Life Mobile Application

FEATURE PROPOSAL: Location Based Services Module

Colin Man, Malina Jiang

**Version 1.0**

**Last Update:** 25 January 2015

Table of Contents

1 Document Control 3

1.1 Change Record 3

1.2 Definitions 3

2 Introduction 4

2.1 Scope 4

2.2 Functional Overview 4

2.2.1 Discovery 4

2.2.2 Maps 4

2.2.3 Lists 4

2.2.4 Safety 5

2.2.5 Scheduler 5

2.2.6 Entertainment 5

2.2.7 Tools 5

2.3 Assumptions/Constraints 5

2.4 Requirements 6

3 Component: Discovery 7

3.1 Navigation 7

3.2 Overview 7

3.3 Description 8

3.4 Feature: Goods and Services 9

3.5 Feature: Places 13

3.6 Feature: Travel 17

3.7 Feature: Food 19

3.8 Combination Features 22

3.9 Use Cases 23

# Document Control

## Change Record

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Date | Author | Changes |
| 1.0 | 01-25-15 | Colin Man | Draft following overview + description |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

## Definitions

|  |  |
| --- | --- |
| Term | Definition |
| LMA | Life Mobile Application |
| LBS | Location Based Services |
| POI | Points of Interest |
|  |  |
|  |  |

Note: “LMA” is consistently used throughout all documents as the name of the mobile application, although it is understood that another name will be chosen for branding purposes.

# Introduction

The goal of this module within the Life Mobile Application is to:

* Provide the user location based services for use in travel and transportation.
* Establish a framework for providing useful information to the user based on location without direct interaction or prompting from the user.
* Present a set of tools that allow the user to more efficiently manage their tasks and activities, during both travel and daily life.

## Scope

This phase of the location based services module will focus on establishing a list of possible features for the system. Subsequent phases may reduce, add, or change the features listed below before development is finalized.

## Functional Overview­

The features of LBS can be summarized in seven core components.

### Discovery

* Provides mostly background services that provide discovery of key features and landmarks nearby.
* Helps user discover points of interest and possible places, goods, or services that they would be interested in nearby.
* Can be customized by user to reflect interests, habits, and preferences.

### Maps

* Customization on top of Google Maps API that provides additional features such as user preference based routing.
* Enables additional integration with components such as “2.2.3 Lists” and “2.2.5 Scheduler” or other modules of LMA such as the “social” module.

### Lists

* Establishes four core lists for organization:
  + To-Do list
  + Packing List
  + Inventory List
  + Wish List
* Each core list embodies deep integration into the rest of the system, allowing for customizations and “smart” decisions to be made based on their content throughout LMA.
* Allows for creation of custom lists as well.

### Safety

* Creates a framework for determining the safety level of a particular location based on publicly available statistics.
* Provides tools for security during travel and transportation.
* “Smart” and responsive design shows tools in an easy-to-access location when necessary (though they can be accessed normally as well).

### Scheduler

* Provides an algorithm to facilitate efficient scheduling of tasks and activities.
* Integrates with “2.2.3 Lists” component in order to determine tasks that need to be scheduled.
* Can be customized and edited to reflect user preferences and criteria used to determine optimal scheduling.

### Entertainment

* Provides entertainment mostly in various forms of media such as music, videos, movies, etc.
* Integrates into the “Social” module of LMA in order to provide a seamless experience.

### Tools

* Useful utilities provided to the user that encompass a wide array of functionality, including:
  + Weather
  + Time
  + Alarm
  + Calendar
  + Messaging
* Integrates with native mobile device functionality to create a smooth LMA experience.
* Extensible platform that allows for third-party developers to create “extensions” with additional functionality.

## Assumptions/Constraints

* This assumes that LMA is run on a system that has a stable Internet connection.
* This assumes that LMA is run on a system that has a stable GPS connection and location data.
* This assumes that users have granted the necessary permissions required to provide services such as:
  + Push notifications
  + Location Based Services
  + Bluetooth LE connections

## Requirements

* The following mobile device operating systems should be supported:
  + iOS 7.0 and above
  + Android 4.1 (*Jelly Bean*) and above
  + Windows Phone 8 and above
* All development of application features should be native and should not be a web-based application in any sense. Components of LMA may be developed using web technologies, but the integration and infrastructure of the core components should be native.

# Component: Discovery

## Navigation

The navigation portion of all components in LBS should allow for easy switching between components as well as modules within LMA. As such, there must be links or buttons that allow for one or two-tap navigation to the following (there need not be a link to the currently running module, depending on the UX/UI design, though having one would most likely benefit the user experience):

* **Home** – The main screen of LMA. Allows the user to easily return to the starting point and re-launch any application or service that he chooses with the LMA launcher.
* **Discovery** – Links to the “Discovery” component of LBS. See “Component: Discovery” for more details.
* **Maps** – Links to the “Maps” component of LBS. See “Component: Maps” for more details.
* **Lists** – Links to the “Lists” component of LBS. See “Component: Lists” for more details.
* **Scheduler** – Links to the “Scheduler” component of LBS. See “Component: Scheduler” for more details.
* **Entertainment** – Links to the “Entertainment” component of LBS. See “Component: Entertainment” for more details.
* **Tools** – Links to the “Tools” component of LBS. See “Component: Tools” for more details.
* **Security** – This should only appear when the security index of the area is determined to pass a certain threshold. If it does pass the threshold, the user should be able to access this module with one tap, otherwise it need not be so accessible, but should still be available through the home screen.

## Overview

The “Discovery” component of LBS includes a multitude of features related to the discovery of points of interest along a travel route.

The categories of points of interest available for discovery can be summarized as follows:

* **Goods and Services** – This encompasses discovery of stores that sell goods or services that the user is looking for. Examples of this include:
  + Hot Deals
  + Coupons and Sales
  + Free things
  + Fuel and repair
  + Tools
  + Equipment
  + Transportation Services
  + Taskrabbits

Goods and services can also be cross referenced with items from the “Lists” component, which will allow for better integration with LBS and smarter suggestions for the user.

* **Places** – This includes specific locations that the user might find interesting, including:
  + Saved Spots
  + Possible parking spots and lots
  + Safe meeting places
  + Speed traps and common police patrolled areas
* **Travel** – This includes place that may be useful for tourists or travelers on vacation in a relatively unknown location:
  + Rest Stops
  + Bathrooms
  + Tourist spots
  + Lodging
* **Food** – This includes any food locations that the user may want to find based on their location. Examples include:
  + Restaurants and Reviews
  + Food Delivery

## Description

The “Discovery” component of the LBS module is responsible for helping users find the locations nearby that they would be interested in looking for and visiting as if they were perusing through the list themselves.

This intelligent search and discovery can be performed with a variety of artificial intelligence algorithms or machine learning. A detailed description of the potential features of the complete “Discovery” component of the LBS module can be found below.

## Feature: Goods and Services

The “Goods and Services” feature includes the discovery of all locations related to obtaining goods or services that the user may be interested in.

This discovery can be further classified into five categories as follows:

* **Coupons, Sales, and Free Things** – There are often free listings, blowout sales, or irresistible coupons in neighborhoods nearby. This feature helps consolidate these deals and allows the user to see the ones that are closest to him and access them immediately.

Prices should also be included when relevant so that the user can decide whether they would like to go to the location.

* + EX: 50% off sale at Macy’s 1.3 miles away
  + EX: Free computer monitor from Craigslist with meeting location 2.3 miles away
* **Fuel and Repair** – Often Location Based Services are used when the user is on the go or somewhere far from home. Thus, it is particularly important for the user to be able to access crucial resources such as fuel and repair services nearby.

Prices should also be included when relevant so that the user can decide whether they would like to go to the location.

* + EX: Shell 12.4 miles away with 89 at $3.12/gallon
  + EX: Kirk’s Auto Repair Shop 8.4 miles away
* **Tools and Equipment** – This differs from the “Repair” portion of the feature, since Tools and Equipment facilitates the discovery of locations that help the user fix their problems themselves. This should allow users to find nearby tools and equipment. Prices should be included when relevant so that the user can decide whether they would like to go to the location.
  + EX: ACE hardware is located 3.4 miles away
  + EX: Home Depot is located 1.3 miles away
* **Transportation Services** – Local transportation services vary widely by area and it is immensely useful to have the top local companies accessible to the user. This should provide a list of the cheapest taxi companies as well as alternative public transportation options by price and convenience.
  + EX: BART is available 2.5 miles away for an average of $2.4 per station
  + EX: Uber is available (not currently surging) for $8/hour
* **TaskRabbits** – TaskRabbit is a company that specialized in allowing people to be hired to perform certain tasks. This can be incorporated into LBS by allowing the detection of local willing “TaskRabbits”. This can be organized using the same features as the TaskRabbit company does.
  + EX: Chad is willing to perform *transportation* tasks and is 2.3 miles away
  + EX: Jenny is willing to perform *delivery* tasks and is 4.3 miles away

Note: If the LMA is successful and gains enough users, we may be able to implement this feature solely with the LMA and not rely on a third-party company such as TaskRabbit to execute the functionality summarized in this portion of the feature.

Additional elements of the “Goods and Services” feature are as follows:

* **“Maps” integration** – This feature should also be incorporated with the “Maps” Component in order to be able to access navigation tools directly and streamline the process of transportation to the actual location.
* **“Lists” integration** – The “Goods and Services” feature should also cross reference with the “Lists” component in order to determine the goods and services that the user would like. It should look at the “Inventory” as well as “Wish” list in order to determine the preferences of the user.

Some qualities that may be used to determine the goods and services that the user may like based on lists include the following:

* + Past purchases
    - EX: Purchase of “Atlas Shrugged” two weeks ago may suggest an interest in Ayn Rand or books.
  + Categories of goods found in “wish” list
    - EX: The “wish” list may consist mostly of electronics, indicating an interest in electronic goods and computers.
  + Quantity of goods purchase
    - EX: Past purchases may indicate a history of bulk purchase, indicating that the user may be interested in buying mass quantities (wholesale location such as Costco or Sam’s Club may be of more interest to the user).
  + Date of purchase
    - EX: The recency of the purchases affects how relevant the information is. If a purchase of “Dove Body Wash” was executed two years ago, it most likely provides less information than a purchase of “Axe body Wash” two days ago.
  + Date added to lists
    - EX: The recency of the addition to lists affects how relevant the information is. If an item “Oculus Rift” was added to the “Wish” list two days ago, it is more likely that the user still wants the product or that he has not purchased it yet.

Data analysis algorithms can be developed to compare the lists of goods and services available nearby with those that the user would like and generate an index based on projected preference. This index can then be used to rank candidates as well as to generate the list of local options (using a threshold).

* **Notifications** – Push notifications should be sent to the user when the preference index generated exceeds a certain threshold set by the user.
* **Export** – The user should have the option of exporting his user data either to the cloud or to a local file that contains a backup of all the data used to calculate his preferences. This may be incorporated with the “Data” module as well.
* **Sync** – The user should have the option of syncing his user data (used to calculate his preferences) either to the cloud or to other devices that may be running LMA. This may be incorporated with the “Data” module as well.

In addition to the automatically generated elements that are found above, the user should also have the option of customizing the results that he obtains. He may interact with the “Goods and Services” feature through one of the following ways:

* **Preferences –** choosing the categories of goods and services that he would like to know about. The preference pane should include various criteria for discovering goods and services such as:
  + Category of good
    - EX: Clothing
    - EX: Electronics (Computers, Mobile Devices)
  + Price range of good
    - EX: $25-70
  + Locations accepted or locations that should not be shown
    - EX: Don’t show: RadioShack, Best Buy
  + Goods similar to certain preset goods
    - EX: Similar to: iPhone

Additional settings are as follows:

* + Radius – only goods and services within this radius should be shown. Any goods and services that are not within this radius will be shown on the map (in the “Maps” module), but will not generate push notifications. These items will also not be shown in the “Browse” screen of the “Goods and Services” feature.
  + Preference Index Threshold - the user should be able to set the preference index threshold for push notifications here. Any item found to exceed this threshold should generate a push notification if it is within the radius.
* S**earch –** the user should be allowed to search for certain goods in the area (similar to performing a Yelp search for restaurants). The search should contain criteria such as:
  + Name of the good or service
    - EX: Car Wash
  + Quantity of the good or service
    - EX: 5 dozen bananas
  + Category of good
    - EX: Clothing
    - EX: Electronics (Computers, Mobile Devices)
  + Price range of good
    - EX: $25-70
  + Locations accepted or locations that should not be shown
    - EX: Don’t show: RadioShack, Best Buy
  + Goods similar to certain preset goods
    - EX: Similar to: iPhone
* **Reference Point Change** – The user should be able to change their reference point if they so desire. This is useful for planning since it allows the user to search for points of interest without actually having to travel to the location.
  + **EX:** The user can search for points of interest near San Francisco even though his current location is actually in Palo Alto.
* **Browse** – this is one of the main elements of the “Goods and Services” feature. It allows the user to view the goods and services that are nearby with only one screen, in a comprehensive list that includes all the details specified above for each item of interest. The user should be able to filter results based on the following criteria:
  + Quantity of the good or service
    - EX: 5 bananas
  + Category of good
    - EX: Clothing
    - EX: Electronics (Computers, Mobile Devices)
  + Price range of good
    - EX: $25-70
  + Locations accepted or locations that should not be shown
    - EX: Don’t show: RadioShack, Best Buy
  + Goods similar to certain preset goods
    - EX: Similar to: iPhone

## Feature: Places

The “Places” feature of the “Discovery” component includes all places that the user may find interesting or useful. Some of these places are generated by the app based on convenience and common use, and some are generated based on user input and previously saved data.

The places that the user may be interested in can be summarized in four categories:

* **Saved Spots** – Saved spots are locations that the user decides to mark in his map as a location that he believes has importance. These can be further categorized into the following default classes:
  + Car location – Losing a car in the parking lot is a common occurrence and this should allow the user to save the location of his car in order to find it later. A more advanced implementation may include automatic detection of parking spots so that the user does not have to manually track the locations that he has parked his car (the phone will be able to detect with GPS technology that the motion has stopped and the car is most likely parked).
  + Bike location – Bikes are often lost amongst arrays of bike racks among the city. This default should allow the user to save his bike location’s GPS coordinates. As with the car location, a more advanced implementation may include automatic detection of bike parking spots so that the user does not have to manually track the locations that he has locked his bike (the phone will be able to detect with GPS technology that the motion has stopped and the bike is most likely locked). This may pose a larger challenge since a person can walk his bike but he cannot walk his car.
  + Favorites – Users should be able to mark their favorite locations and store them as locations in “Saved Spots”. In addition to marking favorites, the user should have the option of marking them with certain tags, including “Restaurants”, “Friends”, “Buildings”, “Stores”, etc.
* **Parking Locations** – Parking locations are often hard to find in a city or in a place that the user is not familiar with. To create more of a convenience for the user, parking locations should be marked and shown. Parking locations include, but are not limited to: parking structures, parking lots, street parking. Paid parking should be listed as well and should include the price if it exists.
  + EX**:** Street parking is available .2 miles away for $2/hour
  + EX**:** There is a parking lot .1 miles away for free
* **Safe Meeting Locations** – Many online services are created on a model that requires users to meet up in person in order to make the most of their experience. Such services include: Craigslist, Tinder, okCupid, etc. However, many crimes have occurred due to the dangerous nature of meeting a stranger. The “Safe Meeting Locations” feature will allow the user to find safe locations nearby to meet up with. Possible features that define such locations include:
  + Surrounding Area – using the safety index from the “Safety” component, we can determine whether the area is a good area to meet a person you don't know.
    - This index takes into account various features, including crime rate, housing prices, etc. See “Safety” component for details.
  + Public – with a large flow of people, there are more witnesses and the location is safer

These locations tend to be public and are not conducive to crime.

* + EX: McDonald’s in a busy mall is a good location to meet. It is safe, public, and not conducive to crime.
* **Speed Traps** – There are many locations that are known to be speed traps and it is very easy to fall into the speed trap and accidentally drive at higher than the speed limit if you are not careful – these locations will be marked and notifications will be sent to the user if he is close to one or about to enter one. He will be reminded of the speed limit, how it changes, and prompted to slow down if he still decides to go a certain amount above the speed limit (default 10% since that is the margin of error for conventional radar speed detection guns).
* **Policeman Frequent Locations** – Policemen have particular locations that they prefer to catch speeding cars. As with “Speed Traps”, the “Policeman Frequent Locations” feature allows for the user to be warned when he is close to a policeman frequent location and these locations will be marked. He will be reminded of the speed limit and prompted to slow down if he still decides to go a certain amount above the speed limit (default 10% since that is the margin of error for conventional radar speed detection guns).

Additional elements of the “Places” feature are as follows:

* **“Maps” integration** – This feature should also be incorporated with the “Maps” Component in order to be able to access navigation tools directly and streamline the process of transportation to the actual location.
* **“Lists” integration** – Favorites should be stored in lists or correlated with lists (in the form of suggestions or data management) in order to provide a more seamless experience with the rest of the components and modules.
* **Notifications** – Push notifications should be sent to the user when one of the locations that the user wishes to be notified about gets close enough the distance is less than a certain threshold set by the user.
* **Export** – The user should have the option of exporting his user data either to the cloud or to a local file that contains a backup of all the preference data. This may be incorporated with the “Data” module as well.
* **Sync** – The user should have the option of syncing his user data either to the cloud or to other devices that may be running LMA. This may be incorporated with the “Data” module as well.

In addition to the automatically generated elements that are found above, the user should also have the option of customizing the results that he obtains. He may interact with the “Places” feature through one of the following ways:

* **Preferences –** choosing the kinds of locations he would like to know about. The preference pane should include the option to show or hide the various options that are discussed above.

Additional settings are as follows:

* + Radius – only locations within this radius should be shown. Any places that are not within this radius will be shown on the map (in the “Maps” module), but will not generate push notifications. These items will also not be shown in the “Browse” screen of the “Places” feature.
* S**earch –** the user should be allowed to search for certain locations in the area (similar to performing a Yelp search for restaurants). The search should contain criteria such as:
  + Type of location
    - EX: Speed Trap
  + Locations accepted or locations that should not be shown
    - EX: Don’t show: RadioShack, Best Buy
  + Hours
    - EX: Open 3pm-10pm; open now
* **Reference Point Change** – The user should be able to change their reference point if they so desire. This is useful for planning since it allows the user to search for points of interest without actually having to travel to the location.
  + EX**:** The user can search for points of interest near San Francisco even though his current location is actually in Palo Alto.
* **Browse** – this is an important view of the elements of the “Places” feature. It allows the user to view the locations that are nearby with only one screen, in a comprehensive list that includes all the details specified above for each item of interest.

## Feature: Travel

One of the main uses of LMA is to help travel and provide a convenient way for tourists and travelers to find the things that they are looking for while on a trip or vacation (as opposed to hours of reading the tourist guide). In terms of LBS, the main purpose would be to help them discover places that are worth going to or places that would be useful on vacation.

Some potential locations for travel include:

* **Rest Stops** – The number one sought out location, especially on road trips, is a rest stop. Continuous driving is much easier said than done, so often drivers have to switch off for safety. This feature provides an indication of rest stops nearby in order to allow more efficient timing of switching off and exiting the highway within the rotating group of drivers.
  + EX**:** After driving on I5 for 6 hours you are tired and you want to switch off with another driver. You can now look up the next rest stop to figure out when you should exit the highway and about how long you should expect to have to continue driving.
* **Bathrooms** – Many places that offer restrooms, especially in large, populated, cities or areas, ask that customers make a purchase before using the restroom. Their bathrooms are not public and may be protected by a key, etc. The “Bathrooms” feature should allow travelers to discover bathrooms nearby as well as determine the properties of the restroom.
  + EX: There is a publicly available restroom in .3 miles at McDonald’s
  + EX: There is a restroom that can be accessed with purchase at Golden Gate Bakery .1 miles away.
* **Tourist Locations** – In any location there are places of interest to the tourist or casual traveler. The “Tourist Locations” portion of LBS will allow the user to discover such locations and find the nearest tourist locations that may interest him.
  + EX: The Empire State Building is .1 miles away from your current location.
* **Lodging** – One of the core components of travel is lodging – whether on extended vacation or just a short trip, everyone needs a place to stay while not at home. The “Lodging” portion of the LBS helps users discover lodging based on certain criteria:
  + Reviews – reviews can be extracted from locally used application such as Yelp
  + Price – the user has the option of only showing lodging options that meet the user’s budget
  + Rating – the user has the option of only showing the lodging otions that meet the user’s rating standards

This portion of LMA may also be incorporated with third-party companies such as AirBnb in order to create an even better experience for the user (since more housing options makes the application better and easier to use).

Additional elements of the “Travel” feature are as follows:

* **“Maps” integration** – This feature should also be incorporated with the “Maps” Component in order to be able to access navigation tools directly and streamline the process of transportation to the actual location.

* **Notifications** – Push notifications should be sent to the user when one of the kinds of locations that the user wishes to be notified about gets close enough the distance is less than a certain threshold set by the user.

In addition to the automatically generated elements that are found above, the user should also have the option of customizing the results that he obtains. He may interact with the “Travel” feature through one of the following ways:

* **Preferences –** choosing the kinds of travel locations he would like to know about. The preference pane should include the option to show or hide the various options that are discussed above.

Additional settings are as follows:

* + Radius – only locations within this radius should be shown. Any travel locations that are not within this radius will be shown on the map (in the “Maps” module), but will not generate push notifications. These items will also not be shown in the “Browse” screen of the “Travel” feature.
* S**earch –** the user should be allowed to search for certain locations in the area (similar to performing a Yelp search for restaurants). The search should contain criteria such as:
  + Type of location
    - EX: Bathrooms
  + Locations accepted or locations that should not be shown
    - EX: Don’t show: McDonald’s
  + Hours
    - EX: Open 3pm-10pm; open now
* **Reference Point Change** – The user should be able to change their reference point if they so desire. This is useful for planning since it allows the user to search for points of interest without actually having to travel to the location.
  + **EX:** The user can search for points of interest near San Francisco even though his current location is actually in Palo Alto.
* **Browse** – this is an important view of the elements of the “Travel” feature. It allows the user to view the locations that are nearby with only one screen, in a comprehensive list that includes all the details specified above for each item of interest.

## Feature: Food

Food is one of the main reasons people use their phones to look up any sort of navigation or maps. Since most people eat three times a day, the need to find good food is often and recurring. The “Food” feature of LBS is much like Yelp in that it helps users locate restaurants and places to eat (and in terms of implementation, may actually use Yelp or build off of Yelp itself), but has a few key added features.

These added features are as follows:

* **Restaurants On the Way** – If the user is already on the way to a certain location (which is often the case when people are looking for restaurants), the “Food” feature of LBS should allow the user to search for places to eat along the route. Yelp currently does not offer this feature since it does not have the full integration into Maps that LMA will have and thus cannot provide the seamless end-to-end support that we can.
  + EX: A person currently in the process of driving two hours to visit his son in college may be able to use this feature to look up a good restaurant on his way and not have to settle for McDonald’s or Burger King.

Criteria for discovering restaurants should be as follows:

* + City – the city of the restaurant
    - EX: Cupertino
  + Distance – the distance of the restaurant from the current location
    - EX: 4 miles
  + Price – the price range of a typical entrée
    - EX: $50 - $100
  + Deals – any deals that might be running and still apply
    - 50% off on Tuesdays (today)
  + Hours – hours of operation
    - Open 2PM – 11PM (open now)
  + Payment Accepted – the payment methods accepted by the restaurant
    - Accepts all credit cards and cash
  + Food type – the kind of food that the restaurant serves
    - Italian Japanese Fusion

The user should be given the option to use any combination of the above criteria to filter his results and determine what restaurants nearby are offered to him as options.

* **Delivery** – All restaurants that deliver to the current location of the user should be listed. This functionality is similar to sites like Eat24 (we may actually use their platform in our implementation), but we are able to encompass a larger range of locations as well as incorporate it with the eat-in restaurant discovery functionality.

Criteria for discovering delivery should be as follows:

* + Time – the time required for the delivery to be completed – depends on the distance and guarantees of the restaurant
    - EX: 35 minute estimated delivery time
  + Price – the price range of a typical entrée
    - EX: $50 - $100
  + Deals – any deals that might be running and still apply
    - 50% off on Tuesdays (today)
  + Hours – hours of operation
    - Open 2PM – 11PM (open now)
  + Payment Accepted – the payment methods accepted by the restaurant
    - Accepts all credit cards and cash
  + Food type – the kind of food that the restaurant serves
    - Italian Japanese Fusion

The user should be given the option to use any combination of the above criteria to filter his results and determine what delivery options nearby are offered to him as options.

Additional elements of the “Food” feature are as follows:

* **“Maps” integration** – This feature should also be incorporated with the “Maps” Component in order to be able to access navigation tools directly and streamline the process of transportation to the actual location.

* **Notifications** – Push notifications should be sent to the user when a restaurant or delivery service that exceeds his preference index threshold set by the user comes within the radius of detection (also set by the user).

Data analysis algorithms can be developed to compare the lists of restaurants and delivery services available nearby with those that the user would like (based on “lists” component, past history, etc.) and generate an index based on projected preference. This index can then be used to rank candidates as well as to generate the list of local options (using a threshold).

In addition to the automatically generated elements that are found above, the user should also have the option of customizing the results that he obtains. He may interact with the “Food” feature through one of the following ways:

* **Preferences –** choosing the kinds of restaurants or delivery services he would like to know about. The preference pane should include the option to show or hide the various options that are discussed above.

Additional settings are as follows:

* + Radius – only locations within this radius should be shown. Any restaurant or delivery services that are not within this radius will be shown on the map (in the “Maps” module), but will not generate push notifications. These items will also not be shown in the “Browse” screen of the “Food” feature.
  + Preference Index Threshold - the user should be able to set the preference index threshold for push notifications here. Any item found to exceed this threshold should generate a push notification if it is within the radius.
* S**earch –** the user should be allowed to search for certain locations in the area. This procedure is described under both “Restaurants” and “Delivery Services” as their criteria are distinct.
* **Reference Point Change** – The user should be able to change their reference point if they so desire. This is useful for planning since it allows the user to search for points of interest without actually having to travel to the location.
  + **EX:** The user can search for points of interest near San Francisco even though his current location is actually in Palo Alto.
* **Browse** – this is an important view of the elements of the “Food” feature. It allows the user to view the locations that are nearby with only one screen, in a comprehensive list that includes all the details specified above for each item of interest.

## Combination Features

In addition to listing all the features separately and finding locations separately as listed in the description for each of the individual features themselves, the user should also have the option of using a composite view that combines all of the “browse” screens specified in the features and allows for the viewing of all relevant locations at the same time.

In addition, there should be an easy way to modify preferences of multiple modules at the same time, including the radius of discovery, preference index threshold, etc.

Note: a more detailed guideline on the division of functionality among screens will be drafted in a functional specification after the features in this proposal are finalized.

## Use Cases

### Figure 3.9.1

Esntaernst eianrst nariet aiersnt aeirnstairens t Esntaernst eianrst nariet aiersnt aeirnstairens t Esntaernst eianrst nariet aiersnt aeirnstairens t Esntaernst eianrst nariet aiersnt aeirnstairens t Esntaernst eianrst nariet aiersnt aeirnstairens t Esntaernst eianrst nariet aiersnt aeirnstairens t Esntaernst eianrst nariet aiersnt aeirnstairens t Esntaernst eianrst nariet aiersnt aeirnstairens t Esntaernst eianrst nariet aiersnt aeirnstairens t Esntaernst eianrst nariet aiersnt aeirnstairens t Esntaernst eianrst nariet aiersnt aeirnstairens t Esntaernst eianrst nariet aiersnt aeirnstairens t Esntaernst eianrst nariet aiersnt aeirnstairens t Esntaernst eianrst nariet aiersnt aeirnstairens t Esntaernst eianrst nariet aiersnt aeirnstairens t Esntaernst eianrst nariet aiersnt aeirnstairens t Esntaernst eianrst nariet aiersnt aeirnstairens t Esntaernst eianrst nariet aiersnt aeirnstairens t

### Figure 3.9.2

Esntaernst eianrst nariet aiersnt aeirnstairens t Esntaernst eianrst nariet aiersnt aeirnstairens t Esntaernst eianrst nariet aiersnt aeirnstairens t Esntaernst eianrst nariet aiersnt aeirnstairens t Esntaernst eianrst nariet aiersnt aeirnstairens t Esntaernst eianrst nariet aiersnt aeirnstairens t Esntaernst eianrst nariet aiersnt aeirnstairens t Esntaernst eianrst nariet aiersnt aeirnstairens t Esntaernst eianrst nariet aiersnt aeirnstairens t Esntaernst eianrst nariet aiersnt aeirnstairens t Esntaernst eianrst nariet aiersnt aeirnstairens t Esntaernst eianrst nariet aiersnt aeirnstairens t Esntaernst eianrst nariet aiersnt aeirnstairens t Esntaernst eianrst nariet aiersnt aeirnstairens t Esntaernst eianrst nariet aiersnt aeirnstairens t Esntaernst eianrst nariet aiersnt aeirnstairens t Esntaernst eianrst nariet aiersnt aeirnstairens t Esntaernst eianrst nariet aiersnt aeirnstairens t

Esntaernst eianrst nariet aiersnt aeirnstairens t Esntaernst eianrst nariet aiersnt aeirnstairens t Esntaernst eianrst nariet aiersnt aeirnstairens t Esntaernst eianrst nariet aiersnt aeirnstairens t Esntaernst eianrst nariet aiersnt aeirnstairens t Esntaernst eianrst nariet aiersnt aeirnstairens t Esntaernst eianrst nariet aiersnt aeirnstairens t Esntaernst eianrst nariet aiersnt aeirnstairens t Esntaernst eianrst nariet aiersnt aeirnstairens t Esntaernst eianrst nariet aiersnt aeirnstairens t Esntaernst eianrst nariet aiersnt aeirnstairens t Esntaernst eianrst nariet aiersnt aeirnstairens t Esntaernst eianrst nariet aiersnt aeirnstairens t Esntaernst eianrst nariet aiersnt aeirnstairens t Esntaernst eianrst nariet aiersnt aeirnstairens t Esntaernst eianrst nariet aiersnt aeirnstairens t Esntaernst eianrst nariet aiersnt aeirnstairens t Esntaernst eianrst nariet aiersnt aeirnstairens t