Software Requirements Specification

for

Caribbean Emergency Registry

Version 1.0 approved

Prepared by:

Shane Okukenu 190902

Tevin Taylor 190901

Ashray Soares 190910

Neo Alexis 190904

Joshua Alkins 190908

Jason Charles 190906

University of the West Indies

06/06/2020

1. Introduction **5**
1.1 Purpose 5
1.2 Document Conventions 5
1.3 Intended Audience and Reading Suggestions 5
1.4 Product Scope 5
1.5 References 5

2. Overall Description **6**
2.1 Product Perspective 6
2.2 Product Functions 6
2.3 User Classes and Characteristics 6
2.4 Operating Environment 6
2.5 Design and Implementation Constraints 7
2.6 Assumptions and Dependencies 7

3. External Interface Requirements **7**
3.1 User Interfaces 7
3.1.1 Landing Page 7
3.1.2 User Login Page 10
3.1.3 Registration Page 11
3.1.4 User Dashboard 12
3.1.5. User Information Form 12
3.2 Software Interfaces 15

4. System Features **16**
4.1 Login 17
4.1.1. Description and Priority 18
4.1.2. Stimulus/Response 19
4.1.3. Functional Requirements 19
4.2 Add Information 20
4.2.1. Description and Priority 21
4.2.2. Stimulus/Response 21
4.2.3. Functional Requirements 22
4.3 Edit Information 22
4.3.1. Description and Priority 23
4.3.2. Stimulus/Response 23
4.3.3. Functional Requirements 24

4.4 Query 25
4.4.1. Description and Priority 26
4.4.2. Stimulus/Response 26
4.4.3. Functional Requirements 27
4.5 Edit User Information 27
4.5.1. Description and Priority 29
4.5.2. Stimulus/Response 29
4.5.3. Functional Requirements 29
4.6 Delete User 30
4.6.1. Description and Priority 31
4.6.2. Stimulus/Response 31
4.6.3. Functional Requirements 32
4.7 Create New Administrator 32
4.7.1. Description and Priority 33
4.7.2. Stimulus/Response 33
4.7.3. Functional Requirements 33
4.8 Delete Administrator 34
4.8.1. Description and Priority 35
4.8.2. Stimulus/Response 35
4.8.3. Functional Requirements 36
4.9 Send Update Reminder Email 36
4.9.1. Description and Priority 36
4.9.2. Stimulus/Response 36
4.9.3. Functional Requirements 36

5. Other Nonfunctional Requirement 37
5.1 Performance Requirements 37
5.2 Security Requirements 37
5.3 Software Quality Attributes 37

Appendix A: Glossary 39

Appendix B: Analysis Models 39

Appendix C: To Be Determined 40

1. Introduction

1.1 Purpose

This is a Requirement Specification document for a creation of an online tool that'll be able to input, retrieve and manage information from Barbadian nationals and diaspora. This document describes the scope, objectives and goals of the system. This document highlights the functional and nonfunctional requirements of the system. This document is intended to direct the design and implementation of the target system in an object oriented language.

1.2 Document Conventions

This document was created based on the IEEE template for System Requirements Specification Documents.

The document also incorporates Software Modeling techniques taught in the SWEN program.

1.3 Intended Audience and Reading Suggestions

This document is intended for:

- Programmers and developers working on the project.
- Dr.Xu, SWEN 3004 lecturer.
- Any individual involved in the assessment of SWEN courses.

1.4 Product Scope

The scope of the project is a web-based system that supports addition, deletion, and management of users information. Users are able to register and login to add and edit necessary information. Admins are able to login in to view , query, edit and delete users information. The implementation of a database is also within the scope of this project as well as website security.

1.5 References

Caribbean Emergency Registry GitHub page:

IEEE template for System Requirements Specification Documents: https://goo.gl/nsUFwy

2. Overall Description

2.1 Product Perspective

This web based system was designed to satisfy the requirements of the Barbadian Government. The recent outbreak of Covid-19 was the inspiration for this system as the Barbadian Government needed a way to keep track of their citizens that are currently living overseas. This data collection website was the decided tool that would satisfy the desired objectives the government wished to fulfill.

2.2 Product Functions

This website aims to allow users to log onto the website via browser and perform the available features the website provides.These features include:

General Users

1. Add Information
2. Edit Information

Administrators

1. Edit User information
2. Query
3. Delete User
4. Create new Administrators
5. Delete Administrators

2.3 User Classes and Characteristics

General User - Uses the system to provide profession and location information to the Barbados Government.

Administrator - Manages the system and users, and queries user information on behalf of the Barbados Government.

2.4 Operating Environment

The Client

The website shall be viewable from Microsoft Internet Explorer 4.0 or later, Google Chrome, Microsoft Edge, Mozilla Firefox and Apple Safari.

Server

The server shall be run on a computer running the latest version of Windows 10.

2.5 Design and Implementation Constraints

CON-1 The application must use the flask framework
CON-2 The application must use MongoDB
CON-3 The application must run on Windows OS

2.6 Assumptions and Dependencies

FlaskLogin - Python library used to handle user login.
Itsdangerous - Python library used to handle data encryption.
Flash-Mail - Python library used to handle automated email sending.
PyMongo - Python Library for interfacing with MongoDB.
MongoDB - Non-relational Database used to store user information.

3. External Interface Requirements

3.1 User Interfaces

The user interface was requested to have a theme based on Barbados, and was designed as such. The color palette for the site borrows from the Barbados National Flag, using Blue as the primary color and Yellow as the secondary color or accent.

3.1.1 Landing Page

Figure 3.1.1.1. Landing Page Section 1 (Top).

The Landing Page acts as the first point of the site that the user will see and interact with. At the top of the Landing Page (as seen in Figure 3.1.1.1.) you will find a background consisting of multiple Barbadian landmarks that interchange in a slideshow as you stay on the page. In the center of the slide show there is a short explanation of the purpose of the system for new users, with the signup and login buttons just below. In the very top left of the page in the navigation bar you will find the Barbados coat of arms to again add to the Barbadian theme and stay “on brand” for the Barbados Government. In the navigation bar you will find links to different sections of the Landing Page as well as another method to reach the Register Page for ease of use.

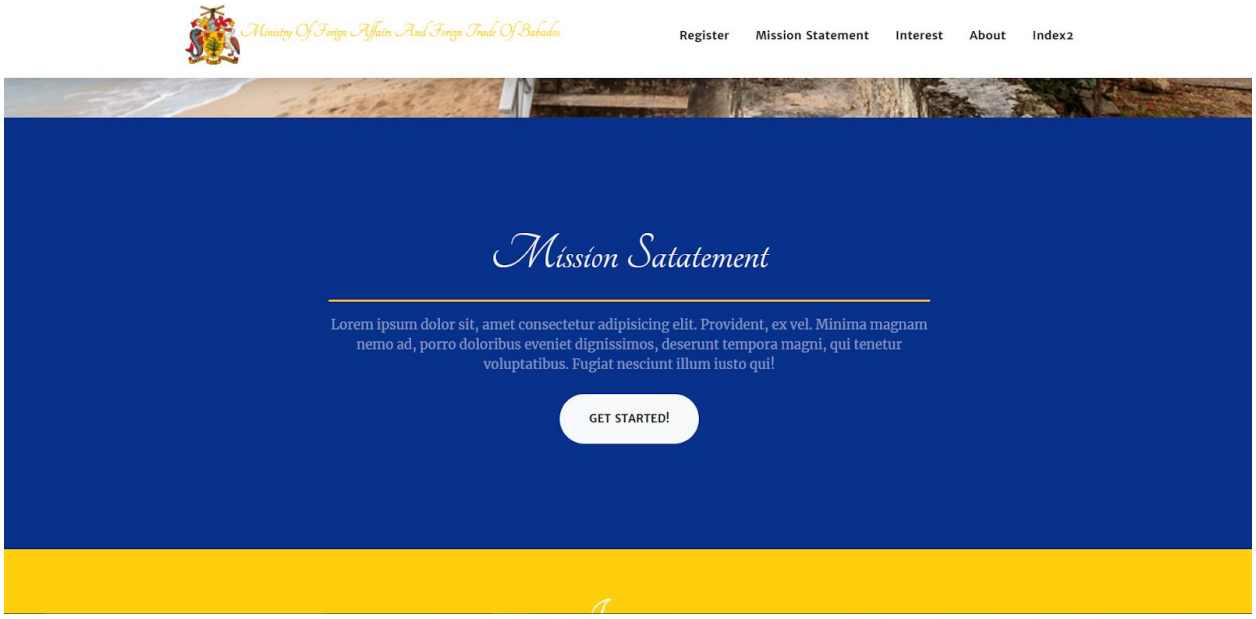

Figure 3.1.1.2. Landing Page Section 2 (Mission Statement)

Just below the slideshow background at the top of the page you will section for a Mission Statement to be filled out by a representative of the Barbados Government to help explain what the Government wants to accomplish with the system. AS seen in Figure 3.1.1.2.

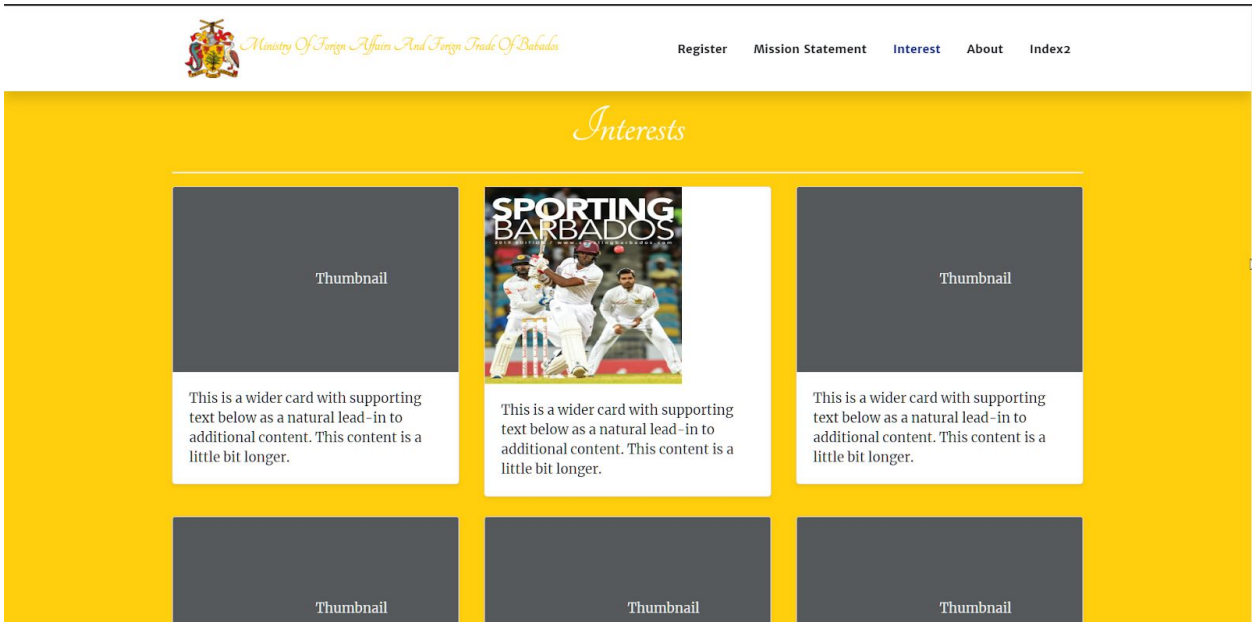

Figure 3.1.1.3. Landing Page Section 3 (External Links)

Just beneath the Mission Statement, there are a series of links that direct users to interesting, important or relevant information relating to Barbados (as seen in Figure 3.1.1.3.). These links may include investment opportunities, sporting events, critical news amongst other things. These links will be managed by an employee or representative of the Barbados Government in charge of the website and changed as regularly as they see fit.

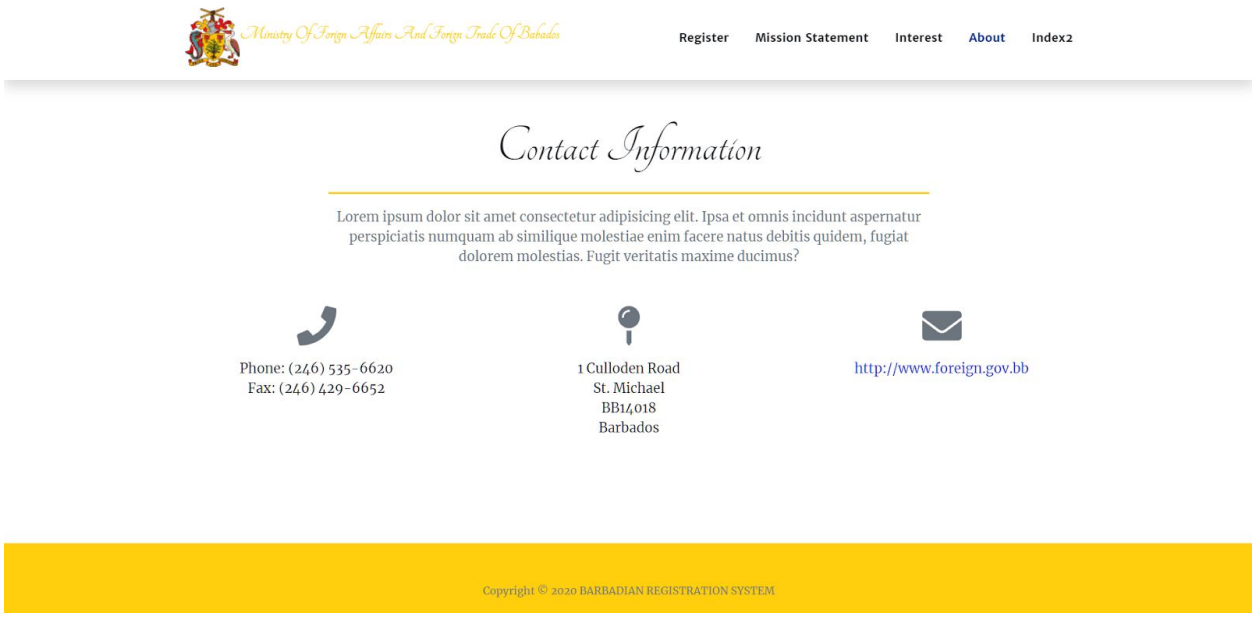

Figure 3.1.1.4. Landing Page Section 4 (Bottom)

At the bottom of the Landing Page you will find contact information for the Barbados Ministry of Forein Affairs. See Figure 3.1.1.4.

3.1.2 User Login Page

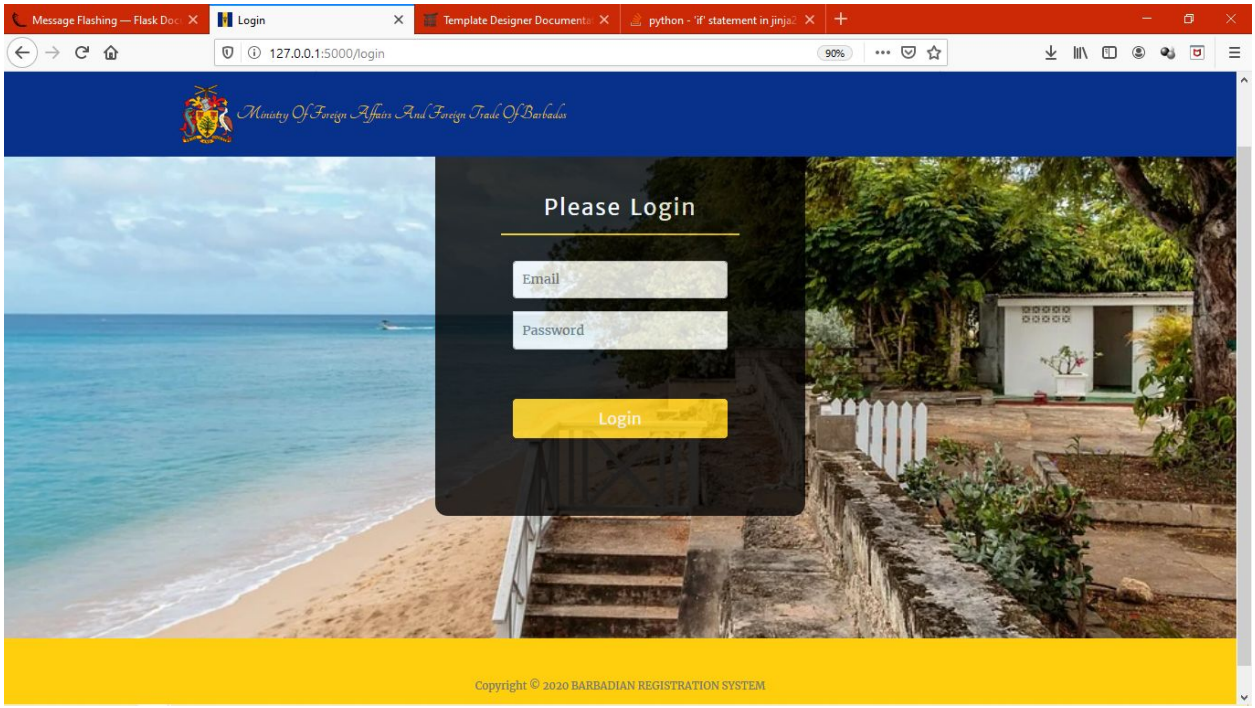

Figure 3.1.2.1. Login Page

The Login Page is very simple consisting of just two text fields, for the user to enter their email and password, and a Login button. This sits on top of the same slideshow background from the Landing Page. See Figure 3.1.2.1.

3.1.3 Registration Page

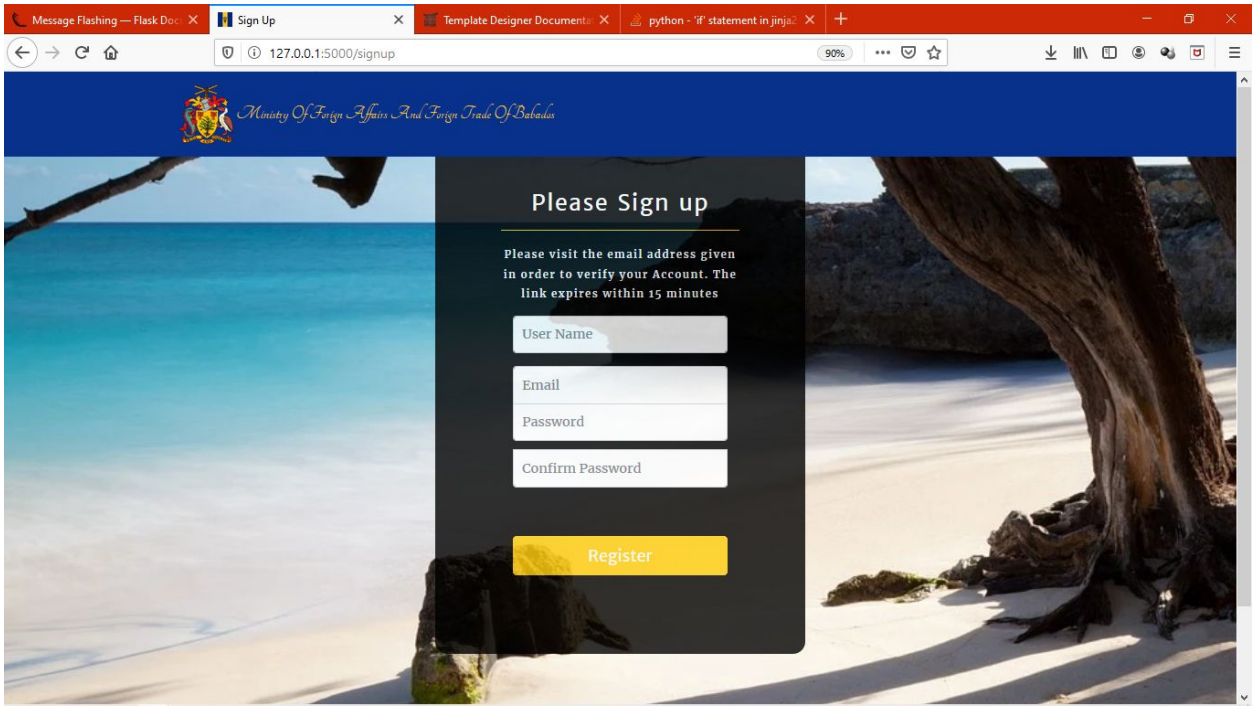

Figure 3.1.3.1 User Registration Page

The Registration Page is very similar to the Login Page with now four text fields to collect login credentials and a Register button. All backed by the same slideshow background. See Figure 3.1.3.1

3.1.4 User Dashboard

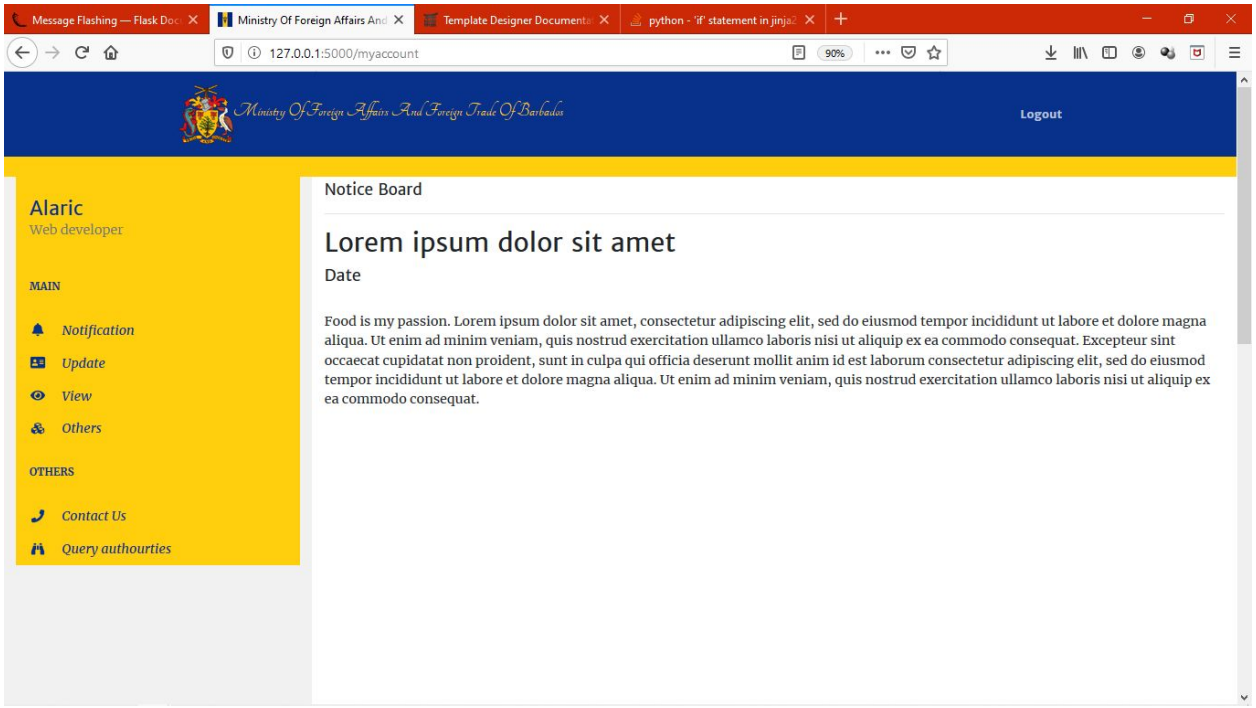

Figure 3.1.4.1 User Dashboard

The User Dashboard consists of a notice board presenting information relevant to the User, for instance in the event of an emergence the notice board may display a warning or advice on accounts whose users are known to have been in the area of the emergence. To the left is a navigation bar which will allow the user to access the various functionality of the system they may use. This includes the Update link, which will navigate the user to a page where the user may update their information,and the View link which will navigate the user to a page where they will be able to view their current information. See Figure 3.1.4.1.

3.1.5. User Information Form

The User Information Form is where the user enters their information into the system for the first time. This consists of two sections with a segment that adjusts the questions to collect specific information on the user depending on whether the user is a National, Diaspora or “A Friend of Barbados”.

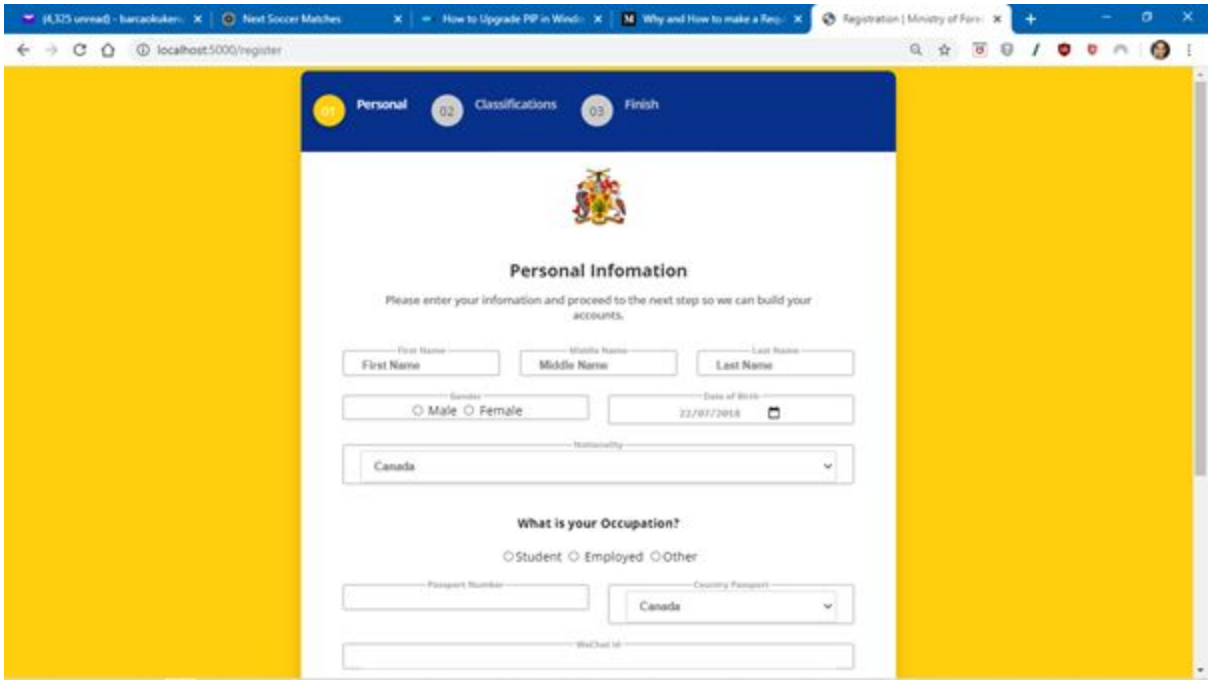

Figure 3.1.5.1. User Information Form Section 1 (Part 1)

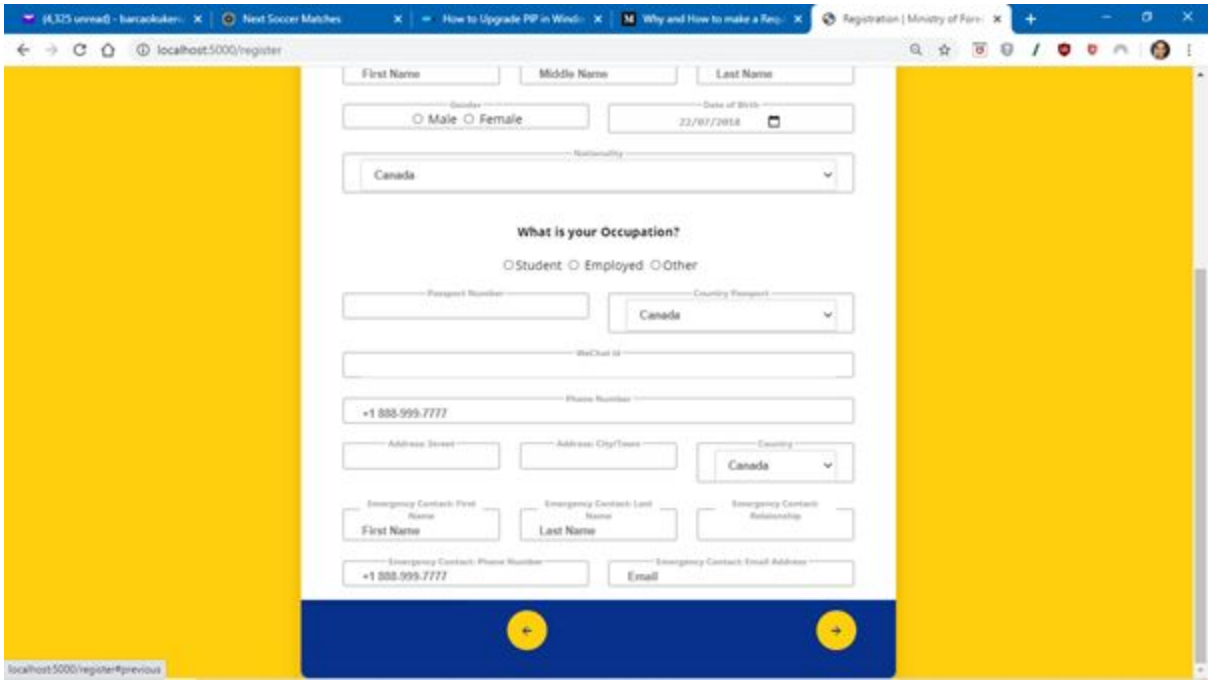

Figure 3.1.5.2. User Information Form Section 1 (Part 2)

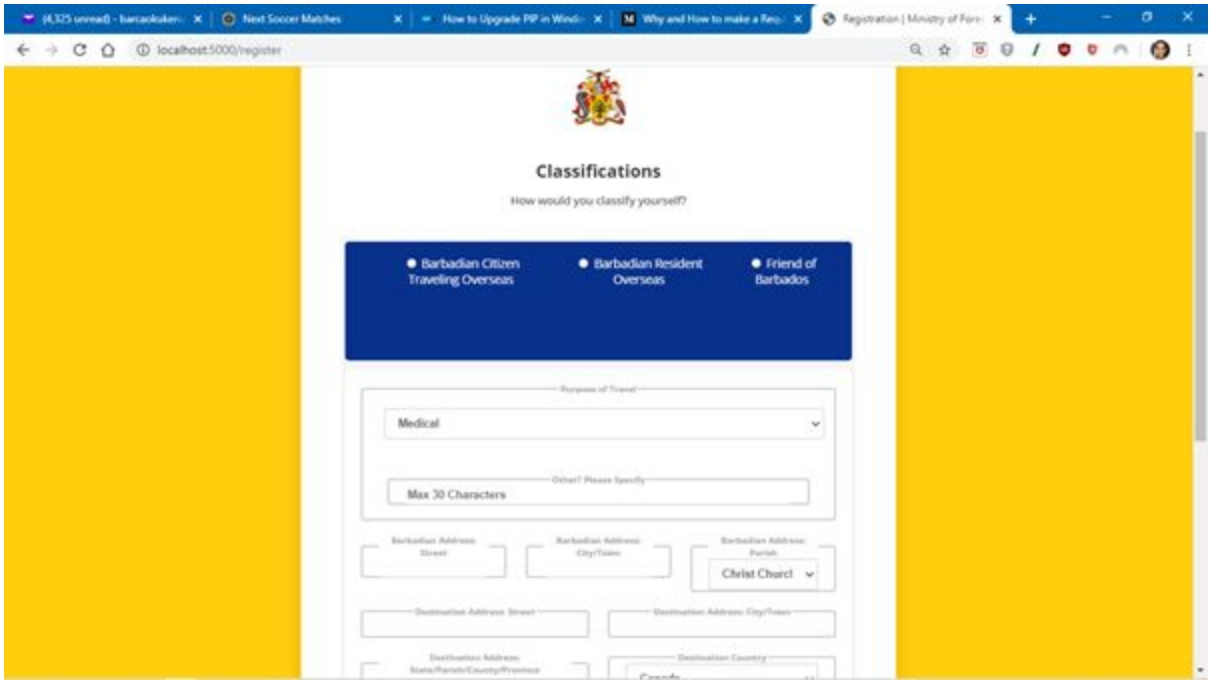

Figure 3.1.5.3. User Information Form Section 2 (Part 1)

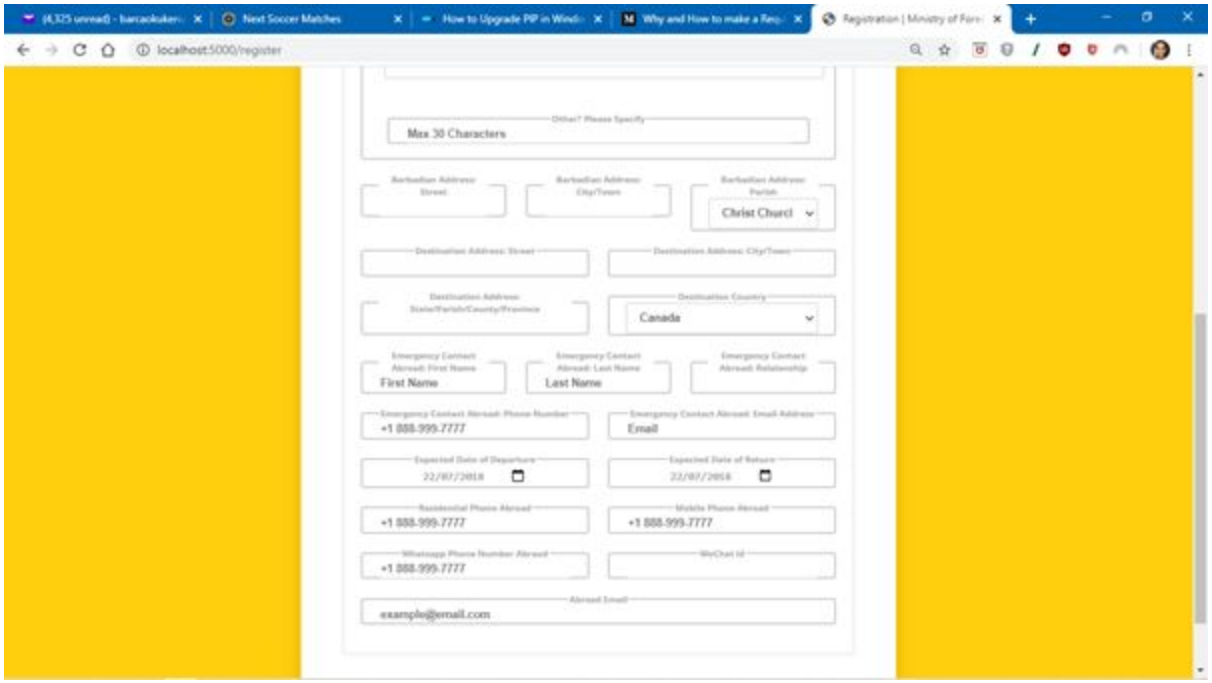

Figure 3.1.5.4. User Information Form Section 2 (Part 2)

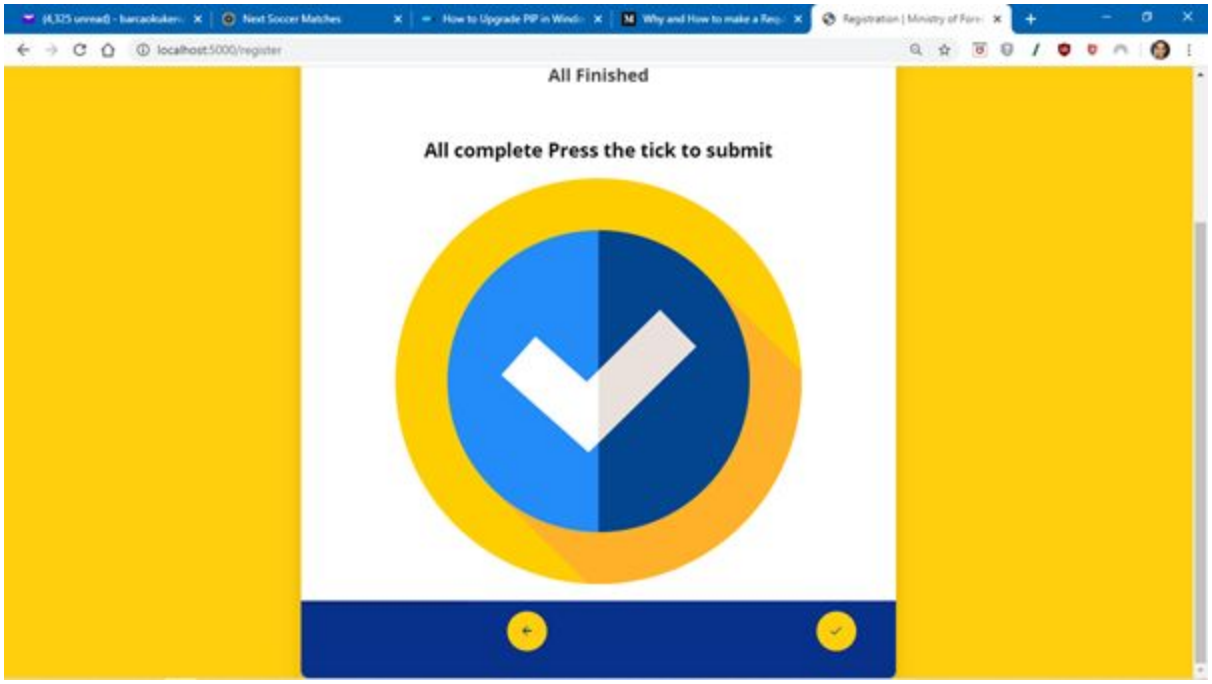

Figure 3.1.5.3. User Information Form Submission

4. System Features

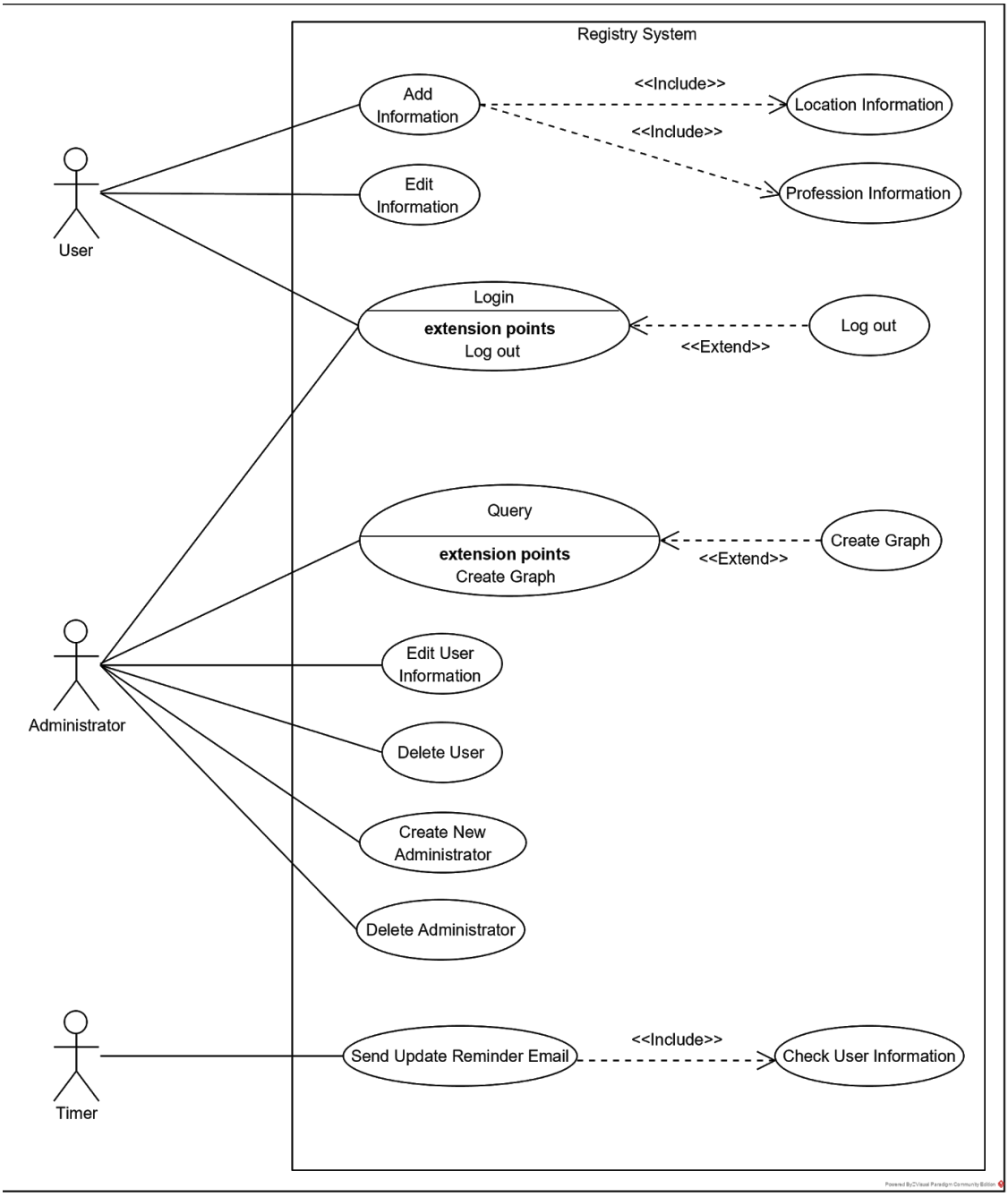

Figure 4.1. Use Case Context Diagram

4.1 Login

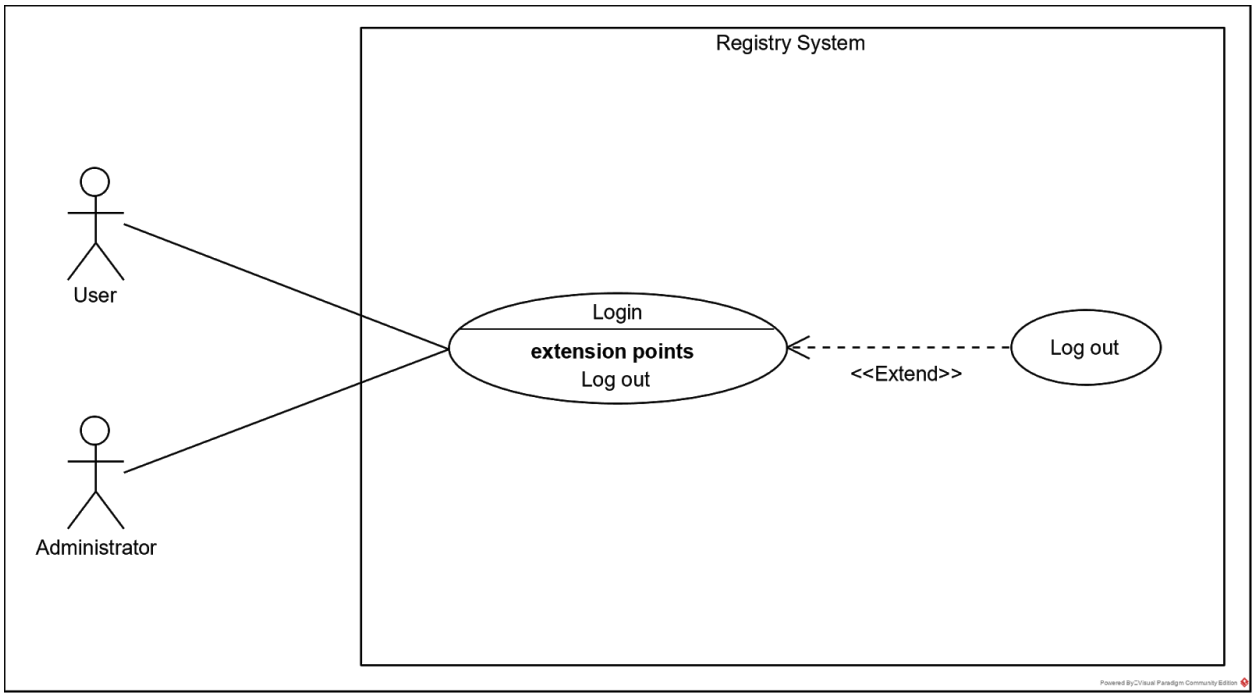

Figure 4.2. Login Use Case Diagram.

Figure 4.3. UC-1 Login Use Case.

Title	Login
Use Case ID	**UC-1**
Description	The user or administrator logs into the system.
Primary Actors	• User • Administrator
Preconditions	• The server is running. • The user or administrator has navigated to their respective Login Page.
Postconditions	The user or administrator is rightfully given access to the system.

Main Flow	1. The user enters their login credentials. 2. The user selects login. 3. The system queries the database to find a user account that matches the user's login credentials. 4. The system verifies the entered password matches the account password. 5. The system grants the user access to the user's account. 6. The system navigates to the user's respective Dashboard.
Alternative Flows	3.1. The application does not find a matching account. 3.1.1. The system denies the user access to the rest of the system. 4.1. The application finds the passwords do not match. 4.1.1. The system denies the user access to the rest of the system. 6.1. The user selects logout. 6.1.1. 6.1.2. The system removes access to the rest of the system.
Non-Functional Requirements	**QA-1** Passwords should be encrypted when stored in the database.. **QA-2** It should take no more than 1 second to verify the user's credentials from the user selecting login.

4.1.1. Description and Priority

Both the user and the administrator will be required to login to access the system's functionality. The user will login through a link on the Landing Page of the site. The administrator login page will not be publicly advertised and will only be accessible by directly typing the URL into the web browser. Both pages will function identically, but will lead to different access to the system's functionality.

Priority: High

4.1.2. Stimulus/Response

The term user will be used in the rest of Section 4.2 to refer to both the user and the administrator.
To login to the system the user enters their login credentials on either the User Login Page or the Admin Login Page depending on the type of user. The user then selects submit, and the user's login credentials are sent back to the server. The server then accesses the database and validates the login credentials. If the credentials match a known user, the server returns a URL that navigates the user to their respective Dashboard and logs the user in with a user session. If the credentials do not match any known user, the server returns a URL directing the user back to the Login Page where with an error message stating that either their username or password is incorrect.

4.1.3. Functional Requirements

REQ-1 Users must be able to login via webpage to gain access to the system.
REQ-2 Administrators must be able to login via webpage to gain access to the system.

4.2 Add Information

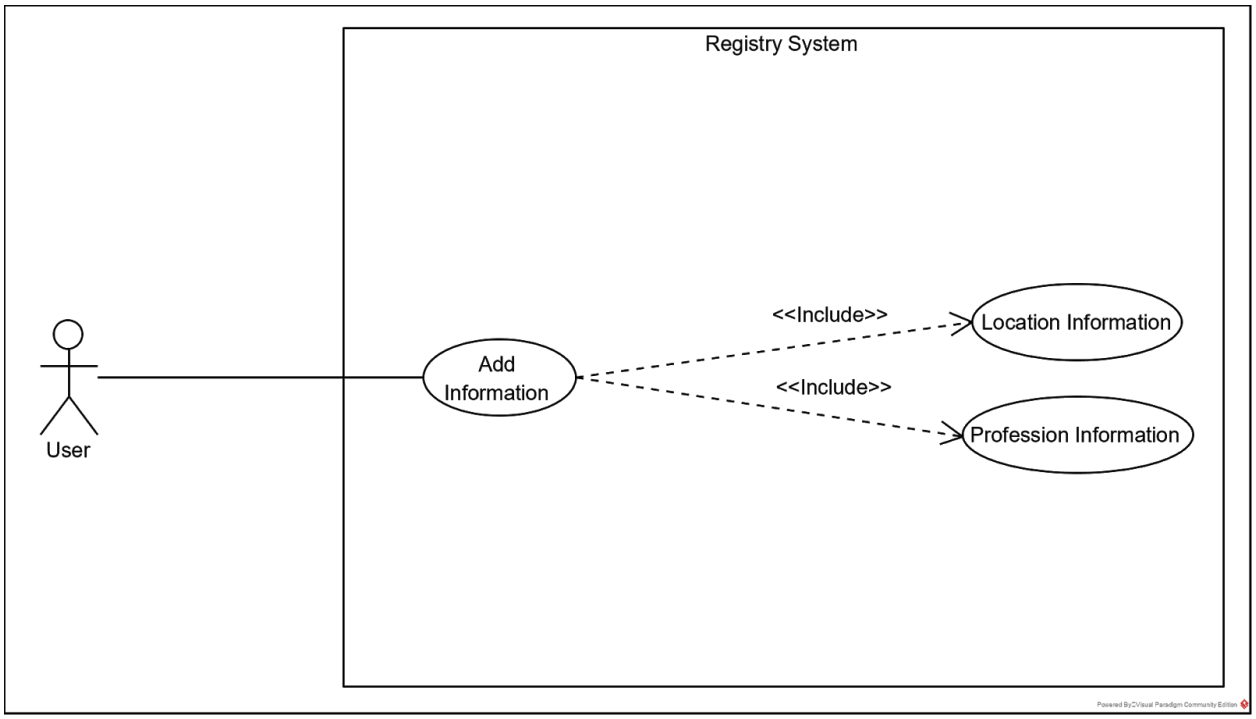

Figure 4.4. Add Information Use Case Diagram.

Figure 4.5. UC-2 Add Information Use Case.

Title	Add Information
Use Case ID	**UC-2**
Description	The user adds location and profession information to their account.
Primary Actor	• User
Preconditions	• The server is running. • The user is logged in. • The user has navigated to the Add Information Page.
Postconditions	The user's information is added to their account and stored in the database.

Main Flow	1. The system presents the user with the Information Form. 2. The user enters their information into respective fields. 3. The user selects submit. 4. The application retrieves the entered information. 5. The application stores the information in the database. 6. The application navigates to the User Dashboard.
Alternative Flows	-
Non-Functional Requirements	**QA-3** The User Information Form should be split into multiple smaller sections. **QA-4** The information from the form should be added to the database within 10 seconds of submit being selected.

4.2.1. Description and Priority

The user is directed to a form with an array of questions the Barbados Government wishes to know about the user. The user is then able to fill out the form via text fields, checkboxes and radial buttons.

Priority: Extremely High

4.2.2. Stimulus/Response

The user has navigated to the Add Information Page via a link on the User Dashboard. The user is presented with a form asking questions about the user's education level, employment status, relationship to the country and location information, as well as other information the government may want to know about the user. The form is presented in three segments, with segment's questions varying depending on what the user selects for "relationship to Barbados", out of Citizen, Diaspora, or Friend of Barbados. Once the user completes a section of the form they are then able to select a button at the bottom of the form to take them to the next section of the form. Once all sections are completed the user will be presented with the option to submit their information. Once the submit option is selected the information is sent to the server via a token. The server then stores the information associated with the user in the database. The server then directs the client back to the User Dashboard.

4.2.3. Functional Requirements

REQ-3 The system must collect information from the user via the use of a form.
REQ-4 The form must adapt to choices of the user for key questions.
REQ-5 The system must store information about users in a database.

4.3 Edit Information

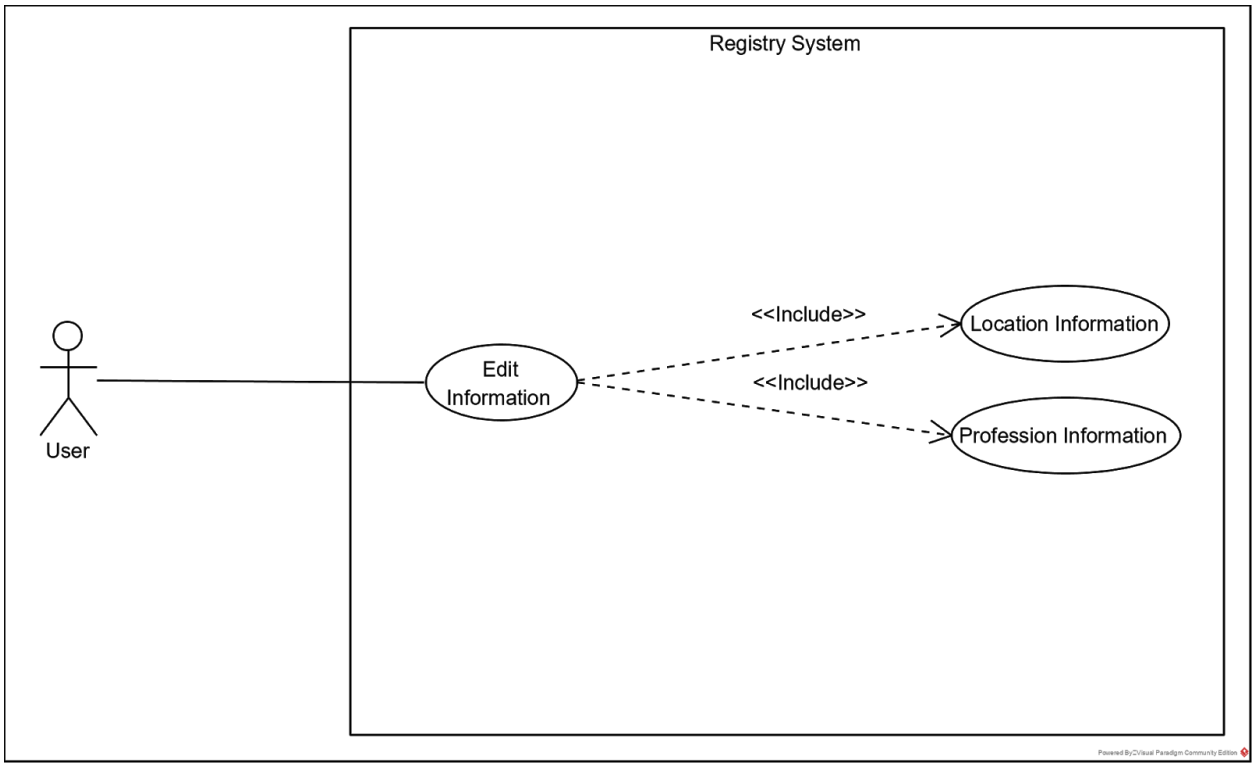

Figure 4.6. Edit Information Use Case Diagram.

Figure 4.7. UC-3 Edit Information Use Case.

Title	Edit Information
Use Case ID	**UC-3**
Description	The user edits details of the information associated with their account.
Primary Actor	• User

Preconditions	● The server is running. ● The user has navigated to the Edit Information Page.
Postconditions	The users information in the database is changed to reflect the edits the user has made.
Main Flow	1. The system retrieves the user's information for fields that are able to be edited from the database. 2. The system presents a form displaying the retrieved user information in their respective fields. 3. The user edits the fields containing information they wish to be altered. 4. The user selects submit. 5. The system retrieves the information from the form. 6. The system stores the information in the database. 7. The system navigates to the User Dashboard.
Alternative Flows	-
Non-Functional Requirements	**QA-5** Users should only be able to edit fields that have been predetermined to likely change over time. **QA-6** The edited information should be added to the database within 10 seconds of the user selecting submit.

4.3.1. Description and Priority

The user is able to edit the information they had previously entered, to better reflect their current circumstance.

Priority: High

4.3.2. Stimulus/Response

The user navigates to the Edit Information Page via a link on the User Dashboard. The user is presented with a form showing the current information about the user that is stored in the database for fields that are able to be edited. The user then alters the fields that are no longer accurate and selects submit. The information is then sent to the server by means of a token. The server then adds the new information about the user in the database. The old information

on the user is kept to allow for the possibility to track trends over time in the future. The server then redirects the user back to the User Dashboard.

4.3.3. Functional Requirements

REQ-6 The user must be able to change previously entered information.
REQ-7 The system must keep archive edited information about the user.

4.4 Query

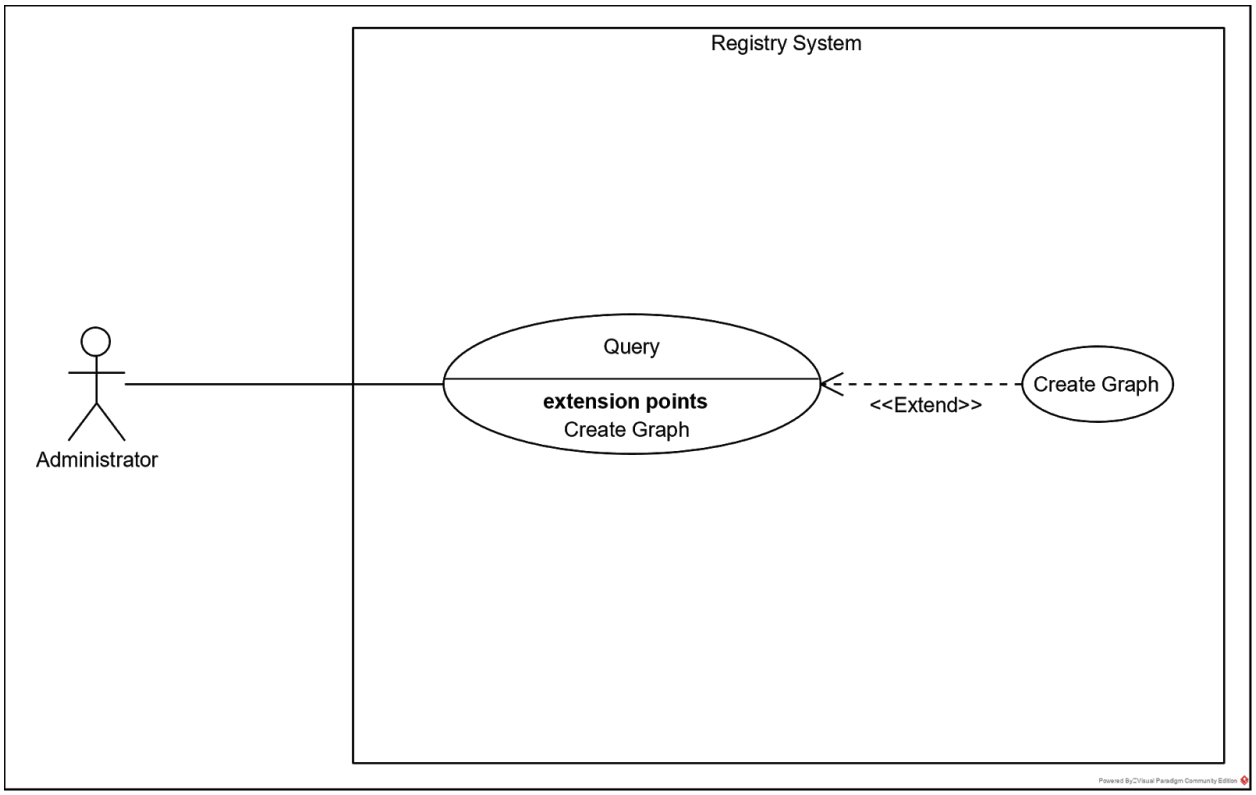

Figure 4.8. Query Use Case Diagram.

Figure 4.9. UC-4 Query Use Case.

Title	Query
Use Case ID	**UC-4**
Description	The administrator queries information in the database.

Primary Actor	● Administrator
Preconditions	● The server is running. ● The administrator is logged in. ● The administrator has navigated to the Query Page.
Postconditions	The administrator is presented with information resulting from the query on the information in the database.
Main Flow	1. The system presents the administrator with options on what information can be queried. 2. The administrator selects the information they wish to query. 3. The administrator selects submit. 4. The system constructs a query to meet the administrator's request. 5. The system accesses the database. 6. The system runs the query. 7. The system retrieves the resulting information. 8. The system presents the information to the administrator.
Alternative Flows	8.1. The administrator selects convert to graph. 8.1.1. The administrator selects the kind of graph to create. 8.1.2. The administrator selects what information from the graph to plot on the graph. 8.1.3. The system constructs the graph. 8.1.4. The system displays the graph.
Non-Functional Requirements	**QA-7** The result of a query should be displayed within 10 seconds of the query being run.

4.4.1. Description and Priority

An administrator creates a query to find relevant information about users, which is run against the information in the database. The information is then presented to the administrator in a

tabular format. The administrator is then given the option to convert the table of information to a graph format to be more human readable.

Priority: High

4.4.2. Stimulus/Response

The administrator navigates to the Query Page from the Administrator Dashboard. The administrator then selects, the kind of query they wish to perform and, the filters for the information in the database. The administrator the selects submit and the query specifications are sent to the server. The server then constructs a query, runs the query against the database and retrieves the results. The server then sends the results of the query to the client, directs the administrator to a new page and the client displays the results of the query in a table. The administrator is then presented the option to convert the table into a graph. If the administrator selects this option they are prompted to select what type of graph they wish to create and what information from the table to plot on the graph. Once the settings have been selected, the client then generates the graph and displays it to the administrator.

4.4.3. Functional Requirements

REQ-8 The system must allow administrators to construct specific queries to gather specific information from the database.
REQ-9 The system must display the results of a query in the form of a table.
REQ-10 The system must allow administrators to generate graphs from the contents of tables.

4.5 Edit User Information

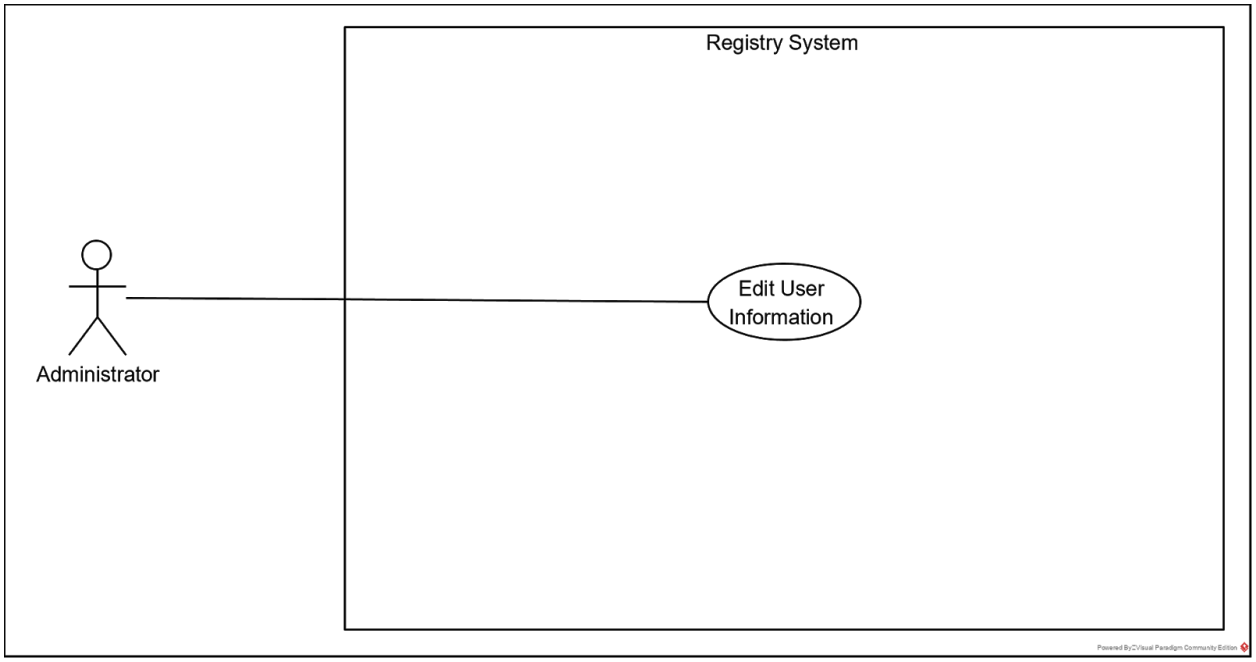

Figure 4.10. Edit User Information Use Case Diagram.

Figure 4.11. UC-5 Edit User Information Use Case.

Title	Edit User Information
Use Case ID	**UC-5**
Description	The administrator edits a user's information on the user's behalf.
Primary Actor	● User
Preconditions	● The server is running. ● The administrator is logged in. ● The administrator has navigated to the Edit User Information Page.
Postconditions	The users information in the database is changed to reflect the edits the administrator has made.

Main Flow	1. The administrator enters the userID of the user whose information they wish to edit. 2. The system accesses the database. 3. The system searches for a user whose ID matches the ID entered. 4. The system navigates to a new page. 5. The system displays the Edit User Form. 6. The administrator edits the necessary user information. 7. The administrator selects submit. 8. The system uploads the new information to the database. 9. The system navigates to the Administrator Dashboard.
Alternative Flows	-
Non-Functional Requirements	**QA-8** Administrators should only be able edit a predetermined selection of fields in the user's information. **QA-9** Only administrators of a sufficient permission level should be able to edit user information. **QA-10** The edited user information should be uploaded to the database within 10 seconds of the administrator selecting submit.

4.5.1. Description and Priority

The administrator edits a user's information on the user's behalf.

Priority: Low

4.5.2. Stimulus/Response

The administrator navigates to the Edit User Information Page via a link on the Administrator Dashboard. The administrator enters the userID of the user whose information they wish to edit and selects the edit option. The userID is sent to the server, which accesses the database and searches for a user whose ID matches the ID entered and retrieves the information for the fields that administrators are permitted to edit. The server then directs the administrator to a new page and displays the Edit User Form.

The administrator enters the necessary user information for the fields to be changed. The administrator then selects submit and the new information is sent to the server, where its uploaded to the database. The server then directs the administrator back to the Administrator Dashboard.

4.5.3. Functional Requirements

REQ-11 Administrators must be able to edit user information.

4.6 Delete User

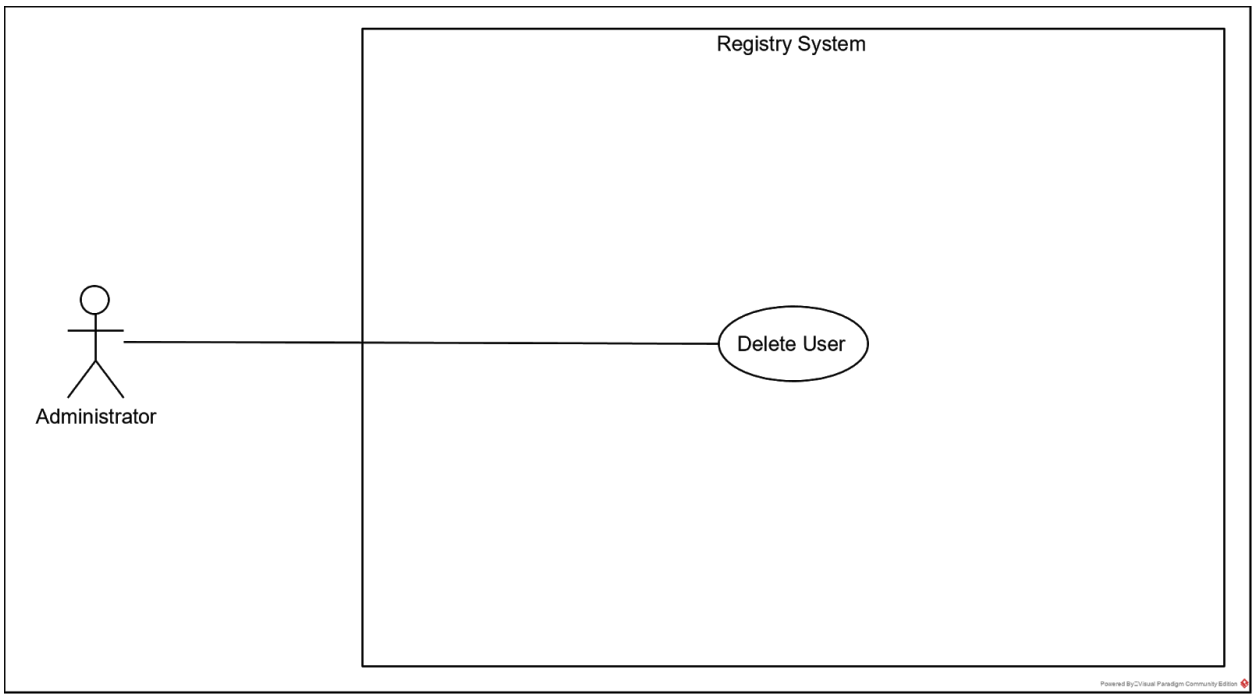

Figure 4.12. Delete User Use Case Diagram.

Figure 4.13. UC-6 Delete User Use Case.

Title	Delete User
Use Case ID	**UC-6**
Description	An administrator removes a user account from the system.

Primary Actor	● Administrator
Preconditions	● The server is running. ● The administrator is logged in. ● The administrator has navigated to the Delete User Page
Postconditions	The selected user account is deleted from the database.
Main Flow	1. The administrator enters the ID of the user to be deleted. 2. The system accesses the database and checks if the user account exists. 3. The system confirms the existence of the account to the administrator. 4. The system prompts the administrator to confirm. 5. The administrator confirms the deletion of the account. 6. The system archives the user account. 7. The system deletes the user account and information from the database. 8. The application navigates to the dashboard.
Alternative Flows	-
Non-Functional Requirements	**QA-11** Deleted user accounts should be archived when deleted from the system, so an account can be restored in the event of an incorrect deletion. **QA-12** Only administrators of a sufficient permission level should be able to delete users. **QA-13** Users should be unable to be deleted unless they meet a set of predetermined requirements.

4.6.1. Description and Priority

An administrator removes a user account that meets certain requirements, from the system.

Priority: Low

4.6.2. Stimulus/Response

The administrator navigates to the Delete User Page via a link on the Administrator Dashboard. The administrator enters the ID of the user to be deleted, the ID is sent to the server where the server accesses the database and checks if the user account exists and is eligible to be deleted. The server confirms the existence of the account to the administrator and displays additional information about the user to allow the administrator to confirm this is the right user to delete. The administrator is then asked to confirm that they wish to delete this user. The administrator confirms the deletion of the account and the server archives the user account, deletes the user account and information from the database, and directs the administrator back to the Administrator Dashboard.

4.6.3. Functional Requirements

REQ-12 Administrators must be able to remove user accounts from the system.

4.7 Create New Administrator

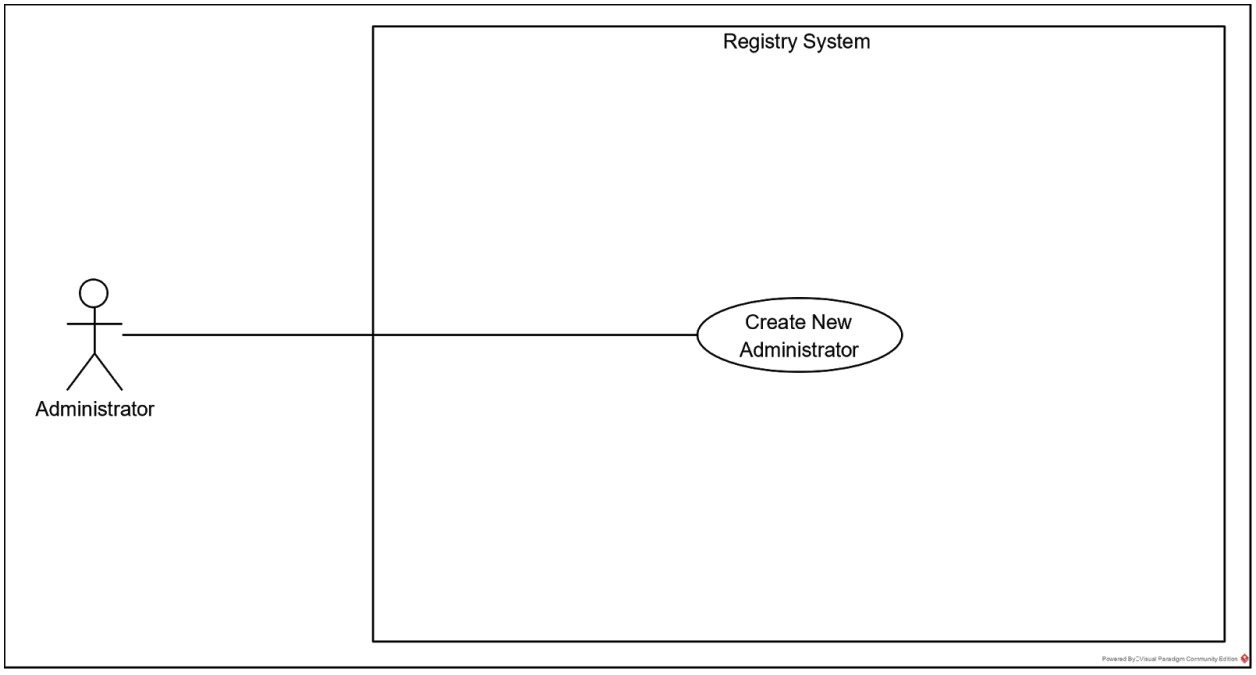

Figure 4.14. Create New Administrator Use Case Diagram.

Figure 4.15. UC-7 Create New Administrator Use Case.

Title	Create New Administrator

Use Case ID	**UC-7**
Description	An administrator creates a new administrator account.
Primary Actor	● Administrator
Preconditions	● The server is running. ● The administrator is logged in. ● The administrator has navigated to the Create New Administrator Page.
Postconditions	A new administrator account is added to the system.
Main Flow	1. The administrator selects create new administrator 2. The application navigates to Create Administrator Page. 3. The administrator enters the login credentials for the new administrator. 4. The application adds the new account to the database. 5. The application navigates to the Admin Dashboard.
Alternative Flows	-
Non-Functional Requirements	**QA-14** Only administrators of a sufficient permission level should be able to create new administrator accounts.

4.7.1. Description and Priority

An administrator creates a new administrator account.

Priority: Medium

4.7.2. Stimulus/Response

The administrator navigates to the Create Administrator Page via a link on the Administrator Dashboard. The administrator enters the login credentials for the new administrator and selects submit. The information is sent to the server which adds the new account to the database. The server then navigates the administrator back to the Administrator Dashboard.

4.7.3. Functional Requirements

REQ-13 Administrators must be able to create new administrator accounts.

4.8 Delete Administrator

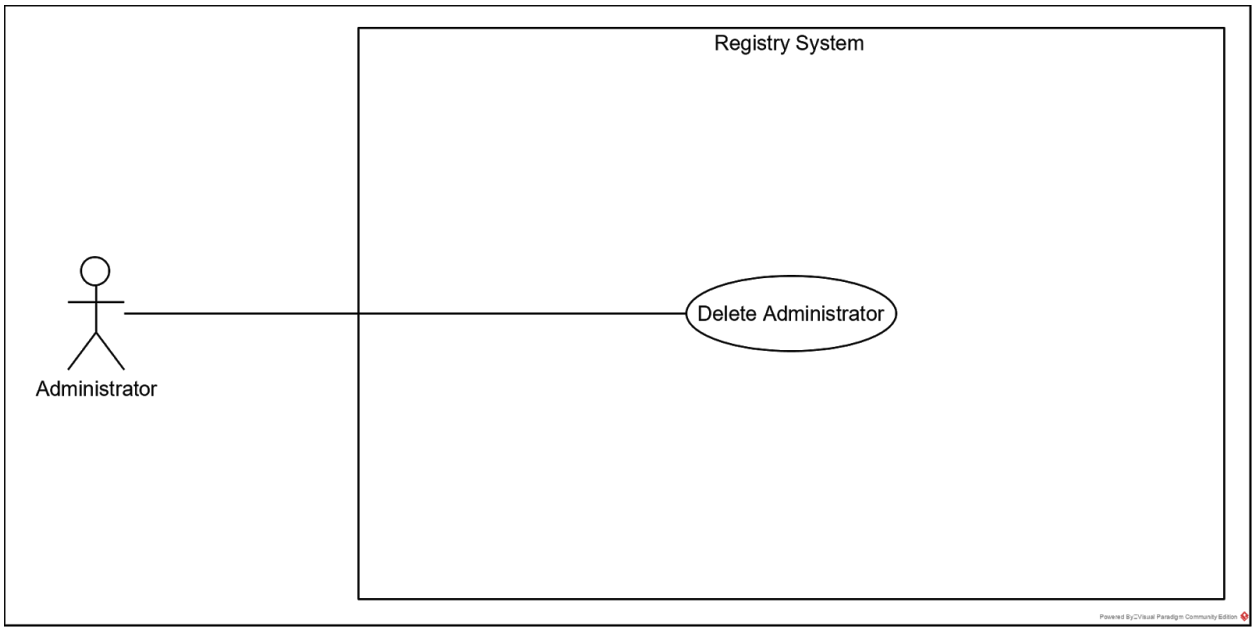

Figure 4.16. Delete Administrator Use Case Diagram.

Figure 4.17. UC-8 Delete Administrator Use Case.

Title	Delete Administrator
Use Case ID	**UC-8**
Description	An administrator deletes another administrator account.
Primary Actor	● Administrator
Preconditions	● The server is running. ● The administrator is logged in. ● The administrator has navigated to the Delete Administrator Page.
Postconditions	The desired administrator account is removed from the system.

Main Flow	1. The administrator enters an administratorID of an account that is to be deleted. 2. The system checks the database for an administrator account with a matching ID. 3. The system confirms the administrator account exists. 4. The system prompts the administrator to confirm this account is to be deleted. 5. The administrator confirms the account is to be deleted. 6. The system archives the account. 7. The system deletes the account from the database.
Alternative Flows	-
Non-Functional Requirements	**QA-15** Only administrators of a sufficient permission level should be able to delete administrators. **QA-16** An administrator should not be able to delete their own account.

4.8.1. Description and Priority

An administrator deletes a different administrator account.

Priority: Low

4.8.2. Stimulus/Response

The administrator navigates to the Delete Administrator Page via a link on the Administrator Dashboard. The admin enters the ID of the administrator account they wish to delete and selects submit. The ID is sent to the server which searches the database for an administrator account with a matching ID. The server then directs the administrator to a new page where the existence of the administrator account is confirmed and additional information about the account is presented to the administrator to ensure that this is the correct account to delete. The administrator is then asked to confirm they wish to delete this account. The administrator confirms and the server then archives the account and deletes the account from the system. The server then directs the administrator back to the Administrator Dashboard.

4.8.3. Functional Requirements

REQ-14 Administrators must be able to delete other administrator accounts.

4.9 Send Update Reminder Email

4.9.1. Description and Priority

The system sends an email reminder to users to update their information after 6 months. This is to ensure that the information in the system is as current as possible.

Priority: Medium

4.9.2. Stimulus/Response

At a fixed time once a month the system queries the database for all users accounts that have not been updated in the range of 6-7 months. The system retrieves the email addresses associated with these accounts and sends a customized email to the user reminding them to update their account information.

4.9.3. Functional Requirements

REQ-15 The system must send emails to users who have not updated their information for 6 months, an email reminder to check their information.

5. Other Nonfunctional Requirement

5.1 Performance Requirements

QA-2 It should take no more than 1 second to verify the user's credentials from the user selecting login.
QA-4 The information from the form should be added to the database within 10 seconds of submitted being selected.
QA-6 The edited information should be added to the database within 10 seconds of the user selecting submit.
QA-7 The result of a query should be displayed within 10 seconds of the query being run.
QA-10 The edited user information should be uploaded to the database within 10 seconds of the administrator selecting submit.
QA-17 The system should be able to support x simultaneous users.
QA-18 The mean time to view a webpage over a 25 Mbps shall not exceed 0.5 seconds.

5.2 Security Requirements

QA-1 Passwords should be encrypted when stored in the database.
QA-5 Users should only be able to edit fields that have been predetermined to likely change over time.
QA-8 Administrators should only be able edit a predetermined selection of fields in the user's information.
QA-9 Only administrators of a sufficient permission level should be able to edit user information.
QA-11 Deleted user accounts should be archived when deleted from the system, so an account can be restored in the event of an incorrect deletion.
QA-12 Only administrators of a sufficient permission level should be able to delete users.
QA-13 Users should be unable to be deleted unless they meet a set of predetermined requirements.
QA-14 Only administrators of a sufficient permission level should be able to create new administrator accounts.
QA-15 Only administrators of a sufficient permission level should be able to delete administrators.
QA-16 An administrator should not be able to delete their own account.
QA-19 The system shall provide password protected access to web pages that are to be viewed only by administrators.
QA-20 The system shall provide email verification as an extra step of security.
QA-21 The system should require users to verify the validity of their login credentials when registering an account by sending a verification email to the user.

5.3 Software Quality Attributes

QA-3 The User Information Form should be split into multiple smaller sections.

Appendix A: Glossary

Dashboard - A web page that provides a user with important information and links out to pages where users can access the system's other functionality.
Diaspora - Nationals living abroad.

Appendix B: Analysis Models

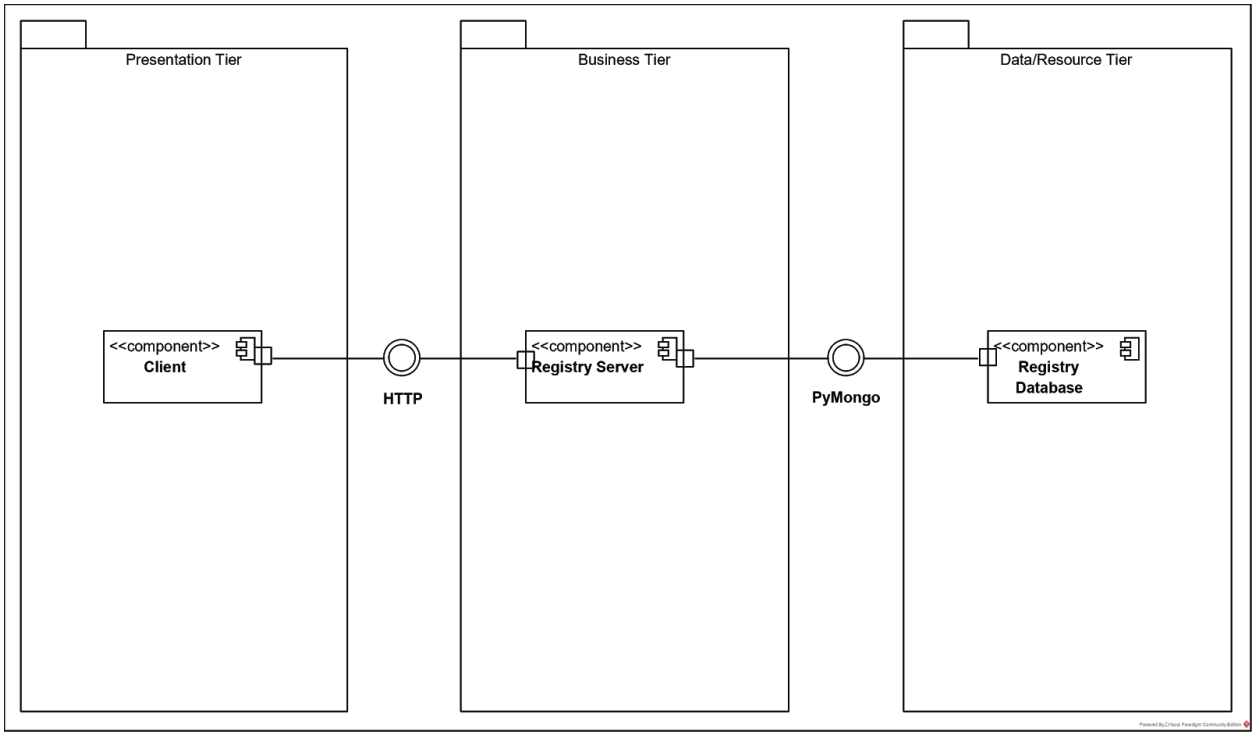

Figure 7.1. System Component Diagram.

The system will be designed to fit the standard 3-Tier architecture that is common for web applications, consisting of a Presentation Tier, a Business Tier and a Data/Resource Tier.

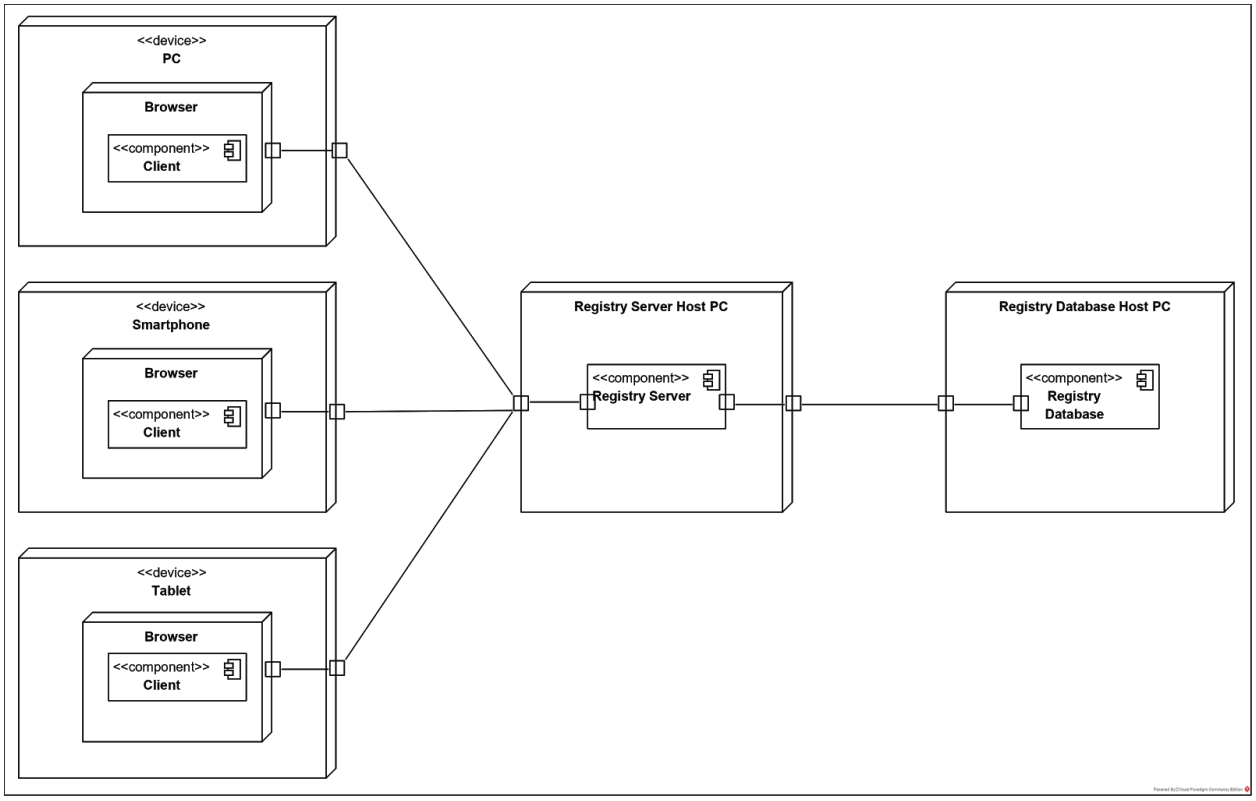

Figure 7.2. System Deployment Diagram.

Appendix C: To Be Determined

Permission Levels - What levels of permissions administrators are required to have in order to access each of the system's functionality.

Requirements to be eligible to be deleted - The specific requirements a User Account must meet in order for an administrator to be able to delete the account.

User editable user information - Which fields in the user information are editable by a user.

Administrator editable user information - Which fields in the user information are editable by an administrator.

Appendix D: To be Completed

Features to be completed in the future. Development of these features are on hold until the client has mad decisions pertaining to the features' implementation.

REQ-2 Administrators must be able to login via webpage to gain access to the system.
REQ-6 The user must be able to change previously entered information.
REQ-7 The system must keep archive edited information about the user.
REQ-8 The system must allow administrators to construct specific queries to gather specific information from the database.
REQ-9 The system must display the results of a query in the form of a table.
REQ-10 The system must allow administrators to generate graphs from the contents of tables.
REQ-11 Administrators must be able to edit user information.
REQ-12 Administrators must be able to remove user accounts from the system.
REQ-13 Administrators must be able to create new administrator accounts.
REQ-14 Administrators must be able to delete other administrator accounts.
REQ-15 The system must send emails to users who have not updated their information for 6 months, an email reminder to check their information.