Conceptual Design Document
for
Dealio

March 2016

Version 1.0

Prepared By

WetSoft Inc.

Executive Summary

Dealio will be an Android based application tasked with assisting users in locating recurring restaurant deals. The application is targeting tourists, university students, downtown workers, retirees and anyone who enjoys a great deal. The initial release of the application will be developed with downtown Victoria, BC as the primary location. Later, the application may be scaled up to include restaurants in surrounding areas and other cities if required.

There are two primary methods of interaction with the application, the first of which involves a search methodology. This functionality will allow users to search for a restaurant deal based upon specific criteria, including:

- Day of the week
- Time of day
- Type of food (Chinese, Italian, etc.)
- Style of eating (Sit-Down, Fast Food, etc.)
- Specific food item (Wings, Pizza, Burgers, etc.)

There are two methods of search, basic and advanced. The advanced search breaks searching into the above categories, allowing users to enter keywords for each category, but not requiring input for every category. The basic search encompasses all categories of the advanced search into a single field where the user may enter multiple keywords belonging to any of the above categories. Upon searching, relevant deals and their respective restaurants will be listed, allowing the user to click on a deal to view more details on the deal and the restaurant offering the deal.

The second method of interaction incorporates the mapping interface from Google Maps. Here the user will be able to view their current location and the restaurants surrounding them, given that location services are enabled on their smartphone. Restaurants with a deal happening at the time of search will be highlighted. From here the user will be able to interact with the map interface and select a restaurant, showing more details on the current deals. The user will be given the option to get directions to the selected restaurant using the Google Maps application external from Dealio.

WetSoft Inc. strongly believes this application will provide a clear, quick and easy way for people in Victoria to discover great meal deals around them.

Revision History

Justyn Houle, Brandon Harvey, Sam Taylor, Graeme Turney	WetSoft Inc.
	Dealio
2016-02-23	Version 1.0

Version	**When**	**Who**	**What**
0.1	2016-02-23	Justyn Houle, Brandon Harvey, Sam Taylor, Graeme Turney	Initial drafting and documentation brainstorming
0.2	2016-02-25	Justyn Houle, Brandon Harvey, Sam Taylor, Graeme Turney	Creation of UI wireframes. Creation of Use Case diagram. Additional Use Cases added.
0.3	2016-03-01	Justyn Houle, Brandon Harvey, Sam Taylor, Graeme Turney	Insertion of UI wireframes into the document.
1.0	2016-03-04	Justyn Houle, Brandon Harvey, Sam Taylor, Graeme Turney	Finalization of document.

Table of Contents

Table of Figures

1.0 Introduction

Victoria is densely populated with restaurants and cafes of all sizes, often offering deals to entice customers. Unfortunately, locating meal deals on-the-fly isn't an easy task, leading to both frustration and possibly having to pay more for something which may be on as a deal at another restaurant down the street.

Dealio is the answer to this ever growing problem. The application will assist the user in locating restaurant and cafe deals using a simple search functionality and map interface, which will allow the user to search for current or future deals based upon restaurant style, food type, food item, day, or time, and locate the restaurant using the provided map interface.

1.1 Purpose and Objectives

Dealio will assist users in the downtown Victoria area in locating restaurant deals based upon specific criteria. The application is targeting all Android smartphone users but particularly university students, employees working downtown, tourists, and retirees. Each group has slightly different requirements determined by their age, linguistic abilities, income, etc.

1.2 Overview

This document depicts the Dealio application in its entirety, specifying all requirements, defining specific use cases, and providing an overview of a prototype of user interface. The document will initially begin describing the system and the requirements it fulfills, detailing both functional and non-functional requirements. The user interaction will then be described through Use Cases, Dialogues, and UI wireframes. The technical interactions and underlying infrastructure and layout of the application will be detailed within the system interaction section. Lastly a glossary contains all of the technical terminology and acronyms used in this document.

2.0 System Description

This section will describe details specific to the system, expectations of the application, and its requirements including constraints on the system and the development process as well as functions and services offered by the application.

2.1 Environment

The Dealio application is in development for the Android operating system. This decision was made because the android market share is the most dominant in mobile phones at the moment. Wetsoft Inc. employs developers who are familiar with the Android development kit and are unfamiliar with iOS. This means that Wetsoft is hesitant to develop for iOS due to the fact that unforeseeable roadblocks in development may occur leading to Wetsoft being unable to provide deliverables on time.

2.2 Functional Requirements

This section describes the functional requirements of the Dealio application. There are two essential features that make up Dealio. The first feature is a real-time navigable map that uses a smartphone's location-based services to determine the location of a user. Based on the user's location, the application will display will various nearby restaurants. The user will be able to determine whether a deal is currently being offered at the restaurant based on its icon on the map, and be able to click on the restaurant icons to view more information.

The second feature is a search function that allows a user to search for a deal based on the day of the week, time of day, style of restaurant, type of food, or specific food item. The user may do a basic or advanced search. In basic search, multiple keywords applying to any of the search categories may be entered. In advanced search, each category has its own input field to allow for more refined results. All fields will accept text input except the day and time, where day will accept input from a drop-down menu and time input will be from a time picker. A user may enter multiple options for each category, which will further refine the results.

2.2.1. Requirement 1 – Navigable Map

Description	The system must provide a real time navigable map that presents the locations of nearby restaurants based on the user's location.

2.2.2. Requirement – Display Deals

Description	The system must provide information on the map page so that a user can discern if a deal is currently available at a given restaurant.

2.2.3. Requirement – Basic Search for Deal

Description	The system must allow a user to search for a deal by searching for a specific keyword or phrase provided by the user
Comments	A user does not need to have all of this information to complete a successful search.

2.2.4. Requirement 3 – Advanced Search for Deal

Description	The system must allow a user to search for a deal using a keyword or phrase. The results can be filtered to have a specific week, time of day, location, style of restaurant, and type of food.
Comments	A user may leave input fields blank and still complete a successful search.

2.2.5. Requirement 4 – Search Multiple Items

Description	The system must allow a user to search for multiple items within the same field.

2.2.6. Requirement 5 – Up to Date Deals

Description	The system must provide a database that maintains up to date data on the deals provided by restaurants.
Comments	How this database is maintained is variable. The most viable option for the current scope of the application is manual input and maintenance.

2.2.7. Requirement 6 - Deal and Restaurant Details

Description	The system must provide a view that displays more detailed information about a particular deal and its restaurant.

2.1.8. Requirement 7 – Directions to Restaurant

Description	The system must allow Google Maps to open to show directions to a selected restaurant through various forms of travel.

2.2.9. Requirement 8 – Invalid Deal Reporting

Description	The system must allow a user to report an invalid deal so that it may be removed from the database.
Comments	A developer or an application administrator would have to review the report before committing a change to the database.

2.2.10. Requirement 9 – View Restaurant Reviews and Ratings

Description	The system must provide a user with a way to view the specified restaurant’s reviews and ratings.

2.2.11. Requirement 10 – Help Overlay

Description	The system must provide a toggleable overlay that displays help information to the user about the various functionalities of the system. This functionality will default to enabled when the application opens for the very first time.

2.3. Technical Requirements (Non-Functional)

Below is a comprehensive list of the technical requirements, dictating the system constraints and project flow of the system design.

2.3.1. Performance

Performance is a key technical requirement in the application. Users will most likely be in the rush of the downtown core, searching for a deal from the nearest available restaurant of their choosing. The application must be able to perform all search requests based upon location and time based sensors within a desirable amount of time (approximately under 5 seconds). If the performance goals are not met the application will lose its customer base.

2.3.2. Scalability

The application must be able to encompass a large and possibly ever growing customer base and restaurant deals. However, the scalability requirement is more so concerned with meeting the ability to display a multitude of restaurant deals in a useful manner; it is not as focused on the ever growing customer base as the app will not need have to store customer information, such as a profile. The application’s database will need to be able to grow and shrink based on the number of restaurants in the area providing deals.

2.3.3. Security

Security is not as focused on the customer’s end as the application will not require a login. Backend security and the prevention of MITM (man-in-the-middle) attacks and tampering are the primary concern. A worst case security breach scenario may allow access to user's smartphone devices via our application which is of course a major concern. Therefore, the application must meet standard security protocols and practices.

2.3.4. Maintainability

The application requires a database to host all the restaurant deals and daily deals around town. The accuracy and reliability of this data is detrimental to the overall performance and customer satisfaction of the application. Unfortunately, the current plan and design of the system does not incorporate any sort of automated way to continually update and ensure the validity of the data. Future implementations may incorporate a web scraping approach to continually update the database, but until then, manual data input will be required.

2.3.5. Usability

Usability is another key requirement as this application is very UI orientated. The UI must be intuitive and effective, providing the user with the means to navigate the application quickly and effectively. This application is very time oriented. The user should not be required to navigate a plethora of tedious screens to get a result; the results should be displayed promptly and in a meaningful manner. Should a user require help, the help information provided by the help overlay should be clear and precise.

2.3.6. Auditing and Logging

If the system encounters an error during operation the application must prompt the user with the option of sending a feedback report to the developers. Logs should be tracked and archived for reference at later dates.

2.3.7. Availability

The application must ideally have a high availability, although downtime is a must have in any software system. Downtime in this particular application isn't detrimental by design and shouldn't have any great affect on the customer's feelings and opinions towards the system. Maintenance would be planned between 12am-6am PST.

2.3.8. Required Hardware

The system will be an Android based application requiring an Android smartphone. The smartphone requires location-based services to be turned on to make use of some of the functionalities of the system. The system must have the necessary specification to be able to run the latest version of Google Maps or equivalent.

2.3.9 Multilingual Support

The system will be designed in such a way that multilingual support can be added later if desired. However, this feature will not be in the initial release as the cost effectiveness and development time cannot be determined without real-time analytics post-release.

3.0 User Interaction

This section will go into details regarding how a user interacts with the Dealio application. This is portrayed through Use Cases, a Use Case Diagram, and Dialogues.

3.1 User Description

The application Dealio is specifically targeting all Android smartphone users in the Victoria area. In particular, university students, employees working downtown, tourists, and retirees. Each group has slightly different requirements and preferences determined by their age, linguistic abilities, income, etc.

Although these demographics have variances, their overall requirements do not have a direct effect on the design and functionality of the application. The application will not have multilingual support until further investigation into the cost effectiveness of this feature has been determined after the initial release.

3.2 Use Cases

The following section describes a set of use cases that describe how a user may interact with the Dealio application.

3.2.1. Use Case 1 - View Restaurant Details Through Map

ID	1
Description	A user views the nearby restaurants based on their location on a map.
Preconditions	● Application is installed and operating as expected. ● Location-based services are enabled on the user's smartphone. ● Database has restaurants.
Basic Steps	1. User opens up the application 2. User presses the "Map" button 3. User is presented with a map with the user's current location in the centre of the screen and nearby deals flagged on the map 4. User presses on a flag and a tooltip of the restaurant's name pops up 5. User presses the name of the restaurant on the newly spawned tooltip 6. User is shown the restaurant's deals and details
Alternate Steps	Press the Menu button (top left corner of every screen, looks like three horizontal bars) and select "Map" from the side menu. This replaces basic steps 1-2.
Exceptions	
Unexpected Results	● User has location-based services off. The user will be given a prompt to turn location services on. If the user choses to keep location services off, Step 3 will present the map with a broad overview of Victoria instead of their current location.

	● User has no internet connection. Pressing the “Map” button will open a popup that will inform the user that there is no internet connection. This prompt will have a “OK” button which will redirect the user to the end of Step 1.
Postconditions	The user is now on the Restaurant Details Screen.

3.2.2. Use Case 2 - Search Location

ID	2
Description	A user wants to search for a specific type of food at a specific time of day and wants to view the location of this deal on a map.
Preconditions	● Application is installed and operating as expected. ● Restaurant deals are available. ● 3.2.1, 3.2.3, or 3.2.4 use cases must be complete to arrive at the Restaurant Details Screen
Basic Steps	1. The use case begins when the user is at the Restaurant Details screen. 2. User presses “Map” button on the restaurant details page and the map is opened with the location of the restaurant with the chosen deal in the centre of the screen.
Alternate Steps	
Exceptions	No search results after the user defines search criteria.
Unexpected Results	● Nothing in database matches the user’s search criteria. After Step 4, the user will be shown “no results found” and a “Search Again” button to redo Step 3. ● User has no internet connection. Pressing the “Search” button in Step 5 will open a popup that will inform the user that there is no internet connection. This prompt will have a “OK” button which will redirect the user to the beginning of Step 4.
Postconditions	The user views the location of the deal on the map.

3.2.3 - Use Case 3 - Basic Search Deals

ID	3
Description	The user searches for a restaurant deal based on a keyphrase.
Preconditions	● Application is installed and operating as expected. ● Restaurant deals are available
Basic Steps	1. User opens application 2. User presses the “Search” option 3. User fills out their desired search preference using the keyphrase field 4. User presses the search icon to search and display relevant results 5. User selects their preferred deal 6. User is taken to the next screen detailing the deal and the time of the deal on the deal’s restaurant screen 7. Highlighted is the day of which the deal the user selected occurs, and the deal itself among a list of other deals
Alternate Steps	Press the Menu button (top left corner of every screen, looks like three horizontal bars) and select “Search” from the side menu. This replaces basic steps 1-2.
Exceptions	No search results after user defines the search criteria.
Unexpected Results	● Nothing in database matches the user’s search criteria. After Step 4, the user will be shown “no results found” and a “Search Again” button to redo Step 3.
Postconditions	The user is now on the Restaurant Details Screen.

3.2.4. Use Case 4 - Advanced Search Deals

ID	4
Description	A user wants to search for a specific type of food at a specific time of day and wants to view the specific details of a deal.
Preconditions	● Application is installed and operating as expected. ● Restaurant deals are available.
Basic Steps	1. User opens application 2. User presses the “Search” option 3. User pressed “Advanced” 4. The user fills out their desired search preferences using the text input fields fields and dropdown filters presented 5. User presses the “Search” button which displays the available search results 6. User selects their preferred deal 7. User is taken to the next screen detailing the deal and the time of the deal 8. Highlighted is the day of which the deal the user selected occurs, and the deal itself potentially among a list of others
Alternate Steps	Press the Menu button (top left corner of every screen, looks like three horizontal bars) and select “Advanced Search” from the side menu. This replaces basic steps 1-3.
Exceptions	No search results after user defines the search criteria.
Unexpected Results	● Nothing in database matches the user’s search criteria. After Step 4, the user will be shown “no results found” and a “Search Again” button to redo Step 4.
Postconditions	The user is now on the Restaurant Details Screen.

3.2.5 - Use Case 5 - Viewing Google Reviews

ID	5
Description	A user views the reviews of the restaurant that external users have posted on Google Reviews.
Preconditions	● Application is installed and operating as expected ● User has internet connection (WIFI or data connection) ● 3.2.1, 3.2.3, or 3.2.4 use cases must be complete to arrive at the Restaurant Details Screen
Basic Steps	1. The use case begins when the user has navigated to the Restaurant Details screen 2. User presses the “Reviews” button 3. User views the reviews of the restaurant on the Google page for the restaurant external from Dealio
Exceptions	
Unexpected Results	
Postconditions	The user views the Google Review for a particular restaurant.

3.2.6 - Use Case 6 - Using Help Overlay

ID	6
Description	A user is unfamiliar with the functionality of the application and wishes to use the “Help Text” overlay to assist them.
Preconditions	● Application is installed and operating as expected
Basic Steps	1. User presses the “Menu” button 2. User selects the “(?) Help” overlay button to toggle the help overlay on and off 3. Upon selecting an element the user is presented with an overlay, describing the elements function and expected input for the user’s current screen
Exceptions	● The “Help Text” overlay is automatically enabled

	upon first-time startup to assist new users
Unexpected Results	
Postconditions	The user is able to familiarize themselves with the functionality of the application via the enabled “Help Text” overlay.

3.2.7 - Use Case 7 - Getting Directions from Google Maps

ID	7
Description	A user wishes to obtain directions to get from one location to the restaurant of their choice.
Preconditions	● Application is installed and operating as expected. ● User has internet connection (WIFI or data connection). ● 3.2.1, 3.2.3, or 3.2.4 use cases must be complete to arrive at the Restaurant Details Screen
Basic Steps	1. The use case begins when the user is at the Restaurant Details screen. 2. User selects “Directions”. 3. User is taken to their selected form of Google Maps (applications or browser). 4. User designates their starting position. 5. User selects form of travel. 6. User views directions to the restaurant.
Exceptions	
Unexpected Results	User has no internet connection. In this situation, the user will receive a dialogue that will inform the user that they have no internet connection. They can either try again or cancel the request to get directions
Postconditions	The user obtains directions to the restaurant via their selected form of travel.

3.2.8 - Use Case 8 - Reporting an Invalid Deal

ID	8
Description	A user has noticed that a deal presented in the application is no longer available at the restaurant. The user informs an application administrator to amend this.
Preconditions	● Application is installed and operating as expected. ● User has internet connection (WIFI or data connection). ● User must be on Restaurant Details screen (complete use case 1, 3, or 4)
Basic Steps	1. The use case begins when the user is at the Restaurant Details screen. 2. User notices that a deal is no longer available. 3. User presses “Report Invalid Deal” 4. User is taken to their selected email application, where the recipient’s email address and subject header are auto filled.(subject will follow the lines of “REPORT:INVALID DEAL FOR Maude Hunter’s - half price wings ”) 5. User can add additional content to the email to inform Dealio about the bad deal. 6. User hits send to send the email.
Exceptions	
Unexpected Results	● The user does not have an email application installed. The user is informed that they are unable to report the error without an email application.
Postconditions	An email is sent to the application administrators with an invalid deal report.

3.3 Use Case Diagram

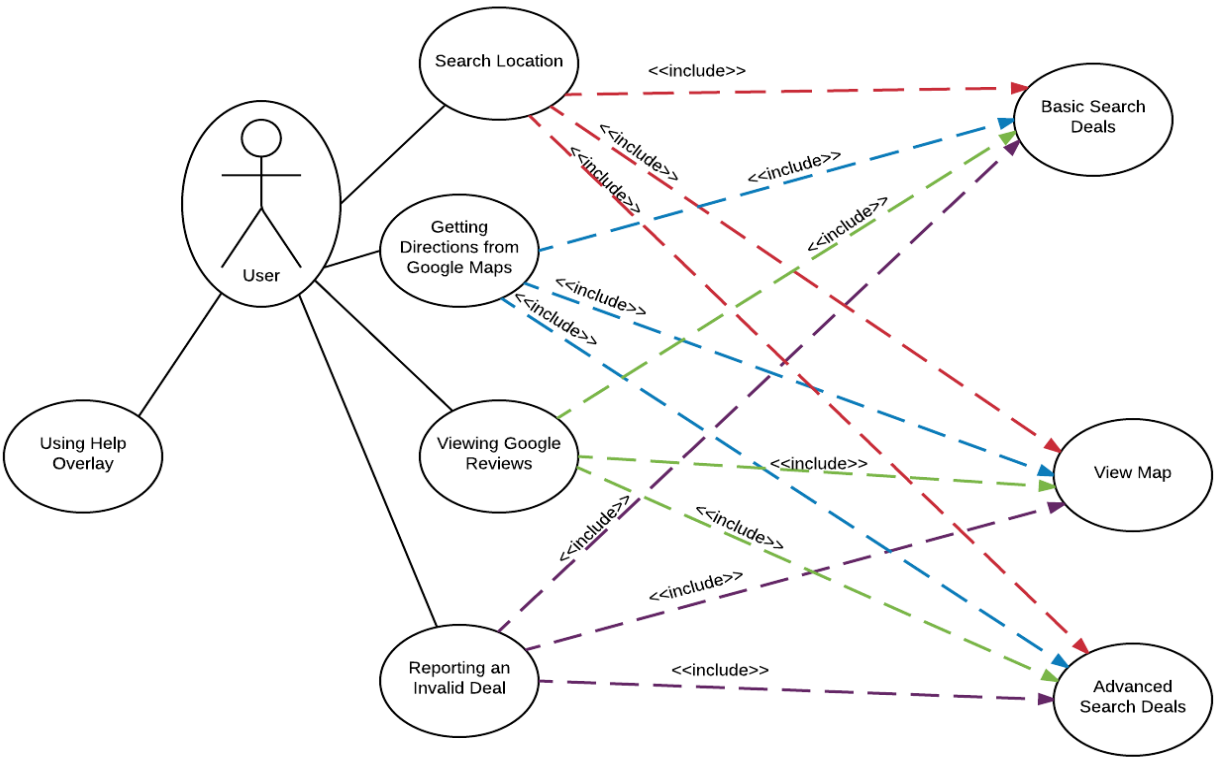

Figure 1. Use Case diagram

The Use Case diagram depicting the relationship between the user and each Use Case.

3.4 Dialogues

Below are two dialogues depicting what a typical user may want to achieve via the application, and how they end up running through the app to meet their end goal.

3.4.1 Dialogue - Map

Patrick is walking around downtown Victoria and decides he wants something quick and cheap to eat. He's not sure what he wants, but knows that he'll be walking there, so the closer the place is the better. Patrick pulls out his Android smartphone and opens Dealio. Since he doesn't have a specific type of food in mind but is looking for a place with location as his priority, he clicks on the "Map" button, and because he has his location services enabled, a Google Maps interface is opened showing his current location and coloured markers indicating nearby restaurants. The markers are coloured differently to indicate restaurants that currently have deals on. Patrick clicks the marker for the restaurant closest to him, which happens to be Cactus Club, and is provided with details about the current deals, including the time the deal ends and what the deal includes. He sees that chicken wings are on deal for the next hour, so decides to walk the short 5 minutes to take advantage of the deal.

3.4.2 Dialogue - Search

One Friday after school, Patrick and his friends decide they want to go out for a beer and a burger. Being students, they hope to find a place where there's a deal going on. Patrick remembers he has Dealio downloaded on his smartphone, and thinks this would be the perfect opportunity to take advantage of its search functionality. He opens the application and clicks the 'Search' button on the application's front page, and is prompted with a number of different search fields, including restaurant style, food type, time, and day of the week. They aren't concerned with location, so he leaves that field blank, and he doesn't have to edit the time and day because they are present to the current time and day. Clicking on the "Restaurant Style" search field, he is presented with a dropdown menu of possible restaurant styles to choose from, including sit-down, take-out, fast food, pub, etc. Patrick chooses "Pub", thinking that will be the best option for finding a deal for a beer and burger. Next he clicks on the food type and, along with a list of types of food, he is given the option to do a text search. Knowing that they're specifically looking for a beer and burger, he types 'beer' and checks that option, then types 'burger' and checks that option. After filling out the search fields, he clicks the 'Search' button and is shown a list of pubs that currently have a beer and burger deal on. The first option in the list is CANOE Brew Pub, which Patrick knows has good burgers, so he clicks on it to see the details. He's shown that there is a beer and burger deal on for the next 2 hours, so Patrick clicks on the "Map" button to see how to get there.

3.5 User Interface Wireframes

The following section showcases all of the screen that a user may encounter within the Dealio application. This is an initial mockup of the Dealio UI design and is subject to change.

3.5.1 Main Menu

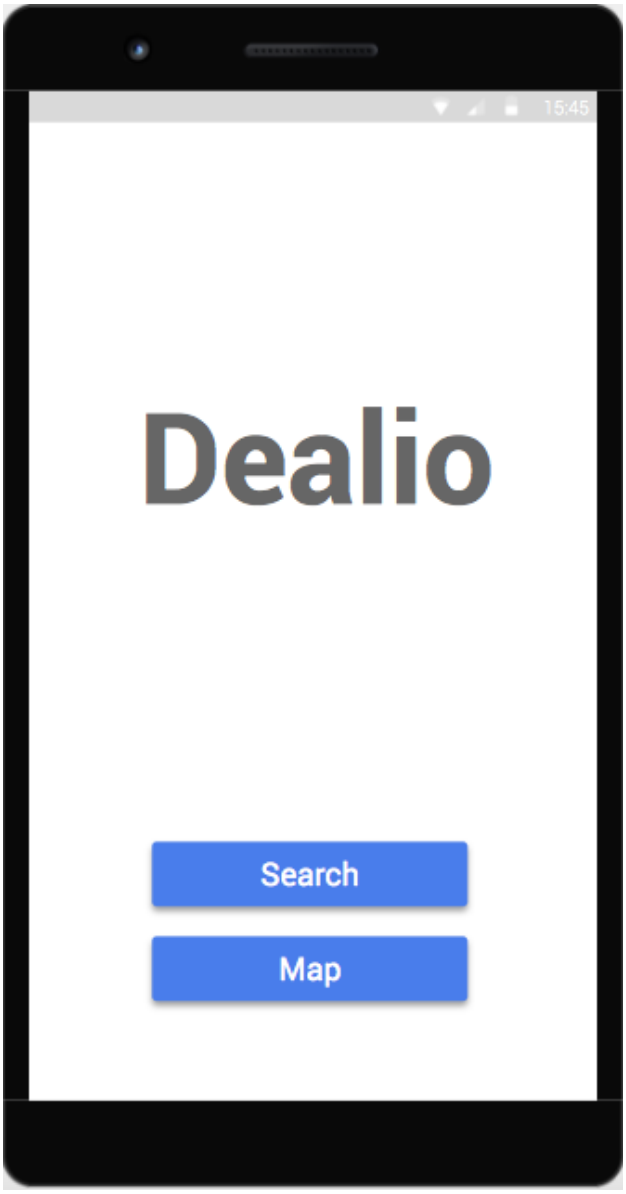

Figure 2. Main Menu Screen

The Main Menu screen is the first screen that a user encounters upon opening the Dealio application. A user is presented with two buttons; a search button and a map button. The button will take the user to the Map screen (section 3.5.2). The search button will take the user to the Basic Search screen (section 3.5.3).

3.5.2 Map

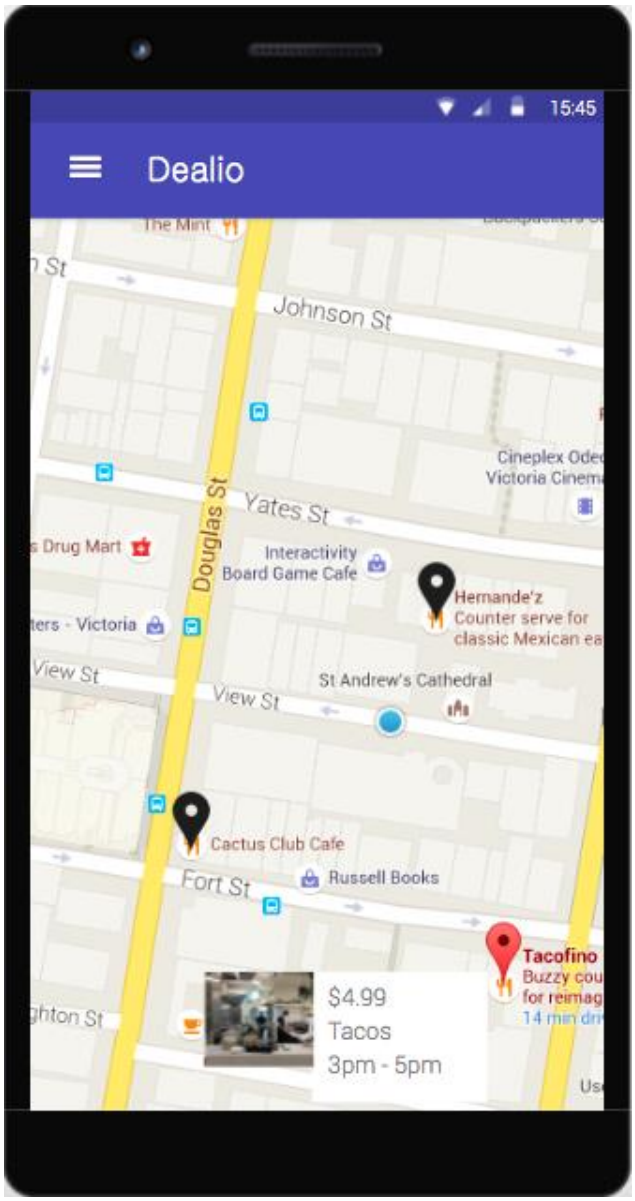

Figure 3. Map Screen

The Map screen showcases all the potential deals that are surrounding the user's location. The user is represented by the glowing blue circle, assuming that the user has location services enabled on their smartphone. Map markers that are coloured red highlight the location of a deal that is currently available. Map markers that are coloured black represent the location of a deal that is currently unavailable. If a user taps a map marker, a pop up appears that tells the user what deal is currently available and the span of time that it is available for. Taping the pop up will take the user to the Restaurant Details screen (section 3.5.6).

3.5.3 Basic Search

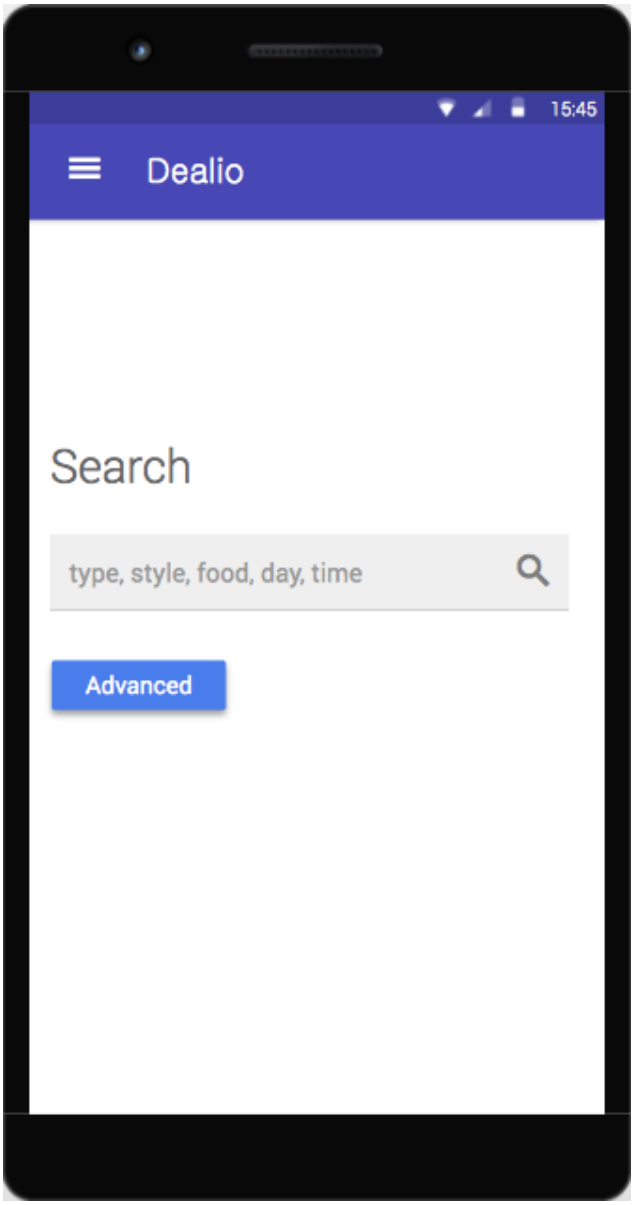

Figure 4. Basic Search Screen

The Basic Search screen allows the user to search for a deal. The text the user enters will search through all qualities of a deal, so this allows the user multiple keywords searching for the type of food, style of the restaurant providing the deal, specific food item offered in the deal, day of the week, or time of day. This screen also gives the user the option to perform an advanced search through the “Advanced” button.

3.5.4 Advanced Search

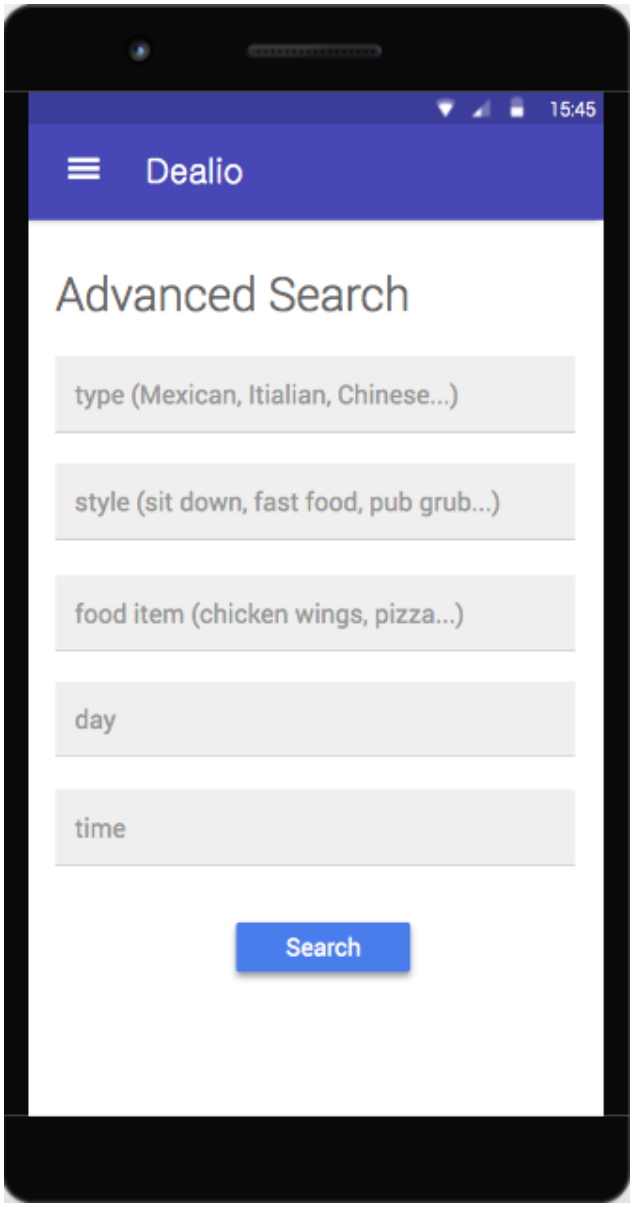

Figure 5. Advanced Search Screen

The Advanced Search screen allows the user to search for a deal by searching specific qualities of a deal. The qualities of a deal include the food type, style of restaurant, food item, day of the week, and time of day. Not every box needs to be filled but at least one needs input to perform a search. Searching for type, style and/or food item is done by the user entering text to search. The day provides a dropdown with the seven days of the week and time provides a time picker. Upon pressing the “Search” button the user will be taken to the Search Results screen (section 3.5.5).

3.5.5. Search Results

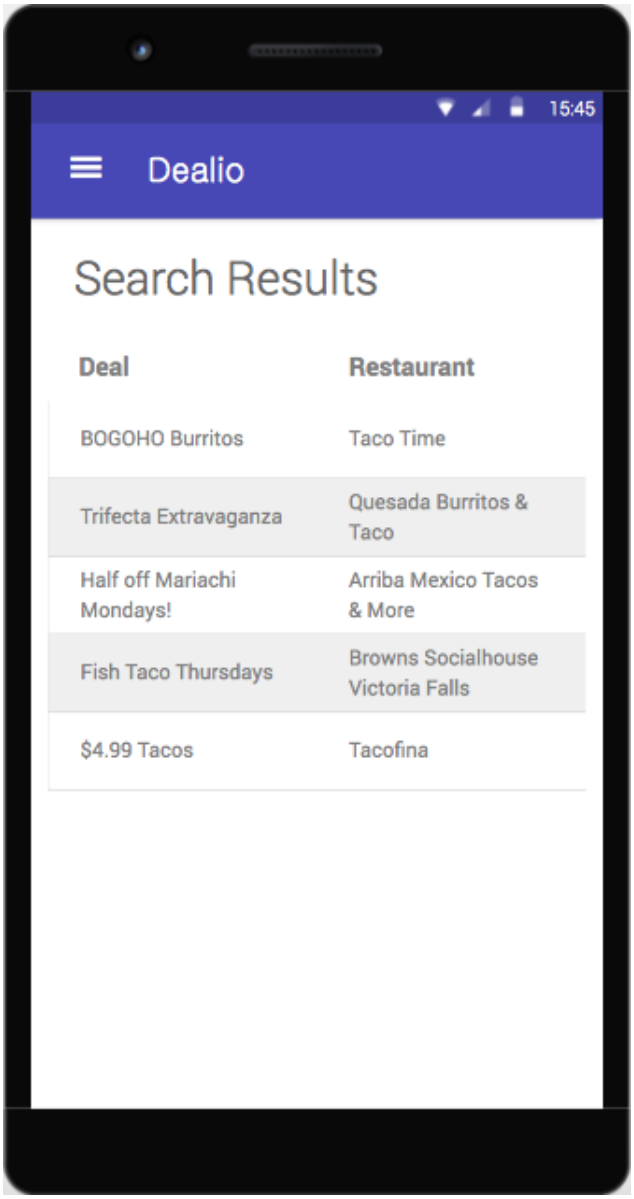

Figure 6. Search Results

The Search Results screen is shown after the user has selected either searched via the Basic or Advanced Search screens. Here the deals are listed, displaying both the deal and the associated restaurant. A user can then select one of the deals, bringing them to the Restaurant Details screen (section 3.5.6).

3.5.6 Restaurant Details Screen

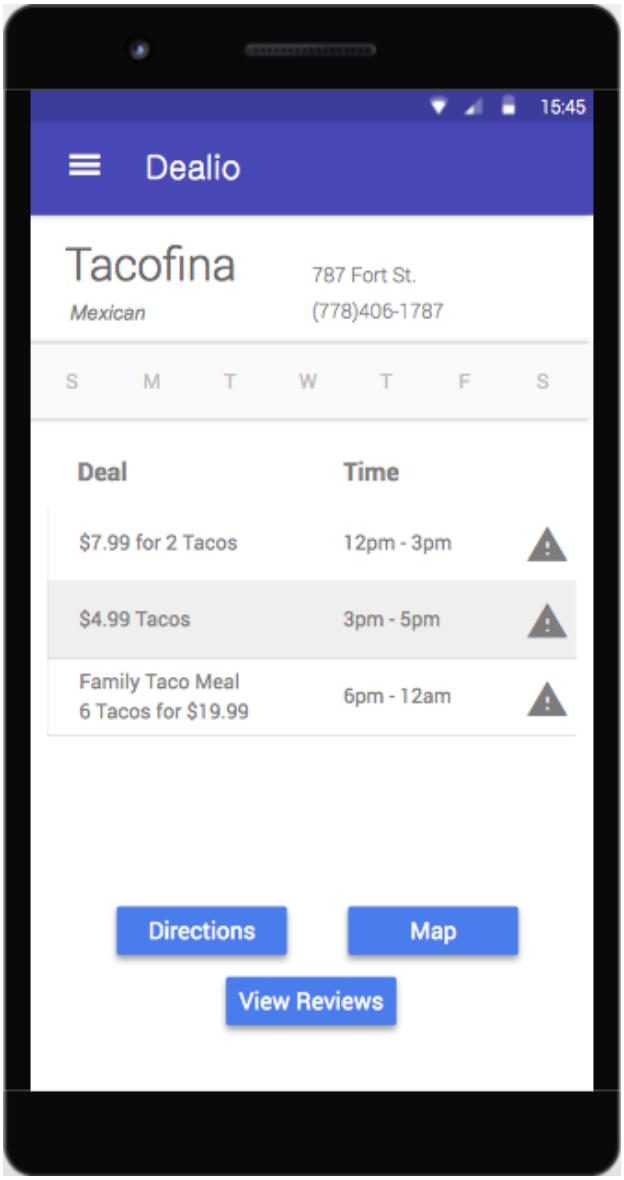

Figure 7. Restaurant Details Screen

The Restaurant Details screen is shown after a user has selected a restaurant from the search results screen, or from a map marker on the map screen. Here the deals for that particular restaurant are displayed, along with the time the deal is running for. A user can select a day of the week near the top of the display, defaulted to the searched or current day, which display the deals which are offered on the selected day. The address, phone number, and food style are also displayed for convenience. Should a user notice that a deal is invalid for a particular reason they can press the “warning” symbol next to the deal, which will direct the user to an external email client, automatically filling the subject line and recipient email so the user can notify a Dealio administrator about an invalid deal. Below are buttons to get directions to the restaurant via Google Maps, view Dealio’s custom mapping interface (section 3.5.2), or view reviews for the restaurant via Google.

3.5.7 Side Menu

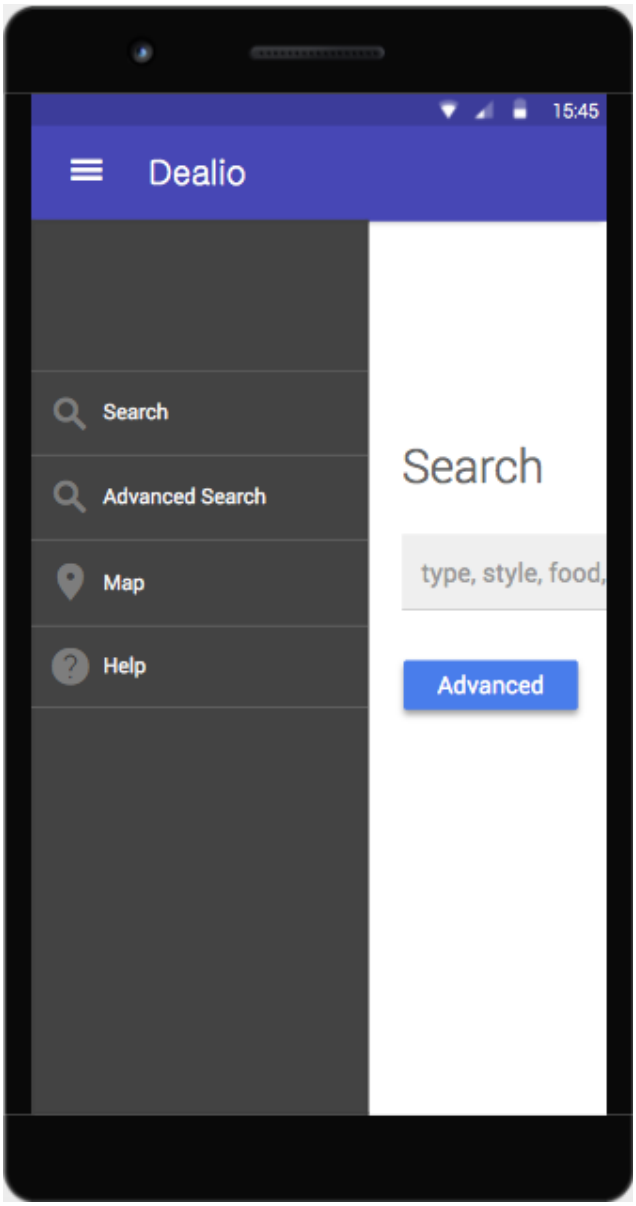

Figure 8. Side Menu Screen

This Side Menu screen depicts the side menu which is accessible via the “Triple Bar” in the top left of every screen except the Main Menu. This side menu functionality is standard within Android and contains buttons allowing the user to jump to a specific screen. This side menu also contains a button labelled “(?) Help” to enable to the “Help Text” overlay.

4.0 System Interaction

This section will go into details regarding the system interaction. The details of the system interaction will be portrayed through a database layout and interaction discussion and an entity-relationship diagram.

4.1 Entity-Relationship Diagram

The system has two main components, Restaurants, and Deals offered by those restaurants. The information will be stored in in the following entities. A Restaurant can have multiple deals, and deals may belong to multiple restaurants in the case of a chain of restaurants.

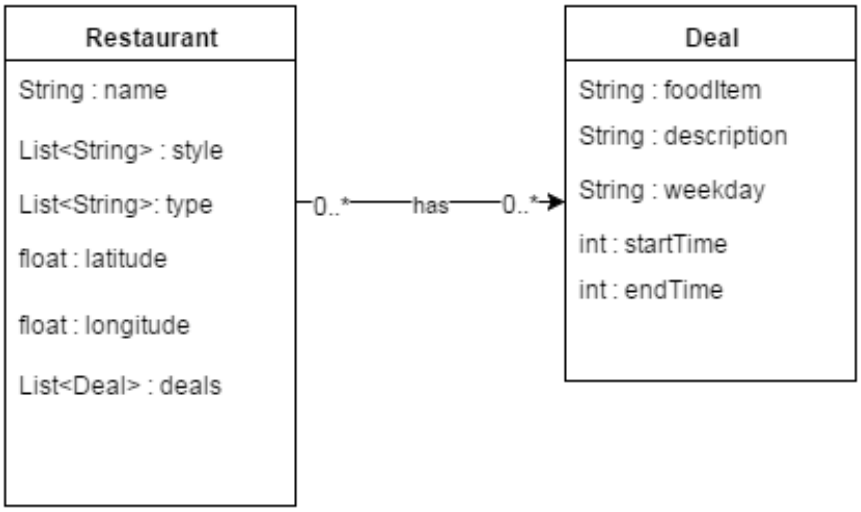

Figure 9: Entity-Relationship Diagram

4.2 Database Layout and Interaction

The layout of the database is represented in the entity-relationship diagram pictured above in section 4.1. When a user executes a search in the Dealio application, both “Restaurant” and “Deal” tables within the database are queried. A successful query will return both the deal and the associated restaurant. The results of this action will be displayed on the Search Results screen (section 3.5.5).

5.0 Glossary

UI	User Interface
Location Based Services	A service provided by smartphones that allows the phone to provide its location to applications to provide more contextual information to the user.
Google Maps	A web based mapping service developed by Google that can utilize location-based services to provide directions to a location.
Man-in-the-middle (MITM) Attack	An attack where the attacker secretly relays and possibly alters the communication between two parties who believe they are directly communicating with each other.
Web Scraping	The task of parsing the web for specific information, the task is often automated via scripts