Software Requirements Specification

for

Citrus

Version 1.0 approved

Prepared by

Christian Ndaye
Michael Anthony
Adam Moritz

Grizzly Mobile

10/21/2018

Table of Contents

Revision History

Name	Date	Reason For Changes	Version

1. Introduction

1.1 Purpose

The citrus smartphone application will enable customers of the application to utilize cheap, environmentally conscientious transportation. Electric bicycles and scooters will be available to the users of the app. Viewing the availability of the rentals will be made available using Google maps. The user will see the rental stations that are spaced out in cities, towns even college campuses. At these rental stations there will be a QR code available to scan using the app. A customer will have a payment channel linked to their account and will release the bike or scooter from the rack once this QR code is scanned with their device. Upon picking a rental the customer is than billed .05 cents a minute immediately after checking a rental out. The user can pause their ride at any time, during the rental period. Doing so will lock the bikes functionalities, rendering it unusable during this time. We wish to bring fun, environmentally friendly transportation to the world, one app user at a time.

1.2 Document Conventions

Our conventions for this product are for the users to have a cleaner alternative to transportation all while enjoying a fun ride in their town or city. The Citrus logo and all other trademarks of our brand are copyright protected. Are company will enable users to pay with cryptocurrencies and we are not subject to or responsible for loss of assets, or fluctuating prices.

1.3 Intended Audience and Reading Suggestions

The Citrus software requirements specifications document is intended for readers of any caliber to understand and appreciate the smartphone application with its many functions and operations. For more technologically advanced readers, you may precede to sections 3 External Interface Requirements and 4 System Features for a more comprehensive view of the system. Stakeholders, marketing staff and users will find it beneficial to start reading from section 2, Overall Description, as it will explain the application from a technologically lighter, top-down approach and feature use case diagrams for easy understanding of the apps processes.

1.4 Product Scope

The Citrus application will be a smartphone app intended for release first on the IOS market. It will enable features such as Google Maps, eventually cryptocurrency payments, and of course the rental of electric bikes and scooters from various locations in the United States. The app will eventually be available on the Android market come early 2020. 2020 will also be the year for the launch of our platforms very own cryptocurrency Citrus Coin. The user will be able to pay for their rides using this mode of payment as well as rack up citrus coin in their wallet while using our service in a reward-based program.

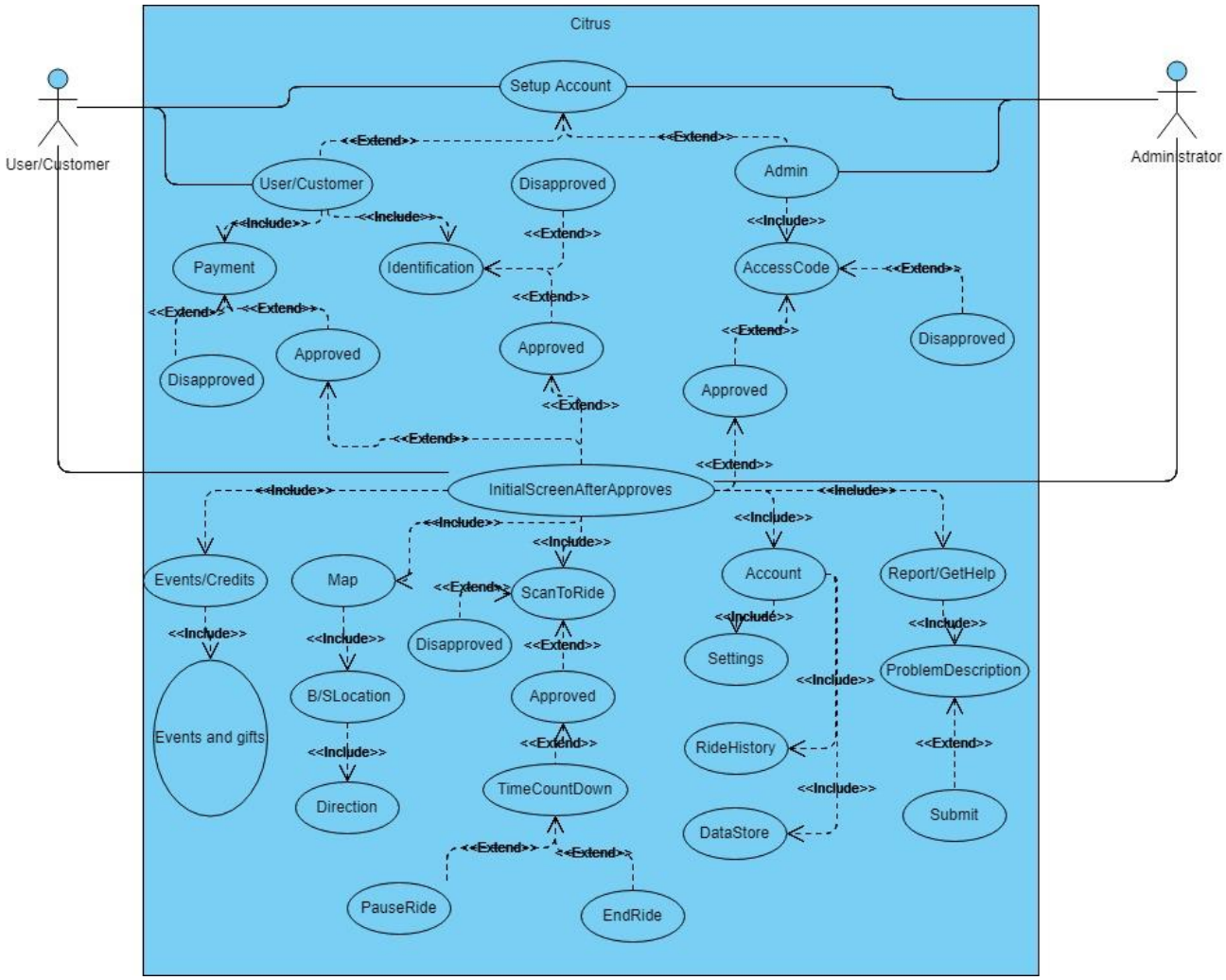
1.5 References

A website will be released mid 2019 that will be the official site of Citrus bikes and scooter shares incorporated. The Citrus cryptocurrency is scheduled to be released on KuCoin and Binance early to mid 2020.

2. Overall Description

2.1 Product Perspective

Citrus application is a new self-contained product that operates with electric bikes and scooters.



2.2 Product Functions

The Citrus App will allow first time users to create an account first, including their payment method and identification, and then allow users to change their info in their account, if needed to, through settings. The initial screen will allow users to view/direct to the location of the available electric bikes/scooters, view their ride history and their data stored, report or ask for help, view events and credits, and mainly to scan a QR code on the electric Bike/scooter to unlock it and ready to use. When an electric bike or scooter is unlocking, the App will start the time countdown so that the bill can be processed based on the time used. The User can either pause a ride if needed to or end the ride.

2.2 User Classes and Characteristics

The Citrus App will have an “account” class to store user’s info e.g. name, address, Student/faulty/staff ID, ride history, and other data to be stored. The account class will be implemented so that a user can first set up an account before using the App and can be accessed later in the App if changes are needed to be made in the future via settings class. A bank info class will be implemented to store banking info so that the App can check if the users has enough funds to scan and ride. A map class will be implemented to view the location of the electric bikes/scooters and direct them to them. If any issue occurs, the App will contain a report class that will allow users to report any issue and get help. To access any of the ride, a switch class will be needed to unlock the electric bike/scooter by scanning a QR code, and will allow a time countdown, so that billing can be processed. In the switch class, the ride can be paused and ended whenever the user needs to. Lastly, the main class will be citrus to act as an umbrella of all the classes.

2.4 Operating Environment

The Citrus App will run on iOS devices, e.g all version of iPhone. The App will require the coexistence of a google map, the school ID system, and banking system.

2.5 Design and Implementation Constraints (Optional)

Access to school ID system.

2.6 User Documentation

The Citrus App will have a built-in report documentation that will allow user to file whenever an issue rises.

2.7 Assumptions and Dependencies

Banking, school ID system, and google map.

3. External Interface Requirements

3.1 User Interfaces

Citrus is intended to have eight different screens with each correlating to one of the core functions of the app. There are screens for account setup, a GPS map view, QR code scanning, an options menu, settings and account information, ride history, gift redemption and events, and problem reports. The account setup screen consists of a series of small pop-up prompts with a “Submit” and “Cancel” at the bottom; in order of appearance on the setup screen, the pop-up prompts for inputting a college-issued ID number, providing credit card/debit/cryptocurrency information, and enabling location services. The GPS map view will show a map of the user’s location and have pins representing rental stations. The GPS map view acts as the main screen and has buttons for accessing the options menu, gifts and events screen, and the QR code scanner along with a button to lock onto the location and show the direction of the user’s phone. The options menu is a screen with a blurred background overlaying the main GPS view; there are buttons for accessing the settings, ride history, and problem report screens stacked on top of each other in the menu screen. Settings is a screen with editable fields for user identification and billing information and a small “Close” button in one of the upper corners. Ride history contains a small “Close” button in the upper corner of the screen and non-editable fields that display ride distances, vehicles rented, and ride times. Problem reports displays a small “Close” button in the upper corner of screen a single editable field with which to write a description of the problem and a “Submit” button at the bottom of the screen. The gift redemption and events screen will have a small “Close” button in the upper corner and a list of redeemable gifts and

rewards for events. Error messages should come in the form of pop-up messages that overlay and blur the current screen in use by the user and contain a short description of the error and an “Ok” button.

3.2 Hardware Interfaces (Optional)

The Electric bike will interface with GPS location tagging. It will be monitored via GPS location to enable location tracking to track distance of travel by customer and will be used incase of theft. The bike racks will interface with the rentals by concluding a customer ride once docked. Once docked the rental will become engaged and locked into the bike share racks and will commence charging.

3.3 Software Interfaces

The Citrus app will be written in XCode for the iOS mobile operating system. Citrus will use the Google Maps API to send and receive location and GPS data between Google’s network and the app’s map view on the user’s phone. The app should also use student and faculty databases from participating colleges for verifying the student or faculty identification number provided by the user. Data shared across software components shall include the user’s location, GPS maps information, and student or faculty ID number.

3.4 Communications Interfaces

Citrus is intended to communicate with Google’s map network, college student and faculty ID databases, bank accounts, and electronic forms regarding problems encountered by users of the app. Electronic forms will be created and sent in a simple text format. The FTP communication standard is likely to be used with this product to send and receive data between various servers and the user’s phone. Connections with each network will be done through Wi-Fi or mobile data streaming as provided by the user’s phone service provider; data transfer rates are expected to vary depending on the type of internet service the user is accessing at the time of using Citrus. Some potential security and encryption issues include the accessing and storage of private user information such as billing information, student or faculty ID’s, and the user’s location.

4. System Features

4.1 Account Setup

4.1.1 Description and Priority

Citrus shall allow the user to create an account that will provide store identification and billing information to be used by the other features of the app. This feature is of High priority.

4.1.2 Stimulus/Response Sequences

1. The User opens the app for the first time.
2. Citrus displays a prompt to input the User's student or faculty ID number.
3. The User inputs their college-issued ID number and presses "OK."
4. Citrus displays a form for the User's credit/debit card information.
5. The User inputs their credit/debit/cryptocurrency information and presses "OK."
6. Citrus prompts the User to activate location services for the GPS map view.
7. The User presses the "Enable location" button to activate location services.
8. Citrus saves all input information and allows the User to access the rest of the app.

4.1.3 Functional Requirements

REQ-1: The app must ask for and verify the identity of the user through credentials they provide within a pop-up prompt.

REQ-2: An account must not already be in use on the user's phone in order for a new account to be created.

REQ-3: All private user-input information must be verified and stored securely to prevent data and identity theft.

REQ-4: An error message should be displayed if incorrect data is input into any of the prompts. The app will re-display the prompt but will include an option to cancel if the user so chooses.

System Feature Conti.

4.2 GPS Map View

4.2.1 Description and Priority

The app shall display a map of the user's surroundings and indicate the location of rental stations. This feature is of Medium priority.

4.2.2 Stimulus/Response Sequences

1. Citrus automatically displays a GPS map view to the User when no other use case is active.
2. The User navigates the map via touch screen to locate nearby rental stations.

4.2.3 Functional Requirements

REQ-1: The app should access map data from Google Maps of the area immediately around the user's phone.

REQ-2: The app must be connected to the internet be capable of sending and receiving data to and from the Google Maps network.

REQ-3: The user must have location services enabled on their phone in order for Citrus to determine the correct map to display at any given time.

REQ-4: Failure to connect to the Google Maps network will display an error message in the form up a pop-up alert and temporarily disable the map view until a connection is reestablished.

REQ-5: Citrus should indicate the exact location of nearby rental stations within a certain range of the user.

4.3 Rent a Vehicle

4.3.1 Description and Priority

The user shall be able to choose and rent a bike or scooter from a Citrus rental station. This feature is of High priority.

4.3.2 Stimulus/Response Sequences

1. The User presses the “Scan Ride” button.
2. Citrus displays a scanner camera screen to the User.
3. The User uses the scanner camera to scan a QR code on the bike or scooter they wish to rent.
4. Citrus unlocks the bike or scooter from the rental station for the User to ride and starts a timer to calculate how long the ride lasts.

4.3.3 Functional Requirements

REQ-1: Citrus should include a button that activates a QR code scanner for renting a bike or scooter of choice from any Citrus rental station.

REQ-2: Errors with the QR code scanner shall display an error message in the form of a pop-up alert and allow the user attempt to scan again or cancel the scan if they so choose.

4.4 Return a Vehicle

4.4.1 Description and Priority

The user shall be able to return a bike or scooter they have rented from a Citrus rental station to any other Citrus rental station. This feature is of High priority.

4.4.2 Stimulus/Response Sequences

1. The User attaches their rented bike or scooter to a rental station and presses the “End Ride” button.
2. Citrus stops the ride timer, locks the bike or scooter and bills the User for the ride based on the total amount of time the vehicle was in use.
3. The User pays for the ride automatically with the credit/debit/cryptocurrency information saved on their account.

4.4.3 Functional Requirements

REQ-1: Citrus should include a button that allows them to return and pay for the bike or scooter they rented.

REQ-2: The user should not have to input their billing information again after providing it during the account setup process.

REQ-3: In case of an error when returning a bike or scooter, the app will display an error message in the form of a pop-up alert and redirect the user to a problem report form. The app will flag the error and lock the bike or scooter in place until someone can be dispatched to fix the issue. If possible, Citrus will charge the user for the ride as it is recorded in their ride history up until the return error.

4.5 View Account

4.5.1 Description and Priority

Citrus shall provide a menu of options related to the user's account. This feature is of High priority.

4.5.2 Stimulus/Response Sequences

1. The User presses the "Menu" button on the main screen of the app.
2. Citrus responds by displaying a series of buttons correlating to ride history, tech support, and settings and User information.
3. The User selects the account option they wish to view.
4. Citrus displays a screen for the selected option.

4.5.3 Functional Requirements

REQ-1: The app should have a button for the user to access an options menu that includes additional buttons for account information and settings, tech support, and ride history.

REQ-2: The user must be able to cancel or exit out of the account menu at any time.

REQ-3: Errors will be resolved by redirecting the user to the main map view screen and displaying a pop-up error message alert.

4.6 Report a Problem

4.6.1 Description and Priority

The user shall have the ability to inform members of tech support and maintenance of any problems they have encountered related to the app, rental stations, or vehicles. This feature is of Medium priority.

4.6.2 Stimulus/Response Sequences

1. The User presses the “Report a Problem” button located within the account menu.
2. Citrus displays a problem report form to the User to fill out.
3. The User fills out the form with a description of the problem they encountered.
4. Citrus sends the completed problem report to the Administrator for review.

4.6.3 Functional Requirements

REQ-1: The user should have access to a button to cancel or otherwise exit out of a problem report at any time.

REQ-2: Problem reports will have no text limit and allow the attachment of image files for proof of physical issues with any Citrus vehicles or rental stations.

4.7 Description and Priority

Citrus shall provide a menu of options related to the user's account. This feature is of Low priority.

4.7.1 Stimulus/Response Sequences

5. The Administrator selects the "Gift CitrusCoins" options on their terminal and specifies a number of CitrusCoins to send to the User.
6. Citrus adds the specified number of CitrusCoins to the User's account.

4.7.2 Functional Requirements

REQ-1: CitrusCoins should be automatically added to the selected user or users' accounts without requiring users to redeem them.

REQ-2: Gifts should not expire after any length of time or under any circumstance.

REQ-3: Errors will result in the displaying of an error message on the administrator's terminal and prompt them to either try to send the gift again or cancel.

5. Other Nonfunctional Requirements

5.1 Performance Requirements

To use the electric bike or scooter, a user must first scan a QR code on the bike or scooter. Citrus App will check whether the user has enough money in their account so that access can be granted, and then it's required to have a time countdown after the access has been granted. The billing depend is depend on the time the user spends with the electric bike or scooter. It's also required to leave the electric bike or scooter at an appropriate and designated area after use.

5.2 Safety Requirements

It's required for a user to ride the electric bike or scooter safely. If needed, a user might wear a helmet. It will require for users to ride on the right side of a path when other people are using that same path.

5.3 Security Requirements

The Citrus App should secure user info/banking and usability data. To secure the electric bikes and scooters, they will always be locked, unless being used with a user.

5.4 Software Quality Attributes

User will be awarded with crypto bonus once they have ridden a specific number of rides. The usage data, like miles per hour, number of rides with their prices, carbon saved, and calories, will be stored. The Citrus App will have special events, like a bike/scooter race, and winners gets prizes.

5.5 Business Rules

Team members are required to maintain the Citrus App while providing what customers need and want, and make the App easily interactive, and fun to use.

6. Other Requirements

Citrus is not responsible for any bodily harm while customers are using our rentals. Repairs and servicing of the rentals will be carried out on a routine basis. Customers will be able to report problems and malfunctions using the citrus app by taking pictures of the damaged rental and sending in a report ticket.

Appendix A: Glossary

Bike and Scooter Sharing (Bike Share) – A bicycle-sharing system, public bicycle system, or bike-share scheme, is a service in which bicycles are made available for shared use to individuals on a short-term basis for a price.

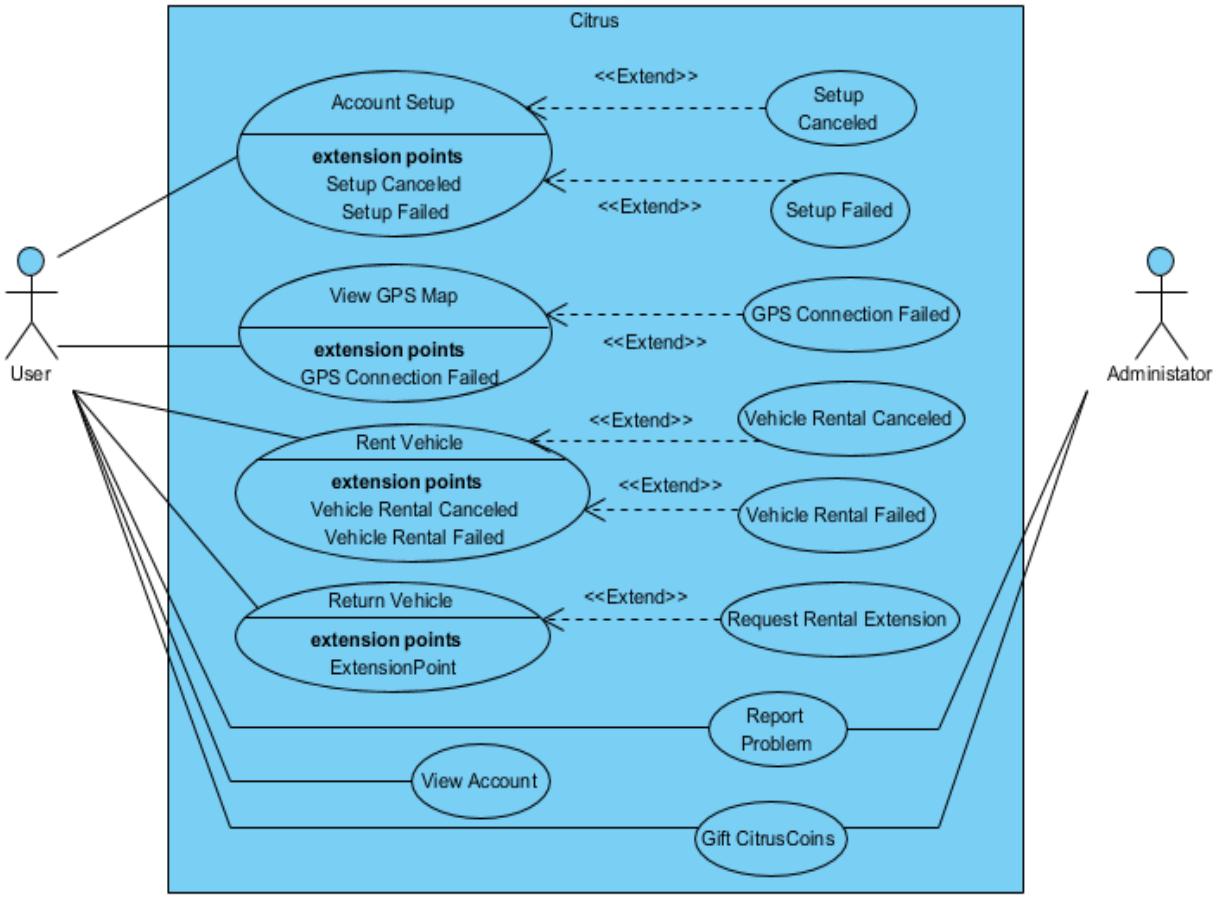
Cryptocurrency – a digital currency in which encryption techniques are used to regulate the generation of units of currency and verify the transfer of funds, operating independently of a central bank.

Citrus – The smartphone app that enables its users to find and select a bike or scooter at a bike share center to rent for a nominal fee.

QR Code – A machine-readable code consisting of an array of black and white squares, typically used for storing URLs or other information for reading by the camera on a smartphone.

Pop-ups – This term is used for when small message boxes pop up to notify the user of errors or for permission to allow certain functionalities of their smartphone to be utilized by the app.

Appendix B: Analysis Models



Use case name	Account Setup
Participating actors	Initiated by User
Flow of events	1. The User opens the app for the first time. 2. Citrus displays a prompt to input the User's student or faculty ID number. 3. The User inputs their college-issued ID number and presses "OK." 4. Citrus displays a form for the User's credit/debit card information. 5. The User inputs their credit/debit/cryptocurrency information and presses "OK." 6. Citrus prompts the User to activate location services for the GPS map view. 7. The User presses the "Enable location" button to activate location services. 8. Citrus saves all input information and allows the User to access the rest of the app.
Entry condition	• The User has opened the app for the first time
Exit condition	• The User has filled out all identification information. • The User has turned on location services.

Use case name	Setup Canceled
Participating actors	Communicates with User
Flow of events	…
Entry condition	This use case extends the Account Setup use case. It is initiated when the User chooses not to proceed with the account setup process.
Exit condition	…

Use case name	Setup Failed
Participating actors	Communicates with User
Flow of events	…
Entry condition	This use case extends the Account Setup use case. It is initiated when the User either inputs invalid identification or billing information or if the app is unable to verify the information input by the User.
Exit condition	…

Use case name	View GPS Map
Participating actors	Communicates with User
Flow of events	1. Citrus automatically displays a GPS map view to the User when no other use case is active. 2. The User navigates the map via touch screen to locate nearby rental stations.
Entry condition	• The User has opened the app.
Exit condition	• The User has initiated a different use case.
Quality requirements	Citrus hides and displays availability indicators at each station as bikes and scooters become available or unavailable.

Use case name	GPS Connection Failed
Participating actors	Communicates with User
Flow of events	…
Entry condition	This use case extends the View GPS Map use case. It is initiated when the network connection between the app and GPS is lost.
Exit condition	…

Use case name	Rent Vehicle
Participating actors	Initiated by User
Flow of events	1. The User presses the “Scan Ride” button. 2. Citrus displays a scanner camera screen to the User. 3. The User uses the scanner camera to scan a QR code on the bike or scooter they wish to rent. 4. Citrus unlocks the bike or scooter from the rental station for the User to ride and starts a timer to calculate how long the ride lasts.
Entry condition	• The User has pressed the “Scan Ride” button.
Exit condition	• The User has successfully scanned a bike or scooter.
Quality requirements	• Citrus notifies the User if the scan is invalid or did not go through. • Citrus redisplays the scanner and allows them to try again.

Use case name	Vehicle Rental Canceled
Participating actors	Communicates with User
Flow of events	…
Entry condition	This use case extends the Rent Vehicle use case. It is initiated when the User chooses not to proceed with the vehicle rental process.
Exit condition	…

Use case name	Return Vehicle
Participating actors	Initiated by User
Flow of events	1. The User attaches their rented bike or scooter to a rental station and presses the "End Ride" button. 2. Citrus stops the ride timer, locks the bike or scooter and bills the User for the ride based on the total amount of time the vehicle was in use. 3. The User pays for the ride automatically with the credit/debit/cryptocurrency information saved on their account.
Entry condition	• The User has successfully rented a bike or scooter.
Exit condition	• The User has paid the bill for the ride.

Use case name	Request Rental Extension
Participating actors	Initiated by User
Flow of events	…
Entry condition	This use case extends the Return Vehicle use case. It is initiated when the user has elapsed 24 hours of ride time and wishes to extend their rental for an additional 24 hours.
Exit condition	…

Use case name	View Account
Participating actors	Initiated by User
Flow of events	1. The User presses the "Menu" button on the main screen of the app. 2. Citrus responds by displaying a series of buttons correlating to ride history, tech support, and settings and User information. 3. The User selects the account option they wish to view. 4. Citrus displays a screen for the selected option.
Entry condition	• The User has pressed the "Menu" button.
Exit condition	• The User has exited the account menu by pressing the "Close" button.
Quality requirements	• Citrus notifies the User if they do not have anything on record in their ride history yet. • The User's account information is automatically filled back in with the previously stored information if they leave a changed field blank upon closing the account menu.

Use case name	Report Problem
Participating actors	Initiated by User Communicates with Administrator
Flow of events	1. The User presses the "Report a Problem" button located within the account menu. 2. Citrus displays a problem report form to the User to fill out. 3. The User fills out the form with a description of the problem they encountered. 4. Citrus sends the completed problem report to the Administrator for review.
Entry condition	• The User has pressed the "Report a Problem" button.
Exit condition	• The User has successfully reported a problem or closed the report.
Quality requirements	• The Administrator will try to respond to critical problems within 24 hours of receiving a report.

Use case name	Gift CitrusCoins
Participating actors	Initiated by Administrator Communicates with User
Flow of events	1. The Administrator selects the "Gift CitrusCoins" options on their terminal and specifies a number of CitrusCoins to send to the User. 2. Citrus adds the specified number of CitrusCoins to the User's account.
Entry condition	• The Administrator has selected the option to gift CitrusCoins to the User.
Exit condition	• The User has received a specified amount of CitrusCoins.
Quality requirements	• The Administrator can choose to gift the User CitrusCoins at timed intervals or randomly rather than manually.

Appendix C: To Be Determined List

White paper for Citrus Coin and Android app will be released early 2020. Website due for release mid-2019.