

**THANTHAI PERIYAR GOVERNMENT INSTITUTE OF  
TECHNOLGY, VELLORE-02**



**MINI PROJECT REPORT**

**PASSPORT AUTOMATION SYSTEM**

*Submitted by*

ANUSHIYA K

GAYATHRI M

THENMOZHI A

**In partial fulfilment for the award of the degree of  
BACHELOR OF ENGINEERING**

**IN**

**COMPUTER SCIENCE AND ENGINEERING**

**ANNA UNIVERSITY,**

**CHENNAI - 600 025**

**THANTHAI PERIYAR GOVERNMENT  
INSTITUTE OF TECHNOLOGY  
VELLORE-02**



**BONAFIDE CERTIFICATE**

Certified that this project on **"PASSPORT AUTOMATION SYSTEM"** is the bonafide work of **ANUSHIYA K (513120104002)**, **GAYATHRI M (513120104008)**, **THENMOZHI A(513120104040)** during the academic year 2020-2021 towards the partial fulfilment of the requirement of the award of BE Degree in **COMPUTER SCIENCE AND ENGINEERING** of Anna University, Chennai – 600 025.

**Guided by**

**Mrs.A.ELAVARASI**

Assistant professor, (Adhoc)

Department of CSE.

TPGIT, Vellore-02

**Dr.RAHILA BILAL**

Head of the Department/CSE

TPGIT, Vellore-02

**Internal Examiner**

**External Examiner**

## ACKNOWLEDGEMENT

The satisfaction that accompanies the successful completion of any task would be incomplete without the mention of people whose ceaseless cooperation made it possible, whose constant guidance and encouragement crown all efforts with success.

We give all honour and praise to GOD ALMIGHTY who gave us wisdom and guided us during the entire course of our project.

We express our heartfelt gratitude towards **Dr.M. Arularasu, Principal, Thanthai Periyar Government Institute of Technology, Vellore** for granting us permission to work on this project.

We express our heartfelt gratitude towards **Dr. Rahila Bilal, Head of the Department /CSE, Thanthai Periyar Government Institute of Technology, Vellore** for granting us permission to work on this project.

We wish to place on record our sincere thanks to our Project Coordinator **Prof.N. Jagadeeswari, Assistant Professor/ CSE, TPGIT, Vellore**, for her thoughtful comments and help.

We wish to place on record our sincere thanks to our Project Guide **Mrs.A. Elavarasi, Assistant Professor (Ad hoc)/ CSE, TPGIT, Vellore**, for her thoughtful comments and help.

We also express our gratitude and thanks all other faculty members of the **Department of Computer Science and Engineering, Thanthai Periyar Government Institute of Technology, Vellore**, for their quick help and expert opinions for completing this project.

**ANUSHIYA K [513120104008]**

**GAYATHRI M [513120104002]**

**THENMOZHI A [513120104040]**

## **ABSTRACT TABLE OF CONTENTS**

<b>CHAPTER NO</b>	<b>TITLE</b>	<b>PAGE NO.</b>
	<b>ABSTRACT</b>	<b>1</b>
	<b>LIST OF DEFINITIONS</b>	<b>2</b>
	<b>LIST OF ABBREVIATIONS</b>	
<b>1</b>	<b>INTRODUCTION</b>	<b>3</b>
	<b>1.1 SCOPE OF THE PROJECT</b>	
	<b>1.2 OVERVIEW OF THE PROJECT</b>	
	<b>1.3 PURPOSE OF THE PROJECT</b>	
<b>2</b>	<b>LITERATURE SURVEY</b>	<b>4</b>
<b>3</b>	<b>PROPOSED SYSTEM</b>	<b>5</b>
	<b>3.1 ADVANTAGES</b>	
<b>4</b>	<b>FUNCTIONAL REQUIREMENTS</b>	<b>6</b>
<b>5</b>	<b>FEASIBILITY STUDY</b>	<b>7</b>
	<b>5.1 TECHNICAL FEASIBILITY</b>	
	<b>5.2 ECONOMICAL FEASIBILITY</b>	
	<b>5.3 OPERATIONAL FEASIBILITY</b>	
<b>6</b>	<b>SYSTEM ANALYSIS</b>	<b>10</b>
	<b>6.1 REQUIREMENT SPECIFICATIONS</b>	
	<b>6.1.1 HARDWARE SPECIFICATIONS</b>	
	<b>6.1.2 SOFTWARE SPECIFICATIONS</b>	
	<b>6.1.3 NON-FUNCTIONAL REQUIREMENTS</b>	

### 6.1.4 SYSTEM FUNCTIONS

### 6.1.5 USER CHARACTERISTICS

### 6.1.6 CONSTRAINTS

### 6.1.7 USE CASE MODEL DESCRIPTION

### 6.1.8 ASSUMPTION AND DEPENDENCIES

<b>7</b>	<b>SYSTEM ARCHITECTURE AND DESIGN</b>	<b>14</b>
	<b>7.1 SYSTEM ARCHITECTURE</b>	
	<b>7.2 USE CASE DIAGRAM</b>	
<b>8</b>	<b>IMPLEMENTATION AND TESTING</b>	<b>22</b>
<b>9</b>	<b>OUTPUTS</b>	<b>38</b>
<b>10</b>	<b>CONCLUSION</b>	<b>45</b>
<b>11</b>	<b>FUTURE ENHANCEMENT</b>	<b>46</b>

## **ABSTRACT**

The project “Passport Automation System” is used in the effective dispatch of passport to all of the applicants. This system adopts a comprehensive approach to minimize the manual work and schedule resources, time in a cogent manner.

The core of the system is to get the online registration form (with details such as name, address etc.,) filled by the applicant whose testament is verified for its genuineness by the Passport Automation System with respect to the already existing information in the database.

It aims at improving the efficiency in the Issue of Passport and reduces the complexities involved in it to the maximum possible extent.

## LIST OF DEFINITIONS

- **ADMINISTRATOR** - Refers to the super user who is the Central Authority who has been vested with the privilege to manage the entire system. It can be any higher official in the Regional Passport Office of Ministry of External Affairs.
- **APPLICATION** - One who wishes to obtain the Passport.

## LIST OF ABBREVIATIONS

- **PAS** - Refers to this **Passport Automation System**.
- **HTML - Markup Language** used for creating web pages.
- **HTTP - Hyper Text Transfer Protocol**.
- **TCP/IP – Transmission Control Protocol/Internet Protocol** is the communication protocol used to connect hosts on the Internet.

# **CHAPTER 1**

## **INTRODUCTION**

### **1.1 SCOPE OF THE PROJECT**

The System provides an online interface to the user where they can fill in their personal details and submit the necessary documents (may be by scanning).

The authority concerned with the issue of passport can use this system to reduce his workload and process the application in a speedy manner.

Provide a communication platform between the applicant and the administrator.

Transfer of data between the Passport Issuing Authority and the Local Police for verification of applicant's information.

Users/Applicants will come to know their status of application and the date in which they must subject themselves for manual document verification.

### **1.2 OVERVIEW OF THE PROJECT**

The ideal structure of online passport registration provides security to the passports to be registered wherein we can fill all the details in an efficient and easy manner. A passport is a document, issued by a national Government, for the purpose of international travel, the identity and nationality of its holder. The Elements of identity are name, date-of-birth, sex and place of birth most often, most often nationality and citizenship are congruent.

A passport does not of itself entitle the passport holder entry into another country, nor to consular protection while abroad or any other privileges it does. However normally entitle the passport holder to return to his country that issued the passport. Rights to consular protection arise from international agreements, and the right to return arises from the laws of issuing country. An individual can register for a passport irrespective of his/her age. The registration of passport is a major step for issuing of a passport. It is system or process in which an individual has to provide exact details of his /her personal information and residential information. Proper registration of a passport is very vital as all the detail filled by the individual are depicted on the passport that is issued.



### **1.3 PURPOSE OF THE PROJECT**

If the entire process of 'Issue of Passport' is done in a manual manner then it would take several months for the passport to reach the applicant. Considering the fact that the number of applicants for passport is increasing every year, an Automated System becomes essential to meet the demand. So, this system uses several programming and database techniques to elucidate the work involved in this process. As this is a matter of National Security, the system has been carefully verified and validated in order to satisfy it.

## **CHAPTER 2**

### **LITERATURE SURVEY**

BIBLIOGRAPHY- Software Engineering A practitioner's Approach Sixth Edition - Roger S. Pressman The Unified Modelling Language User Guide The ultimate tutorial to the UML from the original designers - Grady Booch James Rumbaugh Ivar Jacobson Object – Oriented Software Engineering Using UML Pattern and Java Second Edition Bernd Bruegge Allen H. Dutoit [www.rspa.com/spi](http://www.rspa.com/spi) [www.software-engin.com](http://www.software-engin.com) - For Software Engineering [www.springerlink.com](http://www.springerlink.com) - For Object-Oriented Software Engineering [www.ibm.com/software/rational/www.osalt.com/rational-rose](http://www.ibm.com/software/rational/www.osalt.com/rational-rose) [www.brothersoft.com/downloads/rational-rose.html](http://www.brothersoft.com/downloads/rational-rose.html) - For Rational Rose

## **CHAPTER 3**

### **PROPOSED SYSTEM**

#### **3.1 ADVANTAGES**

The development of this new system contains the following activities, which try to automate the entire process keeping in the view of database integration approach. User friendliness is provided in the application with various controls provided by system rich user interface. This system makes the overall project management much easier and flexible. Various classes have been used for maintain the details of all the users and catalogue. Authentication is provided for this application only registered users can access. Report generation features is provided using to generate different kind of reports. In this system the individual fills all the information or details required for passport reservation through online. The individuals have to fill all the details and can't leave any field as blank.

#### **IMPLEMENTATION MODULES:**

1. Admin
2. Consumer (user)

#### **ADMIN MODULE:**

This module provides administrator related functionalities. Administrator can view the all-applicant requirement.

#### **USER MODULE:**

This module is about users of this portal. By using this module user can lodge any complaint about process. User must be registered with the system.

## **CHAPTER 4**

### **FUNCTIONAL REQUIREMENTS**

This application comprises of **six** modules. They are

1. Passport information
2. Apply for passport
3. Document uploading
4. Payment
5. Verification
6. Authentication
7. Status verification and feedback/complaints

#### **1. Passport information:**

Description: This module is helpful for the people to move to other states or countries for their needs. There are 3 ways to apply for the passport. They are -new passport, Atala and Renewal passport.

#### **2. Applying for passport:**

This model is used for how to apply for the passport and which certificates we need to submit for the particular type of passport.

#### **3. Document uploading:**

In this module we have to submit the documents.

#### **4. Payment:**

This model is used to pay the fee in different: ways

#### **5. Verification module:**

In this model different department have to verify all the certification offline and they change the status.

#### **6. Authentication:**

This module is used to check the valid user or not.

#### **7. Status verification and feedback/complaints:**

This module is used to check the status of verification process .

## **CHAPTER 5**

### **FESABILITY STUDY**

Preliminary investigation examines project feasibility, the likelihood the system will be useful to the organization. The main objective of the feasibility study is to test the Technical, Operational and Economical feasibility for adding new modules and debugging old running system. Any system is feasibility if there are unlimited resources and infinite time. There are aspects in the feasibility study portion of the preliminary and investigation,

- 1 Technical Feasibility
- 2 Operational Feasibility
- 3 Economical Feasibility

#### **5.1 TECHNICAL FEASIBILITY:**

The technical issue usually raised during the feasibility stage of the investigation includes the following: -

1. Does the necessary technology exist to do what 5S suggested?
2. Do the proposed equipment's have the technical capacity to hold the data required to use the new system?
3. Will the proposed system provide adequate response to inquiries, regardless of the number or location of users?
4. Can the system be upgraded if developed?
5. Are there technical guarantees of accuracy, reliability, ease of access and data security?

Earlier no system existed to cater to the needs of 'Secure Infrastructure Implementation System'. The current system developed is technically feasible. The database's purpose is to

create, establish and maintain a workflow among various entities in order to facilitate all concerned users in their various capacities or roles specified.

The work for the project is done with the current equipment and existing software technology. Necessary bandwidth exists for providing fast feedback to the users irrespective of the number of users using the system.

## **5.2 OPERATIONAL FEASIBILITY:**

Proposed projects are beneficial only if they can be turned out into information system. That will meet the organization's operating requirements.

Operational feasibility aspects of the projects are to be taken as an important issue raised to test operational feasibility of a project includes the following: -

- 1.Is their sufficient support for the management from the users?
- 2.Will the system be used and worked properly if it is being developed and implemented?
- 3.Will there be any resistance from the user that will undermine the possible application benefits?

This system is targeted to be in accordance with the above-mentioned issues. Beforehand, the management issues and user requirements have been taken into consideration, so there is no question of resistance from the users that can undermine the possible application benefits.

The well-planned design would ensure the optimal utilization of the computer resources and would help in the improvement of performance status.

### **5.3 ECONOMICAL FEASIBILITY:**

A system can be developed technically and that will be used if installed must still be a good investment for the organization. In the economic feasibility, the development cost in creating the system is evaluated against the ultimate benefit derived from the new systems. Financial benefits must equal or exceed the costs.

The system is economically seasick. It does not require any additional hardware or software. Since the interface for this system is developed using the existing resources and technologies, there is nominal expenditure and economical feasibility for certain.

## **CHAPTER 6**

### **SYSTEM ANALYSIS**

#### **6.1 REQUIRMENT SPECIFICATIONS**

##### **6.1.1 HARDWARE SPECIFICATIONS**

The hardware requirement may serve as the basis for a contract for the implementation of the system and should therefore be a complete and consistent specification of the whole system. They are used by software engineers as the starting point for the system design. It shows what the system does and not how it should be implemented.

PROCESSOR	:	Intel Pentium processor 3
RAM	:	4GB RAM
HARD DISK DRIVE	:	1TB
GRAPHICS CARD	:	4GB

##### **6.1.2 SOFTWARE SPECIFICATIONS**

The software requirement document is the specification of the system. It should include both a definition and a specification of requirements. It is a set of what the system should do rather than how it should do it. The software requirements provide a basis for creating the software requirements specification. It is useful in estimating cost, planning team activities, performing tasks and tracking the team's and tracking the team's progress throughout the development activity.

FRONTEND	:	PYTHON
BACKEND	:	SQL
DATABASE	:	MY SQL 5.5
IDE	:	IDLE, PyCharm

### **6.1.3 NON-FUNCTIONAL REQUIREMENTS**

Non-functional requirements describe aspects of the system that are not directly related to the functional behaviour of the system. Non-functional requirements include a broad variety of requirements as given below:

#### **Usability:**

It is the ease with which a user can learn to operate, prepare inputs for, and interpret outputs of a system or component. Often, clients address usability issues by requiring the developer to follow user interface guidelines on colour schemes, logos, and fonts.

#### **Reliability:**

It & the ability of a system or component to perform its required functions under stated conditions for a specified period of time. More recently, this category is often replaced by dependability, which is the property of the computer system such that reliance can justifiably be place on the service it delivers, Dependability includes reliability robustness, and safety.

#### **Performance:**

Performance requirements are concerned with the quantifiable attributes of the system such as response time, throughput, availability and accuracy.

#### **Supportability:**

Supportability requirements are concerned with the ease of changes to the system after deployment, including adaptability, maintainability, and internationalization.



## 6.1.4 SYSTEM FUNCTIONS

Secure Registration of information by the Applicants.

Schedule the applicants an appointment for manual verification of original documents.

Panel for Passport Application Status Display by the Administrator. SMS and Mail updates to the applicants by the administrator.

Administrator can generate reports from the information and is the only.

## 6.1.5 USER CHARACTERISTICS

**APPLICANT**- They are the people who desires to obtain the passport and submit the information to the database.

**ADMINISTARTOR** - He has the certain privileges to add the passport status and to approve the issue of passport. He may contain a group of persons under him to verify the documents and give suggestion whether or not to approve the dispatch of passport.

**POLICE** - He is the person who upon receiving intimation from the PAS, perform a personal verification of the applicant and see if he has any criminal case against him before or at present. He has been vetoed with the power to decline an application by suggesting it to the Administrator if he finds any discrepancy with the applicant. He communicates via this PAS.

## 6.1.6 CONSTRAINTS

The applicants require a computer to submit their information.

Although the security is given high importance, there is always a chance of intrusion in the web world which requires constant monitoring.

The user has to be careful while submitting the information. Much care is required.

### **6.1.7 USE CASE MODEL DESCRIPTION**

Use Case diagrams identify the functionality provided by the system (use cases), the users who interact with the system (actors), and the association between the users and the functionality. Use Cases are used in the Analysis phase of software development to articulate the high-level requirements of the system. The primary goals of Use Case diagrams include:

- Providing a high-level view of what the system does.
- Identifying the users ("actors") of the system.
- Determining areas needing human-computer interfaces.

### **6.1.8 ASSUMPTION AND DEPENDENCIES**

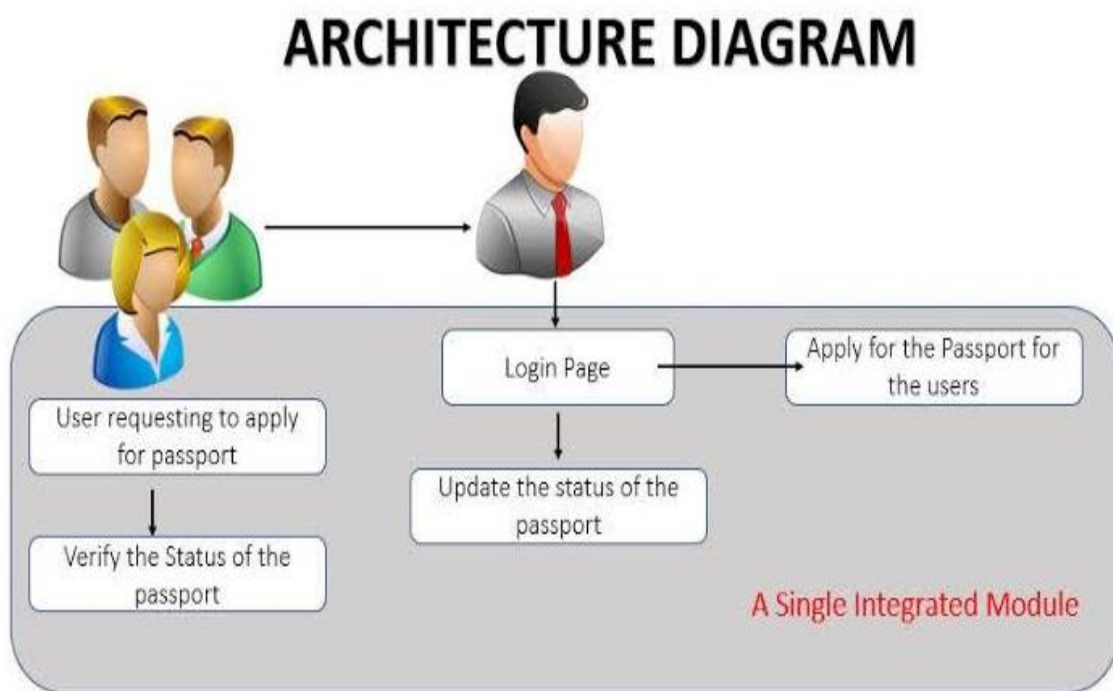
The Applicants and Administrator must have basic knowledge of computers and English Language.

The applicants may be required to scan the documents and send.

## CHAPTER 7

### SYSTEM ARCHITECTURE AND DESIGN

#### 7.1 SYSTEM ARCHITECTURE

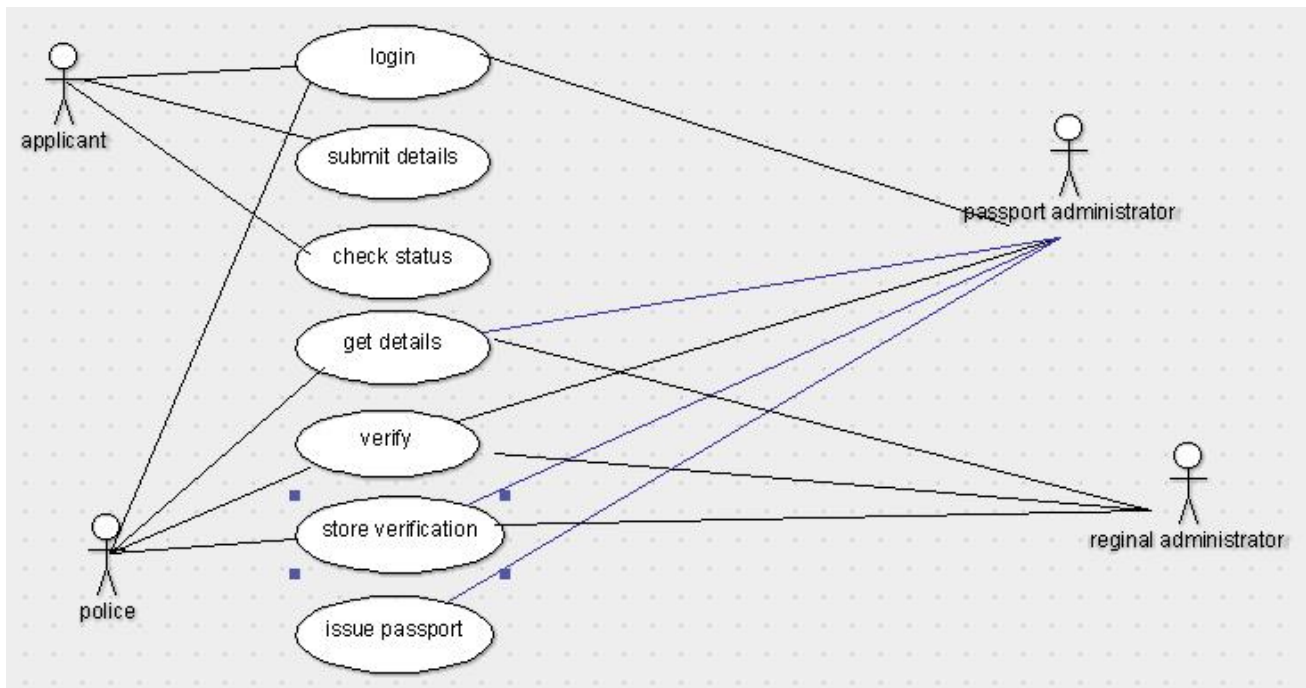


## 7.2 UML DIAGRAMS

### USE CASE DIAGRAM:

To model a system, the most important aspect is to capture the dynamic behaviour. Dynamic behaviour means the behaviour of the system when it is running/operating. Only static behaviour is not sufficient to model a system rather dynamic behaviour is more important than static behaviour. In UML, there are five diagrams available to model the dynamic nature and use case diagram is one of them. The internal and external agents are known as actors. Use case diagrams consists of actors, use cases and their relationships. In brief, the purposes of use case diagrams can be said to be as follows— Used to gather the requirements of a system.

- Used to get an outside view of a system.
- Identify the external and internal factors influencing the system.
- Show the interaction among the requirements are actors.

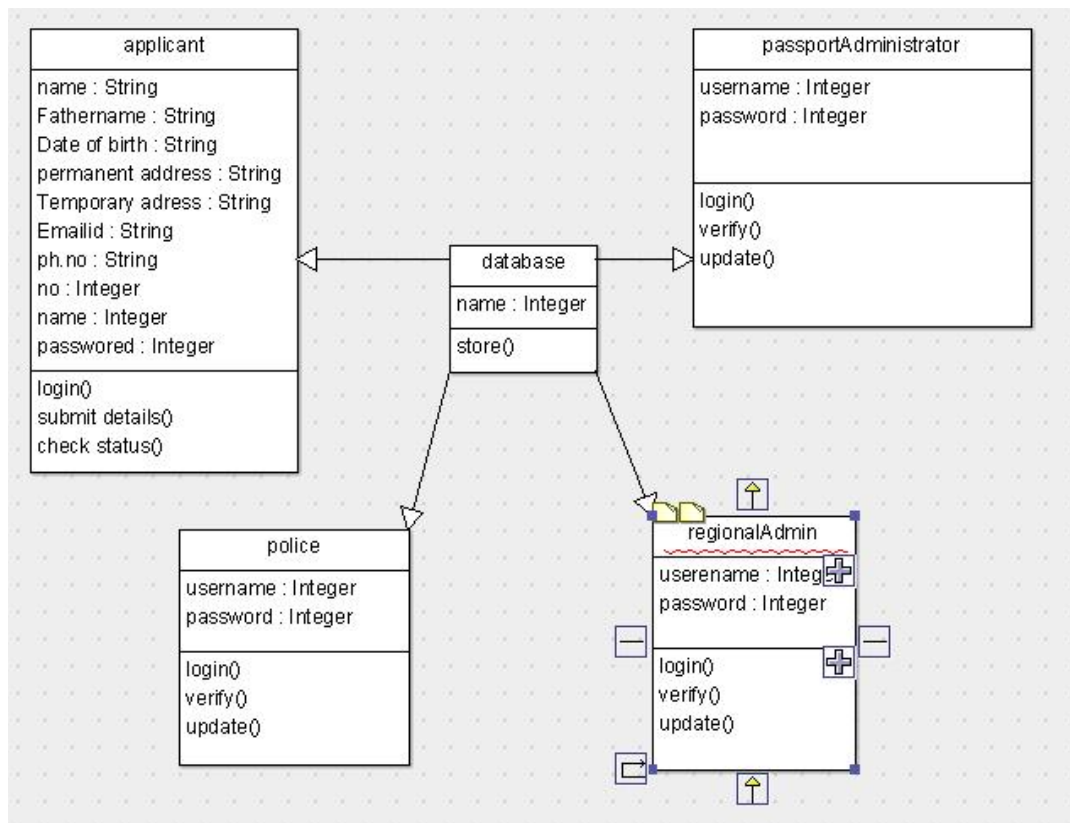


## CLASS DIAGRAM

In software engineering, a class diagram in the Unified Modelling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among objects.

**In the diagram, classes are represented with boxes that contain three compartments:**

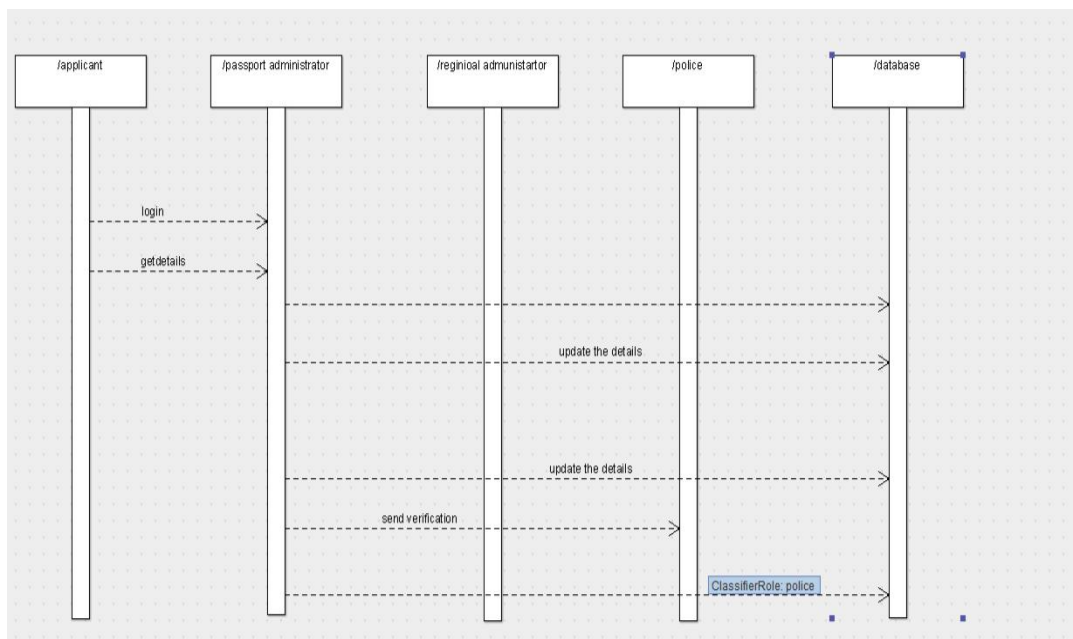
- The top compartment contains the name of the class. It is printed in bold and centred, and the first letter is capitalized.
- The middle compartment contains the attributes of the class. They are left-aligned and the first letter is lowercase.
- The bottom compartment contains the operations the class can execute.
- They are also left-aligned and the first letter is lowercase.



## SEQUENCE DIAGRAM

UML Sequence diagrams are used to represent or model the flow of message, events or action between the objects and components of the system. It is primarily used to design document and validate the architecture, interfaces and logging to the system.

Sequence diagrams provide a dynamic view to the system behaviour which can be difficult to extract from static diagrams or specifications. In the sequence diagrams the time is represented in the vertical direction showing the sequence of interaction of the header element.



## ACTIVITY DIAGRAM

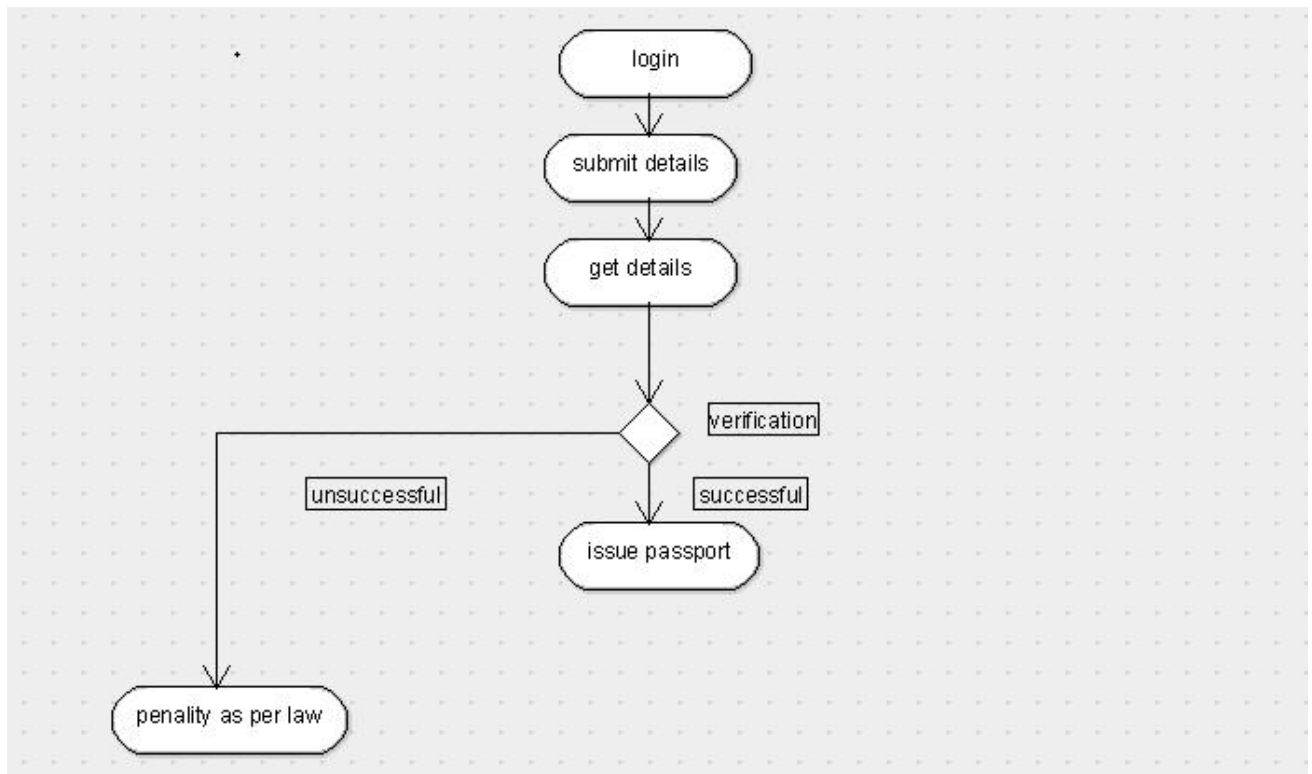
Activity diagram is another important diagram in UML to describe the dynamic aspects of the system. Activity diagram is basically a flowchart to represent the flow from one activity to another activity.

The activity can be described as an operation of the system. The control flow is drawn from one operation to another. This flow can be sequential, branched, or concurrent.

The purpose of an activity diagram can be described as –

Draw the activity flow of a system. Describe the sequence from one activity to another.

Describe the parallel, branched and concurrent flow of the system.



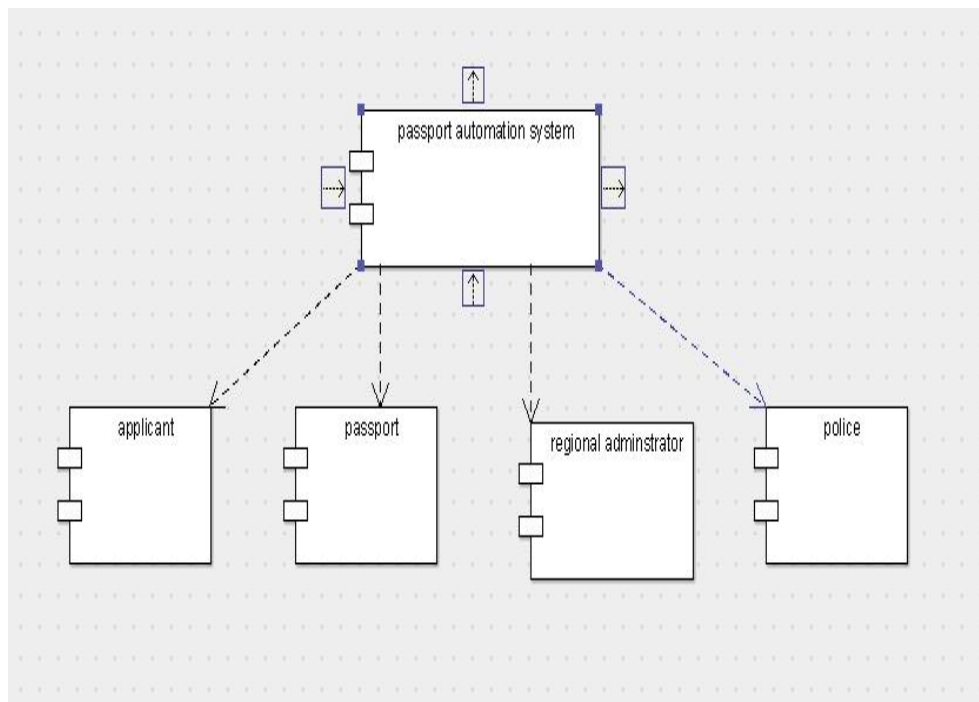
## COMPONENT DIAGRAM

It is used to describe the physical artefacts of the system and implementation perspective. Component diagrams are difficult in terms of nature and behaviour. Because it does not describe the functionality of all systems, but it describes the components used to make those functionalities.

Components diagrams can also be used to describe as static implementation view of the system. Static implementation represents the organization of the components at a particular moment.

**The usages of the component diagram can be described as:**

- Model the components of the system.
- Model database schema.
- Model executable of an application.
- Model systems source code.





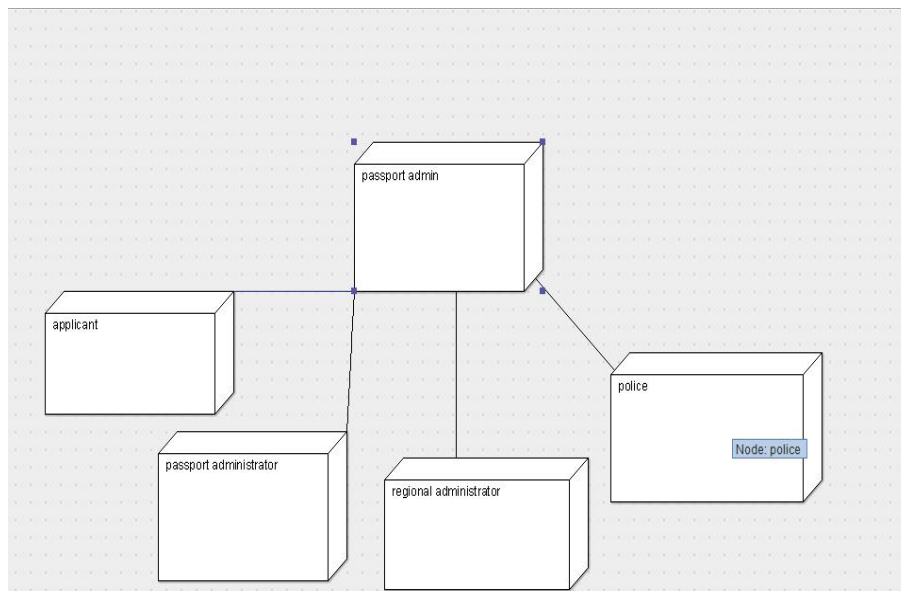
## DEPLOYMENT DIAGRAM

Deployment diagrams are used to visualize the topology of the physical component their distribution and association of a system where the software components are deployed.

Component diagrams are used to describe the component and deployment diagram shows how they are deployed in hardware. UML is mainly designed to focus on software artifacts of a system. But these two diagrams are used to focus on software component and hardware component. The nodes appear as boxes, and the artifacts allocated to each node appear as rectangles within the boxes. Node may have sub nodes, which appear as nested boxes. A single node in a deployment diagram may conceptually represent multiple physical nodes, such as a cluster of database servers.

**The usage of deployment diagrams can be described as follow:**

- To model the h/w topology of a system.
- To model embedded system.
- To model h/w details for a client /server system.
- To model h/w details of a distribution application.
- Forward and reverse engineering.



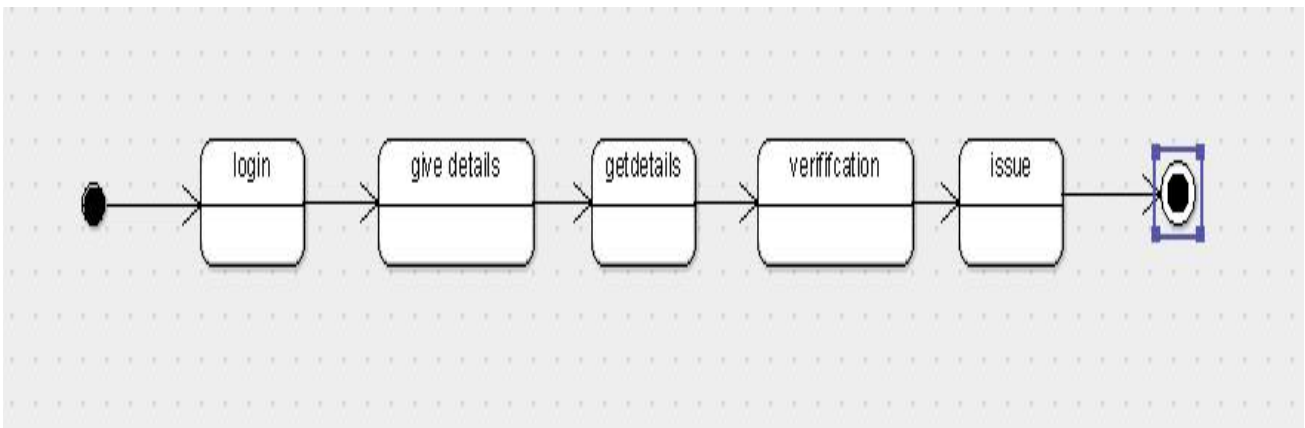
## STATE DIAGRAM

A State chart diagram describes a state machine.

State machine can be defined as a machine which defines different states of an object and these states are controlled by external or internal events. Activity diagram explained in the next chapter, is a special kind of a State chart diagram. As State chart diagram defines the states, it is used to model the lifetime of an object.

**Following are the main purposes of using State chart diagrams:**

- To model the dynamic aspect of a system.
- To model the life time of a reactive system.
- To describe different states of an object during its life time.
- Define a state machine to model the states of an object.



## **CHAPTER 8**

### **IMPLEMENTATION AND TESTING**

#### **ALGORITHM**

**STEP 1:** Provide a user interface for getting the user's login details. Verify login id and if correct, provide access to online passport verification system.

**STEP 2:** If a new user enters and provide a registration.

**STEP 3:** Enter the applicant id and verify the status of the application.

**STEP 4:** Process the application.

**STEP 5:** If the passport is valid, then accept the passport.

## CODE IMPLEMENTATION

### Importing files

```
import sqlite3 from tkinter import *  
from tkinter import messagebox from  
PIL import ImageTk,Image
```

### Selection of Gender

```
def selection():  
    gender = ""  
    selected = int(radio.get()) if  
    selected == 1:  
        gender = "Male"  
    elif selected == 2:  
        gender = "Female"  
    elif selected == 3:  
        gender = "Others" else:  
        gender = "NULL"  
    return gender
```

### Check\_Fields

```
def check_fields():  
    if nameField.get() == "":  
        messagebox.showerror("Failed", "All Fields are Mandatory!") else:  
        return
```

### Details Page

```
def details_page(): top =  
    Toplevel()
```

```
top.title('Details')
top.geometry("1200x1000")
```

```
try:
    conn = sqlite3.connect('db.sql') c =
    conn.cursor()
    c.execute("select * from registration where name = (?)",
              (login_name.get(),)) val =
    c.fetchone() conn.commit()
except Exception as e: print(e)
finally: conn.close()
```

```
label_reg = Label(top, text="REGISTRATION DETAILS:", fg="goldenrod",
font=('Arial Black', 20))
label_reg.place(relx=0.1, rely=0.1)
```

```
label_name = Label(top, text="NAME:", font=('bold', 16))
label_name.place(relx=0.2, rely=0.2)
```

```
fetch_name = Label(top, text=f"{val[1]}", font=('bold', 13))
fetch_name.place(relx=0.3, rely=0.2)
```

```
label_age = Label(top, text="AGE:", font=('bold', 16))
label_age.place(relx=0.2, rely=0.3)
```

```
fetch_age = Label(top, text=f"{val[2]}", font=('bold', 13))
fetch_age.place(relx=0.3, rely=0.3)
```

```
label_dob = Label(top, text="DATE OF BIRTH:", font=('bold', 16))
label_dob.place(relx=0.2, rely=0.4)
```

```
fetch_dob = Label(top, text=f"{val[3]}", font=('bold', 13))
fetch_dob.place(relx=0.4, rely=0.4)
```

```

label_pob = Label(top, text="PLACE OF BIRTH:", font=('bold',16))
label_pob.place(relx=0.2, rely=0.5)

fetch_pob = Label(top, text=f"{val[4]}", font=('bold', 13))
fetch_pob.place(relx=0.4, rely=0.5)

label_gender = Label(top, text="GENDER:", font=('bold',16))
label_gender.place(relx=0.2, rely=0.6)

fetch_gender = Label(top, text=f"{val[5]}", font=('bold', 13))
fetch_gender.place(relx=0.3, rely=0.6)

label_father = Label(top, text="FATHER:", font=('bold',16))
label_father.place(relx=0.2, rely=0.7)

fetch_father = Label(top, text=f"{val[6]}", font=('bold', 13))
fetch_father.place(relx=0.3, rely=0.7)

label_mother = Label(top, text="MOTHER:", font=('bold',16))
label_mother.place(relx=0.2, rely=0.8)

fetch_mother = Label(top, text=f"{val[7]}", font=('bold', 13))
fetch_mother.place(relx=0.3, rely=0.8)

label_address = Label(top, text="ADDRESS:", font=('bold',16))
label_address.place(relx=0.2, rely=0.9)

fetch_address = Label(top, text=f"{val[8]}", font=('bold', 13))
fetch_address.place(relx=0.3, rely=0.9)

```

## Add To Database

```

def add_to_db(): try:
    conn = sqlite3.connect('db.sql') c =
    conn.cursor() c.execute(

```

```

        "insert into registration (name, age, dob, pob, gender, fathename,
mothername, address) values (?,?,?,?,?,?,?)", ( nameField.get(), int(ageField.get()),
dobField.get(), pobField.get(),
        selection(),    fatherField.get(),
        motherField.get(),
        addrField.get(),
        ))
c.execute("select * from registration")
conn.commit() messagebox.showinfo(
    "Success", "Registration Complete"
)
except Exception as e: print(f"Error Has Occured Not able to connect to the
Database :(\n{e}") finally: conn.close()

```

## Login Into Database

```

def login_db(): try:
    conn = sqlite3.connect('db.sql') c =
    conn.cursor()
    c.execute("select * from registration where name = (?)",
        (login_name.get(),)) val = c.fetchone() if login_name.get() == val[1]:
        messagebox.showinfo("Success", "User Logged Successfully")
        conn.commit() details_page()
except Exception as e: print(e)
finally: conn.close()

def login(): login_db()

```

## Registration

```

def register():
    # get the radio button value
    print(selection()) check_fields()
    add_to_db()

```

## New User Registration

```
def new_user(): top =
    Toplevel()
    top.title('Registration') top.geometry("500x500")
    canvase3=Canvas(top,width=800,height=700)

    image3=ImageTk.PhotoImage(Image.open("C:\\Users\\DINESH\\Desktop\\b.jpg"))
    canvase3.create_image(0,0,anchor=NW,image=image3)
    canvase3.pack() global radio radio = IntVar()

    frame1 = Frame(top,bg="skyblue")
    frame1.place(relx=0, rely=0, width=500, height=500)

    registerTitle = Label(frame1, text="REGISTRATION", font=("Arial
        Black", 15), bg="skyblue",fg="black")
    registerTitle.place(relx=0.5, rely=0.03, anchor=CENTER)

    # Name Label and Field global
    nameField
    nameLabel = Label(frame1, text="Name:", font=(
        "Helvetica", 13), bg="skyblue",fg="black") nameLabel.place(x=20,
        y=70)
    nameField = Entry(frame1, font=("Helvetica", 13), bg="white")
    nameField.place(x=150, y=70, width=250)

    # Age Label and Age Field global
    ageField
    ageLabel = Label(frame1, text="Age:", font=(
        "Helvetica", 13), bg="skyblue",fg="black") ageLabel.place(x=20,
        y=110)
    ageField = Entry(frame1, font=("Helvetica", 13), bg='white')
    ageField.place(x=150, y=110, width=250)    # y + 40
```



```

# DoB Label and Field global
dobField
dobLabel = Label(frame1, text="Date of Birth:*", font=(
    "Helvetica", 13), bg="skyblue",fg="black") dobLabel.place(x=20,
y=150)
dobField = Entry(frame1, font=("Helvetica", 13), bg='white')
dobField.place(x=150, y=150, width=250)

# PoB Label and Field global
pobField
pobLabel = Label(frame1, text="Place of Birth:*", font=(
    "Helvetica", 13), bg="skyblue",fg="black") pobLabel.place(x=20,
y=190)
pobField = Entry(frame1, font=("Helvetica", 13), bg='white')
pobField.place(x=150, y=190, width=250)

# gender Label and Field
genderLabel = Label(frame1, text="Gender:*", font=(
    "Helvetica", 13), bg="skyblue",fg="black").place(x=20, y=230)
maleField = Radiobutton(
    frame1, text="Male", variable=radio, value=1, command=selection,
    bg='white').place(x=150, y=230)
femaleField = Radiobutton( frame1, text="Female", variable=radio, value=2,
    command=selection,
    bg='white').place(x=220, y=230)
otherField = Radiobutton(
    frame1, text="Others", variable=radio, value=3, command=selection,
    bg='white').place(x=290, y=230)

# Fathername Label and Field
global fatherField
fatherLabel = Label(frame1, text="Father's Name:*", font=(

```

```

    "Helvetica", 13), bg="skyblue",fg="black").place(x=20, y=270)
fatherField = Entry(frame1, font=("Helvetica", 13), bg='white')
fatherField.place(x=150, y=270, width=250)

# mothername Label and Field global
motherField
motherLabel = Label(frame1, text="Mother's Name:*", font=(
    "Helvetica", 13), bg="skyblue",fg="black").place(x=20, y=310)
motherField = Entry(frame1, font=("Helvetica", 13), bg='white')
motherField.place(x=150, y=310, width=250)

# address Label and Age Field
global addrField
addrLabel = Label(frame1, text="Address:*", font=(
    "Helvetica", 13), bg="skyblue",fg="black").place(x=20, y=350)
addrField = Entry(frame1, font=("Helvetica", 13), bg='white')
addrField.place(x=150, y=350, width=250)

# T&C
tandc = Checkbutton(
    frame1, text="I Agree to the Terms & Conditions", bg='skyblue',
    font=("Calibri", 13), onvalue=1, offvalue=0).place(x=20, y=390)

# Register
registerBtn = Button(frame1, text='Register', font=('Arial Black',
13),fg="green", padx=15, pady=5, cursor="hand2",
    command=register).place(x=20, y=430)

cancelBtn = Button(frame1, text='Cancel', font=('Arial Black', 13),fg="red",
padx=15, pady=5, cursor="hand2", command=top.destroy).place(x=200,
y=430) top.mainloop()

```

## Existing User Login

```
def existing_user(): top =
    Toplevel()
    top.title('Login') top.geometry("900x450")
    canvas2=Canvas(top,width=1300,height=700)

    image2=ImageTk.PhotoImage(Image.open("C:\\Users\\DINESH\\Desktop\\b.jpg"))
    canvas2.create_image(0,0,anchor=NW,image=image2)
    canvas2.pack() head = Label(top, text='LOGIN', font=('Arial
    Black', 15)) head.place(relx=0.5, rely=0.2, anchor=CENTER)

    b_name = Label(top, text='NAME', font=('Arial Black', 14))
    b_name.place(relx=0.1, rely=0.4, anchor=W) global login_name
    login_name = Entry(top)
    login_name.place(relx=0.9,          rely=0.4,          anchor=E)
    login_name.config(width=30)

    b_auth = Label(top, text='PASSWORD', font=('Arial Black', 14))
    b_auth.place(relx=0.1, rely=0.6, anchor=W) global login_password
    login_password = Entry(top,show="*") login_password.place(relx=0.9,
    rely=0.6, anchor=E) login_password.config(width=30)

    bname = Button(top, text="SUBMIT", font=(
        "Arial Black", 10), bg="white",fg="green", command=login)
    bname.place(relx=0.2, rely=0.8, anchor=W) bname.config(height=1,
    width=10)

    bauth = Button(top, text="CANCEL", font=("Arial Black", 10),
        bg="white",fg="red", command=top.destroy)
    bauth.place(relx=0.8,          rely=0.8,          anchor=E)
    bauth.config(height=1, width=10) top.mainloop()
```

## Admin Page

```
def admin(): messagebox.showinfo("ADMIN", " ADMIN PAGE  
ADDED")
```

## Continue Window

```
def continue_window():  
    top = Toplevel() top.title('Passport  
Automation')  
    top.geometry("900x450")  
    canvase=Canvas(top,width=1300,height=700)  
  
    image=ImageTk.PhotoImage(Image.open("C:\\Users\\DINESH\\Desktop\\b.jp g"))  
    canvase.create_image(0,0,anchor=NW,image=image) canvase.pack()  
    head = Label(top, text='USER AUTH', font=('Arial Black', 15))  
    head.place(relx=0.5, rely=0.2, anchor=CENTER)  
  
    bname = Button(top, text="NEW USER", font=(  
        "Arial Black", 10), bg="white", command=new_user, padx=15, pady=10)  
    bname.place(relx=0.1, rely=0.4,) bname.config(height=1, width=10)  
  
    bauth = Button(top, text="EXISTING USER", font=(  
        "Arial Black", 10), bg="white", command=existing_user, padx=15, pady=10)  
    bauth.place(relx=0.4, rely=0.4) bauth.config(height=1,  
width=10)  
  
    adminBtn = Button(top, text="ADMIN", font=(  
        "Arial Black", 10), bg="white", command=admin, padx=15, pady=10)  
    adminBtn.place(relx=0.7, rely=0.4) top.mainloop()
```

## Main Program

```
#=====Main=====
root = Tk()
root.geometry("900x450")    root.title("PASSPORT
AUTOMATION")
canvas=Canvas(root,width=1300,height=700)
image=ImageTk.PhotoImage(Image.open("C:\\Users\\DINESH\\Desktop\\b.jp g"))
canvas.create_image(0,0,anchor=NW,image=image) canvas.pack()
#label=Label(Image=img)
#root.pack()
id = Label(root, text='WELCOME TO THE PASSPORT AUTOMATION SYSTEM',
           font=('Arial Black', 20))
id.place(relx=0.5, rely=0.2, anchor=CENTER)

a = Button(root, text="DO YOU WANT TO CONTINUE", font=(
    "Arial Black", 10), bg="green",fg="white", command=continue_window)
a.place(relx=0.5, rely=0.4, anchor=CENTER)
a.config(height=1, width=30)

b=   Button(root,   text="CANCEL",   font=("Arial   Black",   10),
bg="red",fg="white", command=root.destroy)

b.place(relx=0.5, rely=0.6, anchor=CENTER)
b.config(height=1, width=10)

root.mainloop()
```

## TESTING

Software testing is the process used to assess the quality of computer software, Software testing is an empirical technical investigation conducted to provide Stockholmers with information about the quality of the product or service under test, with respect to the context in which it is intended to operate. This includes, but is not limited to, the process of executing a program or application with the intent of finding software bugs.

Testing is a “fault detection technique that tries to create failures or erroneous states in a planned way”. This allows the developer to detect failures in the system before it is released to the customer.

Testing furnishes a criticism or comparison that compares the state and behaviour of the product against a specification. The software faults occur through the following process. A programmer makes an error (mistake), which results in a defect (fault, bug) in the software source code. If this defect is executed, in certain situations the system will produce wrong results, causing a failure.

There are many approaches to software testing, Reviews, walkthroughs or inspections are considered as static testing, whereas actually running the program with a given set of test cases in a given development stage is referred to as dynamic testing.

Software testing is used in association with verification and validation:

Verification: Have we built the software, right?

Valuation: Have we built the right software?

Software testing can be done by software tester. The development process involves

various types of testing, each test type addresses a specific testing requirement.

### **The most common types of testing involved in the development process are:**

These approaches are used to describe the point of view that a test engineer takes when designing test cases.

#### **Usability Testing:**

Testing that tries to find the faults in the user interface design of the system. Often systems fail to accomplish their intended purpose simply because their users are confused by the user interface and unwillingly introduce erroneous data.

#### **Unit Testing:**

The first test in the development process is the unit test. The source code is normally divided into models, which in turn are divided into smaller units called units. These units have specific behaviour. The test done on these units of code is called unit test.

Unit test depends upon the language on which the project is developed. Unit test ensure that each unique path of the project performs accurately to the documented specifications and contains clearly defined inputs and expected results.

#### **Integration Testing:**

Testing of finding faults when testing the individual components together. Structural Testing is the finale integration testing involving all components of the system. Integration tests and Structural tests exploit knowledge from the SDD using an integration strategy described in the TP (Test Plan).

#### **System Testing:**

Several modules constitute a project. If the project is long-term project, several developers write the modules. Once all the modules are integrated, several errors may arise.

The testing done at this stage is called system test. System testing ensure that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. System testing is based on process descriptions and flown, emphasizing pre-driven process links an integration point.

## **Functional Testing:**

Functional test can be defined as testing two or more modules together with the intent of finding defects, demonstrating that defects are not present, verifying that the module performs its intended functions as stated in the specification and establishing confidence that a program does what it is supposed to do. Functional testing tests the requirements from the RAD and the User manual.

## **Performance Testing:**

Performance test checks the non-functional requirements and additional design goals from the SDD (System Design Document), Functional & Performance testing are done by developers.

## **Acceptance Testing:**

Testing the system with the intent of confirming readiness of the product and customer acceptance. Acceptance testing is the phase of testing used to determine whether a system satisfied the requirements specified in the requirements analysis phase, The acceptance test design is derived from the requirements document. The acceptance test phase is the phase used by the customer to determine whether to accept the system or not.

Acceptance testing and installation testing check the system against the project agreement and is done by the client, if necessary, with help by the developers.

## **Black Box Testing:**

Testing software without any knowledge of the inner workings, structure or Language of the model being tested. It aims to test the functionality according to the requirements. Thus, the tester inputs data and only sees the output from the test object.

## **White Box Testing:**

Testing in which the software tester has knowledge of the inner workings, structure and language of the software, or at least its purpose. White box testing methods include creating tests to satisfy some code coverage criteria (describes the degree to which the source



code of a program has been tested). White Box testing methods can also be used to evaluate the completeness of a test suite that was created with black box testing methods, s

### **Alpha Testing:**

Testing after code is mostly complete or contains most of the functionality and prior to users being involved. Sometimes a select group of users are involved, More often this testing will be performed in-house or by an outside testing firm in co-operation with the software engineering department.

### **Beta Testing:**

Testing after the product is code complete. Betas are often widely distributed or even distributed to the public at large in hopes that they will buy the final product when it is released.

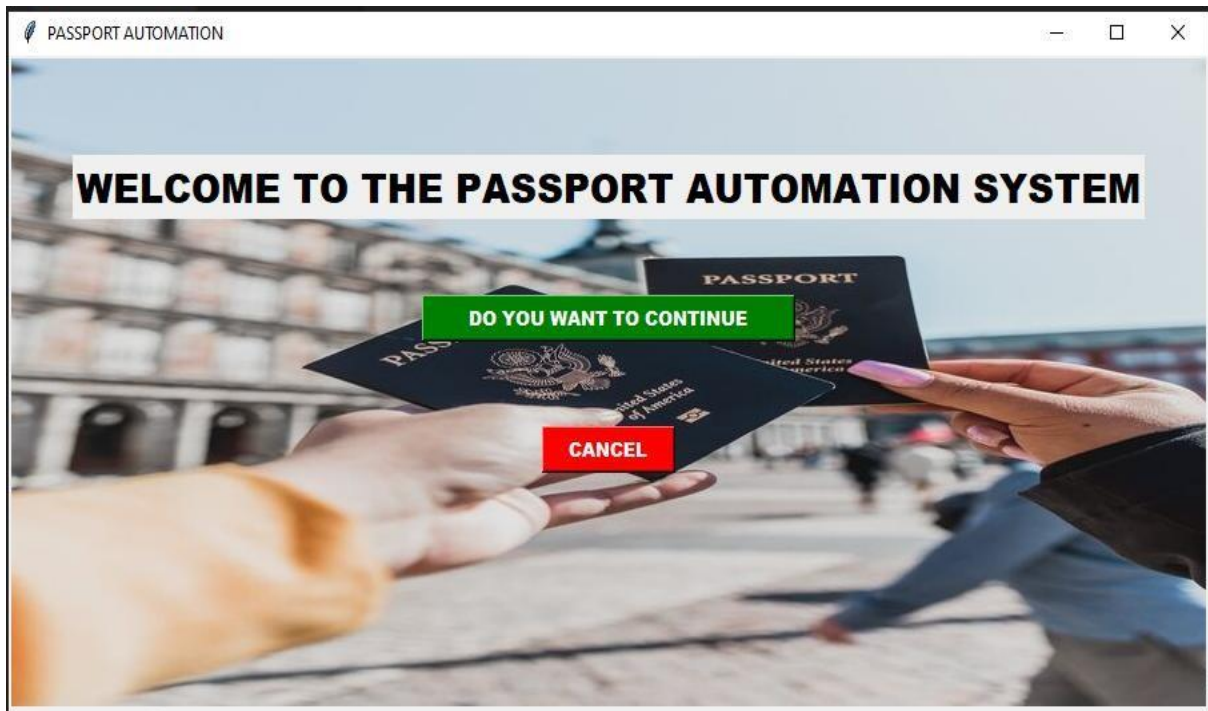
After completing the code generation for each diagram. Sect the tool menu under quality architect performs unit testing and scenario testing for each code.

Testing activities are made several types of testing is carried on.

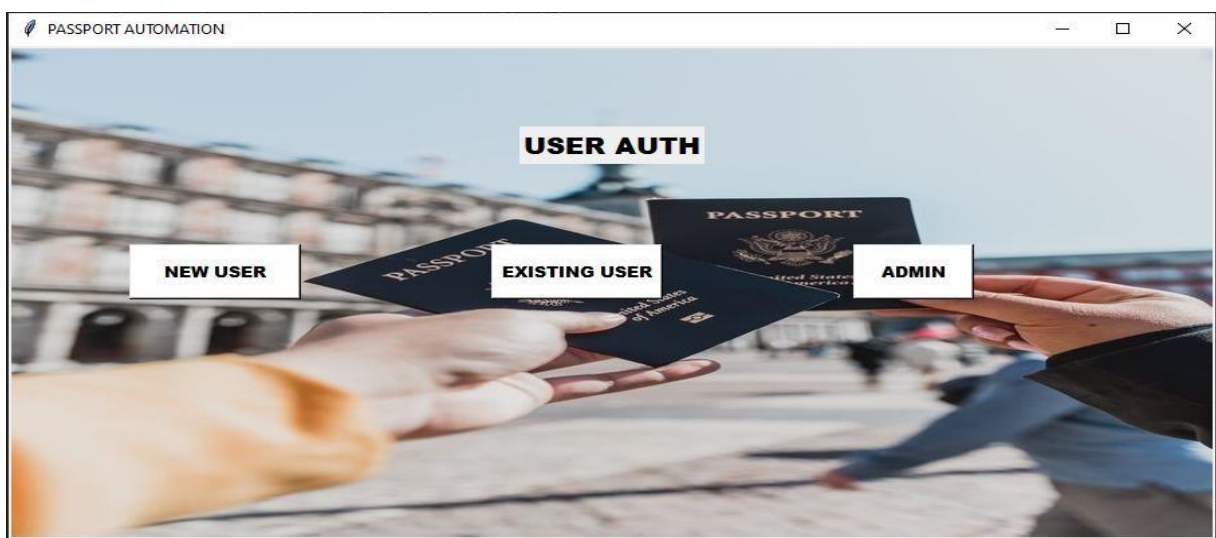
## CHAPTER 9

### OUTPUTS

#### 1. PASSPORT AUTOMATION




#### 2. AUTHORIZATION



### 3.NEW USER

### REGISTRATION

 REGISTRATION

**REGISTRATION**

Name:\*

Age:\*

Date of Birth:\*

Place of Birth:\*

Gender:\*

☐ Male

☒ Female

☐ Others

Father's Name:\*

Mother's Name:\*

Address:\*

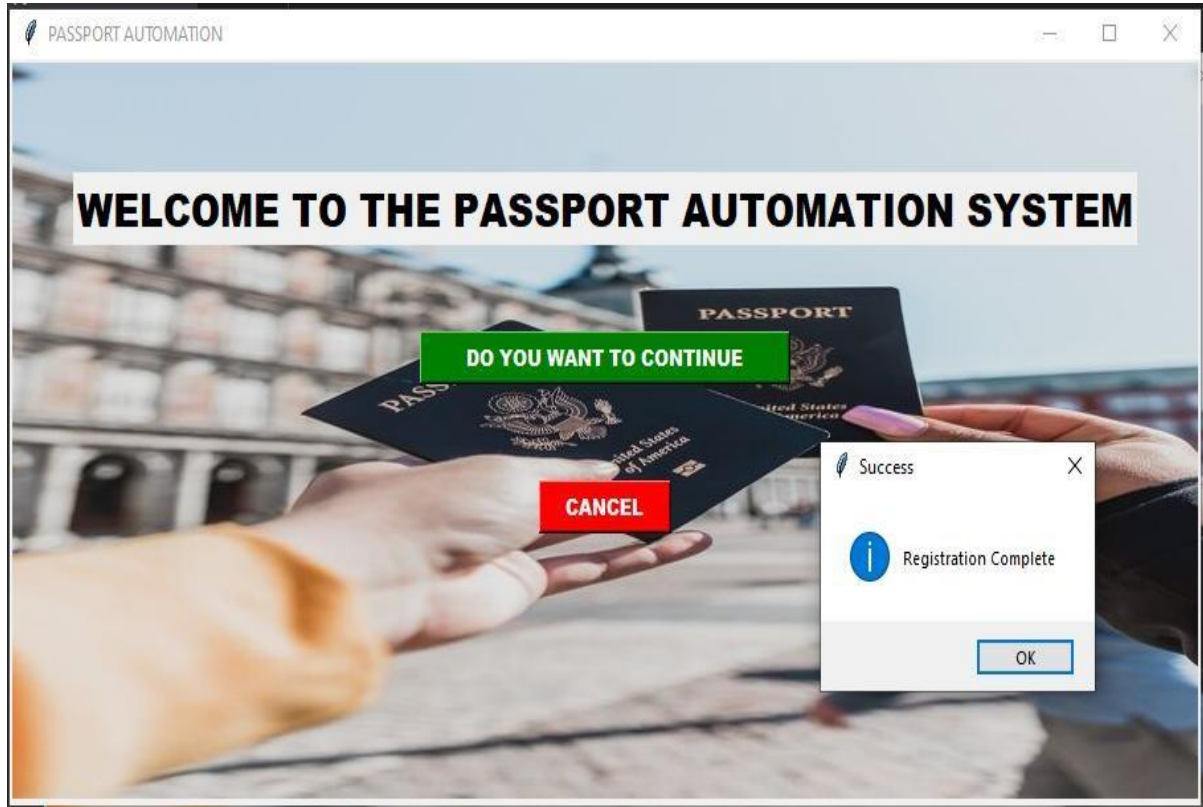
Password:\*

☒ I Agree to the Terms & Conditions

Register

Cancel

## REGISTRATION OF NEWUSER



## MANDATORY FIELD ERROR

REGISTRATION

REGISTRATION

Name:\*

DHARSHINI S

Age:\*

20

Date of Birth:\*

Place of Birth:\*

Gender:\*

Male

Female

Others

Father's Name:\*

Mother's Name:\*

Address:\*

Password:\*

☒ I Agree to the Terms & Conditions

Register

Cancel

Failed

All Fields are Mandatory!

OK

## LOGIN OF EXISTING USER

Login

**LOGIN**

**NAME** DHARSHINI S

**PASSWORD** \*\*\*\*\*

**SUBMIT** **CANCEL**

Login

**LOGIN**

**NAME** DHARSHINI S

**PASSWORD** \*\*\*\*\*

**SUBMIT** **CANCEL**

**Success**

**i** User Logged Successfully

**OK**



## REGISTRATION DETAILS

 Details

— □ ×

### REGISTRATION DETAILS:

NAME: DHARSHINI S

AGE: 20

DATE OF BIRTH: 19/06/2002

PLACE OF BIRTH: SALEM

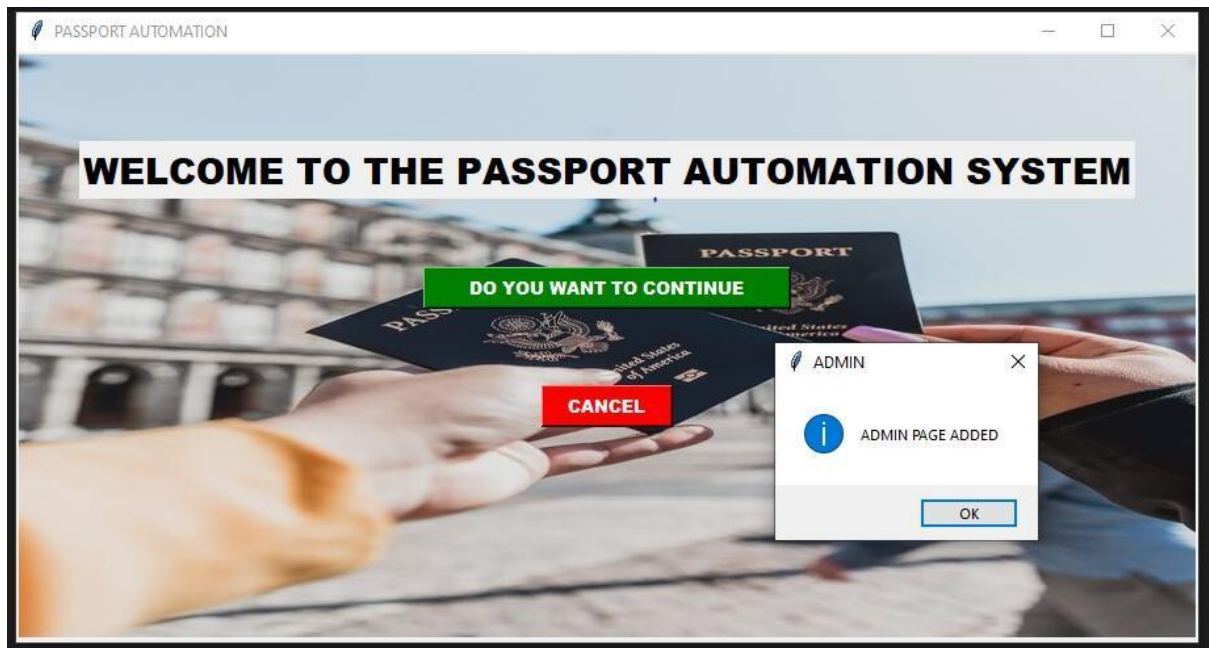
GENDER: Female

FATHER: SAKTHIVEL M

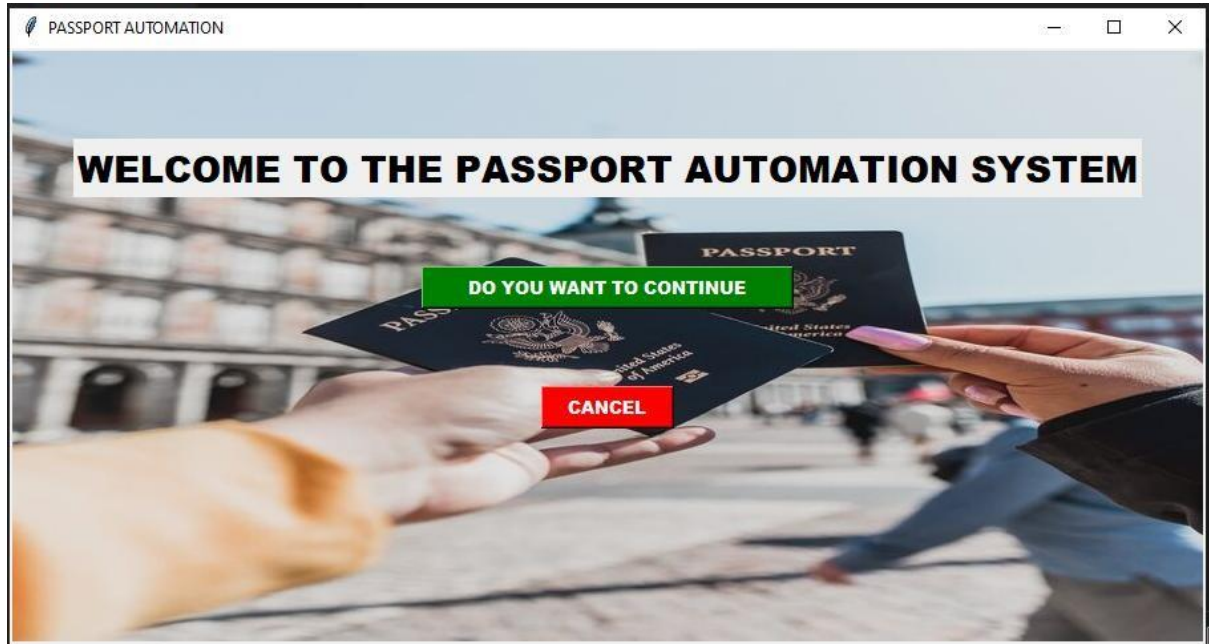
MOTHER: KEERTHI S

ADDRESS: 173,TIRUVALLUVAR STREET,SALEM

## ADDITION OF ADMIN PAGE



## CANCEL





## **CHAPTER 10**

### **CONCLUSIONS**

The main advantage of the Passport Automation System over the existing system is the increase in the speed of information retrieval since that data is maintained systematically and any type of information required for the end user of the system can be generated easily.

Though “Passport Automation System” is so designed that not much amendment is required. This system has provisions for adapting to future enhancements.

To conclude, Passport Automation System is a simple desktop-based application basically suitable for small organization. It has every basic item which are used for the small organization. Our team is successful in making the application where we can update, insert and delete the item as per the requirement. This application also provides a simple report.

This application matches for small organization where there small limited if warehouse. Through it has some limitations, our team strongly believes that the implementation of this system will surely benefit the organization.

## **CHAPTER 11**

### **FUTURE ENHANCEMENT**

The System provides an online interface to the user where they can fill in their personal details and submit the necessary documents (may be by scanning).

The authority concerned with the issue of passport can use this system to reduce his workload and process the application in a speedy manner,

Provide a communication platform between the applicant and the administrator.

Transfer of data between the Passport Issuing Authority and the Local Police for verification of applicant's information.

Users/Applicants will come to know their status of application and the date in which they must subject themselves for manual document verification.