

National College of Ireland
BSc in Computing
2015/2016

Wayne Flanagan
X12112461
wayne.flanagan@student.ncirl.ie

Cloud-StrApp

Technical Report



Declaration Cover Sheet for Project Submission

SECTION 1 *Student to complete*

Name: Wayne Flanagan
Student ID: X12112461
Supervisor: Eamon Nolan

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: _____

Table of Contents

Executive Summary	4
1 Introduction.....	5
1.1 Background	5
1.2 Aims	5
1.3 Technologies	6
1.4 Structure.....	7
2 System	8
2.1 Requirements	8
2.1.1 Functional requirements	8
2.1.2 Requirement 1 <User Registration>.....	9
2.1.3 Requirement 2 <IDE Start-Up>	11
2.1.4 Requirement 3 <Save Files>.....	13
2.1.5 Requirement 4 <Access Files>	15
2.2 Design and Architecture	17
2.3 Implementation	19
2.4 Testing.....	21
2.5 Graphical User Interface (GUI) Layout	23
3 Conclusions.....	25
4 Further development or research	26
5 References	27
6 Appendix	28
6.1 Project Proposal	28
6.2 Project Plan	39
6.3 Monthly Journals.....	40

Executive Summary

The demand for processor hungry software applications drives the need for costly, high-end hardware machines. This report describes and evaluates the viability of the development of a cloud based on-demand, virtualised, ubiquitous and scalable application streaming system that proposes to remove the need for such machines and provide cost savings at every level of user.

It is the intention of this system to fully leverage latest technologies and advancements in Cloud Computing so as to demonstrate the far reaching benefits of the adoption of this model.

The technologies used to deliver this system are complex and convoluted but this document is designed to provide an abstract level of understanding to the reader so as to enable conclusions to be drawn on its viability without an intrinsic knowledge of its implementation.

Within this report the journey from idea inception through requirements gathering, analysis and design, including detailed system architecture diagrams, to final version will be described and detailed.

In conclusion attention is drawn to the fact that this system, in its current state, offers a viable option for the delivery of virtualised applications to students while providing a potential for growth and improvement to become a genuine business opportunity.

1 Introduction

1.1 *Background*

The idea for this project was borne out of my experiences during my time in college coupled with my experiences while on internship. The frustrations of using slow desktop virtualisation systems both on-site and remotely in addition to having to re-sit an exam due to a system crash led me to think about the possible alternatives to this approach of delivering SaaS and ultimately DaaS to students.

From my time on internship I was amazed at the level of cost incurred by the company to ensure that developers had the latest machines and top specification hardware to enable them to run the multiple choices of text editors and IDE's that they use. I began to think about how these costs and necessity for high end machines could be avoided.

1.2 *Aims*

The aim of my project, Cloud-StrApp, is to demonstrate the viability of a system with possible marketable potential that resolves the aforementioned problems.

By leveraging the power of the technology advancements in Cloud Computing I hope to demonstrate the ability to deliver SaaS and DaaS in a way that would enable colleges to change the current delivery model for desktop virtualisation and software availability for students without the need for large capital investment and with the ability to easily manage scaling of resources and in turn providing scope for cost savings.

In addition this method of application software delivery would also enable similar savings for software development companies. By using this method companies could reduce the need for high end development machines. In providing thin clients and shifting to this method of software delivery they

could take advantage of the processing power of off-site distributed and elastically scalable server instances.

1.3 Technologies

Nodes.js: Using Node allows for the web app to run fast. Node.js is a JavaScript framework that uses the V8 engine developed by Google for use in Chrome. V8 compiles and executes JavaScript at lightning speeds mainly due to the fact that V8 compiles JavaScript into native machine code. In addition to lightning fast JavaScript execution, the real magic behind Node.js is the event loop. The event loop is a single thread that performs all I/O operations asynchronously. Traditionally, I/O operations either run synchronously (blocking) or asynchronously by spawning off parallel threads to perform the work. This old approach consumes a lot of memory and is notoriously difficult to program. In contrast, when a Node application needs to perform an I/O operation, it sends an asynchronous task to the event loop, along with a callback function, and then continues to execute the rest of its program. When the async operation completes, the event loop returns to the task to execute its callback. In other words, reading and writing to network connections, reading/writing to the filesystem, and reading/writing to the database, all very common tasks in web apps, execute very, very fast in Node. Node allows you to build fast, scalable network applications capable of handling a huge number of simultaneous connections with high throughput. Node comes with its own package management system, NPM. This will be used to utilise the various available libraries that will enable modular and manageable implementation of additional library frameworks to provide HTTP request libraries and middleware.

Express.js: Express.js is a Node.js web application server framework, designed for building single-page, multi-page, and hybrid web applications. It is the de facto standard server framework for node.js. Express is used in this system as the skeleton layout for the web application files and modules. It handles both client-side and server-side code and enables the use of third-party modules and middleware to handle things such as external HTTP requests to the Access Control Service.

JavaScript: JavaScript is the programming language used to deliver the functionality of the web application. All of the modules within the Express skeleton are written in JavaScript, e.g. the routes and app files.

Bootstrap: The web applications UI is built using Twitter's Bootstrap. This enables the use of the grid layout and responsive design.

jQuery: jQuery is used to provide the client-side functionality for the web application in the form transitions between visible and non-visible elements on click events

Amazon Web Services:

- EC2: Amazon's Elastic Compute Cloud provides resizable compute capacity for the deployment of the applications to be streamed and the access control service necessary to grant users access to the running applications.
- Appstream: Amazon's Appstream is the infrastructure that handles the rendering and streaming of the deployed applications.
- S3: Amazon's Simple Storage Service enables data storage and persistence of system users and the access rights.

1.4 Structure

Chapter 2: A list of the requirements and their use cases

Chapter 3: Conclusions drawn on the system.

Chapter 4: Potential growth of the project through further research and development

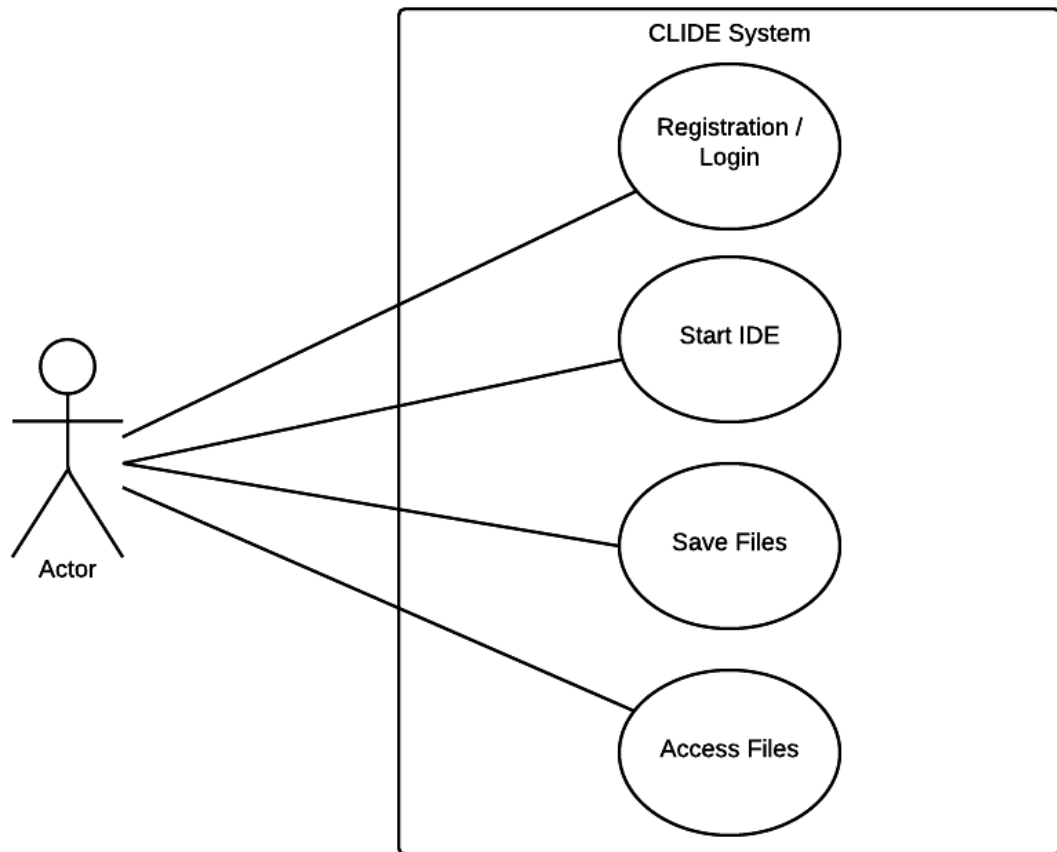
Chapter 5: References for material used to develop the system.

Chapter 6: Appendix

2 System

2.1 Requirements

2.1.1 Functional requirements



2.1.2 Requirement 1 <User Registration>

2.1.2.1 Description & Priority

Allows a new user to register with the system. This is a necessity for use of the system.

Priority 1.

2.1.2.2 Use Case

New User Registration / Login.

Scope

The scope of this use case is to allow for a new user to register / login with the application.

Description

This use case describes the registration / login process for the application.

Flow Description

Precondition

The new user has a stable internet connection, a valid email and access to a web browser.

Activation

This use case starts when a new user navigates to the applications' URL.

Main flow

1. The <Actor> enters their email address in the email field of the registration modal.
2. The <Actor> enters a chosen password in the password field of the registration modal.
3. The <Actor> clicks the register button to submit their credentials.
4. The <System> validates the inputs against a pattern.(See E1)
5. The <System> queries the DB to determine if the user exists.(See A1)
6. The <System> queries the DB to determine if the password matches that of the user.(See E2)
7. The <System> redirects the <Actor> to the home page.

Alternate flow

A1 : <User email already exists>...continues from Main Flow 5.

1. The <System> alerts the <Actor> that there is an existing account for this email address.(See A2)
2. The <Actor> must enter a different email address.
The use case continues at position 3 of the main flow

A2 : <User email already exists>...continues from Alternate Flow A1.1.

1. The <Actor> enters their email address in the email field of the login modal.
2. The <Actor> enters a chosen password in the password field of the login modal.
3. The <Actor> clicks the login button to submit their credentials.
The use case continues at position 4 of the main flow

Exceptional flow

E1 : <The Actor does not enter required input pattern(s)> ...continues from Main Flow 4.

1. The <System> alerts the <Actor> that the pattern does not match the supported pattern.
2. The <Actor> enters a supported pattern.
The use case continues at position 4 of the main flow

Termination

The system presents the home page.

Post condition

The <Actor> is now registered / logged in.

2.1.3 Requirement 2 <IDE Start-Up>

2.1.3.1 Description & Priority

A description of the requirement and its priority. Describes how essential this requirement is to the overall system.

Allows a user to select an IDE to run in the browser.

Priority 1.

2.1.3.2 Use Case

IDE Start-up

Scope

The scope of this use case is to allow an <Actor> to select an IDE to use and enable start-up of the chosen IDE.

Description

This use case describes the IDE selection process and start-up.

Flow Description

Precondition

The <Actor> has logged into the <System>.

Activation

This use case starts when an <Actor> has been redirected to the home page.

Main flow

1. The <Actor> selects an IDE to use.
2. The <System> responds to the selection event and requests an instance of the selected IDE.<See E1>
3. The <System> creates an instance of the selected IDE and returns it to the browser window for the <Actor> to use.

Alternate flow

A1: <not applicable for this use case>

Exceptional flow

E1 : <System does not return a running instance of the IDE>

1. The <System> alerts the <Actor> of the error and the status code, e.g. 404 Not Found, and redirects to the home page.
The use case continues from Main Flow 1.

Termination

The system presents the IDE in the browser window.

Post condition

The <Actor> has use of the selected IDE.

2.1.4 Requirement 3 <Save Files>

2.1.4.1 Description & Priority

A description of the requirement and its priority. Describes how essential this requirement is to the overall system.

Allows a user to save their working files.

Priority 2.

2.1.4.2 Use Case

User can save working files.

Scope

The scope of this use case is to allow an <Actor> to save their files so as they persist for all sessions.

Description

This use case describes the file saving process.

Flow Description

Precondition

The <Actor> has logged into the <System> and has a running instance of an IDE in the browser.

Activation

This use case starts when an <Actor> wants to save their working files.

Main flow

1. The <Actor> selects the save option in the IDE.
2. The <System> saves the files on the file storage system of the OS on the VM.
3. The <System> writes the files to a DB so as they persist for future sessions.(See E1)

Alternate flow

A1: <not applicable for this use case>

Exceptional flow

E1 : <System does not successfully write file to the DB>

The <System> alerts the <Actor> that it failed to write the file to the DB.

The use case continues from Main Flow 1.

Termination

The system saves the files on the file storage system of the OS on the VM.

Post condition

The <Actor> has use of the selected IDE with the files saved and wrote to the DB.

2.1.5 Requirement 4 <Access Files>

2.1.5.1 Description & Priority

A description of the requirement and its priority. Describes how essential this requirement is to the overall system.

Allows a user to access their saved working files.

Priority 2.

2.1.5.2 Use Case

User can access saved files.

Scope

The scope of this use case is to allow an <Actor> to access their saved files so as to continue their work.

Description

This use case describes the file access process.

Flow Description

Precondition

The <Actor> has logged into the <System>, has a running instance of an IDE in the browser and has previously saved files.

Activation

This use case starts when an <Actor> wants to access their saved files.

Main flow

1. The <Actor> selects the open file option in the IDE.
2. The <System> opens the files from the file storage system of the OS on the VM.
3. The <System> displays the files in the viewport of the IDE. (See E1)

Alternate flow

A1: <not applicable for this use case>

Exceptional flow

E1 : <System does not successfully open the file>

The <System> alerts the <Actor> that it failed to open the file.

The use case continues from Main Flow 1.

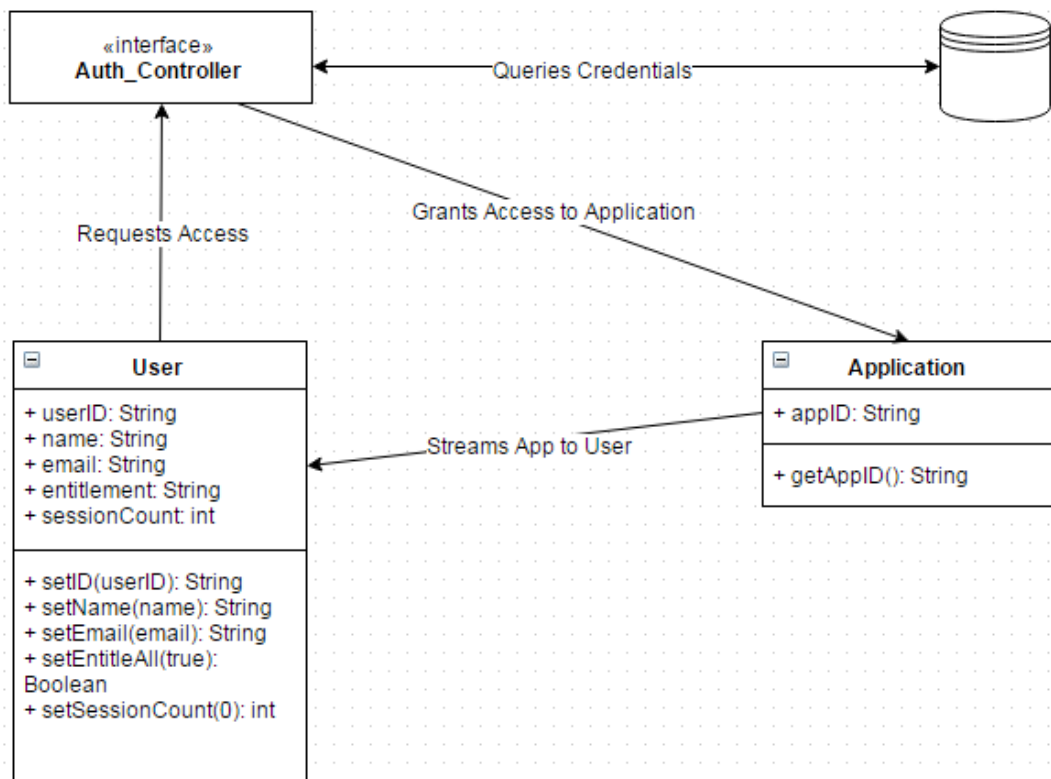
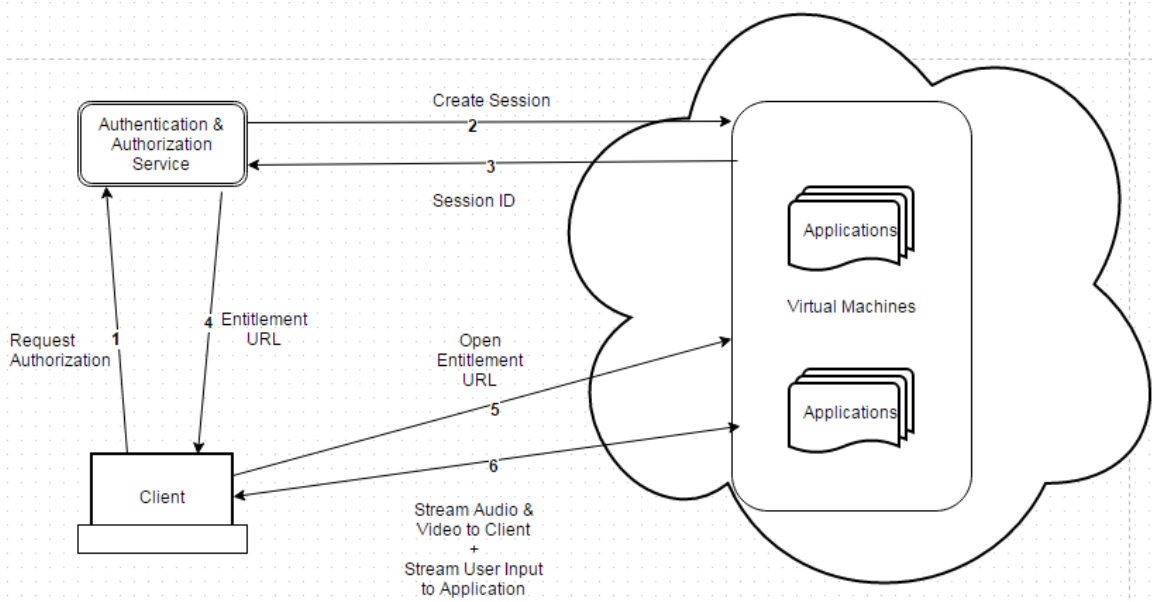
Termination

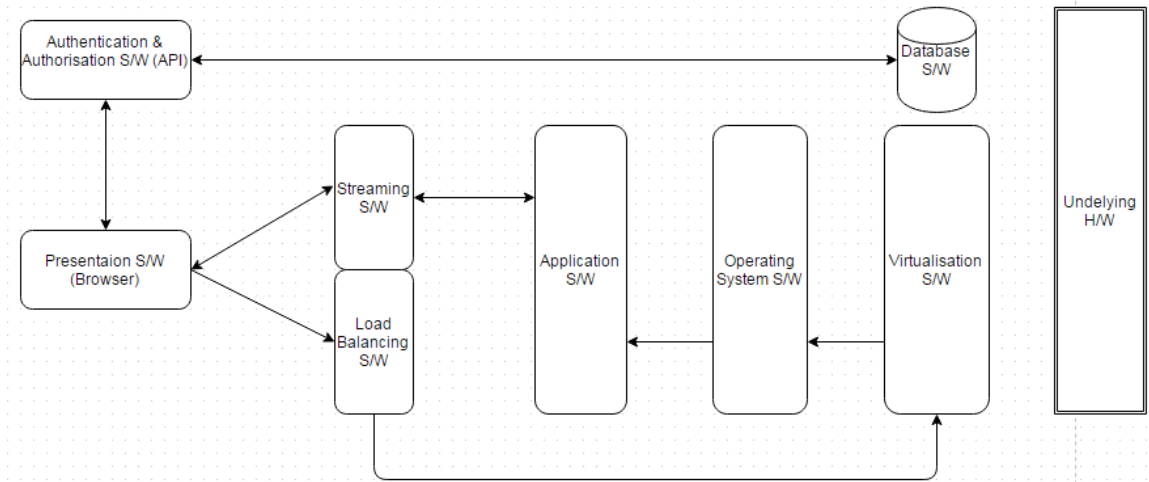
The system opens the files and the <Actor> is free to continue their work.

Post condition

The <Actor> has use of the selected IDE with the files opened.

2.2 Design and Architecture





2.3 Implementation

This system was implemented using some of the latest technologies, frameworks and methods that are currently being used in the industry. By using Node's package manager NPM it gave me access to a huge range of modules that are made available by the Open-Source community. An example of the use of NPM and the installed module is as follows:

```
$ npm install --save request      //get the request module from npm and save it
var request = require("request"); //require the module into the current context
var access_request = function(url, cb){ //define a function
    return request.post(url, function(err, resp, body){ //use the required
module
        if(err){
            cb(err, null);
        }else
            cb(null, body)
        }); //handle errors and return via a callback to the controller
    };
};
```

By using Express, it enabled me to implement the web application part of the system using the Model-View-Controller(MVC) approach. The MVC approach enabled me to organise my code base in a modular fashion which results in code that is easier to test, from both an integration and unit testing point of view.

Express also makes it possible to use templating engines for HTML rendering and passing data from server-side to client-side. In this application I have used the Hogan-Middleware templating engine and the mustache syntax to include data and partials where necessary. An example of the use of this middleware and the syntax is as follows:

```
//in the main app class of the project the middleware is defined  
app.set('views', path.join(__dirname, 'views'));  
app.set('view engine', 'mustache');  
app.engine('mustache', require(' Hogan-middleware').__express);  
  
//this then allows the following syntax to be used
```

```
{{> header}}           //include the header partial
```

```
{{> navbar}}           //include the navbar partial
```

```
<div class="container">
```

```
  <div class="row">
```

```
    <div class="col-md-12">
```

```
      <div class="textCenter">
```

```
        <h1>{{ title }}</h1> //pass in the title value defined in the server code
```

```
        <p>Welcome to {{ title }} @NCI</p>
```

```
      </div>
```

```
    </div>
```

```
  </div>
```

For the backend services of this system I used the AWS console to manage deployment of instances of applications and monitoring of running services. This involves setting up environments and stacks to tie all of the backend features together. This is conceptually quite complex but Amazons management console makes it achievable by removing many of the levels of abstraction.

2.4 Testing

This application was built using the Test-Driven-Development(TDD) methodology. TDD is the process of writing a test for every unit of code before the code is written in order to accurately test the expected behaviour of the code. The tests are written and ran to signal a failing test. The code is then implemented to satisfy the expected behaviour and the tests are re-run to signal a passing test. When the code passes the test it can then be refactored to simplify its implementation if possible or to remove any code duplication. This process is also known as red-green-refactor.

One of the key benefits of using this approach is its ability to mimic user interaction with the system through integration tests. Integration tests allow for the testing of the application as if it was being tested by an end user. Integration tests also allow for the testing of the expected layout and content of the application and how these are expected to respond to different user interactions. When integration tests are combined with unit tests it provides a complete testing solution with full test coverage for both client-side and server-side code.

In order to achieve complete test coverage and automation of these tests it was necessary to use a number of testing suites. The underlying test runner is the Mocha suite. Mocha is a feature-rich JavaScript test framework running on Node.js and in the browser, making asynchronous testing simple. Mocha tests run serially, allowing for flexible and accurate reporting, while mapping uncaught exceptions to the correct test cases. The assertion library I have used for unit tests is Assertive. Assertive aims to make the exact cause of breakage and intent of tests as fast and easy to spot as possible, with much attention paid to both the colour and alignment of expected and actual data, so you should be able to glean what you need immediately. For integration tests I've used the Testium library. Testium is an open-source integration testing library made available by Groupon. It was developed in-house to provide the best possible test coverage for Node applications that experience huge levels of traffic and user interaction. Testium is a complex library that has a large number of dependencies and is difficult to

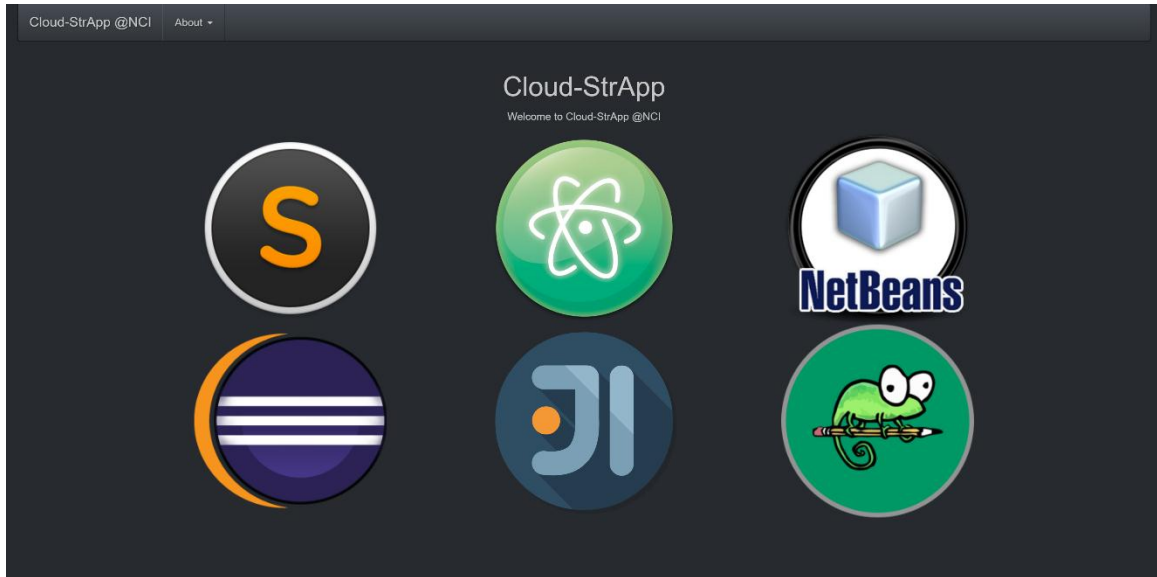
configure. The value of this overhead is realised with the completeness of the test coverage it provides. An example of an integration test using Testium is as follows:

```
var assert, injectBrowser;

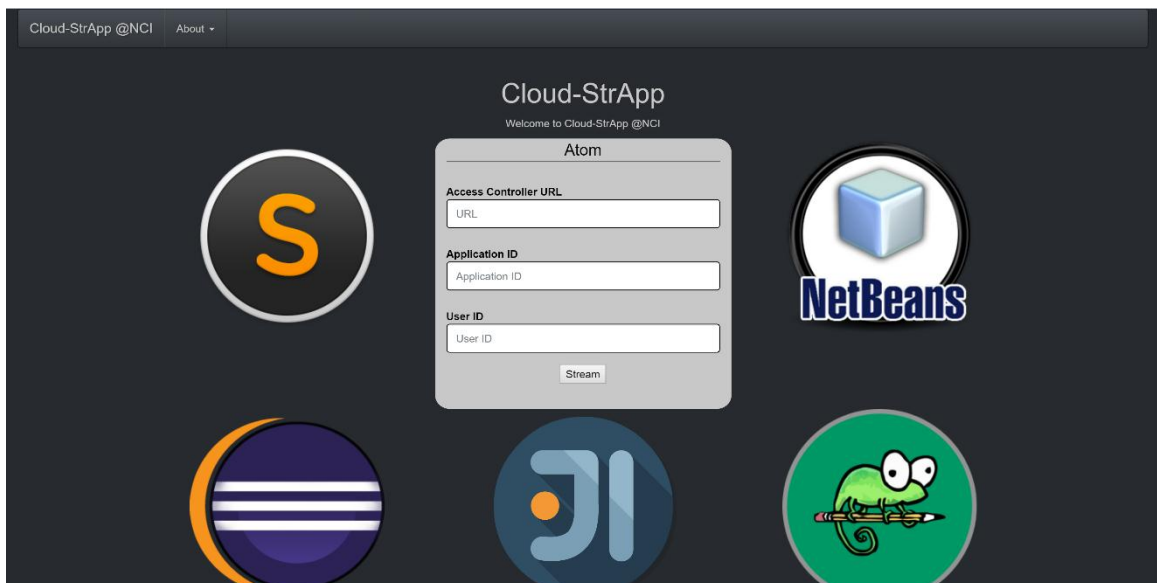
injectBrowser = require('testium/mocha'); //require a standalone browser
assert = require('assertive');           //assertion library of choice
describe('Index page', function() {     //describe what's being tested
  before(injectBrowser());              //start the browser
  before(function() {                   //condition to run before all tests
    this.browser.navigateTo('/');       //navigate to the root of the application
    return this.browser.assert.httpStatus(200); //check for a "Status OK"
    response
  });
  return it('includes a h1 element', function() { //describe a specific test
    return this.browser.assert.elementExists('h1'); //test for its existence
  });
});
```

2.5 Graphical User Interface (GUI) Layout

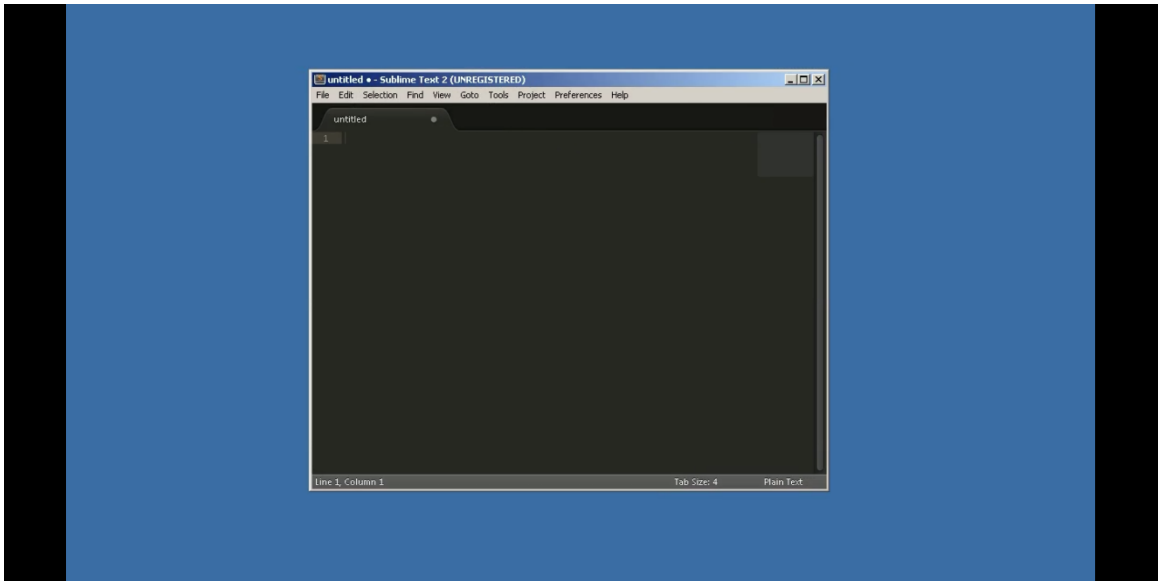
The application landing page. This page allows the user to select an application they would like to stream.



The application landing page with the credentials form displayed after a user chooses an application to stream.



The viewport of the streamed application returned to the user.



3 Conclusions

The idea for this project was to provide an alternative method of virtualised application delivery to students while enabling a reduction in their cost for personal hardware.

The idea was borne out of frustrations experienced while using the current delivery model and the need to have expensive powerful processing hardware in order to run the applications necessary to carry out the development work of a computer science student.

This system provides an example of a possible approach to address these issues. By leveraging the power of the cloud computing vendors' infrastructure and services it is possible to deliver virtualised and streamed applications via a custom built interface to students.

The advantages of this system are listed below:

- Cost effective
- Ease of use
- Always on
- Ubiquitously accessible
- Scalable
- Customisable

The disadvantage of this system is the possibility of vendor lock in. As this system relies on infrastructure provided by a cloud vendor for hosting, etc. it reduces the degree of flexibility at the backend level. The possible option to alleviate this disadvantage would be to develop the technologies locally and deploy them on a private cloud infrastructure which would be a managed and monitored onsite. This method, although alleviating a possible disadvantage, removes the cost effective advantage of using managed services.

4 Further development or research

With greater resources and time for research it is possible that this project could lead to a viable business venture. It would be necessary to spend time to fully understand the best practices for virtualisation and delivery of streamed applications in order to determine if the backend services of this project could be independently developed, managed and delivered in a financially viable way.

In its current state this project offers a viable option to address the issues identified as the basis for the requirements that define the scope of this project. Ideally the mass adoption of this approach would prove it to be a successful venture with a long term view to providing localised managed services of a proprietary backend system to all educational institutes.

The viability of this would first need to be established by way of detailed market research.

5 References

Amazon Web Services Available at: https://aws.amazon.com/?nc2=h_lg
(Accessed: 1 October 2015).

StrongLoop Express - Node.js web application framework. Available at:
<http://expressjs.com/> (Accessed: 25 September 2015).

Mocha - the fun, simple, flexible JavaScript test framework (no date) Available at:
<http://mochajs.org/> (Accessed: 19 October 2015)

Massa, S. and Groupon *I-tier: Testium.* Available at:
<https://engineering.groupon.com/2014/misc/i-tier-testium/> (Accessed: 20 January 2016).

6 Appendix

6.1 *Project Proposal*

Project Proposal

CL-IDE

Student Name: Wayne Flanagan

Student Number: x12112461

Student Email: x12112461@student.ncirl.ie

Degree Programme Name: BSC (Hons) in Computing

Specialisation: Cloud Computing

Date: 02/10/2015

1 Objectives

The objectives of this project are multifaceted:

The primary objective is to develop a working cloud based system that allows users to access a dashboard, via a web browser, that acts as an interface to a number of Integrated Development Environments (IDE's).

The underlying architecture will consist of:

- 1) PaaS (likely to be AWS)
 - a) A server or "virtual host" with the underlying OS
 - b) VMware to manage the virtualisation and create instances of IDE's as needed

From the interface the user will be able to choose which IDE best suits their development needs and launch it in the browser.

- 2) SaaS
 - a) Web browser user interface / dashboard
 - b) Multiple in browser IDE's

The system will also incorporate a Network File System (NFS) to allow users to store and access their development files in the cloud.

- 3) NFS
 - a) Accessible storage

The objective being that this functionality will allow the system to be truly cloud based with only the need for a thin client with internet access to enable ubiquitous use. This will enable the user to use the system from any machine without the concern for third party storage.

The secondary objective is to understand and develop the skills needed to build and maintain this type of cloud based system so as to prepare for a job in this area. This will involve multiple learning outcomes:

- 1) PaaS
 - a) Configuration
 - b) Deployment
- 2) SaaS
 - a) Implementation
 - b) Delivery

- 3) NFS
 - a) Design
 - b) Configuration
 - c) Maintenance

Delivery of these somewhat complex and extensive objectives will produce a project that will be demonstrative of my understanding and ability in the cloud specialisation.

2 Background

The motivation for wanting to build this project stems from both my experience in college and on work placement.

College perspective:

After three years at college and many hours spent frustrated by the lag of using Citrix both onsite and remotely, while trying to multitask and meet deadlines under pressure, I wanted to find an alternative to desktop virtualisation.

The Problem:

While being a student at college I have witnessed the overhead and latency caused by the demand of an IDE on the resources of a distributed virtual desktop infrastructure. Being on the user end of a crashed virtual desktop during an exam I have experienced first-hand what it's like when it goes wrong. These issues along with other restrictions that have to be enforced in a college IT environment led me to think about what the best possible solution was to alleviate the demand on resources while possibly delivering a cost saving.

Solution:

With this in mind I had been considering the tools I have been using outside of college for personal development projects and collaboration. With the likes of Office 365 and Google drive available for use practically anywhere compiling reports, drafting and editing essays and assignments and working on general text based projects was all quite easy to do remotely. This wasn't the case when it came to development projects.

Although there are currently independent cloud based IDE's available, if they are not the ones you are familiar with or prefer then it comes down to a time issue and whether you have it available to familiarise yourself with the new IDE.

Although the download and installation of an IDE is somewhat straight forward, this is not always possible due to machine or other restrictions. From this conundrum I came up with the idea that it would be useful if there was access to a selection of IDE's in one central location and accessed via a browser and without the need for desktop virtualisation which has been the root cause of the initial problem that led to this project idea.

Business Case:

Although there would be an initial outlay for the provision of cloud based services, it would result in the ability to consolidate the currently fragmented infrastructural services such as the student drive and Moodle into one central storage and hosting area. This approach would also lead to cost savings for the college as the need for Citrix would become redundant. This approach would also allow for a flexibility that the college does not currently have and enable savings to the college during off-peak times as they could scale down their cloud resources. Overall I believe that this model would apply to all educational institutions.

Work Placement perspective:

During third year I interned at Groupon in a team of 6 developers including myself. As I was new I asked the team for a suggestion on what IDE to use that would be suitable for the technologies that I would be using. It came down to two IDE's with the decision being made on the one that placed the least strain on my machines resources.

The Problem:

I found this a bit unusual that the deciding factor was based on the machines spec considering Groupon supplies its developers with top spec Apple Mac development machines. It simply came down to the developers desire to reduce the strain on their machines resource's, so as they could continue multitasking without lag issues, something which seems to be an unavoidable character trait of developers.

The Solution:

The team I worked in is one of the smallest at Groupon. They employ hundreds of engineers / developers that work in different programming languages and frameworks every day. The one commonality is that they all have their own preferred IDE for each programming language and framework. In my mind the solution to this was similar to the one for a college. Groupon should subscribe to a cloud service that provides a selection of IDE's in one central location and accessed via a browser without the need for desktop virtualisation.

The Business Case:

This would enable Groupon, and other tech companies, to provide their developers with the option of which IDE to use all the while allowing themselves to provision machines that are mid-spec, thin clients, and less expensive. Having this cloud IDE access would also enable developers on call to view code on their mobile device via its browser in turn possibly saving the company money by limiting downtime.

I had a common problem but with different root causes so how do I address both? This has led me to my current suggestion and project idea:

CL-IDE

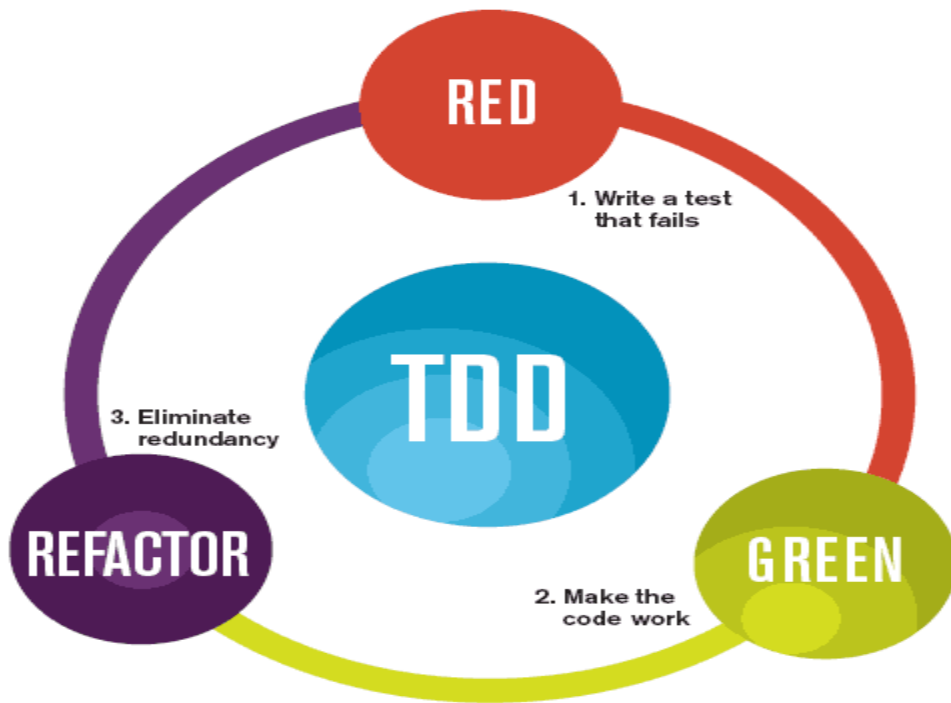
3 Technical Approach

The research for this project will involve reading books, eBooks and online resources as well as the seeking out of examples of similar implementations to understand how the technologies and architecture are combined.

As this is a personal project the requirements will be defined by me with some low level consultation with work colleagues from my internship. They will consist of both functional and non-functional requirements. They will be gathered from the perspective of what is determined as useful and necessary in a college and work environment.

I plan to implement this project using the Test Driven Development (TDD) process and some of the principles from the Agile development methodology. I will conduct Sprint Planning sessions and complete three week Sprints for the duration of the project. I will be using Smartsheets as my project planning and management tool.

My reasoning for using the TDD approach is that it will allow me to better manage the quality of my code. It will enable me to eliminate code redundancy and allow me to minimize the occurrence of bugs in my code by testing, building, testing again and refactoring small units of code in an iterative manner.



The mantra of Test-Driven Development (TDD) is "red, green, refactor."

4 Special resources required

Research resources:

- Books
- E-books
- Online docs
- IEEE

Software resources:

- IDE's / Text Editor's
- Office Suite
- VMware
- VM's
- OS's

Hardware Resources:

- Laptop
- Servers
- PaaS subscription

5 Technical Details

Framework: Node.js

Package management: Node Package Manager (NPM)

Server-side language: CoffeeScript

Client-side language: CoffeeScript, jQuery, AJAX

Front-end framework: Bootstrap

Web Page Mark-up: HTML5, Mustache

Web Page Styling: CSS3

Testing frameworks: Assertive, Mocha, Testium

Version control: Git

6 Evaluation

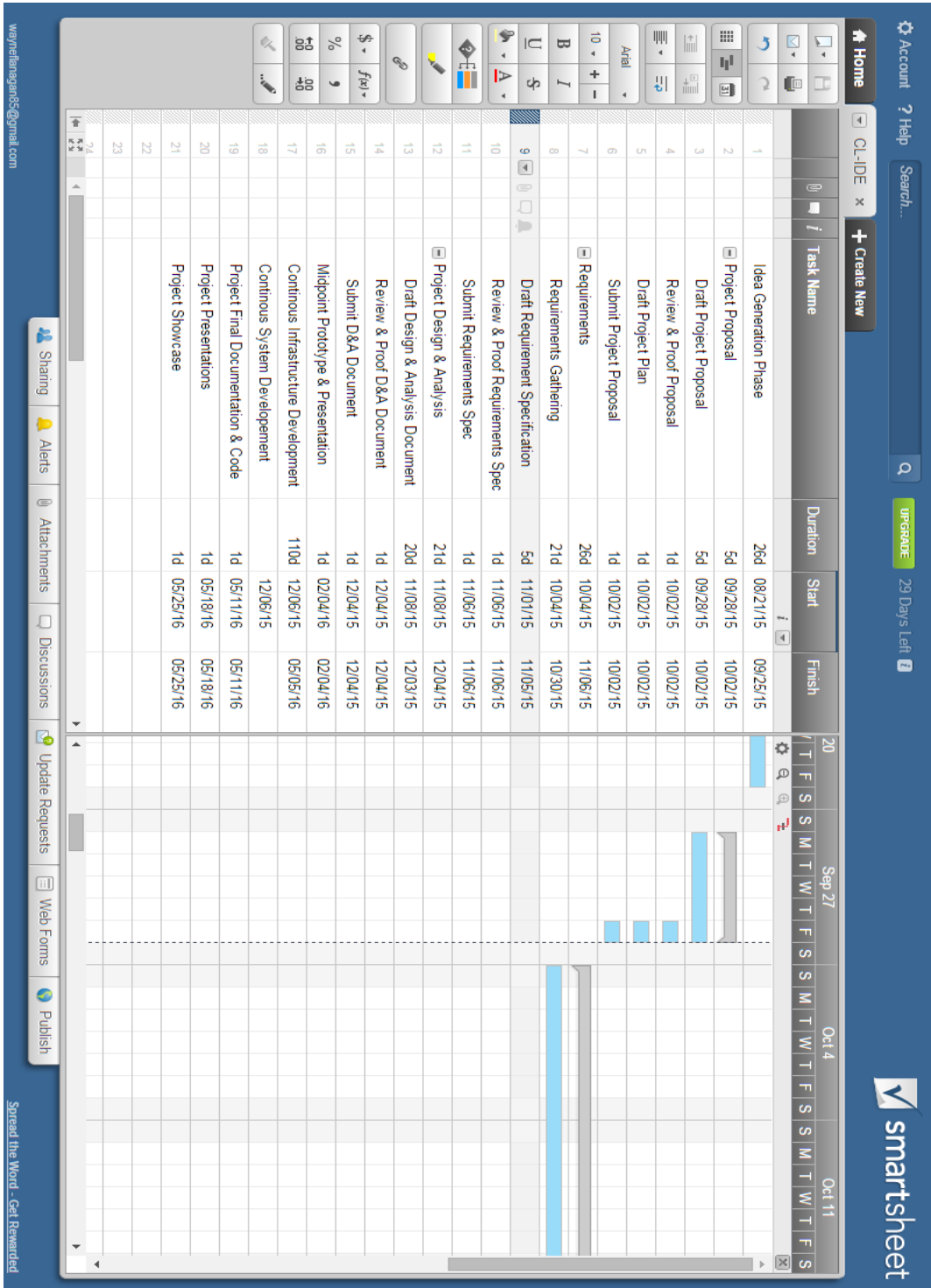
As mentioned above I will be using the TDD approach and so all of my code will be tested as it is developed. This will ensure test coverage across my application code. The testing will take the form of client-side and server-side integration and unit testing. The benefit of this approach is that each component will have to pass its unit tests before moving onto the next component. The integration tests will then highlight any errors or bugs when dependencies are established between components.

The above approach should lead to the System Testing being relatively straight forward. In order to ensure this I plan to provide a group of test users with a description of the expected behaviour of the system and a feedback form. The feedback from these tests will determine whether or not the system is ready for release.

Wayne Flanagan 02/10/2015

Signature of student and date

6.2 Project Plan



6.3 Monthly Journals

Reflective Journal

Student name: Wayne Flanagan

Programme: BSc in Computing

Month: September 2015

My Achievements

This month I was able to narrow down my project ideas and decide on the most viable option that I believe will allow me to demonstrate my current ability and the new skills I will attain during the course of my final year in my chosen specialisation of cloud computing.

My contributions to the project so far included researching the required technologies and architecture needed to deliver my project. I then drafted and submitted my project proposal and project plan.

My Reflection

I feel I have proposed a challenging project that will allow me to showcase my knowledge and ability. With the correct time management and application to the required work I should be able to deliver this project on time and within scope.

Intended Changes

Next month I will arrange to meet with my assigned project supervisor to get feedback on my proposal and possible suggestions on user requirements.

I will define the functional and non-functional requirements for my project and draft and submit the Requirement Specification Document.

I need to continue my research and understanding of some of the technologies I will be using, and so I will do so over the next month.

Supervisor Meetings (yet to be assigned)

Date of Meeting:

Items discussed:

Action Items:

Reflective Journal

Student name: Wayne Flanagan

Programme: BSc in Computing

Month: October 2015

My Achievements

This month, I met with my project supervisor, Eamon Nolan, and discussed my project proposal in detail. This was a good opportunity to go through the specifics of my project and helped my project supervisor establish an understanding of exactly what it is I plan to deliver. The feedback was positive and it made me confident that my project should put me in a position to do well in this module.

I have drafted the first version of the requirements specification document. This was a helpful task as it helped me focus in on the key aspects of my project and clearly define them.

Although I have yet to write any code, specific to this project, my Cloud Computing module project has allowed me to do some research and write some reusable code that I can now apply to my final year project.

I also met with my mentor from my internship and discussed the architecture of my project. His suggestions were really helpful and have given me options for the structure of my project.

My Reflection

October was a very busy month in terms of college work and deliverables. The quantity of work combined with my desire to deliver good quality assignments meant that I had limited time to spend on research for my project. This is something I will have to manage carefully going forward.

Intended Changes

Next month I will arrange to meet with my assigned project supervisor again to get feedback on my requirements specification document.

I also intend to do some more research as I have to gain a better understanding of the technologies needed to make my project work.

I will also try to start to build the first version of the frontend of my project.

Supervisor Meetings

Date of Meeting: 2nd November 2015

Items discussed: Project Proposal, Requirements Specification Document.

Action Items: Requirements Specification Document.

Reflective Journal

Student name: Wayne Flanagan

Programme: BSc in Computing

Month: November 2015

My Achievements

This month I was able to work on the draft for my Design and Analysis document. This document proved both challenging and beneficial in equal measures. The work I had previously done on my Requirements Specification document formed a good basis as a starting point for this document but there was a need to delve deeper into the proposed architecture of my project, in order to better understand the necessary individual components of my system, so as to be able to provide a higher level of abstraction to any reader of this document.

This led me to do some further research on the key component parts of my project, which in turn led me to do some evaluation of third party services to determine which I could utilise and would provide a best to achieve delivery of my project as proposed.

I investigated some UI best practices and noted the ones that were applicable to my project. This is something I intend to revisit over the development lifecycle of my project.

My Reflection

As was the case with October, November was an extremely busy month. I had a Report for Change Management to complete and the delivery of my Cloud Computing project due, which was worth a considerable chunk of the module grade. These ongoing course deliverables create a difficulty in terms of time management and balanced effort across all modules. In saying that I am aware that this is just the nature of it and something I need to be more conscious of.

Intended Changes

Next month, I will try to start the preparations for my mid-point presentation and decide on what work to complete for the prototype. This will have to take place in parallel with preparing for my exams in January and time management is going to be important again.

Supervisor Meetings

Date of Meeting: Email correspondence 16/11/15 – 18/11/15

Items discussed: Requirements Specification document

Action Items: Continue to add to Requirements Specification document.

Reflective Journal

Student name: Wayne Flanagan

Programme: BSc in Computing

Month: December 2015

My Achievements

This month I spent time preparing for my final project presentation for NPDS and studying for my January exams.

Development work on my project was at a minimum and I spent very little time researching.

My Reflection

I felt December was a bad month in terms of progress and overall effort.

Intended Changes

Next month, after my exams, I will try to get back on track with my project work. I need to be more decisive about my choices of third party technologies in order to increase the speed of development work so as to have a prototype ready in time for my mid-point presentation.

Supervisor Meetings

Date of Meeting: 07/12/15 (Missed Meeting due to Work Commitments)

Items discussed:

Action Items:

Reflective Journal

Student name: Wayne Flanagan

Programme: BSc in Computing

Month: January 2015

My Achievements

This month was greatly reduced time wise by my exams which ran until the 14th of the month. Although this was the case I was able to find some solutions to problems that had arose from my design and analysis work and research so far. These problems mainly being related to the streaming protocols that I will have to implement in order to deliver the IDE viewports over a network to my browser interface. This will continue to be a trial and error process to determine the best method.

My Reflection

Although I was not able to apply a full month's effort to my project this month, due to my exams, I still felt comfortable with what I had achieved in terms of identifying possible solutions to problems that where becoming a concern in terms of project viability. This has led to a weight being lifted and I am looking forward to pushing on with my prototype development for the Mid-Point presentation.

Intended Changes

Next month I will focus on preparing for the Mid-Point presentation by getting my documentation up to date and doing more work on my prototype. After the presentation I will consider any feedback I get and plan to address any suggested improvements or changes to my project.

Supervisor Meetings

Date of Meeting: None in January

Items discussed:

Action Items:

Student name: Wayne Flanagan

Programme: BSc in Computing

Month: February 2016

My Achievements

This month I presented my work so far, as a prototype, along with my documentation for my Mid-Point presentation. I felt the presentation went well, bar a small glitch in my prototype demonstration, and the feedback was quite positive. In terms of my documentation the feedback wasn't as positive. This was quite disappointing as I thought I had prepared my documentation to a good standard but it acts as an indicator of what's expected and I now know what I have to work on in that regard.

My Reflection

It is a relief to have the Mid-Point presentation out of the way and to have received the feedback. Although it is different to the setup from previous years, in that there are marks awarded, I thought the presentation was worthwhile and a good mile marker for the progress of my project. It has drawn attention to the remaining time to complete my project and helped me refocus.

Intended Changes

Next month I will try to complete as much as possible of the remaining development work so as to allow for plenty of time to work on my documentation. This again will be dependent on my ability to manage my time between other module projects that have deliverable deadlines approaching.

Supervisor Meetings

Date of Meeting: 16th February 2016

Items discussed: Mid-Point presentation feedback.

Action Items: Improve documentation.