National College of Ireland

Postgraduate Diploma in Cloud Technologies

2011/2012

Louise Harte

John Morris

Dean O’Gorman

Tony O’Keeffe

**Map-A-Doodle**

Project in

Enterprise Framework applications



I hereby certify that this material, which I now submit for assessment of the programme of study leading to the award of Master of Science in Web Technologies is entirely my own work and has not been taken from the work of others save and to the extent that such work has been citied and acknowledged within the text of my work.

Signed:

Date:

Student Number:

**Table of Contents**

[1. Introduction 4](#_Toc330412804)

[1.1. Background Research and Investigations 4](#_Toc330412805)

[1.2. Project Plan 4](#_Toc330412806)

[2. Software development methodology employed 5](#_Toc330412807)

[3. Requirements Analysis 6](#_Toc330412808)

[4. Use Cases 7](#_Toc330412809)

[5. Architecture/Design approach 8](#_Toc330412810)

[6. Models 9](#_Toc330412811)

[7. Implementation of particular OOP constructs 10](#_Toc330412812)

[8. Design patterns and Architectural patterns 11](#_Toc330412813)

[9. Cross-cutting issues 12](#_Toc330412814)

[10. Security 13](#_Toc330412815)

[11. Configuration of the application 14](#_Toc330412816)

[12. Scalability of the application 15](#_Toc330412817)

[13. Testing 16](#_Toc330412818)

[14. Conclusion 18](#_Toc330412819)

[References 19](#_Toc330412820)

# Introduction

Map-A-Doodle is an MVC4-based project written in C# that allows the user to find points of interest on a map…

## Background Research and Investigations

The motivation behind this group project was to design, produce and deploy a fully working application that incorporated a mapping interface.

Blah, blah…

## Project Plan

The application is designed to be scalable. At present, data is being used from New York and Dublin but this can be changed at any date

# Software development methodology employed

C#

MVC4

ASP.NET

Classes – models

Parsers

Map API

Steps

1. Created classes\models
2. Created the oul database
3. Created Parsers
4. Testing
5. Map interface
6. API

Design pattern??

# Requirements Analysis

Blah…..

# Use Cases

Blah…

# Architecture/Design approach

5 layer model…..

# Models

(Class Models / Data Models etc.)

# Implementation of particular OOP constructs

Blah….

# Design patterns and Architectural patterns

Data parser could be a strategy pattern. Singleton maybe??

# Cross-cutting issues

How cross-cutting concerns have been handled . If something went wrong with the project, how did you handle it?

# Security

Security of the application

# Configuration of the application

It is XML, C#, etc…

# Scalability of the application

Yes, very. From the outset it was designed to be a scalable app…

# Testing

Testing Approach (in terms of both functional and non-functional requirements)

**What is it?**

Simply put unit testing is ‘Methods written with the sole purpose of testing another method against a specific scenario. Each test should be readable, independent, and provide a consistent response.

**Which Framework?**

There are many unit testing frameworks available and even more discussions online about which one is the best. NUnit and MSTest are just two that are available, but for this blog I am going to use MSTest as it is built into Visual Studio and Team Foundation Server (TFS). But you need to decide which framework best suits your requirements.

**Misconceptions**

Many people will say that unit testing is not a replacement for functional tests, they add additional time to the development phase or just that due to their application type that unit testing isn’t suitable.

All these arguments can be countered.

* Unit testing isn’t a replacement for functional testing. This is true; you should never consider unit tests as a replacement. Its purpose is completely different, but it will help in making the functional testing process more focused.
* Unit tests add additional time to the development phase. This is true to a certain extent, but the time should be returned later. For example the functional testing can be more focused, regression testing will pick up test failures that might not be picked up until later on.
* Unit tests aren’t applicable for my application type. This occurs because of the development style. For example where the business logic is in the presentation layer or where the code is reliant on external dependences, such a database.

Benefits

There are many benefits with creating unit tests

* Defined, consistent and controlled tests.
* Repeatable tests, which can be rerun in seconds.
* Easy to do regression testing.
* TFS can be setup to automatically run tests on check-in and therefor issues can be caught before functional testing is carried out.
* Code is documented by the tests.

Lets Think!

Before we write any code you need to understand that unit testing requires more than just writing some code. You will probably need to change your programming style as well. It is important that you develop you code so it can be tested, this essentially means that your classes / methods should be independent and have a defined role. It is very hard to test a method that is reliant on a webservice or a database.

# Conclusion

Jazus, this was tough

# References

[1] W3 Schools website (JavaScript) - <http://www.w3schools.com/js/>

Accessed: 12th November 2011

[2] TechTerms.com. <http://www.techterms.com/definition/tooltip>

Accessed: 12th November 2011

[3] jQuery.com. <http://jquery.com/>

Accessed: 15th November 2011

[4] Garrett, J. J., Ajax: A New Approach To Web Applications.

<http://www.adaptivepath.com/ideas/ajax-new-approach-web-applications>

Accessed: 29th November 2011

[5] Flanagan, D., 2011. JavaScript: The Definitive Guide. O’Reilly press

[6] Ulman, C., 2007. Beginning Ajax. Wrox

[7] Ruby, S., Thomas, D., Hansson, D. H. (2011). Agile Web Development With Rails, 4th edition. Pragmatic Bookshelf

**Additional Resources**

Tutorials:How jQuery works. <http://docs.jquery.com/Tutorials:How_jQuery_Works>

Accessed: 30th November 2011

Wilton, P., McPeak J., 2009. Beginning JavaScript. 4th Edition. Wrox