**Self-Training Project: Messenger**

[Brief description 2](#_Toc394860274)

[User roles 2](#_Toc394860275)

[Common functionality 2](#_Toc394860276)

[Functionality for Administrator: 2](#_Toc394860277)

[Functionality for Regular Users: 3](#_Toc394860278)

[General Notes 3](#_Toc394860279)

[Steps in developing the project 3](#_Toc394860280)

## Brief description

The application: **Simple Messaging System**

The system should allow all users to login and logout (by using unique credentials), also the system should allow regular users to register. When logged in the regular users should be able to send messages to other regular users and to see their own sent and received messages.

## User roles

* Administrator
* Regular User

## Common functionality

* Login/Logout : Users should be able to login logout by introducing their correct (there is a match in the system/Database) credentials (username/password) into a general login page and they should be redirected to a specific welcome page (based on the users type). If the credentials are wrong error message should be displayed.
* Registration : Only **Regular Users** shouldbe able to register by entering their specific data (Username, Password, Name, Birthdate, Gender) in a registration form. After registering users should be able to login only after the **Administrator** confirmed them in the system

### Functionality for Administrator

* 1. ***Main Welcome Page*** : this page should be displayed after login, showing general information about the system (number of users, number of messages, etc.)
  2. ***Users Page*** : Showing a list of users with all their details and for the fresh registered users an option (button or link) to confirm their data
  3. ***Messages***: Showing a list of all the messages in the system. For a message it should show the following : Sender Name, Receiver Name, Sent Date, Subject . The subject of each message should be clickable, when clicked a popup should appear showing all the details of the message.

### Functionality for Regular Users

1. ***Main Welcome Page :*** this page should be displayed after login, it should present general information about the user (number of messages sent, number of messages received, number of unread messages)
2. ***Inbox :*** This page will display a list of all the messages received by the user (Sender Name, Received Date, Subject). The unread messages should be highlighted. The subject of each message should be clickable, when clicked a popup should appear showing all the details of the message.
3. ***Outbox :*** This page should display all the messages sent by the user. The subject of each message should be clickable, when clicked a popup should appear showing all the details of the message.
4. ***Send Message :*** This page should allow the current user to send a new messages. It should allow the user to select a recipient (from the registered and confirmed users) introduce the subject and the body of the message. When clicking the “Send Message” button the system should send the message and show a confirmation dialog (this should be an AJAX call).

## General Notes

* 1. All the pages should have a common Master Page
  2. The application should have a Data Layer (separate class for manipulating the DB)
  3. All pages should have their own CSS files for styling

## Steps in developing the project

* 1. Planning/Functional requirements - identify use cases, scenarios that may appear
  2. Database design
  3. Creating a console application which will connect to the DB and will have the basic functionality for querying/manipulating the DB, this will be the Data Layer of the web project. For example it will consist a single class which :
     + Checks user credentials (login)
     + Inserts users in the DB (registration)
     + Gets all the users/messages in the system
     + Gets the sent messages for a specific user
     + Gets the received messages for a given user
     + Etc…
  4. Create the MVC 4 application and add it to the solution the class developed in the previous step, implement Login/logout functionality
  5. Implement registration functionality
  6. Implement user specific pages.