

**Unit:**

**Designing and Developing Object-Oriented**

**Computer Programs**

Assignment title:

**Reference Tracker**

**December 2018**



**Important Notes:**

* Please refer to the Assignment Presentation Requirements for advice on how to set out your assignment. These can be found on the NCC Education *Campus*. Scroll down the left-hand side of the screen until you reach Personal Support. Click on this, and then on Policies and Advice. You will find the Assignment Presentation Requirements under the Advice section.
* You must familiarize yourself with the NCC Education Academic Dishonesty and Plagiarism Policy and ensure that you acknowledge all the sources which you use in your work. The policy is available on *Campus*. Follow the instructions above but click on Policies rather than Advice.
* You must complete the **‘Statement and Confirmation’**. The form is available on the Policies section of *Campus*. Scroll down the left-hand side until you reach Personal Support. Click on this and then click on Policies and Advice.
* You must submit a paper copy and digital copy (on disk or similarly acceptable medium). Media containing viruses, or media which cannot be run directly, will result in a fail grade being awarded for this module.
* All electronic media will be checked for plagiarism.
* This assessment is worth **100%** of the student’s overall grade.

|  |  |  |
| --- | --- | --- |
| Student Name: | Student Number: | Centre: DIA |
| Fahad Md. Kamal | 00171328 |  |



|  |  |  |
| --- | --- | --- |
| Marker’s comments: | | |
| Moderator’s Comments: | | |
|  | | |
|  | | |
|  | | |
|  | | |
| Mark: | **Moderated** | **Final** |
|  | **Mark:** | **Mark:** |



|  |  |  |
| --- | --- | --- |
| ***Each time you submit an assignment you must attach this statement as the cover page for both the hard copy and the electronic version. If the statement is missing your work will not be marked.*** | | |
| Student Declaration | | |
| I have read and understood NCC Education’s Policy on Academic Dishonesty and Plagiarism.  I can confirm the following details: | | |
| Student ID/Registration number: | | 00171328 |
| Name: | | Fahad Md. Kamal |
| Centre Name: | | Daffodil International Academy (DIA) |
| Module Name: | | DDooCP |
| Module Leader: | | Rakibul Hasan |
| Number of words: | |  |
| I confirm that this is my own work and that I have not plagiarized any part of it. I have also noted the assessment criteria and pass mark for assignments. | | |
| Due Date: | | 20.10.2018 |
| Student Signature: | |  |
| Submitted Date: | 20.10.2018 | |



Table of Contents

[Introduction: 4](#_Toc528610460)

[Task-1: 4](#_Toc528610461)

[Form-GUI 5](#_Toc528610462)

[Form-1: User Log in form. 5](#_Toc528610463)

[Form-2: Dashboard Form. 5](#_Toc528610464)

[Form-3: Admin login and Sign Up form with admin control panel. 6](#_Toc528610465)

[Form-3: User Login and signup form 6](#_Toc528610466)

[Form-4: Reference Creation form. 7](#_Toc528610467)

[Form-5: Word Processing form. 8](#_Toc528610468)

[Task-2 9](#_Toc528610469)

[Testing plans and Testing 9](#_Toc528610470)

[Blackbox Testing plan: 9](#_Toc528610471)

[Blackbox Testing: 10](#_Toc528610472)

[White Box Testing Plan: 16](#_Toc528610473)

[Task-3: 28](#_Toc528610474)

[UI-Prototypes: 29](#_Toc528610475)

[Login form (frmLogIn) 29](#_Toc528610476)

[Dashboard Form (frmDeshBoard): 29](#_Toc528610477)

[Form Sign Up (frmCreateAccount): 30](#_Toc528610478)

[Form Reference (frmReference): 30](#_Toc528610479)

[Form Article (frmArticles): 31](#_Toc528610480)

[Custom Message Box(MsgBox): 31](#_Toc528610481)

[Description: 32](#_Toc528610482)

[Overview of user experience: 32](#_Toc528610483)

[Recommendations for Changes: 32](#_Toc528610484)

[Conclusion: 33](#_Toc528610485)

## Introduction:

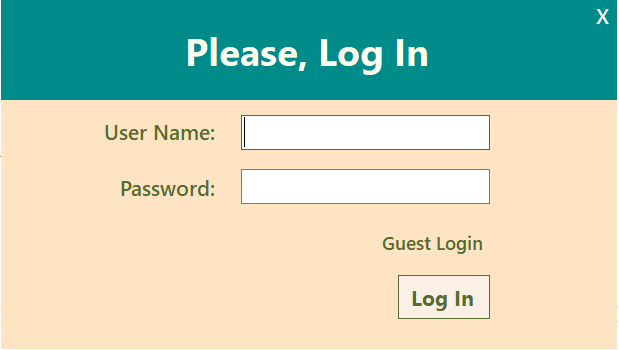
The assignment application “Reference Tracker” was built to enable users to make citation to their articles and print them out if for the sake of avoiding plagiarism crime. The application was built with full-filling all the requirements of the assignment. Here the programming language C-Sharp (C#) was used and the Microsoft’s Visual Studio 2015 was with C# 6 and windows form application. I have done white box testing and black box testing by following some testing plan. For the black box testing purpose I also took a user’s review.

# Task-1:

The application Reference Tracker was made for writhing book, essay or academic paper through proper citation for ensuring honesty in writing. However, the application contains two parts, the GUI or Graphical User Interface and the coding part. At first the GUI part is shown Below:

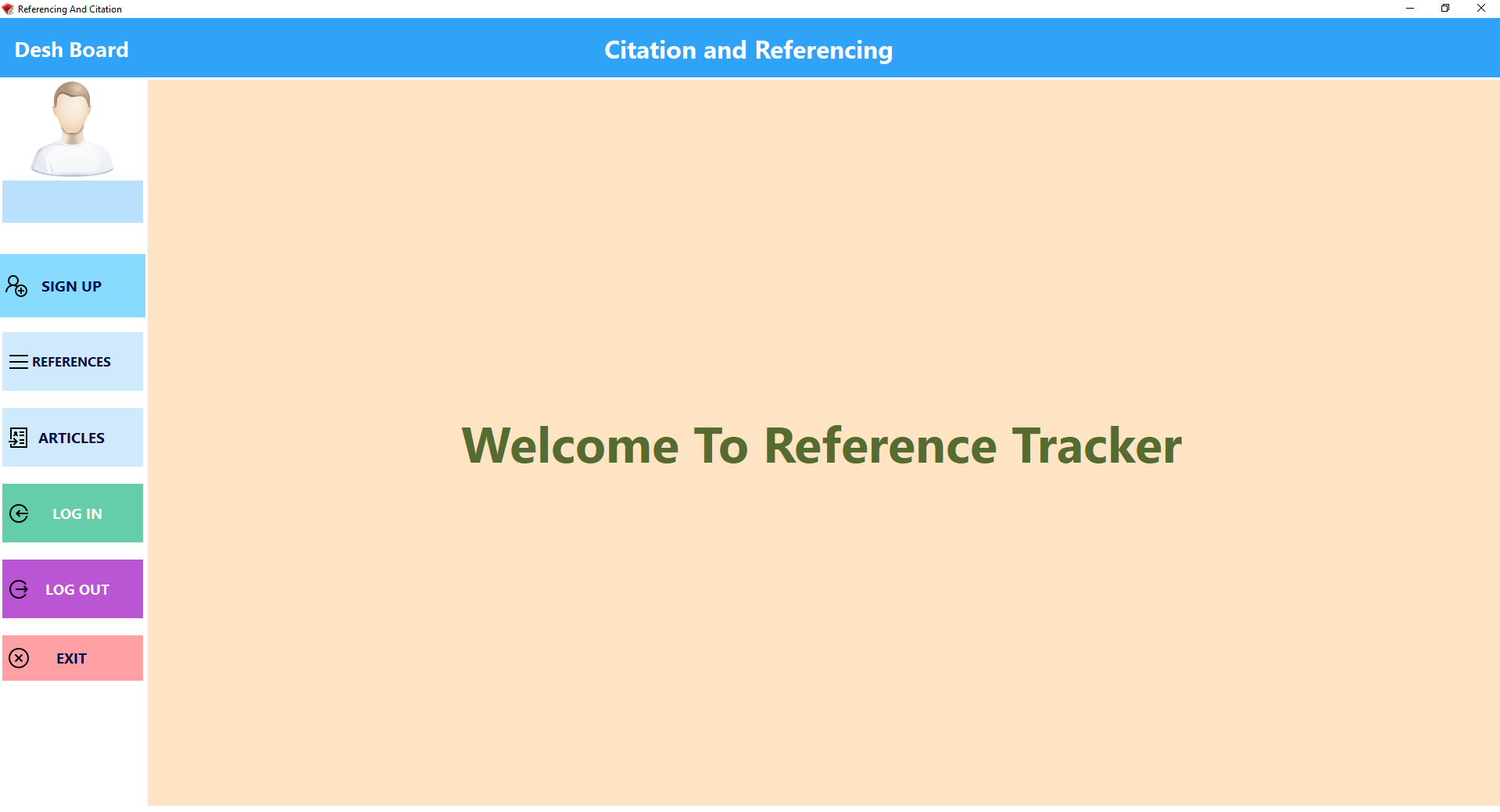
## Form-GUI

### Form-1: User Log in form.



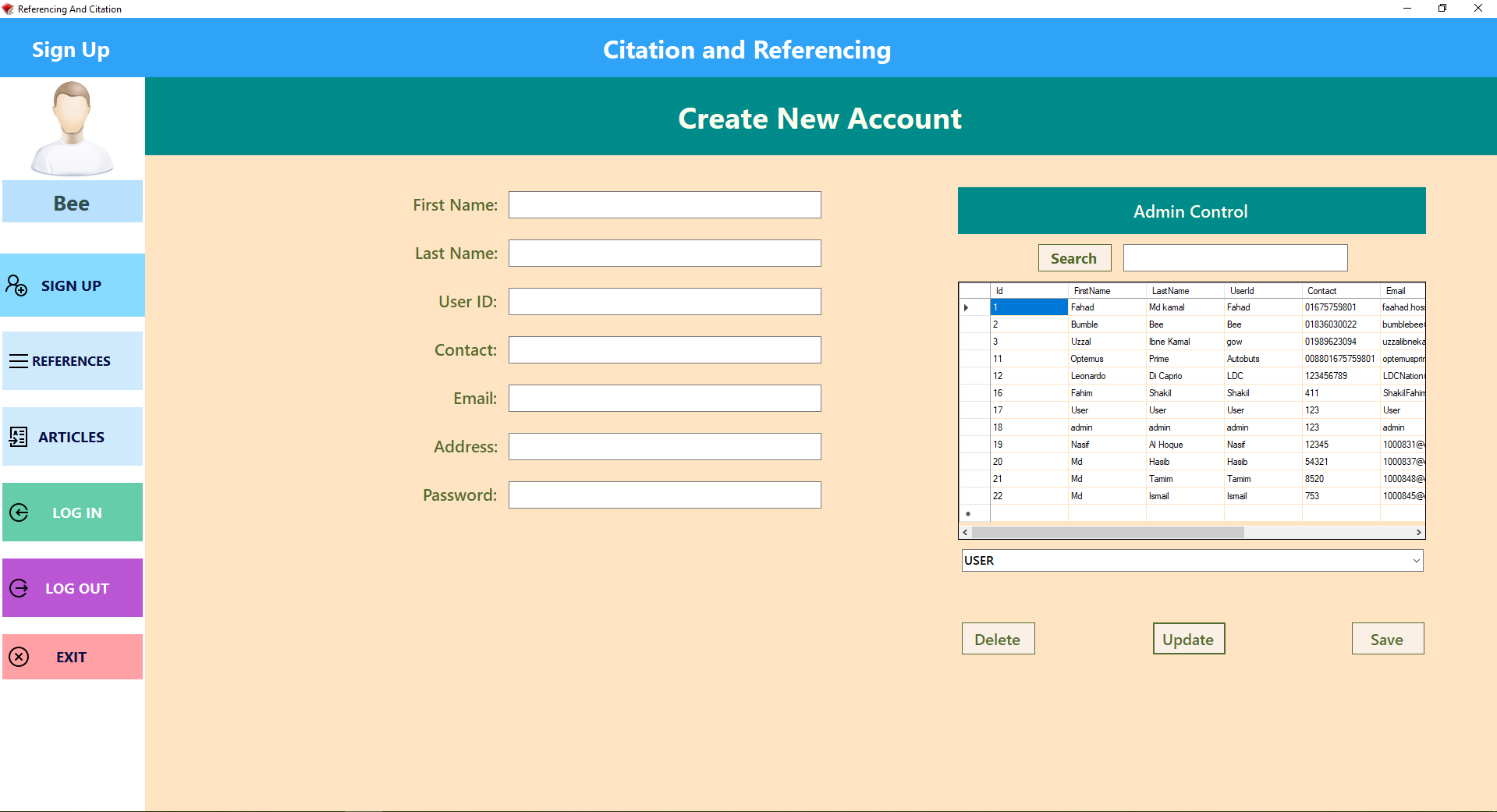
Form- 1: User-Log-In.

### Form-2: Dashboard Form.



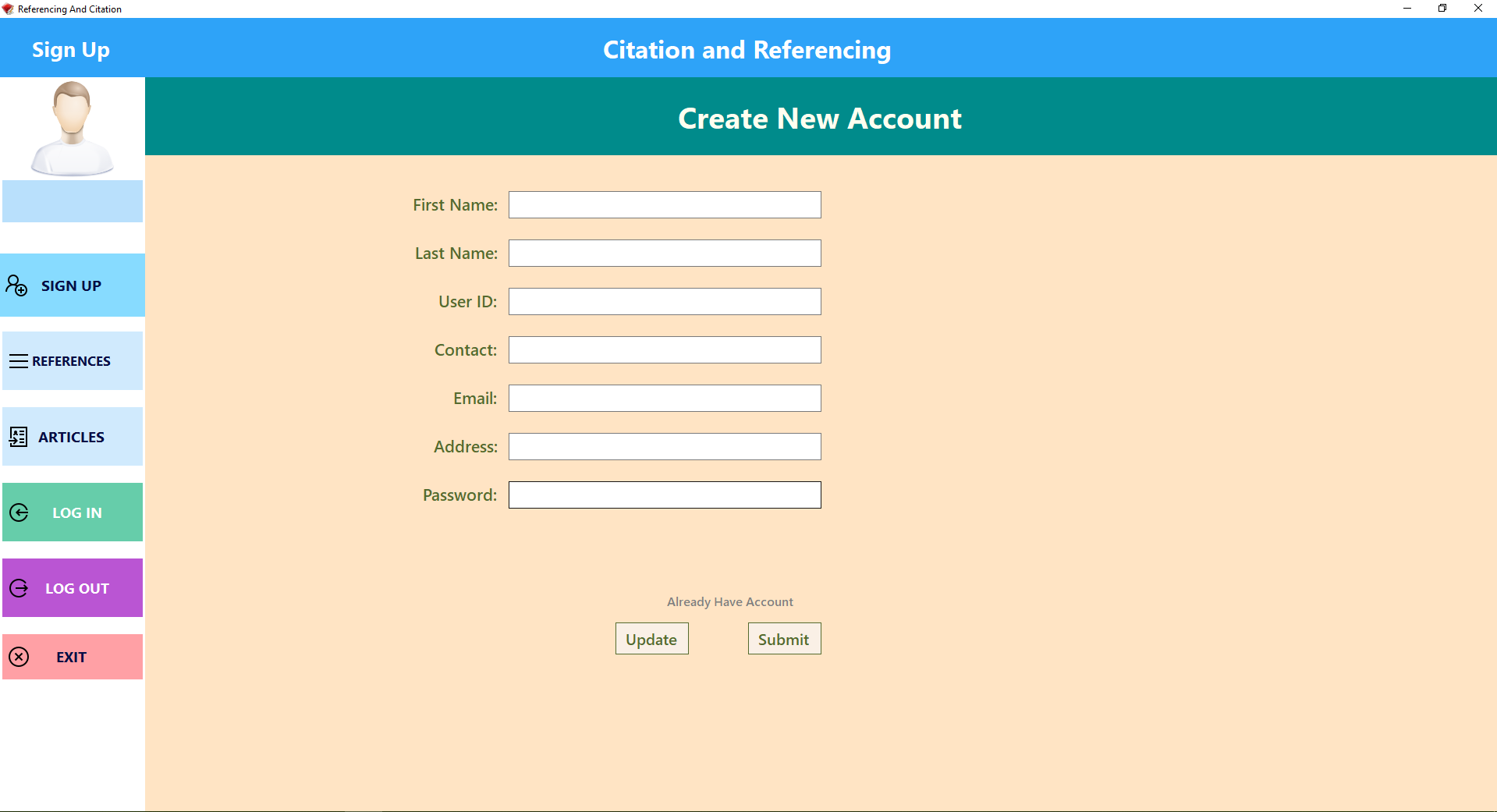
Form- 2:Dashboard-Form.

### Form-3: Admin login and Sign Up form with admin control panel.



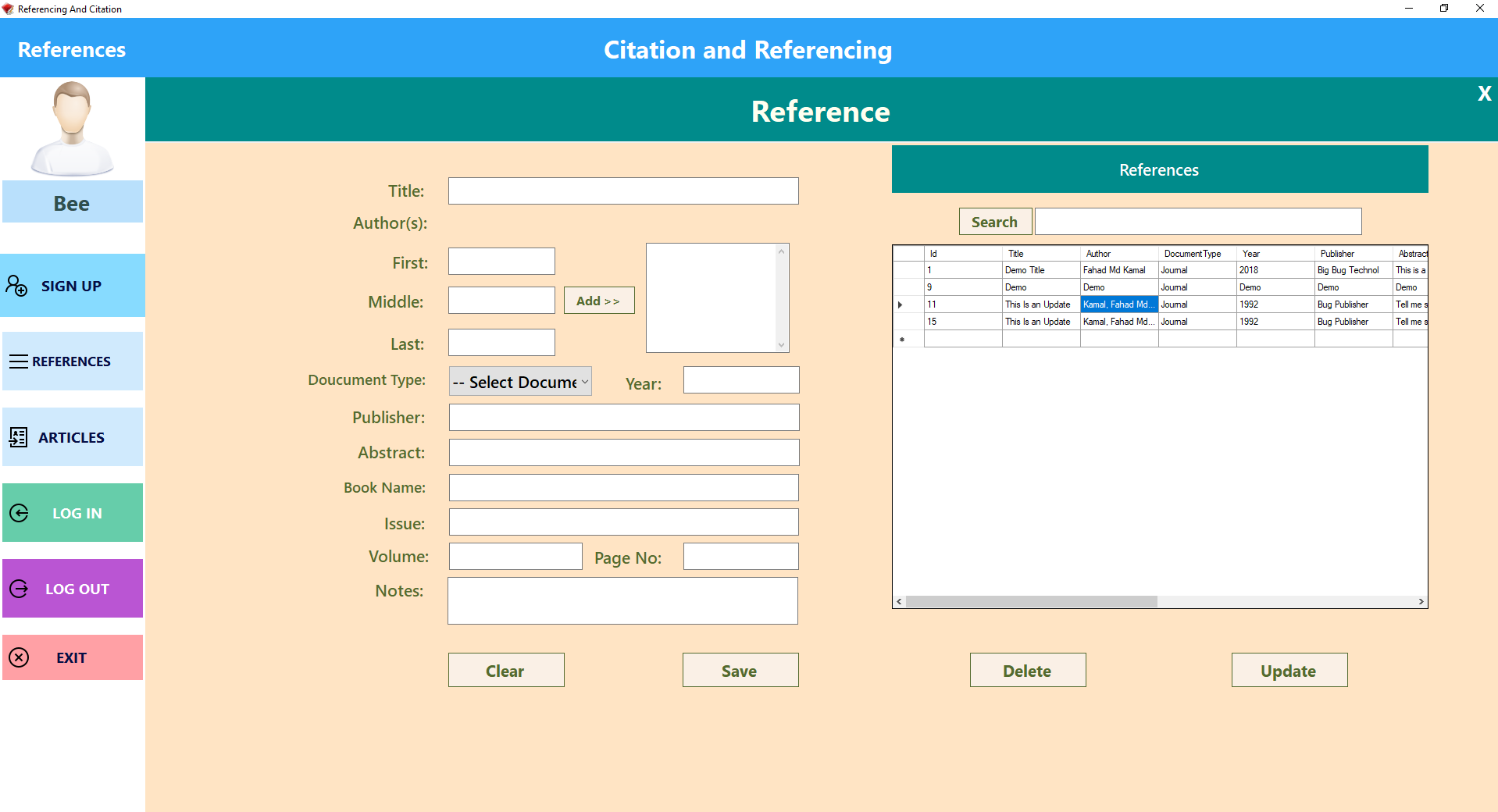
Form- 3:Admin-login-and-SignUp-form-with-admin-control-panel.

### Form-3: User Login and signup form



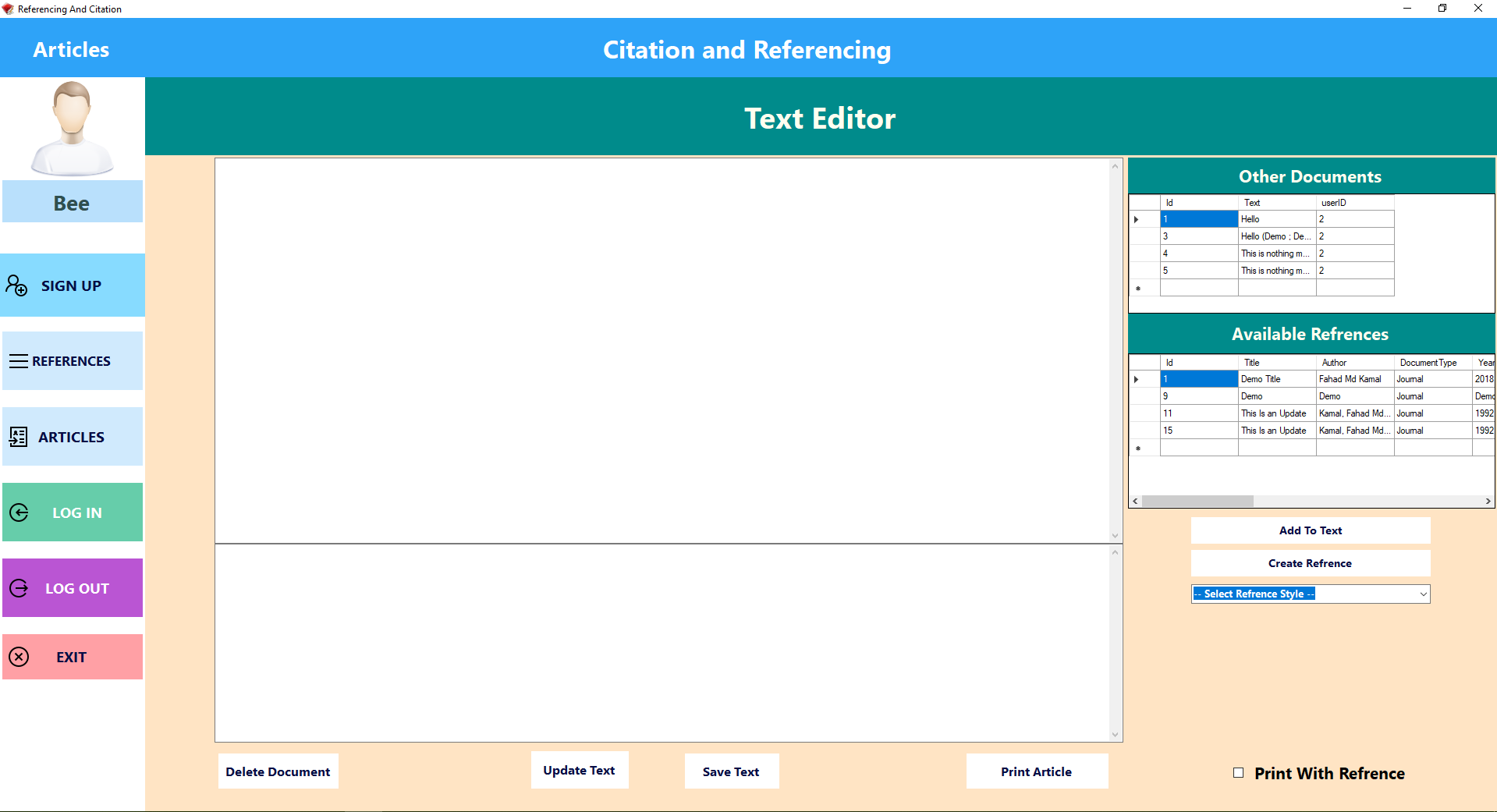
Form- 4:User-Login-and-signup-form

### Form-4: Reference Creation form.



Form- 5:Reference-Creation-form

### Form-5: Word Processing form.



Form- 6:Word-Processing-form.

# Task-2

In this section testing of the newly developed application with the use of appropriate testing data. Black box and white box testing was covered for all equivalence classes and their functions. The testing steps and testing planes were followed among the below processes:

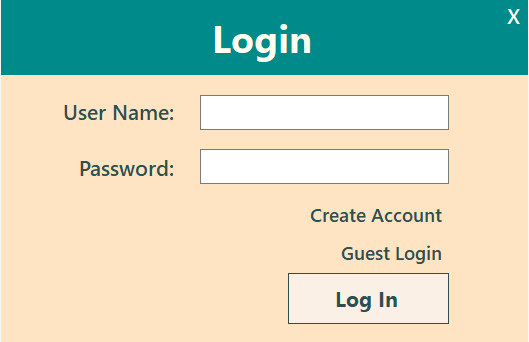
## Testing plans and Testing

### Blackbox Testing plan:

1. At first the user needs to input user name and password into the login form for user log in.
2. If the user is not registered, then they can run the application using Guest login option from the log in window and then from signup form they can create their account.
3. If the user successfully logs in than the dashboard form will show up where the buttons are available to go on different pages of the application.
4. From sign up page users can update their information if they are already logged in. However, if the user is an admin they can update other’s information along with deleting and adding user’s information.
5. The users can create references and add those to the text and print them. But cannot edit or delete other user’s references except using them.
6. However, if the user is as guest they cannot save references.
7. Users can only read and edit their own saved document files.
8. They can choose the referencing style in instructed referencing styles such as Harvard, APA and Chicago styles.
9. Finally, they can print their document files in text files with or without Bibliography of references.

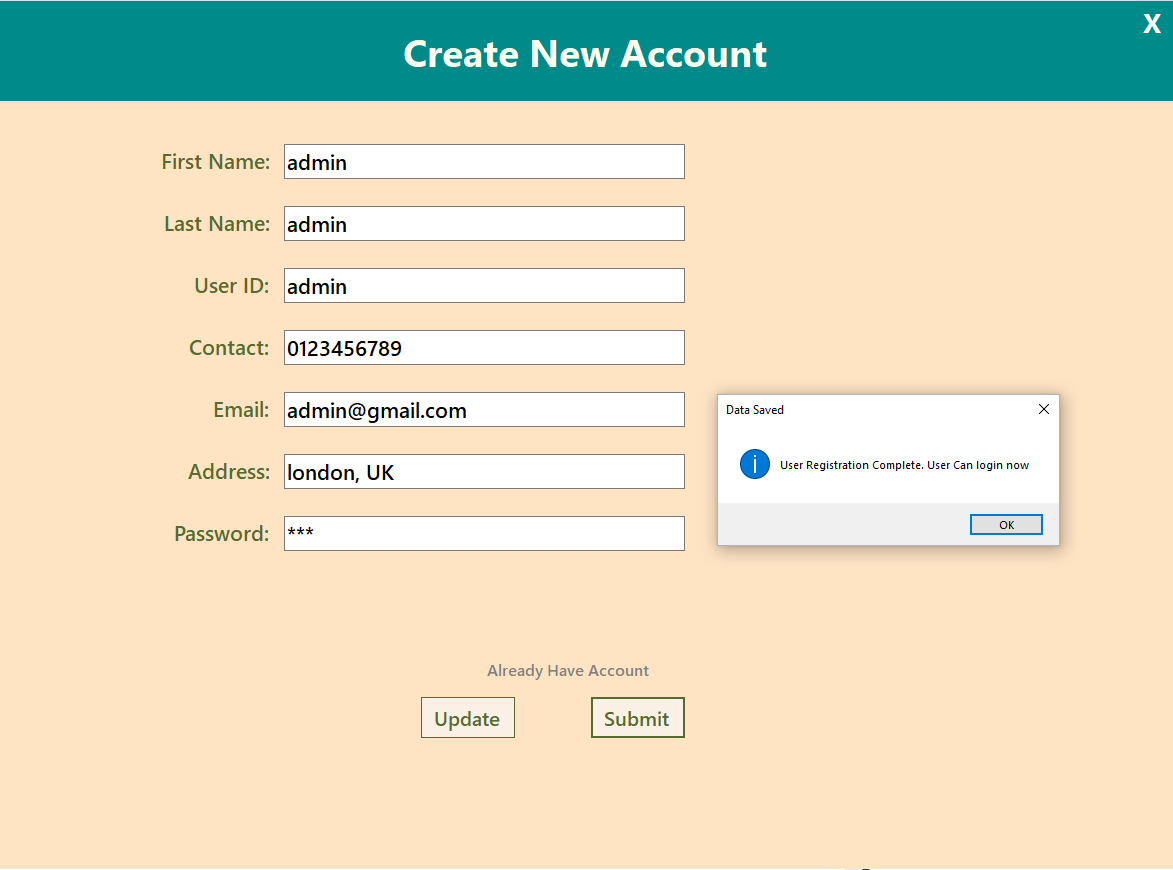
### Blackbox Testing:

1. Login form shows when the application runs. If the user is registered than they can log in else, they can create new account by pressing the create account label.



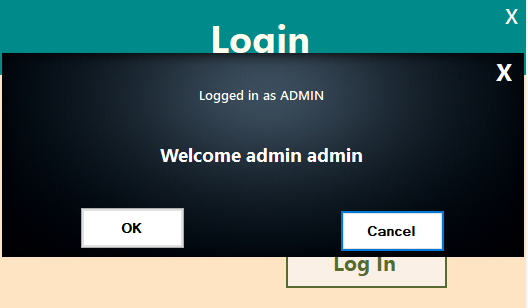
BlackBox-Testing-1

1. After filling all the required fields. User can create user account as regular user. by pressing the submit button. Then the user needs to login with his user ID. However, for admin facilities, an admin needs to update the user as admin.



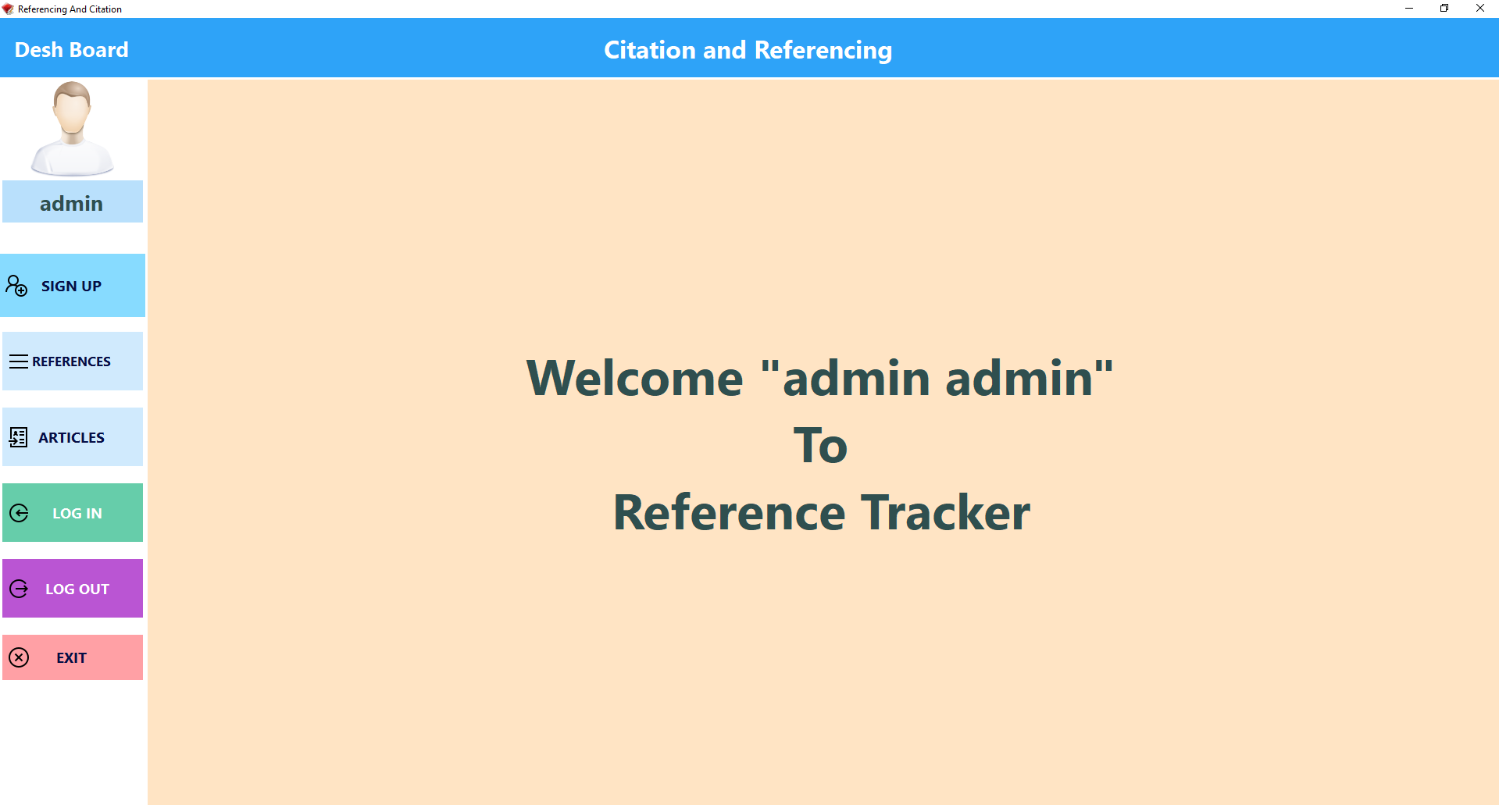
BlackBox-Testing-2

1. After filling the User ID and Password user can log in.



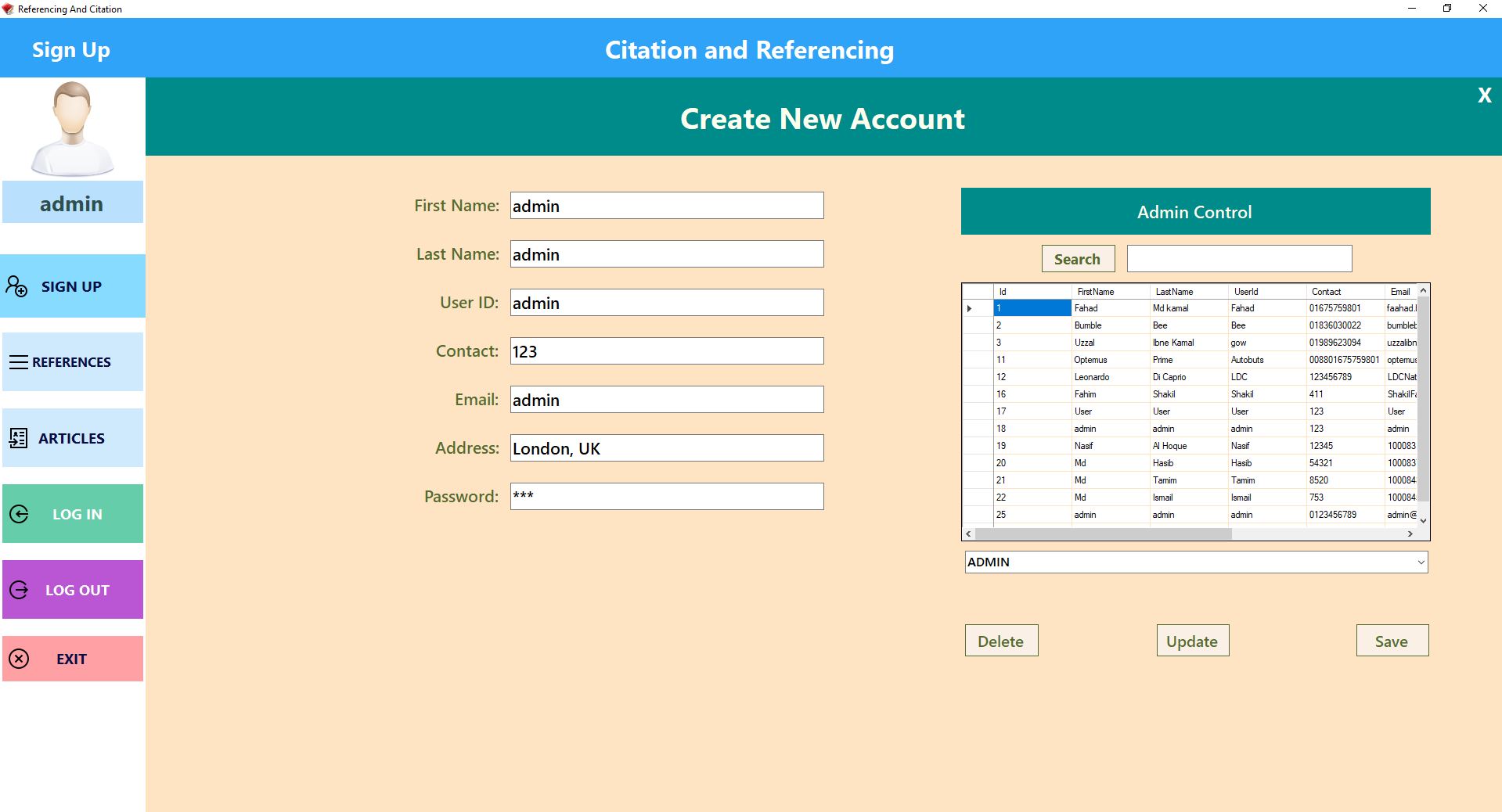
BlackBox-Testing-3

1. And the dashboard will show up. From where user can go to other pages.

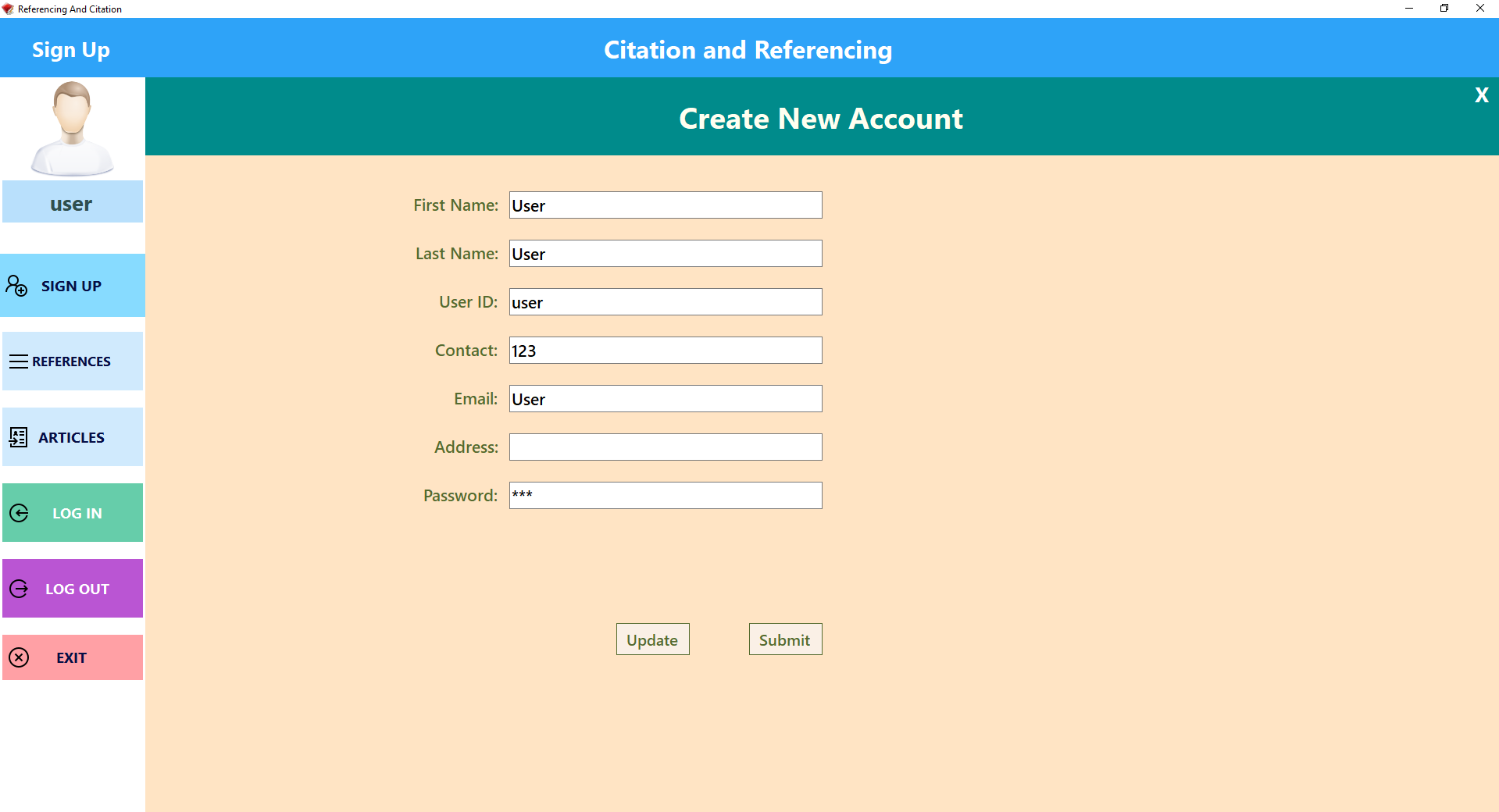


BlackBox-Testing-4

1. If the user is admin than he can see and use admin panel in sign up form otherwise user can update his or her data.

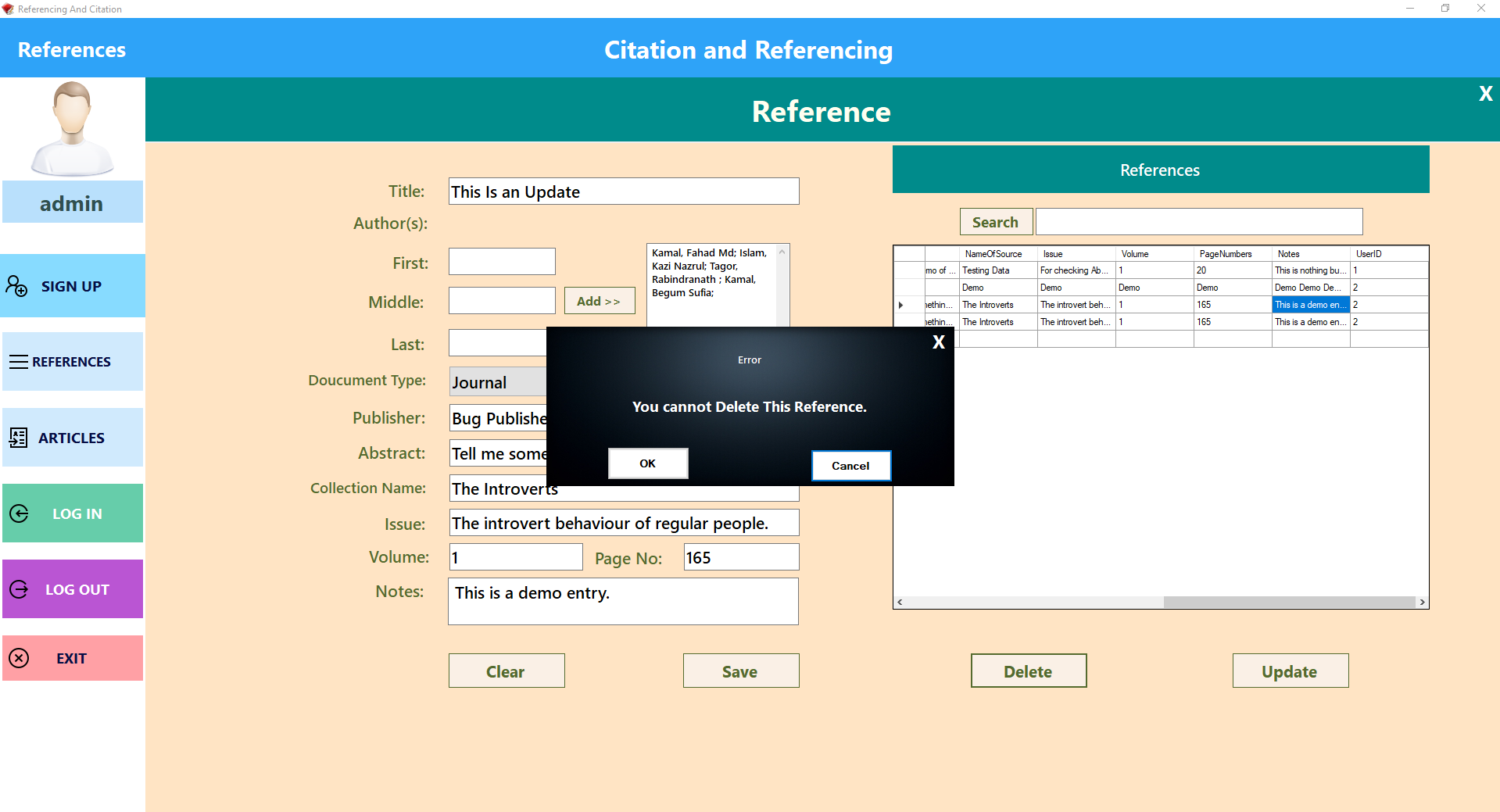


BlackBox-Testing-5(Admin-login-create-account-form-view)



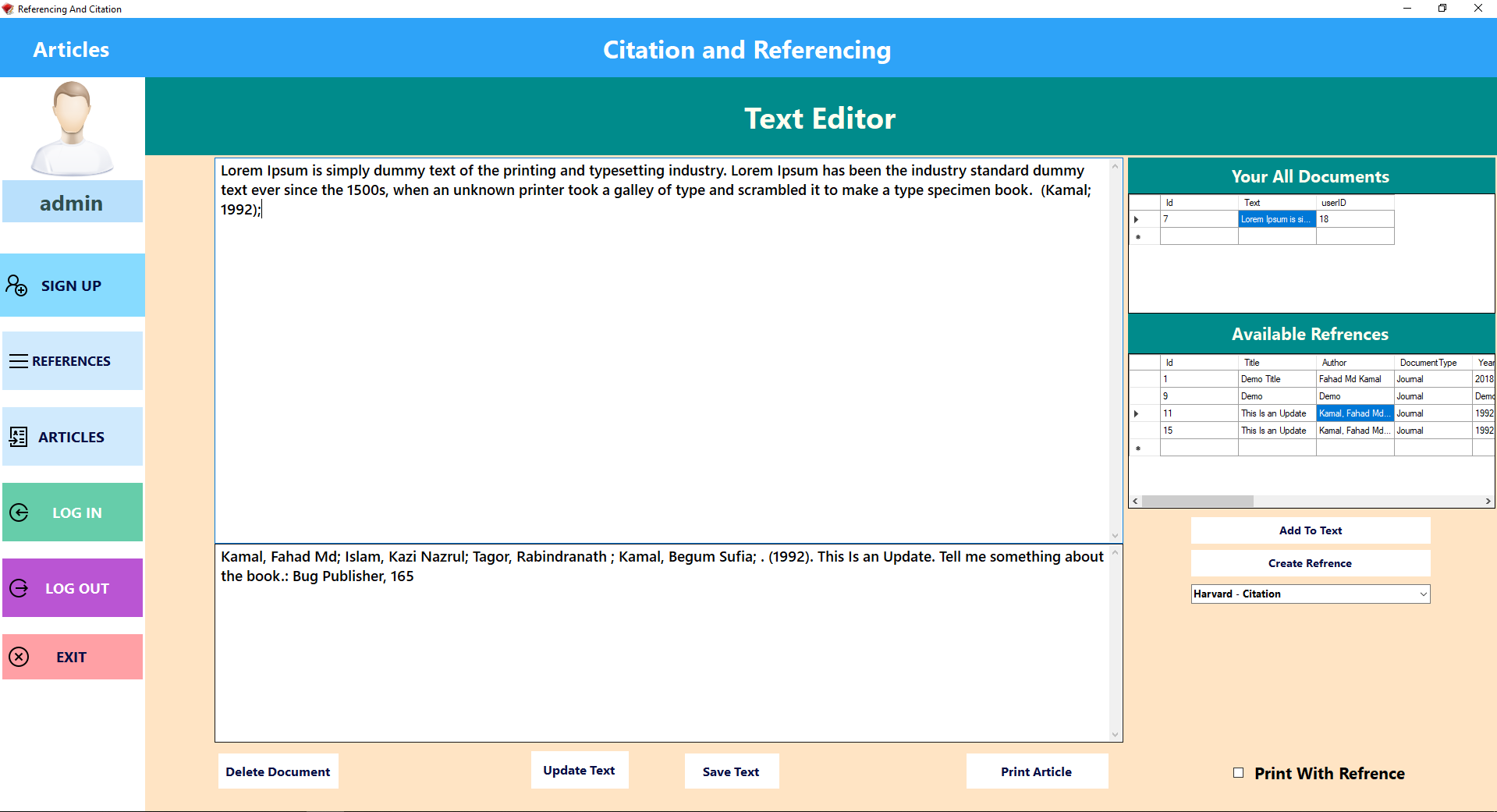
BlackBox-Testing-6(User-Login-Create-Account-form-view)

1. Any user can see other user’s references and create new references. But cannot edit or delete other user’s references except their own ones.



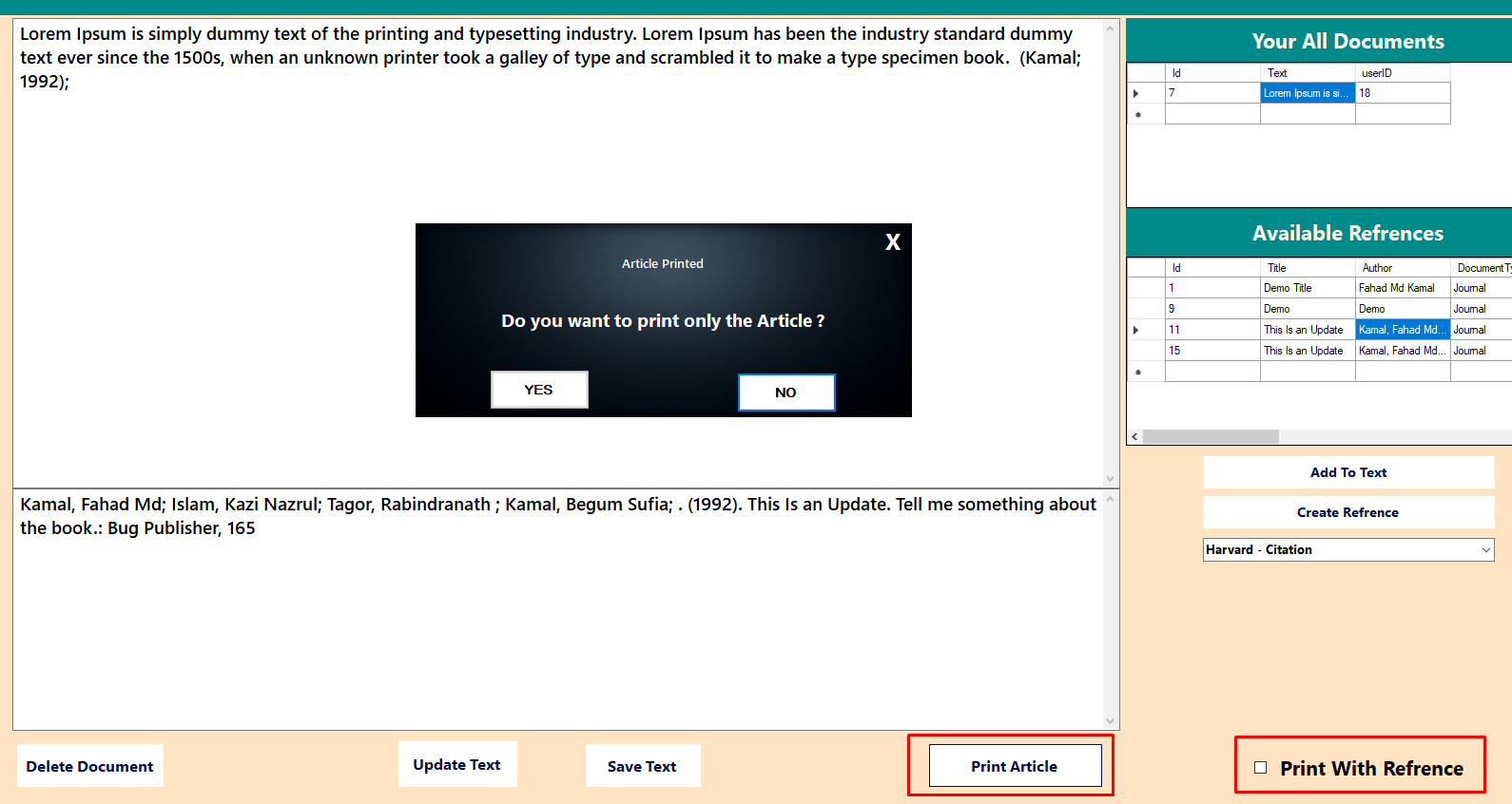
BlackBox-Testing-7

1. Users can see only their own documents of the saved documents that were stored.



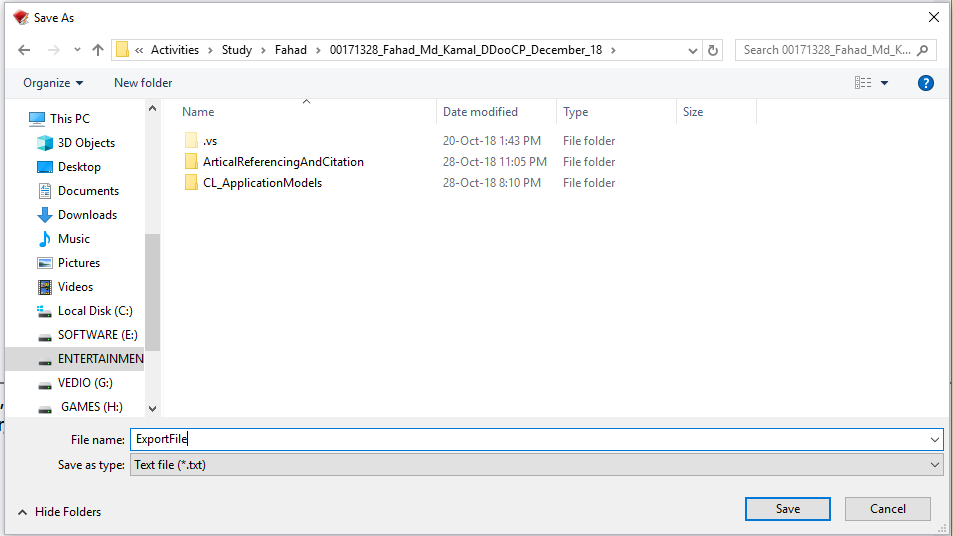
BlackBox-Testing-8

1. If they want to export the article in a text file, they can do it by pressing the print article button. Moreover, if they want to print with the references they can by checking the print with reference checkbox.

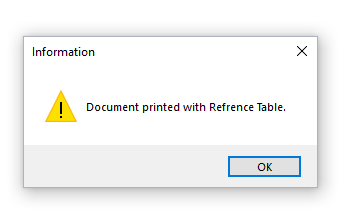


BlackBox-Testing-9

1. After pressing the yes button a dialogue box will show up. After choosing folder location, setting file name and pressing save button the document will be exported and a message box will show up of the success.



BlackBox-Testing-10



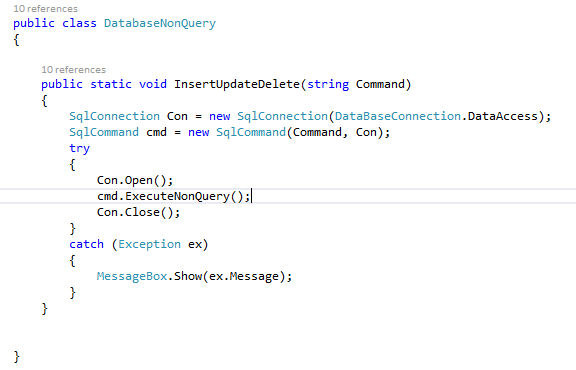
BlackBox-Testing-11

### White Box Testing Plan:

1. **Search for User data:** Checks for user information match with the database. If information matches form dashboard will show up.
2. **New user Create account:** If the user is new and presses the create account label signup form will show up where the user can create new account.
3. **User validation check:** If user id and password matcher a message box will show up and form dashboard will show up.
4. **User and Admin access in Sign up form:** Form dashboard’s signup form user can create update their data and if the user is an admin then they can update delete other user’s data.
5. **See all the references of the database:** Users can see the references of all users in references table but cannot update or delete their data except their own ones.
6. **Create new Reference:** Users can create new references by filling user data field on sign up form.
7. **Search input matching References:** By pressing the search button the user could search title matching references.
8. **Update or delete data:** Regular users can only update their own data from the database. But an admin can update delete any user’s data from the database.
9. **Only the logged in user’s documents show and manipulate able:** own recoded documents in the data grid view and edit or delete them.
10. **Adding reference tags to the text:** by selecting references from the data grid and selecting the reference style from the select reference style combo box.
11. **Exporting the document file:** They can export their documents in text file (.txt) format by pressing on their chosen location and name.

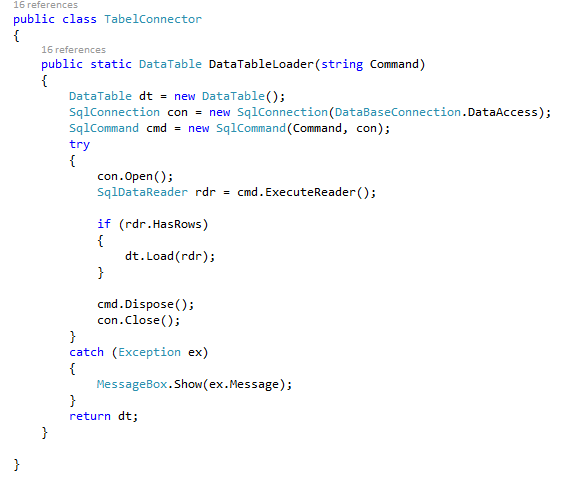
Encapsulation

* **All the non-Query SQL operations are done by “DatabaseNonQuery” class’s method through passing parameter:**

****

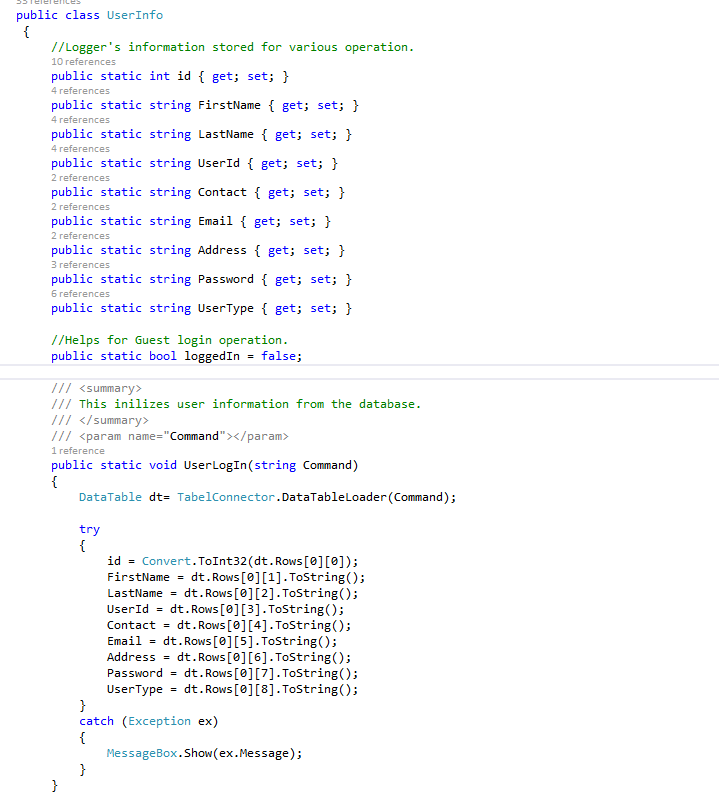
whiteBox-Testing-1

* **All the SQL Query operations are done by “TabelConnector” class’s method through passing parameter:**

****

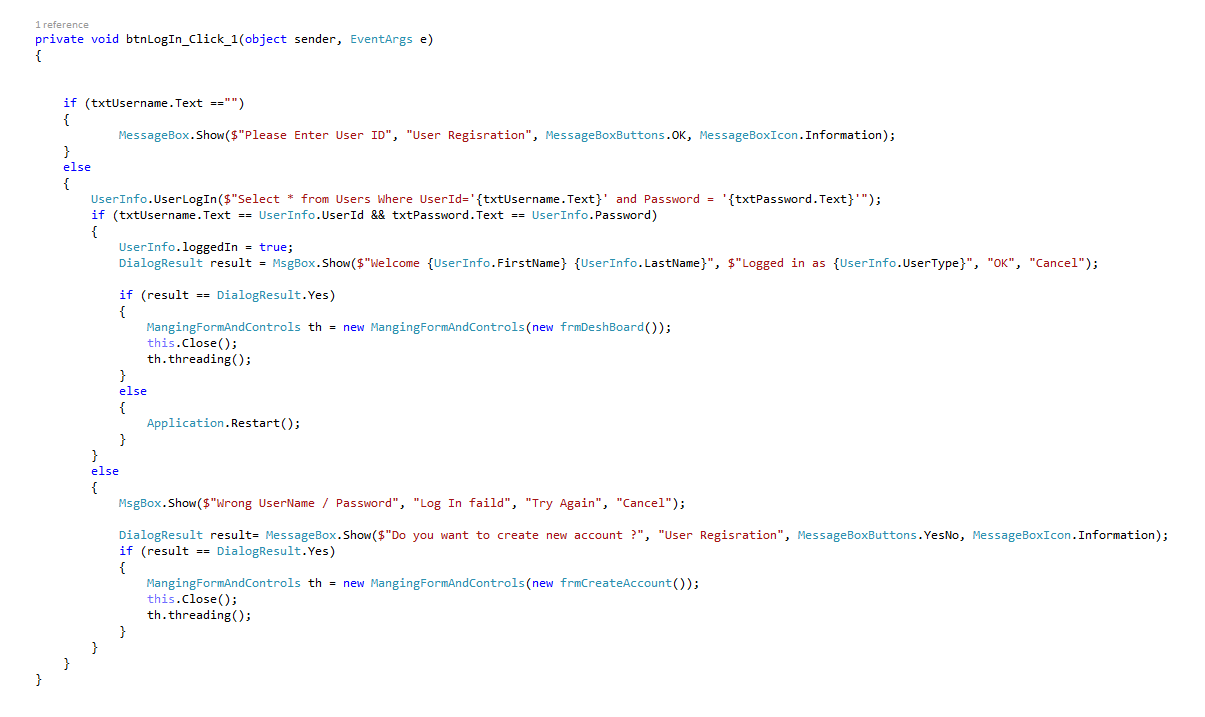
whiteBox-Testing-2

**Encapsulated user information**

****

**Test 1:**

**Search for User data:**



whiteBox-Testing-3

**Expected Output:** With the user Id the method will search for user data and retrieve them in UserInfo table. And will show up form dashboard by logging in the user.

**Actual Result:** Output was expected.

**Comment:** No error was found.

#### Test 2:

**New user Create account**

****

whiteBox-Testing-4

**Expected Output:** User input all the required information and pressed the submit button and the user should see a message box with message of data saved.

**Actual Result:** Output was not that much satisfactory. It cannot always prevent users to save data with the same name field.

**Comment:** This bug needs to be fixed.

#### Test 4:

**Admin access in Sign up form**

****

whiteBox-Testing-5

**Expected Output:** Admin can see the admin control panel where they can edit update and delete user data.

**Actual Result:** The result was satisfactory.

**Comment:** No error was found.

**User Update**

****

whiteBox-Testing-6

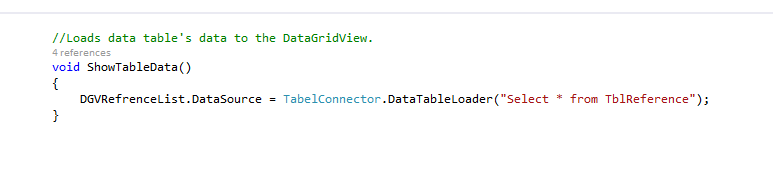
**Expected Output:** User can update his or her information but cannot delete their data.

**Actual Result:** The result was satisfactory.

**Comment:** No error was found.

#### Test 5:

**See all the references of the database**.



whiteBox-Testing-7

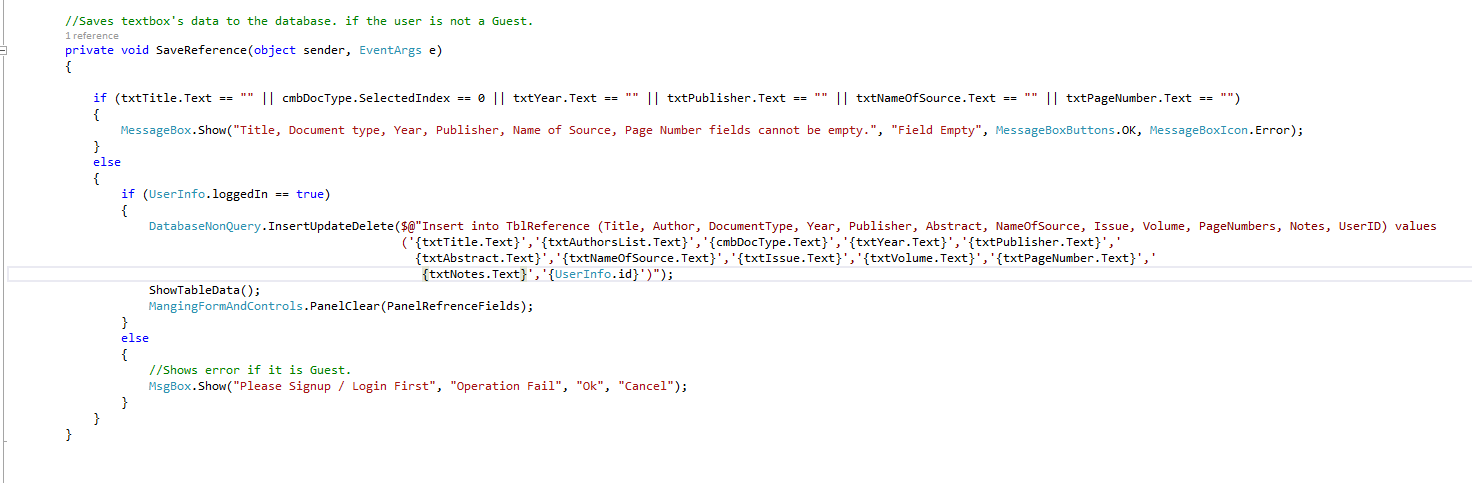
**Expected Output:** When the reference form loads all user’s references could be seen.

**Actual Result:** To result was as expected.

**Comment:** No error was found.

#### Test 6:

**Create new Reference**



whiteBox-Testing-8

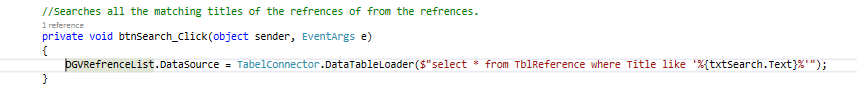
**Expected Output:** After filling all the must have data fields the user can save the references by pressing the save button.

**Actual Result:** The result was satisfactory.

**Comment:** No error was found.

#### Test 7:

**Search input matching References**



whiteBox-Testing-9

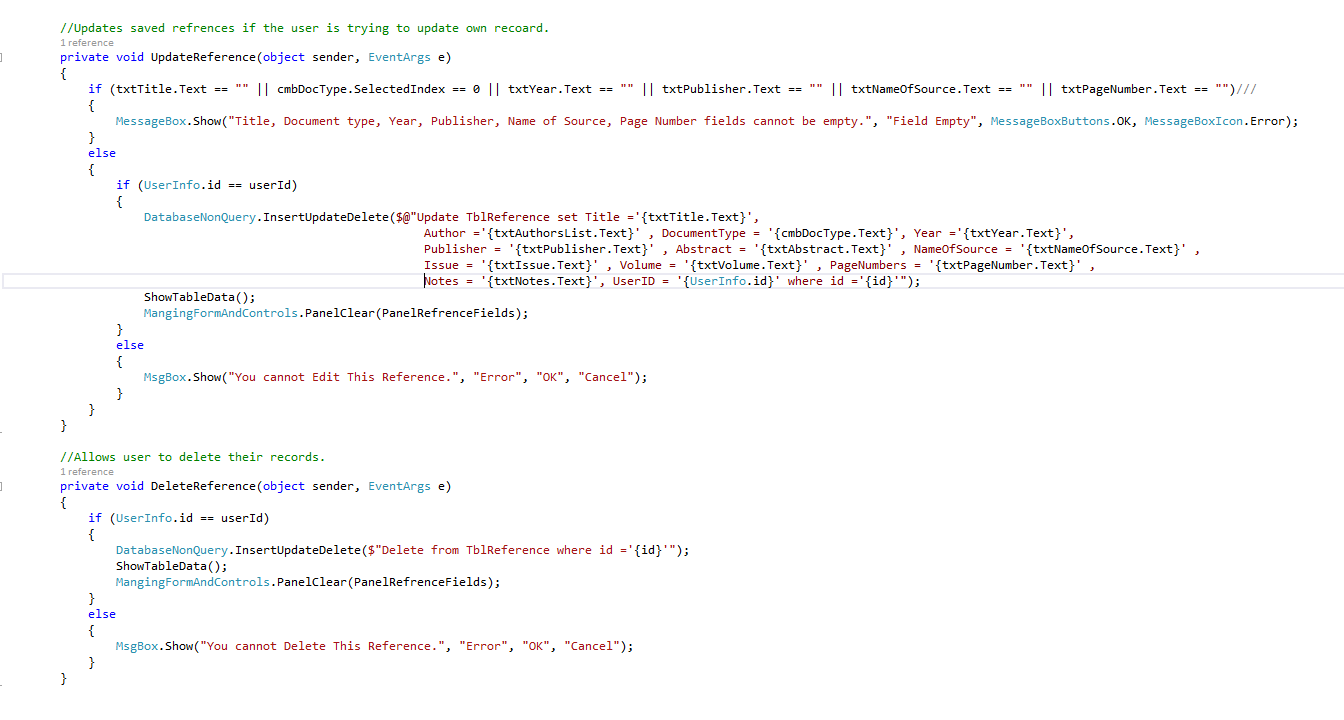
**Expected Output:** user should find all the title matching references like in the textbox.

**Actual Result:** The result is satisfactory.

**Comment:** No error was found.

#### Test 8:

**Update or delete references**



whiteBox-Testing-10

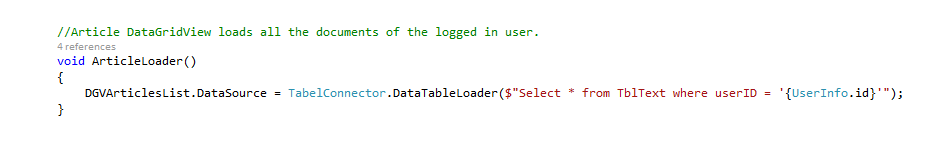
**Expected Output:** After fulfilling all required fields user can save their references and update or delete their own previous records.

**Actual Result:** The result was good.

**Comment:** No error was found.

#### Test 9:

**Only the logged in user’s documents are shown and manipulate able**



whiteBox-Testing-11

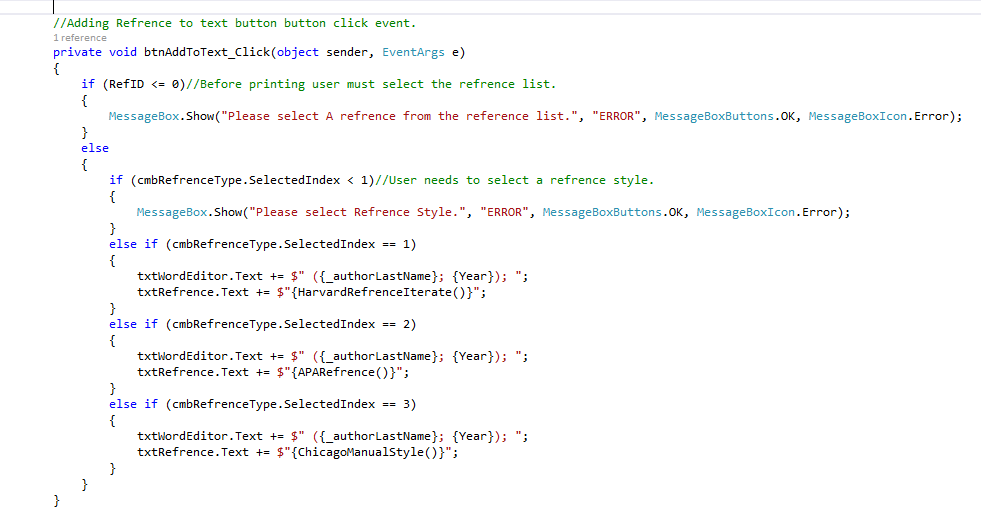
**Expected Output:** Logged in user can only see their documents and manipulate those.

**Actual Result:** Result was good.

**Comment:** No error found.

#### Test 10:

**Adding reference tags to the text**



whiteBox-Testing-12

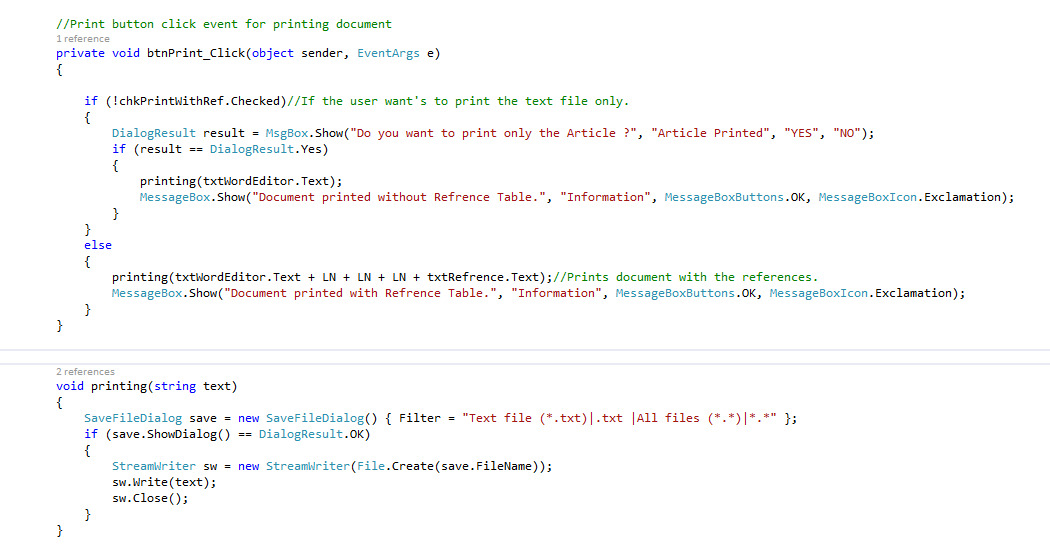
**Expected Output:** After selecting reference from reference list and selecting reference style from combo-box, user can add reference tag to their text file by pressing button add to text.

**Actual Result:** Result was extremely good.

**Comment:** No error was found.

#### Test 11:

**Exporting the document file**

****

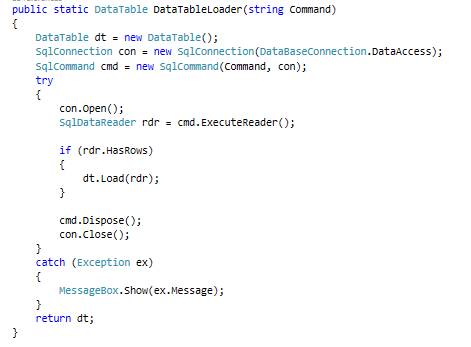
whiteBox-Testing-13

**Expected Output:** User can export the document file to an external text file with their desired name and desired location.

**Actual Result:** The result was good.

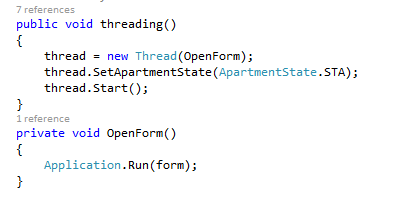
**Comment:** No error was found.

#### Exception Handler:



whiteBox-Testing-14

#### State Threading:

****

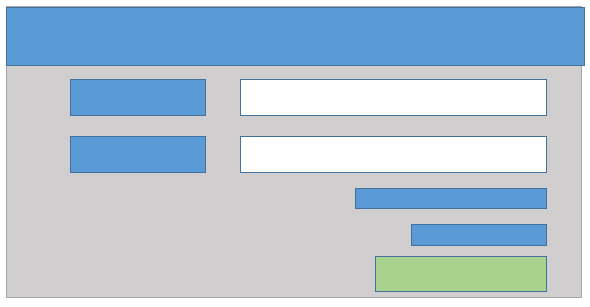
whiteBox-Testing-15

# Task-3:

UI design in Paper prototype of the developed application “Reference Tracker” is given below.

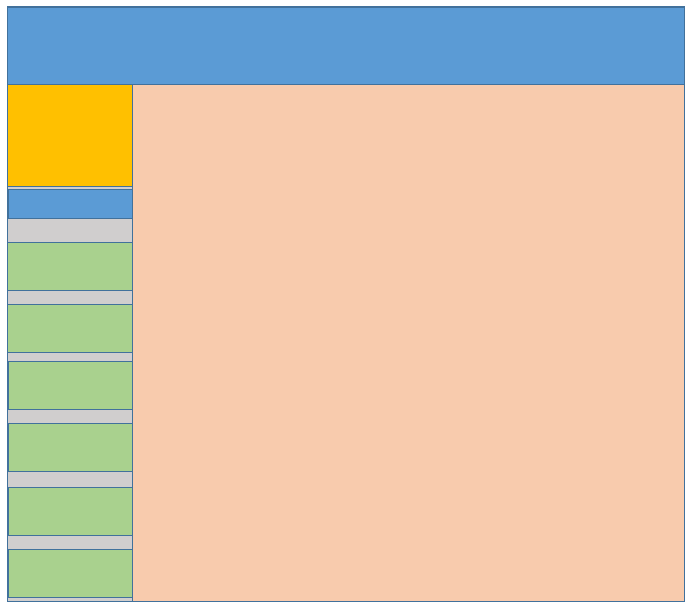
## UI-Prototypes:

### Login form (frmLogIn)



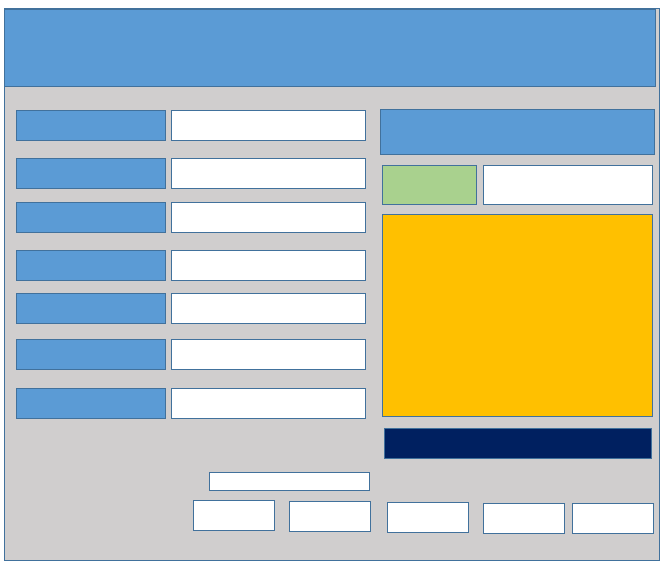
UI-Design-Prototype- 1

### Dashboard Form (frmDeshBoard):



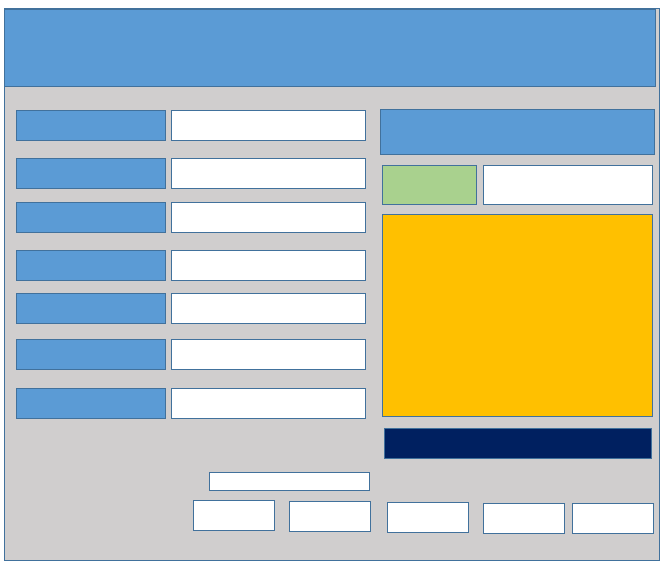
UI-Design-Prototype- 2

### Form Sign Up (frmCreateAccount):

****

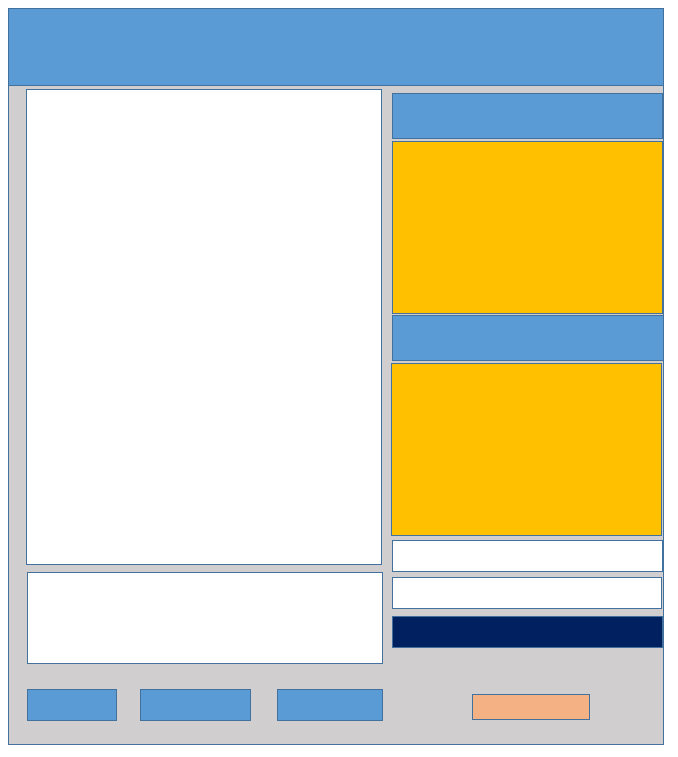
UI-Design-Prototype- 3

### Form Reference (frmReference):



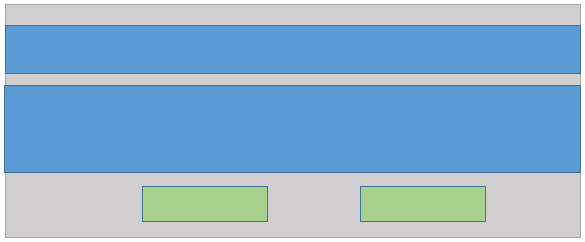
UI-Design-Prototype- 4

### Form Article (frmArticles):



UI-Design-Prototype- 5

### Custom Message Box(MsgBox):



UI-Design-Prototype- 6

## Description:

* **Form:** Gray background color is the form of the application.
* **Label:** Blue color boxes are the label of the application.
* **Textbox:** White Color boxes are the textboxes of the application.
* **Combo-box:** Dark Blue color text boxes are the combo-boxes of the application.
* **Button:** Green color boxes are the buttons of the application.
* **Picture Box:** Red color box is the picture box in the application.
* **Panel:** All the textboxes and buttons were arranged in a different panel.
* **Table Layout Panel:** All the panels were arranged in a table layout pane

## Overview of user experience:

The application “Reference tracker” was examined with a user where the result was satisfactory. As an admin there is a lot of extra functionality where as a regular user there are less. Besides, the security system for application on hiding data and restrictions for manipulation was extremely good. However, there is a little problem on creating new account. It seems that the application does not always responses with an existing user’s registration. Otherwise, all the functionality of the application is really easy and simple. This application could be made even better by adding user’s picture on dash board when they login. And by adding some extra features.

## Recommendations for Changes:

In this application normal SQL CRUD operation was implemented, which are extremely risky of getting hacked. Although, this application is complete, it could be improved even more by adding user picture and multiple types of document printing methods. Besides, if the user registration bug is fixed and adding various types of files saving option to this application will provide extra ordinary service to the users.

## Conclusion:

All the requirements of this assignment was done by following the assignment guide line. The application “Reference Tracker” meets all the required functionality of the assignment by using appropriate objects and using exception handler. The application can interact with database and does the threading for opening and closing forms. Encapsulation and abstraction was used in coding by using class, functions and class fields reuse throughout the whole application. All the recommendations and further changes of this project was described vividly.