Introduction

Within this coursework I will be setting out a plan for a piece of software of my chosen scenario, using the appropriate UML models as well as a series of entity relationship diagrams, both initial and normalised. I will also be coding a basic user interface in order to display a large data set appropriately. All diagrams will be drawn in draw.IO as well as described/annotated.

My chosen scenario will be a design for the police department, to be used to scope out suspected criminal activity by looking into the suspected perps social media message and friend connections.

GIThub Link

<https://github.com/La-Ola/ISAD157>

Requirements

Functional requirement for this application include:

* Storing users names
* Storing user IDs
* Storing users friends
* Store messages
* Store workplace
* Store school
* Allow users to interact with data- interface

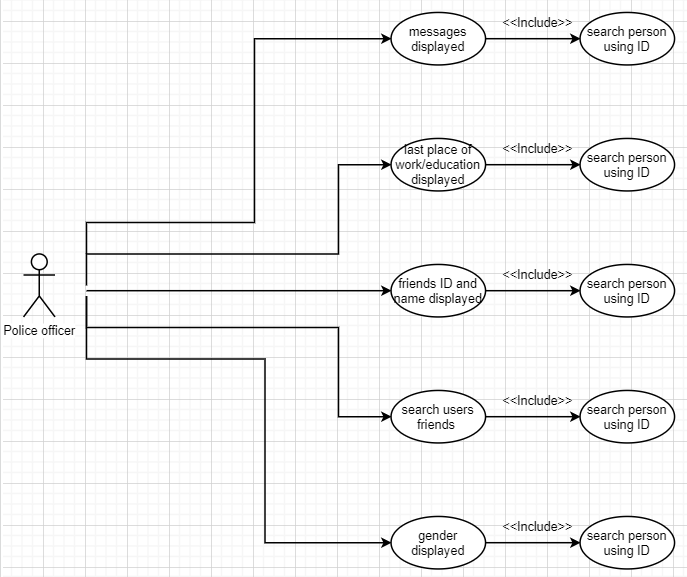
Non-functional requirements:

* Legal
  + Security using a hashing table
  + GDPR – consent to store data
* Ethical
  + Storing peoples data long term
  + Viewing people data
* Usability; interface must conform to user accessibility rules

User stories

* As a police officer I want to enter a userID number and be displayed all their messages so that I can look for potentially incriminating evidence of a perp.
* As a police officer I want to enter a userID and be shown their last place of work/education so that I can follow leads.
* As a police officer I want to enter a userID and be shown their friends ID and names so that I can see if the are friends with other perps.
* As a police officer I want to be able to search through a user’s friends so that I can quickly see if there are associations between perps.
* As a police officer I want to enter a userID and be shown the gender of the perp so that I can match witness statements.

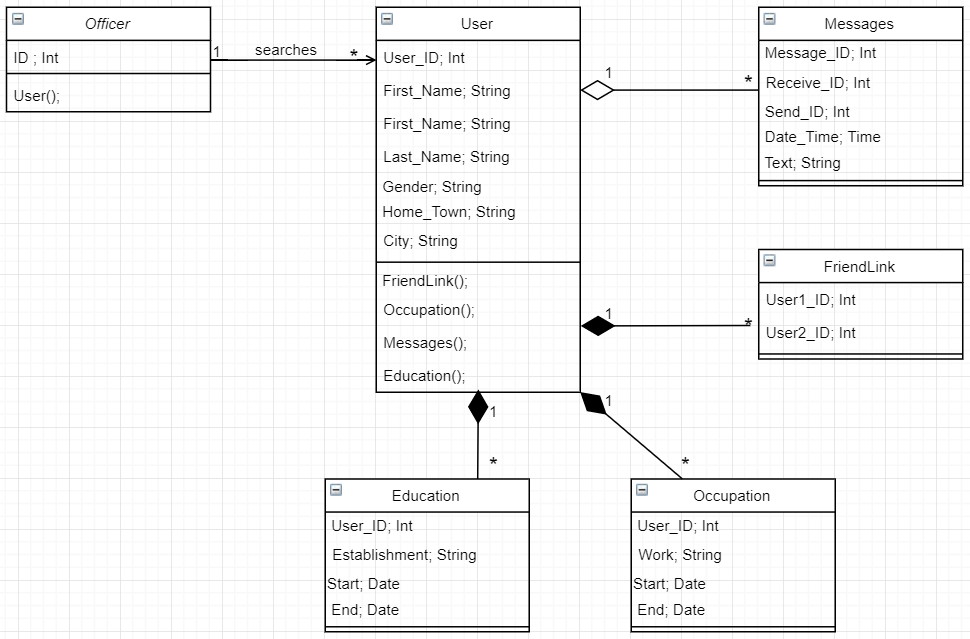
Use Case Diagrams



(Flowchart Maker & Online Diagram Software, 2020)

This use case diagram easily displays the different aspects of the software that the actor can explore; whilst showing the client what the developer grasps as most important functions to include.

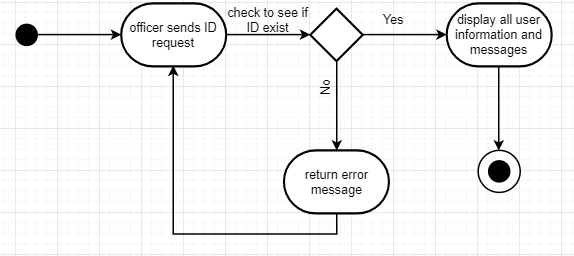
Class Diagram



(Flowchart Maker & Online Diagram Software, 2020)

The class diagram is used to represent the static view of an application. It’s a strong and effective way to structure classes so that they can be translated in to an object orientated languages. It’s also a brilliant way to show the client how their data links together, giving them a higher understanding of their software. It uses both aggregation and composition arrows to convey the dependency on the main class ‘user’.

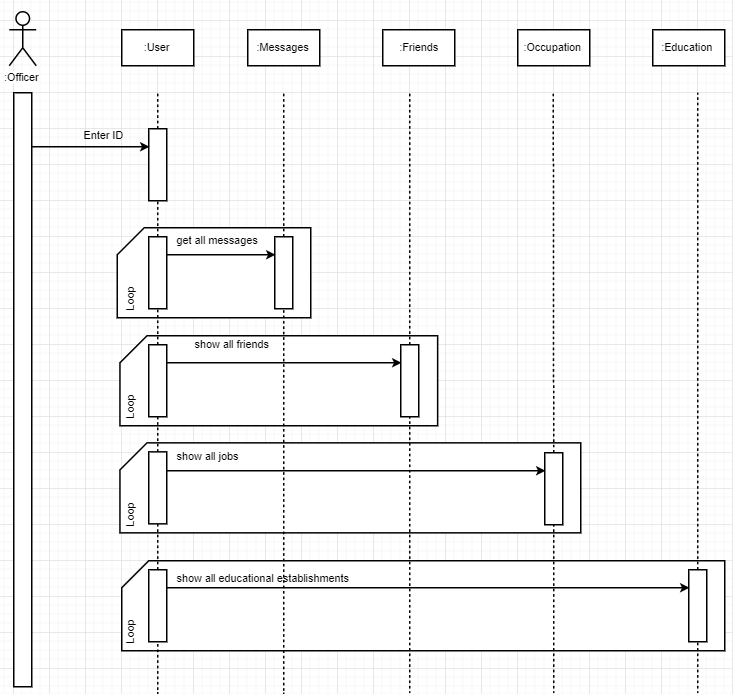
Activity Diagram of information retrieval



(Flowchart Maker & Online Diagram Software, 2020)

I have used an activity diagram to display the flow of information through the application to be made. The diagram moves from the start terminal to the first an action, which is an input from the user. Is then runs through a condition check, allowing the system to come to a termination.

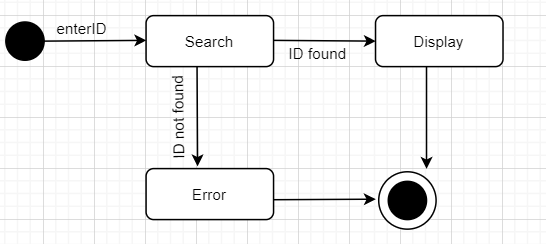
Sequence diagram



(Flowchart Maker & Online Diagram Software, 2020)

A sequence diagram is key to laying out the order of operations within a system. As the diagram descends the time of program execution is increasing; therefore, the first command will be made by the actor, Officer, by inputting the ID of the user they wish to search. After doing this, 4 loops are executed to claim information about the specific user.

State diagram



(Flowchart Maker & Online Diagram Software, 2020)

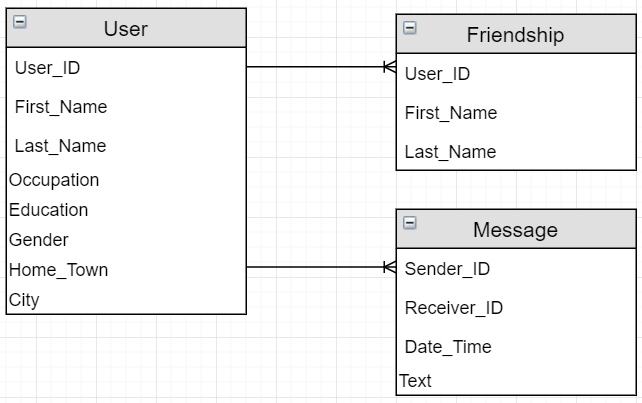
The state diagram is used to emphasise the states in which objects pass through. It is also used to match the systems states to real life states. The comments on each arrow indicate actions, they are used as transition commands between states.

UNF and Corresponding Initial ERD

User

User\_ID

First\_Name

 Last\_Name

Occupation

Education

Gender

HomeTown

City

Friendships

User\_ID

First\_Name

Last\_Name

Messages

Sender\_ID

Receiver\_ID

Date/Time

Text

(Flowchart Maker & Online Diagram Software, 2020)

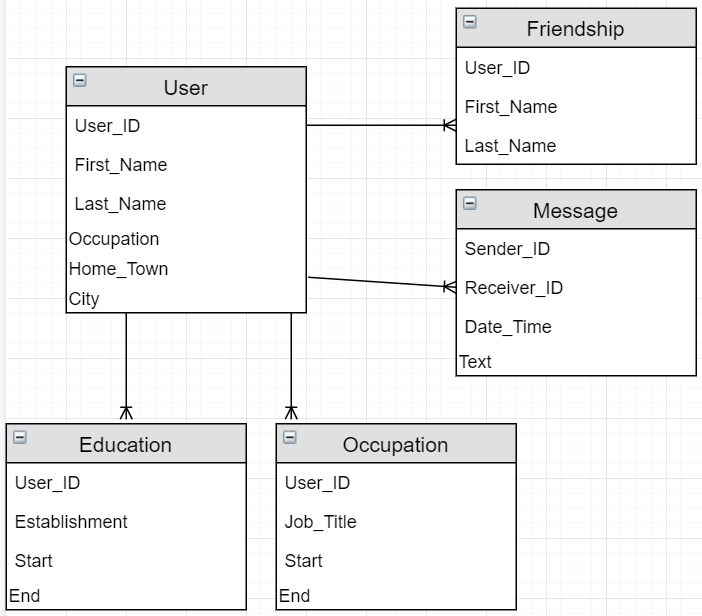
The un-normalised form shows the initial structure of the table, and the ERD shows how they would connect together. Although all relationships are one to many, this ERD and UNF are both in a juvenile state and have a long transformation before they are fully functioning. At this stage, the information laid out is based on the initial table in the brief, table 1.

1NF

User

User\_ID

First\_Name

 Last\_Name

Gender

HomeTown

City

Friendships

User\_ID

First\_Name

Last\_Name

Messages

Sender\_ID

Receiver\_ID

Date/Time

Text

Occupation

User\_ID

Job\_Title

Starting

Ending

Education

User\_ID

Establishment

Starting

(Flowchart Maker & Online Diagram Software, 2020)

Ending

During the first normalisation, it is obvious to see that two new flat files have been added to form our relational database. During the first normalisation, it is key to separate out columns that can contain multiple pieces of data into their own table; this has been done with the education and occupation column because each user could have had more than one job or more than one educational establishment that they have attended. Appropriate columns have been created and named for these new tables. The key from its original table, user, gets propagated into the new tables created.

2NF

User

User\_ID

First\_Name

Last\_Name

Gender

HomeTown

City

Friendships

User\_ID

First\_Name

Last\_Name

Messages

Message\_ID

Sender\_ID

Receiver\_ID

Date/Time

Text

Occupation

User\_ID

Job\_Title

Starting

Ending

Education

User\_ID

Establishment

Starting

Ending

3NF

User

User\_ID

First\_Name

Last\_Name

Gender

HomeTown

City

Friendships

User1\_ID

User2\_ID

Messages

Message\_ID

Sender\_ID

Receiver\_ID

Date/Time

Text

Occupation

User\_ID

Job\_Title

Starting

Ending

Education

User\_ID

Establishment

Starting

Ending

References

Draw.IO. 2020. *Flowchart Maker & Online Diagram Software*. [online] Available at: <https://app.diagrams.net/> [Accessed 20 March 2020].