**Task 1 - Email**

To: Manager

From: Thomas

CC:

BCC:

Date: 19/07/2023

Subject: Program Specifications

-------------------------------------------------------------

Hi Manager,

For the upcoming deployment of Flawless Feedback application I would like to review the Scenario, Specifications, Guidelines and Business Case Scenarios, is it possible for you to email those to me?

Regards,

<Thomas>

**Task 2 – Additional Requirements**

* Deploy our Web Application to the Cloud Server (Azure)
* Deploy our API to the Cloud Server
* Test our Cloud based application
* Create New User that with read/write functions to access the Azure SQL Database.

**Task 3 – Report**

**Development Tools**

**GIT:** Is an open development tool designed to handle both small and large projects. It has a unique branching model that allows users to create multiple local branches independently. Branches are used and are easily merged, created or deleted allowing for smooth context switching and feature-based workflows.

[**ASP.Net**](http://asp.net/) **Core:** is a high performing cross platform framework used for developing modern web based applications. With the use of [ASP.Net](http://asp.net/) developers are able to build web apps and services, Internet of things applications, and mobile backends. Some features of [ASP.Net](http://asp.net/) include:

**Visual Studio:** Visual studio is our selected IDE and what we will bel using to Publish our API to Azure

**Azure:** Our selected cloud based web server

**SQL Server DBMS:** Is the selected database manage system, as it is the business standard and Tim is running his application on windows.

**Big Data**

The term “Big Data” refers to the large sets of data in cloud computing. The large sets of data can be outputted by a variety of computers and can refer to any larger sets of while, which is usually data that is far too big to store or query on a single computer.

Big data is not being used in our application, as we don’t expect the data sets to get large enough for it to be required.

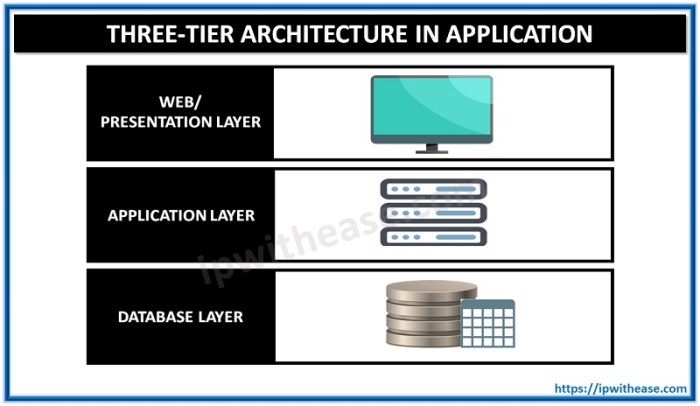
**Architecture and Framework**

Three-tier client and server architecture is a software design that implements three tiers of architecture, these three tiers are known as:

**Presentation tier:** This is top most level of the application and is the user interface, it involves having a digestible UI including tasks that can be easily translated by the user.

**Logic Tier:** The next layer that coordinates the actions and processes of an application as well as all logic decisions and calculations. This layer is also responsible for transferring from the data tier to the presentation tier.

**Data Tier:** The data tier is where the information is stored and retrieved. Using the logic layer we can transfer data from the data tier to the presentation tier for the user to view.

*(Image Source: https://ipwithease.com/three-tier-architecture-in-application/)*

A Model View Controller (MVC) is a software design that implements user interface, data and logic to control it. The User interface is the view and used to display the model data of an object, the data is the model; which represents the shape of the data (like a class in C#) and finally the logic is the controller which is used to handle the user’s requests.

* Model = Data
* View = User Interface
* Controller = Request Handler

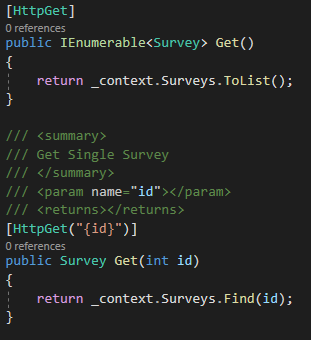
MVVM the view is the entry point to the application. Whilst MVVM does a better job than MVC in breaking down your application to a modular single purpose component MVC can do this as well. A MVVM is used to help organize code and break up programs into modules to speed up development and updating.

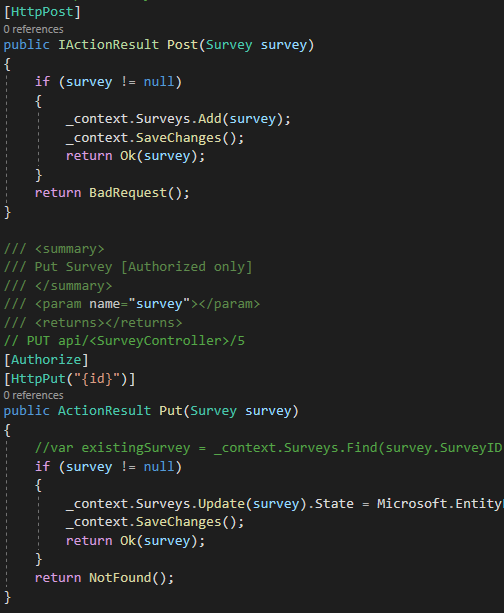
* Model
* View
* View-Model

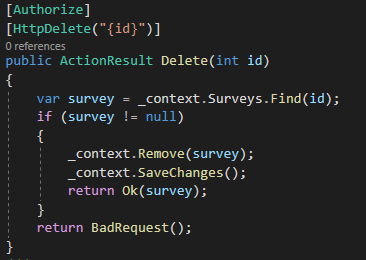
Entity Framework is for .NET development and is used to enable developers the ability to work with data using objects without the need to focus specifically on the underlying database tables. It allows for higher level of abstraction and the creation and maintaining of data oriented applications.

**Web Service**

Flawless feed is a back end web service created using ASP.net Core that will connected to Web application giving the client the ability to access, edit, delete, and create Surveys, Questions and Answers. End users will be required to pass Authorization in or order to delete and edit surveys, but any user can view and create their own. The surveys will be hosted on an sql azure database, so that clients can access the cloud 24/7.







**Cloud Application**

Web Application designed and developed to provide a User Interface for the functionality created in the Web API

* *Create multiple* feedback **questions** per **Survey**
* *Create* multiple possible **options** for a **question**
* *View* a list of all **Surveys**
* *View* a list of **questions** attached to a **Surveys**
* *View* a list of possible **responses** attached to a **question**
* *View* a report of number of **questions** per **Survey**

**Testing**

Testing the back-end can be done through an application known as postman, with postman you can create mock-environments and test the endpoints of a web API and ensure correct functionality.

Testing the front end can be performed by creating a test environment with dummy seed data in our start up class. With this test environment we can test the functionality of our front end without connecting to our API

**Deployment**

For deployment of our web service and web application visual studio has access to Wizard that will allow us to create a resource group on azure, this can be done by going to “Build -> Publish Application” inside our solution. Once our resource group has been created, we must nextly create an SQL Server and on the azure portal; and then finally migrate our locally store SQL database to the Azure Server. A more detailed guide has been attached in the assessments folder under [“Azure\_Deployment”].

**Task 4 – Email for a Meeting**

To: Manager

From: Thomas

CC:

BCC:

Date: 19/07/2023

Subject: Project Meeting

-------------------------------------------------------------

Hi Manager,

For the upcoming deployment of Flawless Feedback application I would like to organise a meeting to confirm the technical details and present our report. Can you please let us know what your availability is?

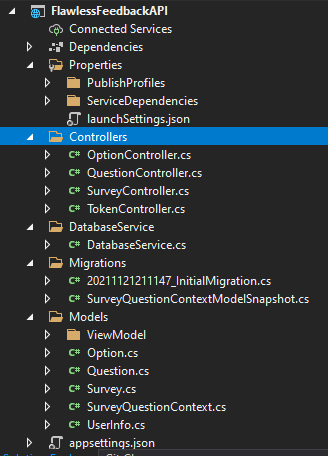
Regards,

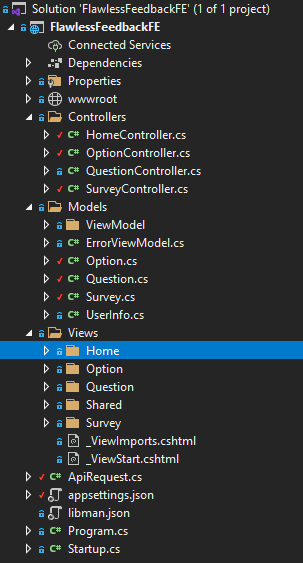
<Thomas>

**Sign-Off**

|  |  |  |  |
| --- | --- | --- | --- |
| **Project Name:** | FlawlessFeedback | | |
| **Project Manager:** | Shaun O’Sullivan | | |
| **Start Date:** | 20/11/2021 | | |
| **Completion Date:** | 22/11/2021 | | |
| **Project Deliverables:** | | | |
| Email to the project manager | | | |
| Project Report | | | |
| Software Installation Plan | | | |
| Azure Deployment | | | |
| Test Plan | | | |
| FlawlessFeedbackFE | | | |
| FlawlessFeedbackAPI | | | |
|  | | | |
|  | | | |
|  | | | |
|  | | | |
|  | | | |
| ***I acknowledge that I have submitted the Technical Report as per your instructions.*** | | | |
| Thomas Abreu | | T.A. | 22/11/2021 |
| **Programmer’s Name** | | **Signature** | **Date** |
| Shaun O’Sullivan | | S.O. | 22/11/2021 |
| **Project Manager** | | **Signature** | **Date** |

**Task 5 & 6 – Development**





**Task 7 – Perform Test**

|  |  |  |
| --- | --- | --- |
| **Test** | **Expected Results** | **Actual Results** |
| Test Create Buttons | New Question/Survey/Option created | All creates work, however entries can be created blank also. This is unintended. |
| Test Edit Buttons | Existing Question/Survey/Option Updates | Edit buttons require a login.  Edits update entry data  Edits can also update fields to blank which is unintended |
| Test Delete Buttons | Existing Question/Survey/Options removed from the database | All Deletes work as intended |
| Test Authentication logged in and logged out | Delete and edit works whilst logged in, otherwise redirect to login page. | Deletes work, Edit Works.  Both require login  Blank entries can be inputted in the edits. |
| Test Details Buttons | Displays details of Questions/Surveys/Options | Details all work as intended |
| Log in | Accepted credentials login successfully, failed will display a failed login message | Log in page works as in intended |
| Login: Username case sensitivity | Username isn’t case sensitive | Username is case sensitive; its more user friendly to remove this constraint. |
| Login: Password case sensitivity | Passwords is case sensitive | Password is case sensitive. The small constraint adds another layer of security in password creation |
| Survey Create: Drag and Drop | Image drags and drops, Logo textbox populates | Working as intended |
| Survey Create logged in: Populates the survey creator field | Field auto populates when logged in | The creator name populates with the logged in account. |
| Log into Guest on two different browser | Second guest remains logged in, first guest doesnt remain log in. | Second guest remains logged in, first guest doesnt remain log in |
| Log into guest then log into admin | Guest name is forgotten, Admin login overwrites | Guest name is forgotten, Admin login overwrites |
| Create two surveys with 2 browsers | ID's conflict causing an error | Successfully allocates an ID to the two newly created surveys. |
| Export CSV button | Writes chart to CSV | Writes chart to CSV |
| Upload an Image on Create Survey | Image saves to directory, available for viewing in the details | As expected results. |

The testing has revealed that logging into multiple users is safe, and will only remember the final user logged in. It has also shown that we can successfully Upload Images and Export files whilst upkeeping working functionality from previous tests, along with these front and back end tests, we have tested the cloud portion of the application, mainly to check stability and performance, we were successfully able to launch our program on multiple web browsers along with finding that if two people create an identical survey, their surveys will not clash with all of this in mind we are happy to conclude our testing and move on to the final stage of operations.

**Task 8 – Email for a Meeting**

To: Manager

From: Thomas

CC:

BCC:

Date: 19/07/2023

Subject: Project Meeting

-------------------------------------------------------------

Hi Manager,

For the upcoming deployment of Flawless Feedback application I would like to organise a meeting to present our deployment plan for our web service and cloud application. Can you please let us know what your availability is?

Regards,

<Thomas>

|  |  |  |
| --- | --- | --- |
| **Date and Time of Meeting:** | 25/11/2021 | 10:00AM |
| **Attendee names:** | Project manager | Shaun O’Sullivan |
|  | Programmer | Thomas Abreu |
|  | Tester 1 | Tom Chaser |
|  | Tester 2 | Jerry Runner |

**Manager Feedback:**

|  |
| --- |
| Excellent progress on the Flawless Feedback project, with a detailed Deployment Plan. I have approved our development team to take the next steps in the Flawless Feedback Project |

**Task 9 – Installation**

***[Please refer to “Azure Deployment” in the Assessment Folder]***

**Task 10 – Installation Sign-Off**

|  |  |  |  |
| --- | --- | --- | --- |
| **Project Name:** | Flawless Feedback | | |
| **Project Manager:** | Shaun O’Sullivan | | |
| **Start Date:** | 20/11/2021 | | |
| **Completion Date:** | 22/11/2021 | | |
| **Project Deliverables:** | | | |
| Development | | | |
| Testing | | | |
| Emails to Manager | | | |
| Azure Deployment | | | |
| Meeting Email | | | |
|  | | | |
|  | | | |
|  | | | |
|  | | | |
|  | | | |
|  | | | |
|  | | | |
| ***I acknowledge that I have ran the install package in the Production Environment and submitted the installation guide as per your instructions.*** | | | |
| Thomas Abreu | | T.A. | 22/11/2021 |
| **Programmer’s Name** | | **Signature** | **Date** |
| Shaun O’Sullivan | | S.O. | 22/11/2021 |
| **Project Manager** | | **Signature** | **Date** |