

Software Requirements Specification for Tentown Parking Finder

EECE 415
Group 3
Assignment 3

November 1, 2011

Elicitation Leader : Jian Liew - 74727066 - munjian2@hotmail.com
Administrative Assistant & Contact Person : Mark Cho - 19976075 - sh89elf@gmail.com
Team Leader : Sean Everest - 68397066 - skandhi@gmail.com

TABLE OF CONTENTS

- 1. Introduction
 - 1.1 Purpose
 - 1.2 Scope
 - 1.3 Definitions, Acronyms, and Abbreviations
 - 1.4 References
 - 1.5 Overview
 - 2. Overall description
 - 2.1 Product Perspective
 - 2.2 Product Functions
 - 2.3 Use Cases
 - 2.4 User Characteristics
 - 2.5 Database Characteristics
 - 2.6 General Constraints
 - 2.7 Assumptions and Dependencies
 - 2.8 Apportioning of Requirements
 - 3. Specific requirements
 - 3.1 Functional Requirements
 - 3.2 External Interface Requirements
 - 3.3 Performance Requirements
 - 3.4 Design Constraints
 - 3.5 Software Quality Requirements
- Appendix A** : Cucumber Scripts
- Appendix B** : Traceability Matrix

1. Introduction

1.1 Purpose

The objective of this project is to design an application that will make it easier for drivers in Tentown to find available parking near their current location. It will be a GUI based, multi-platform application that will attempt to locate the user automatically through local or global positioning systems. Users should be able to check if there are available spots in specific parking lots, either through staff contact information or a vacancy counter which parking lot staff would be responsible for keeping.

The following document will describe the specific functional and non-functional requirements of this system as prescribed from the Tentown representatives during elicitation meetings. Provided are use case diagrams, data flow diagrams and cucumber scripts to better describe the functionality of the system.

1.2 Scope

Our client, The City of Tentown, encountered increased traffic congestion in the city, and analyzed that because of parking lots' unknown locations the streets became more congested with drivers searching for an available parking space. Tentown does not have centralized parking for residents or visitors to park while they dine, shop, or make other use of the vibrant downtown sector. Instead, there are numerous privately owned lots and parkades scattered around the area. In order to resolve this problem, the elected officials of Tentown have asked that a "parking lot finder" be developed that would locate nearby parking lots with empty parking spaces available for drivers to use. If parking were made easier to find, there would be a more efficient use of the parking available and therefore an increase in the flow-through of visitors to the town. This plan would be beneficial to the municipality as well as the shop and parking lot owners.

1.3 Definitions, Acronyms, and Abbreviations

- DBMS : Database Management System
- GUI : Graphical User Interface
- GPS : Global Positioning System
- HTML : HyperText Markup Language
- PC : Personal Computer
- PHP : Hypertext Preprocessor
- RSS : Rich Site Summary
- SMS : Short Message Service
- SQL : Structured Query Language
- TDD : Test Driven Development

1.4 References

- Parking Monitoring Systems
<http://www.cflhd.gov/ttoolkit/flt/FactSheets/ITS/PARKING%20MONITORING.htm>

1.5 Overview

A visitor drives to Tentown for the exemplary shopping, dining and entertainment, but upon arriving they have difficulty finding a parking spot. Parking is available nearby, but the only method the parking lot owners have of notifying drivers of availability is currently a sign near the entrance to their lot. Frustrated, the driver may choose to leave Tentown all together, which would cost the local businesses as well as the city as a whole. In the best case, the driver may spend more time than necessary finding a spot, giving less time for them to enjoy the city.

What if there were a central database, maintained by the city, to hold information on all the parking lots and parkades within Tentown? What if a visitor could use this database through an intuitive and informative interface on their computer or mobile device to find a parking spot quickly and efficiently? The city obviously has a great deal to gain by undertaking this project.

2. Overall Description

2.1 Product Perspective

Based on the type of user, the application will provide a number of different interfaces. The commonality between each of these will be the fact that they are web-based, allowing for the maximum amount of portability. Simply a web-browser, and for certain interfaces an authorized login, will be all that is required to access the features of the application.

A database management interface will show the current list of monitored parking lots and parkades in a simple fashion and allow basic database manipulation functionality. A parking lot status interface will allow remote connection to the main monitoring database to update the availability of parking giving two options, either available or not. Both the database management and parking lot status interface will require authorized logins and be designed solely for use on a PC or laptop.

The main user interface will provide all the necessary information to find an available parking spot quickly, given a desired location. The information, including maps, contact information, and parking rates, will be laid out in a clear and concise manner. This interface will require no authorized login and will have two different designs based on whether it is accessed via PC or mobile device to allow for readability between different resolutions and screen sizes.

All interfaces will interact with an SQL based DBMS running on the municipality's web server, which will hold all the parking lot information.

The major features of the application will include the ability to add and remove parking lots from the monitoring database, update the status of available parking within the monitored parking lots, and to check for the nearest available parking to a specific location. The system will only monitor parking lots and parkades and not street-side metered or free parking. On a map view, the information displayed will include the user's current location as well as the location of parking along with its status, rates and contact information. For a specific parking lot, the status will be described as either "spots available" or "no spots available". The resources required to monitor the location and number of specific open parking spots within a lot would be costly so it was decided upon to use a simple yes or no for availability at any given lot.

2.3 Use Cases

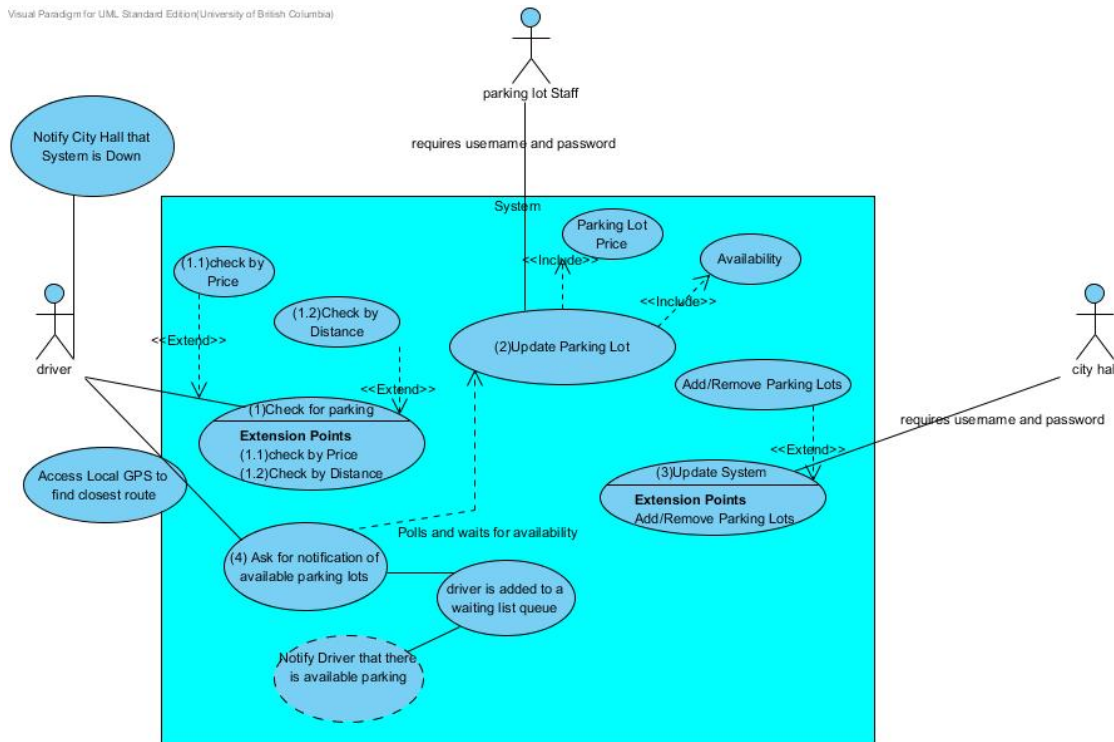


Figure 1. Parking Finder System Use-Case Diagram

Use Case 1.1 - Return Closest Parkade

Actions	Comment
Driver accesses application	
System tries to access remote GPS to find current location of driver	If system cannot access/find remote GPS, the system will notify the driver
Driver asks for closest parkade	
System returns location and direction to the closest parkade	

Use Case 1.2 - Return Cheapest Parkade

Actions	Comment
Driver accesses application	
System tries to access remote GPS to find current location of driver	If System cannot access/find remote GPS, the system will notify the driver
Driver asks for the cheapest parkade	
System returns location and direction to cheapest parkade	

Use Case 2.1 - Updating Parkade Availability

Action	Comment
Parking lot staff accesses application	
System requests authentication	If authentication fails, the system will give error messages
the staff update parking lot's availability	
System notifies the staff when update was successful	an error message will be given in case of failure

Use Case 2.2 - Update Parkade Rate

Action	Comment
Parking lot staff accesses application	
System requests authentication	If authentication fails, the system will give error messages
the staff update parking lot's rate	
System notifies the staff when update was successful	an error message will be given in case of failure

Use Case 3 - Add/Remove Parkade from System

Action	Comment
City hall staff accesses application	
System requests authentication	If authentication fails, the system will give error messages
the staff add/remove parking lot to/from the system	
System notifies the staff when add/remove was successful	an error message will be given in case of failure

Use Case 4 - Notification Setup

Actions	Comment
Driver accesses application	
Driver checks for available parking	If there is no available parking, an error message is returned notifying the driver that there is no available parking
Driver asks to be notified for next available parking	
Driver is added to a waiting list queue to be notified when a parking space is available.	
Driver can choose to accept or decline parking space	If Driver chooses to decline the space, he is kept in the queue and the notification is sent to the next driver in the waiting list.

2.4 User Characteristics

The product will have three types of user: city hall employees, parking lot staff, and drivers looking for available parking. Employees of the city are broken up into two separate users. First is the City Hall Staff who will be able to manage the database of monitored parking locations by adding, deleting or updating locations as well as maintaining the contact information of each.

Parking lot staff will be in charge of updating the status of their lots between full or empty and they will be provided a specific interface to do so. The number of available parking spots is solely dependent on the parking lot staff, so it is assumed that the staff will stay responsible in making hourly checks on the parkade.

The main user of the application, however, will be the drivers entering Tentown and looking for a place to park. It is assumed that they have access to the internet through means of a computer or cell phone that has a GPS device. The GPS device is not necessary as long as the user knows their current location or the location where they wish to park near (a specific shop or restaurant etc.)

2.5 Database Characteristics

One of the main components of the Parking Finder system will be the database, which will be described in detail in the following section. The database will be maintained by the Tentown municipality and hold the information of the cooperating privately owned parking lots and parkades. This information will include a unique name to represent the parking lot, contact information in the form of an address and phone number, the current parking rates, and the availability of parking within the lot. The availability status will take the form of either “vacancy” or “no vacancy”, the specific number of empty spots within the parking lot shall not be kept track of. The following figure (*Figure 2*) shows the single entity within the database, no other entities or relationships are required.

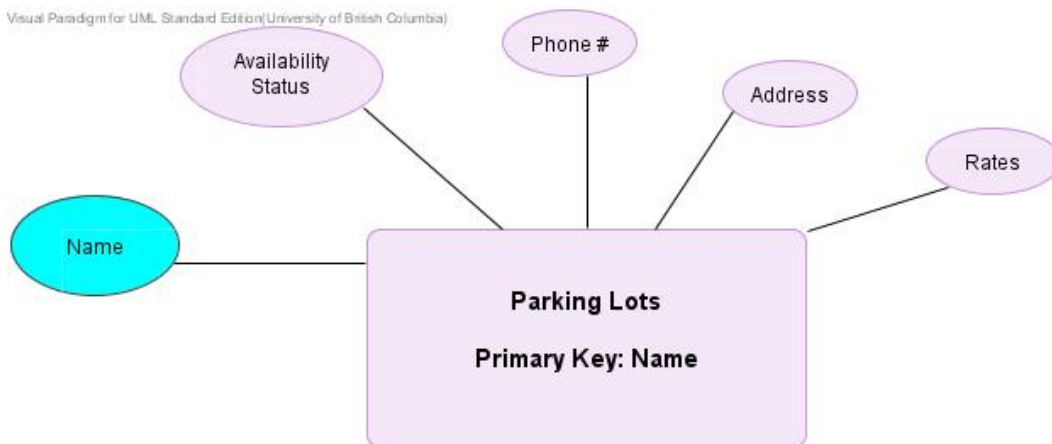


Figure 2. Parking Finder System Database Entity Diagram

Information within the database will be maintained by two different sources. The attendants at each registered parking lot or parkade will update the database with the availability status of their parking spots. If there are spots available, the status will be “vacancy”. If the lot is full, the status will be “no vacancy”. City Hall employees will add and remove the registered parking lots from the database as well as update the correlating contact and rate information for each. As shown in the dataflow diagram below (Figure 3), the information managed by both the city hall employee and the parking lot attendant will be passed to a visitor to Tentown when they use the system to find parking.

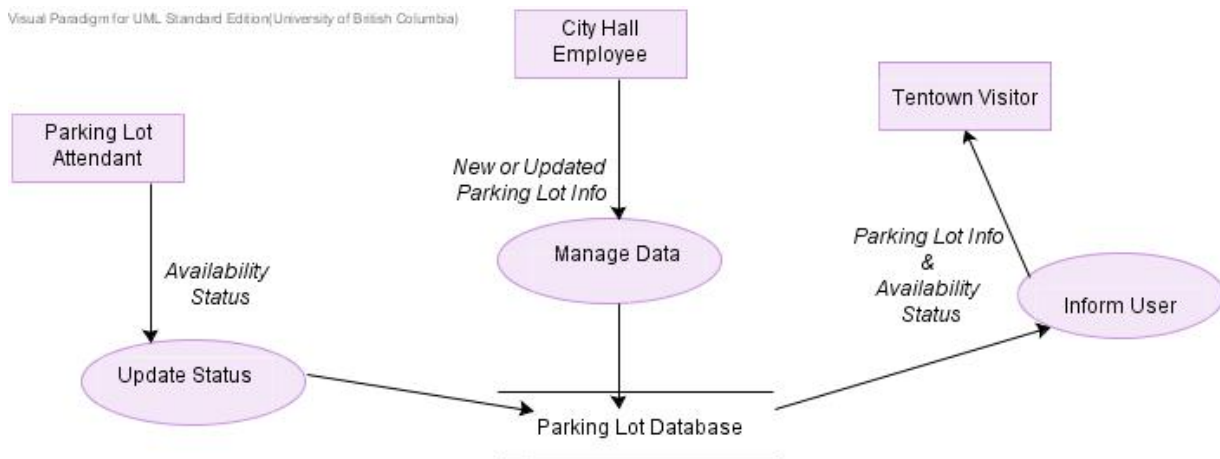


Figure 3. Parking Finder System Dataflow Diagram

2.6 General Constraints

The application will use the infrastructure of the internet in order to distribute parking information to whoever wishes to know. Unfortunately those without access to the internet--whether on a computer or cell phone--will not be able to use the application. In order to best automatically approximate the location of the user, the application recommends users to have GPS. If users do not have GPS within their device, then other less reliable methods will be used (cell tower triangulation, location via IP address, etc.). The automatic location discovery of users is not guaranteed and users may need to manually enter their location into the application.

The application provides an interface for users to easily find the closest available parking, however it does not guarantee that available parking will be found nearby, if at all.

2.7 Assumptions and Dependencies

The entire system depends on the accurate information provided by the parking lot staff monitoring their empty spots. If this information is incorrect due to incompetence, specific intent or otherwise, the sole purpose of the application--finding adequate parking for visitors to Tentown--will be defeated. This may even cause enough frustration in users as to persuade them to abandon the notion of visiting Tentown altogether, thereby doing more harm than if there had been no system in place at all. Steps should be considered to ensure the accuracy of the information provided by the parking lot staff, however this is outside the scope of this document.

2.8 Apportioning of Requirements

Each requirement can be placed into certain categories according to their priority. Below is a chart that shows the priority of each category and describes the reasoning behind our choice.

Category	Cost	Value	Priority
<i>Database</i>	9	10	High
<i>User Interface</i>	6	9	High
<i>Security</i>	8	10	High
<i>Location Services</i>	7	6	Medium
<i>Parking Assistance</i>	7	10	High
<i>Notifications</i>	2	8	Medium
<i>Miscellaneous</i>	1	3	Low
<i>Performance</i>	7	8	High

<i>Interoperability</i>	6	7	Medium
-------------------------	---	---	--------

2.8.1 Database

The database features provide secure and reliable access to the information on which the entire application relies. Despite the high cost of server hardware and maintenance, the system would not function without these features therefore they are high priority.

2.8.2 User Interface

The User interface is the main component in which the users will be interacting with. It is important that it is user friendly and easy to navigate through. Each of the three users (Driver, Parking Staff, City Hall) will have their own interface which will help control different parts of the system. For this reason, the user interface is given one of the highest priorities in our cost value chart.

2.8.3 Security

Since data base is recording the data of a parking lot, the database holds commercial value. To prevent misuse and to provide protected data storing, high standard of security must be implemented. Therefore, the security is at the top of the priority list.

2.8.4 Location Services

This service is really dependent on whether the driver has GPS built into their mobile device. It is important that the locations of the parking lots are accurate for determining the most suitable spot for the driver, but accuracy can vary depending on the user's GPS. Therefore, this service is on a medium priority.

2.8.5 Parking Assistance

These are the features which take the location provided by the Location Services, collect information from the Database features, organize the results based on the user preference and display the information on the User Interface. In short, the Parking Assistance requirements bring the separate parts of the system together and therefore are vital to the success of the project.

2.8.6 Notifications

Notification requirements for functionality that is optional to specific type of user(driver). Since the functionality is opt-in, not every users will benefit from this service, and it is hard to predict how many users will participate. Nonetheless, the value of this set of requirements is noticeable. Thus, notification requirements valued at medium level of priority

2.8.7 Miscellaneous

Miscellaneous requirements are guideline and rules for developers for sustainable system. The miscellaneous requirements are not visible to any types of users, and the system

will function without failure even if miscellaneous requirements are not met. For this reason, Miscellaneous requirements are at the bottom of the priority list.

2.8.8 Performance

Bettering the system's performance is a valuable use of project resources as things like speed and reliability are expected in modern computing. Users may choose not to use a system that is slow or unresponsive. Performance therefore has a high priority. However, an unlimited amount of money may be thrown at the improvement of the system while the noticeable improvement will become harder and harder to achieve.

2.8.9 Interoperability

The only communication we have with the Parkinglot Finder system is its ability to connect to the driver's remote GPS system. This is quite important for helping find the closest, and proper directions to the parking lot. But its not highly important, because if it cannot find the remote GPS, the system can still send the address of the parking lot.

3. Specific Requirements

3.1 Functional Requirements

3.1.1 (*Database*) The database shall hold the following information with respect to each monitored parking lot/parkade: a unique name, location (address), phone number, parking rates, parking availability.

Identifier	FR-1
Category	Functional
Specification	The information of each parkade will be held in a database for city hall to look over.
Fit Criterion	the database must have sections where the information of the parkade can be filled out
Source	Stakeholder meeting, Tentown rep.
Rationale	in order for drivers to find the the best parkade, and in order for city hall to keep track of the parkades, this database is needed
Interaction	3.1.1, 3.1.6, 3.1.8, 3.1.11
Priority Level	High
Stability	High

3.1.2 (*Database*) The application shall provide a secure interface for managing the database information.

3.1.2.1 Database management functionality shall include the following actions: create new entries, update existing entries, delete existing entries.

Identifier	FR-2
Category	Functional
Specification	Database management component shall be secure and allow for adding deleting and updating of the information.
Fit Criterion	No unauthorized user allowed access to database management, authorized user has access to add, delete and update functionality.
Source	Questionnaire
Rationale	The information within the database must be maintained by authorized personnel only.
Interaction	+3.1.1
Priority Level	High
Stability	High

3.1.3 (*Parking Assistance*) The application shall deliver the information of the closest parking lot with empty spaces based on a given location.

Identifier	FR-3
Category	Functional
Specification	The application will show the driver the closest parkade from his current location
Fit Criterion	There are no parkades closer to the driver than the one being shown
Source	Questionnaire and Interview
Rationale	The driver is looking for the closest parkade. There shouldn't be any closer parkade to the driver.
Interaction	+3.1.3, +3.1.4, +3.1.5
Priority Level	High
Stability	High

3.1.4 (*Parking Assistance*) If no location available, the application shall provide all available parking information to the user.

3.1.5 (*Parking Assistance*) If no parking spots available, the application shall show the following error message: "No parking available at this time".

3.1.6 (*Location Services*) The application shall use GPS and other techniques to try and determine location of user (within 1 km).

Identifier	FR-6
Category	Functional
Specification	The application shall use location estimation technologies to determine the location of the user.
Fit Criterion	Location estimate by application is less than 1 km away from actual location of user.
Source	Stakeholder meeting, Tentown rep.
Rationale	The application requires a location to return the nearest available parking
Interaction	+3.1.3
Priority Level	Medium
Stability	High

3.1.7 (*Notifications*) The application shall allow user to set up notifications of special events by e-mail or SMS.

3.1.7.1 Special events shall be classified as: a specified parking lot has become available.

3.1.7.2 Request for notification of available parking lot will be queued and reservation will be void if user does not reply.

Identifier	FR-7
Category	Functional
Specification	Users shall have the option to receive notifications of a specific parking lot becoming available.
Fit Criterion	If a user asks for a notification when a parking lot becomes "vacant", when the lot becomes "vacant" then user should receive a notification.

Source	Stakeholder meeting, Tentown rep.
Rationale	Avoid users having to check the availability of a desired lot continuously.
Interaction	+3.1.1
Priority Level	Medium
Stability	Medium

3.1.8 (*User Interface*) The application shall allow users to organize the list of available parking according to price, distance and time of availability.

3.1.9 (*Security*) The database for the application must not allow unauthorized access.

Identifier	FR-9
Category	Functional
Specification	No unauthorized person shall have access to the database.
Fit Criterion	Only users with authentication may access the database.
Source	Stakeholder meeting, Tentown rep.
Rationale	Data within the database must be reliable, and secure.
Interaction	+3.1.1, +3.1.2
Priority Level	High
Stability	High

3.1.10 (*Notifications*) The application shall notify users when delivering requests from database has failed due to technical problems (e.g. internet connectivity, GPS failure, etc.).

3.1.11 (*User Interface*) The application shall provide an interface for updating each parking lots' status of vacancy.

3.1.11.1 The interface to update a parking lot's availability should require a unique user name and password.

3.1.12 (*User Interface*) The application should provide an interface to add, modify or delete monitored parking lots, as well as their information.

3.1.12.1 The interface to manage parking lot information should require a unique user name and password.

3.1.13 (*User Interface*) The application shall provide an interface to display the information of all parking lots and parkades in the database in a clear and readable way.

3.1.13.1 The interface to display parking lot information should have a separate design for both PC and mobile device use.

3.1.13.2 The interface to display parking lot information should include an interactive overhead view of a map of Tentown.

3.1.13.3 The interface to display parking lot information shall allow a location to manually be entered in case automatic localization techniques malfunction/fail.

3.2 External Interface Requirements

3.2.1 (*Interoperability*) The system must provide an interface for mobile device developers (i.e. mobile phones, GPS navigation) to create their own applications.

Identifier	NFR-1
Category	Non-functional
Specification	The system must provide an interface for 3rd party developers.
Fit Criterion	Mobile device developers are able to create 3rd party applications using the system.
Source	Stakeholder meeting
Rationale	Better integration with more devices requires 3rd party help.
Interaction	+All reqs.
Priority Level	Medium
Stability	Medium

3.2.2 (*Interoperability*) The software must run on the latest version of all popular internet browsers (e.g. Internet Explorer, Firefox, Chrome, Safari, etc.).

Identifier	NFR-2
Category	Non-functional
Specification	The software must be able to run on all internet browsers
Fit Criterion	The software should run consistently and the same on Firefox, Chrome and Safari

Source	Questionnaire
Rationale	The driver should not have any trouble navigating through the software due to the internet browser.
Interaction	+3.2.2, +3.5.1
Priority Level	Low
Stability	Medium

3.3 Performance Requirements

3.3.1 (*Notification*) The system must send the notification of availability to the driver within a minute after the parking lot staff updates the system.

Identifier	NFR-3
Category	Non-functional
Specification	A driver looking for a parking space should be notified that there is free parking within a minute of the parking staff updating the system
Fit Criterion	The software must send a notification within a minute
Source	Questionnaire and Interview
Rationale	Drivers should not need to waste time looking for parking instead of shopping and enjoying themselves in Tentown
Interaction	3.3.1, 3.3.2
Priority Level	medium
Stability	Medium

3.3.2 (*Performance*) The data of the notification should be not greater than 100 kilobytes of information.

Identifier	NFR-4
Category	Non-functional
Specification	Data transferred to a driver's mobile device should be not greater than 100 kilobytes
Fit Criterion	100 kilobytes is the max size of the notification
Source	Future meetings
Rationale	The driver's mobile device shouldn't be clogged with the messages we send to them
Interaction	+3.3.1, +3.3.2
Priority Level	High
Stability	High

3.3.3 (*Performance*) The system speed should not be affected by the number of current users.

Identifier	NFR-5
Category	Non-functional
Specification	The system speed should remain constant, no matter the current system usage.
Fit Criterion	System response (ms) for 1 concurrent user equals or is greater than the system response for 100 concurrent users
Source	Future meetings
Rationale	Many users may access the system at once.
Interaction	+All reqs.
Priority Level	High
Stability	High

3.4 Design Constraints

3.4.1 (*User Interface*) The interface must have Tentown logo visible to users.

Identifier	NFR-6
Category	Non-functional
Specification	The logo must be visible to users when using application
Fit Criterion	Tentown logo appears on GUI
Source	Interview
Rationale	The application should advertise the Tentown
Interaction	+3.1.11, +3.1.12, +3.1.13, +3.1.2
Priority Level	Low
Stability	Medium

3.4.2 (*User Interface*) An independent design should be created for use with PCs and

mobile devices to account for varying screen sizes

3.5 Software Quality Attributes

3.5.1 (*Miscellaneous*) The system will be written in HTML, PHP and Javascript.

Identifier	NFR-7
Category	Non-functional
Specification	the System will be writtin in HTML, PHP and Javascript
Fit Criterion	
Source	Stakeholder meeting
Rationale	The language should be something that most programmers are comfortable using
Interaction	+3.51
Priority Level	Low
Stability	High

3.5.2 (*Miscellaneous*) The software must be commented in detail for easy understanding and modification.

Identifier	NFR-8
Category	Non-functional
Specification	The software shall be easily maintainable.
Fit Criterion	Developers modifying the software are able to easily find the features they are looking for.
Source	Stakeholder meeting
Rationale	More features may be added pending additional funding.
Interaction	+All reqs.
Priority Level	low
Stability	High

APPENDIX A

Cucumber Scripts

The following cucumber scenarios will aid in Test Driven Development of the application. We have chosen 3 features that each describe a user, what services they want the application to perform and why. Each feature also has one or more scenarios associated with it that use the *Given, When, Then* structure to describe the preconditions, the actions, and the desired results.

Feature 1: As a visitor to Tentown, I want to find parking quickly in order to go shopping.

Scenario 1.1: Finding nearest parking

Given the visitor has internet access

And there is parking available

When the visitor enters his/her location

Then the location of the closest available parking is returned

Scenario 1.2: Find the cheapest available parking

Given the visitor has internet access

And there is parking available

When the visitor asks for available parking with the cheapest rates

Then the location of the parking lot that is available with the lowest rates is returned

Scenario 1.3: No available parking

Given the visitor has internet access

And no parking lots have available spots

When the visitor looks for parking

Then an error message will be returned

Feature 2: As a Tentown city hall staff, I want to maintain parking lots in the system in order to bring more business to the city by allowing more efficient parking.

Scenario 2.1: Adding a parking lot into the system

Given the city hall staff has internet access

And he/she has authorization

And the parking lot with the same name is not in the system yet

When he/she enters parking lot information and confirms the entry

Then the parking lot will be registered into the system

Scenario 2.2: Removing a parking lot from the system

Given the city hall staff has internet access

And he/she has authorization

And the parking lot with given name exists in the system

When he/she removes the parking lot from the system

Then the parking lot will no longer be in the system

Feature 3: As a Parking lot staff, I want to be able to update the availability of a parking lot in order to ensure efficient use of my parking lot.

Scenario 3.1: Changing parking lot availability status

Given the employee has internet access
And there are no parking spots available in the parking lot
When a spot opens up and the employee changes the parking lot status to “available”
Then the database will now show that parking lot as “available”

Scenario 3.2: Updating parking lot rate

Given the employee has internet access
When the employee changes the rate of parking in their lot
Then the database will show the new rate of parking in that lot

APPENDIX B

Traceability Matrix

As the project evolves, the connections between the features of the system and the requirements first elicited will become more complex as well as more important. The quality of the traceability management will determine if the process is smooth when features are updated, taken away or new ones are added. If the connections are not maintained, requirements may become unsatisfied or features may become redundant during these evolution stages.

The following is a correlation between the use cases, TDD scenarios and specific requirements of the system.

Use Case	Cucumber Feature/ Scenario	Specific Requirement
1.1	1/1.1	3.1.3, 3.1.6
1.2	1/1.2	3.1.6, 3.1.8
2.1	2/2.1	3.1.9, 3.1.10
2.2	2/2.2	3.1.9, 3.1.10
3	3	3.1.11, 3.1.9, 3.1.10
4	---	3.1.5, 3.1.7