# Content

# 1. ABOUT THE ORGANIZATION

## 1.1 INTRODUCTION

National Informatics Centre (NIC) was established in 1976, and has since emerged as a "prime builder" of e-Government / e-Governance applications up to the grassroots level as well as a promoter of digital opportunities for sustainable development. NIC, through its ICT Network, "NICNET", has institutional linkages with all the Ministries /Departments of the Central Government, 35 State Governments/ Union Territories, and about 625 District administrations of India. NIC has been instrumental in steering e-Government/e-Governance applications in government ministries/departments at the Centre, States, Districts and Blocks, facilitating improvement in government services, wider transparency, promoting decentralized planning and management, resulting in better efficiency and accountability to the people of India.

"Informatics-led-development" programmed of the government has been spearheaded by NIC to derive competitive advantage by implementing ICT applications in social & public administration. The following major activities are being undertaken:

- Setting up of  ICT Infrastructure
- Implementation of National and State Level e-Governance Projects
- Products and Services
- Consultancy to the government departments
- Research and Development
- Capacity Building

In executing all these activities, NIC has been given recognition in terms of awards and accolades in International as well as National levels, which are listed in the Awards Section.

Thus, NIC, a small program started by the external stimulus of an UNDP project, in the early 1970s, became fully functional in 1977 and since then has grown with tremendous momentum to become one of India's major S&T; organizations promoting informatics led development. This has helped to usher in the required transformation in government to ably meet the challenges of the new millennium.

**NIC Headquarters** is based in **New Delhi**. At NIC Headquarters, a large number of Application Divisions exist which provide total Informatics Support to the Ministries and Departments of the Central Government. NIC computer cells are located in almost all the Ministry Bhawans of the Central Government and Apex Offices including the Prime Minister's Office, the Rashtrapati Bhawan and the Parliament House. Apart from this, NIC has various Resource Divisions at the Headquarters which specialize into different areas of IT and facilitate the Application Divisions as well as other NIC Centre's in providing state-of-the-art services to the Govt.

At the **State level**, NICs State/UTs Units provide informatics support to their respective State Government and at the District level laid the NIC District Informatics Offices. **Assam** is a northeastern state of India with its capital at Dispur, in the eastern part of Guwahati city.

## 1.2 NIC GUWAHATI

**Assam State Centre (ASSC)** of the National Informatics Centre (NIC) was set up in April 1986. Various district units were set up from 1990 onwards, and now there is a NIC District Unit in each of 32 districts of the State.

The SIO is supported at the State level by a team of technical officers and some other support staff. The NIC Assam State has a Training Division equipped with the latest training infrastructure; there is also a National Informatics Centre Services Incorporated (NICSI) office, NIC also has dedicated cells in the Guwahati High Court and Regional Passport Office, Guwahati.

## 1.3 RESOURCES

➢ **Network:** The Network Operation Centre in NIC Assam commissioned in 2004 A Block of the Secretariat is the network gateway for the entire NE States. At present, there are two 100 Mbps links from BSNL and PGCIL. The Guwahati Delhi link is a primary link from PGCIL and the Guwahati-Hyderabad link is a failover link by BSNL. All the NE State capitals are connected to this centre either by BSNL or by PGCIL. The NOC is manned in the 24/7 mode by the hired manpower. There are various applications installed locally which are for effective monitoring of the entire network including the other NE states namely What's up, Nagios, MRTG and Open NMS. There are two 63 inch plasma TV screens which are being used to monitor the status and health of the various nodes. All the 8 Blocks of the Secretariat are connected to the NOC with OFC backbone, each block having nearly 120 LAN nodes. All major and important locations inside the campus have secured WEP Wi-Fi access.23 districts of Assam are connected with dedicated 2 Mbps MLLN from the NOC in addition to DVB VSATs for data and DAMA VSATs for VC. Four new districts presently having VSATs only are being provided leased line connectivity shortly. Presently under the Citizen-centric Services Project, a state of the art Wi-Max network (5.876 GHz) is being set up in each district connecting 10 government organizations in each district headquarter in the State. These

ten organizations are being connected to the NIC District Centre network HUB.As regards connectivity to other offices, district head post offices are being connected under the Department of Post (DoP) project. The Regional Passport Office (RPO), Census and Immigration Terminal at the air-port are connected to NOC with MLLN line from BSNL. The Guwahati Metropolitan Area Network (GMAN) was commissioned in 2005-2006 when nearly 42 Directorate offices were connected with RF links. The four sites with RF hubs are connected by 2 Mbps leased line from BSNL and these lines acts as a backbone for the GMAN network. The residence of Governor of Assam and the Chief Minister are also connected with 2 Mbps leased line from BSNL and RF Link. This centre has also got a 3 TB SAN along with several mid-range servers for hosting several Intranet applications both under Windows and Linux platforms. Guwahati is also one of the HUBS for the National Knowledge Commission Network (NKN) in India. The BSNL and PGCIL have commissioned their high end network equipment's to support NKN at NOC Guwahati. BSNL has already installed a STM16 at NOC to give the high bandwidth network to various states in NER. IIT, Guwahati and AAU, Jorhat have been connected through NKN.

➢ **Video Conferencing**: With the first state-level VC Studio set up in 1998, VC facilities presently exist in almost all the districts and at the Chief Ministers Residence Office most of these are IP-based. VC facility is also available in the offices of the Chief Secretary, Minister of IT, Commissioner of IT. VC facility is regularly used by the State Government.

➢ **Training:** NIC, Assam State Unit, Guwahati has been imparting Training almost from its inception. Such activities were intensified since a separate Training Division was set up in 2000. Training Division conducts about 35-40 courses per year ranging from general awareness and Office Productivity courses for normal government users, Sectoral+ Course and Technology Update Courses for NIC personnel for the entire NE

region. Total number of participants trained per year is 600-900. Ten project trainees (from MCA and BE/B Tech courses) also are accepted per semester.

## 1.4 PROJECTS EXECUTED PRESENTLY

1.  Registration: Document Registration System, Assam (e-Panjeeyan).
2.  Land Records
3.  Vahan and Sarathi.
4.  Geographic Information System.
5.  App_Builder.
6.  N.R.C
7.  e-GRAS
8.  Websites of various Government Departments and Institutions.

## 1.5 CONTACT DETAILS

National Informatics Centre

Dispur, Rukmini Gaon

Guwahati-781006, Assam

Phone: +91-361-2237254

Website:dispur.nic.in

Email: sio-asm@nic.in

# 2. PROJECT OVERVIEW

## 2.1 PROJECT TITLE

The project is entitled as **e-PANJEEYAN FOR REVENUE & DISASTER MANAGEMENT, GOVT OF ASSAM**.

## 2.2 WHAT IS IT?

**e-PANJEEYAN FOR REVENUE & DISASTER MANAGEMENT, GOVT OF ASSAM** is an online application designed and implemented for government's registration system for deed and marriage registration. . It is a fully integrated mobile-based android application, which can facilitate the registration process right from Android phones.

## 2.3 ABOUT THE ORGANISATION

The Revenue & Disaster Management consist of two main departments which works for the well-being of the common people. The objectives of those departments are:

**Objective of Revenue Department –** The principal objectives of the Revenue Department are land and land revenue administration in all their facets:

- Formulation of Government land settlement policy.
- Administration of Land Reforms.
- Maintenance of Land Records.
- Land Requisition and Acquisition. (e) Administration of Land and Land Revenue and Local Rate.
- Administration of Assam Survey organization.
- Organization of Relief Operations on account of Natural Calamities.

**Objective of Disaster Department –** The aims & objectives of the Department as envisaged in the bye-laws of the agency are:

- To take up relief, restoration and reconstruction activities and measures for socio-economic revival for mitigating the damages caused or likely to be caused due to any disaster.

- To take up programmes and schemes that will prevent occurrence of any disaster or minimize the damaging effect due to any disaster.

- To undertake specific studies to identify programmes and schemes to be taken up to evaluate the various measures taken and suggest suitable action.

- To accept grants, donation, funds from the State Government, Government of India, bilateral or multi-lateral funding agencies, non-governmental organization (NGOs), semi-governmental organizations, State Government and Central Government undertakings, public or private companies, trusts, private individuals interested in participating in or supporting the disaster mitigation works.

- To raise loans if necessary.

- To utilize funds of the society either directly or through Government Departments, State or Central Government undertakings, semi-government Organizations(NGOs), or private philanthropic organizations for achieving its aims and objectives.

## 2.4 NEED FOR THE SYSTEM

Currently the interaction with e-Panjeeyan has been carried out through a web portal/application. As the usage of mobile applications has been growing tremendously, there is a need for a mobile application which can communicate with e-Panjeeyan and facilitate the registration process right from Android phones. As there have been two different ways of

interaction with e-Panjeeyan – **traditional web portal and modern or the non-traditional mobile app**.

Both methods have their own advantage and disadvantage and are used to make registration process easier.

## 2.4.1 ADVANTAGE of Traditional web portal:

Many organizations today have responsive websites in order to make their information, goods or services available on various platforms and devices. When people look for information, they don't go to the app stores at once. It is more likely that they will start with web browsers. And the results will probably consist mostly of web pages and just a few apps. And if one is surfing an app store itself…well, we can imagine how hard it to break through those thousands of apps.

## 2.4.2 DISADVANTAGE of Traditional web portal:

Mobile apps work differently than responsive websites. They don't run on all devices and need to be installed separately but what they give is unique user experience. While our website is aimed at attracting new or occasional clients our mobile app works with regular clients.

## e-Panjeeyan Mobile Application

A mobile app is a program accessed via the internet or stored locally. Mobile apps can use device's native features, which allows developers to create better, more intuitive and faster user experiences for customers. In e-Panjeeyan too, users can access some parts of the app without being connected to the internet, and additionally the filling of the forms for the registration process has become more user friendly.

## 2.4.3 ADVANTAGE OF e-Panjeeyan app:

For e-Panjeeyan users, the development of the mobile application have resulted the registration process more interactive and customized. A user only have to download and install e-Panjeeyan on their device, and generally, as it is a native app, it offer a faster and more responsive experience compared to the web portal.

## 2.5 PROPOSED SYSTEM

e-PANJEEYAN MOBILE APPLICATION allows user to take an appointment for deed and marriage registration and also to make payment of the associated amount through their android phones.

Some of the benefits and advantage of registering through the app are:

**1. Fee Assessment:** User can calculate the fee associated with deed and marriage registration prior making any registration.

**2. Confidentiality:** Whenever one register oneself with the app, all the personal details and information are kept confidential.

**3. Ease of access:** Once registered with the app, one will be able to check their appointment status through the app itself.

**4. Customization:** User can customize the app according to their desire, whether it be changing of the default language i.e., English to Hindi or changing of the default Day mode to Night mode.

**5. User friendly:** For the convenience of the users, there is a user manual available in the app which would guide the users throughout the whole time.

## 2.5.1 ANALYSIS OF THE EXISTING SYSTEM

Traditionally for registering through the web portal, it only aims at attracting new or occasional clients. But we want a registration system which is accepted by most of the regular clients for their unique user experience. Moreover, in the e-Panjeeyan web portal the user could only make an appointment and do the registration, for the payment he/she has to do it offline. So partially it was still a tedious process of registration. For overcoming this, e-Panjeeyan mobile app was needed to be built.

## 2.5.2 OBJECTIVES OF PROPOSED SYSTEM

This project is aimed at developing a mobile app for REVENUE & DISASTER MANAGEMENT, GOVT OF ASSAM. The system is an online public application that can be accessed with proper login provided. Following Features is integrated in the system:

1. User can calculate the fee associated with deed and marriage registration prior making any registration.
2. Whenever one register oneself with the app, all the personal details and information are kept confidential.

3. Once registered with the app, one will be able to check their appointment status through it.

4. Payment of the associated fee either for deed or marriage registration can be done through the app.
5. User can customize the app according to their desire, whether it be changing of the default language i.e., English to Hindi or changing of the default Day mode to Night mode.

6. For the convenience of the users, there is a user manual available in the app which would guide the users throughout the whole time

## 2.5.3 SCOPE OF THE PROPOSED SYSTEM

The e-PANJEEYAN MOBILE APPLICATION is developed to provide an appointment for deed and marriage registration and also to make payment of the associated amount through their android phones. It is a friendly user interface, provided to facilitate different services. The older traditional web portal process to register is a lengthy one and time consuming, as after registration, one has to go to the nearest counter of Circle office and wait in the queue to make payment themselves. But in the e-Panjeeyan mobile app, payment can also be done online.

The mobile app is notably good when a person needs to complete some goal, here it is to make registration and payment. This well-designed app must give our client all opportunities to perform it just in few taps or scrolls.

# 3. FEASIBILITY STUDY

## 3.1 INTRODUCTION

After initial investigation, feasibility study is carried out to check the workability of registration system. Its impacts on the organizational ability to meet user needs and effective use of the system. Feasibility study is the testing of the proposed system according to its workability. The object of the feasibility system is not to solve the problem but to acquire a sense of its scope. During the study of the problem definition is centralized and aspects of the problem to be included in the system are determined. Consequently costs and benefits are estimated with the greater accuracy at this stage. The result of the feasibility study is a normal proposal. This is simply a report a formal document detailing the nature and scope of the proposed solution. The proposal summarizes what is known and what is going to be done.

There are three key considerations involved in the feasibility analysis. They are:

1. Technical feasibility.
2. Economic feasibility.
3. Operational feasibility.

## 3.1.1 TECHNICAL FEASIBILITY

During technical feasibility, the analyst evaluates the technical merits of the system, at the same time collecting additional information about the performance, reliability, maintain ability and product ability in some cases, this system analysis step also includes limited amount of the research and design.

We can summaries the technical feasibility of the proposed system in following points:-

- The proposed system is capable of holding data to be used.

- The proposed system being modular, if the developer wants can add more features in the future and as well as being able to expand the system.

- As far as the hardware and the software are concerned, the proposed system is completely reliable.

## 3.1.2 ECONOMIC FEASIBILITY

In economic feasibility, which is also known as cost-benefit analysis, benefits expected from registration system are composed with the cost. If the benefits outweigh costs, then decision is made to design and implement the system. Otherwise, further justification or alteration in the proposed system will have to be made.

This project was developed using all open source software like Android Studio, NetBeans, MySQL database tool, etc. No costs regarding purchase of software or license were required.

The proposed system is economically feasible because of the following reasons:-

- This system is one time investment system, where all the features required will be added.
- Since the project was developed using open source tools, therefore the cost of the proposed system will be much less.

## 3.1.3 OPERATIONAL FEASIBILITY

In general, people resistant to change but the usage of mobile applications has been growing tremendously. A survey should be made of how strong a reaction the user staff is likely to have towards the development of the mobile-based system.

This project provides user friendly interactions for getting relevant results.

## 3.2 CONCLUSION

From the observation made in the feasibility study described above, it can be concluded that the proposed system is feasible in all aspects and justified that the feasible study could be followed by the system analysis and system design phase. The proposed system is also capable of holding the data to be used and is capable of providing adequate response. Thus, the proposed system is technically, economically and operationally feasible.

# 4. TECHNOLOGY USED

## 4.1 SOFTWARE SPECIFICATIONS

The proposed system is comprised of the following software tools. These tools were chosen in such a manner taking into account the need for future enhancements, system longevity and maintainability.

- Operating System: Windows 10
- Languages/Scripts: Java (Android), JSP servlet, PHP, HTML
- Application server:  Apache Tomcat, UwAmp (Apache PHP MySQL PHPMyAdmin server)
- IDE used: Android Studio and NetBeans
- Text Editor: Sublime Text
- GUI Design: Bootstrap, Cascading Style Sheets (CSS) 4.0
- Browsers: Google Chrome
- Database: MySQL

## 4.1.1 Java (Android):

Not only is Java the official programming language for Android app development (along with Kotlin), Java itself is used by Google for large parts of the Android internals. There are two distinct parts to learn Java for writing an Android app. One is the Java programming language itself, the other is understanding how to create an app in terms of its user interface, the Android OS, and the Android Software Development Kit (SDK).

Java was first released in the mid-1990s by Sun Microsystems. It was designed to be easy to learn Java by programmers who already knew C and C++. During 2006 and 2007 Sun released Java as free and open-source software, under the terms of the GNU General Public

License (GPL). Sun was bought by Oracle in 2009/2010, and Oracle remains committed to Java.

### 4.1.2 JSP Servlet:

Servlet technology is used to create a web application (resides at server side and generates a dynamic web page). Servlet can be described in many ways, depending on the context.

- o Servlet is a technology which is used to create a web application.
- o Servlet is an API that provides many interfaces and classes including documentation.
- o Servlet is an interface that must be implemented for creating any Servlet.
- o Servlet is a class that extends the capabilities of the servers and responds to the incoming requests. It can respond to any requests.
- o Servlet is a web component that is deployed on the server to create a dynamic web page.

The advantages of Servlet are as follows:

1. Better performance: because it creates a thread for each request, not process.
2. Portability: because it uses Java language.
3. Robust: JVM manages Servlets, so we don't need to worry about the memory leak, garbage collection, etc.
4. Secure: because it uses java language.

### 4.1.3 PHP:

PHP is a server-side scripting language designed for web development but also used as a general-purpose programming language. Originally created by Rasmus Lerdorf in 1995, the reference implementation of PHP is now produced by The PHP Group. While PHP

originally stood for Personal Home Page, it now stands for PHP: Hypertext Preprocessor, a recursive acronym.

PHP code is interpreted by a web server with a PHP processor module which generates the resulting web page: PHP commands can be embedded directly into an HTML source document rather than calling an external file to process data. It has also evolved to include a command-line interface capability and can be used in standalone graphical applications.

PHP is an HTML-embedded scripting language. Much of its syntax is borrowed from C, Java and Perl with a couple of unique PHP-specific features thrown in. The goal of the language is to allow web developers to write dynamically generated pages quickly.

## 4.1.4 HTML:

Hyper Text Markup Language, commonly referred to as HTML, is the standard markup language used to create web pages. It is written in the form of HTML elements consisting of tags enclosed in angle brackets (like <html>). HTML tags most commonly come in pairs like <h1> and </h1>, although some represent empty elements and so are unpaired, for example <img>. The first tag in such a pair is the start tag, and the second is the end tag (they are also called opening tags and closing tags).

Web browsers can read HTML files and render them into visible or audible web pages. Browsers do not display the HTML tags and scripts, but use them to interpret the content of the page. HTML elements form the building blocks of all websites. HTML allows images and objects to be embedded and can be used to create interactive forms. It provides a means to create structured documents by denoting structural semantics for text such as headings, paragraphs, lists, links, quotes and other items. It can embed scripts written in languages such as JavaScript which affect the behavior of HTML web pages.

## 4.1.5 Apache Tomcat:

Apache Tomcat (also referred to as Tomcat Server) implements several Java EE specifications including Java Servlet, JavaServ3er Pages (JSP), Java EL, and WebSocket, and provides a "pure Java" HTTP web server environment in which Java code can run.

Tomcat is developed and maintained by an open community of developers under the auspices of the Apache Software Foundation, released under the Apache License 2.0 license, and is open-source software.

## 4.1.6 UwAmp:

UwAmp is a type of package WAMP involving Apache , PHP , MySQL , SQLite and

phpMyAdmin  simplifies the creation of a Web server on Windows to develop websites in PHP. What characterizes this WAMP package is its management interface to configure Apache and PHP without having to modify the configuration files by hand.

## 4.1.7 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. It is a replacement for the Eclipse Android Development Tools (ADT) as the primary IDE for native Android application development.

## 4.1.8 NetBeans:

NetBeans is an integrated development environment (IDE) for Java. NetBeans allows applications to be developed from a set of modular software components called *modules*.

NetBeans runs on Windows, macOS, Linux and Solaris. In addition to Java development, it has extensions for other languages like PHP, C, C++, HTML5, and JavaScript. Applications based on NetBeans, including the NetBeans IDE, can be extended by third party developers.

NetBeans IDE is an open-source integrated development environment. NetBeans IDE supports development of all Java application types (Java SE (including JavaFX), Java ME, web, EJB and mobile applications) out of the box.

## 4.1.9 Sublime Text:

Sublime Text is a proprietary cross-platform source code editor with a Python application programming interface (API). It natively supports many programming languages and markup languages, and functions can be added by users with plugins, typically community-built and maintained under free-software licenses.

## 4.1.10 BOOTSTRAP:

Bootstrap is a free collection of tools for creating websites and web applications. It is an open-source JavaScript framework developed by the team at Twitter. It contains HTML and CSS-based design templates for typography, forms, buttons, navigation and other interface components, as well as optional JavaScript extensions.

**One Can Use Bootstrap to:**

1) Save time and repetition with Bootstrap's powerful collection of pre-made templates, classes, and grid layouts.
2) Build and deploy "platform agnostic" websites at a pinch.
3) Fluid grid layouts and templates make mobile-first development easy.
4) Light weight and expandable.

5) Compatibility across a huge range of browsers - including Internet Explorer 6!

## 4.1.11 CSS:

**Cascading Style Sheets (CSS)** is a style sheet language used for describing the look and formatting of a document written in a markup language. While most often used to change the style of web pages and user interfaces written in HTML and XHTML, the language can be applied to any kind of XML document, including plain XML, SVG and XUL. Along with HTML and JavaScript, CSS is a cornerstone technology used by most websites to create visually engaging Webpages, user interfaces for web applications, and user interfaces for many mobile applications.

## 4.1.12 MySQL:

**What is a Database?**

A database is a separate application that stores a collection of data. Each database has one or more distinct APIs for creating, accessing, managing, searching and replicating the data it holds. Other kinds of data stores can be used, such as files on the file system or large hash tables in memory but data fetching and writing would not be so fast and easy with those types of systems.

So now a day, we use relational database management systems (RDBMS) to store and manage huge volume of data. This is called relational database because all the data is stored into different tables and relations are established using primary keys or other keys known as foreign keys

**MySQL Database:**

**MySQL** is an open-source relational database management system (RDBMS). It is the most popular Open Source SQL database management system. In July 2013, it was the world's second most widely used RDBMS, and the most widely used open-source client–server model RDBMS. It is named after co-founder Monty Widenius's daughter, **My**. The name of Dolphin in MySQL logo is "Sakila". The SQL part of "MySQL" stands for "Structured Query Language". MySQL is developed, distributed, and supported by Oracle Corporation.

MySQL is becoming so popular because of many good reasons:

➢ MySQL is released under an open-source license. So we have nothing to pay to use it.

➢ MySQL is a very powerful program in its own right. It handles a large subset of the functionality of the most expensive and powerful database packages.

➢ MySQL uses a standard form of the well-known SQL data language.

➢ MySQL works on many operating systems and with many languages including PHP, PERL, C, C++, JAVA, etc.

➢ MySQL works very quickly and works well even with large data sets.

➢ MySQL supports large databases, up to 50 million rows or more in a table. The default file size limit for a table is 4GB, but we can increase this (if operating system can handle it) to a theoretical limit of 8 million terabytes (TB).

## 4.2 HARDWARE SPECIFICATIONS

**Hardware used for developing the system**

- ✓ Processor: Intel Core i5 processor
- ✓ Memory: 8GB RAM
- ✓ Hard disk: 1TB

# 5. SYSTEM ANALYSIS

## 5.1 INTRODUCTION

System analysis is a detailed study of the various operations performed by a system and their relationships within and outside of the system. It is a systematic technique that refines goals and objectives. The goal of the system and development is to deliver the system in line with the user's requirement, and analysis is the heart of the process.

System study has been conducted with the following objectives in mind:

1. Identify the client's needs.
2. Evaluate the system concept for feasibility
3. Perform economical and technical analysis.
4. Allocate functions to hardware, software, people, database and other system elements.
5. Establish cost and schedule constraints.

Both hardware and software expertise is required to successfully attain objectives listed above.

## 5.2 SOFTWARE ENGINEERING PARADIGMS APPLIED

A system engineer must incorporate a development strategy to solve the actual problem in the organization. The strategy that contains the process, method and tests layers and the generic phase referred to as a software engineering paradigm. The software engineering paradigm is chosen on the basis of the nature of the project and application, the method and tools to be used, and the delivers that are required.

In the proposed system, the waterfall model constitute the output of one process becomes the input to another process and this transformation continues till the results is not achieved.

# ITERATIVE WATERFALL MODEL

Iterative Waterfall Model is an approach of software engineering in which the process is structured as a cascade of phases, when the outputs of one phase constitute the input to the next one. If any defect is detected in any phase, then engineers can go back to the phase where the defect had occurred and redo some of the work done during that phase and the subsequent phases to correct the defect and its effect on the later phases. Each phase is regarded as the set of activities that are executed by different people concurrently.

The phases involved are shown with the help of diagram as--
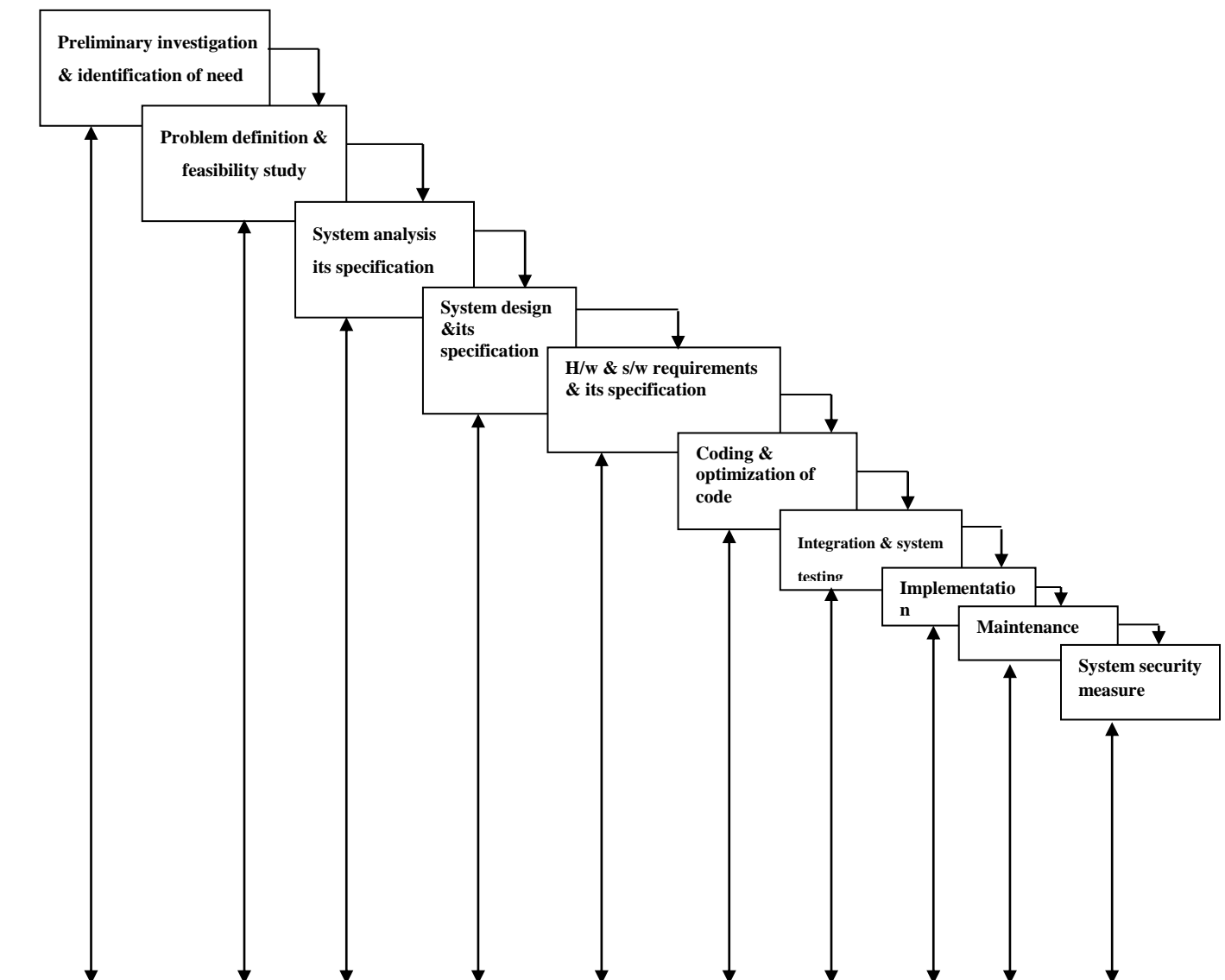
Iterative Waterfall Model:

1. Preliminary investigation and identification of need.
2. Problem definition and feasibility study.
3. System analysis and its specification.
4. System design and its specification.
5. Hardware, software requirements and its specification.
6. Coding and optimization of code.
7. Integration and system testing.
8. Implementation
9. Maintenance
10. System security measure.

Depends upon the organization the waterfall model is describe in various stages, underlying philosophy is the same so that the comments we give here apply to all of them.

The various phases of waterfall model are detailed below:

**Preliminary investigation and identification of needs**- Requirements is defined and determined in the life cycle of the system development.

**Problem definition and feasibility study**- The problem is defined in this phase and technical, economic and operational feasibility is measured.

**System analysis and its specification**- This phase include the detailed information of the requirements of both the user and software and analysis the parameters like

✓ Input to the system
✓ Process required

✓ Output expected

✓ Constraints.

**System design and its specification-** This phase mainly deals with the software architecture and data structure, etc. This phase process the requirements into a representation of the software. This also answers to the questions raised about the tasks.

**Hardware, software requirement and specification-** This phase deals with the various components such as the tangible or non-tangible of the specified system.

**Coding and optimization of code-** In this phase the code required to complete the software is done and each module is tested to achieve the goal. Coding refers to the translation of the design into machine understandable form. The more detail the design the more easy is the coding and better is its reliability.

**Integration and system testing-** After completion of the coding phase of each module it is then required to be integrated with other modules and then finally all the modules are needed to be tested rigorously for correcting of the code and results. Testing may involve the individual unit and the whole system. It requires a detailed plan as to what, when and how to test.

**Implementation-** After completing the integration and system testing phase which implies that the software is not anymore required for any further tests and is corrected completely. It is then needed to be implemented at the client site.

**Maintenance-** After successful installation and implementation, it is evaluated to work satisfactory and perform all the tasks desired from it. If any further requirement arises it is to be maintained and if any modification is needed that also be done in this phase.

**System security measure-** The last phase deals with system security, which is done by authorized user name and regular password check.

## 5.3 STRUCTURED ANALYSIS

One of the best approaches to system analysis is the structural analysis. Structural analysis is a set of techniques and graphical tools that allow us to develop a new kind of system specifications that is easily understandable to the developer. It is the detailed step by step investigation of related procedures to see what must be done and to determine the best way of doing it. The objection is to build a system specification that provides the basis for design and implementation.

### 5.3.1 Data Flow Diagrams:

Data flow diagrams illustrate how data is processed by a system in terms of inputs and outputs. Data Flow Diagramming is a means of representing a system at any level of detail with a graphic network of symbols showing data flows, data stores, data processes, and data destinations/data sources.

The Data Flow Diagram is analogous to a road map. It is a network of all possibilities with different detail shown, on different hierarchical levels. The process of representing different detail levels is called leveling or partitioning by some data flow diagram advocates. Like a road map, there is no start or stop point, no time or timing, or steps to get somewhere. We just

know that the data path must exist because at some point it will be needed. A road map shows all existing or planned roads because at some point it will be detail that is not shown on the different levels of the data flow diagram such as volumes, timing, frequency, etc. is shown on supplementary diagrams or in the data dictionary. For example, data store contents may be shown in the data dictionary.

## 5.3.2 Purpose/Objective:

1. Graphical, eliminating thousands of words, logical representations, modeling what a system does, rather than physical models showing HOW it does it.
2. Hierarchical, showing systems at any level of detail, and
3. Jargon less, allowing user understanding and reviewing.

The goal of data flow diagramming is to have a commonly understood model of a system. The diagrams are the basis of structured systems analysis. Data Flow Diagrams are supported by other techniques of structured systems analysis such as data structure diagrams, data dictionaries and procedure representing techniques such as decision tables, decision trees and structured English. Data Flow Diagrams have the objective of avoiding the cost of User/developer misunderstanding of a system, resulting in a need to redo systems or in not using the system. Having to start documentation from scratch when the physical system changes since the logical system, what gets done often remains the same when technology changes. Systems inefficiencies shows up because a system gets computerized before it gets systematized, being unable to evaluate system project boundaries or degree automation, resulting in a project of inappropriate scope.

**Creating a DFD:**

Step 1: Plan the Solution

- ✓ Identify the inputs, outputs and external entities for the system.
- ✓ Identify the top-level processes in the system.
- ✓ Identify the detailed processes of the system.

Step 2: Implement the Solution

- ✓ Draw the Context Analysis Diagram (CAD).
- ✓ Draw the Top Level DFD.
- ✓ Draw the detailed Logical DFD.
- ✓ Verify the Solution
- ✓ Get approval of the design from your client.

**Context Analysis Diagram:**

A Context Analysis Diagram (CAD) is a diagrammatic tool used to describe a system under study. It shows the people and the organization interacting with the system. This interaction is depicted in terms of the input to and output from the system.

A Data Flow Diagram is of two types, **Physical** and **Logical**.

**Physical DFD:**

The Physical Data Flow diagram (DFD) reveals the actual device and people that perform the functions. It shows the physical components of a system. The emphasis of this type of DFD is on the physical characteristics of a system. It depicts the various people doing jobs in an organization.

**Logical DFD:**

A Logical DFD shows the ongoing activities of the system. It does not show us how these tasks are done or who does these tasks.

## 5.3.3 Data Flow Diagram (DFD) Elements:

The following four elements are used in the Data Flow Diagrams:

1. An External Entity
2. A Data Flow
3. A Process
4. A Data store.

**An External Entity:**

An external entity could either is a source or a destination of data in the system design being constructed. It lies outside the context of the system. It represented by a solid square.

Name of the
External Entity
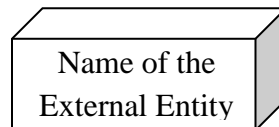
If entity is need to represent more than once then both instance of the entity are represented as follow.

**A Process:**

A process indicates the work that is performed on data. It transforms data from one form to another.

Process No.

Process Name

or

Process No.

Process Name
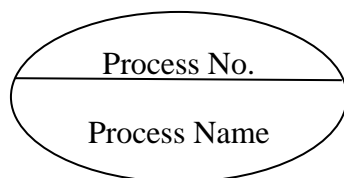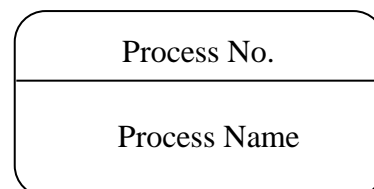
## A Data Flow:

A data flow takes place between the various components of the system. In Data Flow Diagram the data flow is represented as the thin line pointing in the direction in which the data is flowing.

## Data Store:

A data store is repository for the data. While making a logical design if it is require storing the data, data store is used. A data store is represented by open rectangle. It also has a number and name.

Data store name | No. |

## Levels of a DFD:

The DFD is designed to aid communication. If it contains dozens of processes and data stores, it gets too unwieldy. The rule of thumb is therefore to explode the DFD to a functional level so that the next sublevel does not exceed 10 processes. Beyond that it is best to take each function separately and expand it to show the explosion of the single process. If a user wants to know what happens within a given process, then the detailed explosion of that process may be shown.

The following is the DFD for **e-PANJEEYAN FOR REVENUE & DISASTER MANAGEMENT, GOVT OF ASSAM** :
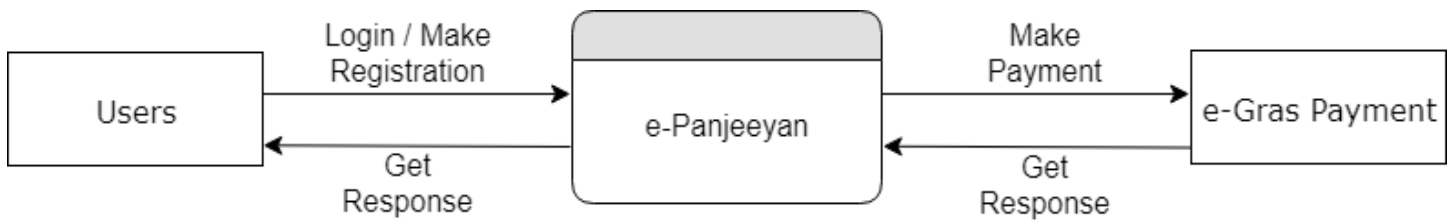


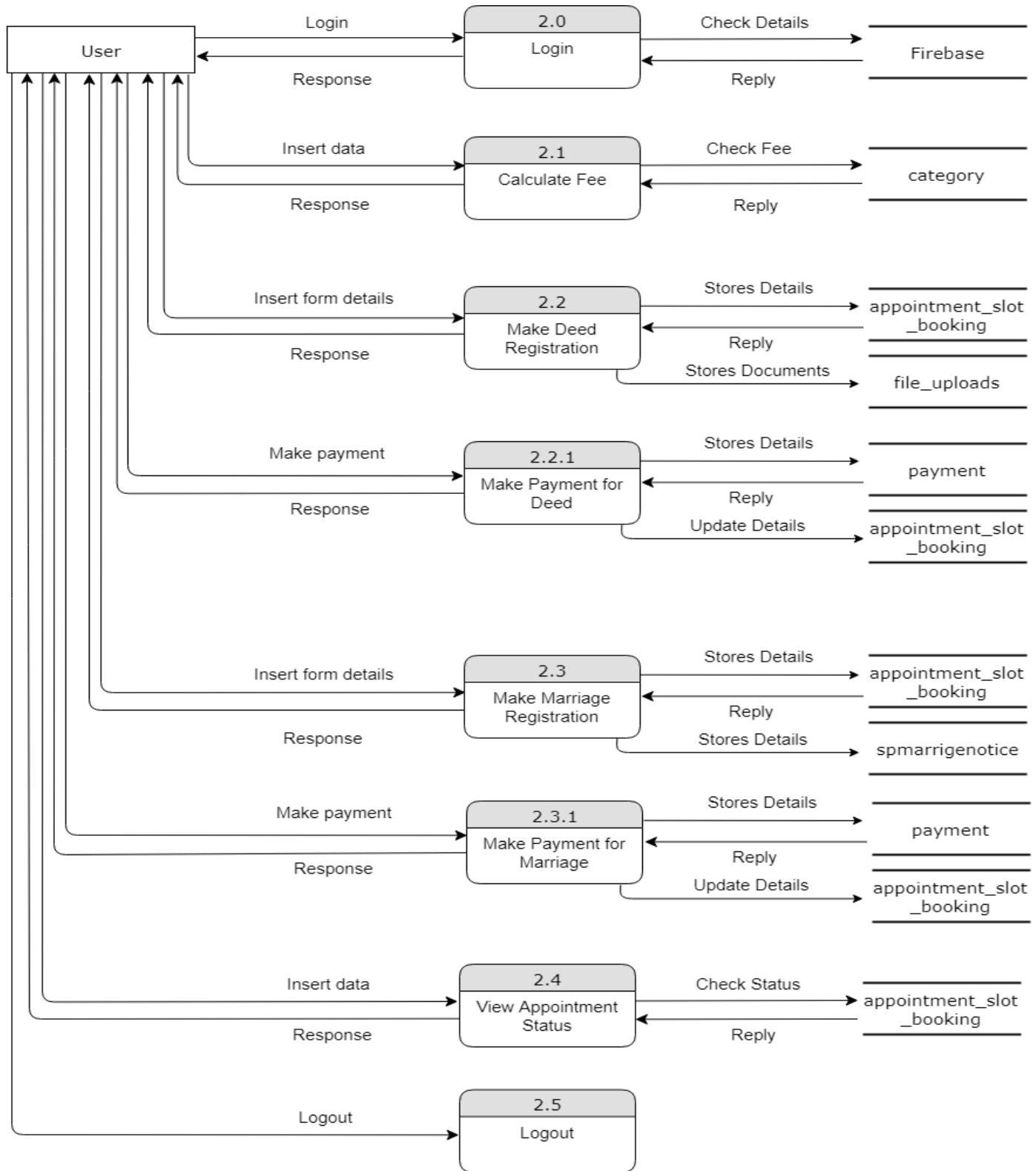Fig: 0-LEVEL DFD (CONTEXT LEVEL)
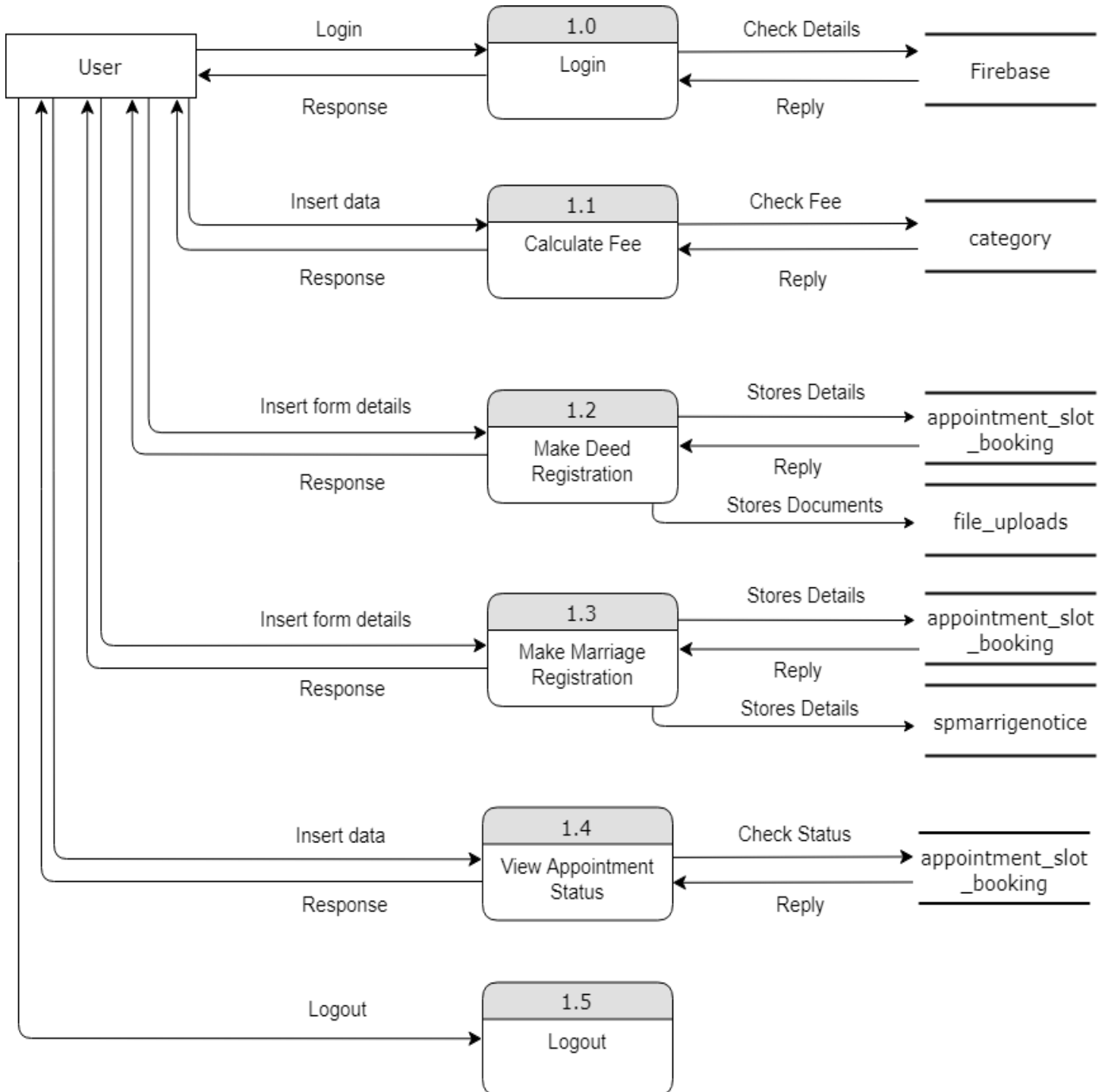
Fig: Level 1 DFD

Fig: Level 2 DFD

# 6. SYSTEM DESIGN

## 6.1 INTRODUCTION

System design is the most crucial phase of a system development life cycle. It produces the details that state how a system will meet the requirements identified during system analysis. Absence of proper design is bound to lead to confusion and errors. A complete design must accommodate all of the implicit requirements desired by the customer. It must be a readable, understandable guide for programmers. It should provide a complete picture of the system addressing the data, functional, and behavioral domains from an implementation perspective. System design starts after the software requirements have been analyzed and specified. System analysis specifies what a system should do to meet the needs of user, but system design specifies how the system will achieve those objectives.

The objectives in designing information system are –

- Specify the logical design elements
- Ensure that system features meet the user requirements
- Physical factors (that effect the performance, comfort and specification of direct users) design
- Provide detailed software development specification

## 6.2 DESIGN ANALYSIS

System design goes through two phase of development. These are –
- Logical Design
- Physical Design

## Logical Design:

Logical design involves developing general specifications for how the basic information system activities of input, processing, output, storage and control can meet the end user requirements.

## Physical Design:

Physical construction, the activity following logical design, produces program software, files and a working system. Design specifications instruct programmers about what the system do. Several development activities are carried out during the design phase, some of the most important are –

- User Interface/System Interface design
- Database design

## User Interface/System Interface Design:

Generally the first step in the system design is the user interface design. The user interface design activity focuses on the preparation of input and design of output reports in a form of acceptable to the users. It concentrates on input/output method and the conversion of data and information between human and machine-readable forms. So, user interface design includes detailed specification for display screens, interactive dialogues, audio responses, reports etc. **Forms Design** is an important phase of System Design in order to produce data in the right form for input and the information produced must be in a format acceptable to the user.

## 6.3 DATABASE DESIGN

Database design is the process of producing a detailed data model of a database. The term database design can be used to describe many different parts of the design of an overall database system. Principally, and most correctly, it can be thought of as the logical

design of the base data structures used to store the data. In the relational model these are the tables and views.
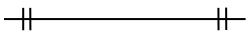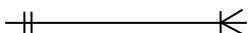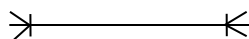
### 6.3.1 Entity Relationship Diagram (ERD)

Entity Relationship (ER) model is a popular high-level conceptual data model. The conceptual schema is a concise description of data requirement of the user and includes detailed description of the entity type, relationships, and constraints. These concepts do not include implementations details; they are easier to understand and can be used to communicate with non-technical users. This approach enables the database designers to concentrate on specifying the properties of data, without being concerned with storage details.

**The Building Blocks: Entities, Relationships, and Attributes**

**Entity**: An entity may be defined as a thing which is recognized as being capable of an independent existence and which can be uniquely identified.

**Relationships:** ERD represents the **relationship** between various entities and their attributes. Relationship between entities makes up a data structure. There are three type of relationship we can find in an ERD,

- One to one
- One to many
- Many to many

**Attributes**: An Attribute is a specification that defines a property of an object, element, or file. It may also refer to or set the specific value for a given instance of such.

Entities and relationships can both have attributes. Every entity (unless it is a weak entity) must have a minimal set of uniquely identifying attributes, which is called the entity's primary key, and a set of attributes that are used to refer other entities, which are called the entity's foreign keys.
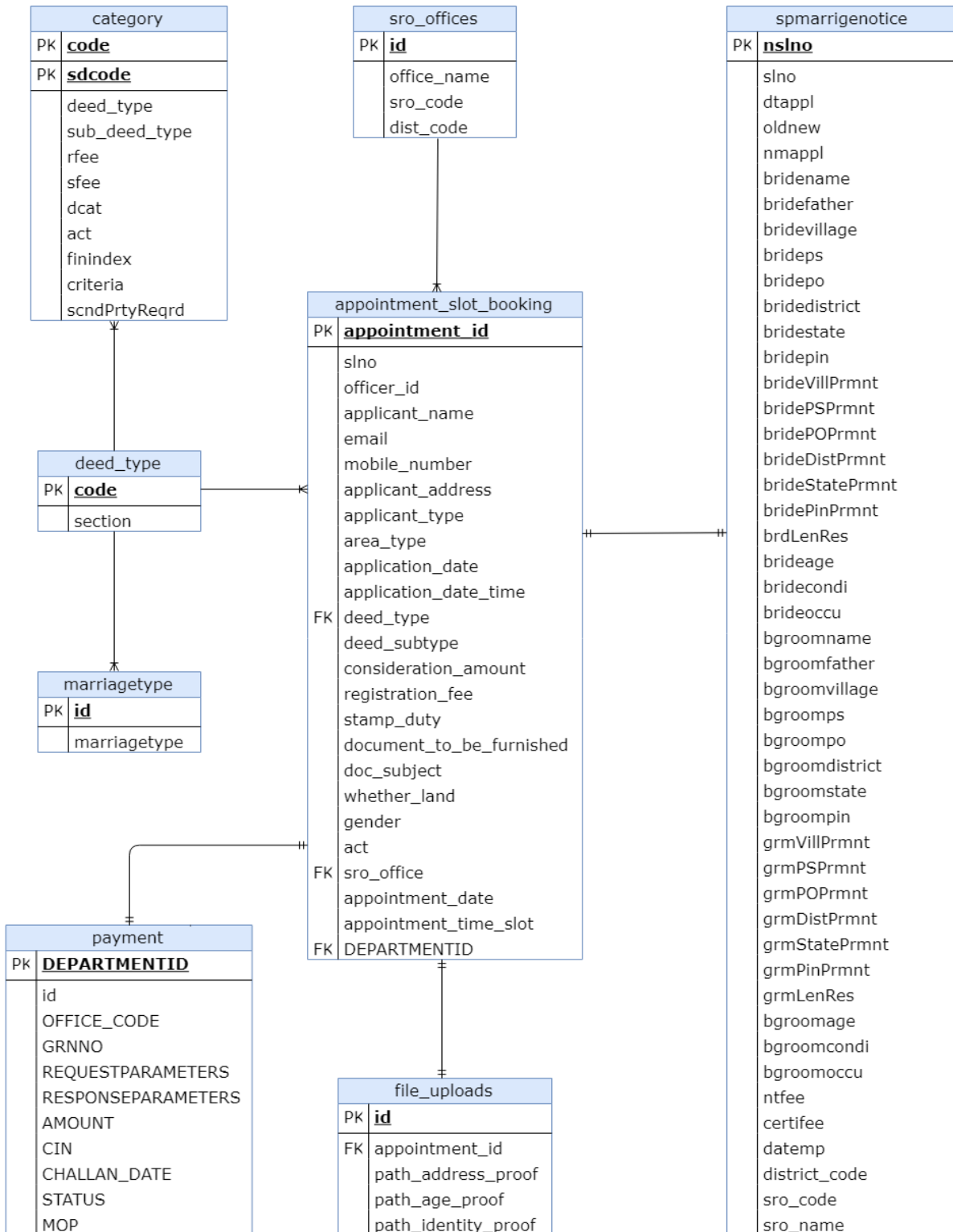
## category

| | |
|---|---|
| PK | **code** |
| PK | **sdcode** |
| | deed_type |
| | sub_deed_type |
| | rfee |
| | sfee |
| | dcat |
| | act |
| | finindex |
| | criteria |
| | scndPrtyReqrd |

## sro_offices

| | |
|---|---|
| PK | **id** |
| | office_name |
| | sro_code |
| | dist_code |

## spmarrigenotice

| | |
|---|---|
| PK | **nslno** |
| | slno |
| | dtappl |
| | oldnew |
| | nmappl |
| | bridename |
| | bridefather |
| | bridevillage |
| | brideps |
| | bridepo |
| | bridedistrict |
| | bridestate |
| | bridepin |
| | brideVillPrmnt |
| | bridePSPrmnt |
| | bridePOPrmnt |
| | brideDistPrmnt |
| | brideStatePrmnt |
| | bridePinPrmnt |
| | brdLenRes |
| | brideage |
| | bridecondi |
| | brideoccu |
| | bgroomname |
| | bgroomfather |
| | bgroomvillage |
| | bgroomps |
| | bgroompo |
| | bgroomdistrict |
| | bgroomstate |
| | bgroompin |
| | grmVillPrmnt |
| | grmPSPrmnt |
| | grmPOPrmnt |
| | grmDistPrmnt |
| | grmStatePrmnt |
| | grmPinPrmnt |
| | grmLenRes |
| | bgroomage |
| | bgroomcondi |
| | bgroomoccu |
| | ntfee |
| | certifee |
| | datemp |
| | district_code |
| | sro_code |
| | sro_name |

## deed_type

| | |
|---|---|
| PK | **code** |
| | section |

## marriagetype

| | |
|---|---|
| PK | **id** |
| | marriagetype |

## appointment_slot_booking

| | |
|---|---|
| PK | **appointment_id** |
| | slno |
| | officer_id |
| | applicant_name |
| | email |
| | mobile_number |
| | applicant_address |
| | applicant_type |
| | area_type |
| | application_date |
| | application_date_time |
| FK | deed_type |
| | deed_subtype |
| | consideration_amount |
| | registration_fee |
| | stamp_duty |
| | document_to_be_furnished |
| | doc_subject |
| | whether_land |
| | gender |
| | act |
| FK | sro_office |
| | appointment_date |
| | appointment_time_slot |
| FK | DEPARTMENTID |

## payment

| | |
|---|---|
| PK | **DEPARTMENTID** |
| | id |
| | OFFICE_CODE |
| | GRNNO |
| | REQUESTPARAMETERS |
| | RESPONSEPARAMETERS |
| | AMOUNT |
| | CIN |
| | CHALLAN_DATE |
| | STATUS |
| | MOP |

## file_uploads

| | |
|---|---|
| PK | **id** |
| FK | appointment_id |
| | path_address_proof |
| | path_age_proof |
| | path_identity_proof |

Fig: ERD for e-Panjeeyan

## 6.4 DATABASE TABLE STRUCTURE

### 1. Table **applicant_type**

| # | Name | Type | Collation | Attributes | Null | Default | Extra |
|---|------|------|-----------|------------|------|---------|-------|
| 1 | slno | int(11) | | | No | None | AUTO_INCREMENT |
| 2 | type | varchar(300) | latin1_swedish_ci | | No | None | |
| 3 | exempted | tinyint(1) | | | No | 0 | |

### 2. Table **deed_type**

| # | Name | Type | Collation | Attributes | Null | Default |
|---|------|------|-----------|------------|------|---------|
| 1 | code | int(11) | | | No | None |
| 2 | section | varchar(50) | utf8_general_ci | | Yes | NULL |

### 3. Table **category**

| # | Name | Type | Collation | Attributes | Null | Default |
|---|------|------|-----------|------------|------|---------|
| 1 | code | int(10) | | | No | None |
| 2 | deed_type | varchar(100) | utf8_general_ci | | Yes | NULL |
| 3 | sdcode | int(10) | | | No | 0 |
| 4 | sub_deed_type | varchar(100) | utf8_general_ci | | Yes | NULL |
| 5 | rfee | int(10) | | | Yes | NULL |
| 6 | sfee | int(10) | | | Yes | NULL |
| 7 | dcat | varchar(100) | utf8_general_ci | | Yes | NULL |
| 8 | act | varchar(100) | utf8_general_ci | | Yes | NULL |
| 9 | finindex | varchar(100) | utf8_general_ci | | Yes | NULL |
| 10 | criteria | varchar(100) | utf8_general_ci | | Yes | NULL |
| 11 | scndPrtyReqrd | tinyint(1) | | | No | 1 |

## 4. Table **appointment_slot_booking**

| # | Name | Type | Collation | Attributes | Null | Default |
|---|------|------|-----------|------------|------|---------|
| 1 | slno | int(6) | | | Yes | NULL |
| 2 | appointment_id | varchar(20) | utf8_general_ci | | No | None |
| 3 | officer_id | int(11) | | | No | None |
| 4 | applicant_name | varchar(50) | utf8_general_ci | | No | None |
| 5 | email | varchar(30) | utf8_general_ci | | No | None |
| 6 | mobile_number | varchar(10) | utf8_general_ci | | No | None |
| 7 | applicant_address | varchar(200) | utf8_general_ci | | No | None |
| 8 | applicant_type | varchar(20) | utf8_general_ci | | No | None |
| 9 | area_type | varchar(20) | utf8_general_ci | | Yes | NULL |
| 10 | application_date | date | | | Yes | NULL |
| 11 | application_date_time | datetime | | | Yes | NULL |
| 12 | deed_type | int(10) | | UNSIGNED | Yes | NULL |
| 13 | deed_subtype | varchar(200) | utf8_general_ci | | Yes | NULL |
| 14 | consideration_amount | int(11) | | | Yes | NULL |
| 15 | registration_fee | int(11) | | | Yes | NULL |
| 16 | stamp_duty | int(11) | | | Yes | NULL |
| 17 | document_to_be_furnished | varchar(200) | utf8_general_ci | | Yes | NULL |
| 18 | doc_subject | varchar(300) | utf8_general_ci | | Yes | NULL |
| 19 | whether_land | tinyint(1) | | | No | 0 |
| 20 | gender | char(2) | utf8_general_ci | | Yes | NULL |
| 21 | act | varchar(10) | utf8_general_ci | | No | 0 |
| 22 | sro_office | int(10) | | | No | None |
| 23 | appointment_date | datetime | | | Yes | NULL |
| 24 | appointment_time_slot | time | | | Yes | NULL |
| 25 | DEPARTMENTID | varchar(150) | utf8_general_ci | | Yes | NULL |

## 5. Table **sro_offices**

| # | Name | Type | Collation | Attributes | Null | Default |
|---|------|------|-----------|------------|------|---------|
| 1 | id | int(11) | | | No | None |
| 2 | office_name | varchar(50) | latin1_swedish_ci | | No | None |
| 3 | sro_code | varchar(2) | latin1_swedish_ci | | No | None |
| 4 | dist_code | varchar(2) | latin1_swedish_ci | | No | None |

## 6. Table **file_uploads**

| # | Name | Type | Collation | Attributes | Null | Default |
|---|------|------|-----------|------------|------|---------|
| 1 | id | int(11) | | | No | None |
| 2 | appointment_id | varchar(300) | latin1_swedish_ci | | No | None |
| 3 | path_address_proof | varchar(300) | latin1_swedish_ci | | No | None |
| 4 | path_age_proof | varchar(300) | latin1_swedish_ci | | No | None |
| 5 | path_identity_proof | varchar(300) | latin1_swedish_ci | | No | None |

## 7. Table **holiday**

| # | Name | Type | Collation | Attributes | Null | Default | Extra |
|---|------|------|-----------|------------|------|---------|-------|
| 1 | id | int(11) | | | No | None | AUTO_INCREMENT |
| 2 | date | date | | | No | None | |

## 8. Table **off_days**

| # | Name | Type | Collation | Attributes | Null | Default | Extra |
|---|------|------|-----------|------------|------|---------|-------|
| 1 | id | int(11) | | | No | None | AUTO_INCREMENT |
| 2 | dist_code | varchar(3) | latin1_swedish_ci | | No | None | |
| 3 | sro_code | varchar(3) | latin1_swedish_ci | | No | None | |
| 4 | date | date | | | No | None | |

## 9. Table **spmarrigenotice**

| # | Name | Type | Collation | Attributes | Null | Default |
|---|------|------|-----------|------------|------|---------|
| 1 | slno | int(11) | | | Yes | NULL |
| 2 | nslno | char(20) | utf8_general_ci | | No | None |
| 3 | dtappl | date | | | No | None |
| 4 | oldnew | char(6) | utf8_general_ci | | Yes | NULL |
| 5 | nmappl | char(40) | utf8_general_ci | | Yes | NULL |
| 6 | bridename | char(100) | utf8_general_ci | | Yes | NULL |
| 7 | bridefather | char(100) | utf8_general_ci | | Yes | NULL |
| 8 | bridevillage | char(250) | utf8_general_ci | | Yes | NULL |
| 9 | brideps | char(50) | utf8_general_ci | | Yes | NULL |
| 10 | bridepo | varchar(50) | utf8_general_ci | | Yes | NULL |
| 11 | bridedistrict | char(50) | utf8_general_ci | | Yes | NULL |
| 12 | bridestate | char(30) | utf8_general_ci | | Yes | NULL |
| 13 | bridepin | int(6) | | | Yes | NULL |
| 14 | brideVillPrmnt | varchar(250) | utf8_general_ci | | Yes | NULL |
| 15 | bridePSPrmnt | varchar(100) | utf8_general_ci | | Yes | NULL |
| 16 | bridePOPrmnt | varchar(50) | utf8_general_ci | | Yes | NULL |
| 17 | brideDistPrmnt | varchar(100) | utf8_general_ci | | Yes | NULL |
| 18 | brideStatePrmnt | varchar(100) | utf8_general_ci | | Yes | NULL |
| 19 | bridePinPrmnt | int(6) | | | Yes | NULL |
| 20 | brdLenRes | varchar(5) | utf8_general_ci | | Yes | NULL |
| 21 | brideage | int(2) | | | Yes | NULL |
| 22 | bridecondi | char(50) | utf8_general_ci | | Yes | NULL |
| 23 | brideoccu | char(50) | utf8_general_ci | | Yes | NULL |
| 24 | bgroomname | char(100) | utf8_general_ci | | Yes | NULL |
| 25 | bgroomfather | char(100) | utf8_general_ci | | Yes | NULL |
| 26 | bgroomvillage | char(250) | utf8_general_ci | | Yes | NULL |
| 27 | bgroomps | char(50) | utf8_general_ci | | Yes | NULL |
| 28 | bgroompo | varchar(50) | utf8_general_ci | | Yes | NULL |
| 29 | bgroomdistrict | char(50) | utf8_general_ci | | Yes | NULL |
| 30 | bgroomstate | char(30) | utf8_general_ci | | Yes | NULL |
| 31 | bgroompin | int(6) | | | Yes | NULL |
| 32 | grmVillPrmnt | varchar(250) | utf8_general_ci | | Yes | NULL |
| 33 | grmPSPrmnt | varchar(100) | utf8_general_ci | | Yes | NULL |
| 34 | grmPOPrmnt | varchar(50) | utf8_general_ci | | Yes | NULL |
| 35 | grmDistPrmnt | varchar(100) | utf8_general_ci | | Yes | NULL |
| 36 | grmStatePrmnt | varchar(100) | utf8_general_ci | | Yes | NULL |
| 37 | grmPinPrmnt | int(6) | | | Yes | NULL |
| 38 | grmLenRes | varchar(5) | utf8_general_ci | | Yes | NULL |
| 39 | bgroomage | int(2) | | | Yes | NULL |
| 40 | bgroomcondi | char(50) | utf8_general_ci | | Yes | NULL |
| 41 | bgroomoccu | char(50) | utf8_general_ci | | Yes | NULL |
| 42 | ntfee | int(3) | | | Yes | NULL |
| 43 | certifee | int(3) | | | Yes | NULL |
| 44 | datemp | date | | | Yes | NULL |
| 45 | district_code | varchar(30) | utf8_general_ci | | Yes | NULL |
| 46 | sro_code | varchar(30) | utf8_general_ci | | Yes | NULL |
| 47 | sro_name | varchar(40) | utf8_general_ci | | Yes | NULL |

## 10. Table **marriage_type**

| # | Name | Type | Collation | Attributes | Null | Default | Extra |
|---|------|------|-----------|------------|------|---------|-------|
| 1 | **id** | int(11) | | | No | None | AUTO_INCREMENT |
| 2 | **marriageType** | varchar(30) | utf8_general_ci | | Yes | NULL | |

## 11. Table **payment**

| # | Name | Type | Collation | Attributes | Null | Default | Extra |
|---|------|------|-----------|------------|------|---------|-------|
| 1 | **id** | int(11) | | | No | None | AUTO_INCREMENT |
| 2 | **DEPARTMENTID** | varchar(150) | latin1_swedish_ci | | No | None | |
| 3 | **OFFICE_CODE** | varchar(6) | latin1_swedish_ci | | No | None | |
| 4 | **GRNNO** | varchar(100) | latin1_swedish_ci | | Yes | NULL | |
| 5 | **REQUESTPARAMETERS** | varchar(1000) | latin1_swedish_ci | | No | None | |
| 6 | **RESPONSEPARAMETERS** | varchar(1000) | latin1_swedish_ci | | Yes | NULL | |
| 7 | **AMOUNT** | int(11) | | | No | 0 | |
| 8 | **CIN** | varchar(100) | latin1_swedish_ci | | Yes | NULL | |
| 9 | **CHALLAN_DATE** | date | | | No | None | |
| 10 | **STATUS** | varchar(5) | latin1_swedish_ci | | Yes | NULL | |
| 11 | **MOP** | varchar(10) | latin1_swedish_ci | | Yes | NULL | |

# 7. CODING

## 7.1 Introduction

Once the design is completed, most of the major decisions about the system have been made. Now the design must be translated into a machine-readable form. The Code generation step performs this task. If the design performed in a detailed manner, code generation can be accomplished mechanistically.

The goal of coding or programming phase is to translate the design of the system produced in design phase into code in a given programming language, which can be executed by a Computer and that performs computations specified by the design. The Code generation should be designed in a manner to reduce input, control errors and accelerate the entire process.

Normally, good software development organizations adhere to some well-defined and standard style of Coding called Standards. Most software development organizations formulate their own coding standards that suit them most, the reasons for adhering to a standard coding style are as follows-

- It gives a uniform appearance to the codes written by different programmers.

- It enhances code understandability.

- It encourages good programming practices.

I have been using the DAO pattern in developing this application as this is the best practice used by experienced object-oriented software developers.

**Design Pattern in Java**: Data Access Object Pattern or DAO pattern is used to separate low level data accessing API or operations from high level business services. Following are the participants in Data Access Object Pattern.

- Data Access Object Interface - This interface defines the standard operations to be performed on a model object(s).
- Data Access Object concrete class - This class implements above interface. This class is responsible to get data from a data source which can be database / xml or any other storage mechanism.
- Model Object or Value Object - This object is simple POJO containing get/set methods to store data retrieved using DAO class.

## 7.2 CODE EFFICIENCY

Code efficiency is one of the major tasks of the System Development Life Cycle (SDLC). It checks as to whether the access time is minimized or not, whether the errors are eliminated or not, whether the data integrity is maintained or not etc. To enhance the efficiency of the system, drop-down list are provided. Besides, drop down list box for item name entries are also provided. It is done so that the user would have the options to select the item name else they may enter incorrectly.

## 7.3 OPTIMIZATION OF CODE

Code optimization is one of the main tasks of the SDLC. It supplements the efficiency of the coding and is the penultimate stage of removal of any data redundancy and unnecessary occupation of the space.

In the proposed system, different types of the codes have been optimized such as looping codes.

Example: (for an unoptimized code)

If (condition 1> condition 2)

{

    If (condition 1> condition 3)

    Statement;

}

Now Example of optimized code-

If (condition 1> condition 2 && condition 1> condition 3)

{

    Statement;

}

Besides, many checking features have been accommodated to stop the entry of unoptimized code. Necessary check constraints have been declared to stop unauthorized accessing retrieving of the data. If the user enters wrong data, the system has been designed to prompt the user, suggesting entering the correct data.

## 7.4 EVALUATION

Evaluation is to identify whether the system is serving the intended purpose of the organization and meeting the expected requirements of the user, its strength and weakness. For evaluation of the system, a list of satisfied users will be given to the organization concern, which is using the identical system for seeking their opinion.

# 8. SYSTEM TESTING

## 8.1 OVERVIEW

The quality of an information system depends on its design, development testing and implementation. One aspects of system testing is its reliability. A system is reliable if, used in a reasonable manner; it does not produce failures that are dangerous or over time costly.

An additional aspect of quality assurance is avoiding the need for enhancement on the one hand and developing software that is maintainable on the other. The need for maintenance is very high and impedes new developments. The greatest amount of maintenance is for user enhancement and improved documentation tasks that can be avoided or at least reduced in frequency through proper systems engineering. So, quality assurance is the review of software products and related documentation for completes correctness and maintainability. Besides these, it also includes assurance that the system meets the specifications and requirements for its intended use and performance.

## 8.2 TESTING

One of the most important level of quality assurance is system testing. Testing is the process of executing a program with the explicit intention of finding errors that is making the program fail. A successful testing is then finding the errors.

There are two strategies for testing software. These are

      1. Code Testing

      2. Specification Testing.

## Code Testing

The code testing strategy examines the logic of the program. To follow this testing method the analyst develops the test cases resulting executing every instruction in the program or modules that is every path through the program is tested. A path is specific combination of conditions that is handled by the program.

## Specification Testing

To perform specification testing, the analyst examines the specification stating what the program is going to do and how it should perform under various conditions. By examining the results, the analyst can determine whether the program performs according to its specified requirements.

## 8.3 LEVEL OF TESTING

Regardless of which strategy the analyst follows, there are preferred practices to ensure that the testing is useful. The levels of tests and types of data, combined with testing libraries are important aspects of the actual test process. Systems are not designed as entire systems nor are they tested as single systems. The analyst must perform.

## UNIT TESTING

Unit testing is testing changes made in an existing or a new program. In the unit testing the analyst test the program making up a system. The software units in a system are the modules and routines that are assembled and integrated to perform a specific function. Unit testing focuses first on the modules, independently of one another, to locate errors. This enables the tester to detect errors in coding and logic that are contained within that module alone. Those resulting from the interaction between modules are initially avoided.

## SYSTEM TESTING

No system design is perfect; several factors like communication gap between user and the system developer, or time constraints create error that must be eliminated before the system is ready for user acceptance and training. Goal achievement of a system can be recognized only after proper testing, rectification of error, if any. A small system error may conceivably explode into a much larger problem with the passage of time. Hence testing is vital to the success of a system.

System testing is executing a program to check logic changes made in it. With the intention of finding errors-making the program fails. Effective testing does not guarantee reliability. Reliability is a design consideration. The purpose of system testing is to identify and correct errors in the system.

System testing does not test software but rather the integration of the modules in the system and its original objectives, current specifications and system documentation. The primary concern is the compatibility of individual modules. The features to be tested and the plan varies from system to system, the first is to see whether it produces the correct output. Following test have been conducted in design and development of the current application system.

## OUTPUT TESTING

The process is testing simultaneously both program (internal processes) and its output. Output produced by the system is compared with the desired output.

Each and every program is tested separately with synthetic data. Actually it is found to be identical. Corrections and modifications were required to be done in some programs.

## VOLUME TESTING

Within this phase of various sizes were entered in the system and run to verify whether the new software function correctly or not. Since seven different data of records with different logic's bulk or volume testing is most important acceptance.

## RECOVERY AND SYSTEM SECURITY TEST

Without the facilities of backup recovery and security, the system is handicapped. Even in some instances it may cause several problems like loss of data. Forced system failures were introduced to test the security of stored data.

There are also facilities to take backup from hard disk to diskette and vice versa. Disk features of the system were tested.

## DOCUMENTATION TESTING

An adequate number of components or messages are provided in each program in order to assist the user in taking awkward situation and for ease of using the system. The user was asked about the understandability and utility of these messages; the response from the user was taken in to consideration to make the system for novice user like them.

## ACCEPTENCE TESTING

Acceptance testing is the last stage of system testing. Live data of the system having different details of information were entered and processed by the actual user of the system. They were sending to the concerned authority and found to be accepted. Then the authority recommended for implementation. Synthetic data were composed by the system developers. During preparation of synthetic data it was in mind that the data should be exhaustive enough to revel all terms and conditions. Live data were taken from different paper files of the existing system of the host firm and inserted by real user.

# 9. DEBUGGING

Debugging occurs as a consequence of successful testing. When a test case uncovers an error, debugging is the process that results in the removal of the error. Although debugging can and should be an orderly process, it is still very much an art. A software engineer, evaluating the results of a test, is often confronted with a "symptomatic" indication of a software problem. That is, the external manifestation of the error and the internal cause of the error may have no obvious relationship to one another. Debug process always begins with the execution of a test case. The results are assessed and a lack of correspondence between the expected and actual outcome is encountered. In many cases, the non-corresponding data are a symptom of an underlying cause that is as yet hidden. The debugging process attempts to match symptom with cause, thereby leading to error correction.

The debugging process will always have one of two outcomes:

➢ The cause will be found corrected and removed, or

➢ The cause will not be found. In the latter case, the person performing debugging may suspect a cause, design a test case to help validate his suspicion, and work toward error correction in an iterative fashion.

In general, three categories for debugging approaches may be proposed:

## 9.1 BRUTE FORCE

The brute force category of debugging is probably the most common and least efficient method for isolating the cause of a software error. This technique is applied when all else fails. Using a "let the computer find the error" philosophy, memory dumps are taken, run-time traces are invoked, and the program is loaded with WRITE/ SYSTEM.OUT/PRINT statements. It is hoped that somewhere in the morass of information

that is produced, we will find a clue that can lead us to the cause of an error. Although the mass of information produced may ultimately lead to success, it more frequently leads to wasted effort and time.

## 9.2 BACKTRACKING

This is a fairly common debugging approach that can be used successfully in small programs. Beginning at the site where a symptom has been uncovered, the source code is traced backward (manually) until the site of the cause is found. Unfortunately, as the number of source lines increases, the number of potential backward paths may become unmanageably large.

## 9.3 CAUSE ELIMINATION

The third approach to debugging - Cause Elimination is manifested by induction or deduction and introduces the concept of binary partitioning. Data related to the error occurrence are organized to isolate potential causes. A "cause hypothesis" is devised and the above data are used to prove or disprove the hypothesis. Alternatively, a list of all possible causes is developed and tests are conducted to eliminate each. If initial tests indicate that a particular cause hypothesis shows promise, the data are refined in an attempt to isolate the bug. Each of the above-mentioned debugging approaches can be supplemented with debugging tools. In all the modules developed, a combination of brute force & cause elimination debugging methodology was used. The unit-testing portion used the former approach wherein either input statement was given to find the error or some piece of code was inserted to trace some values. Functional testing was done using the latter approach. Out of all the debugging exercises, the team members took up debugging at the functional testing stage jointly.

# 10. SYSTEM IMPLEMENTATION AND MAINTENANCE

## 10.1 IMPLEMENTATION

A crucial phase in the system development life cycle is successful implementation of new system design. Implementations simply mean converting new system design into operation. This is the moment of truth the first question that strikes in every one's mind that whether the system will be able to give all the desired results as expected from system. System testing checks the readiness and accuracy of the system to access update and retrieve data from the new files. Once the program becomes available, test data are read into the computer and processed against the files provided for testing. If successful, the program is the run with live data. Otherwise, a diagnostic procedure is used to locate and correct the errors in program.

There are three main types of implementation-

- Implementation of a computer system to replace the manual system.

- Implementation of a new computer system to replace an existing one.

- Implementation of a modified application to replace an existing one.

The term implementation has different meanings, ranging from the conversion of a basic application to a complete replacement of computer system. Implementation is used here to mean the process of converting a new or revised system design into an operational one. Conversion is one aspect of implementation. Since the system is mobile-based system, there is not much procedure involved in implementing the system. After completion of the development of this system and after a thorough testing of the different aspect of the system,

this system will be ready for implementation. The other aspects of system implementation are the post implementation review and software maintenance.

## 10.2 POST IMPLEMENTATION REVIEW

After the system is implemented and system conversion is completed, a review of the system is usually conducted by the user and the analyst to determine whether the system is meeting its desired expectations. It should be a formal process to determine how well the system is working, and how it has been expected.

## 10.3 MAINTENANCE

Software maintenance traditionally denotes the process of modifying a software product after it has been delivered to the customer. Maintenance is inevitable for almost any kind of product.

Most of the software products need maintenance on account of the following three main reasons:

- ✓ **Corrective:** Corrective maintenance of a software product may be necessary either to rectify some bugs observed while the system is in use, or to enhance the performance of the system.
- ✓ **Adaptive:** A software product might need maintenance when the customer need the product to run on new platforms, or when they need the product to interface with new hardware or software.

✓ **Perfective**: A software product needs maintenance to support the new feature that the users want or to change different functionalities of the system according to customer demands.

Software maintenance is an important activity of every organization as the rate of hardware obsolescence is very high. Also the demand of user community to see the existing software product run on newer platforms, run in newer environment with enhanced features.

# 11. SECURITY

Every system must provide built in features for security and integrity of data. Without safeguard against unauthorized access, fraud, embezzlement, fire and natural disaster, a system could be so vulnerable as to threaten the survival of the organization. The end user is always concerned about security along with increased dependence on the computer. In the system development, the developer and the system analyst must consider measures for maintaining data integrity and controlling security at all times.

➢ **Physical Security:** The most costly loss in software is program error. It is possible to eliminate such error through proper testing routines. Parallel runs should be implemented whenever possible. Physical security provides safeguards against the destruction of hardware, databases and documentation; fire, flood, theft, sabotage and eavesdropping; and loss of power through proper backup.

➢ **Database Security:** Two databases are maintained for this project, all the less sensitive data has been stored in MySQL database and for the sensitive data, i.e., the user login credentials, are stored in Firebase's cloud database.

Firebase provides backend as a service. The service provides application developers an API that allows application data to be synchronized across clients and stored on Firebase's cloud. The company provides client libraries that enable integration with Android, iOS, JavaScript, Java, Objective C, Swift and Node.js apps.

Developers using the firebase database can secure their data by using the company's server-side-enforced security rules.

➢ **Application Security:** For app's security we have mainly focused in two measures:

▪ **ProGuard**: To make an application is not good enough, but it also needs to make secure and optimize. It's the basic needs of an application. Security & Optimization play a great role in app ratings, even clients also demands a secure and optimize app. To overcome issues regarding a big size apk and security, ProGuard is the basic and recommended technique to use. It's a great technique to prevent our app from **reverse engineering** (Decompile apk, theft code).

It follows a set of steps: Shrinking, Optimizer, Obfuscator and Preverifier.

o **Shrinking:** It shrink the code to by detecting and removing unused code.

o **Optimizer:** It analyze and optimize the byte code.

o **Obfusator:** It renames the optimize classes, methods with short names. (Used to prevent Reverse Engineering).

o **Preverifier:** It converts the optimize code into optimize jars and libs.

▪ **Firebase and Servlet Backend Authentication**: JSON Web Token (JWT) is being used for web service/api authentication and security. Firebase generates JWTs during the time of user authentication. My job is to retrieve the JWT in the firebase client app, i.e, in the android app, and send it to my backend for verification.

There are two entities which are involved in this whole authentication process. Those are as follows:

• A client android application.

• An authentication server, Google Firebase, in our case, for issuing JWT tokens.

# 12. SCREENSHOTS



Fig: Login Screen
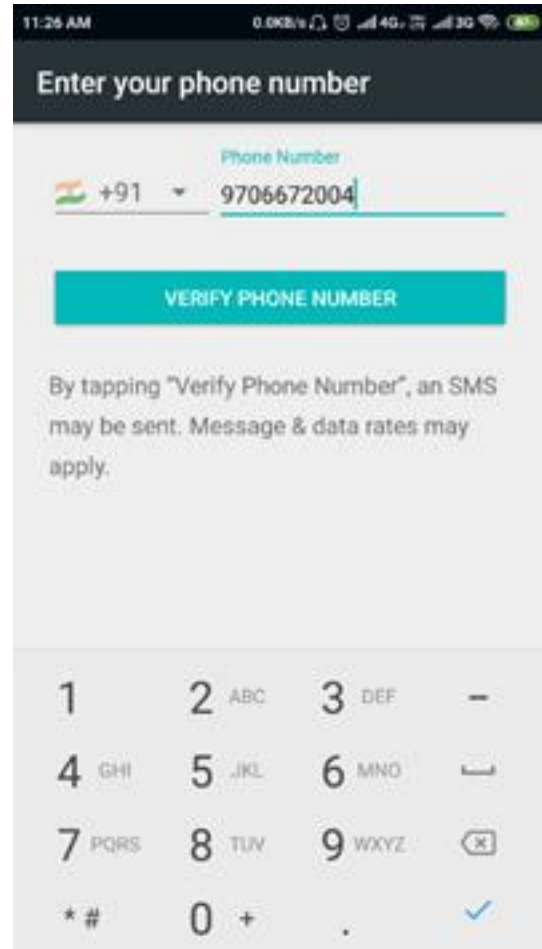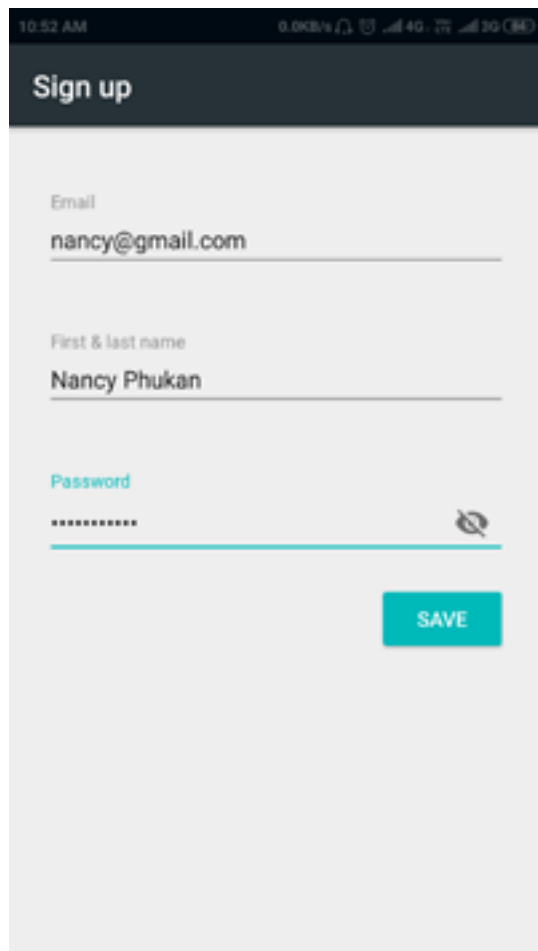


Fig: OTP Login Screen

Fig: Email Login Screen for new user



Fig: Email Login Screen for existing user

Fig: Home Screen



Fig: Navigational Menu

Fig: Fee Assessment Screen

Fig: Fee Assessment (Calculated Amount)

Fig: Deed Registration Screen



Fig: Appointment Confirmed

Fig: Marriage Registration Screen



Fig: Appointment Confirmed

Fig: Deed Payer Details Screen
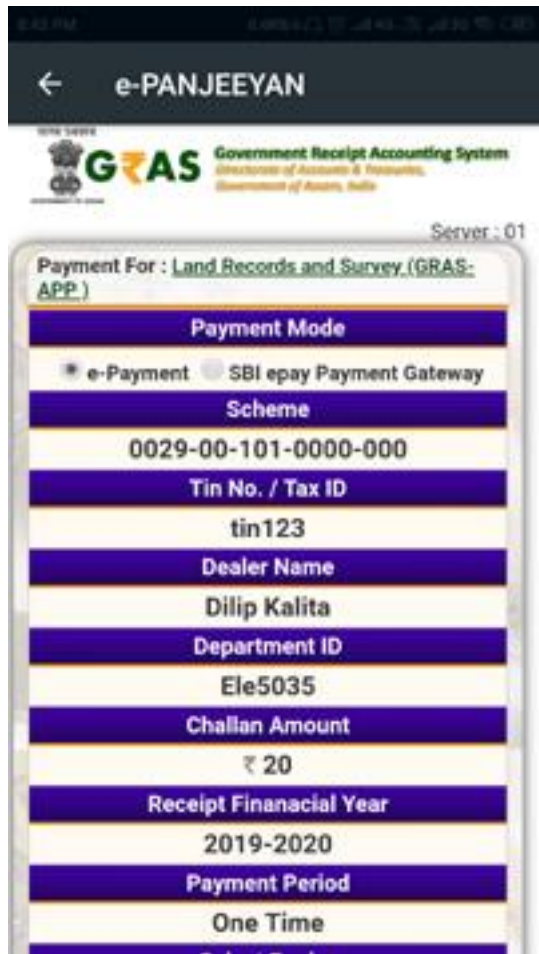


Fig: Marriage Payer Details Screen
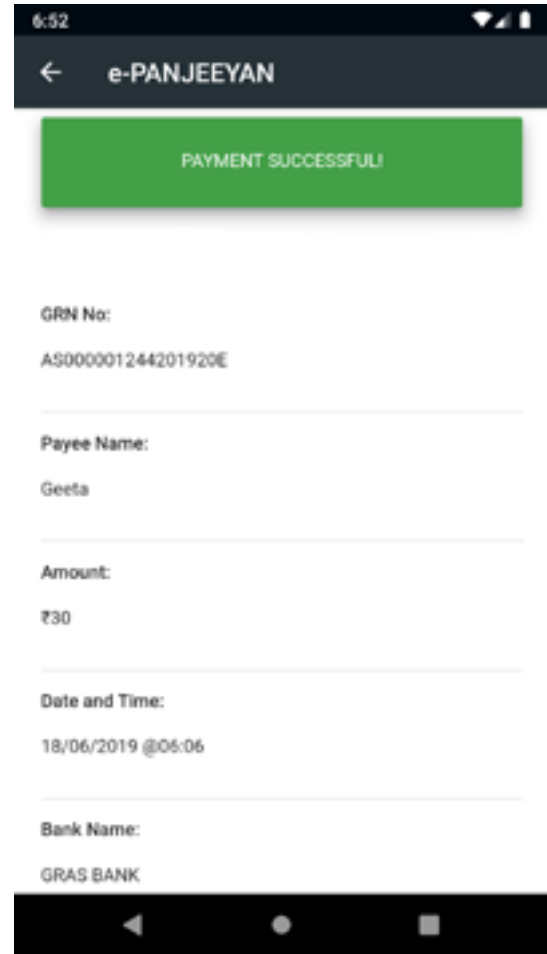
Fig: e-Gras Payment Portal



Fig: Transaction Summary
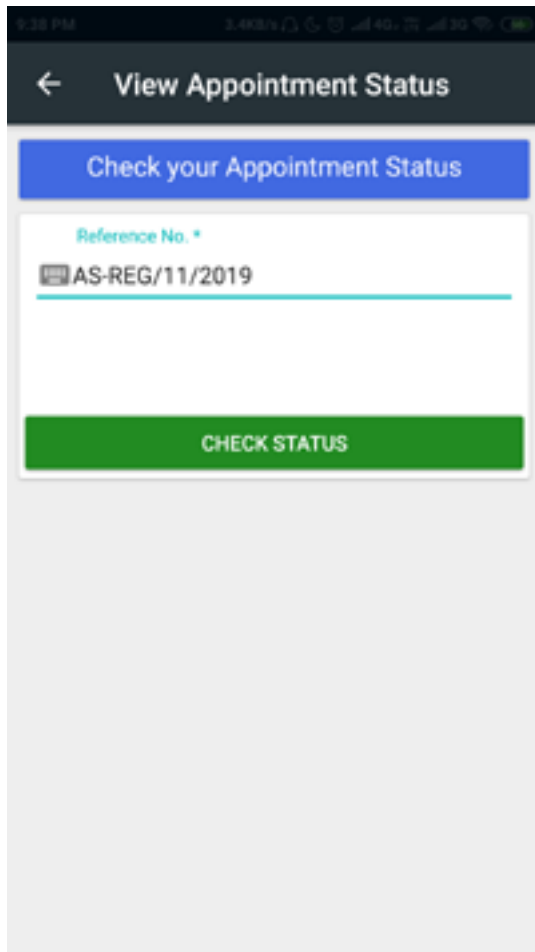
Fig: View Appointment Status
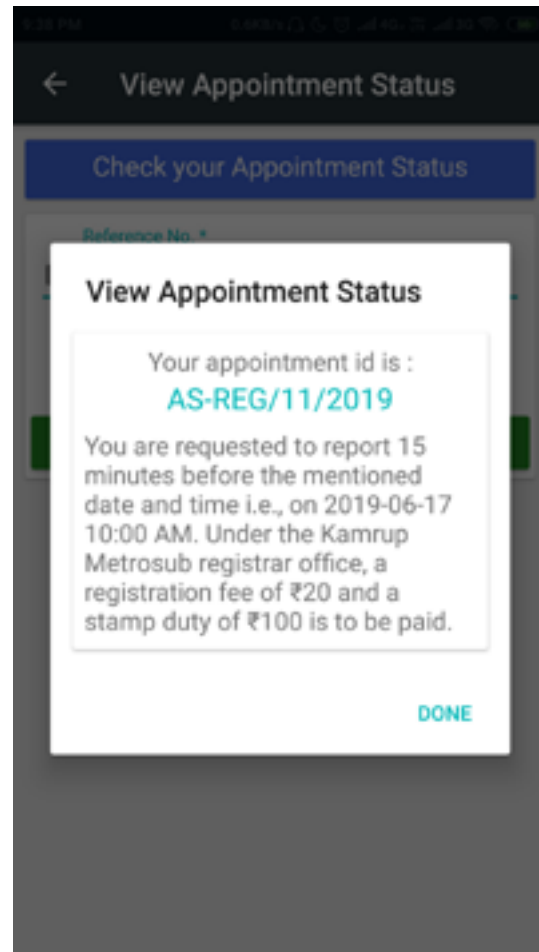


Fig: Appointment Status Screen
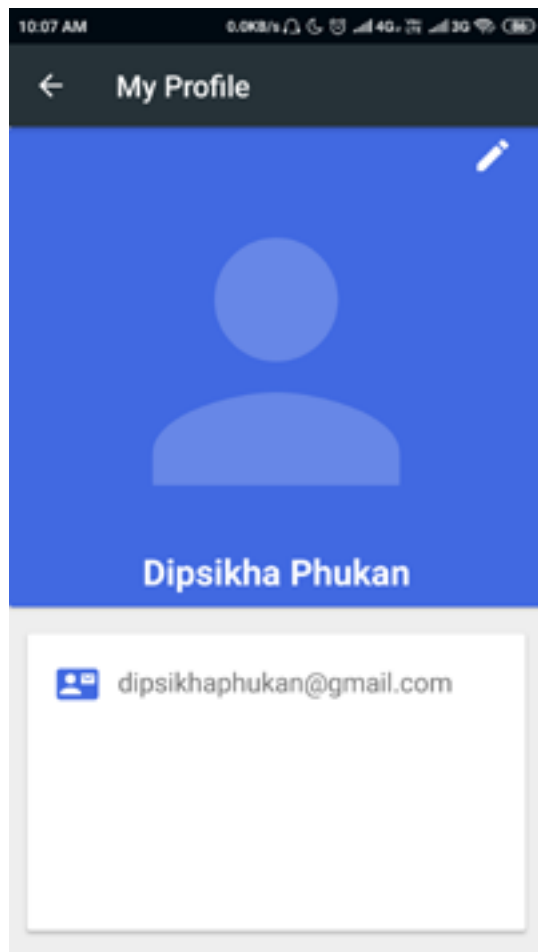
Fig: User Profile Screen
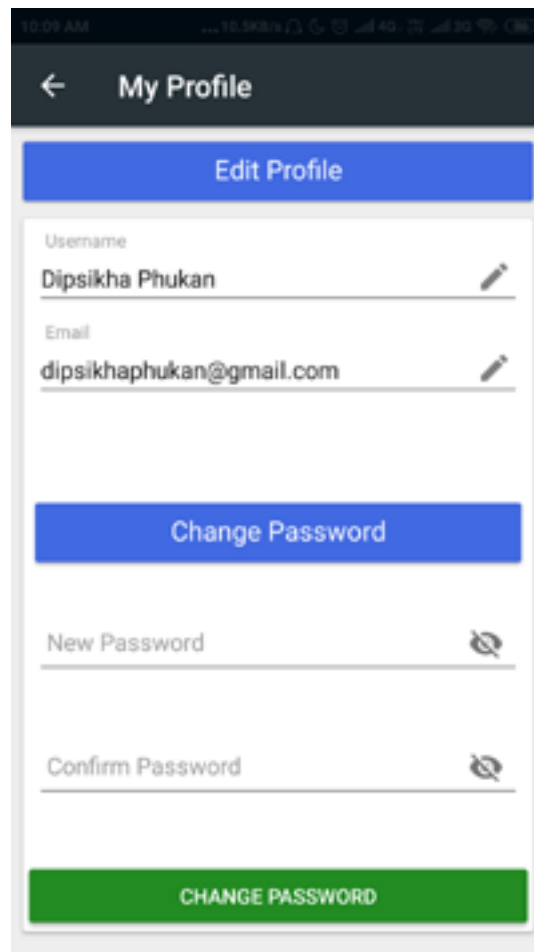
Fig: Edit User Profile Screen
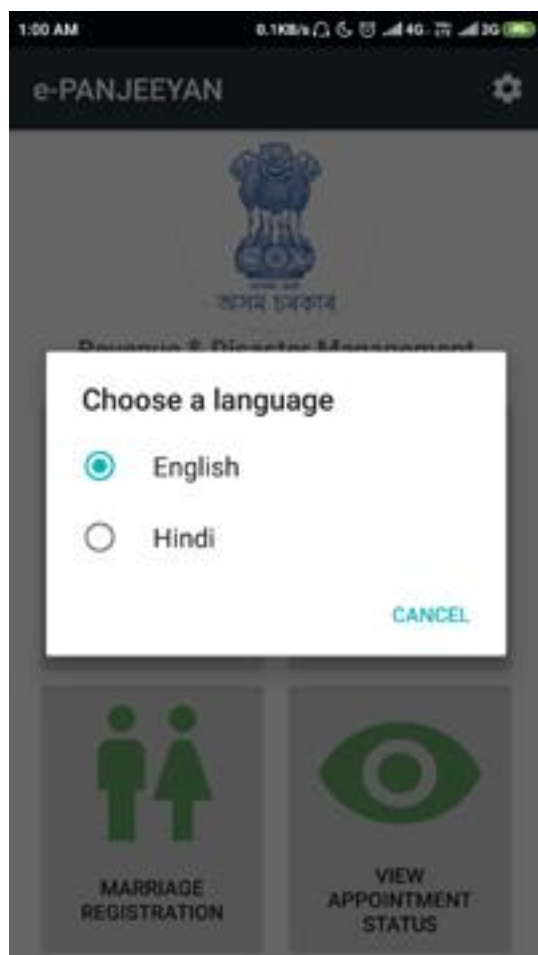
Fig: Language Chooser pop-up menu



Fig: Language Changed to Hindi
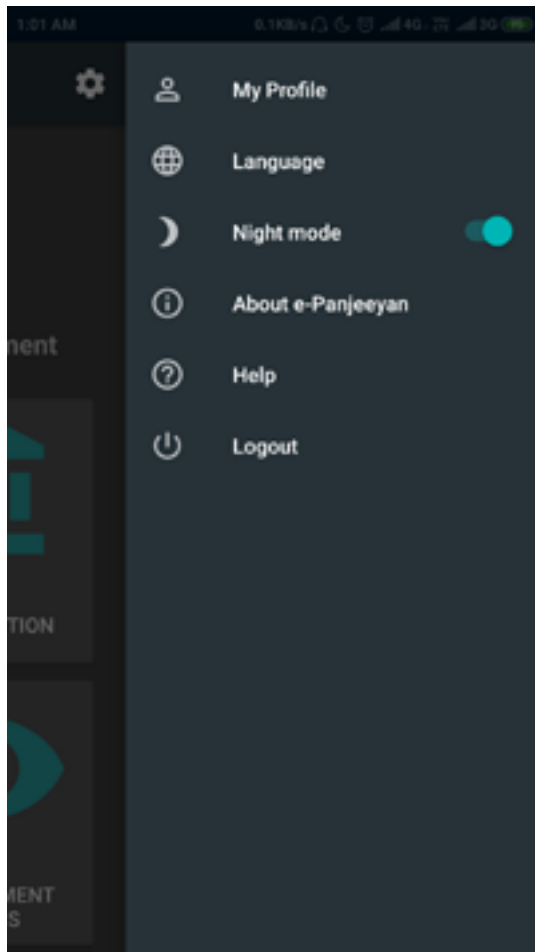
Fig: Night Mode on



Fig: Switched to Night Mode

Fig: About e-Panjeeyan Screen
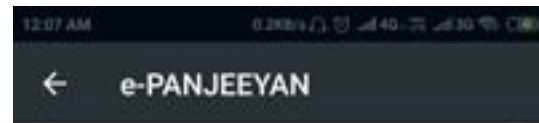
Fig: Help Screen (User Manual)

# 13. CONCLUSION

The project "**e-PANJEEYAN FOR REVENUE & DISASTER MANAGEMENT, GOVT OF ASSAM**" is developed to help the users to make deed and marriage registration online, which also includes the online payment of registration fee without going to the registration counter of the department. Moreover, the user can also calculate the necessary fees associated prior taking an appointment and he/she can even check the appointment status after an appointment is made.

The proposed system was developed by following the requirements given by **Revenue & Disaster Management, Govt of Assam**. Coding was done as per the standards and general coding guidelines were decided so as to have a consistent coding standard across different parts of the proposed system.

The project is still in its growth phase and will be implemented soon as this system is under development, still we are putting our best effort and knowledge to make the system as much error free as possible. Thus the main objective of developing a system that is reliable, efficient and meet all the requirements was carried out.

It can be concluded that developing the System was a real learning experience. The principles of software production were well implemented throughout the system. The project has been as per as the given specification and has been made as user friendly as possible.

As there is always a scope of further development in any software product, and the same is true for this project as well. The end-user will decide its scope in future applications. This will help in developing a better system which will serve them in a better way. This project is flexible enough to incorporate any requirement changes.

# 14. BIBLIOGRAPHY

Books Referred:

1. Fundamentals of Software Engineering - by Rajib Mall
2. PHP MYSQL Vijay Mukhi
3. Servlet & JSP: A Tutorial, Second Edition

Websites referred:

1. www.google.com
2. https://developer.android.com/
3. www.journaldev.com
4. www.stackoverflow.com
5. http://en.wikipedia.org
6. www.tutorialspoint.com
7. https://getbootstrap.com/

Youtube channels referred:

1. PRABEESH R K
2. EDMT Dev

NATIONAL INFORMATICS CENTRE