

# Declaration Cover Sheet for Project Submission

## SECTION 1 *Student to complete*

<b>Name:</b>	Niamh Griffin
<b>Student ID:</b>	x13322636
<b>Supervisor:</b>	Sara Kadry

## SECTION 2 *Confirmation of Authorship*

*The acceptance of your work is subject to your signature on the following declaration:*

I confirm that I have read the College statement on plagiarism (summarised overleaf and printed in full in the Student Handbook) and that the work I have submitted for assessment is entirely my own work.

Signature: \_\_\_\_\_ Date: \_\_\_\_\_

Niamh Griffin

National College of Ireland  
BSc (Hons) in Computing – Cyber Security  
2016/2017

Niamh Griffin  
x13322636  
niamh.griffin1995@gmail.ie

eClip

Final Software Project Report

## Table of Contents

1	Executive Summary .....	5
2	Introduction .....	6
2.1	Glossary .....	6
2.2	Background .....	7
2.3	Aims .....	8
2.4	Technologies .....	9
2.4.1	Security Technologies .....	9
3	System.....	11
3.1	Security .....	11
3.1.1	Data Encryption.....	11
3.1.2	Data Confidentiality .....	11
3.1.3	Data Integrity.....	12
3.2	Functional Requirements.....	13
3.2.1	Use Case Diagram .....	13
3.2.2	Requirement 1.0: User Registration .....	14
3.2.3	Requirement 1.1: User Details Encrypted .....	16
3.2.4	Requirement 2.0: User Login .....	17
3.2.5	Requirement 3.0: Highlight Snippet.....	19
3.2.6	Requirement 3.1.1: Add Experiment .....	20
3.2.7	Requirement 3.1.3: Show Experiments .....	21
3.2.8	Requirement 4.0: User Log Out .....	22
3.3	Non-Functional Requirements.....	24
3.3.1	Performance/Response time requirement.....	24
3.3.2	Availability requirement .....	24
3.3.3	Reliability requirement.....	24
3.3.4	Maintainability requirement .....	24
3.3.5	Portability requirement .....	25
3.4	Design and Architecture .....	25
3.5	Implementation.....	26
3.5.1	Registration .....	26

3.5.2	Login .....	27
3.5.3	Highlight Snippet .....	28
3.5.4	Add Experiment.....	28
3.6	Graphical User Interface Layout.....	29
4	Future development.....	32
5	References .....	35
6	Appendix.....	36
6.1	Project Proposal .....	36
6.1	Supervisor Meetings.....	40

# 1 Executive Summary

eClip is a rich client Java application developed in NetBeans which will aid large scale corporate development teams, student developers and budding developers in the implementation process of a software project. It will be helpful when referencing some code from a certain project, and to look back in future developments where it may be used again by saving experiments or code snippets of their projects.

The best way to describe eClip is to use a scenario we have all had to go through in school. Think of sitting down to do a maths exam, all your answers will be in the front of the exam script along with the question, and all your rough work will be in the back of the paper for your own thought processing and reference. When you finish the exam only your answers get graded, and you'll never be able to use your rough work again. eClip allows you to go back and see your previous rough work noted while developing your software project.

eClip combines all workings together in one place with a major focus on security. Some security features in eClip are authentication of users, access controls to the source code, encryption to the database and sharing of experiments once approved by the experiment owner.

## 2 Introduction

### 2.1 Glossary

IDE	Integrated Development Environment
User	Someone interacting with eClip – mainly developers
Admin	Administrator
C.I.A triangle	Confidentially, Integrity and Availability
CMS	Content Management System
API's	Application Programming Interface

## 2.2 Background

The idea of eClip was thought up during a six-month internship in PayPal. I developed an internal portal with information about my placement team. For most of the web pages, it was required to hard code HTML, PHP, JavaScript, jQuery and CSS into the Joomla framework. During the development of the portal, there were functionalities which required more research and experimentation. One example was to develop a system for managers in PayPal to keep track of what office time zone was on the phone shift at any given time in the year.

As developing with time zones can be very tricky, it is best to work off existing API's, so as not to reinvent the working wheel. This led to trialling out suggestions, subsequently deleting those lines of code and starting the search for another method. It was difficult to keep record of trialled experiments if ever I needed to reference back to them or where they came from. This led to saving all ideas found online onto the bookmarks section of my browser.

Seeing the lack of progress, steered me to search for a more efficient method for version control, combined with note taking of code, ways to revert back to older code versions and search through notes by programming language specific tags. After much online research and talking with developers in my workplace, the search turned up empty. This tool did not exist. Yes, you could use a basic text editor to create a file holding all your notes and code snippets but you would have to be seriously disciplined.

Thoughts pondered about developing this tool as part of my final year software project. On the ground requirements analysis began by asking as many developers as would listen about the theoretical idea and technologies to develop the idea. Their guidance and experience proved invaluable.

Initially the idea was to build a text editor from scratch to implement the functionality. Many of the developers thought it unnecessary to reinvent the working wheel by having to become familiar with a new text editor.

## **2.3 Aims**

The objective of my final year project is to build software which aids real developers, including myself, in the implementation process of software projects.

As a user of eClip myself, I am eager to make eClip very simple and become an instinct reaction for users when organising a software project.

eClip will store information about its users and possibly their intellectual property inside a snippet, therefore securing the data passed through the application is vital. Data classified as sensitive will be encrypted before storing in the database.



## **2.4 Technologies**

As I specialised in cyber security for my final year, I was eager to make eClip secure for all end users. There are two main aspects of how I integrated security with eClip:

1. Access Controls: The source code itself has permissions. eClip contains some private classes to stop potential entry points for an attacker. Classes are also set to final so attackers cannot extend a class and edit some methods in the project.
2. For the experiments the user will create with snippets of code, they will be encrypted before sending to a MySQL database, ensuring the confidentiality and integrity of the user's work.

eClip is programmed using Java as that is my strongest programming language and will allow for functionality that would otherwise be restrictive. I have learned secure coding practises and combined them with my existing knowledge.

After each functionality was developed along the way, I used the agile development lifecycle to check for any errors in the system of eClip.

### **2.4.1 Security Technologies**

eClip was originally intended to be fully available with online capabilities for users to access their experiments from anywhere with an internet connection. These features are not part of eClip at this point, as the time scope of the implementation stage would not have been sufficient to thoroughly learn about server security configurations along with secure coding practises. It was decided to keep eClip offline to minimise security threats to users in the beta version of eClip. As can be seen in the Future Developments section, eClip has only just begun.

A desktop application can still pose security to its database. If an attacker gained access to the database with no access controls, they would have full reign over all information stored.

*“The primary purpose of encryption is to protect the confidentiality of digital data stored on computer systems or transmitted via the Internet or other computer networks.”*

(Rouse, 2014)

The practise of encryption is the modification of information to another form, called cipher text, which cannot be easily understood or reversed, only by authorised personnel.

## **3 System**

### **3.1 Security**

eClip must interact with its hosted database to grant users access to the functionality of the application. The information stored in the database must be kept secure by encrypting the data held inside.

The research target is software developers, who may work as part of a large-scale development team or as a budding developer, so we feel that security is imperative to ensure the intellectual property of our users is maintained.

#### **3.1.1 Data Encryption**

eClip utilises jBCrypt (Miller, 2015) to encrypt data being passed from the desktop application to the hosted MySQL database.

jBCrypt is a Java language implementation of the Blowfish password hashing algorithm (Miller, 2015) designed by Bruce Schneier in 1993. Blowfish is a symmetric block cipher that can take a variable length from 32 to 448 bits.

(Schneier, n.d.)

jBCrypt was designed to disinterest attackers by raising the cost of password cracking as the computation cost of the algorithm varies, so it can be increased as computers get faster.

#### **3.1.2 Data Confidentiality**

The principle of data confidentiality means granting least privileges. This principle states that access to information should be granted only where essential to perform functions necessary for that user.

*“Authentication, and Authorization are principles which are achieved through various access and privacy controls that support Confidentiality.”*

(Mehta, 2016)

### **3.1.3 Data Integrity**

Data integrity ensures that data is not tampered with during the transmission from source to destination. Information should not only be sent over a secure connection but also never sent in plain text.

A practise used in digital forensics to ensure the integrity of examined data is to compare if MD5 values match from before the data was examined and afterwards.

## 3.2 Functional Requirements

### 3.2.1 Use Case Diagram

This use case diagram depicts how the desktop application will interact with the users of the system and the overall behaviour of the system.

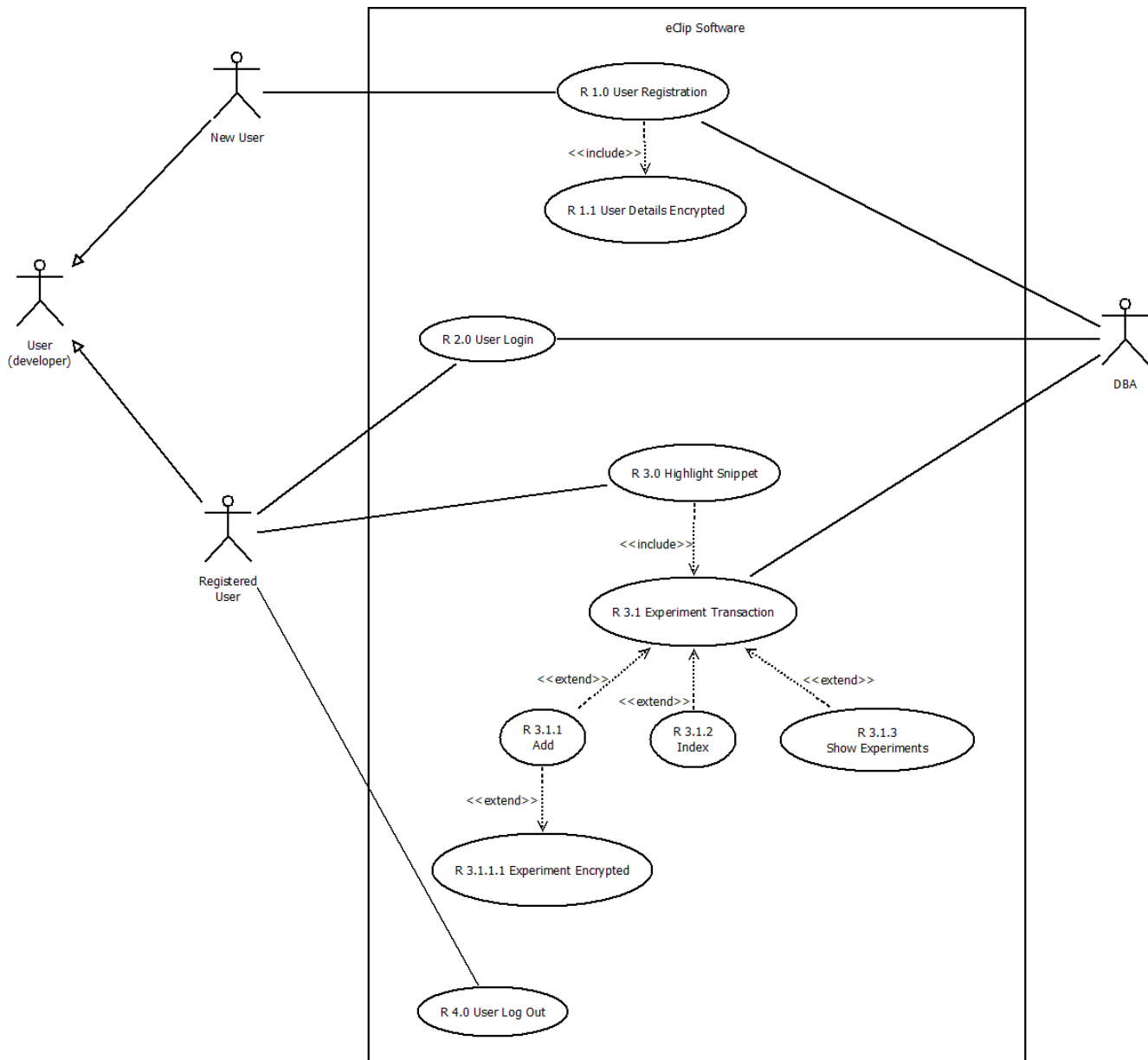


Figure 1 - Use Case Diagram. This use case diagram shows the current functionality of eClip and how each functionality interacts with the actors apart of eClip.

## 3.2.2 Requirement 1.0: User Registration

### Description & Priority

The user can register an account on the eClip desktop application. The user must provide their first name, last name, email, username and a strong password. The users account information will be stored in the hosted MySQL database. This requirement is essential as a user needs to create an account to be granted access to the eClip functionality.

### Use Case ID

R 1.0

### Scope

The scope of this use case is to create a unique account for each user of eClip.

### Description

This use case describes how the user registers a unique account by pressing the signup button.

### Flow Description

### Precondition

The user must enter all five fields in the form. The five fields are: first name, last name, email, username and password.

### Activation

This use case starts when the user clicks signup button.

### Main flow

1. The user launches the eClip desktop application.
2. The system displays a screen with signup and login forms.
3. The user fills in all five fields in the form.
4. The user clicks the signup button.

5. The system grabs all five inputs from the user's signup form.
6. The system creates a registration date variable with the current UTC date.
7. The system encrypts the password of the user.
8. The system saves the first four inputs from the user to the database along with the encrypted password, registration date and encryption string.
9. The user is prompted to login to their account now.

### **Exceptional flow**

E1: <Five fields on form match placeholder text>

1. The user does not add in their details into the form.
2. The system displays an error message to the user that they must enter their details.
3. The use case continues at position 2 of the main flow.

E2: <Any of the five fields are empty>

1. The user does not fill in all the form.
2. The system displays an error message to the user that they must fill in the whole form.
3. The use case continues at position 2 of the main flow.

E3: <Email is not valid>

1. The user does not enter a valid email.
2. The system displays an error message to the user to enter a valid email address.
3. The use case continues at position 2 of the main flow.

E4: <Username or email exists in eClip>

1. The user enters a username or email that is already in existence in eClip.
2. The system displays an error message to the user that the username or email already exists in eClip.
3. The use case continues at position 2 of the main flow.

### **Termination**

The system prompts the user to login to their account.

### **Post condition**

On successful completion, an account is created for the user and then the user will be prompted to login. If a failure occurs the user will be prompted to create an account correctly.

## **3.2.3 Requirement 1.1: User Details Encrypted**

### **Description & Priority**

The user's password is encrypted before setting up their account in the database. This requirement is also essential to protect the confidentiality and integrity of the user.

### **Use Case ID**

R 1.1

### **Scope**

The scope of this use case is to encrypt the user's password so as not to be vulnerable to an attack if the eClip database was breached.

### **Description**

This use case describes the process of how the user's password is encrypted during the signup process.

### **Flow Description**

### **Precondition**

Use case R 1.0 must have gotten to its main step 8 with no errors for this use case to proceed.

### **Activation**

Use case R 1.0 must have gotten to its main step 8.



### **Main flow**

1. The system uses the jBCrypt [4] class methods to generate an encryption string of 12 characters long.
2. A new variable is created which stores the plain text password entered by the user which is combined with the encryption string.

### **Termination**

Use case R 1.0 continues its main flow as step 9.

### **Post condition**

On successful completion, the plain text password the user entered will never be seen in plain text in the eClip system.

## **3.2.4 Requirement 2.0: User Login**

### **Description & Priority**

Once a user has registered an account with eClip, the user will be able to login to their account and begin using eClip's functionality. The users login details are checked against the users table in the database and access is either granted or declined.

### **Use Case ID**

R 2.0

### **Scope**

The scope of this use case is to verify the user's credentials before granting access to eClip.

### **Description**

This use case describes how the user must enter their unique credentials and click the login button.

### **Flow Description**

## **Precondition**

The user must enter their email and password into the login form.

## **Activation**

This use case starts when the user clicks login button.

## **Main flow**

1. The user launches the eClip desktop application.
2. The system displays a screen with signup and login forms.
3. The user fills in their email and password unique to their account into the form.
4. The user clicks the login button.
5. The system grabs both inputs from the user's login form.
6. The system queries the database.
7. The system grants the user access to the eClip editor page.

## **Exceptional flow**

E1: <Email is not contained in the database>

1. The email address entered by the user is not registered with eClip.
2. The system displays an error message to the user that no account exists with the credentials entered.
3. The use case continues at position 2 of the main flow.

E2: <Password does not match up with the email entered>

1. The password entered does not match up with the account the user is trying to access.
2. The system displays an error message to the user that the username or password entered is incorrect.
3. The use case continues at position 2 of the main flow.

## **Termination**

The system grants the user access to the editor page.

## **Post condition**

On successful completion, the user can access eClip for the first time or begin to add code snippets to their account. If a failure occurs the user must enter their credentials correctly or signup to eClip.

## 3.2.5 Requirement 3.0: Highlight Snippet

### Description & Priority

Once the user is logged into their account, they now have access to eClip's main functionality.

#### Use Case ID

R 3.0

#### Scope

The scope of this use case is to allow a registered user to add an experiment or code snippet to a database along with a name, tags and notes.

#### Description

This use case describes how the user can add an experiment to eClip.

#### Flow Description

#### Precondition

1. The user must be registered.
2. The user must be logged in.

#### Activation

This use case starts when user highlights some text in the editor area and clicks on the add experiment button.

#### Main flow

1. The user can highlight some text in the editor.
2. The user will click the add experiment button.
3. The highlighted or selected text from the editor will be brought to a text area on the Add Experiment page.

### **Exceptional flow**

E1: <Highlighted is empty>

1. Nothing is highlighted by the user in the editor.
2. The system displays an error message to the user notifying them to highlight
3. The use case will not have activated yet.

### **Termination**

The system grants the user access to the Add Experiment page.

## **3.2.6 Requirement 3.1.1: Add Experiment**

### **Description & Priority**

The main functionality will allow the user to note a piece of code from their editor, keeping this snippet of code saved for functionality they may want to use in the future. This is one of the core functionalities of eClip, making its priority high.

### **Use Case ID**

R 3.1.1

### **Scope**

The scope of this use case is to allow the user to create an experiment and can refer back to the experiment for future use.

### **Flow Description**

### **Precondition**

The user must be logged in.

### **Activation**

The user must have highlighted the text they want to add and clicked the Add Experiment button on the Editor page.

### **Main flow**

1. The snippet field will have been populated from the Editor page.
2. The user should fill out the experiment form with a name, tags and notes.
3. The user should click the save button.
4. The system will grab all four fields from the form.
5. The system will create two variables called ExpCreated and ExpModified which are set to the current UTC date.
6. The system then generates an encryption string and encrypts the snippet.
7. The system saves the name, encrypted snippet, tags and notes to the database along with the user identification number, ExpCreated, ExpModified and encryption string.

### **Exceptional flow**

E1: <Name, snippet and tags fields are empty>

1. The name, snippet and tags fields are left empty by the user.
2. The system displays an error message to the user notifying them to enter all three fields.
3. The use case continues at position 2 of the main flow.

E2: <Public experiment>

1. The user can choose to not encrypt their snippet.
2. The use case skip position 6 of the main flow.

### **Termination**

The system will alert the user that this experiment was added.

### **Post condition**

The system will allow the user to continue working on their project.

## **3.2.7 Requirement 3.1.3: Show Experiments**

### **Description & Priority**

This functionality will allow the user to view all their saved experiments. This is one of the core functionalities of eClip, making its priority high.

### **Use Case ID**

R 3.1.3

## **Scope**

The scope of this use case is to allow the user view their experiments in one place for future use.

## **Flow Description**

### **Precondition**

The user must be logged in.

### **Activation**

The user should click the Show Experiments button from the Editor and Add Experiment pages.

### **Main flow**

1. The page will show the user a brief of a few experiments.
2. The user can click on the snippet of an experiment where they can view all the information saved with that experiment.

### **Exceptional flow**

E1: <No experiments saved>

4. There are no experiments saved by the user.
5. The system tells the user they have no experiments saved.
6. The use case continues at position 1 of the main flow.

### **Termination**

The user can direct onto another page as they wish.

## **3.2.8 Requirement 4.0: User Log Out**

### **Description & Priority**

When a user has registered an account with eClip and has logged into eClip, the user will be able to logout at any time they wish. The user's log in details are permanently stored in the database for future access.

### **Use Case ID**

R 4.0

## **Scope**

The scope of this use case is to allow the user to exit their account.

## **Description**

This use case describes how the user logs out of their account by pressing the log out button.

## **Flow Description**

### **Precondition**

1. User has an existing account.
2. User is logged in.

### **Activation**

This use case starts when a user clicks the logout button.

### **Main flow**

1. The user decides to logout.
2. The system logs the user out of their account.

### **Termination**

The system logs the user out of their account.

### **Post condition**

On successful completion, the user is logged out and the start screen is displayed.

## **3.3 Non-Functional Requirements**

### **3.3.1 Performance/Response time requirement**

The eClip application should be fast and responsive. The application should switch between displays within minimum time required. Login and adding experiments should take no more than 5 seconds once all the required fields are filled in. This is vital to the success of eClip as it is intended for users to take advantage of the functionality throughout their working day or for users currently developing software.

eClip will however be dependent on how fast the user's computer can process Java applications and how fast the hosted database is.

### **3.3.2 Availability requirement**

The application will be available 24/7 so that the user can access all functionality of the application no matter the time. The application should be connected to the internet in order for the application to interact with the database.

### **3.3.3 Reliability requirement**

To ensure to security and efficient of eClip and Java should always be kept up to date with new releases.

### **3.3.4 Maintainability requirement**

As the threat of black hat hackers grows, eClip developers will have to maintain its security with software patches and new techniques to maintain the confidentiality of eClip users.

The latest cryptographic practices will also be developed in the coming months and will be maintained for user's privacy.



### 3.3.5 Portability requirement

Future developments include accessing eClip experiments while not being able to access the user's laptop/desktop. This will be done on eClip's website where the user can log into their account and view a dashboard of their experiments.

## 3.4 Design and Architecture

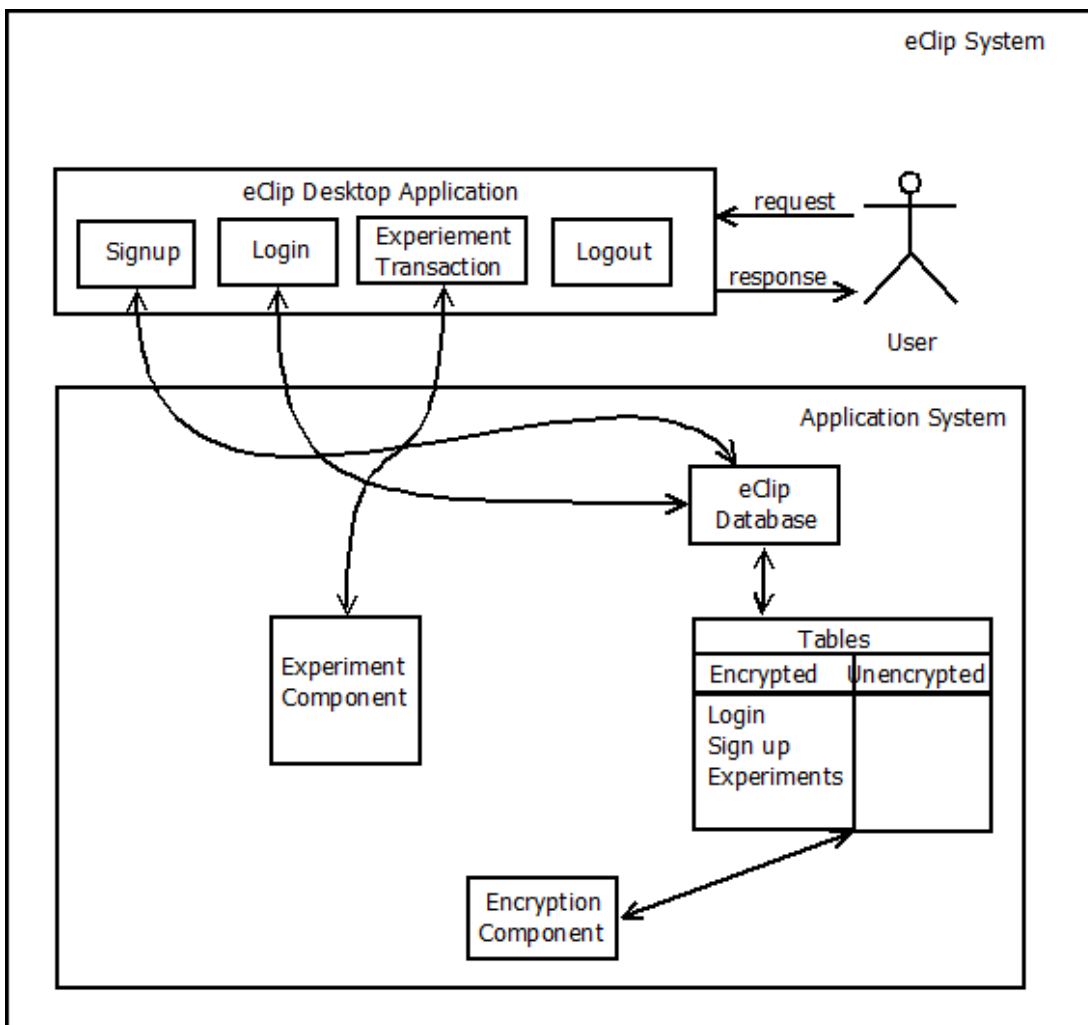


Figure 2 - System Architecture

## 3.5 Implementation

### 3.5.1 Registration

```
try {
    // get first name, last name, email, username, password from textfields
    String FName = fnameSignupTf.getText();
    String LName = lnameSignupTf.getText();
    String Email = emailSignupTf.getText();
    String Username = usernameSignupTf.getText();
    Plain_Password = pswdSignupTf.getText();

    // if all fields are filled in
    if ((!FName.matches("first name")) && (!LName.matches("last name")) && (!Email.matches("email")) && (!Username.matches("username"))) {
        if ((!FName.trim().isEmpty()) && (!LName.trim().isEmpty()) && (!Email.trim().isEmpty()) && (!Username.trim().isEmpty()) &&
            (!Plain_Password.trim().isEmpty())) {
            // if the email entered is valid
            if (isEmailValid(Email) == true) {
                // if the username and email entered is not already in the db
                if ((CheckUsernameExists(Username) == false) && (CheckEmailExists(Email) == false)) {

                    // Creating registration date variable
                    Timestamp RegDate = new Timestamp(System.currentTimeMillis());

                    // Hashing password
                    String Salt = BCrypt.gensalt(12);
                    Pswd_Hash = BCrypt.hashpw(Plain_Password, Salt);

                    String sql = "INSERT INTO u3XY3Mq_users (fname, lname, email, username, password, reg_date, salt)"
                        + "VALUES ('" + FName + "', '" + LName + "', '" + Email + "', '" + Username + "', '" + Pswd_Hash + "', '" + RegDate + "', '"
                        + "'" + Salt + "')";

                    pst = con.prepareStatement(sql);
                    pst.execute();
                    JOptionPane.showMessageDialog(null, "You have been registered. Please proceed to log into your account.");
                }
            }
        }
    }
}
```

Figure 3 - Registration Method

## 3.5.2 Login

To ensure the identity of the user is matched to the details they provide when signing up for eClip, the user must go through some checks before they are provided with access to the eClip application and before accessing their eClip account.

```
try {
    // get password from textfield
    String Email = emailLoginTf.getText();
    String InputPassword = pswdLoginTf.getText();

    String sql = "SELECT password FROM u3XY3Mq_users WHERE email=?";

    pst = con.prepareStatement(sql);
    pst.setString(1, Email);
    rs = pst.executeQuery();

    if (rs.next()) {
        // set the output of the password column in db to
        Pswd_Hash = rs.getString(1);

        // use BCrypt to check if the password is correct
        if ((CheckPassword(InputPassword, Pswd_Hash)) == true) {
            // Show the editor GUI to allow the user to add a code snippet
            EditorGUI myEditorGUI = new EditorGUI();
            myEditorGUI.setVisible(true);
            myEditorGUI.setEmail(Email);
            dispose();
        } else if ((CheckPassword(InputPassword, Pswd_Hash)) == false) {
            JOptionPane.showMessageDialog(null, "The username or password entered is incorrect, please try logging in again.");
        }
    } else if (!rs.next()) {
        JOptionPane.showMessageDialog(null, "The username or password entered is incorrect, please try logging in again.");
    }
} catch (SQLException e) {
    System.out.println(e);
} finally {
    try {
        rs.close();
    }
}
```

Figure 4 - Login Code

### 3.5.3 Highlight Snippet

```
if (editorTa.getSelectedText() != null) {
    try {

        String highlighted = editorTa.getSelectedText();

        AddExperimentGUI myExperimentGUI = new AddExperimentGUI();
        myExperimentGUI.setHighlighted(highlighted);
        myExperimentGUI.setVisible(true);
        dispose();
    } catch (HeadlessException e) {
        System.out.println(e);
    }
} else {
    JOptionPane.showMessageDialog(null, "Please select some text to be added into an experiment.");
}
```

Figure 5 - Highlighting Snippet Code

### 3.5.4 Add Experiment

```
try {
    // get Name, Snippet, Tags, Notes from the JTextField and JTextArea
    String Name = nameTf.getText();
    String Plain_Snippet = snippetTa.getText();
    String Tags = emailTf.getText();
    String Notes = notesTf.getText();
    Timestamp ExpCreated = new Timestamp(System.currentTimeMillis());
    Timestamp ExpModified = new Timestamp(System.currentTimeMillis());

    // Name, Snippet and Tags must be entered
    if ((!Name.trim().isEmpty()) && (!Plain_Snippet.trim().isEmpty()) && (!Tags.trim().isEmpty())) {
        // If private is selected
        if (privateExpBtn.isSelected()) {
            // Hashing snippet
            String Salt = BCrypt.gensalt();
            String Snippet_Hash = BCrypt.hashpw(Plain_Snippet, Salt);

            // Save Inputs to DB
            String sql = "INSERT INTO k09mFrH_experiments (U_Id, name, snippet, tags, notes, date_created, date_modified, salt)"
                + "VALUES ('" + getUserId() + "', '" + Name + "', '" + Snippet_Hash + "', '" + Tags + "', '" + Notes + "', '" + ExpCreated + "', '"
                + ExpModified + "', '" + Salt + "')";

            pst = con.prepareStatement(sql);
            pst.execute();
            JOptionPane.showMessageDialog(null, "Your Experiment has been saved!");
        }
    }
}
```

Figure 6 - Add Experiment Code

### 3.6 Graphical User Interface Layout

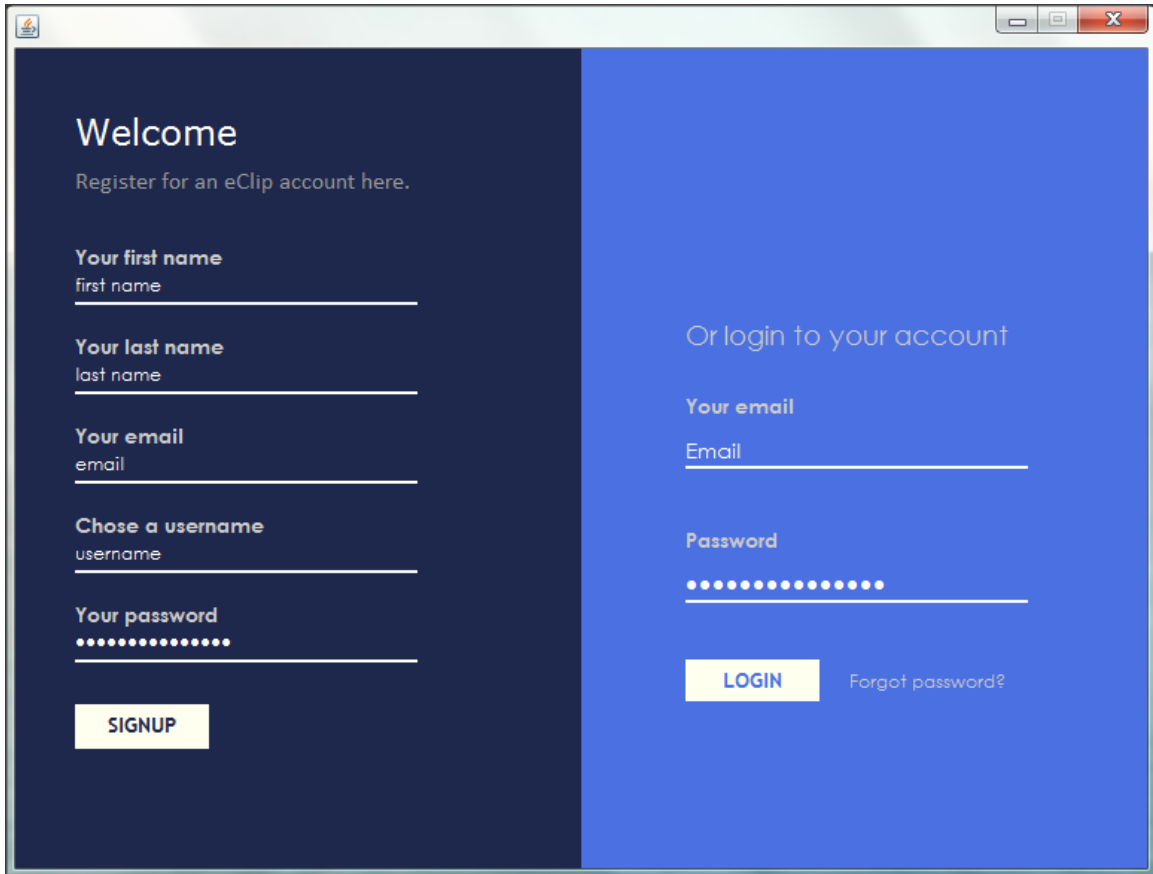
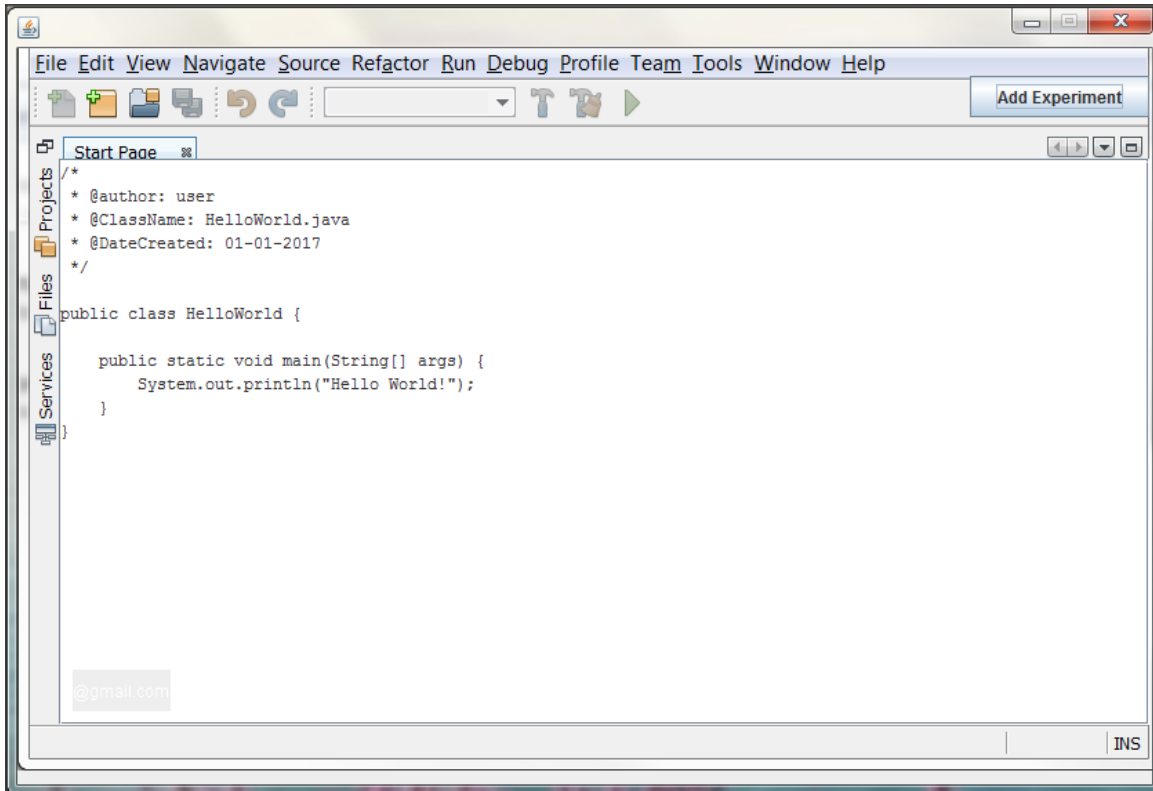


Figure 7 - Home page for login and signup



**Figure 8 - Editor page where the user can develop as they like and add experiments as they please.**

\* Name:

\* Snippet: 

```
public static void main(String[] args) {  
    System.out.p  
rintln("Hello World!");  
}
```

\* Tags:

Notes:

Public Experiment  Private Experiment

\* Required Fields

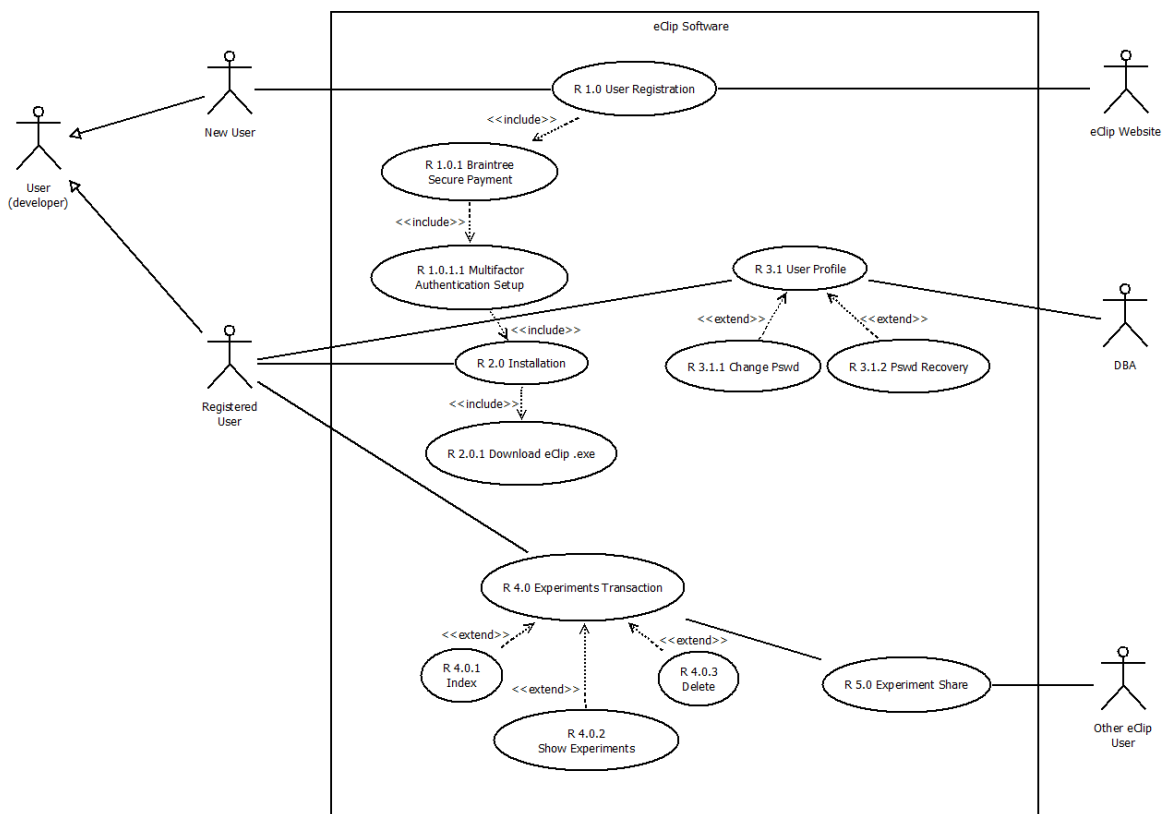
[View All Experiments](#)

Figure 9 - Adding Experiment form

## 4 Future development

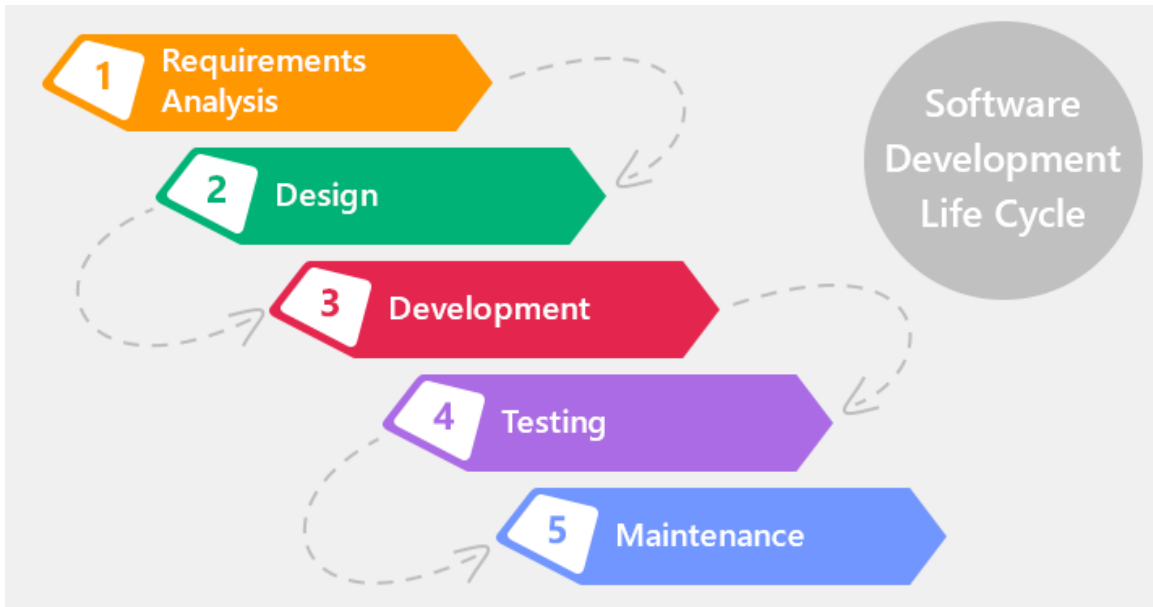
eClip was envisioned in July 2016 and began development in January 2017. Now coming to the end of the beta version there are so many more functionalities I can think of to integrate with eClip.

My project supervisor, Sara Kadry, and I will begin working on making eClip a fully commercial product in the coming months.



**Figure 10 - Use Case Diagram. This diagram presents the intended future functionality to be added to eClip.**





**Figure 11 - Waterfall Software Development Lifecycle (Gordiyenko, 2014)**

When I began to list functionality, I intended to be part of eClip, it was evident that all would not be achieved in the five-month implementation stage. I have been compiling a list of major functionalities that I will add to eClip in the coming months.

1. Multifactor authentication
2. Experiment sharing
3. Integrate with Eclipse
4. Develop the eClip website
5. Online dashboard of experiments

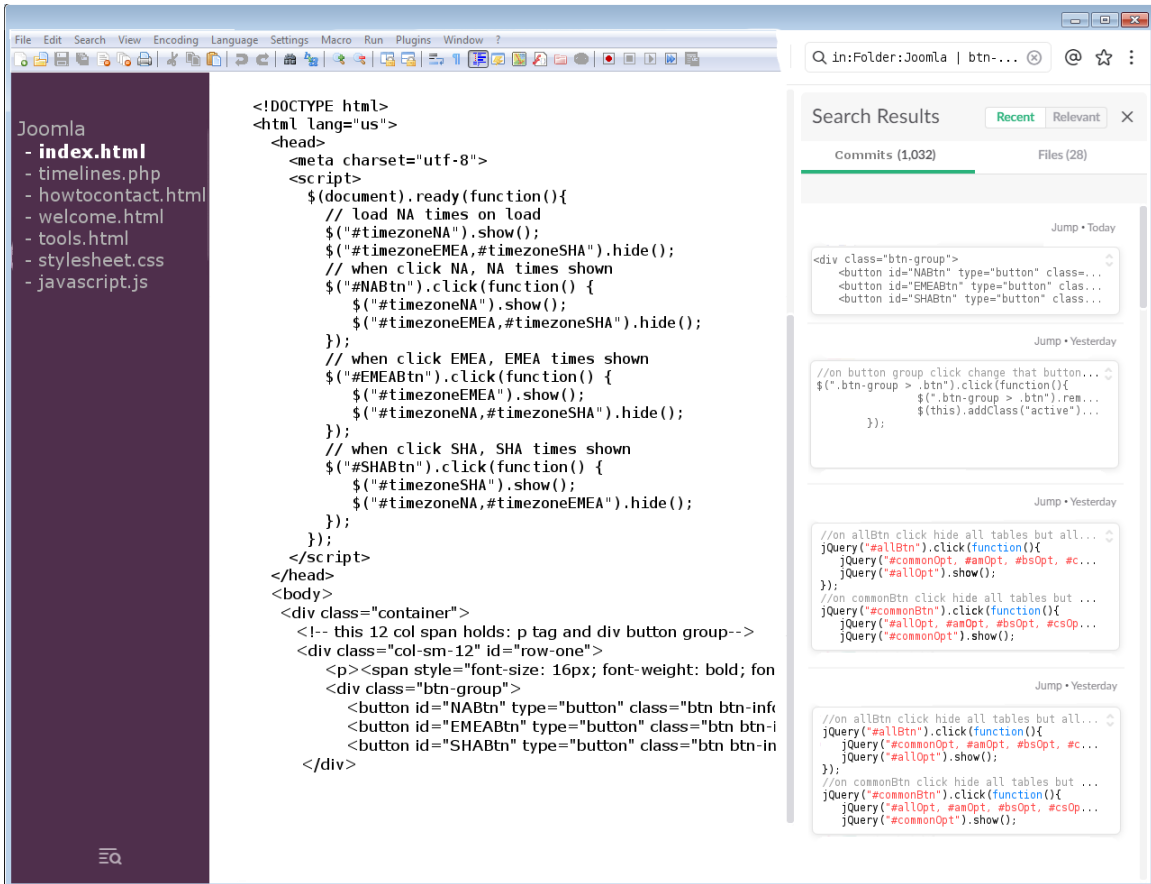


Figure 12 - How eClip will look similar to this with more development

## 5 References

- [1] Gordiyenko, S., 2014. *XB Software*. [Online]  
Available at: <https://xbsoftware.com/blog/software-development-life-cycle-waterfall-model/>  
[Accessed 02 05 2017].
- [2] Hoffman, C., 2016. *What Is a Virtual Machine?*. [Online]  
Available at: <http://www.makeuseof.com/tag/virtual-machine-makeuseof-explains/>  
[Accessed 06 12 2016].
- [3] Mehta, L., 2016. *InfoSec Institute*. [Online]  
Available at: <http://resources.infosecinstitute.com/category/certifications-training/cissp/domains/security-and-risk-management/the-security-cia-triad/>  
[Accessed 02 05 2017].
- [4] Miller, D., 2015. *jeremyh/jBCrypt*. [Online]  
Available at:  
<https://github.com/jeremyh/jBCrypt/blob/master/src/main/java/org/mindrot/BCrypt.java>  
[Accessed 14 03 2017].
- [5] Rouse, M., 2014. *Search Security*. [Online]  
Available at: <http://searchsecurity.techtarget.com/definition/encryption>  
[Accessed 02 05 2017].
- [6] Schneier, B., n.d. *Schneier*. [Online]  
Available at: <https://www.schneier.com/academic/blowfish/>  
[Accessed 14 03 2017].

# 6 Appendix

## 6.1 Project Proposal

### 1 Objectives

My final year project is called eClip. Think of a maths exam, you have your exam in the front of the workbook and your rough work in the back, after the exam you don't get to keep the rough work together.

My idea is to 'clip' all your work together. I will develop an Eclipse plug-in to allow developers, already using the Eclipse IDE, to save 'experiments' or code snippets with their projects. This will be helpful when referencing some code for a certain project, and do look back for future work. Functionality will also provide searching through experiments and reverting back to previous saves.

The objective of my final year project is to build software which aids real developers, including myself, in the implementation process of software projects and to incorporate everything I have learned so far in my four years of college together to best represent myself as a computer student and where I want my career to go.

I am specialising in cyber security for my final year and am anxious to make my software secure for the end users. There are three main aspects of how I will integrate security with my project:

1. I will secure the code. This means that the end user will not be able to modify the source code, which could compromise the safety of their code.
2. To activate the plug-in, the end user will need a product key, which they will have received in the purchasing process.

3. For the experiments the user will create with snippets of code, this will be saved in a database which will be encrypted, ensuring the integrity of the user's work.

As I am specialising in cyber security, I know that as I learn new things along the way, I will use new approaches and technologies where applicable for my project.

## **2 Background**

I came up with my project idea on my six-month internship in PayPal. I was developing an internal portal with information about the team I was in. My manager requested I use Joomla. For most of the sections on the website, it required me to hard code HTML, PHP, Javascript, jQuery and CSS into the Joomla framework. Inside the portal, I completed lengthy projects where I used Notepad++ for writing. For a project, I was developing a system for managers in PayPal to keep track of what office time zone was on the phone shift at any given time in the year.

As I had never dealt with time zones before, and never want to again, I was testing out existing Javascript frameworks, then deleting those lines of code and starting again. It was very hard to keep record of my experiments if I ever needed to look at them again, this lead me to saving all ideas I found online into my browser bookmarks. This let me to research into tools for version control, combined with note taking of code, ways to revert back to older code versions and search through your notes by programming language specific tags.

After much online research and talking with developers in my work, I was left stumped that this tool did not exist. Yes, you can use notepad++ to create a text file holding all your notes and code snippets

but you can revert your java project back two weeks or view you code snippets with ease while on Eclipse.

So, I started to think, “maybe I should make this a reality”. So, I asked as many developers as would listen about by theoretical idea and they provided much guidance.

I originally had thought about building a text editor to implement my ideas, but many of the developers didn’t like the idea of reinventing the wheel or downloading and having to become familiar with a new text editor, which may be missing features they have developed for themselves on Eclipse.

### **3 Technical Approach**

I will be implementing my project using Eclipse’s Plug-in Development Environment (PDE).

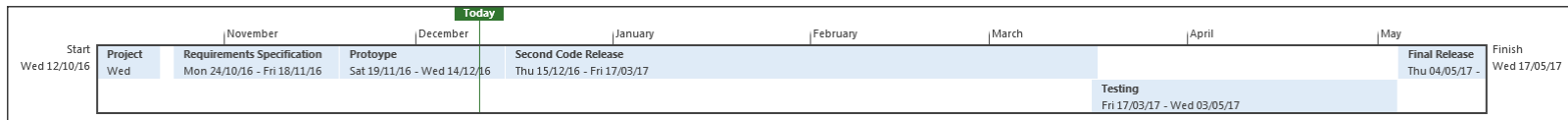
As Eclipse supports all major operating systems, therefore my plug-in will too.

I will be programming my plug-in with Java as that is my strongest programming language. I have no doubt that I may need to use other languages for different aspects of the project, therefore I will learn new languages along the way.

### **4 Special resources required**

As I am developing an Eclipse plug-in, I will not require any special hardware. The only requirements for users to use my software is to have Eclipse and my plug-in installed.

## 5 Project Plan



Task Name	Duration	Start	Finish
<b>eClip Final Project</b>	<b>156 days</b>	<b>Wed 12/10/16</b>	<b>Wed 17/05/17</b>
<b>Requirements</b>	<b>28 days</b>	<b>Wed 12/10/16</b>	<b>Fri 18/11/16</b>
Project Proposal	8 days	Wed 12/10/16	Fri 21/10/16
Requirements Specification	20 days	Mon 24/10/16	Fri 18/11/16
<b>Implementation</b>	<b>86 days</b>	<b>Sat 19/11/16</b>	<b>Fri 17/03/17</b>
Technical Report	18 days	Sat 19/11/16	Tue 13/12/16
Prototype	19 days	Sat 19/11/16	Wed 14/12/16
Second Code Release	67 days	Thu 15/12/16	Fri 17/03/17
<b>Verification</b>	<b>34 days</b>	<b>Fri 17/03/17</b>	<b>Wed 03/05/17</b>
Testing	34 days	Fri 17/03/17	Wed 03/05/17
Final Release	10 days	Thu 04/05/17	Wed 17/05/17

## 6 Evaluation

While on my internship I discussed with my teammates if they would test out my plug-in and provide feedback. For this plug-in to be as helpful as I plan it to be, I need it to be tested by real developers. With all the feedback I receive, I will take it on board and modify the plug-in to fulfil my project objectives while making it useful for developers.

## 6.1 Supervisor Meetings

<b>Date</b>	27 <sup>th</sup> of October 2016
<b>Time</b>	4:45pm
<b>Duration of Meeting</b>	30 minutes
<b>Current Challenges</b> <ul style="list-style-type: none"><li>• Deciding on a name for the project which incorporates a security element</li></ul>	
<b>Goals of Meeting</b> <ul style="list-style-type: none"><li>• Talk through all the possible functionality of the project and decide which to focus on first</li><li>• Decided to not to make project open source so I can secure my code and send paying users the .exe file to download</li></ul>	
<b>Goals/Actions for next Meeting</b> <ul style="list-style-type: none"><li>• Have the requirements specification already approved by Sara and ready to submit the following day (Friday 11<sup>th</sup> November)</li></ul>	



<b>Date</b>	10 <sup>th</sup> of November 2016
<b>Time</b>	4:00pm
<b>Duration of Meeting</b>	30 minutes
<b>Current Challenges</b>	
<ul style="list-style-type: none"> <li>• Struggling to write up the requirements specification.</li> </ul>	
<b>Goals of Meeting</b>	
<ul style="list-style-type: none"> <li>• Ask about thesis shared folder.</li> <li>• Work out most important functionality to focus on.</li> </ul>	
<b>Goals/Actions for next Meeting</b>	
<ul style="list-style-type: none"> <li>• Talk about the implementation process</li> </ul>	
<b>Date</b>	24 <sup>th</sup> of November 2016
<b>Time</b>	4:00pm
<b>Duration of Meeting</b>	30 minutes
<b>Current Challenges</b>	
<ul style="list-style-type: none"> <li>• Technical report</li> <li>• Deploying an eclipse plugin on windows machine</li> </ul>	
<b>Goals of Meeting</b>	
<ul style="list-style-type: none"> <li>• Realised I am struggling unusually with the technical report and so will apply for an extension due to my leaning difficulty.</li> <li>• Will make a virtual machine to run eclipse plugin.</li> <li>• Defined the three functionalities will have working for mid-point presentation.</li> </ul>	
<b>Goals/Actions for next Meeting</b>	
<ul style="list-style-type: none"> <li>• Work on time management examples.</li> </ul>	

<b>Date</b>	1 <sup>st</sup> of December 2016
<b>Time</b>	2:30 pm
<b>Duration of Meeting</b>	20 minutes
<b>Current Challenges</b>	
<ul style="list-style-type: none"> <li>• Confusion about extension after Eamon's email</li> </ul>	
<b>Goals of Meeting</b>	
<ul style="list-style-type: none"> <li>• Clear up confusion about submission date</li> <li>• Reevaluate prototype for presentation</li> </ul>	
<b>Goals/Actions for next Meeting</b>	
<ul style="list-style-type: none"> <li>• Pseudo code for 2 functionalities</li> </ul>	
<b>Extra:</b>	
Doing pseudo code for the presentation: <ol style="list-style-type: none"> <li>1. eClip website authentication *Integrity *Pay *Access</li> <li>2. Database Encryption</li> <li>3. Sharing Code through eClip servers</li> </ol>	
<b>Date</b>	2 <sup>nd</sup> of February 2017
<b>Time</b>	3:00 pm
<b>Duration of Meeting</b>	40 minutes
<b>Current Challenges</b>	
<ul style="list-style-type: none"> <li>• Issues deploying an Eclipse plugin on Windows machine.</li> <li>• Trying to look for online tutorials about Eclipse plugins is proving difficult</li> </ul>	
<b>Goals of Meeting</b>	
<ul style="list-style-type: none"> <li>• Talk about mid-point presentation. Where to improve and what was well presented.</li> <li>• How to work around plugin deploying issues.</li> <li>• List put together of sections to improve on in the documentation.</li> </ul>	
<b>Goals/Actions for next Meeting</b>	
<ul style="list-style-type: none"> <li>• Have implementation to show (methods)</li> </ul>	

<b>Date</b>	10 <sup>th</sup> of February 2017
<b>Time</b>	12:30 pm
<b>Duration of Meeting</b>	30 minutes
<b>Current Challenges</b>	
<ul style="list-style-type: none"> <li>• Needed to make the JFrame for all menus on NetBeans</li> </ul>	
<b>Goals of Meeting</b>	
<ul style="list-style-type: none"> <li>• Talk briefly about masters programmes</li> </ul>	
<b>Goals/Actions for next Meeting</b>	
<ul style="list-style-type: none"> <li>• Have implementation to show (methods)</li> </ul>	

<b>Date</b>	16 <sup>th</sup> of February 2017
<b>Time</b>	12:30 pm
<b>Duration of Meeting</b>	30 minutes
<b>Current Challenges</b>	
<ul style="list-style-type: none"> <li>• Issues linking database connection with JFrame on NetBeans. Issues because of private ActionListener in the MyGUI class, not linking to DBConnection.java separate class.</li> </ul>	
<b>Goals of Meeting</b>	
<ul style="list-style-type: none"> <li>• Check I'm on track</li> </ul>	
<b>Goals/Actions for next Meeting</b>	
<ul style="list-style-type: none"> <li>• Have issue with class connections fixed and can add an experiment which sends to the DB and switches to experiment completion page.</li> </ul>	

<b>Date</b>	2nd of March 2017
<b>Time</b>	4:00 pm
<b>Duration of Meeting</b>	45 minutes
<b>Current Challenges</b>	
<ul style="list-style-type: none"> <li>• Issue sending snippet that the user can highlight, to the page to fill in all details of the experiment.</li> </ul>	
<b>Goals of Meeting</b>	
<ul style="list-style-type: none"> <li>• Ask about encryptions methods to use.</li> <li>• Check if on track with timing of project.</li> </ul>	
<b>Goals/Actions for next Meeting</b>	
<ul style="list-style-type: none"> <li>• Have all issues fixed.</li> <li>• Adding experiment functionality completed.</li> </ul>	

<b>Date</b>	9 <sup>th</sup> of March 2017
<b>Time</b>	3:45 pm
<b>Duration of Meeting</b>	30 minutes
<b>Current Challenges</b>	
<ul style="list-style-type: none"> <li>• Issue showing list of all experiments. In arraylist so won't be able to define amount of experiments.</li> <li>• Now where to start with database encryption.</li> </ul>	
<b>Goals of Meeting</b>	
<ul style="list-style-type: none"> <li>• Work out next tasks.</li> <li>• Understand database encryption methods to add.</li> </ul>	
<b>Goals/Actions for next Meeting</b>	
<ul style="list-style-type: none"> <li>• Have completed authentication of users, access permissions for users adding to database, search method or policy contract.</li> </ul>	

