

**ESKİŞEHİR TECHNICAL UNIVERSITY**

**FACULTY OF ENGINEERING**

**COMPUTER ENGINEERING**

**2021-2022 SPRING SEMESTER**

**BIM312**

**DATABASE MANAGEMENT SYSTEMS**

**TERM PROJECT**

PHASE-3

**G9-PNo9:** University Club Membership Database System

**Çağla ŞEN**

**Radwan Rachid HOUSSEIN**

Contents

[Phase 1 3](#_Toc104760610)

[Requirement Analysis 3](#_Toc104760611)

[Requirements and Functionalities 3](#_Toc104760612)

[Some of the Possible Queries: 4](#_Toc104760613)

[Entity Sets: 4](#_Toc104760614)

[Atributes & Primary Keys: 4](#_Toc104760615)

[Relationships: 4](#_Toc104760616)

[Phase 2 7](#_Toc104760617)

[E/R Diagram [HERE] 7](#_Toc104760618)

[The Relational Schema [HERE] 8](#_Toc104760619)

[SQL EER Diagram [HERE] 9](#_Toc104760620)

[Our Programs UI 10](#_Toc104760621)

[**Contribution** 29](#_Toc104760622)

# Phase 1

In the first step, we have done requirement analysis, wrote requirements and functionalities and some of the possible queries. In the second step, we have written entity sets, attributes, primary keys, relationships and we have created E/R diagram.

## Requirement Analysis

University students' clubs are one of the most important things in the university life as it's also considered as a legacy from the senior students to freshman students sharing with them their experiences. A student club needs a database as much as any other facility as the need of storing data is increasingly necessary.

We think developing this database would actually help us gain real hands-on experience by using a full-fledged database management system and it would be skill boosting and fun experience because it has a good potential.

In our journey of gathering information and requirements for our database, after thinking about the functionalities of this database and consulting some of our friends who manage certain clubs in the university considering them as our customers we came up with following requirements and functions.

## Requirements and Functionalities

This university club membership database stores and keeps track of the club's members, subunits, committees, events and workshops that are organized by the subunits each semester. The system will be used by several different users (with different authorities) like a normal member who can see the list of events and workshops or an authorized member for checking active members’ list and register new members or another authorized member for managing the budget. The system can also be used by a committee member to study the club's status, members, events, achievements and goals in order to make future plans and draw a map to achieve the club's short and long term goals.

1. ‌ The system stores each member's unique ID, Name, gender, faculty, department, year, birthdate, phone number, email, address, and status (active or passive member)
2. ‌The members of the club can join subunits which are like sub clubs each subunit is focused on specific field. Those subunits are managed by some member of the club and each one of them has unique identifier, name and a capacity of members that can join.
3. Subunits organize workshops in which they provide seminars, courses and all other educational activities that can be joined by any member of the club whether he/she is a member of the subunit or not. Each workshop has unique identifier, unique Name (even if the workshop is organized several times if will labeled as name + the year/month it took place), type which indicates what kind of workshop it is, starting date, ending date, time, tutor/specialist and a capacity of members that can join.
4. ‌ Subunits also organize events; an event can be organized by one or more subunits also a subunit can organize one or more events which are attended by any member. An event has a unique identifier, Name, date and a time.
5. ‌The club has committees like administrative board, economic board and so on. Which also have their unique identifier, unique name and a foundation date. Each committee consists of members in different positions. A member can only join one committee.
6. We want to keep truck of the budget that is issued to each subunit on different times for different purposes. We store each budget's amount, type (type can be a descriptive word for the reason that budget was issued) and date.

## 

## Some of the Possible Queries:

1. What is the total of budgets spent on a given subunit in a given semester?
2. What is the current number of members of the club?
3. Which subunit has the most number of members?
4. Provide a ranking of managers based on the period of time they served as mangers.
5. Who is the most active member (calculating how many events and workshops they attended)?
6. Provide a ranking of members based on how active a member is.
7. Which members participated in less than a given minimum number of events and workshop (so can be marked as passive member)?

## Entity Sets:

* MEMBER (from the 1st point)
* SUBUNIT (from the 2nd point)
* COMMITTEE (from the 5th point)
* EVENT (from the 4th point)
* WORKSHOP (from the 3rd point)
* BUDGET (from the 6th point)

## Atributes & Primary Keys:

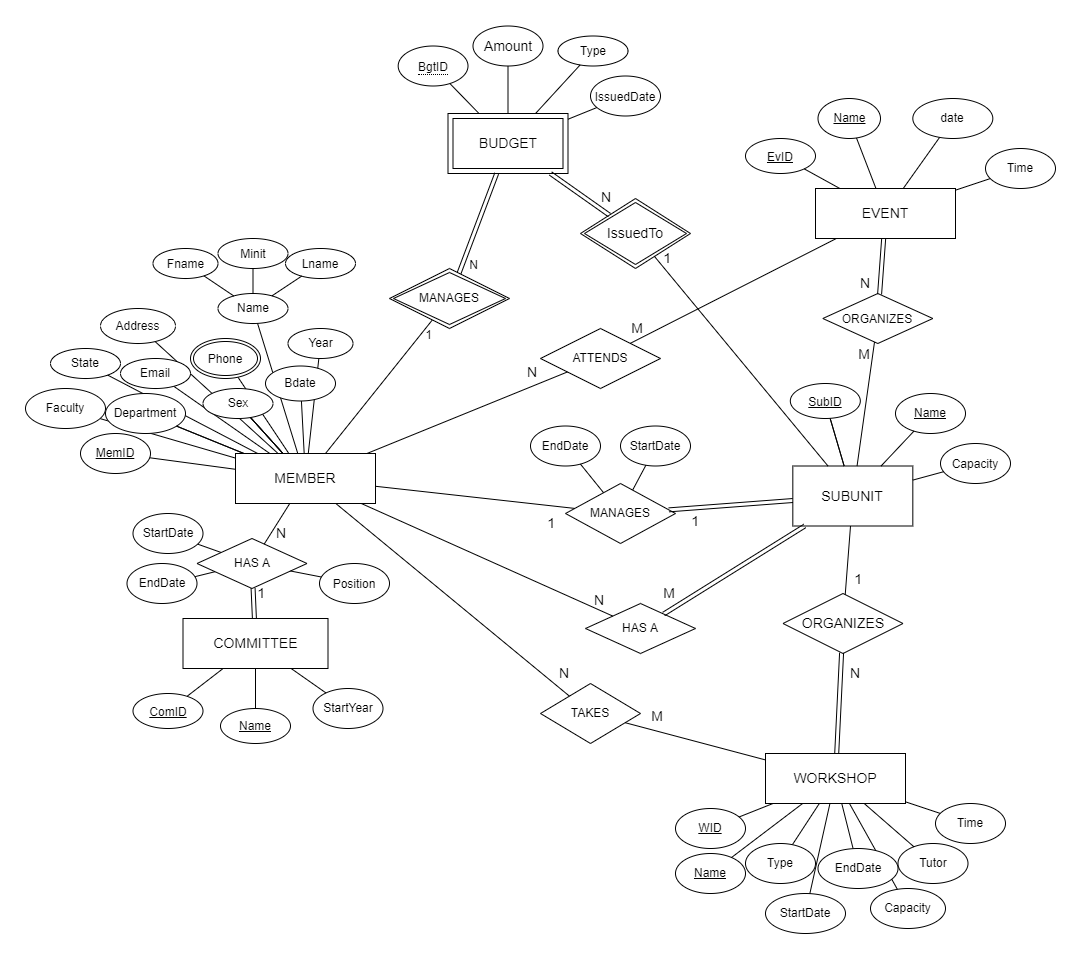
* MEMBER **(MemID**, Name, Sex, Faculty, Department, Year, Bdate, Phone, Email, Address, State)
* SUBUNIT **(SubID**, **Name**, Capacity)
* COMMITTEE (**ComID**, **Name**, StartYear)
* EVENT (**EvID**, **Name**, date, Time)
* WORKSHOP (**WID**, **Name**, Type, Tutor, Time, Capacity, StartDate, EndDate)
* BUDGET (BudID, Amount, Type, IssuedDate)

## 

## Relationships:

* **HAS A,** a 1:N relationship type between MEMBER and COMMITTEE. MEMBER participation is partial. COMMITTEE participation is total as users indicate in the requirements list. After consultation with users we decided to add Position attribute (which describes member’s position in the committee), start date and end date (to figure out which period of time a specific member held that position) to this relationship type.
* **MANAGES**, a 1:1 relationship type between MEMBER and SUBUNIT. MEMBER participation is partial. SUBUNIT participation is total. We found it appropriate to add the StartDate and EndDate (indicates when a member started and finished their duty as the manager of the sub unit) attributes to this relationship type like we did it in MANAGES relationship between MEMBER and COMMITTEE.
* **HAS A**, a M:N relationship type between MEMBER and SUBUNIT. MEMBER participation is partial. SUBUNIT participation is total. This indicates that a SUBUNIT can have several MEMBERs and a MEMBER can join several SUBUNITS as understood from the requirements.
* **ORGANIZES**, a 1:N relationship type between SUBUNIT and WORKSHOP. SUBUNIT participation is partial. WORKSHOP participation is total, after consulting the users we decided that a SUBUNIT can organize several Workshops that are related to their field of interest and only one SUBUNIT is enough to organize a workshop.
* **TAKES**, a M:N relationship type between MEMBER and WORKSHOP. MEMBER and WORKSHOP participations are partial; this indicates that several members can take/participate in several workshops.
* **ORGANIZES**, a M:N relationship type between SUBUNIT and EVENT. SUBUNIT participation is partial. EVENT participation is total, that indicates that several SUBUNITs might team up to organize an event and of course a single SUBUNIT can organize many events as required.
* **ATTENDS**, a M:N relationship type between MEMBER and EVENT. MEMBER and EVENT participations are both partial; several members can attend several events.
* **IssuedTo**, a 1:N relationship type between SUBUNIT and BUDGET, which is also the identifying relationship for the weak entity type “BUDGET”, considering a budget entity may only make sense and be worth storing if its issued to some SUBUNIT as the users pointed out. SUBUNIT participation is partial. BUDGET participation is total.
* **MANAGES**, a 1:N relationship type between MEMBERS and BUDGET, which is also the identifying relationship for the weak entity type BUDGET. It shows that there is a specific member who is responsible of the budget. MEMBER participation is partial. BUDGET participation is total.

[E/R Diagram](https://github.com/RadwanH/University-Club-Membership/blob/main/Diagrams/ER-Diagram.drawio.png?raw=true)



# Phase 2

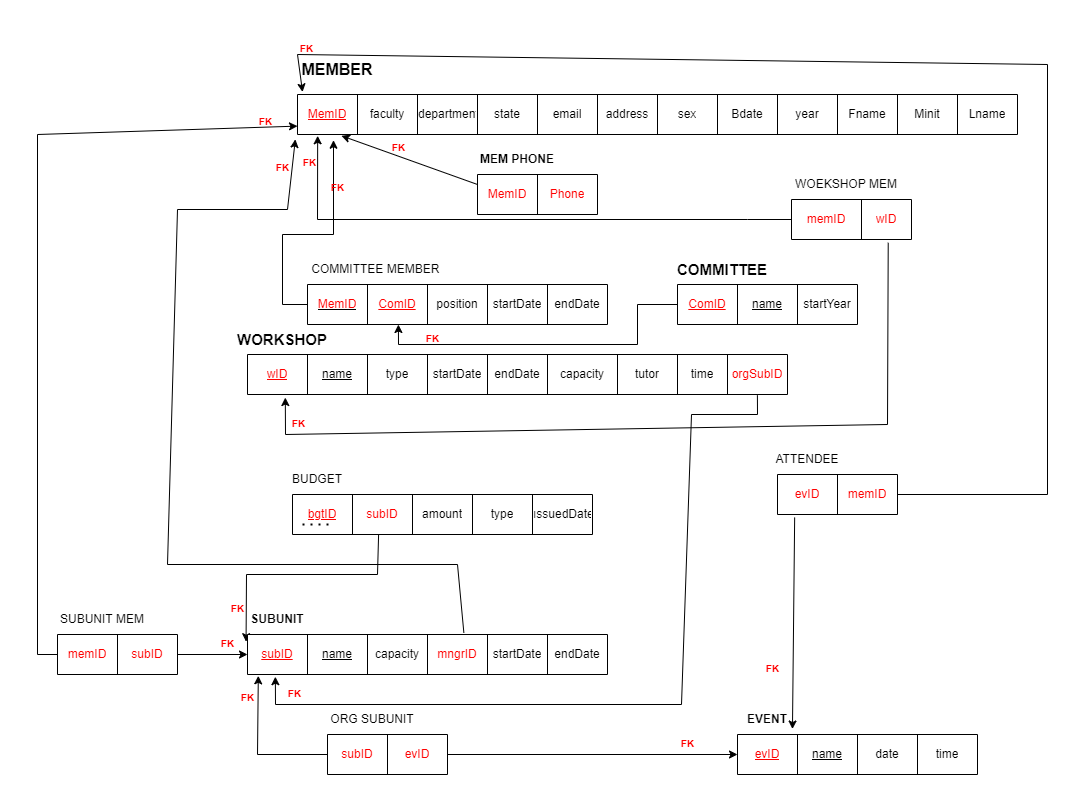
We got full point from phase 1 but, to maintain the simplicity and keep our database in BCNF we decided to make the following changes:

* The HAS A relationship between MEMBER and COMMITTEE entity sets was changed to M:N relationship as a specific member with position can actually be member in several COMMITEEs.
* The relationship ORGANIZES relationship between MEMBER and BUDGET entity sets was removed as we decided that it would make more sense if there is actually just one member that can be responsible for all budget transactions and that member’s position can actually be declared in the COMMITTEE table under the position attribute so this relationship would cause us redundancy (very unwanted thing) and there for was removed.

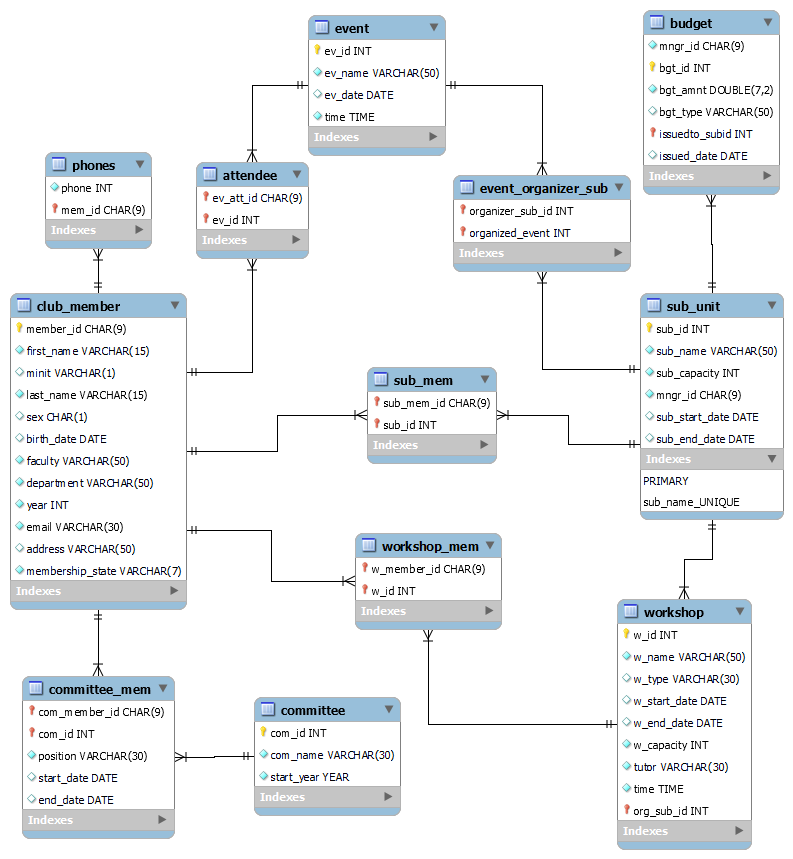
We updated our diagram according to these changes. We have made relational schema according to the diagram. Then we applied normalize. We used MySQL to create the database. We created our tables and relations between tables with appropriate commands. We entered enough data into the database that we have created.

## [E/R Diagram](https://github.com/RadwanH/University-Club-Membership/blob/main/Diagrams/ER-Diagram.drawio.png?raw=true) [[HERE]](https://github.com/RadwanH/University-Club-Membership/blob/main/Diagrams/ER_diagram.png?raw=true)

# The Relational Schema [[HERE]](https://github.com/RadwanH/University-Club-Membership/blob/main/Diagrams/Schema.png?raw=true)



# SQL EER Diagram [[HERE]](https://github.com/RadwanH/University-Club-Membership/blob/main/Diagrams/SQL_EER.png?raw=true)



Phase 3

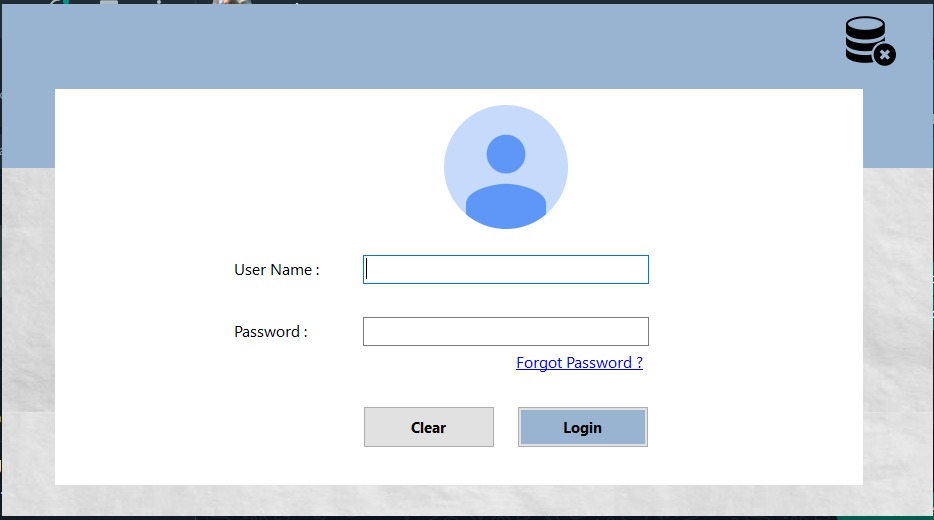
In phase 3 we demonstrated the database in a friendly UI user winforms with C# where we made a form every table in our database.

We made two different faces for the application: one for admin which is like a control panel where the admin can make all kinds of possible operations on the table and another face for normal members where they can see the tables without any authority of editing it as we expplained in the following images.

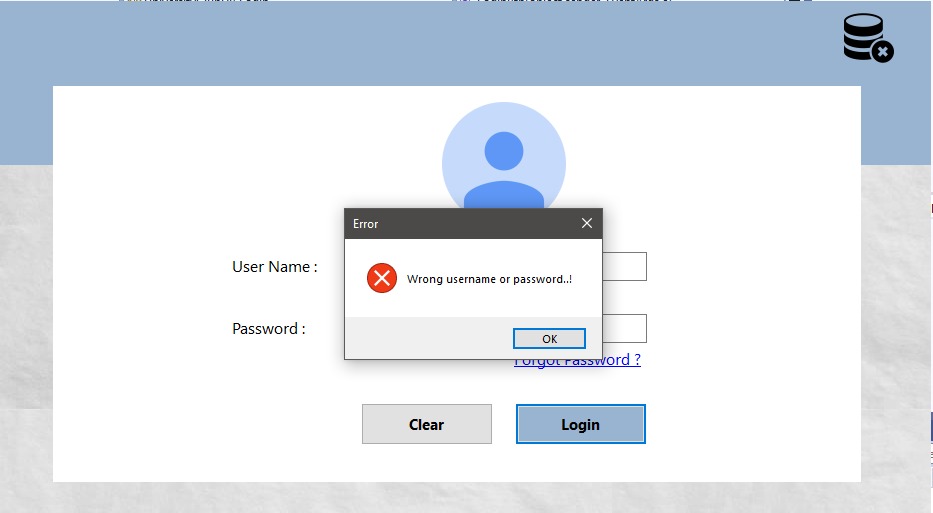
# Our Programs UI

Login:

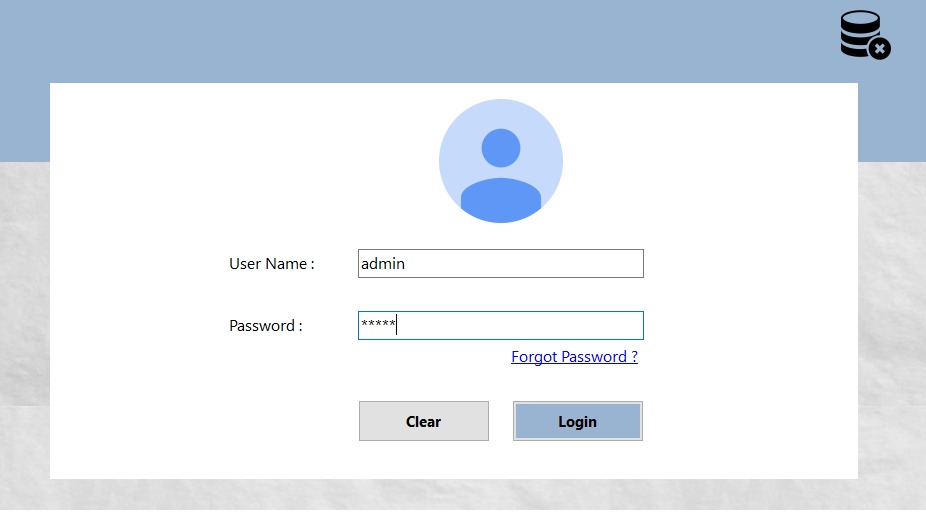
By pressing the login button, we are directed to either the member page or the admin page according to the password and user name we entered. We can delete the username and password we entered by pressing the Clear button.



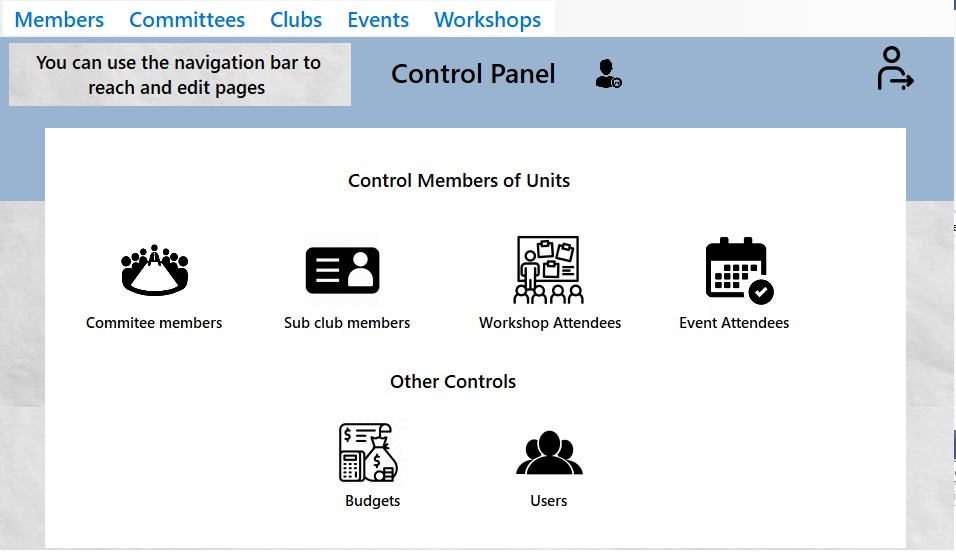
If we enter wrong username or password, warning message pops up.

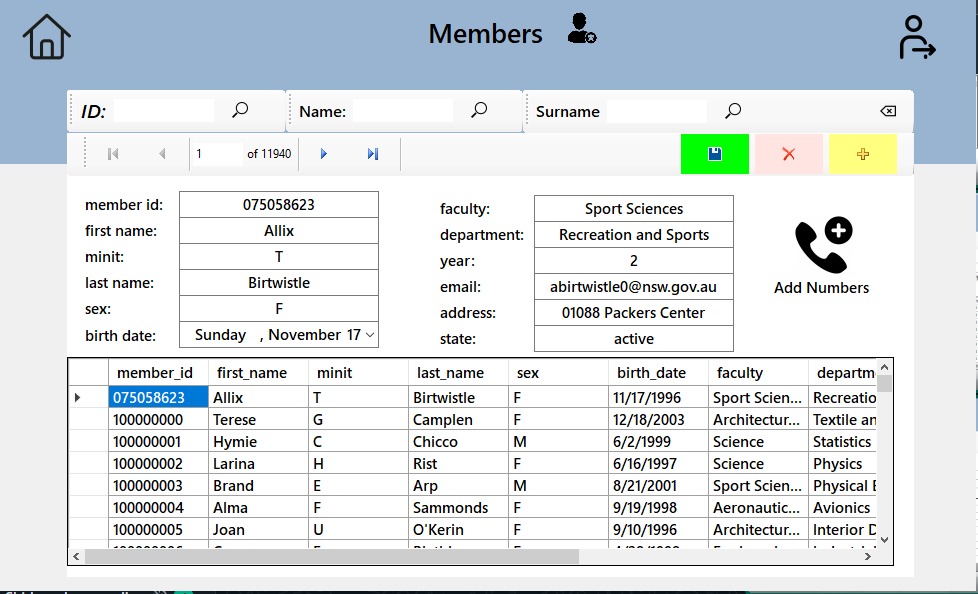


**ADMIN LOGIN**

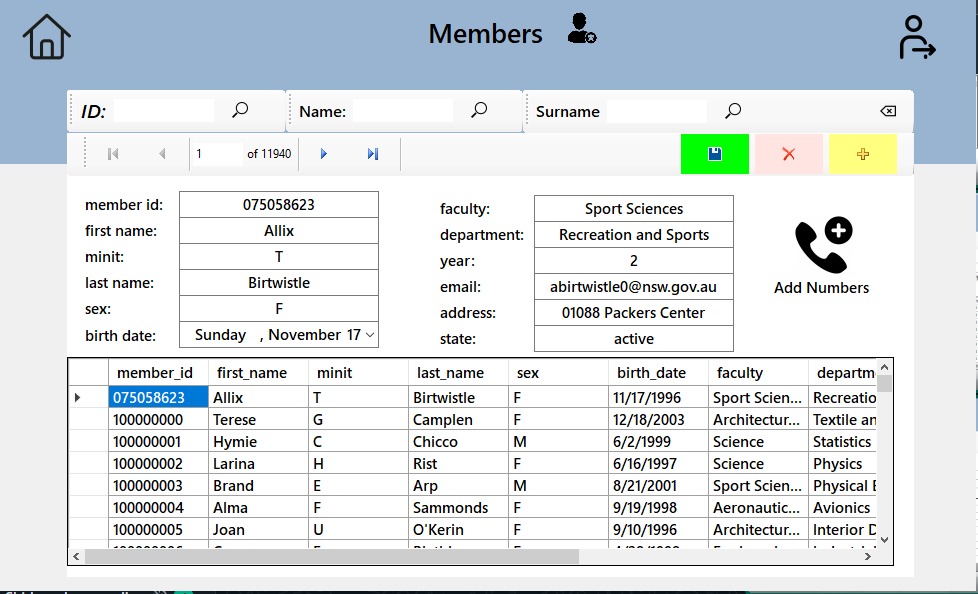


When admin logs in, he/she sees a page like this:

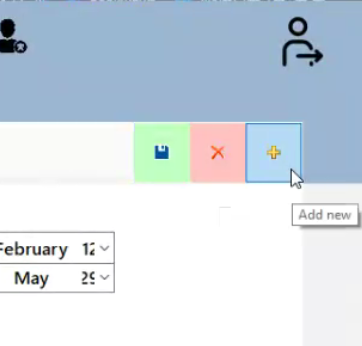
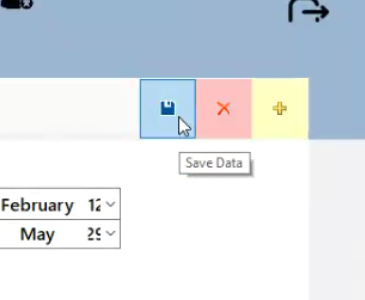


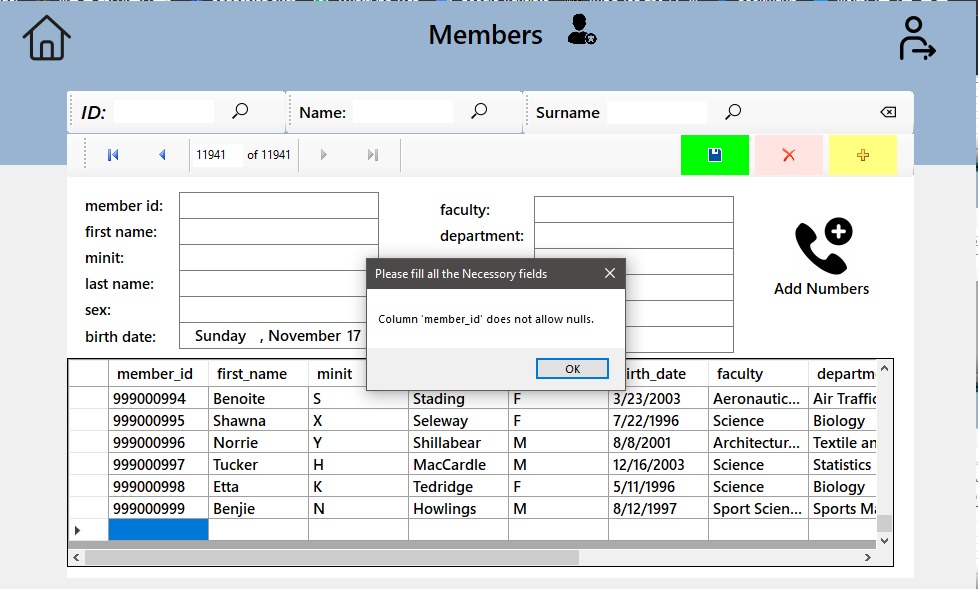
* The navbar buttons of the page opened with the admin login
* When admin clicks on the Members button: 

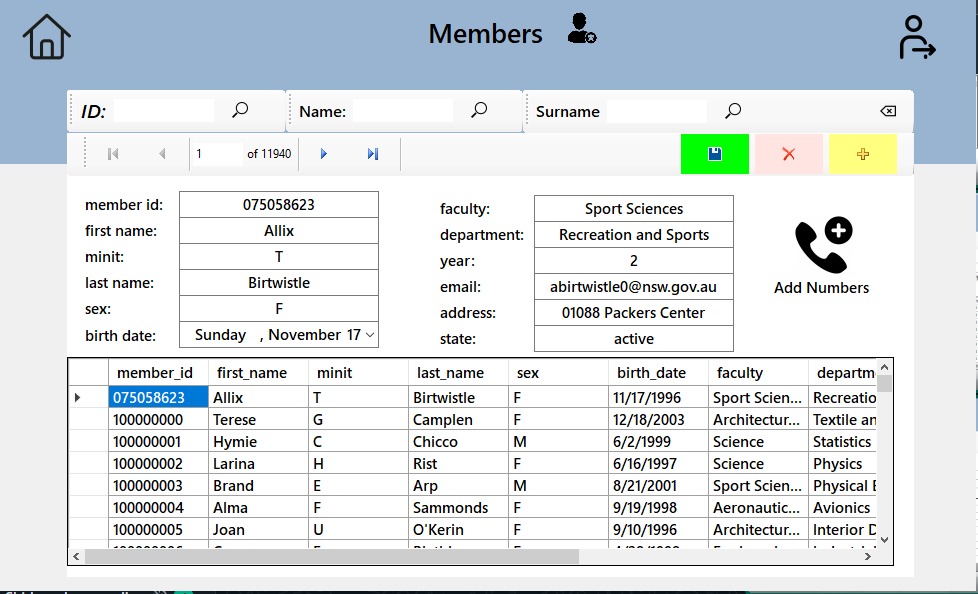
Admin can search for users by entering id, name and surname:



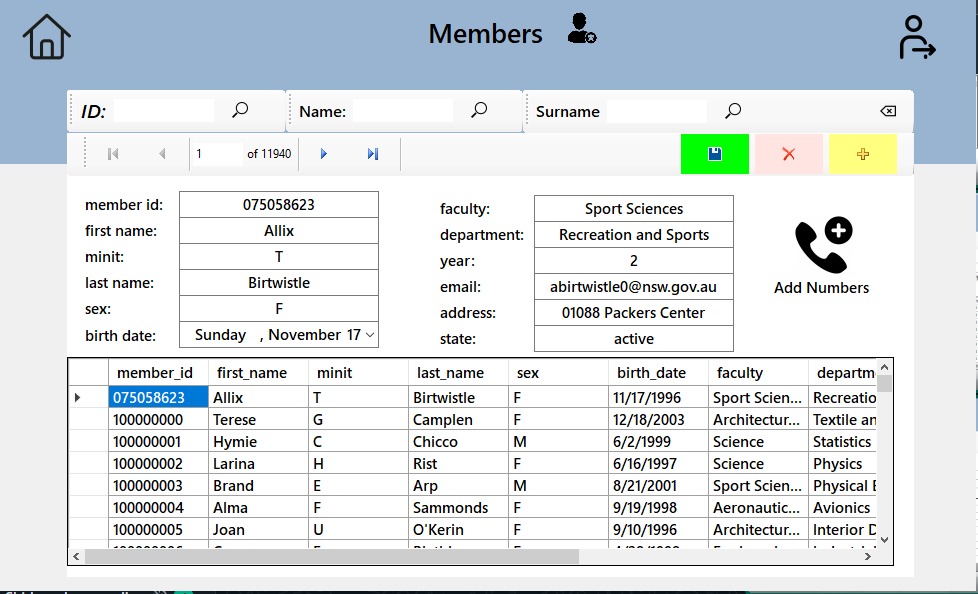
After entering the necessary information; admin can add a new user by pressing the add new button and then the save data button, can delete the selected user by pressing the delete button and then the save data button, can change the user information and update the user information by pressing the save data button:



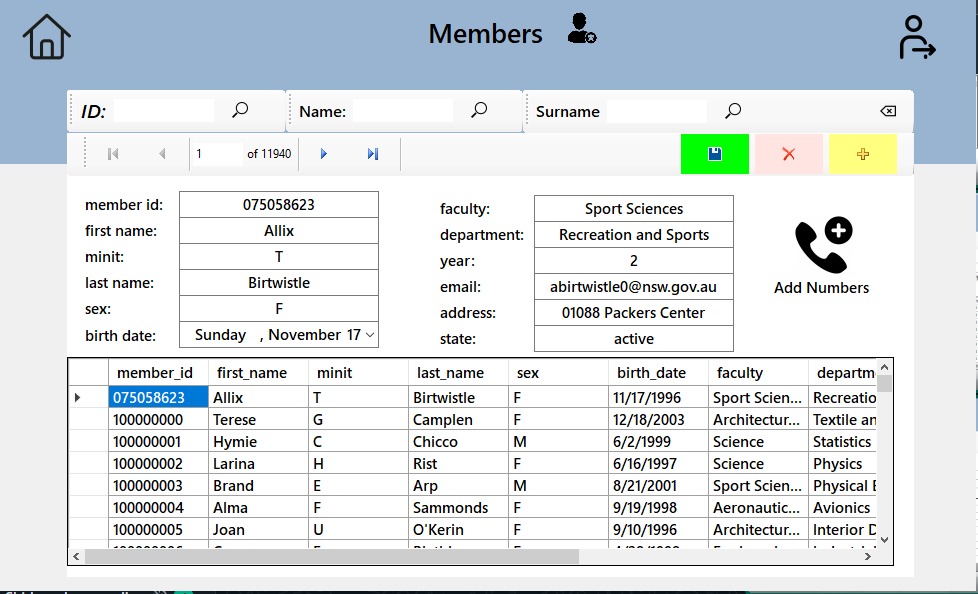


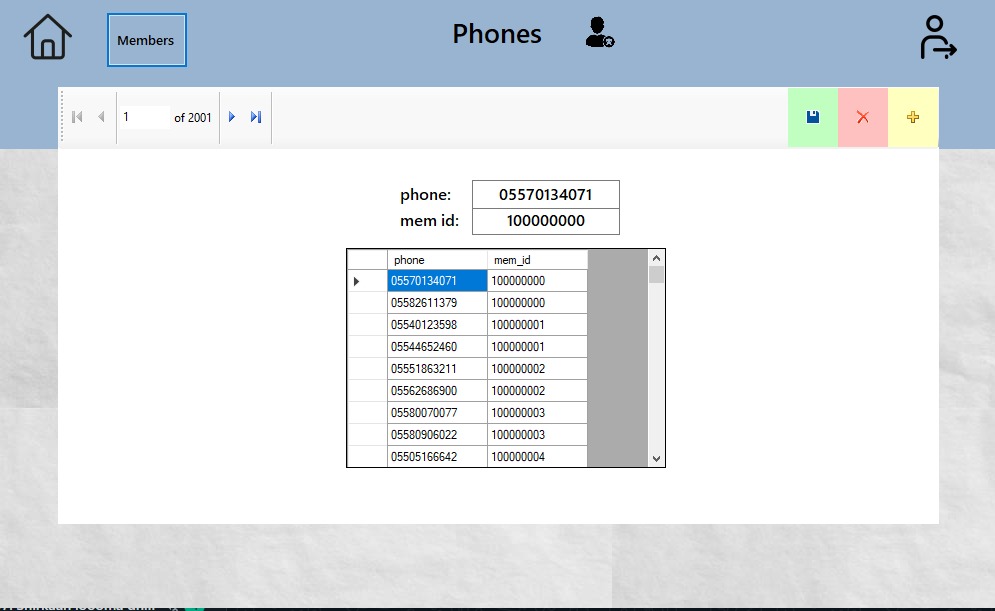


with this button admin can turn back the main page.

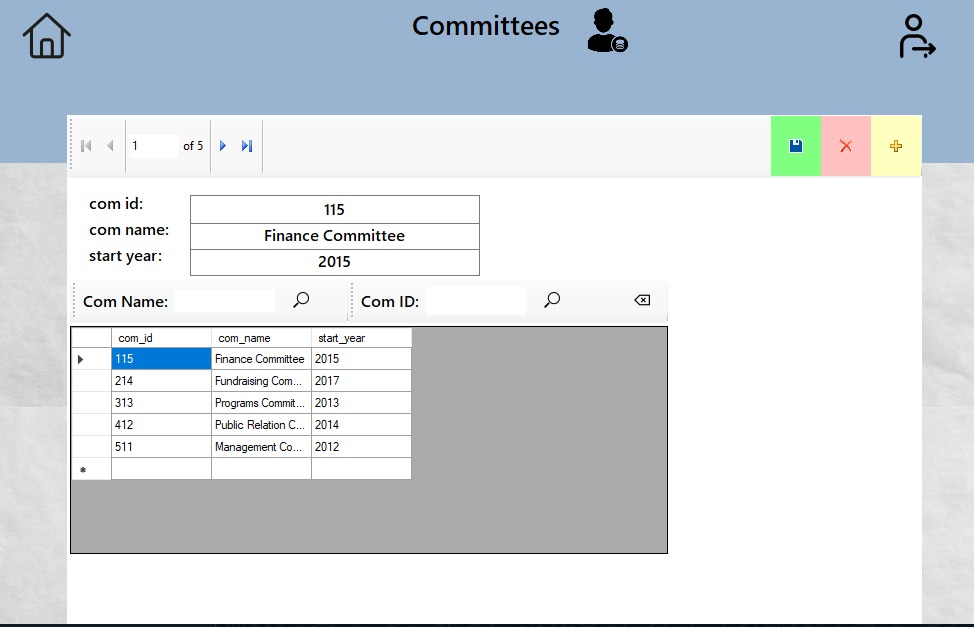


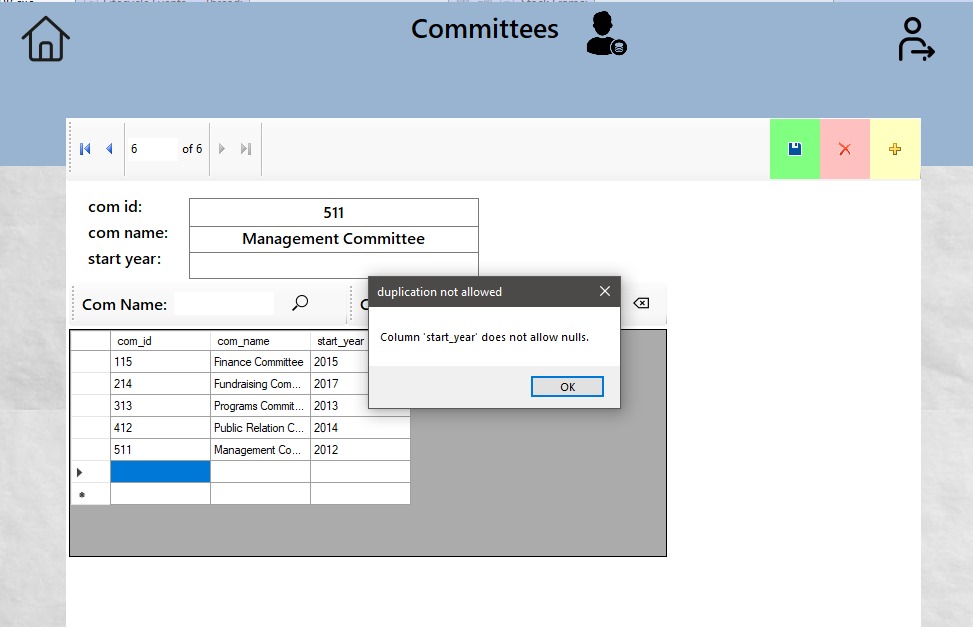
with this button admin can log out.

By pressing the add numbers button, admin can add multiple numbers for users on the page that opens: 



* When admin click on the Committees button: This page opens and admin can do the same things on the member page.

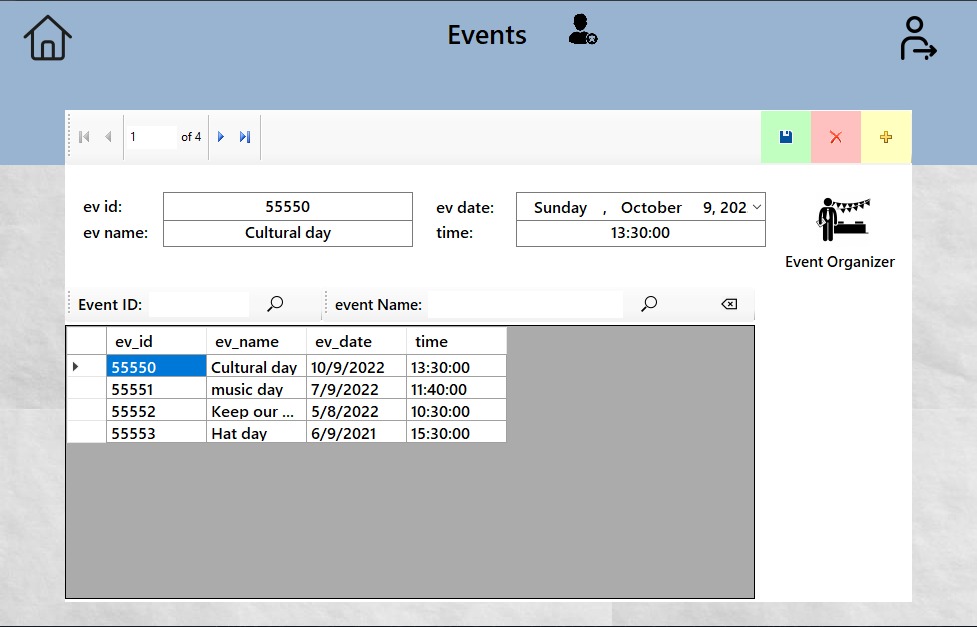




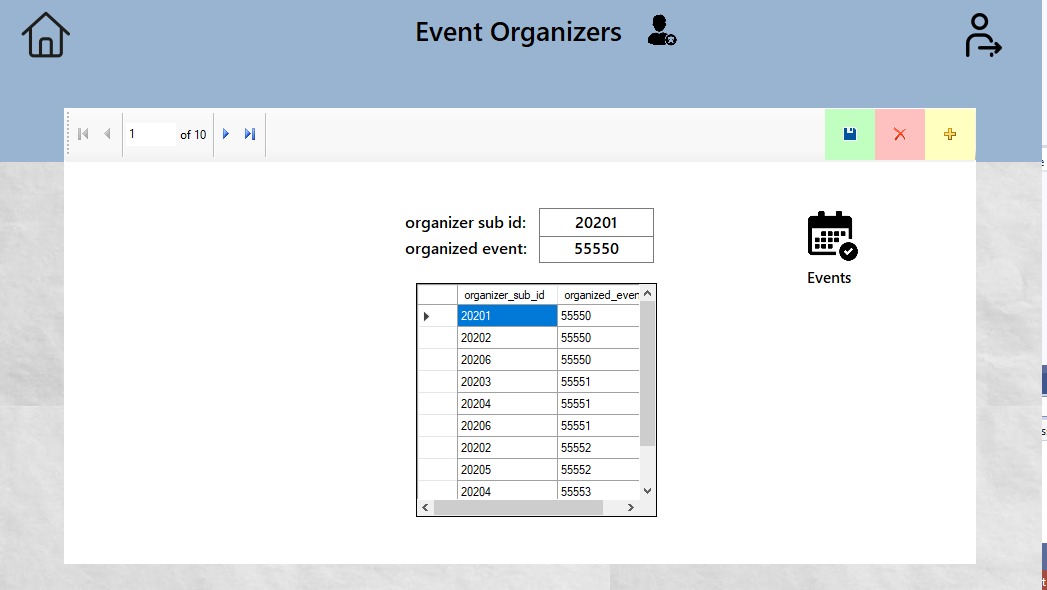
* When admin click on the Clubs button: This page opens and admin can do the same things on the member page.



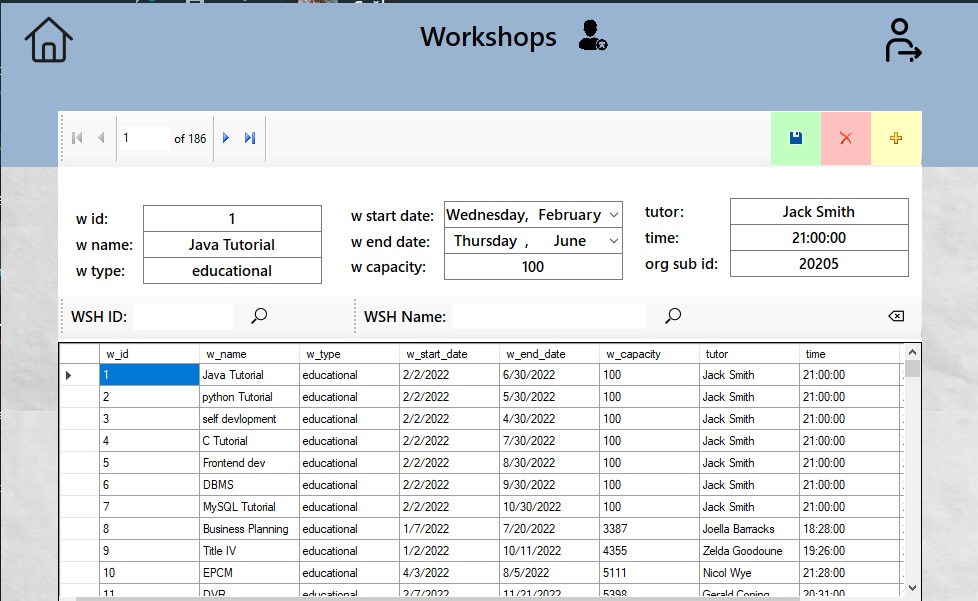
* When admin click on the Events button: This page opens and admin can do the same things on the member page.

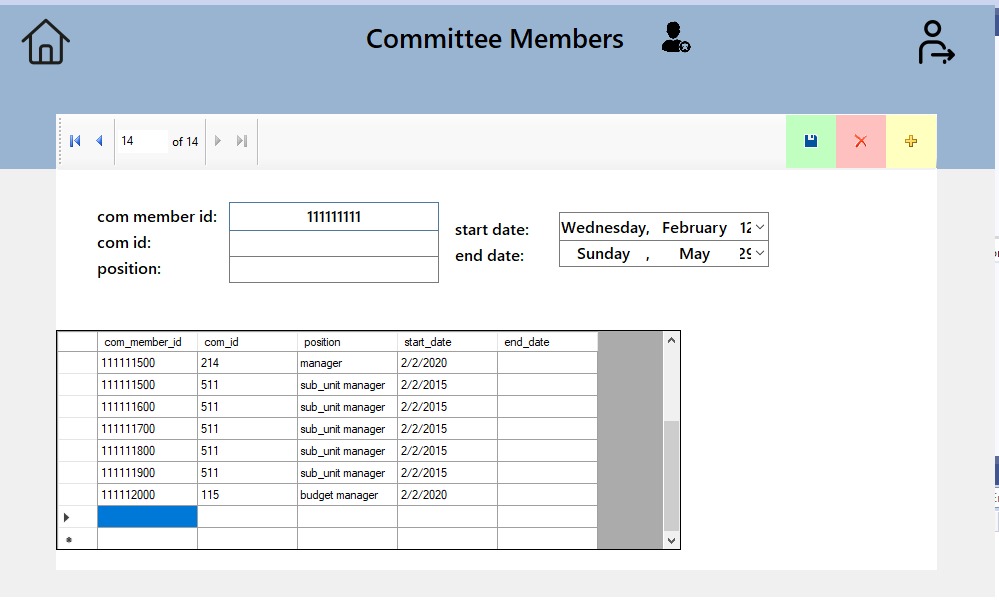


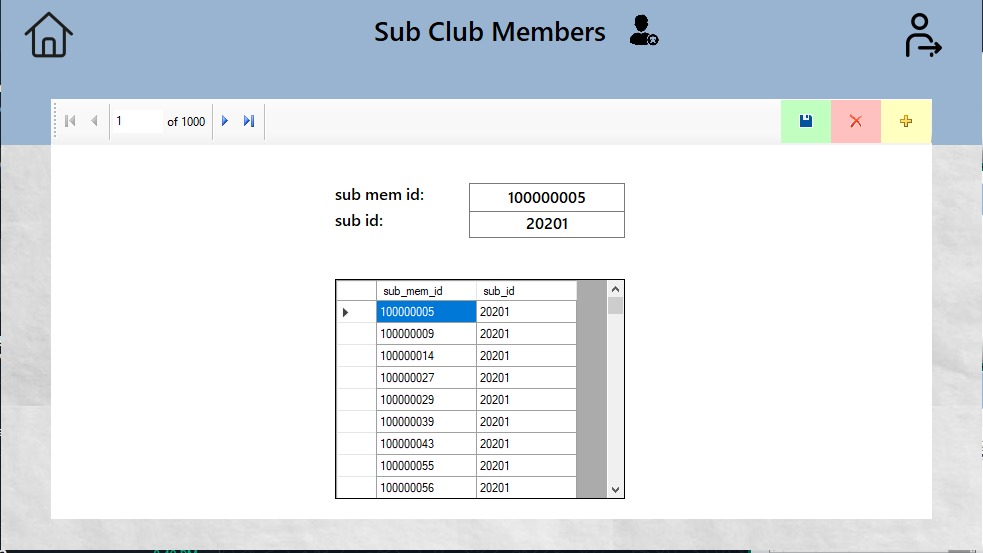
And admin can assign organizers to events with clicking the Event Organizer button

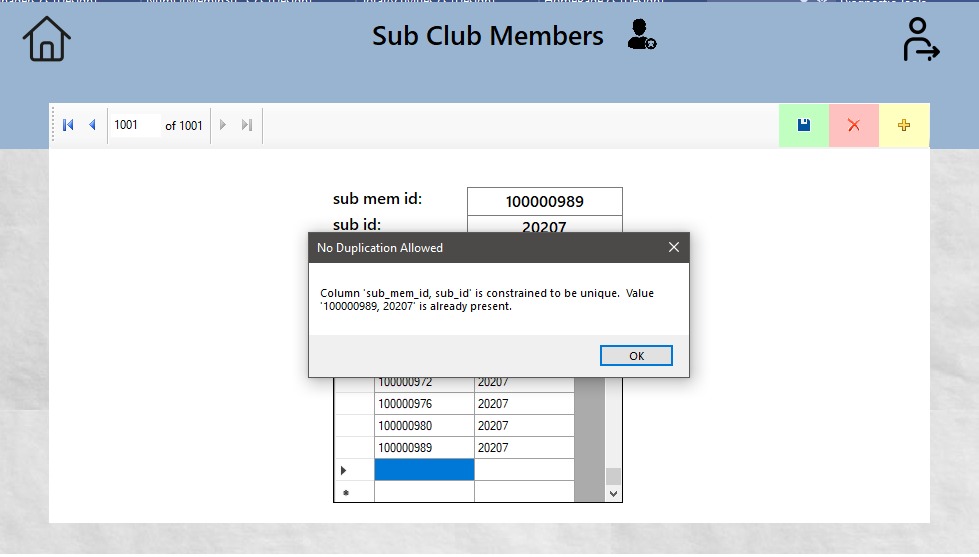


* When admin click on the Workshops button: This page opens and admin can do the same things on the member page.

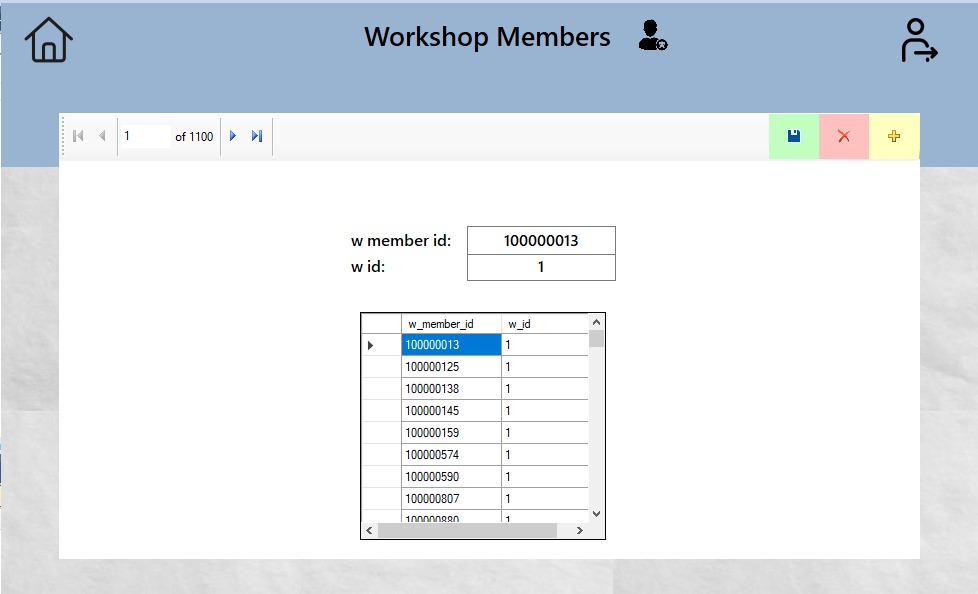


* The Control Members of Units buttons of the page opened with the admin login
* When admin click on the Committee members button: This page opens and admin can add, members to the committee and delete or update them.
* When admin click on the Sub club members members button: This page opens and can add, members to the subclub and delete or update them.

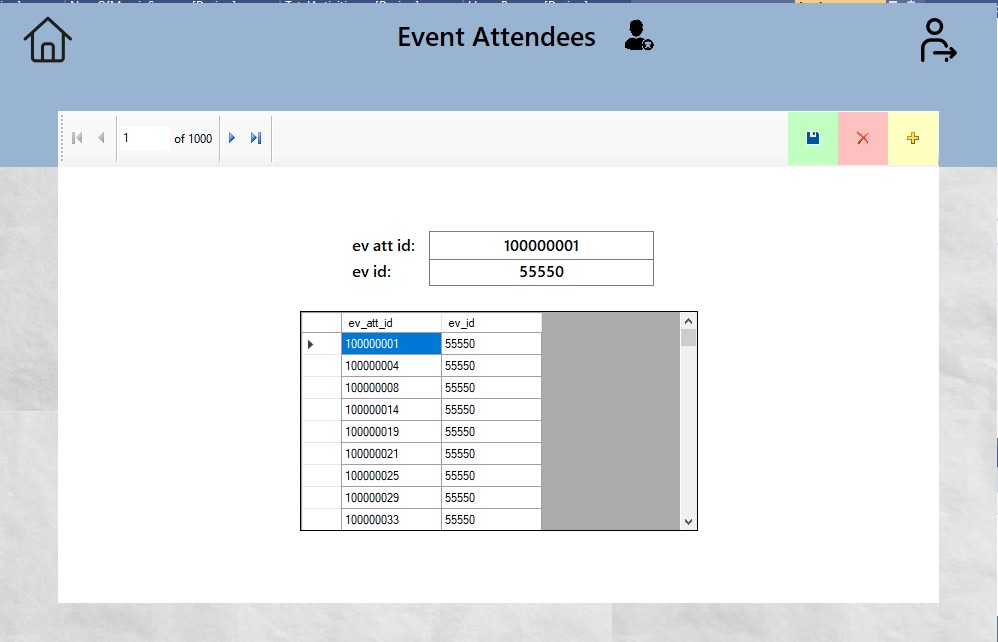


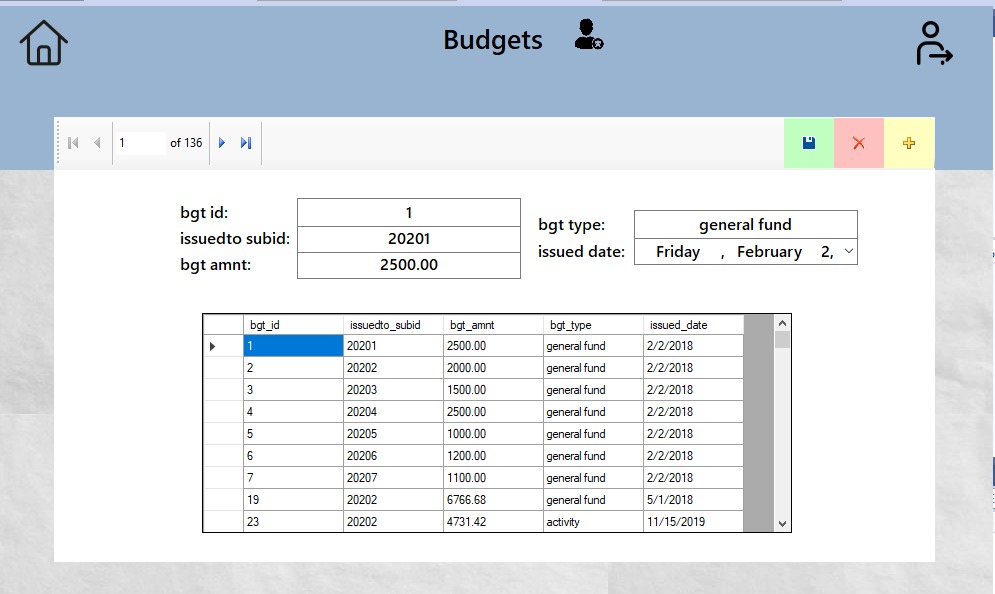


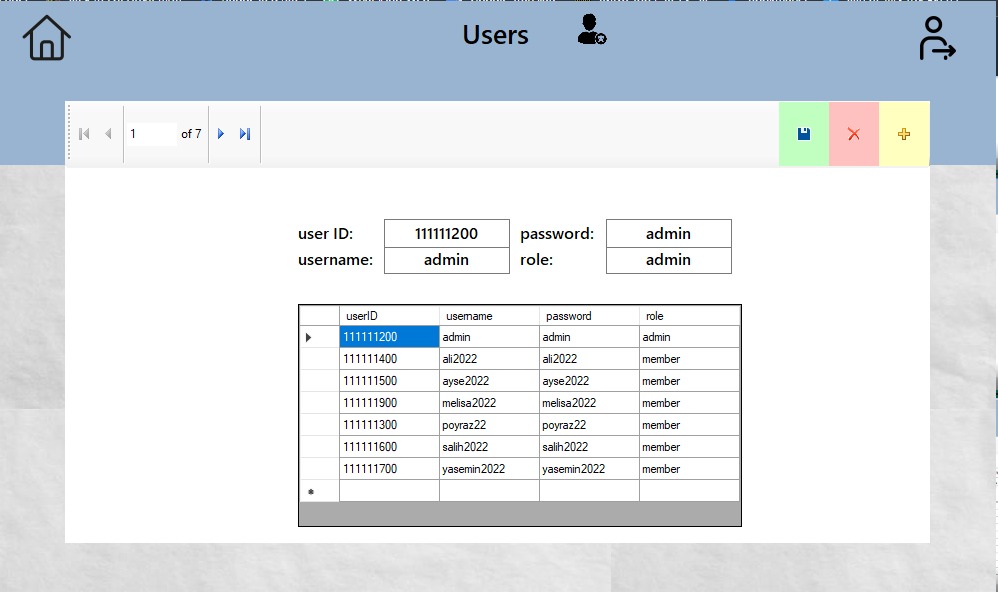
* When admin click on the Workshop Attendees button: This page opens and can add, members to a workshop and delete or update them.



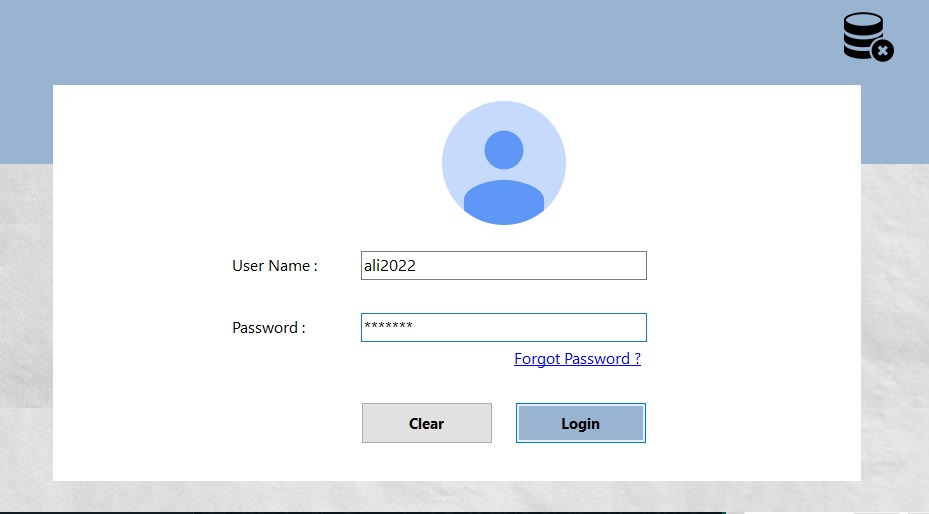
* When admin click on the Event Attendees button: This page opens and can add, members to an event and delete or update them.



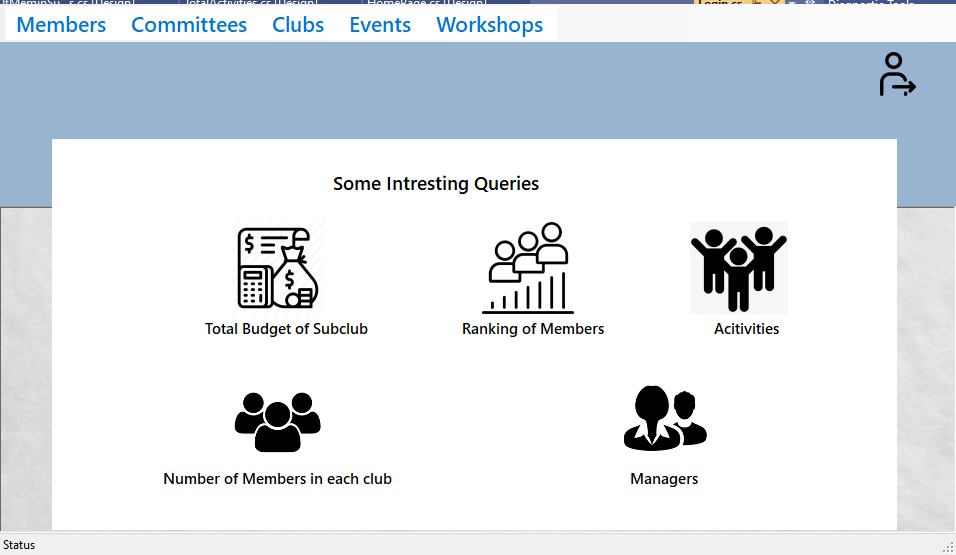
* The Other Controls buttons of the page opened with the admin login
* When admin click on the Budgets button: This page opens and can add, remove and update the budgets assigned to subunits.
* When admin click on the Users button: This page opens and can add, remove and update the users login informations.



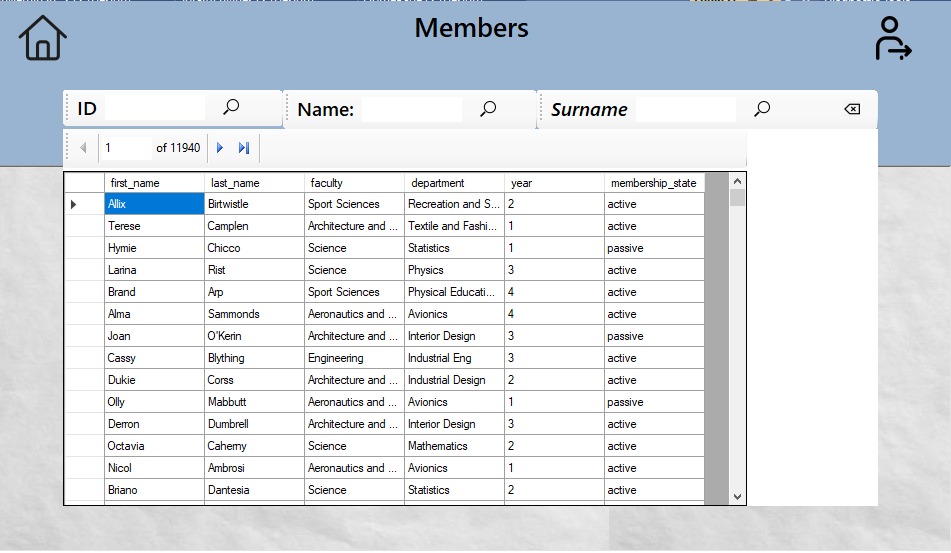
**MEMBER LOGIN**



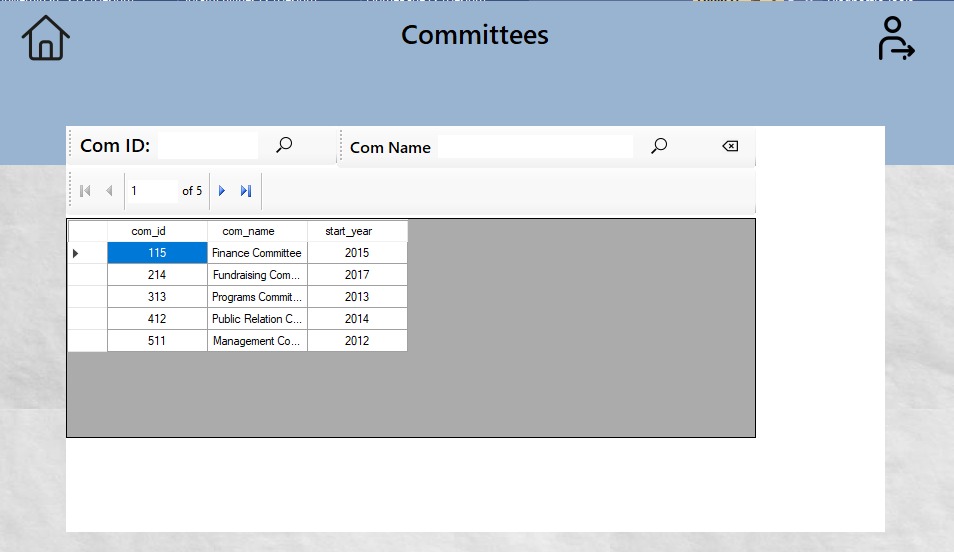
When member logs in, he/she sees a page like this:



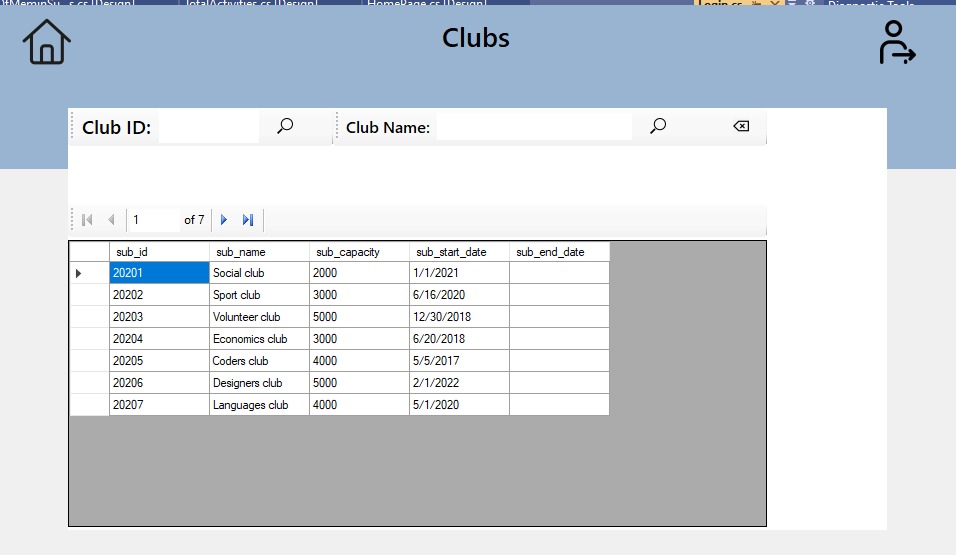
* The navbar buttons of the page opened with the member login
* When member clicks on the Members button: This page opens and member can see all members. (Members cannot see all the information as the admin sees it.) Members can search for users by entering id, name and surname:



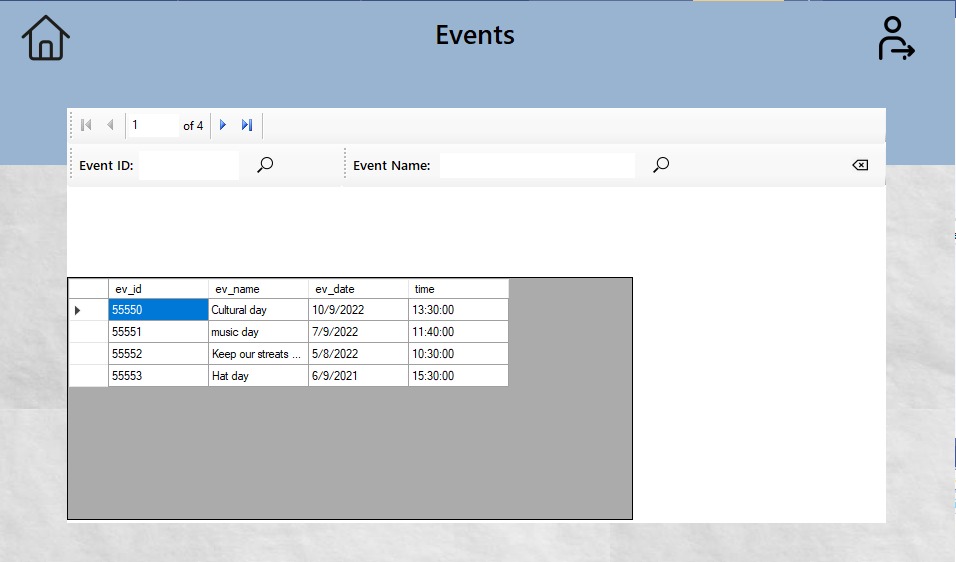
* When member clicks on the Committees button: This page opens and member can see all committees. (Members cannot see all the information as the admin sees it.) Members can search for committees by entering id, name.



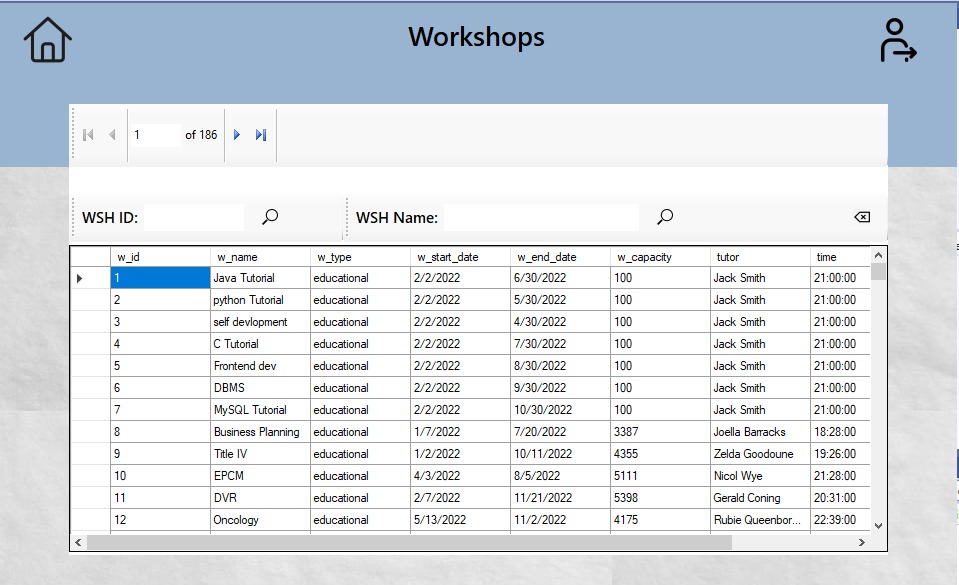
* When member clicks on the Clubs button: This page opens and member can see all clubs. (Members cannot see all the information as the admin sees it.) Members can search for clubs by entering id, name.



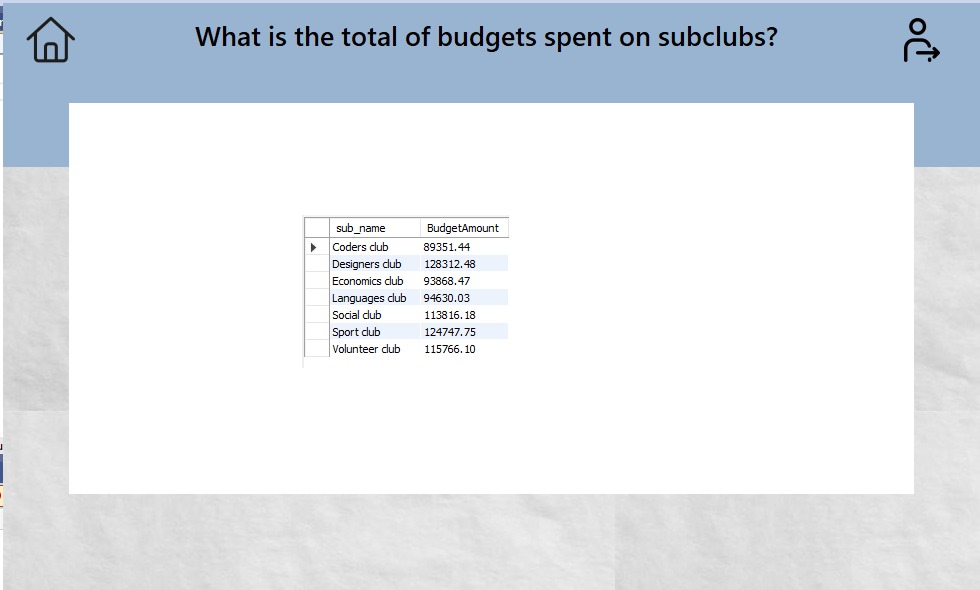
* When member clicks on the Events button: This page opens and member can see all events. (Members cannot see all the information as the admin sees it.) Members can search for events by entering id, name.



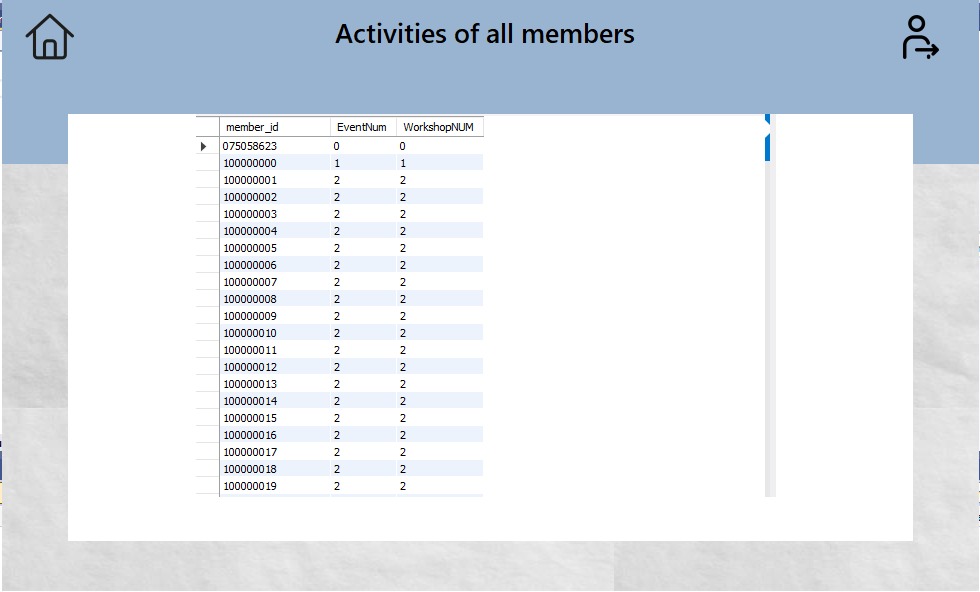
* When member clicks on the Workshops button: This page opens and member can see all workshops. (Members cannot see all the information as the admin sees it.) Members can search for workshopss by entering id, name.



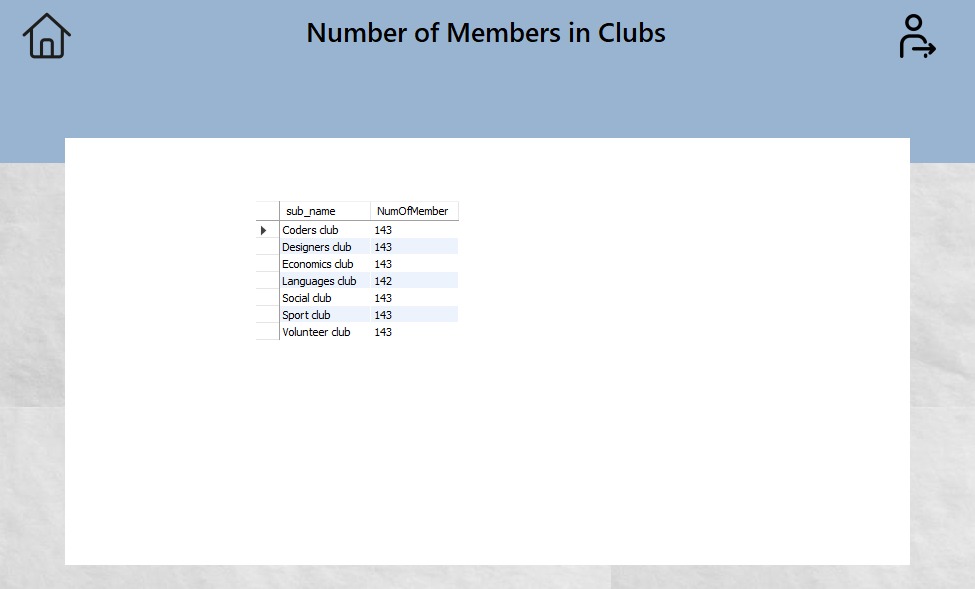
* The Some Interesting Queries buttons of the page opened with the member login
* When member clicks on the Total Budget of Subclub button: This page opens



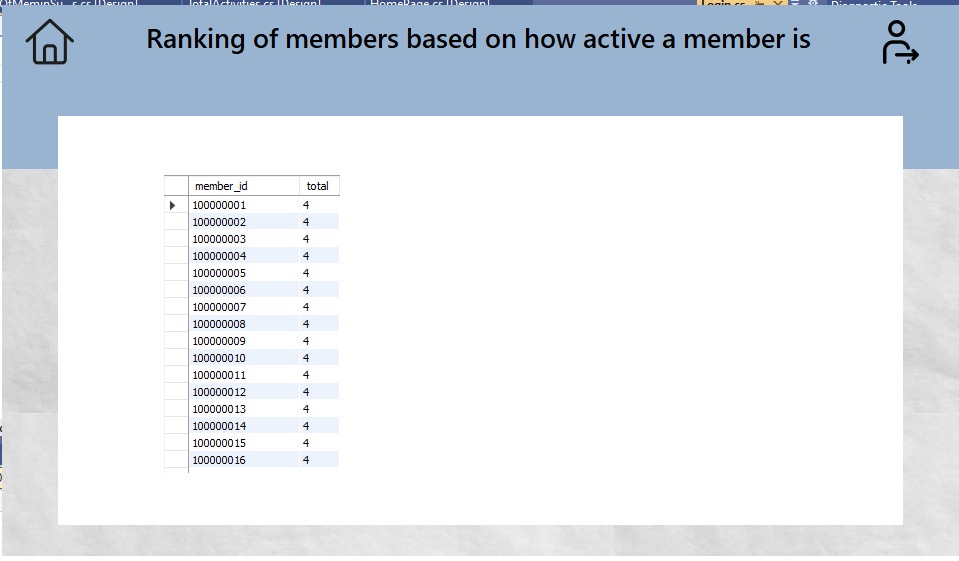
* When member clicks on the Activities button: This page opens



* When member clicks on the Number of Members in each club button: This page opens



* When member clicks on the Ranking of Members button: This page opens



* When member clicks on the Managers button: This page opens

