

# PASSPORT AUTOMATION SYSTEM

## Problem Statement:

Develop a robust and secure Password Automation System to streamline user account management and enhance security for organizations. The system should:

- **Generate Strong Passwords:** Create complex passwords with a mix of characters meeting security best practices.
- **Unique Password Generation:** Ensure each user and application has a unique password.
- **Secure Password Storage:** Store passwords securely using robust encryption methods.
- **Seamless Integrations:** Integrate with existing IAM and other relevant systems.
- **Robust Auditing:** Track all password activities for security and compliance.
- **User-Friendly Interface:** Provide an intuitive interface for easy password management.
- **Compliance Focus:** Adhere to industry standards and relevant regulations.

# SOFTWARE REQUIREMENTS SPECIFICATION

classmate

Date 23/1/24

Page

## Software Requirement Specification (SRS)

### 1.0 Introduction

#### 1.1 Purpose of this Document

The SRS document outlines the detailed requirements for Hotel Management System. It serves as a blue print for the development team ensuring final product meets the specified specification and fulfills the customer requirements.

#### 1.2 Scope of this document

It aims to provide clear specifications for the system's development, including overview of costs and timelines involved. The HMS (Hotel Management System) is designed to provide streamline hotel operations and enhance customer experience.

### 1.3 Overview

The HMS will provide solution for managing hotel operations including reservations, customer management, billings and reportings. This will enable staff to handle daily tasks and offer user friendly interface for bookings.

### 2.0 General Description

#### 2.1 User characteristics - The general user for HMS will be hotel staff, guests and administrators.

### 2.2 Features and Benefits

- o User-friendly Interface
- o Mobile Compatibility
- o Real time availability
- o Automated Reporting

3.2 Importance  
The HMS is crucial for optimizing hotel operations, improving customer satisfaction and increasing revenue through efficient management of resources.

3. Functional Requirements  
3.1 Reservation Management  
• Users can search for available rooms.  
• Users can create, modify, or cancel reservations.

3.2 Customer management  
• System can store guest info, like contact details and preferences.  
• Staff can view guest history.

3.3 Billings and Payments  
• Generate invoices and process payments via various methods.  
• Send automated payment confirmation emails.

3.4 Reporting  
• Generate occupancy reports, revenue reports.  
• Schedule regular reports to be sent to admin.

#### 4. Interface Requirements

- 4.1 User Interfaces  
Web and Mobile interfaces for staff and guest to manage bookings.
- 4.2 System Interface  
Database connection - Interface for the HMS to communicate with database for data storage.

Payment Gateway - Integration with third party payment processors for handling transactions.

#### 5. Performance Requirements

- System should respond to users within 10 seconds.
- Support simultaneous access for 1000 users.
- Maximum error acceptable rate for payments should be less than 1%.

#### 6. Design Constraints

- Must utilize specific set of technologies like Java for backend and React for frontend.
- Use of relational database (e.g. MySQL) for data management.

#### 7. Non-functional

- Security - Implementing user authentication mechanisms.
- Portability - System should operate on various devices.
- Reliability - The system should have 99.9% uptime.
- Scalability - Ability to scale resources on demand.

#### 8. Preliminary schedule and Budget

##### 8.1 Schedule

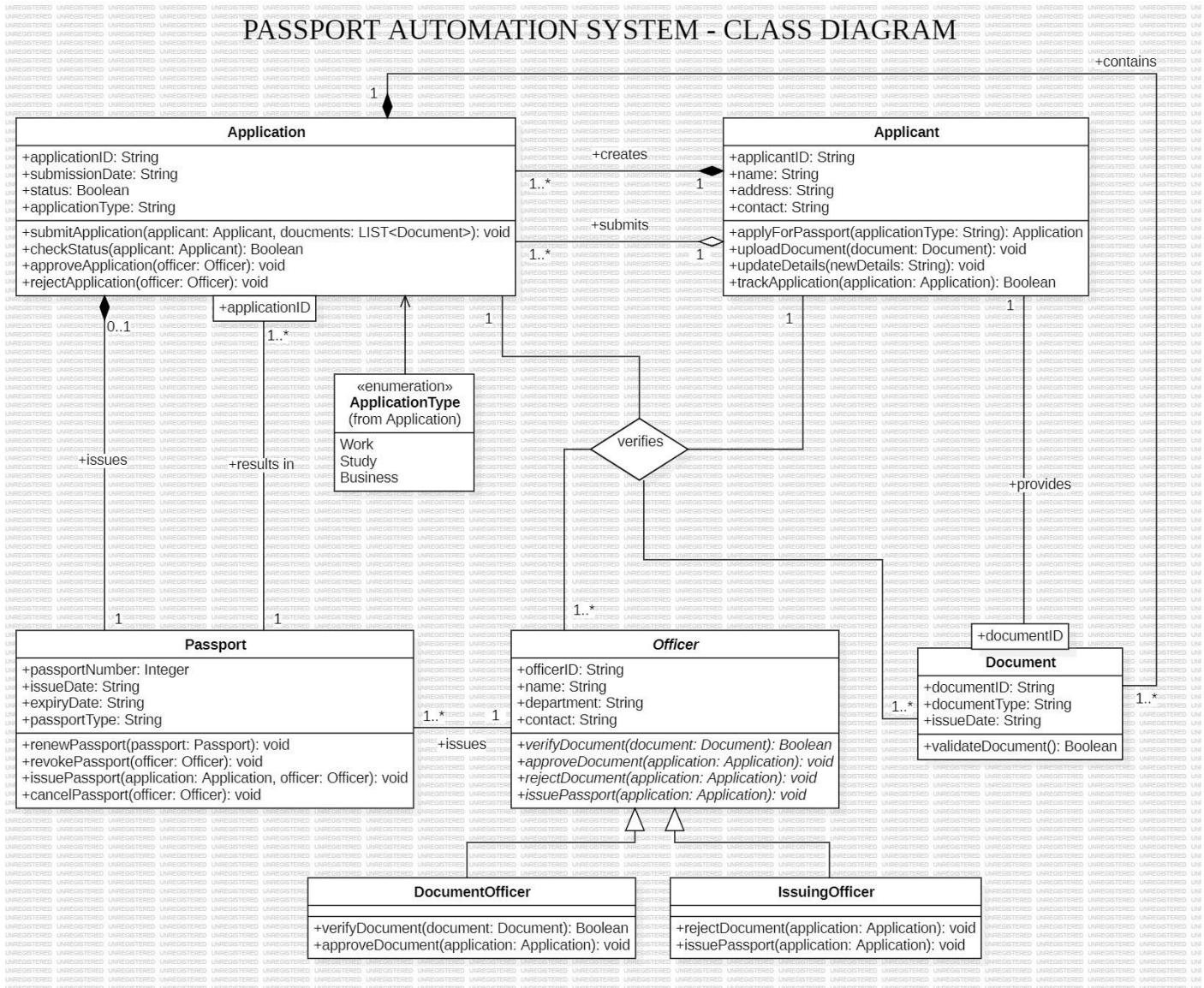
- Project Duration - 6 months
  - Requirements Gathering - 1 month
  - Design phase - 1 month
  - Development phase - 3 months
  - Testing phase - 1 month

##### 8.2 Budget

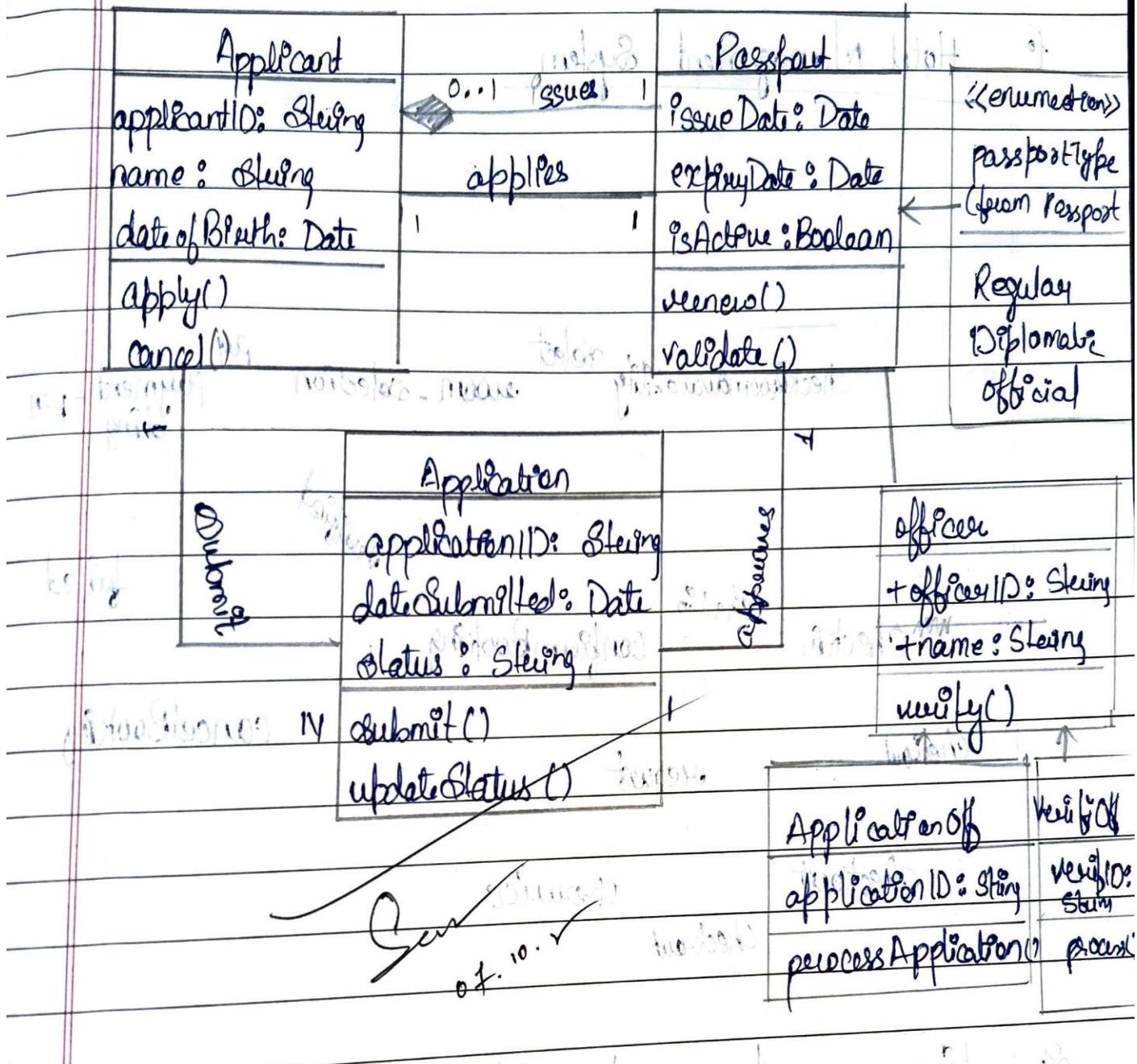
- |                                 |
|---------------------------------|
| Estimated Total cost - 1,50,000 |
| Development - 100,000           |
| Software License - 30,000       |
| Hardware - 15,000               |
| Miscellaneous - 20,000          |

# UML DIAGRAMS

## CLASS DIAGRAM



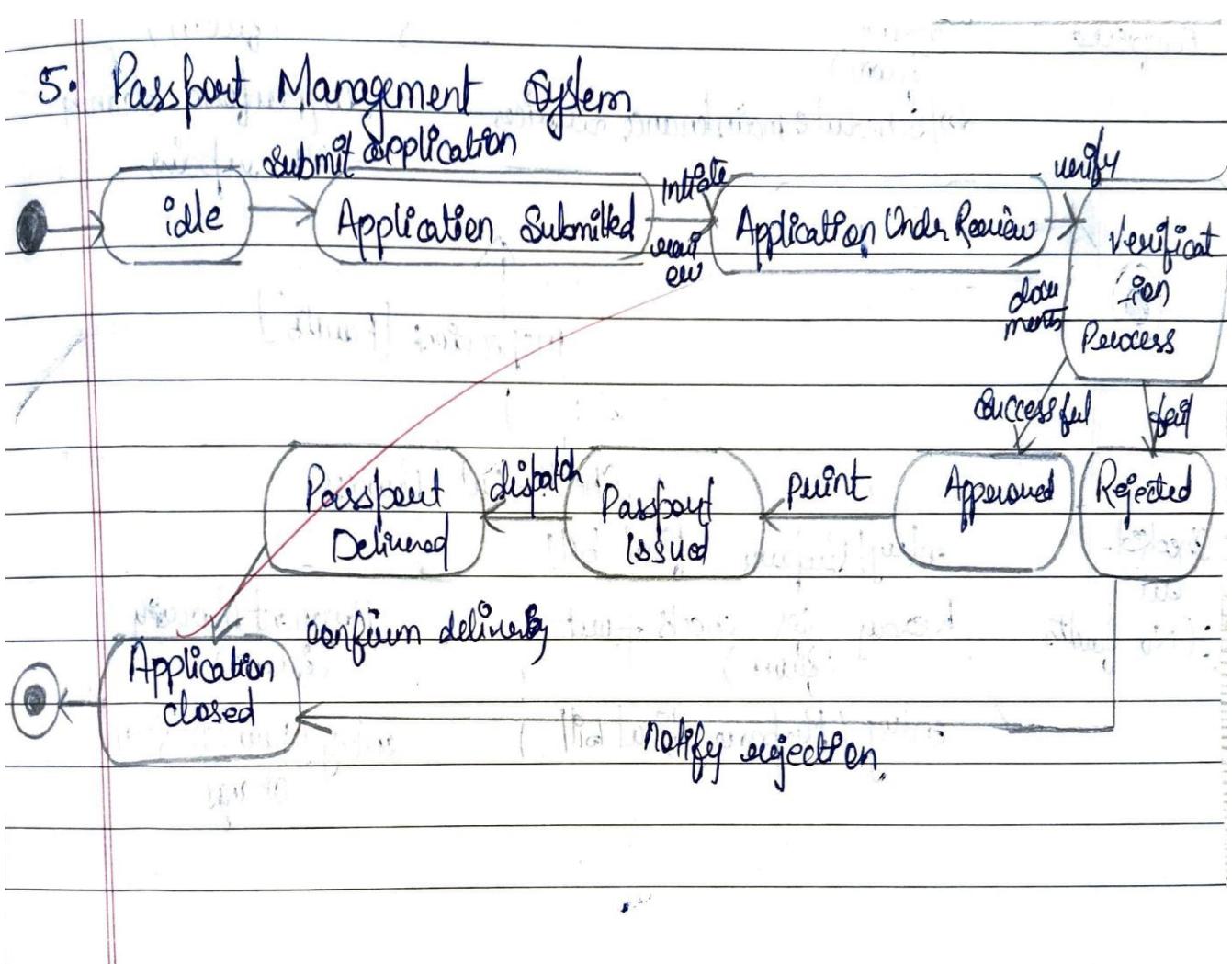
# Passport Automation System



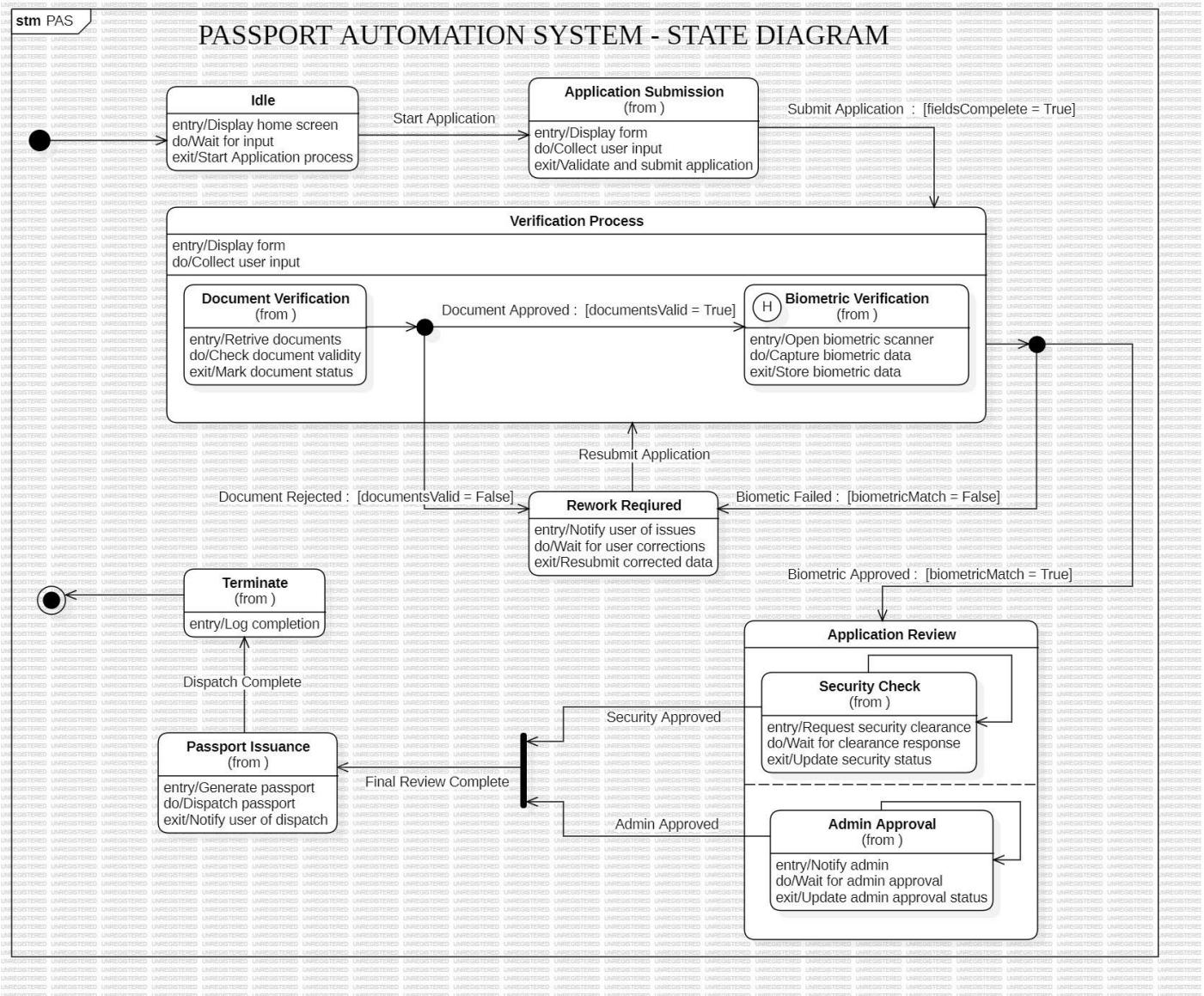
**Brief Description:**

1. **Applicant:** Represents the individual applying for a passport, holding their personal information and application-related methods.
2. **Application:** Represents a passport application with details like status, type, and submission date, including methods for submission, status checks, and approvals.
3. **Passport:** Represents the issued passport document with information like number, issue/expiry dates, and methods for renewal, revocation, and cancellation.
4. **Officer:** Represents a passport officer with information like ID, name, and department, responsible for verifying documents, approving/rejecting applications, and issuing passports.
5. **Document:** Represents a document submitted with the application, storing details like ID, type, and issue date, and a method for validation.
6. **DocumentOfficer:** Specializes in verifying documents, inheriting from the Officer class with additional verification methods.
7. **IssuingOfficer:** Specializes in issuing passports, inheriting from the Officer class with additional issuance methods.
8. **ApplicationType:** An enumeration defining the different types of passport applications (e.g., work, study, business).

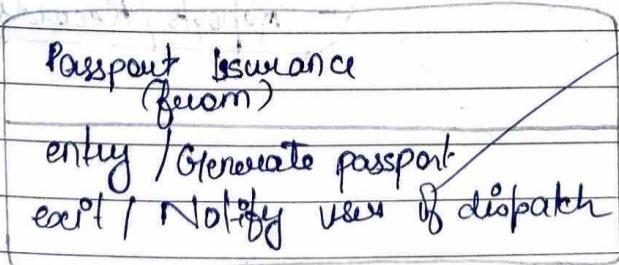
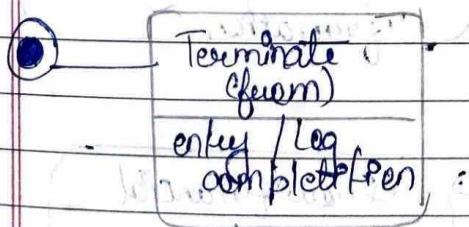
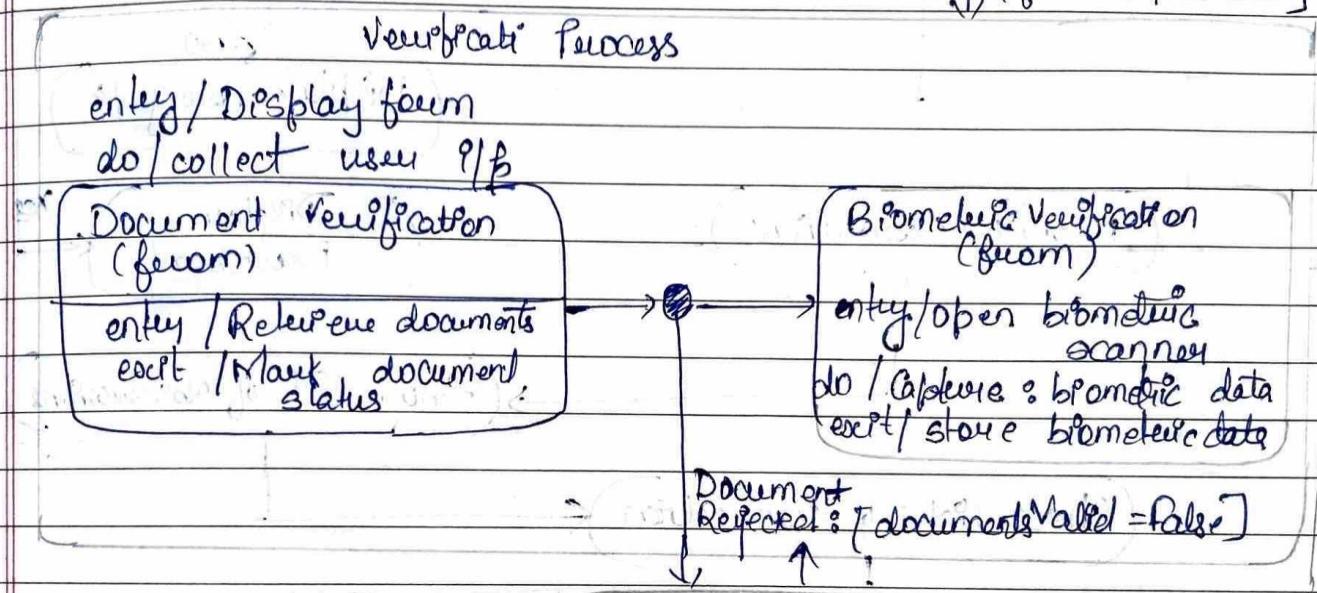
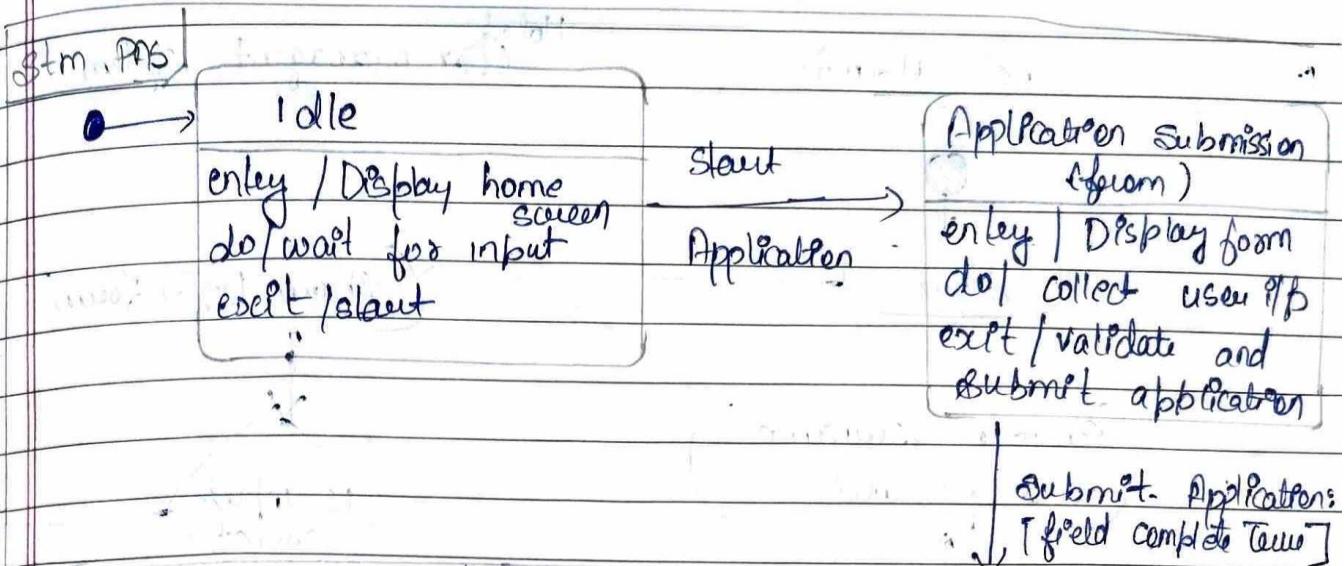
## SIMPLE STATE DIAGRAM



# ADVANCE STATE DIAGRAM



# Passport Automation



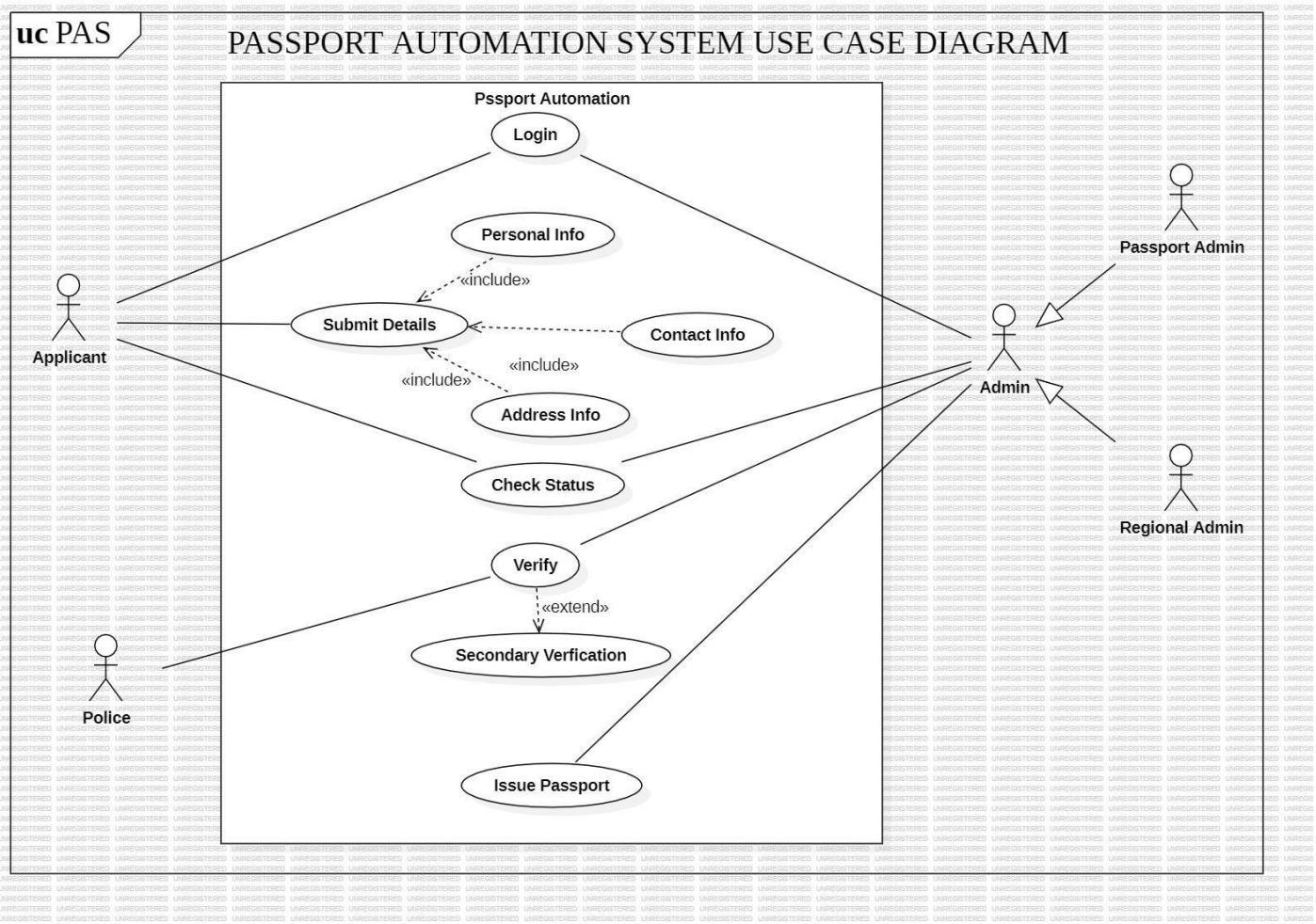
Bio metric  
Paired [bio metric Match = True]

Cur 02/10

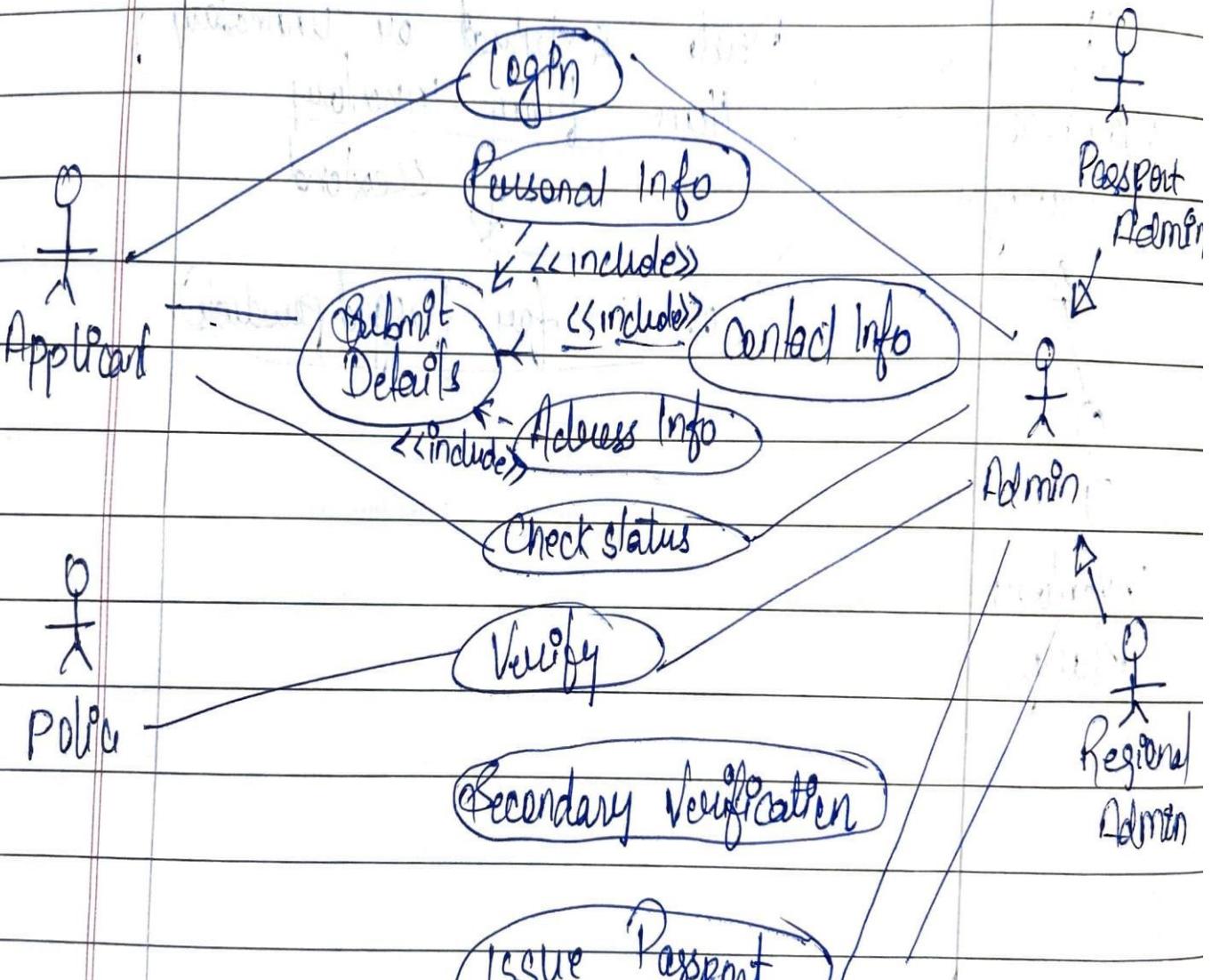
**Brief descriptions:**

- **Start Application:** The system transitions from the Idle state to the Start Application state when the user initiates the application process.
- **Application Submission:** The system transitions from the Start Application state to the Application Submission state when the user submits the completed application form.
- **Document Verification:** The system transitions from the Application Submission state to the Document Verification state to begin checking the validity of the submitted documents.
- **Biometric Verification:** The system transitions from the Document Verification state to the Biometric Verification state if the documents are valid.
- **Rework Required:** The system transitions from the Document Verification or Biometric Verification state to the Rework Required state if the documents are invalid or the biometric data does not match.
- **Terminate:** The system transitions from the Rework Required state to the Terminate state if the user fails to correct the issues or resubmit the application.
- **Dispatch Complete:** The system transitions from the Admin Approval state to the Dispatch Complete state when the passport has been generated and dispatched to the user.
- **Application Review:** The system transitions from the Biometric Verification state to the Application Review state if the biometric data is valid.
- **Security Check:** The system transitions from the Application Review state to the Security Check state to perform security clearance checks.
- **Admin Approval:** The system transitions from the Security Check state to the Admin Approval state for final administrative review and approval.

# USE CASE DIAGRAM



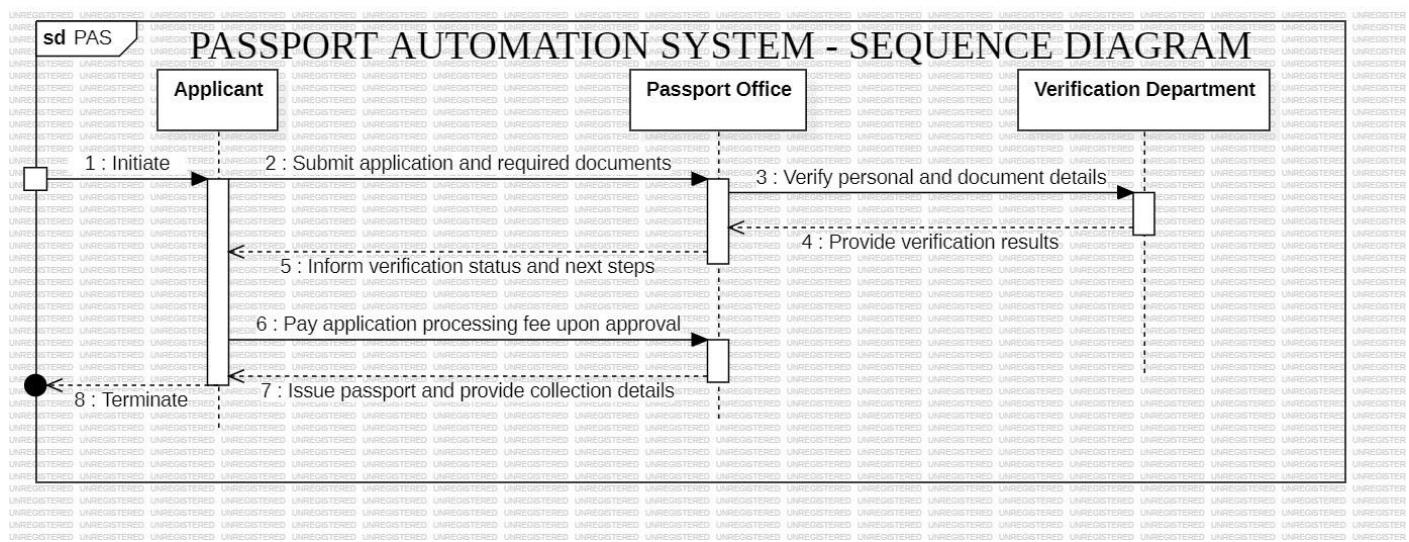
# Passport Automation System



### **Brief descriptions:**

- **Login:** Allows users (Applicants, Admin, Passport Admin, Regional Admin) to authenticate and access the system.
- **Submit Details:** The core use case where applicants provide personal, contact, and address information for their passport application.
- **Personal Info:** Includes entering personal details like name, date of birth, and nationality.
- **Contact Info:** Includes entering contact details like phone number and email address.
- **Address Info:** Includes entering current and permanent address details.
- **Check Status:** Allows applicants to track the status of their passport application.
- **Verify:** The process of verifying the applicant's information and documents.
- **Secondary Verification:** An optional extension of the verification process, potentially involving police checks or background investigations.
- **Issue Passport:** The final step where the passport is issued to the applicant.

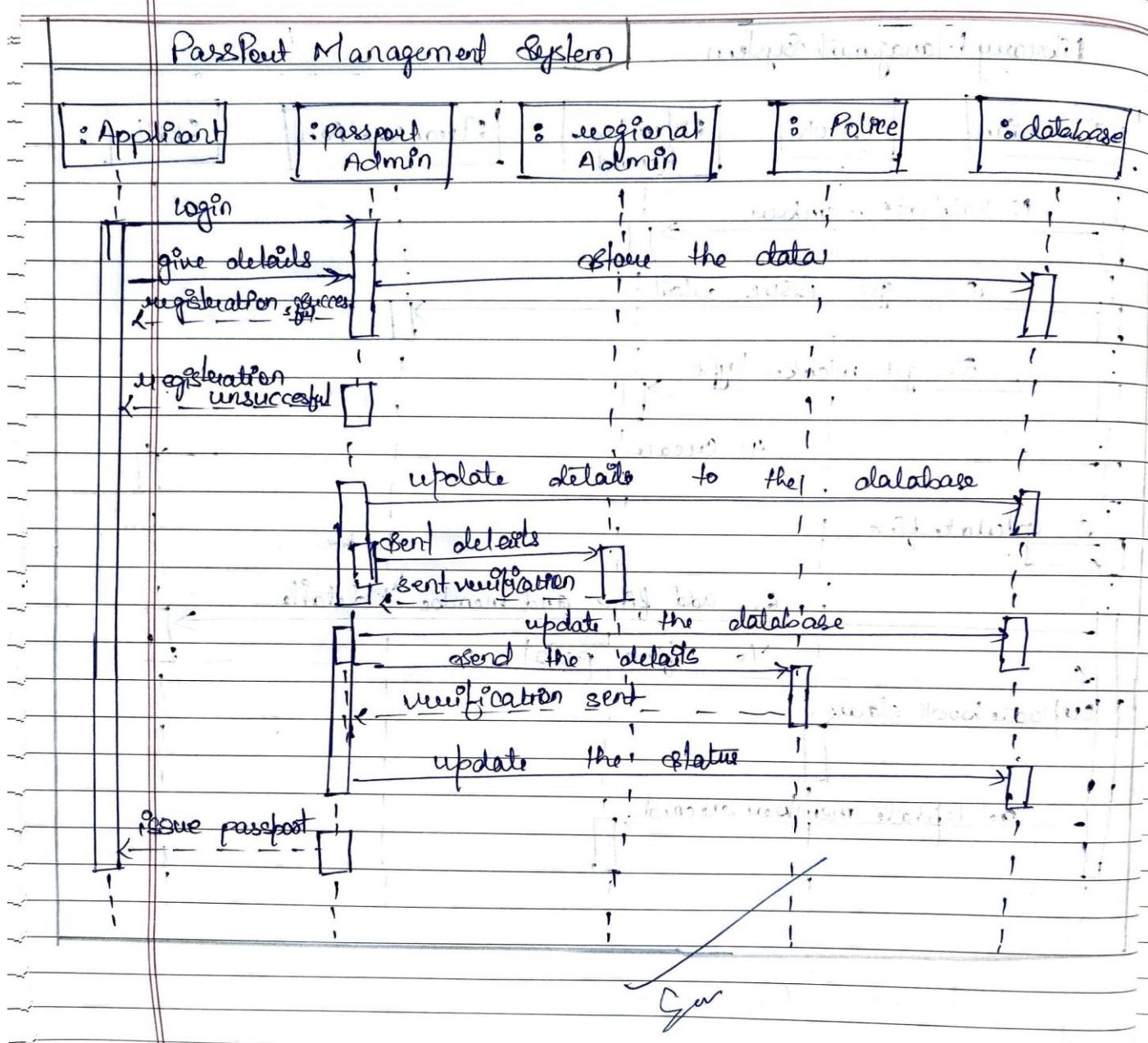
## SEQUENCE DIAGRAM



classmate

Date 16/12/29

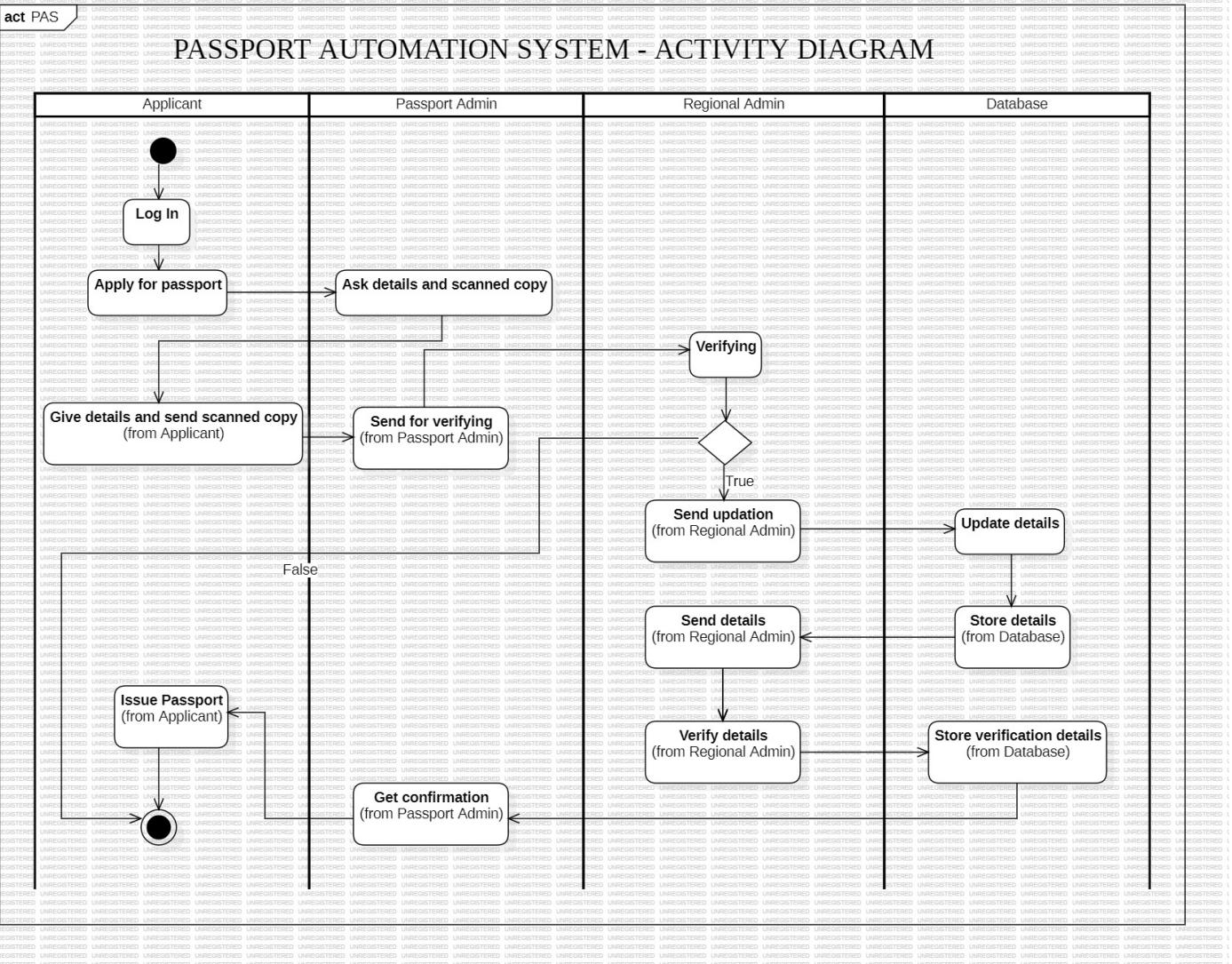
Page \_\_\_\_\_



**Brief descriptions:**

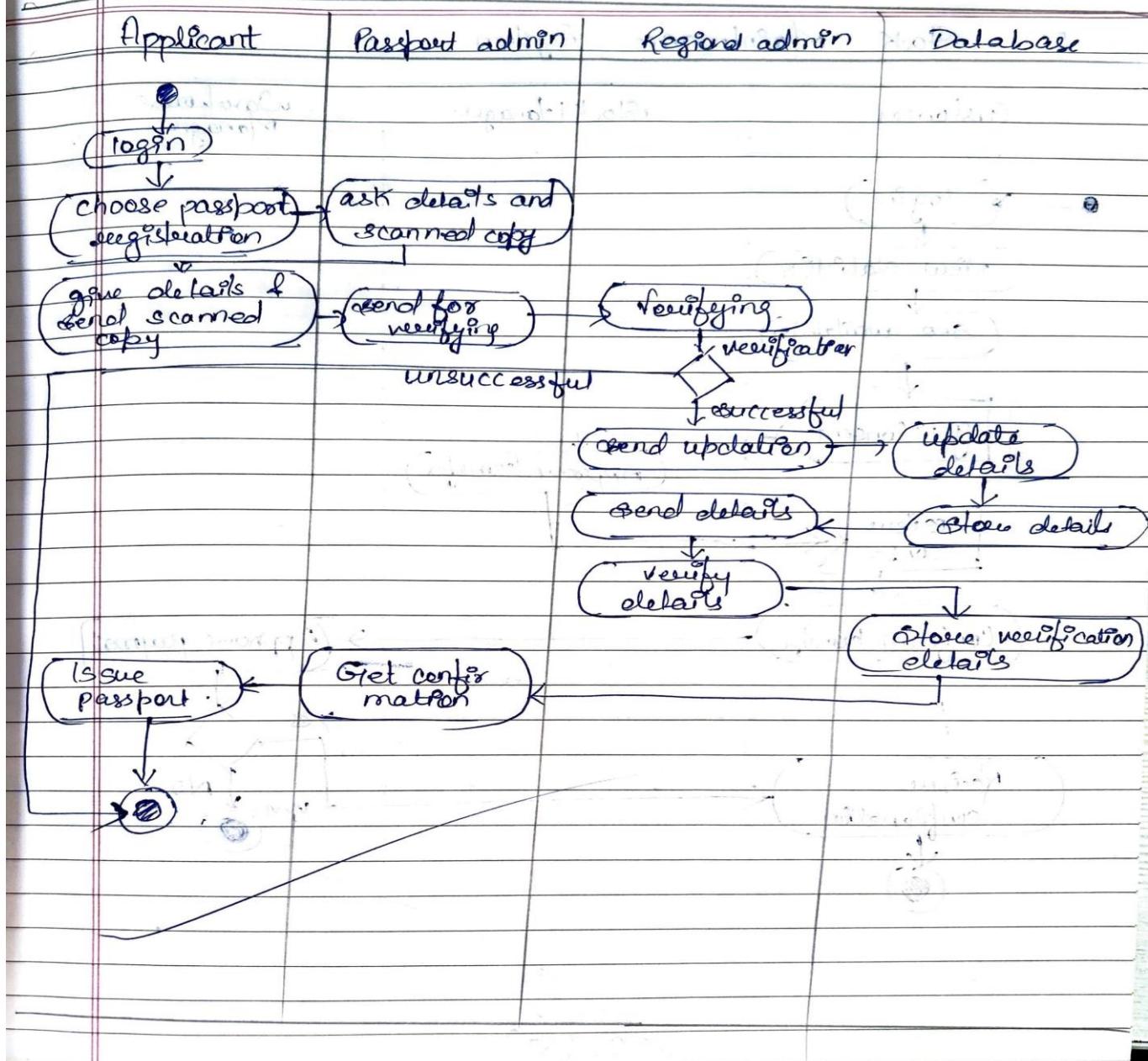
1. **Initiate:** The applicant initiates the passport application process.
2. **Submit application and required documents:** The applicant submits the passport application form and required supporting documents to the Passport Office.
3. **Verify personal and document details:** The Passport Office receives the application and forwards it to the Verification Department for verification.
4. **Provide verification results:** The Verification Department verifies the applicant's personal information and documents and sends the results back to the Passport Office.
5. **Inform verification status and next steps:** The Passport Office informs the applicant about the verification status and the next steps in the process.
6. **Pay application processing fee upon approval:** If the application is approved, the applicant is notified and asked to pay the application processing fee.
7. **Issue passport and provide collection details:** After the fee is paid, the Passport Office processes the application and issues the passport. The applicant is notified about the passport issuance and provided with details on how to collect it.
8. **Terminate:** The application process is completed.

# ACTIVITY DIAGRAM



## 4. Passport Management System

Date \_\_\_\_\_  
Page \_\_\_\_\_



### **Brief descriptions:**

- **Log In:** The Applicant initiates the process by logging into the system.
- **Apply for Passport:** The Applicant applies for a passport.
- **Ask details and scanned copy:** The Passport Admin asks the Applicant for details and scanned copies of required documents.
- **Give details and send scanned copy:** The Applicant provides the details and sends the scanned copies to the Passport Admin.
- **Send for verifying:** The Passport Admin sends the details and documents to the Regional Admin for verification.
- **Verifying:** The Regional Admin verifies the details and documents.
- **Send updatation:** If the verification is successful (True), the Regional Admin sends an update to the Database.
- **Send details:** If the verification is unsuccessful (False), the Regional Admin sends the details back to the Passport Admin.
- **Store details:** The Database stores the updated details or the details returned by the Regional Admin.
- **Verify details:** The Regional Admin verifies the details again.
- **Store verification details:** The Database stores the verification details.
- **Get confirmation:** The Passport Admin receives confirmation from the Regional Admin.
- **Issue Passport:** The Passport Admin issues the passport to the Applicant.