6COSC006W – Final Year Project Report

Project Title

Student first and last name (student number)

**Supervisor:** Supervisor Name

This report is submitted in partial fulfillment of the requirements for the BSc (Hons) XXX degree at the University of Westminster.

School of Computing & Engineering

University of Westminster

Date

# Declaration

This report has been prepared based on my own work. Where other published and unpublished source materials have been used, these have been acknowledged in references.

Word Count:

Student Name:

Date of Submission:

# Abstract

*500* words

Summarise here the problem statement, your aims (how you propose to solve the problem). Provide a brief description of the methodology, main results and observations.

# Acknowledgements

Thank those you wish to.

# Table of contents

Provide a table of contents, linking Sections and Subsections to page numbers. If you can, hyperlink the page numbers/sections.

# List of figures

Provide a list of figures, linking figure numbers to page numbers. If you can, hyperlink the page numbers/figures.

# 1. Introduction

Include the problem statement, project aim and objectives.

## 1.1 Problem statement

*500* words

Explain background to the problem you intend to solve and the need for the software/application. Use illustrations, diagrams, figures, if needed.

## 1.2 Aim and Objectives

*300 words*

The **aim(s)** describe, in a few sentences, the overarching purpose(s)/intention(s) of the software/application. What is the point of developing the software/application and what you wish to achieve.

**Objectives** describe with some detail the individual steps you will take to achieve your aim(s).

# 2. Background

Include a literature survey in the research topic, discuss existing similar or relevant applications to yours and the result of a review of tools and techniques that are used to tackle projects similar to yours.

## 2.1 Literature survey

*800* *words*

Describe initial results of a literature survey on research topics related to your project subject. Use relevant books, published research articles as well as Internet content for the purpose.

## 2.2 Review of projects / applications

*800 words*

Describe your background research on existing projects/software/applications relevant to yours, their advantages and disadvantages. Use illustrations, diagrams, screenshots for the purpose.

Produce a **Table of Features** this section, comparing the main features of the above projects/software/applications and the one you are currently building.

## 2.3 Review of tools and techniques

*800* words

Describe results of a survey on relevant tools, programming languages and environments, libraries. List their advantages and disadvantages. Use illustrations, diagrams, screen-shots for the purpose.

# 3. Requirements

Introduce the project stakeholders, the methods for the elicitation of the project requirements, how you model your requirements and relevant diagrams. Finally discuss legal, ethical, social, professional and security issues associated with your research and the software/application you are building.

## 3.1 Stakeholders

*100 words*

Describe people and, if relevant, organisations who will be using, maintaining, or/and will be affected by your software/application.

## 3.2 Gathering requirements

*600 words*

Describe the techniques/procedures you used for gathering requirements (other than the exploration of existing applications, listed in Section 2). Results will be summarised in this section (use graphs, if necessary).

Notes: Detailed results/raw data from the process of gathering requirements will be included in an Appendix in the Final Project Report.

## 3.3 Modelling requirements and relevant diagrams

*600 words + diagrams*

CS & SE (use formal analysis, design, and final code representation, plus any diagram that assists understanding)

DMD (use formal analysis, design, and final code representation, plus any diagram that assists understanding). Below is a list of diagrams that could be included:

* Context diagram
* Stakeholders and their goals
* Use Cases
* Personas
* Flow charts (depending on the project)

BIS

* SWOT – PEST - CSF
* Context diagram
* Use case modelling - Use case diagrams and use case descriptions
* Activity modelling – Actions and activity diagrams
* Class modelling – classes, attributes, class diagrams
* Entity relationship diagrams

Games (use formal analysis, design, and final code representation, plus any diagram that assists understanding). Below is a list of diagrams that could be included:

* Context diagram
* Stakeholders and their goals
* Development Stage Diagram
* Use Cases/control Diagrams
* Activity diagrams / GUI diagrams
* Level Design Analysis

## 3.4 List of project requirements

*100 words*

List (or create a able with) Functional and Non-Functional requirements of the software/application and classify them in “Essential”, “Desirable” and “Luxury”.

## 3.5 Legal, social and ethical issues

*200 words*

Consider any legal, ethical, social, professional and security issues associated with your research and the software/application you are building.

# 4. Methodology

Describe fully the methodologies, processes and development techniques you followed to assist the design and development of the software/application.

*700 Words*

CS & SE (use appropriate methodology for project. e.g. UX, UI, unit testing for typical client-server applications, white box for algorithmic and mission critical code etc. Use prototypes with outputs every 2-3 weeks where appropriate.)

DMD - use appropriate methodology for digital media projects e.g. discuss how software development methodologies like the ones in the following list best suit the project and why:

* Waterfall
* Spiral
* Rapid Application Development (RAD)
* Agile Software Development
* Crystal Methods
* Dynamic Systems Development Model (DSDM)
* Extreme Programming (XP)
* Feature Driven Development (FDD)
* Joint Application Development (JAD)
* Lean Development (LD)

for example:

* Multimedia Development (flowcharting, menu maps, data flow diagrams – storyboarding – prototyping – assets development)
* Web based application development (requirements – architecture envisioning – prototype development and acceptance – resource procurement and contract negotiation – feature driven development – SCRUM (delivery sets are the prioritized feature/task list producted in a sprint) – evolution process
* Mobile development approaches – Web app – Native – Hybrid (inception, design (UX, UI, development, stabilization (prototype, alpha, beta, release), deployment)
* Game development (initiation, pre-production, production, testing, beta, and release)

BIS

* Development approach- structured / object oriented etc.
* Agile / traditional; describe the chosen methodology and the reasons (merits) of the chosen methodology.
* Describe how/if hybrid methodology is used. i.e. which techniques and/or processes are used from which methodology and why.

Games (use appropriate methodology for project. e.g. UX, UI, unit testing for typical client-server applications, white box for algorithmic and mission critical code etc. Use thow-away prototypes with outputs every 2-3 weeks where appropriate including Camera-view outfits.)

Provide clear representation of the SDLC selected methodology

Development Stage Diagram with clear indicators of testing over Alpha- beta and gold edition releases of the prototype

# 5. Design

Describe design issues related to your software/application, discuss how the proposed project design will be implemented and the tools you will be using for doing so.

*700 Words*

Discuss in some detail issues relating to:

* User Interface
* Infrastructure
* Functionality
* Algorithm development (if applicable)
* Content creation
* Other

Discuss how these address the project requirements.

CS & SE (use appropriate design methods for project e.g. make use of activity/state diagrams for complex algorithms and workflows, use UI design methodology and heuristics for predominately UX based projects)

DMD (use appropriate design methods for project e.g. make use of Flowcharting, menu maps, data flow diagrams – storyboarding – prototyping – assets development)

BIS

Developing a business strategy to solve a business problems – business process mapping – SWOT – PEST – Business models etc.

Systems design to include ‘current system’ and proposed ‘new’ systems

The techniques used will vary according to the methodology chosen.

Example – for traditional methodology - Data Flow diagrams, ER diagrams etc can be used

for agile – user stories, estimation techniques, acceptance test etc. are required

Prototypes used/developed

Games use appropriate design methods for project e.g. make use of activity/state transition diagrams for complex algorithms and workflows, use UI design methodology and heuristics for predominately UX based projects, storyboarding level designs (Walkthrough)

# 6. Tools and implementation

## 6.1 Tools

*300 words*

Describe the tools (programming environments & languages, libraries,) you used for the development of your application. Justify your choices.

State acquired or/and new skills you employ.

## 6.2 Implementation

*2200 words*

Explain implementation of main code for key functions, indicate any novel code clearly and code that is adopted/adapted and the original sources.

# 7. Testing

## 7.1 Functional testing

*800 words*

Discuss black box or/and white box testing.

## 7.2 User testing

*700 words*

Discuss how you obtained used feedback, using expert and non-expert users.

# 8. Conclusions and reflections

*1000 words*

Include conclusions and reflections (strengths and weaknesses) on the resulting application, acquisition of any new knowledge and skills and further work to improve its workings.

# 9. References

Include a list of cited in your text items (books, papers, websites, etc.). Use Harvard style for the purpose, or any other preferred standard referencing style.

# 10. Bibliography

Include here a list of general reading items (books, papers, websites, etc.). List the items in alphabetical order, using Harvard style to describe them.

# Appendix I

Provide additional material, if appropriate, in separate appendices.

Do not include the entire code in print.