Mobile Application Based Parking Reservation System

By

Ammar Mohammad Baitalmal

Thesis submitted to the
Faculty of Graduate and Postdoctoral Studies
In partial fulfillment of the requirements
For the M. Sc. degree in Systems Science

School of Electrical Engineering and Computer Science
Faculty of Engineering
University of Ottawa

Abstract

Finding a suitable parking space in busy cities is a time consuming and challenging task. During the searching process, drivers become frustrated and distracted. Motor vehicle drivers might not initially find a free spot, and will leave the area by making a loop to find another spot close to their destination. As a result, there is increased congestion on the road, sometimes causing accidents, and wasting valuable time. To address this problem, we believe that a parking reservation system is necessary and will help reduce the high volume of congestion that might otherwise lead to accidents and have many other environmental and health impacts. The objective of our research is to propose a mobile-based reservation system. The process of finding a free parking space shall be made easy and fast; customers will only be a few taps away from guaranteed and paid parking, based on their preferences. The model presented considers all nearby parking service providers' ability to satisfy customers' requirements and will reserve the best parking for the user.

Acknowledgements

“To my very loved family: Dunia, Mohammed (my parents), Sahar, Hussam, Rabaa, Sumaya (my brother and sisters), and Esraa (my lovely wife).”

Special Thanks to:

Prof. Dr.-Ing. Abdulmotaleb El Saddik, my supervisor.
Ministry of Higher Education, my sponsor.
Sumaya Baitalmal, my sister.
MCR and DISCOVER labs’ members, especially Abdulmajeed AlKhalidy, Dr. Jamal Saboni, Dr. Dewan Ahmad, Dr. Mukesh Saini, Hawazin Badawi, Abdulrhman Alshareef , and Dr. Mohamad Hoda

Table of Contents

List of Tables

List of Figures

List of Abbreviations

2D: Two Dimensional

3D: Three Dimensional

CAS: Consolidated Availability String

CRS: Computerized Reservation System

DATAS: Delta Automated Travel Account System

DB: Database

ERD Entity Relationship Diagram

GSM: Global System for Mobile Communication

HCI: Human Computer Interaction

PARS: Programmed Airline Reservation System

PGI: Parking Guidance Information Systems

PR: Parking Reservation, the proposed iPhone Application

PRS: Parking Reservation System

RFID: Radio Frequency Identification

SABRE: Semi-Automatic Business Research Environment

SDK: Software Development Kit

SMS: Short Message Service

UI: User Interface

VMS: Variable Message Signs

Chapter 1: Introduction

We all know how important transportation is in our lives, and we realize the significant role of motorcars in our daily activities. Despite their personal need for vehicles, however, some people are not aware of the overall growth rate of the vehicle population. John Sousanis states in his article [1] that in a period of 24 years, the number of vehicles transcended 500 million, doubling the number recorded in 1986. The demand for parking is also increasing, in proportion to the increase in vehicle use. In many big cities, finding a vacant parking space is becoming a major problem. In this thesis, we will address this issue and present our systematic solution to this challenging obstacle by introducing a mobile-based parking reservation system. We believe that Internet reservation has many successful scenarios related to parking, as it uses a bidirectional approach [2]. Furthermore, online booking makes it easy, fast, and safe to grant a parking spot, and also enables the businesses of service providers to grow [3].

1.1 Motivation

In the last two decades, governments and companies have realized that digital transformation is the new trend due to its speed and broad exposure to people and consumers. Many services are moving, or have already made the move, to paperless and cloud transactions. Additionally, these days a greater number of people are using technology, and are more interested in online information gathering than stopping at various local businesses. For many years, personal computers were the principal intermediate between users and Internet services, as few handheld devices in the market had wireless capabilities and could be connected with a decent connectivity speed and good Internet browsers. Recently, since the evolution of Apple’s iPhone in 2007, touchscreen devices have had a remarkable impact in the display market. This growth multiplied ten times more than all previous screen developments [4]. With this technology in hand, many institutions, system developers, and designers redesigned and rebuilt webpages and apps to fit the screen size of these devices. Interestingly, not only former computer users are adopting these touch devices, but kids and elderly people are also helping the market expand because of the devices' usability and user-friendly interfaces. James O'Toole recently stated that as of January 2014, in the United States, not only does mobile Internet usage overtake that of personal computers, but the use of mobile apps also surpasses it [5]. In spite of all these great inventions, we are still in the early stages of making our life easier on a day-to-day basis. In this study, parking is the topic that will be discussed, in order to make a better, more reliable, and user-friendly system. Many online reservation websites provide the traditional user interface, where there are a number of fields to be filled and several drop down lists to make different choices; these specific inputs now have to be more graphical and adapted to touch capabilities. We therefore decided to simplify these inputs and make the reservation process easier and faster, with new methodologies to find the best parking for the user.

1.2 Problem Statement

Motor vehicles are a major mode of transportation, which has seen a significant growth over the years. The need for parking spaces is increasing in conjunction with this growth, and becoming a major problem in busy cities. There are many problems associated with this overuse such as pollution, fuel consumption, wasted time because of the looping process [6], a higher percentage of accidents, and drivers' frustration due to traffic congestion [7]. The open loop strategy is defined as the Blind Search, where drivers keep cruising the area looking for a vacant parking and will stop once they reach a free spot [8]. A study shows that approximately 45% of road traffic is caused by motorists looking for a free parking spot [9]. Another study by Donald Shoup [10] specified that about 30% of traffic is mainly due to cruising vehicles in congested area such as downtown. Moreover, traditional methods like static or digital parking signs at sites are no longer relevant, because of the significant increase in drivers who are looking for parking. Drivers' eyes are often busy off-road and they lose their focus on what is happening on the road, by searching for parking signs or free parking. Even if they temporarily wait for a spot to be vacant, parking illegally on-road has a direct impact on road traffic [6]. Nowadays, we have advanced technologies that are used in many different fields to solve problems like reservations. Evoking these technologies and merging them with a single travel related system is still in the early stages and needs to be used by drivers in their day-to-day travels. Certain companies are already applying several reservation methodologies and are making huge investments to achieve customer satisfaction and maximize their profits. In the last decade, there have been many online parking reservation systems that were developed to serve people using computer based web browsers and allow them to book their parking in advance. Unfortunately, these systems were developed to work on personal computers and laptops, before the era of smartphones. With the growth of technology, the idea of finding and reserving parking online can be realized by creating a reservation system that assimilates the concept of a smartphone parking reservation application with parking service providers everywhere. This system will save users time and allow them to know ahead of time when and how to reach their guaranteed parking space, instead of having to travel to their desired destination unsure of where to park their vehicle.

1.3 Thesis Contributions

Here is a list of contributions that we will discuss in our thesis:

- The design and development of several algorithms for an advanced real-time parking reservation system as follows:
    - Building the Consolidated Availability String (CAS) algorithm
    - Selecting hour algorithm

- Finding a free spot algorithm
- Parking price calculation algorithm
- Priority algorithm

- The design and development of an intuitive graphical user interface for the system.
- Designing and implementing the methodologies behind our proposed Consolidated Availability String (CAS) as a proof of concept to develop one of the best user experiences available to help find available parking spots in real-time.

1.4 Thesis Overview and Goal

In this work, we will cover the most relevant solutions to these rising parking issues. Our aim is to provide a solid solution by exploiting the most recent technologies, used day-to-day, to eliminate driver frustration and decrease environmental risks, leading to safer driving and a healthier environment.

1.5 Thesis Organization

The structure of this thesis is organized as follows:

* Chapter 2 discusses the background and related work. It will cover different systems and methodologies used nowadays for informative parking systems and parking reservation systems, and how these systems are related to our proposed system.

* Chapter 3 covers the design of our proposed system, which includes flowcharts, diagrams, and algorithms.

* Chapter 4 presents the implementation phase of the system. In this chapter, a detailed structure of the system is organized and illustrates all of the system components. There is also a description of all of the compartments of the proposed application, and screen shots of the implemented iPhone application.

* Chapter 5 provides the system’s usability test and experimental results.

* Chapter 6 summarizes and concludes this thesis and gives a glance on possible future work.

Chapter 2: Background Information and Related Work

2.1 History of Reservation Systems

In the past few decades, computerized reservation systems (CRS) have gained much popularity. Before detailing its history, here is a definition of CRS:

“a computerized system containing information about, inter alia, air carriers' schedules, availability, fares and related services with or without facilities through which reservations can be made or tickets may be issued to the extent that some or all of these services are made available to subscribers.” [11].

In the 1950’s, some companies started using dedicated CRS before the technology expanded to distribute such systems to travel agencies [11]. The first machine, shown in Figure 1, was called the Magnetronic Reservisor [12], and was used for booking by American Airlines. Later, after the collaboration between American Airlines and IBM, they introduced the Semi-Automatic Business Research Environment (SABRE). In 1968, other airlines started to develop their own systems, for example Delta Airlines, who created the Delta Automated Travel Account System (DATAS). Three years later, Trans World Airlines launched the Programmed Airline Reservation System (PARS) [13].

As the technology advanced and the services improved, CRS were promoted from airline companies to other services such as hotels, car rentals, and cruises [11].

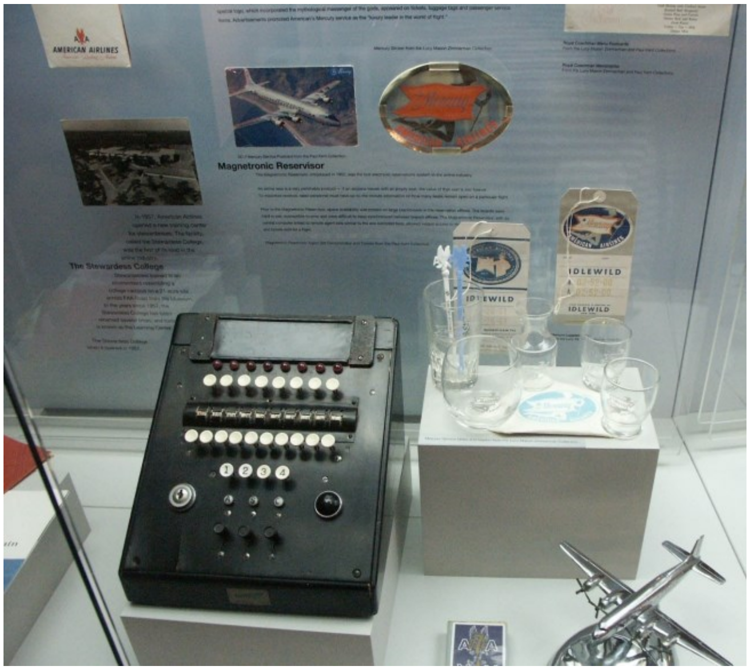


Figure 1 Agent set of the Magnetronic Reservisor system [12]

In the last few decades, an alternative solution came popular called Parking Guidance and Information (PGI) systems mainly for car parking. As car population is increasing, the demand for parking also raises. Few studies aimed due to that fact to lower the time for looking to a parking space and to improve accessibility by adopting PGI systems [14][15][16] as a study [17] indicated that in some cities in Germany awareness of PGI systems is increasing after few months. This invention consists of four components as described in [18]: Information detection, information dissemination, controlling center and telecommunication networking, and lastly, variable message signs (VMS) to provide guidance and display information such as available parking spots in the parking facility. PGI systems implementation started in Aachen, Germany in 1970s and gained popularity worldwide thereafter [14].

2.2 Text-Based Applications VS Map-Based Applications

Although map-based informative or reservation systems look more appealing to the user, we believe that our lite text-mode Parking Reservation System (PRS) is superior in many ways. Table 1 shows a comparison of these two different approaches.

	PRS Application	Map-Based Parking Systems
Time	Visual selection	Drop-down lists (with related timings)
Real-Time Time Selection	Yes	No
Instant Results in Time Selection	Yes	No, results only after clicking search

Table 1 PRS application VS map-based parking systems

If we also include a network reliance to this comparison, the results will be as shown below, in Table 2.

	PRS Application	Map-Based Parking Systems
Data Usage	Text only	Text and graphics (maps)
Data Load Used	Find service providers in the background Download and upload data from and to databases	Webpage interface Displaying results (once or multiple times depending on availability)

Table 2 Network reliance comparison

2.3 Related Work

As we mentioned earlier in this chapter, finding a parking for motor vehicles is a well known problem. As a matter of fact, drivers already struggle to find a free parking space in big cities, and the issue is becoming more severe as time goes on. Few studies touch upon this subject. Mouskos et al. [19] explain the issues related to finding a parking space in a business area and states that it leads to productivity loss, higher pollution, and driver dissatisfaction. Indeed, parking difficulties means more cars on the road, which causes a slowdown in traffic. They concluded by using the Advanced Parking Information Systems (APIS) to find a parking spot and obtain the directions to that specific location, considering the growing PRS. Mouskos et al.

used the concept of PRS to “consider either parking revenue maximization or user parking cost minimization or both, which can be formulated as a max-min type of problem” [19]. Another study by Yanfeng Geng et al. [16] used a dual concepts, PGI and parking allocation via the Internet, allowing drivers to assign and reserves a parking spot. Inaba et al. [20] pointed out the impact of a traditional parking system, which is a threat to public safety. Car accidents result in a high rate of injuries and raise costs, because of the increase in the number of emergencies, in the higher need for health care services, in property damages, and in high traffic jams that they cause. In their paper [20], the authors claim that narrow roads like in Asia increase traffic congestion because of roadside parking. Intelligent Parking Reservation (IPR) systems can solve this problem by allowing customers to select a parking facility, in real-time, through the Internet. This parking service can be used along with a smart card, to identify the drivers when they enter the parking facility, thus eliminating the time needed to stop and get an actual ticket. Leaving the parking facility would also be faster because of the non-stop exit allowed by having more options of payment methods. An IPR can increase safety and eliminate illegal parking using necessary information such as location, service rate, and available spots [20]. A similar approach has been implemented by Chia el al [21]. They developed a smart parking reservation system that allows users to pay through PayPal for the first reserved hour. The system then generates a QR-Code for entering and exiting the car parking during that time. The user is also required to pay the remaining balance prior to his/her exit via PayPal. Hanif et al [22] present one of the solutions described to address the importance of using new technologies. They believe that car parking lacks systematic technologies, and that in most cases, either the whole process is manually operated or the technologies are misused. Motorists circle a lot and waste time to find an available parking spot. They claim that to tackle this problem, Short Message Services (SMS) can be used to reserve a parking space and provide confirmation with the necessary credentials, parking lot number, and expiry date and time. Hanif et al. [22] claim that there are more SMS and MMS users than Internet users, and so they made their system to run over GSM. Similarly, another study [23] uses the same GSM technology to overcome the parking challenge in addition to an RFID sensor to check the parking lot status. Their approach provides a simple and easy solution to a wide range of users, however, users have to send at least two text messages in order to reserve, since the first one is informative and the last one is a reservation. This is an issue especially if those text messages have service charges. In addition, the message has to be in a specific format [23]. Lastly, the user can only obtain information from one parking service provider at a time. Another approach on a different case study was more focused on securing a parking spot and monitoring the parking lot by using the same technology mentioned in [22][23]. When a driver with a reservation reaches the gate, they use a secure password to open it [6]. As for the reservation process, there are several algorithms not only for parking since the procedure can be utilized in other pre-scheduled activities like airline booking, hotels, car rentals, ground services, and many other sectors. In this study [24] they have invented an on-line reservation system specifically for chauffeured car services. As most of nowadays reservation systems, clients would make a reservation either online or via CRS systems and then the reservation system execute two functions: Acquisition, and then validation. After that if there is no valid service provider then the client has to change his/her options and start again. The following Figure 2 shows the reservation system overview of their ground service.

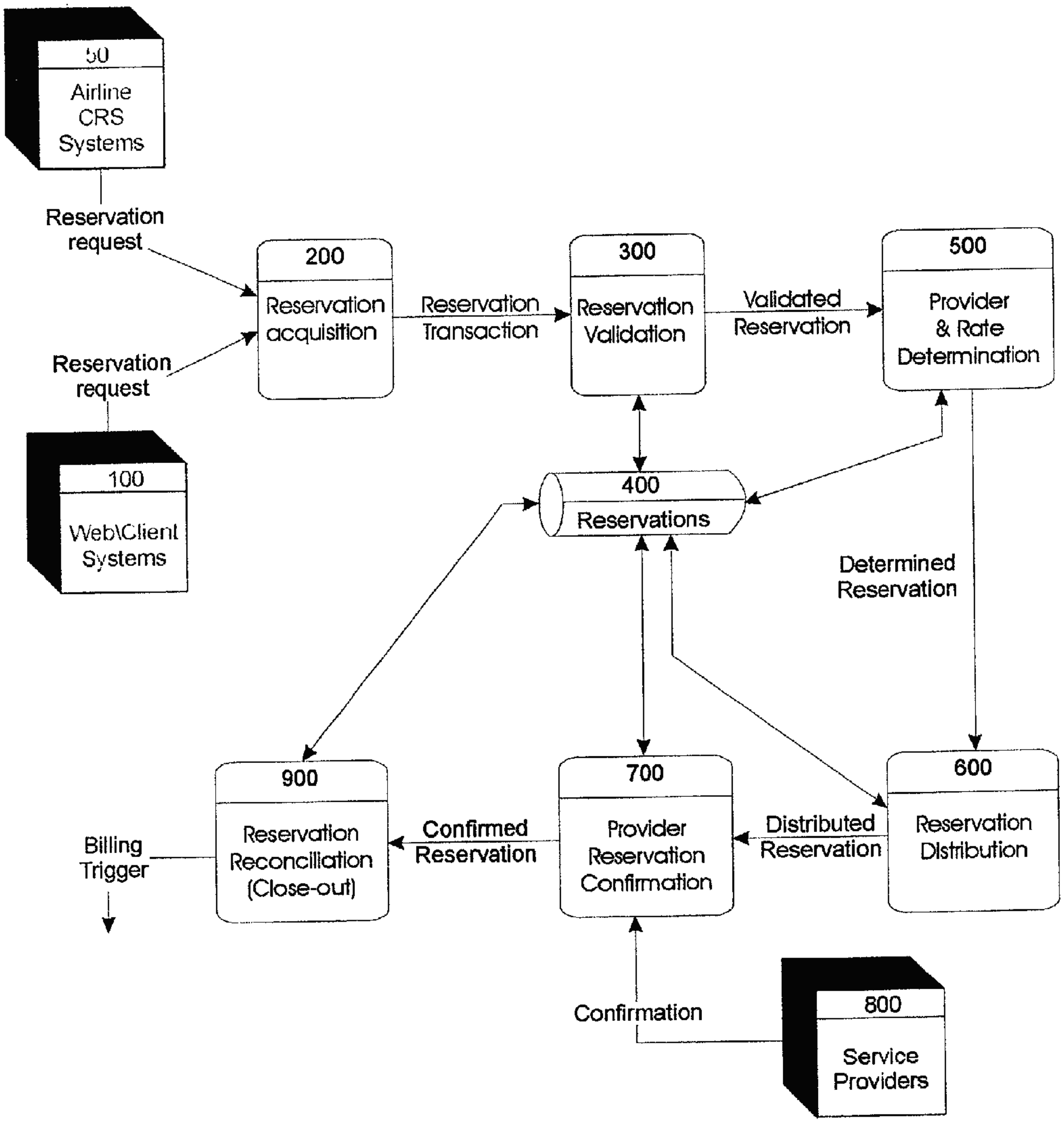


Figure 2 Reservation system overview [24]

A detailed chart of their invention shows the validation process that ends with a valid or an invalid reservation in Figure 3. In our system, however, we used several algorithms to present to the client a valid options before indicating their choice as we are going to discuss in details our system design in Chapter 3.

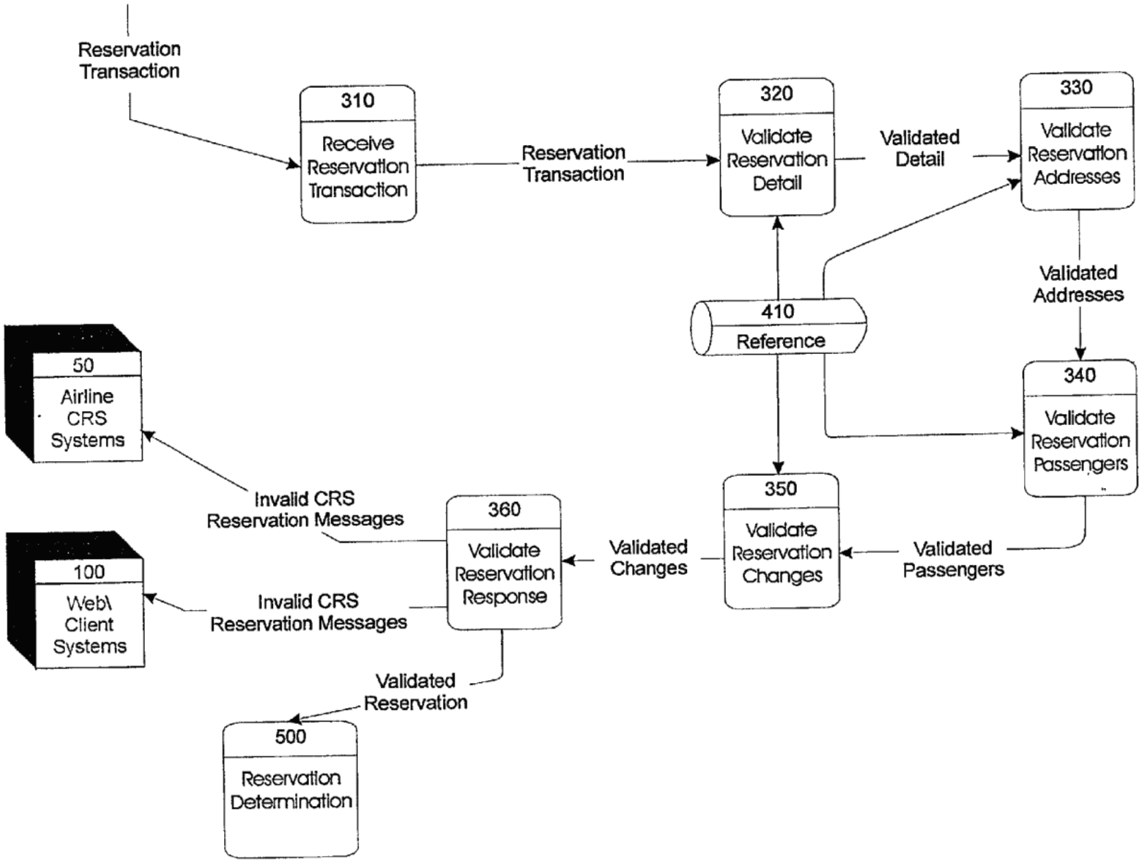

Reservation Transaction
310
Receive Reservation Transaction
Reservation Transaction
320
Validate Reservation Detail
Validated Detail
330
Validate Reservation Addresses
Validated Addresses
410
Reference
340
Validate Reservation Passengers
50
Airline CRS Systems
Invalid CRS Reservation Messages
360
Validate Reservation Response
Validated Changes
350
Validate Reservation Changes
Validated Passengers
100
Web\ Client Systems
Invalid CRS Reservation Messages
Validated Reservation
500
Reservation Determination

Figure 3 Reservation validation [24]

Another common step in most reservation systems is showing the user multiple results. Here is an example as shown on Figure 4 of a hotel enquiry system [25] that shows the reservation flow:

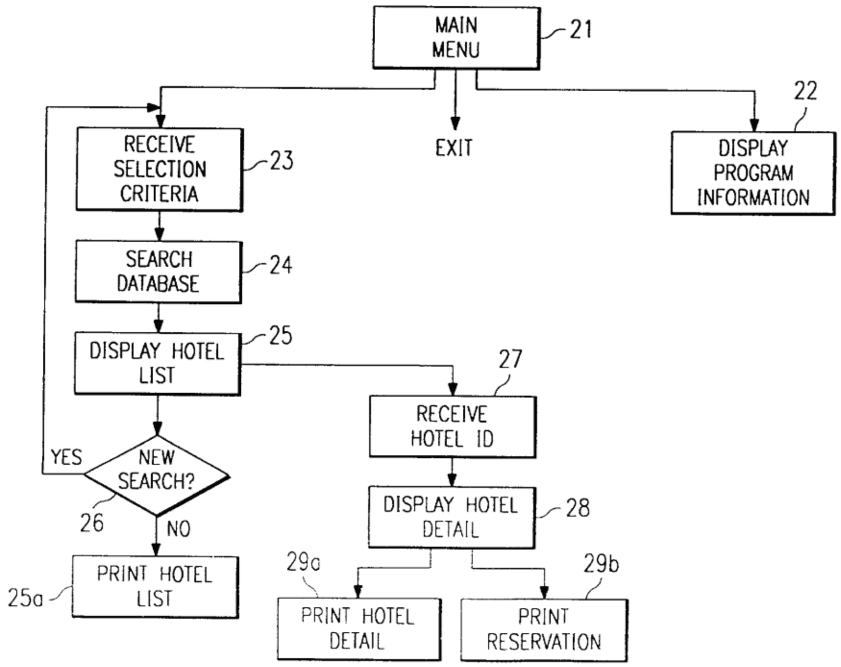
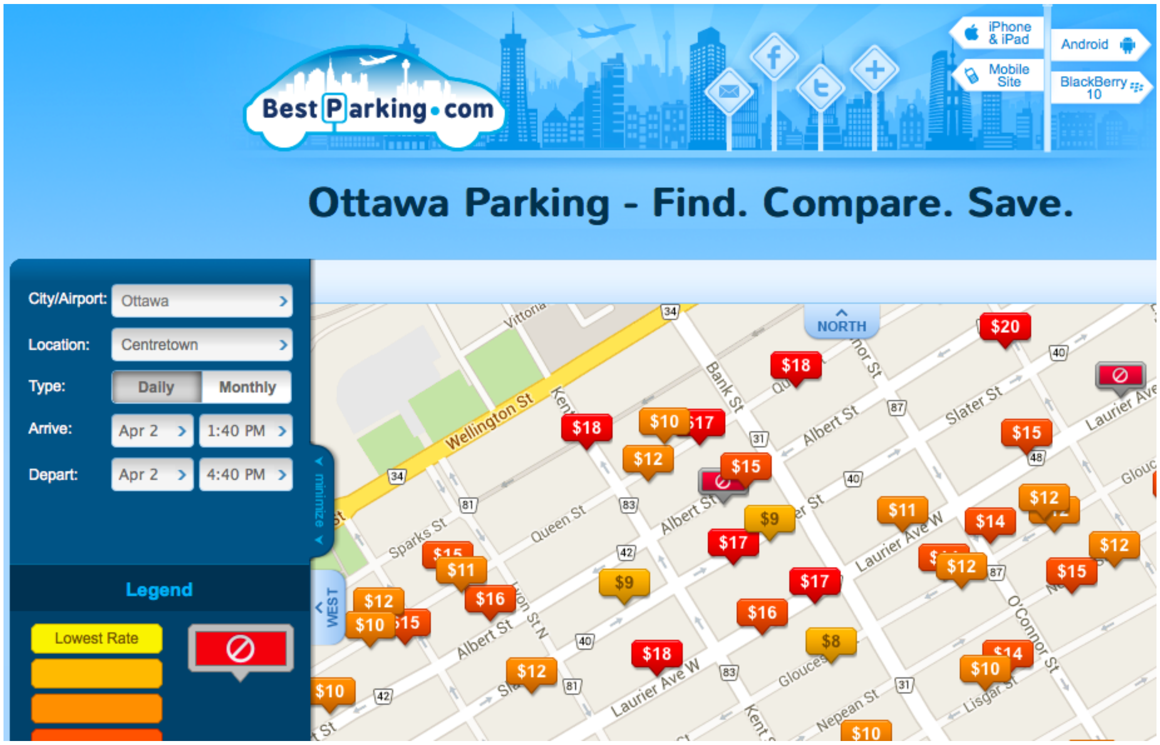


Figure 4 Hotel enquiry process [25]

The user enters his search criteria to narrow the results to find his/her best hotel and choose from the displayed list. Although this method is effective and gives the user more options to choose from, but one of the drawbacks is that the user already applied some conditions during his/her search to satisfy their needs but they still have a list of hotels for comparison thereafter. In our system however, the system nominates one result based on the user criteria, which we will cover in the next chapter.

There are few comparable systems that allow users to both find nearby parking and obtain useful information; it seems as though some have a reservation option, while the others are only informative. In addition, few other systems work with online parking reservation using a GSM network to make a reservation with specific service providers. In this section, we will briefly discuss and compare some of these systems.

2.3.1 Best Parking [26]



Ottawa Parking - Find. Compare. Save.

City/Airport: Ottawa

Location: Centretown

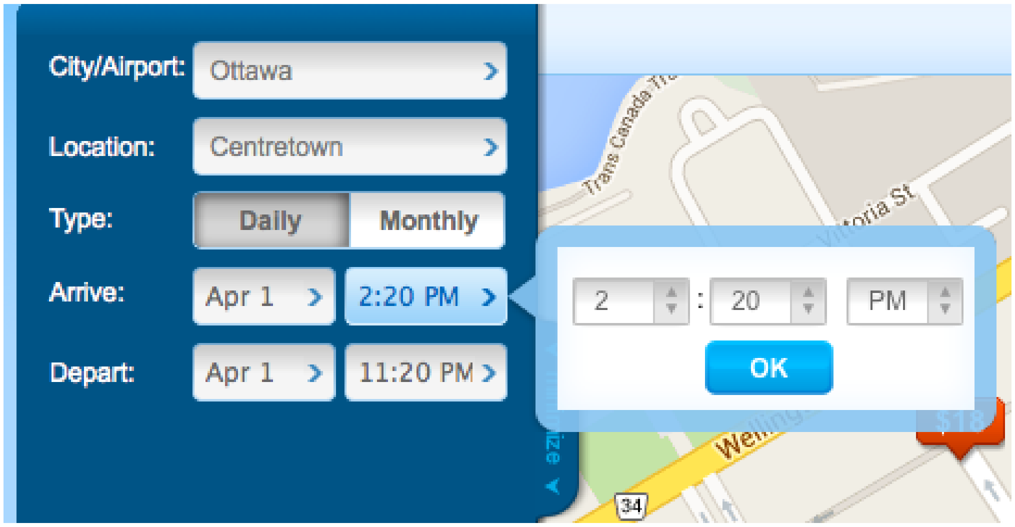
Type: Daily Monthly

Arrive: Apr 2 1:40 PM

Depart: Apr 2 4:40 PM

Figure 5 Best Parking map view

Best Parking is an online-based parking search engine. They provide motorists with hourly, daily, weekly, and monthly rates. Drivers can use their system online or by downloading the mobile application; they can search for parking by current location, specified address, cross street, or attraction site. Users have to select from the drop down lists the city, location, parking type, and duration of the parking. Since the results are map based as displayed in Figure 5, the results will appear as priced pins. Prices are automatically calculated as the user chooses the arrival and departure times. Despite the fact that these calculations are accurate and satisfy the user, he/she still needs to check the sites by clicking on each individual pin for more information. In addition, the results are informative and do not reserve a parking spot. Furthermore, clicking on drop down lists and choosing arrival and departure times is time consuming. For instance, choosing an arrival time or a departure time, as shown in Figure 6, takes about 8 clicks (from choosing an hour, minute, and morning or evening, until pressing the OK button), to which we still have to add the time needed to choose the date. Last but not least, if the user is more interested in a table view, which is an optional feature where the user can see a list view of the facilities, he/she needs to locate the desired option, e.g. sort by price, and then the list will appear. The table view is a great way to show the user all of the results in one place, and adding a sorting functionality makes it even more convenient, but it is still not ideal for people who just want to get the best rates right away. Also, we think that in most cases, users are not interested in sorting by street name or facility name but are just looking to find a suitable parking, as shown in Figure 7.



City/Airport: Ottawa

Location: Centretown

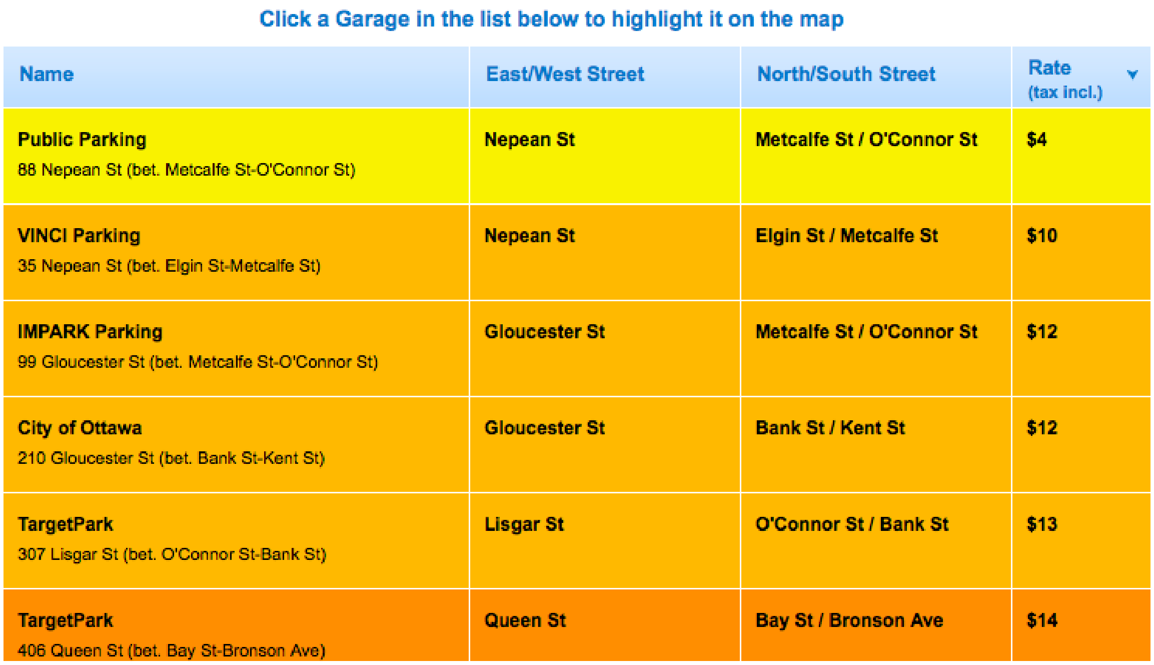
Type: Daily Monthly

Arrive: Apr 1 2:20 PM

Depart: Apr 1 11:20 PM

2 : 20 PM

Figure 6 Best Parking search criterion



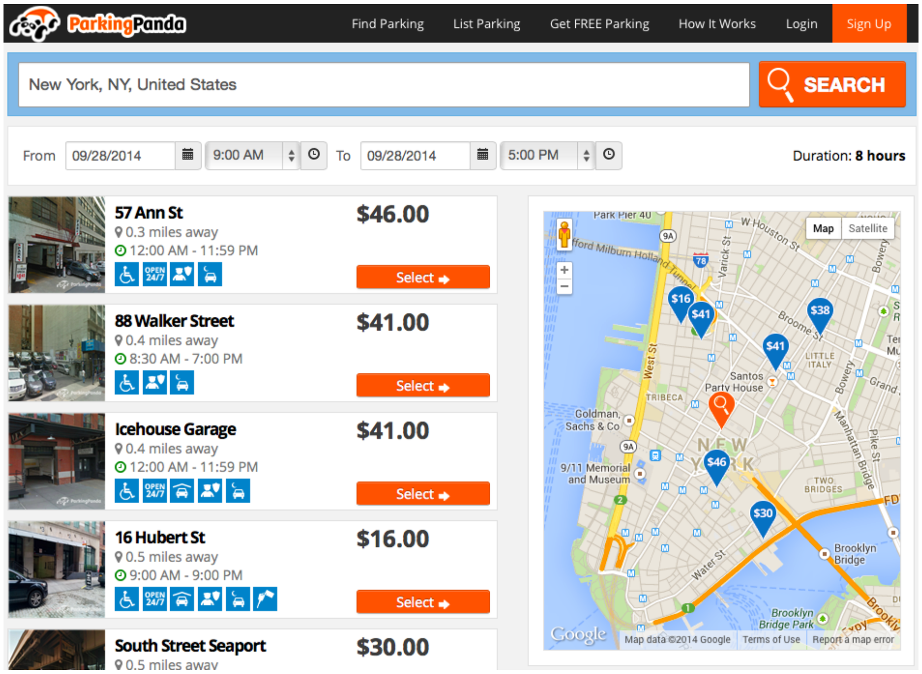
Click a Garage in the list below to highlight it on the map

Name	East/West Street	North/South Street	Rate ▼ (tax incl.)
Public Parking 88 Nepean St (bet. Metcalfe St-O'Connor St)	Nepean St	Metcalfe St / O'Connor St	$4
VINCI Parking 35 Nepean St (bet. Elgin St-Metcalfe St)	Nepean St	Elgin St / Metcalfe St	$10
IMPARK Parking 99 Gloucester St (bet. Metcalfe St-O'Connor St)	Gloucester St	Metcalfe St / O'Connor St	$12
City of Ottawa 210 Gloucester St (bet. Bank St-Kent St)	Gloucester St	Bank St / Kent St	$12
TargetPark 307 Lisgar St (bet. O'Connor St-Bank St)	Lisgar St	O'Connor St / Bank St	$13
TargetPark 406 Queen St (bet. Bay St-Bronson Ave)	Queen St	Bay St / Bronson Ave	$14

Figure 7 Best Parking list view

2.3.2 Parking Panda [27]

Parking Panda is an informative parking system that also allows drivers to reserve a parking space in advance. The main advantage of this service is the possibility of finding a guaranteed a spot [27]. When the user searches for a parking space, as displayed in Figure 8, both the website and the mobile application display the results in a map and a list format. The results show the real-time prices and the distance to the facilities; however, the list view is fixed and can't be sorted by price or by any other attribute. This inconvenience causes the user to spend more time navigating and comparing between different options, until he/she finds the best location. Payment is made afterwards via either debit or credit.



ParkingPanda | Find Parking | List Parking | Get FREE Parking | How It Works | Login | Sign Up

New York, NY, United States | SEARCH

From 09/28/2014 9:00 AM To 09/28/2014 5:00 PM Duration: **8 hours**

57 Ann St — 0.3 miles away — 12:00 AM - 11:59 PM — **$46.00** — Select

88 Walker Street — 0.4 miles away — 8:30 AM - 7:00 PM — **$41.00** — Select

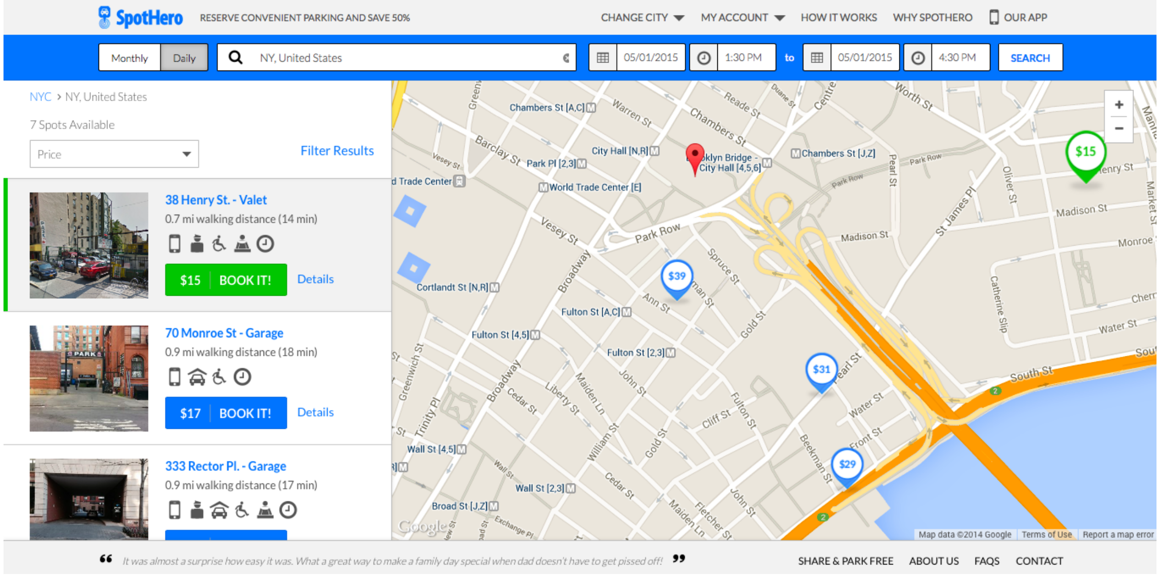
Icehouse Garage — 0.4 miles away — 12:00 AM - 11:59 PM — **$41.00** — Select

16 Hubert St — 0.5 miles away — 9:00 AM - 9:00 PM — **$16.00** — Select

South Street Seaport — 0.5 miles away — **$30.00**

Figure 8 Parking Panda map and list view

2.3.3 Spot Hero



SpotHero RESERVE CONVENIENT PARKING AND SAVE 50% | CHANGE CITY | MY ACCOUNT | HOW IT WORKS | WHY SPOTHERO | OUR APP

Monthly | Daily | NY, United States | 05/01/2015 | 1:30 PM | to | 05/01/2015 | 4:30 PM | SEARCH

NYC > NY, United States

7 Spots Available

Price | Filter Results

38 Henry St. - Valet
0.7 mi walking distance (14 min)
$15 BOOK IT! Details

70 Monroe St - Garage
0.9 mi walking distance (18 min)
$17 BOOK IT! Details

333 Rector Pl. - Garage
0.9 mi walking distance (17 min)

“ It was almost a surprise how easy it was. What a great way to make a family day special when dad doesn't have to get pissed off! ” | SHARE & PARK FREE | ABOUT US | FAQS | CONTACT

Figure 9 Spot Hero map and list view

Spot Hero is another informative parking reservation system with a similar concept, where users can find a parking spot by address and get the results on a map [28]. What makes this service different than Parking Panda is that on the left-hand side, they have a filter to prioritize the list of results. Although this feature gives more focus to the user, it can only filter by price or by distance from the searched address. In the end, the user might simply look to the map to see other options. Figure 9 shows Spot Hero’s map and list view.

2.4 Conclusion

In this chapter, we discussed the emergence of the computerized reservation systems and how other sectors have adopted these systems. Moreover, we discussed other work related to our findings. Although existing systems are providing reliable parking information and some has a full reservation process, unity of services provided, easiness of the reservation process, and eliminating uninteresting parking spots is not available in one unique system. It now becomes obvious that there is a need for a better parking reservation system to facilitate the reservation process and improve the quality of such systems. The design of our proposed system will be discussed in the next chapter.

Chapter 3: System Design

In our system, we have set quality and usability as the top priorities. Designing the user interface is a major part of our system, in order to make it as user-friendly as possible. For us to make this possible, we designed a user interface and tested it to make the parking reservation with a minimal number of steps and taps. A few algorithms were implemented to fulfill this requirement, and all these algorithms are implemented in one single screen on the PRS. The first algorithm is used to reduce the human error by accepting only slots of one hour, as in Figure 10.

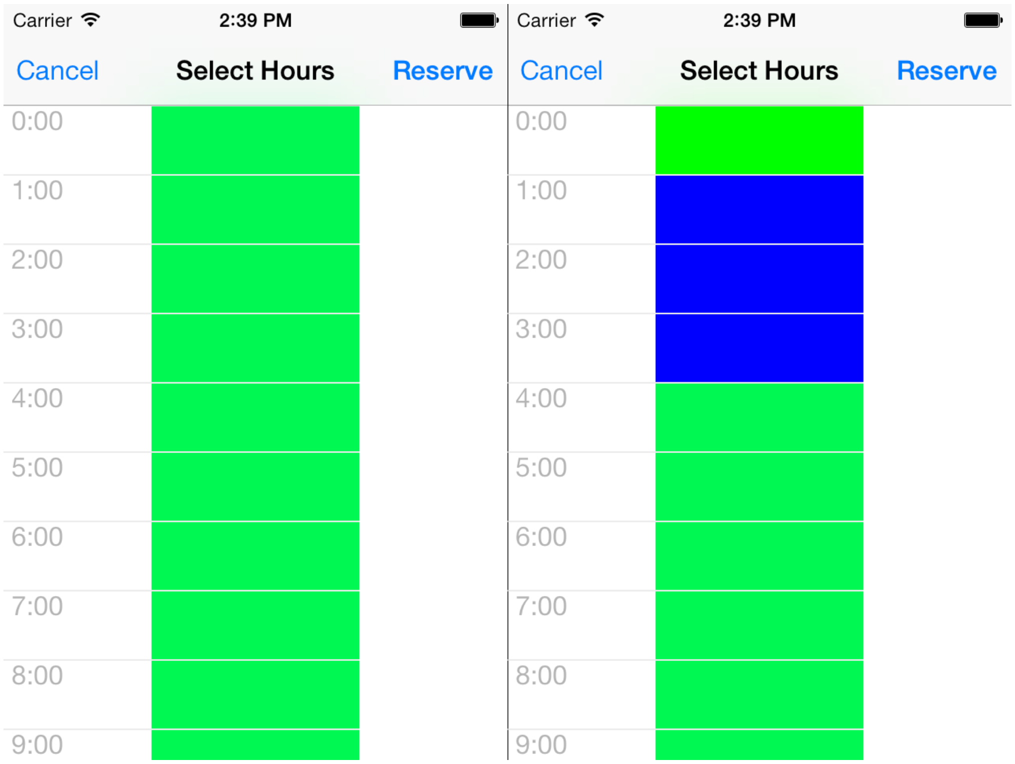


Figure 10 Consolidated availability string-hour selection

In Figure 10, the user can see a list of available hours in green, and can select the desired hour by tapping on it. After tapping on a specific hour, the hour cell will turn blue, which indicates that this blue cell is reserved if the user confirms his/her request.

3.1 System Inputs and Outputs

In this section, we will breakdown all of the system's inputs and outputs.

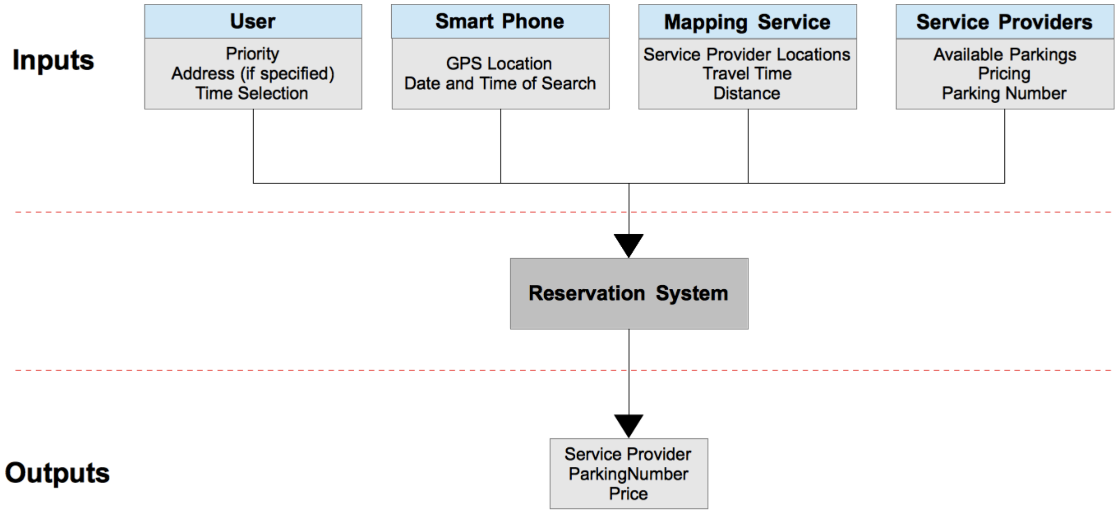


Figure 11 Abstract view of the searching methodology

The system has four main input components: user, smartphone, mapping service, and service provider. The first input is from the user; they have to set their priorities in order for the results to be filtered based on their needs. The user will only make his/her time selection further into the reservation process. The next input is based on the user's interest; whether they want to enter the destination address or, if they are looking for a nearby parking spot, the smartphone itself can detect any available spots. The smartphone's GPS chip will be the next input, only if the user did not specify an address. Also when the user is looking for parking, the system will capture the date and time of the search as an input, which will help find vacant spots from different parking service providers. The third input is the mapping service, Google maps, where service provider locations, time, and distance are collected and processed, once all inputs are gathered. The available parking spots, as well as the prices of multiple parking service providers, are used as the last input; the system can then determine the best results for the user. Once the system collects all of the inputs, the system will reserve the parking and the final output to the user is a complete reservation with the service provider name, parking number, and price. Figure 12 shows an activity sequence of all the inputs and the outputs.

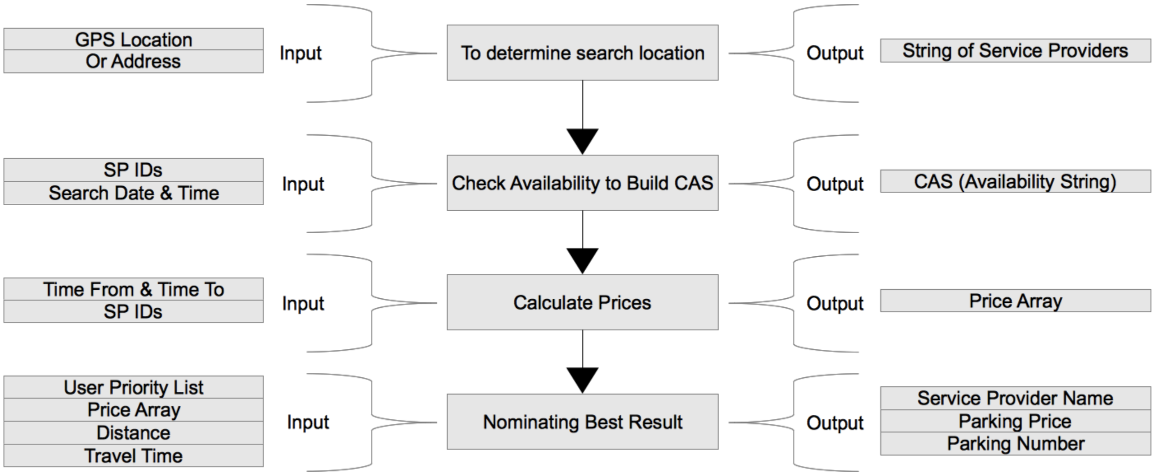


Figure 12 Activity Sequence, inputs and outputs

The above Figure 12 illustrates the activity sequence that covers the order of the reservation tasks in time. Each task requires few inputs to proceed to the next task block. The first task is to search for parking locations. This requires a specified address or could be detected by the GPS chip as an input and the outcome is a list of multiple parking service providers in a string. Moving to the next task, the system will use the string of each individual service provider along with the search date and time to combine results in one string, which is CAS. As a result, the user is now able to choose his desired time based on the availability and their input cooperates with all previous inputs to activate the third task block. The Third task block calculates the prices in order to get a string of price for each service provider. After all, the system will determine upon all the inputs form the user in addition the presets: Priority List, Distance, and Travel Time to nominate one parking spot from all previous results. The output would be the total price, name of service provider, and at last the parking number.

3.2 Parking Reservation Methodology

In this part, we will explain how our system works and discuss the system's inputs and outputs. A high level of our systems reservation process is shown in Figure 13.

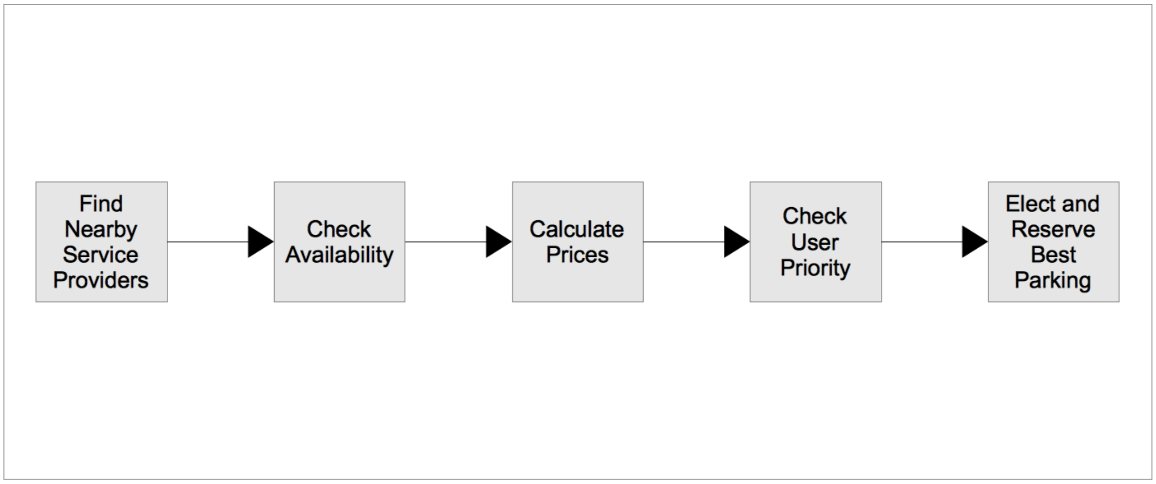


Figure 13 High level reservation process

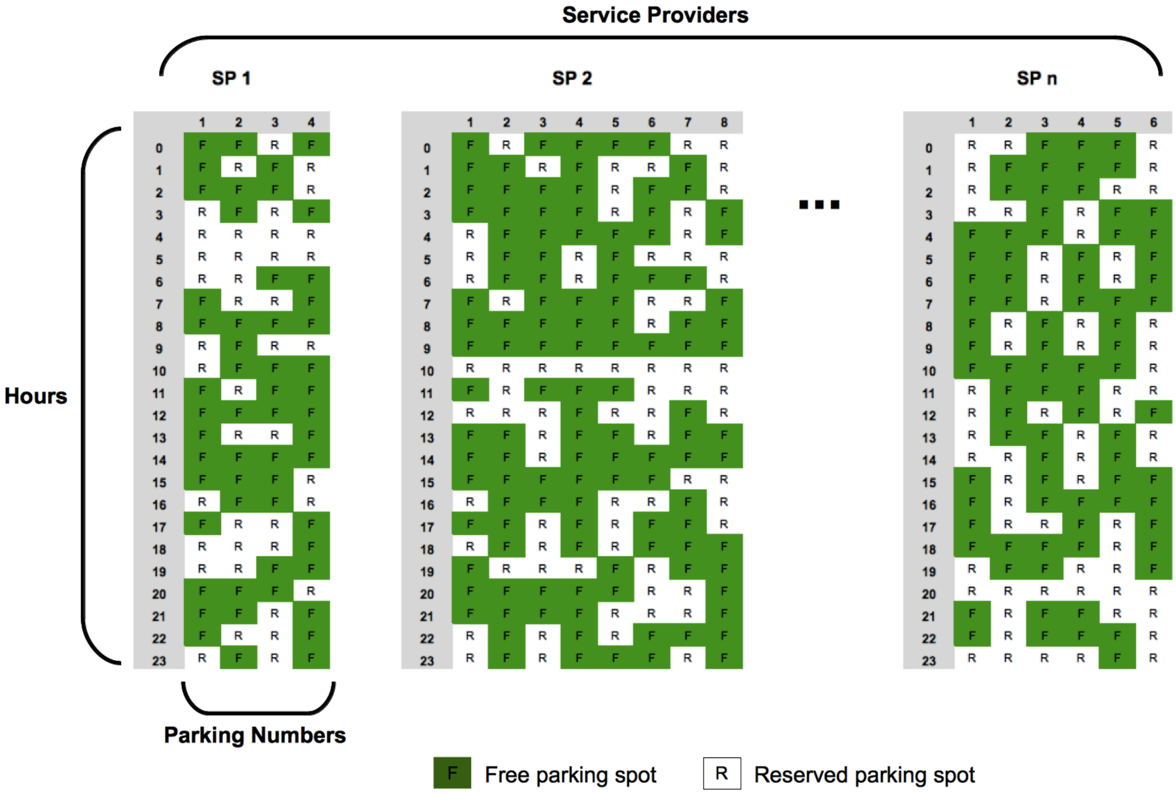
As shown on the previous Figure 13, the system's methodology begins with finding nearby service providers. After that it narrows the list by checking available spots among all service providers in the following process after the user specifies the time period of the reserved parking spot. The next process calculates the prices based on the predecessor completion leading the successor process to add more variables such as the user priority, where they can set their priorities to help the system providing best result such as lowest price, shortest path, or shortest time, to direct the best parking for the user. Finally, the system will elect the best parking spot and reserve it for the user.

3.3 Finding Vacant Parking

The first input to the system is determined when the user chooses the destination location, either by writing an address or based on his/her current location. This key input gives the system the starting point to find all nearby parking service providers; at this point, the system also detects the date and time of usage and registers the information as another input. For prototype testing purposes, the available hours are limited to twenty-four hours after the time of search, which satisfies the conditions of a same-day reservation system.

With this input, the system can evoke the availability of a variety of service providers. At this point, CAS eliminates all of the reserved parking spots and only shows the available spots, making it easier for the user to view the options and to choose the desired hours; the green cells show the available hours and the grey cells indicate that the hour for that parking spot is reserved by others.

Once the user taps on the find button, a query is executed to capture four variables from the parking service providers' database: Service Provider ID, Parking Number, Reservation Time From, and Reservation Time To. At this point the system gets a two dimensional array of multiple hourly reservations on the locations found. The system will use these variables to build three dimensional arrays. The system will initially fill all cells as free spots, but then it uses the captured data to fill all reserved hours to the new three dimensional array. The first dimension is the hours, the second is the parking numbers, and the third dimension is the service provider IDs. Figure 14 shows few different service providers; each service provider represents a dimension and has the second and the third dimensions (hours and parking numbers, respectively).



SP 1

	1	2	3	4
0	F	F	R	F
1	F	R	F	R
2	F	F	F	R
3	R	F	R	F
4	R	R	R	R
5	R	R	R	R
6	R	R	F	F
7	F	R	R	F
8	F	F	F	F
9	R	F	R	R
10	R	F	F	F
11	F	R	F	F
12	F	F	F	F
13	F	R	R	F
14	F	F	F	F
15	F	F	F	R
16	R	F	F	R
17	F	R	R	F
18	R	R	R	F
19	R	R	F	F
20	F	F	F	R
21	F	F	R	F
22	F	R	R	F
23	R	F	R	F

SP 2

	1	2	3	4	5	6	7	8
0	F	R	F	F	F	F	R	R
1	F	F	R	F	R	R	F	R
2	F	F	F	F	R	F	F	R
3	F	F	F	F	R	F	R	F
4	R	F	F	F	F	F	R	F
5	R	F	F	R	F	R	R	R
6	R	F	F	R	F	F	F	R
7	F	R	F	F	F	R	R	F
8	F	F	F	F	F	R	F	F
9	F	F	F	F	F	F	F	F
10	R	R	R	R	R	R	R	R
11	F	R	F	F	F	R	R	R
12	R	R	R	F	R	R	F	R
13	F	F	R	F	F	R	F	F
14	F	F	R	F	F	F	F	F
15	F	F	F	F	F	F	R	R
16	R	F	F	F	R	R	F	R
17	F	F	R	F	R	F	F	R
18	R	F	R	F	R	F	F	F
19	F	R	R	R	F	R	F	F
20	F	F	F	F	F	R	R	F
21	F	F	F	F	R	R	R	F
22	R	F	R	F	R	F	F	F
23	R	F	R	F	F	F	R	F

SP n

	1	2	3	4	5	6
0	R	R	F	F	F	R
1	R	F	F	F	F	R
2	R	F	F	F	R	R
3	R	R	F	R	F	F
4	F	F	F	R	F	F
5	F	F	R	F	R	F
6	F	F	R	F	R	F
7	F	F	R	F	F	F
8	F	R	F	R	F	R
9	F	R	F	R	F	R
10	F	F	F	F	F	R
11	R	F	F	F	R	R
12	R	F	R	F	R	F
13	R	F	F	R	F	R
14	R	R	F	R	F	R
15	F	R	F	R	F	F
16	F	R	F	F	F	F
17	F	R	R	F	R	F
18	F	F	F	F	R	F
19	R	F	F	R	R	F
20	R	R	R	R	R	R
21	F	R	F	F	R	R
22	F	R	F	F	F	R
23	R	R	R	R	F	R

Figure 14 Service providers' three dimensional array

3.3.1 Building the Consolidated Availability String (CAS)

CAS (as shown earlier on figure 7) is an integrated string that consists of three main functionalities: First function, it is a graphical input for the user to choose parking hours from, which replaces the traditional drop-down list that we see on most online applications, forms, and other types of pages. Second function, it is a graphical representation to show available parking spots in all near-by parking providers in two colors: green if available or grey if not available before any selection made by the user. Third function, while the user select each individual hour, it always updates results simultaneously in real-time with the captured available spots from various database sources then it keeps that hour reserved until the user adds another hour or taps on the reserve button.

Having a unique three dimensional array makes it easier for the system to build CAS. The system will go hour by hour to find the available hours, based on all of the results. If one free hour at least were found among all spots at service providers at any specific hour then the first cell of CAS would be green otherwise it will be grey. The method will build up CAS by going though the next twenty-four hours for all of the service provider results obtained. Figure 15 below shows the algorithm that builds CAS.

The algorithm starts with declaring pointers, counters, and CAS string. After that, the first condition checks the time counter, if the CAS array cells has been successfully built, if so then it will show it on the screen, otherwise the loop will check for each service provider for any free spot cross them all one-by-one. If any free hour was found, then the hour cell on CAS will be set to free, otherwise the parking pointer will increase by one to check for free hour in the next parking provider. Another pointer after that checks for the total number of spaces per parking facility, if it doesn't reach the total number then it well go back to check for another parking spot in the same parking facility, otherwise it will resets the parking pointer and will increase the service provider pointer in order to move to the next parking facility. Before this method ends, one last check is made to make sure if the total number parking facilities has been reached, if not, it will do the related pointer resets and increases the time counter.

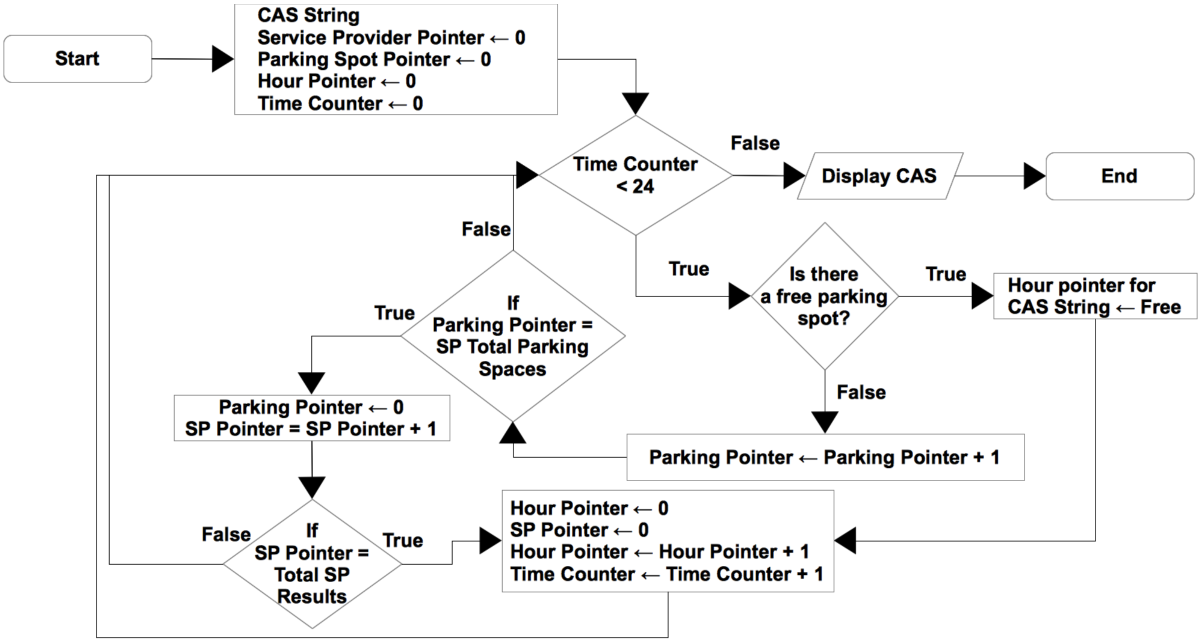

Start
CAS String
Service Provider Pointer ← 0
Parking Spot Pointer ← 0
Hour Pointer ← 0
Time Counter ← 0
Time Counter < 24
False
Display CAS
End
False
True
Is there a free parking spot?
True
Hour pointer for CAS String ← Free
If Parking Pointer = SP Total Parking Spaces
True
Parking Pointer ← 0
SP Pointer = SP Pointer + 1
False
Parking Pointer ← Parking Pointer + 1
False
If SP Pointer = Total SP Results
True
Hour Pointer ← 0
SP Pointer ← 0
Hour Pointer ← Hour Pointer + 1
Time Counter ← Time Counter + 1

Figure 15 Building the CAS algorithm

3.3.2 Selecting Hour and Finding a Spot

At this point, the user can only see the parking spots that are available, and has not yet made any selections. The user then needs to tap on an hour, or more, to make his/her choice. Each tap on any hour will activate a number of methodologies, starting from this algorithm that reduces human errors by only allowing the user to select an individual hour or a chain of hours. Figure 16 shows the algorithm for the user’s hour input.

Here we have six probabilities, each is represented by a condition statement: The first condition is to check if the this is the first tap, which means the first hour selected by the user. In this case the time-in and time-out variables will be set to the same hour. The second condition is to check if this input is to remove that one individual hour, if so it will deactivate the confirmation button. The third probability is that the user wants to add another hour after a previous choice, and then the time-in would increase. The fourth condition if the user wants to add an hour before a selected hour, this will result in changing the time-in to that specific tapped hour. The fifth condition will test if the user has a chain of hours and wants to reduce the period of his parking then time-out will get a new value by decreasing one hour. The last condition is similar to the fifth except that the user decided to delay his time-in, which will change accordingly. These six conditions restrict any other taps on CAS, resulting to accurate single hour or chain of hours of selection.

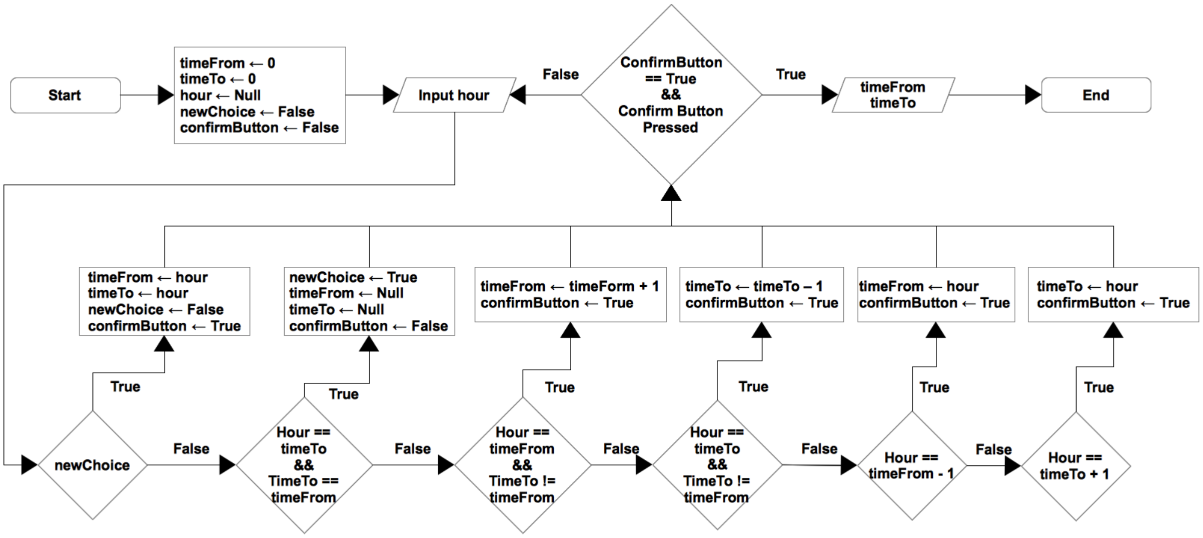


Figure 16 Hour selection algorithm

To find a spot, the following algorithm shown in figure 17 is activated once the user selects an hour as we mentioned earlier.

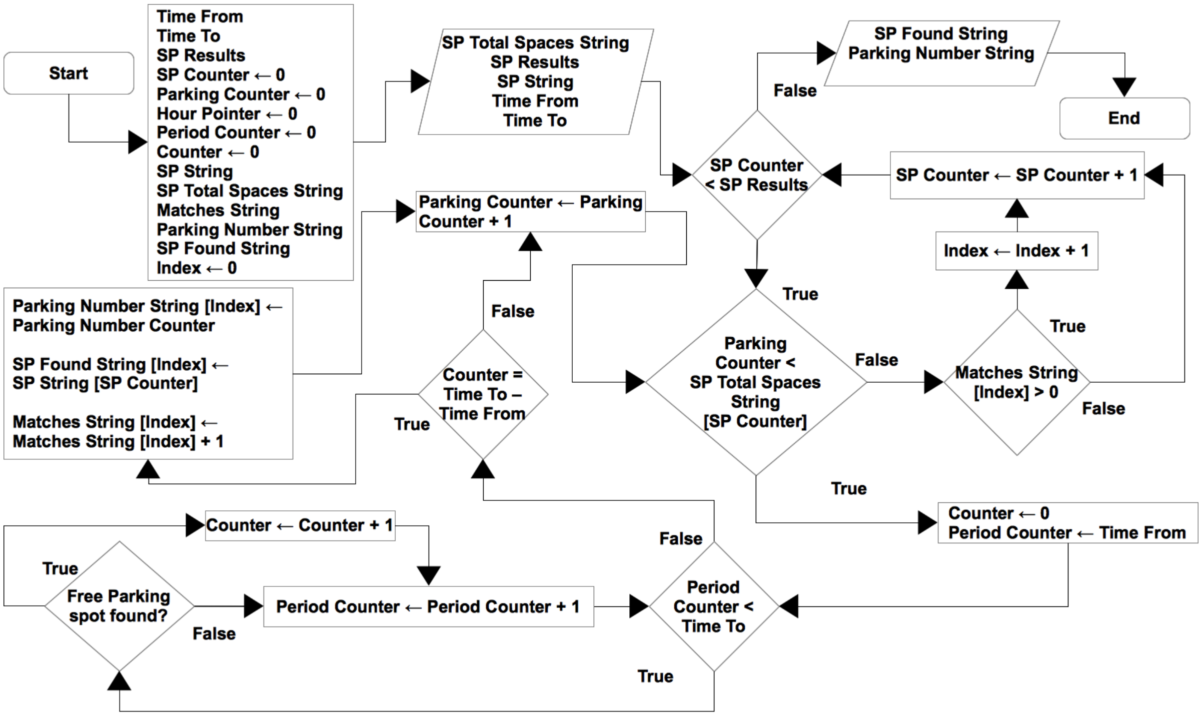


Figure 17 Finding a parking spot algorithm

The algorithm chooses the service providers one-by-one, in order to find an individual hour or a chain of hours, based on the user's input. If the user selects one hour, then the algorithm provides two outputs, which are the ID's of one or more service providers, along with their parking numbers, stored in a string for the next step. If the user adds another hour to their selection, then the system will update the results according to the new inputs.

To understand the mechanism behind this algorithm, it has few loops: The first loop is to check if the number of service providers found has reached the algorithms service provider's counter, if it is still searching it will go to the next loop, if not, then it will store an array that has the service provider ID and the parking number of any spot found. The next major loop will check for the number of spaces for each service provider found, if not completed it will reset the counter and move to the third loop, otherwise it will move to the next service provider and will increment an index used for counting. The third loop condition will test a found hour but still has to check if it's a connected chain of hours. So in case it's a period of time selected then the period counter will still work under this loop, however, if not then it will check if the counter reaches the number of the selected hours, if so, then the corresponding arrays will record the number of parking and the service provider number, if no, then the loop will still we active for the next parking spot.

3.3.3 Parking Service Provider Price Calculation

Price calculation starts once we have the service provider ID and the parking number. Reservation time-in and time-out are also used as inputs to run another query and bring the data needed for the pricing algorithm from each service provider. A generic table of pricing and criterion for a parking facility can be seen in Table 3.

Time Period	Price/ Hour	Max./ Day
6am - 7:59am	$0.75	$15
8am - 4:59pm	$1	
5pm - 11:59am	$0.75	
12am - 5:59am	$0.50	

Table 3 Parking facility - pricing and criterion

The data collected from the database contains the following information: service provider ID, rate period (time from and time to), price per hour, and max. per day. By using the collected data, the system then calculates the final price for each service provider, taking into account any criterion such as max. per day or max. per weekend, and stores it. The next figure shows the pricing algorithm.

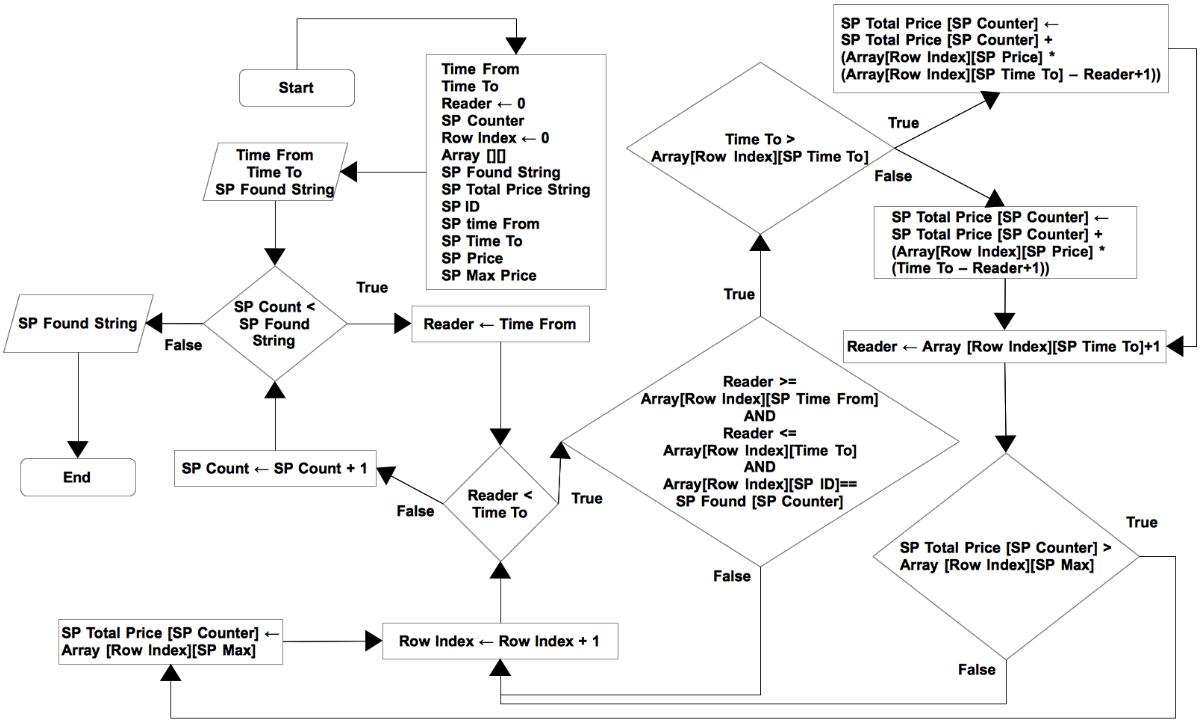

Start
Time From
Time To
Reader ← 0
SP Counter
Row Index ← 0
Array [][]
SP Found String
SP Total Price String
SP ID
SP time From
SP Time To
SP Price
SP Max Price
Time From
Time To
SP Found String
SP Count <
SP Found
String
True
False
SP Found String
End
Reader ← Time From
SP Count ← SP Count + 1
Reader <
Time To
False
True
Reader >=
Array[Row Index][SP Time From]
AND
Reader <=
Array[Row Index][Time To]
AND
Array[Row Index][SP ID]==
SP Found [SP Counter]
True
False
Time To >
Array[Row Index][SP Time To]
True
False
SP Total Price [SP Counter] ←
SP Total Price [SP Counter] +
(Array[Row Index][SP Price] *
(Array[Row Index][SP Time To] – Reader+1))
SP Total Price [SP Counter] ←
SP Total Price [SP Counter] +
(Array[Row Index][SP Price] *
(Time To – Reader+1))
Reader ← Array [Row Index][SP Time To]+1
SP Total Price [SP Counter] >
Array [Row Index][SP Max]
True
False
SP Total Price [SP Counter] ←
Array [Row Index][SP Max]
Row Index ← Row Index + 1

Figure 18 Pricing calculation algorithm

The first condition once this algorithm begins is making sure that all prices for founded service providers have been calculated. The next step is to check the time-in and time-out using a counter called reader. The condition in this step will test if the pricing has ended to for each service provider, if not finished yet, then the service provider counter will increase for the sake of the next one, otherwise the algorithm continues by checking with the captured data as shown on Table 3. Since every service provider has a different pricing ranges as well as variant pricing periods, the reader is checked if it fits into that specific period of time, if it is in that range then it will check also if the time-out is on the same range of not because the price for differs between a period and another. Two different equations are used to calculate the price as shown on Figure 18. One uses the time-out if the users time-out is within one service provider price range; the other equation uses the end-time of the price period and then loops again to the next period of pricing. Then both equations lead to update the reader counter. Then another condition approaches to check if the calculated price reaches the max price per day, if that was true, then the price for that service provider will be updated accordingly. Lastly, row index counter increases to complete this loop.

3.3.4 Priority Algorithm

The final stage of the system is to provide the best choices to the user. The priority algorithm (Figure 19) takes place to determine which service provider better suits the user's needs based on his/her pre-input order in the priority list. Travel time, distance, and price are other inputs that are considered before making the final decision. If two or more matches have equal values, then the algorithm includes the next item on the priority list to the methodology, and so on.

Starting with the first condition on this algorithm, it tests if more than one service provider available. If in case only one result has been found then it stores that service provider ID, parking number, and total price, otherwise it goes the major switch case. This switch directs the algorithm to choose the right path, in case the user prefers best price, then it will compare the results to store that for him/her. A temporary price variable captures the first service provider's total price. Then a counter increases to check for the next service provider. Each time the algorithm compares if the next service provider's total price is less than the temporary price then it will replace the temporary price and the service provider's ID. In case two service providers have an equal price, it will still store that for another test using the service provider temporary index. Same methodology applies for the time and distance comparisons.

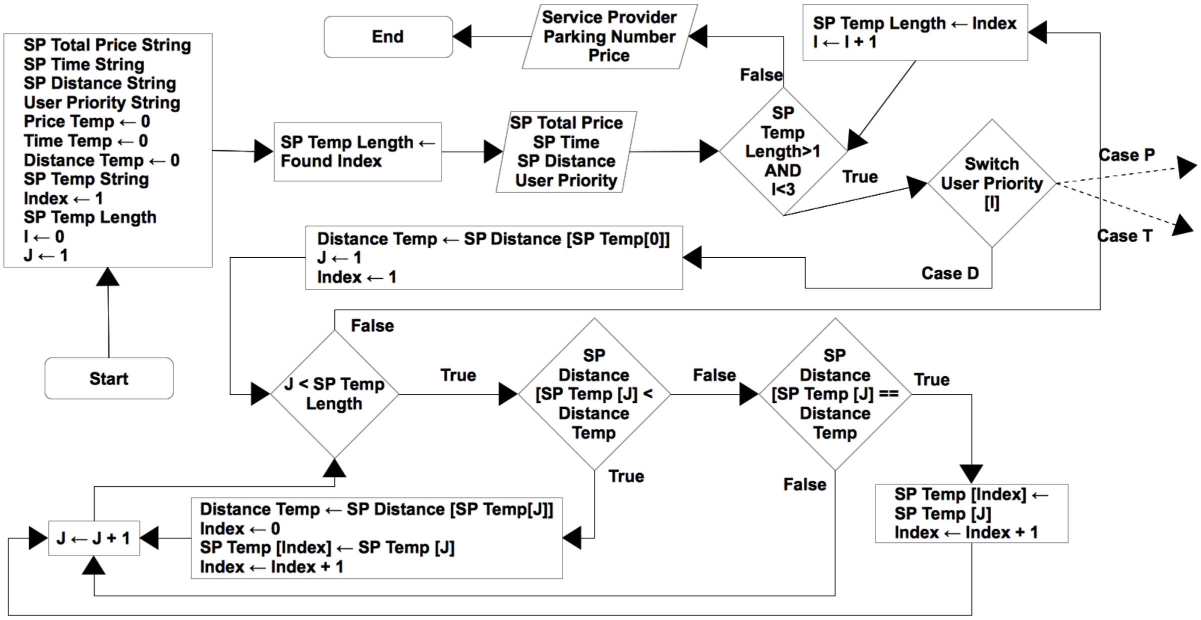
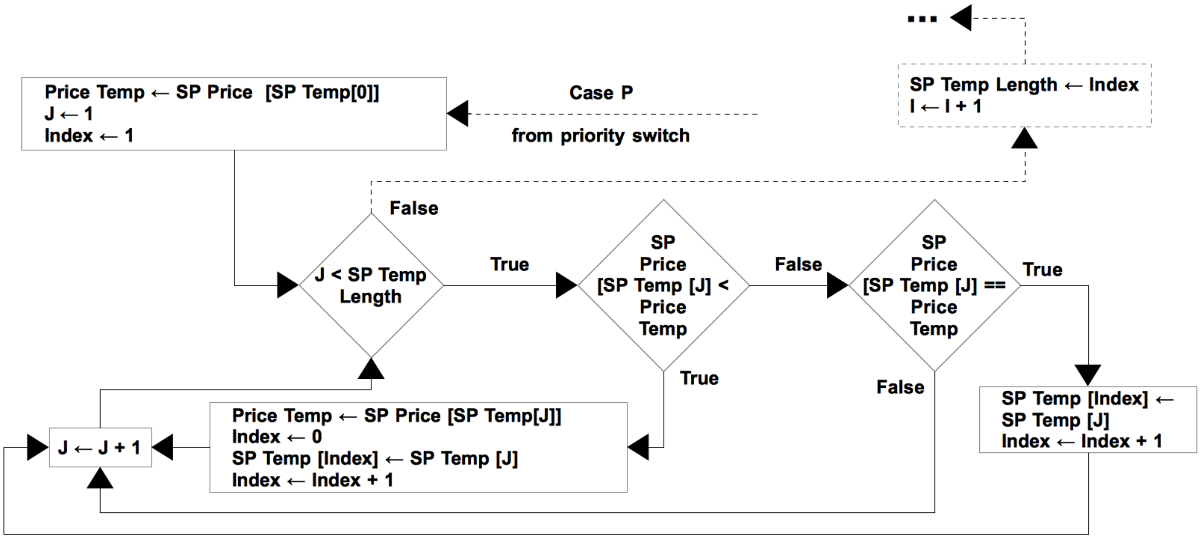


Figure 19 Priority algorithm

Figure 19 shows the switch case for the distance, where Figure 20 and Figure 21 shows price and time cases respectively. Both cases have same functionalities as the price case but only different variables.



...
SP Temp Length ← Index
I ← I + 1
Price Temp ← SP Price [SP Temp[0]]
J ← 1
Index ← 1
Case P
from priority switch
False
J < SP Temp Length
True
SP Price [SP Temp [J] < Price Temp
False
SP Price [SP Temp [J] == Price Temp
True
True
False
SP Temp [Index] ← SP Temp [J]
Index ← Index + 1
Price Temp ← SP Price [SP Temp[J]]
Index ← 0
SP Temp [Index] ← SP Temp [J]
Index ← Index + 1
J ← J + 1

Figure 20 Price case for priority algorithm

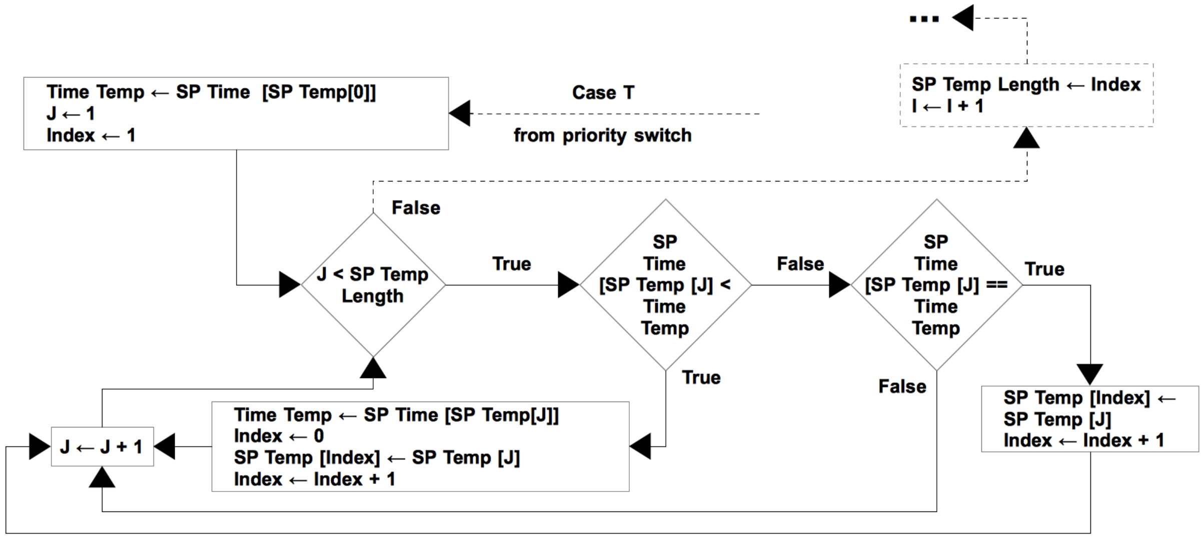


Figure 21 Time case for priority algorithm

3.4 Conclusion

In this chapter, we presented the design of the proposed PRS. We identified the systems inputs and its output from a systematic point of view, and described the methodology of the system's reservation process. In addition, we detailed all of the calculation functions and algorithms, starting from finding a free parking to making a guaranteed parking reservation. In the following chapter, we will present the system's architecture and implementation.

Chapter 4: System Architecture and Implementation

4.1 Proposed Model

In this chapter, we will be present our Parking Reservation System and its key characteristics. The proposed model has three main components: customers/end-users, middleware, and parking service providers. Customers have to log-in to the proposed PRS in order to access information or to make a new reservation. Each autonomous parking service provider subscribed to the system provides parking details such as location and pricing, as well as any other facility details, to the middleware. The middleware is the link between the PRS and all service providers. After consulting with service providers, it prepares a list of parking options satisfying the customer's request, as shown in Figure 22.

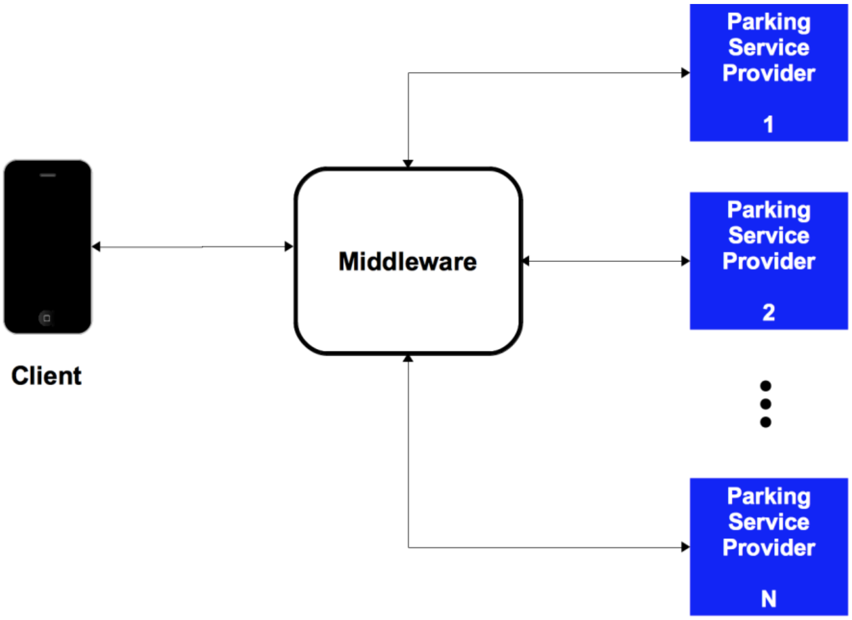

Parking Service Provider 1
Client
Middleware
Parking Service Provider 2
Parking Service Provider N

Figure 22 The parking reservation system's overall architecture

The service provider represents all parking facilities that have a workstation connected to the system.

4.1.1 Parking Reservation System Components

In this section, we will describe the three main components of our parking reservation system: The client who is using the mobile device, the middleware that has webserver and the database server, and finally parking service providers who will have the guaranteed parking spots for their clients. Details about each component as followed:

4.1.1.1 Client

Clients of the system would use their smartphones to access the system. They may access the middleware through the Parking Reservation System installed on their smartphones. This system connects to the middleware using a cellular network such as 3G or 4G data networks. Since the PRS is a mobile based application, the smartphone manages all the functionalities based on its hardware capabilities and all the data related to interactions between the PRS and the mobile system, such as user profile information, car information, parking availability, etc. are stored in the middleware zone. Clients have to be connected to the network in order to get all the up-to-date and real-time information.

4.1.1.2 Middleware

The middleware zone is the hub between clients and parking service providers. It has two hardware components: The web server, which is an Apache HTTP server, and the data base (DB) server. The web server is the only gate to the DB that stores all the information for both ends. All data will be synchronized between both ends (the client and the service provider workstations), and through the web server. The web server uses software called PHPMyAdmin to administrate the database server.

The web server will be handling all administrations in MySQL DB in order to access the DB. Since the user interface is a mobile application based system, there is no web interface required for the client. Thus, the middleware works mainly as a data store and synchronizes its data with both ends simultaneously, in real-time. Figure 23 shows the breakdown of the middleware.

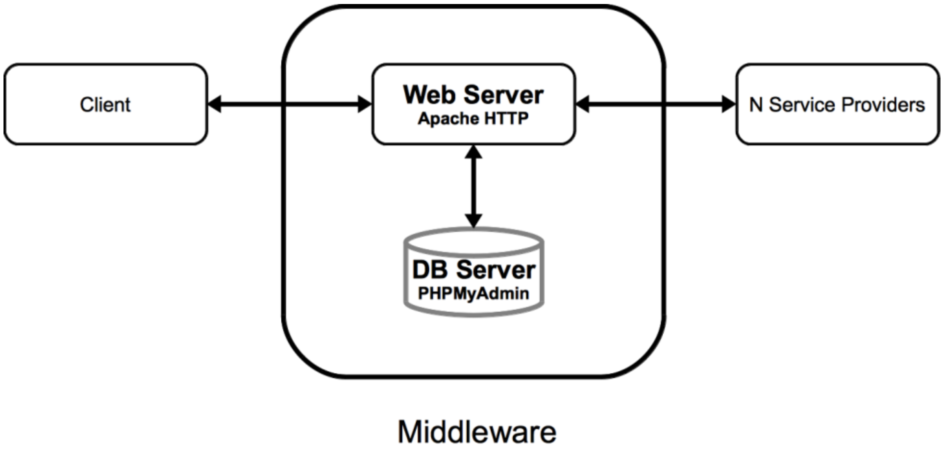


Figure 23 Middleware of the parking reservation system

4.1.1.3 Parking Service Provider

Parking lot providers have two hardware device types on their sites. The first is a workstation that is used for administration proposes such as to update the pricing, closures, or any site-related information. The second hardware is an optional automated car sensor that determines the status of each spot, as it is placed in front of each parking space. With the help of the sensors, which provide a true/false signal for each spot, the workstation syncs all updated parking spaces on its site. Sensors are cable connected to the workstation, and the workstation is connected to the Internet (wired or wirelessly). The other option for the service provider is that the system administrator manually checks the number of spots available at the location and updates the vacant parking spaces using the workstation at the site. The choice of setup is at the service provider’s discretion. Figure 24 shows the breakdown of the technical service provider environment.

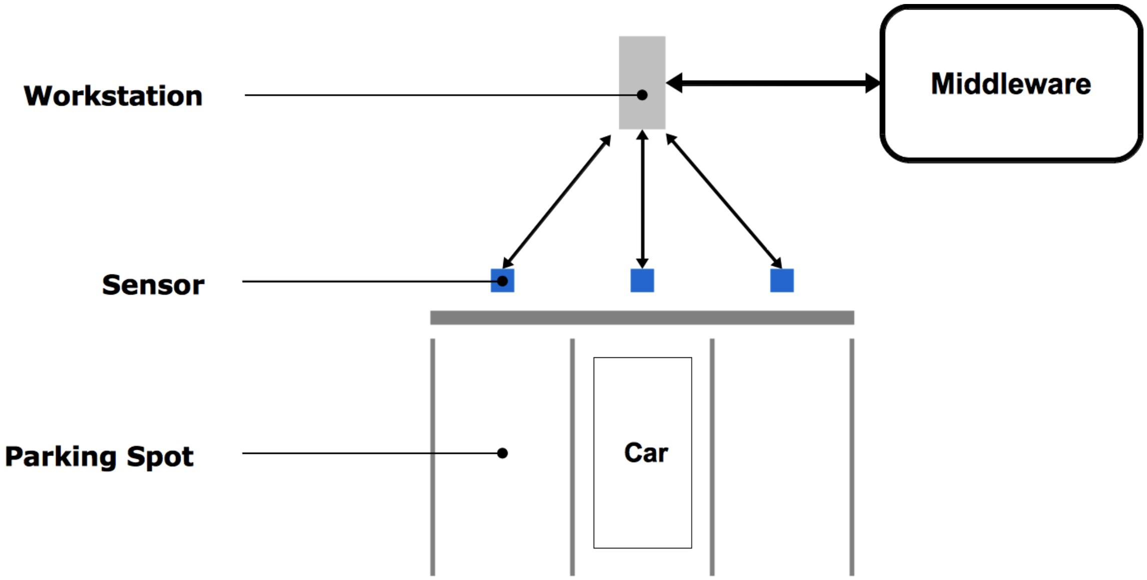


Figure 24 Service provider component breakdown

4.1.2 Parking Reservation System Requirement

In this section, we will briefly explain the system requirements for the overall system. Table 4 shows each zone and its system components:

	Client	Middleware	Service Provider
Hardware	iPhone	PC or server	Workstation, sensors
Software	Proposed PRS	Apache server PHPMyAdmin MySQL	Internet browser to communicate with the middleware

Table 4 System requirements

4.1.3 Prototype Model

As a proof of concept, and in order to implement our system in a real-time experimentation, we used the Apple Inc. Software Development Kit (SDK) Xcode version 6.1. This tool works simultaneously with an SQL database engine SQLite, running under Mozilla's Firefox web browser, to execute SQL queries from the library in real-time. After the implementation, we used an iPhone for testing purposes.

4.2 Database Design and Tables

The following Figure 25 shows the entity relationship diagram for the database used in our system:

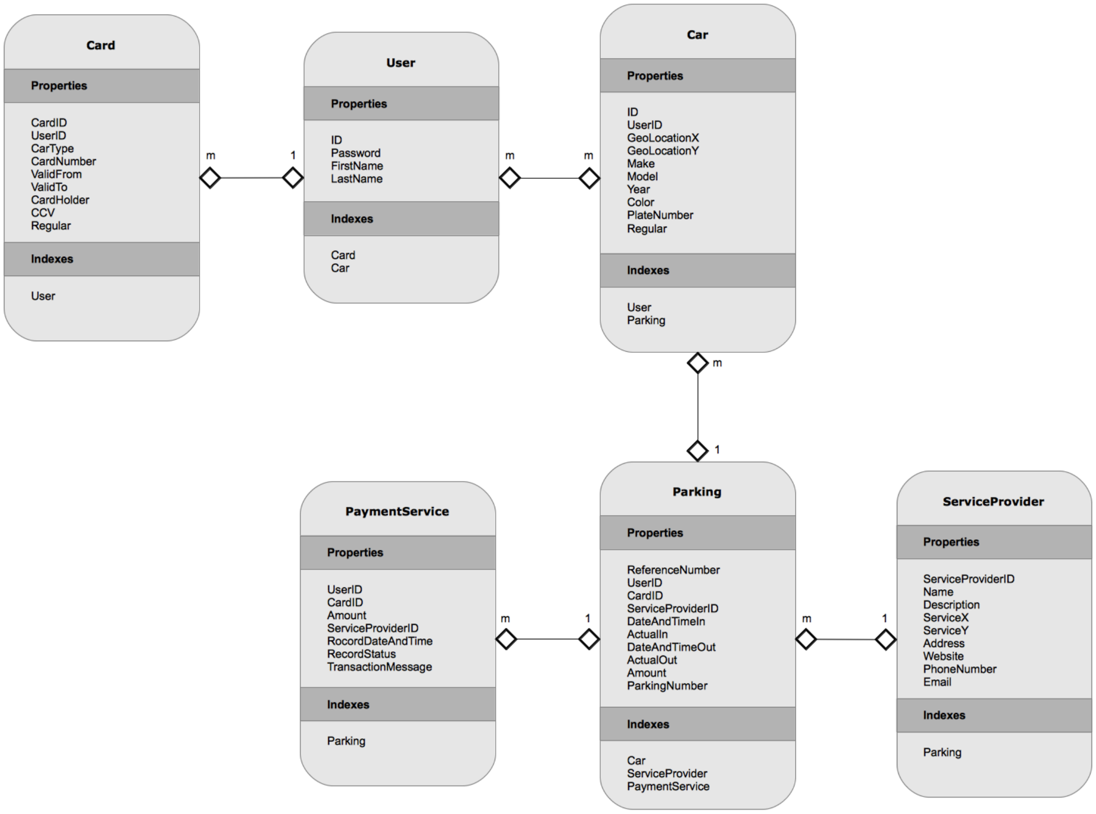


Figure 25 Parking reservation system's ERD

4.2.1 Database Tables

In this section, we will describe the tables used in our reservation system.

4.2.1.1 Database Table: ServiceProvider

The ServiceProvider DB table contains all of the necessary information about the various service providers, each of which has a unique ID. This table has a many-to-many relationship with the users in our system. Detailed information is shown in Table 5.

ServiceProvider		
Field Name	**Data Type**	**Description**
ServiceProviderID	INTEGER	Service provider’s ID
Name	CHAR(50)	Parking provider’s name
Description	TEXT	Description of the service provider such as parking location name, etc.
ServiceX	DECIMAL(24,20)	Geolocation altitude of the service provider
ServiceY	DECIMAL(24,20)	Geolocation longitude of the service provider
Address	TEXT	Full address for the service provider
Website	CHAR(100)	URL for the service provider
PhoneNumber	CHAR(16)	Service provider’s phone number
Email	CHAR(50)	Service provider’s email address

Table 5 ServiceProvider DB table

4.2.1.2 Database Table: PaymentService

The PaymentService DB table records the information for every payment made, regardless of the status of the transaction. Detailed information about this table is shown in Table 6.

PaymentService		
Field Name	**Data Type**	**Description**
UserID	INTEGER	User ID
CardID	INTEGER(5)	Card ID
Amount	INTEGER(5)	Payment amount to the financial institution
ServiceProviderID	FLOAT(5,2)	Service provider’s ID
RecordDateAndTime	INTEGER(4)	Transaction date and time
RecordStatus	CHAR(10)	Accepted or declined
TransactionMessage	CHAR(255)	Details about the transaction

Table 6 PaymentService DB table

4.2.1.3 Database Table: Card

The Card DB table holds the user payment accounts such as credit cards. Detailed information about this table is shown in Table 7.

Card		
Field Name	**Data Type**	**Description**
CardID	INTEGER	Card ID
UserID	INTEGER	User ID
CardType	CHAR(20)	Type of card, such as: VISA, MasterCard, American Express, etc.
CardNumber	CHAR(20)	Card number
ValidFrom	DATE	Validity date from
ValidTo	DATE	Expiry date
CardHolder	CHAR(30)	Name of card holder
CCV	INTEGER(5)	Card Code Verification of the card
Regular	INTEGER	Indicates the default card

Table 7 Card DB table

4.2.1.4 Database Table: Parking

The Parking DB table is used to create a record of every parking provider; it saves a record once a reservation is made, but will also a record each time a user enters or exits a parking lot. Detailed information about this table is shown in Table 8.

Parking		
Field Name	**Data Type**	**Description**
ReferenceNumber	INTEGER	Parking reference number for the confirmed booking
UserID	INTEGER	User ID
CardID	INTEGER	Card ID
ServiceProviderID	INT(4)	Service provider's ID
DateAndTimeIn	DATETIME	Reservation check-in date and time
ActualIn	DATETIME	Actual date and time of arrival
DateAndTimeOut	DATETIME	Reservation check-out date and time
ActualOut	DATETIME	Actual date and time of departure
Amount	FLOAT(5,2)	Amount paid by the user for the reservation service
ParkingNumber	INTEGER	Parking number in the parking facility

Table 8 Parking DB table

4.2.1.5 Database Table: Car

The Car DB table contains essential information about the vehicles of all the users. Detailed information about this table is shown in Table 9

Car		
Field Name	**Data Type**	**Description**

ID	INTEGER	Car ID
UserID	INTEGER	User ID
GeoLocationX	DECIMAL(24,20)	Geolocation altitude of the vehicle’s current position
GeoLocationY	DECIMAL(24,20)	Geolocation longitude of the vehicle’s current position
Make	CHAR(20)	Make of the car
Model	CHAR(20)	Model of the car
Year	YEAR	Year of manufactor
Color	CHAR(10)	Color of the car
PlateNumber	CHAR(10)	License plate number
Regular	INTEGER	Indicates the default car

Table 9 Car DB table

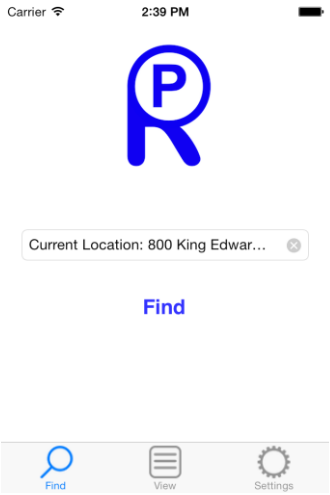
4.2.1.6 Database Table: User

The User DB table has the user’s information that allows them to log into the system. Detailed information about this table is shown in Table 10.

User		
Field Name	**Data Type**	**Description**
ID	INT(5) AUTO_INCREMENT	User ID
Password	CHAR(50)	User password
FirstName	CHAR(20)	User first name
LastName	CHAR(20)	User last name

Table 10 User DB table

4.3 User Interface



Current Location: 800 King Edwar…

Figure 26 Main tab bar

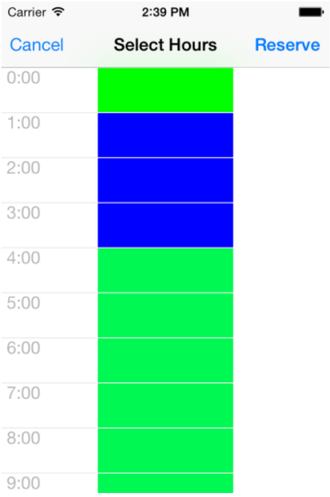
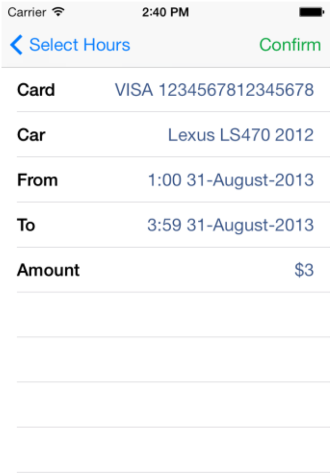


Figure 27 CAS



Card	VISA 1234567812345678
Car	Lexus LS470 2012
From	1:00 31-August-2013
To	3:59 31-August-2013
Amount	$3

Figure 28 Confirmation screen

As shown in Figure 26, our parking reservation system has three main tabs to choose from: find, view, and settings.

4.3.1 Find Tab

When a user wants to make a reservation, the first step is to find a location either by current location, which will be filled automatically by the system (this is the default configuration), or by tapping the text field to enter the desired location. There are two strategies for the calculations behind these two different approaches. We will address the differences below.

4.3.1.1 Current Location

If the user wants to find parking near his current location, he/she can immediately tap on find; the system will then locate several service providers in a certain range (i.e. one kilometer radius for this work). In this case, the distance and time are calculated from the current geographical point to the location of the service providers that have been found. In this case, the user is travelling by car from his current location to the parking lot, are we only need to measure the distance between two points as shown in Figure 29.

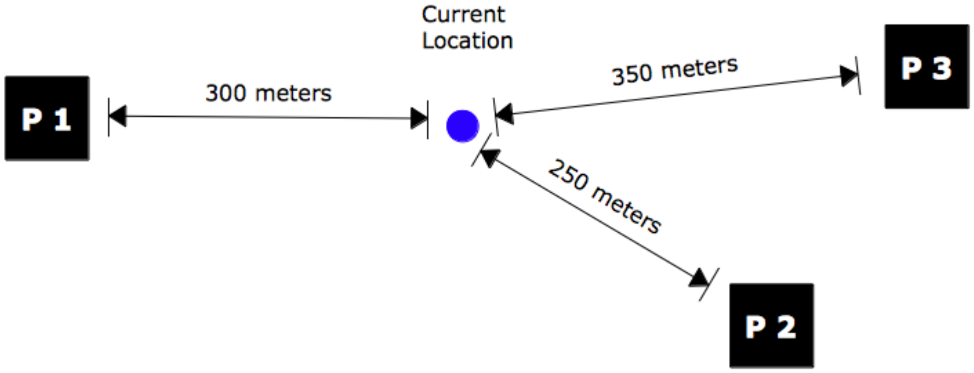


Figure 29 Current location distance and time calculation

4.3.1.2 Entered Address

The user has the option of entering a specific address in order to find parking around that location. In this case, there are tree points (two distances) to measure: the user's current location, the service provider's location, and the destination address. The time and distance calculated are the walking time and distance from that service provider to the final address, as shown in Figure 30, and not the driving time and distance from the current location to the parking service provider. The system will choose the best parking for the user, based on their priority list. Figure 30 illustrates this approach.

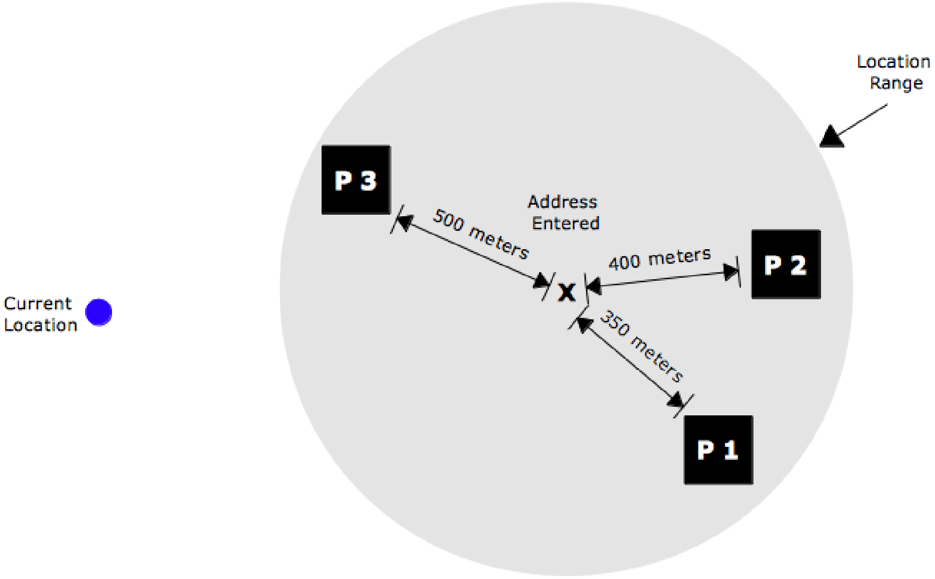


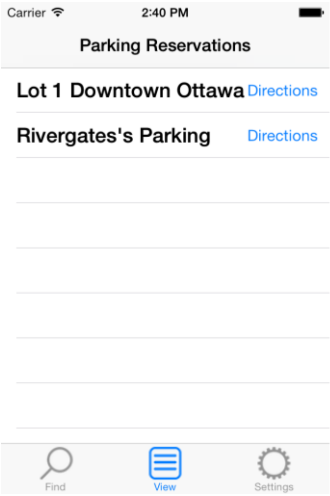
Figure 30 Specified address distance and time calculation

After finding the available parking, the CAS will show the available parking hours in green and the unavailable hours in grey. If the user selects an hour, it changes to blue, meaning that this hour is guaranteed if the user later decides to reserve it. Figure 27 shows the CAS hour selection.

The next step is the reservation confirmation. The user can see which payment card will be used, as well as which vehicle, the parking duration, and the total price for the parking, as in Figure 28.

Once the user confirms the parking reservation, he/she is able to check it at anytime from the view tab.

4.3.2 View Tab

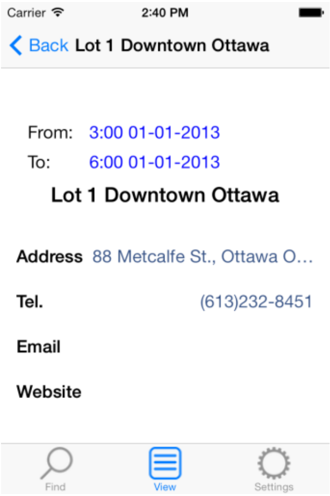


Parking Reservations

Lot 1 Downtown Ottawa Directions

Rivergates's Parking Directions

Figure 31 View reservation



Lot 1 Downtown Ottawa

From: 3:00 01-01-2013
To: 6:00 01-01-2013

Lot 1 Downtown Ottawa

Address 88 Metcalfe St., Ottawa O...
Tel. (613)232-8451
Email
Website

Figure 32 Reservation details

Figure 31 shows the view tab with all the upcoming reservations. The user can easily tap on Directions, and the system will open the maps application. If the user wants to have more information regarding the service provider, for instance their phone number, website, address, etc., in addition to the reservation period, then he/she can tap on the facility name. Figure 32 shows the details of the service provider and the reservation.

4.3.3 Settings Tab

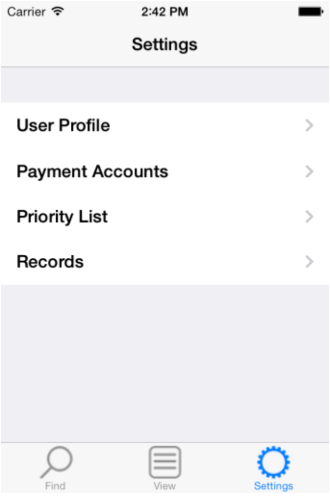
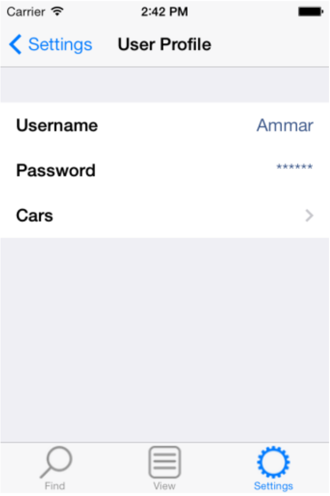


Figure 33 Settings tab



Username Ammar

Password ******

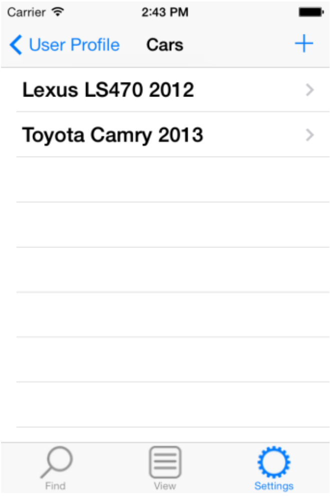
Figure 34 User profile

The settings tab on Figure 33 contains the most valuable information for the user. Details about each of the options are as follows:

4.3.3.1 User Profile

The user profile consists of the user name and the password, as well as the cars linked to the user, as seen in Figure 34. The user can also add, edit, or delete a car. Figure 35 shows how the user sees his/her car(s).

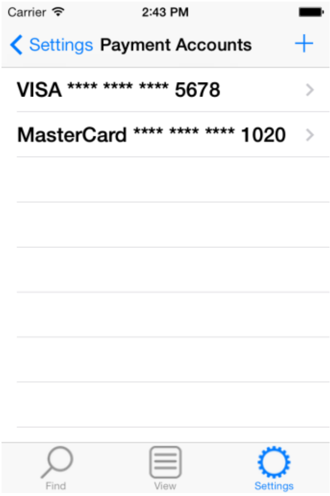
4.3.3.2 Payment Accounts



Lexus LS470 2012

Toyota Camry 2013

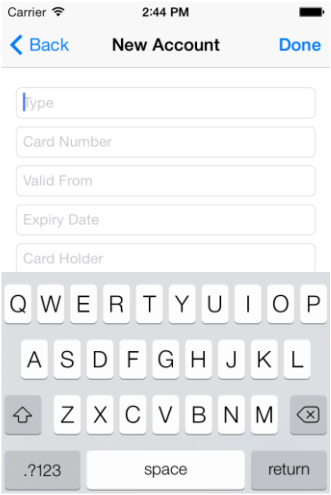
Figure 35 Car list



VISA **** **** **** 5678

MasterCard **** **** **** 1020

Figure 36 Cards



Type

Card Number

Valid From

Expiry Date

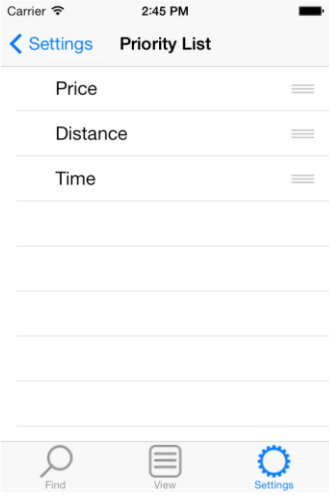
Card Holder

Figure 37 Adding new card

In this list, the user can add one or more payment accounts to make a direct payment to the parking provider. In Figure 36, we can see the list of cards. The user is able to add a payment account by tapping on the plus sign. Figure 37 shows the screen displayed when adding a new card.

4.3.3.3 Priority List

The priority list is a critical feature of this system as it is essential to the search methodology, and the suggested parking is based on this valuable input. The user can modify the order of his/her priorities by sorting the priority list; the top choice will be considered the highest priority, then the next, then the one after, as shown in Figure 33.

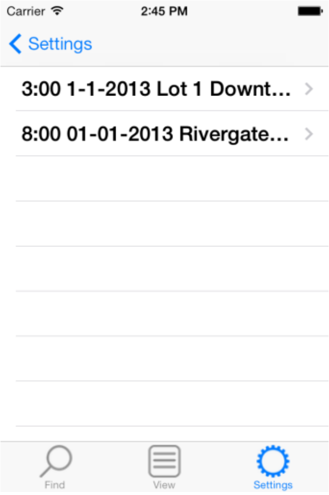


Price

Distance

Time

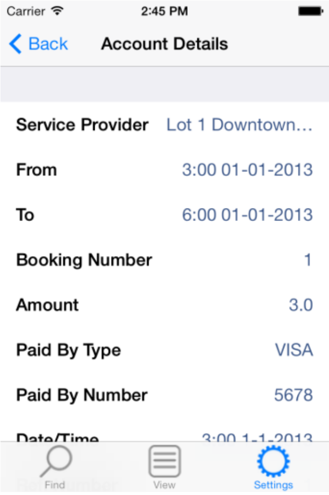
Figure 38 Priority list



3:00 1-1-2013 Lot 1 Downt…

8:00 01-01-2013 Rivergate…

Figure 39 Records



Service Provider	Lot 1 Downtown…
From	3:00 01-01-2013
To	6:00 01-01-2013
Booking Number	1
Amount	3.0
Paid By Type	VISA
Paid By Number	5678
Date/Time	3:00 1-1-2013

Figure 40 Record details

4.3.3.4 Records

This tab shows all of the historical transactions made, whether or not it was successful. Figure 39 shows the list of records made, and Figure 40 has the details of a specific record.

4.4 Conclusion

Our goal for this chapter was to describe the system architecture and the major blocks of it’s components. Moreover, we discussed both hardware and software requirements for each component. We also explained the characteristics of the user interface. In the next chapter, we will see how this user interface facilitates the usability of the system, in order to increase the number of drivers using the system.

Chapter 5: System Usability and Experimental Results

In this usability test, we were able to gather the information in order to measure the quality of our system compared to current informative/reservation parking systems.

5.1 Objectives

The goals of this experiment are to: (a) evaluate the methodologies we used to perform a complete task scenario of a typical parking reservation, to (b) give the participants the opportunity to rate their experience on the different approaches between our proposed text-mode PRS and the traditional map-based systems available online or on smartphone applications, which is Best Parking [26], and lastly, to (c) measure their satisfaction.

The objective of this usability test is to let the user experience the two different approaches: map-based parking reservation systems and our proposed text-mode parking reservation system. We will then measure the usability of both systems, to determine how satisfied they were with both of them.

5.2 Data Collection Scenario

In our study, we conducted a questionnaire consisting of four parts. The first part was an optional section for participants who drive a car. The second part of the questionnaire was in relation to their experience using one of the map-based parking reservation systems, and the third part was in relation to their hands-on experience with our PRS. Once the participants finished all of the previous parts, they completed the fourth part, which was a comparison between the two systems.

Appendix A includes all of the questions of our usability test.

Our questionnaire contains a variety of types of questions. In the first optional part, we used nominal questions in order to examine how familiar the participants were with the currently available online parking services. In the second and third part, we measured the quality of both the map-based systems and our proposed PRS, using the Likert scale [29]. Typical descriptors are: (1) “Strongly disagree”, (2) “Disagree”, (3) “Neither agree nor disagree”, (4) “Agree”, (5) “Strongly agree”. In the last part of our questionnaire we used comparison questions containing the following items: (a) “System A”, (b) “System B”, (c) “Another system I used before”, and lastly (d) “None”, where System B is our implemented system.

5.3 Participants

In total, there were 15 participants in our usability test. The age range was between 25 and 59 years, and the group had different academic backgrounds. There were 8 males and 7 females, and all participants were able to perform all tasks assigned to them on both systems. Since not all participants drive, only 12 of the 15 users participated in the optional questionnaire, and we will refer to them later on as the drivers group.

5.4 Evaluation and Results

5.4.1 Familiarity with Online Parking Systems

We started our questionnaire by measuring the drivers’ familiarity with online parking systems. The results show that more than 65% of participants in the drivers group don’t use any online parking website or reservation mobile application to find parking locations; a third of them claim that they don’t because they don’t know of a specific website or a mobile application. Only 25% of the group often used online informative parking or parking reservation systems. As little as 25% of this group generally search for a parking to get directions ahead of their actual trip, where 50% were neutral. In addition, we found that 75% of the group have never tried online parking reservations, and none of the drivers who actually tried one have used it to find various locations.

From these valuable inputs, obtained from the drivers group, we can see that a high percentage of them are still not using online parking information or reservation systems. For this reason, we think that more versatile systems are required, to capture driver attention and attract them to these services, as it has a direct impact on reducing many issues related to traffic, safety, and the environment.

5.4.2 Experimental Tests

After the participants performed the experiment using the reservation scenarios in both systems, the results showed that their experience with each system differed significantly. We analyzed the results using a method called an average agreement of data, where 5 is strongly agree, and 1 is strongly disagree, on each of the elements being measured. Figure 41 below is a representation of the results, in averages.

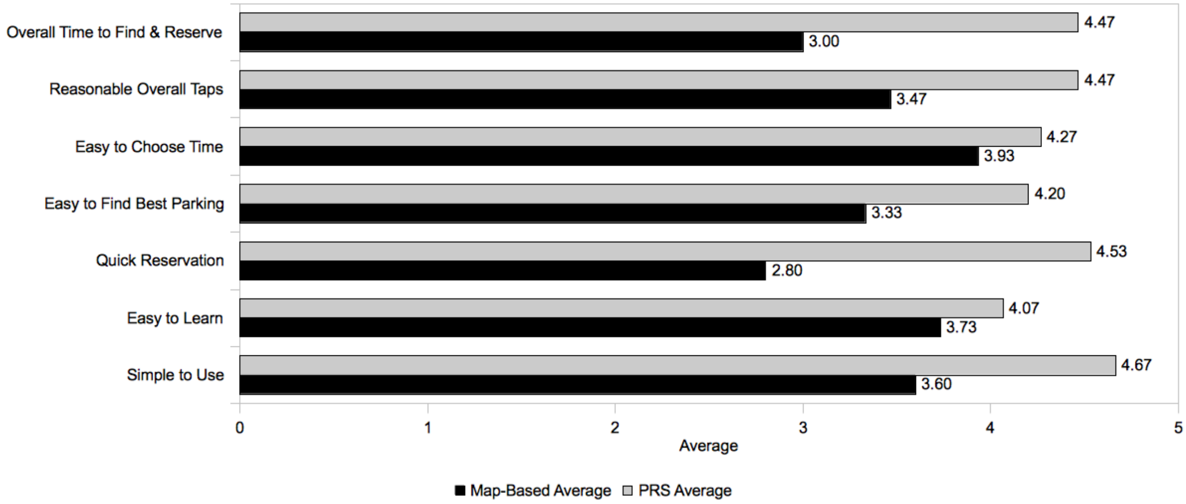


Figure 41 Experimental responses between the map-based system and our PRS

It is obvious how a hands-on test can influence the usability in many different aspects of an online parking reservation system, and our system clearly shows promising experimental outcomes.

5.4.3 Comparison Results

This was the last part of the whole questionnaire, where the participants answered the comparison questions from their own perspective. The objective of this section is to determine which system they prefer based on the variant measurement questions. More than 90% believe that our proposed PRS is more user-friendly and contains less numbers, which helps them achieve their goal. Moreover, more than 85% think that it is faster and more usable to make a reservation with our system. Finally, 80% like our PRS and might use it to find parking; they would use it more frequently than a map-based online system. The following chart (Figure 42) shows the responses based on these system characteristics:

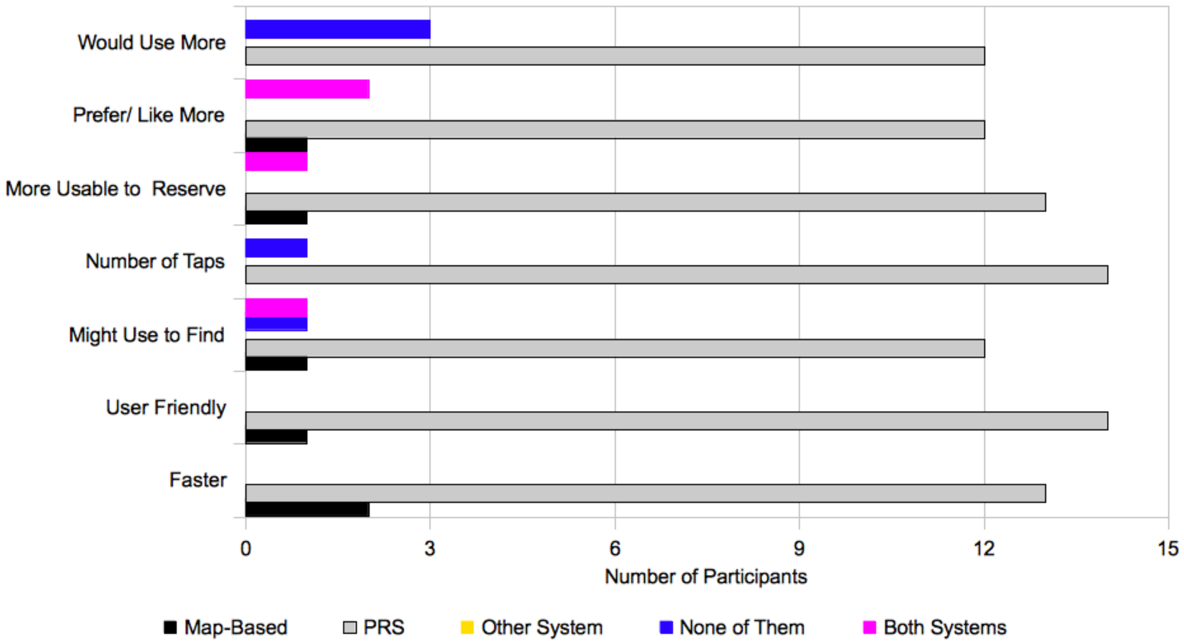


Figure 42 Comparison chart of the map-based system vs our PRS

5.4.4 User Suggestions

Although our PRS score well and received positive responses from most participants, some of them did suggest very useful features and add-ons that they would like to see in the future for a better reservation experience. One participant would like to have additional parameters for more flexibility and therefore better results. Another user suggested that the priority list would be more efficient if it could combine two priorities, for instance distance and time, working simultaneously as one search criterion. An interesting input from a participant was the ability to speak to the system, to make a reservation via voice-command. In addition, one suggested that there should be a map right after the reservation confirmation. Another suggestion is to have additional details or information regarding the reservation itself after confirmation, for example the distance to the reserved parking.

5.5 Conclusion

In this chapter, we believe that a usability test is essential to evaluate our methodologies, do an experimental test, and finally measure the users' satisfaction. In order to achieve this goal we compared our system with a map-based system using a questionnaire to observe how familiar the participants are with online parking services, to measure the quality of both systems, and lastly to generate a realistic comparison between both systems. At the end of the experiment, our findings clearly show that our system's reservation process is more favorable than other online map-based systems.

Chapter 6: Conclusion and Future Work

The need for a parking space in busy cities is becoming a major problem. It is a very challenging task for drivers, who often need to make loops just to find a place to park their vehicles. This time wasting process may result in road accidents and have other environmental impacts. Drivers never know ahead of time what the price or the location are going to be, and they have to take this risk every time they want to park in a lot. For these reasons, in this work, we introduced a reservation system that works in real-time to capture information and transmit this information regarding free parking spaces to the users. We believe that this system would always suggest the best parking to the user, so that he/she would not have to worry about the parking, the price, or the directions. In addition, from a design prospective, the reservation process offers a unique approach that requires the least amount of taps, therefore making the reservation faster and very user friendly.

Even though this proposed system has fast and easy-to-use characteristics, after the positive feedback from the usability study, there is still an opportunity to provide additional services to make it even better. First, the system can feature license plate recognition via the smartphone camera, instead of asking the users to enter their car information manually to add a new car. This feature would ensure accuracy by reducing human error, especially if was integrated with the Ministry of Transportation. Second, an integrated administrative webpage linked directly to the system would ease the parking service provider’s task of registering their facility and adding all features and pricing information into the system. Therefore, the system would function as a single system, serving vehicle drivers and service providers simultaneously. Fourth, the system could feature a reminder, notifying the user if his/her parking is about to expire, and providing an optional parking renewal, if the space is available. Fifth, street parking could become an extension, widening the searching zone outside of the current parking lots. Finally, the system could features not only short-term reservations, but also a long-term parking such as weekly or monthly reservations.

All in all, we hope that this work captures attention and encourages system modelers, designers, and developers to enhance the current state of reservation systems. We also hope that it provides a different prospective for new systems, helping advance the usability and create more intuitive and flawless designs, for a better use of the technology that surrounds us.

References

[1] J. Sousanis. (Aug 15, 2011). *World Vehicle Population Tops 1 Billion Units*. Available: http://wardsauto.com/ar/world_vehicle_population_110815.

[2] F. Kurauchi and Y. Iida, "Simulation Analysis on the Evaluation of Parking Reservation System," 2008.

[3] F. Faheem, S. Mahmud, G. Khan, M. Rahman and H. Zafar, "A Survey of Intelligent Car Parking System," *Journal of Applied Research and Technology,* vol. 11, 2013.

[4] D. Takahashi. (Aug 17, 2011). *Touchscreen market growing 10 times faster than other displays* [Accessed: 2014, Mar 16]. Available: http://venturebeat.com/2011/08/17/touchscreen-market-growing-10-times-faster-than-other-displays/.

[5] (Feb 28, 2014). *Mobile apps overtake PC Internet usage in U.S.* [Accessed: 2014, Mar 20]. Available: http://money.cnn.com/2014/02/28/technology/mobile/mobile-apps-internet/.

[6] Y. Rahayu and F. N. Mustapa, "A Secure Parking Reservation System Using GSM Technology,".

[7] P. Trusiewicz and J. Legierski, "Parking reservation-application dedicated for car users based on telecommunications APIs," in *Computer Science and Information Systems (FedCSIS), 2013 Federated Conference on,* 2013, pp. 865-869.

[8] H. Wang and W. He, "A reservation-based smart parking system," in *Computer Communications Workshops (INFOCOM WKSHPS), 2011 IEEE Conference on,* 2011, pp. 690-695.

[9] P. White. (Feb 27, 2007). *No Vacancy: Park Slopes Parking Problem And How to Fix It* [Accessed: 2014, Nov 15]. Available: http://www.transalt.org/news/releases/126.

[10] D. C. Shoup, "Cruising for parking," Transp. Policy, vol. 13, pp. 479-486, 11, 2006.

[11] European Parliament. (1999). *Computerizer reservation Systems* [Accessed: 2014, Dec 1]. Available: http://www.europarl.europa.eu/workingpapers/tran/105/chap3_en.htm.

[12] Airways News. (). *1950s - Magentronic Reservisor reservations machine at The American Airlines C.R. Smith Museum* [Accessed: 2014, Dec 1]. Available: http://airwaysnews.com/html/museums/american-airlines-cr-smith-museum-ft-worth-texas/1950s-magentronic-reservisor-reservations-machine/845.

[13] G. Ross. (Nov 1, 2013). *Airline Reservation Systems History 101* [Accessed: 2014, Dec 1]. Available: https://www.cbtravel.com/blog/2013/11/airline-reservation-systems-history-101/.

[14] [Accessed: 2015, Jun 21]. Available: http://www.its.leeds.ac.uk.

[15] Yang Jun, "A system framework of active parking guidance and information system," in Information Engineering (ICIE), 2010 WASE International Conference on, 2010, pp. 150-154.

[16] Yanfeng Geng and C. G. Cassandras, "A new “smart parking” system based on optimal resource allocation and reservations," in Intelligent Transportation Systems (ITSC), 2011 14th International IEEE Conference on, 2011, pp. 979-984.

[17] R. G. Thompson and P. Bonsall, "Drivers’ response to parking guidance and information systems," Transport Reviews, vol. 17, pp. 89-104, 1997.

[18] M. Y. I. Idris, Y. Leng, E. Tamil, N. M. Noor and Z. Razak, "Car park system: A review of smart parking system and its technology," Information Technology Journal, vol. 8, pp. 101-113, 2009.

[19] K. Mouskos, J. Tvantzis, D. Bernstein and A. Sansil, "Mathematical formulation of a deterministic parking reservation system (PRS) with fixed costs," in *Electrotechnical Conference, 2000. MELECON 2000. 10th Mediterranean,* 2000, pp. 648-651.

[20] K. Inaba, M. Shibui, T. Naganawa, M. Ogiwara and N. Yoshikai, "Intelligent parking reservation service on the internet," in *Applications and the Internet Workshops, 2001. Proceedings. 2001 Symposium on,* 2001, pp. 159-164.

[21] W. C. Chia and N. Salimi, "A smart parking and reservation system for QR-code-based car park," in Innovation in the High-Tech EconomyAnonymous Springer, 2013, pp. 91-101.

[22] Hanif, Noor Hazrin Hany Mohamad, M. H. Badiozaman and H. Daud, "Smart parking reservation system using short message services (SMS)," in *Intelligent and Advanced Systems (ICIAS), 2010 International Conference on,* 2010, pp. 1-5.

[23] K. Sushma, P. R. Babu and J. N. Reddy, "Reservation Based Vehicle Parking System Using GSM and RFID Technology,".

[24] C. Black and G. Tuccillo, Ground Transportation Internet Reservation System, 2001.

[25] R. L. Schneider, K. D. Turner and K. L. Stavnes, Hotel Database Inquiry System, 1998.

[26] [Accessed: 2014, Dec 1]. Available: www.bestparking.com.

[27] [Accessed: 2014, Dec 1]. Available: https://www.parkingpanda.com/.

[28] [Accessed: 2014, Dec 1]. Available: https://spothero.com/.

[29] (2015, Apr 24). *Likert scale* [Accessed: 2014, Dec 1]. Available: http://en.wikipedia.org/wiki/Likert_scale.

Appendix A

A.1 Participant Information

First Name: ____________________ Last Name: _____________________

Age Range:

a. 15-19
b. 20-24
c. 25-29
d. 30-39
e. 40-49
f. 50-59
g. 60 and above

Gender:

a. Male
b. Female
c. I would rather not specify

Occupation: ______________________________________

A.2 Collection Questionnaire (Optional to Participant)

1- I use online parking website or parking app to find parking locations.

a. Yes, I always do
b. Yes, I usually do
c.Yes, I often do
d. No, I have never used online

2- In general, I search for a parking to get directions ahead of my actual trip.

a. Strongly disagree
b. Disagree
c.Neutral
d. Agree
e. Strongly agree

3- I have tried online parking reservations at least once.

a. Yes
b. No

4- If you are using online parking website(s) or app(s), please specify the reason(s):

a. To find various locations
b. To get directions
c. To compare different prices
d. To find the shortest path to my final destination
e. Other: ____

5- If you are not using a parking website(s) or app(s), please specify the reason(s):

a. I don't know a specific website or an app to find a parking.
b. Parking website(s) or app(s) are not useful.
c. I like to find a parking when I reach my destination.
d. Other: ____

A.3 Map-Based System Usability Task Scenario Questionnaire

1- The system was simple to use.

a. Strongly disagree
b. Disagree
c. Neutral
d. Agree
e. Strongly agree

2- It was easy to learn how to use the system.

a. Strongly disagree
b. Disagree
c. Neutral
d. Agree
e. Strongly agree

3- I was able to make the reservation (if available) quickly using the system.

a. Strongly disagree
b. Disagree
c. Neutral
d. Agree
e. Strongly agree

4- I was able to find the best parking of my choice easy.

a. Strongly disagree
b. Disagree
c. Neutral
d. Agree
e. Strongly agree

5- Choosing the arrival time and departure time was easy.

a. Strongly disagree
b. Disagree
c. Neutral
d. Agree
e. Strongly agree

6- The number of taps to find and reserve a parking (if applicable) was reasonable.

a. Strongly disagree
b. Disagree
c. Neutral
d. Agree
e. Strongly agree

7- The time to find and reserve a parking (if applicable) was reasonable.

a. Strongly disagree
b. Disagree
c. Neutral
d. Agree
e. Strongly agree

A.4 Proposed PRS Usability Task Scenario

1- The system was simple to use.

a. Strongly disagree
b. Disagree
c. Neutral
d. Agree
e. Strongly agree

2- It was easy to learn how to use the system.

a. Strongly disagree
b. Disagree
c. Neutral
d. Agree
e. Strongly agree

3- I was able to make the reservation (if available) quickly using the system.

a. Strongly disagree
b. Disagree
c. Neutral
d. Agree
e. Strongly agree

4- I was able to find the best parking of my choice easy.

a. Strongly disagree
b. Disagree
c. Neutral
d. Agree
e. Strongly agree

5- Choosing the arrival time and departure time was easy.

a. Strongly disagree
b. Disagree
c. Neutral
d. Agree
e. Strongly agree

6- The number of taps to find and reserve a parking (if applicable) was reasonable.

a. Strongly disagree
b. Disagree
c. Neutral
d. Agree
e. Strongly agree

7- The time to find and reserve a parking (if applicable) was reasonable.

a. Strongly disagree
b. Disagree
c. Neutral
d. Agree
e. Strongly agree

A.5 Comparison Questionnaire

System A is the Map-Based Parking Reservation System.
System B is the proposed Text-Mode Parking Reservation System.

1- Which system do you believe makes you achieve your goal faster?

a. System A
b. System B
c. Another system I used before.
d. None

2- Which system do you find more user friendly?

a. System A
b. System B
c. Another system I used before.
d. None

3- Which system do you believe you might use when you look for a parking?

a. System A
b. System B
c. Another system I used before.
d. None

4- Which system do you believe has a reasonable number of taps to achieve your goal?

a. System A
b. System B
c. Another system I used before.
d. None

5- Overall, which system do you find more usable to make a reservation?

a. System A
b. System B
c. Another system I used before.
d. None

6- Overall, If you are about to use parking reservation systems, which system do you prefer or like more?

a. System A
b. System B
c.Another system I used before.
d. None

7- I would use online parking reservation systems more frequently if I use:

a. System A
b. System B
c.Another system I used before.
d. None

Please let us know if you have any suggestions or concerns regarding our system or any parking reservation systems.

____

____

____

____

____

____

Thank you for your participation!