**3rd Year Project**

|  |  |
| --- | --- |
| Deliverables | **Git Repository** |
|  | **Design Document** |
|  | **Implementation** |
|  | **Video** |

Demonstration of the project will take place during the week of April 24th. Have your video and presentation ready to go. Have your solution tested so that you can demonstrate a working version of the project. All deliverables will be submitted at the time of the presentation.

# Git Repository

This is the main hand in mechanism. This link is submitted, or your project supervisor is added as a collaborator. You should set up the git repository for the project team and everybody should have a history of commits with comments in the project. Remember, prospective employers are now looking at your git repository and how you use it to show that you understand the collaboration process for developing software.

# Design Document

A Design Document is a detailed technical document. It is intended to provide both user and developer with background information for the system. It provides information on the system from two perspectives.

The first details what the finished product will do, how the user will interact with it and what the system will look like.

The second documents how the application will be implemented, concentrating on the technical aspects of the user interactions rather than what the system looks like.

As part of the development process, this document allows the development team to scope out the different technologies that are available to provide a solution to different aspects of system implementation. These technologies can be compared and decisions made on the merits of each as to whether it is appropriate to use in the development of the new system.

The requirements for the system have been previously documented and it is here that the system is detailed in the functional elements from which it is composed.

Some of the questions to answer in the writing of this document include:

* Will this be a multi-user system of a single-user system (for one user at a time)?
* Are there special business rules and/or calculations that pertain to this industry?
* How many data entry screens are required?
* What kind of output do the users require?
  + Will the system produce web pages, printed reports, mail merge files or other types of output?
* What is the subject of the database and what purpose does the data stored serve?
* Will the system require static or dynamic updates to a website?
* Will the product be localised into more than one language?
* What are the hardware and software requirements to run this system?
  + Example: MS Access on a PC with 500 Mb of disk space for storage.
* Provide a list all of the tables to be included in the database and a brief description of each, describe the file structure and the relationship between the tables.
* What resource files are required for multi-language support?

This is only a suggested list and should not be taken as complete or exhaustive as each system is different. In writing the Design Document, take the following into consideration:

* Architectural Overview of the system.
  + Provide a functional breakdown of the system to be developed.
* Identify and compare the possible technologies, justify the choice of technology to be used in the implementation.
* Database design.
  + Develop the database to 3rd Normal Form for the implementation.
  + Specify data types, field sizes, default values and any constraints on the values to be stored.
* Screen Layout providing a blueprint for how the particular application will look to the end user.

A **possible** set of headings for the design document include:

* Introduction
* System Requirements
* Technology Used and Why
* Architecture of the Solution
* Design Methodology
* Features of the Implementation
* Limitations
* Known Bugs
* Recommendations for Future Development
* Conclusions

# Implementation

The implementation of the system is to be considered as three separate parts. These are:

## Code

Code should be written in a modular fashion implementing all of the best practice guidelines that have been provided during your courses. All code functions, modules and classes should be well commented. Best practice is also expected on the naming of classes, functions, variables, modules and forms that are used throughout the system.

Source code needs to be managed through Git. Set up a project repository, use the Wiki feature and the Issue tracker feature on Git to create a history of work and collaboration on the project for your supervisor and your project team (regardless of team size). In this way, you can create a task list, set deadlines and update weekly to keep your supervisor informed.

## Installation & Configuration Guide

This is a document that guides the user through the installation or set-up process. The

Installation Guide details any supporting packages that must be installed. For example, if the database engine is to be MySQL, then the user must be told how to download and begin installation of MySQL. If there is any specific configuration to be done in a supporting package, it should be detailed in the Installation Guide. You do not have to incorporate the Installation Guide for supporting packages, but you must provide a means for the user to locate and read them, possibly a URL would be appropriate.

The Installation Guide is a step-by-step guide through the successful installation and initial configuration of the system being installed.

## User Guide

The User Guide is the documentation that users will turn to in the event that they are not familiar with some part of the system. This document is easy to follow and incorporates screen shots where appropriate for the user. It can be in HTML, Word or an online help format.

# Video

You should create a short video that will be used to demonstrate your project to the external examiner. The video should be no more than 2 minutes in length depending on the project and should show the major features for the project. You can do a voice over on the video and explain what is happening.