Functional Specification and Management Plan for Dealio

February 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 specials. 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 special 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.)

The user isn't required to fill out all of the available search criteria to perform a search. Upon searching given specified criteria the application will list out the restaurants and their specials which meet the user's' criteria where they can then select one to see more details about the special.

The second method of interaction incorporates a mapping interface such as 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 special happening at the time of search will be highlighted. From here the user will be able to interact with the mapping interface and select a restaurant, showing more details on the current specials.

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

Revision History

Justyn Houle, Brandon Harvey, Sam Taylor, Graeme Turney	WetSoft Inc.
	Dealio
2016-02-16	Version 1.0

Version	**When**	**Who**	**What**
0.1	2016-02-04	Justyn Houle, Brandon Harvey, Sam Taylor, Graeme Turney	Initial drafting, organizational discussion.
1.0	2016-02-05 to 2016-02-16	Justyn Houle, Brandon Harvey, Sam Taylor, Graeme Turney	Finalized document, wrote sections 2 through 5.

Table of Contents

1. Introduction 1
    1.1. Purpose and Objectives 1
    1.2. Glossary 1
2. Functional Specifications 1
    2.1. Important Features 1
    2.2. Technical Requirements (Non-Functional) 2
        2.2.1. Performance 2
        2.2.2. Scalability 2
        2.2.3. Security 2
        2.2.4. Maintainability 2
        2.2.5. Usability 3
        2.2.6. Auditing and Logging 3
        2.2.7. Availability 3
        2.2.8. Required Hardware 3
3. User Interaction 3
    3.1. User Categories and Characteristics 3
    3.2. Use Cases 4
        3.2.1. View Map 4
        3.2.2. Search Details 4
        3.2.3. Search Location 5
    3.3. Interaction Dialogues 6
        3.3.1. Map Dialogue 6
        3.3.2. Search Dialogue 6
4. Management Plan 7
    4.1 Project Features 7
        4.1.1. Navigable Map 7
        4.1.2. Display Specials 7
        4.1.3. Search for Special 7
        4.1.4. Search Multiple Items 7
        4.1.5. Up to Date Specials 8
        4.1.6. Special and Restaurant Details 8
        4.1.7. Directions to Restaurant 8
        4.1.8. Invalid Deal Reporting 8
        4.1.9. Google Reviews Integration 8
        4.1.10. Help Overlay 9
    4.2. Minimal Deliverable System 9
    4.3. Possible Implementations 9
        4.3.1. Directions to Restaurant 9
        4.3.2. Invalid Deal Reporting 10
        4.3.3. Google Reviews Integration 10
        4.3.4. Help Overlay 10
    4.4. WetSoft Inc. 10
5. Conclusion 11

1. Introduction

Victoria is densely populated with restaurants and cafes of all sizes, often offering specials to entice customers. Unfortunately, locating meal specials 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 special 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 specials using a simple search functionality and map interface, which will allow the user to search for future deals based upon restaurant style, day, and meal time, and locate the restaurant swiftly without the need to switch to a different application.

1.1. Purpose and Objectives

Dealio will assist users in locating restaurant specials based upon specified search criteria and/or the location of the user and the time of day. The application is targeting all smartphone users who live in Victoria; specifically targeting tourists, university students, employees working downtown, and retirees.

1.2. 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.

2. Functional Specifications

This sections details the required functional specifications of Dealio. The application will be developed in the Android environment. A user will be able to view a real time location based map that displays restaurants and their specials, as well as search for restaurant specials based on the day of the week, time of day, style of restaurant, and type of food.

2.1 Important Features

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 special is currently on at the restaurant based on its icon on the map.

The second feature is a search function that allows a user to search for a special based on the day of the week, time of day, style of restaurant, and type of food. These fields will work as a multiple selectable drop down. A user can either scroll through the options to find what they are looking for, or type into the field to see if it appears.

2.2. Technical Requirements (Non-Functional)

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

2.2.1. 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 special 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.

2.2.2. Scalability

The application must be able to encompass a large and possibly ever growing customer base and restaurant specials. However, the scalability requirement is more so concerned with meeting the ability to display a multitude of restaurant specials 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 specials.

2.2.3. 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.

2.2.4. Maintainability

The application requires a database to host all the restaurant specials 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 implementation and design of the system doesn’t 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 intervention will be required.

2.2.5. 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.

2.2.6. 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.

2.2.7. 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 and shouldn't have any great affect on the customer's feelings and opinions towards the system.

2.2.8. 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.

3. User Interaction

This section describes the typical user of the application, as well as common use cases. Also included is a set of dialogues written to give contextual situations to aid in describing the application and its use cases.

3.1. User Categories and Characteristics

There are four intended demographics for the Dealio: tourists, students/young adults, downtown workers, and retirees. The tourist demographic describes the group of users that are unfamiliar with the restaurants in Victoria. The students/young adults demographic describes users that may be on a tight budget but are looking to go out with friends for a meal. Local employees are users that potentially work nearby restaurants and are able to take advantage of deals during lunch breaks or after work. Retirees is the demographic that describes users that have ample free time not restricted by work or school and are able to plan their meals around specific deals. These four demographics, however, are amalgamated into one mass user group simply defined as "User". This is because all demographics have the same permissions within the application. There is currently no administrator user type as administration of the application will be managed external to the actual application.

3.2. Use Cases

This section lists and details the use cases of the system. Each use case discusses a set of preconditions that must be met before the use case begins, the steps to take to complete the use case, and a discussion of possible but unexpected results.

3.2.1. View Map

ID	1
Description	A user views the nearby restaurants based on their location on a map.
Preconditions	● Location based services are turned on on the user's smartphone. ● User has internet connection ● Database has restaurants
Basic Steps	1. Open up the application 2. Press the "Map" button 3. User is presented with a map with the user's current location in the centre of the screen.
Exceptions	
Unexpected Results	● User has location based service 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 views the map.

3.2.2. Search Details

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 specific details of a deal.
Preconditions	● Restaurant specials are available. ● User has internet connection
Basic Steps	1. User opens application 2. The user then selects the 'search' option 3. The user fills out their desired search preferences using the dropdown filters presented 4. The user then presses the 'search' button displaying the available search results

	5. The user selects their preferred special 6. The user is then taken to the next screen detailing the special, the time of the special.
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 views the details of a special.

3.2.3. Search Location

ID	3
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	● Restaurant specials are available. ● User has internet connection
Basic Steps	1. The use case begins when the user opens the application. 2. The user then selects the 'search' option. 3. The user is then prompted with specific dropdown menus entailing type of food, time of day, location, style of food, etc. 4. The user fills out their desired search preferences. 5. The user then presses the 'search' button displaying the available search results. 6. The user selects their preferred restaurant. 7. The user is then taken to the next screen detailing the deals. User selects a deal and is shown details of the deal they have chosen. 8. User presses "See on map" button on the details page and the map is opened with the location of the restaurant with the chosen deal in the centre of the screen.
Exceptions	No search results after 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 there is no internet connection. This prompt will have a "ok" button which will redirect the user to **Step 4**.
Postconditions	The user views the location of the special on the map.

3.3 Interaction Dialogues

This section details two use case dialogues that showcase how a common user would utilize Dealio. The common user for these dialogues is a 3rd year university student named Patrick who often likes going out for food after class on Friday and on the weekend. Patrick owns an Android smartphone and uses it to text, email, browse social media and occasionally play mobile games.

3.3.1 Map Dialogue

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 specials 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 specials, including the time the special ends and what the special includes. He sees that chicken wings are on special for the next hour, so decides to walk the short 5 minutes to take advantage of the deal.

3.3.2 Search Dialogue

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 special 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 special 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 special on for the next 2 hours, so Patrick clicks on the "Map" button to see how to get there.

4. Management Plan

The following section details the management plan for the development of Dealio. This includes a discussion of the features of the application, an overview of possible implementations, and a description of the development team.

4.1. Project Features

This sections details the functional requirements of Dealio. Features are labeled low to high based on their priority of being completed for a minimum deliverable.

4.1.1. 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.
Priority	High

4.1.2. Display Specials

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

4.1.3. Search for Special

Description	The system must allow a user to search for a special based on day of the week, time of day, location, style of restaurant, and type of food.
Priority	High
Comments	A user does not need to have all of this information to complete a successful search.

4.1.4. Search Multiple Items

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

4.1.5. Up to Date Specials

Description	The system must provide a database that maintains up to date data on the specials provided by restaurants.
Priority	High
Comments	How this database is maintained is variable. Two potential options are data scraping or manual input.

4.1.6 Special and Restaurant Details

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

4.1.7. Directions to Restaurant

Description	The system must show directions to a selected restaurant. It must provide the option to select form of travel from a set of options.
Priority	Medium

4.1.8. Invalid Deal Reporting

Description	The system must allow a user to report a deal that a restaurant no longer provides so that it may be removed from the database.
Priority	Low
Comments	A developer or an application administrator would have to review the report before committing a change to the database.

4.1.9. Google Reviews Integration

Description	The system must provide a user with the ability to see the Google Review and Ratings for a selected restaurant.
Priority	Low

4.1.10. Help Overlay

Description	The system must provide a toggleable overlay that displays help information to the user about the various functionalities of the system.
Priority	Medium

4.2. Minimal Deliverable System

The following is a description of the minimal system WetSoft Inc. can guarantee to have produced by the end of the contract. This covers the requirements designated as high priority in section 5.1. This minimal deliverable system consists of two methods of interaction.

Firstly the user will be able to search for a restaurant special based upon their specified search criteria. This search criteria will include: the day of the week, the time of day, the type of food (chinese, italian, etc.), the style of eating (sit down, fast food, etc.), and the specific food item (wings, pizza, burgers, etc.). Users will be able to select multiple items in a search field, for example: burgers and wings. Users are not required to fill all search fields . A list of restaurants and their specials which apply to the user's search criteria will be displayed, allowing the user to click on a list item to see details.

Secondly the user will be able to interact with a mapping interface that is similar to or is Google Maps. From here, their current location will be displayed (should their location services be enabled) as well as an overlay displaying the surrounding restaurants. Restaurants which currently have a special on will be displayed with a different color on the mapping interface, differentiating them from restaurants which do not currently have a special on. The user can then select a restaurant from the map which will open a page on the details of the restaurant and the available special.

In regards to the database, a static, manually filled database will be created to store the information regarding restaurants and their specials. The database will only be set up this way for the minimum deliverable. Please refer to section 4.1.5 for a brief example of potential ways of maintaining the database.

4.3. Possible Implementations

Features with priority of medium or below in 4.1 are considered possible implementation that were either proposed as 'nice-to-haves' from our clients or considered heavily beneficial to the application by WetSoft Inc.

4.3.1. Directions to Restaurant

This feature relates to the real time navigable map feature. Given that a user has location services turn on on their smartphone, once a user has selected a restaurant they wish to eat at, the application will provide the option to map a route to the destination. The user would be able to receive the most efficient route based on their chosen form of travel (car, transit, bike, etc.).

4.3.2. Invalid Deal Reporting

If a user discovers that a restaurant no longer offers a deal, or that it has changed, the user should have an easy way of informing someone who can change the deal. This feature would allow a user to report to the administrators of the restaurant specials database that the special is no longer available.

4.3.3. Google Reviews Integration

A feature that would benefit users would be to integrate Google Reviews into the application. This feature would allow users to read and write reviews about a restaurant and their specials on the restaurant details page of the application. Reviews would provide a way for users to inform others of whether they feel a special is actually a special.

4.3.4. Help Overlay

This feature will provide assistance to users who are still becoming familiar with the interface. A user can opt to display a “Help Overlay” via a simple click of a button. Opting for the “Help Overlay” will display tooltips and small pieces of text which will appear upon clicking a particular element of the interface. From this help text, we wish to familiarize the user with the interface and answer any questions such as “what is this element's function?” or “how do I use this?”.

4.4. WetSoft Inc.

The WetSoft Inc. software development team is a new software company out of Victoria BC working on the development of Dealio. All four members are proficient in web application development with a focus on the MEAN Stack framework. The team also has a strong background in technical report writing and are familiar with the paradigms that revolve around determining the functional and nonfunctional requirements of software applications.

The WetSoft Inc. development team is made up of four members, CEO Justyn Houle and developers Brandon Harvey, Sam Taylor, and Graeme Turney. When Justyn is not busy managing Wetsoft Inc. his other duties include application analysis and toolkit development. Brandon and Sam are the current webmasters at WetSoft Inc., so when they aren’t working on making it easier for users to find meal specials they can often be found maintaining the company website. Sam is currently hosting the WetSoft Inc. website on his personal webspace through the UVic domain. He is responsible for making sure the website is up to date for due dates of deliverables. Graeme’s development roles are often put on hold to write technical documentation for WetSoft’s products. Overall, WetSoft Inc. is a well rounded team of individuals working towards the common goal of creating an application that is beneficial to food eaters around downtown Victoria.

5. Conclusion

Dealio will satisfy the need for consumers to know about daily specials happening in downtown Victoria. Its main functionalities include a filter search which allows users to specify certain traits of a deal and display all deals fitting the specified criteria, and a map which shows the current location of the user and available nearby deals. The target users include tourists, university students, downtown workers, retirees and anyone who enjoys a great deal. Initially, Dealio will only contain information for restaurants found in downtown Victoria, but if Dealio turns out to be favoured by users, expanding to different areas is a definite possibility. WetSoft Inc. is taking the initiative to develop the software necessary to build this application, and believes that it can provide a service that will keep customers coming back for more.