**Vision and Scope Document**

**for**

**OCC Veterans Club Login Project**

**Version 1.0 approved**

**Prepared by Johnny Dao, Huan Do, Hung Phan, Gene Strickland, and Derek Tran**

**TBD Group**

**October 12, 2017**

**CS A220 - Software Engineering**

**Instructor: Sara Ghadami**

**Table of Contents**

**Table of Contents 2**

**Revision History 3**

**Project Plan 3**

**User Stories**

Functional User Stories **7**

Non-Function User Stories **8**

**Use-Cases**

Signing in UC-001 **8**

Formatting User’s Input UC-002 **9**

Viewing Database UC-003 **10**

Making Users Information Private UC-004 **11**

Making the Login Run Non-Stop UC-005 **12**

**Non-Functional Requirements 13**

**Use-Case Diagram 14**

**Sprint Backlog 15**

**Pre-game Planning 15**

**Staging/Grooming 16**

**Development Process 16**

**User Manual 17**

**References 27**

**Revision History**

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Date | Reason For Changes | Version |
| Gene Strickland | 9/24/2017 | Creating an outline of the application | 1.0 |
| Derek Tran | 9/24/2017 | Sign-in and sign-up functionality for club member access | 1.0 |
| Johnny Dao | 9/27/2017 | Functionality to view club member information in database | 1.0 |
| Hung Phan | 9/29/2017 | Sign-in functionality for administrator access | 1.0 |
| Derek Tran | 9/30/2017 | Added start-up menu for user to navigate between sign-in and sign-up menus | 1.0 |
| Derek Tran | 10/01/2017 | Made sure users had to enter student ID in the correct format | 1.0 |
| Huan Do | 10/03/2017 | Use Cases for Functional Requirements | 1.0 |
| Derek Tran | 10/03/2017 | Allowed users to go straight to sign-up menu from sign-in menu if correctly formatted student ID is not found in database | 1.0 |
| Gene Strickland | 10/04/2017 | Organized/simplify the code so that implementation of other functions could be easier | 1.0 |
| Derek Tran | 10/05/2017 | Allowed user to return to startup menu from sign-in and sign-up menu without restarting program | 1.0 |
| Derek Tran | 10/06/2017 | Ensured that user couldn’t sign-up with student ID that already exists in database | 1.0 |

**1. Business Requirements**

**1.1 Background**

The Orange Coast College Veterans Club (OCCVC) is currently using a traditional pen-and-paper sign-in sheet to keep a record of the attendance of both current and new club members. The stationery needed to continue this method of sign-ing-in as club activities commence will not only take up needed space in the club room, but also increase the cost of supplies for the OCCVC. In addition, valuable time will be spent combing through all the sign-in papers to verify the validity and attendance of club members.

**1.2 Business Opportunity**

As more campuses continue to adopt electronic forms to inquire students’ information, it is necessary for student clubs to adapt as well to help simplify the workflow of club activities. This login product will provide the OCCVC, as well as other clubs, a simpler and more elegant method of signing in and recording their members’ attendance and information. Keeping electronic records rather than paper records will reduce storage and costs, and improve accessibility and workflow efficiency. It is also a step towards modernizing the OCCVC through use of digital records as opposed to traditional pen and paper.

**1.3 Business Objectives and Success Criteria**

The aim of the product is to increase efficiency while minimizing use of resources by eliminating the need for paper records to record members. In turn, it will allow OCCVC members more time to commit to other, more important club activities. The project will be considered successful if the OCCVC experiences a reduction in the amount of paper used and in the costs going towards stationery. Those factors will indicate whether OCCVC members feel comfortable enough to transition to the new electronic system.

**1.4 Customer or Market Needs**

While multiple programs exist to ease keeping records of club members and/or their contribution to club activities, none exist that meet the specific needs of various clubs at OCC. The software will run on a personal computer running the Windows operating system with a mouse and keyboard for users to input their information. The product will have a graphical user interface to make it user friendly as well as aesthetically appealing in order to maintain the simplicity and ease the transition to our electronic form of logging in.

**1.5 Business Risk**

The risks associated with the development of this product include but are not limited to:

* A team member leaves the group during the development process. We will ensure all data is backed up where all team members can access it easily, and we will distribute the ex-member’s workload amongst the remaining members.
* Timing issues. All developers have additional responsibilities outside the development of this product, such as their jobs and their school work. We will take into account each member’s daily schedule and adjust the development process accordingly in order to meet the goals of this project while tending to our other responsibilities. In addition, we will contact each other if a change in schedule is needed.
* Loss of data. We will keep backups of all versions of the product on our personal computers, external drives, and on the cloud to ensure any corruption or loss of data is minimal.
* OCCVC acceptance of the product. To ensure that the product will be adopted by the OCCVC as their new login process, we will develop our front-end to be as simple as possible to avoid having club members and administrators return to their original method of paper sign-in sheets.

**2. Vision of the Solution**

**2.1 Vision Statement**

The product will allow the OCCVC members to spend less time verifying logins and more time on productive club activities. The product will also be broad in its design to allow other potential clubs to adopt the software. With this application, club administrators will be able to keep and search records of the OCCVC members more accurately and swiftly. Furthermore, the members’ information will be kept safe since only administrators with a secure password will be able to view the members’ information when needed; ending the need to keep club members’ records insecurely on paper or on excel.

**2.2 Major Features**

* Member login: Permits current club members to sign in with just their OCC Student ID
* Member sign-up: Allows new members to sign-up for the club
* Input validation/verification: Ensures proper input in certain fields and notifies users of any errors (i.e. student ID must be formatted correctly)
* Member database: tracks the current and past members and their information
* Member login record: records the date, time, and information when a member logs in
* “Tamper” proof: prevents member misuse of product (i.e. closing or minimizing the application)
* Administrator controls: provide separate login credentials for facility administrator with special functions
* View member database: administrator control that will display a list of members in a readable format from the user database
* View member login record: administrator control that will display a list of login records in a readable format from the user database
* Print controls: administrator control that will print the user database or login record
* Graphical user interface: simple and elegant design
* User-friendly: clearly labeled fields and simple instructions

**2.3 Assumptions and Dependencies**

This project assumes that the OCCVC, or any other clubs on campus, have or can acquire a personal computer running on a Windows operating system to run the software.

**3. Scope and Limitations**

**3.1 Scope of Initial Release**

In the initial release, our product will utilize a database to store club member information and will focus on using a console-based interface for administrators to access that data. This will ensure that the logic behind the data access is working correctly. There will be a sign-in and a sign-up page for club members to check-in to the OCCVC, and a separate sign-in page for the administrator in order to access the administrator controls. The administrator controls include allowing the administrator to view the records of the club members. In addition, we will include a user login verification feature to ensure that our product can handle input verification.

**3.2 Scope of Subsequent Releases**

In our subsequent releases, we will convert our user interface from a console-based interface to a graphical user interface. Our login verification feature will be expanded to include further administrator privileges, such as viewing and printing login records with timestamps for the times a club member has signed-in. Lastly, to make our application “tamper-proof”, we will take away the user’s ability to close or minimize the login window to prevent members from abusing the login feature.

**3.3 Limitations and Exclusions**

The product won’t have the ability to change the database nor will the stakeholder be able to customize the user interface to their tastes.

**4. Business Context**

**4.1 Stakeholder Profiles**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ***Stakeholder*** | ***Major Value*** | ***Attitudes*** | ***Major Interest*** | ***Constraints*** |
| *OCCVC club members* | *streamlined sign-in and sign-up* | *see product as faster way to sign-in to club* | *simpler and faster sign-in process* | *must have valid OCC Student ID* |
| *administrators* | *fewer errors in work* | *reduces monetary and time costs for the club* | *easier to look up and verify members and attendance* | *must run on Windows OS* |

**4.2 Project Priorities**

|  |  |  |  |
| --- | --- | --- | --- |
| **Dimension** | ***Driver***  ***(state objective)*** | ***Constraint***  ***(state limits)*** | ***Degree of Freedom***  ***(state allowable range)*** |
| *Schedule* | *release 1.0 to be available by 10/12, release 1.1 by 12/1* |  |  |
| *Feature* | *Login credentials, member database, and command-line UI in release 1.0* |  |  |
| *Quality* |  |  | *70%-80% of user acceptance tests must pass for release 1.0, 95%-98% for release 1.1* |
| *Staff* |  | *maximum team size is 5 developers* |  |
| *Cost* |  |  |  |

**4.3 Operating Environment**

The users of this product will be focused to the OCCVC and other potential clubs on the OCC campus. Users will first use this system when they enter the club in order to login their attendance. The system will continue to run throughout the day until the administrator closes the program. Continuous access to our system is not critical for the operation of our business. Data will be generated and used when a user log into our system. That data can be accessed by the administrator, who has the credentials to view the member database. Security controls include our user login verification feature to prevent user fraud*.* In addition, users won’t be able to close or modify the window to tamper with the user login.

**User Stories**

1. As user I want to be able to sign in electronically with my ID number so that I don’t have to write all of my information down when I sign in.
2. As an administrator I want input validation so that members input information in the correct format.
3. As an administrator, I want to be able to view the user database so that I can verify members’ attendance at OCCVC.
4. As an administrator, I want the application to only allow the members to signup/sign-in and that only I, the administrator, can view the members’ database so that the members’ information will be kept private.
5. As a user I want the program to be able to run non-stop without constant maintenance during club hours so that it is always easily accessible.

**Non-Functional User Stories**

1. As an administrator, I want my application to be secured so that the club members won’t be able to view the club’s database.
2. As a user, I want my information to be kept safe and secure so that no one besides the administrator can view my information.
3. As the owner, I want this application to be easily modifiable so that it can suit the need of other clubs who requested this application.
4. As the administrator, I want my application to be user-friendly so that club members won’t have to spend too much time figuring out the application and use their time for more club activities.
5. As the administrator, I want this login application to be reusable so that club members won’t have to sign-up every club meeting but instead they can just sign-in since the previous time the application was used, it had created a text file of all club members that signed up.

**Use Cases**

***Functional Requirements***

# Use Case: Signing in

**Id**: UC- 001

**Description**

Users are able to login into the application with their student ID. Their information will be stored into the application’s database. If the users’ information is not in the database, the user can add their information into the database.

**Primary Actor**

* Users
* Club Members

**Pre-Conditions**

The application must be running prompting the user to enter their student ID.

**Post Conditions**

Success end condition

To indicate that the sign-in process was successful, the application will display a welcome message once the user entered their student ID.

Failure end condition:

If the sign-in process fails, the user will be asked to repeat the ID input or will be asked to sign-up as a new club member. This process will repeat until the user either enter the correct input, successfully signup as a new club member, or wait for the application to reset back to the sign-in menu.

## Main Success Scenario

1. Once the application is launched, the sign-in menu will be displayed
2. The users will be asked to enter their student ID
3. A welcome message will be displayed
4. The application will revert back to the sign-in page prompting the next users to enter their student ID

## Extensions

2. In step 2 of the main success scenario, if the user’s student ID is not in the database

2a. The user will be asked to sign-up as a new club member

2a.a. The user will be asked to enter their student Id

2a.b. The user will then be asked to enter their first name

2a.c. The user will be asked to enter their last name

2a.d. The user will be asked to select among the branches provided

# Use Case: Formatting User’s Input

**Id**: UC- 002

**Description**

The user must input their information with the format that was asked. If the format of the user’s input is incorrect, the user will be asked to re-enter their student ID in the correct format. This process will repeat until either the format is correct or until the application refresh itself. If the format of the user’s input is correct, the user will be able to proceed onto the next step.

**Primary Actor**

* Users
* Club Members

**Pre-Conditions**

The user will have to input their student ID in the incorrect format in order for the “invalid input” message to be displayed.

**Post Conditions**

Success end condition

If the user were to happen to enter their information in the incorrect format, the “Invalid input format” message will be displayed and the user will be able to re-enter their information in the correct format.

Failure end condition:

If the user were to happen to enter their information in the incorrect format, if user was able to continue without the application displaying the “Invalid input format” message, then the function as failed.

## Main Success Scenario

1. The application is launched
2. The user will be asked to enter their student ID
3. If the user’s input is invalid, the invalid format message will be displayed
4. User will be able to re-enter their student ID

## Extensions

2. In step 2, if the user enters their student ID in the correct format

2a. The user will be able to proceed onto the next step

4. In step 4, if the user happens to still enter the incorrect formatting

4a. The user will be repeatedly asked to enter their student ID in the correct formatting

4b. The application will refresh back to the sign-in page

# Use Case: Viewing Database

**Id**: UC- 003

**Description**

The administrator will be able to view the club members’/users’ information in the database. The members’/users’ information will be saved onto a text file.

**Primary Actor**

* The Administrator

**Pre-Conditions**

The administrator can only view the club database only if the text file is not empty.

**Post Conditions**

Success end condition

If the text file contains information on the club members, the administrator will be able to view the club members’ information.

Failure end condition:

If the text file is empty, the administrator won’t be able to view the club database.

## Main Success Scenario

1. The application is launched
2. The administrator will have to enter their password
3. The administrator can view the club’s database

## Extensions

No extensions

# Use Case: Making Information Private

**Id**: UC- 004

**Description**

The club members are not given an option to view the club database. The only options the club members are provided are to sign-up or to sign in. The only person with access to the club database will be the administrator; thus, the clubs’ members’ information is being kept private.

**Primary Actor**

* Users
* Club Members

**Pre-Conditions**

If the users were to enter new data, the information they enter will be kept private since the only way for anyone to view the club database is through the administrator.

**Post Conditions**

Success end condition

It will be successful only if the other club members are unable to view the club database. The club member will be able to login then the application will direct them back to the sign in page; thus, preventing the club member from viewing the database.

Failure end condition:

If the club members were able to view the club database, then the function failed.

## Main Success Scenario

1. The application is launched
2. The user enters their student ID
3. The user receives a welcome message
4. The application refreshes back to the sign-in page asking the next user to input their student ID

## Extensions

2. In step 2, the user is a new club member

2a. The user will enter their student ID

2b. The user will enter their information

# Use Case: Make the Login Run Non-Stop

**Id**: UC- 005

**Description**

The application should be able to run non-stop during club meetings.

**Primary Actor**

* The application
* Administrator

**Pre-Conditions**

The application is launched with no errors.

**Post Conditions**

Success end condition

The application will run continuously through the club meetings without any problems/conflicts.

Failure end condition:

The application will crash during the club meeting. Club members will not be able to use the application do to the error.

## Main Success Scenario

1. The application is launched successfully
2. The users’/club members’ enter their student ID and information
3. The welcome message is displayed for each successful student ID Entry
4. The administrator closes the application without any issues

## Extensions

No Extensions

***Non-Functional Requirements***

**Security:**

The administrator will be given a passcode that enable the them access additional features such as viewing the clubs’ members’ database, printing the database, etc. Since only the administrator will be given the passcode, only admin will be able to see the club members’ information. Once the member input their student ID, the application will display a welcome message then automatically return back to the application’s menu; thus, not giving the club members a chance to do anything else with the application.

**Privacy:**

Club members are not allowed to access other club members’ information. Only the administrator/s is/are allowed access to the club members’ information.

**Extensibility:**

The application can be easily modified to suit the need of other clubs. We, the application developers, are able to add additional features or remove features depending on what the client wants.

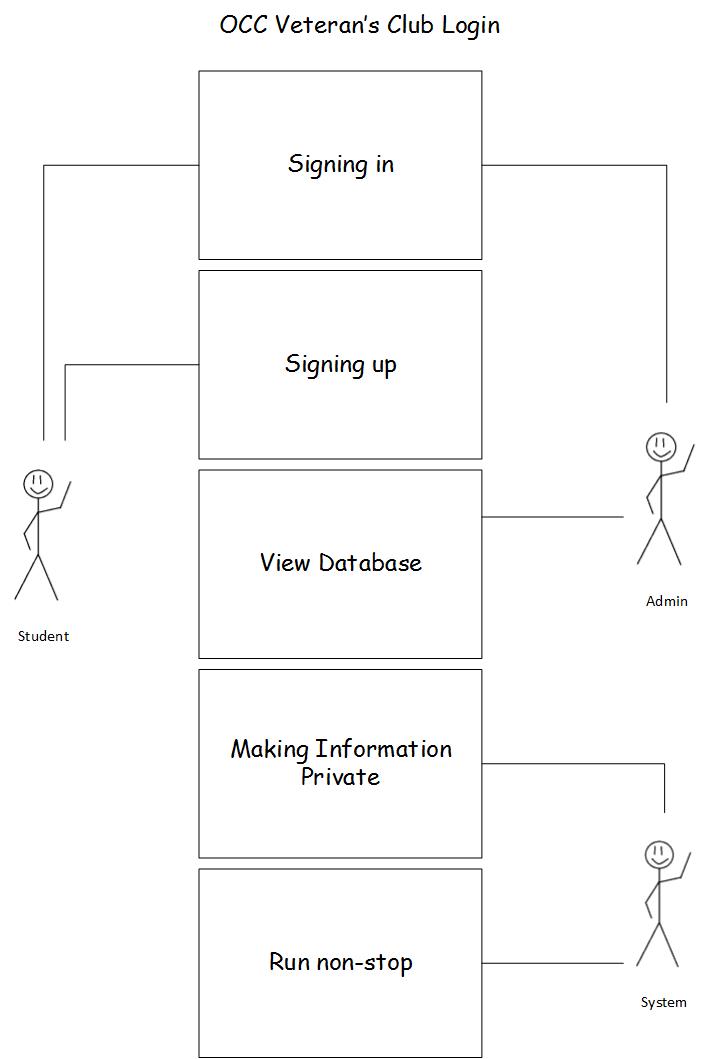
**Usability:**

The login will be user-friendly. The application when in use will display one line asking the club members for their student ID. Once the member input their student ID, if the application’s database contains their information, the login will display a welcome message then automatically return to the application’s menu. Else, the application will ask the member to input their first name, last name, and their branch of service (for the OCC Veteran Club. Varies depending on the club).

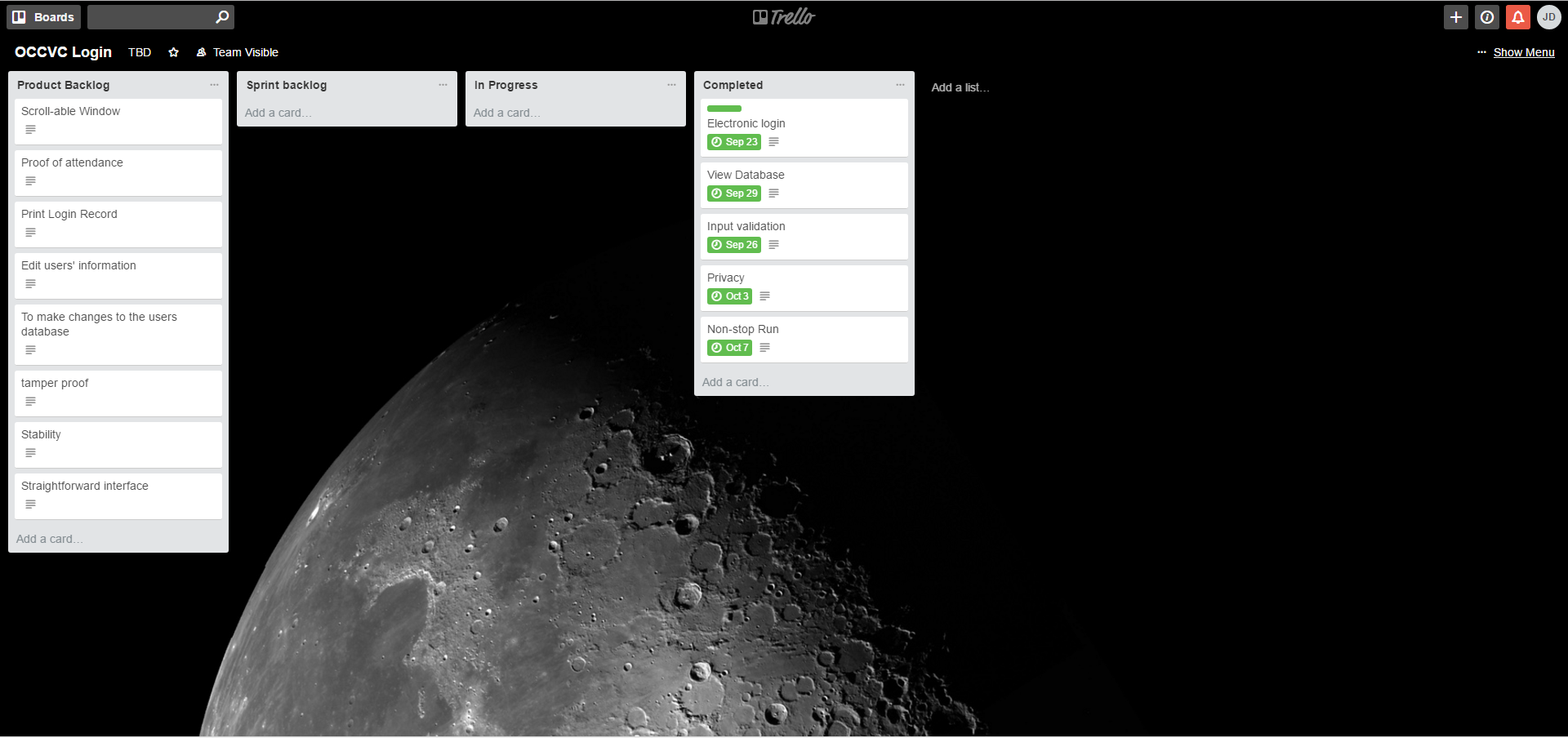
**Reusability:**

The club administrator is able to reuse this application each club meeting. By having the administrator reuse the application - that already contain the database - the club members that already sign up for the club will only need to sign in with their student ID.

**Use-case Diagram**



**Sprint Backlog Screenshot**



**Pre-game Planning**

We will first work on creating an electronic login page that will allow the user sign-in into the OCCVC. The information of each user that sign-in would be added to a database. We all believe that feature would be completed within three days. We then agreed to spend another three days to add input verification into our login page to ensure that the user inputs the correct credentials into the sign-in. Afterwards, we will include a feature that allows an administrator to view information on the user database through a console. Hung believes that it will take one day to implement due to his familiarity with developing similar features, while Derek believes it will take four days to implement due to his unfamiliarity with file streams. The group agreed to spend three days on the feature to both learn and implement this feature. At that point, the core functionality of our electronic page would be finished, so we will add features to our application to ensure security, efficiency, and accessibility. First we will work on making the information stored in the user database only viewable and editable to the administrator, in other words making that information private to those who sign-in. Hung believes that feature is simple enough to implement in a day while Huan believes a week is needed to perform security tests and manage memory. The group agreed to spend four days on this task. Afterwards we will make sure that our application can run non-stop without constant maintenance during certain hours so that it will always be easily accessible. Again, we all believe it will take about four days to implement this feature.

**Staging or Grooming**

For the first iteration, we decided to start with a command-line interface to get the main idea and function before we move on to the GUI application. This project has some basic requirements such as a standard menu with a sign-in and sign-out option, a format validation to correctly format the users’ inputs, an admin login function that allows admin to view and close the application when needed, and to make the application run non-stop during the clubs meeting so that the application run only once.

Before we could add any additional functions in this project, we need to make the basic outline of the application and it is to have a menu that allows users to enter data have it responds to the various actions we are implying. In the first iteration, we are just using standard cin to retrieve the users’ information. Once that has been completed, we then need a function that would correctly format the members’ input so that we can organize the members’ information so that we can easily retrieve information when needed. The next step in this project is have the application store the users’ input into the database; for now, we are having the application store the users’ data into a text file. The final step in the first iteration is to have the application run non-stop so that the administrator only has to log into the application once per club meeting.

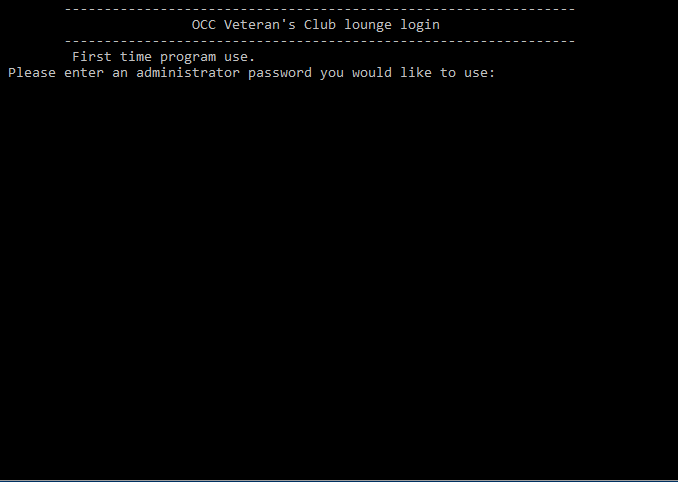
**Development Process**

As a group, we made a login application for the OCC Veterans Club; however, any club on campus is able to use this login for their club as well (with minor modifications). We first started this project by playing the poker game to see what user stories will benefit our application. The user stories we decided on were to have a sign-up and sign-in menu, to have a function that would check for correctly formatted user input, to have an admin function that would give admin control over the application, to make the members’ information private, and to make the application run non-stop. Once we decided on what functions to include in the first iteration for this project, we start the coding process. Gene first started by making an outline of the login application. As soon as he finished with the outline, Derek continue the application by making the login and signup function so that the application would be able to retrieve user inputs. Once the login menu has been completed, worked how we can save the users’/members’ input into a text file. After we ensured that we could write and retrieve data from the text file, we began working on a way for an administrator to view the contents of the file and to close the application. At the end, we disabled the ‘close’ button on the console to prevent accident premature termination of the program to allow it to run continuously. We also ensured that the console would clear itself between entries to provide a clean and easy to read interface that could run an extended duration (7+ hours a day). As we completed features they were tested, both isolated and in conjunction with other features to make sure everything worked together seamlessly.

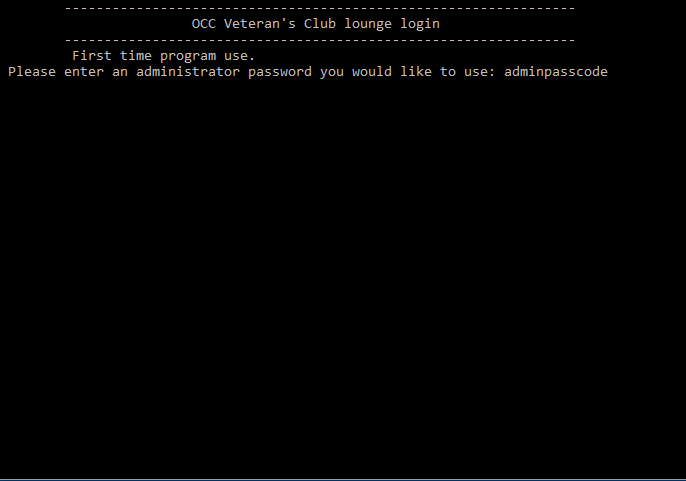
**User Manual**

**Administrator Setup**

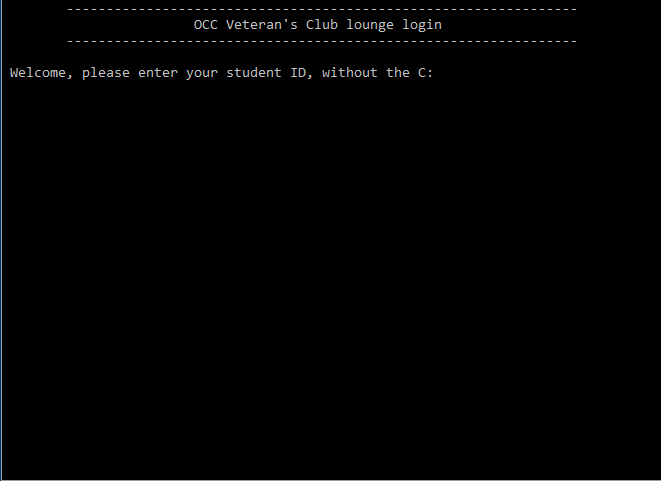
1. Application’s first launched.



1. When the application is first launched, the administrator will have to set up their password. This password will be used to close and view the club members’ database.

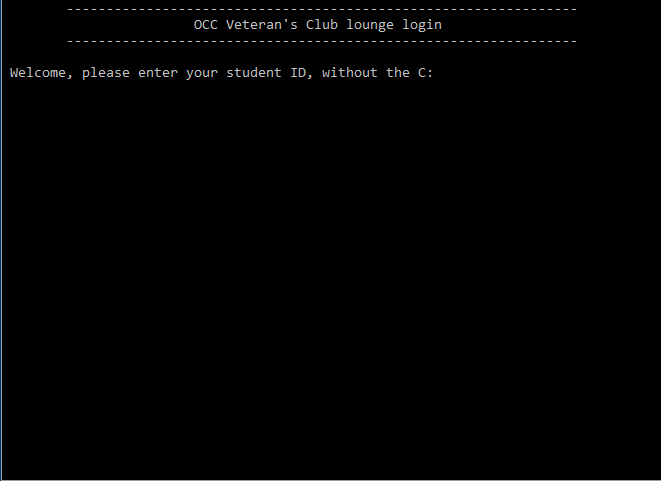


1. Once finished, the application’s login page will be displayed.

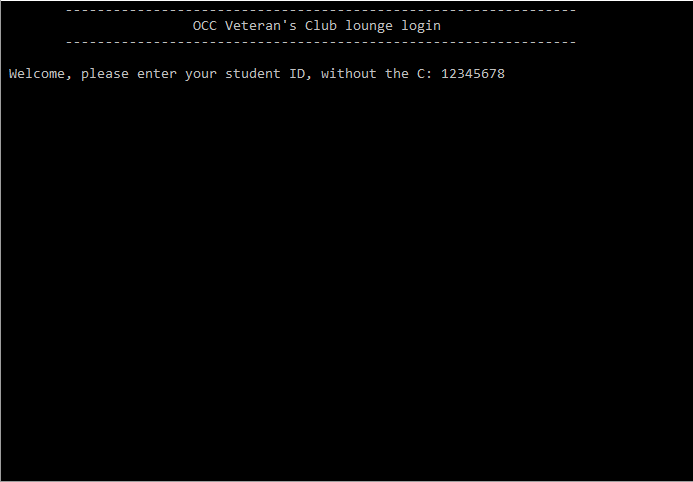


**Using the application for club members**

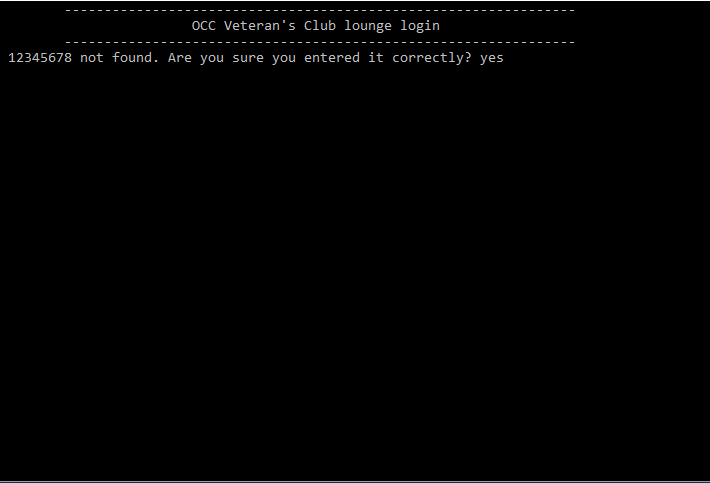
1. The application starts off with the application’s login page.



1. The user must enter their student ID number without the C. The student ID must be 8 digits long.



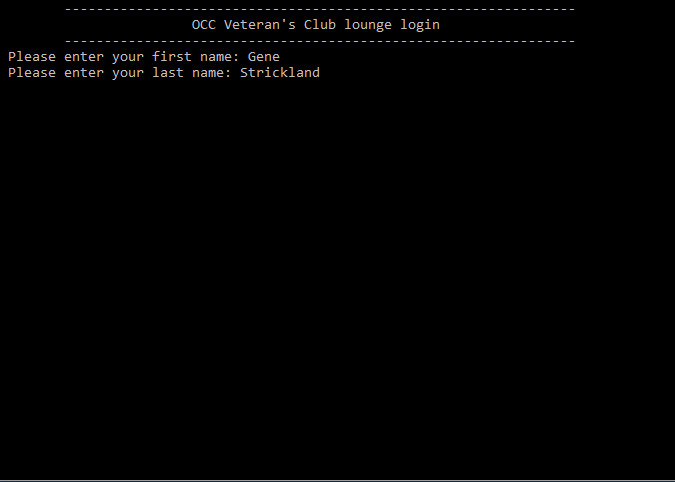
1. If the user is a new club member, the user must sign-up. The application will ask the user whether they inputted the correct student ID.



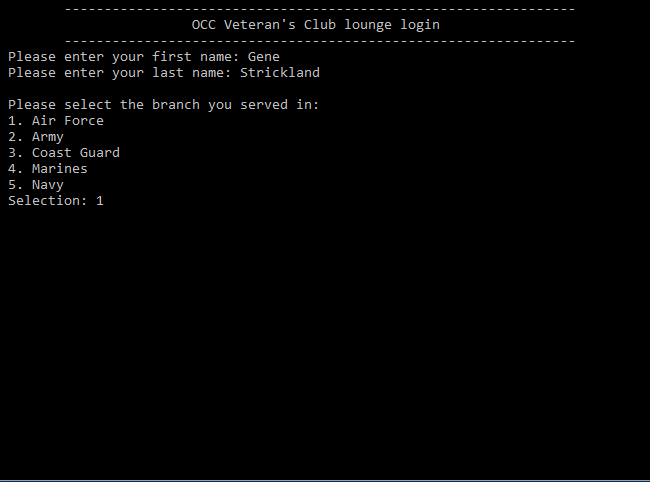
1. When the user inputted the correct student ID, they will then be able to proceed onto entering their information. First, the application will ask for their first name.



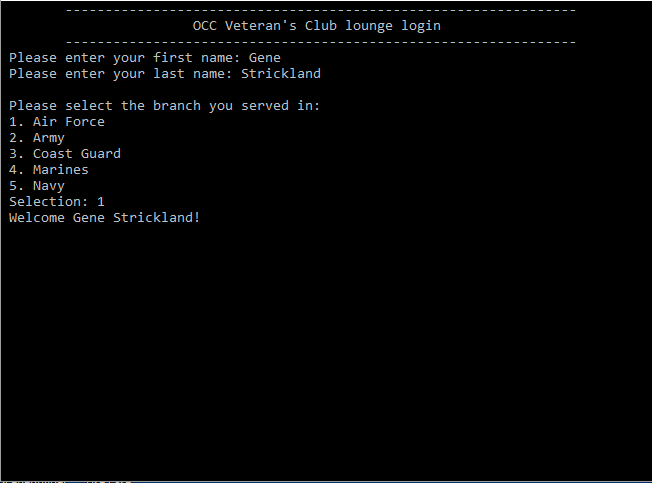
1. Then, enter last name.



1. Last, select the branch of service.

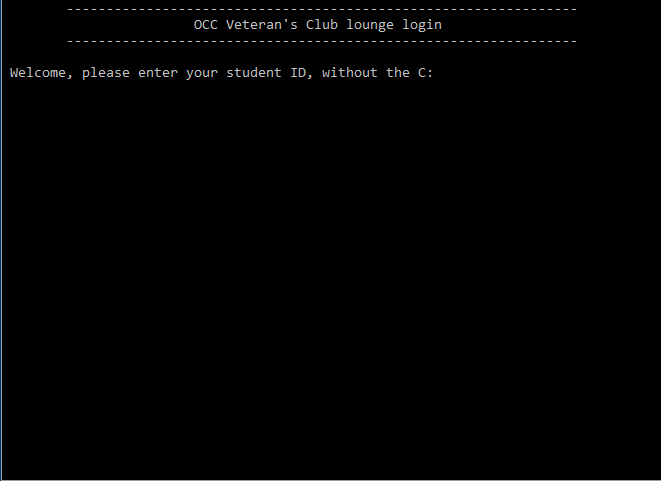


1. If the process was successful, the application will display a welcome message to the user.

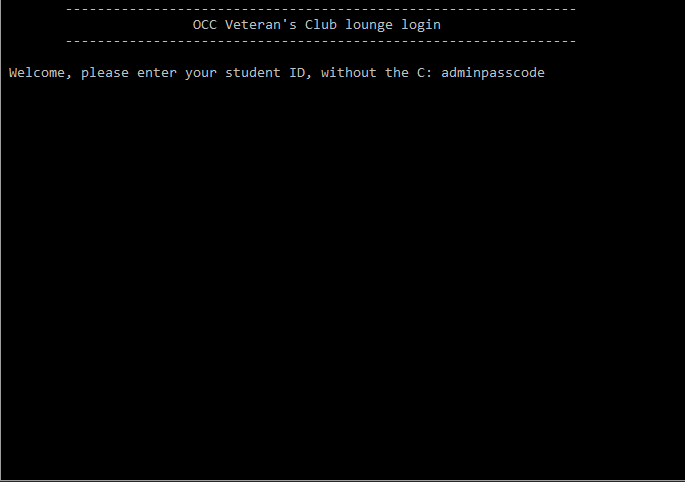


**View the Club Database**

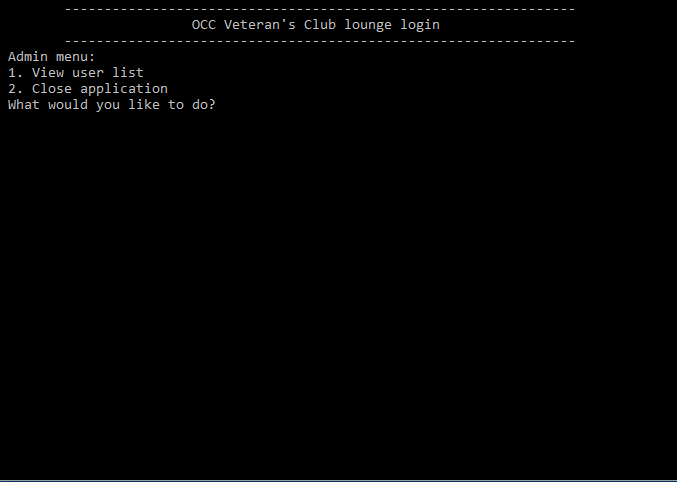
1. Launch the application.



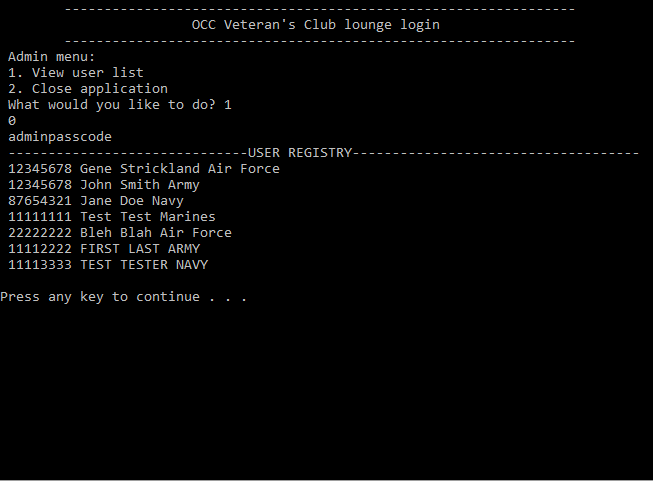
1. Enter the admin password that the administrator created at the beginning when the application was first launched.



1. The admin menu will be displayed once the admin password had been entered.

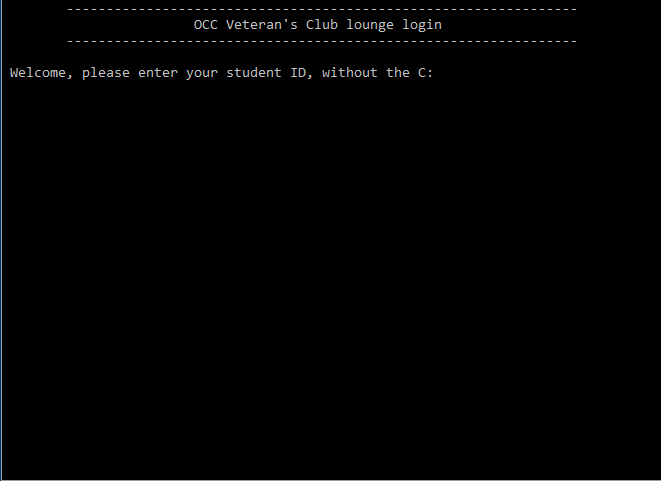


1. Enter 1 to view the club database.

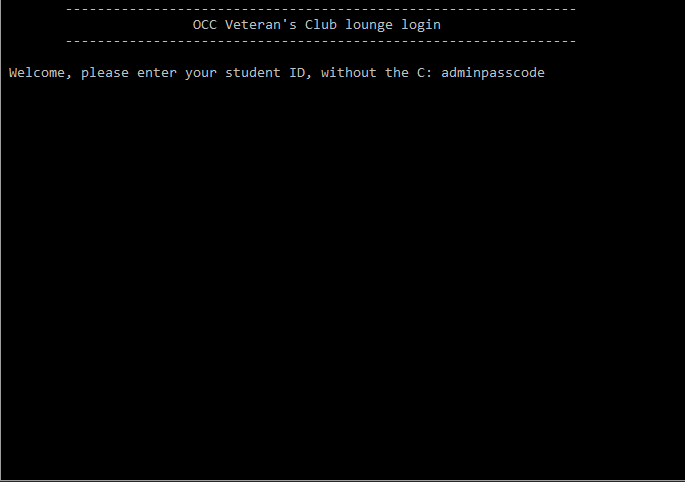


**Closing the Application**

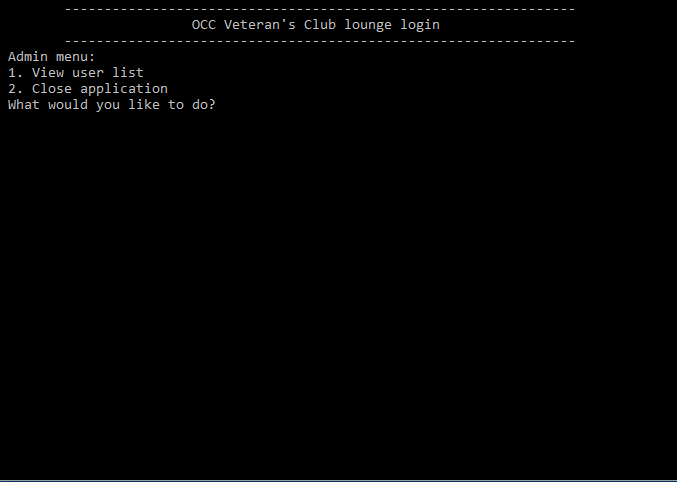
1. Launch the application



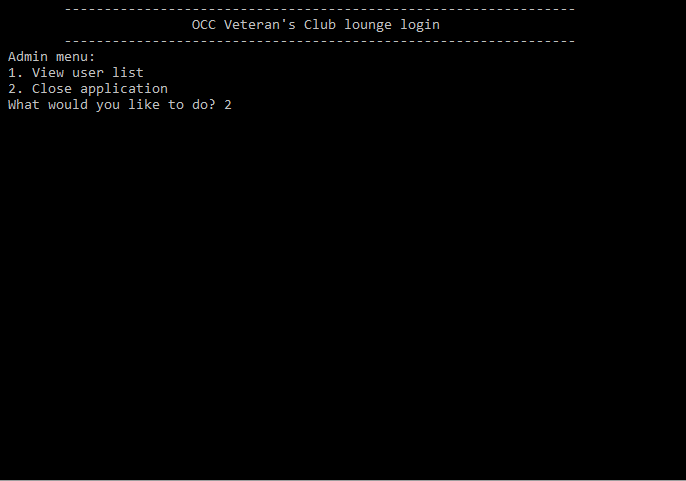
1. Enter the admin password that was created at the beginning when the application was first launched.



1. The admin menu will be displayed once the admin password had been entered.



1. Enter 2 to safely close the application.



**References**