

Project Report

Institiúid Teicneolaíochta Cheatharlach



INSTITUTE *of*
TECHNOLOGY

CARLOW

At the heart of South Leinster

Project: Smart Shopping Application

Student: Jason O'Hara

Student Number: C00168956

Supervisor: Paul Barry

Contents

1. Introduction	3
1.1 Smart Shopping Application.....	3
1.2 Purpose of this Document	3
2. Description of Project	4
2.1 Core Application Features	4
2.1.1 Create Shopping List.....	4
2.1.2 Sync Shopping List to Cloud	8
2.1.3 Sort Shopping List.....	8
2.1.4 Move Product Within Shop.....	11
2.1.5 Signup and Login System	13
2.2 Supplementary Application Features	16
2.2.1 My Lists.....	16
2.2.2 My Shops.....	18
2.3 Administration Website	20
2.3.1 Create/Delete Shop.....	21
2.3.2 Move Product With Shop.....	22
2.3.3 Signup and Login System	23
3. Description of Conformance to Specification and Design	24
3.1 Core Application Features	24
3.2 Supplementary Application Features	24
3.3 Administration Website	24
4. Description of Learning	26
4.1 Technical	26
4.2 Personal	26
5. Review of Project	28
5.1 Project Successes	28
5.2 Project Weaknesses	28
5.3 Review of Technology Choices.....	29
5.4 Next Steps.....	30
6. Acknowledgements	30

1. Introduction

1.1 Smart Shopping Application

The Smart Shopping Application is a cross-platform, mobile shopping list application. Its key features are:

- Allow a user to create a shopping list in a quick and easy manner.
- Allow a user to save this list to a cloud database.
- Allow the user to add information regarding the location of a product in a given shop to a cloud database.
- Allow a user to sort a shopping list based on the location of products in a given shop.

The functionality that separates this application from the many other basic shopping list applications available is the list-sorting functionality. This uses crowd-sourced data to maintain a database describing the location of products across a selection of different shops and supplies users with the ability to integrate this information directly into their personal shopping list.

The intention is to develop this application in a cross-platform manner with a particular focus on mobile platforms. This application will be developed across three iterations from October 2015 until April 2016 by a single developer.

1.2 Purpose of this Document

The purpose of this document is to give an overview of what was planned for this project and to what extent the implementation of these plans were successful. This document will also describe the learning outcomes of the project (technical and personal) as well as a review of what was done well, what was done poorly, how the project could have been approached differently and what would be the next logical steps for this project if it were to continue.

The details of project implementation can be found in the Functional Specification and Design Documents of this project and details of research conducted as part of this project can be found in the Research Document.

2. Description of Project

This section describes the features to be delivered as part of the project. For further details on these processes please refer to the functional specification of this project.

2.1 Core Application Features

This section describes some of the core features of the application.

2.1.1 Create Shopping List

The application allows a user to maintain an on-screen shopping list.

Users may add a product to this list by selecting “Add” and then selecting the product they wish to add from the product menus. These product menus are populated upon start-up with information from the cloud database and contain all commonly purchased goods a user may wish to add to their list. If a user wants to add multiple instances of the same product they may do so by selecting it repeatedly, this will then be represented by a number beside the list item on the shopping list screen.

Users may remove products from this list by tapping on the product on the list and selecting “Remove” from the submenu that pops up. This will remove once instance of the product from the list. If a user selects remove for a product that there is only one instance of on the list, the product will be removed from the list entirely.

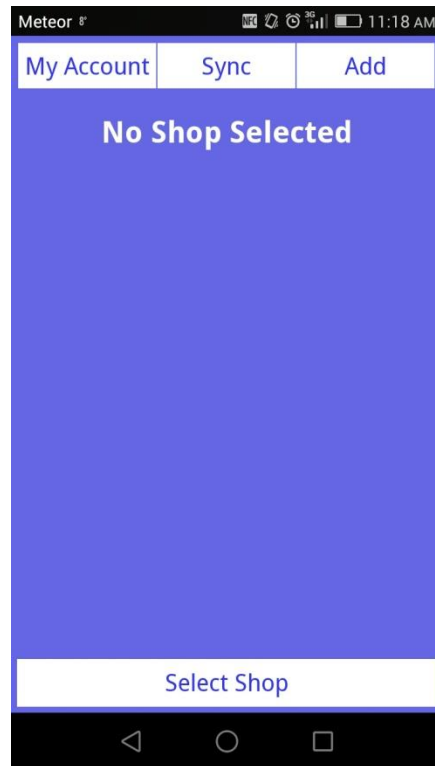


Fig 2.1 An Empty Shopping List. From here the user may select “Add” to add to the list

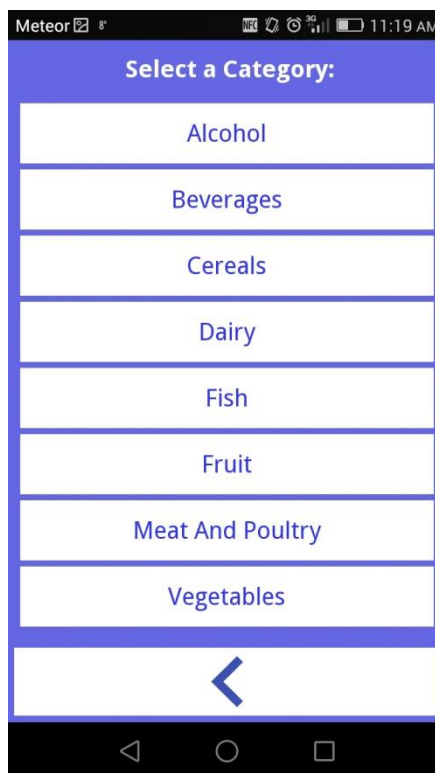


Fig 2.2 The main product selection menu. The user may select any of these categories to see the products within.



Fig 2.3 The fruit submenu. The user may now select any of these items to add to their list. The list above is populated with incomplete sample values.



Fig 2.4 The state of the list screen after selecting 3 apples, one banana(s) and one coconut .

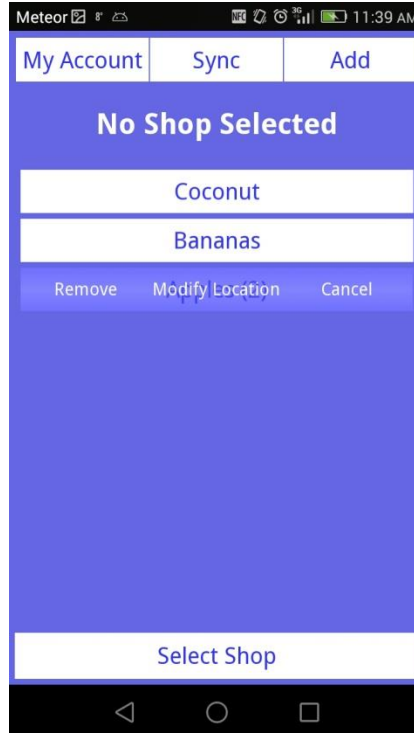


Fig 2.5 The submenu brought up by tapping on the apples list item. From here the user may tap “Remove” to remove once instance of this product.



Fig 2.6 The list screen after once instance of apples and once instance of bananas are removed.

2.1.2 Sync Shopping List to Cloud

The application allows a user to save their current list to the cloud.

When the application is first opened the user's last synced list is loaded. If the user has never synced with the cloud database then this defaults to an empty list.

To sync a list a user must select the "Sync" button from the list screen. The application will then save the user's current list to the cloud database so that when they next log the same items will be on the list.

2.1.3 Sort Shopping List

The application allows the user to sort a shopping list based on cloud data for a given shop.

Users may sort their list by selecting "Select Shop" on the main list screen and then selecting the shop they wish to sort on from the shop menus. These shop menus are populated upon start-up with information from the cloud database and contain all shops currently supported by the application, categorised by location.

Once a user has selected a shop they will be returned to the list screen where they will be shown their list, now sorted according to the selected shop. Products will be categorised by aisle and products that have no known location within the selected shop will be categorised as "Unknown Location".

A user may continue adding to this list and new products will be inserted into the correct aisle for the current shop. A user may also remove products from this list as before. If the user removes all products from a given aisle, that aisle information will no longer be displayed on screen.



Fig 2.7 An unsorted shopping list. From here a user may select the “Select Shop” functionality



Fig 2.8 The location menu. The user may select a location to view the shops at that location. The list above is populated with incomplete sample values.



Fig 2.9 The Dublin submenu. From here the user may select a specific shop to sort on.



Fig 2.10 The list from 2.7 now sorted based on data from shop "Iceland" in location "Dublin"

2.1.4 Move Product Within Shop

Once a user has a shop selected they may modify the location of an item of their current list for the current shop.

To do this they can tap on the list item to bring up its submenu and then select “Modify Location”. They are then presented with an aisle menu, this menu allows the user to select a new aisle for the product between aisle 1 and aisle 50. They may also select “Unknown Location”, this will make the product’s location unknown for the current shop. Any change will affect all users that select that same shop.

Once they have selected the new aisle for the product they will be brought back to the main list screen, updated to reflect the change. If the user selects this option with no current shop selected they will be prompted to select a shop.



Fig 2.11 The submenu brought up by tapping on the energy drink list item. From here the user may tap “Modify Location” to modify the location of this item within Iceland.

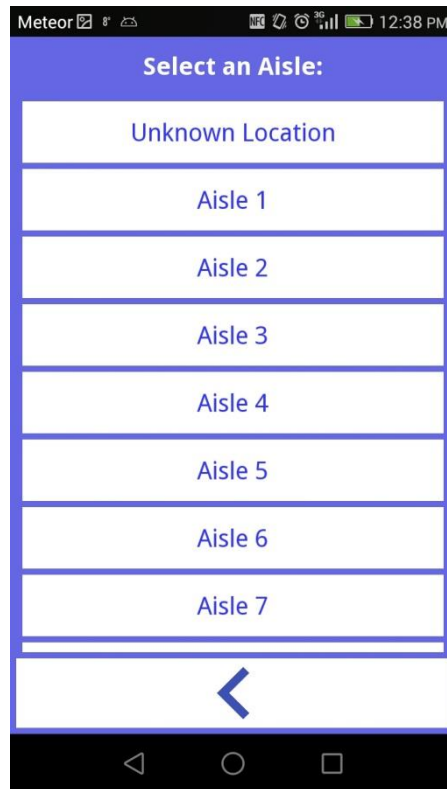


Fig 2.12 The aisle menu. From here the user may select a new aisle.



Fig 2.13 The updated list screen after the user moves energy drink to aisle 5.

2.1.5 Signup and Login System

The application allows users to create an account, login and logout.

If a user is already logged in on a device, the application will go straight to their main list screen upon start-up. If no user is logged in on a device the application will show the login screen on start-up.

From the login screen the user must enter a valid username and password in order to log in. Alternately, they may select the “Sign up” option, which takes them to the signup screen. On the signup screen the user may create a new account by entering a username and password (password is entered twice). The username selected must not be in use by any other user and both the username and password must conform to certain formatting rules. If the username or passwords entered on the signup or sign-in screens are invalid, an error message will be shown.

If a user wishes to log out they may select the “Logout” feature from the “My Account” screen. This will return them to the login screen. They may also delete their account by selecting “Delete Account” from the same screen.

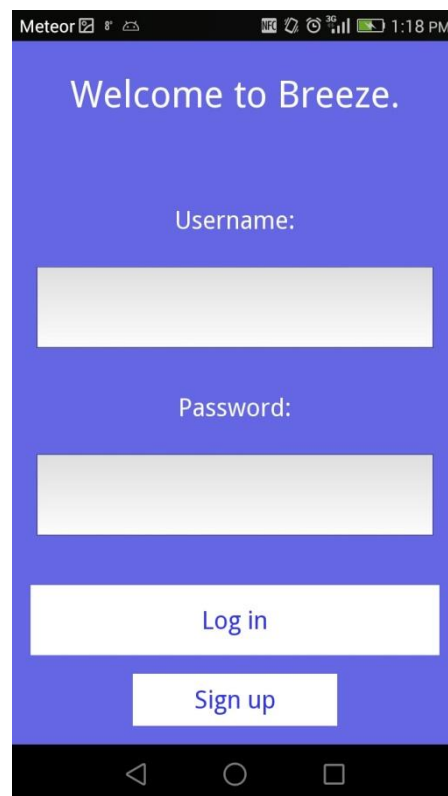


Fig 2.14 The login screen.

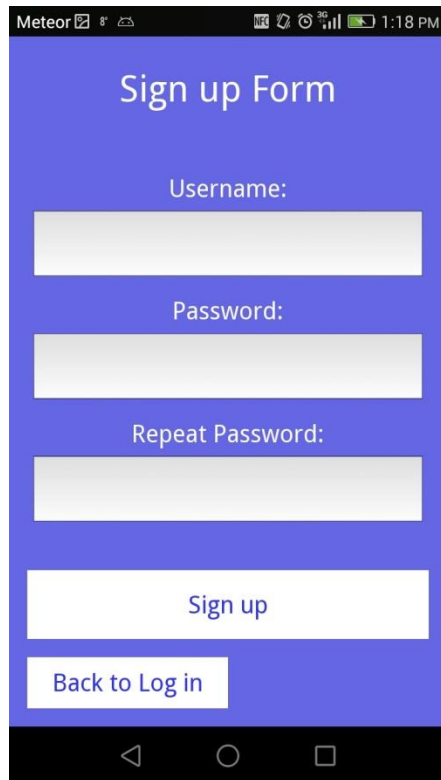


Fig 2.15 The signup screen.

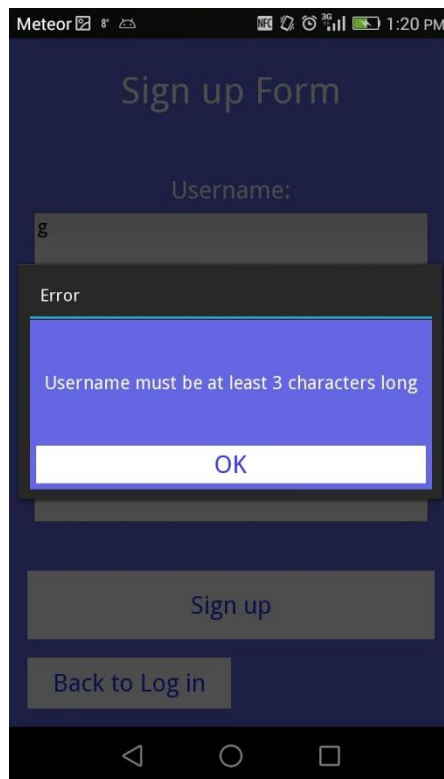


Fig 2.16 One of several error messages the user may receive when attempting to sign up.

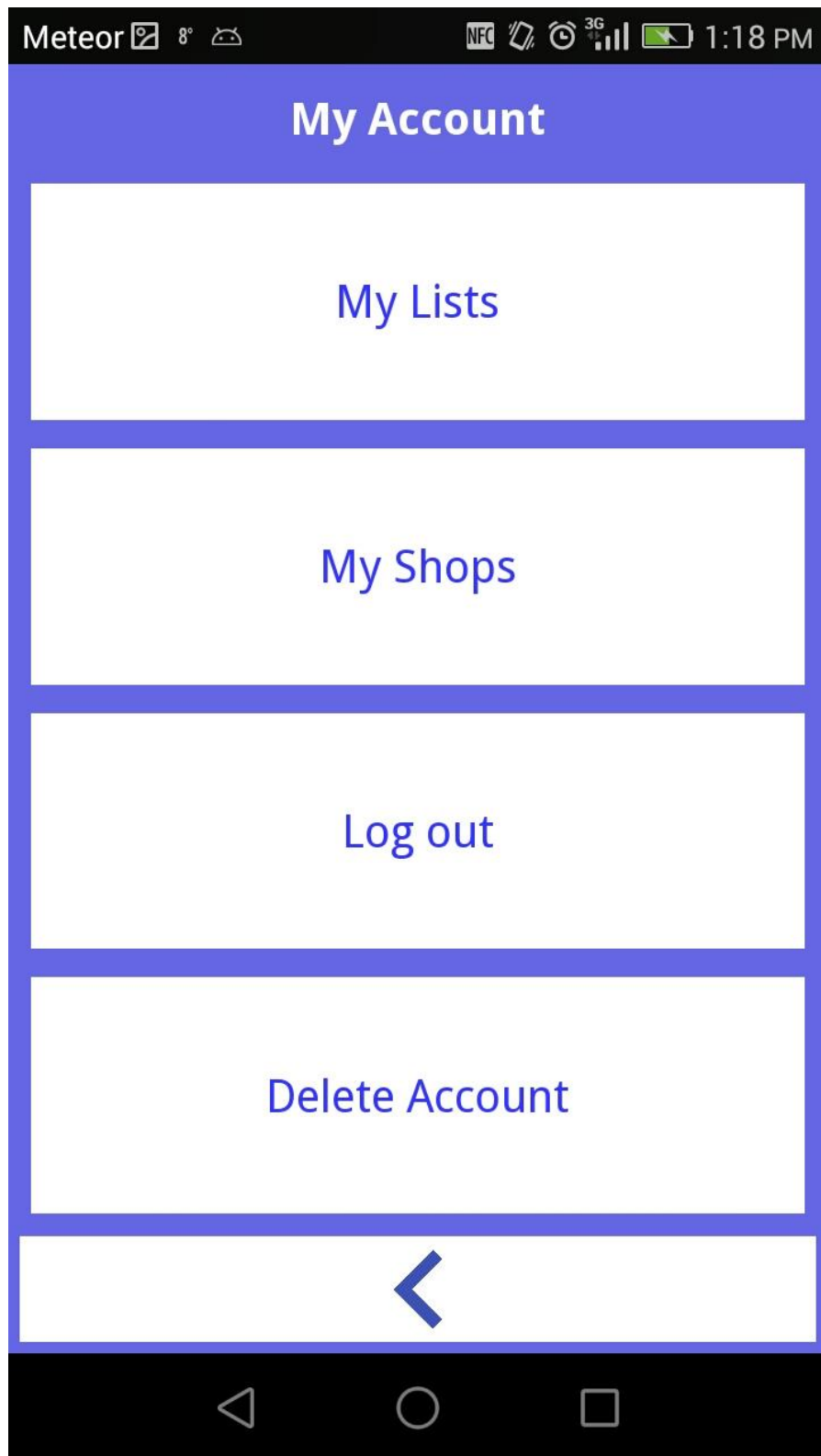


Fig 2.17 The “My Account” screen. Note the “Logout” and “Delete Account” buttons.

2.2 Supplementary Application Features

This section describes some of the non-core features of the application that were added during iteration three.

2.2.1 My Lists

The user may save more than one list at a time by using the My Lists feature. This allows users to save lists under a given name, load the lists at a later time and delete any of their saved lists. All activities associated with this feature are accessible from the My Lists screen, this screen is accessed from the My Account menu.

To add a new list a user will select “Save Current List”. From here they will enter the name they wish to save the current list under and it list will be added to their saved lists.

To select a previously saved list a user will simply tap on a list’s name on the My Lists screen. This will load the saved list to main list screen and display it.

To delete a previously saved list a user will select “Delete List”. They will then be shown the list deletion screen where they can select a list from their current saved lists for deletion. They will then shown their My Lists screen with the selected list removed.



Fig 2.18 The My Lists screen with two saved lists.

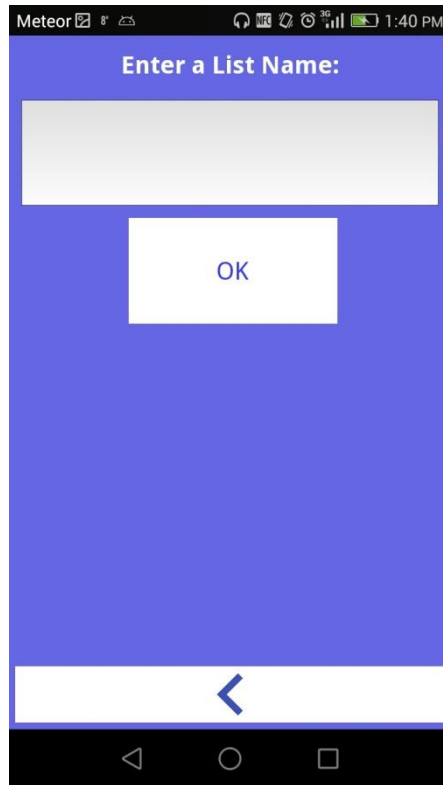


Fig 2.19 The prompt to create a new list.



Fig 2.20 The list deletion screen.

2.2.2 My Shops

The user may save a list of favourite shops by using the My Shops feature. This allows users to save shops to a screen other than the main shop selection screens, select shops from this screen as the current shop and delete shops from this screen. All activities associated with this feature are accessible from the My Shops screen, this screen is accessed from the My Account menu.

To add a shop to My Shops the user must select “Save Current Shop”. The current shop will then be added to My Shops.

To select a shop from My Shops as the current shop the user must tap its name. The user’s list will then be sorted on the selected shop and the main list screen will be shown.

To delete a previously saved shop a user will select “Delete Shop”. They will then be shown the shop deletion screen where they can select a shop from their current saved shops for deletion. They will then be shown their My Shops screen with the selected shop removed.

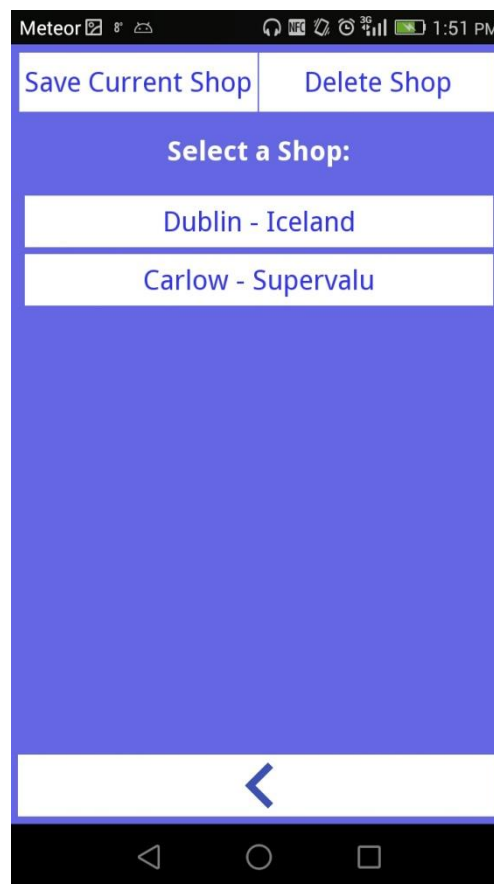


Fig 2.21 The My Shops screen with two saved shops.

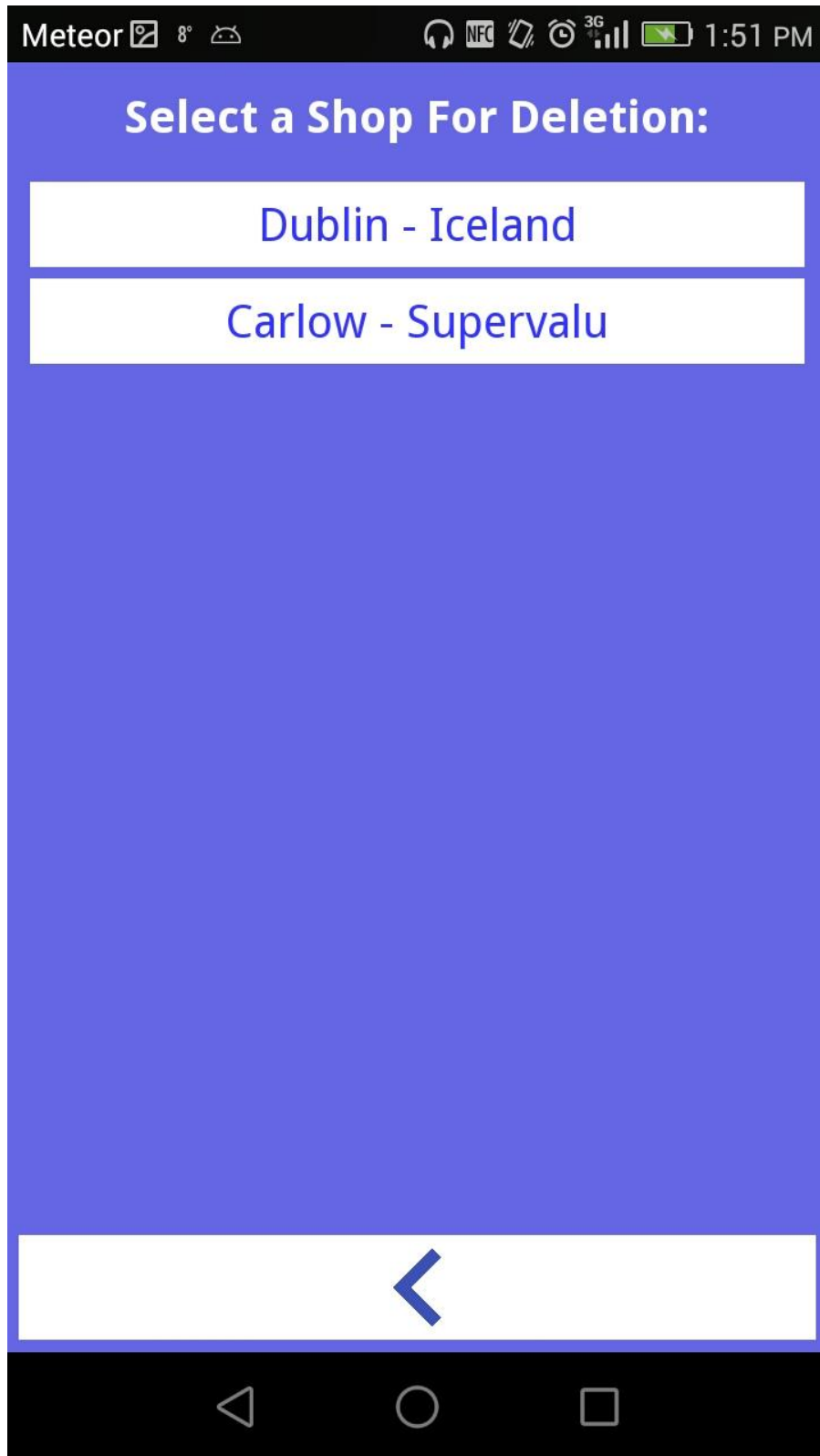


Fig 2.22 The shop deletion screen.

2.3 Administration Website

This section describes the features available to administrators only via the administration website.

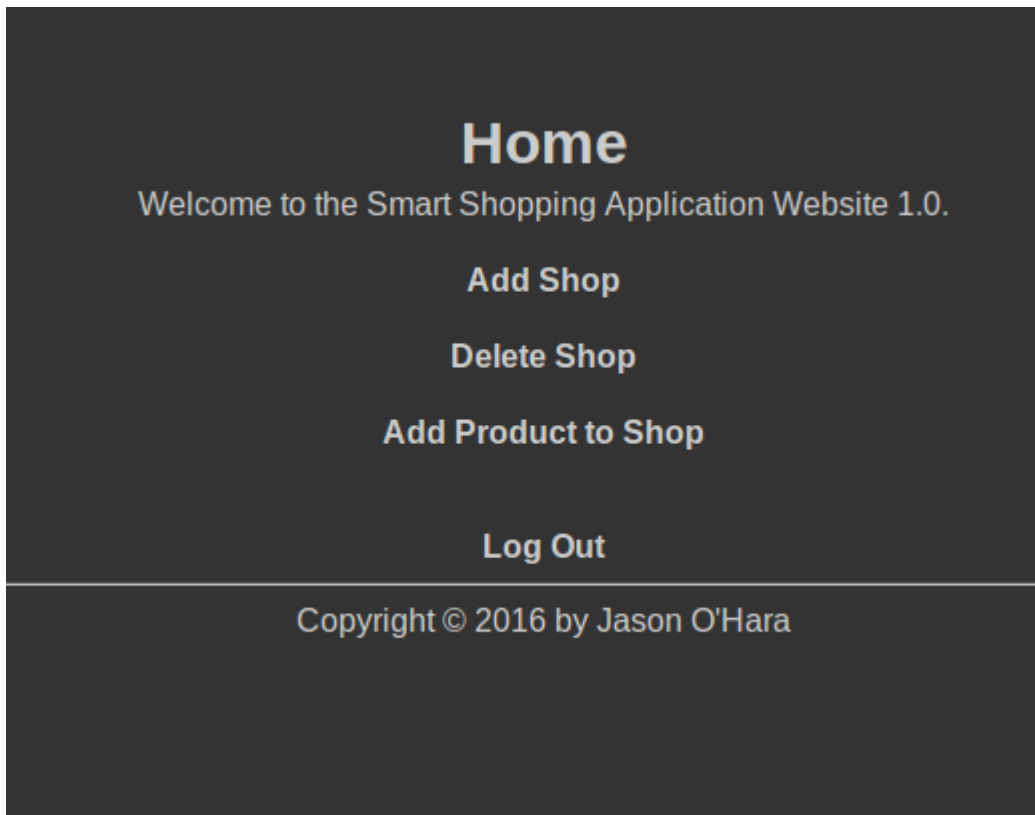



Fig 2.23 A screenshot of the administration website homepage.

2.3.1 Create/Delete Shop

From the administration website an administrator is able to add or delete shops and locations. These shops and locations will then appear on the shop and location menus for each application user.

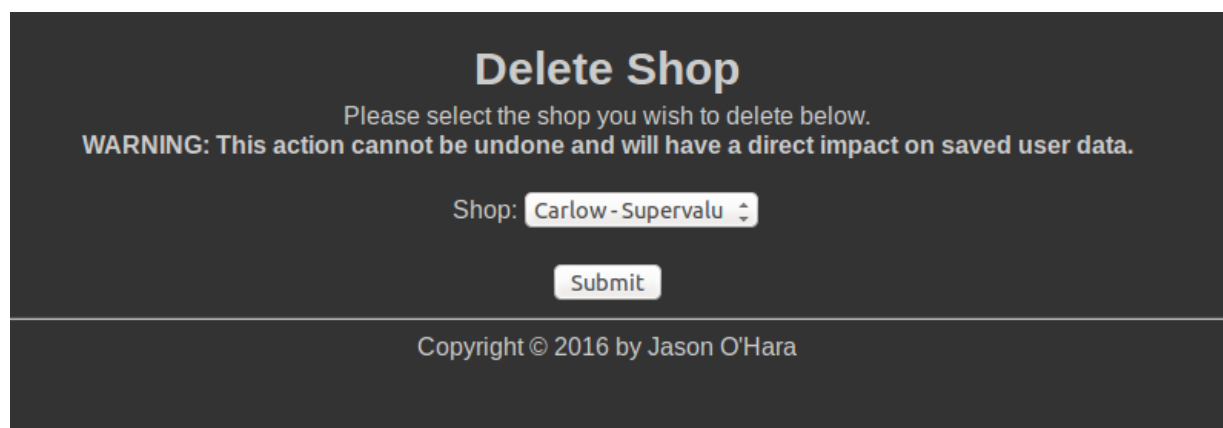
To add a shop an administrator will go to the “Add Shop” page. They will then enter the name and location of the shop they wish to create. If the details entered were valid then a new shop will be created. If the location entered does not already exist it will also be created.

To delete a shop an administrator will go to the “Delete Shop” page. They will then select the shop to be deleted from a drop-down menu. The selected shop will be deleted.



The screenshot shows a dark-themed web form titled "Add Shop". It contains two text input fields: "Enter Shop Name:" and "Enter Location:". Below these fields is a "Submit" button. At the bottom of the form, there is a copyright notice: "Copyright © 2016 by Jason O'Hara".

Fig 2.24 A screenshot of the Add Shop page



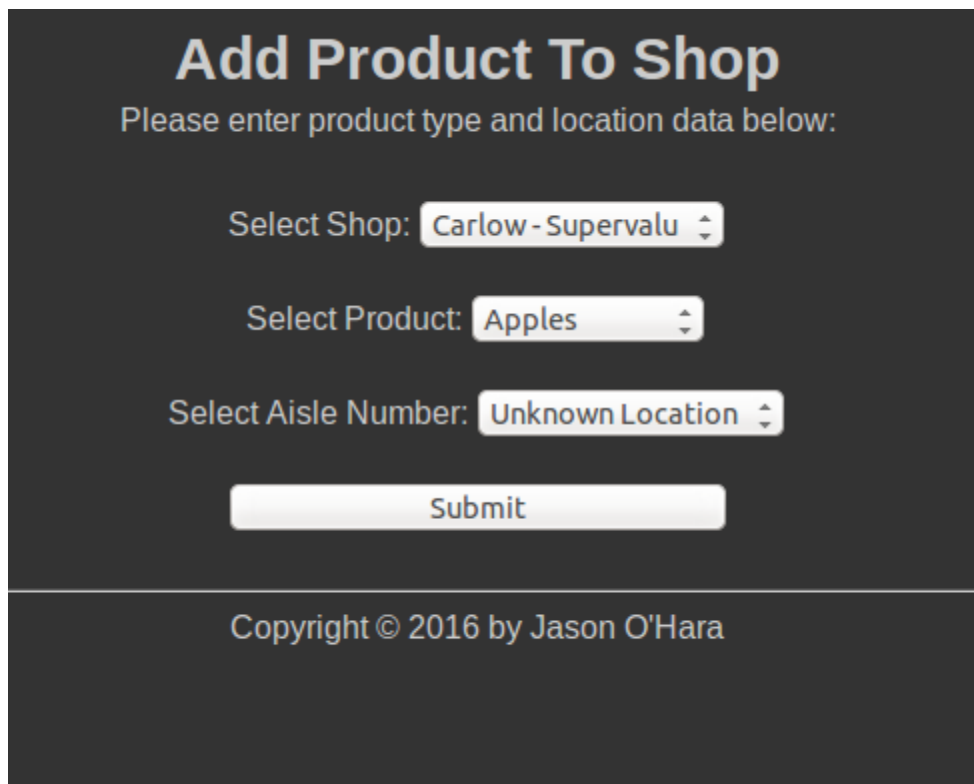
The screenshot shows a dark-themed web form titled "Delete Shop". It features a warning message: "Please select the shop you wish to delete below. **WARNING: This action cannot be undone and will have a direct impact on saved user data.**". Below the warning is a dropdown menu labeled "Shop:" with the selected value "Carlow - Supervalu". A "Submit" button is located below the dropdown. At the bottom of the form, there is a copyright notice: "Copyright © 2016 by Jason O'Hara".

Fig 2.25 A screenshot of the Delete Shop page

2.3.2 Move Product With Shop

From the administration website a administrator may modify the location of a product within a shop. This is the exact same functionality as described in section 2.1.4 (Move Product Within Shop), the difference being that this is accessible from a web browser and is designed to be easier for making a large number of changes in a row.

To modify product location an administrator will select the “Add Product to Shop” page. They will then select a shop, a product and an aisle number. When they submit the form shop data will be changed so that the product selected in the shop selected will be placed in the aisle selected.



Add Product To Shop

Please enter product type and location data below:

Select Shop: Carlow - Supervalu

Select Product: Apples

Select Aisle Number: Unknown Location

Submit

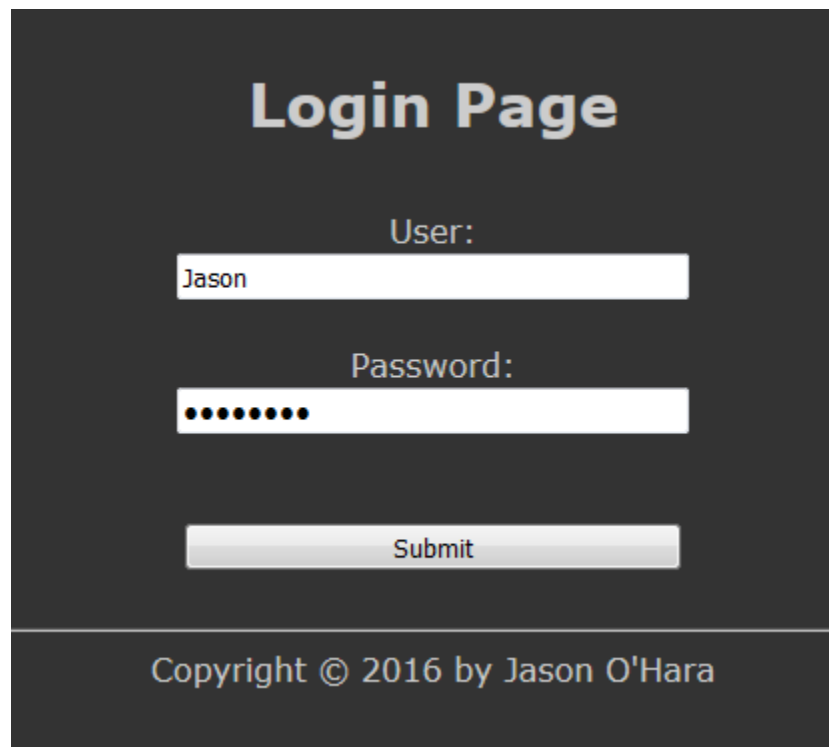
Copyright © 2016 by Jason O'Hara

Fig 2.25 A screenshot of the Add Product to Shop page

2.3.3 Signup and Login System

An administrator must log into the administration website in order to access any page. They are logged out when they close their browser session or when they select “Logout” from the homepage.

To login the administrator must simply enter a valid username and password for an administrator account.



The image shows a dark-themed login page. At the top, the text "Login Page" is displayed in a large, bold, white font. Below this, there are three main input areas: a "User:" label followed by a white text box containing the name "Jason"; a "Password:" label followed by a white text box containing seven black dots; and a light gray button with the text "Submit" centered on it. At the bottom of the page, a thin horizontal line separates the login form from a footer area that contains the text "Copyright © 2016 by Jason O'Hara" in a smaller white font.

Fig 2.26 A screenshot of the Login page

3. Description of Conformance to Specification and Design

This section describes instances where the project deviated from the initial plans and the reasons for this.

3.1 Core Application Features

It was originally intended that a user would type in the name of each product they wanted to add to their list, possibly with the assistance of some search hints. This was altered so that users instead select products for a series of menus. This change was made to abide by the principle that actions should be completed in the minimum button presses possible. This also reduced the amount of variables the system had to deal with by giving the developers more control over how the user can interact with the system.

The system for modifying the location of products in a given shop was altered significantly from initial plans. The original idea was to allow users control over the creation, deleting and naming of aisles. This would not only add quite a bit of complexity to the system but it would make the entire process much slower for users, resulting in them being less likely to contribute. This was modified so that a user need only select a product and an aisle number (or “Unknown Location”). The creation, deletion and naming of aisles is then handled for them by the system.

3.2 Supplementary Application Features

There was some deviation from plans in regards to the process by which users delete from My Lists or My Shops. It was originally intended that the user would delete lists/shops directly from the My Lists/My Shops screen. However it was technically easier to create a second, very similar screen for this purpose in both cases. This less elegant solution was chosen due to time pressure.

Moving into sprint three an options menu was considered as a feature. This menu would allow the user to make some small modifications to the behaviour, look and feel of the application. Ultimately, this was de-scoped due to time pressure.

3.3 Administration Website

It was intended at one point that the administration website would also have a feature for creating and deleting specific product types. This was de-scoped due to technical complications and the fact that this use case is not a common one.

The administration website is lacking a feature where someone accessing the website from a browser can create an administrator account. This is intentional, the only way to create an administrator account is to add it directly to the database. This was the only secure way the

feature could be implemented on schedule. This was not prioritised over other use cases as it is unlikely to be a common one in small projects such as this.

4. Description of Learning

This section describes some of the key technical and personal learning outcomes of this project.

4.1 Technical

Python. Going into the project my understanding of Python was limited to a brief introduction. Now that it's concluded I feel I have a strong understanding of many of the key principles of the language and would feel comfortable working on a larger Python project in future.

Kivy. I'm glad to get a large amount experience in this emerging and exciting cross-platform interface development framework. I will look forward to using this technology in future projects.

Flask. As a result of the work done on this project and the work done in the Web Development module I'm now confident in my ability to create Flask applications of various kinds and deploy them to the cloud if needed.

MySQL. While the SQL used in this project was rarely complex in nature I found it quite useful for reinforcing some of the concepts I learned during my 3rd year work placement as an SQL developer.

Mobile Application Development. Going into this project I had minimal experience with mobile application development using the standard Android/Java methods. Through research and coding conducted for this project I learned a vast deal about mobile application development, the different options available and the unique challenges it presents.

Cloud Computing. My limited experience with cloud development was also greatly expanded during this project. Going in I had quite a poor understanding of the processes and structures underlying cloud applications. Now that I've developed a cloud application of medium complexity I feel I have a much better grasp of this domain.

Documentation. The collection of documentation that I have written is the largest amount I have written for any one project. I found the experience of doing so very educational.

Testing. This project represents the largest amount of testing I have done to date in a general-purpose language. Not only have I seen firsthand the invaluable role that testing plays in delivering quality software but I have gained substantial experience in writing my own test cases in Python.

4.2 Personal

Communication. Despite this being a solo project there were many times in which I had an opportunity to improve my communication skills. I had to clearly communicate the technical

details of my project clearly to my supervisor during meetings, in writing while doing the project documentation and formally during the project presentations.

Formal Presentations. I have had difficulty in the past with presentations in front of large groups. I found the process of giving formal presentations as part of this project to be great practice.

Time Management and Personal Organisation. Managing the various aspects of this project and balancing it against the significant pressures from other modules has been a highly educational experience in time management, personal organisation and prioritisation.

Solo Development. Developing a project of this size as the sole developer is not something I've done before and it has opened my eyes to some of the challenges a project of this nature presents. In particular, I learned the importance of getting external feedback in this kind of situation.

5. Review of Project

This section will provide an overall assessment of the successes and failures of this project as well as a review of the technology used and what the next steps would be if this project were to continue

5.1 Project Successes

Overall, I would say this project was successful.

All features specified at the start of this project were implemented in what I believe to be a satisfactory manner. In addition, some supplementary features were included due to having free development time near the end of the project.

5.2 Project Weaknesses

Naturally, the project has also had several failures that can be learned from.

At one point it was intended that this project would be deployed to IOS as well as Android and Ubuntu. This was de-scoped for several reasons. Firstly, it was decided that the effort required to get involved in the somewhat restrictive process of IOS development was not worth the return. Also, neither the college nor myself could afford to purchase an expensive target machine for deployment and testing. This means that the application is only running on two platforms. While this is satisfactory to prove that it is indeed cross-platform, it is less than ideal.

The application in its current state also has some security concerns. All passwords are encrypted in the database and most URLs require that a secret key be sent as part of a request. However, there are a small amount of URLs where this process was not possible and they remain somewhat vulnerable to anyone who has deduced the inner workings of the application.

Finally, I feel that more could have been achieved with this application if I had started coding sooner. While the first few weeks of research were very valuable there was also a few weeks where little of value was learned. Upon reflection this time would have been better spent experimenting with the various technologies more than merely reading about them.

5.3 Review of Technology Choices

Overall, I feel the technologies chosen for this project were well-suited.

Python

Python was an unusual choice of language for a project such as this but I found it to be highly suited when the right tools were used. Compared to languages such as Java and C++ I found the code to be easier to read, write and understand. At no point in this project did I feel that the lack of lower-level details in the code restrictive and at no point did I find the performance of Python to be problematic.

The use of Python as a backend language is less unusual and it was highly effective in this role also.

Kivy

The use of Kivy was one of the more interesting aspects of this project. Despite the bad reputation that cross-platform tools of this nature have among some, found it to be very well suited to this project.

The main difference I immediately noticed between Kivy and other mobile development tools was how easy it was to create a basic program within the framework. There is a certain overhead of code associated with even the simplest application developed in Android Studio, with Kivy this overhead is much smaller. Kivy makes it very easy to start coding a basic prototype immediately.

I found the style of coding used by Kivy to be highly intuitive and easy to understand. A junior developer would have no problem picking up the fundamental ideas in a few hours. The Kivy mark-up language has a simple syntax and is very readable.

The flexible nature of Kivy is also helpful. Widgets can be created in a variety of different ways depending on the process, Kivy mark-up and Python can be mixed in a seamless and very natural way. The danger of this flexibility is that an extra degree of care is required in order to keep the code structured in a logical and consistent manner.

A difficulty when using Kivy was deployment to Android devices. Despite the Buildozer tool simplifying it, this is still a very heavyweight and delicate process. It has a large amount of dependencies, many of which are version-sensitive and not well documented. Poor documentation and online resources was an issue in general throughout this project, as can be expected with an emerging technology such as Kivy.

Other Technologies

Flask, PythonAnywhere and MySQL all suited this project well. All were chosen on the basis of being simple and robust technologies that supplied the services needed. This approach worked out well, as there were no significant negative issues involved in the use of any of them.

5.4 Next Steps

Should development on this project continue I would focus on addressing some of the weaknesses and going back to some of the de-scoped features (the following list is in no particular order):

- I would address the security concerns of the application.
- I would take steps to deploy the application to IOS and Windows.
- I would perform more testing with more users.
- I would develop a process for that would allow administrators to create and delete different product types via the administration website.
- I would create a process that allowed for more streamlined creation of administrator accounts, possibly involving email verification and review by other administrators.
- I would modify the process of deleting from My Lists so that it could be done from one screen.
- Likewise, I would modify the process of deleting from My Shops so that it could be done from one screen.
- I would add the de-scoped settings screen to the application with options regarding look, feel and behaviour.

6. Acknowledgements

I would like to thank the following people for their assistance with this project:

My supervisor, Paul Barry. His guidance and advice during this project was truly invaluable and sincerely appreciated.

All 4th year software development lecturers for adjusting their class and examination schedules to accommodate this project, often at their own inconvenience.

All of my classmates who have provided advice and constructive criticism during the course of this project.

The developers of Kivy, PythonAnywhere and all other free and/or open-source technologies used during this project. Without free tools a project of this nature would not be possible.