BEB801 PROJECT PROPOSAL & PROGRESS REPORT

FreeFoodNearMe

Abstract

A web service to assist people from low SES backgrounds find food outlets

Jason (Chia-Sheng) Chang n9182004

Supervisor: Dr. Dhaval Vyas

Executive Summary

The following project proposal outlines the gap in current HCI systems within Brisbane addressing both emergency relief centres and social connectivity. This issue is reinforced in literature of previous case studies. Through this it was found that there is a common problem with access to information and importance of social networks within the low SES community. The solution proposed is targeted at assisting in relief of food and providing a network for users to socialise and share stories. This will provide access to information allowing for week-to-week survival and providing reassurance. In addition, the solution aims to relieve stress and help improve quality of life for participants by easing their budget for food.

The proposed solution to address this gap would be a web service called “Free Food Near Me” (FFNM). It was decided that a webservice would be the most appropriate HCI system for the proposed solution due to its ease of access across multiple devices and how integrated web services are in our current society. A design plan was conjured for the development of FFME, through which it was decided the web service would be built with the following:

- ASP.net MVC web application (Backend Framework)
- Custom Html, CSS, JS (Frontend Framework)
- Microsoft Azure CosmosDB (Content Management System)
- Google Maps API

The projected timeline for the solution is split into four main phases – Environment Setup, Solution Code Sprint, Solution Polish and Solution Wrap Up. The development process would also follow an AGILE & SCRUM approach with several iterations of code sprints, feedback from clients and code reiteration.

In addition, a progress update has also been included in this report. To date all requirement set in the initial project proposal has been completed and a functional web service has been developed. Improvements and changes have also been made from the initial project proposal. These improvements are mainly changed to improve user experience of both administrators and regular users. Some of these improvements include:

- Use of Bootstrap data tables for ease of search and navigation of data
- Redesign of look and feel of webservice
- Additional Google Maps API integration

The project timeline followed when implementing the solution has also been updated. At the start of the code sprint in phase 2 a week break was taken to prioritise other subjects and tasks. Through this week delay tasks had to be reordered in order to be completed on time. Rather than creating a fundamental template for each webpage to be built further upon, each webpage was fully completed before moving on to the next. This decision resulted in a successful project delivery on time and within scope.

Table of Contents

List of Figures

1.0 Background and Literature Review

1.1 Introductory Statement

The project focuses on the benefits of technology on people from low socioeconomic status (SES) backgrounds within Brisbane. More specifically the literature review will entail details regarding the background of the target audience, overview of human computer interaction (HCI), effects HCI had on low SES communities and how HCI has a positive effect on food crisis.

1.2 Literature Review

1.2.1 HCI & Low Socioeconomic Status Communities

Currently technology has a huge influence on making the everyday person's life easier. Whether it is at school, in a workplace or at home, there is always some form of technology to make things simpler. However, in the entirety of HCI studies one of the lesser explored topics is how great the impact of technology on low SES communities.

According to several studies and research it was found that the homeless community greatly benefit from the use of technology. One of these benefits is the use of mobiles phones to stay connected to friends and family and establishing social networks. This is addressed in a case study where several participants found the sense of disconnection from an extended social circle was a considerable source of stress (Dantec & Edwards, 2008).

Technology has also greatly assisted the low SES community with access to information and employment. It was found in a study targeted at homeless young people that technology has allowed them to have access to resources assisting with writing resumes and cover letters increasing employability in addition to finding more job opportunities online (Woelfer & Hendry, 2010).

For the low SES community technology has also played a major role in managing low income. An article sheds light on this matter with a member of the community stating they previously recorded their bills, financial statements and financial management annotations on paper (Vines, Dunphy & Monk, 2014). This often led to human error, loss of information and difficult record management. With the use of technology, it greatly assisted with record keeping allowing all incoming and outgoing expenses to be easily viewed. This promotes confidence through awareness and helped developed financial plans and balance budgets (Vines, Dunphy & Monk, 2014).

1.2.2 HCI and Food

There have also been multiple breakthroughs within HCI studies regarding food. One of which is combating food insecurity in the future. A paper written by Jaz Hee-jeong Choi and Eli Blevis explore the role of HCI design in encouraging individuals to participate in creating sustainable food cultures in urban environments. It is stated that urban environments are particularly problematic in their segregation from areas where natural food is grown leading to a lack of understanding in access to fresh produce (Choi & Blevis, 2010). It is predicted that the rapid population growth in urban areas will amount to 60% of the entire population by 2030, which suggests that availability of food resources in these areas will increasingly become more scarce (Choi & Blevis, 2010). Technology has been used to help levitate this

problem through promoting awareness of this issue. Some of these include tracking of food production and source, interactive systems to help ensure planning and preparation and interactive systems to assist people in urban environments how to survive with fewer resources (Choi & Blevis, 2010).

1.2.3 Food Crisis and Technology

Combining HCI studies in both fields of low SES communities and food there have also been HCI systems addressing local food needs within these communities. The department of informatics within the university of California released a paper based on a 18-month qualitative study for a prototype location-based information system (LBIS) to address food insecurity in one U.S country (Dombrowski et al, 2013). The study addresses current problem to effectively match clients with relevant information, services and food resources depending on their type of needs. This issue was the bottleneck in the process and greatly reduced efficiency in assisting clients. LBIS was used to combat this problem and support alleviation of hunger by streamlining and automating clients categorisation-based needs to appropriate resources (Dombrowski et al, 2013).

Technology has had a proven track record globally to positively impact low SES communities and address scenarios of food crisis. Through this HCI could also be used locally within Australia to help our communities locate emergency food outlets or food pantries. In recent years it is recorded that nearly 4-6% of Australians live in chronic poverty (CEDA, 2015) and an increase of 8% of people seeking food assistance (FHR, 2014). The current issue expands past homeless people as elderly, single parents and refugees are also in need of food assistance (FHR, 2014). A recent study conducted by Dr. Dhaval Vyas and Stephen Snow delves into this issue within the Brisbane community through visits to the local relief centre “Communify”. Within the article there have been several similarities to previous explored literature such as the importance of survival, reassurance, access to information and strong influence by social networks. The article is summarised by stating the importance these food reliefs have on low SES communities and possible HCI solutions. Not only food reliefs allow for week to week survival, it also provides reassurance and opportunities to create social connections (Vyas & Snow & Mallett, 2015). This is touched upon in the HCI section stating “sharing” is viewed as an empowering mechanism and can be ultilsied to support disadvantaged communities balance reciprocity, collective efficacy and employment generation through vertical capital (Vyas & Snow & Mallett, 2015).

Reflecting on the researched literature there are common problem factors regarding access to information and social connectivity. This gap is also evident within Brisbane SES communities and food services. It is found that services such as Communify addresses the importance of staying connected and providing reassurance. Although there are several services similar to Communify, there currently isn’t an official HCI system to incorporate both information providing and social connectivity capabilities.

1.3 Research Problem

The research problem for this project addressed a current void in Brisbane food outlet services. Currently there are numerous food outlets in scattered throughout Brisbane, however not a way for people in need to easily locate these services. This is a significant problem for the financially struggling community as food is a vital necessity for survival. By giving people the ability to locate free food it not only allows them to survive for another day, but also offer peace of mind. In addition, solving this problem also directly correlates to improving quality of life. By not needing to purchase food it relieves budgets allowing for financial allocation into other necessities such as shelter and clothing.

2.0 Program and Design of The Proposed Research Investigation

2.1 Objectives, Methodology and Research Plan

The project proposes a web service "*FreeFoodNearMe*" (FFNM) as the optimal solution for this problem by allowing all food outlet services around the area to be located in one convenient location. A web service was chosen for the solution type as in the current day and age websites are easily accessible through multiple mediums. Referring to the target audience of the low SES background community the solution can be easily accessed via their phones through free WIFI locations, library computers or any other device capable of browsing the web.

Due to complications with double degree courses the scope of this project proposal will be factored to a single semester duration. It was agreed upon with the supervisor that the project will be handed over to future students partaking in BEB801 and BEB802. The following design specifications and build choices consider the potential longevity and scale of the web service in addition to ease of project hand over.

Considering these factors, the proposed web service solution will be built with the following frameworks and features:

- ASP.net MVC web application (Backend Framework)
- Custom Html, CSS, JS (Frontend Framework)
- Microsoft Azure CosmosDB (Content Management System)
- Google Maps API

Each of these features and frameworks will be covered in detail within this project proposal in their individual sections.

2.1.1 FreeFoodNearMe Web Service Structure

The web page flow of the proposed solution is highlighted in figure 1 below. This shows the navigation path and individual pages of the web service.

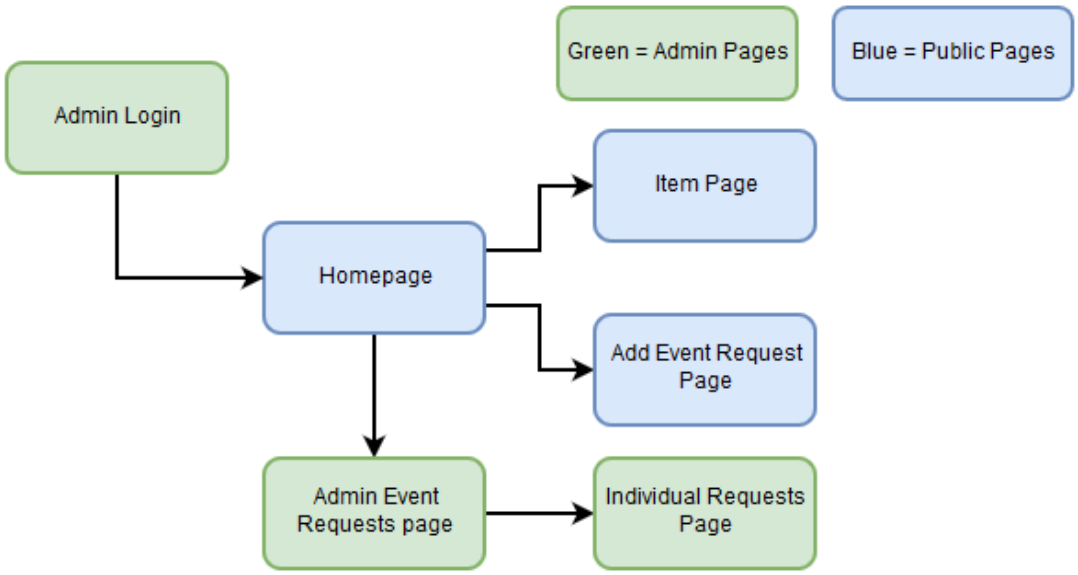


Figure 1: Flow Diagram of Web Service

For the initial proposal the web service will consist of five separate webpages allowing for basic functionality allowing users to search, view and post stories of their experiences with food outlets. FFNM will also have the functionality for users to contribute to the overall database by requesting events to be added. These events can be categorized as permanent food outlets which operate on a schedule or upcoming one-off events that the community may be interested in. Requests posted by the community will be reviewed by an administrator of FFNM and if accepted will be added to the database.

In more detail a mockup of the homepage can be found in Appendix A – Homepage Wireframe. This page will allow users to search and filter food outlets and events in the database. These results are then populated on both a tabulated format and on google maps allowing for simple navigation. Simple details for each food outlet or event will also be shown when hovering over a google maps pin or within the tabulated results.

A mockup of the item page can be found in Appendix B – Item Page Wireframe. This page provides additional information about the food outlet/event and the services they provide. The page also allows the community to leave stories of their experiences with the food outlet/event. These stories are displayed on the bottom of the page and act as a way for the community to promote and motivate other people in need to ultilise these services.

A template form for the Add Event Request Page is located in Appendix C – Add Event Request Page. This page allows users to register their services or upcoming events to be added to the FFNM database. In order for service to be validly registered all inputs are mandatory and the service must also be approved by a FFNM administrator.

The administrator login mockup wireframe can be found in Appendix D – Admin Login. This page allows the administrator to login to access administrator exclusive features and pages. For the initial proposal

this will allow the administrator to access the list of service register requests. The administrator is able to then review these requests individually and approve or decline them.

Upon signing in with an administrator account the navigation bar will change from “Request an Event” to “Service Registration Requests”. Accessing this page will show something similar to the wireframe found in Appendix E – Admin Event Request Page. This page tabulates all service registrations showing basic details for each request. To view additional details and review each service registration the administrator is able to click into each item.

This will show a similar user interface to that of Appendix F – Individual Request Page. The page will show inputted details for the service registration request. In this page the administrator is able to review the submission and choose to either accept or decline the request.

2.1.2 ASP.Net MVC – Backend Framework

Model View Controller (MVC) is a product development architecture pattern. This pattern is commonly used for architecting web applications and user interfaces. Unlike the traditional approach of programming, the MVC pattern separates the different aspects of the application (input logic, business logic and UI logic), while providing a loose coupling between these elements (Socratic Solution, 2017).

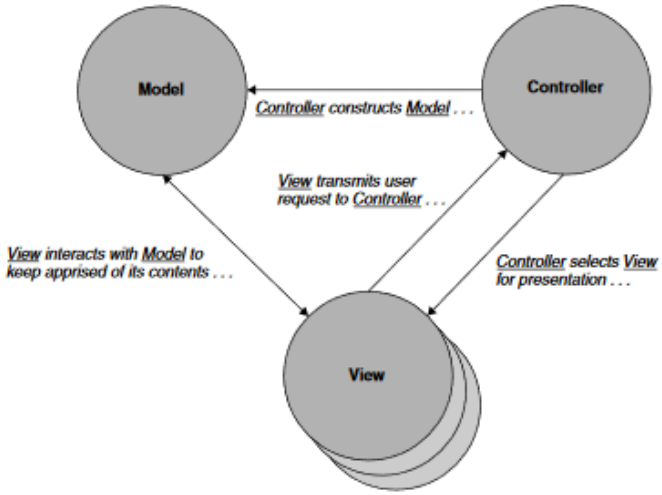


Figure 2: Model-View Controller design pattern

Referring to Figure 2 the user interface logic belongs in the view, input logic belongs in the controller and business logic belongs in the model (Socratic Solution, 2017).

More specifically the model defines what data the application should contain. If the state of this data changes the model will notify the view either directly or through controller logic allowing for dynamic change in the display (MDN, 2018). These model objects are often stored and retrieved in a database, within the project scope the model will be stored as a JSON format in CosmosDB.

The view defines how the application's data passed from the controller should be displayed. Typically, these are defined from the model data and represents the components that are displayed on the application's user interface (Socratic Solution, 2017).

The controller typically acts as the middle man between the model, view and the user. It handles user interaction, work with the model, and selects a view to render the user interface (Socratic Solution, 2017).

Considering the potential longevity and scale of the project the MVC model was chosen as the web application architecture for its completeness in structure. The separation of components promotes the maintainability, testability as well as scalability of the application. This is done through separation of responsibilities and logical grouping allowing simultaneous development for multiple developers (Socratic Solution, 2017).

In addition, ASP.NET was chosen as the backend framework through its extensive resources online and support for existing ASP.Net features (Microsoft, 2018). This design choice promotes ease of handover as it allows future developers to quickly integrate themselves into the project and find resources or libraries to assist in development.

2.1.3 Frontend Framework

Custom HTML, CSS and JavaScript will be used as the primary framework for creating views within the application. This design choice was made over existing frontend libraries such as Angular 6 or ReactJS for simplicity and ease of project handover. Currently within QUT's coursework these libraries are not taught, thus it would be difficult to integrate a new developer into the existing project. In addition, these libraries would not be needed to achieve the current project scope.

CSS libraries such as Bootstrap however, will be used in the solution for style consistency and time efficiency. Unlike Angular 6 or ReactJS this library does not have a substantial learning curve and at its core simply pure CSS. This allows the project to be easily handed over to a new developer with minimal complexity in knowledge transfer.

2.1.4 Azure CosmosDB – Content Management System

Azure CosmosDB is a globally distributed, multi-model database service which is highly scalable and responsive. Cosmos includes compatibility with MongoDB API which allows for all existing MongoDB libraries, drivers, tools and applications to be used (Microsoft, 2018).

For the proposed solution Azure CosmosDB will be ultilsied as the database to store model data in JSON format. This allows the web application to store and retrieve model objects through controller logic to create dynamic views. CosmosDB will be used in the proposed web service solution through the following data flow as represented in Figure 3.

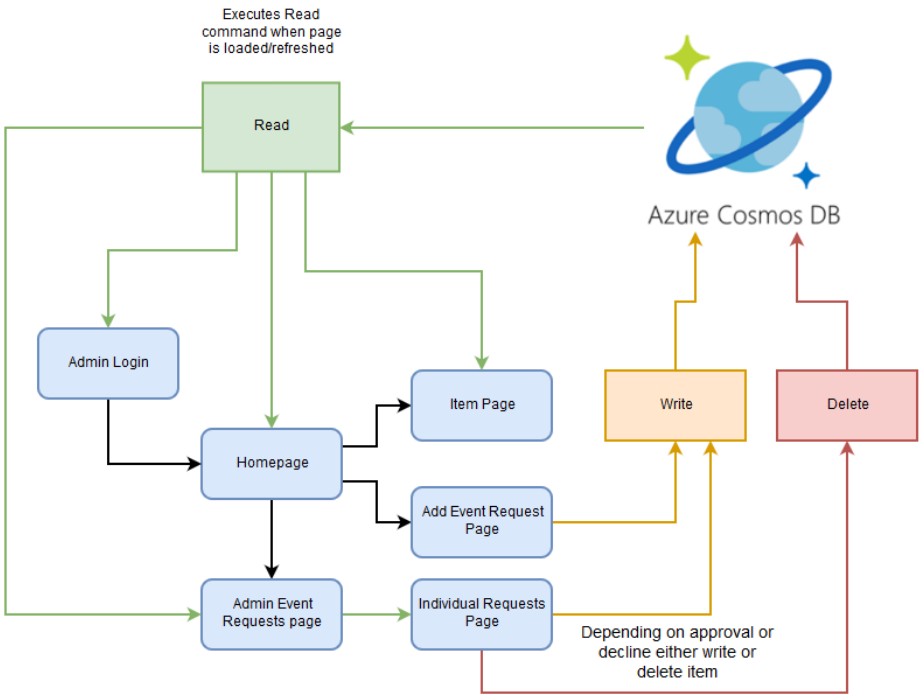


Figure 3: Web Service Cosmos Dataflow

The controller will issue read commands to retrieve model objects when loading pages which require dynamic data. For pages which sends data through user interaction such as service registration requests and accepting service registrations the write command will be issued to create a new document within CosmosDB. If a service registration is declined the controller issues a delete command to free up space in the database.

The design decision of choosing CosmosDB over traditional SQL databases was primarily due to the advantages of NoSQL databases. NoSQL databases were created in response to the limitations of traditional relational database technology. NoSQL databases are able to handle (mongoDB, 2018):

- Large volumes of structured, semi structured, and unstructured data.
- Agile sprints, quick iteration, and frequent code pushes
- Object-oriented programming that is easy to use and flexible
- Efficient, scale-out architecture

2.1.5 Google Maps API

Google Maps API is used to personalize maps with personal content and imagery for display on web pages and mobile devices (Google, 2018). This will be ultilsied in FFNM as the main service identification feature. This will allow users to find services through the use of location markers for each individual service. For the proposed solution the JavaScript Maps API will be used to convert addresses to latitude and longitude coordinates which translates to the display map.

2.2 Timeline for Completion

Considering the double degree complication, the project progression timeline was factored to a single semester project. The project plan is split into four main phases highlighted in Figure 4.

Figure 4: FreeFoodNearMe Project Progression

Phase one covers all the initial set up required to commence development of the web service. This will include configuration of CosmosDB, GitHub setup and ASP.net MVC web application solution layout setup. In addition, all libraries used such as Bootstrap, jQuery and Google maps API must be downloaded and included into the solution file. It is estimated that this phase should take minimal time to complete.

Phase two covers the bulk of development work and light testing for the webservice. This phase will follow an agile approach to software development. Agile software development is a software development methodology based on iterative development, where requirements and solutions evolve through collaboration between self-organising cross-functional teams (CPrime, 2018). During this phase frequent development cycles will take place. Each cycle will include an initial code sprint, feedback from the project supervisor and code refactor. It is estimated that this phase will take the most time to complete.

Phase three covers a polish of the web service once all components are functioning correctly. This phase will mainly involve updating web service views to improve look and feel, code cleanup and intensive testing. If extensive time is permitted the web service solution will be made to be responsive allowing for native use across all devices of varying resolutions. Dependent on time remaining this phase could take a short amount of time to complete or a moderate amount of time to complete.

Phase four covers the preparation for knowledge transfer of the web service solution, progress report writing and project presentation. The preparation for knowledge transfer includes tasks such as commenting code, CosmosDB backup, web service solution backups and GitHub cleanup. Considering all tasks within this phase it is estimated that this phase should take a moderately-high amount of time to complete.

Factoring the breakdown of the four phases in the project plan, a detailed project progression timeline was constructed and can be found in Appendix G – Gantt Chart of Project Progression. This timeline will be closely followed when developing FreeFoodNearMe.

3.0 Progress Update

Since the original project proposal, a lot of progress has been covered in implementing the solution and several alterations have been made to initial plan. This section describes the design changes from the original project proposal, a reflection of the progress and a comparison between the original project timeline and updated timeline.

3.1 Design Changes

3.1.1 Azure CosmosDB

Several changes were made from the original database architecture due to the limitations of the free trial. Originally the database was planned to be separated into several collections each categorized by a different model each. However due to the limitations of the Azure free trial only allowing for a single collection all models types were inserted into this one collection. A visualization of this can be seen below in Figure 5.

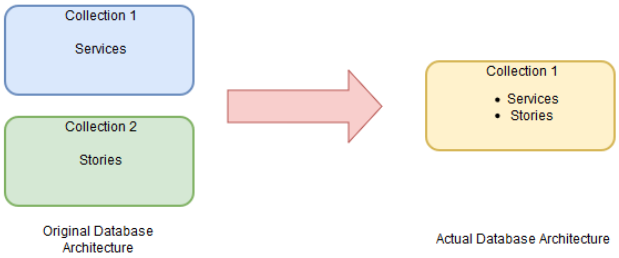


Figure 5: Database Architecture Change

3.1.2 Web Service Flow

During the implementation of the solution web service flow was also changed to improve user experience for administrators. The updated web service flow is described in Figure 6.

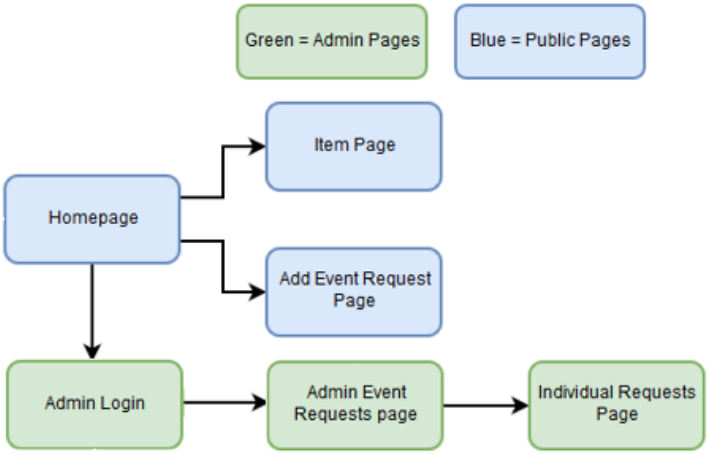


Figure 6: Updated Web Service Flow

The navigation of administration login was changed to redirect to the event request page rather than the homepage. This decision was made based on a user experience standpoint as all services on the homepage can be accessed publicly, therefore it does not make sense for the redirection page to land on the homepage after successful sign in.

3.2 Progress

From the initial project proposal all aspects of the solution have been completed in terms of both functionality and aesthetic. Improvements have been made from the initial wireframes of each webpage and additional functions were also added. The progress and changes of each individual webpage will be described in detail in the sections below.

3.2.1 Homepage

A screen capture of the updated homepage can be found in Appendix H – Updated Home Page. Comparing this page to the initial mock up wireframe in Appendix A – Homepage Wireframe several improvements and changes have been made. These include:

- General Layout Change – The layout of the page was altered to improve user experience as the original mockup had unused real estate on the webpage. In addition, the new layout is also more convenient for users with smaller screens as prevents the need to scroll down to view results on each search.
- Bootstrap Data Tables – Bootstrap data tables were used in the construction of the service results to also improve user experience. This allows the search results to compute in real time and not require a physical reload in the webpage.
- Google maps Geolocation – Google maps geolocation was also used to detect the location of the user. This allows the map to be centered at the user's current position making it easier to view services nearby.
- Search autocomplete – An autocomplete feature was also integrated into the search bar to improve user experience. This feature triggers on the user's keystroke with suggestions depending on services recorded in the database.

3.2.2 Service Details Page

A screen capture of the updated service details can be found in Appendix I – Updated Service Details Page. Comparing this page to the initial mock up wireframe in Appendix B – Item Page Wireframe several improvements and changes have been made. These include:

- Google Maps Pin – Google maps is centered around the location of the particular service for easy identification.
- Service Details – The details shown about the service has also expanded to include more information such as operating hours and service type. This prevents user's needing to search for additional information elsewhere.
- Stories – The functionality of stories has also been implemented in the solution. This focuses on the importance of staying connected for the low SES community by allowing them to interact with each other and leave reviews and experiences that they have had.

3.2.3 Add Event Request Page

A screen capture of the updated event request page can be found in Appendix J – Updated Event Request Page. Comparing this page to the initial mock up wireframe in Appendix C – Add Event Request Page several improvements and changes have been made. These include:

- Google Places Auto Complete – Improvement to user experience and ease of use by autocompleting fields based on google places dropdown list of suggestions when filling out the address.
- More Information – More information is also captured such operating hours and service type to prevent users from needing to search for additional information elsewhere.
- Preceding Fields Logic – Depending on the service type field the date picker fields will automatically hide or show. This allows for ease of use when applying for one-off services that only occur during a specific period.

3.2.4 Admin Login

A screen capture of the updated admin login page can be found in Appendix K – Updated Admin Login Page. The page follows the exact structure of the initial mock up wireframe in Appendix D – Admin Login. Minor improvements made to this page include aesthetic improvements and also responsive animations to user interactions.

3.2.5 Admin Event Request Page

A screen capture of the updated admin event request page can be found in Appendix L – Updated Admin Event Request Page. Comparing this page to the initial mock up wireframe in Appendix E - Admin Event Request Page several improvements and changes have been made. These include:

- Bootstrap Data Tables – Bootstrap data tables were used in the construction of the pending event request results to also improve user experience. This allows the search results to compute in real time and not require a physical reload in the webpage.
- Search Bar – A search bar feature has also been implemented for ease of filtering for the administrator. This allows the administrator to easy identify high priority services to review first.

3.2.6 Admin Individual Request Page

A screen capture of the updated admin individual request page can be found in Appendix M – Updated Admin Individual Request Page. Comparing this page to the initial mock up wireframe in Appendix E – Individual Request Page several improvements and changes have been made. These include:

- More Information – This reflects on the event request page as more fields were added they are also represented on this page. This provides more information for the administrator allowing them to make a more accurate decision with the review of the service.
- Ability to Edit – The feature for the administrator to be able to edit each request has also been added into the solution. This allows the administrator to correct minor mistakes with the application such as typo's and format issues. This improves user experience and saves time by preventing the user to have to fill out another event application form.

3.2.7 General Webservice

During the process of implementing the web service general improvements have also been made throughout each stage. From the initial wireframes each page has been redesigned to fit an appropriate theme and the general aesthetic has been improved using custom CSS, bootstrap and jQuery. User interaction and user experience were prioritized when making these choices. This led to improvements in overall website responsiveness and flow from the initial design.

3.3 Timeline Update

The process taken during the implementation of the web service followed the structure described in section 2.2 Timeline for Completion. However, several alterations were made in the time spent on each task due to other subject commitments and complications. The updated task breakdown of time spent is described in the Gantt chart in Appendix N – Updated Gantt Chart of Project Progression. From the updated project progression plan several changes have been made to the ordering of the tasks and timeframe spent. Firstly, a week was taken off initially due to other subjects and assignments. Due to this delay it was decided that rather than creating a foundation template for each webpage to be built upon it would be more time efficient to finish each webpage completely before moving on to the next. The order of webpages implemented follows:

1. Template
2. Homepage
3. Event request page
4. Admin login page
5. Admin request page
6. Admin individual event request page
7. Service page

The order of implementation is important perquisite pages must first be completed before creating other pages. Following the new format, the coding aspect of the project was completed on time and code refactorization from supervisor feedback were also made.

References

CEDA (2015). *Committee for Economic Development of Australia*. Retrieved from http://adminpanel.ceda.com.au/FOLDERS/Service/Files/Documents/26005~CEDAAddressingentrenchedisadvantageinAustraliaApril2015.pdf

Choi, J., & Blevis, E. (2010). *HCI & Sustainable Food Culture: A Design Framework for Engagement* [PDF]. Brisbane: Queensland University of Technology.

cPrime. (2018). What is Agile/Scrum. Retrieved from https://www.cprime.com/resources/what-is-agile-what-is-scrum/

Dantec, C., & Edwards, W. (2008). *Designs on Dignity: Perceptions of Technology Among the Homeless* [PDF]. Florence: Georgia Institute of Technology.

Dombrowski, L., Brubaker, J., Hirano, S., Mazmanian, M., & Hayes, G. (2013). *It Takes a Network to Get Dinner : Designing Location - Based Systems to Address Local Food Needs* [PDF]. Zurich: University of California.

FHR (2014). *Foodbank Hunger Report 2014*. Retrieved from http://www.foodbank.org.au/wp-content/uploads/2014/10/Foodbank-Hunger-Report-2014.pdf

Google. (2018). Maps JavaScript API. Retrieved from https://developers.google.com/maps/documentation/javascript/tutorial

Jose, B., & Abraham, S. (2017). *Exploring the Merits of NoSQL: A study based on MongoDB* [PDF]. Kottayam: Manhatma Gandhi University. Retrieved from https://ieeexplore-ieee-org.ezp01.library.qut.edu.au/stamp/stamp.jsp?tp=&arnumber=8076778

Microsoft. (2018). ASP.NET MVC Overview. Retrieved from https://msdn.microsoft.com/en-us/library/dd381412(v=vs.108).aspx

Microsoft Azure. (2018). Introduction to Azure Cosmos DB. Retrieved from https://docs.microsoft.com/en-us/azure/cosmos-db/introduction

Modzilla. (2018). MVC architecture. Retrieved from https://developer.mozilla.org/en-US/docs/Web/Apps/Fundamentals/Modern_web_app_architecture/MVC_architecture

MongoDB. (2018). Advantages of NoSQL. Retrieved from https://www.mongodb.com/scale/advantages-of-nosql

Shklar, L., & Rosen, R. (2006). *Web application architecture*. Chichester: John Wiley & Sons

Socratic Solution. (2017). Why MVC Architecture? – Socratic Solution – Medium. Retrieved from https://medium.com/@socraticsol/why-mvc-architecture-e833e28e0c76

Vines, J., Dunphy, P., & Monk, A. (2014). *Pay or Delay: The Role of Technology When Managing a Low Income* [PDF]. Newcastle: Newcastle University.

Vyas, D., Snow, S., & Mallett, M. (2015). *More than just Food: Field Visits to an Emergency Relief Centre* [PDF]. Brisbane: Queensland University of Technology.

Woelfer, J., & Hendry, D. (2010). *Homeless Young People's Experiences with Information Systems: Life and Work in a Community Technology Center* [Ebook]. Seattle: University of Washington.

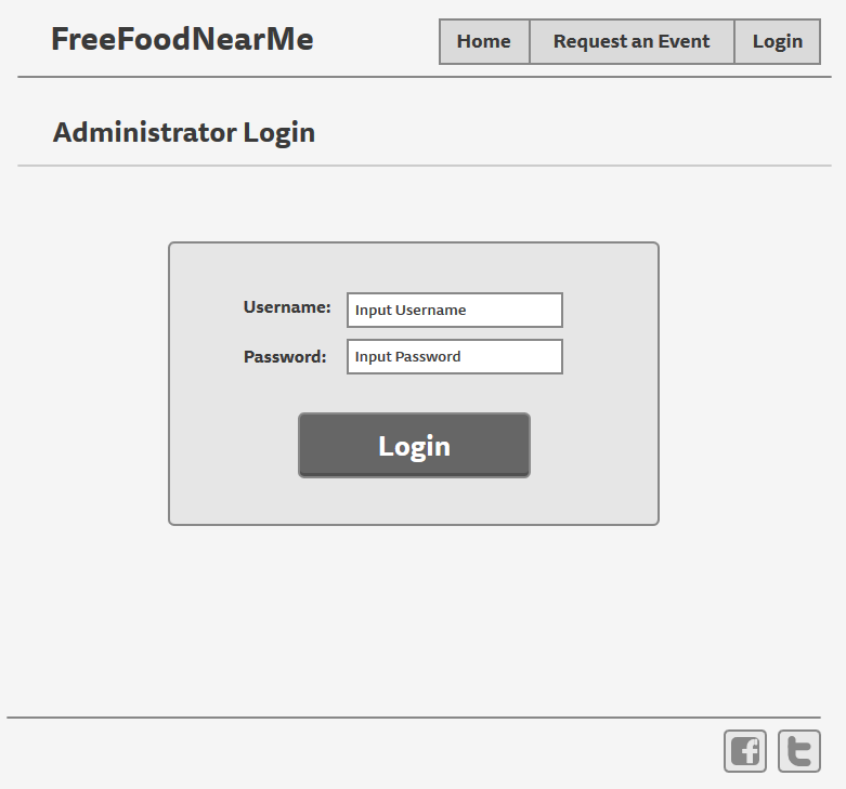
Appendices

Appendix A – Homepage Wireframe

Appendix B – Item Page Wireframe

Appendix C – Add Event Request Page

Appendix D – Admin Login

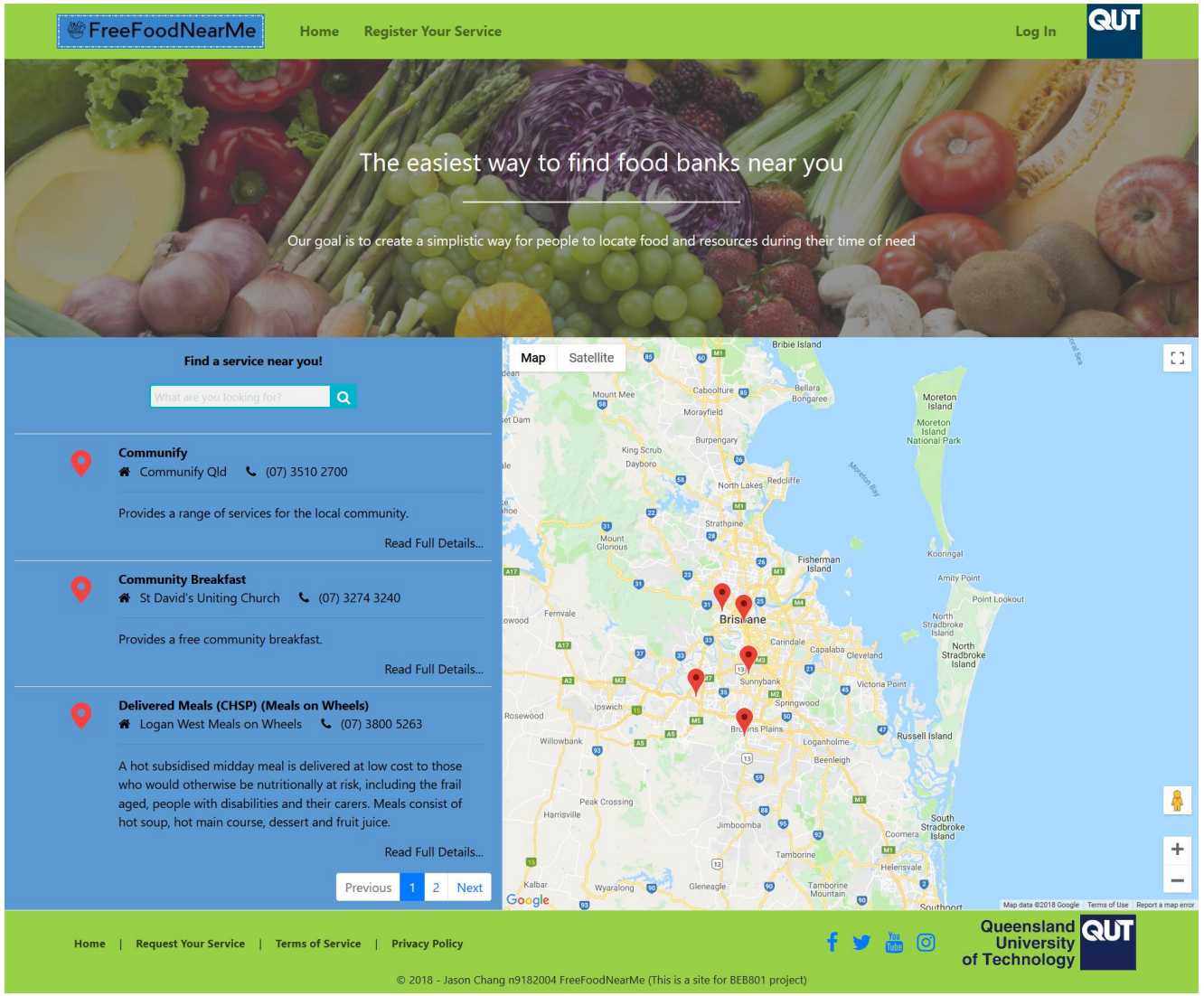


Appendix E – Admin Event Request Page

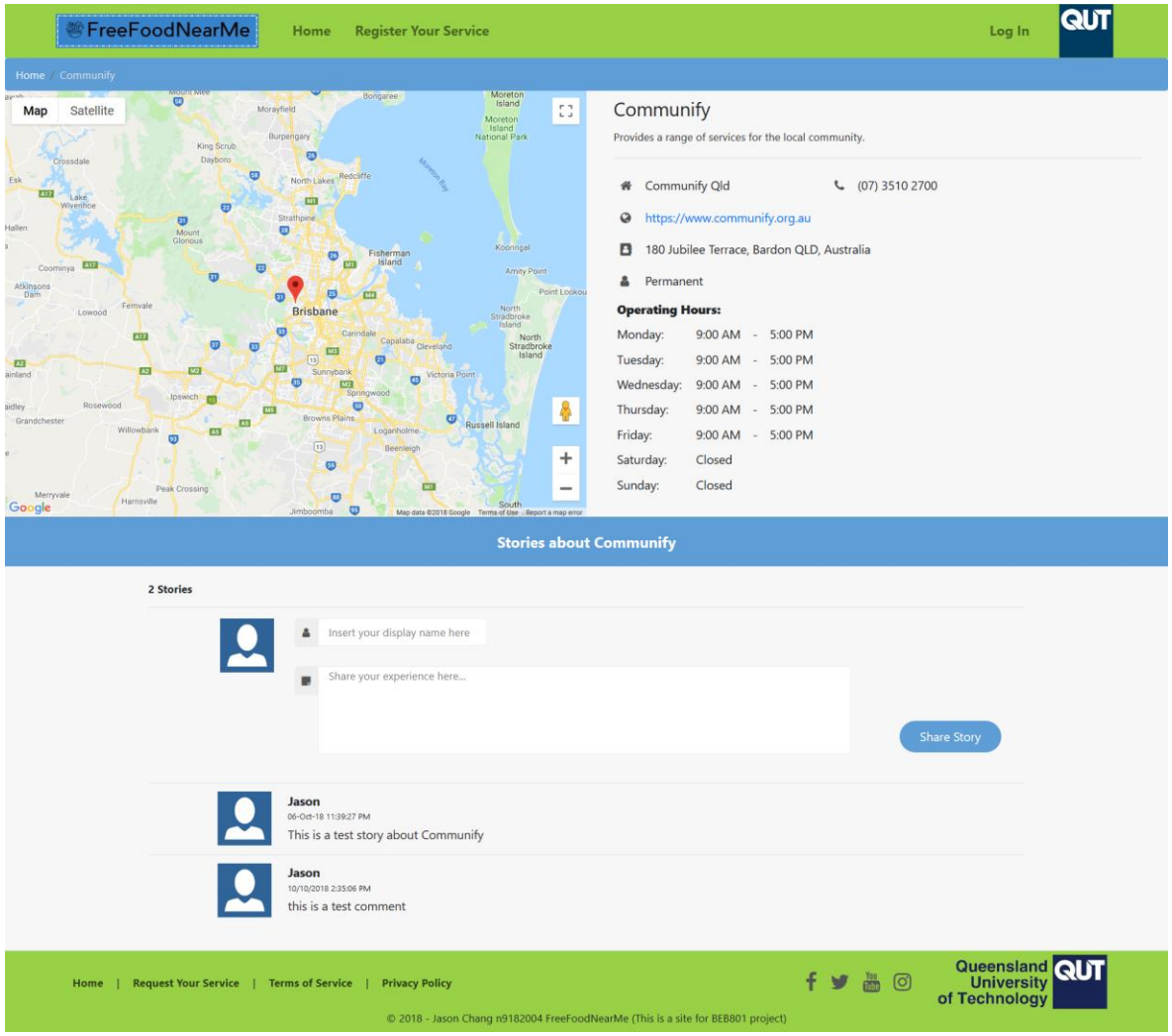
Appendix F – Individual Request Page

Appendix G – Gantt Chart of Project Progression

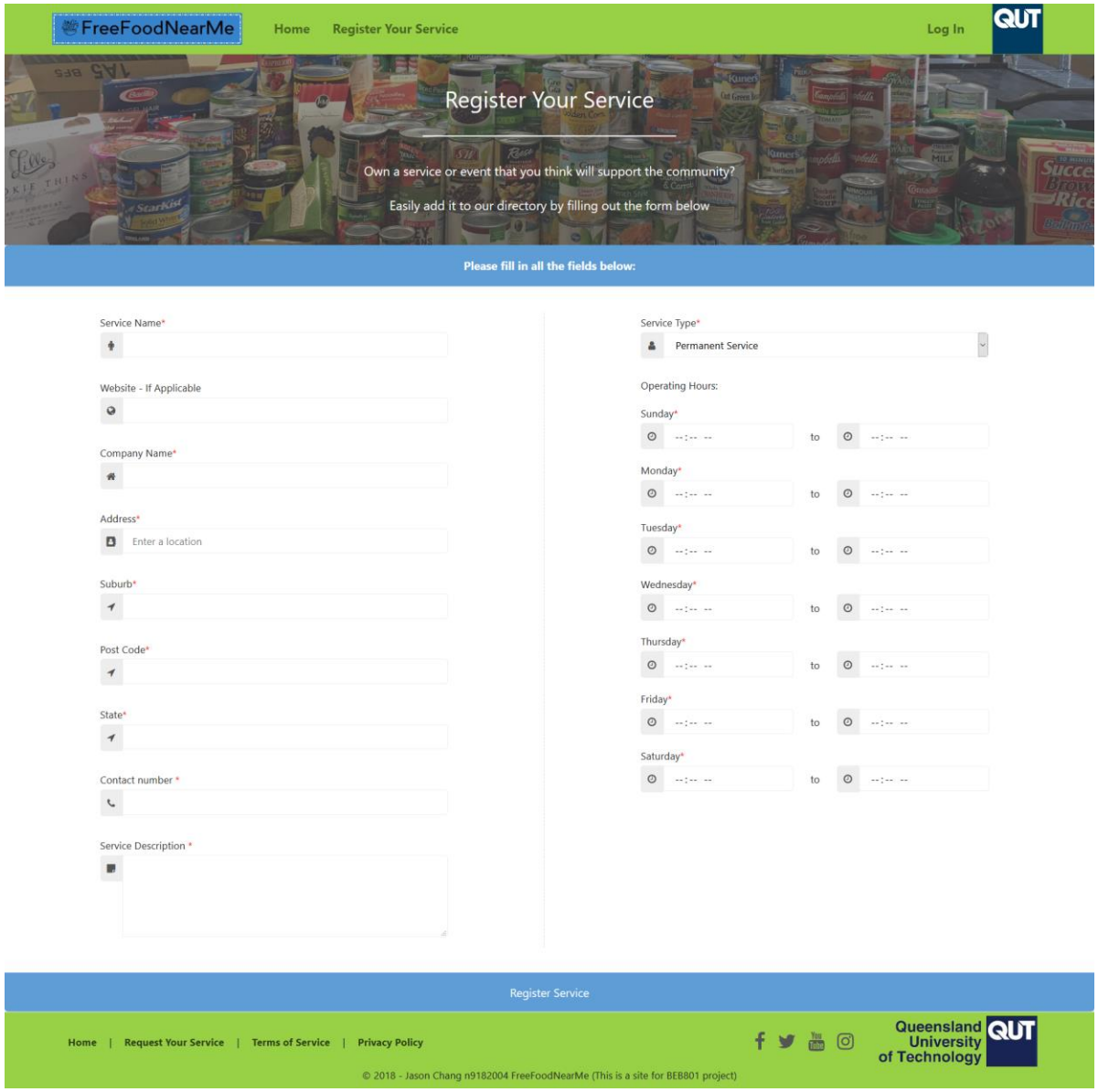
Appendix H – Updated Home Page



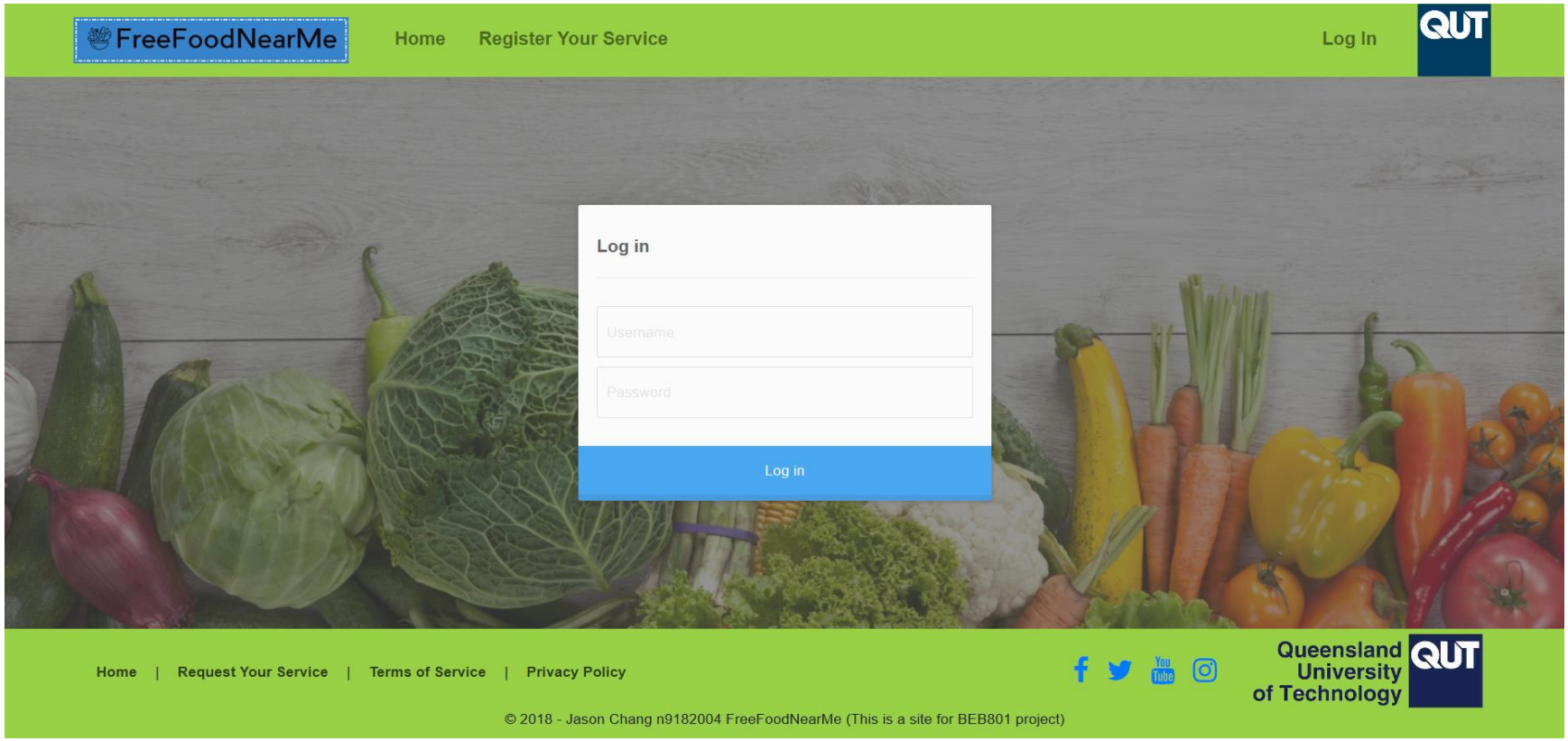
Appendix I – Updated Service Details Page



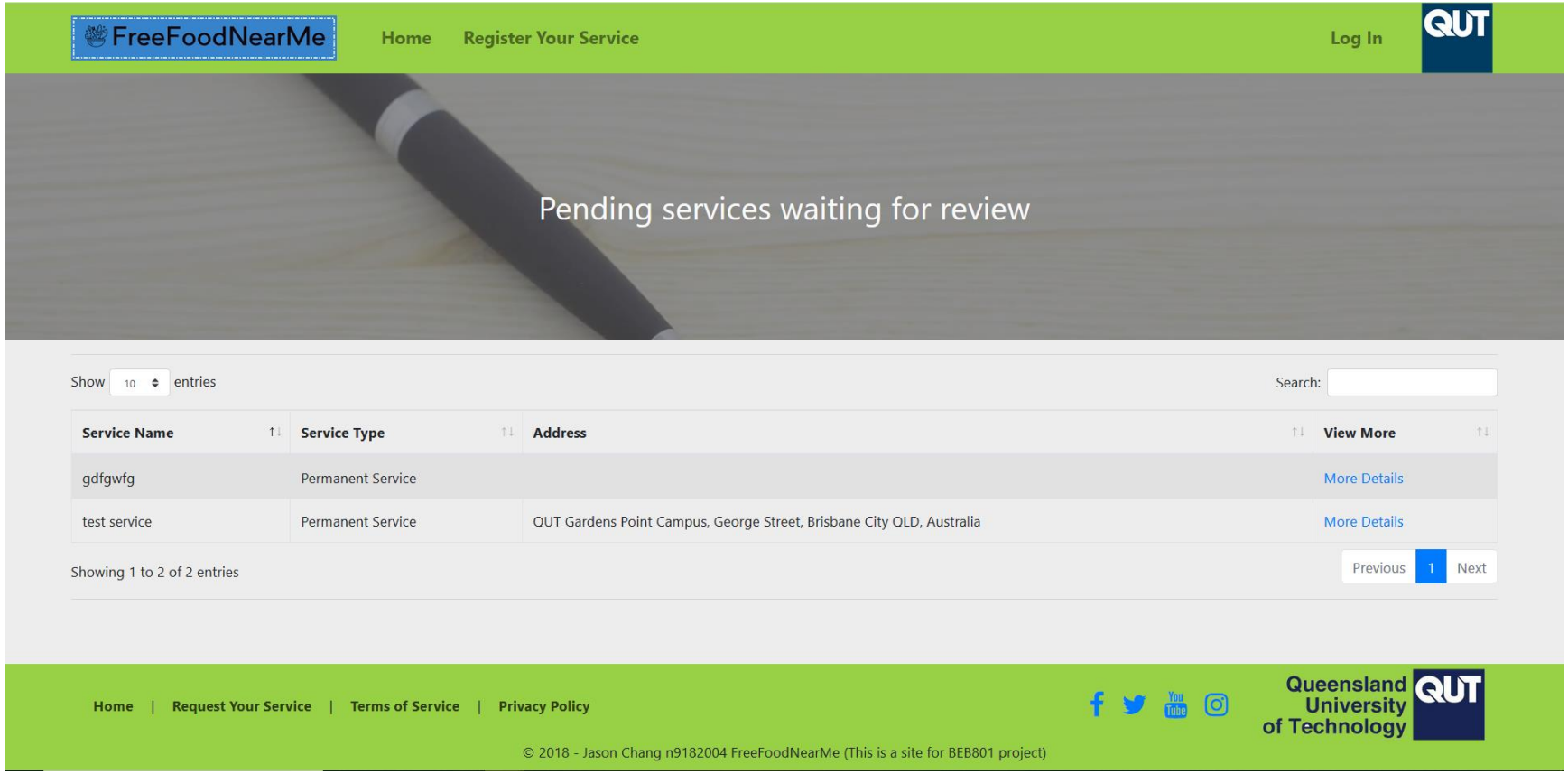
Appendix J – Updated Add Event Request Page



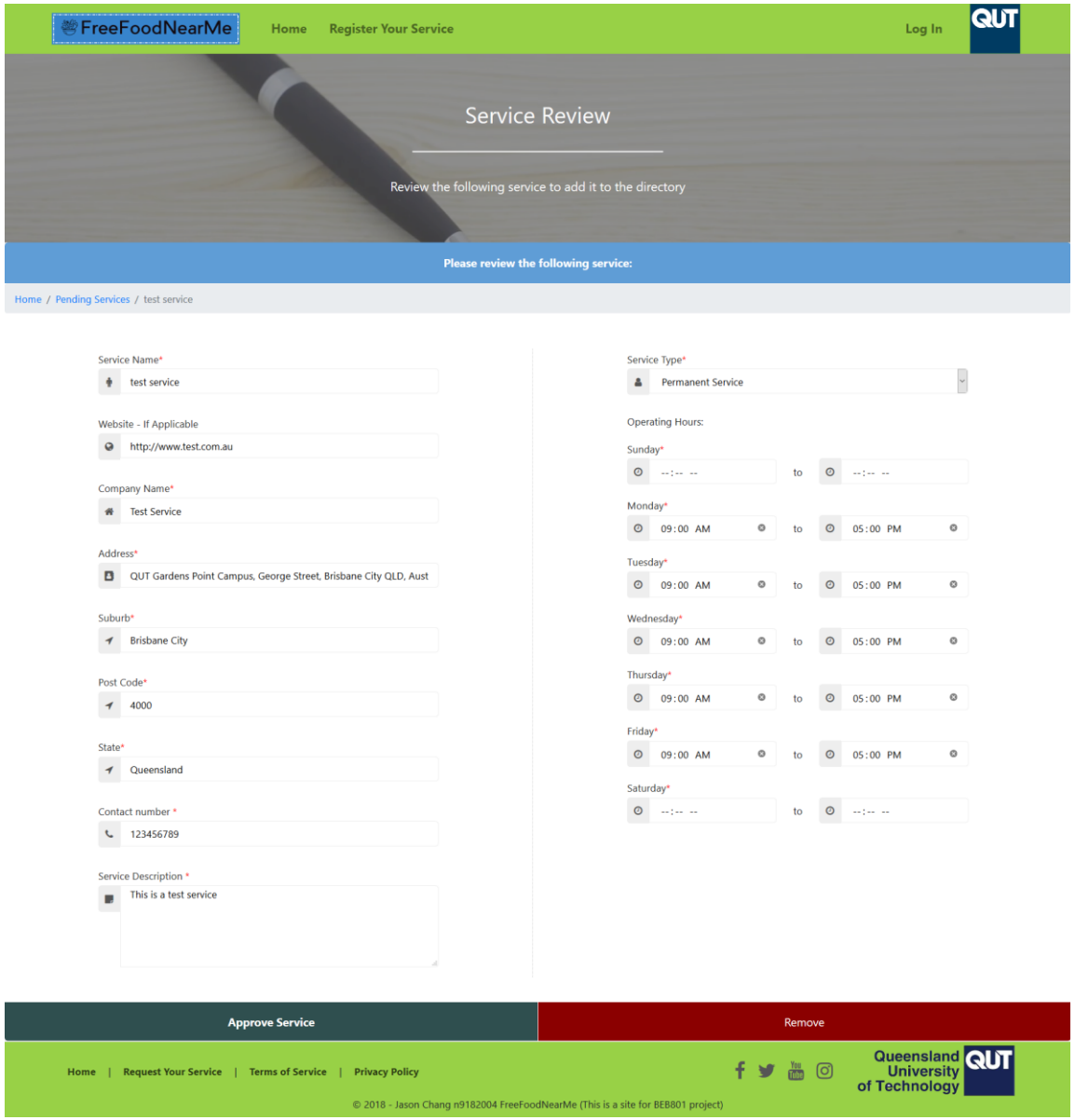
Appendix K – Updated Admin Login Page



Appendix L – Updated Admin Event Request Page

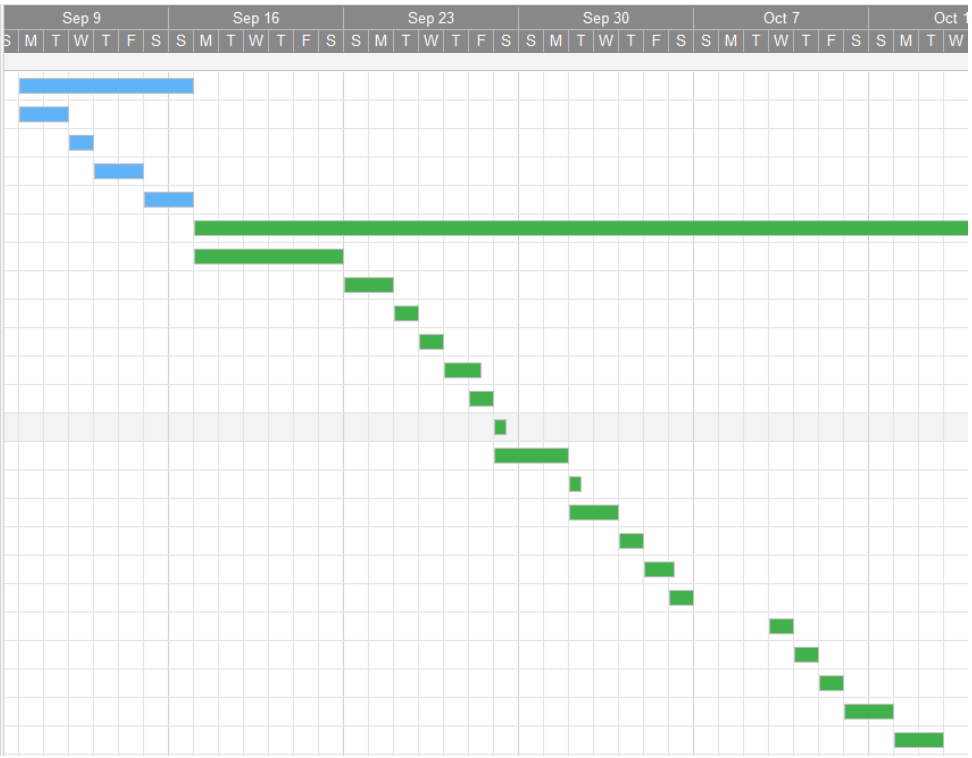


Appendix M – Updated Admin Individual Request Page



Appendix N – Updated Gantt Chart of Project Progression

Task Name	Start	Finish
Phase 1	09/10/18	09/16/18
Github Setup	09/10/18	09/11/18
ASP.NET web application setup	09/12/18	09/12/18
Libraries download and include	09/13/18	09/14/18
CosmosDB configuration	09/15/18	09/16/18
Phase 2	09/17/18	10/17/18
Other subject commitments	09/17/18	09/22/18
Develop draft site layout	09/23/18	09/24/18
Develop static homepage	09/25/18	09/25/18
Integrate google maps - static data	09/26/18	09/26/18
Implement CosmosDB interaction	09/27/18	09/28/18
Implement Homepage to be dynamic	09/28/18	09/28/18
Develop template event request page	09/29/18	09/29/18
Implement functionality for event request	09/29/18	10/01/18
Develop template admin login page	10/02/18	10/02/18
Develop static admin event request page	10/02/18	10/03/18
Update admin request page functionality	10/04/18	10/04/18
Develop static individual request page	10/05/18	10/06/18
update individual request page functionali	10/06/18	10/06/18
Supervisor meeting - feedback	10/10/18	10/10/18
Code refactor from feedback	10/11/18	10/11/18
Develop static item page	10/12/18	10/12/18
Update item page to be dynamic	10/13/18	10/14/18
Implement functionality for admin login	10/15/18	10/16/18



Task Name	Start	Finish
Implement functionality for admin login	10/15/18	10/16/18
Phase 3	10/17/18	10/21/18
Supervisor meeting - feedback	10/17/18	10/17/18
Code refactor from feedback	10/17/18	10/17/18
Web service testing	10/18/18	10/19/18
Update look and feel	10/18/18	10/19/18
Code refactor and cleanup	10/20/18	10/21/18
Phase 4	10/22/18	11/02/18
Preparation for project presentation	10/22/18	10/25/18
Progress report writing	10/26/18	11/02/18
Preparation for knowledge transfer	10/31/18	11/02/18