**FACULTY OF SCIENCE, ENGINEERING AND COMPUTING**

**School of *Computing and Information Systems***

**BSc DEGREE**

**IN**

**Computer Science**

**PROJECT PROPOSAL**

Name: Daniel Carnovale

ID Number: K1336511

Project Title: Individual Project (CI6300)

Project Type: Design

Date: 14/10/2016

Supervisor: Pushpa Kumarapeli



Did you discuss and agree the viability of your project idea with your supervisor? Yes

Did you submit a draft of your proposal to your supervisor? Yes

Did you receive feedback from your supervisor on any submitted draft? Yes

**Abstract**

Many low-cost cloud-computing resources and new free open-source software components are now available to software developers. These have reduced the costs that were associated with the creation of web-based software systems. Although a large team of IT professionals and expensive hardware and communications resources are no longer necessary, and can be replaced by just a few individuals, it’s still essential for those individuals to be able to identify which tools and resources will meet their needs. They must also know how to effectively combine these tools to build their proposed system.

This project is an attempt to itemise and instruct (in a “HOW-TO” fashion) one such combination of tools and resources that will allow a complete web application system to be built with minimal code, costs and manpower. The project will address many of the considerations that are essential for real-world feasibility, such as Security, Source control, Issue management, Disaster Recovery, Scalability and Maintainability. To achieve the above aims, I have chosen to build a system called “UniHood” which will allow a university community to review and comment on courses, their modules and lecturers in a more structured way than the typical forums and social networks.

**Background**

**Stakeholders**

It’s important to recognise that the primary audience for this report is not the community of users of UniHood, but it is rather any organisations or individuals who are interested to see how a complete web application system can be assembled with minimum cost and effort using open-source and cloud-based resources as far as possible.

**Problem outline**

There is a wide range of new tools which are becoming available, and one characteristic of the open-source environment is that each tool has a different owner; therefore there are very few resources that can guide how to combine these tools. This report attempts to provide at least one such guideline.

My project aims to:

* Build a public facing database-backed website
* Provide the tools and structure that would allow it to scale to higher complexity
* Document all steps of the procedure in a way that will allow others to reach similar level of functionality, with minimal effort
* Report, itemise and minimise any expenses
* Minimise actual coding via maximum deployment of existing open-source components.
* Provide system documentation

**Importance of Project**

This work is particularly important given that the huge diversity of Open-Source components now available makes it extremely difficult for many students and individuals to get started with building a system such as UniHood. This project will help them begin to realise the cost-savings that are being achieved via cloud-based computing.

**Benefits to the stakeholders**

Stakeholders should be able to use these instructions to build a completely different application. Many individuals and organisations would find the HOW-TO deliverables of this project extremely valuable because of the significant time-saving if they want to start a project using similar tools.

**Relation to previous work and existing knowledge**

There is a long tradition of “HOW-TOs” in the open-source community, and some of these may have similar goals, however the range of available web-building tools and resources (such as Amazon AWS Free Tier and Google Single Sign-On) is evolving so rapidly that one might prepare the same report in 10 months’ time and recommend different tools in it.

**Aim and objectives**

**Primary Deliverables**

The primary deliverables are:

* Running cloud-based service, ie. the UniHood system operating as a live public-facing website
* Collection of specific HOW-TOs including screenshots
* Explanation of Components
* Design Document
* Discussion of functionality achieved
* Observations

**Intermediate Deliverables**

There are many intermediate deliverables, each of which is self-contained and may be of value to many different types of projects. Some of these deliverables will be self-contained discussions of:

* Authentication
* Authorisation
* Security
* Configuration
* Source control
* Issue management
* Backup
* Disaster Recovery
* Scalability
* Productivity
* Automated testing
* Continuous Integration
* Continuous Delivery
* Continuity of Service
* Domain Name Management

**Technologies & Resources**

**Major resources required**

* Online documentation pages for Amazon AWS, MySql and other Software resources
* Forums and discussion sites such as StackOverflow
* Client-side PC workstation(s)

**Technologies Utilised**

For Cloud-Computing resources I expect to use various Amazon AWS Resources such as:

* EC2 – Elastic Compute Cloud, dedicated to managing Amazon Virtual Machines.
* S3 - Simple Storage Service, allows upload and download of very large files
* EBS – Elastic Block Storage, mainly used for adding extra drives to currently running Virtual Machine instances.
* Elastic IPs – A concept in which static IPs are able to be assigned to an Amazon Account and because of their “elasticity”, provide a means of mapping this same IP straight to another instance when the previously IP assigned instance fails.

Amazon offers a “Free Tier” service which allows these above resources to be used for free or very low-cost during the first year.

Additional resources to be chosen from:

* GitHub for Source Control and as a Cloud-Based repository of project documentation. This is a free service for Open-Source project work.
* MySQL for database back-end
* PuTTY as an SSH client
* Programming language eg. PHP, Python, Ruby or Perl.
* HTML5 / CSS / Javascript for browser-side programming
* Javascript UI Framework eg. Bootstrap, Angular, JQuery UI
* Database Technology eg. mySQL, PostGressQL
* Configuration tools eg. Ansible, Docker

**Data Sources**

Sample test data will have to be improvised according to the evolution of the project. The reasoning for this is that it is always impossible to speculate how the final database structure will come together, so as it evolves with the project so will the test data. No alternatives for the test data will be discussed nor will they be necessary; although this is a practical project it is also an Open Source one as stated earlier, which means the data cannot be limited to just one organisation (in this case Kingston University).

**Method & Work Plan**

I have already noted above some of the technologies I have proposed to use, but an important aspect of this project is to survey and in some cases trial some alternatives to these and the additional components which will help me put a system together with minimal coding. Because of this requirement, I propose an unusual approach compared with the traditional structured methodologies.

I aim to rapidly survey product and forum websites, to identify choices. To be a candidate for trial a component must be:

* A necessary or highly desirable component of the finished system
* Able to be quickly and easily configured into the application.

I will use this methodology in an iterative fashion, while examining and speculating as to how each attempt will affect how the database system and web presentation comes together. There will be a level of trial and error in the steps which will determine the tasks in the final HOW-TO document, and this will ensure all the correct steps from this trialling will end up as concrete statements in this document.

**Use of Technologies**

I will focus on an Amazon (Free Tier) virtual machine providing these functionalities and in particular the optimal / custom alterations of the machine's OS' specifications such as RAM, Hard Disk space (as well as the procedure of adding an extra virtual drive known as an Elastic Block Store (EBS) to the virtual machine and so on.

This project proposal aims to outline all objectives involved in the technical backend aspects of a practical website while aiming to minimise the associated costs. The objectives involved include setting up and managing a database environment (planning to use SQL Developer or similar), creation, configuration and connection to Virtual Machine (VM) instances (through Amazon Web Services), reservation of domain name through GoDaddy and its assignment to the UK registry.

Additional objectives regarding connection to the VM instance via an SSH client (PuTTY in this case) include a series of steps provided by Amazon on the "connection" window, and include implementation of private key files (both .pem and .ppk extensions) which provide a secure connection between PuTTY and the VM.

**Context**

The significance of this work to my study of the Computer Science course is of utilising my existing skills to create a unique system, unique in a way that it differs from my usual projects such as coded applications for external theoretical companies. At the same time, it will benefit the students of Kingston University through offering their lecturers insight as to which teaching methods / techniques are most effective, and is open for adaptation in the foreseeable future. The only relation to previous work is the implementation of the database hierarchy, the majority consists of working with new cloud-based technologies and implementing the services like Virtual Machine management they offer. My existing knowledge of Virtual Machines and the like however, will be fully applied to the construction of this database-backed website.

By the completion of the website, there will be situations where problems may still arise and will need to be investigated promptly; for the purposes of the final report these will be investigated ahead of time. If for example there is a case to recreate a VM instance, then resolution via the "Elastic IP" concept and ip reset for the domain name on Godaddy will need to be addressed. My hypothesis aims to test whether these problems can be resolved within a reasonable time that, theoretically, would not affect development of the main project.

**Major phases of the project and milestones**

Investigation Phase

* 30th September : Verification of Amazon "Free Tier" account including creation of and access to a virtual Linux server instance and progressive documentation
* 14th October : Proposal Due

Skeletal Phase

* 31st October : Database technologies evaluated, skeletal database established
* 30th November : Programming technologies evaluated, initial screens implemented
* 16th December : Early Prototype Demo Presentation
* 31st December : Final technical direction in place

Construction Phase

* 31st January : Body of code construction implemented
* 28th February : Review and re-evaluation of technical options and re-implementation where necessary, essential functionality complete

Finalisation Phase

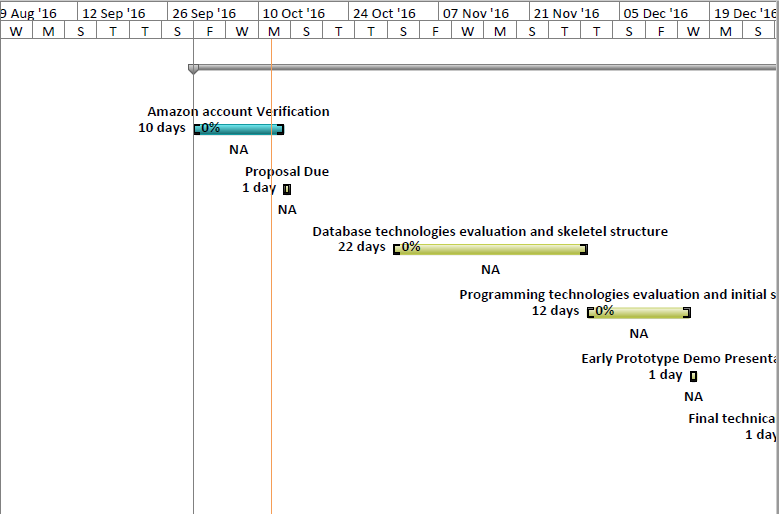
* 31st March : Compilation of all screenshots and explanations into report
* 24th April : Final Report Due
* 30th April : Final presentation ready
* 5th May : Viva Presentation

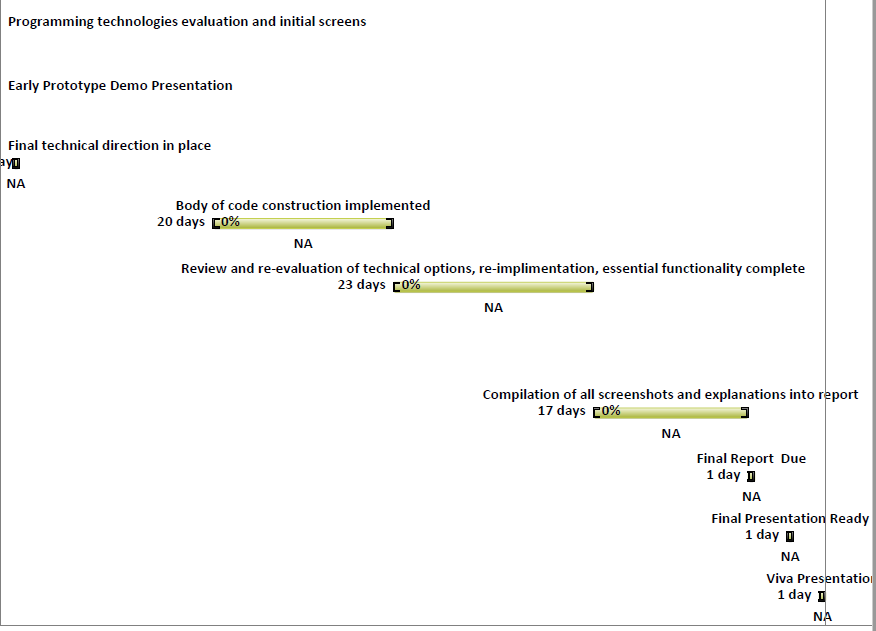
**Major risks and contingencies**

1: Given that I do not have a high level of expertise and experience as a programmer, it may not be possible to complete all of the functionality that would make Uni-Hood a real-world production system. While I will attempt to implement whatever useful functionality I can, I do not see this as a major issue because my goal is to investigate and report on how zero or low-cost components can be assembled to allow this type of project to proceed.

2: Some of the chosen components eg. Javascript libraries, Database libraries may turn out to be bad choices. This would obviously affect how much functionality I can deliver. The contingency here will be that in the evaluation phase, I will try to identify better choices and at least demonstrate that the short-comings of the initially chosen option can be resolved.

**Gantt Chart**

****

****

**Ethical Considerations**

The system will use Open-Source components as much as possible and will always acknowledge the source of any Open-Source software. Other resources, such as the Amazon Free-Tier, are encouraged for student use. For these reasons I see this system as having no ethical conflict, in terms of how it will be put together. Regarding the proposed application system "Uni-Hood", if it does actually get exposed to the community for use then this potentially raises questions of privacy and confidentiality of the users involved. If I succeed in getting the system to a stage where there is a genuine community of users, it will be necessary to get professional advice on whether there are any concerns here.

**Data Protection**

A feature of my method and an illustration of the modern approach to this type of project is that all code and documentation will be committed to GitHub, giving in effect a high confidence level that not only all work is easily accessible from the cloud, but a full audit trail will always be available. In order to protect from the extremely unlikely scenario of a catastrophic failure of the GitHub ecosystem, the system itself being built will work from its own local Git repository, which is a clone of the GitHub repository. In order to demonstrate the database backup and recoverability concepts, I expect to use Amazon's S3 facility which is an extremely reliable data storage resource.

**Intellectual Property Rights / Access to Code**

All of my own work and code will itself be Open-Source and visible to any GitHub user, and I will not be duplicating any similar work, hence I do not believe there are any issues regarding intellectual property rights. As is the convention with GitHub projects, I am allowing my work to be publicly available and freely fork-able by these users, with no limits in place such as freedom of distribution.