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

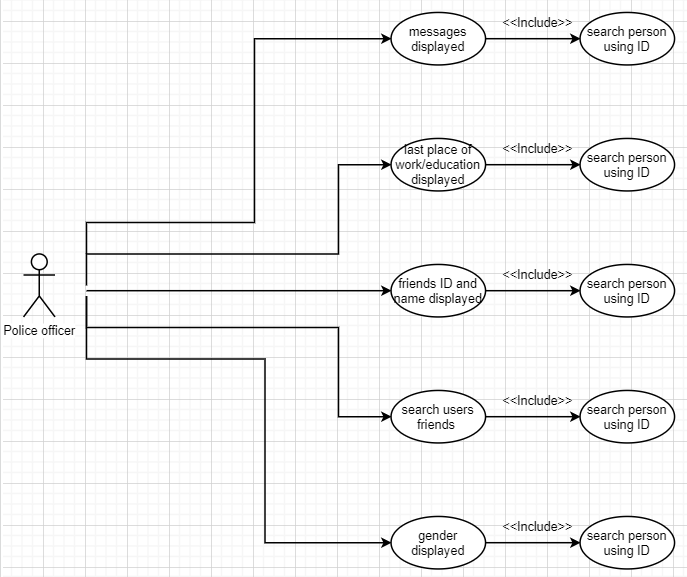
Non-functional requirements:

* 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

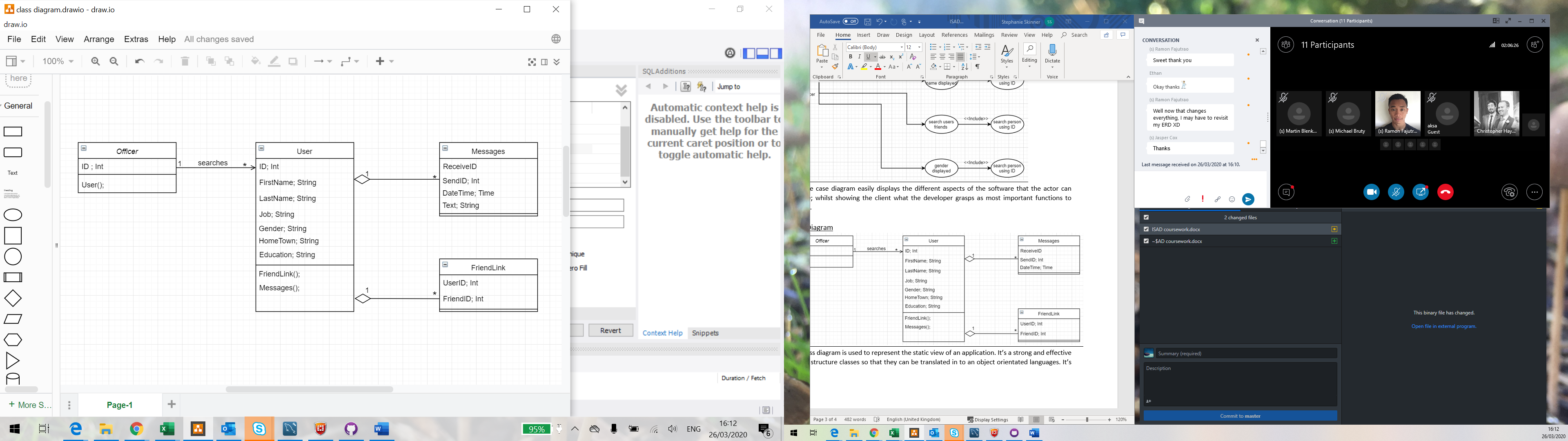
* As a police officer I want to enter a userID number and be displayed all their messages so that I can look for potentially incriminating evidence of a perp.
* As a police officer I want to enter a userID and be shown their last place of work/education so that I can follow leads.
* As a police officer 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



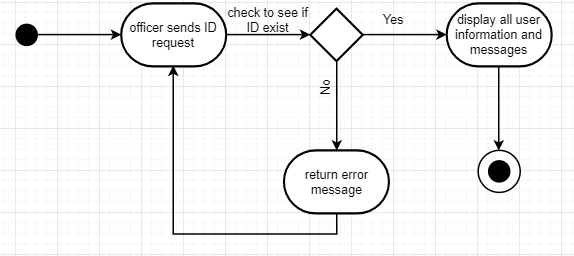
This use case diagram easily displays the different aspects of the software that the actor can explore; whilst showing the client what the developer grasps as most important functions to include.

Class Diagram



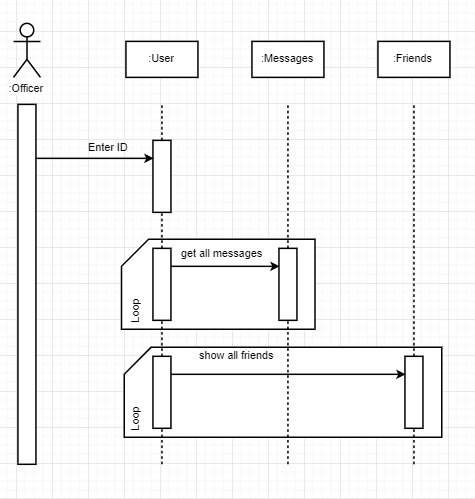
The class diagram is used to represent the static view of an application. It’s a strong and effective way to structure classes so that they can be translated in to an object orientated languages. It’s also a brilliant way to show the client how their data links together, giving them a higher understanding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

Activity Diagram of information retrieval

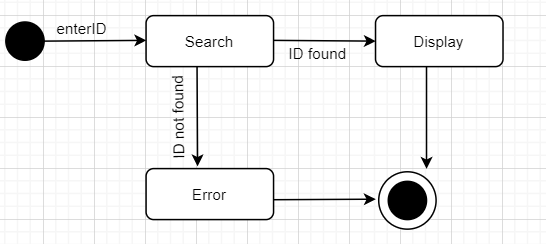


I have used an activity diagram to display the flow of information through the application to be made. The diagram moves from the start terminal to the first an action, which is an input from the user. Is then runs through a condition check, allowing the system to come to a termination.

Sequence diagram



State diagram



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.

UNF

Users

User\_ID

First\_Name

Last\_Name

Occupation

Education

Gender

Home\_Town

Residents

Frienships

User\_ID

First\_Name

Last\_Name

Messages

Sender\_ID

Receiver\_ID

Date/Time

Text

1NF

Users

User\_ID

First\_Name

Last\_Name

Gender

Residents

Frienships

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

Ending

2NF

Users

User\_ID

First\_Name

Last\_Name

Gender

Residents

Frienships

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

Users

User\_ID

First\_Name

Last\_Name

Gender

Residents

Frienships

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