Introduction:

The following report will be covering the project for a Facebook database, to demonstrate identified requirements via diagrams, creating an application for the base user. The user I have selected for the database is a database manager.

For this project, I will be using MySQL and C# to store and to create an application to illustrate the requirements models. In order to make this happen, I have picked out the required requirements for the project, and as I want it to have access and be able to alter any data. And this system is going to be created for the database manager. The required requirements I want the systems to execute are:

- Edit and update users details
- Select and view users details
- Create a new user.

The database manager will have the basic power of being able to gain access to the database and being able to gain all the information it needs to about the user's data. The user's data contains information of their full name, friends, study, workplace and messages.

The database manager will be able to view all tables of data regarding all the users, viewing each and every information available. Another task set out for the database manager is to be able to create a new user or make any altercation on an existing user. Uses of UML diagram will be used to elaborate this, you can find types of diagrams used below:

- Use case Diagram
- Activity Diagram
- Entity Diagram
- Sequence Diagram
- State Diagram
- Class Diagram

Normalisation

UNF

UserID:
First Name
Last Name
Gender
Relationship
Hometown
Town/City
Workplace
Workplace Date
School/University
School/University Date
IDFriendship
Friend's First Name
Friend's Last Name
Text Date/Time
Text

1NF

UserID:
First Name
Last Name
Gender
Relationship
Hometown
Town/City

UserID:
User Workplace
Workplace Date

UserID:
User School/University
School/University Date

UserID:
IDFriendship
Friend's First Name
Friend's Last Name
Text Date/Time
Text

2NF
UserID:
First Name
Last Name
Gender
Relationship
Hometown
Town/City

UserID:
User Workplace
Workplace Date

UserID:
User School/University
School/University Date

UserID:
IDFriendship
Text Date/Time
Text

IDFriendship
Friend's First Name
Friend's Last Name

3NF
UserID:
First Name
Last Name
Gender
Relationship
Hometown
Town/City

UserID:
User Workplace
Workplace Date

UserID:
User School/University
School/University Date

UserID:
IDFriendship
Text Date/Time
Text

IDFriendship
Friend's First Name
Friend's Last Name

MySQL Sample

In MySQL, the source of the database, there are multiple tables that needed to create in order to import given data. Creating tables were very simple same as importing the data into the correct table. However, below the sample code belongs to the stage of creating foreign keys to make a relationship tables to tables. The foreign keys are placed in all the tables except the main one being "users" and also the "friendlist". Here are the sample codes of creating the foreign key:

```
ALTER TABLE isad157_ssyantantamang.universities

ADD INDEX userid_uni_idx (UserID ASC) VISIBLE;

;

ALTER TABLE isad157_ssyantantamang.universities

ADD CONSTRAINT userid_uni
```

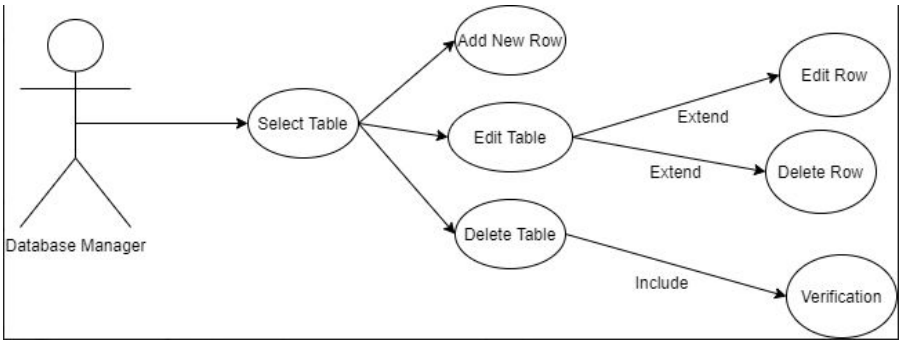
```
  FOREIGN KEY (UserID)
  REFERENCES isad157_ssyantantamang.users (UserID)
  ON DELETE CASCADE
  ON UPDATE CASCADE;
ALTER TABLE isad157_ssyantantamang.messages
ADD INDEX UserID_messages_idx (UserID ASC) VISIBLE,
ADD INDEX UserID2_messsages_idx (UserID2 ASC) VISIBLE;
;
ALTER TABLE isad157_ssyantantamang.messages
ADD CONSTRAINT UserID_messages
  FOREIGN KEY (UserID)
  REFERENCES isad157_ssyantantamang.users (UserID)
  ON DELETE CASCADE
  ON UPDATE CASCADE,
ADD CONSTRAINT UserID2_messages
  FOREIGN KEY (UserID2)
  REFERENCES isad157_ssyantantamang.friendslist (UserID2)
  ON DELETE CASCADE
 ON UPDATE CASCADE;
```

MySQL is the main source of the database, where all files regarding users information are stored into different tables, all allocated to its own category so that it will allow for the database to create relationships between each table using those foreign key and primary key within the table.

The rest of the tables needed to be altered so It can include the foreign key as it needs it to create a relationship between the tables. The user table was the simplest table as it did not need any adjustment, it did not need a foreign key as the main table was the user whereas the other table was connected to the user table.

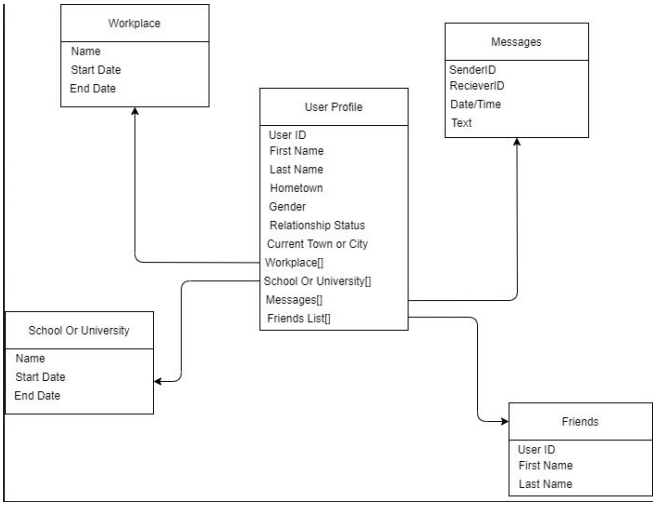
UML Diagrams

Use Case Diagram:



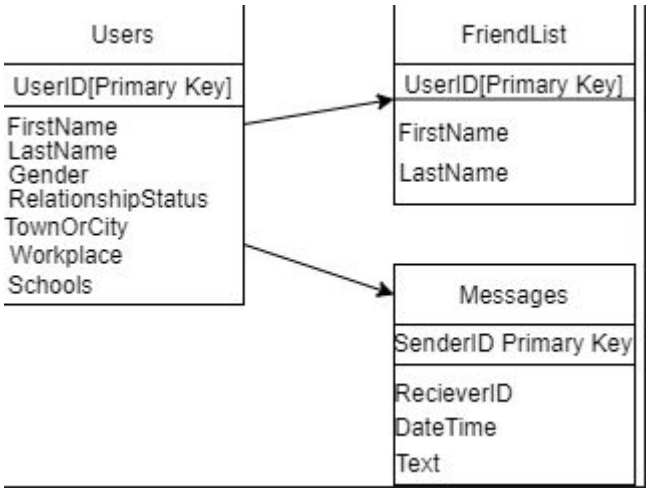
The use case diagram is simple and very understanding, draws out specifically what the database manager is supposed to do and the task it is given to execute. From the diagram, we can see that as a database manager you will select a table before being able to edit, create or delete any assets within the table.

Class Diagram

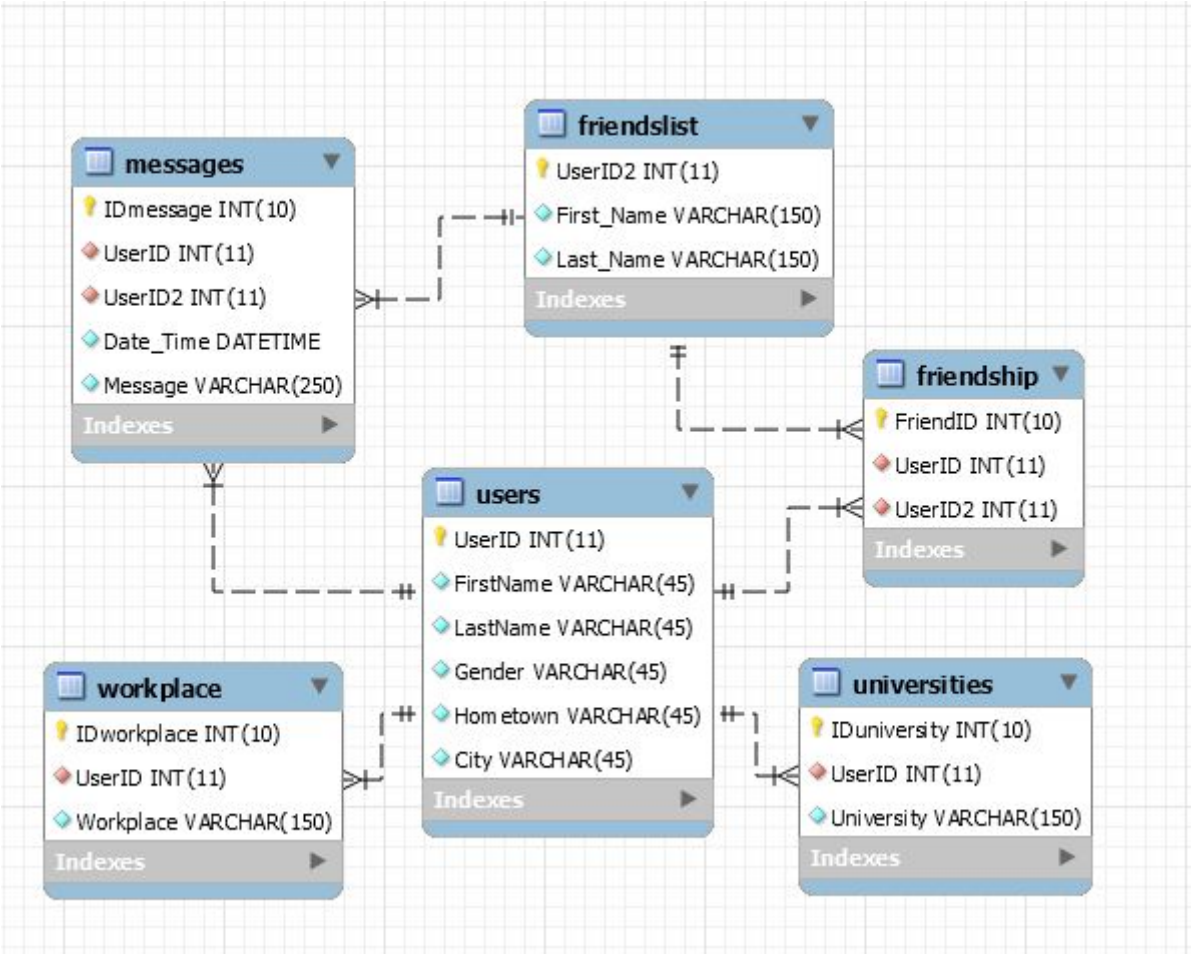


The diagram illustrates all box linked to its main class which is the middlebox, “User Profile”. The other boxes are the array for the datatypes used by the main class. The 4 different datatypes are all linked to the main class, “Workplace”, “Messages”, “School Or Univerisity” and “Friends”.

Entity Diagram

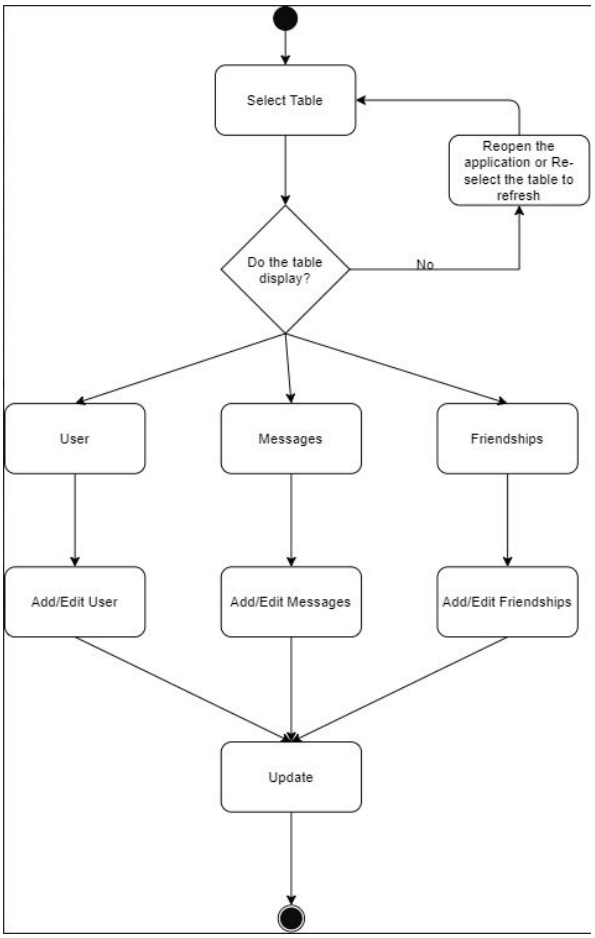


This is the initial entity diagram and it is very simple as there is only three entity. They are “Users”, “FriendList” and “Messages”. Both the “FriendList” and “Messages” are both in connection with “Users”. There are not many relationships in this stage so we are not able to fully in-depth see other information about the users just yet. This information can be about the user workplace or relationship.



This is the final entity diagram, this diagram is much more populated than the initial entity diagram as expected. This diagram was created in MySQL after completing all the relationship with the main table, “users”. Compared to the initial diagram we see three more tables, “workplace”, “universities” and “friendship”. In the final diagram, we see that “FriendList” is not in a relationship with “messages” and “friendship”. Foreign keys are now filled in the necessary tables with links being made to show where it belongs.

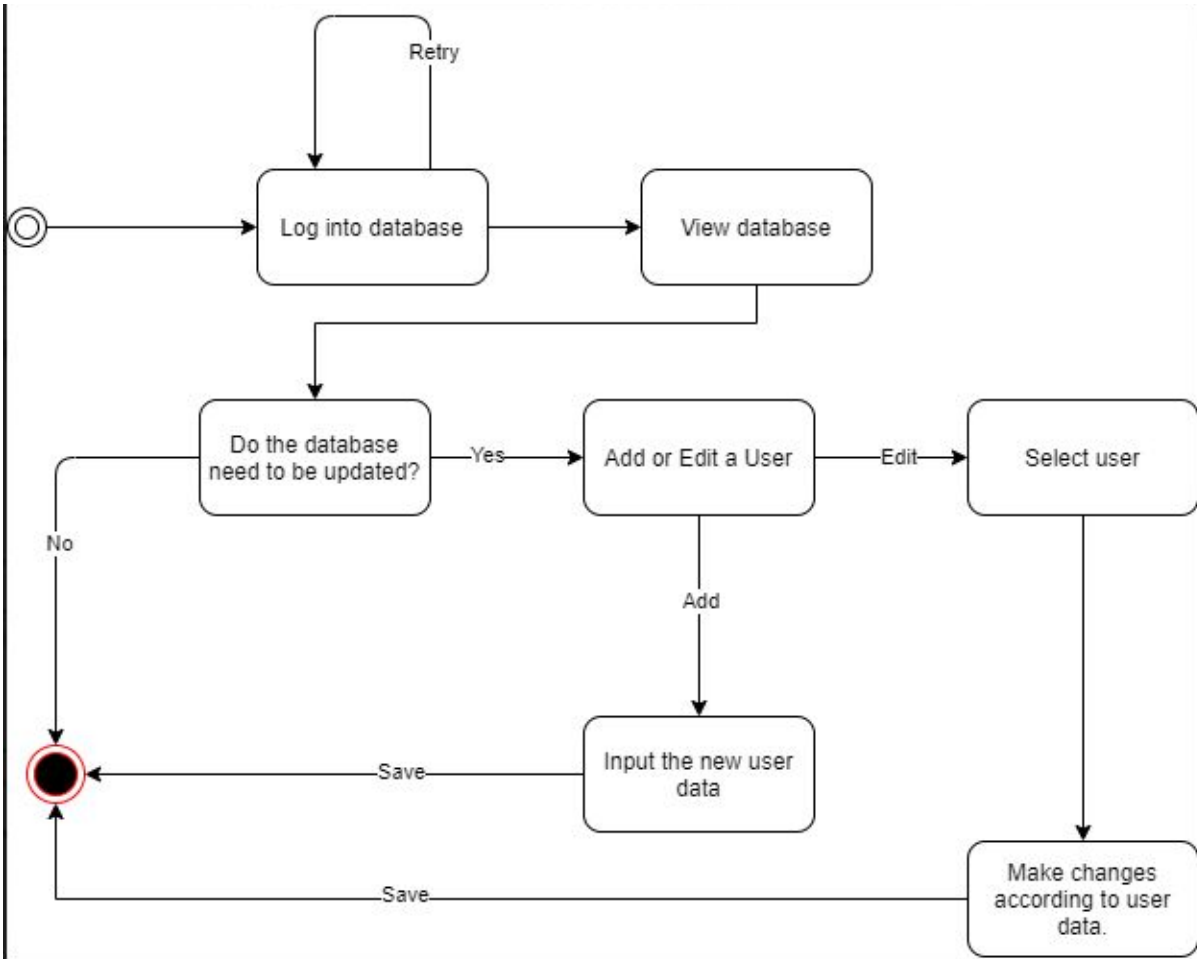
Activity Diagram



This is the activity diagram which showcases how the system user will be using the system, the system is very simple and it does not require much interaction between the system user and the system to complete its task. As basic, the system is designed to display and make altercation to user, messages and friendships tables. Each of the tables will be having the same function, as shown above, add/edit. The activity diagram is a great start to creating the system as it makes it clear what the system needs to be able to do, it only includes the main function and it is enough o create the required system.

State Diagram

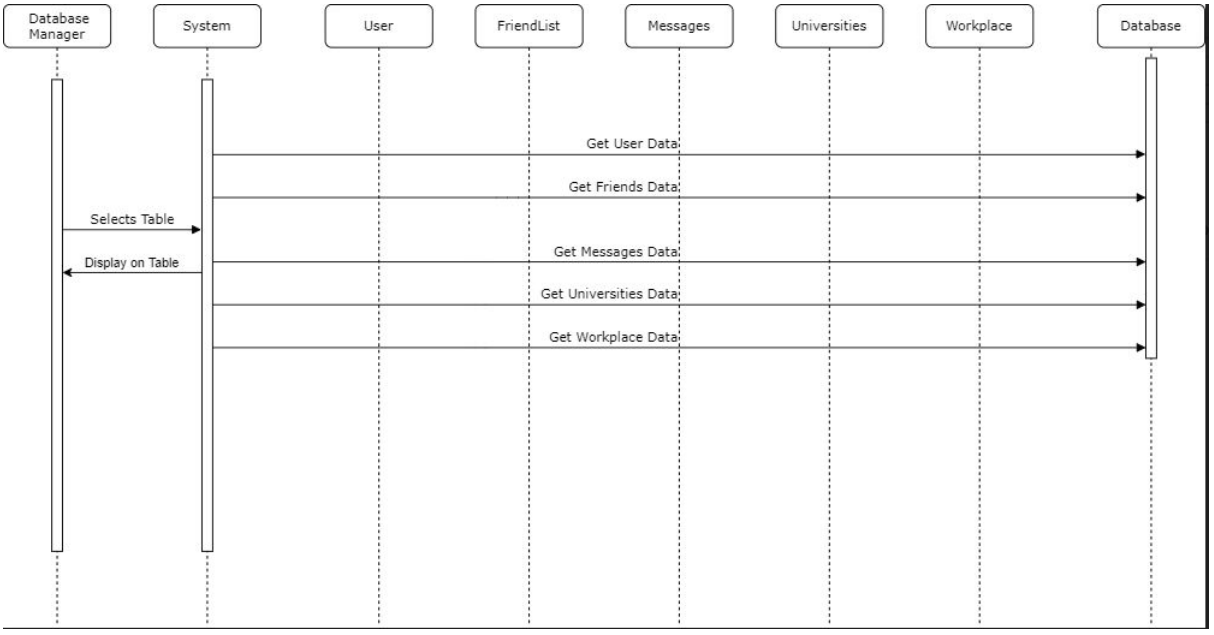
Computing



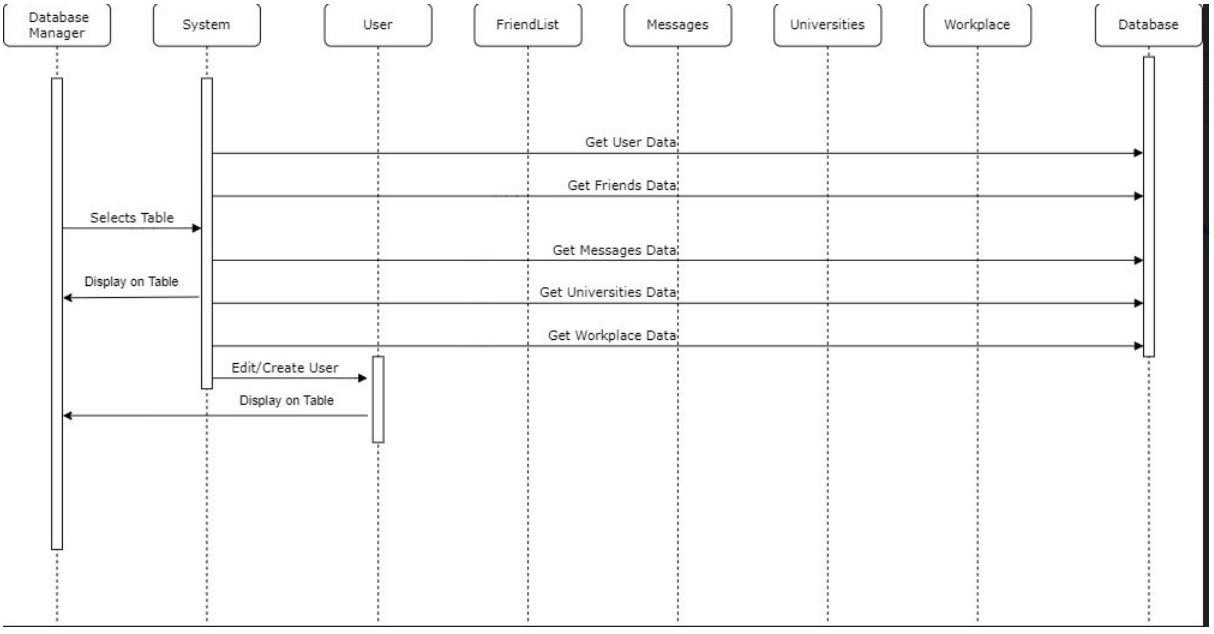
This is the state diagram and the diagrams read how the system user will gain access to the database and whether the user will make any changes or not. If the changes are needed then it will go with the workflow of the diagram and complete the series of activities.

Sequence Diagram

Display Sequence Diagram

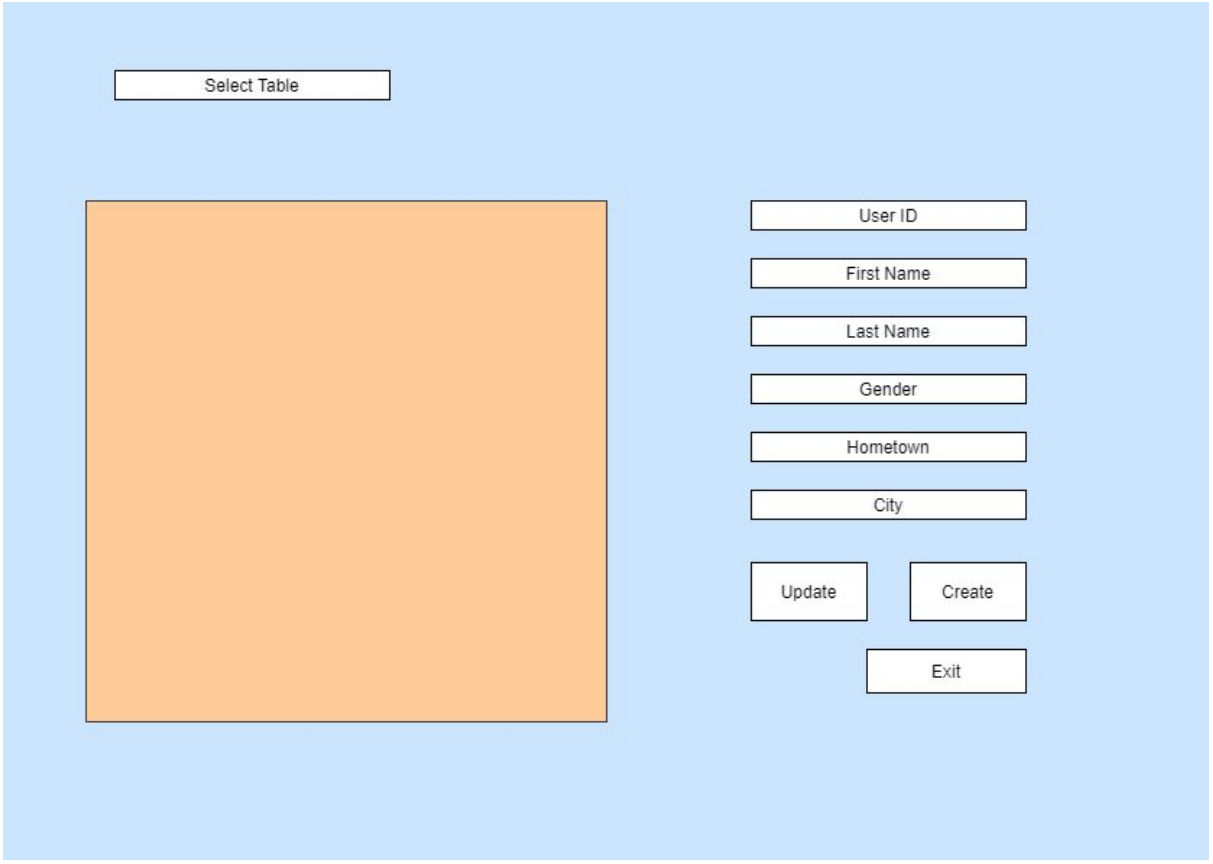


This is the display sequence diagram, to get the data to display is a not so difficult therefore it is only simple that the system has to only connect to the database to retrieve the data and have it read and displayed onto the system. As shown above, all the line are each connection made to the database trying to retrieve the required data. The line is labelled as “Get User Data”, etc. once the data are retrieved it is only then displayed on the system to the database manager.

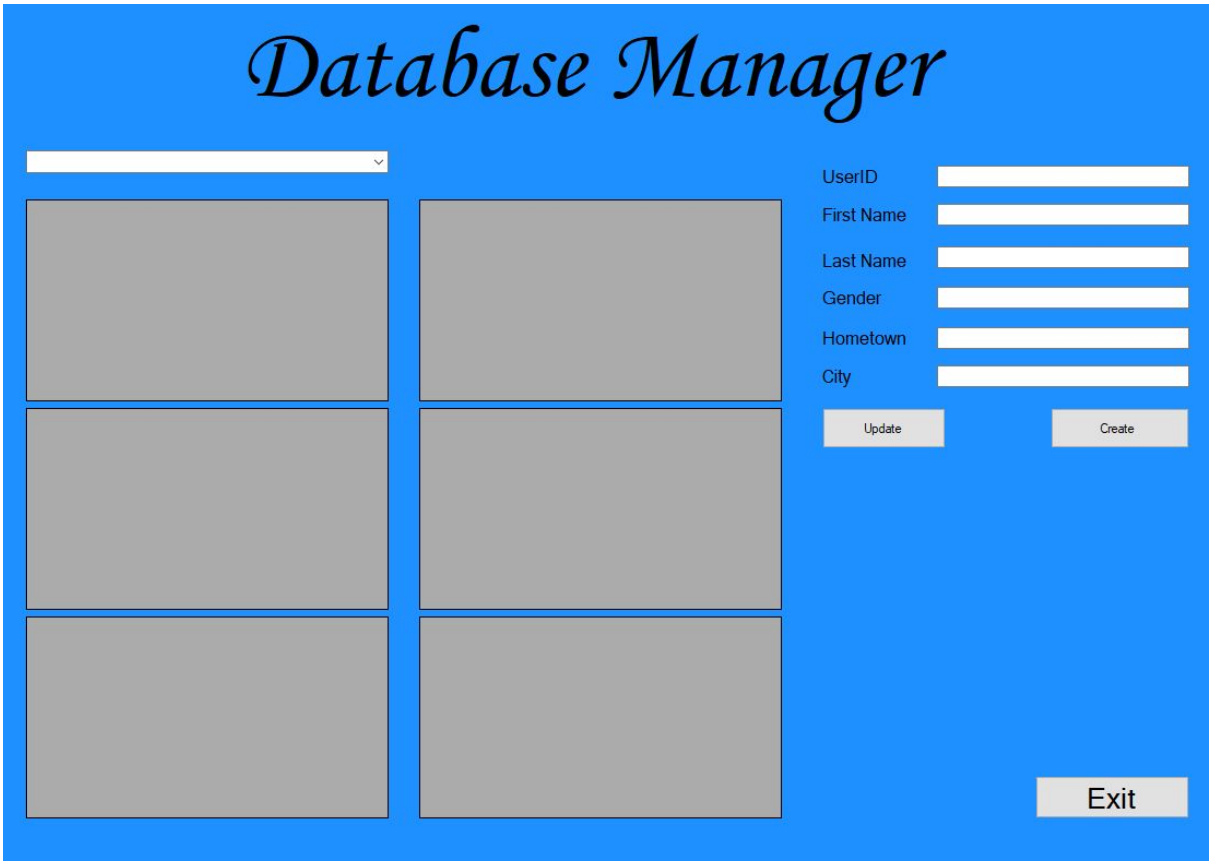


This is a similar diagram to the with the display diagram as above. The difference in this diagram is it includes the function of being able to edit or create a new user which is what we want the system user to be able to do. Once the data has been retrieved into the system, only then will the system user can make an adjustment to the user. In order to create a new user and have it into the database, it still needs to retrieve the data from the database. Once the system user has either created or made an alter to a user data it can then be displayed on the system.

System Design Brief



The system that is going to be implemented will have to be easy to use and be easy to understand without any confusion. The design above has everything I want the system to have in order to function how it is supposed to. The orange box is for the display of all the table data from the database, and you are able to select this from the select table bar at the top of the design. On the right, I have sets of boxes to display the user's details from the table when clicked. This is also the section where the system user will be able to add a new user or update an existing user.



This is the final implemented design. There was only one major change and that is the data display box. Before there was one box and you were able to view different tables on that one box. I have now implemented 6 boxes, each box is coded to display one table so now you are able to view more than one table at once.

I have kept the rest of the design the same, the select box and the user details box alongside the update, create and exit button.

Evaluation:

With the series of diagrams created, it gives a simple understanding of what the diagrams are illustrating. Although the diagrams are minimum it still shows how it is suitable for the required requirements and how it is in correlation with the implementation of the whole database system which is to be used by the database manager. The report has shown via varies of the diagram the function of the system for the based user, database manager. The function which was clearly laid out by the diagrams is editing, creating and deleting users within the database. However, there are diagrams which lack some expected functions which can forecast the idea of the function may not be implemented with the final implementation.

There could be more functions that can be implemented that can support the user more because so far there is only 3 main and only function are create, delete and edit. However, it later shows the delete is not implemented in the end so, with the fall of one function, there are only 2 real functions shown and implemented. The report and the system can be an improvement if the delete function was kept on with the rest of the diagram and also have it implemented into the system.

Mainly the function is focused on the user's details only, it does not have an effect on the **workplace** and **university** or **school**. The report does not base any function on the mentioned. Implementing this could really improve and give more purpose to the user. This could be useful as these also needs to be changed if the user has any changes and yet it is not possible to make any altercation to these data as shown by the diagram, it can only change the user's name, gender and hometown/city.

The final system designed brief is designed and implemented well, all the table are shown on the available boxes and it is an advantage of having these multiples boxes as the user can see all sets of tables at once whereas with the previous design there was only one box to display which mean the user would have to keep changing the tables to view it so it takes extra time and effort to work around it.

The application is very useful however the boxes which are used for displaying the sets of tables are confusing and this can be improved by having them labelled. As the six boxes are all compact and lose to each other it can be confusing on which to look at and select. Without labels, they all look the same and create confusion. This can result in user editing the wrong set of data.

Despite the lack of function, the system database works well as it displays all the tables filled with the correct and necessary data. The create and update/edit function implemented works as it supposed to. The diagrams created could be improved and make it more details to give a lot more understanding as it only just gives the basic understanding. The textboxes which are there are to fill with information when selecting rows works and changes when selecting each and every row however during the testing it did not work until closing and reopening. This could be a problem with the code or with the user's system.

Link to GitHub:
https://github.com/SubodhTMG/ISAD157