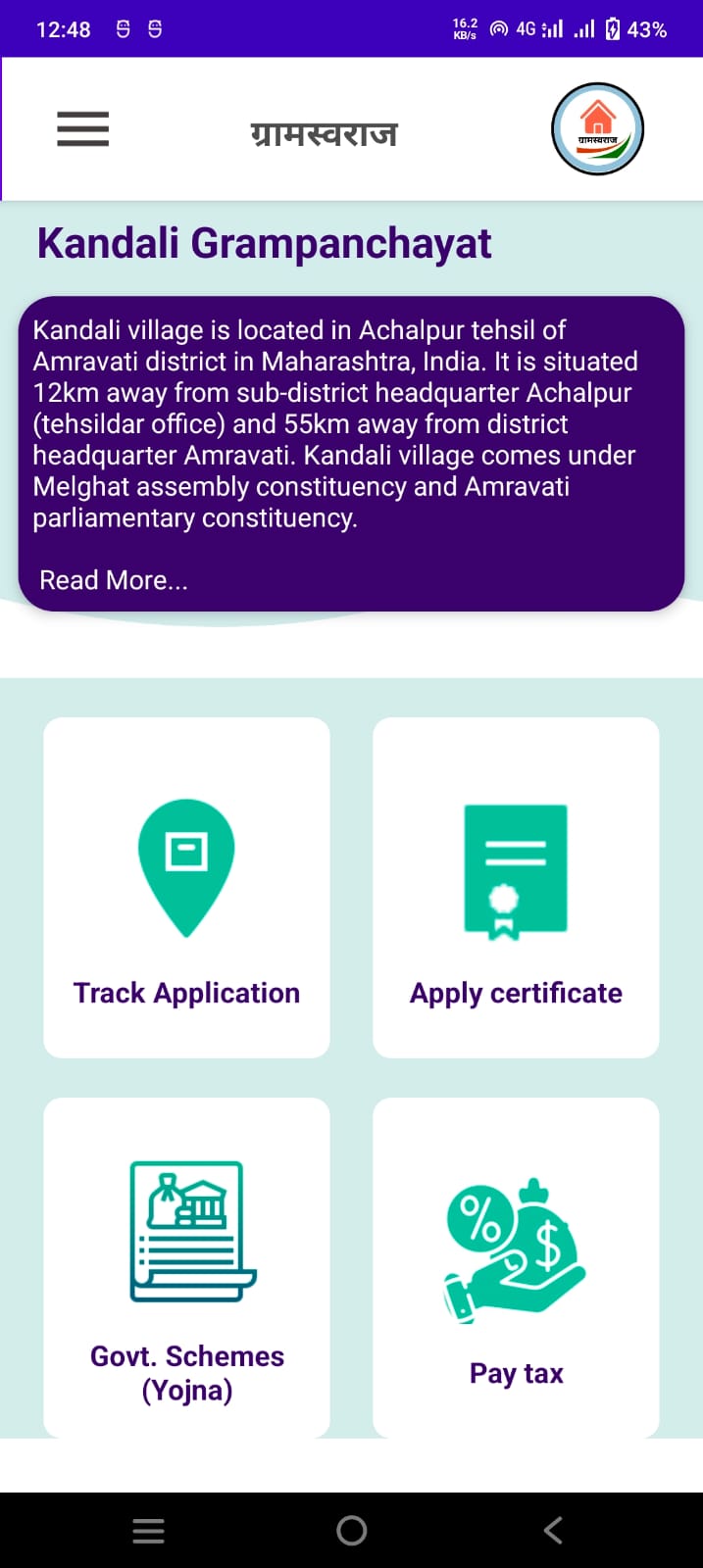
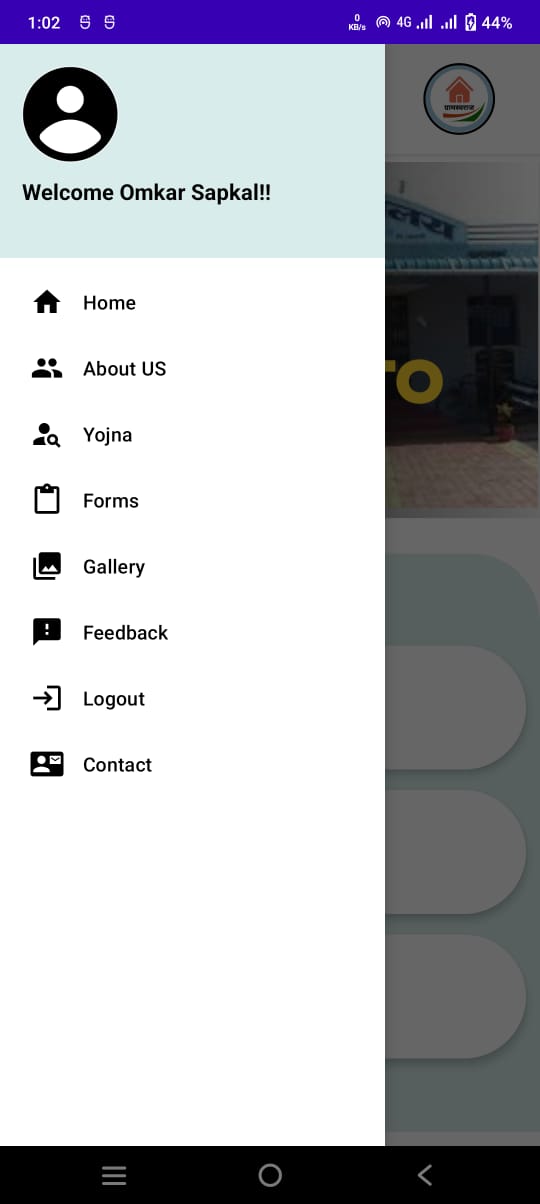
****

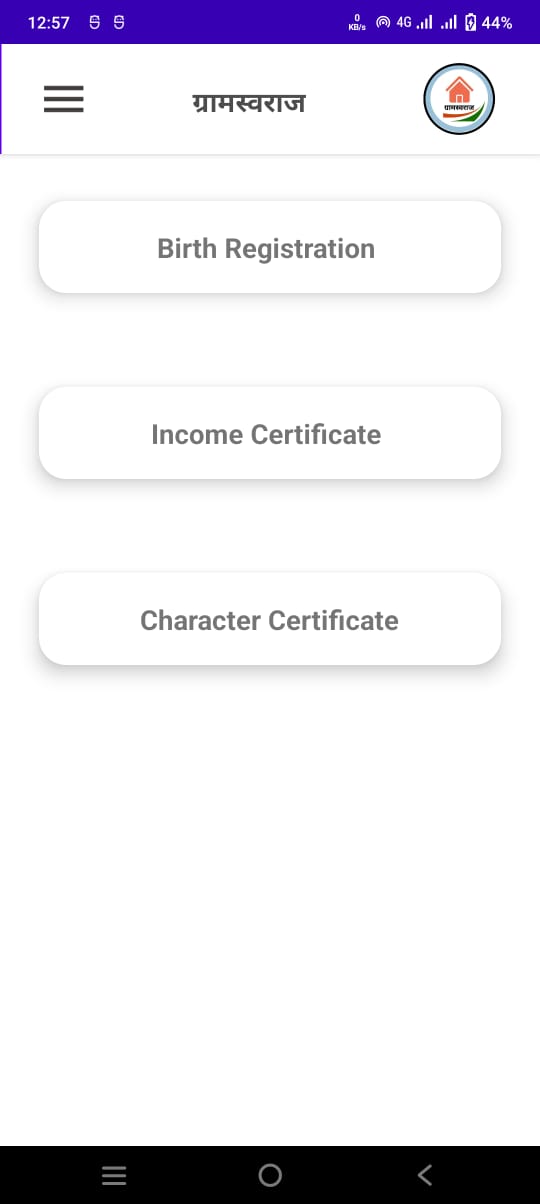
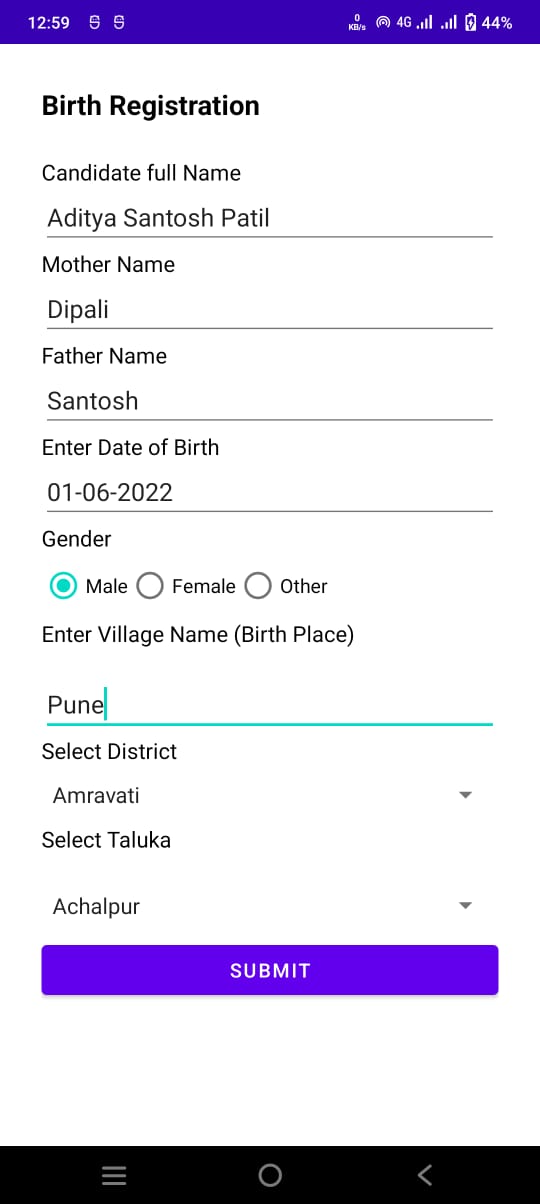
Figure 12: Object Diagram

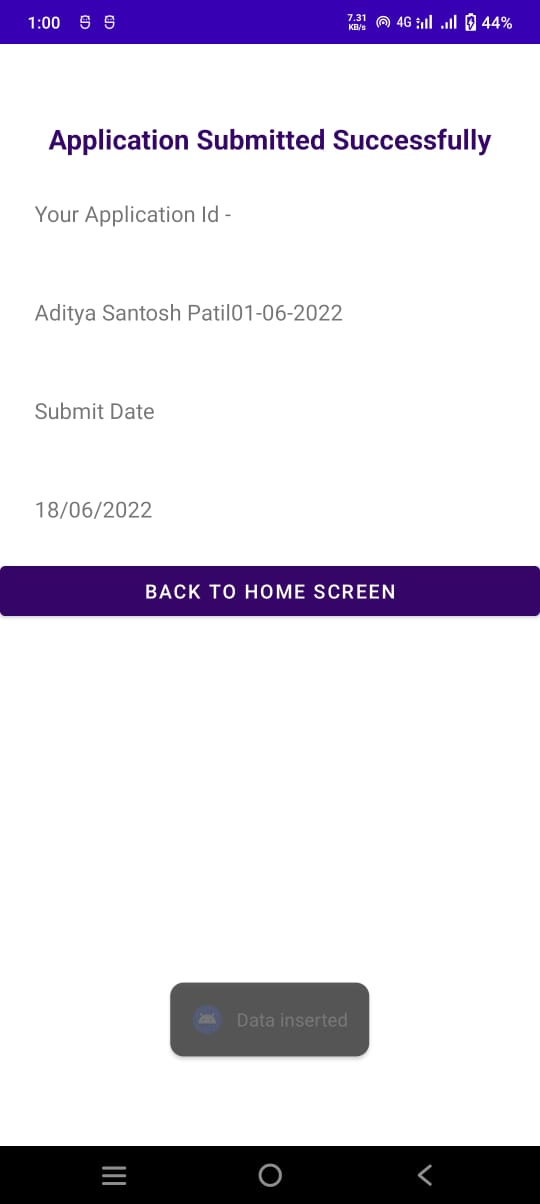
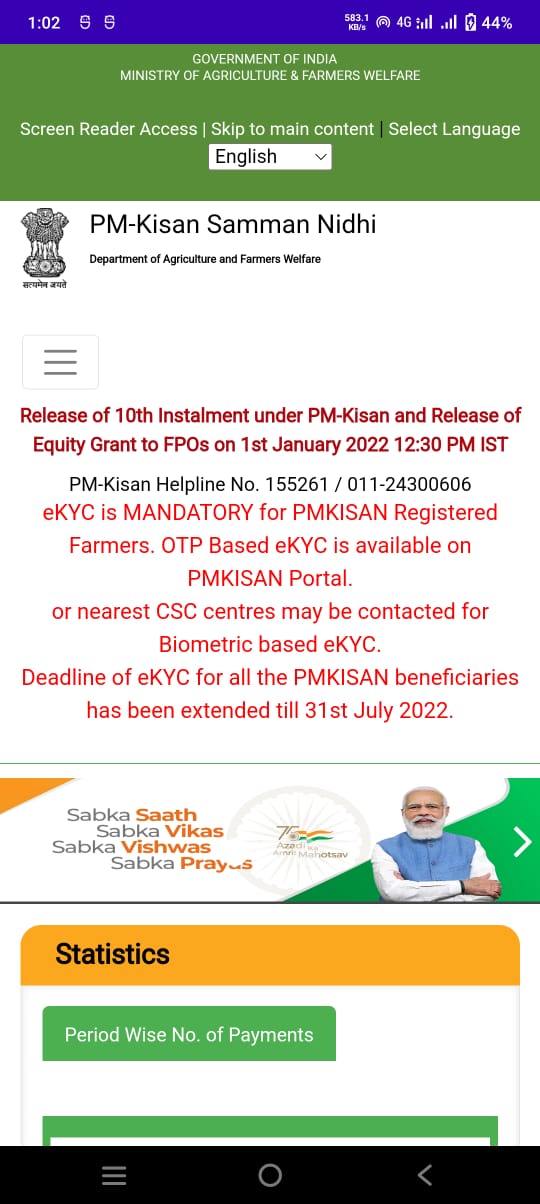
**4.5 Project Output**

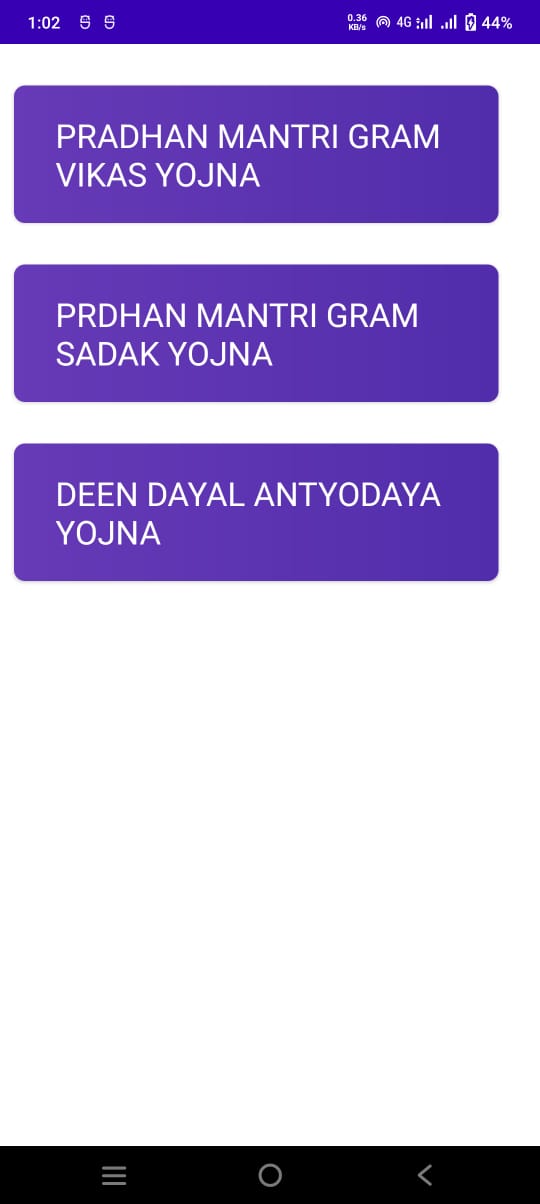
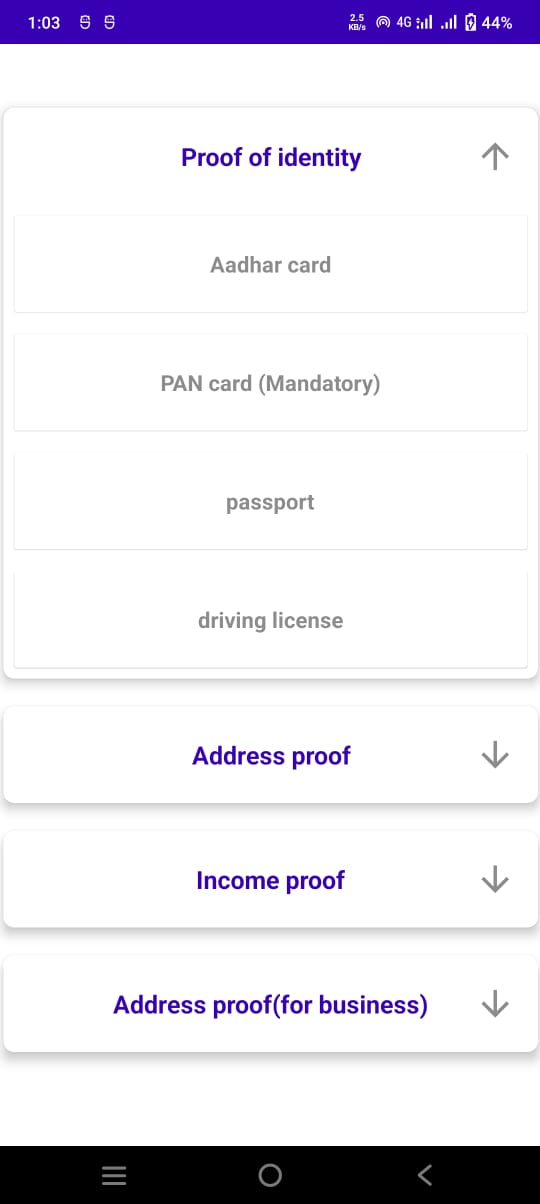
 

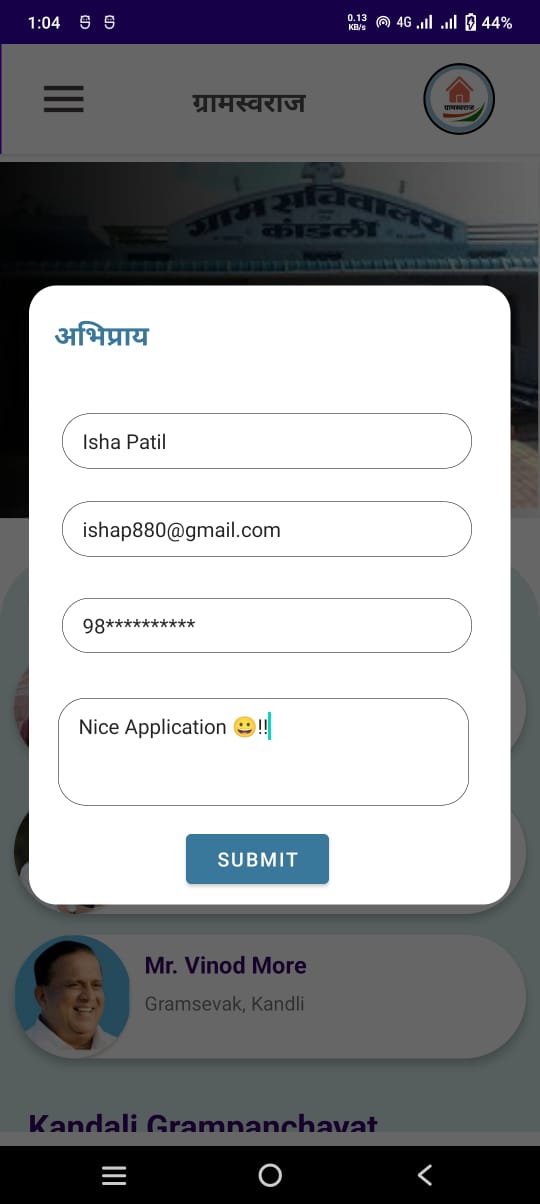
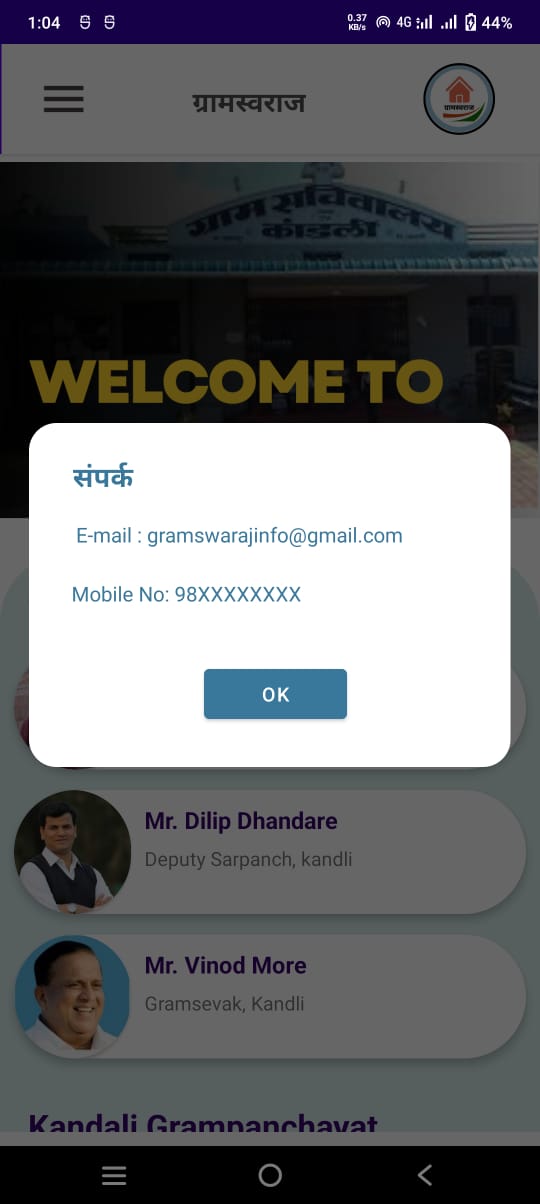
 

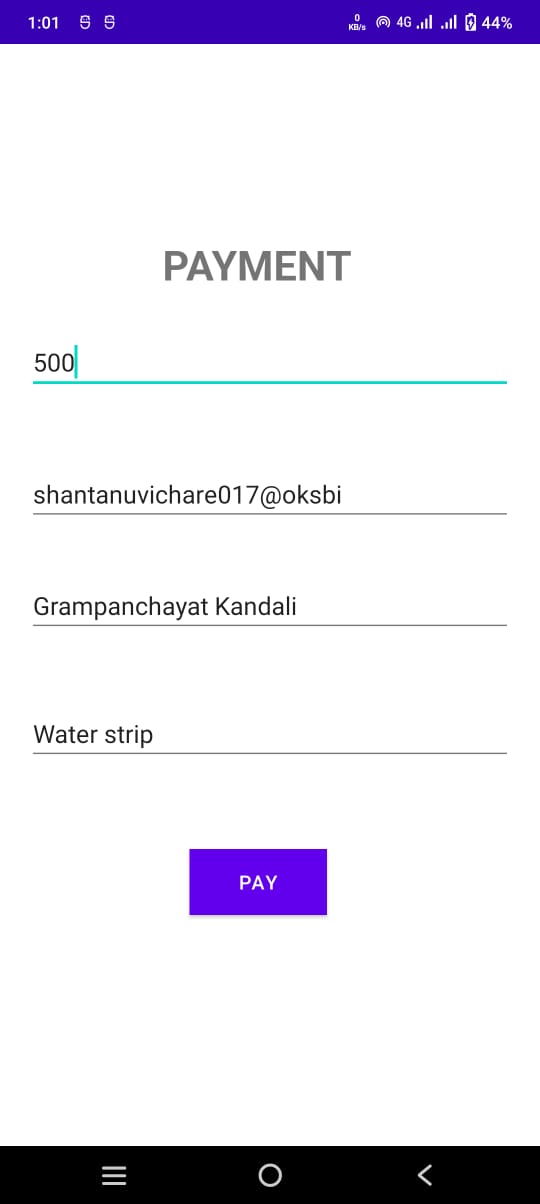
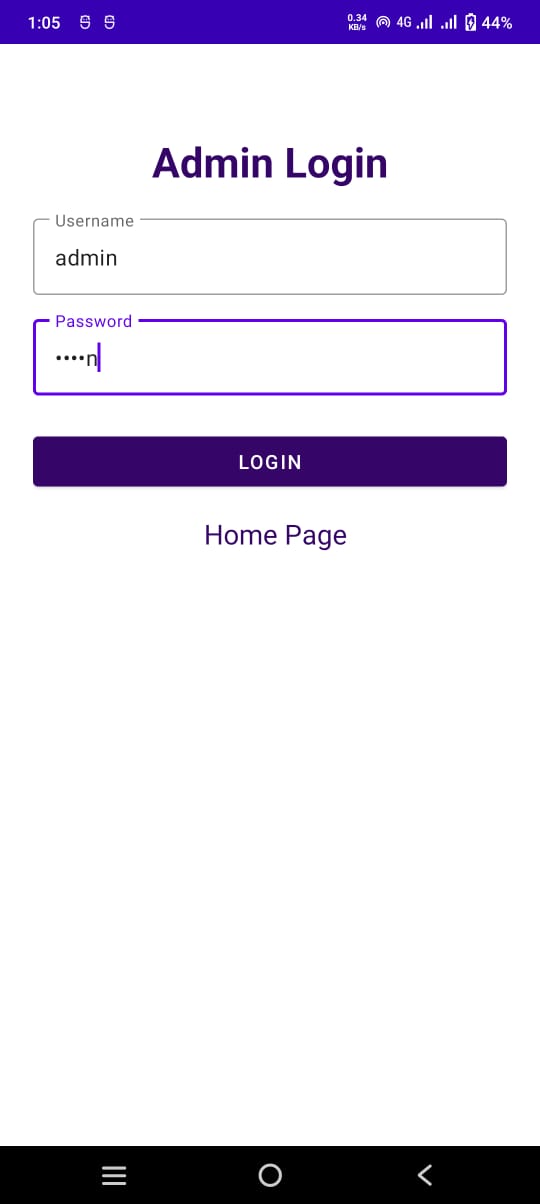
 

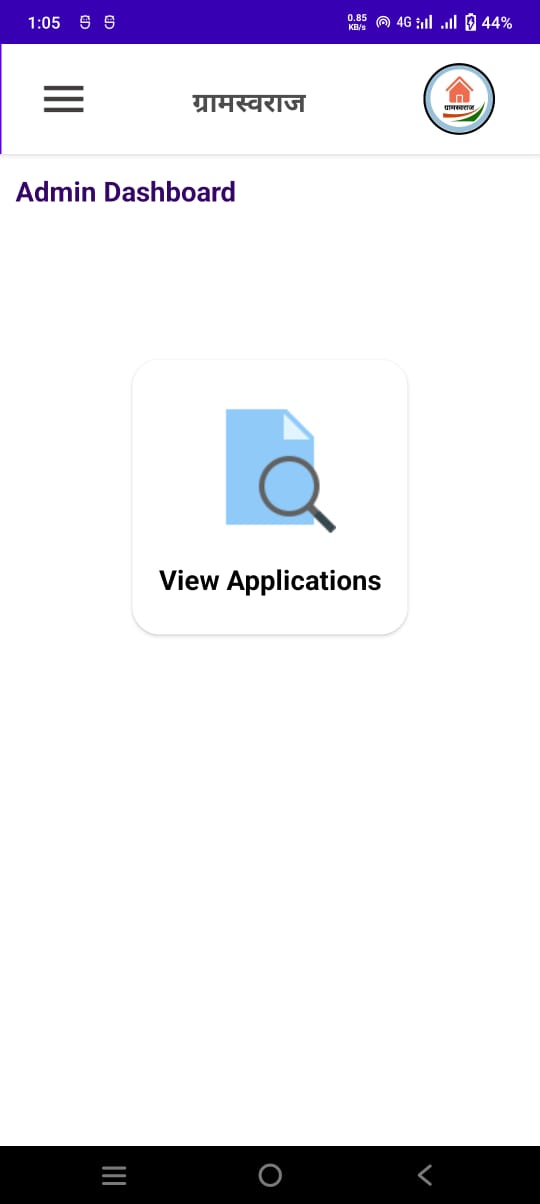
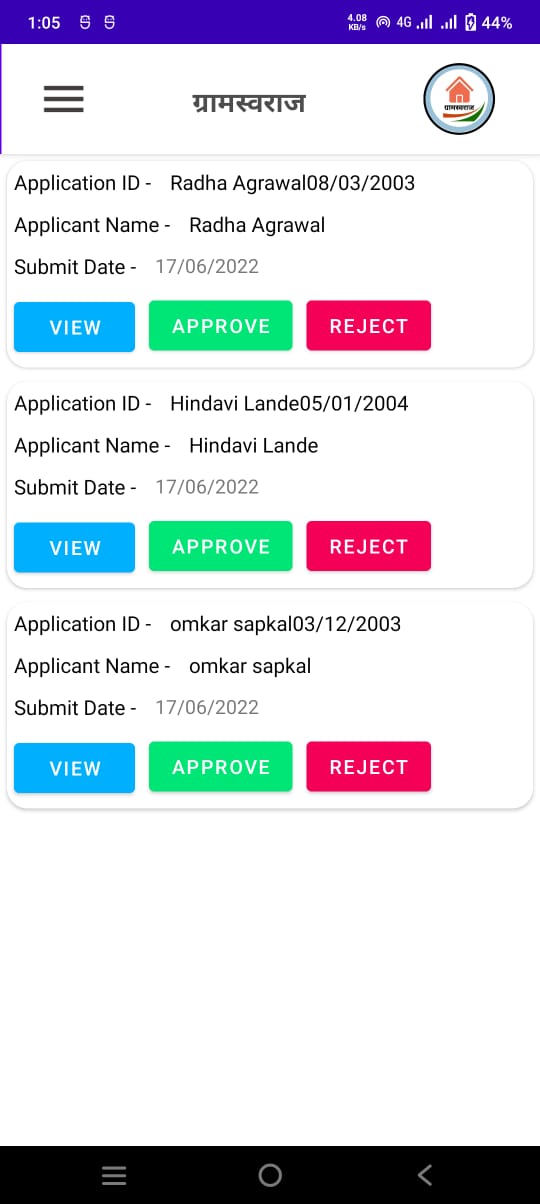
 

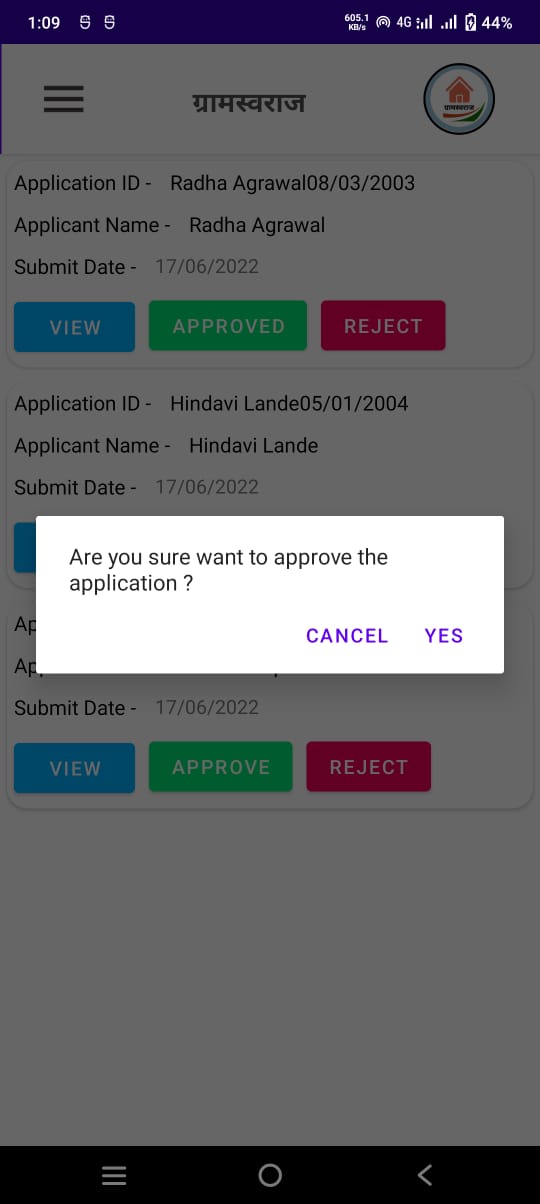
  s 

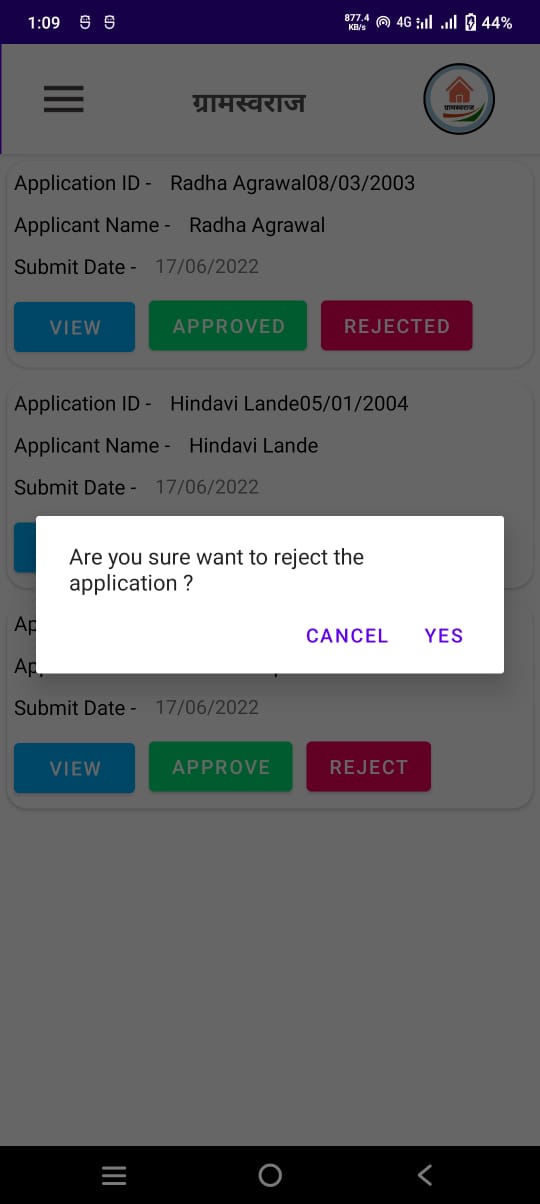
 



**CHAPTER 5 : SYSTEM TESTING**

**5.1 Types of Testing**

|  |  |  |  |
| --- | --- | --- | --- |
| **Sr No.** | **Type of Test** | **Description** | **Software Component** |
| 1 | Requirement Testing | This testing is required because we need to verify whether our requirements are able to solve the current problem or not | Complete Software including GUI |
| 2 | Unit Testing | This testing allows us to test individual modules before integrating hem together to form a single software | Data Pre-processing |
| 3 | Integration | This test is important to check whether the modules are giving the same results after integration as before | All the software components individually |
| 4 | Performance | This test is important to calculate the efficiency of the software also helps us to find any performance issue related to the system | All the software components individually |
| 5 | Security | We have performed this test to check whether privacy is maintained | Uploaded assignments |

Table No. 3 :Types of Testing

**5.2 Test Cases:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test Step** | **Test Case** | **Test Data** | **Expected Result** | **Actual Result** | **Status** |
| 01 | Verify if a user will be able to login with a valid username and valid password. | Email: landehindavi@email.com Password: Pass@1234 | Login Successful | Login successful | Pass |
| 02 | Verify if a user cannot login with a valid username and an invalid password. | Email: landehindavi@email.com Password: Pass@1234 | Login successful | Login failed | Fail |
| 03 | Verify the login page for both, when the field is blank and Submit button is clicked. | Email : none  Password: none | Email cannot be empty. | Email cannot be empty | Pass |
| 04 | Verify can you navigate between various modules as expected | Click on the module | Redirect to that module | Redirect to that module | Pass |
| 05 | Verify the gram panchayat name typed in searchbar shows that gp. Option. | Search : Kandali | Redirected to kandali grampanchyat page. | Redirected to kandali grampanchayat page. | Pass |
| 06 | Submit the birth registration form with valid inputs. | Name : Ram  Mother Name : Sangita  Father Name : Vikas  Dob : 01/05/2003  Gender : male  District : Amravati  Talaka : Achalpur  Village (Birth Place) : Shreerampur | Submited successfully | Submitted successfully | Pass |

Table No 4: Test cases

**CHPATER 6 : FUTURE SCOPE**

This system has been designed keeping in mind the requirements of gram panchayat staff and enables the admin and staff of panchayat to make entries in the database about villagers, personal details, and their related services. This system also provides him the authority to manipulate his account. we can add much more feature in the system i.e., alert system, receive notification to user and gram panchayat staff about some action. By using this system data collected from different gram panchayats will be helpful for implementing different schemes and will help in natural calamities and data can be useful in other fields. Paying various taxes which comes under local governance authority such as property and water tax can paid using this app.

**CHAPTER 7: CONCLUSION**

This mobile application will be helpful to the villagers of that village. it will bring transparency, accountability, and efficiency in administration. Document and their related record will be available on this application. It helps to make administration more accountable as well as more transparent. The above survey and proposed system will help the Gram panchayat system to work efficiently. This system provides ID and password for the villager. Account history provides information about the services that are previously submitted by the villagers. Both analysis and evaluation results confirm that the proposed web base system can provide an effective solution for submitting the services/schemes that are useful for the villagers in online mode. This will help to minimize corruption in the system, and also save the effort and time of common man and government officers.

**CHAPTER 8: REFERENCES**

[**https://www.android.com/intl/en\_in/**](https://www.android.com/intl/en_in/)

[**https://www.tutlane.com/tutorial/android**](https://www.tutlane.com/tutorial/android)

[**https://firebase.google.com**](https://firebase.google.com)

[**https://www.javatpoint.com/android-tutorial**](https://www.javatpoint.com/android-tutorial)