National College of Ireland

BSc in Computing

2016/2017

Nathan Ryan

X13448212

nathan.ryan@student.ncirl.ie

NCIProjects

Technical Report

Declaration Cover Sheet for Project Submission

SECTION 1 *Student to complete*

Name: Nathan Ryan
Student ID: X13448212
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 (summarized 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:

NB. If it is suspected that your assignment contains the work of others falsely represented as your own, it will be referred to the College's Disciplinary Committee. Should the Committee be satisfied that plagiarism has occurred this is likely to lead to your failing the module and possibly to you being suspended or expelled from college.

What constitutes plagiarism or cheating?

The following is extracted from the college's formal statement on plagiarism as quoted in the Student Handbooks. References to "assignments" should be taken to include any piece of work submitted for assessment.

Paraphrasing refers to taking the ideas, words or work of another, putting it into your own words and crediting the source. This is acceptable academic practice provided you ensure that credit is given to the author. Plagiarism refers to copying the ideas and work of another and misrepresenting it as your own. This is completely unacceptable and is prohibited in all academic institutions. It is a serious offence and may result in a fail grade and/or disciplinary action. All sources that you use in your writing must be acknowledged and included in the reference or bibliography section. If a particular piece of writing proves difficult to paraphrase, or you want to include it in its original form, it must be enclosed in quotation marks and credit given to the author.

When referring to the work of another author within the text of your project you must give the author's surname and the date the work was published. Full details for each source must then be given in the bibliography at the end of the project

Penalties for Plagiarism

If it is suspected that your assignment contains the work of others falsely represented as your own, it will be referred to the college's Disciplinary Committee. Where the Disciplinary Committee makes a finding that there has been plagiarism, the Disciplinary Committee may recommend

- That a student's marks shall be reduced.
- That the student be deemed not to have passed the assignment.
- That other forms of assessment undertaken in that academic year by the same student be declared void.
- That other examinations sat by the same student at the same sitting be declared void.

Further penalties are also possible including

- Suspending a student college for a specified time.
- Expelling a student from college.
- Prohibiting a student from sitting any examination or assessment.
- The imposition of a fine.
- The requirement that a student to attend additional or other lectures or courses or undertake additional academic work.

Technical Report

Document Control

Revision History

Date	Version	Scope
11/12/16	1.0	Requirements Specification + Initial Analysis
20/02/17	2.0	Analysis & Design
04/05/17	3.0	Testing

Distribution List

Name	Title	Version
Eamon Nolan	Supervisor	1.0
Eamon Nolan	Supervisor	2.0
Eamon Nolan	Supervisor	3.0
Eugene McLaughlin	Lecturer	3.0

Related Documents

Title	Comments
-	-

Table of Contents

Executive Summary

The main objective of the NCIProjects assignment was to develop a web-based portal for use within National College of Ireland wherein 4th Year Students can upload all their personal details and information regarding their final year project in a specified format. The project was undertaken to replace the outdated and inefficient methods of developing the Project Showcase Booklet used in the annual Project Showcase in the college and is also sent to employers and guests who attend Project Showcase.

This project was also a study into Usability with careful consideration given to design patterns and users in regards to how they will interact with and use the system.

The project as developed to make the process of creating the Showcase Booklet easier and efficient for both college staff and students.

The web portal was developed using with ASP.NET MVC framework, SQL Server and Azure web hosting amongst other technologies and can be found at http://nciprojects.azurewebsites.net/.

Areas of Interest: Teaching, Education, Allocation, Usability, Web Technologies.

1 Introduction

1.1 Background

Every year NCI and most colleges in the country will have a day to showcase their students' Final Year Projects. With this showcase a booklet must be prepared to inform guests of what the project does, the work done and what technologies were used in developing the project. This booklet is a great way to show off an individual's work done throughout their final year and it is great for the college as it shows employers and potential students the diverse range of skills on offer from the college.

1.2 Aims

Every year when the Project Showcase Booklet must be assembled, staff at NCI must contact the students to get details of their project such as descriptions, technologies used and profile pictures to be used in the booklet. The problem with this approach is that it may be a long process trying to contact the students to provide this information and even more time to correct any errors students may have submitted such as the descriptions being too long, too many spelling errors or their picture not being in the right format/size.

NCIProjects aims to make this process as efficient as possible for staff and students through the development of a web portal where Final Year Students can upload all details of their project in a timely and straightforward manner. This project aims to be a helpful addition for the staff in the college and shorten the process in assembling the Project Showcase Booklet.

NCIProjects also aims to remove the ambiguity students face when delivering project details for use in the Project Showcase. This portal will guide users through the process of entering short and long summaries about their projects for approval by an administrator (Lecturer). Short summaries will contain 40-60 words with the student's picture and LinkedIn profile link. A longer summary will contain 200 words with the student's picture and technologies used in developing the project.

The system will ensure data is entered correctly using strict guidelines such as number of words entered, spellcheck, list of technologies used, required picture format etc...

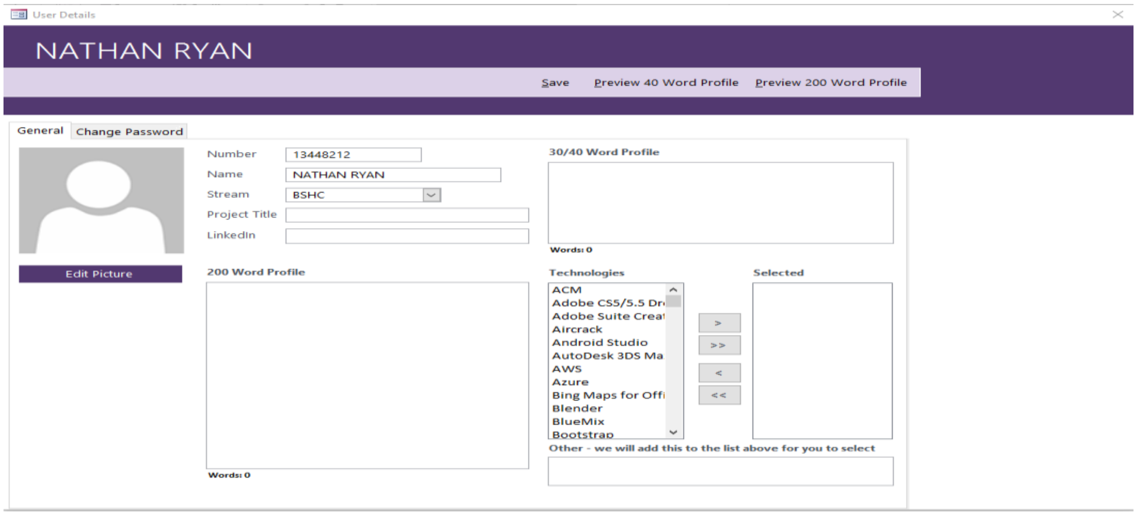
When all submissions have been approved by an administrator, the Showcase Booklet will be produced by the system for use in the Project Showcase.

Students can also preview their submissions in either the 40-word or 200-word booklet.

1.3 Research

Because this project is mainly an internal project to be used within the college itself there was not many resources from which to research and reference from e.g. finding a market, target audience, similar projects in existence etc. The main part of my research was taking the information given from the user requirements and evaluating the current system used by the college for the project showcase.

The process for managing Final Year Projects has been done using Microsoft's Access Database Management System. Students had to log in through the college's network to find the Access program in which they would use to submit their information.



NATHAN RYAN

Number: 13448212
Name: NATHAN RYAN
Stream: BSHC
Project Title:
LinkedIn:

Words: 0

Words: 0

During ongoing research, several students were asked about how they used and interacted with the current system. Feedback received included comments such as availability of the system – it was only available on the college network; students could not always access the system at home to edit their details. Students also wanted a more efficient way to check the status of their submission (approved or rejected).

1.4 Technologies

.NET MVC5: the framework for building the web portal. ASP.NET allows this project to be scalable and allows for separation of concerns with the Model-View-Controller.

Web Forms: another aspect of the .NET framework. Web Forms differ from MVC in that they can be standalone applications but may also be included within a MVC application. In this project, Web Forms provide the crucial functionality for producing the Showcase Booklet pdf.

C#: the programming language used to create the .NET platform. The Controllers (or logic) and Models of the application is developed using C#.

SQL Server Database: Microsoft's Relational Database Management System. Used to hold all data relating to Students and Project Submissions.

Razor: This is syntax based on the C# language used to provide logic to Web Pages (View).

Azure: Microsoft's cloud computing platform. The project will be deployed through the Azure service.

HTML5: the fifth version of the HTML standard, it is a mark-up language for the structure and presentation of content on the Web. Used to create the front end of the application with the Bootstrap framework.

Bootstrap: open-source JavaScript front-end framework used in designing web sites.

Git: used for Version Control. The project is hosted on GitHub in a private repository.

MSTest: testing framework used for conducting Unit Tests.

1.5 Structure

This section will provide a brief overview of each chapter.

The Introduction chapter gives a summary of the background of the project and it's aims.

The System chapter describes the project requirements, design and architecture, implementation of core functionality, the graphical user interface and testing carried out.

The Conclusion chapter provides a summary of what I have learned throughout the development of this project and the opportunities and limits of this project.

The Further Development or Research chapter describes how this project could develop in the future and its direction going forward.

References will outline all resources used in the development of this application.

The Appendix includes all other relevant information to the project including the Project Proposal and Monthly Journals.

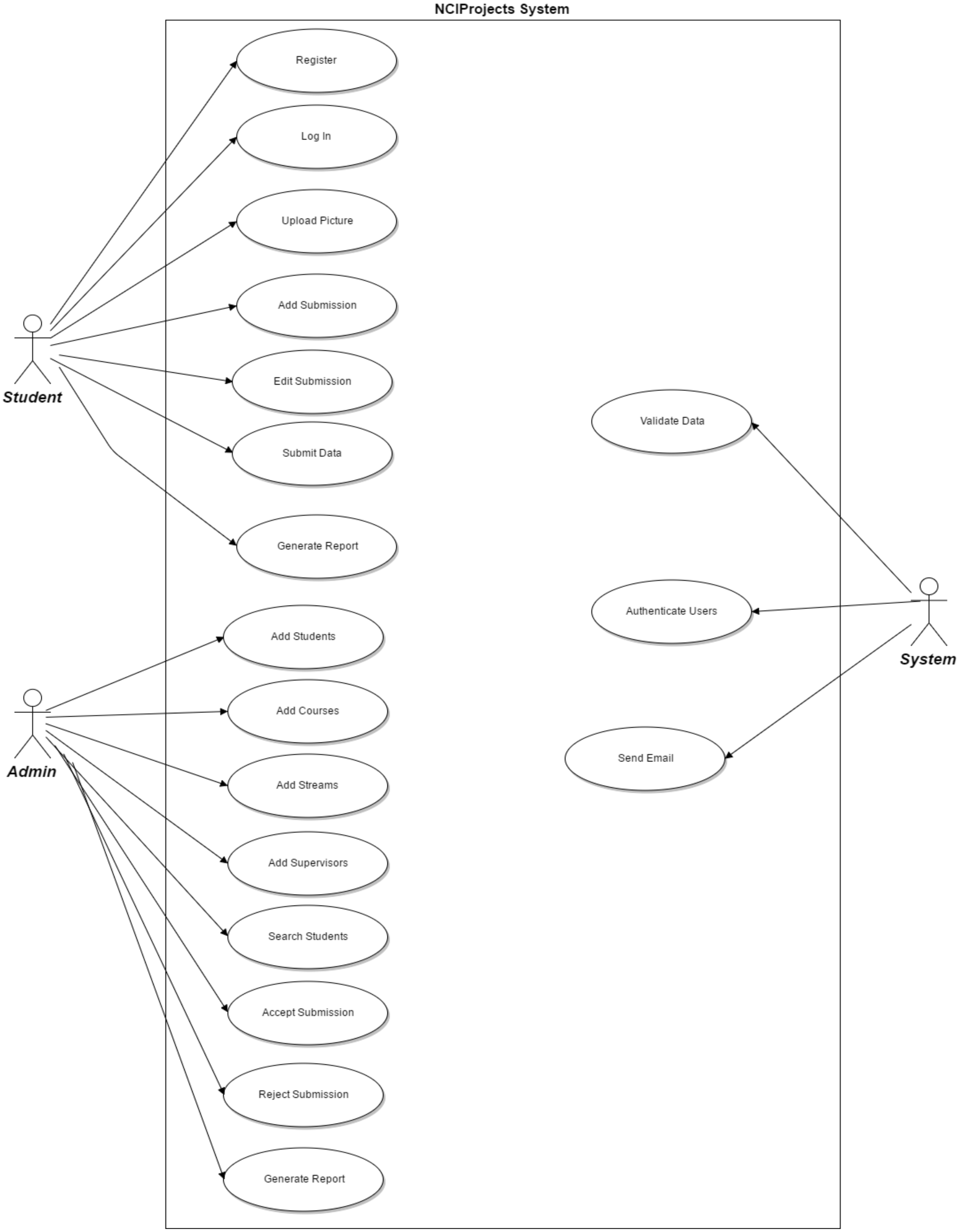
2 System

2.1 Requirements

2.1.1 Functional requirements

ID	Requirement	Comment
FR_01	The system shall allow Admin User to register Student details to the application.	Admin shall be able to pre-load Student's details into system before Students register.
FR_02	The system shall allow a student user to register to the application.	All final year students must register to use the system.
FR_03	The system shall allow a student user to log in to the application.	Students must log in to system to submit data.
FR_04	The system shall allow a student user to input project data.	Students shall enter project details into web portal.
FR_05	The system shall allow a student user to upload a profile picture from device camera.	Students must submit one profile picture for project showcase.
FR_06	The system shall allow a student user to upload a profile picture from device file system.	Students must submit one profile picture for project showcase.
FR_07	The system shall validate student user data on input.	Project details must be valid to submit.
FR_08	The system shall allow a student user to submit data for approval.	If data is valid it can be submitted for approval by Admin user.
FR_09	The system shall capture student user data in a database.	All data will be saved in database for report generation.
FR_10	The system should associate a supervisor with student(s).	Many supervisors may have more than one student.
FR_11	The system shall allow users to view student user profiles.	All users shall be able to view student profiles.
FR_12	The system shall allow admin user to approve a student user submission.	If data submitted is of required standard the admin may approve submission for use in showcase booklet.
FR_13	The system shall allow admin user to reject a student user submission.	If data submitted is not of required standard the admin may reject submission.
FR_14	The system shall send email when condition is met by admin user (approval or rejection of submission).	Regardless of approval/rejection the system will inform the student of the status of their submission.
FR_15	The system shall allow admin user to search for student users.	Admin user may search for any current Final Year Student and note their status.
FR_16	The system shall output report from database.	Admin user may request for project details booklet.

Use Case Diagram



Use Case 1: FR_01: Add Students

Scope

The scope of this of this use case is to allow the Admin to register (upload) Student details to database.

Description

This use case describes how the Admin will add Student details to database.

Flow Description:

Precondition

The user has an internet connection.

The user is an Admin user.

The Admin user is logged in to system.

Activation

The use case starts when the User selects ‘Upload CSV’ from Admin Dashboard index page.

Main Flow:

1. The system presents an interface with field to select a CSV file from the file system.
2. The user navigates to selects correct file with Student data on their file system.
3. The user selects ‘Upload Details’. (See A1)
4. The system informs the user that their data has been accepted with confirmation of number of rows inserted to database. (E1)

Alternate Flow:

A1: Data not Valid

1. The system informs user it cannot upload file selected due to data validation.
2. The user corrects all errors.
3. The use case continues at position 3 of the main flow.

Exceptional Flow:

E1: File not recognized

1. The system informs user it cannot upload file selected due to unrecognized file format.
2. The system presents an interface select another file from file system.
3. User refreshes page.
4. Use case ends.

Termination

The system redirects the User to Admin Dashboard Index page.

Post Condition

The system goes into a wait state.

Use Case 2: FR_02: Student Register

Scope

The scope of this of this use case is to allow a Student to register to the system.

Description

This use case describes how the Student will register to the system.

Dependencies with other Requirements

FR_01: The system shall allow Admin User to register Student details to the application.

Flow Description:

Precondition

The user has an internet connection.

The user’s details have been pre-loaded into system database.

Activation

The use case starts when the User selects ‘Register’ from Web Portal index page.

Main Flow:

1. The system presents an interface with fields to input Student Email (username), Student Number and Password.
2. The user enters all required data in the fields.
3. The system validates data. (See A1)
4. The user selects the ‘Register’ button.
5. User confirms they wish to submit this data.
6. The system informs the user that their data has been accepted. (E1)

Alternate Flow:

A1: Data not Valid

1. The system displays errors in data entry.
2. The user corrects all errors.
3. The use case continues at position 4 of the main flow.

Exceptional Flow:

E1: Student details not pre-loaded

1. The system informs user it cannot find Student Number in system database.
2. Student navigates to ‘Email Admin’
3. The system presents an interface to email Admin to request Student details be entered into system.
4. Student sends email.
5. Use case ends.

Termination

The system returns the User to Web Portal homepage.

Post Condition

The system goes into a wait state.

Use Case 3: FR_08: Submit Data

Scope

The scope of this of this use case is to allow a Student User of the system to submit their project detail data.

Description

This use case describes how a Student User will submit project details for approval.

Dependencies with other Requirements

FR_07: The system shall validate student user data on input.

Flow Description:

Precondition

The user has an internet connection.

The user has an account registered on the system.

The user has logged in to their account.

Activation

The use case starts when the Student User selects ‘Create New’ button from Submissions page.

Main Flow:

1. The system presents an interface with fields to input LinkedIn Profile Link, Project Title, Project Details (200 Word and 30/40 Word Profile), Technologies Used and Profile Picture upload.
2. The user enters all required data in the fields. (E1)
3. The system validates data. (See A1)
4. The user selects the ‘Submit’ button.
5. User confirms they wish to submit this data.
6. The system informs the user that their data has been submitted for approval by Admin.

Alternate Flow:

A1: Data not Valid

1. The system displays errors in data entry.
2. The user corrects all errors.
3. The use case continues at position 4 of the main flow.

Exceptional Flow:

E1: Technologies used not listed

1. The system displays list of technologies Student used in project development.
2. Technology used not listed.
3. Student navigates to ‘Email Admin’ to request Technology is listed.
4. Student sends email.
5. Use case ends.

Termination

The system redirects the Student User to Submission Details page.

Post Condition

The system goes into a wait state.

Use Case 4: FR_12: Approve Submission

Scope

The scope for this use case is for the Admin user to approve a Student User submission.

Description

The Admin user may approve the Student user submission if it is of required standard.

Dependencies with other Requirements

FR_09: The system shall capture student user data in a database.

Flow Description:

Precondition

Student user submission has been validated by system.

Student user submission has been saved to database.

Activation

The use case starts when the Admin user navigates to the Administration dashboard.

Main Flow:

1. On the Administration dashboard, the Admin selects ‘View Submissions’ button.
2. The system presents a list of submissions that are awaiting approval or rejection. (E1)
3. The Admin may select a submission.
4. The system presents that individual submission.
5. The Admin selects ‘Approve Submission’
6. Admin confirms they wish to approve that submission.

Exceptional Flow:

E1: No submissions

1. System tries to present list of submissions from Student Users
2. No submissions yet, nothing to approve

3. Admin user has nothing to approve
4. Use case ends

Termination

The system sends email to Student User informing them of the approval status.

Post Condition

The system goes into a wait state.

Use Case 5: FR_07: Validate Data

Scope

The scope of this use case is for the system to validate user data as they enter it.

Description

This use case describes how the system will validate the Student User submission per the rules specified (max number of words, spelling etc.)

Flow Description:

Precondition

Student user must be logged in to system.

Activation

The use case starts when the Student User enters project details data.

Main Flow:

1. Student User navigates to Project Submissions page.
2. Student User begins to enter data in required fields.
3. The system will verify contents on input (See A1)
4. The system informs Student User that data is valid and ready for submission.

Alternate Flow:

A1: Data not valid

1. The system has found errors in data submission.
2. The system will warn Student User their data is not valid.
3. Student User must fix errors before submission.
4. The use case continues at position 4 of the main flow.

Termination

The system has informed the Student User that their data is valid.

Post Condition

The system goes into a wait state.

Use Case 6: FR_16: Generate Report

Scope

The scope of this use case is for the system to produce a report of submissions as requested by Admin user.

Description

This use case describes how an Admin user may request for the system to generate the project showcase booklet.

Flow Description:

Precondition

All Student User submissions have been approved.

Activation

The use case starts when the Admin User wishes to generate project showcase draft booklet.

Main Flow:

1. Admin user accesses the Administration dashboard.
2. Admin user selects ‘Generate Draft Profile Booklet’.
3. The system presents a list of all approved submissions.
4. Admin user selects ‘Export as PDF’ at top of page. (See E1)

Exceptional Flow:

E1: Picture not present

1. System tries to export data to PDF file.
2. A submission does not have a picture attached to the student user.
3. Report cannot be generated
4. Use case ends

Termination

The system has produced the project showcase booklet PDF document.

Post Condition

The system goes into a wait state.

Use Case 7: FR_14: Send Email

Scope

The scope of this use case is for the system to email a student user about the status of their submission.

Description

After an Admin user, has approved or rejected a Student User submission the system shall send an email to that Student User.

Dependencies with other Requirements

FR_12: The system shall allow admin user to approve a student user submission.

FR_13: The system shall allow admin user to reject a student user submission.

Flow Description:

Precondition

Student User is registered to system with valid college email address.

Activation

Use case starts when Admin user approves or rejects Student User submission.

Main Flow:

1. On review of a Student User project details, the Admin User may approve or reject the submission.
2. Admin selects ‘Approve’ or ‘Reject’.
3. The system opens new page with default text regarding approval or rejection.
4. Admin User may alter text.
5. Admin selects ‘Send’.
6. The system informs Admin User that email has been sent to Student.

Termination

The system returns Admin User to Administration dashboard.

Post Condition

System goes into a wait state.

2.1.2 Data requirements

ID	Requirement	Comment
DR_01	Storing Student User Data	Student table holds data about a student. Data to be captured: student number, student name, course, stream, supervisor, email and password.
DR_02	Storing Student Stream Data	Steam table holds data about the different Fourth Year Streams. Data to be saved: stream id, stream name
DR_03	Storing Student Supervisor Data	Supervisor table holds data about all project supervisors. Data to be saved: supervisor id, supervisor name
DR_04	Storing Student Course Data	Course table holds information on the different courses each student is registered to. Data to be saved: course id, course name
DR_05	Storing Submission Data	Submission table holds all data associated with each student's submission. Data to be captured: student id, student name, LinkedIn url, project title, short project description, long project description, technology id, profile picture image
DR_06	Storing Student Technologies Data	Technologies table holds information on technologies used by students. Data to be captured:
DR_07	Storing Report Data	Report table holds information on the generated reports. Data to be saved: technology id, technology name
DR_08	Storing Admin Data	Admin table holds information on the Administrator. Data to be saved: admin id, staff email, password

2.1.3 Business Rules

ID	Business Rule	Related Requirement
BR_01	Student user must be 4th Year Degree Student completing Final Year Project.	-
BR_02	Student user must have valid NCI email address.	FR_02
BR_03	Student user must have a supervisor.	FR_10
BR_04	Student user must register to system to submit project details.	FR_04
BR_05	Student user must input all data required to submit details.	FR_08
BR_06	Student user profile picture must adhere to specific rules.	FR_07
BR_07	All input data must be valid to submit details.	FR_09
BR_08	Admin user must approve or reject Student user submission.	FR_08
BR_09	The system must send email on approval or rejection of submission.	FR_14
BR_10	The system must output PDF file from report generation.	FR_16

2.1.4 User Requirements

1. The Process

a. The Careers section in the College manage this Showcase and produce a professional booklet containing two main features as shown in Appendix I:
    i. The 40-word Profile
    ii. The 200-word profile

2. Project High-Level Requirements

a. A front-end Web application to allow the students to enter the required information as seen in Appendix I
b. Two User levels:
    i. Admin
    ii. Student
c. When the Application opens, Users can view all Profiles
d. Students Details Pre-Loaded by Admin
e. To edit your profile User must register and then Login
f. Register:
    i. Use Student email as Username
    ii. Email verified by Application:
    iii. If not entered not there, email the User details to Course coordinator for validation
    iv. Else create user with password
g. When User logs in they will be brought to their own record for editing and add the following:
    i. The Project Title
    ii. The Students LinkedIn URL address
    iii. The Project Description (30 word)
    iv. The project Description (200 Word)
    v. The Students Photo (30mm width x 37mm height)
    vi. The Students Name
    vii. The Students Number
    viii. Technologies Used –
        1. This is a drop down allowing multiple choices
        2. If Technology IDs not there, user can select a button to email the Course Coordinator with the Technology to get it validated and added.
h. All this data to be captured in a back-end database

i. Reports to be created from the database using the correct Formats (as specified) that can be delivered directly to marketing to create the brochure.

3. Fonts & Photo

a. The font to be used is Lucida Sans 10
b. Photo resolution is to be 300pdi and sized 30mm width x 37mm height

2.1.5 Usability requirements

Usability is defined in the ISO Standard 9241-11 as "The extent to which a product can be used by specified users to achieve specified goals with effectiveness, efficiency and satisfaction in a specified context of use." In this project, it is important to understand user needs (both Admin and Students) and find to a way to develop the User Interface through heuristics (specified in testing) in such a way that users will always know where they are and what is required of them on the site.

Intuitiveness: does the system match the real-world language and processes users would expect to see?

Efficient: how does the system handle user tasks – is much effort required by user to complete tasks?

Consistency: does the site promote a consistent designed across all pages? It is important to employ a uniform interface and navigation.

Error Tolerance: it is important for this web app to minimize user error as much of the processes carried out by users are user-generated.

2.1.6 Non-Functional Requirements

This section will specify the non-functional requirements for the system. These requirements will detail how the system will perform.

Availability requirement

- The system will be available to all valid users through a web browser.

- The system should be available for normal use between 06.00 – 22.00 Monday to Saturday.
- The system should be available for maintenance use between 23.00 – 02.00 every day.

Recover requirement

- The system should backup to database daily
- The system should be able to recover from a database backup

Security requirement

- The system must validate a Student User email address ensuring they have valid NCI email address.
- Student Users may modify only their submission.
- Only Admin User may modify all submissions.

Reliability requirement

- Using the Azure Cloud Services platform, the system should be online and running always.

Maintainability requirement

- Using the MVC and separation of concerns, the system should be easily maintainable.

Portability requirement

- Following Web Standards, it should be easy to port this application to many different web browsers.
- ASP.NET is open source and cross-platform allowing for greater portability than other platforms.

Reusability requirement

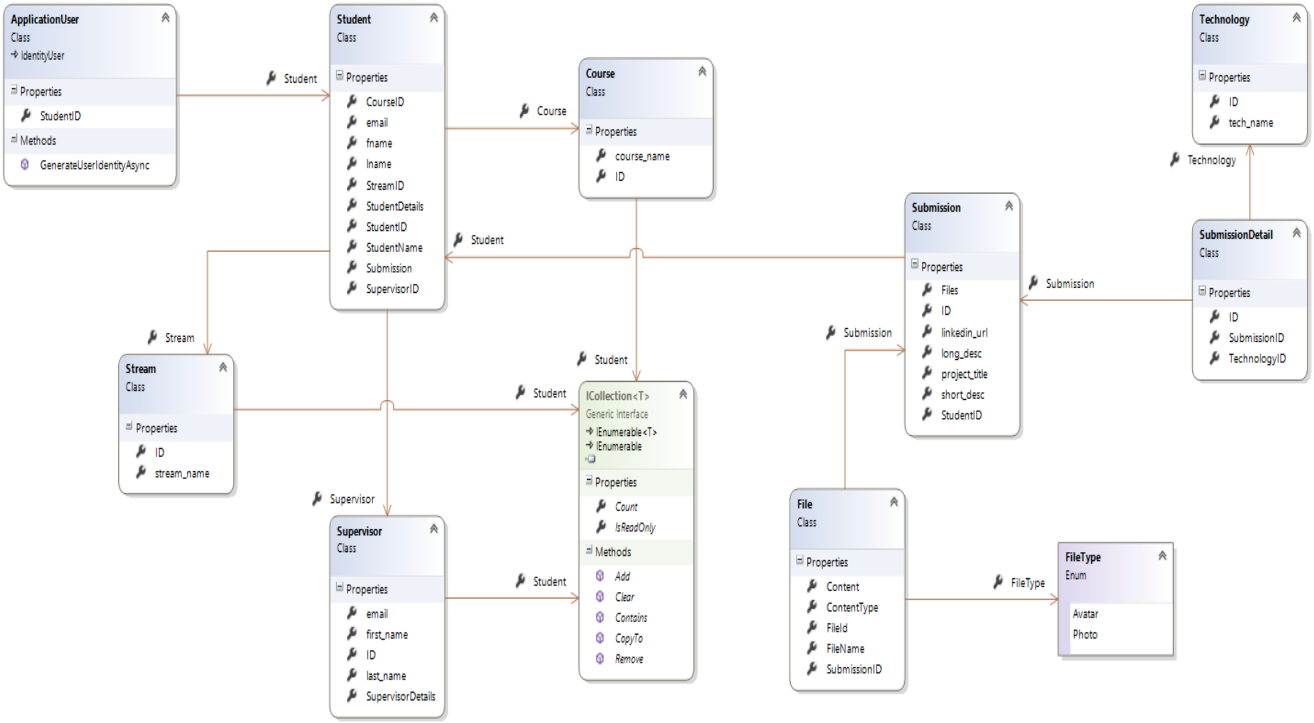
- The system database should allow for use in new web applications within the college.
- Components are separated and should be available for use in new MVC applications within the college i.e. registering student to database

Physical requirement

- The system will require a cloud-based service from which to run. Microsoft’s Azure Cloud Platform will be utilized.
- Users can access the system using any machine with a web browser and capability of file access/access to camera.

2.2 Design and Architecture

2.2.1 Logical View



2.2.2 Software Architecture

The architecture of this project will follow the design pattern of a typical MVC Web Application. The project will be divided between the Model View Controllers.

Client:

- The application portal will be presented in a Web Brower rendered by the View in the Web Application Logic.

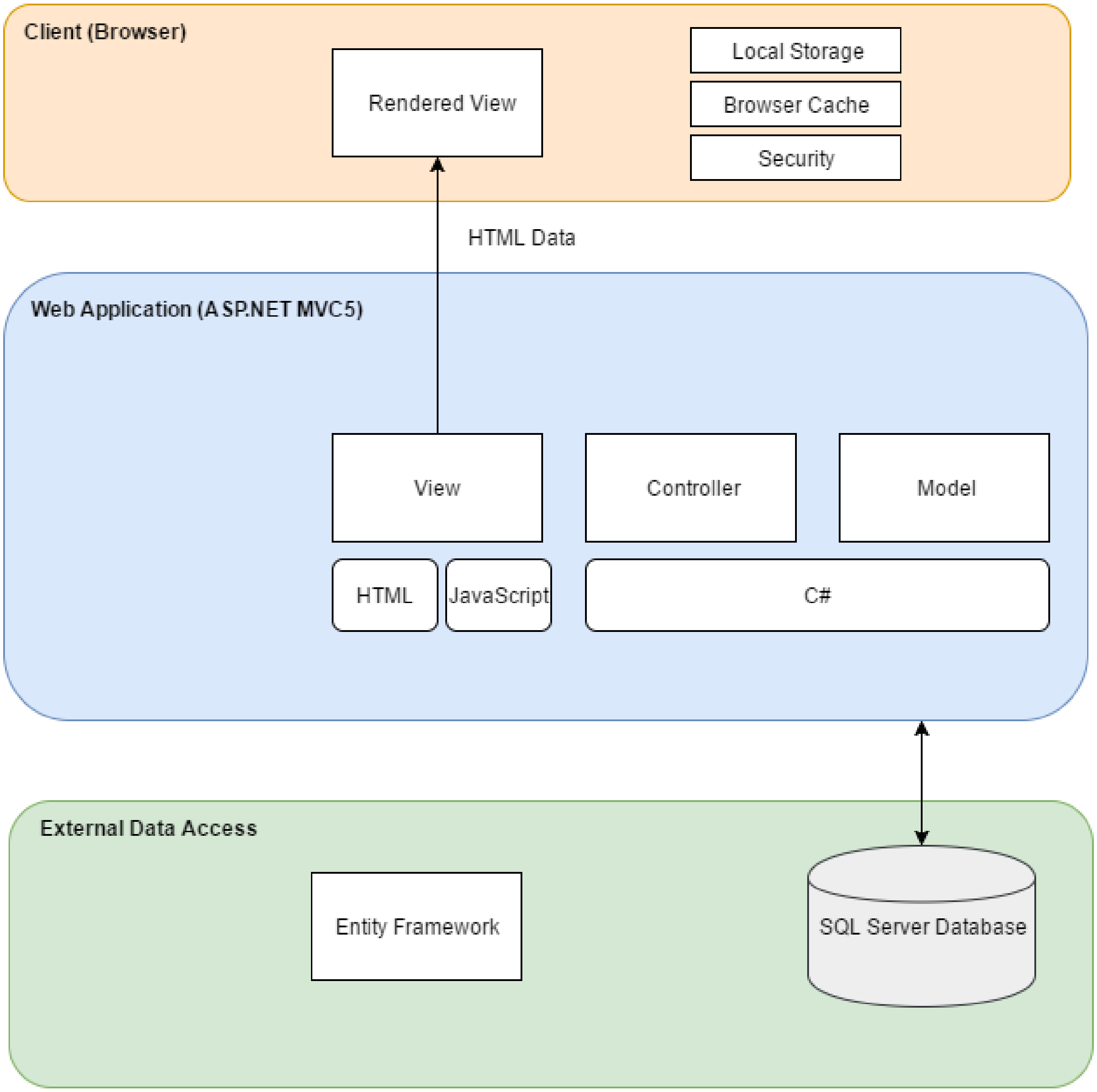
Web Application:

- Divided between Models, Views and Controllers. This division of the application into different, interconnected parts enables the application to decouple major components allowing for separation of concerns and reuse of code.
- The Model is the lowest level of the pattern and manages the behaviour and data of the application. In this project, the code-first entity framework will be employed. Code-first allows for the development of Domain Driven Design of the model over the database design.

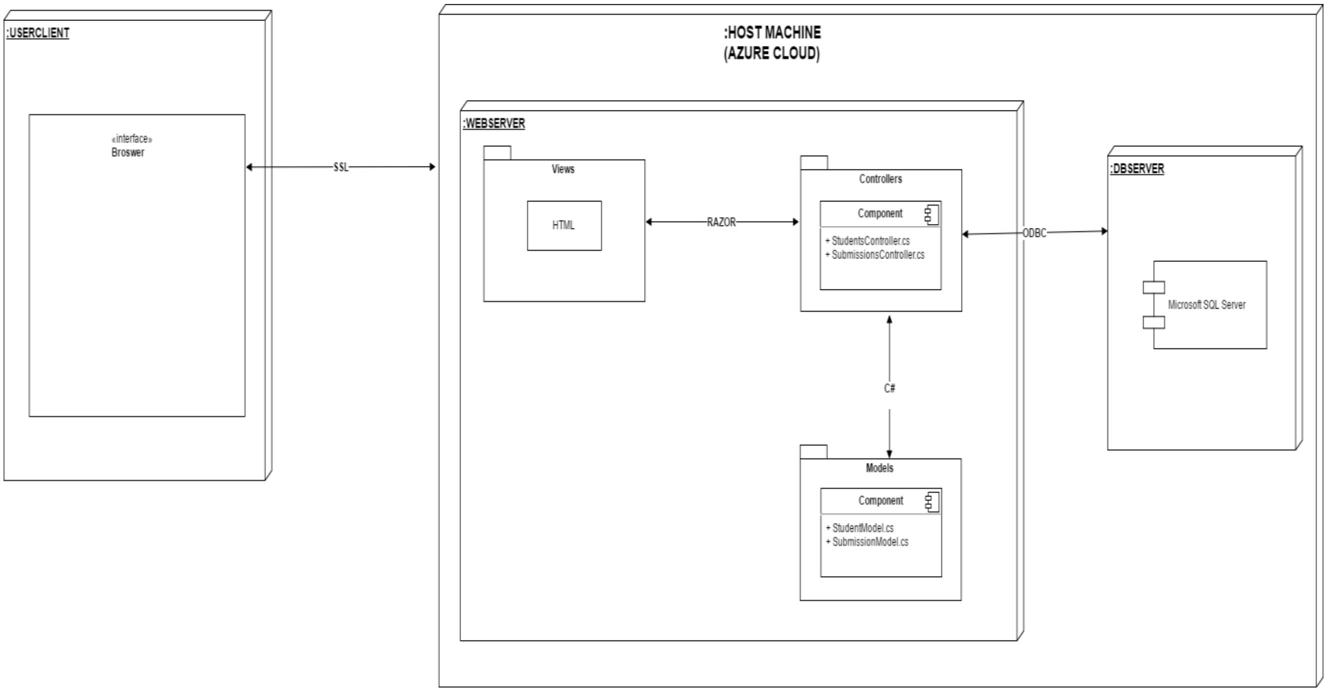
- The Controller controls the interactions between the Model and View. The controller accepts inputs and converts it to commands for the model or view.
- The View provides the user interface of the Web Application. Views render data from the Model into a form that is suitable for end users.

External Data Access:

- The Model, in this case, will describe the Database Design. The application will create the SQL Server Database on the fly.
- The Entity Framework layer contains classes and objects used in interfacing with the database.

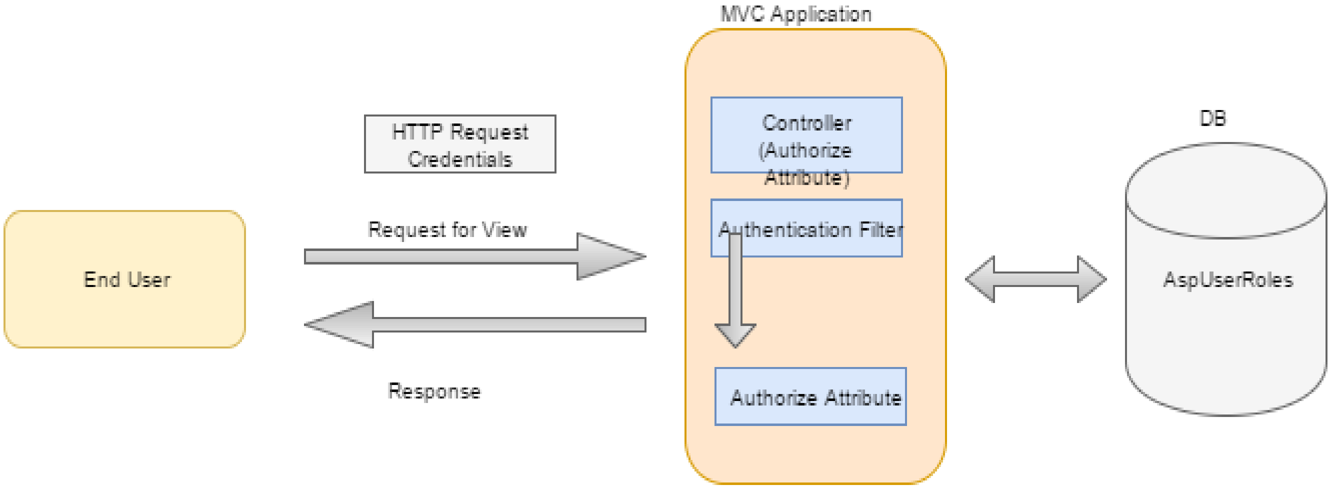


2.2.3 Deployment Diagram



2.2.4 Security Architecture

Role-based Authorization:

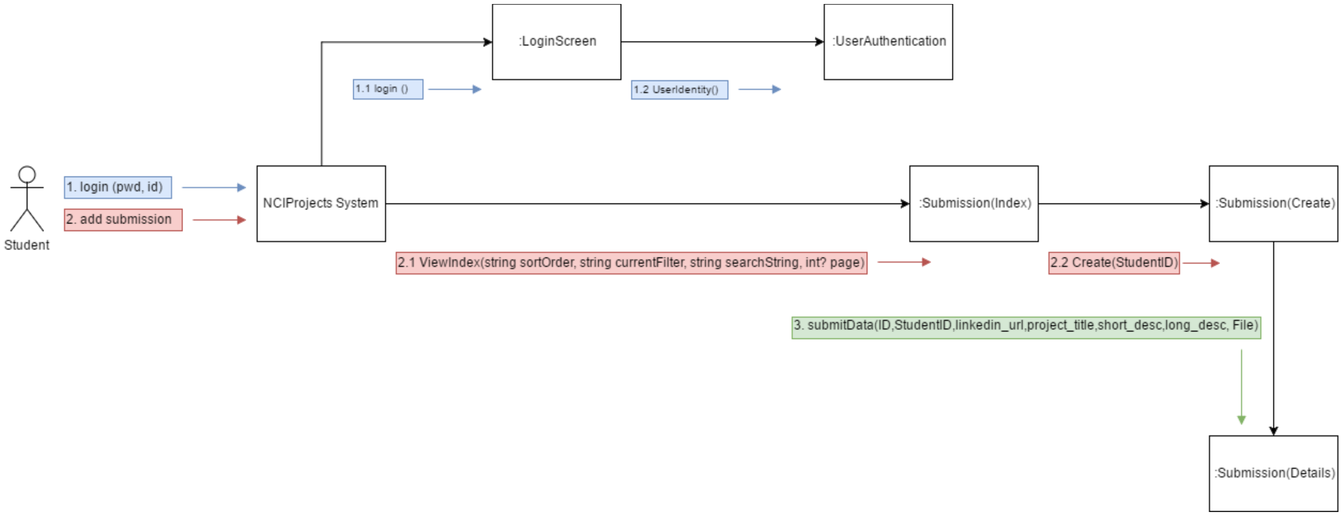


- MVC has robust authorization and authentication system (ASP.NET Identity)– allows admin access to parts of the system students are restricted from entering
- Role based authorization checks are declarative. The authorization is declared in the Controller code to limit the type of user that can access a certain page or area.

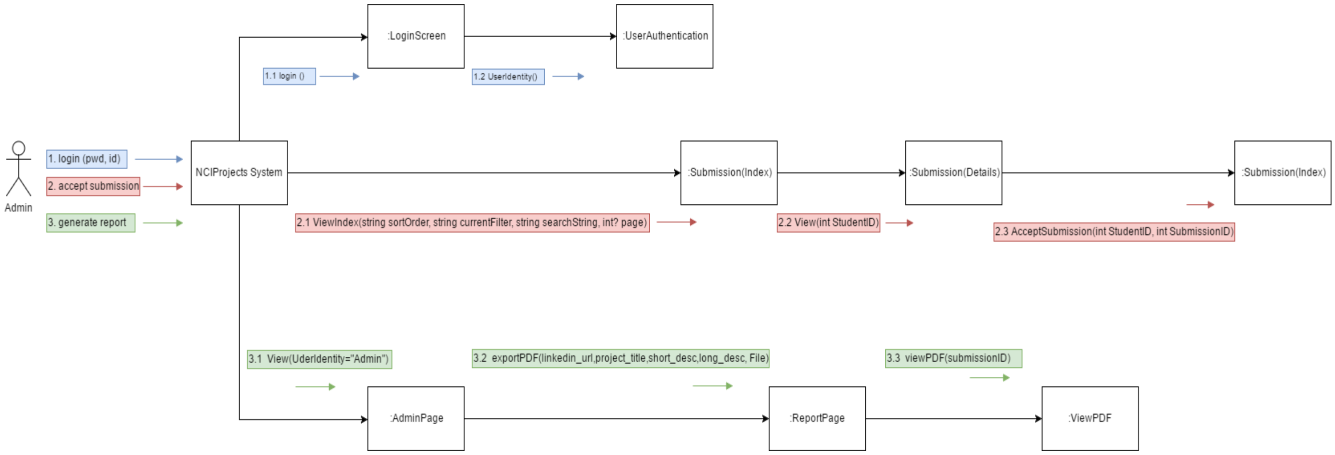
- [Authorize(Roles = "Admin")]
- This Authorized User Role is validated from the Database in the AspUserRoles tables where a user is assigned a RoleID, in this case “Admin”.
- Student Users of the system must be registered at NCI. They must have a student number or they will not be able to register to the system.

2.2.5 Communication Architecture

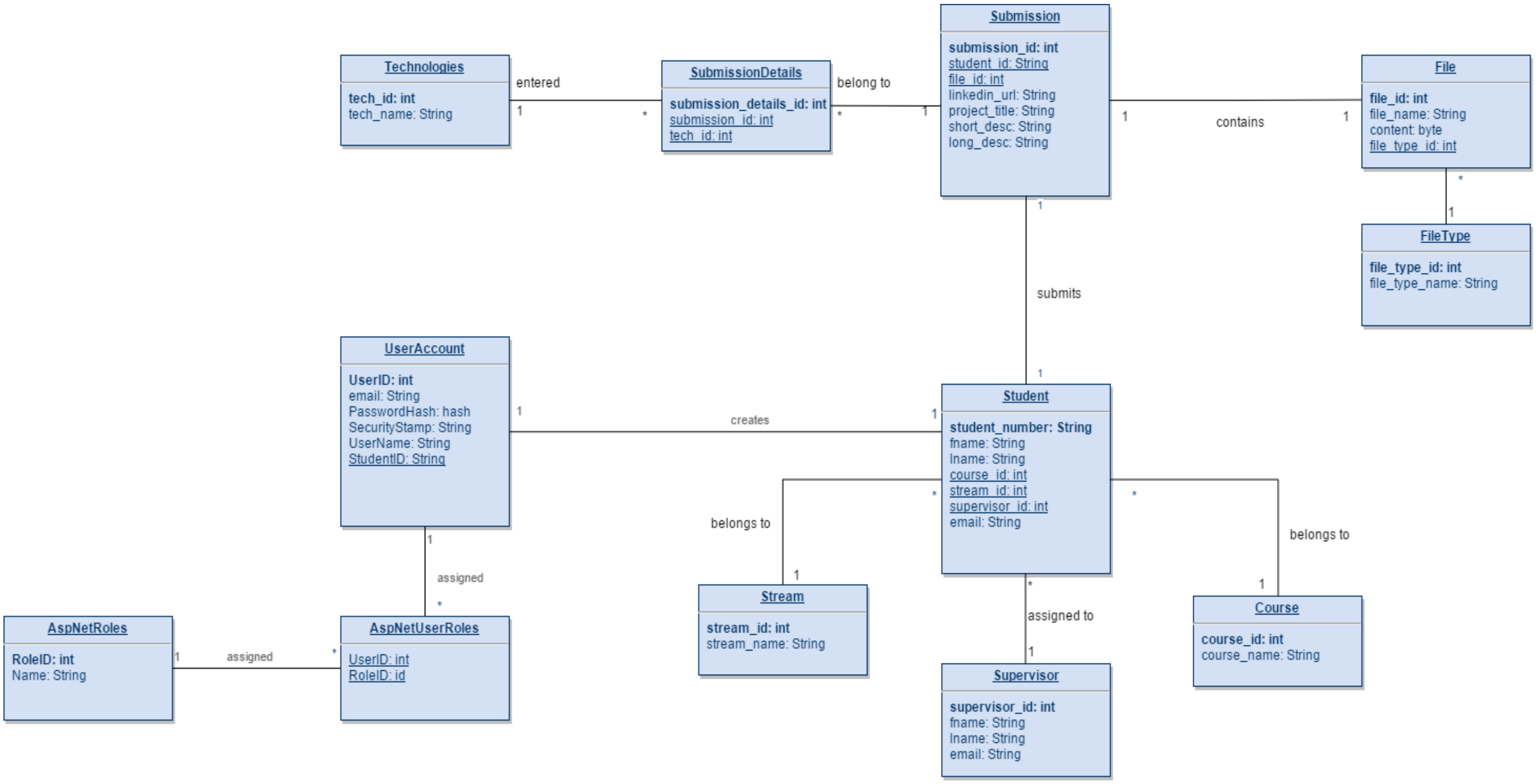
Student:



Admin:


:LoginScreen
:UserAuthentication
1.1 login ()
1.2 UserIdentity()
Admin
1. login (pwd, id)
2. accept submission
3. generate report
NCIProjects System
:Submission(Index)
:Submission(Details)
:Submission(Index)
2.1 ViewIndex(string sortOrder, string currentFilter, string searchString, int? page)
2.2 View(int StudentID)
2.3 AcceptSubmission(int StudentID, int SubmissionID)
3.1 View(UderIdentity="Admin")
3.2 exportPDF(linkedin_url,project_title,short_desc,long_desc, File)
3.3 viewPDF(submissionID)
:AdminPage
:ReportPage
:ViewPDF

2.2.6 Database Design



2.3 Implementation

This section will describe the main classes and functions used in the development of the project.

2.3.1 Application Programming Interfaces

The NuGet Package Manager (https://docs.microsoft.com/en-us/nuget/consume-packages/overview-and-workflow) in Visual Studio allows for the easy installation of different libraries including:

The ASP.NET Web API (Microsoft.AspNet.WebApi.Client). Allows for easy implementation of the REST architecture. This library adds support for formatting content management. This library also includes support for JSON, XML and form URL encoded data.

TinyMCE – a small but powerful text editor enables word count and spellcheck to ensure the quality of Submissions.

The iTextSharp library was utilized to create the PDF's required to produce the showcase booklets.

A number of other packages are available including Mircosoft's OAuth security and PagedList MVC which provides an easy way of implementing pagination into the application.

The MSTest Framework was used to carry out unit tests.

```
<packages>
  <package id="Antlr" version="3.4.1.9004" targetFramework="net46" />
  <package id="bootstrap" version="3.0.0" targetFramework="net46" />
  <package id="EntityFramework" version="6.1.3" targetFramework="net46" />
  <package id="jQuery" version="1.10.2" targetFramework="net46" />
  <package id="jQuery.Validation" version="1.11.1" targetFramework="net46" />
  <package id="Microsoft.AspNet.Identity.Core" version="2.2.1" targetFramework="net46" />
  <package id="Microsoft.AspNet.Identity.EntityFramework" version="2.2.1"
targetFramework="net46"/>
  <package id="Microsoft.AspNet.Identity.Owin" version="2.2.1" targetFramework="net46" />
  <package id="Microsoft.AspNet.Mvc" version="5.2.3" targetFramework="net46" />
  <package id="Microsoft.AspNet.Razor" version="3.2.3" targetFramework="net46" />
  <package id="Microsoft.AspNet.Web.Optimization" version="1.1.3" targetFramework="net46"/>
  <package id="Microsoft.AspNet.WebPages" version="3.2.3" targetFramework="net46" />
  <package id="Microsoft.CodeDom.Providers.DotNetCompilerPlatform" version="1.0.0"
targetFramework="net46" />
  <package id="Microsoft.jQuery.Unobtrusive.Validation" version="3.2.3" targetFramework="net46" />
  <package id="Microsoft.Net.Compilers" version="1.0.0" targetFramework="net46"
developmentDependency="true" />
  <package id="Microsoft.Owin" version="3.0.1" targetFramework="net46" />
  <package id="Microsoft.Owin.Host.SystemWeb" version="3.0.1" targetFramework="net46" />
  <package id="Microsoft.Owin.Security" version="3.0.1" targetFramework="net46" />
  <package id="Microsoft.Owin.Security.Cookies" version="3.0.1" targetFramework="net46" />
  <package id="Microsoft.Owin.Security.Facebook" version="3.0.1" targetFramework="net46" />
  <package id="Microsoft.Owin.Security.Google" version="3.0.1" targetFramework="net46" />
  <package id="Microsoft.Owin.Security.MicrosoftAccount" version="3.0.1" targetFramework="net46"/>
  <package id="Microsoft.Owin.Security.OAuth" version="3.0.1" targetFramework="net46" />
  <package id="Microsoft.Owin.Security.Twitter" version="3.0.1" targetFramework="net46" />
  <package id="Microsoft.Report.Viewer" version="11.0.0.0" targetFramework="net46" />
  <package id="Microsoft.ReportViewer.2015" version="12.0.2.2402" targetFramework="net46"/>
  <package id="Microsoft.ReportViewer.Runtime.Common" version="12.0.2402.15"
targetFramework="net46" />
  <package id="Microsoft.SqlServer.Types" version="11.0.2" targetFramework="net46" />
  <package id="Microsoft.Web.Infrastructure" version="1.0.0.0" targetFramework="net46" />
  <package id="Modernizr" version="2.6.2" targetFramework="net46" />
  <package id="Newtonsoft.Json" version="6.0.4" targetFramework="net46" />
  <package id="Owin" version="1.0" targetFramework="net46" />
  <package id="PagedList" version="1.17.0.0" targetFramework="net46" />
  <package id="PagedList.Mvc" version="4.5.0.0" targetFramework="net46" />
  <package id="Respond" version="1.2.0" targetFramework="net46" />
  <package id="TinyMCE" version="4.5.3" targetFramework="net46" />
  <package id="WebGrease" version="1.5.2" targetFramework="net46" />
</packages>
```

2.3.2 File Upload

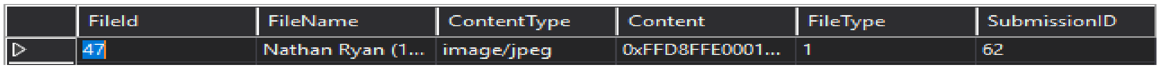
One of the main functions of this application is to provide a way for Students to upload a profile picture for use in the Showcase Booklet. This was achieved using the FileType and File class, saving files to the database as binary data.

```
public enum FileType
    {
        Avatar = 1, Photo
    }
```

Here we take the FileType and set it as a Photo to denote the purpose of that file. This FileType class means that different file types may be added at a later date to the project, if needed.

```
public class File
    {
        public int FileId { get; set; }
        [StringLength(255)]
        public string FileName { get; set; }
        [StringLength(100)]
        public string ContentType { get; set; }
        public byte[] Content { get; set; }
        public FileType FileType { get; set; }
        public int StudentId { get; set; }
        public virtual Student Student { get; set; }
    }
```

In the File class, we specify the model. Each file will be given a unique ID to be associated with a Student, the ContentType will specify if the Photo file is a JPEG or PNG file and the file will then be written to the Database as a byte[] object in the Content field.



	FileId	FileName	ContentType	Content	FileType	SubmissionID
▷	47	Nathan Ryan (1...	image/jpeg	0xFFD8FFE0001...	1	62

To display this File in the View a FileController class is also created. This Controller class allows us to specify how the file will be retrieved and displayed in the browser.

```
public class FileController : Controller
        private CollegeContext db = new CollegeContext();
        // GET: File
        public ActionResult Index(int id)
        {
            var fileToRetrieve = db.Files.Find(id);
            return File(fileToRetrieve.Content, fileToRetrieve.ContentType);
        }
```

This Controller will identify the ID of the file to be retrieved and return the file with Content and ContentType fields to the View.

The View in this case will be the Student Details page. The File is retrieved by opening an <img> tag and specifying the location of the file and associated user from the Database.

```
@if (Model.Files.Any(f => f.FileType == FileType.Avatar))
        {
            <dt>
                Avatar
            </dt>
            <dd>
                <img src="~/File?id=@Model.Files.First(f => f.FileType ==
FileType.Avatar).FileId" alt="avatar" />
            </dd>
        }
```

Using this method of storing files to the Database as binary data we can also retrieve a Student's Photo for the PDF report generation. The report is generated in a Web Form through a ReportViewer object associated with an .RDLC (Report Definition Language Client-side) file. The .RDLC file allows us to build a report and specify the Field in which the Photo file is stored. In this case, it is the Content field where the binary data is stored.

```
<Image Name="Avatar">
	<Source>Database</Source>
	<Value>=Fields!Content.Value</Value>
	<MIMEType>image/png</MIMEType>
	<Sizing>FitProportional</Sizing>
	<Top>0.74083cm</Top>
	<Left>11.53583cm</Left>
	<Height>2.77cm</Height>
	<Width>3.06104cm</Width>
	<ZIndex>4</ZIndex>
	<Style>
		<Border>
			<Style>None</Style>
		</Border>
	</Style>
</Image>
```

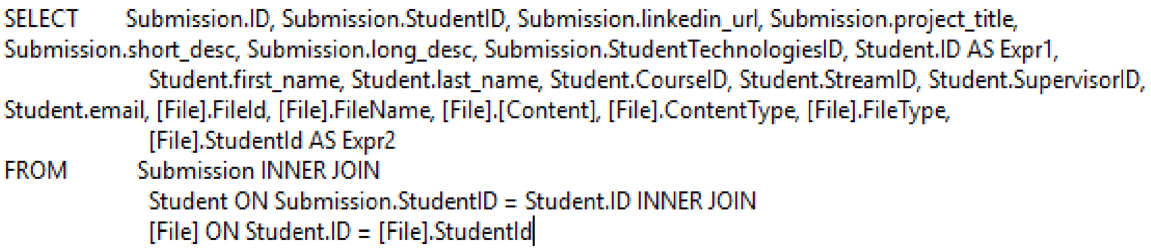
The Value field retrieves the Content values from each user.
The MIMEType field specifies that the File will be rendered as a PNG file in the PDF report.

2.3.3 Report Generation

2.3.3.1 Report Viewer

The Full Project Showcase Booklet is produced using the Microsoft Report Viewer tool. This tool allows us to specify what data we want from the database and display it in a Report.RDLC file to be used in a Web Form.

To retrieve the Student Submission data that will be displayed in the generated report a number of tables must be called.



```
SELECT      Submission.ID, Submission.StudentID, Submission.linkedin_url, Submission.project_title,
Submission.short_desc, Submission.long_desc, Submission.StudentTechnologiesID, Student.ID AS Expr1,
                   Student.first_name, Student.last_name, Student.CourseID, Student.StreamID, Student.SupervisorID,
Student.email, [File].FileId, [File].FileName, [File].[Content], [File].ContentType, [File].FileType,
                   [File].StudentId AS Expr2
FROM         Submission INNER JOIN
                   Student ON Submission.StudentID = Student.ID INNER JOIN
                   [File] ON Student.ID = [File].StudentId
```

From the Web Form: StudentSubmissions.aspx we specify the ReportViewer object that allows the end user to view and generate a PDF report.

```
<form id="form1" runat="server">
  <asp:ScriptManager ID="ScriptManager1" runat="server"></asp:ScriptManager>
  <div>
       <rsweb:ReportViewer ID="ReportViewer1" runat="server" Font-Names="Verdana"
Font-Size="8pt" Height="600px" WaitMessageFont-Names="Verdana" WaitMessageFont-
Size="14pt" Width="100%">
            <LocalReport ReportPath="Report1.rdlc">
                <DataSources>
                    <rsweb:ReportDataSource DataSourceId="ObjectDataSource1"
Name="SubmissionDataSet" />
                </DataSources>
            </LocalReport>
        </rsweb:ReportViewer>
        <asp:ObjectDataSource ID="ObjectDataSource1" runat="server"
SelectMethod="GetData"TypeName="NCIProjects.StudentSubmissionDataSetTableAdapters.
SubmissionTableAdapter"></asp:ObjectDataSource>
  </div>
</form>
```

2.3.3.2 Student Preview

The ‘student preview’ booklet comes in two forms: a 40-word profile version (Large booklet) and a 200-word profile version (Small booklet). Users can select to view these PDF’s from their submission page. These previews were created using the iTextSharp library.

The Student Preview booklets are designed in a way to show the student what their submission will look like in the full Showcase Booklet.

Both the Draft booklet and the Student Preview booklets use the GetData method in the SubmissionsController.

```
private DataTable GetData(string query) //get Query from Booklet Generation method
        {
            string conString =
ConfigurationManager.ConnectionStrings["DefaultConnection"].ConnectionString;
            SqlCommand cmd = new SqlCommand(query);
            using (SqlConnection con = new SqlConnection(conString))
            {
                using (SqlDataAdapter sda = new SqlDataAdapter())
                {
                    cmd.Connection = con;
                    sda.SelectCommand = cmd;
                    using (DataTable dt = new DataTable())
                    {
                        sda.Fill(dt);
                        return dt;
                    }
```

GetData is a generic function to execute SELECT queries. This method returns DataTable. This method will take the SQL query specified in any one of the GenerateLargeBooklet, GenerateSmallBooklet or GenerateDraftBoooklets. The method will then add data to rows in the DataTable that is returned in the MemoryStream of the selected Booklet method.

Both the large and small booklet are generated in the same fashion with slight changes to the layout.

```
string strDDLValue = form["Options"].ToString();
```

```
var dt = GetData("SELECT * FROM Students JOIN Submissions ON Submissions.StudentID
= Students.StudentID JOIN Courses ON Students.CourseID = Courses.ID JOIN Files ON
Submissions.ID = Files.SubmissionID WHERE Students.StudentID = " + strDDLValue);
```

The student submission number is passed from the view to the controller using the FormCollection class. This class allows us to create a form in the view and specify a name, in this case “Options”. The form will also take the current student submission number and pass it to the strDDLValue string. Using the SQL command, we can select the required fields where the StudentID is equal to the strDDLValue.

```
using (MemoryStream memoryStream = new MemoryStream())
                {
                    PdfWriter writer = PdfWriter.GetInstance(document,
memoryStream);
                    PdfPTable table = null;
                    //Open Document to add elements
                    document.Open();
                    Paragraph heading = new Paragraph("Large Profile Booklet",
FontFactory.GetFont("Lucida Sans", 20, BaseColor.DARK_GRAY));
                    heading.Alignment = Element.ALIGN_CENTER;
                    document.Add(heading);
```

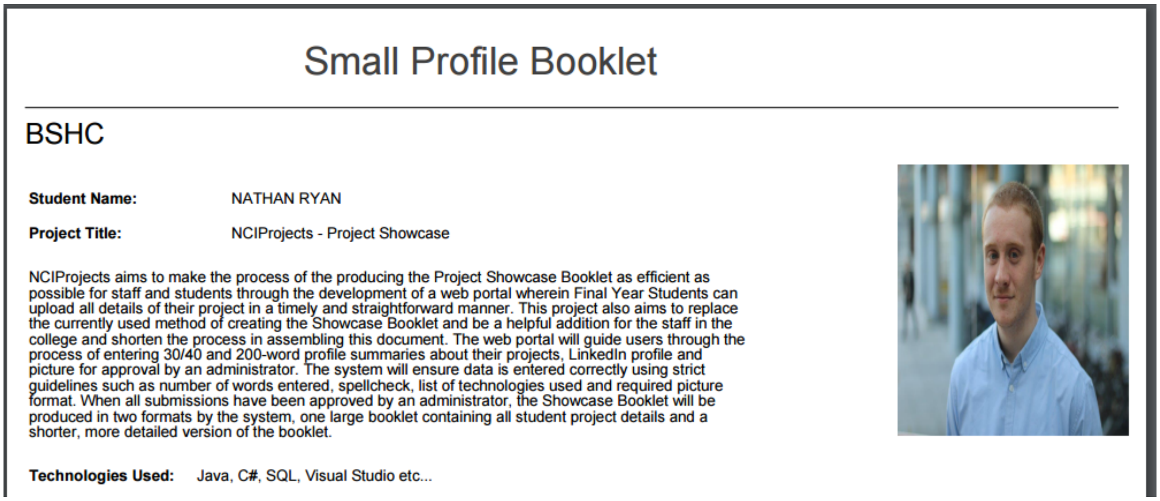
The MemoryStream class is derived from the Stream class in ASP.NET. The MemoryStream is a representation of bytes and is needed to produce the PDF file with the Student’s binary image from the database.

When all elements are added to the document we can then close the document to specify the end of the file and return a PDF document.

```
document.Add(table);
document.Close();
```

```
byte[] bytes = memoryStream.ToArray();
memoryStream.Close();

return File(memoryStream, "application/pdf");
```



Small Profile Booklet

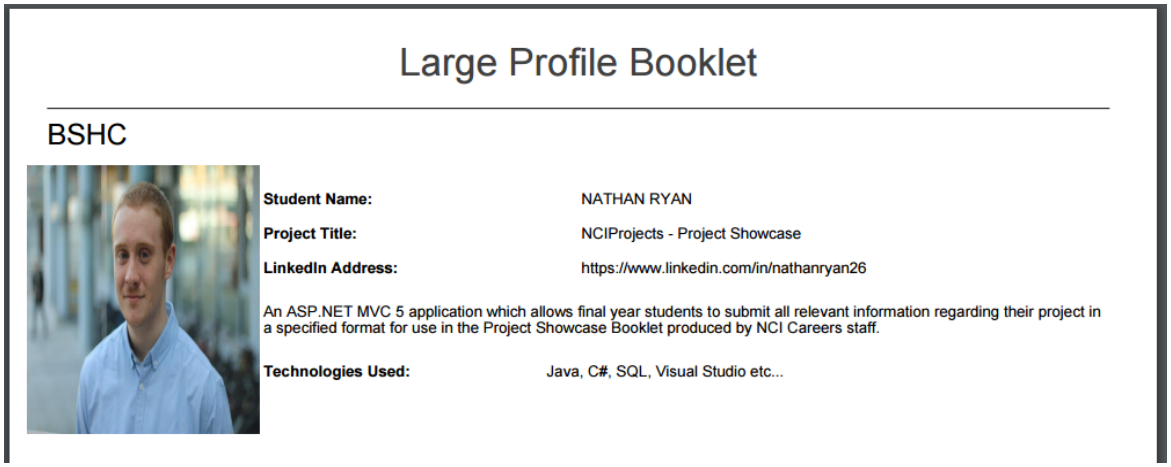
BSHC

Student Name: NATHAN RYAN

Project Title: NCIProjects - Project Showcase

NCIProjects aims to make the process of the producing the Project Showcase Booklet as efficient as possible for staff and students through the development of a web portal wherein Final Year Students can upload all details of their project in a timely and straightforward manner. This project also aims to replace the currently used method of creating the Showcase Booklet and be a helpful addition for the staff in the college and shorten the process in assembling this document. The web portal will guide users through the process of entering 30/40 and 200-word profile summaries about their projects, LinkedIn profile and picture for approval by an administrator. The system will ensure data is entered correctly using strict guidelines such as number of words entered, spellcheck, list of technologies used and required picture format. When all submissions have been approved by an administrator, the Showcase Booklet will be produced in two formats by the system, one large booklet containing all student project details and a shorter, more detailed version of the booklet.

Technologies Used: Java, C#, SQL, Visual Studio etc...



Large Profile Booklet

BSHC

Student Name: NATHAN RYAN

Project Title: NCIProjects - Project Showcase

LinkedIn Address: https://www.linkedin.com/in/nathanryan26

An ASP.NET MVC 5 application which allows final year students to submit all relevant information regarding their project in a specified format for use in the Project Showcase Booklet produced by NCI Careers staff.

Technologies Used: Java, C#, SQL, Visual Studio etc...

2.3.3.3 Draft Booklet

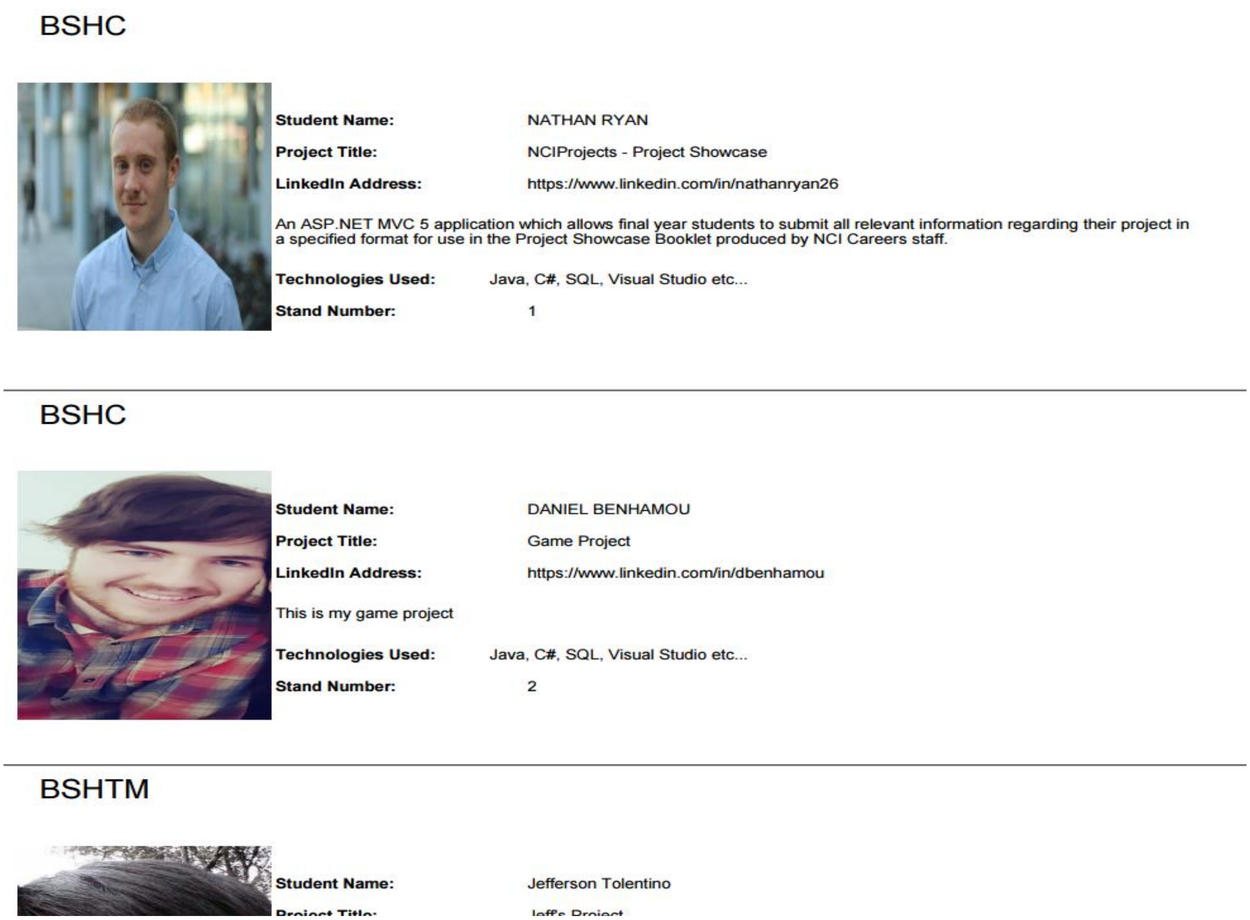
The 'draft' booklet is a full list of all submitted profiles that the Admin user can generate from the Administrator dashboard. This booklet is also created using the iTextSharp library.

Unlike the Student Preview booklets in the previous section where one page will be returned with that submission's specific data, the Draft booklet will output all student submissions to the file. This is achieved using a foreach loop that will iterate through each DataRow in the DataRow.

```
foreach (DataRow dr in dt.Rows) //start loop
{
```

In this foreach loop all objects are specified and the layout is created. At the end of each DataRow a new page must be specified to continue to output the results.

```
} //end loop
document.NewPage();
```



BSHC

Student Name: NATHAN RYAN
Project Title: NCIProjects - Project Showcase
LinkedIn Address: https://www.linkedin.com/in/nathanryan26

An ASP.NET MVC 5 application which allows final year students to submit all relevant information regarding their project in a specified format for use in the Project Showcase Booklet produced by NCI Careers staff.

Technologies Used: Java, C#, SQL, Visual Studio etc...
Stand Number: 1

BSHC

Student Name: DANIEL BENHAMOU
Project Title: Game Project
LinkedIn Address: https://www.linkedin.com/in/dbenhamou

This is my game project

Technologies Used: Java, C#, SQL, Visual Studio etc...
Stand Number: 2

BSHTM

Student Name: Jefferson Tolentino
Project Title: Jeff's Project

2.3.4 CSV File Upload

The Excel/CSV File Upload feature allows the Admin user to upload a file of all student details as stated in section 2.1.4 User Requirements as the ‘pre-load’ requirement.

This feature makes use of the SqlBulkCopy class which lets us efficiently bulk load a SQL Server table from another source (CSV file). Here we define the ConnectString to tell the method where to find the SQL Server Database connection on the network. We then specify the table we will load the data into –

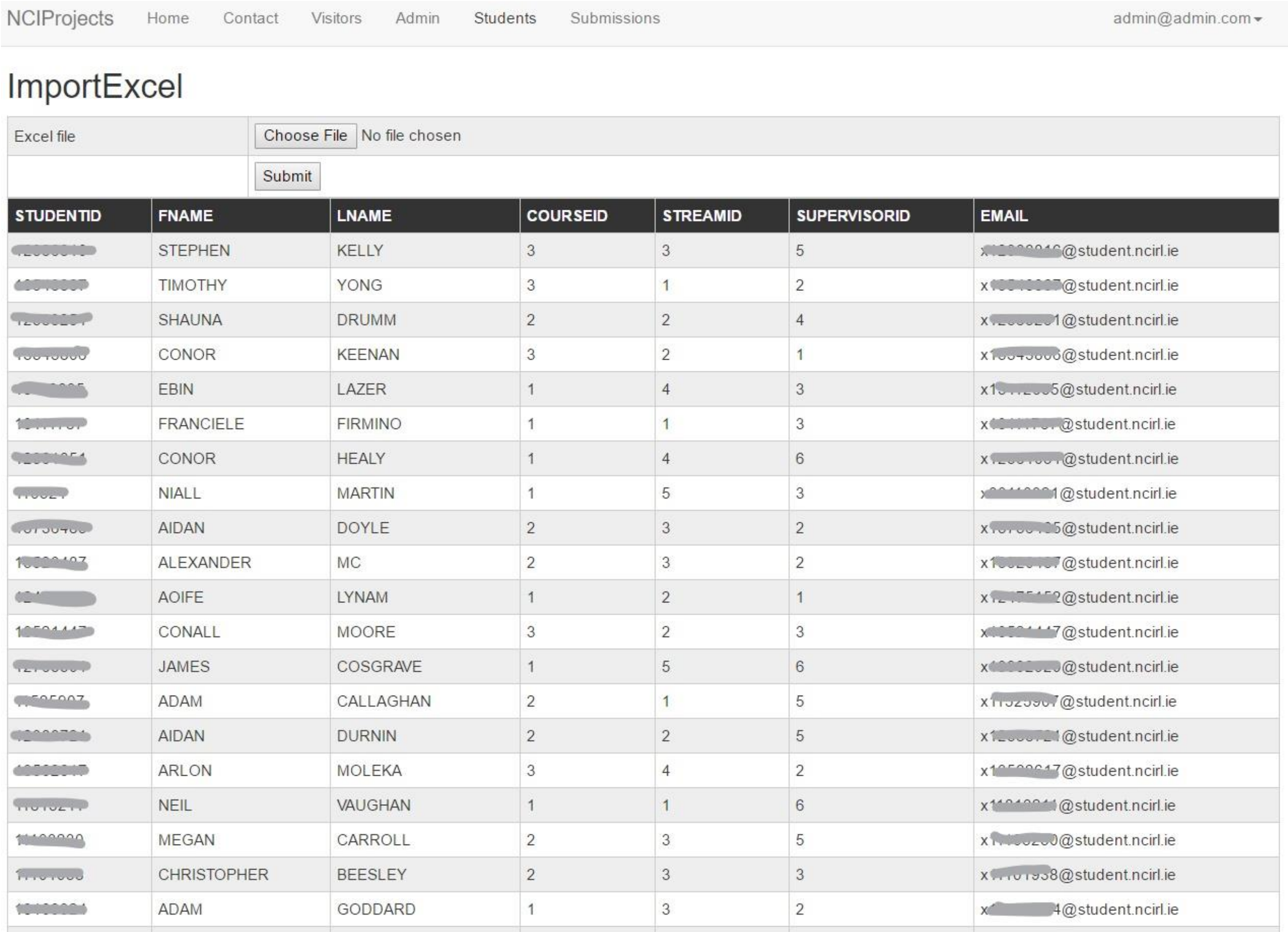
“dbo.Students”, we open the connection, write the data to the DataTable defined in the `ConvertCSVtoDataTable` method.

```
string consString =
ConfigurationManager.ConnectionStrings["DefaultConnection"].ConnectionString;
                using (SqlConnection con = new SqlConnection(consString))
                {
                    using (SqlBulkCopy sqlBulkCopy = new SqlBulkCopy(con))
                    {
                        //Set the database table name
                        sqlBulkCopy.DestinationTableName = "dbo.Students";
                        con.Open();
                        sqlBulkCopy.WriteToServer(dt);
                        con.Close();
                    }
                }
```

```
public static DataTable ConvertCSVtoDataTable(string strFilePath)
        {
            DataTable dt = new DataTable();
            using (StreamReader sr = new StreamReader(strFilePath))
            {
                string[] headers = sr.ReadLine().Split(',');
                    dt.Columns.AddRange(new DataColumn[7] {
                        new DataColumn("StudentID", typeof(int)),
                        new DataColumn("fname", typeof(string)),
                        new DataColumn("lname",typeof(string)),
                        new DataColumn("CourseID", typeof(int)),
                        new DataColumn("StreamID", typeof(int)),
                        new DataColumn("SupervisorID", typeof(int)),
                        new DataColumn("email",typeof(string))
                     });
```

The method loops through the CSV file reading each row and writing these records to the DataTable

```
while (!sr.EndOfStream)
                {
                    string[] rows = sr.ReadLine().Split(',');
                    if (rows.Length > 1)
                    {
                        DataRow dr = dt.NewRow();
                        for (int i = 0; i < headers.Length; i++)
                        {
                            dr[i] = rows[i].Trim();
                        }
                        dt.Rows.Add(dr);
                    }
                }
```



ImportExcel

Excel file	Choose File No file chosen
	Submit

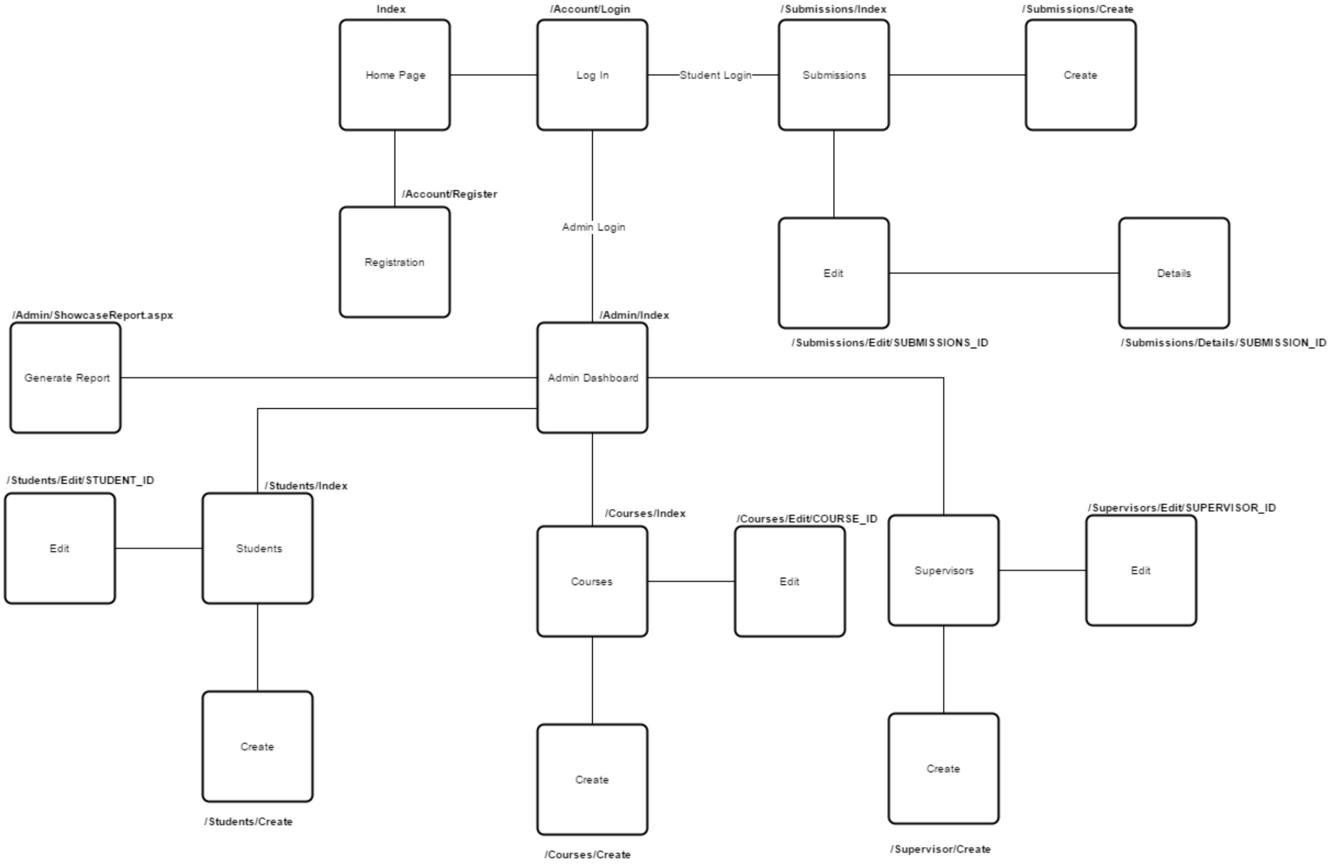
STUDENTID	FNAME	LNAME	COURSEID	STREAMID	SUPERVISORID	EMAIL
[redacted]	STEPHEN	KELLY	3	3	5	x[redacted]@student.ncirl.ie
[redacted]	TIMOTHY	YONG	3	1	2	x[redacted]@student.ncirl.ie
[redacted]	SHAUNA	DRUMM	2	2	4	x[redacted]@student.ncirl.ie
[redacted]	CONOR	KEENAN	3	2	1	x[redacted]@student.ncirl.ie
[redacted]	EBIN	LAZER	1	4	3	x[redacted]@student.ncirl.ie
[redacted]	FRANCIELE	FIRMINO	1	1	3	x[redacted]@student.ncirl.ie
[redacted]	CONOR	HEALY	1	4	6	x[redacted]@student.ncirl.ie
[redacted]	NIALL	MARTIN	1	5	3	x[redacted]@student.ncirl.ie
[redacted]	AIDAN	DOYLE	2	3	2	x[redacted]@student.ncirl.ie
[redacted]	ALEXANDER	MC	2	3	2	x[redacted]@student.ncirl.ie
[redacted]	AOIFE	LYNAM	1	2	1	x[redacted]@student.ncirl.ie
[redacted]	CONALL	MOORE	3	2	3	x[redacted]@student.ncirl.ie
[redacted]	JAMES	COSGRAVE	1	5	6	x[redacted]@student.ncirl.ie
[redacted]	ADAM	CALLAGHAN	2	1	5	x[redacted]@student.ncirl.ie
[redacted]	AIDAN	DURNIN	2	2	5	x[redacted]@student.ncirl.ie
[redacted]	ARLON	MOLEKA	3	4	2	x[redacted]@student.ncirl.ie
[redacted]	NEIL	VAUGHAN	1	1	6	x[redacted]@student.ncirl.ie
[redacted]	MEGAN	CARROLL	2	3	5	x[redacted]@student.ncirl.ie
[redacted]	CHRISTOPHER	BEESLEY	2	3	3	x[redacted]@student.ncirl.ie
[redacted]	ADAM	GODDARD	1	3	2	x[redacted]@student.ncirl.ie

)

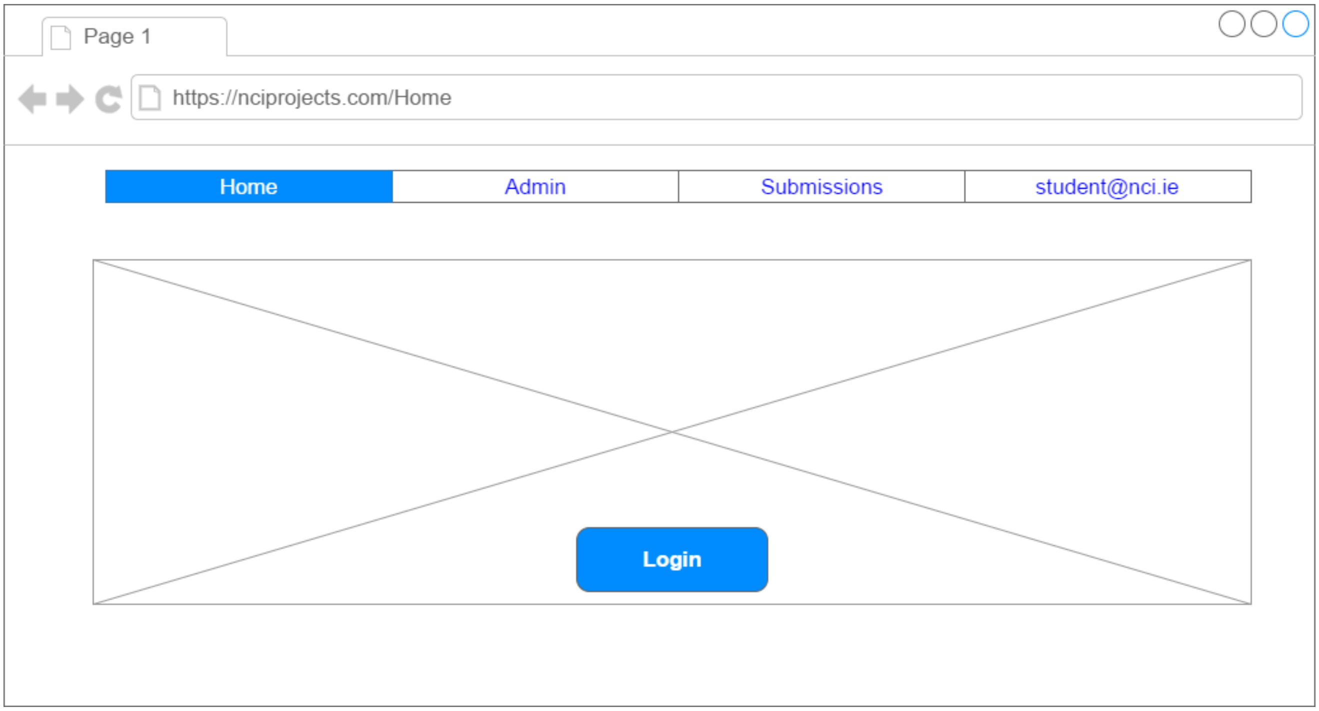
2.4 Graphical User Interface (GUI) Layout

This web application will run using a Graphical User Interface (GUI) in a web browser on a device connected to the internet. Users will operate the application using a mouse and keyboard for data input, or in the case of a mobile device they will use a touch screen.

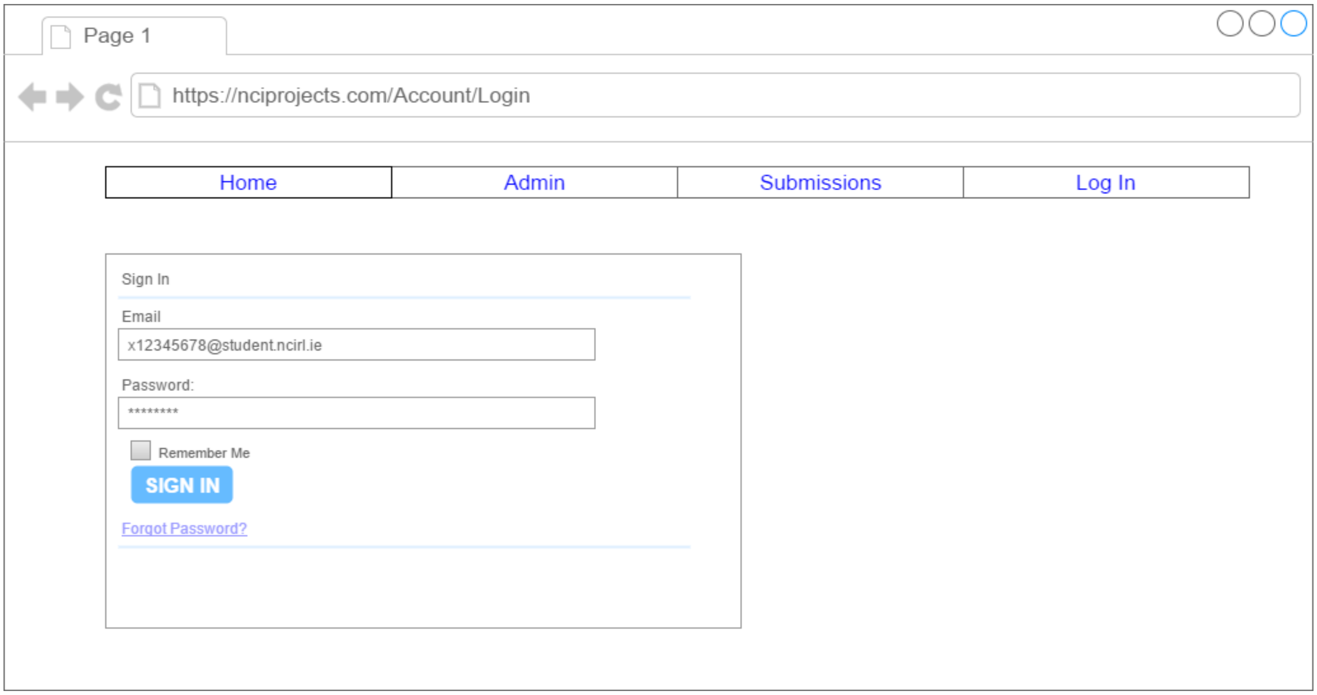
2.4.1 Flow Diagram



2.4.1.1 Index



2.4.1.2 Login

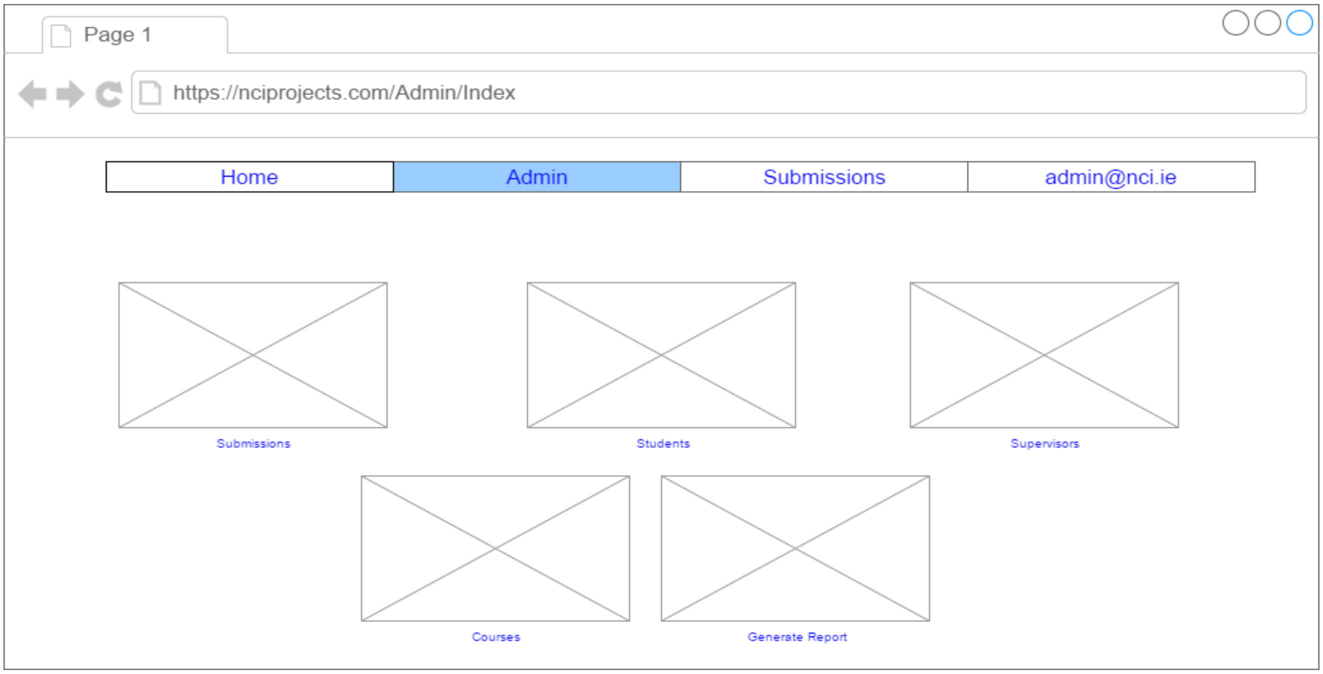


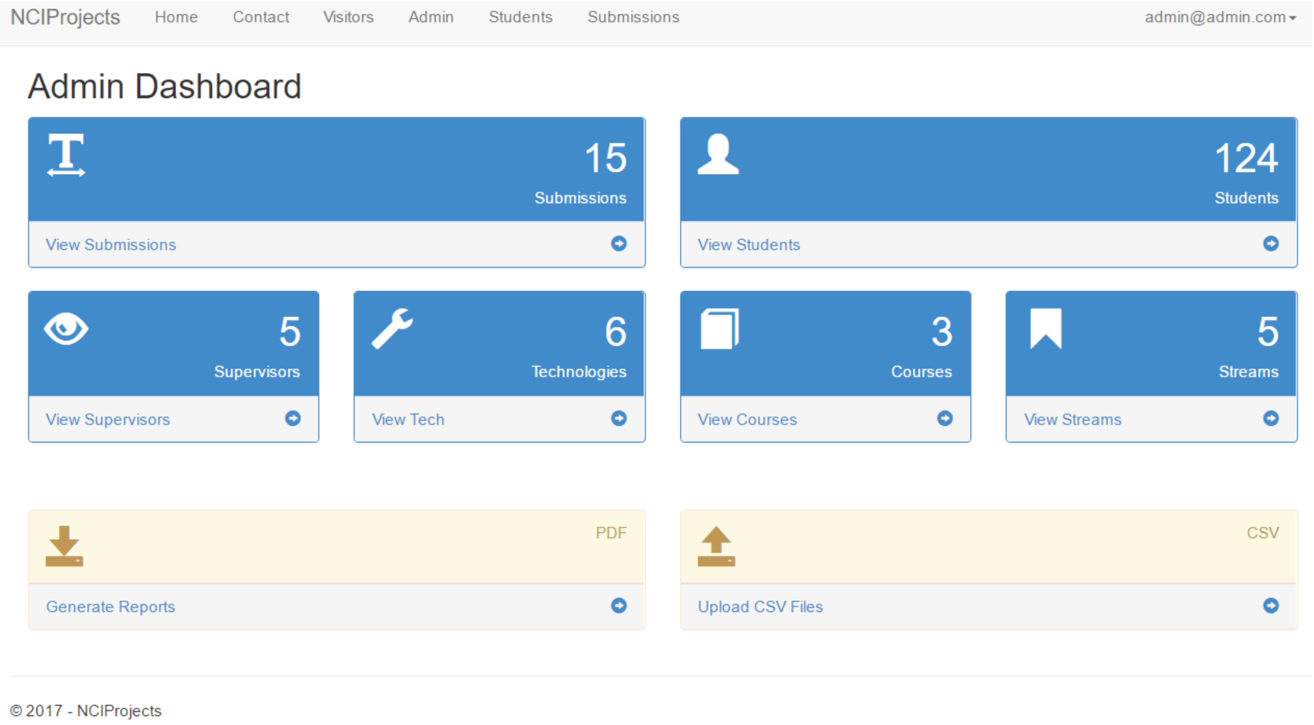
Email
x12345678@student.ncirl.ie

Password:
****

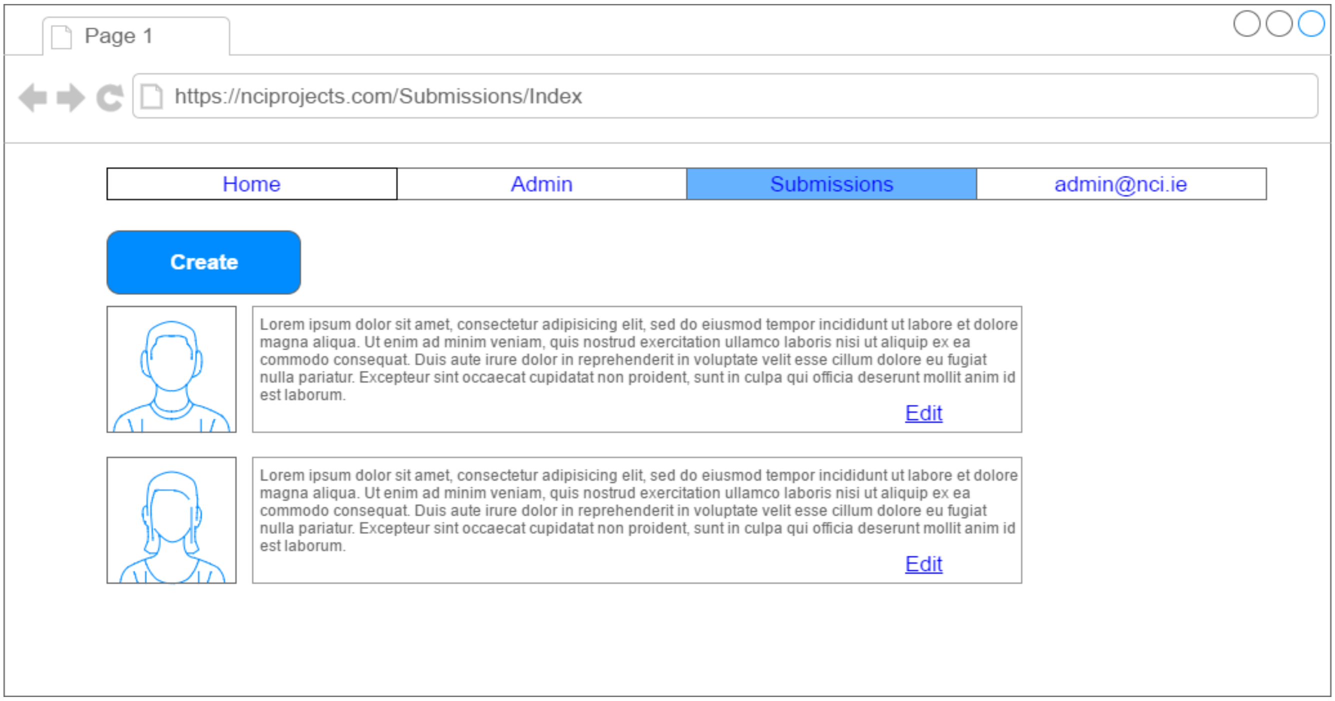
□ Remember Me

2.4.1.3 Admin Dashboard





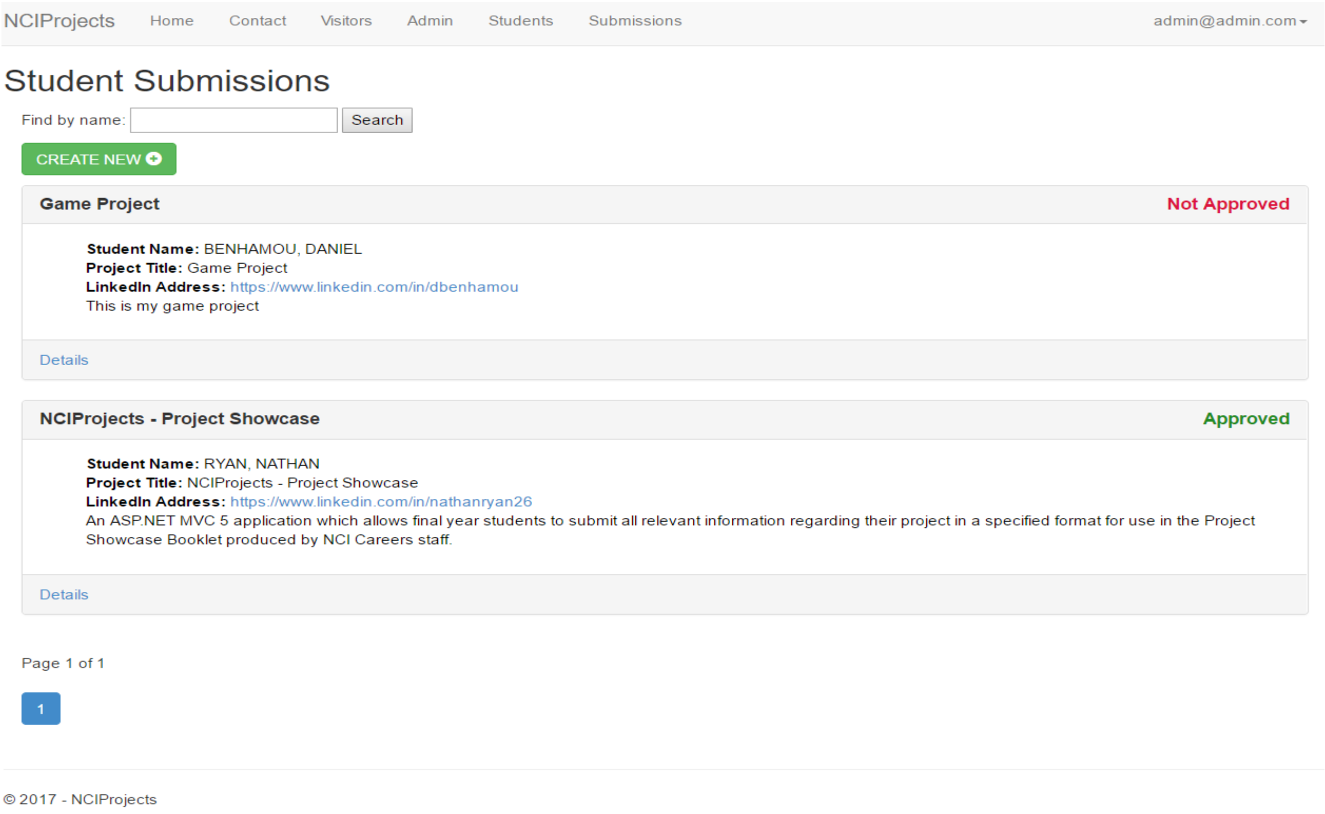
2.4.1.4 Student Submissions



https://nciprojects.com/Submissions/Index

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.



Student Submissions

Find by name:

Game Project Not Approved

Student Name: BENHAMOU, DANIEL
Project Title: Game Project
LinkedIn Address: https://www.linkedin.com/in/dbenhamou
This is my game project

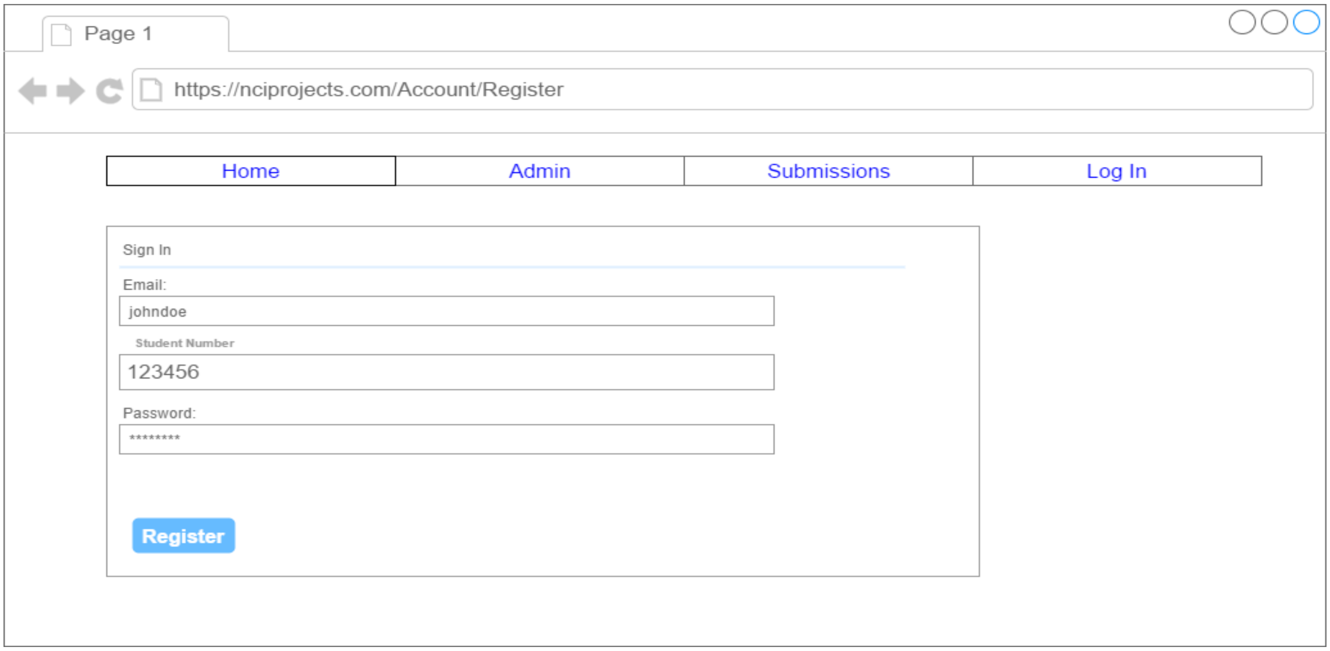
NCIProjects - Project Showcase Approved

Student Name: RYAN, NATHAN
Project Title: NCIProjects - Project Showcase
LinkedIn Address: https://www.linkedin.com/in/nathanryan26
An ASP.NET MVC 5 application which allows final year students to submit all relevant information regarding their project in a specified format for use in the Project Showcase Booklet produced by NCI Careers staff.

Page 1 of 1

© 2017 - NCIProjects

2.4.1.5 Register



https://nciprojects.com/Account/Register

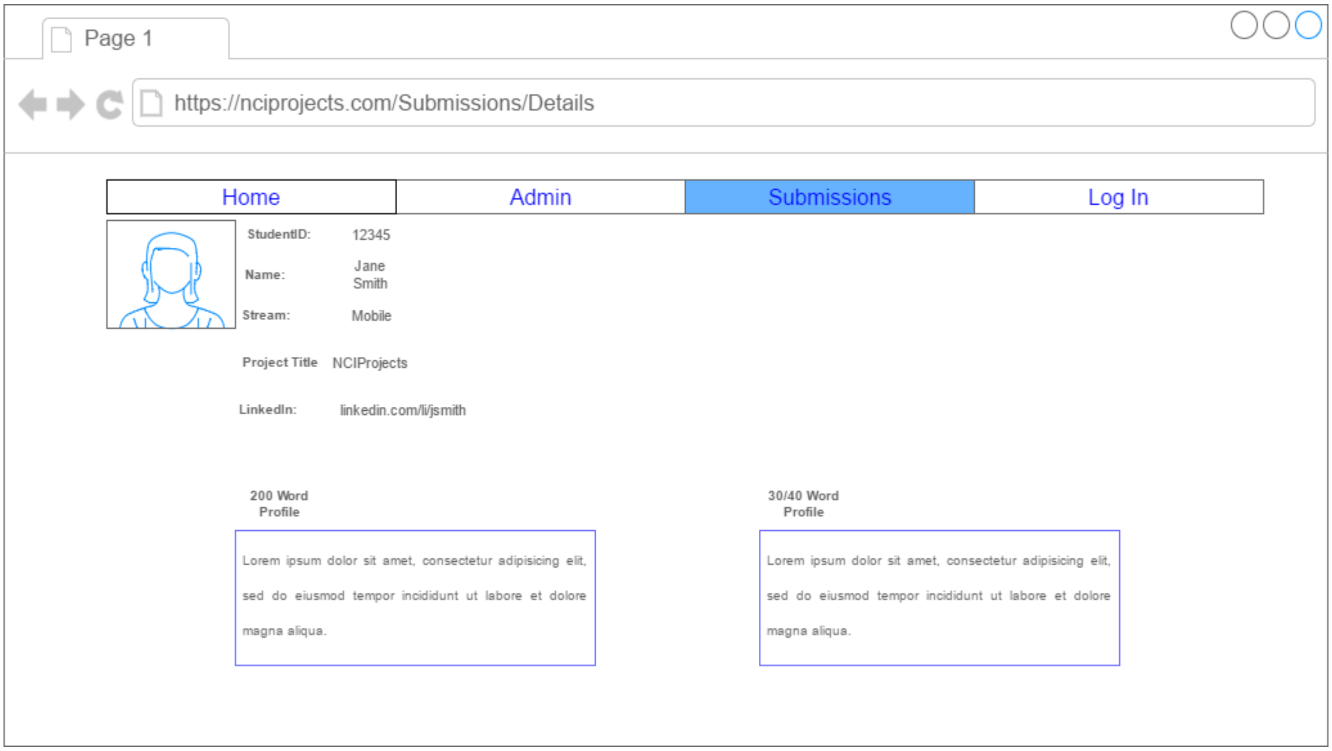
Sign In

Email: johndoe

Student Number 123456

Password: ********

2.4.1.6 Submissions Details



StudentID: 12345

Name: Jane Smith

Stream: Mobile

Project Title NCIProjects

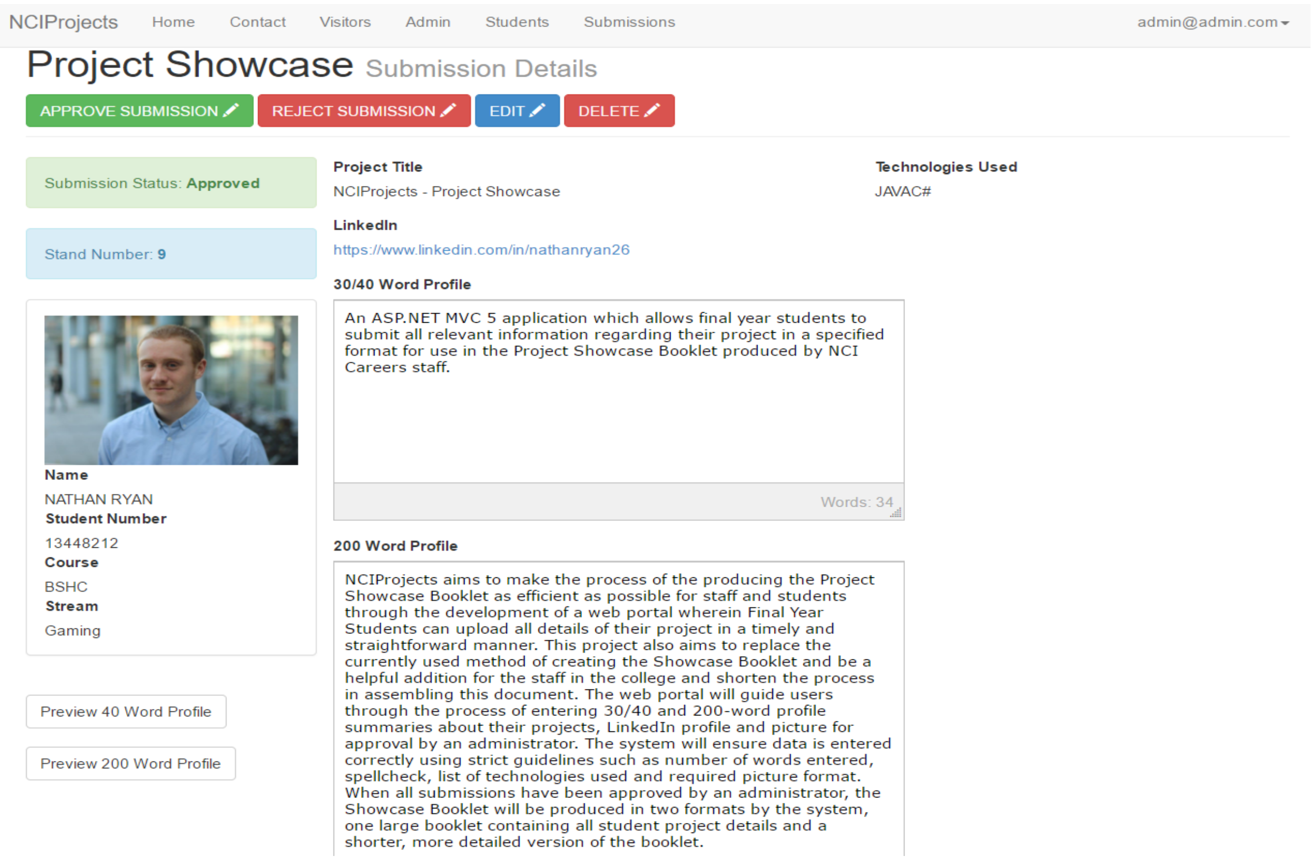
LinkedIn: linkedin.com/li/jsmith

200 Word Profile

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.

30/40 Word Profile

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.



Project Showcase Submission Details

Submission Status: **Approved**

Stand Number: **9**

Name
NATHAN RYAN
Student Number
13448212
Course
BSHC
Stream
Gaming

Project Title
NCIProjects - Project Showcase

Technologies Used
JAVAC#

LinkedIn
https://www.linkedin.com/in/nathanryan26

30/40 Word Profile

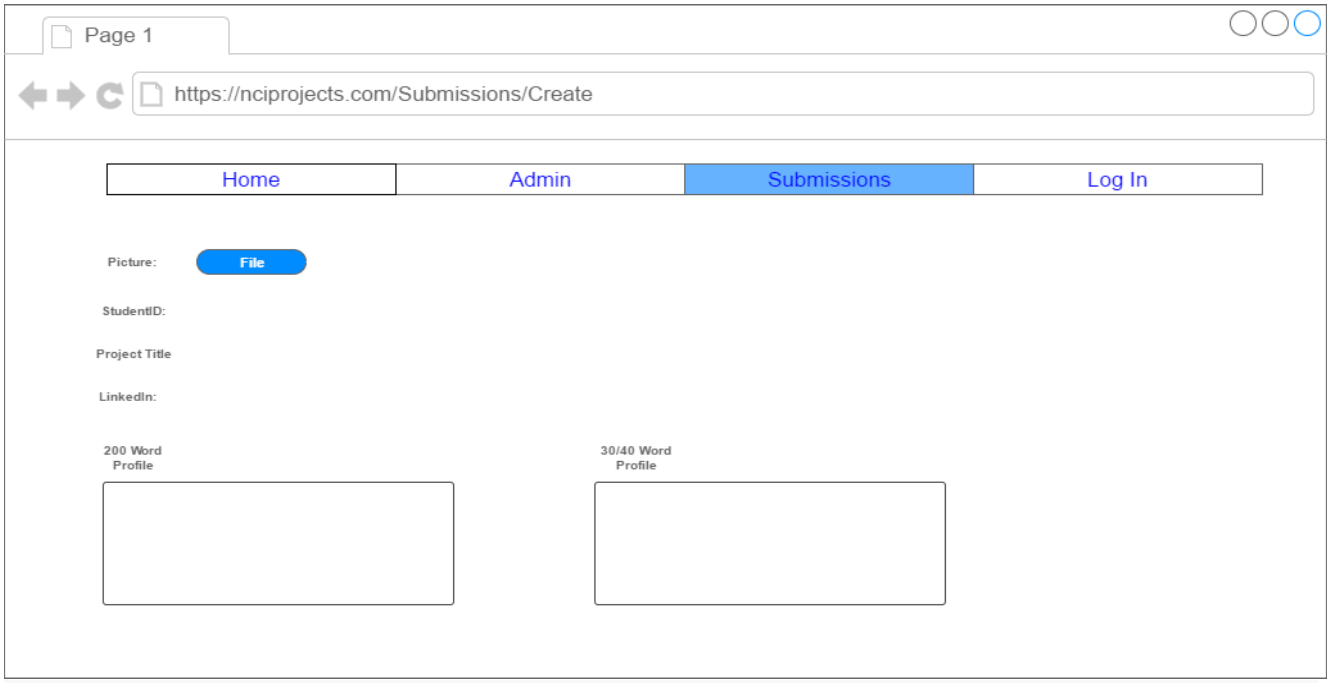
An ASP.NET MVC 5 application which allows final year students to submit all relevant information regarding their project in a specified format for use in the Project Showcase Booklet produced by NCI Careers staff.

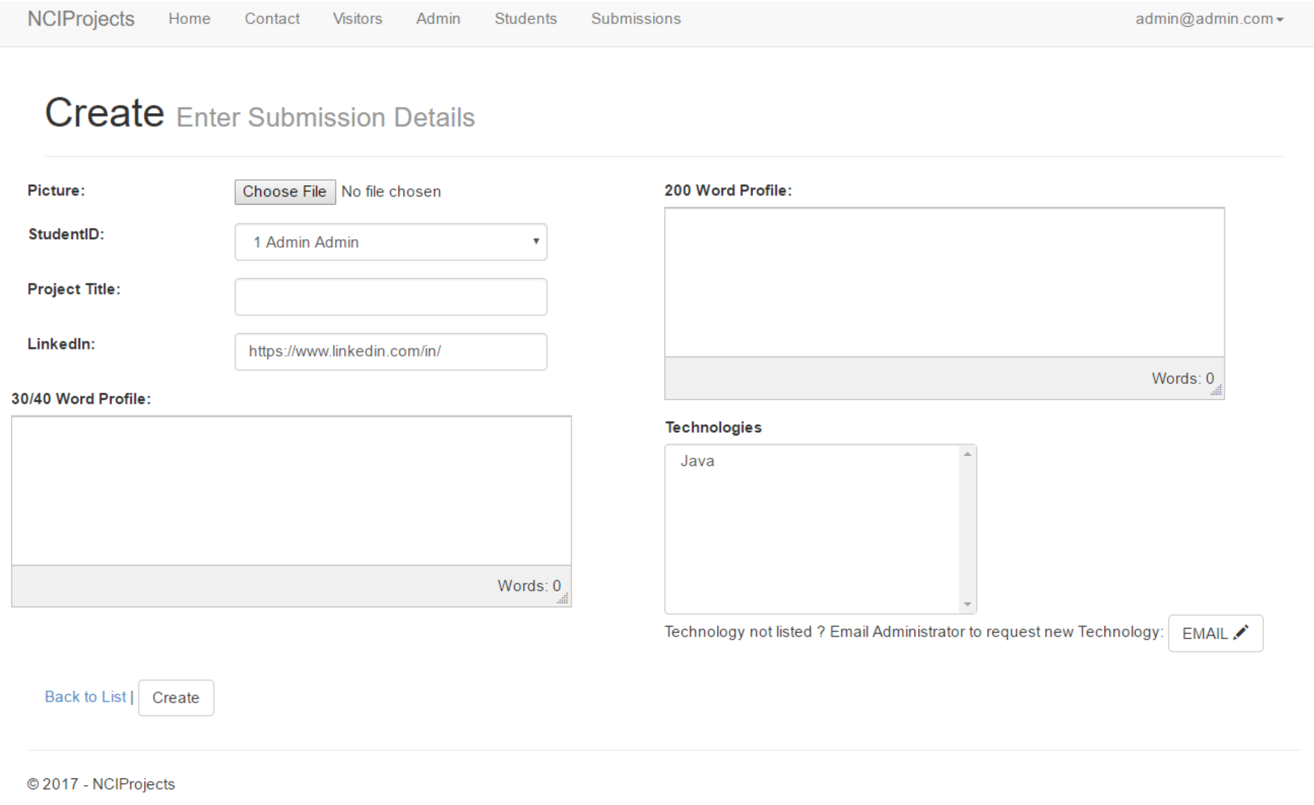
Words: 34

200 Word Profile

NCIProjects aims to make the process of the producing the Project Showcase Booklet as efficient as possible for staff and students through the development of a web portal wherein Final Year Students can upload all details of their project in a timely and straightforward manner. This project also aims to replace the currently used method of creating the Showcase Booklet and be a helpful addition for the staff in the college and shorten the process in assembling this document. The web portal will guide users through the process of entering 30/40 and 200-word profile summaries about their projects, LinkedIn profile and picture for approval by an administrator. The system will ensure data is entered correctly using strict guidelines such as number of words entered, spellcheck, list of technologies used and required picture format. When all submissions have been approved by an administrator, the Showcase Booklet will be produced in two formats by the system, one large booklet containing all student project details and a shorter, more detailed version of the booklet.

2.4.1.7 Students Create





Create Enter Submission Details

Picture: No file chosen

StudentID: 1 Admin Admin

Project Title:

LinkedIn: https://www.linkedin.com/in/

30/40 Word Profile:

Words: 0

200 Word Profile:

Words: 0

Technologies

Java

Technology not listed ? Email Administrator to request new Technology:

© 2017 - NCIProjects

2.5 Testing

This section outlines the testing methodologies carried out over the course of this project.

2.5.1 Unit Testing

Test Driven Development was conducted when developing the web application. This approached allowed me to gain good code coverage when designing, building and executing unit tests. The MSTest framework was used within the Visual Studio environment, allowing quick test design alongside the application. Some Controllers could be not accurately unit tested, therefore Manual testing and Usability testing was also conducted.

2.5.1.1 Supervisor Controller

Module ID	Module Name	Description
SU_UT	Supervisor Controller	This module allows the user to create, read, update and delete Supervisors objects

Test Author:	Nathan Ryan x13448212
Test Date:	16/03/17 – 4/5/17

Test Case ID	Description	Input Data	Expected Result	Actual Result	Pass /Fail
SUP_UT _01	Returns user to Supervisors Index page after Supervisor creation	`ID = 1, first_name = "John", last_name = "Smith", email = "jsmith@nci.ie"`	Return Supervisors/Index.html	Return Supervisors/Index.html	Pass
SUP_UT _02	Returns user to Supervisors Index page after Supervisor edit	`ID = 7, first_name = "Jim", last_name = "Jones", email = " jimjones@nci.ie"`	Return Supervisors/Index.html	Return Supervisors/Index.html	Pass
SUP_UT _03	Returns user to Supervisors Index page after Supervisor deletion	`ID = 7`	Route value to Supervisors/Index.html	Route value to Supervisors/Index.html	Pass
SUP_UT _04	Supervisor Index returns List to View	`var result = controller.Index() as ViewResult;`	Return List	Return List	Pass

Test Code snippet example:

```
// POST: Supervisors/Create
// TEST CASE ID: SUP_UT_01
        [TestMethod]
        public void Supervisors_Create_Redirect_To_Index()
        {
            // Arrange
            var controller = new SupervisorsController();
            // Act
            RedirectToRouteResult result = controller.Create(new Supervisor()
            {
                ID = 12,
                first_name = "John",
                last_name = "Smith",
                email = "jsmith@nci.ie"
            }) as RedirectToRouteResult;
            // Assert
            Assert.AreEqual(result.RouteValues["action"], "Index");
        }
```

2.5.1.2 Courses Controller

Module ID	Module Name	Description
CUT	Courses Controller	This module allows the user to create, read, update and delete Course objects

Test Author:	Nathan Ryan x13448212
Test Date:	16/03/17 – 4/5/17

Test Case ID	Description	Input Data	Expected Result	Actual Result	Pass /Fail
COU_UT _01	Returns user to Courses Index page after Course creation	ID = 1, course_name = "HCC"	Return Courses/Index.html	Return Courses/Index.html	Pass
COU_UT _02	Returns user to Courses Index page after Course edit	ID = 7, first_name = "Jim", last_name = "Jones", email = " jimjones@nci.ie"	Return Courses/Index.html	Return Courses/Index.html	Pass
COU_UT _03	Returns user to Courses Index page after Course deletion	ID = 2	Route value to Courses/Index.html	Route value to Courses/Index.html	Pass
COU_UT _04	Courses Index returns List to View	var result = controller.Index() as ViewResult;	Return List	Return List	Pass

Test Code snippet example:

```
// POST: Courses/Create
// TEST CASE ID: COU_UT_02
        [TestMethod]
        public void Courses_Edit_Redirect_To_Index()
        {
            // Arrange
            var controller = new CoursesController();
            // Act
            RedirectToRouteResult result = controller.Create(new Course()
            {
                ID = 1,
                course_name = "BSHC"
            }) as RedirectToRouteResult;
            // Assert
            Assert.AreEqual(result.RouteValues["action"], "Index");
        }
```

2.5.1.3 Streams Controller

Module ID	Module Name	Description
STR_UT	Streams Controller	This module allows the user to create, read, update and delete Stream objects

Test Author:	Nathan Ryan x13448212
Test Date:	16/03/17 – 4/5/17

Test Case ID	Description	Input Data	Expected Result	Actual Result	Pass /Fail
STR_UT _01	Returns user to Streams Index page after Stream creation	`ID = 1, stream_name = "Mobile"`	Return Streams/Index.html	Return Streams/Index.html	Pass
STR_UT _02	Returns user to Streams Index page after Stream edit	`ID = 1, stream_name = "Internet of Things"`	Return Supervisors/Index.html	Return Supervisors/Index.html	Pass
STR_UT _03	Returns user to Streams Index page after Stream deletion	ID = 1	Return Supervisors/Index.html	Return Supervisors/Index.html	Pass

Test code snippet example:

```
// POST: Streams/Edit/5
// TEST CASE ID: STR_UT_03
        [TestMethod]
        public void Streams_Delete_Redirect_To_Index()
        {
            // Arrange
            var controller = new StreamsController();
            // Act
            var result = controller.DeleteConfirmed(2) as RedirectToRouteResult;
            var routeValue = result.RouteValues["action"];
            // Assert
            Assert.AreEqual("Index", routeValue);
        }
```

2.5.1.4 Students Controller

Module ID	Module Name	Description
STU_UT	Students Controller	This module allows the user to create, read, update and delete Students objects

Test Author:	Nathan Ryan x13448212
Test Date:	16/03/17 – 4/5/17

Test Case ID	Description	Input Data	Expected Result	Actual Result	Pass/ Fail
STU_UT_01	View Student Details	`ID = 1`	Return Students/Details/1.html	Return Students/Details/1.html	Pass
STU_UT_02	Student ID Null Returns Error 400 (Bad Request)	`var result = controller.Details(null) as HttpStatusCodeResult;`	Return Bad Request page	Return Bad Request page	Pass
STU_UT_03	Student not found returns Error 404	`var result = controller.Details(1234) as HttpStatusCodeResult;`	Return Not Found page	Return Not Found page	Pass
STU_UT_04	Student Creation returns Index	`StudentID = "1111", fname = "Jim",lname = "Smith",StreamID = 1, CourseID = 1,SupervisorID = 1, email = "jsmith@nci.ie"`	Return Students/Index.html	Return Students/Index.html	Pass

Test Code snippet example:

```
// GET: Students
// TEST CASE ID: STU_UT_03
        [TestMethod]
        public void Student_Not_Found_Returns_404()
        {
            // Arrange
            var controller = new StudentsController();
            // Act
            var result = controller.Details(1) as HttpStatusCodeResult;
            // Assert
            Assert.IsNotNull(result);
            Assert.AreEqual(404, result.StatusCode);
        }
```

2.5.1.5 Submissions Controller

Module ID	Module Name	Description
SUB_UT	Submissions Controller	This module allows the user to create, read, update and delete project submissions and generate reports.

Test Author:	Nathan Ryan x13448212
Test Date:	16/03/17 – 4/5/17

Test Case ID	Description	Input Data	Expected Result	Actual Result	Pass /Fail
SUB_UT _01	Draft Booklet Returns PDF File	var result = controller.GenerateDraftBooklet() as ActionResult;	Return PDF	Return PDF	Pass

Test Code snippet example:

```
// TEST CASE ID: SUB_UT_01
        [TestMethod]
        public void Draft_Booklet_Returns_PDF()
        {
            // Arrange
            var controller = new SubmissionsController();
            // Act
            var result = controller.GenerateDraftBooklet() as ActionResult;
            // Assert
            Assert.IsInstanceOfType(result, typeof(ActionResult));
        }
```

2.5.2 Manual Testing

System testing was carried to test functions where unit-testing was not appropriate such as generating reports, uploading files and verifying authorizations.

2.5.2.1 Submissions Controller

Test Case ID	Date	Description	Scenario	Prerequisites
SUB_ST_01	12/04/17	Large Booklet Generation returns PDF	Student wants to preview large booklet with details submitted	Student must be registered to system

Step #	Step Details	Expected	Actual	Outcome
1	Log in to application (user= 1234, pass= Hello!123)	Enter correct login details, redirected to Submissions page	As expected	Pass
2	Open user submission	Navigation to submission	As expected	Pass
3	Produce report	Click preview '40-word' profile button	As expected	Pass
4	PDF Output	Report is produced in browser window	Returned to Submissions page (picture not uploaded)	Fail
5	PDF Output	Report is produced in browser	As expected	Pass

Test Case ID	Date	Description	Scenario	Prerequisites
SUB_ST_02	12/04/17	Small Booklet Generation returns PDF	Student wants to preview small booklet with details submitted	Student must be registered to system

Step #	Step Details	Expected	Actual	Outcome
1	Log in to application (user= 1234, pass= Hello!123)	Enter correct login details, redirected to Submissions page	As expected	Pass
2	Open user submission	Navigation to submission	As expected	Pass
3	Produce report	Click preview '200-word' profile button	As expected	Pass
4	PDF Output	Report is produced in browser	As expected	Pass

Test Case ID	Date	Description	Scenario	Prerequisites
SUB_ST_03	12/04/17	Submit Project Details	Student wants to submit 40-word profile	Student must be registered to system

Step #	Step Details	Expected	Actual	Outcome
1	Log in to application (user= 1234, pass= Hello!123)	Enter correct login details, redirected to Submissions page	As expected	Pass
2	Create new submission	Click 'Create New' button on submissions page	As expected	Pass
3	Enter all required data	Enter project details data (select student number, enter title, linkedin url, 40 word description, select technologies, select picture)	As expected	Pass
4	Return to Submissions details	All data is valid, return to details page of submission.	As expected	Pass

2.5.2.2 Admin Controller

Test Case ID	Date	Description	Scenario	Prerequisites
ADM_ST_01	15/04/17	Upload CSV File	Admin wants to 'pre-load' new student details to system	Admin must have CSV file

Step #	Step Details	Expected	Actual	Outcome
1	Log in to application (user= admin, pass= Admin!123)	Enter correct login details, redirected to Submissions page	As expected	Pass
2	Open Admin Dashboard	Navigate to Dashboard from navigation	As expected	Pass
3	Open CSV File Upload area	Navigate to CSV File Upload	As expected	Pass
5	Choose file	File is selected from machine	As expected	Pass
6	Upload	Select 'Submit' button	Error returned (columns in incorrect order)	Fail
7	Upload	Select 'Submit' button	As expected	Pass
8	Rows returned	Rows of data entered displayed in View	As expected	Pass

2.5.3 User Testing

User testing carried out includes the Trunk Test and Think Aloud tests.

2.5.3.1 Trunk Test

Trunk testing follows the logic that a typical user will scan a web page instead of reading all information. The Trunk Test is carried out on a random page on the web application and the user is asked to locate the following items in the table as quickly as possible.

Questions	Participant Answers
Task No.	TT_01
Start Time	11.33 4/4/17
End Time	11.35
Site ID	NCIProjects
Page Name	Student Submissions
Sections (What are the major sections? Are the major sections outlined?)	Home, Contact, Submissions, Email
Local Navigation (What are my options at this level)	There is a way to go scroll through different pages at the bottom of each page. Can also select button under search bar to create new submission on new page.
Where am I?	The Student Submissions (clear from the header at top of page)
How can I search?	There is a search input under the navigation bar to search for students by name.
Additional Notes	User says site is laid out well and easily navigable. Likes the search function.

Questions	Participant Answers
Task No.	TT_02
Start Time	11.37 4/4/17
End Time	11.40
Site ID	NCIProjects
Page Name	Project Showcase Submission Details
Sections (What are the major sections? Are the major sections outlined?)	Home, Contact, Submissions, Email
Local Navigation (What are my options at this level)	There is an edit and delete button at the top of the page and preview 40 and 200-word profile links underneath student details.
Where am I?	The Student Submissions (clear from the header at top of page)
How can I search?	There is no search.
Additional Notes	Preview links were not immediately visible when trying to navigate page.

Questions	Participant Answers
Task No.	TT_03
Start Time	10.22 5/4/17
End Time	10.25
Site ID	NCIProjects
Page Name	Admin Dashboard
Sections (What are the major sections? Are the major sections outlined?)	Home, Contact, Submissions, Email
Local Navigation (What are my options at this level)	You can select different areas of the website from the links provided on the page. There are links to edit submissions, students and supervisors. There are also buttons to generate reports and upload CSV files at the bottom of the page.
Where am I?	Admin Dashboard
How can I search?	There is no way to search.
Additional Notes	User says dashboard is useful for Admin user to be able to navigate to different areas of the site.

Questions	Participant Answers
Task No.	TT_04
Start Time	10.30 5/4/17
End Time	10.33
Site ID	NCIProjects
Page Name	Students
Sections (What are the major sections? Are the major sections outlined?)	Home, Contact, Submissions, Email
Local Navigation (What are my options at this level)	You can create a new student using the ‘Create New’ button at top of the page.
Where am I?	Students details area
How can I search?	There is a search input above student panels to search for student by name and student number.
Additional Notes	-

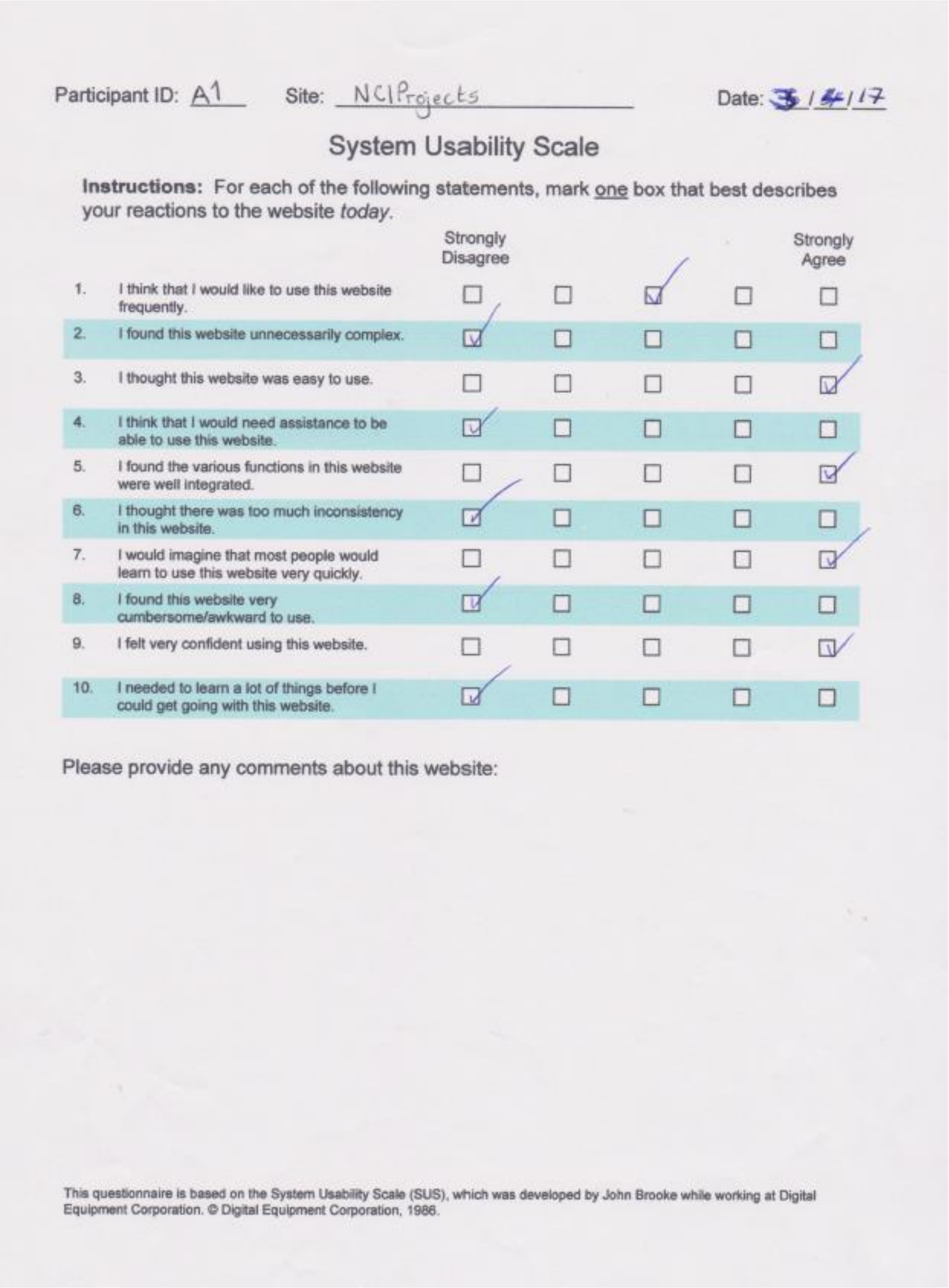
2.5.3.2 Think Aloud

Think Aloud tests are conducted to allow the participant to interact with the system – perform tasks. The participant is asked to provide feedback as the tasks are conducted. Think Aloud tests allows us to understand how users will approach the system and how usable the system is.

Task No.	ADMIN01_TASK01
Task Goal	Accept Student Submission
Start Time	13.32 3/4/17
End Time	13.34
Expected Behaviour	- Admin user logs in with Admin credentials. - Admin is redirected to Submissions Index page. - System presents list of submissions created by students. - Admin selects a Submission from the list. - Submissions Details for that Student is returned. - If all information is relevant and of satisfies requirements, Admin selects 'APPROVE SUBMISSION' button. - System confirms that Submission has been approved.
Actual Behaviour	- Admin logged in with Admin credentials. - Admin is redirected to Submissions Index page. - Admin selected student submission from list. - Admin reviewed information to make sure all is up to standard. - Admin selected 'APPROVE SUBMISSION' button. - System changed status of Submission to 'Approved'.
Notes	- User would like to have the details link on Submissions Index page to be more visible (possibly a button).

Task No.	ADMIN01_TASK02
Task Goal	Edit Student Details
Start Time	13.36 3/4/17
End Time	13.39
Expected Behaviour	- Admin user logs in with Admin credentials. - Admin is redirected to Submissions Index page. - Admin navigates to Admin Dashboard. - Admin selects Students Tab from Admin Dashboard. - System presents Admin with panel list of all Students registered to system. - Admin selects 'EDIT' link on student panel. - System presents Admin with form with Student info retrieved from Database. - Admin edits student information. - Admin selects 'SAVE' button. - Admin is returned to list of Students.
Actual Behaviour	- Admin user logged in with Admin credentials. - Admin navigated to Admin Dashboard. - Admin selected Students Tab from Admin Dashboard. - System presents Admin with panel list of all Students registered to system. - Admin selected 'EDIT' link on student panel. - System presents Admin with form with Student info retrieved from Database. - Admin edits student information. - Admin selected 'SAVE' button. - Admin is returned to list of Students.
Notes	- Admin User would like to have the system return the student details page after saving info instead of returning to list of students.

Task No.	ADMIN01_TASK03
Task Goal	View Preview Booklet
Start Time	13.41 3/4/17
End Time	13.43
Expected Behaviour	- Admin user logs in with Admin credentials. - Admin is redirected to Submissions Index page. - System presents list of submissions created by students. - Admin selects a Submission from the list. - Submissions Details for that Student is returned. - Admin reviews submitted data. - Admin selects either 'Preview 40 Word Profile' or 'Preview 200 Word Profile' button under Student Information panel. - System outputs PDF file to user's machine.
Actual Behaviour	- Admin user logged in with Admin credentials. - Admin is redirected to Submissions Index page. - System presents list of submissions created by students. - Admin selected a Submission from the list. - Submissions Details for that Student is returned. - Admin reviewed submitted data. - Admin selected either 'Preview 40 Word Profile' button under Student Information panel. - System returned PDF file to user's machine.
Notes	- Admin User would like the 'Preview' buttons to have colours as they were not immediately visible (white buttons). - User would like the PDF file to be shown in browser window with option to download rather than download automatically.



Participant ID: A1 Site: NCIProjects Date: 3/4/17

System Usability Scale

Instructions: For each of the following statements, mark <u>one</u> box that best describes your reactions to the website *today*.

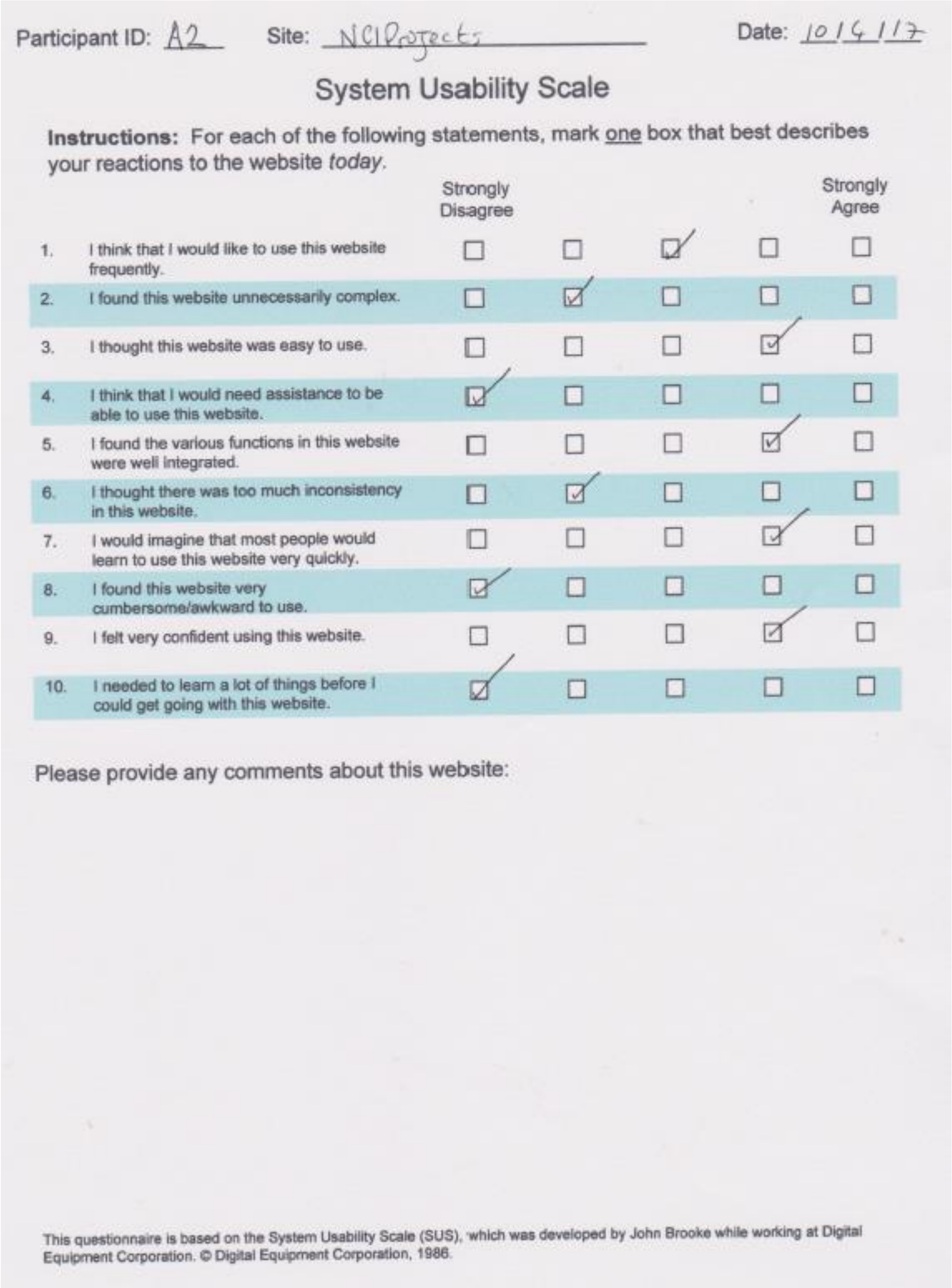
		Strongly Disagree				Strongly Agree
1.	I think that I would like to use this website frequently.	□	□	☑	□	□
2.	I found this website unnecessarily complex.	☑	□	□	□	□
3.	I thought this website was easy to use.	□	□	□	□	☑
4.	I think that I would need assistance to be able to use this website.	☑	□	□	□	□
5.	I found the various functions in this website were well integrated.	□	□	□	□	☑
6.	I thought there was too much inconsistency in this website.	☑	□	□	□	□
7.	I would imagine that most people would learn to use this website very quickly.	□	□	□	□	☑
8.	I found this website very cumbersome/awkward to use.	☑	□	□	□	□
9.	I felt very confident using this website.	□	□	□	□	☑
10.	I needed to learn a lot of things before I could get going with this website.	☑	□	□	□	□

Please provide any comments about this website:

This questionnaire is based on the System Usability Scale (SUS), which was developed by John Brooke while working at Digital Equipment Corporation. © Digital Equipment Corporation, 1986.

Task No.	ADMIN02_TASK01
Task Goal	Upload CSV File
Start Time	10.32 10/4/17
End Time	10.36
Expected Behaviour	- Admin user logs in with Admin credentials. - Admin is redirected to Submissions Index page. - System presents list of submissions created by students. - Admin navigates to Admin Dashboard. - Admin selects ‘Upload CSV File’ link. - System presents user with form to select CSV file from machine. - User selects the relevant file and clicks ‘Submit’ to upload file. - System presents user with list of data written to Database.
Actual Behaviour	- Admin user logged in with Admin credentials. - Admin is redirected to Submissions Index page. - System presented list of submissions created by students. - Admin navigated to Admin Dashboard. - Admin selected ‘Upload CSV File’ link. - System presents user with form to select CSV file from machine. - User selects the wrong file and clicks ‘Submit’ to upload file. - System tells user that they must upload valid CSV file. - User selects correct file from machine and clicks ‘Submit’ to upload file. - System confirms upload and presents user with list if data written to Database.
Notes	- User likes the list output that informs them of the files written to database.

Task No.	ADMIN02_TASK02
Task Goal	View Draft Booklet
Start Time	10.40 10/4/17
End Time	10.44
Expected Behaviour	- Admin user logs in with Admin credentials. - Admin is redirected to Submissions Index page. - System presents list of submissions created by students. - Admin selects ‘Draft’ button at top of page. - System outputs PDF file to user’s browser.
Actual Behaviour	- Admin user logged in with Admin credentials. - Admin is redirected to Submissions Index page. - System presented list of submissions created by students. - Admin selected ‘Draft’ button at top of page. - System outputs PDF file to user’s browser.
Notes	-



Participant ID: A2 Site: NCIProjects Date: 10/4/17

System Usability Scale

Instructions: For each of the following statements, mark one box that best describes your reactions to the website *today*.

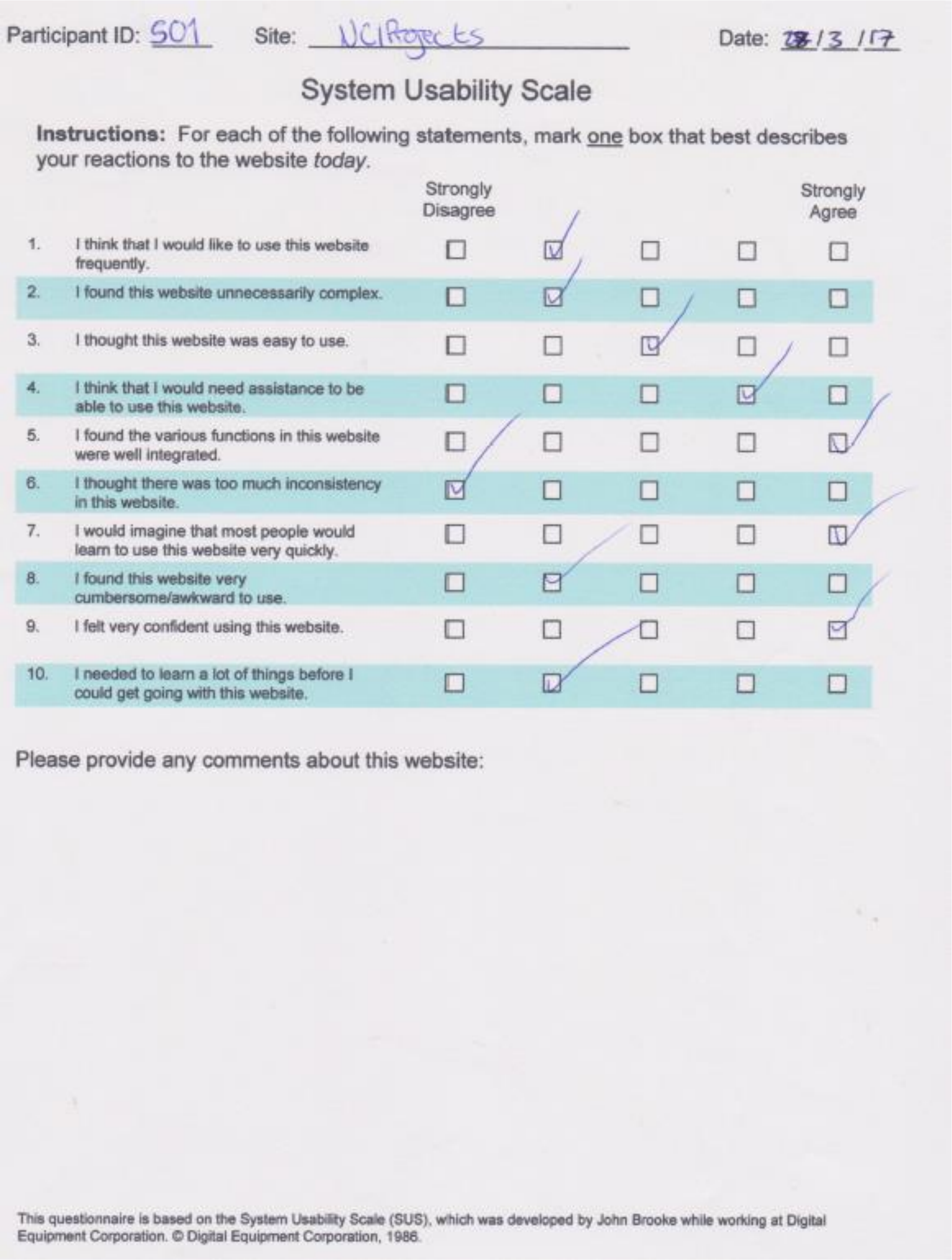
		Strongly Disagree				Strongly Agree
1.	I think that I would like to use this website frequently.	□	□	☑	□	□
2.	I found this website unnecessarily complex.	□	☑	□	□	□
3.	I thought this website was easy to use.	□	□	□	☑	□
4.	I think that I would need assistance to be able to use this website.	☑	□	□	□	□
5.	I found the various functions in this website were well integrated.	□	□	□	☑	□
6.	I thought there was too much inconsistency in this website.	□	☑	□	□	□
7.	I would imagine that most people would learn to use this website very quickly.	□	□	□	☑	□
8.	I found this website very cumbersome/awkward to use.	☑	□	□	□	□
9.	I felt very confident using this website.	□	□	□	☑	□
10.	I needed to learn a lot of things before I could get going with this website.	☑	□	□	□	□

Please provide any comments about this website:

This questionnaire is based on the System Usability Scale (SUS), which was developed by John Brooke while working at Digital Equipment Corporation. © Digital Equipment Corporation, 1986.

Task No.	STUDENT01_TASK01
Task Goal	Register Student Account
Start Time	10.45 28/3/17
End Time	10.48
Expected Behaviour	- From the ‘Home’ page the student user selects ‘Register’ from the navigation bar. - The system presents the user with fields to input student email, student number and password. - The student enters the required information and selects the ‘Register’ button. - The system redirects user to the Submissions index page.
Actual Behaviour	- User selected ‘Log In’ from navigation bar. - User selected ‘Register’ link under form on ‘Log In’ page. - User entered required information into form for registration. - Password entered in to the system did not meet criteria defined in code. - System presents user with information as to why password did not meet criteria (must have at least one symbol and uppercase letter). - User corrected password to match restrictions. - User Registers details to system. - User is redirected to Submissions index page.
Notes	- User was unsure whether to enter student email or personal email into Registration form - Student was unsure whether to add an ‘x’ at the start of their student number. - Password restrictions were unclear.

Task No.	STUDENT01_TASK02
Task Goal	Enter Submission
Start Time	10.52 28/3/17
End Time	10.56
Expected Behaviour	- From the 'Home' Page the (Registered) student user selects 'Log In' from navigation bar or Jumbotron bootstrap element. - The user enters relevant information into Log In form. - The user logs in to system. - The user selects 'Create New' button. - The system presents the user with fields to input their student profile picture, student number, project title, LinkedIn URL, project description (30/40 and 200 Word Profile) and selected technologies. - The user enters all required data in the fields. - The user selects 'Submit' button. - User is returned to their submission details page.
Actual Behaviour	- User selected 'Log In' from Jumbotron on the 'Home' page. - User entered correct login details and logs in to system. - User navigates to Submission Index page. - User selected 'Create New' button. - User entered all required data into fields presented by system. - User selected 'Submit' button. - User is returned to their submission details page.
Notes	- User would like to be redirected to Submissions Index page on Log In to system. - User has concerns about other users being able to select a different user when editing their own submission.



Participant ID: S01 Site: NCIProjects Date: 28/3/17

System Usability Scale

Instructions: For each of the following statements, mark one box that best describes your reactions to the website *today*.

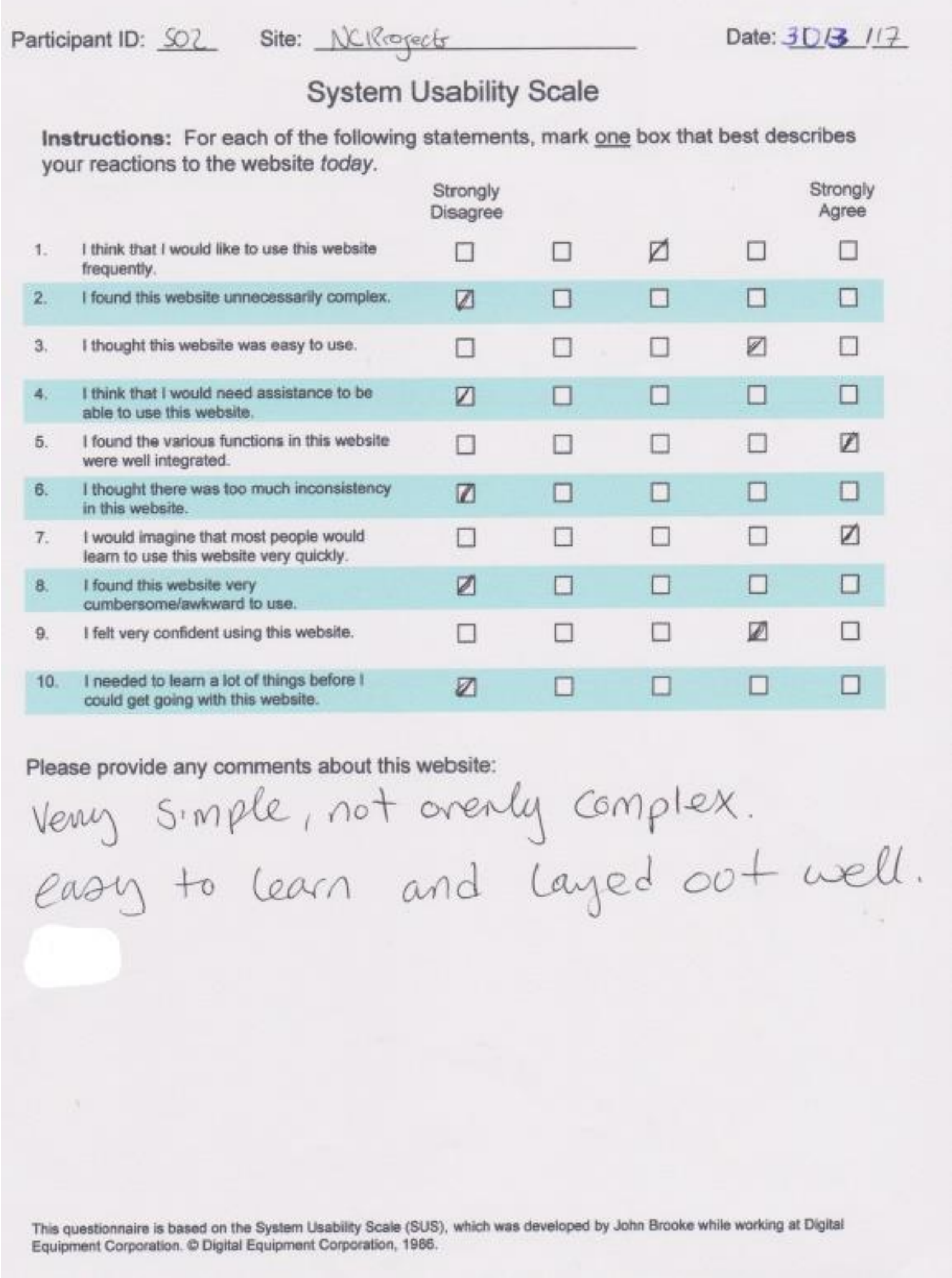
		Strongly Disagree				Strongly Agree
1.	I think that I would like to use this website frequently.	□	☑	□	□	□
2.	I found this website unnecessarily complex.	□	☑	□	□	□
3.	I thought this website was easy to use.	□	□	☑	□	□
4.	I think that I would need assistance to be able to use this website.	□	□	□	☑	□
5.	I found the various functions in this website were well integrated.	□	□	□	□	☑
6.	I thought there was too much inconsistency in this website.	☑	□	□	□	□
7.	I would imagine that most people would learn to use this website very quickly.	□	□	□	□	☑
8.	I found this website very cumbersome/awkward to use.	□	☑	□	□	□
9.	I felt very confident using this website.	□	□	□	□	☑
10.	I needed to learn a lot of things before I could get going with this website.	□	☑	□	□	□

Please provide any comments about this website:

This questionnaire is based on the System Usability Scale (SUS), which was developed by John Brooke while working at Digital Equipment Corporation. © Digital Equipment Corporation, 1986.

Task No.	STUDENT02_TASK01
Task Goal	Register Student Account
Start Time	12.36 30/3/17
End Time	12.38
Expected Behaviour	- From the ‘Home’ page the student user selects ‘Register’ from the navigation bar. - The system presents the user with fields to input student email, student number and password. - The student enters the required information and selects the ‘Register’ button. - The system redirects user to the Submissions index page.
Actual Behaviour	- User selected ‘Register’ from navigation bar. - User entered required information into form for registration. - User entered required information. - User Registers details to system. - User is redirected to Submissions Index page.
Notes	- User did not see password restriction criteria at first. - User thinks the password restriction alert could be placed closer to input.

Task No.	STUDENT02_TASK02
Task Goal	Enter Submission
Start Time	12.40 30/3/17
End Time	12.44
Expected Behaviour	- From the 'Home' Page the (Registered) student user selects 'Log In' from navigation bar or Jumbotron bootstrap element. - The user enters relevant information into Log In form. - The user logs in to system and is redirected to Submissions Index page. - The user selects 'Create New' button. - The system presents the user with fields to input their student profile picture, student number, project title, LinkedIn URL, project description (30/40 and 200 Word Profile) and selected technologies. - The user enters all required data in the fields. - The user selects 'Submit' button. - User is returned to their submission details page.
Actual Behaviour	- User selected 'Log In' from Navigation Bar on the 'Home' page. - User entered correct login details and logs in to system. - User is redirected to Submissions Index page. - User selected 'Create New' button. - User entered all required data into fields presented by system. - User selected 'Submit' button. - User is returned to their submission details page.
Notes	- User likes layout of create and details submission pages. - User thinks the Preview booklet buttons could be different colours to make them more visible.



Participant ID: S02 Site: NCIProjects Date: 3/13 /17

System Usability Scale

Instructions: For each of the following statements, mark one box that best describes your reactions to the website *today*.

		Strongly Disagree				Strongly Agree
1.	I think that I would like to use this website frequently.	□	□	☒	□	□
2.	I found this website unnecessarily complex.	☒	□	□	□	□
3.	I thought this website was easy to use.	□	□	□	☒	□
4.	I think that I would need assistance to be able to use this website.	☒	□	□	□	□
5.	I found the various functions in this website were well integrated.	□	□	□	□	☒
6.	I thought there was too much inconsistency in this website.	☒	□	□	□	□
7.	I would imagine that most people would learn to use this website very quickly.	□	□	□	□	☒
8.	I found this website very cumbersome/awkward to use.	☒	□	□	□	□
9.	I felt very confident using this website.	□	□	□	☒	□
10.	I needed to learn a lot of things before I could get going with this website.	☒	□	□	□	□

Please provide any comments about this website:

Very Simple, not overly complex.
easy to learn and layed out well.

[redacted]

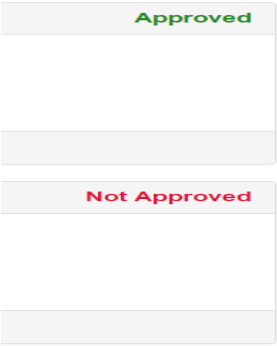
This questionnaire is based on the System Usability Scale (SUS), which was developed by John Brooke while working at Digital Equipment Corporation. © Digital Equipment Corporation, 1986.

2.5.3.3 Heuristic Evaluation

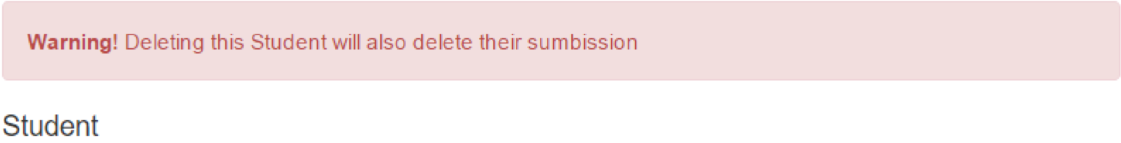
For the heuristic evaluation, each of the testing participants were asked to review one heuristic. Their feedback has been summarised in this section.

Visibility of System Status

The site will inform users of the status of the submission on the Submission Index and Details page.



The site will also inform Admin users what their actions will do when modifying contents of the database, such as deleting a student profile.



Warning! Deleting this Student will also delete their sumbission

Student

Match between System and the Real World

The site uses clear and understandable language. The forms for data input are labelled clearly and are relevant to the data that is needed. No ambiguity.

User Control and Freedom

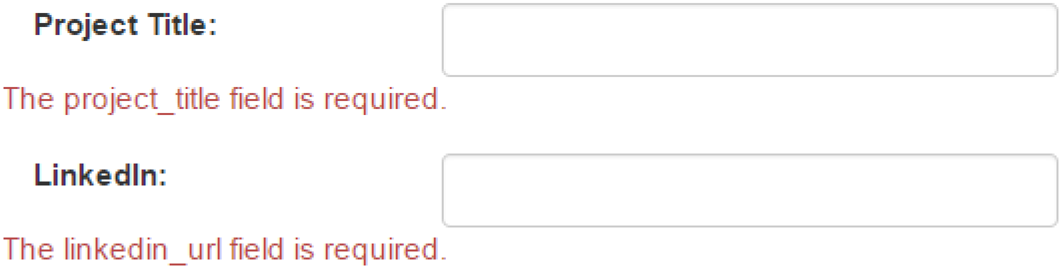
User freedom is somewhat restricted as student users will be required to just enter their project submission. Users can exit from creation of their project submission and can edit or delete their submission.

Consistency and Standards

The site is consistent in a simple, professional design, using the Bootstrap this is achieved easily. The navigation is fixed to the top of every page allowing for easy navigation to anywhere on the site in a few clicks.

Error Prevention

Error prevention can be found on the project submission creation page. Student users must input the required fields to submit their data.



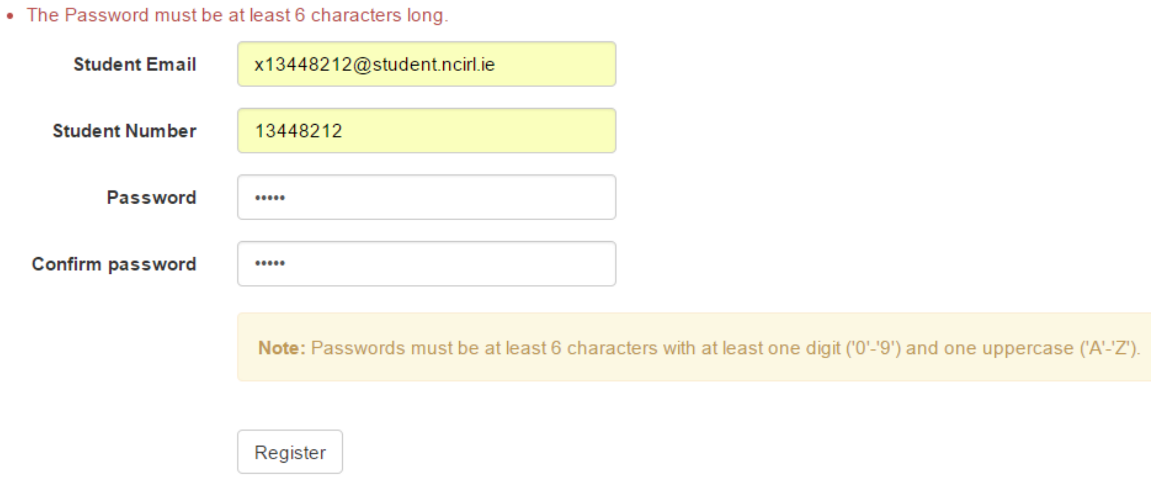
Project Title:

The project_title field is required.

LinkedIn:

The linkedin_url field is required.

Users also required to have a valid NCI email address and provide a secure password required by the system to register.



- The Password must be at least 6 characters long.

Student Email: x13448212@student.ncirl.ie

Student Number: 13448212

Password: •••••

Confirm password: •••••

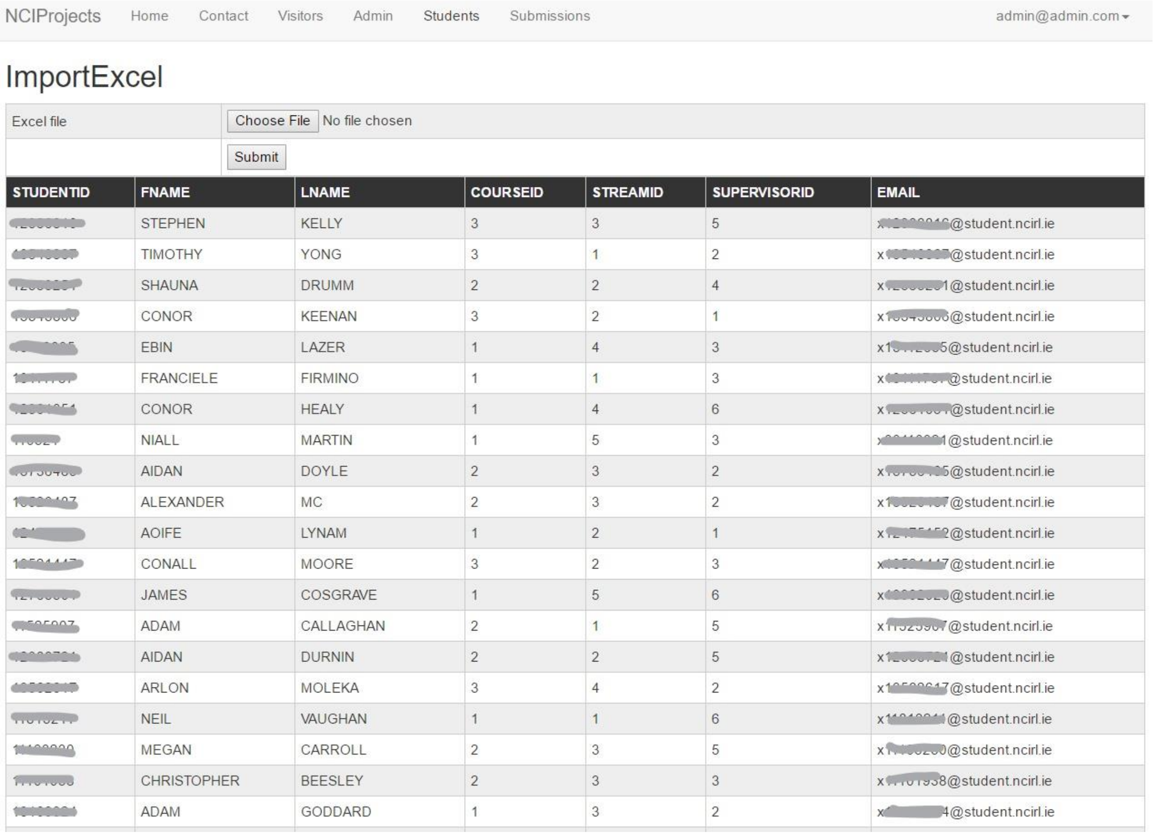
Note: Passwords must be at least 6 characters with at least one digit ('0'-'9') and one uppercase ('A'-'Z').

These error controls prevent data integrity from being compromised.

Flexibility and Efficiency of Use

The site is responsive to mobile and web users.

The system allows for the uploading of CSV files to enable the Admin user to quickly add new student info. This page also displays all rows added to database ensuring the Admin user knows what data has been written.



ImportExcel

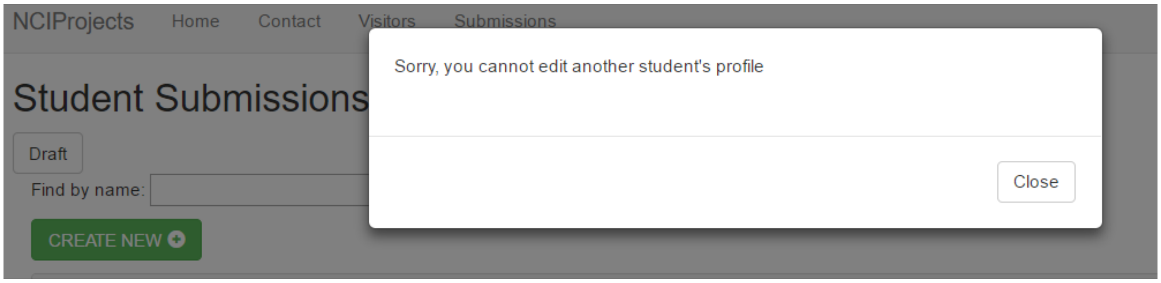
Excel file: Choose File No file chosen

Submit

STUDENTID	FNAME	LNAME	COURSEID	STREAMID	SUPERVISORID	EMAIL
[redacted]	STEPHEN	KELLY	3	3	5	[redacted]@student.ncirl.ie
[redacted]	TIMOTHY	YONG	3	1	2	x[redacted]@student.ncirl.ie
[redacted]	SHAUNA	DRUMM	2	2	4	x[redacted]@student.ncirl.ie
[redacted]	CONOR	KEENAN	3	2	1	x[redacted]@student.ncirl.ie
[redacted]	EBIN	LAZER	1	4	3	x[redacted]@student.ncirl.ie
[redacted]	FRANCIELE	FIRMINO	1	1	3	x[redacted]@student.ncirl.ie
[redacted]	CONOR	HEALY	1	4	6	x[redacted]@student.ncirl.ie
[redacted]	NIALL	MARTIN	1	5	3	[redacted]@student.ncirl.ie
[redacted]	AIDAN	DOYLE	2	3	2	x[redacted]@student.ncirl.ie
[redacted]	ALEXANDER	MC	2	3	2	x[redacted]@student.ncirl.ie
[redacted]	AOIFE	LYNAM	1	2	1	x[redacted]@student.ncirl.ie
[redacted]	CONALL	MOORE	3	2	3	x[redacted]@student.ncirl.ie
[redacted]	JAMES	COSGRAVE	1	5	6	x[redacted]@student.ncirl.ie
[redacted]	ADAM	CALLAGHAN	2	1	5	x[redacted]@student.ncirl.ie
[redacted]	AIDAN	DURNIN	2	2	5	x[redacted]@student.ncirl.ie
[redacted]	ARLON	MOLEKA	3	4	2	x[redacted]@student.ncirl.ie
[redacted]	NEIL	VAUGHAN	1	1	6	x[redacted]@student.ncirl.ie
[redacted]	MEGAN	CARROLL	2	3	5	x[redacted]@student.ncirl.ie
[redacted]	CHRISTOPHER	BEESLEY	2	3	3	x[redacted]@student.ncirl.ie
[redacted]	ADAM	GODDARD	1	3	2	x[redacted]@student.ncirl.ie

Help users recognize, diagnose, and recover from errors

Students cannot edit another student’s submission, trying to do this will result in that user being redirected to the Submissions Index page with the following message.

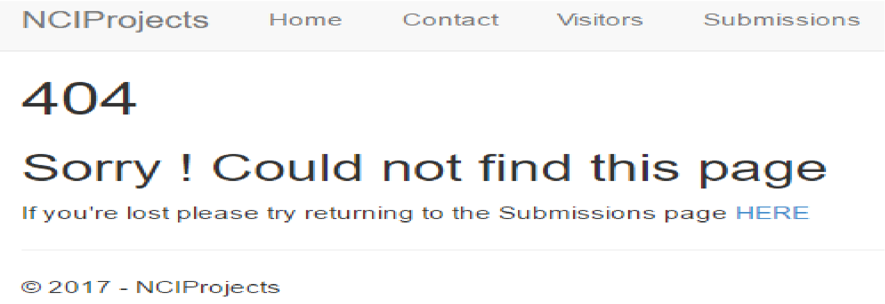


Student Submissions

Find by name:

Sorry, you cannot edit another student's profile

The site also provides a custom ‘404’ Error page guiding users back to the Submissions page should they stray onto a link or area that may have expired.



404

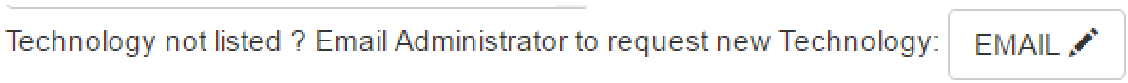
Sorry ! Could not find this page

If you're lost please try returning to the Submissions page HERE

© 2017 - NCIProjects

Help and Documentation

Users can email the Administrator of the site if the technology they used in their project is not listed



Technology not listed ? Email Administrator to request new Technology:

2.5.3.4 Usability Testing Outcome

Usability testing was a quick way to get immediate feedback on different aspects of the site and changes were implemented rapidly after feedback was received. From carrying out the Think Aloud tests with users I could see how they would interact with the system without intervention. This was a good way of gauging if the features on the site were too complex or required too much guesswork to be completed. From the System Usability Scale (SUS) feedback forms I got an average score of: 85 which is above the SUS average. This score made me feel confident about the usability aspect of the site going forward. The navigation aspect of the site was reviewed by conducting Trunk Tests. Trunks tests allowed me to sort out any issues with navigation quickly and effectively. Finally, the Heuristic Evaluation was very useful to the project. Each user that participated in Usability Testing was asked to review one heuristic, this allowed me site down with the user and go through the site and gain feedback in a more detailed way.

3 Conclusions

The purpose of this document was to outline the processes that went into planning, designing, developing and testing NCIProjects to build a modern and efficient Final Year Project Showcase Web-Portal to replace legacy systems and methods employed in the college. The project has satisfied the requirements outlined at the start of development including user, functional and non-functional and usability requirements and will hopefully be of assistance to the college staff and students in the future project showcases.

Usability Testing was an important aspect of this project as it was important to understand how users would interact with the system and what aspects of the site were particularly hard or easy to work with.

Opportunities of NCIProjects:

- The biggest opportunity that this project has is the efficiency the system can bring to students and staff when compiling the annual project showcase booklets, freeing up time to commit to other crucial tasks.
- Administrators (lecturers) can check the status of each individual submission and track the number of submissions entered.
- Built with ASP.NET MVC allowing for portability and expansion.
- Could be integrated within the college network and further developed to include a number of different course years, streams etc.

Limits of NCIProjects:

- Has a particular user base, would only mainly be used within NCI.
- Free tier of Azure used in development of the project limits the database size to 30mb.
- Free tier of Azure used in development also limits API/Plugin use, some features such as email could only be achieved using external methods.

4 Further development or research

With more resources, this project could develop in several ways. The main resources that were limiting factors was time and money. Some services offered through Azure and AWS were not available due to being limited to the free tier. For this project the user requirements were clear about exactly what is needed i.e. generating reports for users, but in future the system could incorporate other aspects of the showcase including an area for Showcase Poster design tutorials, resources and upload for Administrators to view and approve poster designs before they are sent to the printers for use in the project showcase.

The web portal could also be extended to include other course year’s projects, much like Moodle. Users could be required to upload details of their project and the system will collate the information for Administrators for use in end-of-semester presentations.

If the college required the system could also be extended to incorporate third-party logins, such as if the college wanted to add the web portal to their internal network like NCI360.

5 References

Brind, M. (2015). ASP.NET MVC 5 with EF 6 - Working With Files. [online] Mikesdotnetting.com. Available at: http://www.mikesdotnetting.com/article/259/asp-net-mvc-5-with-ef-6-working-with-files [Accessed 1 Dec. 2016].

Chapter 3: Architectural Patterns and Styles. (n.d.). [online] Msdn.microsoft.com. Available at: https://msdn.microsoft.com/en-us/library/ee658117.aspx [Accessed 19 Feb. 2017].

Communication Diagram (formerly Collaboration Diagram) [EA User Guide]. (n.d.). [online] Icodeguru.com. Available at: http://www.icodeguru.com/design/Enterprise-Architect/communicationdiagram.htm [Accessed 17 Feb. 2017].

Danylko, J. (2016). The Ultimate Guide To Unit Testing in ASP.NET MVC. [online] DanylkoWeb. Available at: https://www.danylkoweb.com/Blog/the-ultimate-guide-to-unit-testing-in-aspnet-mvc-E2 [Accessed 2 Mar. 2017].

Developers.itextpdf.com. (n.d.). Examples hub | iText Developers. [online] Available at: http://developers.itextpdf.com/examples [Accessed 25 Feb. 2017].

dotnetthoughts. (2014). Creating Unit Tests for ASP.NET MVC 6 Applications. [online] Available at: http://dotnetthoughts.net/creating-unit-tests-for-asp-net-mvc-6-applications/ [Accessed 1 Mar. 2017].

Dykstra, T. and Anderson, R. (2014). Getting Started with Entity Framework 6 Code First using MVC 5. 1st ed. Microsoft Technologies.

Forms Authentication in ASP.NET MVC 4. (2013). [online] Code.Util. Available at: https://codeutil.wordpress.com/2013/05/14/forms-authentication-in-asp-net-mvc-4/ [Accessed 14 Feb. 2017].

Khan, M. (2013). Create PDF Report from database in ASP.Net using C# and VB.Net. [online] Aspsnippets.com. Available at: http://www.aspsnippets.com/Articles/Create-PDF-Report-from-database-in-ASPNet-using-C-and-VBNet.aspx [Accessed 26 Feb. 2017].

Khan, M. (2014). Import (Upload) CSV file data to SQL Server database in ASP.Net using C# and VB.Net. [online] Aspsnippets.com. Available at: http://www.aspsnippets.com/Articles/Import-Upload-CSV-file-data-to-SQL-Server-database-in-ASPNet-using-C-and-VBNet.aspx [Accessed 5 Feb. 2017].

Lin, C. (2014). Lifecycle of an ASP.NET MVC 5 Application. [online] Docs.microsoft.com. Available at: https://docs.microsoft.com/en-us/aspnet/mvc/overview/getting-started/lifecycle-of-an-aspnet-mvc-5-application [Accessed 19 Feb. 2017].

Msdn.microsoft.com. (n.d.). ASP.NET MVC Overview. [online] Available at: https://msdn.microsoft.com/en-us/library/dd381412(v=vs.108).aspx [Accessed 2 Nov. 2016].

Punhani, R. (2016). Read Data From Excel File (xls, xlsx, csv) In ASP.NET MVC. [online] C-sharpcorner.com. Available at: http://www.c-sharpcorner.com/UploadFile/dbb82e/read-data-from-excel-filexlsxlsxcsv-in-Asp-Net-mvc/ [Accessed 12 Mar. 2017].

Security, Authentication, and Authorization. (n.d.). [online] The Official Microsoft ASP.NET Site. Available at: https://www.asp.net/mvc/overview/security [Accessed 15 Feb. 2017].

6 Appendix

6.1 Project Proposal

Project Proposal

Showcase Framework Application

Nathan Ryan, x13448212, nathan.ryan26@gmail.com

BSc (Hons) in Computing

Mobile Application Development Stream

21/10/16

6.1.1 Objective

The main objective of this project is to develop a working web-based Project Showcase portal intended for use by Final Year Students and College Lecturers. It will endeavor to assist both students and faculty in ensuring personal and project details are entered into the system in a clear and uniform manner.

This project aims to remove the ambiguity students face when delivering project details for use in the Project Showcase. This portal will guide users through the process of entering short and long summaries about their projects for approval by an administrator (Lecturer). Short summaries will contain 40-60 words with the student’s picture and LinkedIn profile link. A longer summary will contain 200 words with the student’s picture and technologies used in developing the project.

The system will ensure data is entered correctly using strict guidelines such as number of words entered, spellcheck, list of technologies used, required picture format etc...

Students can preview their submissions on entry and view their individual upload to review their entry.

When data from user has been entered correctly, an administrator will have the task of ensuring the quality of the summaries. If approved, the student will receive an email detailing why their submission was denied. Similarly, students will receive an email if their submission has been approved.

The administrator will have their own dashboard enabling them to view the number of students in the year and how many have made submissions.

The system will be available as a web-based portal accessible on both PCs and mobile devices allowing users to upload a profile picture and screenshots of their project from their device for use in the project showcase.

The system will store all data from users and when all submissions have been approved by an administrator, Marketing users from the college can download the project submissions as the Showcase Booklet.

I also aim to develop a high quality application that can be used within the college for future final year students.

6.1.2 Background

Every year the college (NCI) requires Final Year students to submit details about their projects in a specified format – a small summary of 40 words and a larger summary of 200 words. These summaries are then compiled by NCI staff into a project showcase booklet. The problem with this approach is there are numerous times when students do not follow the specified format e.g. going over the allotted 200 words or submitting a profile picture that is larger than required. NCI staff must then fix the work submitted by students costing them time.

This project aims to provide a portal that will guide students into uploading correct details in the correct format, ensuring an efficient process for all concerned

6.1.3 Technical Approach

Defining an approach to tackle this project, I felt it was important to divide the work into meaningful categories to ensure I have a clear picture of what is required throughout the year.

Research and Design:

- Initial research (proposal), finding out the scope of the project and identifying stake holders.
- Requirements Analysis – setting out clear requirements and use cases required for development.
- Testing schedule – formulating a clear strategy for testing such as integration testing and user feedback.

Implementation:

- Development stages – I plan to divide the process into three key development stages, one phase after the requirements analysis in preparation for the prototype presentation, a development phase after prototype feedback and the third major phase to be carried out from February 2017 to April 2017.
- Development will be carried using Visual Studio.

Feedback review from Midpoint presentation:

- Following the midpoint presentation, I plan on reviewing and implementing recommendations provided by lecturers and supervisor.

Review and Completion:

- Complete technical document and final submission of code.

Final Presentation

6.1.4 Special Resources Required

Professional ASP.NET MVC 5 1st Edition by Jon Galloway is one of the top books listed by Microsoft for ASP.NET MVC development.

I will also be making use of thee extensive ASP.NET documentation https://docs.asp.net/en/latest/

To host the project, I will need a Microsoft Azure account.

6.1.5 Technical Details

From research I found **ASP.NET MVC** most suitable for this type of web application. The ASP.NET framework is based on the Model-View-Controller architecture model and applications are developed using **C#**. The benefit of using this framework is that it is relatively lightweight and highly testable.

For the frontend development I will HTML5, CSS, JavaScript and AJAX functionality for presentation. Bootstrap will be used for the front-end design.

For version control I will be using Git which will allow me to safely backup my project files and allow my supervisor to track my progress.

Database Management System: Microsoft SQL Server

6.1.6 Evaluation

I will evaluate the system by two methods, unit testing and user testing.

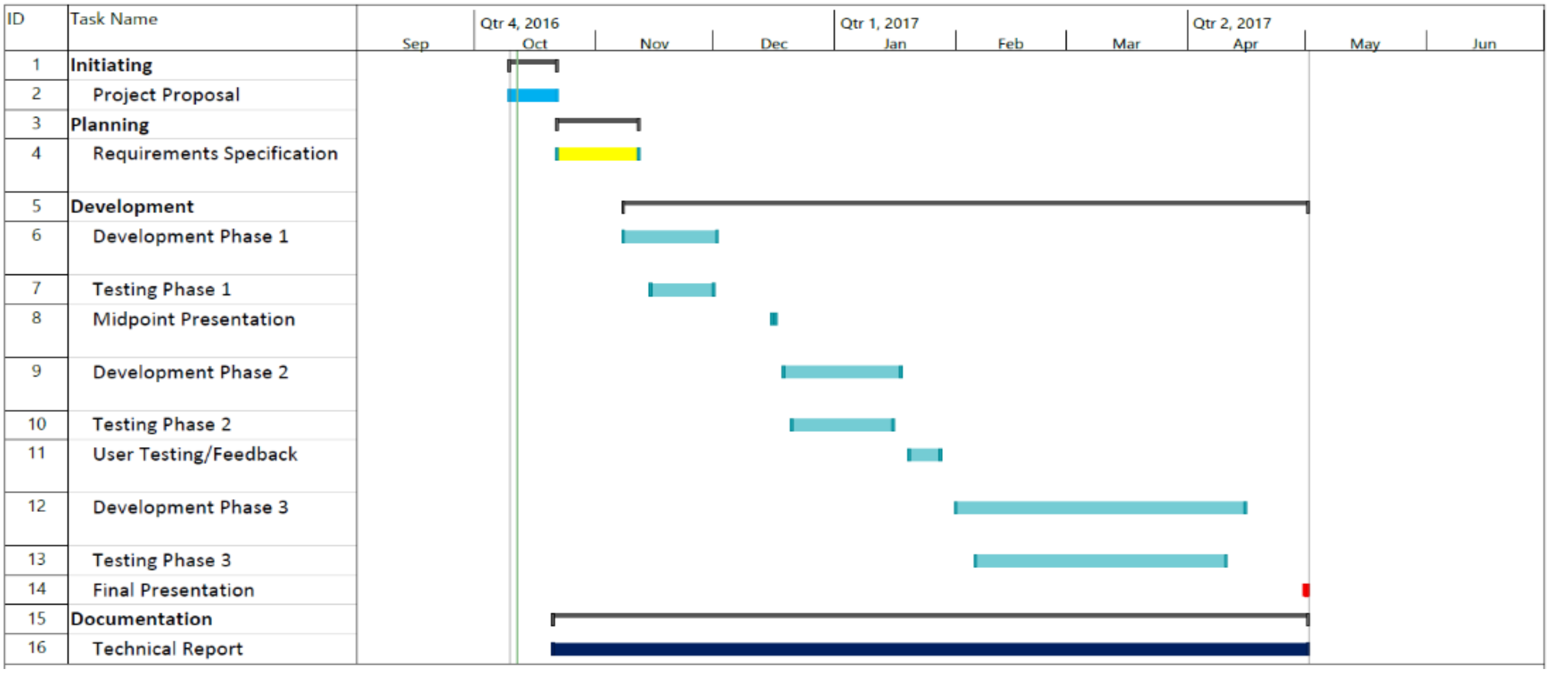
For the duration of this project I will implement a test-driven approach with unit testing. This approach will allow me to continually improve the application with each iteration of the system.

User testing will require students and faculty to use the system and provide feedback through questionnaires. I will also use previous student's project details to test how the quality control features work.

6.1.7 Proposed Supervisor

Eamon Nolan

6.2 Project Plan



ID	Task Name
1	**Initiating**
2	Project Proposal
3	**Planning**
4	Requirements Specification
5	**Development**
6	Development Phase 1
7	Testing Phase 1
8	Midpoint Presentation
9	Development Phase 2
10	Testing Phase 2
11	User Testing/Feedback
12	Development Phase 3
13	Testing Phase 3
14	Final Presentation
15	**Documentation**
16	Technical Report

6.3 Monthly Journals

6.3.1 September

My Achievements

This month, I identified a number of different project ideas and narrowed the choice down to just one. I then had to work on the Project Pitch to be presented to a panel of three lecturers who could approve or reject the proposal if it was not good or complex enough.

My Reflection

September was the first month of the 4th year Software Project. It's the big, individual project where students can showcase their talents and all they have learned in the years at NCI.

Prior to returning to NCI for my final year, I was thinking and researching different projects I could undertake. I wanted to choose something that could play to my strengths and also something that would be interesting to develop. I had initially chosen the Game Development stream for 4th year and was looking into games I could design for the project but changed my mind to Mobile App Development as I realised I would have probably lost interest in gaming within the first semester. With this in mind, I decided my area of interest for the project would be between Web Development or Mobile App Development.

Outside of the Tech world my favourite hobby is music and going to concerts. I thought this may be an opportunity to combine my interests and develop something worthwhile and useful. I thought about the different aspects of the music world and going to concerts so my idea was to create a web site where users could buy, sell or swap tickets for concerts only at face value. I thought this could be a useful project as it can often be hard to buy tickets for sold out event without paying exorbitant prices from scalpers or ticket re-sale sites. This site would enable genuine music fans to connect and have the opportunity to see their favourite music acts. The site would also have forum features where fans can discuss up-coming events.

Intended Changes

Next month, after the project has been approved I will create the official project proposal and establish exactly what I will need to complete this project in an efficient and timely way.

6.3.2 October

My Achievements

This month I submitted my Project Proposal on 21/10/16. The proposal includes my objectives, background, project plan and my technical approach.

This month I was also assigned a Project Supervisor and started work on the Requirements Specification documentation that is due 11/11/16.

My Reflection

After having my initial idea rejected I chose a new project from the Pre-Approved Projects list posted on the Project Moodle page. Although at first I was disappointed in not having my music app project approved I realised it was for the best as I would not have gotten the required grade needed to pass the final year as it was not complex enough.

I found an interesting project about a Project Showcase application proposed by Eamon Nolan and after a short meeting he allowed me to take it on, with him as my project supervisor. I feel confident I can make this project a success and learn new technologies and develop the skills I have been working on over the last three years.

I made the most of the Reading Week by completing most of the sections for the Requirements Specification after my Supervisor emailed with a list of User Requirements. I felt that some parts of the Requirements Specification were difficult such as specifying the Business Rules and detailing the GUI. I researched these topics to ensure I completed the document to the best of my abilities as a good Requirements Spec is a crucial part of any project and could make all the difference when developing the application.

Intended Changes

Next month, before November 11th, I will complete the Requirements Document. I will also begin work on the Project Prototype for the Mid-point presentation.

Supervisor Meetings

Date of Meeting: 17/10/16

Items discussed: Project idea from approved projects list

Action Items: What the project is about & Initial Requirements.

6.3.3 November

My Achievements

This month I submitted my Requirements Specification on 17/11/16. The Requirements Spec specifies functional requirements, non-functional requirements, user requirements and data requirements for this Web project – NCIProjects.

I also began developing a prototype for the Mid-Point Presentation that will take place on the 19th and 20th of December. I am developing the project using the ASP.NET framework and for the Mid-Point presentation I hope to have the webcam image and image file upload function working to demonstrate how to application will be used by students.

My Reflection

Trying to balance the workload of the Final Year is tough, with assignments and projects to complete every week it was hard to prioritise one thing each week so I may not have given my full attention to this software project this month but with the Mid-Point presentation approaching I feel I can push on now and develop a good working prototype to demonstrate the application’s intended functionality.

With the upcoming Mid-Point presentation, I have so far completed the image upload and PDF report generation functions for the prototype and will continue to refine them after the Mid-Point presentation.

Intended Changes

Next month, I intend to have the first part of the Technical report finished and uploaded to Moodle and will attend the Mid-Point presentation to gain the first marks of the project. After the Mid-Point presentation, I will take the feedback given from the lecturers and apply it to my project.

Supervisor Meetings

Date of Meeting: 28/11/16

Items discussed: Technical Report review and Prototype progress.

6.3.4 December

My Achievements

This month, I was able to develop functionality for the prototype of my application for demonstration in the Mid-Point presentation.

My contributions to the projects include areas for the students to register their details to the system and an area for the students to input project details for use in the project showcase booklet. I was also able to implement a file upload feature allowing students to upload a profile picture that will also be used in the booklet. Finally, this month I was also able to implement and finalise the design of the report generation feature which allows the end user (NCI staff) to export data from the database into a functional PDF document for use in the project showcase.

My Reflection

I found it easier to manage project work this month as many lecturers were able to push back deadlines for other classes meaning I could prioritise this project for a couple of weeks.

My Mid-Point presentation was scheduled for the 20th of December and I feel that it went well. The two lecturers assigned to mark my presentation gave me good feedback and I feel confident going forward that I can complete the project on time and deliver a quality product.

Intended Changes

Next month, after the exams are finished (5th of Jan – 12th of Jan) I will have 10 days until Semester 2 starts so I will use this time efficiently and begin to develop the next stage of the project which includes developing functionality for Webcam picture upload and Admin approval or rejection of a student's submission. Feedback from the Mid-Point presentation will also be taken into consideration and implemented into the project.

I will also begin to implement testing in my application using Unit Tests.

Documentation will also be revised and updated to reflect changes in the project.

6.3.5 January

My Achievements

This month, I focused my efforts on expanding the prototype developed during Semester 1, securing the application through Authorizations and developing the Database File upload to allow the Administrator to add new Students to the database through CSV Excel files.

Following a meeting with my Project Supervisor I have also redesigned the Student and Submission pages to accommodate the uploading of profile pictures to the Submission itself instead of the Student's profile page.

I also got to grips with the Microsoft's Azure platform and have deployed the project to a test server. When the final database design has been finalised and tested, I will deploy to a production server.

My Reflection

This month I found it tough getting back into a development mind frame after the Christmas break and numerous exams.

I found the documentation available for the .NET platform to be very helpful when starting a new feature. When designing the Custom Authorization there was extensive documentation on the different Authorization tags and best practices.

I have also found the Azure platform to be restrictive on the free tier – there is a limit of 32mb on the free SQL database and the PDF generation will also not work on the free feature – this is something I will have to discuss with my project supervisor.

Intended Changes

By the end of this February I intend to have completed and uploaded the Analysis & Design document. This is not a mandatory upload and will not carry any grades on its own but will be useful in compiling my final Technical Document due for completion in May.

I also plan to have completed the Webcam feature to allow users to upload their profile picture directly from the laptop or computer webcam in addition to the file system upload currently in place.

The next phase of development will also include the Admin approval or rejection feature which will allow the Admin to approve or reject a Student's submission.

Supervisor Meetings

Date of Meeting: 06/02/17

Items discussed:

- Reviewed current application the College uses to create the Project Showcase Booklet
- Database design changes
- User Interface design

6.3.6 February

My Achievements

This month, I focused my efforts on the testing aspect of the Development cycle – Unit testing and Usability testing. I also focused on completing the Analysis & Design document.

I attended the seminars on Unit and Usability testing. These seminars helped with starting the Unit tests as I have had little experience in this area.

Completing the Analysis & Design document was another important step towards completing the Technical document as I can focus my efforts on development from now until May.

My Reflection

Trying to find the time to learn about the different aspects of testing was hard, between working on the Analysis & Design document and completing other module work. Having consulted different MVC Unit testing methods I feel confident in completing the testing part of the project. Having never done a lot of testing in other projects it was interesting to learn how to test code using unit tests and think this will benefit me during the project development and in the future.

Intended Changes

In March, I plan to conduct the Usability Testing with some students from the class and also some of the NCI staff as the Admin user. The feedback received from these tests will be analysed and put to good use in the project.

Supervisor Meetings

Date of Meeting: 06/02/17

Items discussed:

- Reviewed current application the College uses to create the Project Showcase Booklet
- Database design changes
- User Interface design

6.3.7 March

My Achievements

For March, I focused on getting the usability testing done and getting the core functionality tested and developed.

The core functionality included the report generation. I had developed an ASP.NET Web Forms for the report generation but Web Forms seem to be outdated and I could get the authorization aspect of the project to work with Web Forms. I decided to convert the Web Forms page to a method in the Submissions Controller.

I also got my project showcase description signed off by my supervisor which will be used in the booklet at the showcase.

My Reflection

With this semester being a much shorter semester than usual it has been hard to find the time to divide between projects as there are new uploads and assignments expected each week. I focused my efforts during reading week on a research assignment for the Distributed Systems Module so there wasn’t much development that – more so testing.

I found that Usability testing was very helpful in correcting errors – even from the first session. I took feedback from different students and refine and fix the bugs to be ready for the next session – and so on. Through this testing I was able to alter aspects of the site such as how complex passwords should be and where to place different alert boxes to ensure the user was not confused.

Intended Changes

Get the Usability and Unit testing documents completed.

Get the project showcase poster finalised and printed off for April.

Supervisor Meetings

Date of Meeting: 06/03/17

Items discussed:

- Usability Testing Session