Software Requirements Specification

for

CarMa

Version 1.0

Prepared by Team 05
O'neal, Emily (emioneal)
Huang, Tianze (tianhuan)
Prabowo, Gabriela (gprabowo)
Dianprakasa, Arif (arifdian)

Associate Instructor
Shah, Kunaal (kupshah)

November 24, 2019

Table of Contents

1. Introduction

1.1 Purpose

This document fully describes the full requirements for the application Car-Ma, which acts as a service medium for user/customer to rent a car online on their smartphone. The full information and technical details about the application structure and development plans are addressed below.

1.2 Document Conventions

The functional requirements for this project are classified by the use case list with respect to mainly one user class which is the customer using this car-sharing app.

1.3 Project Vision and Product Scope

1.3.1 Vision Statement

The concept of car-sharing app has been applied to many different areas. However, relatively college town like Bloomington had not been met with satisfaction when it comes to transportation. With Car-Ma, the application provides many different options for the Bloomington community be it for the students or the local residents. features such as different car sizes from trucks to hatchback provide the specific needs for a wide variety of customers which the town Bloomington is lacking. Besides, wide availability of parking spot around the campus area would serve as an advantage for an alternate commute especially when time is on the essence.

1.3.2 Product Scope

Car-Ma is an iOS application that provides a car sharing platform for the customer to register and rent an electric car for a period of time. For the age of 18 and above together with valid US driver's license, a user is able to register and create an account. Car locations, size and time availability will be listed on the main page for direct accessibility for the user. User can also modify / cancel reservations prior to the pick up time. User can also add payment methods and view past history of renting.

1.4 References

Existing websites used as an inspiration for ideas regarding the functionality documented here include:

www.grab.com.sg
www.uber.com
www.zipcar.com/about
www.stickerride.com/about
www.lyft.com

2. Overall Description

2.1 User Classes and Their Use Cases

2.1.1 Customer

A Customer is someone who uses a Car-Ma app to reserve a car.
Use cases:

UC-C-1: Register as a new user
UC-C-2: Login as a registered user
UC-C-3: Reset password
UC-C-4: Enter / add payment methods (Credit card information)
UC-C-5: Enter driver's license information
UC-C-6: Use map to locate a vehicle
UC-C-7: Choose the type of vehicle
UC-C-8: Choose a timeframe for pickup and dropoff
UC-C-9: Reserve a vehicle
UC-C-10: Summary of reservation (Confirmation)
UC-C-11: Edit / Cancel a reservation
UC-C-12: Report car's condition (upload photos of the car)

2.1.2 Administrator

An Administrator is responsible for maintaining the Car-Ma app.
Use cases:

UC-A-1: Organize and verify user information (license information)
UC-A-2: Suspend user (delete account)
UC-A-3: Process and validate refund requests
UC-A-4: Edit / Add vehicle information
UC-A-5: Generate reports
UC-A-6: Generate payment request
UC-A-7: Infrastructure maintenance

2.2 Operating Environment

Car-Ma shall be compatible with standard iOS smartphone platforms. Supported browsers are Opera 10 and 11; Google Chrome 10 and later. [iOS platform is chosen due to wide usage of Iphone and the language has great view controller model]

2.3 Design and Implementation Constraints

CO-1: The administrator shall be able to maintain the application by creating reports and organizing reports from users to the developer team working in the background (not included as full requirement).

CO-2: The administrator needs some sql background knowledge to tabulate reports / requests from user.

CO-3: We do not have a way to actually confirm a driver's license as of right now.

2.4 User Documentation

The developer is obliged to provide necessary documentation and support assistance to enable an Administrator to perform the use cases listed in section 2.1.4 Administrator.

2.5 Assumptions and Dependencies

There is an assumption that the user is at least 18 years old. A dependency is that we will be using PayPal and Google Maps API.

3. Functional Requirements

3.1 Use Cases for Customers

UC-C-1: Register as new user

Description: The user needs to create an account in order to use the service of our vehicles.

register.create:	In order for the user to user must create an account. Each registration will have the following fields: 1. users email (required) 2. users password (required) 3. user retype password (required) 4. users birthday (required) 5. users phone number (required)	(Priority=H)
register.agreement:	In order for the user to finish creating their account they must click the box to agree to the terms and conditions of our app. This will be represented by a box at the bottom of the page and will have a check in it once they have	(Priority=H)

	clicked it. The user will be held accountable for knowing the terms.	
register.valid.email:	The user will receive an error if they enter an invalid email. To be valid it will only contain letters and numbers, hyphens(-), and underscores(_) are allowed before the "@." It must contain only one "@." Periods (.) are allowed before and after the "@." There has to be at least one letter after the "@." The password cannot already be in use for the app.	(Priority=H)
register.match.password:	The password and re-type password must match each other or the user will receive an error to try typing them again to make sure they match each other.	(Priority=H)
register.secure.password	In order to have a secure password there are some requirements when creating one. It must be at least 8 characters, contain at least 1 number, at least one special character.	(Priority=H)
register.valid.agreement:	If the user does not click the agreement box then when the screen reloads, an error message will show that says it is required to agree in order to make an account.	(Priority=H)

3.2 UC-C-2: Login as registered user

Description: The user needs to login into their account if they already have one in order to use the app.

login.email:	The first thing to do to sign in is to enter the email that they registered with.	(Priority=H)
login.email.error:	If the user enters an email that is not registered there will be an error. There will also be an error if the format of the email entered is incorrect, see register.valid.email for the format.	(Priority=H)
login.password:	The user has to enter the password that they used when creating the account.	(Priority=H)
login.password.error:	If the user password does not match the account of the user email then the user will get an error that the password and email do not match. If the password does not meet the password requirements or was entered wrong, see register.secure.password to see the requirements.	(Priority=H)

UC-C-3: Reset Password

Description: If the user needs to reset their password because they forgot it or it is no longer secure.

reset.password:	if the user is trying to login and they forgot their password they can click "forget password?" The user is then prompted to enter their email or phone. A link to reset it is then set to their email or phone.	(Priority=H)
reset.timeout:	The user will have 10 minutes to use the link before it expires. This is a security measure that we want to take.	(Priority=H)
reset.error:	If the user tries to reset their password more than 2 times in one day then the user will get an error that their account is locked for the day.	(Priority=H)
reset.password.email:	This is 1 of 2 options to reset their password. The user can enter the email associated with their account.	(Priority=H)
reset.password.phone:	This is 1 of 2 options to reset their password. The user can enter the phone number associated with their account.	(Priority=H)
reset.link:	This is the link that will be sent to their email or phone. They must click the link in order to enter a new password.	(Priority=H)

UC-C-4: Enter / add payment methods (Credit card information)

Description: In order for the user to pay for their ride they need to enter their credit card information.

card.info: The user must enter their credit card information. This information includes: (Priority=H)

1. card number (required)
2. card CVC (required)
3. card expiration date, format mm/yyyy (required)
4. cardholders first name (required)
5. cardholders last name (required)
6. users billing address (required)
    a. city (required)
    b. state (required)
    c. zipcode (required)

card.number.error:	If the format of the card number is not entered correctly, there will be an error that it was entered wrong. The format is nnnn nnnn nnnn nnnn. There is also an error if the user tries to enter anything that is not a number.	(Priority=H)
card.cvc.error	If the format or the cvc does not match the card the user will see an error.	(Priority=H)
card.name.error	If the user tries to put anything that is not a letter in the space then they will see an error. This means no numbers or symbols even if people have symbols in their name such as an apostrophe (').	(Priority=H)
card.required.error:	If one of the required spaces is not filled the user will not be able to go on before filling it in correctly.	(Priority=H)

UC-C-5: Put in the driver's license information

Description: In order to use our service the user must enter their drivers license information.

license.info:	The user must have their approved drivers license information in their profile if they want to use the app. The information needed is: 1. their name (required) 2. location their license was issued in 3. drivers license number (required) 4. re-type drivers license number (required) 5. the address on their license (required) a. address (required) b. city (required) c. state (required) d. zipcode (required)	(Priority=H)
license.number.match:	The user must enter their drivers license number twice to make sure they do not make a mistake. There will be an error if the two do not match each other.	(Priority=H)
license.name.error:	If the user does not follow the format of a name then there will be an error or if they do not enter a name at all there will also be an error.	(Priority=H)
license.required:	If the user does not fill in one of the required spaces then there will be an error until the space is filled properly.	(Priority=H)

UC-C-6: Use map to locate vehicle

Description: To continue, we must use the map to locate the vehicles that are available around the area

locate.vehicle:	User must have their location on in order for them to find the nearest vehicle available for them to reserve.	(Priority=H)
locate.vehicle.error:	If the user does not have their location on, they will not be able to find the nearest vehicle available.	(Priority=H)
location.error	Not able to locate the user's location, need to make sure that the internet is working or quit the application and open it up again.	(Priority=H)

UC-C-7: Choose the type of vehicle

Description: After picking the location, users must choose the type of vehicles that are available in a location that is the nearest to them

pick.vehicle:	There will be a certain number and type of vehicles that are available for reservation in that location. Users are able to pick which vehicles that they are interested in.	(Priority=H)
pick.vehicle.error:	If the users are not quick in deciding which vehicles they are planning to reserve, someone else might have reserve the car and users could opt for another type of vehicle	(Priority=H)
internet.vehicle .error:	Not being able to reserve a vehicle since the internet on the phone is not working. Users should make sure that the cellular network and location is on.	(Priority=H)

UC-C-8: Give a time frame for pick-up and drop-off

Description: Time frame must be given for users to pick-up and drop-off the vehicles that they chose.

time.vehicle:	Users will have to give a pick-up and drop off time frame in order to know if the car that they have chosen are available in the specific location that they chose	(Priority=H)
pick.vehicle.error:	If the users are not quick in deciding which vehicles they are planning to reserve, someone else might have reserve the car and users could opt for another type of vehicle	(Priority=L)

UC-C-9: Reserve a vehicle

Description: Users must click the reserve button once they have decided the vehicle and the pick-up and drop-off time

reserve.vehicle:	User successfully reserved the vehicle that they were planning on reserving.	(Priority=H)
reserve.vehicle.error:	Most of the time, other users will book the vehicle a few minutes ahead, and if this happens, you will have to choose another vehicle and time at the same location. Another option is to look for an area that is closest to you.	(Priority=H)

UC-C-10: Summary of reservation

Description: An admin should be able to summarize all the information the user has chosen before they pay for the reservation

summary.vehicle:	Admin must be able to create a summary of the user's reservation	(Priority=H)
summary.vehicle.location:	Admins must be able to know the location of the vehicle	(Priority=H)
summary.vehicle.time:	Admins must be able to know the time of the vehicle	(Priority=H)

summary.vehicle.type:	Admins must be able to know the type of the vehicle	(Priority=H)
summary.vehicle.battery:	Admins must be able to know the battery of the vehicle	(Priority=H)
summary.vehicle.price:	Admins must be able to know the price of the vehicle	(Priority=H)

UC-C-11: Modify reservation

Description: Users must click the change button once they have changed vehicle or the pick-up or drop-off time (reservation) and confirmed the changes to reservation.

modify.reservation:	User successfully changed the reservation and information are confirmed.	(Priority=M)
modify.reservation.error:	While modifying the reservation, sometimes people might fail to change the reservation since other user reserve the same time or same car earlier than user did, so user can try to modify the reservation again to find either cars or location that fits to the user.	(Priority=M)

UC-C-12: Take photos of the car before and after operating the vehicle

Description: Users must take photos before and after operating the vehicle in order to check the status of the vehicle and the photo will be saved to administrator database in order to keep the record.

take.photos:	User takes photos of the car successfully and the photo is uploaded into the database.	(Priority=H)
take.photos.error:	While taking photos, sometimes people are not giving access to photo, and user has to change them in their phones' settings. While uploading photo failed, user has to retake the photo and update photos.	(Priority=H)

3.2 Use Cases for Administrators

UC-A-1: Verify License

Description: An administrator has to verify the license through BMV or other sources to verify the license of users and report if they are accepted or declined because of out of date or wrong information.

license.verify:	An Administrator should be able to record the license information and contact BMV to ensure the information is correct.	(Priority=M)
license.verify.error:	If a license is verified error, then we need to require user to upload a new license since the old one has expired or the information is wrong.	(Priority=M)

UC-A-2: suspend user

Description: If the user does not follow the rules then their account can be suspended.

suspend.access:	If the user does to follow the terms and conditions of the app and it is brought to the administrations attention then their access will be taken away.	(Priority=L)
suspend.error:	If the system is down then there will be an error that the access denied was not complete.	(Priority=L)

UC-A-3: Give refunds

Description: When users had modified reservation or user cancels the trip, system or administrator has to refund the user due to the condition happened to the current trip.

refunds.given:	An administrator should record the current status of a trip and decide if the refunds should be given or not.	(Priority=M)
refunds.type:	If a trip is cancelled, the administrator should refund all costs but not including cancellation fee. If a trip is modified such as changing types of cars, or location before the due time,	(Priority=M)

administrator should refund all price, or modified difference to users.

UC-A-4: Edit / Add vehicle information

Description: An Administrator shall be able to modify the contents of any page on the website.

info.modify:	An Administrator will be able to modify information on the app such as the car descriptions or terms of service.	(Priority=H)
info.upload:	An Administrator will be able to upload new information or new vehicles that are available to the customers.	(Priority=H)

UC-A-5: Generate reports

Description: The app will generate reports for the administration to see if and how everything is maintaining/working.

report.system:	The app will collect data when the user rents a car such as: 1. the average time of rental 2. how many rentals in the month 3. popular rental times 4. popular locations 5. photos of the vehicles 6. crashes 7. total users	(Priority=L)
report.error:	If the app at any time during the month is down for any reason it will not be able to generate information during that time.	(Priority=L)

UC-A-6: Modify vehicle options and information

Description: When new updates had happened to vehicles, administrator should update the information and collect data, put them into the database. Administrators have to follow the latest information and update them online on time.

Vehicle.modify:	An administrator should update data and condition for new vehicles and vehicles that need to be updated. Collect the data and upload them into database.	(Priority=H)
vehicle.types:	New car and old cars should be saved in the same database with time and information provided to demonstrate the current	(Priority=H)

condition of the vehicle. Different types of cars should have different type modules to seperate them.

UC-A-7: Infrastructure maintenance

Description: The system and vehicles will need to go through maintenance at times.

app.maintenance:	The administrators shall be able to access the code that is needed to be fixed or maintained.	(Priority=M)
app.error:	If the code is not organized it will be hard to find and fix problems.	(Priority=M)
vehicle.maintenance:	Since we have an app that deals with vehicles and they will be driven regularly they will need to be maintained if we want to take care of the vehicles.	(Priority=M)

Full Use Case (User, Original Format)

UC-C-1: Register as new user

Use Case ID:	**UC-C-1**		
Use Case Name:	Sign Up/register		
Created By:	emioneal	Last Updated By:	arifdian
Date Created:	November 5, 2019	Date Last Updated:	November 22, 2019
Actors:	User		
Description:	The user will create an account with CarMa. The user will enter a series of information in exchange for a profile.		
Preconditions:	1. The User has the CarMa app on their device		

	2. The user does not already have an account
Postconditions:	User is registered and has an account
Normal Flow:	2.0 Create an account 1. User open the app 2. User will have the option to login or create an account 3. User taps create an account 4. User is required to enter the email as username 5. System checks if the username (email) is taken 6. User is required to enter a password 7. User has to retype password 8. User is required to enter a birthday 9. User is required to enter phone number 10. User checks if the user is at least 18 11. User picks a security question 12. User answers the question 13. The user taps the box to agree to terms 14. The user taps create 15. The system has saved the users information 16. The user is registered
Alternative Flows:	2.1 email / username is taken 1. User is prompted to enter a different email 2. User continues process 2.3 User stops the process 1. User decides to not create an account 2. User taps "cancel" 3. User is taken back to main page
Exceptions:	2.0.E.1 User is not eligible due to age If user is not at least 18 to sign a contract then they will not be able to create an account 2.0.E.2 User is already registered 1. System will notify user that account already exists 2. The user ends registration and logs in
Priority:	High
Frequency of Use:	Once per user and administration
Special Requirements:	1. User must enter all fields of the registration 2. User must have a unique email (username)

	3. User must tap the box to agree to term before signing up
Assumptions:	1. The user has the app 2. The user does not already have an account
Notes and Issues:	1. Users must be 18 years old or older to use the app because they have to agree to our terms which are a contract. Also, the user must be old enough to drive. 2. One issue is that a user could lie about their age and create an account but preventable by license check on the later stages.

UC-C-2: Login as registered user

Use Case ID:	**UC-C-2**		
Use Case Name:	Login		
Created By:	emioneal	Last Updated By:	arifdian
Date Created:	November 3, 2019	Date Last Updated:	November 22, 2019
Actors:	User		
Description:	The user will be asked to enter their already created information in order to enter their account, this allows the user to proceed with their action of requesting a car.		
Preconditions:	1. The user has previously made an account 2. The user is not already logged into the system		

Postconditions:	1. The user will then be logged in 2. The user will be able to see the features that are offered 3. The user can request a vehicle
Normal Flow:	**1.0 Login to the system** 1. The user opens the application 2. The user is prompted to put in their username 3. The user is prompted to enter their password 4. The system will check if this information is valid
Exceptions:	**1.0.E.1 User enters the wrong information** 1. The system will notify the user of the error 2. The user will be able to try again **1.0.E.2 User is already logged in** 1. User will see their account **1.0.E.1 User does not have an account** 1. User will tap "create an account" 2. User will start use case 2
Priority:	High
Frequency of Use:	When the user is not logged in. The user is never automatically signed out
Special Requirements:	None
Assumptions:	1. The user already has an account 2. The user knows their information
Notes and Issues:	None

UC-C-3: Reset Password

Use Case ID:	**UC-C-3**

Use Case Name:	Reset password		
Created By:	emioneal	Last Updated By:	arifdian
Date Created:	November 5, 2019	Date Last Updated:	November 22, 2019
Actors:	User		
Description:	If the user already has an account and needs to login but has forgotten their password there is an option to retrieve or change a password.		
Preconditions:	1. User already has an account 2. User has forgotten password		
Postconditions:	1. User has changed password 2. User has access to account		
Normal Flow:	**3.0 Reset Password** 1. User open app 2. User enters username (Email) 3. User enters password 4. User gets an error about the wrong password 5. User taps "forgot password?" 6. User is prompted to enter either email or phone number 7. User enters email or phone number 8. User goes to their email and opens the email from CarMa or User receives text message with the link to reset password 9. User taps the link in the email or text 10. The link leads to a page 11. User enters a new password 12. User enters the new password again 13. System updates password 14. User goes to the app and enters the new password 15. User has access to account		
Alternative Flows:	**3.1 User forgot email**		

	1. User taps "forgot password?" 2. Prompted to enter email 3. User forgot email used 4. User taps "security question" 5. User enters username 6. User is prompted with their security question 7. User answers the question 8. User can resume to account **3.2 User does not have an account** 1. User is prompted to log in 2. User enters username and password 3. Error message that the account does not exist 4. User realizes they do not have an account 5. User then taps "create an account"
Exceptions:	**3.0.E.1 User has too many attempts** 1. User enters the wrong password 4 times 2. Account is frozen **3.0.E.2 User does not open email within a certain time** 1. User requests an email to be sent 2. User has 10 minutes to open the email and use the link 3. Link expires after 10 minutes
Priority:	High
Frequency of Use:	Whenever user forgets their password, could be more than once per user but also could happen 0 times
Special Requirements:	1. The security question is based on username since each one is unique 2. User must have an account in order to use this
Assumptions:	1. User already has an account 2. Only the user knows the answer to the security question

Notes and Issues:	1. An issue could be that the wrong person gets into the account

UC-C-4: Enter / add payment methods (Credit card information)

Use Case ID:	**UC-C-4**		
Use Case Name:	Enter / add payment methods (credit card information)		
Created By:	emioneal	Last Updated By:	arifdian
Date Created:	November 5, 2019	Date Last Updated:	November 22, 2019
Actors:	User		
Description:	In order for the user to rent a vehicle, they must pay for it. This will be done by the system storing the user's card information. There has to be a system to take payments for the service.		
Preconditions:	1. User has an account 2. User is logged in 3. User has a credit/debit card		
Postconditions:	User will be able to use the service because they provided a payment		
Normal Flow:	**4.0 Enter credit card information** 1. User opens app 2. User goes to settings 3. User goes to payment options 4. User taps "add payment" 5. User enters the name on the card 6. User enters the card numbers 7. User enters the expiration date 8. User enters the CVC 9. User taps "save card"		

	10. User taps "confirm" 11. System authorizes payment
Alternative Flows:	**4.1 User pays when requesting vehicle** 1. User open app 2. User request vehicle 3. A message pops up to add payment before it can continue 4. Return to step 4 5. User receives verification for service
Exceptions:	**4.0.E.1 User has insufficient funds** 1. The user requests a vehicle 2. User does not have payment saved 3. User is prompted to enter payment 4. The user enters payment method 5. System processes 6. User does not have the amount in their account 7. System denies user 8. User is prompted to enter a new method **4.0.E.2 User cannot provide CVC** 1. User request ride 2. Payment is saved a. User has payment saved 3. System requests user to enter CVC 4. User does not know CVC 5. System denies payment method 6. System prompts the user to enter a new method **4.0.E.3 User cancels request** 1. User does not want the vehicle and the time requested 2. User cancels transaction
Priority:	High
Frequency of Use:	At least once per user but could be as frequent as every time the user uses the service
Special Requirements:	1. User can choose to enter payment every time or save it 2. User must confirm CVC before every use

Assumptions:	1. User has an account 2. User is logged in
Notes and Issues:	1. User can have more than one payment type 2. Will be using PayPal's API

UC-C-5: Enter driver's license information

Use Case ID:	**UC-C-5**		
Use Case Name:	Enter drivers licence information		
Created By:	emioneal	Last Updated By:	arifdian
Date Created:	November 5, 2019	Date Last Updated:	November 22, 2019
Actors:	User		
Description:	Since this is a service that provides a vehicle the user must enter their driver's licence in order to use and rent the vehicle.		
Preconditions:	1. User has an account 2. User is logged in 3. User has a valid driver's licence		
Postconditions:	User will be eligible to use service and licence information will be confirmed		
Normal Flow:	**5.0 Enter driver's licence information** 1. User open app 2. User goes to settings 3. User taps "licence information" 4. User taps "add information" 5. User enters name on licence 6. User enters licence name 7. User enters address on licence 8. User will take a photo of licence 9. User will tap "confirm information" 10. System will verify licence and driving record from a specialized license database 11. User will be notified if they are approved		

Alternative Flows:	None
Exceptions:	**5.0.E.1 Users licence becomes invalid** 1. User tries to reserve vehicle 2. Their licence becomes invalid from database check 3. System runs information 4. System will notify the user that their account has been suspended **5.0.E.2 Users allows another person to use account** 1. User allows someone besides themselves to use their account and information 2. If CarMa is alerted of this user will be notified that their account is suspended
Priority:	High
Frequency of Use:	Each person is required to enter one licence number
Special Requirements:	1. One licence per account 2. No learners permits 3. No major violations on record
Assumptions:	1. The licence will match the driver every time
Notes and Issues:	1. Issue: as of now we have no way to confirm this 2. Issue: another person without a licence could use the user's account

UC-C-6: Use map to locate vehicle

Use Case ID:	**UC-C-6**		
Use Case Name:	Use a map to locate a vehicle		
Created By:	gprabowo	Last updated By:	arifdian
Date Created:	November 6, 2019	Date Last Updated:	November 22, 2019
Actors:	User		
Description:	The user will use the map to locate themselves and look for a vehicle nearest to them.		
Preconditions:	1. User has an account 2. User is logged in 3. User has their location on		
Postconditions:	User will be able to locate the nearest vehicle to them.		
Normal Flow:	**6.0 Use a map to locate a vehicle** 1. User logs in 2. Map is on main screen 3. Alerts user if location is not turned on 4. User taps the map to locate where they are 5. Zoom in to find the nearest vehicle in the area		
Alternative Flows:	None		
Exceptions:	**6.0.E.1 Unable to locate their current location** 1. User has difficulty locating their current location 2. Make sure the internet is working 3. Quit the application and open it up again **6.0.E.2 Users location is not on** 1. User does not have the location on in their phone settings 2. User turns on the location on their phone		

Priority:	High
Frequency of Use:	Every time a user is looking to rent/book a car, they need to use the maps in order to see which location nearest to them have a car available for them to rent/book
Special Requirements:	1. Location must be on in the user's phone in order for the maps to work
Assumptions:	1. There will be no problem locating the cars available on the maps
Notes and Issues:	1. Issue: as of right now, we still do not know how to let the users see (from the maps) if there are cars available at the location

UC-C-7: Choose the type of vehicle

Use Case ID:	**UC-C-7**		
Use Case Name:	Choose the type of vehicle		
Created By:	gprabowo	Last updated By:	arifdian
Date Created:	November 6, 2019	Date Last Updated:	November 22, 2019
Actors:	User		
Description:	After locating the nearest location, users are able to choose the type of vehicles that they are looking to book		
Preconditions:	1. User has an account 2. User is logged in 3. User has their location on		
Postconditions:	User will be able to locate the nearest vehicle to them.		

Normal Flow:	**7.0 Use a map to locate a vehicle** 1. User logs in 2. Map is on main screen 3. User taps the map to locate where they are 4. Zoom in to find the nearest vehicle in the area 5. User has the option to choose the different vehicle types and sizes with respect to availability. 6. User tab on the image of the desired vehicle type and size
Alternative Flows:	None
Exceptions:	**7.0.E.1 Unable to locate their current location** 1. User has difficulty locating their current location 2. Make sure the internet is working 3. Quit the application and open it up again **7.0.E.2 Users location is not on** 1. User does not have the location on in their phone settings 2. Alerts user if location is not turned on 3. User turns on the location on their phone
Priority:	High
Frequency of Use:	Every time a user is looking to rent/book a car, they need to use the maps in order to see which location nearest to them have a car available for them to rent/book
Special Requirements:	1. Location must be on in the user's phone in order for the maps to work
Assumptions:	1. There will be no problem in choosing the vehicle.
Notes and Issues:	1. Issue: as of right now, we still do not know what type of vehicles are available

UC-C-8: Give a time frame for pick-up and drop-off

Use Case ID:	**UC-C-8**		
Use Case Name:	Give a time frame for pick-up and drop-off		
Created By:	gprabowo	Last updated By:	arifdian
Date Created:	November 6, 2019	Date Last Updated:	November 22, 2019
Actors:	User		
Description:	After choosing the vehicle, users are able to pick up the car during the time that is available		
Preconditions:	1. users logged in 2. users picked the location for where they are going to pick up the car		
Postconditions:	User will be able to set pick-up and drop-off time		
Normal Flow:	**8.0 Set the time to pick up the vehicle** 1. users log in 2. go to maps and pick the location 3. set the time to pick-up and drop-off the car 4. the time is fixed for 30 minutes interval for example, 10:30 - 11:30. hence a minimum of 30 minutes rent.		
Alternative Flows:	None		

Exceptions:	**8.0.E.1 Unable to set the pick up time** 1. users are not able to set the pick up time 2. one of the reasons could be someone reserve the car ahead of time (conflict) 3. pick another car that is available in the nearest area. 4. user's pick up time and drop off time adds up to more than 7 days.
Priority:	Low
Frequency of Use:	Every time a user is looking to rent/book a car, they need to pick the car that they are planning to book/reserve.
Special Requirements:	1. make sure to book/reserve the car quickly so someone won't reserve the car first and drop-off
Assumptions:	1. There will be no problem setting the pick-up and drop-off time
Notes and Issues:	1. Issue: user might not return the car on time when the other user want to pick up the car

UC-C-9: Reserve a vehicle

Use Case ID:	**UC-C-9**		
Use Case Name:	Reserve a vehicle		
Created By:	gprabowo	Last updated By:	arifdian
Date Created:	November 6, 2019	Date Last Updated:	November 22, 2019
Actors:	User		

Description:	After setting up the pick up time, users are able to reserve the vehicle for them to pick up
Preconditions:	1. users logged in 2. users picked the location for where they are going to pick-up and drop-off the car 3. users reserve the car
Postconditions:	User will be able to reserve the vehicle that they have chosen
Normal Flow:	**9.0 Set the time to pick up the vehicle** 1. users log in 2. go to maps and pick the location 3. set the time to pick-up and drop-off the car 4. user confirm the total price 5. user reserve the car
Alternative Flows:	None
Exceptions:	**9.0.E.1 Unable to reserve the vehicle** 1. users are not able to reserve the vehicle 2. another users may have reserved it a few minutes quicker 3. users will need to choose another vehicle or in a different location 4. User's bank transaction/payment gets declined 5. Driver's license expired
Priority:	High
Frequency of Use:	Every time a user is looking to rent/book a car, they are required to reserve the car ahead of time.
Special Requirements:	1. make sure that the vehicle that you reserve is the vehicle that you originally wanted
Assumptions:	1. There will be no problem reserving the vehicle.

Notes and Issues:	1. Issue: user might not be able to reserve the car as someone else reserved it ahead of time.

UC-C-10: Summary of reservation

Use Case ID:	**UC-C-10**		
Use Case Name:	Summary of reservation		
Created By:	gprabowo	Last Updated By:	arifdian
Date Created:	November 6, 2019	Date Last Updated:	November 22, 2019
Actors:	Admin		
Description:	After reserving the car, users are given the summary of reservation.		
Preconditions:	1. users logged in 2. users picked the location for where they are going to pick-up and drop-off the car 3. users reserve the car 4. reservation summary is given		
Postconditions:	User will be given the reservation summary of the vehicle that they reserved		
Normal Flow:	10.0 Summary of reservation 1. users log in 2. go to maps and pick the location 3. set the time to pick-up and drop-off the car 4. user reserve the vehicle 5. summary of reservation is given to the users		
Alternative Flows:	None		
Exceptions:	None		

Priority:	Low
Frequency of Use:	Every time a user reserve a vehicle, they will be given the reservation summary
Special Requirements:	1. make sure that the summary indicated the correct information for your reservation
Assumptions:	1. There will be no problem with the summary.
Notes and Issues:	1. Issue: the reservation summary might be different from what the user originally reserved

UC-C-11: Edit / Cancel a reservation

Use Case ID:	11		
Use Case Name:	Modify reservation		
Created By:	tianhuan	Last Updated By:	arifdian
Date Created:	November 9, 2019	Date Last Updated:	November 22, 2019
Actors:	User		
Description:	Users make choices about their reservation. Either changing contents in the reservation, or cancelling a reservation of vehicle.		
Preconditions:	1. User has an account 2. User is logged in 3. User has a valid driver's licence 4. User has placed a reservation		
Postconditions:	Reservation will update once user confirmed to change.		

Normal Flow:	**11.0 Modify reservation** 1. User already has a reservation 2. User click on button “modify reservation” 3. User reenters the information in reservation 4. User confirms changes applying to the reservation
Alternative Flows:	None
Exceptions:	**11.0.E.1 User reservation failed** 1. User modified reservation 2. The current type of vehicle/time is not available 3. System replies to users “Change failed” **11.0.E.2 Users input wrong information** 1. User modified reservation again 2. User confirmed the new request for reservation
Priority:	High
Frequency of Use:	Every time when user is changing location/ car model/ reservation time
Special Requirements:	1. User must input all information which is in the required areas from the form
Assumptions:	1. Every time when users are trying to make changes to their reservation, they are going to modify their reservation.
Notes and Issues:	1. We can accept the same information while modifying the reservation.

UC-C-12: Report car's condition (upload photos of the car)

Use Case ID:	12		
Use Case Name:	Take photos of the car before and after operating the vehicle		
Created By:	tianhuan	Last Updated By:	arifdian
Date Created:	November 9, 2019	Date Last Updated:	November 22, 2019
Actors:	User		
Description:	Users take photos of the car in front, left, right and back before unlocking and after locking the car to record the car's condition		
Preconditions:	1. User logged in account 2. User made a reservation 3. User arrived at the car pool 4. User clicks "unlock car button" in the application		
Postconditions:	1. Users will then be required to take photos of current car condition 2. Users operate the car 3. User lock the car after operation 4. Users take photos again 5. User successfully finished the trip		
Normal Flow:	**12.0 Take photos of the car before and after operating the vehicle** 1. Users logged into their account 2. Users make a reservation 3. Users arrived at the car pool 4. Users unlock the car 5. Users take photos of the current car's condition 6. Users start their trip 7. Users lock the car		

	8. Users take photos of the current car's condition 9. Users end their trip
Alternative Flows:	None
Exceptions:	**11.0.E.1 User upload photo failed** 1. User takes photos of the car's current condition 2. User upload photo failed, and the system returns "failed to upload photo" 3. User takes photos of the car's current condition again **11.0.E.2 Users take photos of other objects instead of the car they are using** 1. User takes photos of other objects 2. System returns "Wrong car/value in the photo" 3. User takes photos of the car's current condition again
Priority:	High
Frequency of Use:	Every time when users are unlocking / locking the car before and after their trip
Special Requirements:	None
Assumptions:	1. Users phone has a camera
Notes and Issues:	None

4. External Interface Requirements

3.3 User Interfaces

UI-1: All fields should have a maximum of 100 words. A keyboard will appear when the user taps the text box. Each text field will have a placeholder with what the user is meant to put in the field. Feedback text will be under the field if the user does not follow a rule when filling out the specific field. Also, if something is entered wrong the field will be outlined in red.

UI-2: If the user has to enter a credit card number or phone number then the boxes will be seperated. If it is a card number it is will in 4 boxes and if it is a phone number it will be in 3 boxes. This is so the user can keep track of what they need to enter and it will allow them to follow the information.

UI-3: All of the main pages need to have a logo on them.

UI-4: Each page needs to have a navigation that takes the user back home, see their reservation, and settings. If they tap the menu the other stuff in the background will get darker so that it focuses on the menu item. When the user taps on the item the first time the background color of that option will change to a different color and the second time the user taps it will take them to that page.

UI-5: If there is a major error, there will be a popup alert.

UI-6: If the person gets their password wrong a few times or they need to reset their then an email will be sent. The user will have to open the link from there.

UI-7 When a date is entered it will automatically place the "/" between the different texts.

3.4 Software Interfaces

External links will exist from CarMa to the following sites:

4.2.1 PayPal
4.2.3 Google Maps
4.2.3 mail

3.5 Communications Interfaces

There are no specific communications interfaces other than allocated phone line.

4. Other Nonfunctional Requirements

4.1 Performance Requirements

PE-1: Each page should not take more than 3 seconds to load.

PE-2: 1 second is the best time that it takes to load.

PE-3: It should take no longer than 3 minutes to fill out any information in the app such as register, reservations, or cancel reasons.

PE-4: There will be feedback if the user is waiting longer for any of these that provides information of how long it will take.

4.2 Security

SE-1: The website shall use standard Web security protocols when transferring any private information regarding a Customer.

SE-2: The system will track

SE-3: Sensitive data is not released to third parties.

SE-4: Sensitive data will be encrypted inside a secured database.

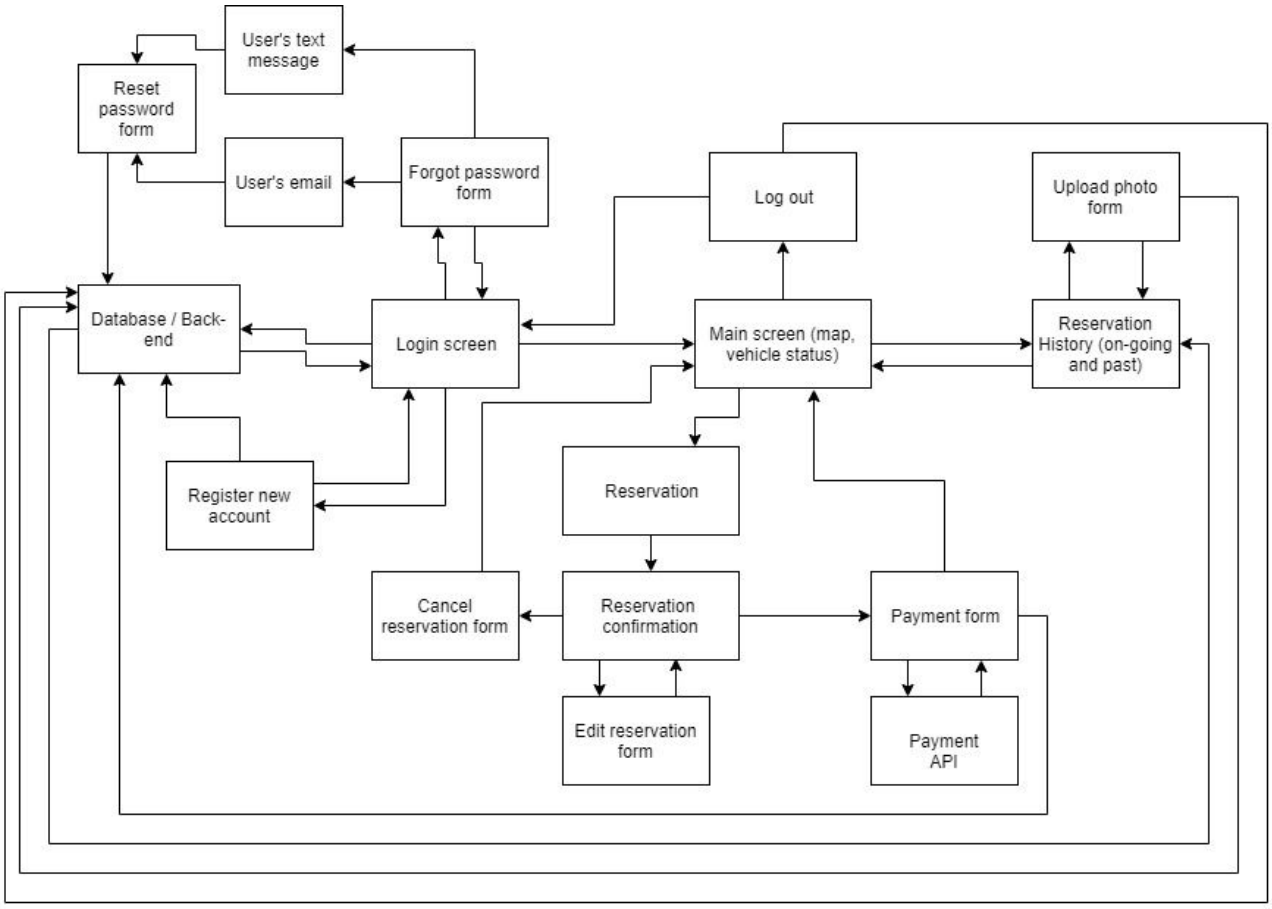
SE-5: Passwords will not be shown when being typed.

SE-6: Each user will be uniquely identified.

4.3 Extensibility

EX-1: The application shall be designed to permit adding features and database component for support.

Appendix A: Analysis Models



Appendix B: Issues List

1. Unable to verify license due to security compromise that must be accepted without Indiana's bureau of motor vehicle.
2. An offline bank database would result in payment decline and not being able to reserve a vehicle from the user's perspective.

Revision History

Name	Date	Reason For Changes	Version
Team 05	11/24/19	1. initial baselined version	1.0