Police Verification Reporting (PVR) System

Comprehensive Project Report

By

National Informatics Centre, Howrah

Acknowledgement

We would like to express our heartfelt gratitude to National Informatics Centre, Howrah and Dr. Subrata Roy Gupta (Scrum Master) for giving us the opportunity to work under their guidance and help us gain immensely enriching professional experience. Our sincere thanks to Scrum Masters, Mr. Sudipta De for giving us valuable inputs and ideas right from the allocation of project till its successful completion.

The successful completion of our project would not have been possible without the dedicated support from all our mentors and friends.

Srijon Mallick
Rupak Pal
Jit Saha
Pritam Roy
Soumyadeb Dhar
Indranil Sen
Souvik Saha
Suprity Paul
Anik Das Mondal

Project Certificate

Contents

1. Introduction

1.1. Problem Statement:

Develop an e-governance web portal that provides a platform for online processing of police verification applications received from various organizations under the government sector and government undertaking sectors. The platform should have a dashboard to facilitate new applications submission, send an application for verification, verify applications, search applications, generate verification letters, and generate reports based on the applications received at the district level.

1.2. Problem Solution:

Software as a service (SaaS) is a model of software deployment where an application is hosted as a service provided to customers across the Internet. By eliminating the need to install and run the application on the customer's own computer, SaaS alleviates the customer's burden of software maintenance, ongoing operation, and support.

Our solution, Police Verification Report System is a web portal based on the principle of SaaS. PVR Portal will provide a dashboard-type platform for processing police verification applications received from various government sectors to get the candidates verified by the appropriate authority.

1.3. Assumptions

SaaS is based upon the assumption that the services provided are commonplace and well defined, hence economies of scale and balancing of supply and demand becomes possible. This assumption holds true for those areas of IT that are ubiquitous, a cost of doing business and commodity-like. SaaS is therefore not suitable for innovative or highly specialized niche systems, though SaaS may be used to provide one or more components in such systems.

About PVR System:

1. The PC is connected to the internet/ connected to the intranet.
2. A Web Browser.

1.4. Constraints

Minimum number of computers is at least 2 which will limit the development process if they are not available.

1.5. Modules

- Login
- Dashboard
- Application
    - New Application

    - Add Office
    - Check Status
    - Update Application
- Letter
    - Verified letter to employer
    - Unverified letter to employer
- Report
    - Monitoring Report
    - Daily Application Report
- Activity Log
- Notification
- Profile
- Super Admin Panel

1.6. System Features

- Anywhere any device is accessible.
- Multimodal delivery of project status - mobile app, mobile-based website & desktop-based access.
- Dashboards show updates regarding submitted, pending and verified applications.
- Submitted applications can be tracked through dedicated check status operation.
- Alerts and Notifications will be shown at dashboards when new applications are processed.
- Existing scheme's data that are available online will be pre-fetched by this portal to avoid duplication of efforts.
- Strict adherence to NIC recommended SDLC - software development life cycle and quality standards.

1.7. System Benefits

- Dashboard based monitoring.
- Accuracy of reported data.
- One-click report generation.
- One-click letter generation.
- Ease of access & use.
- Reduced duplication of efforts.

2. Project Plan

2.1. Task Sheet

Scrum Masters:
Mr. Sudipta De (ADIO cum Scientist-C)
Dr. Subrata Roy Gupta (DIO cum Scientist-D)

Number of Sprints: 4

Sprint 1:

Name of the Developer	Tasks
Srijon Mallick	● Planning and work distributing ● Database Schema Preparation ● Attribute Selection ● ER Diagram ● Use case Diagram ● PVR System Study ● Swimlane Diagram
Souvik Saha	● Use case Diagram ● Attribute Selection ● PVR System Study ● Swimlane Diagram

Sprint 2:

Name of the Developer	Tasks
Srijon Mallick	● Database Design ● Backend Coding and Connecting ● Dashboard Design ● Login Design ● Application Form Design
Rupak Pal	● Dashboard Design ● Login Design ● Application Form Design
Jit Saha	● Web layout design ● CSS styling of the entire system ● Downloadable PDF Design
Soumyadeb Dhar	● Data fetching in printable PDF ● Dashboard Design
Indranil Sen	● DOM PDF Generation ● Dashboard Design

Sprint 3:

Name of the Developer	**Tasks**
Srijon Mallick	● Database Administrator ● Full DB redesign with corrected attributes names and checks ● Redesign of data dictionaries ● Master data entries ● Notifications
Rupak Pal	● Activity Log Design ● Profile Design ● Issues lodged Design
Jit Saha	● Mobile view Design ● Bug fixes
Soumyadeb Dhar	● Captcha Integration and validation ● Form validations ● Contact us
Indranil Sen	● All page view redesigns ● Contact us

Sprint 4:

Name of the Developer	**Tasks**
Srijon Mallick	● Super Admin creation and integration ● Forgot password ● Redundant code removing ● Project Report Documentation
Pritam Roy	● API Integration ● OWASP ZAP Testing ● OWASP ZAP Report ● Project Report Documentation
Rupak Pal	● API Integration ● Project Report Documentation ● Test cases and user cases testing and reporting
Jit Saha, Soumyadeb Dhar, Indranil Sen	● Project Data Documentation ● ER Diagram
Suprity Paul, Anik Das Mondal	● User manual documentation

2.2. Project Management Roles and Responsibilities

Roles	Responsibilities	Participants
Project Owner	● Ultimate decision-maker and tiebreaker. ● Provide feedback with requirements if any in Scrum meetings.	District Administration
Developers	● Sets user stories, test cases, draft cases, assessment, time period and artifacts for respective sprints. ● Update project according to the user stories. ● Complete backlogs, if any. ● Provide Project Reports to the Scrum Master and Project Owner in Scrum Meetings. ● Update Project Plan according to the requirements. ● Start the next sprint with the feedback and iterate.	● Srijon Mallick ● Rupak Pal ● Jit Saha ● Pritam Roy ● Soumyadeb Dhar ● Indranil Sen ● Souvik Saha
Scrum Master / Project Coordinator	● Periodically assess each and every development in successive sprints. ● Direct link between developers and Project Owner. ● Moves to the next sprint with previous backlogs and current feedback.	● Dr. Subrata Roy Gupta (DIO cum Scientist-D) ● Mr. Sudipta De (ADIO cum Scientist-C)
Project Guide	● Provide project oversight and guidance. ● Review/approve project elements. ● Getting the Team Off on the Right Foot.	● Mr. Parag Dhali ● Mr. Rakesh Koley

Table 1 Project Roles and Responsibilities

3. Requirement Analysis

3.1. Introduction

3.1.1. Purpose

The purpose of the PVR portal is for online processing of police verification applications received from various organizations under the government sector and government undertaking sectors. VR section under District administration will be verifying candidates for jobs in various government offices. PVR portal will help to achieve this purpose by passing on the application digitally between the departments and administration to verify candidates. Our proposed system has dedicated modules like new application for submitting new candidate details, check status for checking application state and update application to approve or reject applications by appropriate authority. Also, different letter generation modules are there to generate government letters for verified/unverified candidates which will help get rid of pen and paper method as this might create redundancy issues. Lastly to monitor all the activities like reports can be generated based on the queried date range. Notifications and activity logs have been made for better interactivity. The system is so designed that each user gets to see information which is meant for him/her only, thus providing a better user management system.

3.1.2. Intended Audience and Reading Suggestions

The intended audience of this document will be the developers of PVR, project managers, user testers, and documentation writers. It is recommended that the SRS document be read sequentially.

3.1.3. Project Scope

PVR which is a SaaS aims to be a online application processing tool and dashboard for police verification applications received from various organizations under the government sector and government undertaking sectors. Users can see real-time number of applications, their state, details, etc. It will provide a user-friendly dashboard that will serve all user needs.

3.2. Overall Description

3.2.1. Product Perspective

The public sector faces other challenges when it comes to implementing ERP solutions. On top of budget restrictions and increased service-level demands, many public sector agencies require highly specialized functionality and controls. Data collection and maintenance in a government administration are crucial and prone to redundancy.

With PVR, we hope to resolve these issues.

3.2.2. Product Features

- Application installation is not required; users access the application on the web.
- Updates and upgrades to newer versions are automatic.
- Any computer with an Internet connection can become an access point to an application, no matter what operating system is installed.
- The risk of viral infection is greatly decreased when running an application on the web instead of an executable.

3.2.3. Users

There are fundamentally 4 dedicated users in our system. Each user has different abstract views of the system. From VR section, a Data Entry Operator will be a user responsible for new application submission. Officer-In-charge of VR Section (OC-VR) will be next higher-level user who will approve the applications for further proceeding in IB department. ADM will be next higher level of user who will confirm all the verified Defence category applications received from IB department. Super admin is the user with highest privileges responsible for new user addition, audit log, check issues, monitor user activities etc.

User Type	**Dashboard**	**Profile**	**New Application**	**Check status**	**Add Office**	**Super Admin Panel**
Super Admin	Y	Y	N	Y	Y	Y
Additional District Magistrate	Y	Y	N	Y	Y	N
Officer-In-Charge (VR Section)	Y	Y	N	Y	Y	N
Data Entry Operator (VR Section)	Y	Y	Y	Y	Y	N

Table 2 User Types and Their Access hierarchy

User Type	**Update Application**	**Monitoring Report**	**Daily Application Report**	**Verified Letter**	**Unverified Letter**	**Issues Lodged**

Super Admin	N	Y	Y	N	N	Y
Additional District Magistrate	N	Y	Y	Y	N	Y
Officer-In-Charge (VR Section)	N	Y	Y	Y	Y	N
Data Entry Operator (VR Section)	Y	Y	Y	N	N	N

Table 3 User Types and Their Access hierarchy (Contd.)

3.2.4. Operating Environment

PVR will operate on a web browser. It is cross-browser (can be run on most web browsers including Internet Explorer, Mozilla, etc.) and cross-platform (windows vista/XP, Linux) software.

3.2.5. Design and Implementation Constraints

The minimum number of computers is at least 2 which will limit the development process if they are not available.

3.2.6. User Documentation

- User Manuals
- Data Documentation (consisting ERDs and Data Dictionary)

3.2.7. Assumptions and Dependencies

Assumptions:

- Latest Web browsers like Google Chrome, Mozilla Firefox, Microsoft Edge installed in the browsing device
- Internet Connectivity
- PDF Viewer installed for viewing generated reports.

3.3. System Features

3.3.1. Mobility

Because of the SaaS architecture, the software does not require itself to be locally installed in the user's device. Users can access the portal anytime anywhere with the basic requirement of a web browser and a stable internet connection.

3.3.2. Responsive

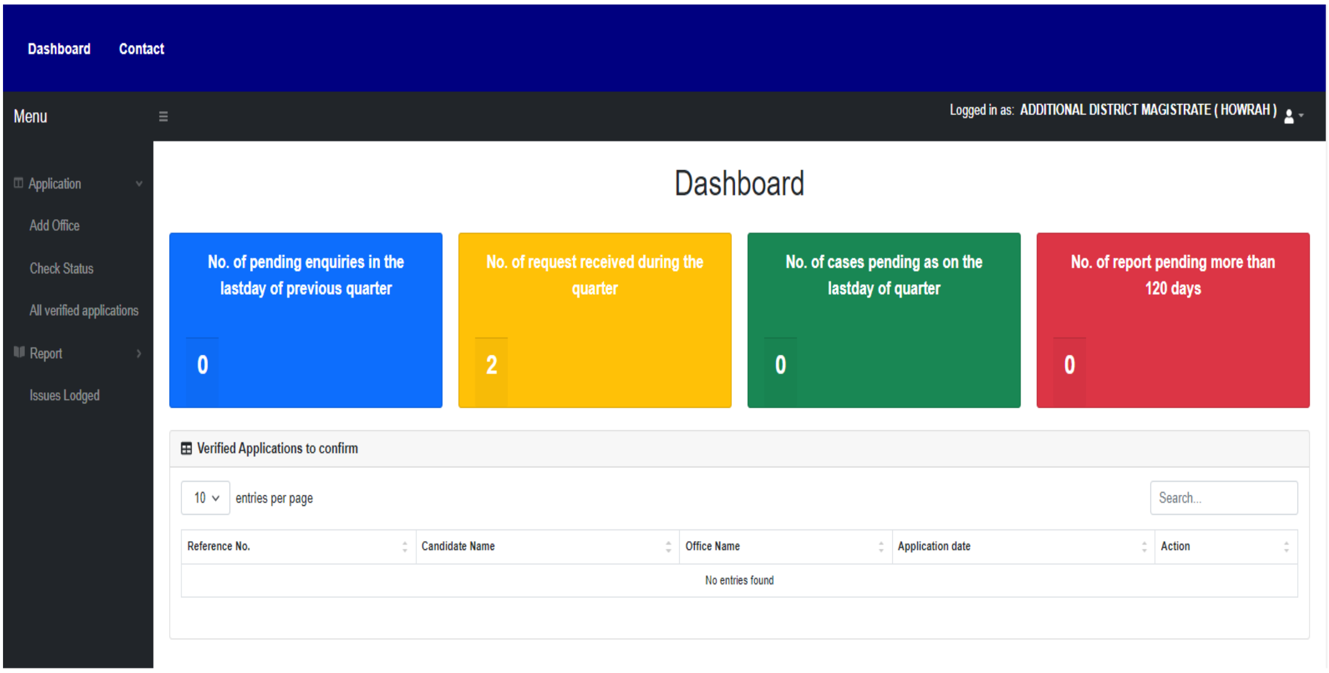
Responsive Web design is the approach that suggests that design and development should respond to the user’s behavior and environment based on screen size, platform, and orientation. In other words, the website should have the technology to automatically *respond* to the user’s preferences. This would eliminate the need for a different design and development phase for each new gadget on the market. Thus, we have made our system responsive which makes it easy to view in portable devices.

3.4. External Interface Requirements

3.4.1. User Interface

User interface (UI) design is the process designers use to build interfaces in software or computerized devices, focusing on looks or style. Designers aim to create interfaces which users find easy to use and pleasurable. UI design refers to graphical user interfaces and other forms—e.g., voice-controlled interfaces.

A dashboard is provided to the user for summary progresses. All the actions in the portal are point and click based.



Logged in as: ADDITIONAL DISTRICT MAGISTRATE (HOWRAH)

Dashboard

No. of pending enquiries in the lastday of previous quarter 0

No. of request received during the quarter 2

No. of cases pending as on the lastday of quarter 0

No. of report pending more than 120 days 0

Verified Applications to confirm

10 entries per page

Reference No.	Candidate Name	Office Name	Application date	Action
No entries found				

Fig 1 Dashboard

3.4.2. Hardware Interfaces

TCP/IP: It is a reliable connection-oriented protocol that allows byte streaming originating on one machine to be delivered without any error on any other machine in the network. It fragments the incoming byte stream into discrete messages and parses each one onto the internet layer. At the destination, the receiving TCP process reassembles received messages into the output stream. TCP also handles the flow control to make sure a fast sender cannot swamp a slow receiver with more messages than it can handle.

3.4.3. Communication Interfaces

HTTP: Hypertext transfer protocol (HTTP) is a method used to transfer or convey information on the World Wide Web. Its original purpose was to provide a way to publish and retrieve HTML pages. An HTTP client initiates a request by establishing Transmission Control Protocol (TCP) connection to a particular port on a remote host. An HTTP server listening on that port waits for the client to send a request message. Upon receiving the request, the server sends back a status line, such as “HTTP/1.1 200 OK” and a message of its own, the body of which is perhaps the requested file, an error message, or some other information. Resources to be accessed by HTTP are identified using Uniform Resource Identifiers (URIs) (or, more specifically, URLs). Using the HTTP: or HTTPS URI schemes.

SMTP: SMTP is an application layer protocol. The client who wants to send the mail opens a TCP connection to the SMTP server and then sends the mail across the connection. The SMTP server is always-on listening mode. As soon as it listens for a TCP connection from any client, the SMTP process initiates a connection on that port (25). After successfully establishing the TCP connection the client process sends the mail instantly.

3.5. Other Non Functional Requirements

3.5.1. Performance Requirements

The web server must be able to handle and support multiple instances of application. The time between request and reply should be less in case of online help. Minimum time should be taken by the application to display preview the web pages developed by the user. In case of power failure, the data should be stored in the state that was last saved by the user

3.5.2. Safety Requirements

- A user can be able to view/update only his web pages.
- CSRF tokens must be implemented to prevent unauthenticated transactions

3.5.3. Software Quality Attributes

There are various software quality attributes that are taken into consideration

- **Availability** – As PVR is a web-based service provided to the users, it will be available as long as the server is up.
- **Interoperability** – PVR is interoperable on various operating systems, thus increasing the applications usability and flexibility.
- **Maintainability** – PVR is a SaaS based service. Hence, all the updates are centralized making the maintenance convenient.
- **Usability** – Special care has been taken to make the system user friendly with features like dashboard and GUI.

3.6. Software Development Methodology

3.6.1. Agile Methodology

AGILE methodology is a practice that promotes continuous iteration of development and testing throughout the software development lifecycle of the project. Both development and testing activities are concurrent, unlike the Waterfall model.

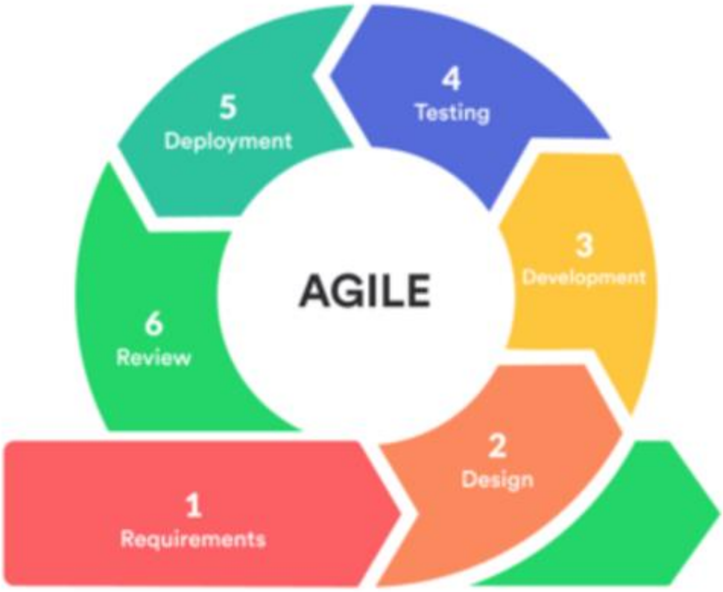


Fig: Agile Model

Agile software development emphasizes four core values:

- Individual and team interactions over processes and tools
- Working software over comprehensive documentation
- Customer collaboration over contract negotiation
- Responding to change over following a plan

Agile Methodologies Frameworks: Today, the word Agile can refer to these values as well as the frameworks for implementing them, including Scrum, Kanban, Extreme Programming (XP), and Adaptive Project Framework (APF). Simply put, it is a process for managing a project characterized by constant iteration and collaboration in order to more fully answer a customer's needs.

Agile metrics:

Metrics that can be collected for effective usage of Agile is:

- Drag Factor
  - Effort in hours that do not contribute to the sprint goal
  - Drag factor can be ipvroved by reducing the number of shared resources, reducing the amount of non-contributing work
  - New estimates can be increased by a percentage of drag factor -New estimate = (Old estimate + drag factor)

- Velocity
    - Amount of backlog (user stories) converted to shippable functionality of sprint
- No of the Unit Tests added
- Time interval taken to complete daily build
- Bugs detected in an iteration or in previous iterations
- Production defect leakage

3.7. Technologies Used

3.7.1. CodeIgniter

CodeIgniter is a powerful PHP framework with a very small footprint, built for developers who need a simple and elegant toolkit to create full-featured web applications. CodeIgniter was created by EllisLab and is now a project of the British Columbia Institute of Technology.

CodeIgniter is an application development framework, which can be used to develop websites, using PHP. It is an Open-Source framework. It has a very rich set of functionalities, which will increase the speed of website development work.

If you know PHP well, CodeIgniter will make your task easier and saves a lot of time if you are developing a website from scratch. It has a very rich set of libraries and helpers. A website built in CodeIgniter is very secure, as it has the ability to prevent various attacks that take place through websites.

Feature:

- Model-View-Controller Based System
- Extremely Light Weight
- Full-Featured database classes with support for several platforms.
- Query Builder Database Support
- Form and Data Validation
- Security and XSS Filtering
- Session Management
- Email Sending Class. Supports Attachments, HTML/Text email, multiple protocols (Sendmail, SMTP, and Mail), and more.
- Image Manipulation Library (cropping, resizing, rotating, etc.). Supports GD, ImageMagick, and NetPBM
- File Uploading Class
- FTP Class
- Localization
- Pagination
- Data Encryption
- Benchmarking
- Full Page Caching
- Error Logging
- Application Profiling
- Calendaring Class
- User Agent Class
- Zip Encoding Class

- Template Engine Class
- Trackback Class
- XML-RPC Library
- Unit Testing Class
- Search-engine Friendly URLs
- Flexible URI Routing
- Support for Hooks and Class Extensions
- Large library of “helper” functions

CodeIgniter Usage:

Sl. No.	Method	Sub-methods used	Summary
1.	Helpers	● Array helper ● Captcha helper ● Cookie helper ● Date helper ● File helper ● Form helper ● URL helper ● Security helper	Each helper file is simply a collection of functions in a particular category. They are not loaded by default. Once loaded they become globally available. They are loaded when needed to reduce complexity.
2.	Libraries	● Email class ● Java Script class ● Session Library ● Encryption Library	Libraries are the classes located in libraries directory. They can be created or extended.
3.	Class	● Query Builder Class ● Pagination Class ● Loader class ● HTML table class ● File Uploading Class ● FTP Class	Classes should have a basic prototype for codeigniter where class names and file names must match and class declarations must be capitalized.
4.	Security	● Post method ● CSRF Token ● XSS protection ● Cross-Origin Resource Sharing	Codeigniter’s own security methods can be configured either by adding headers or changing in the config file.
5.	MVC	● Model ● View ● Controller	● Model includes all the data and its related logic ● View presents data to the user or handles user interaction ● Controller is an interface between model and view
6.	Miscellaneous	● Database caching ● Cookie ● Flexible URI routing ● Data encryption(SHA 256)	● Caching and cookies for better user experience ● Dynamic URL ● Password encryption with salting ● Messaging system via email ● Arranging the dependencies

		● SMTP Configuration ● Composer(dependencies) ● Time reference	● Variable time zone

Table 4 CodeIgniter Usage Table

3.7.2. PostgreSQL

PostgreSQL is an **ORDBMS** [**Open-Source Object-Relational Database Management System**]. It is used to store data securely; support best practices, and allow recovering them when the request is processed.

- PostgreSQL is initially introduced on 8th July 1996 at the University of California.
- It is the first DBMS, which performs MVCC [Multi-Version Concurrency Control] feature, even before Oracle. The multi-version concurrency control attribute is known as snapshot isolation in Oracle.
- It is written in C programming language.
- PostgreSQL is cross-platform and runs on various operating systems such as Microsoft Windows, UNIX, FreeBSD, Mac OS X, Solaris, HP-UX, LINUX, and so on.
- PostgreSQL is the existing database for the macOS server.
- PostgreSQL is also pronounced as Post-gress-Q-L, which is developed by the PostgreSQL Global Development Group (a worldwide team of volunteers), any organization or other private entity does not control it.
- PostgreSQL will offer us the facility to add custom functions with the help of various programming languages such as Java, C, and C++, etc.
- In this, we can describe our functional languages, index types and data types, and we can also create a custom plugin to increase the reliability of our needs.
- Its source code is accessible under PostgreSQL license; thus, we can use it freely, change and allocate PostgreSQL in any form.
- PostgreSQL follows the transaction along with the ACID (Atomicity, Consistency, Isolation, and Durability) properties.
- The primary objective of PostgreSQL is to handle a variety of jobs from single technologies to web service or the data warehouse with several parallel users.

3.7.3. OWASP ZAP

The OWASP Zed Attack Proxy (ZAP) is one of the world's most popular web application security testing tools. It is made available for free as an open-source project and is contributed to and maintained by OWASP. The Open Web Application Security Project (OWASP) is a vendor-neutral, non-profit group of volunteers dedicated to making web applications more secure. The OWASP ZAP tool can be used during web application development by web developers or by experienced security experts during penetration tests to assess web applications for vulnerabilities.

The OWASP Zed Attack Proxy is a Java-based tool that comes with an intuitive graphical interface, allowing web application security testers to perform fuzzing, scripting, *spidering,* and *proxying* in order to attack web apps. Being a Java tool means that it can be made to run on most operating systems that support Java. ZAP can be found by default within the Kali Linux Penetration Testing Operating System, or it can be downloaded and run-on OSs that have Java installed. The OWASP ZAP proxy borrows heavily in GUI appearance from the *Paros Proxy Lightweight* Web Application security testing tool. Kindly see this article for a detailed look at the Paros Proxy tool.

3.8. Libraries Used

Sl. No.	Name of the library	Link/URL/Description	Summary
1.	Admin LTE	https://adminlte.io/	Design template for our project
2.	Bootstrap	https://getbootstrap.com/docs/4.4/getting-started/introduction/	Responsive UI with adaptive screen size
3.	Fontawesome icon	https://fontawesome.com/	Icon pack
4.	JQuery	https://jquery.com/	Real-time dynamic functionality
5.	PHP	Functional Language	Functionality, query
6.	SHA 256	https://www.npmjs.com/package/js-sha256	Used for hashing passwords or any other sensitive data
7.	HTML	Markup Language	Design
8.	CSS	Styling Sheets	Styling
9.	PDF (mime)	DOM PDF	Generate PDF

Table 5 Libraries Used

4. Design Document

4.1. Introduction

A software design description (a.k.a. software design document or SDD; just design document; also Software Design Specification) is a written description of a software product, that a software designer writes in order to give a software development team overall guidance to the architecture of the software project. An SDD usually accompanies an architecture diagram with pointers to detailed feature specifications of smaller pieces of the design. Practically, the description is required to coordinate a large team under a single vision, needs to be a stable reference, and outline all parts of the software and how they will work.

4.2. Process Flow Diagram

Process Flowchart or PFD is used to show relations between major parts of the system. Process Flowcharts for single unit or multiple units differ in their structure and implementation. Process flow diagram (PFD) illustrates the arrangement of the equipment and accessories required to carry out the specific process; the stream connections; stream flow rates and compositions; and the operating conditions. The PFD is a diagrammatic representation of the process, which is normally drawn in a stylized pictorial form using international standards symbols.

The main reason of using Process Flowchart or PFD is to show relations between major parts of the system. Process Flowcharts are used in process engineering and chemical industry where there is a requirement of depicting relationships between major components only and not include minor parts.

Why to make a process flow diagram

- More easily communicate what the user flow should be.
- Make sure we don't miss important steps in the process.
- Make decisions and changes at a low fidelity before investing a lot of time in design.
- Get feedback from different stakeholders.

What is user flow?

User flow: This visual includes all the various interactions users should have on your site. Your product and UX teams should determine how they want users to navigate your site, taking into account the users' needs.

User journey: User flow is one part of a user journey. The user journey looks at the entire environment of a purchase, from offline influences to online ads. The user flow is only concerned with your visitors once they reach your site or application. Once they've left your site, your visitors continue their user journey.

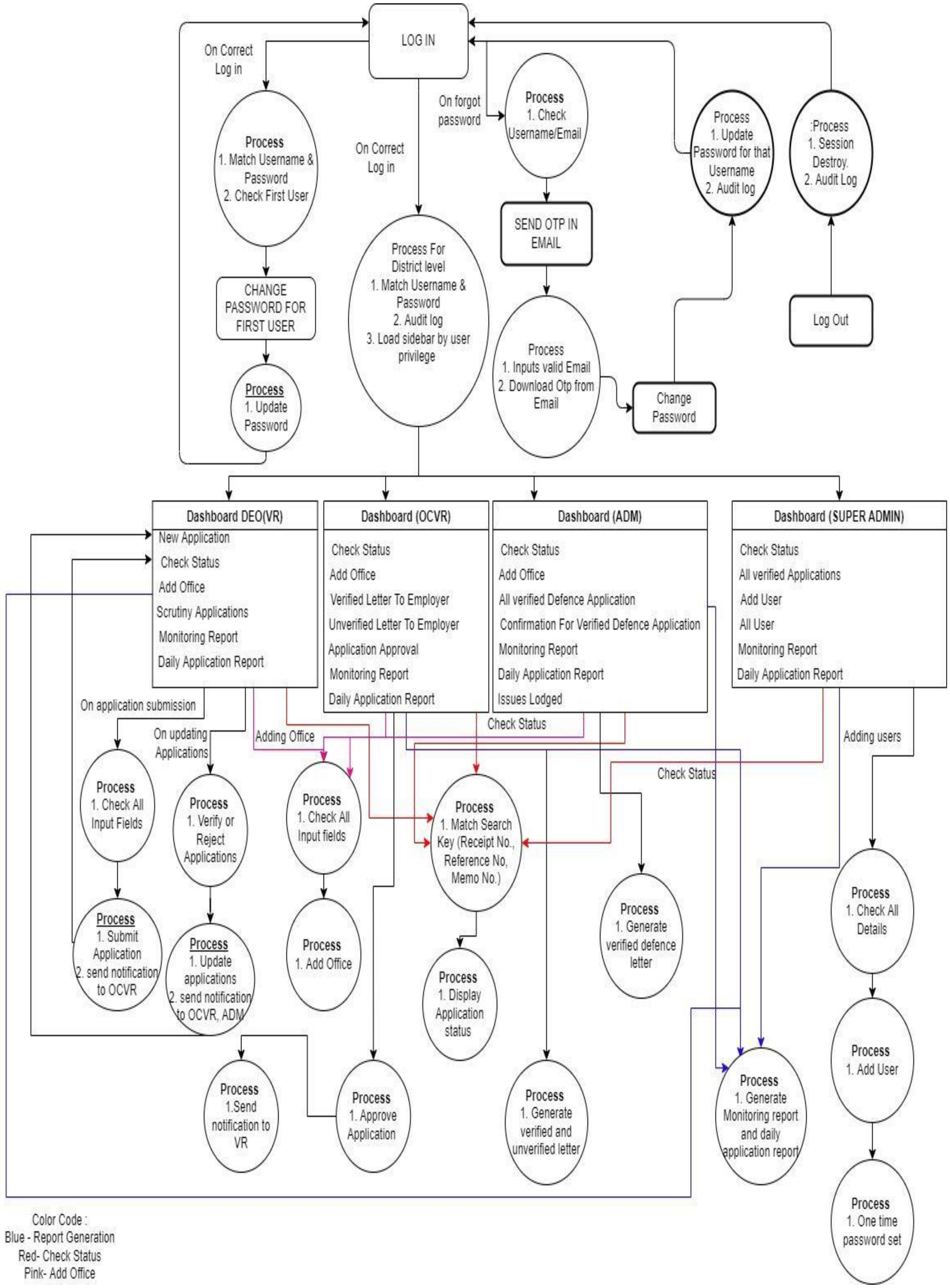


Fig 2 Process Flow Diagram

4.3. Database Design

Database Design is a collection of processes that facilitate the designing, development, implementation and maintenance of enterprise data management systems. Properly designed database are easy to maintain, ipvroves data consistency and are cost effective in terms of disk storage space. The database designer decides how the data elements correlate and what data must be stored.

The main objectives of database designing are to produce logical and physical designs models of the proposed database system.

The logical model concentrates on the data requirements and the data to be stored independent of physical considerations. It does not concern itself with how the data will be stored or where it will be stored physically.

The physical data design model involves translating the logical design of the database onto physical media using hardware resources and software systems such as database management systems (DBMS).

4.3.1. Master Tables

Master data tables are the core data that is used as a base for any transaction. If you are producing, transferring stock, selling, purchasing, doing physical inventory, whatever your activity may be, it requires certain master data to be maintained.

The master tables used in our system are:

- public.“pvr_master_caste"
- public."pvr_master_defence"
- public."pvr_master_department"
- public."pvr_master_designation"
- public."pvr_master_district"
- public."pvr_master_employee_type"
- public."pvr_master_gender"
- public."pvr_master_notification"
- public."pvr_master_office"
- public."pvr_master_pincode"
- public."pvr_master_policestation"
- public."pvr_master_sent_to"
- public."pvr_master_state"

4.3.2. Transactional Tables

A transaction is an activity performed by entities(master tables) within the system. These activities are captured in transaction tables and usually, these transaction entries have foreign keys to master records.

Transaction tables are designed to store events in the system. These events are associated with master records to ensure normalization because the transactions can quickly grow in large numbers. The analytics tools, OLTP, partitioning are applied on transaction tables. Most of the querying is done on transaction tables.

Every scheme table has id_pk has as its primary key and login_id_fk as the foreign key referencing public. ”pvr_semitrans_login”. The field is "location_code” is not included in public. ”pvr_master_table” to ease data entry operations in the development end. Theses tables are transactional in nature. Following are the scheme tables :

- pvr_audit_log
- pvr_candidate_details
- pvr_candidate_profile
- pvr_employer
- pvr_employer_profile
- pvr_final_status
- pvr_issues
- pvr_login
- pvr_memo
- pyr_privilege
- pvr_receipt_no
- pvr_reference_no
- pvr_report
- pvr_trans_notification
- pvr_user_privilege
- pvr_user_profile
- pvr_user_type
- pvr_vr_detail
- pvr_with

4.3.3. Constraints Information

All the Fields are implicitly defined with **Not NULL** constraint. The fields which are nullable are implicitly mentioned. All the tables defined and designed in such a way that there is no provision for null values.

All other constraints like Referential Integrity Constraints (Foreign Keys), Primary Keys, Check and Unique constraint have been described explicitly for each table.

4.3.4. Indexing

All the Primary Keys and Unique keys in PostgreSQL uses B-Trees as an Indexing Strategy by default. However, to ease transactions in scheme tables, we have introduced indexing in the scheme tables based on 3 columns: session, month and location_code since insertion and updation is done by checking these columns. Indexes are named as idx_<tablename>.

4.3.5. Stored Procedures

Stored Procedures are used to automate database processes. A stored procedure "notification_status()", is incorporated in the public."pvr_trans_notification" to deactivate the records in the notification table after 2 days of entry.

4.3.6. ER Diagram

Entity Relational (ER) Model is a high-level conceptual data model diagram. ER modeling helps you to analyze data requirements systematically to produce a well-designed database. The Entity-Relation model represents real-world entities and the relationship between them. It is considered a best practice to complete ER modeling before implementing your database.

ER modeling helps you to analyze data requirements systematically to produce a well-designed database. So, it is considered a best practice to complete ER modeling before implementing your database.

For Detailed Data Description, Refer to the PVR Data Documentation.

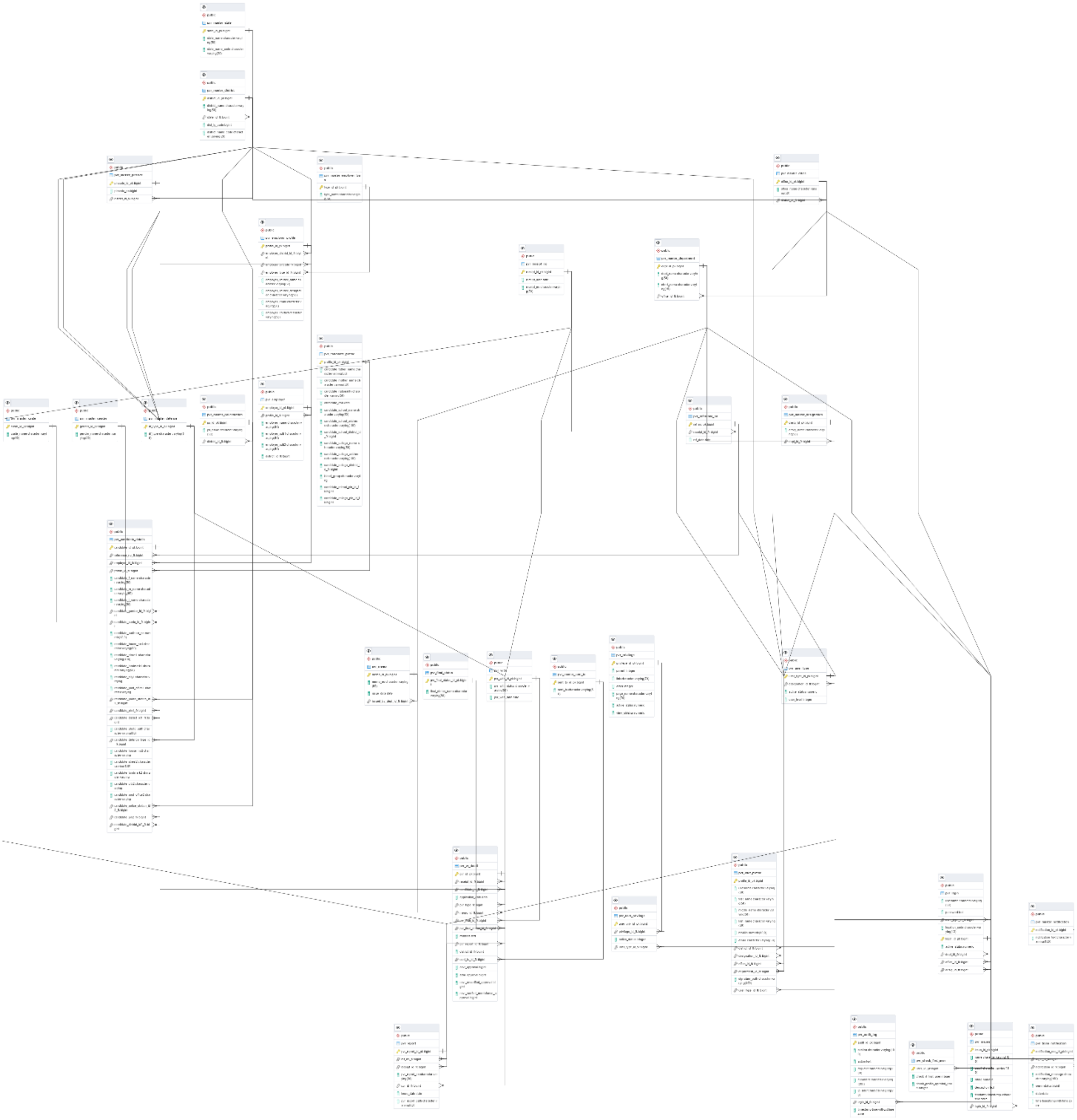


Fig 3 ER Diagram of PVR.

4.4. Module Summary

4.4.1. Login

First Time Login:

The username for log in is the registered email address that the user will provide to the System Administrator during account creation. A one-time password will be provided to the user which they can find in their mail inbox after successful account creation from the administrator’s end.

Forget password:

If the user forgets his/her password, then he needs to head on to the Forget Password option where he/she needs to enter in his/her email address and a password reset link will be sent to his/her email address. By tapping on to that link user will be redirected to the reset password portal and from there user can successfully change his/her password.

First Time Login Change Password:

After successful login with one time password the user need to reset their password and then login.

4.4.2. Dashboard

The main attraction of the system is the Dashboard. Dashboard clearly indicates number of pending enquiries in last day of previous quarter, number of pending requests received during the quarter, number of cases pending as on the last day of quarter, and number of reports pending more than 120 days. The menu holds different functionalities used for online processing. Dashboard has various other features like activity log, notifications to make the dashboard more interactive.

4.4.3. Application

4.4.3.1. New Application

This page contains a application form which will be filled up by Data entry operator from VR section. This option is only available to this user only. From name to address, reference number to contact details, all this data will be captured here.

4.4.3.2. Add office

This page contains a form which can be filled up by data entry operator (VR Section), Officer-In-Charge (VR Section), Additional-District-Magistrate (ADM) A new office can be added in database by this form. Office name and Office addresses will be captured here.

4.4.3.3. Check Status

This module will be used to check status of an application. User can search application status by application's receipt number, reference no, memo no. This page can be accessed by data entry operator (VR Section), Officer-In-Charge (VR Section), Additional-District-Magistrate (ADM). After searching tracking of application status will be displayed.

4.4.3.4. Scrutiny Applications

This page will show all the applications which have been processed by IB. Here data entry operators can manually verify or reject individual applications as per feedback received from IB in offline mode.

4.4.4. Letter

4.4.4.1. Verified letter to employer

This page will show all the non-defence applications which have been verified by IB. Here Officer-In-Charge (VR Section) can manually confirm and generate printable verified non-defence letters to employers for individual application.

4.4.4.2. Unverified letter to employer

This page will show all the applications which have been rejected by IB. Here Officer-In-Charge (VR Section) can manually confirm and generate printable unverified letters to employers for individual application.

.

4.4.5. Report

4.4.5.1. Monitoring Report

This module takes start and end dates as inputs and then shows a printable monthly report No of cases pending enquiries in the last day of previous quarter, No of request received during the quarter, No of enquire report received from D.I.B during quarter under report, No of request sent to the employing agencies during quarter under report, No of cases pending as on the last day of quarter, No of report pending more than 120 days within the mentioned period. This functionality can be accessed by data entry operator (VR Section), Officer-In-Charge (VR Section), Additional-District-Magistrate (ADM).

4.4.5.2. Daily Application Report

This module takes start and end dates as inputs and then shows a printable daily application report containing all application's details like Name of Candidates with Address, Name of the Employer with Address, Employer's Memo No. & Date, DM Office memo No.& Date, Receive Date in a tabular format within the mentioned period.

4.4.6. Activity Log

It shows every audit trail by every user and the system administrator can search any particular user’s audit trail by their username or timestamp.

4.4.7. Notifications

It allows OCVR and ADM to give a pop-up reminder whenever application submitted by Data Entry Operator and also shows approval notification to ADM.

4.4.8. Profile

It shows the details such as name, username, designation etc. about the particular user who login in the current timestamp.

4.4.9. Super Admin Panel

The super admin dashboard can only be accessed by the System administrator only and he/she can find the option in the sidebar as Super Admin. From there a dashboard will open with the following options User Registrations, Report, Audit Log, issues.

User registration:

The user registration tab allows the system administrator to add new users to the PVR portal. For that he/she needs a valid email id(Personal email id of the users) and their departments, designations, locations, user types so that the particular user can be registered and a one-time password will be sent to that user in his/her personal email on successful registration.

4.4.10. Issues Lodged

It allows the system administrator, ADM to check the issues given by the users about the system.

4.5. API

In our application, api is included so that the android app developed can use it to fetch data from PVR web database. JWT is used to achieve the same. Now first of all we look into why use JWT.

JSON Web Token (JWT) is an open standard (RFC 7519) that defines a compact and self-contained way for securely transmitting information between parties as a JSON object. This information can be verified and trusted because it is digitally signed. JWTs can be signed using a secret (with the HMAC algorithm) or a public/private key pair using RSA or ECDSA.

Although JWTs can be encrypted to also provide secrecy between parties, we will focus on *signed* tokens. Signed tokens can verify the *integrity* of the claims contained within it, while encrypted tokens *hide* those claims from other parties. When tokens are signed using public/private key pairs, the signature also certifies that only the party holding the private key is the one that signed it.

Here are some scenarios where JSON Web Tokens are useful:

- **Authorization**: This is the most common scenario for using JWT. Once the user is logged in, each subsequent request will include the JWT, allowing the user to access routes, services, and resources that are permitted with that token. Single Sign On is a feature that widely uses JWT nowadays, because of its small overhead and its ability to be easily used across different domains.

- **Information Exchange**: JSON Web Tokens are a good way of securely transmitting information between parties. Because JWTs can be signed—for example, using public/private key pairs—you can be sure the senders are who they say they are. Additionally, as the signature is calculated using the header and the payload, you can also verify that the content hasn't been tampered with.

Now in our case we have, in the libraries, CodeIgniter API controller and Authorization_token.php file for API Token generation and validation is used. Along with that third party php JWT package is included to achieve the JSON Web Token implementation. Following is the way the api will work.

1. The application or client requests authorization to the authorization server. This is performed through one of the different authorization flows.

2. When the authorization is granted, the authorization server returns an access token to the application.

3. The application uses the access token to access a protected resource (like an API).

Our codes integrated with the PVR web codes is used as the authorization server, where we only return the token on proper login credentials from PVR Android app(Client end).

4.6. Use Case Diagram

Use case diagrams are used to gather the requirements of a system including internal and external influences. These requirements are mostly design requirements. Hence, when a system is analyzed to gather its functionalities, use cases are prepared and actors are identified.

When the initial task is complete, use case diagrams are modeled to present the outside view.

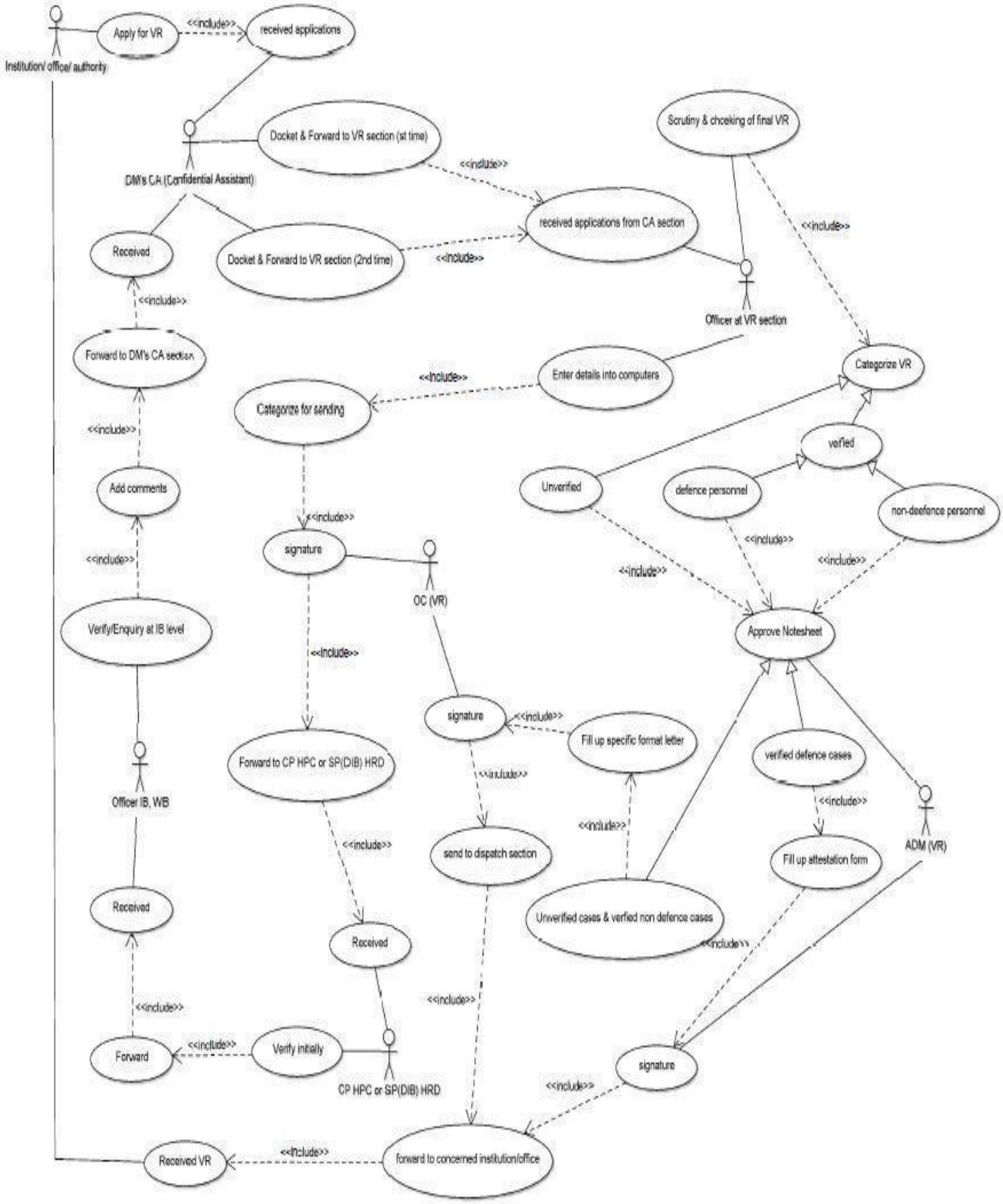


Fig Use Case Diagram

4.7. Sequence Diagram

Sequence Diagrams are interaction diagrams that detail how operations are carried out. They capture the interaction between objects in the context of collaboration. Sequence Diagrams are time focus and they show the order of the interaction visually by using the vertical axis of the diagram to represent time, what messages are sent and when.

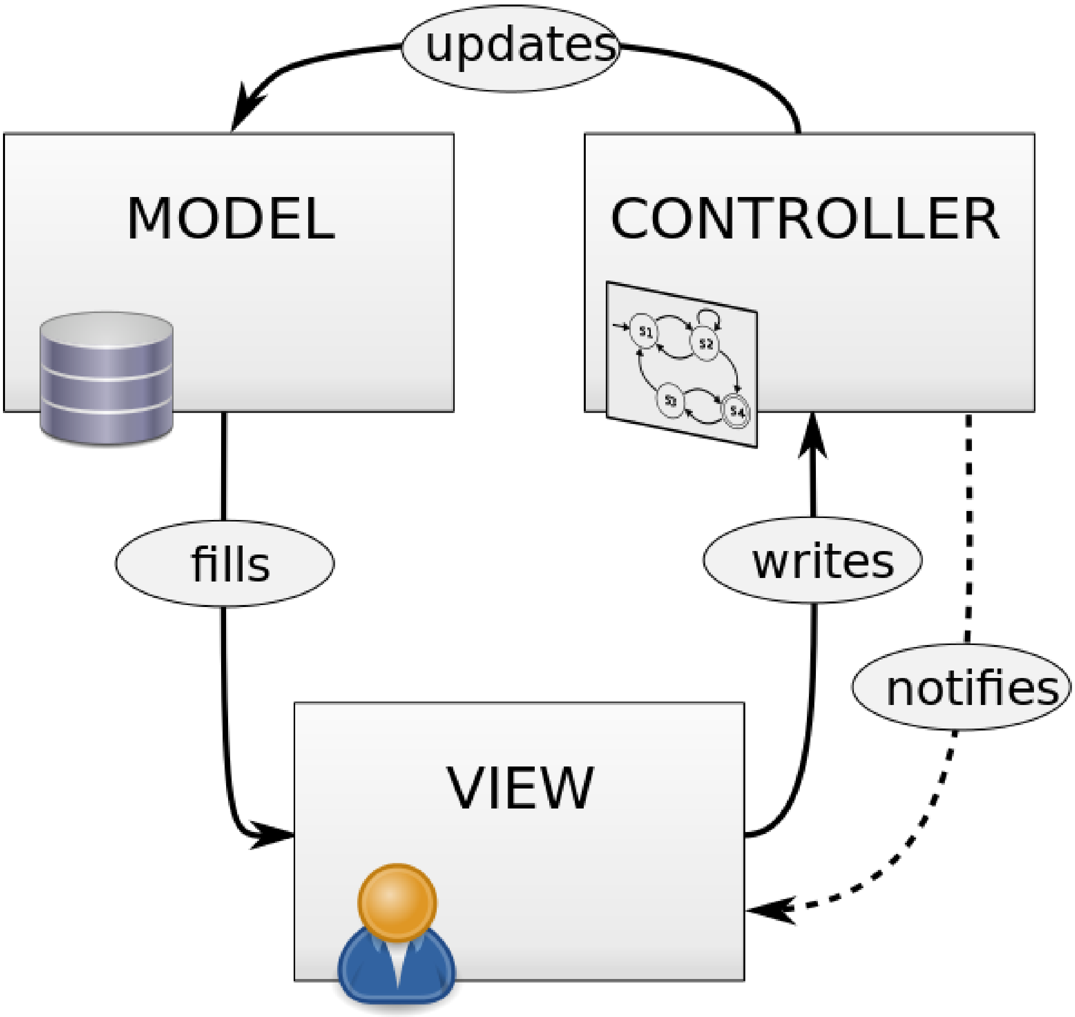


Fig: Sequence Diagram

4.8. Development

Number of developers -

Required skills:

- Knowledge about HTML, CSS, Javascript
- Database Administration in PostgreSQL
- Security testing with OWASP ZAP

Hardware requirements:

- Intel® Pentium® or AMD processor, 1 GHz with MMX or equivalent
- 512 MB of RAM
- 700 MB available hard disk space

Environment and development tools required for the development process:

- Text Editor, Preferably an IDE like Visual Studio Code
- Codeigniter 3
- pgAdmin 4

5. Test Cases

A **Test Case** is a set of actions executed to verify a particular feature or functionality of your software application. A Test Case contains test steps, test data, precondition, post condition developed for specific test scenario to verify any requirement. The test case includes specific variables or conditions, using which a testing engineer can compare expected and actual results to determine whether a software product is functioning as per the requirements of the customer.

List of Test Cases:

SL. NO	Work Done	Test Cases	Pass	Fail
1.	Login	Basic Login and password encryption	Y	
		Password Validation	Y	
2.	Super Admin	User Registration	Y	
3.	Application Fill up	Validation Of Inputs	Y	
		Receipt Number generation	Y	
		Memo Number generation	Y	
4.	Check Status	Check Status	Y	
5.	Update Application	Application Approval	Y	
6.	Add office	Validation Of Inputs	Y	
7.	View Application Details	Generate letter	Y	
		Print Application	Y	
		Approve or Reject Application Working	Y	
8.	Report	Date Range Search	Y	
		Date Specific Search	Y	
		Print Report	Y	
9.	Dynamic Sidebar	Sidebar Selection	Y	
		Sidebar Loading for specific Users	Y	
10.	Notification + Bell	Real-time Notification Prompt	Y	
		Bell Count Loading	Y	
		Notification loading in dropdown menu	Y	
11.	View Application	Data Loading for Submitted Application	Y	
		Application Search	Y	
		View details and Check status for Application working	Y	
12.	Redesigned Table names compatibility	Naming conventions of attributes with proper types	Y	
		Code checking to be compatible after rework	Y	
		Key Mapping and Triggers and Events in DB side	Y	
13.	Commit Rollback for data	DB Commit rollback for every DB transaction	Y	

14.	User Creation	Password mailing	Y	
		User Creation	Y	
15.	User Privilege	User Privilege	Y	
16.	CSRF	CSRF for all pages	Y	
		CSRF for login	Y	
17.	Captcha Verification	Captcha Verification	Y	
18.	SMTP Configuration	SMTP Configuration	N	
19.	Account Recovery	Account Recovery	Y	
20.	Notifications	Notifications	Y	
21.	Seek Record	Validation Of Inputs	Y	

Table 6: Test Cases Table

6. Quality Assurance

Software Quality Assurance (SQA) is simply a way to assure quality in the software. It is the set of activities which ensure processes, procedures as well as standards suitable for the project and implemented correctly.

Software Quality Assurance is a process which works parallel to development of a software. It focuses on improving the process of development of software so that problems can be prevented before they become a major issue. Software Quality Assurance is a kind of an Umbrella activity that is applied throughout the software process.

List of SQA Attributes in ensured in the system:

Sl. No	Feature Description	Yes	Not Applicable
1	Responsive design - User is able to View and interact with every page of the application on any device with a minimum of resizing, panning or scrolling.	Y	
2	Dashboard - Dashboards are designed and implemented in the application to display exactly what is relevant to different users based on either user role or any other criteria.	Y	
3	Alerts, Notification, Exceptions can be configured to notify user of events that require attention.	Y	
4	Error Messages - Depending on user requirement, language preference and display mode of error messages can be configured.	Y	
5	Search Option - The search option is designed such that the appearance and behavior of the search pages can be configured easily. Like search result grid, personalized search, option to save search criteria etc.	Y	
6	The masters like State, District, Block, Village etc. are taken from standard directories like LG directory to facilitate standardization and interoperability.	Y	
7	The application uses Metadata standards (MDDS eGov standards) and standard international codes for country, region etc. along with any domain specific codes like Unit of measurement, port codes, vessel codes, Harmonized System (H.S.) Commodity codes wherever applicable.	Y	
8	The application allows management of Masters Data i.e. the local Code Directories like designation, relation, location etc. and the same are configurable and manageable by the user.	Y	
9	The Workflow and routing is fully configurable using database or other mechanism without any hard coding.	N	
10	The Business Rules are configurable using graphical user interface by the user (at Administrative level) without any programming.	N	
11	The application provides RBAC (Role Based Access Controls). The application Administrator is able to define and manage user roles and controls using intuitive graphical user interface.	Y	

12	The application provides built-in auto feedback mechanism i.e. option is provided to the user to add feedback at the page/form level directly while using the application.	Y	
13	UI/UX, business logic & database are separate and configurable to accommodate frequent business logic/flow change requirements.	Y	
14	The Geo-location based user identification (IP Geo-location) is implemented in the application to present localized content.		N
15	The application is built on modular architecture with clear segregation of presentation, business and data layer as per best practices.	Y	
16	The tiers are configurable with minimal effect on other tiers.	Y	
17	The application is designed and developed on Service Oriented Architecture to avoid dependencies.	Y	
18	The application is compatible with the latest versions of popular browsers (Edge, Chrome, Mozilla Firefox, Safari, Internet Explorer etc.).	Y	
19	The deployment architecture provides high availability and is horizontally scalable.	Y	
20	The application supports packaged installation with minimum user intervention. The application can be managed remotely. The application provides auto-update features and updates can be pushed from central servers.		Y
21	The application supports external configuration files with run-time management option i.e. the system properties can be set through external configuration files without requirement of compilation and redeployment.		Y
22	The application is designed in such a way that changes to the core software can be made without impacting externally integrated systems/components.		Y
23	The application implements API versioning and maintains backward compatibility for functionality.		Y
24	The application implements data segregation and other security features to ensure data security and prevention of unauthorized access to data of one tenant by other tenants.	Y	
25	The application has facility to identify, classify and protect sensitive data items and implements mechanisms such as encryption, tokenization and access control, wherever appropriate, to secure data both in transit and at rest.	Y	
26	The application provides the user(s) to have full control over their data and use it for any further analysis	Y	
27	The application has built-in mechanisms and provides tools to reduce or eliminate the need for direct access of Database.	N	
28	The application implements automatic backup and recovery mechanism and provide tools for the user organizations to backup their data.	N	
29	The application provides graphical reporting using graphics, dashboards and templates.	Y	
30	The application provides reporting features allowing users to customize existing reports.	N	
31	The application provides ad-hoc query options to generate dynamic reports by selecting parameters.	Y	
32	The reporting framework provides comprehensive data export feature in CSV, PDF, Excel and HTML formats.	Y	

33	The application is able to audit all system and user actions and ensures that all actions performed on data are recorded, keeping track of users, date & time.	Y	
34	The application ensures that the log captured, includes all of the event and activity logs	Y	
35	The application implements mechanism to store and maintain log data securely for audit.	Y	
36	Mechanism for real-time monitoring of logs is implemented with alert and notifications to handle security incidents and data leaks.	Y	
37	Implementation of Authentication	Y	
38	Sign-up/login Processes	Y	
39	The terminology/language used for the form is familiar to the user and well understood	Y	
40	Field labels on forms clearly explain what entries are desired	Y	
41	Labels are placed close to the data entry fields	Y	
42	Fields in data entry screens contain default values when appropriate and show the structure of the data and the field length	Y	
43	The application automatically formats the entered data in the desired format		Y
44	The user is able to Tab through the form using keyboard and Tabs are indexed properly	Y	
45	Clear visual ‘notification’ is provided by using color change, highlighting border or fade in box	Y	
46	Pull-down menus, radio buttons and check boxes are used in preference to text entry fields on forms (i.e. text entry fields are not overused)	Y	
47	The application is carrying out the field level validation and form level validations at appropriate times	Y	
48	Both Server Side validation and Client side validations are incorporated appropriately	Y	
49	For file upload Restriction on Document type and extension is incorporated in validation	Y	
50	[Alpha-numeric CAPTCHA with minimum 6 characters]	Y	
51	Implement proper validations on all input parameters in client and server side (both).	Y	
52	Use parameterized queries instead of inline SQL queries	Y	
53	Do not reference components (such as JavaScript) directly third-party sites.	Y	
54	Hash the password before it is relayed over network, or is stored in database.	Y	
55	Implement Change Password and Forgot password applications	Y	
56	Use Post methods to pass parameters as values from one page/website to another.	Y	
57	Implement proper error-handling.	Y	
58	Store uploaded files in database, rather than storing them in file system	Y	
59	Implement proper Session Timeout	Y	
60	Implement Logout buttons in all authenticated pages	Y	
61	Use the latest and non-vulnerable versions of Application Server (IIS/Apache etc.), JQuery etc.	Y	

62	Name of the data element are meaningful to users and clear at the database level	Y	
63	Reserve words have not been used for data elements.	Y	
64	Name of the element are unique in the data dictionary to avoid any ambiguity	Y	
65	A naming convention has been adopted to enhance clarity, consistency, meaningful data and better understanding among team members	Y	
66	The source of data is recorded for each data element in the database.	Y	
67	Data Type has ideally been defined. Choice of data type has been governed by what data type is optimal for the data element in terms of minimum storage, ease of retrieval and the nature of its usage.	Y	
68	Data length has been defined for all the fields optimally.	Y	
69	A pre-defined set of values has been defined for the data elements wherever applicable.	Y	
70	Proper constraints have been defined in database for mandatory, optional and other validations.	Y	
71	Email ID is verified by sending the verification link to that email.	Y	
72	Mobile Number is verified by sending OTP to that number or getting a missed call from that number.		Y
73	Aadhar Number is being authenticated online and managed as per UIDAI guidelines and Aadhaar Act.		Y
74	Wherever applicable, the verifiable fields are being verified from its source for correctness. For example, Driving License Number to be verified from Saarthi DB.		Y
75	Data type validations (allowing only numbers in a numeric data element), allowed values validations, range validations, validation against regular expressions such as email etc have been implemented.	Y	Y
76	Wherever input data is being taken through API, same is being validated before storing it in DB	Y	
77	The masters like State, District, Block, Village etc. are taken from standard directories to facilitate standardization and interoperability?	Y	
78	The application uses Metadata standards (MDDS eGov standards) wherever applicable.		Y
79	The application uses standard international codes for country, region etc. wherever applicable.	Y	
80	The application uses domain specific codes like Unit of measurement etc wherever applicable.	Y	
81	Appropriate control such as radio button, check box, combo, list box or date picker control etc have been used as an input mechanism for data entry	Y	
82	The field label/caption should be meaningfully defined. For example, for group of data element day_birth, month_birth, and year_birth, the caption may be Date of Birth.	Y	
83	In-line/tooltip text is available for each form field including menu/sub-menu items	Y	
84	The identifiers satisfy the criteria such as Availability, Longevity, Privacy, Numeric Value to cater to multi-lingual society.		
85	Check digit has been provisioned for identifiers wherever applicable.	Y	

86	First digit of the identifiers is not zero.	Y	
87	Primary key has been properly identified, defined and are not composite.	Y	
88	Unique key has been properly identified and defined.	Y	
89	Referential integrity is being maintained in the database. The naming convention used is such that it easily recognizes the relation between filed and its table.	Y	
90	Database has been designed to handle multilingual/unicode data.	Y	
91	Instead of using Null value allowed, a default value or Not Available option has been provided to avoid system errors getting generated at the back-end as exceptions due to non-checking of null value.	Y	
92	Every record along with data should also have timestamp for creation, last updating, user identification like user-id, IP address etc.	Y	
93	Hidden field validation	Y	
94	Add the security header for parent controller	Y	
95	CSRF on Ajax post	Y	
96	Use of Base Path	Y	
97	No external links/CDN	Y	
98	No hard coding in code	Y	
98	Removal of redundant functions/code	Y	
99	Removal of console logs from the system	Y	
100	Every form in a view file shall be able given through CodeIgniter form_open() for CSRF	Y	
101	Headers Should be added in the base controller	Y	
102	Code Authoring	Y	

Table 6.1.: Quality Assurance Table

7. ZAP Reports

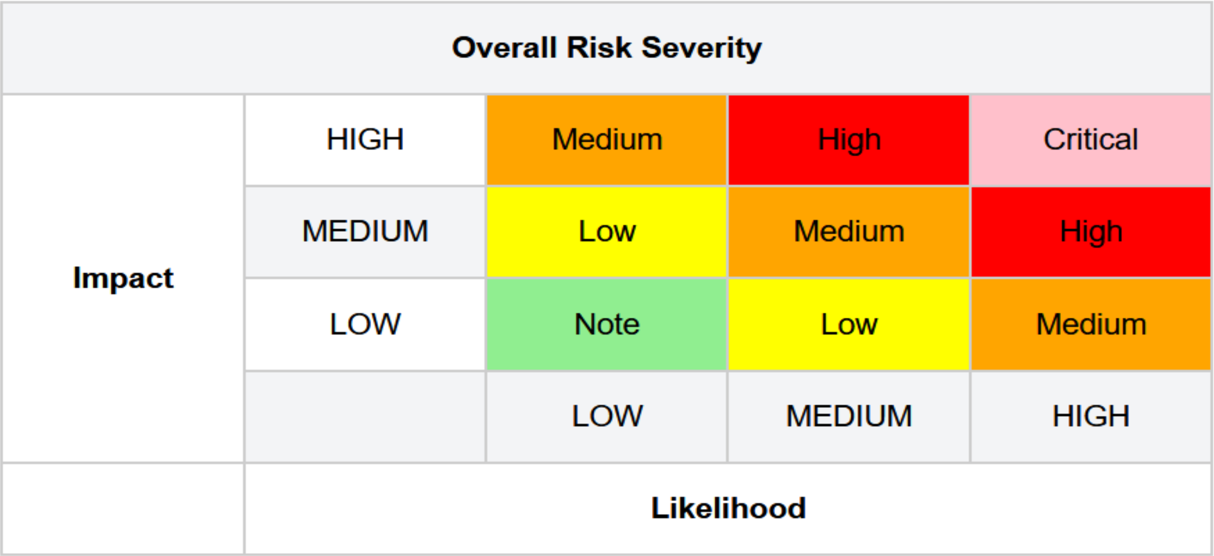
As ZAP spiders the web application, it constructs a map of the web applications' pages and the resources used to render those pages. Then it records the requests and responses sent to each page and create alerts if there is something potentially wrong with a request or response.

ZAP assess the risks and classifies them into four categories:

- High
- Medium
- Low
- Informational

Determining Severity

However the tester arrives at the likelihood and impact estimates, they can now combine them to get a final severity rating for this risk. Note that if they have good business impact information, they should use that instead of the technical impact information. But if they have no information about the business, then technical impact is the next best thing.



Overall Risk Severity				
Impact	HIGH	Medium	High	Critical
	MEDIUM	Low	Medium	High
	LOW	Note	Low	Medium
		LOW	MEDIUM	HIGH
	Likelihood			

In the example above, the likelihood is medium and the technical impact is high, so from a purely technical perspective it appears that the overall severity is high. However, note that the business impact is actually low, so the overall severity is best described as low as well. This is why understanding the business context of the vulnerabilities you are evaluating is so critical to making good risk decisions. Failure to understand this context can lead to the lack of trust between the business and security teams that are present in many organizations.

Ref: https://owasp.org/www-community/OWASP_Risk_Rating_Methodology

Module:

1) Login

Risk Level	No. of Alerts
High	0
Medium	2
Low	5
Informational	0

2) Dashboard

Risk Level	No. of Alerts
High	0
Medium	1
Low	4
Informational	1

3) New Application

Risk Level	No. of Alerts
High	0
Medium	1
Low	4
Informational	2

4) Add Office

Risk Level	No. of Alerts
High	0
Medium	1
Low	3
Informational	0

5) Check Status

Risk Level	No. of Alerts
High	0
Medium	1
Low	3
Informational	0

6) Update Application

Risk Level	No. of Alerts
High	0
Medium	0
Low	2
Informational	0

7) Monitoring Report

Risk Level	No. of Alerts
High	0
Medium	1
Low	3
Informational	0

8) Monthly Application Report

Risk Level	No. of Alerts
High	0
Medium	1
Low	3
Informational	0

8. Future Scope

1) **Scalability:**
   Currently the project will work at district (Howrah) level only. But it has been prepared as such it can be scaled up to state level (West Bengal) or national level (India).