Software Requirement Specification

Tax Payment System

Version 7.0



Course Code: SWE 332

Course Title: Software Engineering project-II

Course Teacher: Asif khan Shakir

Md.Janip Ali Helal 171-35-2046

Md.Fazle Rabbi 171-35-2027

Sadman sakib khan sourav 171-35-1971



Copyright © 2019 by Software Engineering, Daffodil International University. Permission is

required to use, modify, and distribute this document.

Table of Contents

Topic Page

1. Introduction

1.1 Purpose …………………………………………………………………………………………………………….. 4  
 1.2 Documentation Conventions……………………………………………………………………………… 4

1.3 Case Study ………………………………………………………………………………………………………... 5

1.4 Product Scope …………………………………………………………………………………………………… 5

1.5 Glossary ……………………………………………………………………………………………………………. 5

1.6 Overview ………………………………………………………………………………………………………….. 6

2. User Classes & Characteristics ……………………………………………………………………………….. 6

2.1 Area Officer ……………………………………………………………………………………………………... 6

2.2 Circle Officer …………………………………………………………………………………………………….. 6

2.3 Tax Payer ………………………………………………………………………………………………………….. 6

3. Design & Implementation Constraints ……………………………………………………………………. 7

3.1 Operating Environment …………………………………………………………………………………….. 7

3.2 Software Language Used ……………………….………………………………………………………... 7

3.3 Development Tools ……………………….…………………………………………………………………… 7

3.4 Database Support ……………………………………………………………………………………………… 7

4. List of Functional & Non-Functional Requirement ………………………………………………. 8

4.1 Functional requirement ………………………………………………………………………….……….. 8

4.2 Non-Functional Requirement …………………………………………………………………………… 10

5. Use Case ……………………………………………………………………………………………………………. …. 12

6.Use case diagram …………………………………………………………………………………………………… 13

7. Activity Diagram …………………………………………………………………………………….……………… 17

8. Sequence Diagram ……………………………………………………………………………….……………….. 20

9. Class Diagram ………………………………………………………………………………………………………. 23

10. ERD Diagram ………………………………………………………………………………………..……………… 24

*11.UI/UX …………………………………………………………………………………………………………………………………………….. 26*

*11.1 Starting page ………………………………………………………………………………………………………………………… 26*

*11.2 Login from …………………………………………………………………………………………………………………………… 26*

*11.3 Taxpayer home page ……………………………………………………………………………………………………………. 27*

*11.4 Tax Calculation from ……………………………………………………………………………………………………………. 27*

*11.5 Taxpayment from ………………………………………………………………………………………………………………… 28*

*11.6 Assessment Report ……………………………………………………………………………………………………………… 28*

*11.7 Admin dashboard ……………………………………………………………………………………………………………….. 29*

*13. Github ……………………………………………………………………………………………………………………………………… 30*

*14. Mailtrap …………………………………………………………………………………………………………………………………… 30*

**1. Introduction**

**1.1 Purpose**

The purpose of this document is to describe all the requirements for the targeted Tax payment Govt. of Bangladesh. The intended audience includes all the Citizen of Bangladesh in the potential system. These include, but are not necessarily limited to, the following: Government Employee, Tax officer, Circle, Officer, Tax payer.

Developers should consult this document and its revisions as the only source of requirements for the project. They should not consider any requirements statements, written or verbal as valid until they appear in this document or its revision.

The Government Employee, Tax officer, Circle, Officer, Tax payer should use this document and its revisions as the primary means to communicate confirmed requirements to the development team. The development team expects many face-to-face conversations that will undoubtedly be about requirements and ideas for requirements. Please note that only the requirements that appear in this document or a future revision, however, will be used to define the scope of the system.

**1.2 Documentation Conventions**

This SRS is divided up into sections detailing an overall description, the external interface requirements, system features, and other non-functional requirements. As this is the final draft, any future modifications of this document would involve adapting the product to changing systems and uses. We hope to have the product evolve to changing times as to ensure continued use and success. The Document and Specification team have prepared the overall information in this document to the best of their ability. Once read, it is evident that each section is important to the overall SRS and significant to the project in its own right.

**1.3 Case Study**

According to National Board of Revenue (NBR) data, as of June 2018, there are about 3.5 million tax identification number (TIN) holders, of which about 1.95 million submitted tax returns. However, this does not mean that those who have paid tax have not evaded tax.

**1.3 Gap**

Tax evasion can have many forms. In Bangladesh, businessmen who collect value added tax from consumers also evade tax by under-reporting the same. Importers avoid tax by under-invoicing. Tax avoidance is thus a major problem.

**1.4 Product Scope**

The proposed software product is Tax Payment System for Citizen. It will be used to maintain various Tax payer records such as Submit Income Tax Return, Tax Calculation, Tax Payment, Audit and Report.

The system is expected to

* Ensure better revenue distribution,
* Efficient collection of revenue,
* User-friendly service,
* Fraud detection,
* Higher rate of collection,
* Standard solutions for all tax types in whole tax system,
* Integrated solution for all revenue types,
* Significant decrease of costs and faster tax collection.

**1.5 Glossary**

Here there are some clarifications of the terms uses in this document and also some explanation related to Tax Payment System (TPS)

**Terms Definition:** A Citizen must register to use this system, they can add their whole information of their properties and make a list of their legal properties like Gold, Other valuable jewelry. The system automatically calculates the tax over the citizen’s properties, so the citizen can view how much tax he/she have to pay.

On the other side the circle officer can see the Assessment report of the citizen and submit the from to Area officer.

And the Area officer can see the citizen and Circle officer audit report and give the tax clearance.

**1.6 Overview**

This Software Requirements Specification (SRS) specifies all the requirements for TPS. Various techniques such as interviews, brain storming and idea reduction, use cases and prototyping were used to elicit the requirements and we have identified the needs, analyzed and refined them. The objective of this document therefore is to formally describe the system’s high-level requirements including functional requirements, non-functional requirements and constraints.

**2. User Classes & Characteristics**

**2.1 Area Officer**

In the TPS the major and sensitive role is played by the Area Officer, in TPS this role is played by the Main character. The Area officer will check and validate all the process of Tax payment. Audit Circle officer work and tax payer details report.

**2.2 Circle Officer**

In the TPS the secondary important role plays by circle officer, He can check the report of tax payer and audit tax payer properties information.

**2.3 Tax Payer**

The role of a tax payer is also a most important role because, it’s the main primary actor of our system. A taxy payer can register in the system and input validated information for calculate how much max he/she have to pay. And see his/her report, and track his/her tax payment process in home. He /She can payment the tax using online mobile banking system or credit card.

**3. Design & Implementation Constraints**

**3.1 Operating Environment:**

The TPS will be web-based system. Thus, anyone having a browser can hit the specific link and can get access to it. Thus, it will ensure its best usage and will ease the means of getting access to the system. Moreover, it will remove the complexities of running the system in multiple platforms as it will be deployed in a web server.

**3.2 Software Language Used:**

The application will be developed using Laravel 6.0 Framework. The used language will be PHP and the front end will be developed using Laravel MVC. Besides for eye soothing user interface experience CSS 4 will also be used.

**3.3 Development Tools:**

For the development purpose ATOM edition will be used. For handling different database operations XAMPP MySQL server will be used.

**3.4 Database Support:**

The database that will be used is XAMPP server. Entity framework 4.1 will be used from the applications end to insert, update and delete the data.

# 4. List of Functional & Non-Functional Requirement

## **4.1 Functional requirement**

## **Functional requirement: Tax Payer**

|  |  |
| --- | --- |
|  |  |
| **Requirement ID** | FR.TP.1 |
| **Requirement Name** | Log in |
| **Description** | Tax Payer can log in using user name and password |

|  |  |
| --- | --- |
|  |  |
| **Requirement ID** | FR.TP.2 |
| **Requirement Name** | Calculate Tax |
| **Description** | Customer can Calculate Tax |

|  |  |
| --- | --- |
|  |  |
| **Requirement ID** | FR.TP.3 |
| **Requirement Name** | Select Assessment From |
| **Description** | Customer can Submit Tax Information in Assessment From |
|  |  |

|  |  |
| --- | --- |
|  |  |
| **Requirement ID** | FR.TP.4 |
| **Requirement Name** | Tax Payment |
| **Description** | Customer can Pay Tax |

|  |  |
| --- | --- |
|  |  |
| **Requirement ID** | FR.TX.5 |
| **Requirement Name** | Select Report |
| **Description** | Customer can See Details about his Tax |

## **Functional requirement: Circle Officer:**

|  |  |
| --- | --- |
|  |  |
| **Requirement ID** | FR.CO.1 |
| **Requirement Name** | Log in |
| **Description** | Circle Officer can log in using user name and password |

|  |  |
| --- | --- |
|  |  |
| **Requirement ID** | FR.CO.2 |
| **Requirement Name** | See Tax Payer Submission |
| **Description** | Circle Officer can see all Tax Payer Submission |

|  |  |
| --- | --- |
|  |  |
| **Requirement ID** | FR.CO.3 |
| **Requirement Name** | Submit Report for Tax Payer |
| **Description** | Circle Officer can Submit Report |

## **Functional requirement: Area Officer:**

|  |  |
| --- | --- |
|  |  |
| **Requirement ID** | FR.AO.1 |
| **Requirement Name** | Log in |
| **Description** | Area Officer can log in using user name and password |

|  |  |
| --- | --- |
|  |  |
| **Requirement ID** | FR.AO.2 |
| **Requirement Name** | Assessment Tax Payer Return |
| **Description** | Area Officer can Assessment all Tax Payer |

|  |  |
| --- | --- |
|  |  |
| **Requirement ID** | FR.AO.3 |
| **Requirement Name** | Action for Tax Payer |
| **Description** | Area Officer can Action for Tax Payer |

**4.2 Non-Functional Requirement**

* System Security  
  The objective of system security is the protection of information and property from theft, corruption and other types of damage, while allowing the information and property to remain accessible and productive. System security includes the development and implementation of security countermeasures. There are a number of different approaches to computer system security, including the use of a firewall, data encryption, passwords
* Maintainability  
  Software maintenance is not about fixing wear and tear. Software is not physical, and therefore it does not degrade by itself the way physical things do. Yet most software systems are modified all the time after they have been delivered. This is what software maintenance is about. Four types of software maintenance can be distinguished:
* Bugs are discovered and have to be fixed (this is called *corrective maintenance*).
* The system has to be adapted to changes in the environment in which it operates—for example, upgrades of the operating system or technologies (this is called *adaptive maintenance*).
* Users of the system (and/or other stakeholders) have new or changed requirements (this is called *perfective maintenance*).
* Ways are identified to increase quality or prevent future bugs from occurring (this is called *preventive maintenance*).
* Performance

Software performance testing is the practice of determining whether a given application has the capacity to perform in terms of scalability and responsiveness under a specified workload. Responsiveness refers to the ability of a given application to meet pre-determined objectives for throughput, while scalability is the number of activities processed within a given time. Performing this type of testing is a key factor when ascertaining the quality of a given application.  
The entire process of software performance testing is done to accomplish a set of four goals:

* To determine the throughput or the rate of transaction.
* To determine the server response time, which is the time taken by a given application node to give a response to a request made by another node.
* To determine the response time of the render, which requires the inclusion of functional test scripts in the test scenario.
* To determine the performance specifications and document them in the test plan.

**5. Use Case**

**User profile: 1**

|  |  |  |
| --- | --- | --- |
| **User Class: Tax Payer** | **Characteristics** | **Requirement Implied** |
| **User type** | Primary | Must give input |
| **Age range** | 18-65 | Minimal Design |
| **Number of users** | Unlimited | Bandwidth should be high |
| **Education** | Educated | Simple interface |
| **Language Skill** | English | Simple English |
| **Computer/ Mobile Knowledge** | Yes | Type option or Touch |
| **Training** | Not required | Not required |
| **Goal** | Payment Tax, Submit Return, Calculation | Must see from and Submit Return, Tax Calculator |

# User profile: 2

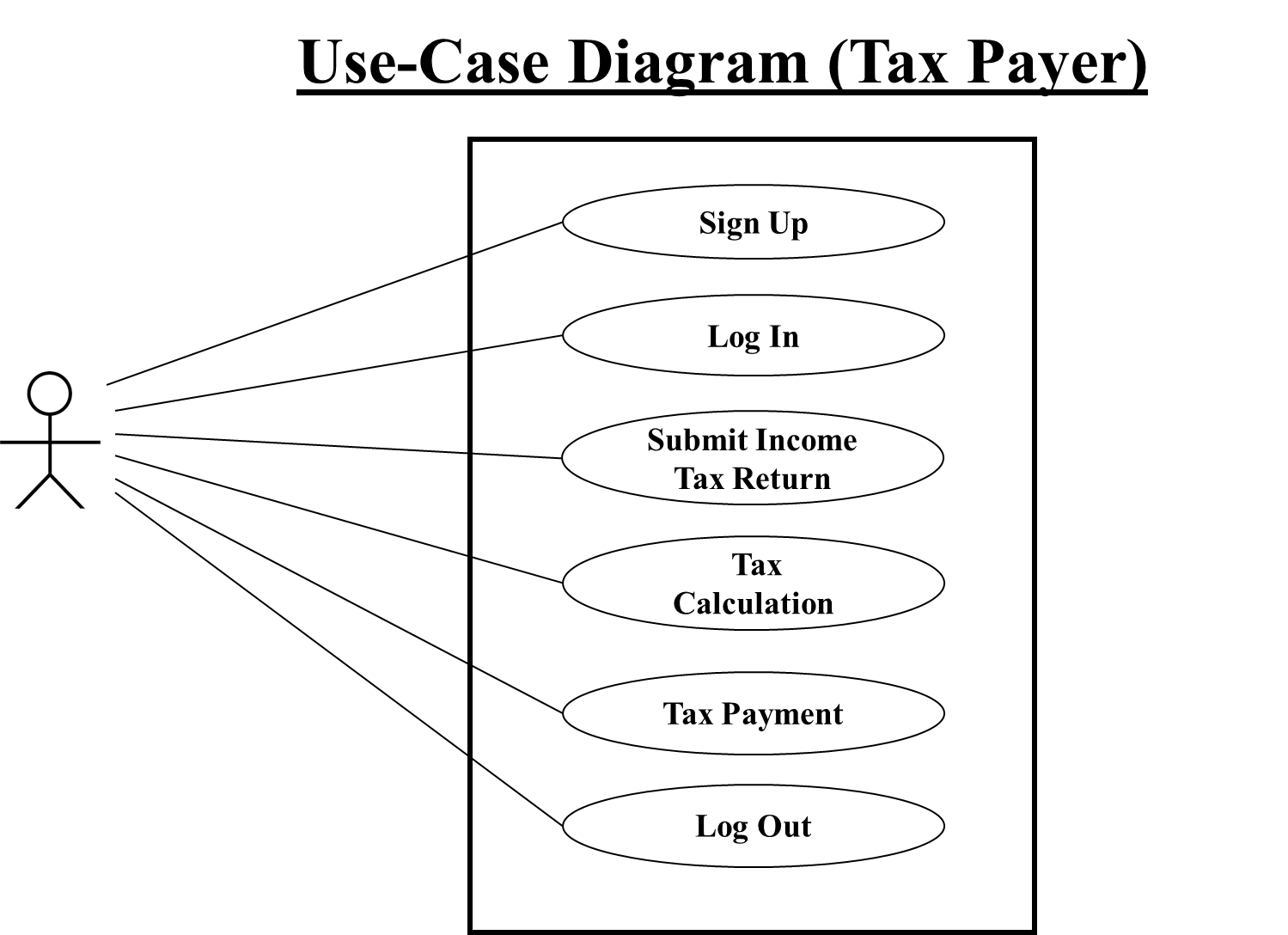
|  |  |  |
| --- | --- | --- |
| **User Class: Circle Officer** | **Characteristics** | **Requirement Implied** |
| **User type** | Primary | Must see output |
| **Age range** | 35-40 | Minimal Design |
| **Number of users** | 5 | Bandwidth should be normal |
| **Education** | Higher education | Simple interface |
| **Language Skill** | English | Simple English |
| **Computer/ Mobile Knowledge** | Yes | Only type or click option |
| **Training** | Required | 2 days training will be provided |
| **Goal** | Assessment, Report | Must See Dashboard Assessment, Report |

# User profile: 3

|  |  |  |
| --- | --- | --- |
| **User Class: Area Officer** | **Characteristics** | **Requirement Implied** |
| **User type** | Primary | Must see output |
| **Age range** | 35-40 | Minimal Design |
| **Number of users** | 2 | Bandwidth should be normal |
| **Education** | Higher education | Simple interface |
| **Language Skill** | English | Simple English |
| **Computer/ Mobile Knowledge** | Yes | Only type or click option |
| **Training** | Required | 2 days training will be provided |
| **Goal** | Report, Action Against Tax Payer | Must See Dashboard Report, Action Against Tax Payer |

**6.Use case diagram**

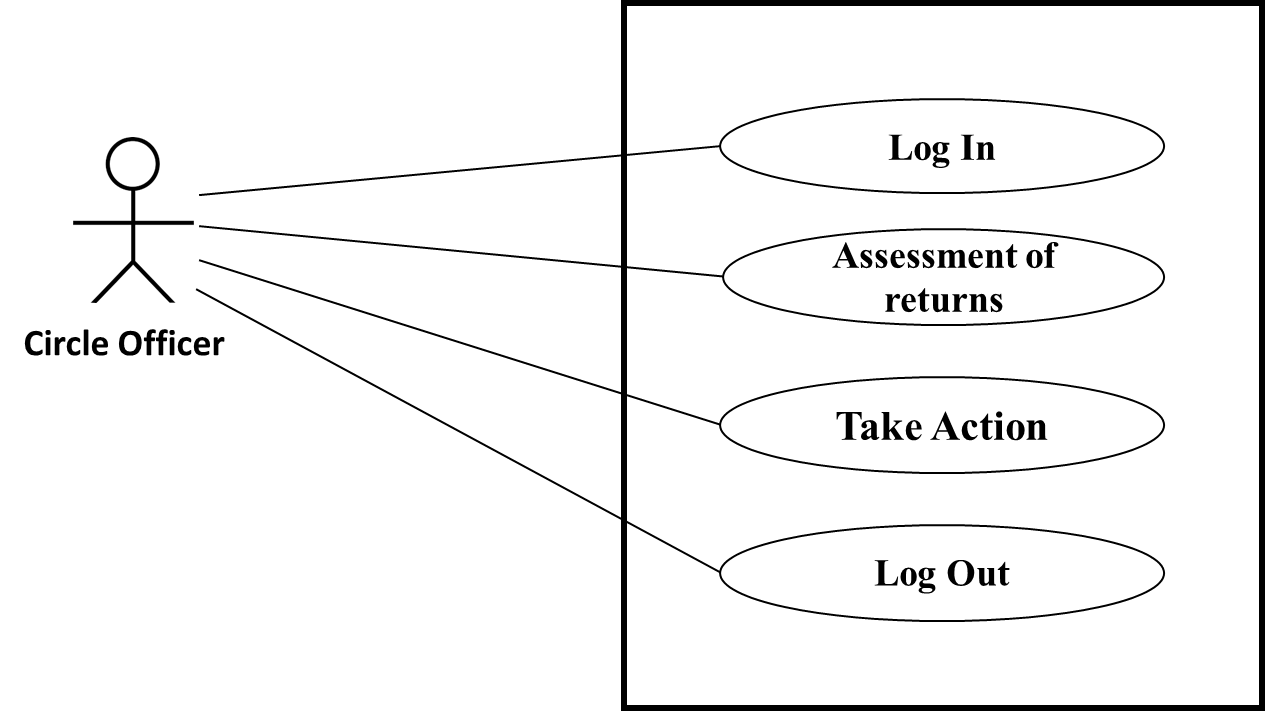
**Use case diagram: Tax Payer**

****

**Use Case Description**

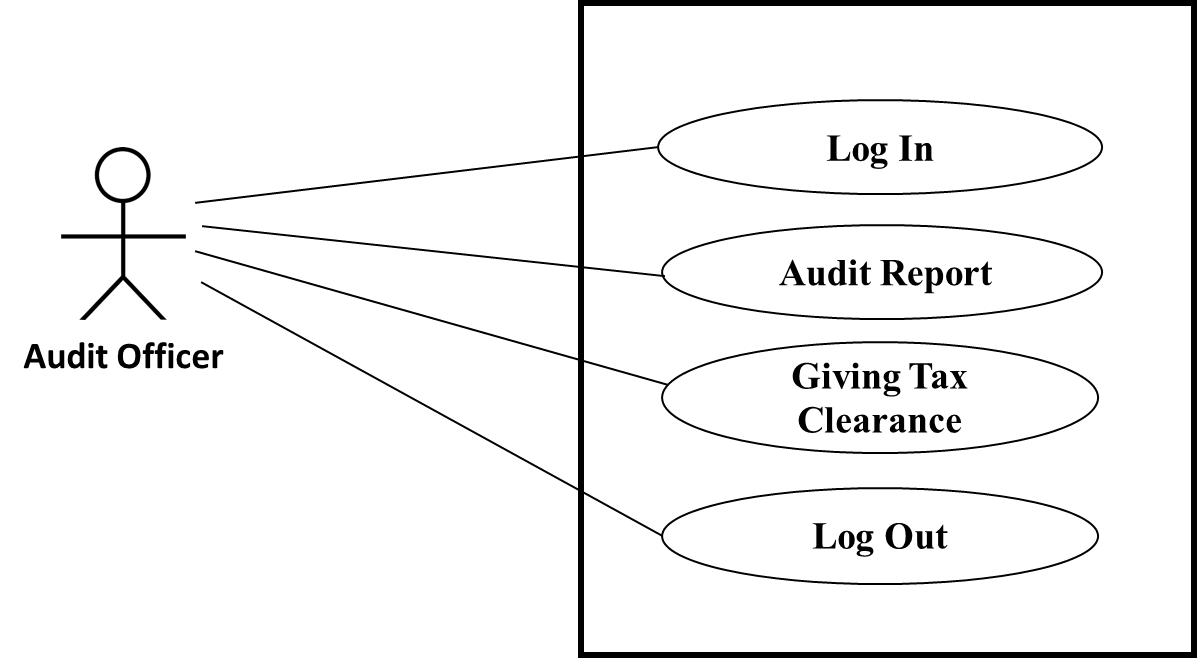
|  |  |  |  |
| --- | --- | --- | --- |
| **Name of Use Case:** | Tax Payer | | |
|  | | |
| **Description:** | | | Tax Payer Input Information, Submit Return, Calculate Tax, Tax Payment |
| **Actors:** | | | Tax Payer |
| **Preconditions:** | | | 1. Tax Payer must be Registered 2. Must be Submitted Information 3. Must be Calculate Tax |
| **Postconditions:** | | | 1. Tax Payer must use Payment gateway option 2. See Report |
| **Flow:** | | | 1. Tax Payer Registration Account for Payment 2. Select Tax Return Information 3. Input All Information 4. Calculate Tax Charge 5. Tax Charge Payment 6. View Feedback /Report |
| **Alternative Flows:** | | | 5. In step 5 of the normal flow, if the tax payer Payment by Credit Card or Mobile Banking. |
| **Exceptions:** | | |  |
| **Requirements:** | | The following requirements must be met before execution of the use case   1. Tax Payer Must be Registered 2. Must Be Input Details Tax Return Information | |

# Use case diagram: Circle Officer



|  |  |  |
| --- | --- | --- |
|  | |  |
| **Description:** | | Circle Officer Assessment Tax Payer Information and Submit Report |
| **Actors:** | | Circle Officers |
| **Preconditions:** | | 1. Officer Must be Registered |
| **Postconditions:** | |  |
| **Flow:** | | 1. Circle Officer Login their Portal 2. See Tax Payer Details 3. Assessment Tax Payer Return Document 4. Create Reports |
| **Alternative Flows:** | |  |
| **Exceptions:** | |  |
| **Requirements:** | | The following requirements must be met before execution of the use case   1. Must Be Registered | |

# Use case diagram: Area Officer

****

|  |  |  |
| --- | --- | --- |
|  |  | |
| **Description:** | Area Officer See Assessment Report, See Payment, See Details and Action against Tax Payer | |
| **Actors:** | Area Officers | |
| **Preconditions:** | 1. Officer Must be Registered | |
| **Postconditions:** |  | |
| **Flow:** | 1. Area Officer Login their Portal 2. See Tax Payer Details 3. See Assessment Reports   4. Take Action | |
| **Alternative Flows:** |  | |
| **Exceptions:** |  | |
| **Requirements:** | | The following requirements must be met before execution of the use case   1. Must Be Registered |

1. **Activity Diagram (Tax payer)**

Tax Payment

Tax Calculation

Tax Return Submit

Re

Registration

Login

No

Yes

Yes

No

Mobile Banking

Credit Card

Transaction ID

Report

Logout

**Activity Diagram (Circle Officer)**

Logout

Report Against Tax Payer

Tax Payer Details

Registration

Login

Yes

No

Yes

No

**Activity Diagram (Area Officer)**

Tax Payer Details

Registration

Login

Payment Details

Action Against

Tax Payer

Logout

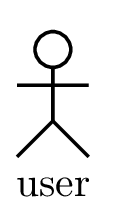
Yes

No

Yes

No

1. **Sequence Diagram (Tax Payer)**



Tax Payer

Tax Return Submit

Tax Calculation

View Report

Payment

Login Success

Login to Page

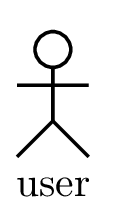
Tax Return Submission

Tax Calculation

Tax Payment

View Report

**Sequence Diagram (Circle Officer)**



Tax Payer Assessment

Report

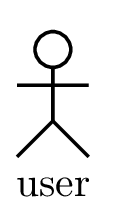
Circle Officer

Database

Assessment

Report

**Sequence Diagram (Area Officer)**



Assessment Details

Area Officer

Payment Details

Database

Report & Action

Assessment

Payment Details

Report & Action

1. **Class Diagram**

1

|  |
| --- |
| *Area Officer* |
| -A\_id : string  -A\_name : string  -A\_address : string  -A\_phone :int  -C\_id : string  -T\_id : string |
| +Tax\_payer\_report()  +Action \_against\_expire() |

1…\*

1

1…\*

1

1…\*

1

|  |
| --- |
| *Tax Payer* |
| -T\_id : string  -T\_name : string  -T\_address : string  -TIN\_number : int  -T\_phone : int |
| +Tax\_return submit()  +Tax\_calculation()  +Tax\_payment():boolean  +getincome():double |

|  |
| --- |
| *Circle Officer* |
| -C\_id : string  -C\_name : string  -C\_address : string  -C\_phone : int  -T\_id : string |
| +Assesment\_audit()  +Report\_generate() |

|  |
| --- |
| *Report* |
| -Report\_id : string  -Report\_name : string |
| +Tax\_payer\_report()  +Action \_against\_expire() |

1

|  |
| --- |
| *Payment* |
| -Tran\_no : string  -Created\_date :Date  -Payment\_amount :double  -Payment\_name:string |
| +Card()  +Tax\_payer () |

1

1

|  |
| --- |
| *Card* |
| -Card\_no :int  -Card\_type:string |
| +Payment\_option():boolen |

1. **ERD Diagram**

Tax Payer

1…\*

1…\*

1

Assessment

Pay

1

Circle Officer

Action

1…\*

Report

1

1

1

Manage

1

Area Officer

Payment

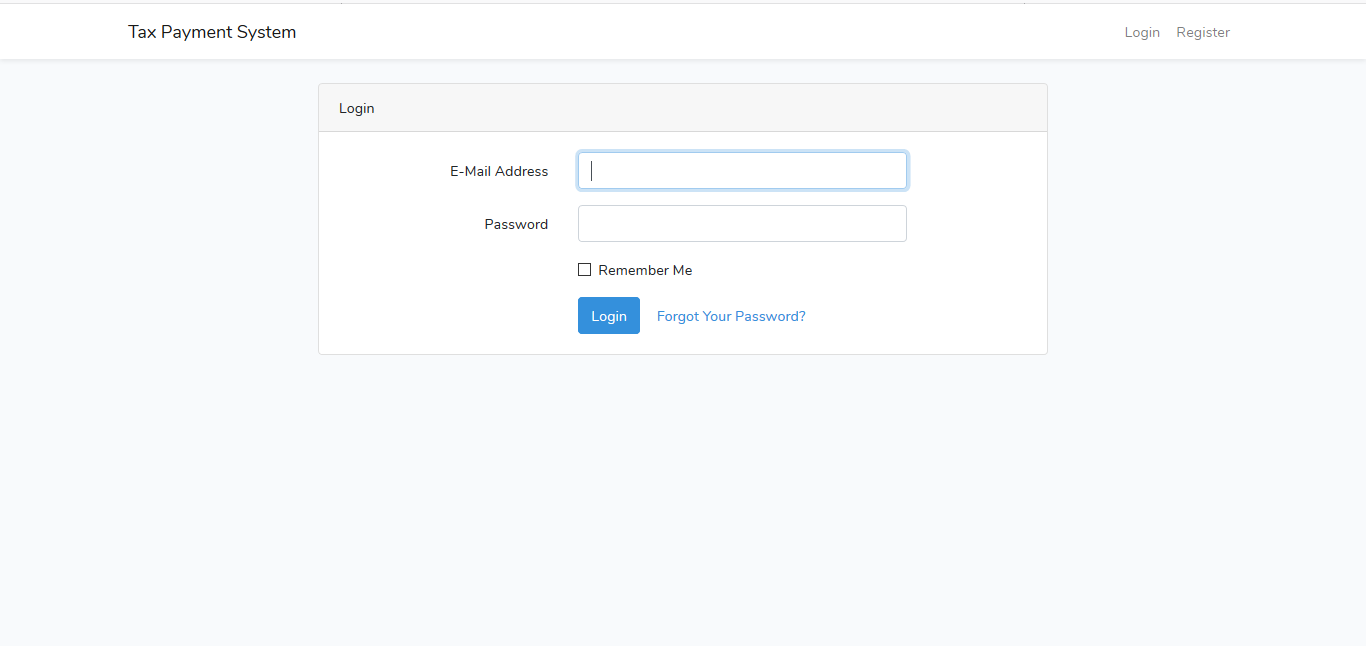
1…\*

1. **UI/UX**

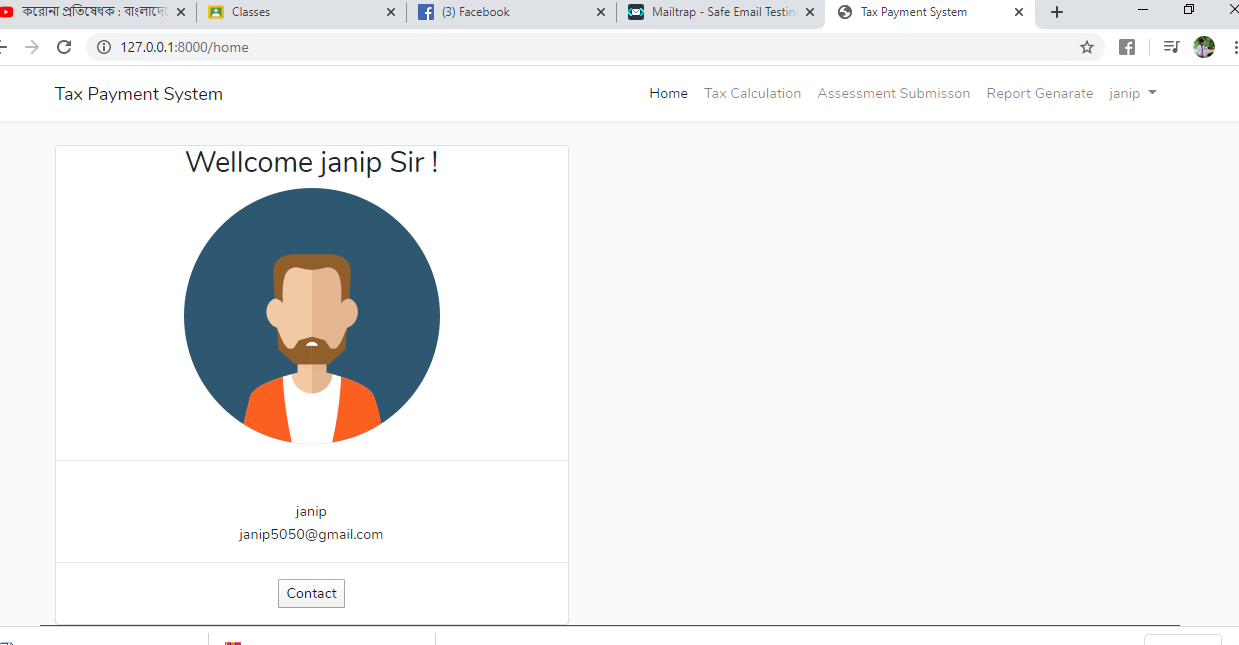
**1.Starting Page**

****

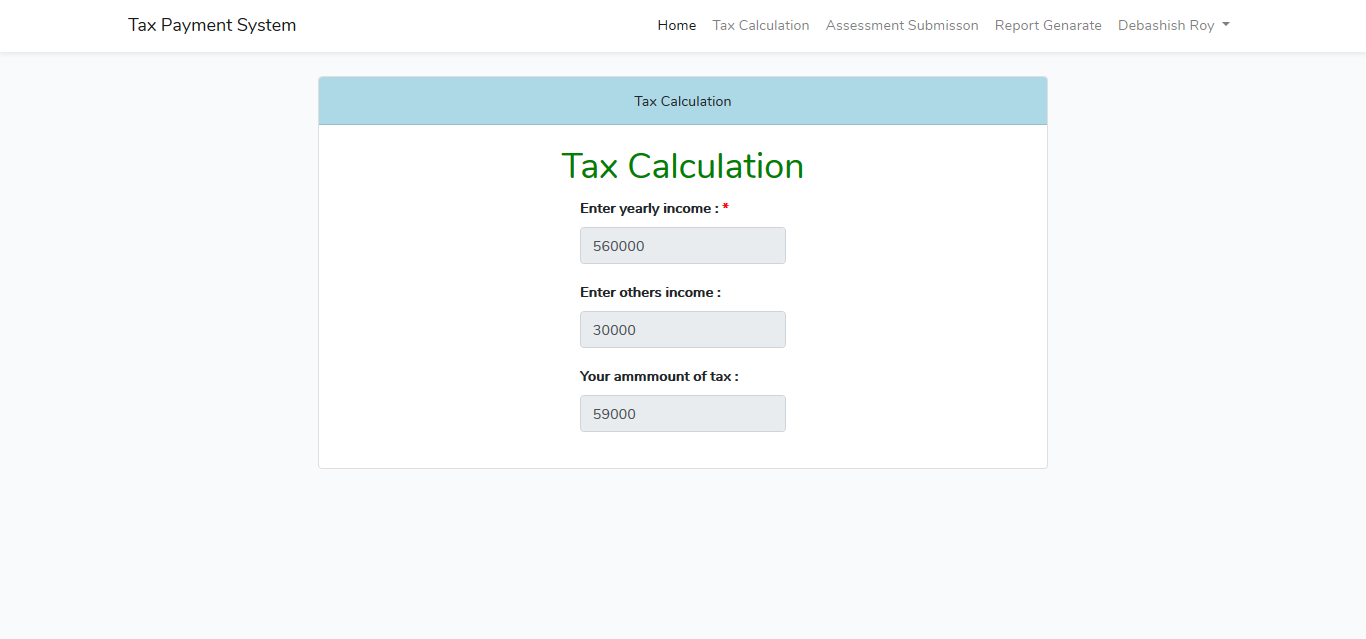
**2.Login from**

****

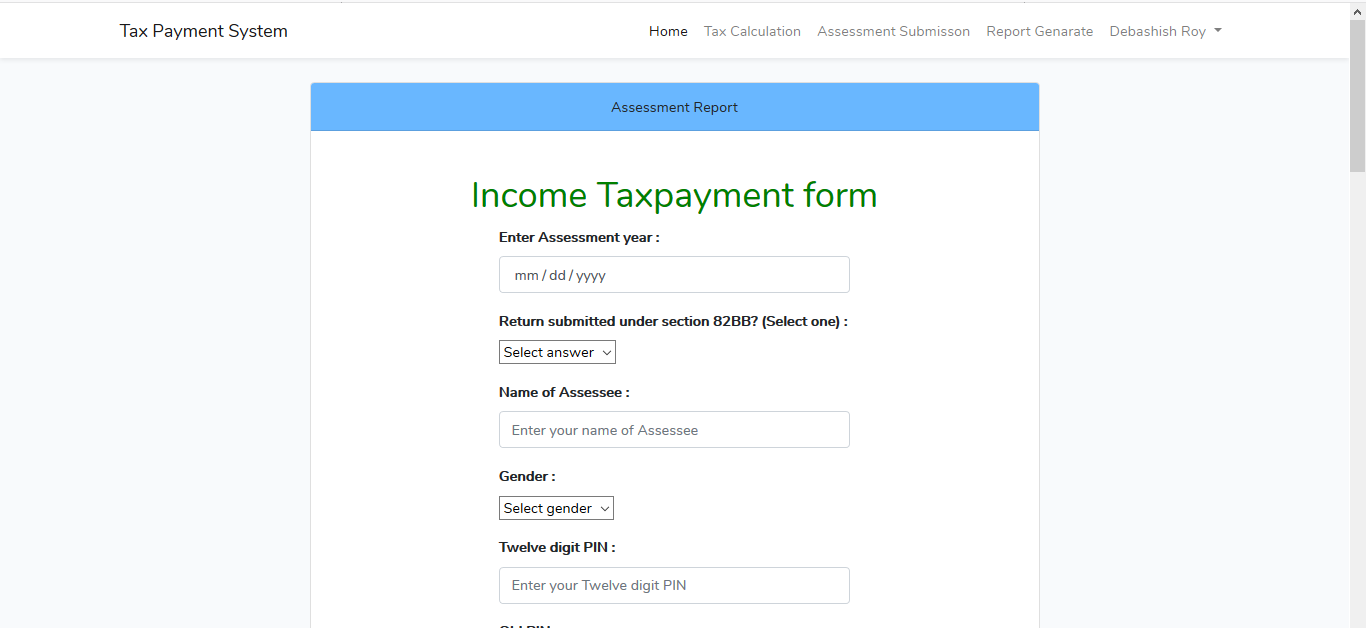
**3.Taxpayer home page**

****

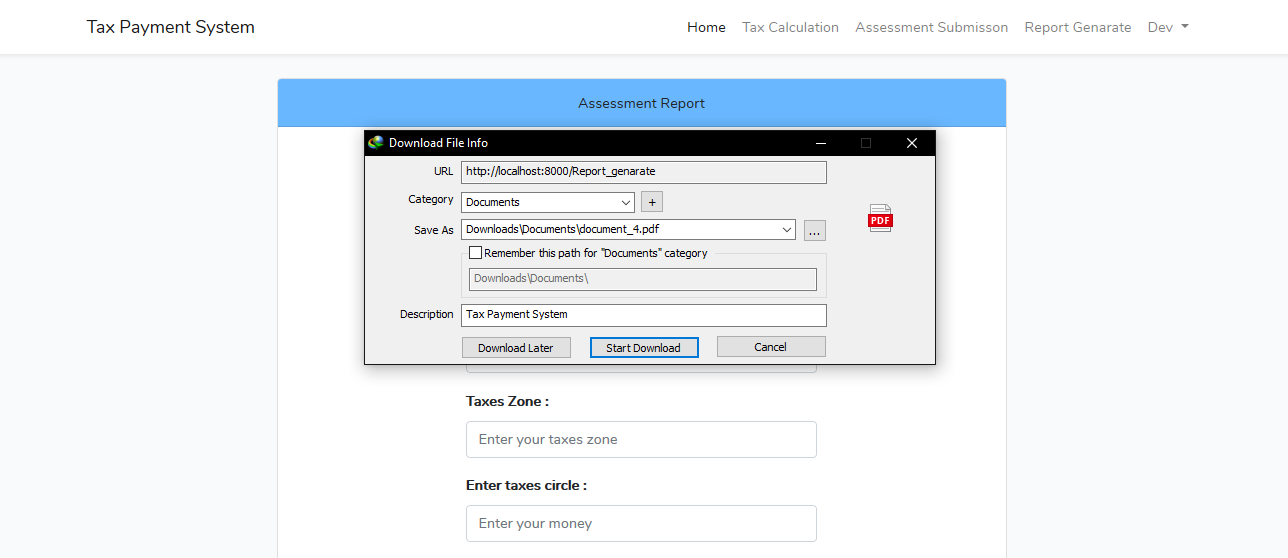
**4.Tax Calculation from**

****

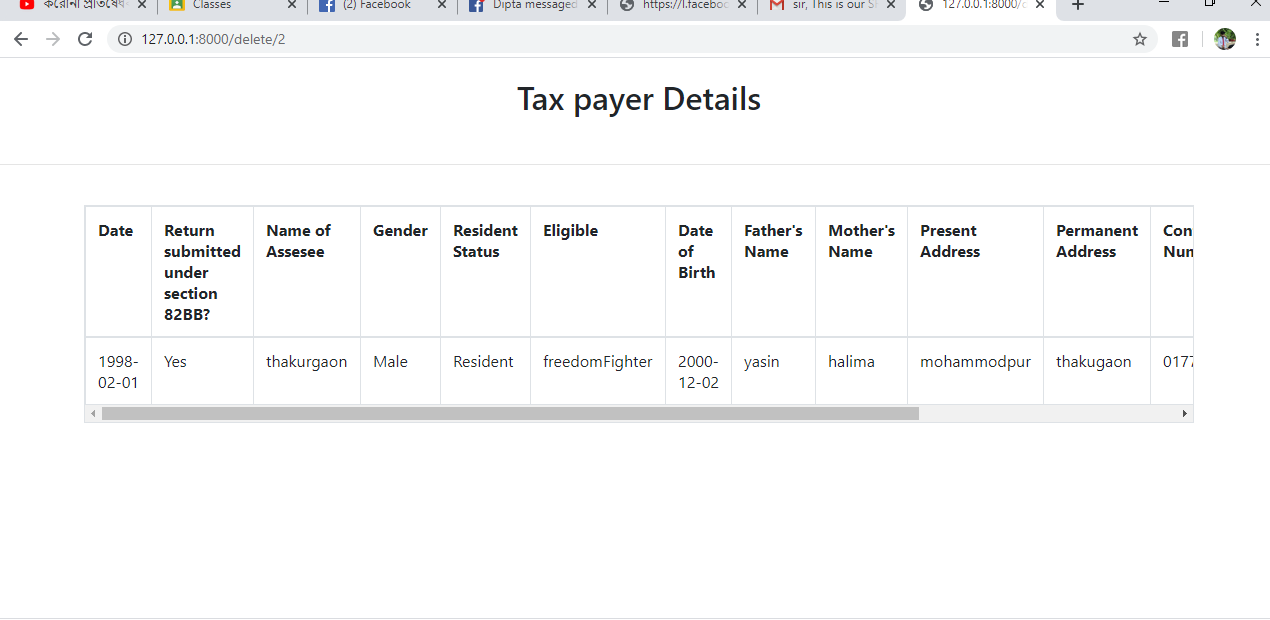
**5.Taxpayment from**

****

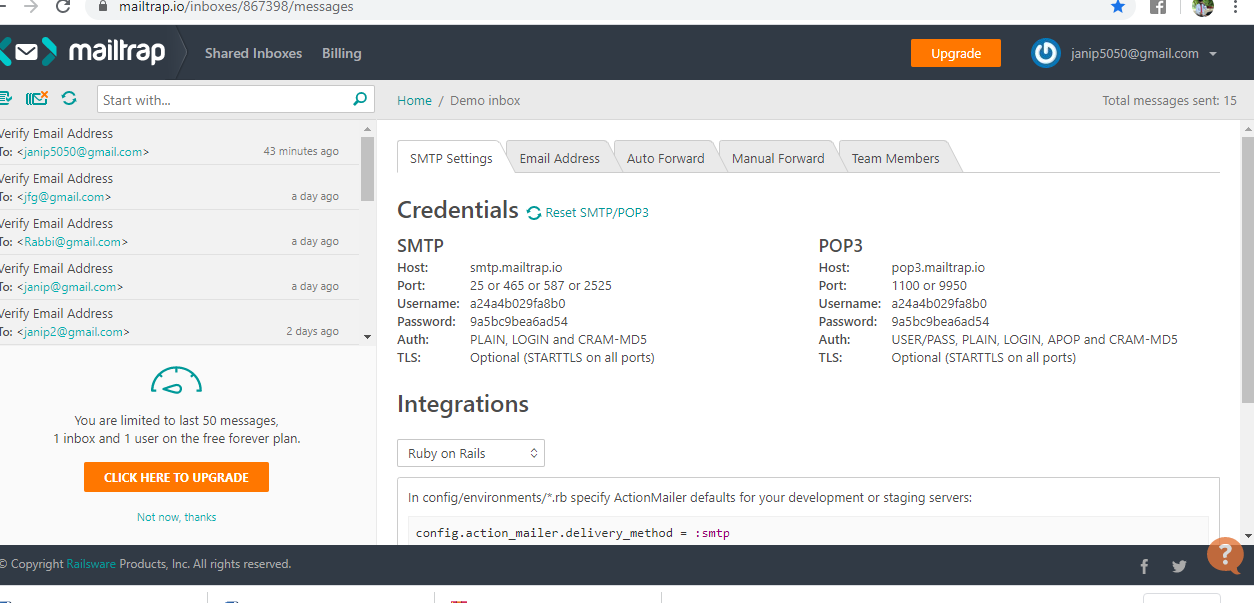
**6.Assessment Report**

****

**7.Admin Dashboard**

****

**14.Mailtrap**

****