## B.M.S COLLEGE OF ENGINEERING BENGALURU

Autonomous Institute, Affiliated to VTU



OOMD Mini Project Report on

## DIGIVERIFY

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

Bachelor of Engineering in

Computer Science and Engineering

*Submitted by:*

### MOHAMMED ZEESHAN UMAR (1BM22CS160)

**MUKUND RAGHAVAN SADAVARTHI (1BM22CS166)**

**NAVANEETH V N (1BM22CS171)**



Faculty In charge **SANDHYA A K**

Assistant Professor

Department of Computer Science and Engineering

B.M.S College of Engineering

Bull Temple Road, Basavanagudi, Bangalore 560 019 2024-2025

## B.M.S COLLEGE OF ENGINEERING DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING



***DECLARATION***

We, MOHAMMED ZEESHAN UMAR (1BM22CS160) , MUKUND RAGHAVAN SADAVARTHI (1BM22CS166) and NAVANEETH V N (1BM22CS171) , students of 5th Semester, B.E, Department of Computer Science and Engineering, BMS College of Engineering, Bangalore, hereby declare that, this OOMD Mini Project entitled "DIGIVERIFY" has been carried out in Department of CSE, BMS College of Engineering, Bangalore during the academic semester September - February 2023. I also declare that to the best of our knowledge and belief, the OOMD mini Project report is not from part of any other report by any other students.

**Signature of the Candidate**

MOHAMMED ZEESHAN UMAR (1BM22CS160)

MUKUND RAGHAVAN SADAVARTHI (1BM22CS166)

NAVANEETH V N (1BM22CS171)

# BMS COLLEGE OF ENGINEERING

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**



***CERTIFICATE***

This is to certify that the OOMD Mini Project titled “**DIGIVERIFY”** has been carried out by Mohammed Zeeshan Umar (1BM22CS160) , Mukund Raghavan Sadavarthi(1BM22CS166) and Navaneeth V N(1BM22CS171) during the academic year 2024-2025.

Signature of the Faculty in Charge

# Table of Contents

|  |  |  |
| --- | --- | --- |
| **Sl No.** | **Title** | **Page No.** |
| 1 | Problem Statement | 1 |
| 2 | Software Requirement Specification | 2-8 |
| 3 | Class Diagram | 9-10 |
| 4 | State Diagram | 10-11 |
| 5 | Interaction Diagram | 12-18 |
| 6 | UI Design | 19-27 |

1. **INTRODUCTION**

The rapid advancement of technology has revolutionized various industries, including healthcare. One notable development in the healthcare sector is the emergence of online pharmacy systems. An online pharmacy system refers to a digital platform that allows individuals to purchase pharmaceutical products and medications over the internet, eliminating the need for physical visits to brick-and-mortar pharmacies. This report aims to provide an in-depth analysis of online pharmacy systems, their features, benefits, and potential implications.

### Problem Statement:

Traditional pharmacy systems often present several challenges and limitations. Individuals with restricted mobility, limited access to transportation, or residing in remote areas may face difficulties in obtaining necessary medications. Moreover, long queues, limited availability of certain drugs, and potential privacy concerns contribute to an overall suboptimal experience for customers. Addressing these challenges and improving the overall pharmacy experience has become crucial in the digital age.

### Motivation:

The motivation behind studying and understanding online pharmacy systems stems from the need to leverage technology to overcome the limitations of traditional pharmacy setups. Online pharmacies offer a convenient alternative that allows users to browse a wide range of pharmaceutical products, place orders, and have them delivered to their doorstep. This increased accessibility and convenience have the potential to enhance medication adherence, particularly for individuals with chronic conditions who require regular prescriptions. Additionally, online pharmacy systems can promote patient privacy, streamline the prescription verification process, and optimize inventory management for pharmacies.

By comprehensively exploring the features, benefits, and potential implications of online pharmacy systems, this report aims to shed light on their impact on the healthcare industry, patient experience, and the future of pharmaceutical services. This knowledge will enable healthcare professionals, policymakers, and stakeholders to make informed decisions and advancements in the field of online pharmacy systems.

# DIGIVERIFY(online validation system)

1. **SOFTWARE REQUIREMENT SPECIFICATION**

### Introduction:

* 1. **Purpose of this Document:** The main aim of this document is to briefly and clearly describe the problem statement, how this document aims to solve the problem statement and what are the requirements, specifications etc required to solve the problem.
  2. **Scope of this Document:** This document just provides a brief look on how the problem will be solved and what the functionalities of the application developed to solve the problem is. This application developed is used to verify the documents issued by government bodies only to individual or corporate entities(both private and public). The application would require an initial amount of Rupees 1,00,00,000 and a time period of 1 year to implement the project.
  3. **Overview:** The Application acts as a medium for government departments to verify the Personal Identification Information(PII) of various entities when they provide some documents for any of the work involves disclosure of various government issued documents of the entities.

### General Description:

The application developed will act as a median between the various departments established under the different levels of the Government and the various individual and organisational entities and also helps to verify these documents.

* 1. **Individual User and/or Organizations:**

These users will be able to apply for the issuance of new documents for some purposes, extension of validity of these documents and updating details in these documents. The users will also be able to use this application to prove the correctness of their documents to the concerned departments. This class of users must provide correct details to the concerned departments at the time of applying for the new document to ensure that the system is reliable and flawless.

* 1. **Departments Under Government and Statutory Bodies:**

The application will cater the needs of these entities with a first priority basis. This class of users will use this application to issue documents and certificates to the Individual Users or Organizations and will also use this application to verify the details of the documents and to check the correctness of the information present in the documents provided by the latter when any work related to the process is to be done by the government. This class of user must enter the correct information to make the system reliable and flawless. They also should verify details of the User before inserting the details into the document and uploading the document/data onto the application database.

Main functionality of this application is to act as a guardian against false information and to protect the integrity of the Public system that both classes of Users mentioned above use.

### Functional Requirements:

* 1. **Functional Requirements of User/Organisations:**
     1. **Document Management**
        1. **Input Details**: Users must input accurate and comprehensive details to initiate document verification or issuance.
        2. **Request Services**:
        3. Apply for new documents.
        4. Request extensions or updates to existing documents.
        5. Validate the correctness of their documents using the system.
     2. **Authentication**
        1. Secure login for each user to prevent unauthorized access.
        2. Multi-factor authentication for sensitive processes like document submission or updates.
     3. **Data Privacy and Security**
        1. Implement encryption for sensitive Personally Identifiable Information (PII) during transmission and storage.
        2. Ensure compliance with data protection regulations to safeguard user details.
     4. **View Current Status**
        1. Users can check the status of their requests and receive notifications about updates or required actions.
     5. **Error Reporting**
        1. Provide clear feedback in case of invalid input, missing information, or verification failure.
     6. **Integration with Government Databases**
        1. Seamless integration with existing government databases for faster validation of submitted documents.
     7. **User-Friendly Interface**
        1. Easy navigation for all types of users to interact with the application effectively.
        2. **Input Details**: Users must input accurate and comprehensive details to initiate document verification or issuance.
        3. **Request Services**:
           1. Apply for new documents.
           2. Request extensions or updates to existing documents.
           3. Validate the correctness of their documents using the system.
  2. Functional Requirements (for Government Departments)Functional Requirements (for Government Departments)
     1. **Verification of Documents**
        1. Verify the authenticity of documents submitted by users/organizations in real-time.
        2. Generate validation reports accessible by concerned departments.
     2. **Document Issuance and Updates**
        1. Issue new government documents securely and quickly.
        2. Update records and issue revisions as needed.
     3. **Database Management**
        1. Maintain a central repository for issued documents with robust access control.
        2. Audit trails for all operations performed by department users.
     4. **Inter-Department Collaboration**
        1. Enable collaboration between multiple government departments for processing shared cases or cross-verifications
     5. **Analytics and Reporting**
        1. Provide dashboards and analytics for monitoring usage, detecting anomalies, and improving operations.
     6. **Security Protocols**
        1. Implement role-based access to ensure only authorized personnel can manage sensitive information

### Interface Requirements:

* 1. **User Interface (UI) Requirements**
     1. **Dashboard:**
        1. A clean and intuitive dashboard for users to view their requests, document status, and notifications.
        2. Include distinct sections for "Apply for Documents," "Verify Documents," and "Track Requests."
     2. **Form Fields:**
        1. Clearly labeled fields for document uploads, personal details, and other information.
        2. Provide tooltips and validation messages for accurate data entry.
     3. **Notifications:**
        1. Real-time notifications on application status, errors, and other updates.
        2. Notifications should appear in the app and via email or SMS if enabled.
     4. **Accessibility**
        1. Ensure compatibility with screen readers.
        2. Offer multi-language support for broader reach.
        3. Implement high contrast themes for users with visual impairments.
  2. **Backend Interface Requirements**
     1. **Database Integration:**
        1. Seamless integration with government databases for fetching and verifying document details.
        2. API endpoints for data exchange between the app and external systems.
     2. **Security:**
        1. Encrypted communication protocols (e.g., HTTPS, SSL/TLS) for all data exchanges.
        2. Token-based authentication for API access.
  3. **Admin Interface Requirements**
     1. **Access Control:**
        1. Role-based dashboards to restrict access to specific features (e.g., read-only for certain staff).
        2. Ability for administrators to manage user roles and permissions.
     2. **Reports and Analytics:**
        1. Interfaces to generate usage statistics, error logs, and performance metrics.
     3. **Document Management:**
        1. Easy upload, search, and edit capabilities for government-issued documents.

### Performance Requirements:

1. **Response Time**
   1. The system must process and verify documents within **15 seconds** for individual requests.
   2. The dashboard should load in under **5 seconds** for typical users.
   3. API calls for document validation or status updates should respond within **2-5 seconds** under normal load conditions.
2. **Scalability**
   1. The system should handle at least **10,000 concurrent users** without performance degradation.
   2. It must support a 10x increase in users during peak times, such as government campaign periods or application deadlines.
3. **Availability**
   1. The application must maintain **99.9% uptime**, ensuring minimal downtime throughout the year.
   2. Scheduled maintenance should not exceed **2 hours per month**, with advance notice provided to users.
4. **Throughput**
   1. The system should handle **1,000 document uploads per minute** during peak operations.
   2. It must support **5,000 verification requests per minute** at maximum load.
5. **Data Consistency**
   1. All submitted data should reflect accurately in the system within **1 second** of submission, ensuring real-time consistency across all modules.
6. **Error Rate**
   1. The application should have an error rate of less than **0.1%** for all critical operations such as document verification and submission.
7. **Resource Utilization**
   1. The application should use no more than **75% of available server resources** under normal operating conditions, leaving headroom for unexpected spikes.
8. **Load Handling**
   1. The system must support stress testing scenarios, ensuring smooth operation under conditions such as:
      1. **10x normal user load**.
      2. Continuous operation for **72 hours** without degradation.

### Design Constraints:

1. **Compliance Constraints**
   1. The system must comply with data protection regulations, such as:
   2. Information Technology Act, 2000 for Indian users.
   3. Other regional and international privacy laws as applicable.
2. **Technology Constraints**
   1. The application should be developed using **open-source technologies** wherever possible to reduce costs.
   2. The backend must support integration with **existing government databases** using secure and standardized APIs.
3. **Security Constraints**
   1. All data in transit must be encrypted using **TLS 1.2 or higher**.
   2. The system should implement role-based access control (RBAC) to limit permissions for users and administrators.
   3. Personal Identifiable Information (PII) must be stored using **256-bit encryption**.
4. **Platform Constraints**
   1. The system must be accessible on both **desktop (web)** and **mobile devices (Android and iOS)**.
   2. The application must be compatible with all major web browsers, including Chrome, Firefox, Edge, and Safari.
5. **Resource Constraints**
   1. The budget for the project is capped at **INR 1 crore**, covering development, testing, and deployment.
   2. The development timeline is constrained to **12 months** from project initiation.
6. **Database Constraints**
   1. The database must handle large datasets of PII with a guaranteed query performance under high load.
   2. Must support **relational database architecture** to ensure data consistency.
7. **Interoperability Constraints**
   1. The application must integrate with third-party systems (e.g., Aadhaar, PAN databases) via standardized protocols such as RESTful APIs.
   2. The system should support multiple file formats (e.g., PDF, JPG, PNG) for document uploads.
8. **UI/UX Constraints**
   1. The user interface must be designed for **accessibility standards**, such as colour blindness, contrast website, imcrease font size etc.
   2. Interfaces should load within **5 seconds**, even on lower bandwidth networks.
9. **Hardware Constraints**
   1. The system must run on **cloud infrastructure** to ensure scalability and high availability.
   2. It must also support deployment on government-approved data centers for compliance.

### Non-Functional Attributes:

1. **Security:** The system should ensure the security of user data through encryption, secure authentication, and access controls to prevent unauthorized access and data breaches.
2. **Performance:** The system should provide fast response times and handle a large number of concurrent users and transactions without significant performance degradation.
3. **Reliability and Availability:** The system should be reliable and available to users at all times, with minimal downtime for maintenance or upgrades.
4. **Scalability:** The system should be able to scale to accommodate future growth, handling larger product catalogs, user bases, and transaction volumes.
5. **Usability:** The system should offer a user-friendly interface and intuitive navigation, ensuring ease of use and a positive user experience.
6. **Compatibility:** The system should be compatible with various devices, operating systems, and web browsers, ensuring a consistent user experience across different platforms.

**Preliminary Schedule:**

* Requirements Gathering:2 months
* System Design: 3 months
* Development: 1 year
* Testing and Quality Assurance: 6 months
* Total Time:1 year 11 months.

**Preliminary Budget: 100000000 INR**

# CLASS MODELING

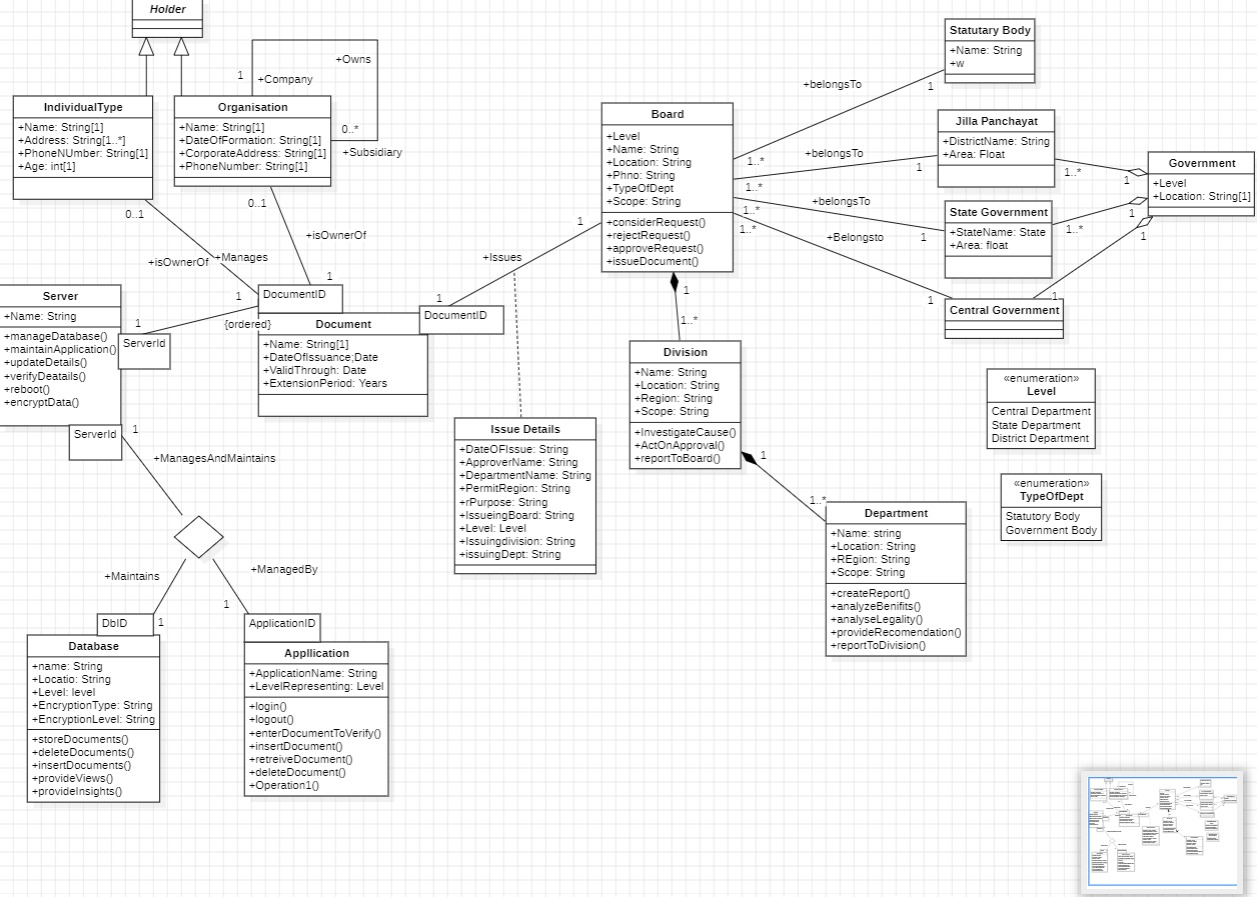
****

Fig 3.1

 **Holder**  
Represents a generic entity with associations to individuals or organizations. Acts as a superclass for IndividualType and Organisation.

 **IndividualType**  
Manages individual-specific details such as name, address, phone number, and age. Associated with ownership of documents.Here The document can be an individual person who can use the application to apply for a new document, renew a document or verify the document.

 **Organisation**  
Contains details like name, date of formation, addresses, and phone numbers. Can manage or own documents and subsidiaries. Here The document can be an individual person who can use the application to apply for a new document, renew a document or verify the document.

 **Server**  
Represents a system server performing database and application management. Functions include maintaining applications, verifying, updating, rebooting, and encrypting data.

 **Database**  
Holds data for name, location, level, and encryption details. Supports operations like document deletion, access, and insights.

 **Application**  
Represents an application with functionalities such as login, logout, verifying documents, and basic CRUD operations.

 **Document**  
Captures document metadata like name, issue date, validity, and extension period. Tightly coupled to Server, Database, and Application.

 **Issue Details**  
Stores document approval-related information such as issue date, approver name, purpose, and issuing entities like boards and divisions.

 **Board**  
Governing body with attributes for location, type of department, and scope. It manages requests and issues documents.

 **Division**  
Contains location, region, and scope. Handles functions like investigating causes, approving actions, and reporting to boards.

 **Department**  
Similar to Division, but more granular, focusing on reporting, analysis, legal checks, and benefit assessments.

 **Statuary Body**  
A specific named body which does not work under the government but helps in the running of ht ecountry.

 **Jilla Panchayat**  
Manages district-level geographic details with a name and area specification.

 **State Government**  
Represents state-level governance entities, holding state names and area information.

 **Central Government**  
The highest governing body linking levels of departments.

# 5.STATE MODELING

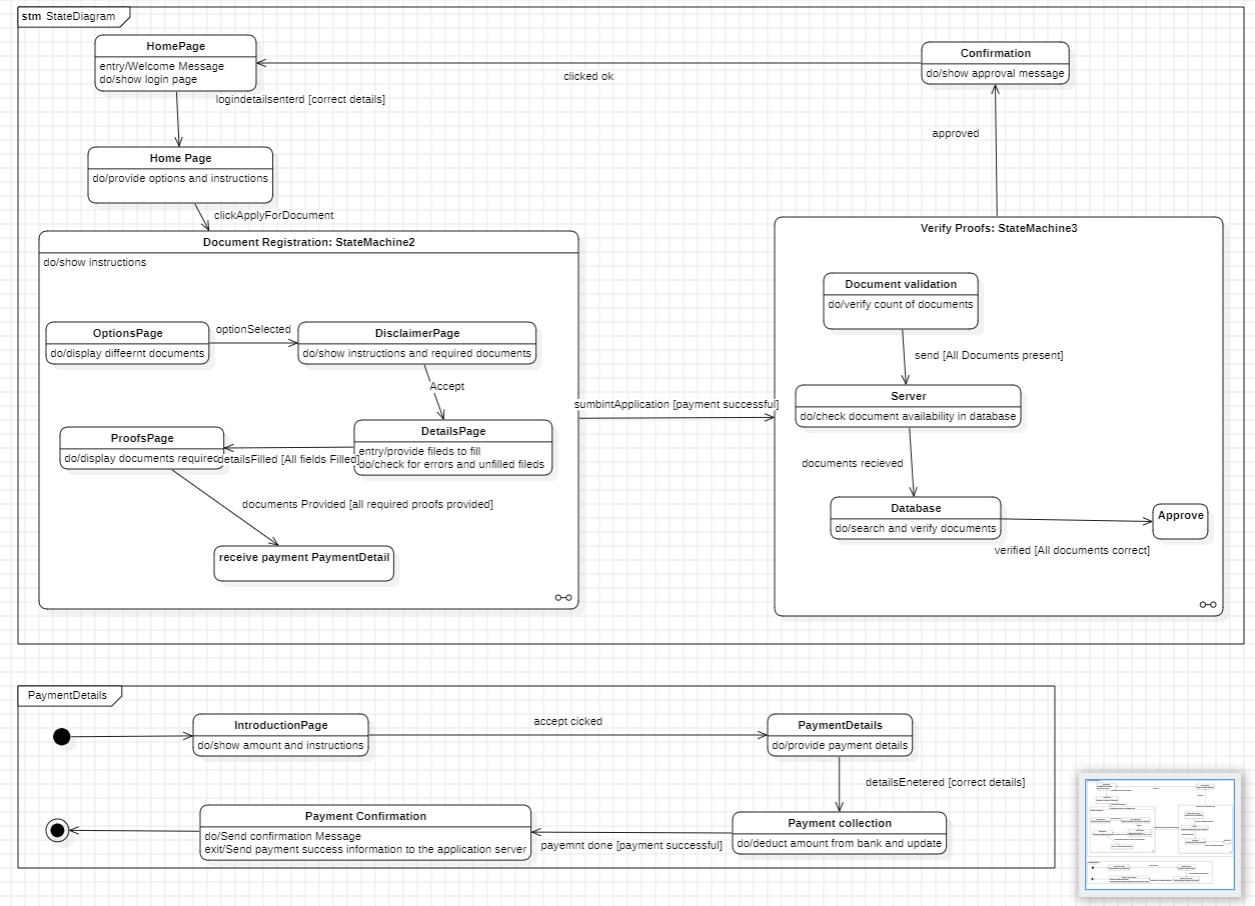
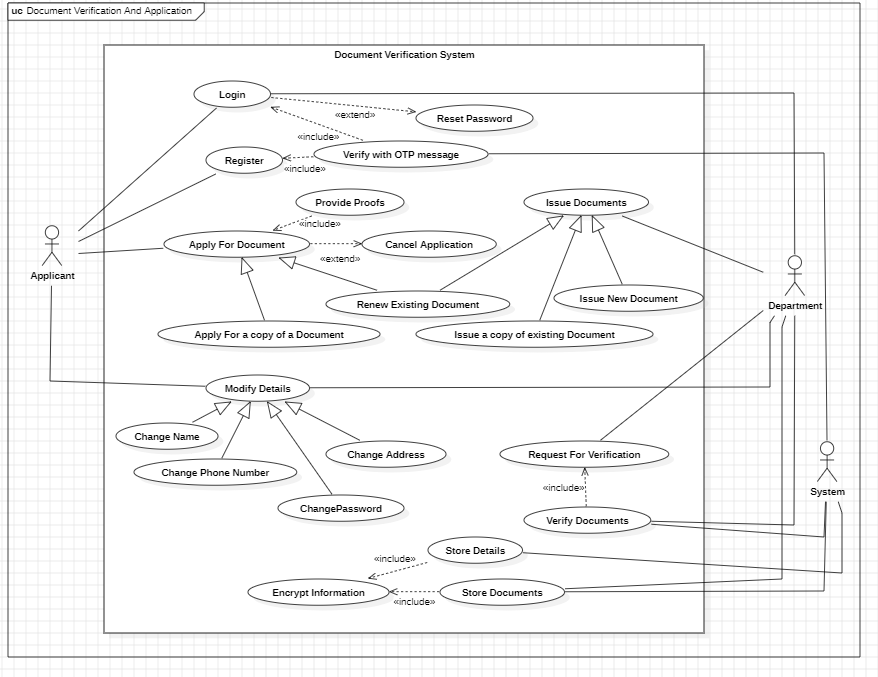
****

Fig 4.1

* **HomePage**:
  + The process begins with a welcome message and displays the login page.
  + Upon correct login details, options and instructions are provided to the user.
* **Document Registration**:
  + **OptionsPage**: Displays different document options.
  + **DisclaimerPage**: Shows instructions and required document details.
  + **DetailsPage**: Allows users to fill fields and checks for errors/unfilled fields.
  + **ProofsPage**: Displays required document proofs. Once provided, payment details are collected.
  + Upon successful payment, the application proceeds for verification.
* **Verify Proofs**:
  + **Document Validation**: Verifies the count of submitted documents.
  + **Server**: Checks document availability in the database.
  + **Database**: Searches and verifies the documents. If all are correct, approval occurs.
  + After verification the user is shown the confirmation page and directed to the home page.
* **Payment Process**:
  + **IntroductionPage**: Displays payment amount and instructions.
  + **PaymentDetails**: Allows payment details entry.
  + **Payment Collection**: Deducts the amount and updates the system.
  + **Payment Confirmation**: Sends success information to the server.
  + **Confirmation State**: Finally, an approval message is shown after all documents and payments are verified.

# 5. INTERACTION MODELING

## USE CASE DIAGRAM

****

### Actors and Their Roles:

* **Applicant**: Performs primary tasks such as document applications, modifications, and proof submissions.
* **Department**: Issues new, existing, or renewed documents and verifies them.
* **System**: Stores and encrypts documents, manages verification, and data updates..

### Use Case:

1. **Login**: Allows the applicant to log into the system.
2. **Reset Password**: Extends login functionality to reset a forgotten password.
3. **Register**: Enables the applicant to create an account.
4. **Verify with OTP message**: Confirms registration using an OTP.
5. **Provide Proofs**: Includes document submission by the applicant.
6. **Apply For Document**: Allows applying for new documents.
7. **Cancel Application**: Extends the application process to cancel a pending request.
8. **Renew Existing Document**: Reapply for an expired document.
9. **Apply For a Copy of a Document**: Request a duplicate of a document.
10. **Issue New Document**: Department issues a new document.
11. **Issue a Copy of Existing Document**: Department issues document copies.
12. **Modify Details**: Modify user information (name, address, phone number, password).
13. **Change Name**: Update user’s name.
14. **Change Phone Number**: Update contact number.
15. **Change Address**: Update residential details.
16. **Change Password**: Modify login credentials.
17. **Request For Verification**: Submit documents for verification.
18. **Verify Documents**: System verifies submitted documents.
19. **Store Details**: Save user or document information.
20. **Encrypt Information**: Securely encrypt stored information.
21. **Store Documents**: Includes saving submitted documents in the system.
22. **Issue Documents**: Department processes and issues documents.

## SEQUENCE DIAGRAM

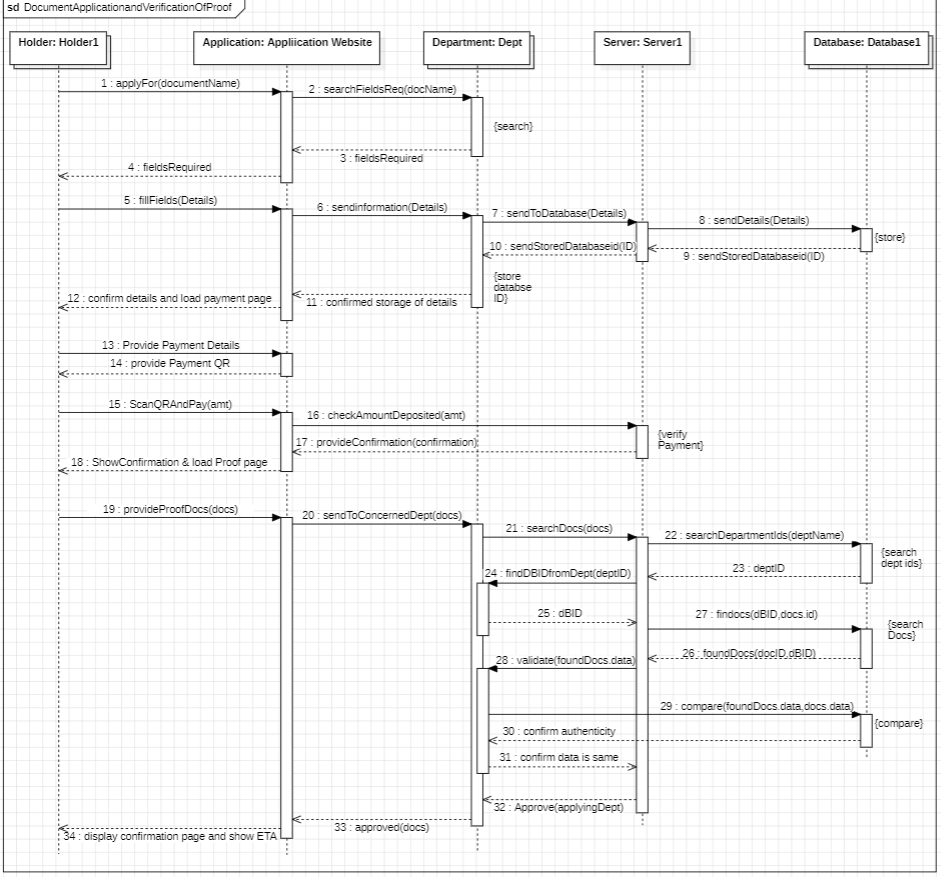
****

Fig 5.2

**1. Initiating the Application**

1. **Holder** begins by initiating an application for a specific document using the **Application Website** (applyFor(documentName)).
2. The **Application Website** identifies the required fields for the specified document by querying its system or backend (searchFieldsReq(docName)).
3. The **Application Website** retrieves and provides the list of required fields back to the **Holder** (fieldsRequired).

**2. Filling and Submitting the Details**

1. The **Holder** fills in the necessary details for the application (fillFields(Details)) based on the list of required fields.
2. The filled details are submitted to the **Application Website** (sendInformation(Details)).
3. The **Application Website** forwards these details to the **Server** (sendToDatabase(Details)).
4. The **Server** then stores the details in the **Database** (sendDetails(Details)).
5. The **Database** confirms that the details have been successfully stored by sending back an acknowledgment and a reference ID (sendStoredDatabaseid(ID)).
6. The **Application Website** displays a confirmation message to the **Holder** that the details have been stored successfully (confirmed storage of details).

**3. Payment Process**

1. The **Holder** confirms the details and navigates to the payment page (confirm details and load payment page).
2. The **Holder** provides payment details (Provide Payment Details) and is shown a QR code for payment (provide Payment QR).
3. The **Holder** scans the QR code and pays the required amount (ScanQRAndPay(amt)).
4. The **Application Website** verifies the payment by checking if the correct amount has been deposited (checkAmountDeposited(amt)).
5. Once the payment is verified, the **Application Website** sends a payment confirmation message (provideConfirmation) and allows access to the proof page (ShowConfirmation & load Proof page).

**4. Submission of Proof Documents**

1. The **Holder** uploads or provides the required proof documents (provideProofDocs(docs)).
2. The **Application Website** forwards the proof documents to the **Concerned Department** responsible for verifying and approving the application (sendToConcernedDept(docs)).

**5. Validation by the Department**

1. The **Department** initiates the document verification process:

* It searches for the submitted documents in its system (searchDocs(docs)).
* It queries its database to find the relevant department ID associated with the document type (searchDepartmentIds(deptName)).
* The **Department** fetches the specific database ID for the department responsible for the application (findDBIdFromDept(deptID)).
* Using the database ID, it retrieves the stored documents from its system (findDocs(dbID, docs.id)).

1. The **Department** validates the submitted documents against the stored records (validate(foundDocs.data)) by:

* Checking if the document details are authentic (confirm authenticity).
* Comparing the submitted data with the existing data in the system (compare(foundDocs.data, docs.data)).

**6. Approval Process**

1. If the validation is successful:

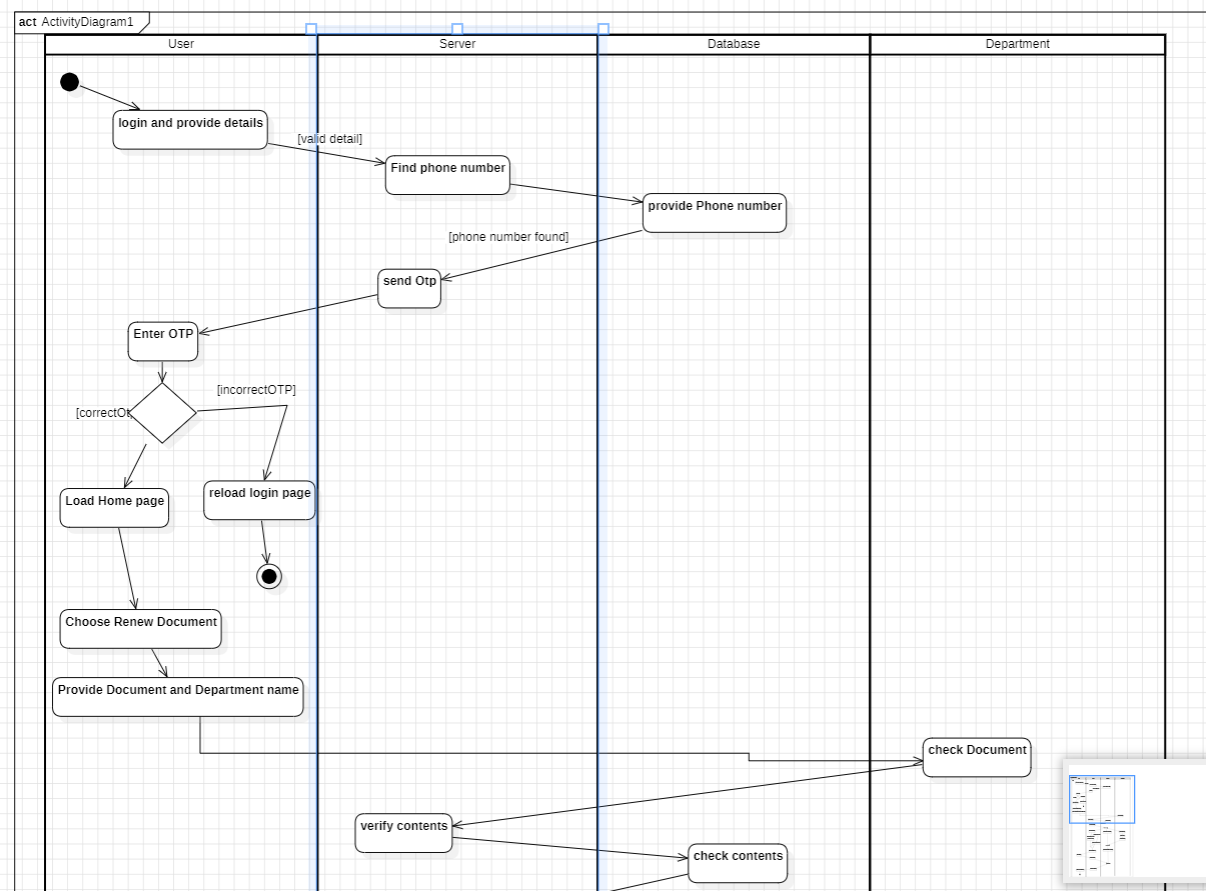
* The **Department** confirms that the submitted data matches the stored data (confirm data is same).
* It approves the application on behalf of the concerned department (Approve(applyingDept)).

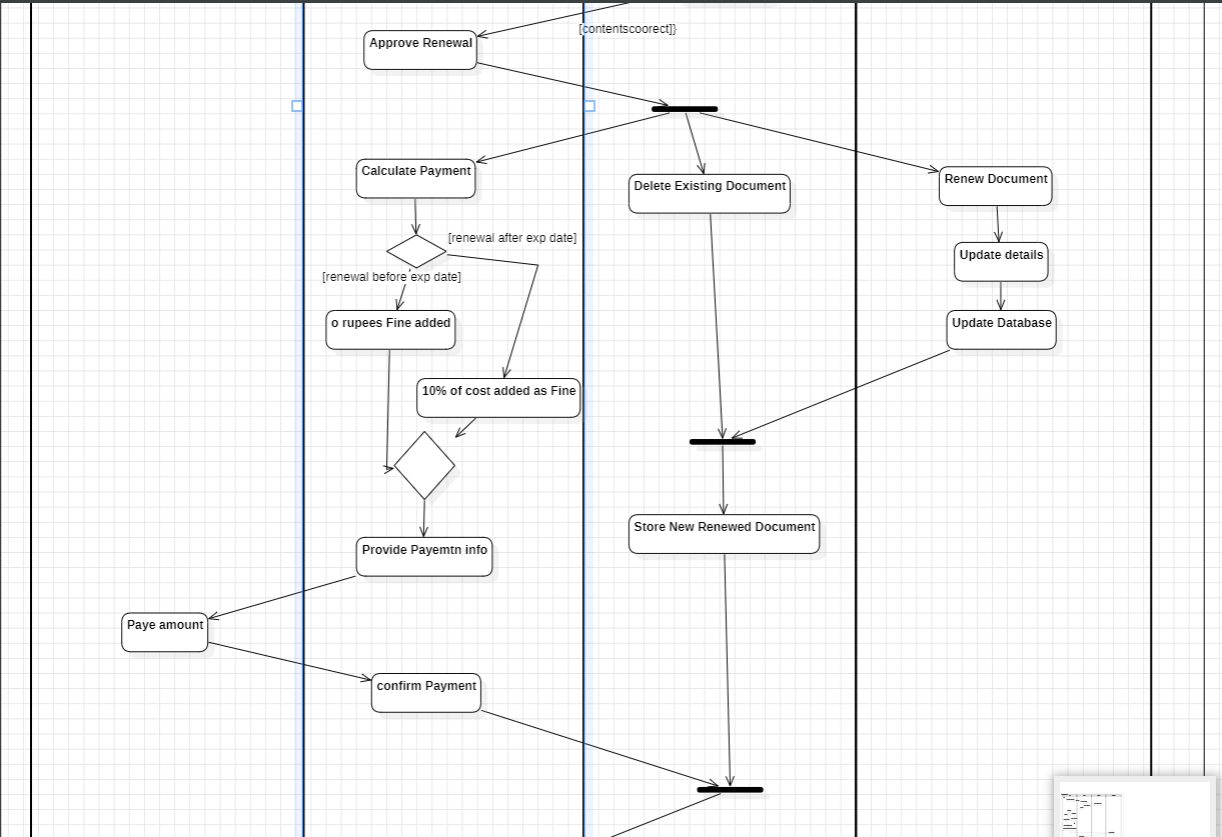
1. A final approval message is sent back to the **Application Website** (approved(docs)).

**7. Final Confirmation and ETA Display**

1. The **Application Website** displays a confirmation page to the **Holder**, indicating that the document application is successful.
2. It also provides an estimated time of arrival (ETA) for the requested document (display confirmation page and show ETA).

## ACTIVITY DIAGRAM

****

****

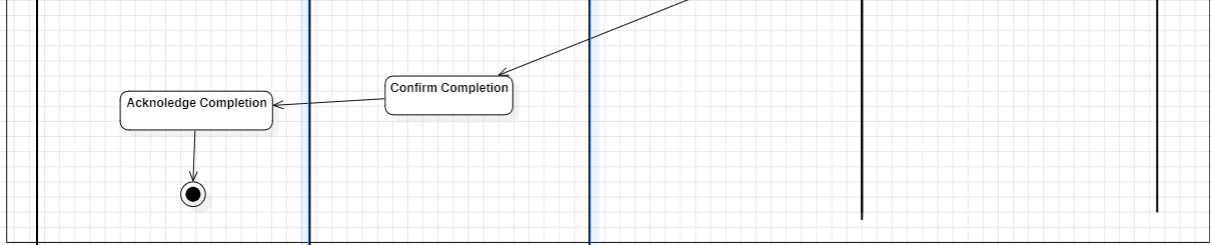
****

Fig 5.4

The diagram represents a **process for document renewal** using a swimlane diagram. Each swimlane represents a specific entity or role involved in the workflow: **User**, **Server**, **Database**, and **Department**. Here's a detailed breakdown of the diagram:

**1. User Login and Authentication**

* The **User** logs in to the system and provides their details.
* The **Server** checks the user's record in the **Database** by verifying the phone number.
* If the phone number is found, the **Server** sends an OTP (One-Time Password) to the user.
* The **User** enters the OTP, and the **Server** verifies it:
  + If the OTP is valid, the system allows access to the **Home Page**.
  + If invalid, the login page is reloaded, and the process restarts.

**2. Document Renewal Selection**

* On the **Home Page**, the **User** chooses the option to renew a document.
* The **User** provides the document name and the department responsible for it.

**3. Document Validation by Department**

* The **Department** checks the document:
  + It validates the existing content and identifies if the document can be renewed.
  + The result of this check is communicated back to the **Server**.

**4. Renewal Approval and Processing**

* If the document is approved for renewal:
  + The **Department** performs the renewal process:
    - Deletes the existing document.
    - Updates the details for the new document.
    - Saves the new document in the **Database**.
  + The **Server** calculates the payment required for the renewal.
* If there is a delay in renewal (beyond a due date), a **fine of 10%** of the renewal amount is added to the payment.

**5. Payment**

* The **User** provides payment details and pays the calculated amount.
* The **Server** confirms the payment by updating the **Database**.

**6. Completion**

* Once the payment is confirmed:
  + The **Server** informs the **User** about the successful renewal.
  + The **User** acknowledges the completion of the process.

**Key Entities and Responsibilities**

1. **User**:
   * Logs in, verifies identity, chooses the renewal option, provides necessary details, and makes the payment.
2. **Server**:
   * Acts as the intermediary, handling communication between the user, database, and department.
   * Manages OTP verification, payment calculation, and payment confirmation.
3. **Database**:
   * Stores user records, verifies phone numbers, stores new document details, and updates payment statuses.
4. **Department**:
   * Validates the document, processes renewal by updating or deleting existing records, and communicates results to the **Server**.

# UI DESIGN

