Liam Jones

COMP-1004 design documentation

Contents

[Introduction 2](#_Toc163837655)

[Software development life cycle 2](#_Toc163837656)

[Requirement analysis 2](#_Toc163837657)

[Design 2](#_Toc163837658)

[Waterfall model 2](#_Toc163837659)

[Agile model 2](#_Toc163837660)

[Project vision 4](#_Toc163837661)

[Background 4](#_Toc163837662)

[User stories and use case scenarios. 4](#_Toc163837663)

[References 6](#_Toc163837664)

# Introduction

In this documentation I will be outlining the steps taken to design, plan, implement, and test my project for the COMP-1004 module. Due to my focus on the course being cyber security, I will be creating a program which caters towards this aspect of my course.

For this project, I will be developing a single page web application which will aim to act as a username and password storage system. The user should be able to enter a username and password to be stored and the system will also indicate to the user the strength of their password.

# Software development life cycle

A crucial part to the development of any software development is conforming to a software development cycle designed to keep developers on track and to effectively resolve any time delays and issues faced during the development process. The five stages of the SDLC include:

* Requirement analysis
* Design
* Implementation
* Testing
* Evolution

If these steps are not followed as intended, a project could take end up taking much longer to complete than initially planned or even fail entirely if it is deemed that the development cost is higher than the potential return.

*Hackr.io. (2023).*

## Requirement analysis

There are 3 main requirement types in the first step of the SDLC, those being the functional, non-functional, and usability, each having different considerations that must be discussed and planned for. Firstly, the usability requirements of a project refer to how easy it is to use the program upon completion. Secondly, non-functional requirements refer to the limitations of a software and how it can meet the requirements of the program as well as how it may not be possible to meet specific requirements. Finally, functional requirements refer to what the program can do and how well it can handle different use cases such as being able to handle extreme or incorrect data inputs without crashing. External requirements are another part which must be considered, this refers to any legal, ethical, or social issues which must be acknowledged.

## Design

The next step is to take the requirements and design a piece of software based on them. The main ways of designing a software from requirements is to look at the style and details of the software. The style section of this design refers to the outline of how the program will work on a basic level and detailed design refers to a more in-depth design template, outlining how specific functions will be implemented into working code. Furthermore, UML’s will be created to visualise the project design and include models such as state, sequence, and component diagrams.

Upon completing these steps, the program will be implemented into working code before being tested and any issues resolved with further code (evolution). During testing, each function of the program will be subjected to various use cases to identify any logical errors within the program such as by testing with extreme or invalid data, or no data for inputs, or trying to request data which is not present. Once testing results are obtained, the code will undergo evolution where necessary to resolve any issues such as certain functions or data cases not performing as intended.

## Waterfall model

The waterfall model is a straightforward method of software development that has been in use for many years by which developers follow a linear progression between each stage by completing one section before beginning the next such as completing implementation then beginning testing. This methodology allows developers to follow clear deadline and gives clients a better understanding of when goals may be completed. However, the flaw with this methodology is that if changes to the project requirements arise, it could lead to significant delays as the current stage must be completed before re-evaluating and designing new features.

## Agile model

The agile model is designed to provide fast results whereby stages are completed in sprints usually lasting a few weeks. First a very basic version of the software is developed (minimum viable product).

## 

# Project vision

My vision for this project is for it to be made use of by non-professional web users who wish to securely store their usernames and passwords whilst being informed on the individual security of each password being stored to prevent the use of easily cracked passwords.

## Background

Username and password management systems have been available for public use for over 20 years and aid the user in keeping track of each individual login for various websites. The use of password managers has been widely observed to encourage higher standards of login security as the user does not need to remember various usernames and passwords as the manager keeps track for them, therefore security standards are more likely to be adhered to.

Due to the nature of data being stored by username and password management software, legal, and ethical considerations must be taken to comply with current regulations. The’ Data Protection Act 1998’ and ‘Data Protection Act 2018’ are the UK legislation which govern how data can be collected and stored as well as what steps an organisation must take to ensure the security of said data. In this case steps must be taken to ensure that only the necessary data is stored, e.g., Usernames, Passwords, and the site they are for. Furthermore, the program should securely store any data collected to comply with legislation and prevent data leaks.

## User stories and use case scenarios.

in this section, the user stories and use cases will be outlined in the following diagram. A diagram of a user account

Description automatically generated

# References

Hackr.io. (2023). *Top 7 SDLC Methodologies [Complete Guide]: Agile, Waterfall.* [online] Available at: <https://hackr.io/blog/sdlc-methodologies>. [Accessed 15 November 2023]