USING EMERGING TECHNOLOGIES TO FIND A GEOGRAPHIC PARKING LOCATION IN A DYNAMIC ENVIRONMENT

By

PRAJWALA KALAKUNTLA

Bachelor of Technology in Computer Science

Jawaharlal Nehru Technological University

Hyderabad, AP, India

2012

Submitted to the Faculty of the Graduate College of the
Oklahoma State University in partial fulfillment of the
requirements for the Degree of
MASTER OF SCIENCE
December, 2013

USING EMERGING TECHNOLOGIES TO FIND A GEOGRAPHIC PARKING LOCATION IN A DYNAMIC ENVIRONMENT

Thesis Approved:

Dr. Eric Chan-Tin

Thesis Adviser

Dr. K M George

Dr. Nophill Park

ACKNOWLEDGEMENT

I dedicate my thesis work to my parents, Kalakuntla Satyanarayana Rao and Kalakuntla Rupa Devi, my sister, Kalakuntla Shravani and my brother-in-law Madhavaneni Srikar Rao for their support, love and encouragement throughout my life.

I am grateful to Dr. Eric Chan-Tin for providing a chance to contribute to his research. I appreciate his encouragement and support throughout the period of my MS. I also thank Dr. K M George and Dr. Nophill Park for their advices in completing my thesis work successfully.

Name: PRAJWALA KALAKUNTLA

Date of Degree: DECEMBER, 2013

Title of Study: USING EMERGING TECHNOLOGIES TO FIND A GEOGRAPHIC PARKING LOCATION IN A DYNAMIC ENVIRONMENT

Major Field: COMPUTER SCIENCE

Abstract: GPS (Global Positioning System) has now-a-days become very popular because of the ease it provides to find the best and most easiest way to the destination of the user. The most widely used applications based on GPS that are available now are the map based navigation applications and the applications that help in tracking vehicles. This research explores ways to combine GPS technology and smart phone technology to solve the location identification problem. An instance of this problem is finding the location of a vacant parking space by a moving vehicle. Finding the information about the vacancy of a parking lot is an important issue because it involves the continuous monitoring of the parking lot in order to ensure the accuracy of the system. The user must be updated on a regular basis so that the user may not face any difficulty when he tries to find a vacant parking space. The location coordinates retrieved by the inbuilt GPS of the smart phone are based on the accuracy level set. Navigation applications always need the most accurate location coordinates for giving exact results.

This thesis develops an iOS application based on the location co-ordinate values retrieved by the inbuilt GPS in order to find a vacant parking space in a very large parking lot. The thesis also tries to stabilize the location coordinates which may change depending on the weather conditions and wind direction in that particular area. The thesis does not involve any usage of sensors but only makes use of the location coordinates. The accuracy of the location coordinates is very important to ensure accurate updated information about the parking spots that are vacant when delivering the information to the user. The underlying idea is the availability of vacant parking spaces using signals sent by the smart phone and its inbuilt GPS.

TABLE OF CONTENTS

LIST OF FIGURES

Figure Page

CHAPTER I

INTRODUCTION

1.1 Overview

Parking a car in a large parking lot is the most common challenge faced by millions of people these days. The increase in the number of vehicles is tremendous. So the occupancy of the parking lot needs to be increased when it is a crowded place. Therefore finding a vacant parking space in a large parking lot with so many cars is a tough task now-a-days.

GPS has become a very important feature of any smart phone today. The navigation is the most important application where GPS is widely used. There are few other useful applications that use GPS like Navigon USA, Gokivo, MapQuest and so on. These applications are mainly for navigation.

Almost everyone has a smart phone now-a-days. Every smart phone has an inbuilt GPS. The thesis makes use of the inbuilt GPS to help users in finding a vacant parking space. When the user is in a large city, he finds it difficult to find a vacant parking space in a large parking lot. If the user wants to find a vacant parking space manually, he has to move around in the parking lot regardless of how large the parking lot is in order to find a vacant parking space to park his vehicle. This involves a lot of time as the user needs to search the whole parking lot in order to find a single vacant spot and there is a waste of fuel to find a vacant parking spot.

1.2 Organization of Thesis

The thesis is to develop an iPhone application that can easily find a vacant parking space for the user in a large parking lot. The thesis is based on the concept of Resource Allocation Graphs (RAGs) for the development. There are few challenges involved in the development of the application like the accuracy of the location coordinates, as the values change depending on the wind direction, height of the buildings around and so on. The other challenge is the timeliness of the updates to the database. The database needs to be updated very often in order to information correct all the time. The application is to be developed using Xcode, the integrated development environment for the Mac to use the iOS platform.

Basic Assumptions

The basic assumptions to be made for the design and development of this iPhone application are that, each and every user needs to have the application in order to find the parking space and park, along with the timely updates to the database. If every user does not have the application, only the user with the application can update the database when he parks his vehicle. If any other user without the application has parked his vehicle in a parking spot, that will not be updated to the database and so the busy spot might be shown as a vacant spot to some other user using the application. So, there is a need for every user to have the application installed on their smart phones.

The timely updates to the database is an important issue, as that helps in maintaining the accuracy of the application in showing the user the exact vacancy status of the parking lot by saving the user's time and fuel than searching a vacant spot manually.

CHAPTER II

REVIEW OF RELATED LITERATURE

This thesis is concerned with the development of an iPhone application which is used to find a vacant parking spot in a very large parking lot as an instance of the general location identification problem. There are few applications with the similar functionality but they have used sensors to find the information about the vacant parking spaces of the parking lot.[8]

2.1 Parker - An application by *Streetline*

There is an existing application developed by Streetline called Parker[6]. This application is used to find a vacant parking space along with the availability and pricing information. Smart-parking Technology by Streetline makes use of many *ultra-low power wireless sensors* in the form of a network in every parking lot and tries to determine the vacancy. This data is later transmitted back to the smart phone with the information in the form of blocks with nearest parking and most available spaces.

The interesting part of the application is that the drivers can easily get the information about the pricing, the on time limits and also about the payment mode of the meters like credit cards or coins. This application makes use of wireless sensors of high cost and those sensors need to be arranged in every parking lot available to get the information of every parking lot in the city.

Arranging these ultra-low wireless sensors all over the country is very costly and all the sensors need to be designed to communicate with the Streetline Head Quarters to access the smart-parking technology.

Fig 1: Parker - An existing application to find a parking space.

This thesis application makes use of the inbuilt GPS of the smart phone. There are no sensors needed for the development of the application, instead it is solely based on the location coordinate values retrieved by the GPS. The location coordinates retrieved by the GPS are used to update the database about the vacant and busy spots in a parking lot.

2.2 Comparison to a Resource Allocation Graph

The goal of the thesis is to allocate a vacant parking space to the user easily as soon as he enters the parking lot. The correctness of the application is another important issue. A single parking spot cannot be assigned to two or more users and also a vacant spot cannot be shown as busy spot because of a late update or update failure. This is analogous to Resource Allocation Graph in Operating Systems.

2.2.1 Resource Allocation Graph [7]

A Resource Allocation Graph is a graph which consists of both resources and processes in a system. Resource Allocation Graph is mainly used to know the existence of deadlock and also gives the information about the resources and also the processes of the system that are involved in the deadlock.

The edge pointing from a process to a resource is a *request edge* which indicates that the process requests the resource. The edge pointing from a resource to a process is an *assignment edge* which indicates that the resource has been assigned to that particular process.

The Resource Allocation Graph is a set of vertices and edges. The set of vertices is further divided as two: set of *Processes* and set of *Resources*.

The set of processes is

$$P_i = \{P_1, P_2, P_3, ...\}.$$

The set of resources is

$$R_i = \{ R_1, R_2, R_3, ...\}.$$

The request edge is denoted as

$$P_i \longrightarrow R_i.$$

The assignment edge is denoted as

$$R_i \longrightarrow P_i.$$

The edge set is obtained by the type of edge associated with a process and a resource from the set of process and resource vertices respectively. The edge set is represented by E.

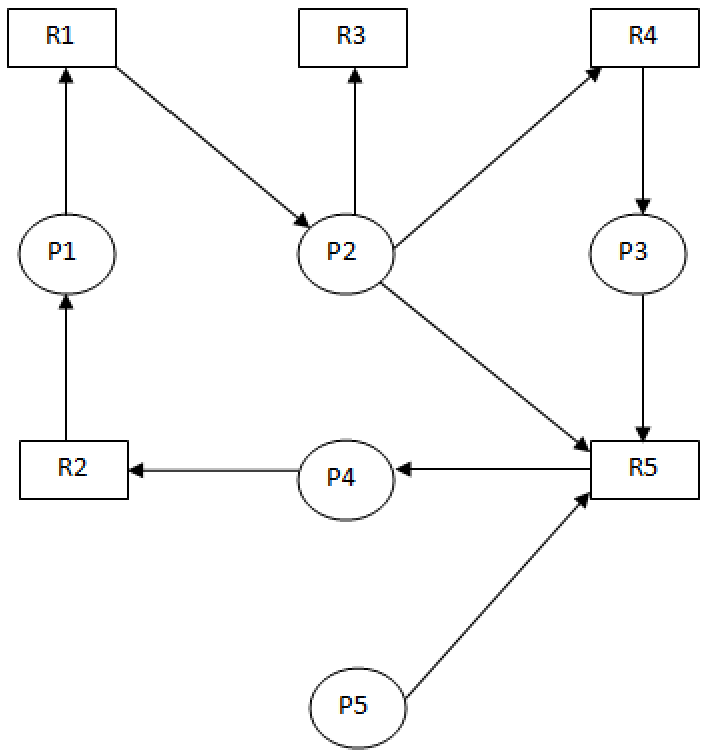


Fig 2: Sample Resource Allocation Graph

The set of vertices for the above graph:

Pi = {P1, P2, P3, P4, P5}

Ri = {R1, R2, R3, R4, R5}

The edge set of the above sample resource allocation graph is:

$E = \{ P1 \rightarrow R1, P2 \rightarrow R3, P2 \rightarrow R4, P2 \rightarrow R5, P3 \rightarrow R5, P4 \rightarrow R2, P5 \rightarrow R5,$

$R1 \rightarrow P2, R2 \rightarrow P1, R4 \rightarrow P3, R5 \rightarrow P4\}$

2.2.2 Comparison

The thesis involves the allocation of vacant parking spot to the user with the help of a mobile application. Comparing the application to the Resource Allocation Graph (RAG), every car is considered as a node i.e., Process set of vertices and the parking spots are the resource set of vertices. The edges between the vertices are the allocation edges which indicates the allocation of a particular parking spot to a particular car.

A cycle in the resource allocation graph usually indicates a deadlock situation with hold-wait conditions. The deadlock situation in this thesis refers to the failure in updating the database which may result to more than one users trying to park the car in a single parking spot thinking it is vacant or the application showing a busy spot as a vacant spot or so.

The update operation on the database is one of the very important step in the thesis. The allocation of vacant parking spots can be decided based on the database values or updates. The resources (parking spots) must be vacant or free in order to get allocated for the processes (cars).

CHAPTER III

PROBLEM STATEMENT

This thesis is about developing an iOS application based on the location coordinates obtained from the GPS in the iPhone. Whenever a user wants to park his car in a parking lot, he will just need to check with the application for the available parking space. If a user parks his car, the application obtains the latitude and longitude values at that particular position and checks if that value exists in between the latitude and longitude values of the parking lot.[10] If they exist, the application updates the database that the particular parking space is occupied. In this way, the user will be able to obtain the exact information about available parking spaces in a parking lot.

3.1 Proposed Work

The development of the thesis is based on the Resource Allocation Graphs where the processes are the vehicles of the users and the resources are the parking spots. The goal is to allot a resource (Vacant parking spot) in such a way to the process (Vehicle of the user) that it does not complicate any user in deciding between the parking spots that are currently available.

Initially, we need to develop an iPhone application that retrieves the latitude and longitude values accurately. We use the Core Location framework (A review of iOS frameworks and programming is given in Appendix A) in order to get the location co-ordinates. Later, the location coordinates of a particular car are recorded and checked if they are in the range of the values of the parking lot.

If the values exist within the range, we need to update the database that the particular parking space is occupied. In this way, the GPS keeps updating the database continuously and so the user can easily find a vacant parking space in a very large parking lot.[11]

3.1.1 Theoretical Model

Resource Allocation Graph basically allows the user or the system detect deadlock situations. The resources must be allocated very carefully so that no process should keep starving and no process should terminate. This is used to develop the application to find a vacant parking spot based on the GPS inbuilt in the smart phone.

There are several kinds of resource allocation graphs as there might be a single unit of resource or there might be a single resource with multiple units in it.

Representation:

Fig 3: Types of Resources

In the above representation, the resource R1 is a single resource and it is to be used as a single unit on the whole. The resource R2 is a single resource that has four other units of R2.

In this kind, a process that needs only 1 unit of R2 can be allocated one single unit of R2 and the other 3 units can be allocated to any other process that is in need of R2 units.

Example:

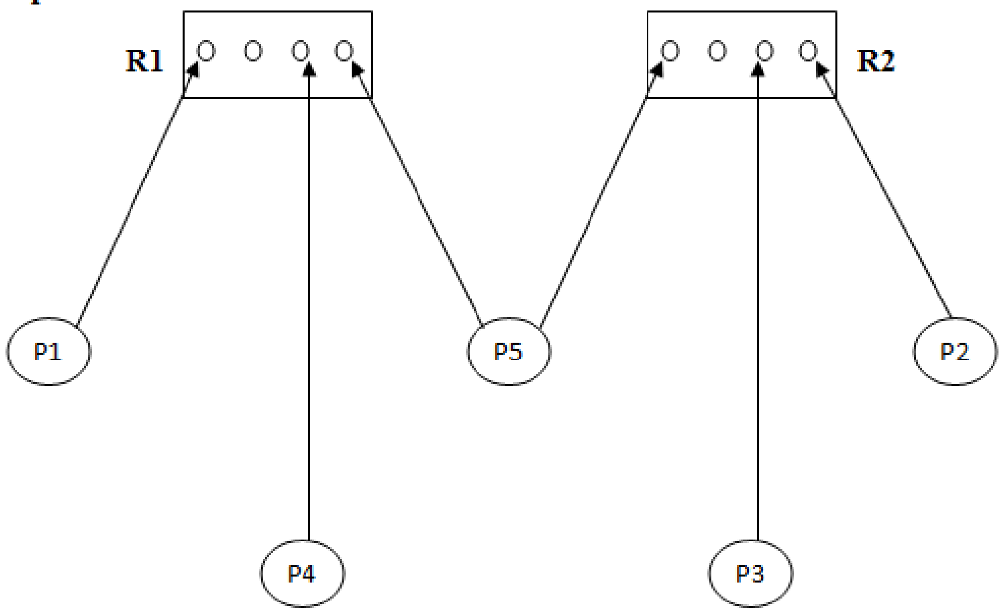


Fig 4: RAG with multiple units of a single resource.

In this case, the processes P1, P4, P5 are allocated single unit of resource R1 and the processes P2, P3 and P5 are allocated single unit of resource R2.

Race Condition:[15]

Race Condition is a condition where two or more processes compete for a single resource.

Comparison:

The above resource allocation graph can be used as a theoretical model to the thesis. The development can be based on the concept of above type of resource allocation graph.

Suppose that there are two parking lots A and B in a particular area. Let us suppose that there are 10 parking spots in each parking lot. We are comparing the resources R1 and R2 in the above graph to A and B parking lots and the processes to the vehicles of the users. Let there be 10 vehicles which are allotted different parking spots. These 10 vehicles need to be allotted parking spaces in such a way that no two vehicles must be allotted the same parking spot which is similar to the deadlock situation in Operating Systems.

The above comparison can be represented as:

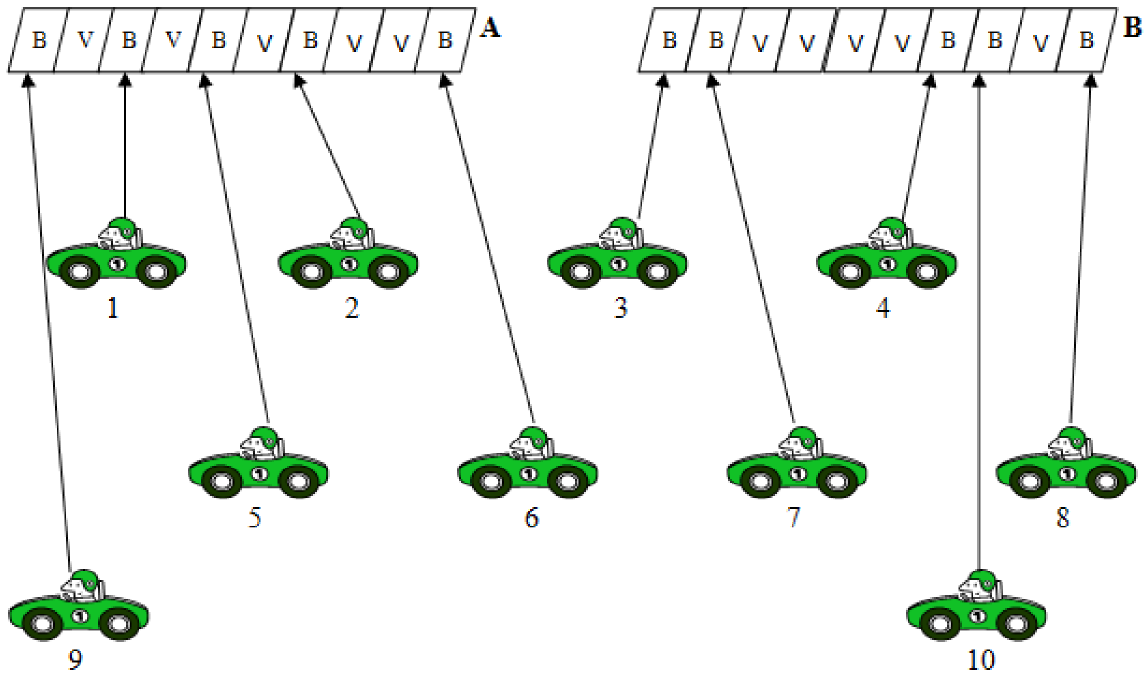


Fig 5: Comparison to multiple units of single resource concept

In this way, every car must be allotted a vacant parking spot being very particular that the allotted slot is not going to be a busy spot in any way like the failure of update to the database or any such kind of issue. The resource utilization is also a very important feature in managing a system to be reliable. So, the allocation of the vacant parking spots must be very effective taking into consideration of distance the user need to travel to reach the allotted parking spot.

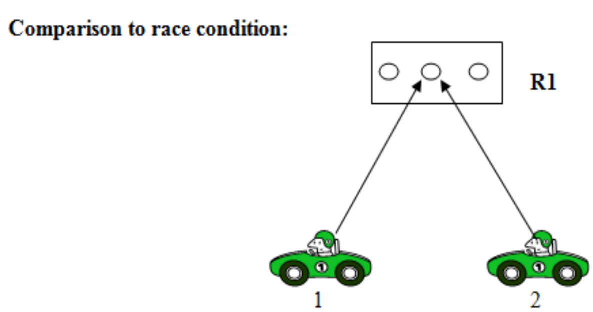


Fig 6: Comparison to race condition

The failure to update the database regularly may lead to a situation similar to race condition in Operating Systems. If the database is not updated, more than one user may see that a particular parking spot is vacant. This may lead to a situation where multiple users may try to park their vehicles in the same parking spot which if vacant can be allotted only to a single user or if is already busy cannot be allotted to any of the user. This condition is the main issue that is to be considered during the development of the thesis.

CHAPTER IV

DESIGN AND DEVELOPMENT

4.1 Motivation

Finding a vacant parking space in a large parking lot manually is a very time consuming process. The user needs to travel all around the parking lot in order to find a single vacant parking space. The user may have to move around and so there is a waste of fuel in the process of searching and the user's time is wasted just to find a vacant spot in the parking lot. As everyone has a smart phone now-a-days and every smart phone has an inbuilt GPS, it would be of great help if there is an application that can be installed on their smart phone which gives the user the information about the vacant parking spots than the user searching manually for one.

Survey Results:[17]

1. "28 % to 45 % of traffic on some streets in New York City is generated by people circling the blocks" - A study released in June by *Transportation Alternatives, a public transit advocacy group[16]*.

2. "Drivers searching for metered parking in just a 15-block area of Columbus Avenue on Manhattan’s Upper West Side drove 366,000 miles a year" - A study released in June by *Transportation Alternatives, a public transit advocacy group.*

3. "There is a need to ensure that at any time on-street parking is no more than 85 percent occupied." - *The city's planners* strategy based on research by *Mr.Shoup.*

4. "Estimate that drivers searching for curbside parking are responsible for as much of 30 percent of the traffic in central business districts." - *Research by Mr.Shoup.*

5. "In one small Los Angeles business district, over the course of a year, cars cruising for parking created the equivalent of 38 trips around the world, burning 47,000 gallons of gasoline and producing 730 tons of carbon dioxide." - *Research by Mr.Shoup.*

6. " In the Harvard Square, there is about 30% cruising (parking space search time) for an average of 11.5 minutes. Between 8% & 74% of cars looking for parking. They take between 3.5 and 13.9 minutes to find a parking space." - *Research by Mr.Shoup.*

7. Some points from a research by Donald Shoup[18]:

- Average time looking for a parking space (6 min/ hour)= 1/10 of hour.
- Average income =$40,000/year or $20/ hour.
- It takes at least 3 minutes to find a space . Average cost of wasted time: $2.
- 30 extra minutes of driving per space per day is wasted. If average car travels at 10 miles per hour, 30 minutes results in 5 extra miles of driving per space per day.
- If the average block has 33 spaces, then the block results in cars cruising 165 miles per day or 60,000 miles per year (twice around the world!)
- Gasoline: 60,000/ 30mpg = 2000 Gallons a year/ block.

Fig 7 : Marked parking spaces in a city of Los Angeles.

YEAR	CITY	TRAFFIC CRUISING	SEARCH TIME (minutes)
1927	Detroit (1)	19%	-
1927	Detroit (2)	34%	-
1933	Washington	-	8.0
1960	New Haven	17%	-
1965	London (1)	-	6.1

1965	London (2)	-	3.5
1965	London (3)	-	3.6
1977	Freiburg	74%	6.0
1984	Jerusalem	-	9.0
1985	Cambridge	30%	11.5
1993	Cape Town	-	12.2
1993	New York (1)	8%	7.9
1993	New York (2)	-	10.2
1993	New York (3)	-	13.9
1997	San Francisco	-	6.5
2001	Sydney	-	6.5
	AVERAGE	**30%**	**8.1**

Fig 8: Twentieth century cruising

As we try to make use of the inbuilt GPS and not any sensors, every user can easily install the application on their phone and use it with a minimal cost.

4.2 Base work

The base work done for the thesis includes developing an iPhone application that can accurately obtain the latitude and longitude values using the GPS present in the iPhone. Creating separate tables in the database and retrieving the data from the database using PHP. Passing the latitude and longitude values obtained between the view controllers and trying to update the database using the latitude and longitude values obtained in the previous view corresponding to the current location of the user.

4.3 Design of the Map view

Designing the map view is the initial step in the design and development of our application. In this the user interface consists of a map view which automatically drops a pin at the current location of the user as soon as the application launches. This map view makes use of the hybrid view of the map. Whenever the user long presses on a particular spot on the map view, the map view drops a pin with the latitude and longitude values of that location.

The map view designed as a part of the thesis allows the users to zoom in and zoom out in order to find the information about the location that the users need to reach. The thesis has been developed and tested based on the Whitehurst Lane of Oklahoma State University, Stillwater. The thesis was tested when the weather condition was sunny mostly.

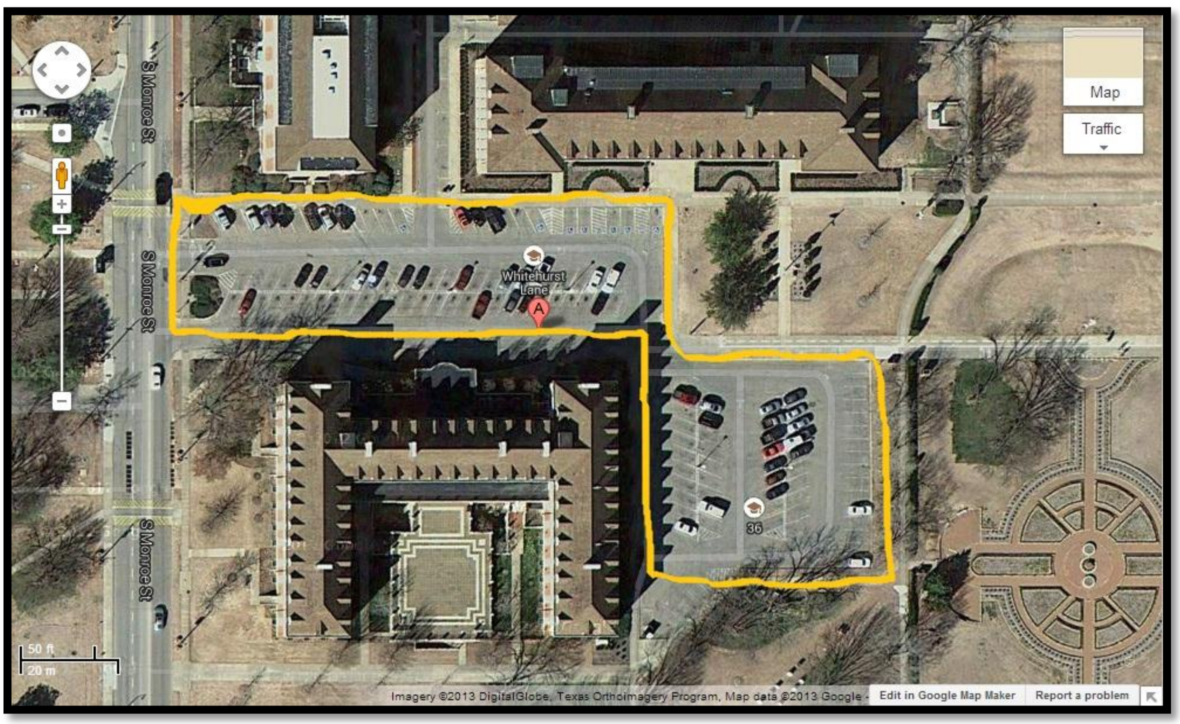


Fig 9: Tested parking lot - Whitehurst Lane, Oklahoma State University, Stillwater.

4.4 Accessing the database

The user interface consists a find button along with the map view. When the user clicks the find button, the user will be navigated to a new view controller where the user will be able to know the current status of the parking lot with the information about all the parking spots vacancy status. These values are used by the user to choose a vacant parking spot to park his vehicle. The user needs to move to that particular vacant spot and click Park button available on the interface.

4.5 Updating the database

The database is updated with the current status of the parking spots in the parking lot. The updating process can be when the user parks his vehicle or when the user wants to vacate the parking spot. For a user to park his vehicle, he first decides among one of the vacant parking spots to park and will move to that parking spot, park his car and clicks the park button on the user interface which will mark that parking spots status from Vacant (V) to Busy (B). The vacate is exactly the same process where the database will be updated for the current status of the parking spot from Busy (B) to Vacant (V).

4.6 Challenges

There are few challenges involved in the development of the proposed parking application. Some of them are the accuracy of the GPS coordinates, necessity to use the application for every user, updating the database on a regular timely basis and the battery consumption.

4.6.1 Accuracy of the GPS coordinates

There are several factors that affect the GPS coordinate values. The accuracy of the GPS coordinates vary based on the signal strength, weather conditions and the wind directions.

When the device receiving the GPS signals has a poor signal strength, the GPS coordinates retrieved are varied with a minute difference. This minute difference is also an important factor when we try to locate if the user is in a particular parking space or not.

The accuracy of the GPS decreases in adverse weather conditions because of the noise present in the surroundings which cause the GPS signals to be effected and the wind direction effects the GPS signals received by the device.

4.6.2 Necessity to use the application

There is an important challenge which is the necessity of using the application. This means that every user needs to have the application installed on their smart phone in order to ensure the accuracy of the application. The user having the application installed can park the vehicle and also update the database by setting the current status of that particular spot to busy. If a user who does not have the application parks his car, the database will not be updated and the database still has the current status as vacant for that parking spot. The challenge here is when another user tries to use the application to find a vacant parking space, the database shows that the parking space which is already occupied is vacant. By this, the accuracy of the application cannot be ensured. So, each and every user having the application installed on their smart phone is a very big challenge involved in the work.

4.6.3 Updates to the database

The database needs to be updated regularly and quickly as soon as the user parks his vehicle in a vacant parking spot. The regular and timely database updates will help us in avoiding the race conditions. Suppose a scenario where a user parks his vehicle in a vacant parking space and does not update the database manually. Then the database still shows that the busy spot is still vacant. In the mean time, if any other users try to occupy that particular vacant spot, then that results in a race condition. A race condition is where two or more processes compete for a single resource. Here, two users need the same vacant parking spot which might already been occupied and failed to update the database or it can be allotted to any one of the users and the other needs to move to other available parking spot which is a waste of time in trying to occupy an already busy spot. This challenge can be faced by updating the database regularly on a timely basis and by ensuring that every user definitely updates the database whenever he parks or vacates the parking lot.

4.6.4 Battery Consumption

The very minute difference in the value of the GPS coordinates retrieved by the applications can cause inappropriate updates to the database and can hence effect the accuracy of the application. So, the accuracy of the GPS coordinates is a very important challenge. In the process of ensuring the accuracy of the application, there is a need to make sure that the accuracy of the GPS coordinates is the best. The Apple API provides a flexibility to make use of the accuracy levels which can be set depending on the user convenience. The available accuracy levels are:

1) kCLLocationAccuracyBestForNavigation

2) kCLLocationAccuracyBest

3) kCLLocationAccuracyNearestTenMeters

4) kCLLocationAccuracyHundredMeters

5) kCLLocationAccuracyKilometer

6) kCLLocationAccuracyThreeKilometers.

The accuracy level *'BestForNavigation'* is to be used to ensure the best accuracy of the GPS. This accuracy level consumes a lot of battery. So, the battery consumption can be considered as one of the challenge.

S. No	***Time tested***	***Battery consumption with GPS on.***	***Battery consumption with GPS off.***
1	1 min	< 1%	< 1%
2	5 mins	< 1%	< 1%
3	10 mins	1% - 2%	< 1%
4	15 mins	1% - 3%	1% - 2%
5	30 mins	5% - 7%	2% - 3%

Fig10:Comparison of battery consumption when GPS is on and off in the smart phone.

4.7 Working of the application

Initially as soon as the application launches, the user's current location is displayed by dropping a pin on the map view. The dropped pin has the location coordinate values of the current location of the user. The user can zoom in and zoom out the map view to select any place on the map view by dropping a pin with the long press. There is a button "Find" and on clicking, the vacancy status of the parking lot is displayed. The user chooses a spot and clicks "Park" button which will update the database for that particular parking spot that the spot has been busy from now. Here when 'Park' button is clicked, the current location coordinates are retrieved and stored. These location coordinates are used to find the distance between the coordinates of all the parking spots in the parking lot. The database will initially have all the values of a parking spot. It includes the latitude and longitude values of four edges of the rectangular parking spot. On clicking 'Park' those values are used and a distance function runs by calculating the distances of current location latitude and longitude with the four edges of every parking space in the parking lot. The distances from the four edges of the current location are calculated and the mean of the distances is performed. Based on the mean value, the least mean value is taken and checked the parking spot to which those coordinates belong to and there by decides that the particular parking space is busy and updates the database.

There are two levels of check performed in the developed application. The level 1 check initially is used to know if the location coordinates are valid coordinates or invalid coordinates. The level2 check is used to know the parking spot in which the user had parked his vehicle.

Level 1 check:

In level 1 check, we initially consider the current location latitude and longitude values retrieved. Based on the parking lot selected by the user, the range of the latitude and longitude that belong to the parking lot are initially considered and stored in the database. The boundaries set in this way are used to check if the given coordinates is a valid coordinate or not by verifying the boundary of the latitude and longitude values for that parking lot. Only the coordinates that are valid are sent further for level 2 check.

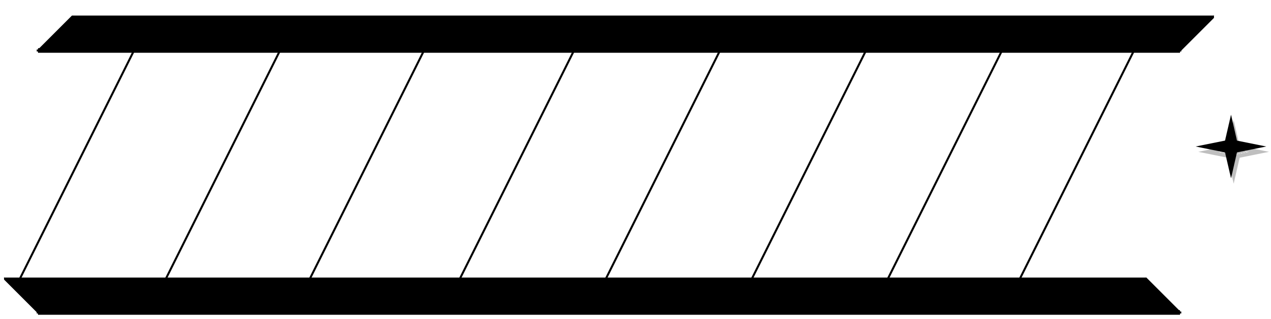


Fig11: Level 1 check - Invalid location coordinates

Level 2 check:

The level 2 check is performed only for the valid coordinates. When we can conclude that the retrieved current location latitude and longitude are valid, we need to know to which parking spot does those values belong. It can be found using the distances of the coordinates from all other parking lot coordinates. We initially have a database that contains a relation which stores the coordinate values of all four edges of a parking space. In that way, all the readings of all the parking spots in a parking lot are taken and inserted into the database. When we find a valid coordinate that has to be checked in level 2, we will try to find the distances of that coordinate from every other coordinates of the parking lot performing the calculation for every single parking spot at a time. We then take the average of the four distances of every single parking spot each time. After the calculation of all the averages, we need to check for the least average value. The least value of the averages can be found and verified to which parking spot the coordinates belong and therefore we can decide that the particular location coordinates belong to that particular parking space.

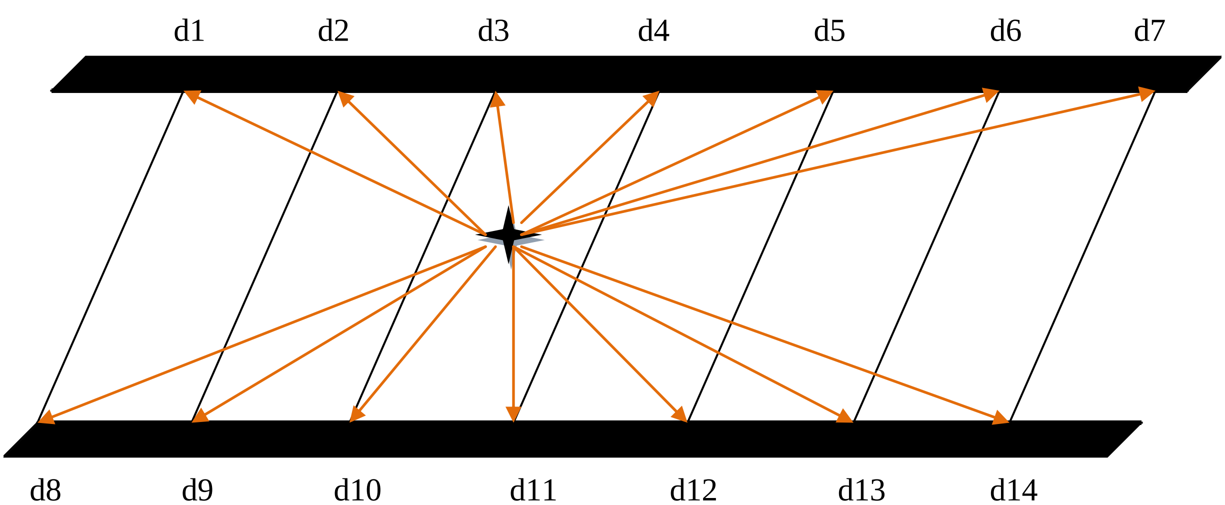


Fig 12: Level 2 check - Calculating distances from the current location coordinates

d1..d14 - distances calculated from the coordinate.

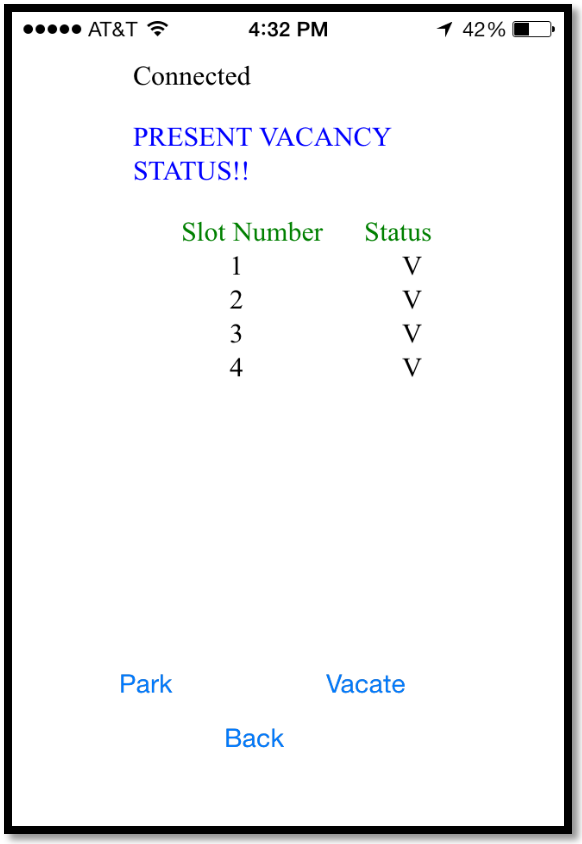


Fig 13: Screenshot of working of the application with four sample slots

4.8 Testing of the application

The application is tested by considering some of the parking spots in the parking lot. Initially as soon as the application launches, the user's current location is displayed by dropping a pin on the map view. The dropped pin has the location coordinate values of the current location of the user. The user can zoom in and zoom out the map view to select any place on the map view by dropping a pin with the long press. The testing can use the current location latitude and longitude values retrieved by the GPS.

4.8.1 Test Methodology

In the testing process, we initially zoom in and zoom out the map view and try to find the desired location. After selecting the location, a long press on the map view will drop a pin on the map view pointing the location selected. Find button in our application will find the vacancy status of the parking lot in the selected location on the map view. The user needs to select a parking spot, reach to it and park his car. When he clicks Park, the distances from his current location to every other edge of every parking spot in that particular lot is calculated. Whenever the application finishes to calculate the distances of four edges of a parking space, automatic arithmetic mean of the distances is performed and stored. In this way the parking space coordinates which end up with least arithmetic mean will be the parking spot where user has parked his vehicle.

The application is tested considering different test cases. The testing of the application involves testing if the application is capable of pointing out the exact parking spot where the user has parked his vehicle or not. The application is also tested to see if the application assumes a user to be in a parking space if he is at the cutting edge of the spot but is not really in that spot.

Testing the application under different weather conditions as the atmospheric changes do effect the accuracy of the GPS coordinates. There were 22 tests taken, of which 19 were accurate and 3 were false. The testing has been accurate in almost many of the cases that have been tested. The cases that have shown inaccurate results because of the wind direction and speed as a future work.

The reason for the inaccuracy of first case of the 3 test cases included a slight difference in pointing out the user's current location exactly. The test was performed by launching the application at the edge of a parking space and with the difference in pointing out the current location by the GPS, the parking space adjacent to the original parking space was marked as busy.

The reason for the inaccuracy for second and third cases also included a bit greater difference than the first case in pointing out the user's current location exactly. The test was performed by testing the application in the middle of the parking space and the attempt to park was unsuccessful by the message invalid location coordinates. The reason was the GPS retrieved current location was outside the parking space but the test was actually performed within the parking space in a particular parking spot.

The difference in the third case might be because of the wireless network access. The test was performed by making use of the cellular data and switching off the Wifi. There was an alert to turn on the wifi to improve the location accuracy before the map view was loaded and so the internet access may also be a factor that effects the location accuracy of the GPS in the smartphone.

The test cases considered are:

- Testing by assuming that a user is at the edge of a parking spot.
- Testing at all possible positions inside a parking spot.
- Testing outside the parking space.
- Testing the accuracy of the database updates to avoid the race condition.

4.9 Results

The results of the test cases are:

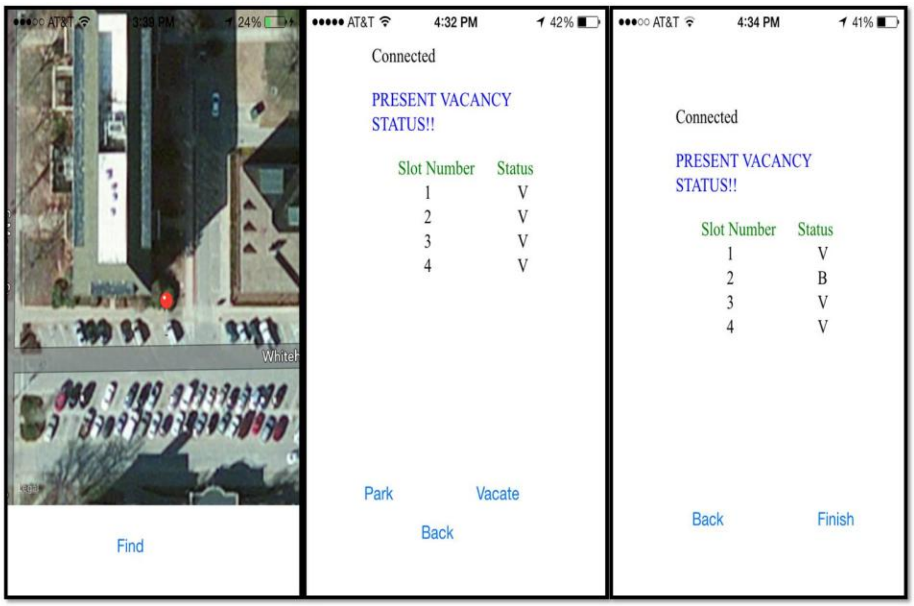


Fig14: Testing by assuming that a user is at the edge of a parking spot.

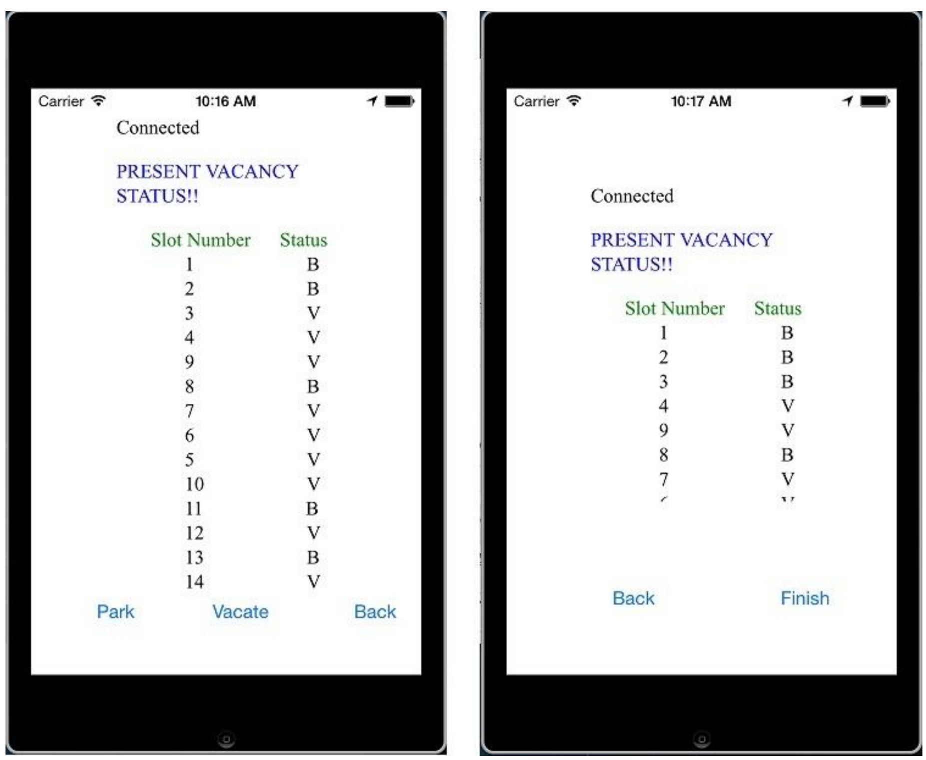


Fig 15: Testing at all possible positions inside a parking spot.

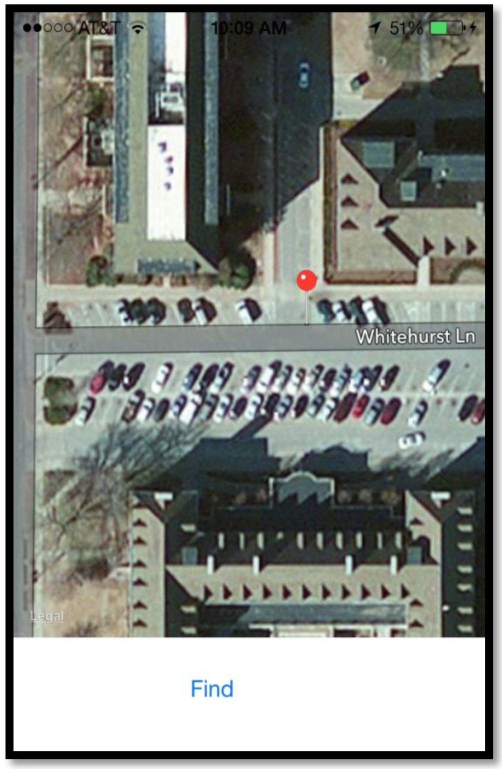


Fig 16: Testing when the user is outside the parking space

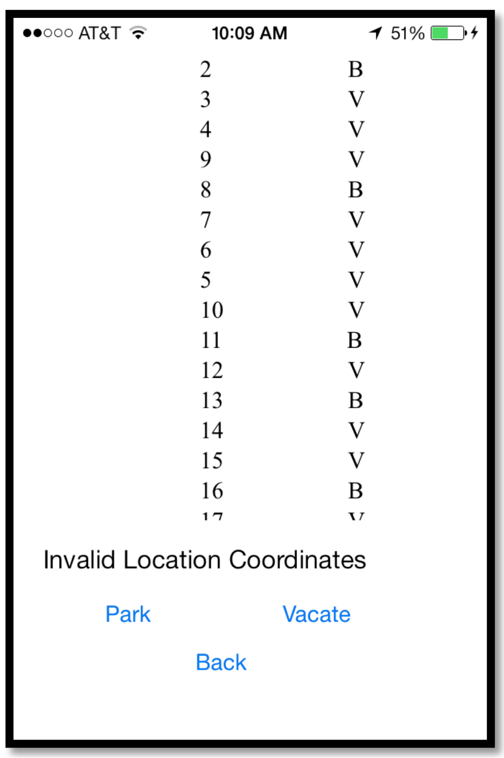


Fig 17: Test results

CONCLUSION

This thesis is the development of a helpful and cost-effective application. This application does not involve any sensors and makes use of only the inbuilt GPS of the smart phone and so is a low cost application.

The application helps the user in saving time in finding a vacant parking spot in a very large parking lot. The user even has a chance to choose the parking spot from a set of vacant spots which is comfortable or suitable to the location the user can easily get back, vacate the spot and move on. The application can be widely used as a low cost parking application. In order for this application to be fully reliable, the following conditions must be met.

1. All users have to get the application installed on their smartphone.
2. The application should be turned on for use.

FUTURE WORK

The future work for the thesis includes the improvements in the application by making the accuracy level of the GPS being a user defined property. The user can decide the accuracy level based on the battery level or the necessity of accuracy by the user.

The testing of the application under adverse weather conditions is also a part of the future work by which we can come out with a solution to the inaccuracy of the GPS in adverse weather conditions so that the user do not find any difficulty in using the application under any weather condition. The future work also includes improving the application in a way such that the application can automatically update the database on occupying and vacating a parking spot rather than the users doing it manually.

REFERENCES

[1] Apple Inc., "Location Awareness Guide", iOS development Library.

[2] Vandad Nehavandipoor, "iOS 6 Programming Cookbook" , O'Reilly Media, 978-1-4493-4275-3, 2012.

[3] Erica Sadun, "iOS 5 Developer's Cookbook: Core Concepts and Essential Recipes for iOS Programmers", Third Edition, 2012.

[4] Apple Inc., " Data Management in iOS and Core Data" , iOS development Library.

[5] Gonzalo Mendez, Pilar Herrero, and Ramon Valladares, "SIAPAS: A Case Study on the Use of a GPS-Based Parking System".

[6] "Find Parking : Parker Mobile" Retrieved from : http://www.streetline.com/find-parking/

[7] Roger Henriksson, "Resource Allocation Graphs", Dept. of Computer Science, Lund University.

[8] Massaki Wada, Kang Sup Yoon, Hideki Hashimoto, "Development of Advanced Parking Assistance System", IEEE Transactions on Industrial Electronics, Vol 50.

[9] Stefano Panzieri, Federica, Giovanni, "An Outdoor Navigation System Using GPS and Inertial Platform", IEEE Transactions on Industrial Electronics, Vol 7.

[10] David M. Belly, J.C Gerdes, C Wilson, G Zang, "The use of GPS Based Velocity Measurements for Improved Vehicle State Estimation", Dept. of Mechanical Engineering, Stanford University and DaimlerChrysler Research & Technology, North America, Inc.

[11] Reginald G. Golledge, Roberta L. Klatzky, "A Geographical Information System for a GPS based personal guidance system", University of California, Carnegie-Mellon University.

[12] Erica Sadun, "The iPhone Developer's Cookbook", Addison - Wesley.

[13] Eugenie V. Uhlmann, Desmond, Kenneth, Niall R.Lynam, "Vehicle based navigation system with smart map filtering, portable unit home-base registration and multiple navigation system preferential use".

[14] Jay Werb, Kelvin Underriner, Martin Long, "Asset and Personnel Tagging System Utilizing GPS", RF Technologies, Inc.,

[15] Cormac Flanagan, Stephen N.Freund, "Detecting Race Conditions in large programs", Compaq Systems Research Centre.

[16] Transportation Alternatives, a public transit advocacy group.

http://www.smrpc.org/transportation/dm/otherplanning/Workshops/Parking%20101.pdf

[17] John Markoff, "Can't find a parking spot? Check Smartphone", The New York Times.

[18] Donald Shoup, "Cruising for Parking".

http://shoup.bol.ucla.edu/CruisingForParkingAccess.pdf

APPENDIX A

REVIEW OF iOS PLATFORM

1. Overview of iOS

iOS is the name of Apple's platform for mobile applications. Mobile applications are a kind of software applications that are designed for the smart phones mainly which are of a great demand. iOS Software Development Kit (SDK) is the most important thing used to develop mobile applications using iOS. To build these mobile applications we mainly need to make use of two things. They are:

1. The iOS system frameworks

2. Objective-C language.

These two are the basic requirements to develop any kind of application on a Mac operating system that is compatible with iPhone, iPad etc. The iOS system framework is used by coding in Objective-C to build an application.

"An iOS system framework is a directory that contains a dynamic shared library and the resources (such as header files, images, helper apps, and so on) needed to support that library."[1] Objective-C is one of the object-oriented programming languages used to develop the iPhone applications in Xcode. This language is the extension of 'C' programming language.

The iOS platform is made up of several layers.

Layers of iOS:

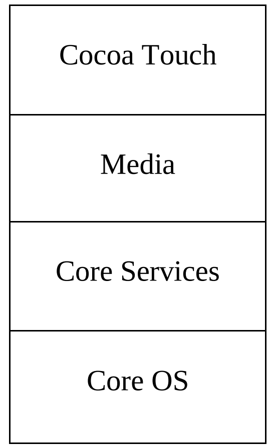


Fig 18: Layers of iOS

- The Cocoa Touch layer mainly has all the important iOS system frameworks that are used in the development of the applications.
- The Media Layer has all the technologies needed to develop the multimedia mobile applications.
- The Core Services Layer provides the basic fundamental services and the Core OS Layer deals with the security of the application we develop.

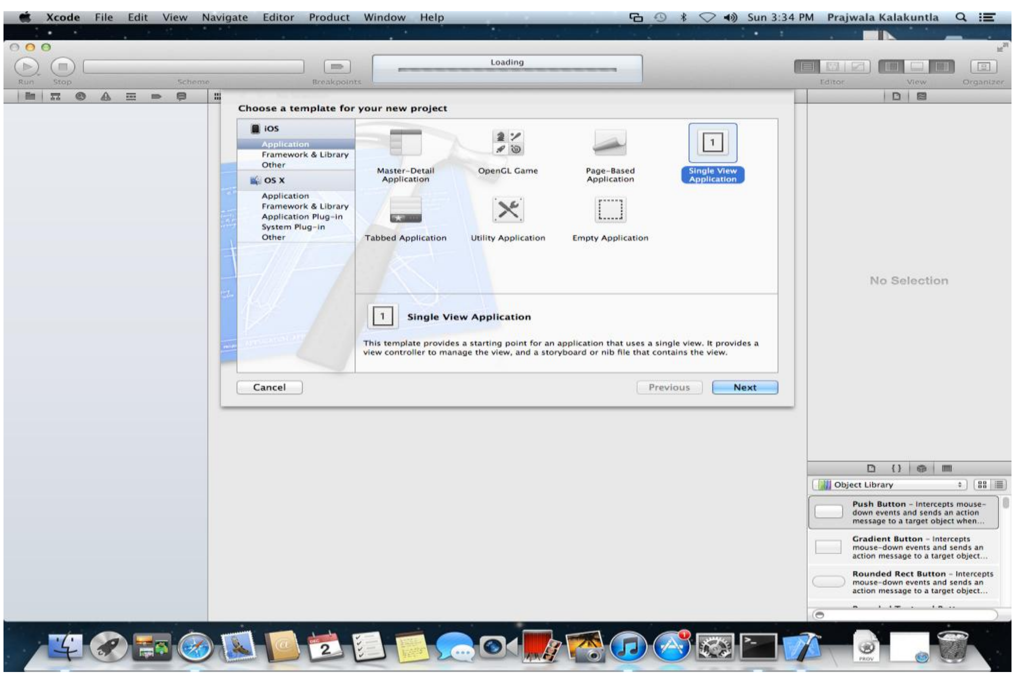
2. Xcode[2]

The development of iOS applications makes use of an Integrated Development Environment (IDE) called Xcode. Xcode provides various varieties of tools which are helpful in building an iPhone application. Xcode has the capability to import and export the necessary frameworks from the library. Xcode gets updated with the latest versions of the iOS and provides the user with the most latest features. The latest full version of iOS is 6.1.3 and Apple Inc. has recently released a beta version of iOS 7. Xcode has an organizer that retrieves the information about the connected iOS device.

The organizer also has the information about the provisioning profiles associated with the device and also the type of the developer profile being used to deploy the applications and the registered authorized developer group details.

Steps to develop a project using Xcode

Fig 19: Creating a new Xcode project.



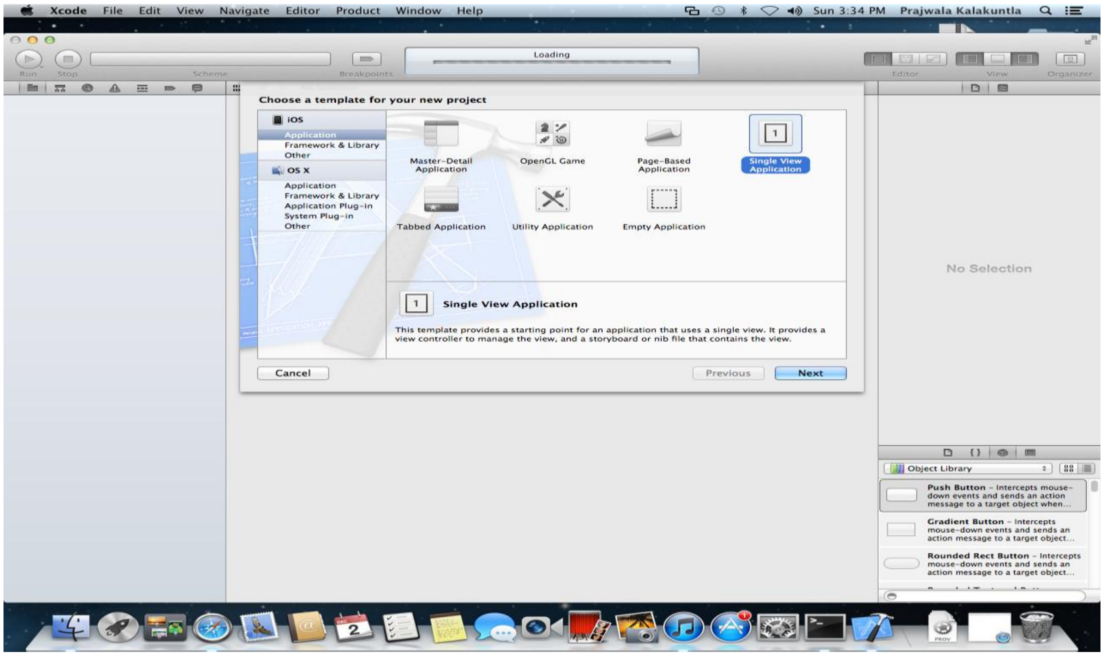


Fig 20:Selecting the type of application we need to develop.

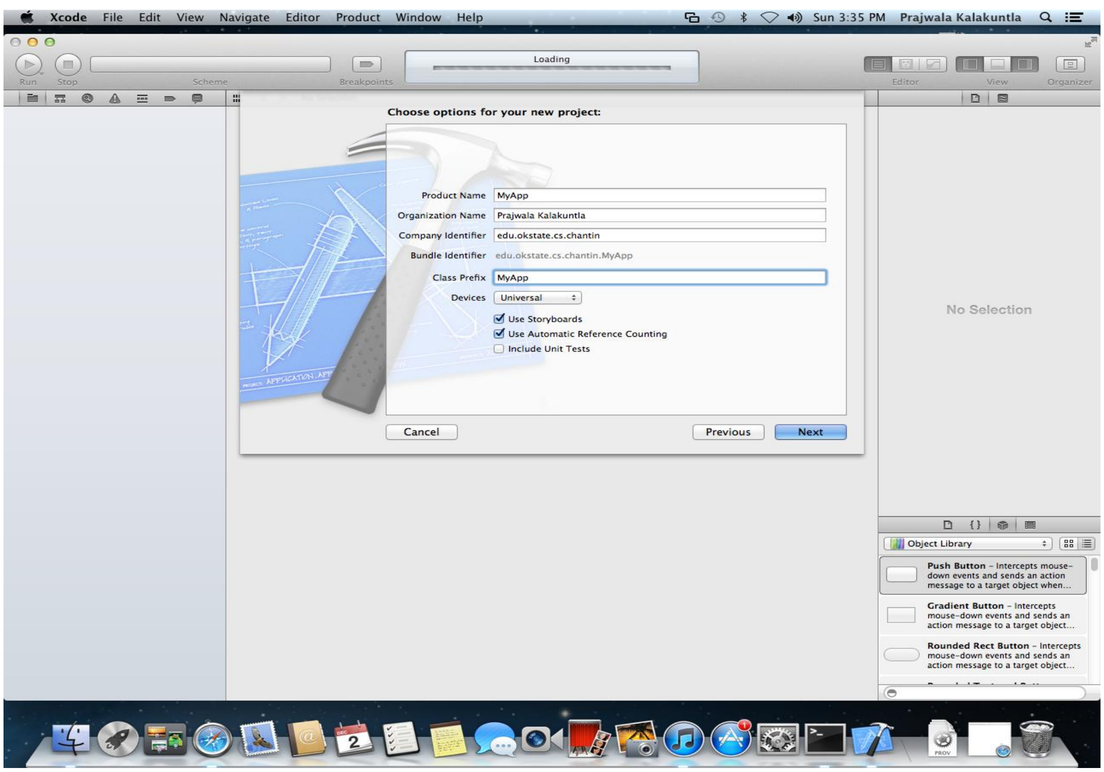


Fig 21: Choosing a name for the application

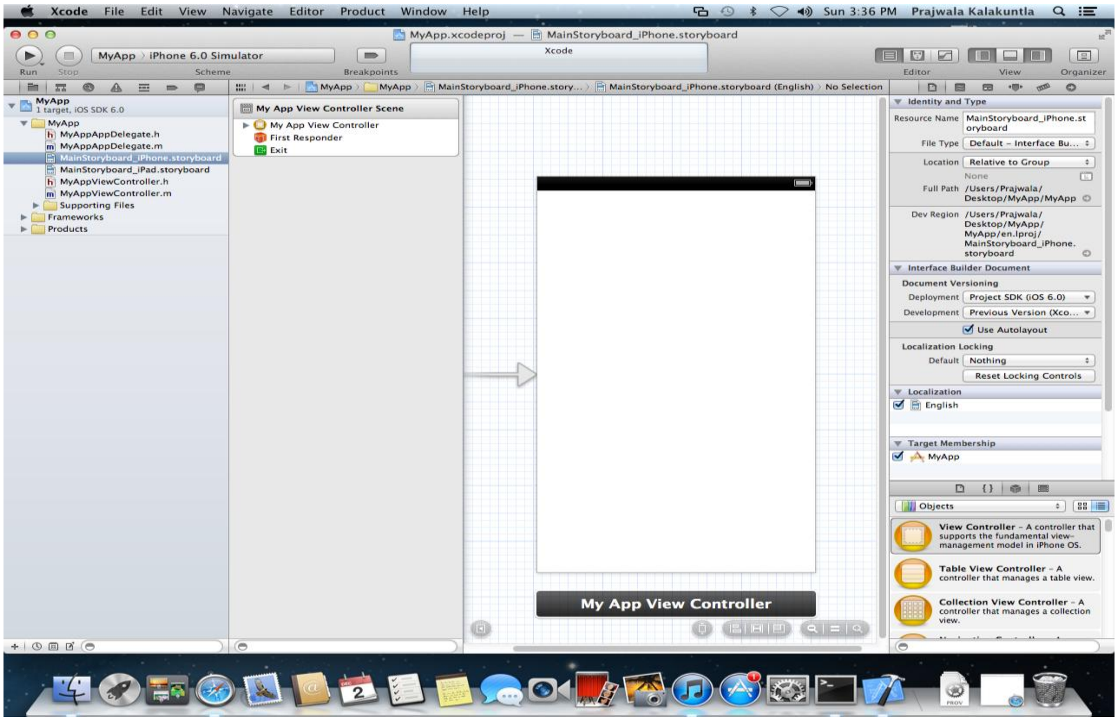


Fig 22: Storyboard, the interface to add UI elements.

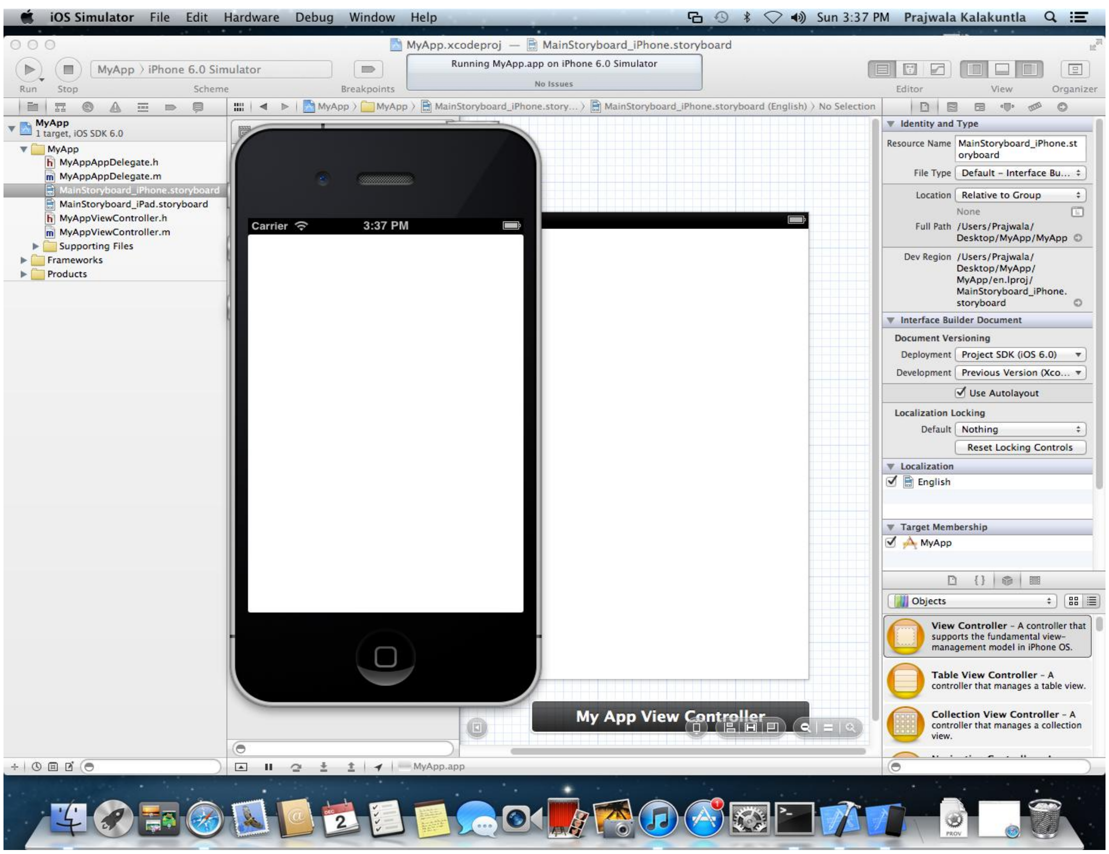


Fig 23: The iPhone Simulator.

The iPhone Simulator[12] is used to test or run the developed applications. It helps us in knowing how an application actually runs on the device. However, the applications involving the usage of GPS and Cellular Data need to be deployed in the original Apple registered devices.

Whenever we create a project using the Xcode, the created project contains four files by default. They are:

1. NameAppDelegate.h

2. NameAppDelegate.m

3. NameViewController.h

4. NameViewController.m

These files contain the basic code needed for the successful completion of the application that is intended to be developed by the user and run on any authorized iOS device.

2.3 Core Location Framework[4]

The Core Location framework provided in the library of Objective-C by Apple is used for the location coordinates to be retrieved from the satellite data by the GPS and these values are used for finding out the vacancy of the parking lot.

"Core Location framework allows us to find the current location of the user based on the user's hardware position and heading."[1] There are various class references in the Core Location framework.

Some of the class references available are:

- CLGeocoder
- CLLocation

- CLHeading
- CLLocationManager
- CLPlacemark
- CLRegion

In this thesis we make use of two of these class references. They are CLGeocoder and CLLocationManager.

CLGeocoder:

"The CLGeocoder class provides services for converting between a coordinate (specified as a latitude and longitude) and the user-friendly representation of that coordinate."[1]

CLLocationManager:

"The CLLocationManager class defines the interface for configuring the delivery of location and heading-related events to your application."[1] This framework helps us in obtaining the latitude and longitude values of the current location.

VITA

PRAJWALA KALAKUNTLA

Candidate for the Degree of

Master of Science

Thesis: USING EMERGING TECHNOLOGIES TO FIND A GEOGRAPHIC PARKING LOCATION IN A DYNAMIC ENVIRONMENT

Major Field: Computer Science

Biographical:

Education:

- Master of Science in Computer Science at Oklahoma State University, Stillwater, Oklahoma, December, 2013.
- Bachelor of Technology in Computer Science and Engineering at Jawaharlal Nehru Technological University, Hyderabad, Andhra Pradesh/India, 2012.

Experience:

- Graduate Teaching and Learning Assistant at Computer Science department, OSU from August 2012 to December 2013.