ParkMe: Software Requirements Specification

Daniel Agostinho
agostd - 001414323
Group 5

Michael Bitzos
bitzosm - 001405050
Group 5

Kathryn Brownlee
brownlks - 001408416
Group 5

Anthony Chang
changa7 - 001413615
Group 5

Ben Petkovsek
petkovb - 001417104
Group 5

November 4, 2018

Contents

1 Introduction **3**

1.1 Behaviour Overview . . . 3

1.2 Scope . . . 3

1.2.1 Stakeholders . . . 4

2 Constants **4**

3 Context Diagrams **5**

4 Monitored and Controlled Variables **5**

4.1 Monitored Variables . . . 6

4.2 Controlled Variables . . . 7

5 Functional Decomposition Diagram **7**

6 Use Case Diagrams **8**

7 Requirements **10**

7.1 Functional Requirements . . . 10

7.1.1 Parking Detection Requirements . . . 10

7.1.2 Application Requirements . . . 10

7.1.3 Analytics Requirements . . . 11

7.1.4 Business User Requirements . . . 11

7.2 Nonfunctional Requirements . . . 12

7.2.1 Look and Feel Requirements . . . 12

7.2.2 Usability Requirements . . . 12

7.2.3 Performance Requirements . . . 12

7.2.4 Security Requirements . . . 13

7.2.5 Operational Requirements . . . 13

7.2.6 Scalability Requirements . . . 13

7.2.7 Maintainability and Portability Requirements . . . 14

7.2.8 Cultural and Political Requirements . . . 14

7.2.9 Legal Requirements . . . 15

7.2.10 Health and Safety Requirements . . . 15

7.3 Undesired Event Handling . . . 15

8 References **15**

1 Introduction

At one point or another everyone who drives has encountered this scenario; you plan to go to a mall for the day so you drive all the way there only to discover that the parking lot is seemingly full. You must frustratingly drive around the parking area to try and find a single spot, if you even find one. This problem has existed for as long as vehicles have, and while some rudimentary and limited solutions exist such as a parking structure noting how many open spots exist when you enter, there are no solutions that an average consumer can use to solve the problem preemptively. Imagine if there was a way that you could know exactly how many spots were open in a parking lot at a given time and where they were, before you even left your house.

On the other hand, imagine being a business owner and having the ability to see trends in how many visitors arrived on a given day, how long the average visitor stayed, or even if a vehicle has been parked in a paid spot for longer than they should. ParkMe is a mobile application and parking availability system that aims to provide valuable information not only to customers, but also businesses.

1.1 Behaviour Overview

To solve the problems mentioned above, ParkMe will implement several general systems. The first and most important of these is the ability to monitor the occupancy of parking spots in a given parking area through a series of physical sensors. This data will then be used by the mobile application to update the customer on the number of available spots in the lot, as well as the location of those spots. Specific data on information such as accessible parking, expectant mothers, paid parking...etc will also be displayed, and will utilize preferences set by the user.

Ideally, the application will also include functionality to navigate the user to a parking spot/row/area, but this integration will depend on the accuracy and availability of GPS systems and the complexity of those systems. In addition, a secondary section of the application which will only be available to verified business owners will include various statistics and trends based on the data gathered by the system, allowing those owners to better tailor their experience to the customers, and increasing returns as well.

1.2 Scope

ParkMe will allow customer users to receive up-to-date information on the availability of parking spots in a desired lot. They will be able to know where their desired parking spot is, and may also be able to get navigation to the spot. Additionally, they will be able to mark where they parked. The business users will be able to access a separate interface where they can view information and statistics that can be made from the parking data. These may include things such as average occupancy rates for parking spots, overall number of visits and departures based on time/day...etc.

1.2.1 Stakeholders

There are two main stakeholders who will be interested in our product, the consumers, and businesses. The consumers are the everyday people who use the application for parking availability. The businesses can constitute two groups, but are handled in the same way. One of these groups is the small business, who owns a smaller lot and is interested in customer information and trends. The second group is the large business. This group is composed of entities such as the owner of an entire mall, rather than the individual businesses inside of the mall who wouldn't be able to glean any useful information from the parking trends. Making sure that both of these groups find our system useful is imperative for its success.

The consumer will find use in knowing that parking spots are open for a store they wish to visit, and will have less frustration physically finding a spot. This is good for both the consumer and the business, as it will ensure that not being able to find a parking spot will not deter a potential customer.The business will find use in being able to recognize trends in their customer attendance, allowing them to make changes which improve their own potential goals as well as improving the experience for the customer. In this way, ParkMe satisfies both major stakeholders and the relationship between them is strengthened as well.

2 Constants

- PARKING_SPOT_RESPONSE_TIME= 500ms
  - How long it takes for the availability of a parking spot to update on the application.
- PARKING_SPOT_EXPIRATION_TIME_WARNING= 10 minutes
  - How long before the application will notify the user their parking spot is about to expire.
- MAX_ANALYTICS_DATA= 50gb
  - How much analytics data will be recorded before older data will be deleted.
- MAX_CONCURRENT_USERS= 1000 users
  - How many concurrent users our system will support.
- DETECTION_ERROR_RATE= 0.01
  - Max errors in detection in percentage the system should maintain before.

3 Context Diagrams

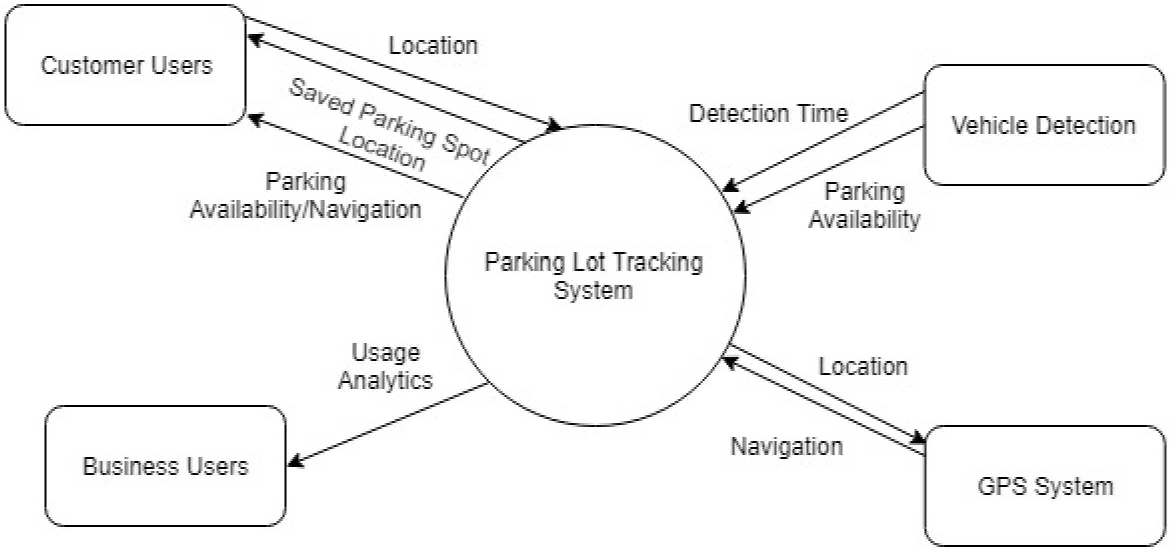


Figure 1: Context Diagram showing data flow of the system.

4 Monitored and Controlled Variables

ParkMe only contains two systems where it interacts with the environment around it: The mobile application where the user interacts with our system directly and the system responds back. The parking identifier is the second system where the users only interact with it and it does not directly interact with it.

4.1 Monitored Variables

Name	Type	Range	Units	Physical Interpretation
Vehicle_Detection	Status	"yes"/"no"	Unitless	If the Parking Spot Identifier has detected a vehicle or obstacle in the way.
Vehicle_Detection_Time	Number	[0, 31536000]	Seconds	The amount of time that has passed since an object has been detected by the sensor.
User_GPS_Data	GPS	Longitude: -180 ,180 Latitude: -90, 90	Longitude, Latitude in Degrees	The GPS location data of the users mobile device.
User_Device_Rotation	Vector2 (Number, Number)	Does not have a range	Unitless	The rotation of the mobile device relative to the normal of the device.
User_Application_Input	Action	Does not have a range	Unitless	The user's varied input to the mobile application**

Table 1: Monitored Variables for ParkMe

** - User inputs will be later described in the requirements section, this is just the general idea of inputs being registered from the environment from the user that all falls under potential application actions.

4.2 Controlled Variables

Name	Type	Range	Units	Physical Interpretation
User_Navigation	–	–	–	The navigation the application uses to guide the user to the parking lot.
Parking_Spot_Availability	Status	On/Off	Unitless	If the parking spot is occupied.
Parking_Lot_Statuses	List of Statuses	List of On/Off	Unitless	A mapping of the parking lot that the user requested with their availabilities.
Parking_Lot_Analytics	Numbers	Varied	Varied	The statistics and trends relating to a parking lot/parking spot.**

Table 2: Controlled Variables for ParkMe

** - User inputs will be later described in the requirements section, this is just the general idea of inputs being registered from the environment from the user that all falls under potential application actions.

5 Functional Decomposition Diagram

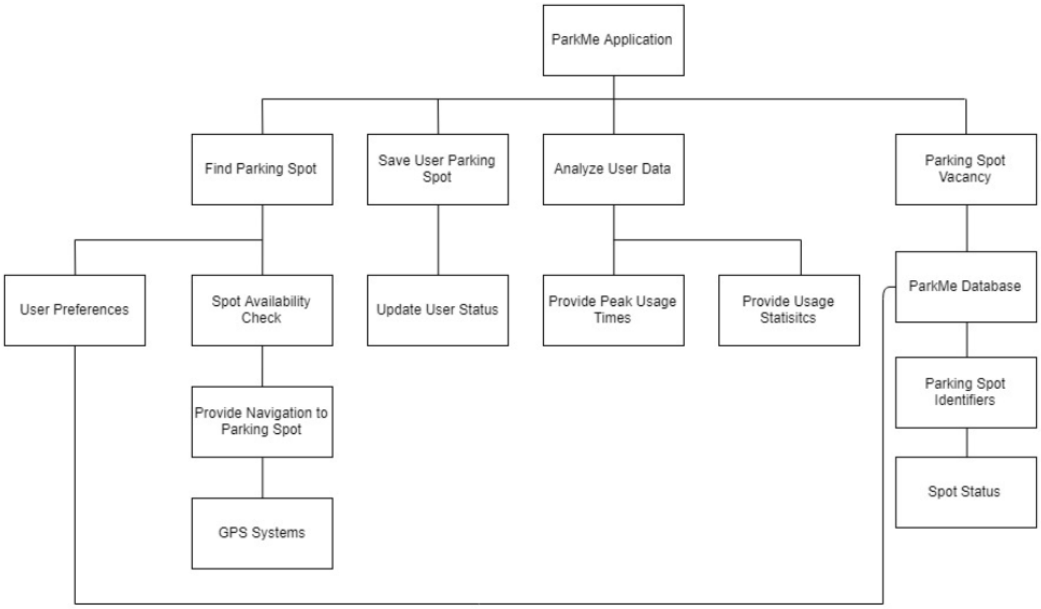


Figure 2: Functional Decomposition

6 Use Case Diagrams

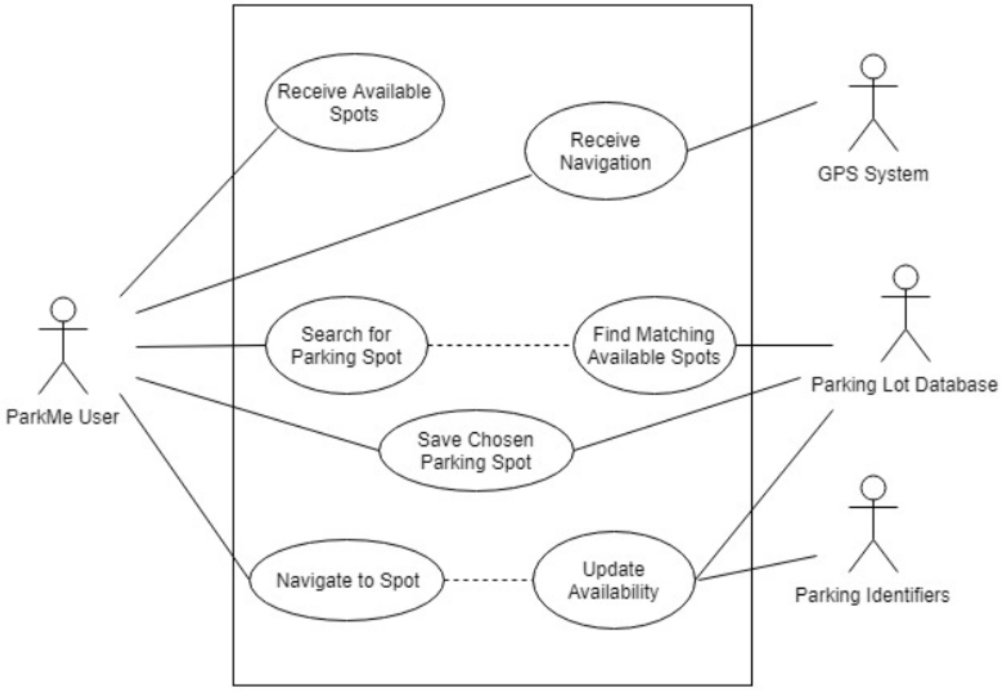


Figure 3: Use Case diagram for a driver using the app to find and navigate to an available parking spot.

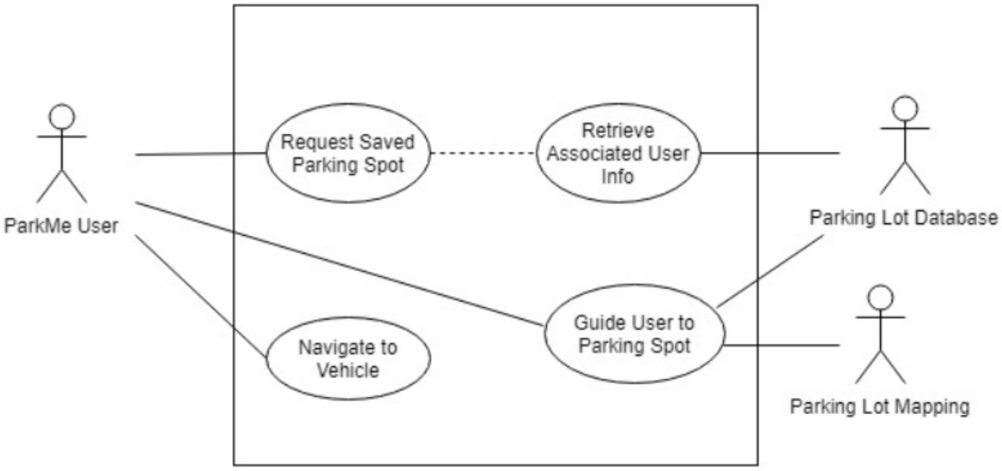

ParkMe User
Request Saved Parking Spot
Retrieve Associated User Info
Parking Lot Database
Navigate to Vehicle
Guide User to Parking Spot
Parking Lot Mapping

Figure 4: Use Case diagram for a customer returning to their vehicle using the saved parking spot location.

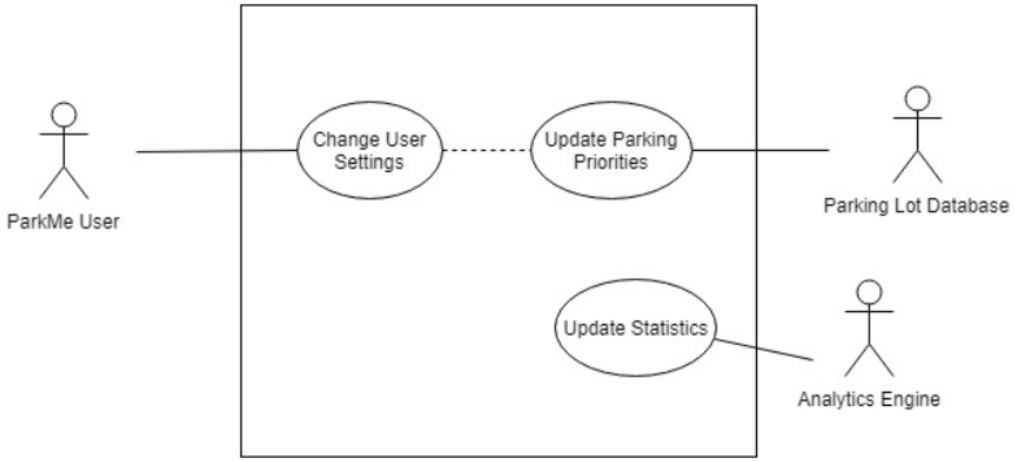


Figure 5: Use Case diagram for a user's personalized settings being changed. This affects the user's computed spot suggestion based on given choices and accessibility info.

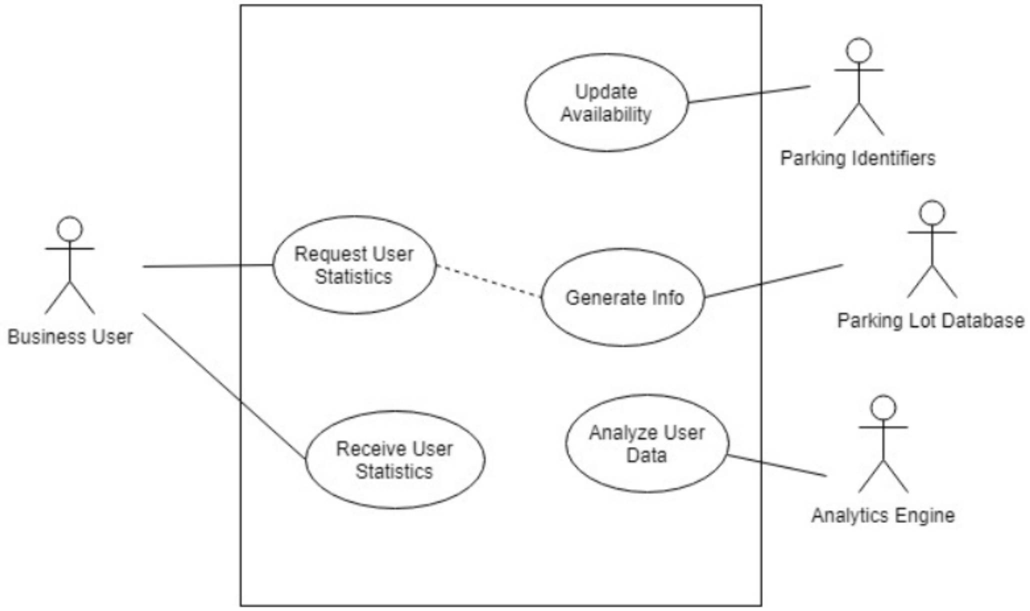


Figure 6: Use Case diagram for Business Users requesting user and parking lot statistics.

7 Requirements

For the following requirements, let LC denote the requirements that are likely to change and UC denote the requirements that are unlikely to change.

7.1 Functional Requirements

7.1.1 Parking Detection Requirements

- PDR1: The system must be able to recognize the state of vacancy of all parking spots in a parking lot.
- PDR2: The system must update the application with the status of parking spaces.

7.1.2 Application Requirements

- APR1: The system must provide the user with the number of available parking spots in a parking lot.
- APR2: [LC] The system should show a view of all available parking lots supported by ParkMe.
  - This requirement might not be included in future revisions.
- APR3: The system must allow the user to select a parking lot that is supported by ParkMe.
- APR4: [LC] The system must provide navigation to the parking lot selected.
  - This requirement might not be included in future revisions.
- APR5: The system must keep track of the location where the user parked their vehicle in the parking lot.
- APR6: The users will be able to save what parking spot they parked in.
- APR7: The system must show if the parking spot is an accessibility parking only.
- APR8: The system must not show any vacant accessibility parking spots as available if the user has not specified accessible parking.
- APR9: [LC] The system must show if there is a time limit associated with a parking space.
  - This requirement might not be included in future revisions.

- APR10: [LC] The user must be warned if their time limited parking space will expire in PARKING_SPOT_EXPIRATION_TIME_WARNING.
  - This requirement is dependent on if APR9 still holds.
- APR11: [LC] When the user is navigating to the parking lot, the system must update the user's position on the map and provide navigation instructions.
  - This requirement is dependent on if APR4 still holds.
- APR12: The system must notify the user when the parking lot does not have any vacant spots.
- APR13: [LC] The system must show the distance to the closest building entrance from the user's selected parking space.
  - This requirement is dependent on whether during development, access to entrance information is provided.
- APR14: [UC] The application will warn the user to be cautious during the hours of 8pm and 8am and/or when parking lots are less than 10 percent occupied. (The purpose of this is to remind users to be safe at night where parking lots/garages can be unsafe.)
  - While simple to implement it is an additional non-critical feature that may extend development time.

7.1.3 Analytics Requirements

- ANR1: The application will show the user's statistical averages of the vacancy of the parking lot.
- ANR2: The application will show the user peak traffic times for each parking lot.

7.1.4 Business User Requirements

- BUR1: [LC] The application must notify the business owner when a vehicle has been parked in a spot for longer than allowed.
  - This requirement depends on the information given to the system by the business.
- BUR2: The application should provide the business users with information on each specific parking space by location and type.
- BUR3: The application should provide business users with trends about the parking lot.

- BUR4: [LC] The system will record up to MAX_ANALYTICS_DATA worth of analytics data while it is online.
    - The amount of data that will be used could easily change based on other requirements

7.2 Nonfunctional Requirements

7.2.1 Look and Feel Requirements

- LF1: The applications user interface will adjust with device's screen size specifications.

7.2.2 Usability Requirements

- UR1: The mobile application should be easy to use and understand. Users should be able to perform basic functionality without addition instructions.
    - Interface should be simple with high discoverability.
- UR2: Interface will follow standard conventions (buttons, hamburger menu, etc.) and constraints will prevent users from performing undesired actions.
    - The user interface should be familiar to the user and mappings should be intuitive to minimize learning time and mistakes.
    - Constraints can help limit the possible actions a user can take and prevent some errors from occurring.
- UR3: The application will provide sufficient error messages on error occurrences.

7.2.3 Performance Requirements

- PR1: [LC] Users should be able to use the mobile application with low response time. Any communication done between client and server should be completed within PARKING_SPOT_RESPONSE_TIME.
    - The application need to receive a response and update to provide accurate information (parking spot is actually available and was not taken).
    - This is likely to change as PARKING_SPOT_RESPONSE_TIME is likely to change.
- PR2: [LC] Servers should be able to sustain MAX_CONCURRENT_USERS.
    - Depending on parking lot sizes and if a single server will host multiple parking lots, it needs to be able to receive and respond to requests without delay

  - The supported amount of concurrent users can change with increased budget and better server capabilities

- UR3: [LC] Parking space availability detection should have a error rare less than DETECTION_ERROR_RATE
  - inimal false positives (reading a spot is occupied when it's not or vice versa)
  - This is likely to change as DETECTION_ERROR_RATE is likely to change.

- PR3: [LC] Parking space availability detection should have a error rare less than DETECTION_ERROR_RATE
  - inimal false positives (reading a spot is occupied when it's not or vice versa)
  - This is likely to change as DETECTION_ERROR_RATE is likely to change.

7.2.4 Security Requirements

- SR1: Communication between the android application and the server should be done through a secure connection
  - Private information should be kept private. Safeguard against middleman attacks

- SR2: Gathered statistics should be anonymous
  - Privacy is a major concern especially since most people do not wish to be tracked. In the event of a data breach, this would also limit the amount of damage done

7.2.5 Operational Requirements

- OR1: Users phone must have enough storage to save data
  - User must be able to install and run the application.

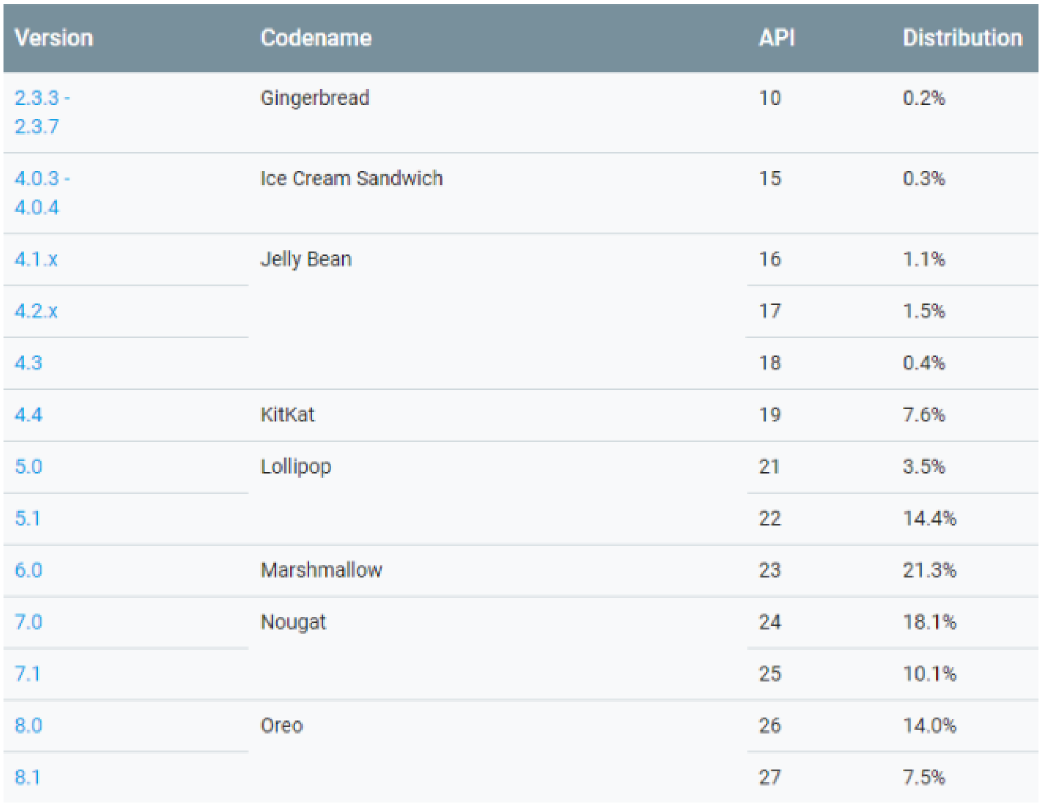
- OR2: GPS data is needed for navigational support
  - In order to provide navigational support, the location of the user is needed

7.2.6 Scalability Requirements

- SCR1: The application should support different sizes of parking spaces.
  - Some parking lots are larger/smaller than others. The application should still function correctly.

7.2.7 Maintainability and Portability Requirements

- MPR1: Application will only run on android devices version 5.0 or newer
  - This is to ensure the application supports most android devices.



Version	Codename	API	Distribution
2.3.3 - 2.3.7	Gingerbread	10	0.2%
4.0.3 - 4.0.4	Ice Cream Sandwich	15	0.3%
4.1.x	Jelly Bean	16	1.1%
4.2.x		17	1.5%
4.3		18	0.4%
4.4	KitKat	19	7.6%
5.0	Lollipop	21	3.5%
5.1		22	14.4%
6.0	Marshmallow	23	21.3%
7.0	Nougat	24	18.1%
7.1		25	10.1%
8.0	Oreo	26	14.0%
8.1		27	7.5%

7.2.8 Cultural and Political Requirements

- CPR1: The product will not include any messages or images that will offend parties of any culture or religion.
  - The application is targeted to users age 16+ (or driving age). Users should not see any offensive messages or images. If ads are present, they should be curated first to enforce this.
- CPR2: The product will comply with current Canadian cultural standards

7.2.9 Legal Requirements

- LR1: All assets used are either original works. It will not infringe on any copyrights and appropriate permission/licensing will be acquired.
- LR2: The product will comply with Canadian copyright laws.

7.2.10 Health and Safety Requirements

- HSR1: Our application interface will pose no threat to epileptic prone users.
- HSR2: The system will provide hands-free features that will allow the user to use some functionality of the application without using their hands such as speech-to text input and text-to-speech navigation output.
  - As the user is operating a motor vehicle, the system must not distract the driver to adhere to driving laws.

7.3 Undesired Event Handling

In the event of a undesired event that is not a result of a previously stated assumption, the system will be designed in such a way that it will self-correct any discrepancies. This means that any desynchronization issues between the application and the servers will be fixed with the use of multiple users supplying data to the database. Additionally this will be solved by having periodic updates so that the most recent data is used. Finally, in the case of an undesired event that is not covered, the use of error handling and error messages will be utilized to prevent any confusion for the user and preventing any disastrous errors within the system.

8 References

1. Norman, D. The design of everyday things. Basic Books, New York, NY, 2013.

2. Google, Distribution Dashboard, Android Developers, *https://developer.android.com/about/dashboards/* 2018