Individual Software Development Project

**Date: 28/02/2017**

**Author: Colin Maher**

**K00186813**

Table of Contents

**Chapter 1: Requirements Elicitation**

**1.1** Software Engineering Synopsis

**1.2** Requirements Elicitation – Application of requirements elicitation to SD3 project.

**Chapter 2: Analysis**

**2.1** Classdiagram

**2.2** Sequence Diagrams

**Chapter 3: Design**

**3.1** Part I System

**3.2** Part II Object

**Chapter 4:** **Implementation**

**Chapter 5: Testing**

**Conclusion**

**Chapter 1: Requirements Elicitation**

* 1. **Software Engineering with Agile Synopsis**

Agile software engineering is “a set of methods and practices based on the values and principles expressed in the Agile Manifesto.” (The Agile Alliance, [www.agilealliance.org](http://www.agilealliance.org)). It is based on iterative development, where requirements and solutions evolve through frequent collaboration between members of the development team and through meetings with the customer. Agile favours adaptive and evolutionary development and it is less rigid than other development methods.

The Agile Manifesto succinctly outlines the goals of Agile development ([www.agilemanifesto.org](http://www.agilemanifesto.org)).

1) Individuals and interactions over processes and tools

The focus of development must be on the people and not the tools used to create the software. Time must be spent interacting with fellow developers and customers to nail down the requirements of the product, before the tools and processes are thought about.

2) Working software over comprehensive documentation

Agile favours an iterative approach to development instead of a Waterfall approach. Under a Waterfall approach the requirements are documented and the customer signs off on them, then the developer builds the product described. This method is completely rigid and doesn’t take into account changing requirements, either from the customer or through problems encountered during development.

3) Customer collaboration over contract negotiation

Agile favours an iterative approach to Development. This has a better chance of delivering a well designed product that satisfies the customer, compared to a product which follows a rigid contract signed early in development.

4) Responding to change over following a plan

Agile favours dealing with problems and making changes as the project progresses instead of sticking to a rigid plan. Projects with complex detailed plans can fail as they are difficult to change.

In conclusion, Agile leads to better software which meets the customer’s requirements and is problem free as problems were dealt with throughout development.

**1.2 Requirements Elicitation – SD3 Group Project**

My group project “Penalty Points Bureau” consisted of many requirements elicitation activities. We had weekly meetings with our customer where we asked questions about the product and over time we built up a picture of the customer’s requirements.

Between meetings we developed use cases, class diagrams, scenarios etc. We created textual use cases which outlined functionality of each component to the customer’s requirements. We created class diagrams which contained what type of information the customer needed to be stored on each entity Eg. Employee, driver, regulation etc. We created scenarios which showed the customer examples of use cases in realistic events. Every week we would show the customer the new and updated use cases, class and other diagrams, he would give feedback and suggest changes.

**Use Cases**

**User Login**

**Participating Actor:** User

**Entry Conditions:** A user wishes to access the system and the User Login function has been invoked.

**Flow of Events:**

**1.** The System responds by presenting blank fields requesting the following information:

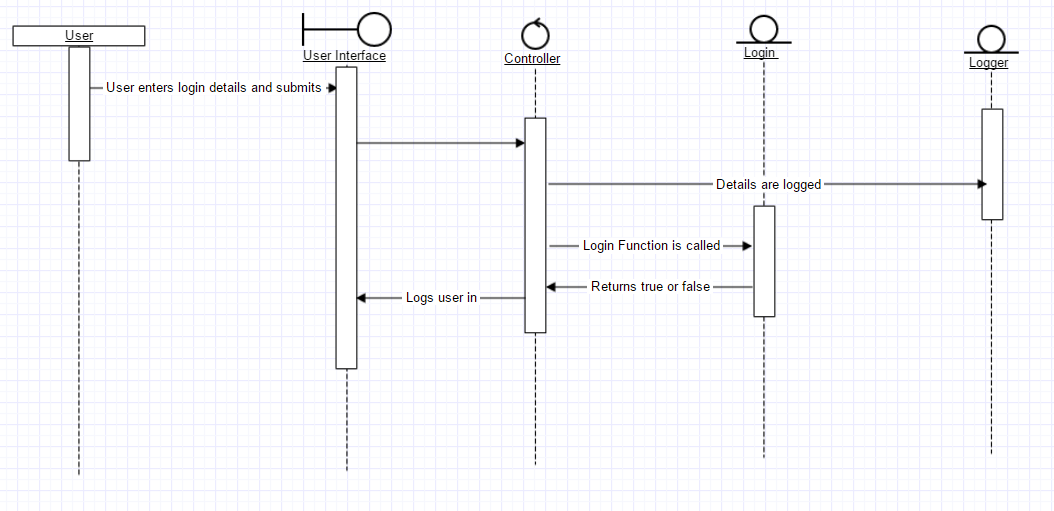
a. User ID

b. Password

The system presents the following options to “Submit” or "Cancel”

**2.** The User enters their User ID and Password and submits the information.

**3.** The system first logs the event, then checks if the User ID has been deactivated and if so the system proceeds to step 7 below.

**4.** Otherwise the system appends the Password to the end of the User ID and passes the concatenated string into an MD5 hashing function to produce a 256bit Cipher. The system users the plaintext User ID to acquire the associated Cipher stored in the database.

**5.** If the Cipher acquired from the user matches the Cipher recorded in the database then the user is assumed to have provided the correct credentials. The system grants the user access to the System. The system retrieves the appropriate Black List and White List Rules from the Access Control system and execution proceeds by invoking the View User Inbox Use Case.

**6.** If the Ciphers do not match then the system logs the event and counts the number of failed attempts and presents the user with a message stating that the user ID or password are incorrect and that they have n attempts remaining. If the count:

a. Is less than three attempts, the system displays a Forgot Password button, which when pressed invokes the Forgot Password Use Case, and the system proceeds to step 1 above.

b. Is equal to three attempt, the system deactivates the user account

**7.** The system notifies the User that the account has been deactivated, and informs the User to contact the administrator to reactivate the account. The system exits when the User closes this Message.

Exit Conditions:

User has gained access to the system and the system has logged the event.

**List Regulations**

**Participating Actor:** Supervisor

**Entry Conditions:** This use case both acts standalone and is invoked by related use cases. The Supervisor must inspect the List of Regulations. The Supervisor invokes the View List of Regulations function.

**Flow of Events:**

1. The System presents a list, which has been prepopulated with the details of existing regulations, and presents blank spaces to enter new records. The system does not display records referring the Regulations that have a Termination Date field which is not null. The system displays the following information for all current and future effective regulations:
   1. A short Definition of the regulation.
   2. The Amended Regulation ID referring to a regulation which has been amended. This field must either be blank or a valid id.
   3. The Publishing Date of the new Regulation.
   4. The Effective Date upon which the new Regulation will take effect.
   5. The Number of Penalty Points upon payment of a fine prior to a court summons.
   6. The Number of Penalty Points upon conviction.
   7. The Amount Payable prior to an appearance in court.
   8. The Amount Payable upon conviction.
   9. A checkbox indicating that the offence requires a Mandatory Court Appearance.
2. The Supervisor inspects the information.

**Exit Conditions:**

The list of regulations have been viewed.

**View User Details**

**Participating Actor:** Admin, Or the User that this account represents

**Entry Conditions:**

An admin wishes to view the recorded details associated with a user.

The admin has invoked the “View User Details” function.

**Flow of Events:**

1. The System displays a read only form containing the following:
   1. The **Organisation ID** which is defaulted to PP for Penalty Points Bureau

All possible choices are:

* + - PP for Penalty Point Bureau
    - LA for Licence Authority
    - GS for Gardaí *Síochána*
  1. The **User ID**.
  2. User Role which is a list containing Admin, Supervisor, Staff, Speed Camera, but is not limited to these choices and can be defined as needed.
  3. The **Date Added**.
  4. The **Date Revoked**.

And If the User is an employee of the Penalty Points Bureau:

* 1. the First Name
  2. the Last Name(s)
  3. Email Address
  4. Home Address
  5. Telephone Number

**Exit Conditions:**

The system has displayed the details recorded in the users account.

**Notify Driver of Summons**

**Participating Actor:** Staff

**Entry Conditions:**

This action is governed by the Data Protection Acts 1988 and 2003. Staff members must sign a Non-Disclosure Agreement before being exposed to any sensitive information.

Members of Staff has received notifications of newly Reported Summons. The member of Staff must issue a Notice of Summons and has invoked the Notify Driver of Summons function.

**Flow of Events:**

1. The System acquires and displays the following information associated with the Offence:
   1. The Reference No of the Offence.
   2. The Time and Date of the incident.
   3. The Location at which the incident occurred.
   4. The Drivers Licence Number.
   5. The Vehicle Registration.
2. The member of Staff presses the Notify Driver button.
3. The System responds by generating a Letter of Notification of Summons which must include the following information:
   1. The Name and Address of the Driver positioned to fit in a windowed envelope.
   2. The Drivers Licence Number.
   3. The Reference No of the Offence, both in numeric form and as a Bar Code.
   4. The Time, Date and Location at which the incident occurred.
   5. The Section and Act (Section 5 of the Road Traffic Act 2002).
   6. The Description of the Regulated Offence.
   7. Date and Location of the Court Hearing.
   8. The Number of Penalty Points to be endorsed as defined by the Regulation.
   9. The Amount Payable and a Payment Deadline of 28 days from the current date.
   10. Instructions on how to make payments at a local Post Office Branch.
   11. Contact Details for the Penalty Points Bureau to facilitate enquiries.
4. The member of Staff presses the Print button.
5. The System responds by printing the Letter of Notification of Summons. The System also records the Notification in the Database. The System returns to the Inbox.

**Exit Conditions:**

The Notification of Summons has been recorded in the Database.

The Letter of Notification of Summons has been printed and is ready to be sent by Registered Post

**Searching a List**

**Participating Actor:** User

**Entry Conditions:**

A user wishes to search a list and show only relevant results.

**Flow of Events:**

1. The System presents a search box which accepts regular expressions.
2. The User enters text in the search box.
3. The System detects that the search box has changed and uses the text to pattern match the fields in the list being presented. Each record in the list that matches the pattern is made visible, all the other records in the list are hidden.
4. The user continues to enter text until the list has been narrowed down considerably.
5. The system presents the final list of all matches.

**Exit Conditions:**

The list has been narrowed down to only relevant entries.

**Scenarios**

**Tom Ryan Logs in to the System**

**Participating Actor:** Member of staff Tom Ryan

**Entry Conditions:** Tom wishes to log in to the system

**Flow of Events:**

**1**. The system presents a blank field requesting:

a. User ID

b. Password

**2.** Tom enters his username and password

**3.** Tom clicks the login button

**4.** The system logs the event, then checks if the User ID has been deactivated.

**5.** The User ID has not been deactivated

**6.** The system appends the Password to the end of the User ID and passes the concatenated string into an MD5 hashing function Cipher stored in the database

**7.** The cipher acquired from Tom’s username and password match the cipher recorded in the database.

**8.** The system retrieves the appropriate Black List and White List Rules from the Access Control system. Execution proceeds by invoking the View User Inbox Use Case.

**Exit Conditions:**

**1.** Tom has gained access to the system and the system has logged the event

**Tom Ryan Notifies Driver of Summons**

**Participating Actor:** Member of staff Tom Ryan

**Entry Conditions:** Tom has received a notification of a newly reported summons. Tom must issue a Notice of Summons and has invoked the Notify Driver of Summons function.

**Flow of Events:**

**1.** The system acquires and displays the following information associated with the offence

a. The Reference No of the offence

b. The Time and Date of the incident

c. The Location at which the incident occurred

d. The Drivers Licence Number

e. The Vehicle Registration

**2.** Tom presses the Notify Driver Button

**3.** The system responds by generating a Letter of Notification of Summons which must include the following information:

a. The Name and Address of the Driver positioned to fit in a windowed envelope.

b. The Drivers Licence Number.

c. The Reference No of the Offence, both in numeric form and as a Bar Code.

d. The Time, Date and Location at which the incident occurred.

e. The Section and Act (Section 5 of the Road Traffic Act 2002).

f. The Description of the Regulated Offence.

g. Date and Location of the Court Hearing.

h. The Number of Penalty Points to be endorsed as defined by the Regulation.

i. The Amount Payable and a Payment Deadline of 28 days from the current date.

j. Instructions on how to make payments at a local Post Office Branch.

k. Contact Details for the Penalty Points Bureau to facilitate enquiries.

**4.** Tom presses the Print button

**5.** The System responds by printing the Letter of Notification of Summons. The System also

records the Notification in the Database. The System returns to the Inbox.

**Exit Conditions:**

**1.** The Notification of Summons has been recorded to the Database

**2.** The Letter of Notification of Summons has been printed and Tom can now send it by Registered Post

**Jimmy Reilly Searches a List**

**Participating Actor:** Member of staff Jimmy Reilly

**Entry Conditions:** Tom wishes to log in to the system

**Flow of Events:**

**1**. Tom enters a search term into the search box

**2.** Tom clicks the login button

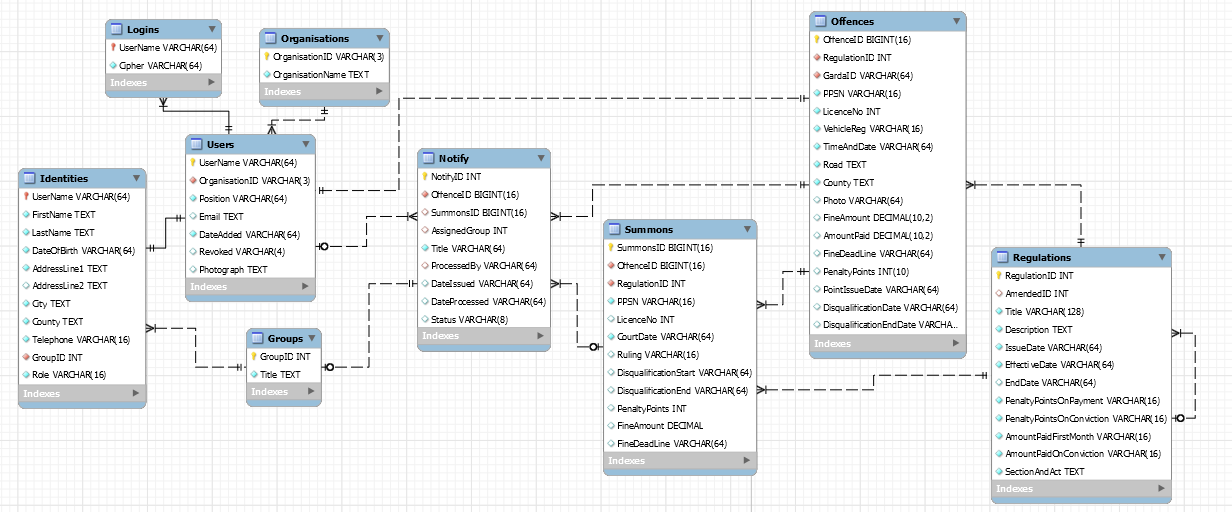
**3.** The system filters the list as Jimmy types

**Exit Conditions:**

Jimmy finds what he is looking for once the list is narrowed down enough

**Chapter 2: Analysis**

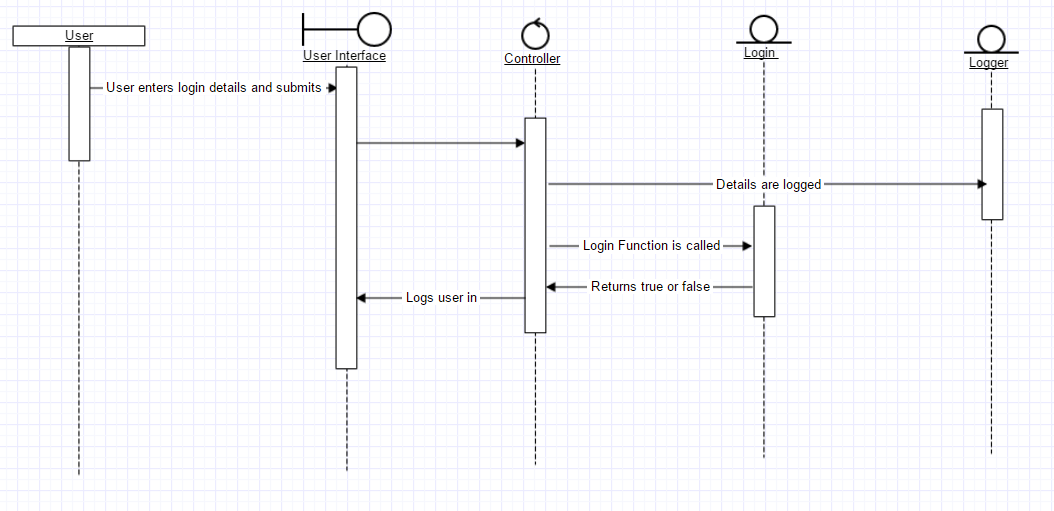
**2.1 Class Diagram**



This class diagram shows the entities in the system along with all their attributes and relationships.

**2.2 Sequence Diagrams**

**Login Sequence Diagram**



**Chapter 3: Design**

**Chapter 4: Implementation**

**Chapter 5: Testing**

Conclusion

In conclusion my group project thought me a lot about requirements elicitation. It thought me how to deal with customer meetings. I learned about keeping minutes and presenting ideas to the customer. I got experience in working with a team.