Beta Transportation Network Company

Report

Semester 2 2018 - Requirements Engineering Assignment

Prepared by

Lauchlan Derrick

Dakota Epton

Simeon Leatherland

Table of Contents

1 Introduction

As a company, Beta Transportation Network Company, we have decided to develop a Software Requirements Specification for our new application, Beta Transport, which will allow users to order car rides, order food delivered to their house or rent a bicycle in selected areas on demand. It will also allow drivers to earn an income by delivering food, taking users in their car for rides or to maintain the bicycles around selected areas. This report outlines the requirements elicitation and analysation through various models and processes and various goals, problems and resolutions we have encountered and developed along the way to form the Software Requirements Specification.

2 Project Scope

The goal of the project as derived by our company is as follows:

> *Develop one application that offers ridesharing, food delivery and bike drop off/pick up services to customers, job opportunities for drivers and restaurant owners.*

2.1 Initial requested features

When the company started, we developed a list of features that were required to make this application successful and rival market leading applications:

- Allow customers to order a bicycle, food to a specified location.
- Allow customers to order a ridesharing service from a specified location to another location.
- Allow drivers to earn money for dropping off bikes and food to locations.
- Allow drivers to earn money for picking up and dropping off customers.
- Allow restaurants to place an orderable menu on the application such that customers can order from their location.
- Develop one application that allows a customer, driver or restaurant owner to switch between their respective accounts within the application.

4 Requirements Elicitation

Before developing this product, we wanted to be able to fully understand the requirements involved with every aspect and interaction of this system. We identified and elicited information from key stakeholders using individual interviews, and a key stakeholder meeting to determine requirements and constraints on this application. We built upon this with a questionnaire to gather requirements and uses from majority of external stakeholders. Further information regarding specific elicitation can be seen in Appendix 9.2.

5 Requirements Analysis

Post elicitation, out team concluded a list of functional and non-functional requirements, Appendix 9.3, that need to be met, which have been modeling using use case and sequence diagrams. See modelling in Appendix 9.4 for all our functional and non function requirements.

6 Software Requirements Specification

After the analysis of the elicited requirements, a formal SRS was formed to aid the development of the application. The SRS can be found at Appendix 8.7.

7 Wireframe Prototypes

After the creation of our SRS, wireframe diagrams were created for the main interfaces of the application. This was done in the aim to outline very basic prototypes of the requirements of the SRS. These an be found in Appendix 9.9.

8 Problems Encountered

Over the development process of this project, our company BTNC ran into a few problems which affected elicitation, requirements gathering and questions regarding the scope defined above. A full analysis and how we broke down each problem is detailed further in Appendix 9.1.

9 Appendices

9.1 Appendix - Problems Encountered and Analysis

9.1.1 Application Use Problems

Q: How are we going to ensure food won’t be touched by drivers?

R: To ensure food health and safety we will ensure restaurants stable and sticker the top of the food bad we provide. This will show any tampering to the customer so they can trust that the food would not have been touched.

Q: How are we going to know when someone is picked up by a car?

R: Have clear buttons for the user and driver application so they can both set that they have been picked up, which will ensure the driver is being paid the full ride amount.

Q: How do we ensure the safety of passengers?

R: Make sure that drivers have had background checks and have valid insurance details such that if anything goes wrong the company has done all it can do in its powers to ensure the safety of the passengers as much as we can.

Q: How do we incentivise drivers to stay online?

R: Pay surge multipliers for trips in certain areas at certain times where lots of people are using the service.

9.1.2 Elicitation Problems

Q: Potential bias of Questionnaire towards those people who are likely to use the service.

R: We ran the questionnaire in multiple locations at different times of days to ensure the least bias as possible. We did however place these questionnaires in areas where the service will most likely be used to gain an idea of specific requirements the majority of service users will need.

Q: Potential Gold plating on the development side of the project.

R: Ensure an agenda has been specified before the meeting with the specific requirements that will be discussed, so that a target goal of the specific discussions can be taken place without exploring different options. This will in turn allow stakeholders to have an idea of what to discuss with each topic rather than have a broad agendas which will allow gold plating to occur.

Q: Differing opinions in stakeholder meetings.

R: Make sure that an agenda is specified and sent to all stakeholders beforehand so everyone has a clear idea of what to discuss. This doesn't specifically target differing opinions, to target differing opinions we will have a meeting leader which will try and be a bridge in the middle to impartially negotiate.

9.1.3 Management Problems

Q: Cash problems.

R: Within a short time frame, limited cash will limit the scope and growth of the application, so we need to generate a burn rate of our application before we make profit.

Q: Lead developers want the best developers for their team.

R: To get the best talent we will need to spend lots of money. We will try and incentivise our work place to have a good work life balance to ease the acquisition cost and rather advertise the benefits and the work life balance. This will in turn hopefully attract the best talent.

Q: Not all android versions will support the application.

R: Hard to develop for a large version history, so we need to find a balance between users and limiting users.

9.2 Appendix - Requirements Elicitation

9.2.1 Domain Properties

The domain properties are listed below:

- Drivers have a driver's licence.
- Restaurants have a valid food, health and safety licence.
- Majority of people have a iPhone or android with WIFI, Bluetooth or cellular connection.
- Majority of people have valid payment methods, credit card, Paypal.

9.2.2 Key Stakeholders

A board meeting with the company was arranged to identify the stakeholders of the ride-sharing / food delivery industry. A focus was placed on how these stakeholders would be affected by the introduction of the new system, and the consideration of how this could potentially mitigated/managed if necessary. The following stakeholders were identified:

- **Existing 'Instant Response' Companies** – Existing companies will become competitors with the new system. They may try to incorporate features new to the proposed system, or at the least preserve their market share in some other way.

- **Administration / Management staff at the BTNC** – This includes both front and back end staff of the new company (excluding software developers). Depending on the sourcing of staff, they may require either complete training, or at least retraining to assist them in the transition from their previous position.

- **Development staff at the BTNC** – Development staff will be impacted as they will now likely be working alongside a greater number of staff, all with their own specialised position within the company.

- **Drivers** – Driver responsibilities and interactions with customers will be impacted. They will have to be confident with both food delivery and the driving of passengers, and also the stowing and removing of bikes on their vehicles. The lifting of bikes may bring with it some physical limitations that may also impact drivers.

- **Public transport users** – Users will be impacted as their needs for various services will be centralised. Habits will need to be adjusted to accommodate for this centralisation. They will also have to adjust to being comfortable with food potentially being in the car, and potential stops for bike pickups / drop-offs.

A further meeting was conducted, and a restaurant owner that presently made use of Deliveroo was brought in to assist in the identification of any further stakeholders that may have been previously missed for the dining part of the system. Identified in this meeting were:

- **Restaurant owners –** Restaurant owners will be impacted with a further choice of service to use (the BTNC vs Deliveroo, etc). This may lead to them demanding a more competitive proposition from such companies, in return for them selecting one over the other.
- **'At home' diners** – Diners at home will likely be impacted as their choice of services will diversify. For example, they may now need to keep accounts with multiple services, if one service has exclusive access to a food provider over another. However, if they choose to use the BTNC service, their dining / transportation needs will be centralised, and they will be impacted by needing to become accustomed to this change.
- **'In Restaurant' diners** – As (hopefully) the BTNC service reduces people's need to go out to eat as often, diners who still prefer to attend restaurants will be impacted by the increased presence of delivery drivers in the restaurants themselves. They will be impacted as they will need to become accustomed to this increased presence. To keep business' happy, the BTNC may need to ensure its physical presence is minimised in some restaurants to ensure the satisfaction of these diners and the restaurant owners.

Finally, a conclusionary meeting was held to consider the scope of the project. The following stakeholders were identified as being involved, but outside the scope of the project:

- **Government Ministers / Staff** – Government staff are (currently) outside the scope of this project. As it stands, all the services BTNC will be based on are legal, and do not interact with the government to any considerable degree. BTNC will be the same, and as a private project (with no government endorsement or funding), will similarly not interact with the government to any great degree.

9.2.3 Information Gathering Techniques

After identifying the stakeholders, we used the information gathering techniques below to gather the specific requirements for the project.

9.2.3.1 Interviews

Interviews were conducted on an individual basis with key stakeholders to identify potential requirements by the Owner and Developer of BTNC. We decided to conduct interviews as we found they would be a good source of information, with little pressure on the people to discuss or gold plate their opinion, as they may be prone to do when in front of others. We found these interviews effective in uncovering goals, and suggesting follow up questions that could be discussed at a later time in a group setting with all interviewees.

Interviews were conducted with the following stakeholders:

BTNC Management (Jump Bikes Manager)

One of the new management employees. They were hired based on their experience in this style of application and their notable managerial skills in software development projects.

The following questions were used to establish if BTNC were on the right track and to develop some possible requirements specifications:

- Why do we want to create this application?
- What will make us stand out against other applications?
- What new features should the system do?
- Are there any problems with current applications?
- How long did you work at your last company? What did they do well?

The following questions were used to establish the features specific to operations of the application and past applications:

- What are the peak hours of ride sharing companies?
- How should we work around peak traffic and high demand?
- How should we ensure safety of food and health of meals?
- How are we going to validate drivers and restaurants?

BTNC Head Developer (Ex- Uber Developer)

As developers were identified as one of the key stakeholders in the project, it was important that we discovered any needs such individuals may have, as well as any requirements that should be discussed formally.

The following questions were used to discuss BTNC and the direction of the company with any requirements that would aid development:

- Why do we want to create this application?
- What will make us stand out against other applications?
- What new features should the system do?
- Are there any problems with current applications?
- How long did you work at your last company? What did they do well?

The following questions were used to discussed specific to the infrastructure and development of the application:

- What infrastructure do other companies use?
- What and how big does the new hardware cost?
- How will we maintain the hardware?
- How will be back up and ensure customer data?
- What features could we add to the application itself?

Driver (UberX Driver)

Arguably the biggest stakeholder in terms of use of the application is those drivers who will be using the application. Alongside questions similar to those previously mentioned, we asked questions designed to expand on how the application could and should be used, and identify further features that should also be implemented.

The following questions were used to discuss BTNC and the direction of the company with any requirements that would aid development:

- Why do we want to create this application?
- What will make us stand out against other applications?
- What new features should the system do?
- Are there any problems with current applications?
- How long did you work at your last company? What did they do well?

The following questions were used to discussed specific to the UI and specific features?

- What phone do you mainly use?
- How do you use the phone in the car?
- What features could be implemented to make driving or use better?

Interview Conclusion

Upon concluding the interviewing process, we established the following requirements and constraints that should be brought up at company meetings to be further discussed.

- System will be ran on Android and OS
- Backend infrastructure will be ran and maintained in-house.
- Application will allow the following accounts:
    - Driver
    - Rider/Customer
    - Restaurant
- The main payment methods will be online, and feature no cash transactions. Potential payment methods include credit card, paypal and apple pay.
- System should have data security and backups.
- Drivers using the system should have their driver licence, riders should be 13+ and restaurants should have appropriate food and safety licences.
- The application will be available to use 24/7.
- Drivers should have a driver dashboard that gives easy access to their core features on one screen, ease of use whilst in a car.

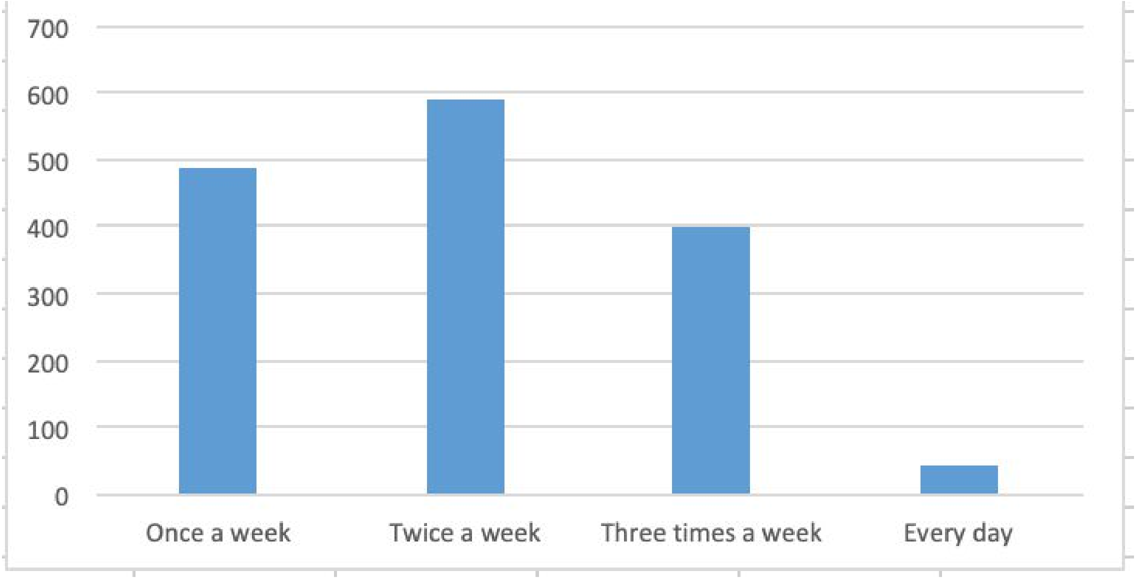
9.2.3.2 Questionnaires

A questionnaire was developed and handed out at restaurants, at a bike festival and relevant conventions, and to people around the Northbridge area at popular nightlife locations. The following questions were asked, and their respective results below are recorded below.

Format : Question? Answer (number of respondents)

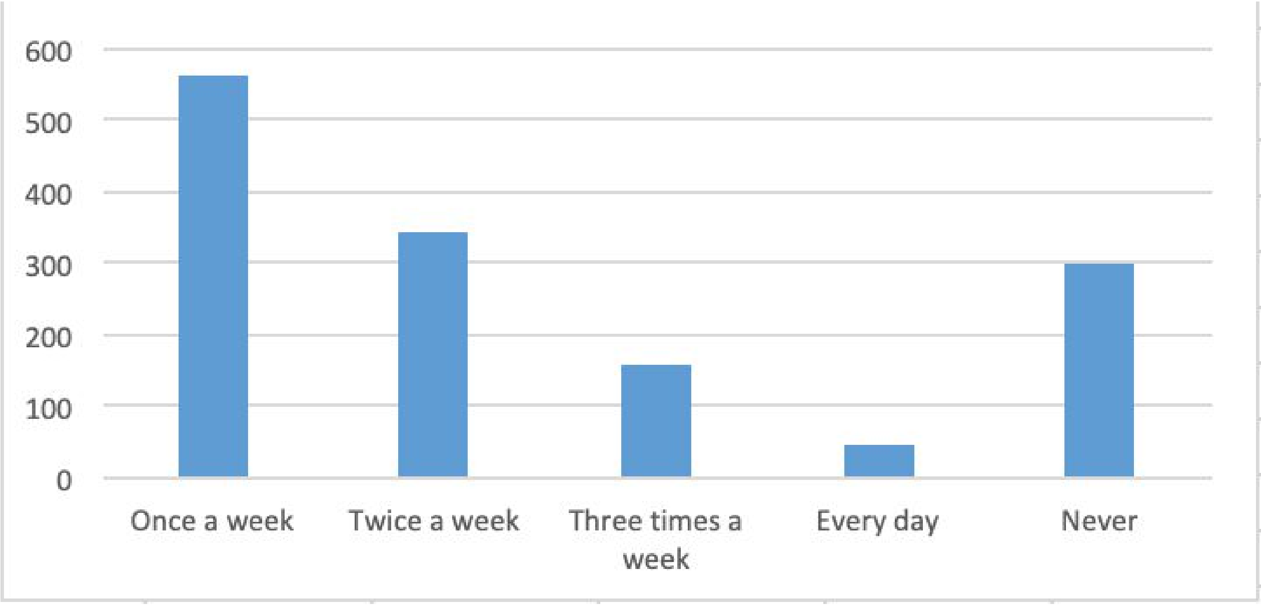
How often do you eat out?

- Once a week (489)
- Twice a week (589)
- Three times a week (400)
- Every day (45)



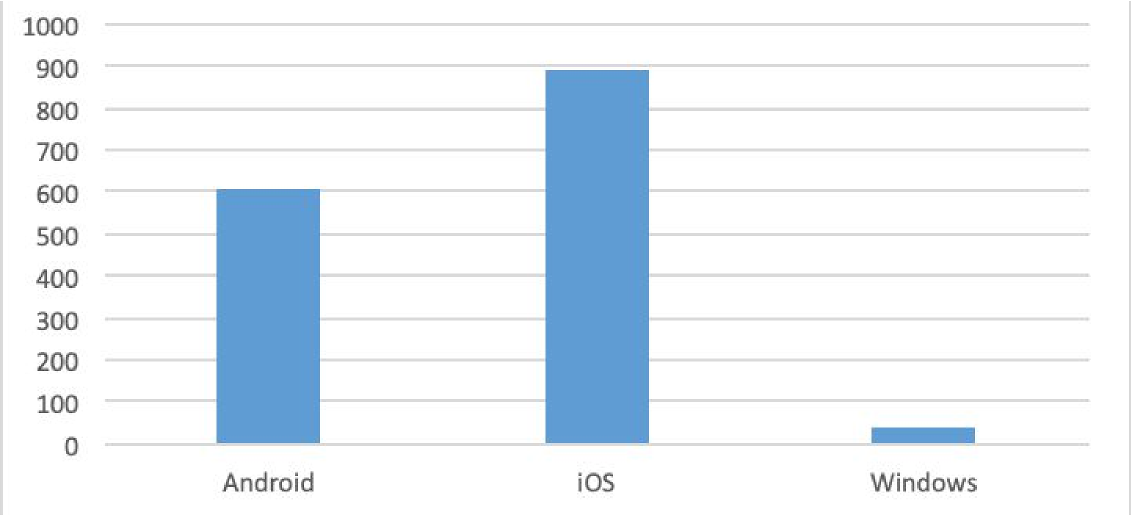
How often do you get take out food to your house?

- Never (302)
- Once a week (564)
- Twice a week (342)
- Three times a week (158)
- Every day (45)



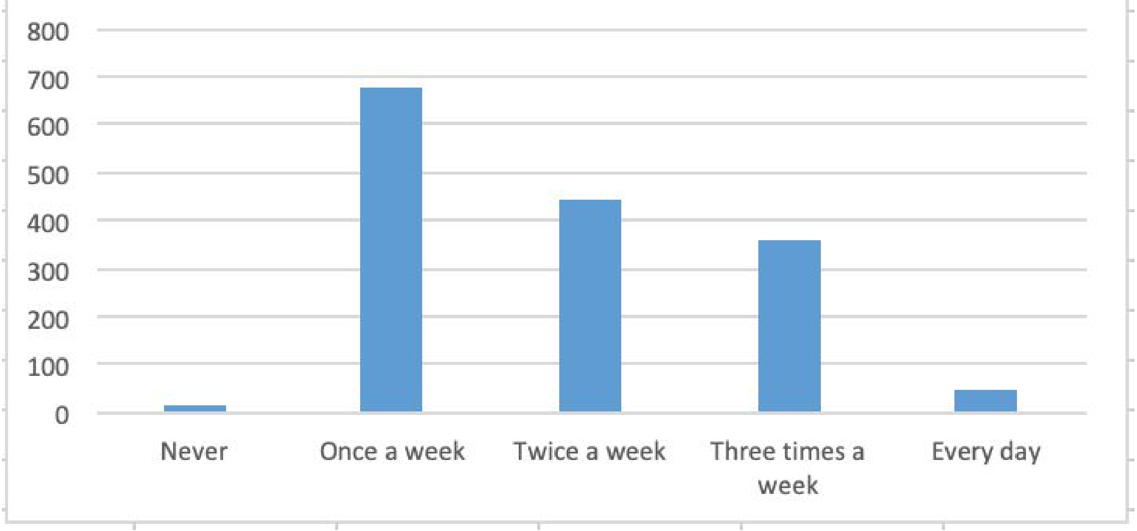
What mobile phone do you currently use?

- Android (603)
- iOS (887)
- Windows (42)



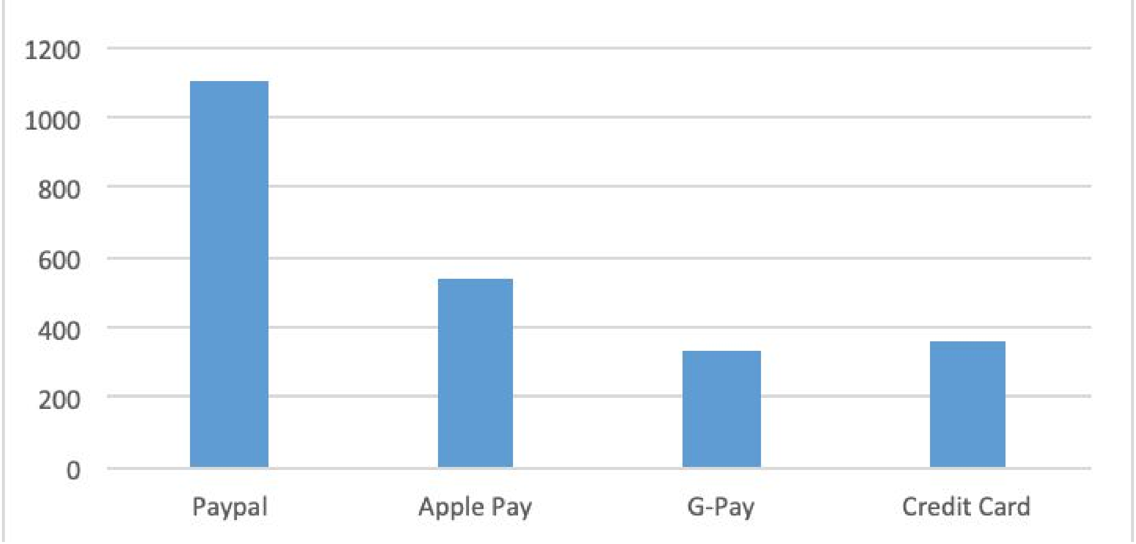
How often do you catch a taxi or ride sharing service?

- Never (12)
- Once a week (678)
- Twice a week (447)
- Three times a week (360)
- Every day (45)



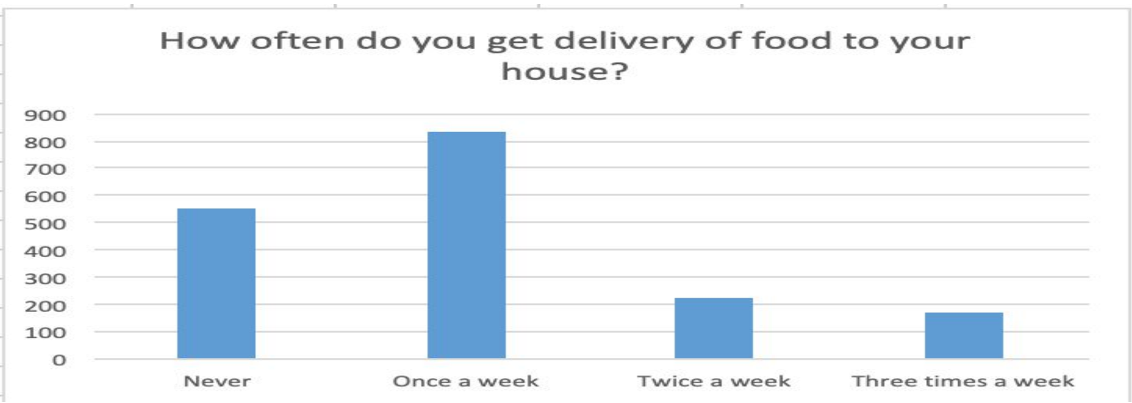
Do you currently have Paypal, Apple pay or GPay?

- Paypal (1100)
- Apple Pay (543)
- G-Pay (335)
- Credit card (1489)



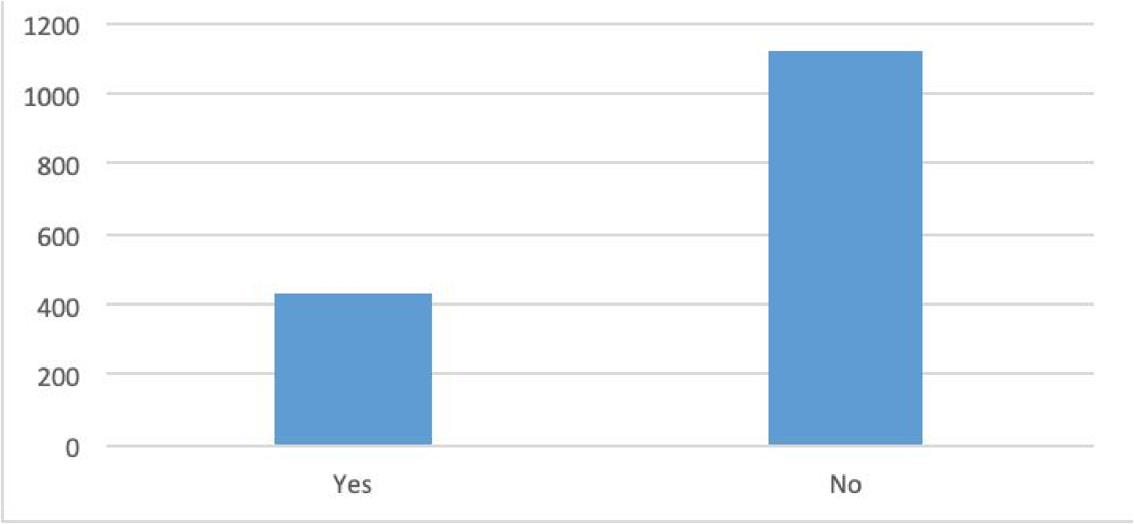
How often do you ride a bicycle?

- Never (554)
- Once a week (831)
- Twice a week (224)
- Three times a week (170)
- Every day (3)



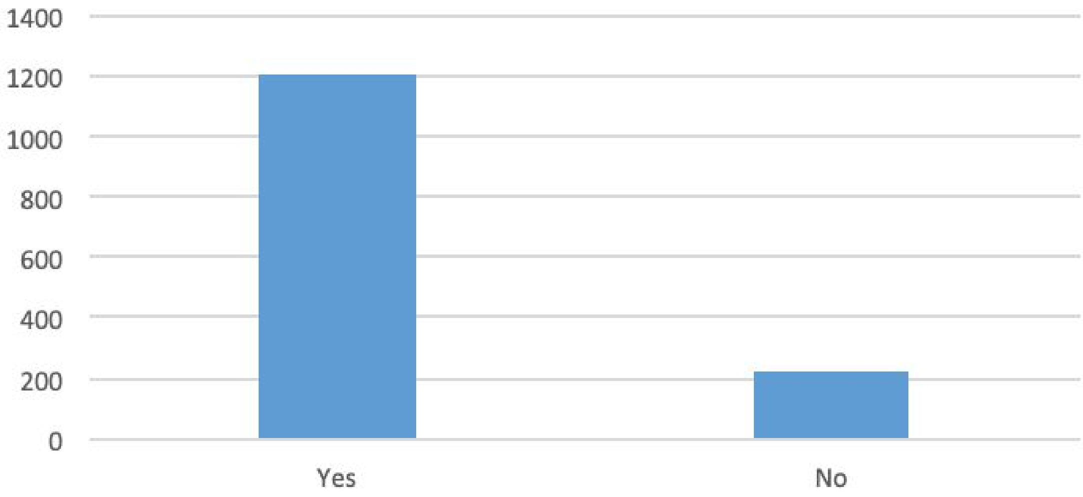
Have you ever been been in a situation where you needed a bike dropped off to you?

- Yes (430)
- No (1123)



Do you trust food delivery services?

- Yes (1203)
- No (220)



9.2.3.2.1 Questionnaire Conclusion

After conducting the questionnaire, we concluded that majority of people eat out more than twice a week, with over 60% of people who eat out ordering food delivery services to their houses, with a 87% trusting rate of the food delivery features. We concluded that only 3% of people do not catch a taxi or ride sharing service each week, which means 97% people use a service like Uber or taxi. In terms of payment methods, 76% of people have paypal and 97% having a credit card, so use of these payment methods would be ideal. In terms of the bicycle services, over 64% ride somewhere more than once a week, with more than 30% of people concluding they would use a drop off bike service. In regards to the phone OS user use, over 66% use iOS with 42% using android and 2% using windows. We surveyed 1523 people for each question on average.

9.2.3.3 Meetings

Meetings were conducted with the three main stakeholders that would be interacting with the application's design, progression and use. This was done with the aim of developing a summarization of all the requirements that were discussed in the individual interviews, and gathering feedback on what the group thought were important requirements. Before the meeting, an agenda was sent out to all the stakeholders so that they could prepare for what they wished to discuss. The following meeting minutes were written and as a result, the project was able to proceed with a sense of agreement upon the requirements discussed, and a cohesive focus.

9.2.3.3.1 Meeting 1

Beta Transport Network Company Meeting Minutes (Role-play Meeting One)

Date: Thursday 11th October 2018
Time: 3:00pm - 4:00pm
Location: Building 314, Curtin University
Meeting Type: Requirements Elicitation
Attendees: Lachlan Derrick (Owner), Simeon Leatherland (Developer), Dakota Epton (Ex-Uber Developer), Amelia Smith (Uber/Uber Eats Driver), Ching Chong (Restaurant owner)

Agenda Items:

- Initial contact between domain experts and key stakeholders.
- Discussion of requirements.
- Initial drafting of requirements.

Requirements Proposed

Customer Related

- Customer can request to:
    - Get a ride to a location from a registered driver
        - Can select regular or XL car
    - Get a Bike drop off to a location from a registered driver
    - Pick up a bike from a bike rack
    - Get food delivered to a location from a registered restaurant delivered by a registered driver
- Customer is required to provide:
    - Name
    - Mobile Number
    - Email Address
    - Payment Method (Credit Card or PayPal)

    - Billing Address
- Customer is able to:
    - Get a quote on their trip/delivery/bike cost before confirmation
    - Change personal details
    - Change payment details

Driver Related

- Driver can request to:
    - Pick up a customer and drive them to a given location
    - Pick up a bike and deliver it to a customer
    - Pick up a delivery from a restaurant and deliver it to a customer location
- Driver is required to provide:
    - Name
    - Mobile Number
    - Email Address
    - Bank Account Details
    - Billing Address
    - Police Clearance
    - Valid Driver's License (And have held it for at least 4 years)
- Driver is able to:
    - Receive payments for their trips

Restaurant Related

- Restaurant is able to:
    - Receive food order requests from customers
    - Receive payments for their food orders
    - Give feedback on their delivery driver
    - Add or remove menu items to their account for purchase by customers
- Restaurant is required to provide:
    - Restaurant Name
    - A Primary Contact:
        - Name
        - Contact Number
        - Email Address
    - Business Number
    - Valid Health And Safety Certification
    - Valid Food Outlet License
    - Bank Account

9.2.3.3.2 Meeting 2

Beta Transport Network Company Meeting Minutes (Role-play Meeting Two)

Date: Saturday 13th October 2018
Time: 3:00pm-4:00pm
Location: Building 314, Curtin University
Meeting Type: Finalising Requirements
Attendees: Lachlan Derrick (Owner), Simeon Leatherland (Developer), Dakota Epton (Ex-Uber Developer), Amelia Smith (Uber/Uber Eats Driver),Ching Chong (Restaurant owner)

Agenda Items:

- Finalise previous meetings requirements.
- Propose new requirements.

<u>Finalised Requirements</u>

- All requirements from the previous meeting were agreed upon.

<u>Finalised New Requirements</u>

Customer Related

- **Customer is able to:**
  - Only order food from a restaurant that’s within a 10km radius
  - Give feedback on their experience
    - Submit complaint
    - Rate out of 5
  - Enter a promotional code to receive a discount or promotion
  - Contact their driver before and after their ride

Driver Related

- **Driver is able to:**
  - Only take customer requests from customers within a 10km radius
  - Give feedback on their customers
    - Submit complaint
    - Rate out of 5
  - Contact their customer before or after their ride
  - Contact their pick up restaurant before or after delivery

Restaurant Related

- **Restaurant is able to:**

- Contact their driver before or after order has been picked up
- Contact their customer before order has been dispatched

9.2.4 Elicitation Discussion/Conclusion

After creating the company, BTNC held a meeting to discuss how to move forward with the gathering of requirements and observing of stakeholders. The team concluded upon a list of stakeholders, and organised interviews with each of the main stakeholders in order to gain personal insight on their perspective and the requirements that they would like to see.. Once the interviews concluded, we organised meetings with all the stakeholders who were interviewed to bring together their requirements and further discuss what was and was not needed moving forward. This was done to separate and allow people who may not have felt confident to discuss their own requirements in front of other people the chance to do so. Interviews and meeting minutes can be found at 7.2.3.

After constructing a list of satisfactory key stakeholder constraints and requirements, questionnaires were given to key users to get access to many people's opinions within a short time frame. We formulated questions with the aim of limiting and constraining certain requirements gathered during the meetings and interviews. This was done to formulate a larger picture of how best to create the application with a good balance between user and stakeholder needs. All data collected can be found under questionnaire section above in 7.2.3.2.

9.3 Appendix - Requirement Analysis

9.3.1 Non-Functional Requirements

1. The application must be developed using version control so that code is maintainable, and so a timeline of the code history can be seen and reverted to when necessary.
2. Any application faults with code must be maintained and fixed with a software update within 24 hours of it occurring / being reported to the company.
3. Database must undergo a continuous incremental backup with a daily full backup to ensure there will be no loss of business or user data should an incident occur.
4. Database and its full backup must be stored in different states to limit loss data if a natural disaster was to occur.
5. Incremental backups must be stored in different buildings, such that in the case of a fire or similar emergency, data loss will be minimised.
6. Database must be secured by lock and key and must be under 24/7 surveillance to deter and prevent theft or intentional damage.
7. User account passwords must require at least one capital letter, be longer than 9 characters and contain 2 numbers. This ensures passwords are hard to guess.
8. Data sent from device to server/database must be encrypted end to end using the DES standard to ensure that middle man attacks and spying on packets cannot occur.
9. Payment and credit card details must be encrypted and stored with user's personal details, however the 3/4 digit CVN number must never be stored or sent to the database. Must be entered upon each use of the card.
10. The application must require the user to login once every month on their mobile phone.
11. User passwords must be changed once a year.
12. Security team must have employees hired to monitor attacks on the database 24/7 to detect and recover from any hacks or unauthorized access into the system.
13. The application must display a walk through visual helper on the first login or upon creation of a new account in the application. This feature must also be accessible from the settings menu inside the application.
14. The website must be easy to use, with 90% of users being able to book a service after watching a walk through within 2 minutes of opening the application.
15. The application must work both on iOS and android devices.

9.3.2 Functional Requirements

1. Create an account, specifying which accounts they wish to create (driver, rider or restaurant), along with their details.
2. Allow user to login into the app with the option of logging into their restaurant, driver or rider account.
3. View and change account details through the application.
4. Add a credit/debit card to pay for trips with the option to use apple pay or google pay (this must be a hard constraint on the system and the user must not be able to create an account without including card details).
5. Order a ride from a specific location, whether its current or at a specified location.
6. Order food from a chosen restaurant and have it delivered to a current or specified location.
7. Pick up a bike from a bike rack nearby, and upon confirmation of payment, have the bike unlock from the rack.
8. Order a driver to drop off a bike to your current location.
9. View quotes and cost of trips or food delivery before the user confirms their trip / order.
10. Display visually on a map the location of the driver upon their acceptance of a request from a user, along with their ETA.
11. Switch between the 3 different account modes within the app, Driver, Rider and Restaurant.
12. Driver mode allows drivers to turn their status to waiting.
13. Allow driver to view income and past trips with all details and ratings/feedback.
14. When driver is in waiting mode, notify a driver if a customer request a rideshare service, OR a food delivery is available OR bike delivery.
15. Allow a driver to accept or reject a customer request (this means the driver does not want to undertake this request).
16. Allow a driver to drop off a bike to a user's location.
17. Notify the users when a driver is near by their specified location after a request has been made.
18. Charge customers pay method after they have confirmed their request for a service.
19. Allow a restaurant account owner to open or close their store for food delivery orders.
20. Allow restaurant owner to add or remove menu items.
21. Allow restaurant to change and view their public information.
22. Allow restaurant to have access to a drivers information so they can verify pickups.
23. Provide a user interface for restaurants to see how much they have sold.
24. Allow site administrators access to collected data, and provide means for them to maintain servers and user accounts if issues arise.
25. For drivers and restaurants, allow them to modify the bank account details that their earnings will be paid into.

9.4 Appendix - Requirement Modeling

9.4.1 Use Case Diagrams

Title:	Create User Account	
Use Case ID:	1	
Description:	This use case explains the process of an User creating an account to use the BTNC system.	
Primary Actor:	Guest User	
Secondary Actors:	Database, Bank system, application system	
Preconditions	BTNC System is online and User is on App create account page.	
Trigger	User selects ‘Create User Account’	
Flow of Events		**Events:**
	1	Application requests users personal information. ***1**
	2	User inputs required information.
	3	Application validates payment info with bank
	4	Bank confirms valid card details
	5	System requests email confirmation, sends confirmation email to address user provided.
	6	User presses confirmation link.
	7	System create new user in accounts database.
	8	Database tells application successful
	9	Application logs user in
Alternatives		**Events:**
	2A1	User presses cancel
	2A2	App redirects to loading page
	4A1	Bank rejects payment details
	4A2	GOTO EVENT **2.** Display error message
	5A1	Email doesn’t exists
	5A2	GOTO EVENT 2. Displaying error message
Notes:	**1.**	***Personal Information:** Full name, billing address, mobile number, email address, payment information (Credit Card

		Details/PayPal)

Title:	Delete Account	
Use Case ID:	2	
Description:	This use case explains the process of an User deleting their account from the BTNC system.	
Primary Actor:	Registered User	
Secondary Actors:	BTNC Database, App	
Preconditions	BTNC System is online and User is logged in on Apps delete account page.	
Trigger	User selects ‘Delete Account’	
Flow of Events		**Events:**
	1.	System prompts User to confirm whether they are sure they wish to delete their account.
	2.	User selects confirm.
	3.	System checks if user account has an outstanding balance.
	4.	User’s account information is removed from database.
Alternatives		**Events:**
	3A1	If user has an outstanding balance deny delete request.
	3A2	Output error message to user and display outstanding balance.
Notes:		

Title:	Log In	
Use Case ID:	3	
Description:	This use case explains the process of an User logging into the BTNC system.	
Primary Actor:	Registered User	
Secondary Actors:		
Preconditions	BTNC System is online, User has a valid account and is on either App or Website’s login page.	
Trigger	User selects ‘Log In’	
Flow of Events		**Events:**
	1.	System requests User’s email and password.
	2.	User inputs email and password.
	3.	System validates given login credentials.
	4.	System redirects User to user home page.
Alternatives		**Events:**
Notes:		

Title:	View Account Details	
Use Case ID:	4	
Description:	This use case explains the process of an User viewing their account information* 1	
Primary Actor:	Registered User	
Secondary Actors:	Database	
Preconditions	BTNC System is online and User is logged in.	
Trigger	User selects ‘View Account Details’ OR User selects ‘Update Account Details’	
Flow of Events		**Events:**
	1	Application requests user information from BTNC Database
	2	BTNS database returns user information
	3	Display personal information to screen
Alternatives		**Events:**
	3A1	BTNCDatabase returns no user exist (which shouldn’t really happen if the user is in the application already)
	3A2	App moves to login screen
Notes:		

Title:	Update Account Details	
Use Case ID:	5	
Description:	This use case explains the process of an User updating their account information.	
Primary Actor:	Registered User	
Secondary Actors:		
Preconditions	BTNC System is online and User is logged in on either App or website’s update account details page.	
Trigger	User selects ‘Update Account Details’.	
Flow of Events		**Events:**
	1.	System displays current personal information associated with logged in users account. ***1**
	2.	User selects which detail they wish to update.
	3.	User inputs the updated detail.
	4.	System prompts the user to confirm they wish to update the selected detail.
	5.	User selects confirm.
	6.	User’s account information is updated in accounts database..
Alternatives		**Events:**
	2A1	User selects cancel.
	2A2	System redirects User to user home page..
	5A1	User selects cancel.
	5A2	GOTO Event 1.
Notes:	**1.**	***Personal Information:** Full name, billing address, mobile number, email address.

Title:	View Payment Methods	
Use Case ID:	6	
Description:	This use case explains the process of an User viewing the payment methods currently linked to their account	
Primary Actor:	Registered User	
Secondary Actors:	Database, Bank, Application System	
Preconditions	BTNC System is online and User is logged in.	
Trigger	User selects ‘View Payment Methods’.	
Flow of Events		**Events:**
	1.	System displays current payment methods associated with the logged in account. *1
	2.	User selects a payment method.
	3.	System retrieves payment method information from BTNC Database.
	4.	System displays available details associated with the payment method. *2
	5.	User selects cancel.
	6.	System returns to payment method display page.
	7.	User selects cancel.
	8.	System redirects User to user home page.
Alternatives		**Events:**
	2A1	User selects add payment method.
	2A2	System redirects User to add payment method page (Use Case: 7).
	2B1	User selects cancel.
	2B2	System redirects User to user home page.
	4A1	User selects edit payment method.
	4A2	System redirects User to update payment method scene (Use Case: 8).
Notes:	**1.**	**Payment Method:** Credit/Debit Card Details/PayPal.
	2.	**Available Details:** Last 4 digits of credit card, expiry date, etc.

Title:	Update Payment Method	
Use Case ID:	7	
Description:	This use case explains the process of an User updating a payment method already linked to the logged in account.	
Primary Actor:	Registered User	
Secondary Actors:	Application/System, Database, Bank	
Preconditions	BTNC System is online and User is logged in on either App or website display payment method details page.	
Trigger	User selects ‘Edit Payment Method’	
Flow of Events		**Events:**
	1	Application requests payment information from BTNC Database
	2	Application displays current details of the selected payment methods used. __*1, *2__
	3.	User selects which detail they wish to update.
	4.	System prompts the user to confirm they wish to update the selected detail.
	5.	User inputs payment details
	6	Application confirms with banking method the details are valid
	7	Selected payment method information is updated in accounts database..
	8.	Application Prompts user of the result
Alternatives		**Events:**
	3A1	User selects cancel.
	3A2	System redirects User to payment method details display page (Use Case: 6, Event 3).
	4A1	User selects cancel.
	4A2	GOTO Event 1.
Notes:	**1**	**Payment Method:** Credit/Debit Card Details/PayPal.
	2	**Required Details:** Card number, expiry date, country, paypal username, etc.

Title:	Add Payment Method	
Use Case ID:	8	
Description:	This use case explains the process of an User adding an additional payment method. ***1**	
Primary Actor:	Registered User	
Secondary Actors:	BTNC Database, Bank	
Preconditions	BTNC System is online and User is logged in on either App or website's payment methods page.	
Trigger	User selects 'Add Payment Method'.	
Flow of Events		**Events:**
	1.	System displays two selectable icons, listed as 'Credit or Debit Card' and 'Paypal'.
	2.	User selects 'Credit or Debit Card'.
	3.	System requests required information. ***1**
	4.	User inputs required information.
	5.	System externally validates inputted information.
	6.	System creates new payment method in payment methods database.
Alternatives		**Events:**
	2A1	User selects cancel.
	2A2	System redirects User to payment method display page (Use Case: 6).
	2B1	User selects 'Paypal'.
	2B2	System redirects User to paypal login.
	2B3	User logs in via paypal.
	2B4A1	System creates new payment method in payment methods database.
	2B4B1	Validation fails (incorrect details, etc).
	2B4B2	GOTO Event 3
	6A1	Validation fails (incorrect details, etc).
	6A2	GOTO Event 3

Notes:	**1.**	**Required Information:** Card number, expiry date, country, security code.

Title:	Order Ride	
Use Case ID:	10	
Description:	This use case explains the process of an User requesting a ride from a BTNC driver.	
Primary Actor:	Registered User	
Secondary Actors:	Registered Driver, Mapping Service	
Preconditions	BTNC System is online and User is logged in on the App's main page.	
Trigger	User selects 'Request Ride'	
Flow of Events		**Events:**
	1.	System requests the destination from the Primary Actor
	2.	Primary Actor inputs the destination.
	3.	System validates the location.
	4.	System displays route details. ***1**
	5.	System prompts Primary Actor to confirm ride.
	6.	Primary Actor selects confirm.
	7.	System notifies Driver, directs Driver to Primary Actor location.
	8.	Driver notifies system that User has been picked up.
	9.	Driver notifies system that trip is complete.
	10.	System displays finalised trip information to driver and user.
	11.	System stores trip details in BTNC Database under user and driver accounts.
	12.	Server directs trip details to payment processing.
Alternatives		**Events:**
	2A1	User selects cancel.

	2A2	System redirects user to main page.
	4A1	Validation fails.
	4A2	System notifies user of invalid location.
	4A3	GOTO Event 1.
	7A1	Primary Actor selects cancel.
	7A2	GOTO Event 1.
Notes:	**1**	**Route Details:** Cost, trip length, trip time, wait time.

Title:	Order Food	
Use Case ID:	11	
Description:	This use case explains the process of an User requesting a food delivery from a registered food outlet delivered by a BTNC driver.	
Primary Actor:	Registered User	
Secondary Actors:	Registered Driver (Driver), Registered Food Outlet (Restaurant)	
Preconditions	BTNC System is online and User is logged in on the App’s main page.	
Trigger	User selects ‘Order Food’	
Flow of Events		**Events:**
	1.	System requests delivery destination.
	2.	Primary Actor inputs the destination.
	3.	System validates the location.
	4.	System redirects Primary Actor to Active Restaurant List page. *1
	5.	Primary Actor selects desired restaurant.
	6.	System redirects Primary Actor to Restaurants Menu page.
	7.	Primary Actor adds food items to cart.
	8.	Primary Actor selects ‘View Order’.
	9.	System redirects Primary Actor to Checkout page and displays cart contents as well as trip details. *2
	10.	System prompts Primary Actor to confirm order.
	11.	Primary Actor selects ‘Confirm’.

	12.	System notifies restaurant of order and finds available driver.
	13.	System directs driver to Restaurant for pickup.
	14.	System directs driver to Primary Actor for drop-off.
	15.	Driver travels to drop off location.
	16.	Driver notifies server that delivery is completed..
	17.	User notifies server that delivery is completed.
	18.	Server directs order details to payment processing.
	19.	System stores trip/order details in BTNC Database under user and driver accounts.
Alternatives		**Events:**
	2A1	Primary Actor selects cancel.
	2A2	System redirects Primary Actor to main page.
	3A1	Validation Fails
	3A2	System notifies Primary Actor of invalid location.
	3A3	GOTO Event 1.
	10A1	Primary Actor requests to add more items to cart
	10A2	GOTO Event 4.
	10B1	Primary Actor removes item from cart.
	10B2	System updates cart contents.
Notes:	**1.**	**Active Restaurant List:** List of restaurants in system that are within range and are open for order.
	2.	**Trip Details:** Cost, estimated time of arrival.

Title:	Order Bike	
Use Case ID:	12	
Description:	This use case explains the process of an User requesting a bike drop-off from a BTNC driver	
Primary Actor:	Registered User	
Secondary Actors:	Registered Driver, MappingService, Database, Application/System, Server	
Preconditions	BTNC System is online and User is logged in on the App's main page.	
Trigger	User selects 'Order Bike to Location'	
Flow of Events		**Events:**
	1.	System requests the drop off location from the Primary Actor .
	2.	User inputs the destination.
	3.	MappingService validates the location.
	4.	Application prompts server of a request for a bike drop off
	5.	Server sends a service request from any driver online
	6.	Driver Accepts service
	7.	Server requests mapping service for directions
	8.	Driver is displayed the drop off location of the bike
	9.	Application and mapping service directs driver to drop off location.
	10.	User selects that bike has been received.
	11	Driver Selects bikes been dropped off
	12	Server directs details of the order to payment processing (another use case)
	13	Server updates order details into the database of driver and user
Alternatives		**Events:**
	2A1	Primary Actor selects cancel.
	2A2	System redirects Primary Actor to main page.
	3A1	Validation fails.

	3A2	System notifies Primary Actor of invalid location.
	3A3	GOTO Event 1.
	10A1	User selects bike hasn’t been received
	10A2	Contact details of driver and uber is displayed on screen for resolution
	9A1	Driver cancels drop off
	9A2	User is notified the driver canceled, and prompts server to search for another one GOTO 5
Notes:	**1.**	**Bike Rack Location:** the closest bike rack that actually has bikes available.
	2.	**Trip Details:** Cost, estimated time of arrival.

Title:	Pick Up Bike	
Use Case ID:	13	
Description:	This use case explains the process of an User requesting a bike pickup from a BTNC Bike Rack.	
Primary Actor:	Registered User	
Secondary Actors:	MappingService, BTNC Database, Application, BTNC Server	
Preconditions	BTNC System is online and User is logged in on the App’s main page.	
Trigger	Primary Actor selects ‘Request to Pick-up Bike’	
Flow of Events		**Events:**
	1.	System requests the current location of the Primary Actor.
	2.	Primary Actor inputs current location.
	3.	System validates location.
	4.	System displays closest bike rack location to Primary Actor(***1**) and pickup details (***2**).
	5.	System prompts Primary Actor to confirm pickup.
	6.	System directs Primary Actor to bike rack location and gives bike unlock passcode.
	7.	Primary Actor unlocks bike using passcode.
	8.	Bike rack notifies system that bike has been picked up.
Alternatives		**Events:**
	2A1	Primary Actor selects cancel.
	2A2	System redirects Primary Actor to main page.
	3A1	Validation Fails
	3A2	System notifies Primary Actor of invalid location.
	3A3	GOTO Event 1.
Notes:	**1.**	**Bike Rack Location:** the closest bike rack that actually has bikes available.
	2.	**Pick-up Details:** Cost, distance from bike rack..

Title:	Pickup Passenger/Start Trip	
Use Case ID:	14	
Description:	This use case explains the process of an User requesting to pickup a passenger and drop them to a destination.	
Primary Actor:	Registered Driver	
Secondary Actors:	Registered User	
Preconditions	BTNC System is online, driver is logged in on the App’s main page.	
Trigger	Primary Actor selects ‘Pickup a Passenger’	
Flow of Events		**Events:**
	1.	System uses Primary Actor’s device to find current location.
	2.	System finds the closest Secondary User looking for pickup.
	3.	System displays found pickup location and prompts Primary Actor to select ‘Pickup Rider’. ***1**
	4.	Primary Actor selects ‘Pickup Rider’.
	5.	System directs Primary Actor to Secondary User’s pickup location.
	6.	Primary Actor notifies system that Secondary User is in the vehicle.
	7.	System directs Primary Actor to Secondary User’s drop-off location.
	8.	Upon arrival system prompts Primary Actor to select ‘Finish Trip’ .
	9.	Primary Actor selects ‘Finish Trip’.
	10.	System displays trip details to Primary Actor. ***2**
Alternatives		**Events:**
	1A1	System unable to find Primary Actor location.
	1A2	System displays error message.
	1A3	GOTO App Home page.
	2A1	There are no Secondary Actors waiting for pickup.
	2A2.	System notifies Primary Actor.
	2A3.	GOTO Event 1.
	2B1	Primary Actor selects cancel.

	2B2	System redirects Primary Actor to main page.
	4A1	Primary Actor selects cancel.
	4A2	System redirects Primary Actor to main page.
	8A1	Primary Actor selects 'Finish Trip' before reaching destination.
	8A2	System updates trip details ***2**
	8A3	GOTO Event 10
Notes:	**1**	**Found Pickup Location:** pickup location of Secondary User
	2	**Trip Details:** payment amount, total time, distance travelled.

Title:	Pickup Food/Drop Off Food	
Use Case ID:	15	
Description:	This use case explains the process of an User requesting to pickup a food order and deliver it to a user.	
Primary Actor:	Registered Driver (Driver)	
Secondary Actors:	Registered User (Customer), Registered Food Outlet (Restaurant)	
Preconditions	BTNC System is online, driver is logged in on the App's main page.	
Trigger	Primary Actor	
Flow of Events		**Events:**
	1.	System uses Drivers device to find current location.
	2.	System finds the closest Restaurant that has an order without a driver.
	3.	System displays Restaurant location and prompts Driver to select 'Accept Delivery'.
	4.	Driver selects 'Accept Delivery'.

	5.	System directs Driver to Restaurant location.
	6.	Restaurant notifies System that driver has picked up delivery.
	7.	System directs driver to User delivery location.
	8.	Driver notifies system that delivery is completed.
	9.	System displays delivery details to Driver.
Alternatives		**Events:**
	1A1	System unable to find Driverslocation.
	1A2	System displays error message.
	1A3	GOTO App Home page.
	2A1	There are no Restaurants waiting for driver.
	2A2.	System notifies Driver.
	2A3.	GOTO Event 1.
	2B1	Driver selects cancel.
	2B2	System redirects Primary Actor to main page.
	4A1	Primary Actor selects cancel.
	4A2	System redirects Primary Actor to main page.
	8A1	Primary Actor selects ‘Finish Delivery’ before reaching destination.
	8A2	System updates trip details ***2**
	8A3	GOTO Event 9.

Title:	Creating Restaurant Account	
Use Case ID:	16	
Description:	This use case explains the process of an User creating a restaurant account to offer food on the BTNC service.	
Primary Actor:	Restaurant Personnel	
Secondary Actors:	BTNC Database, BTNC Server	
Preconditions	BTNC System is online and User is on either App or website's create account page.	
Trigger	User selects 'Create Restaurant Account'.	
Flow of Events		**Events:**
	1.	System requests Users restaurant information. ***1**
	2.	User inputs required information.
	3.	System requests Users menu information. ***2**
	4.	User inputs required information.
	5.	System requests email confirmation, sends confirmation email to address user provided.
	6.	System create new user in accounts database.
Alternatives		**Events**
	2A1	User selects cancel.
	2A2	System redirects User to login page.
	4A1	User selects cancel.
	4A2	System redirects User to login page.
	6A1	Confirmation is not received within sufficient time.
	6A2	Upon next attempt to create account, system will GOTO Event 1.
Notes:	**1.**	***Restaurant Information:** Name, Location, Manager, etc.
	2.	**Menu Information:** Dishes, Price, Ingredients.

Title:	Update Menu Items	
Use Case ID:	17	
Description:	This use case explains the process of an User updating the Menu items linked with the logged in restaurant account	
Primary Actor:	Registered Restaurant (Restaurant)	
Secondary Actors:	BTNC Server, BTNC Database	
Preconditions	BTNC System is online and User is logged in on either App or website's update menu items page,	
Trigger	User selects 'Update Menu Items	
Flow of Events		**Events:**
	1.	System displays current Menu Items associated with logged in Users account.
	2.	User selects which item they wish to update.
	3.	System displays the item details. ***1**
	4.	User selects the detail they wish to update.
	5.	System prompts User to input the updated detail.
	6.	User inputs the updated detail.
	7.	System prompts the user to confirm they wish to update the selected detail.
	8.	User selects confirm.
	9.	Menu Item information is updated in the Menu Items database..
Alternatives		**Events:**
	2A1	User selects 'Add New Menu Item'.
	2A2	System redirects user to 'Add New Menu Item' page *(Use Case: 18).
	2B1	User selects cancel.
	2B2	System redirects User to user home page..
	6A1	User selects cancel.
	6A2	GOTO Event 1.
	8A1	User selects cancel.

	8A2	GOTO Event 3.
Notes:	**1.**	***Item Details:** Name, Price, Ingredients, Picture.

Title:	Add Menu Item	
Use Case ID:	18	
Description:	This use case explains the process of an User adding an additional Menu Item. ***1**	
Primary Actor:	Registered Restaurant (Restaurant)	
Secondary Actors:	BTNC Server, BTNC Database	
Preconditions	BTNC System is online and User is logged in on either App or website's update menu items page,	
Trigger	User selects 'Add Menu Item'.	
Flow of Events		**Events:**
	1.	System prompts User for item details. ***1**
	2.	User inputs required details.
	3.	System creates new Menu Item in menu items database.
Alternatives		**Events:**
	2A1	User selects cancel.
	2A2	System redirects User to Menu Item display page (Use Case: 17).
Notes:	**1.**	***Item Details:** Name, Price, Ingredients, Picture.

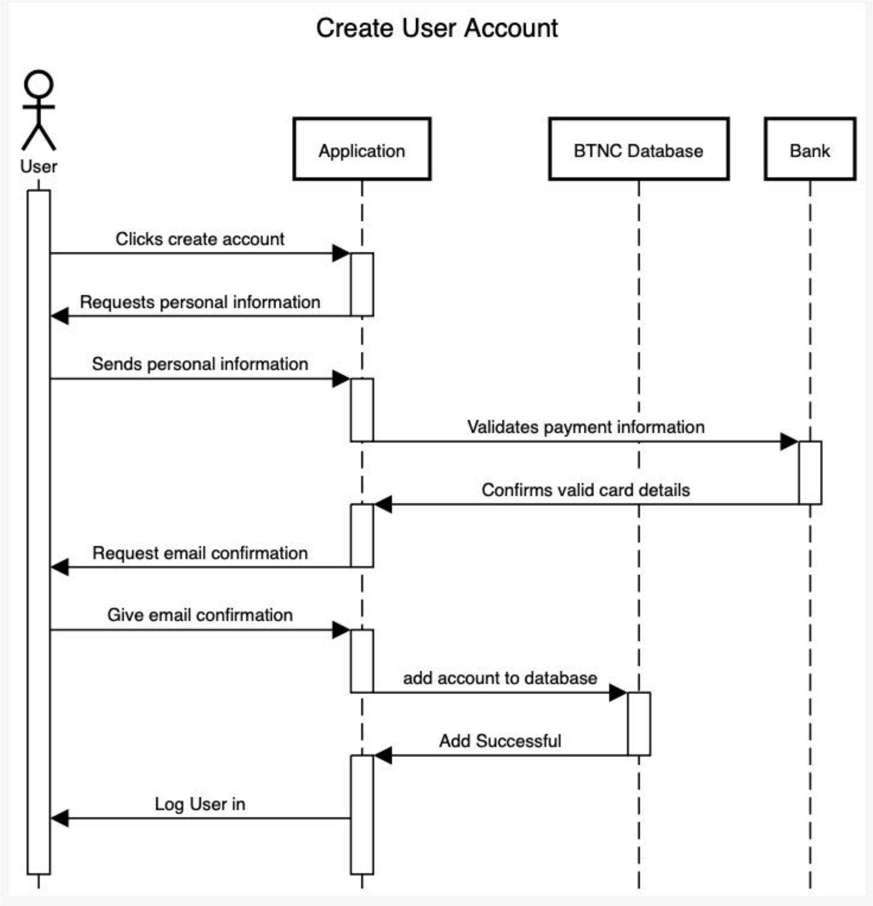
Title:	Logout	
Use Case ID:	19	
Description:	This use case explains the process of an User logging out of the BTNC system.	
Primary Actor:	Registered User	
Secondary Actors:	Application	
Preconditions	BTNC System is online and User is logged in on either App or website's main page.	
Trigger	User selects 'Logout'.	
Flow of Events		**Events:**
	1.	System prompts to confirm their decision to logout.
	2.	User selects confirm.
	3.	System logs User out of BTNC App or website.
Alternatives		**Events:**
	2A1	User selects cancel.
	2A2	System redirects User to the main page.
Notes:		

Title:	Create Driver Account	
Use Case ID:	20	
Description:	This use case explains the process of an User creating an account to use as a Driver on the BTNC system	
Primary Actor:	Guest User	
Secondary Actors:		
Preconditions	BTNC System is online and User is on either App or website’s create account page.	
Trigger	User selects ‘Create Driver Account’	
Flow of Events		**Events:**
	1.	System requests users personal information. ***1**
	2.	User inputs required information.
	3.	System requests license and vehicle information. ***2**
	4.	User inputs required information.
	5.	System requests email confirmation, sends confirmation email to address user provided.
	6.	User presses confirmation link.
	7.	System create new Driver in accounts database.
	8.	System sends email to User advising them on how to proceed with preparation of themselves and their vehicle to act as a BTNC Driver.
Alternatives		**Events:**
	2A1	User selects cancel.
	2A2	System redirects user to main page.
	4A1	User selects cancel.
	4A2	System redirects user to main page.
	6A1	Confirmation is not received within sufficient time.
	6A2	Upon next attempt to create account, system will GOTO Event 1.
Notes:	**1.**	***Personal Information:** Full name, billing address, mobile number, email address, payment information (Credit Card Details/PayPal)
	2.	***Vehicle Information:** Vehicle manufacturer, model, year of

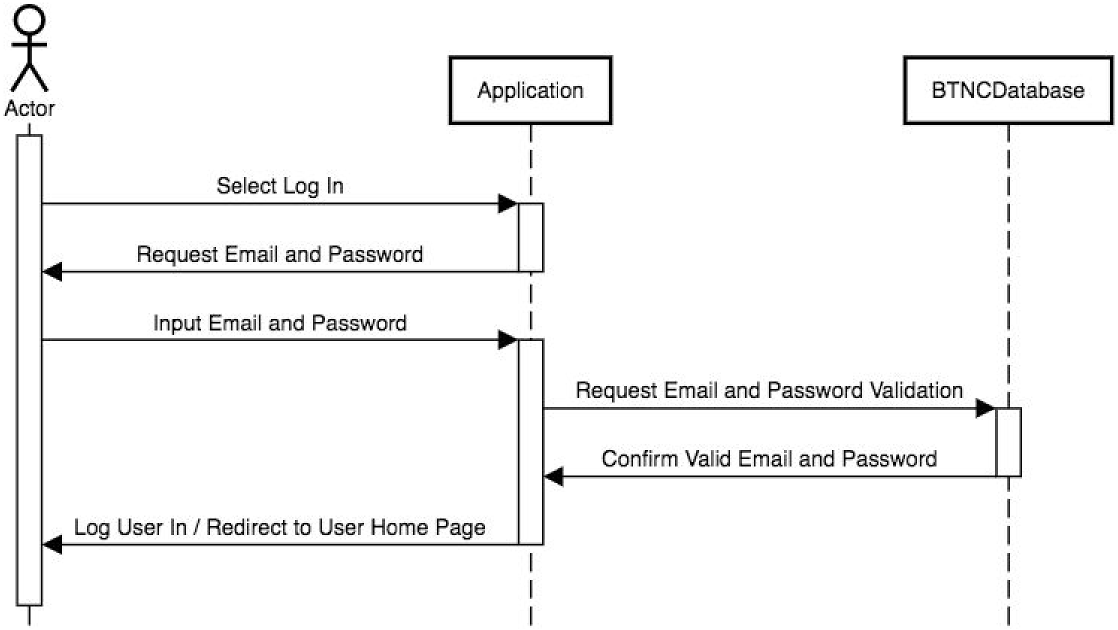
		manufacture. plate number, copy of driver's license.

Title:	Payment Processing	
Use Case ID:	21	
Description:	Payment processing, updating the drivers earning and charging the users account	
Primary Actor:	Database	
Secondary Actors:	Bank, server	
Preconditions	Service has finished, user and driver both have valid payment details.	
Trigger	Both the user and driver select finish ride/drop off	
Flow of Events		**Events:**
	1.	Server requests database for order details
	2.	Server requests user payment details from database
	3.	Server requests driver/restaurant details
	4	Server request bank charge user and put x amount into drivers/restaurant owner account
Alternatives		**Events:**
	4A1	Not enough money in bank account
	4A2	Bank follows that up with user
Notes:		

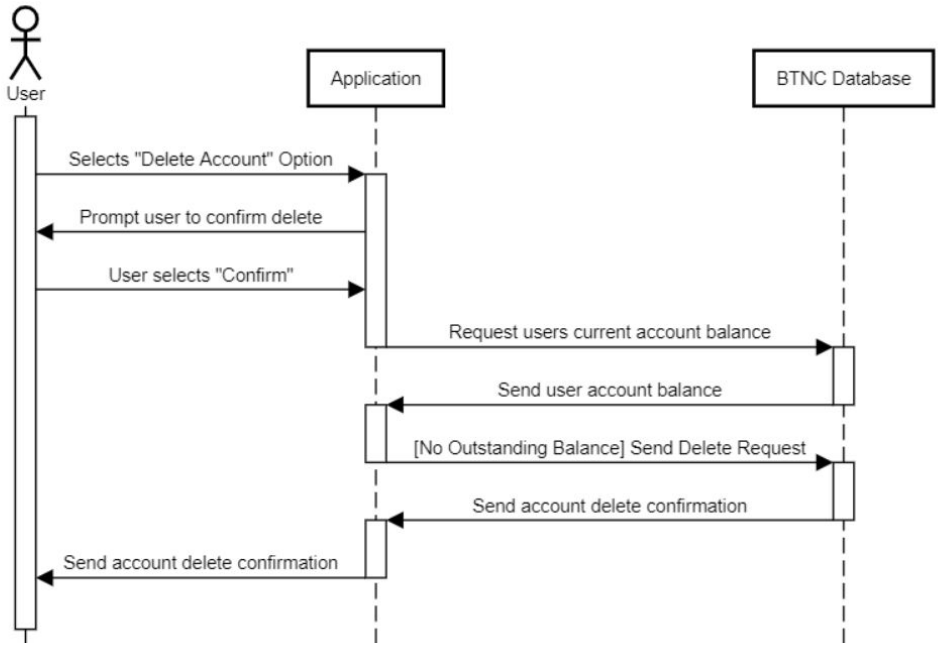
9.4.2 Sequence Diagrams



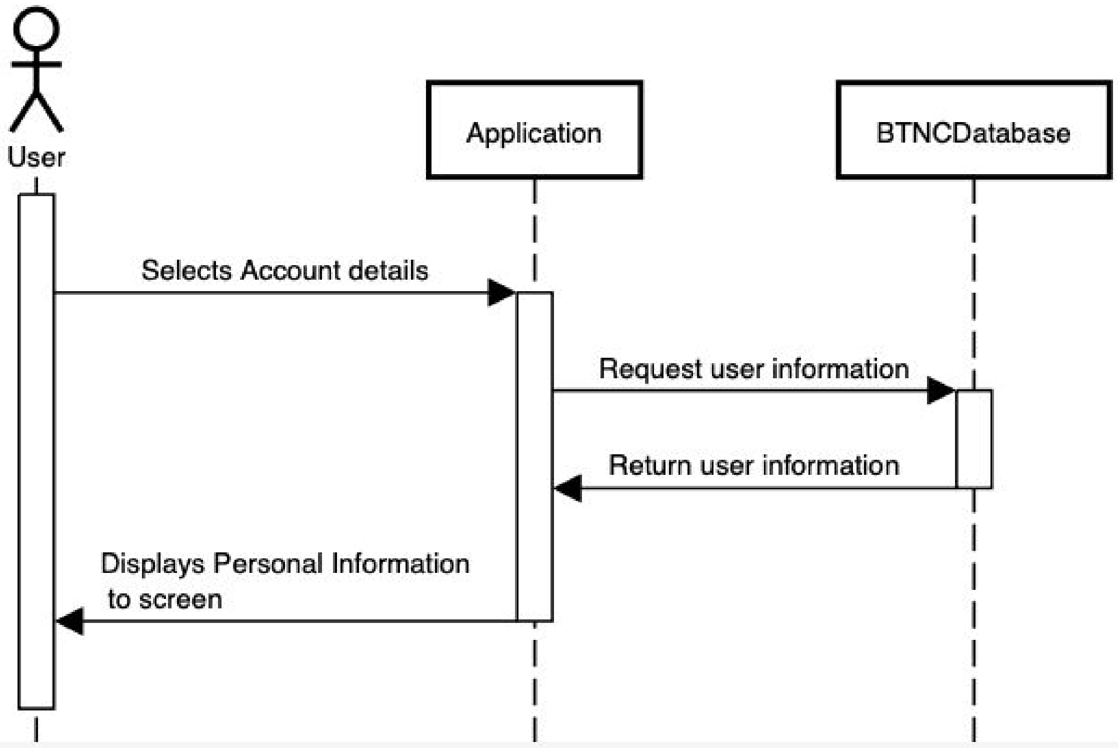
Log In


Actor
Application
BTNCDatabase
Select Log In
Request Email and Password
Input Email and Password
Request Email and Password Validation
Confirm Valid Email and Password
Log User In / Redirect to User Home Page

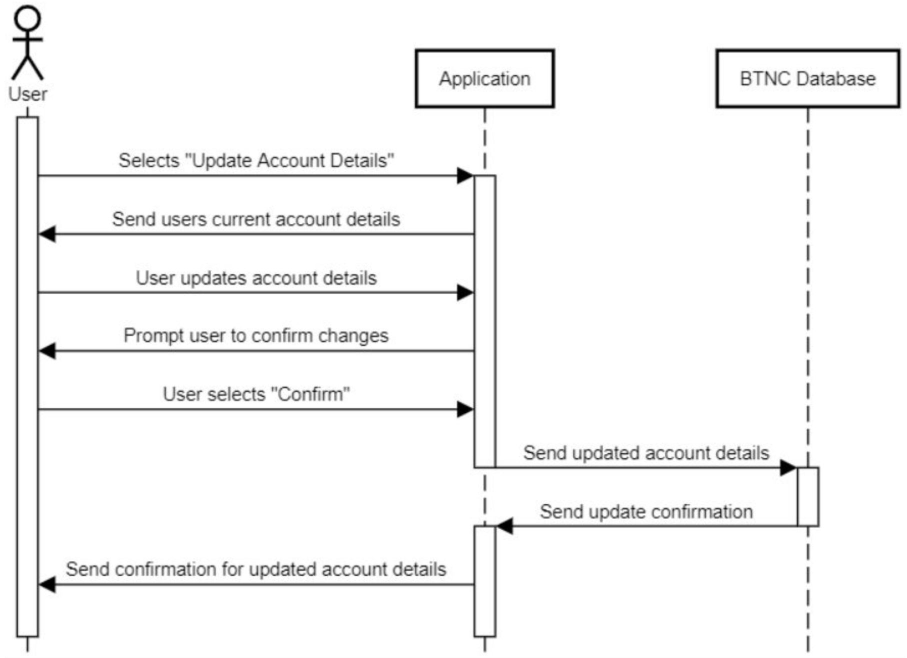
Delete User Account


User
Application
BTNC Database
Selects "Delete Account" Option
Prompt user to confirm delete
User selects "Confirm"
Request users current account balance
Send user account balance
[No Outstanding Balance] Send Delete Request
Send account delete confirmation
Send account delete confirmation

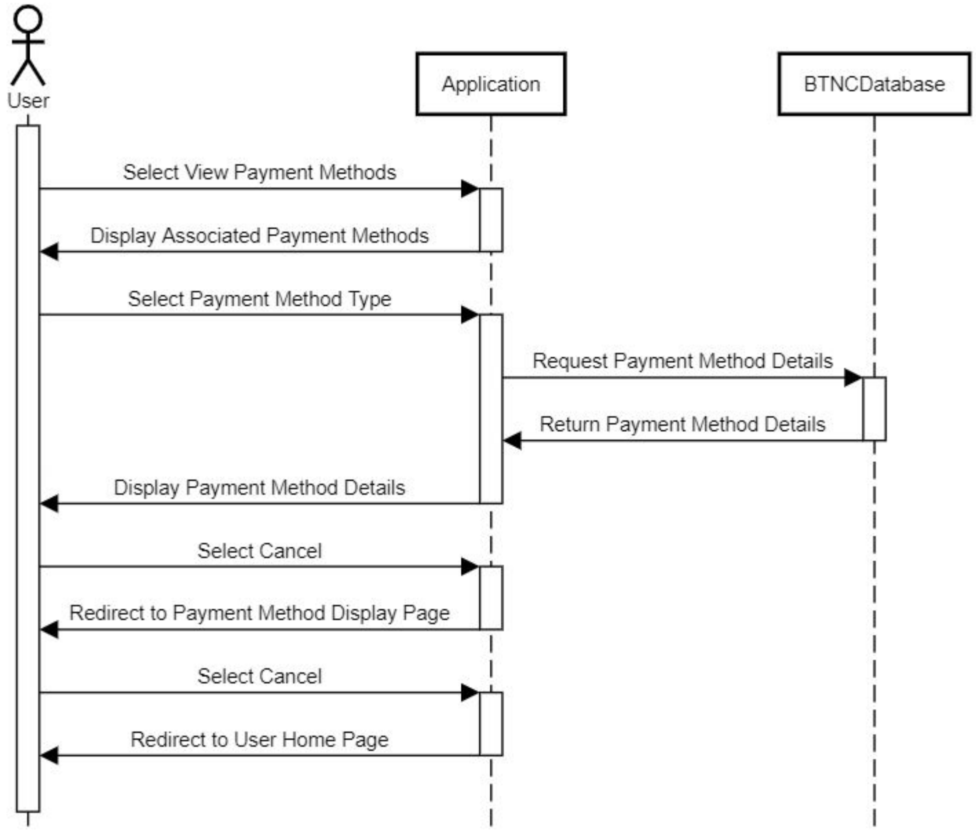
View Account Details



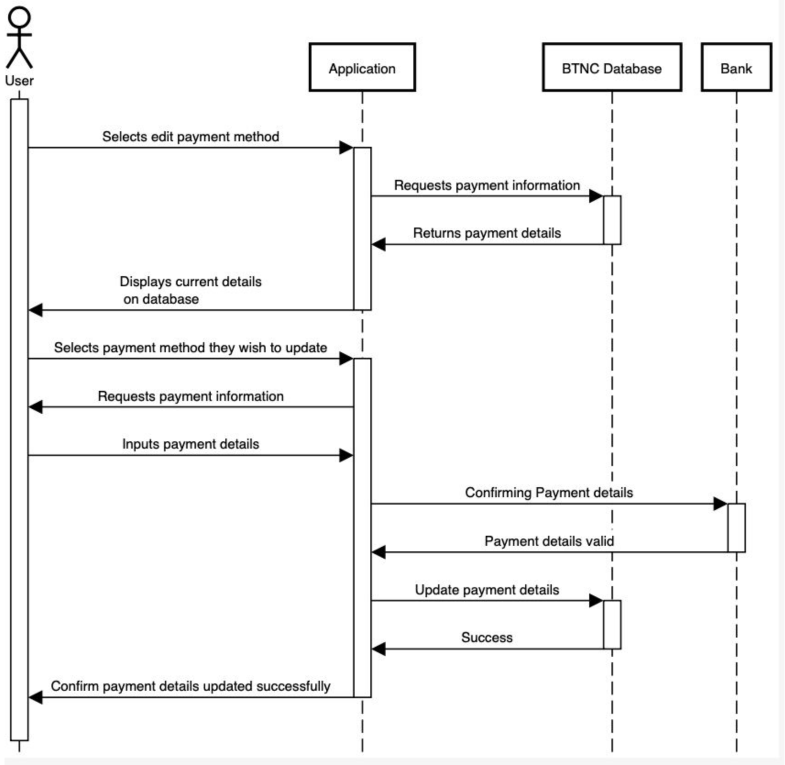
Update Account Details


User
Application
BTNC Database
Selects "Update Account Details"
Send users current account details
User updates account details
Prompt user to confirm changes
User selects "Confirm"
Send updated account details
Send update confirmation
Send confirmation for updated account details

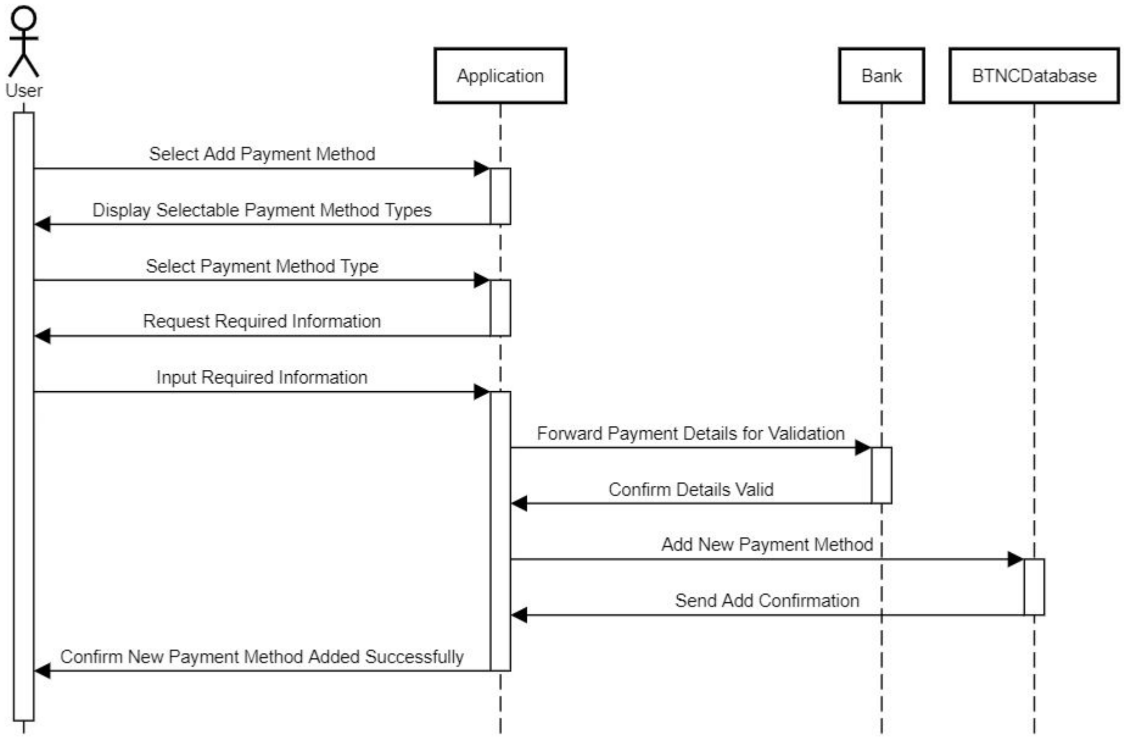
View Payment Methods


User
Application
BTNCDatabase
Select View Payment Methods
Display Associated Payment Methods
Select Payment Method Type
Request Payment Method Details
Return Payment Method Details
Display Payment Method Details
Select Cancel
Redirect to Payment Method Display Page
Select Cancel
Redirect to User Home Page

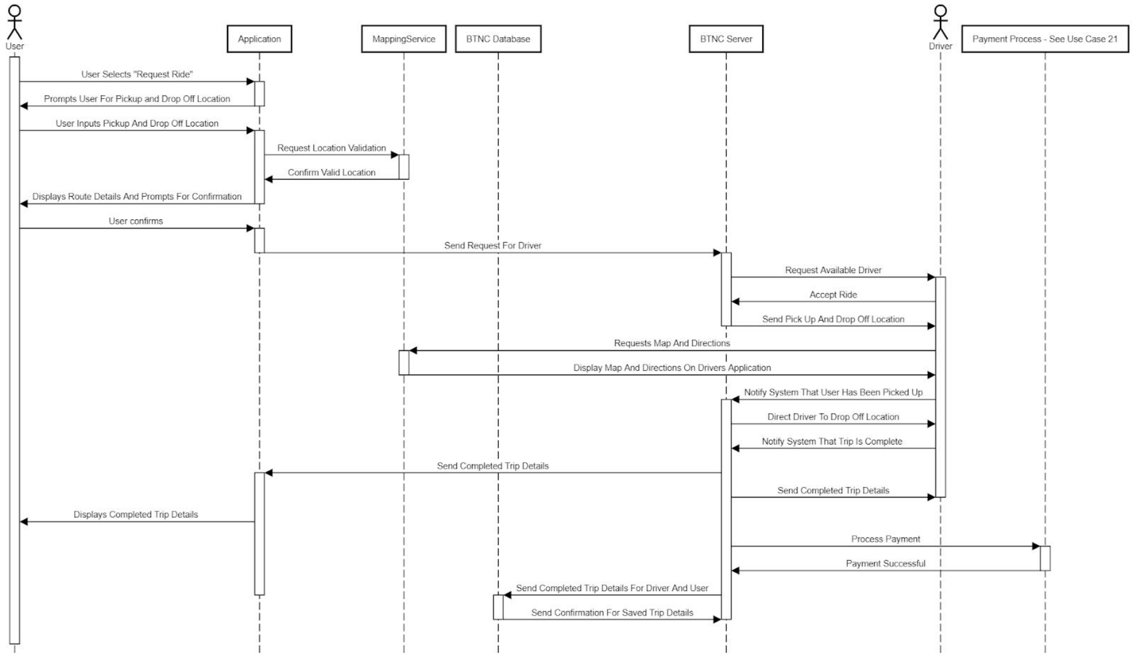
Update Payment Method


User
Application
BTNC Database
Bank
Selects edit payment method
Requests payment information
Returns payment details
Displays current details
on database
Selects payment method they wish to update
Requests payment information
Inputs payment details
Confirming Payment details
Payment details valid
Update payment details
Success
Confirm payment details updated successfully

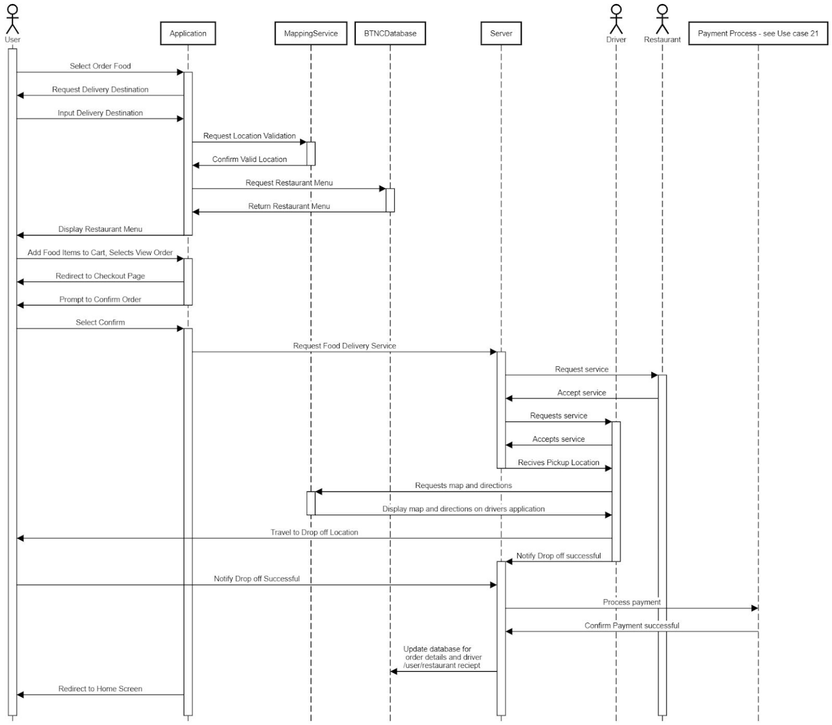
Add Payment Method


User
Application
Bank
BTNCDatabase
Select Add Payment Method
Display Selectable Payment Method Types
Select Payment Method Type
Request Required Information
Input Required Information
Forward Payment Details for Validation
Confirm Details Valid
Add New Payment Method
Send Add Confirmation
Confirm New Payment Method Added Successfully

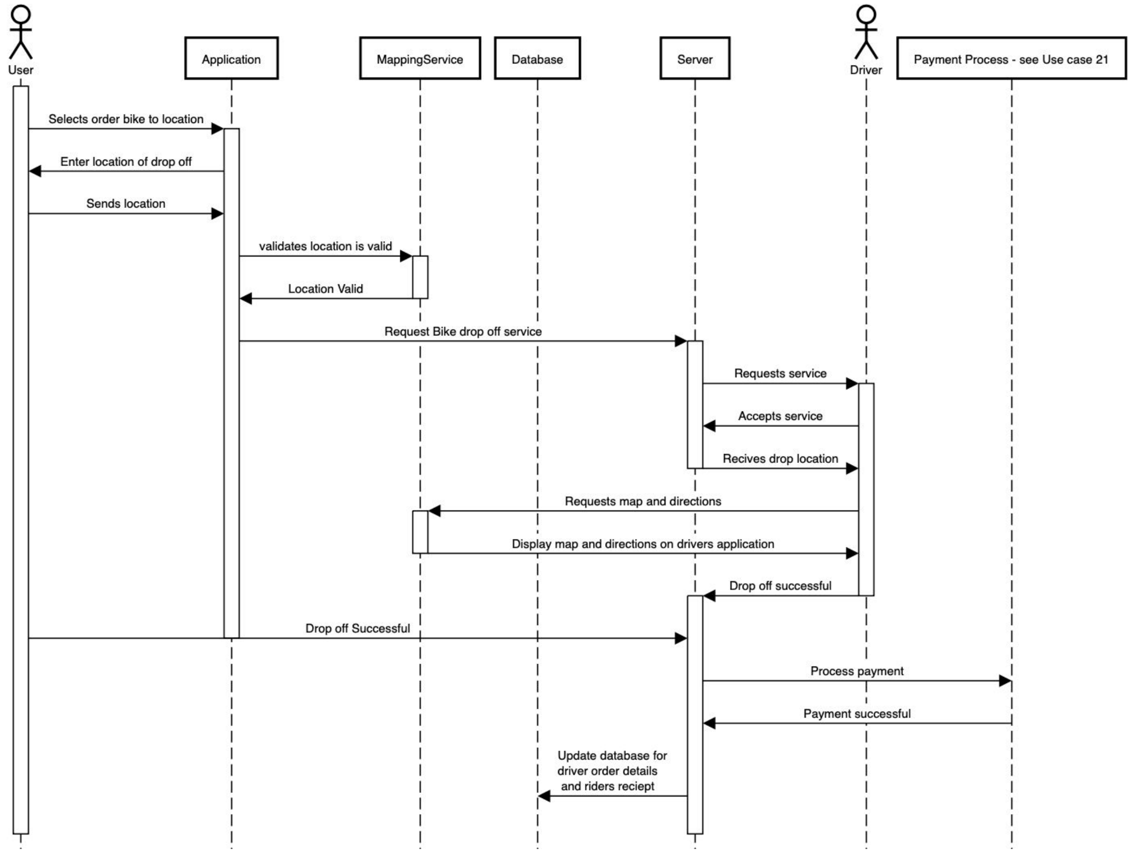
Order Ride


User
Application
MappingService
BTNC Database
BTNC Server
Driver
Payment Process - See Use Case 21
User Selects "Request Ride"
Prompts User For Pickup and Drop Off Location
User Inputs Pickup And Drop Off Location
Request Location Validation
Confirm Valid Location
Displays Route Details And Prompts For Confirmation
User confirms
Send Request For Driver
Request Available Driver
Accept Ride
Send Pick Up And Drop Off Location
Requests Map And Directions
Display Map And Directions On Drivers Application
Notify System That User Has Been Picked Up
Direct Driver To Drop Off Location
Notify System That Trip Is Complete
Send Completed Trip Details
Send Completed Trip Details
Displays Completed Trip Details
Process Payment
Payment Successful
Send Completed Trip Details For Driver And User
Send Confirmation For Saved Trip Details

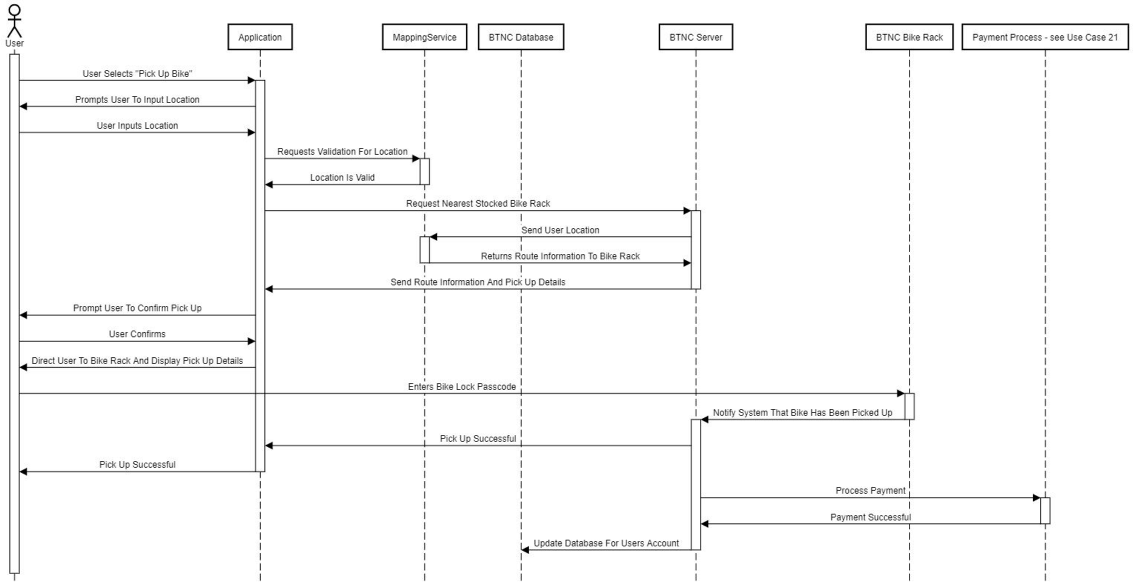
Order Food



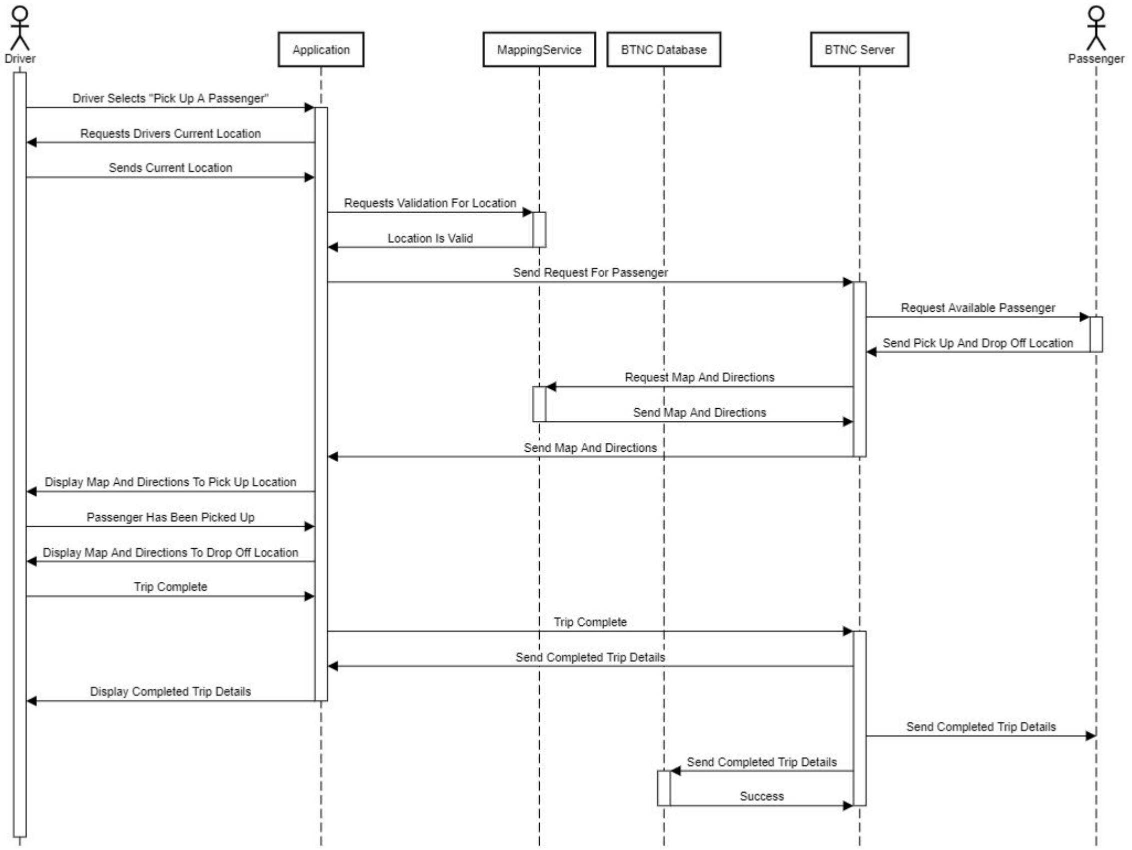
Order Bike



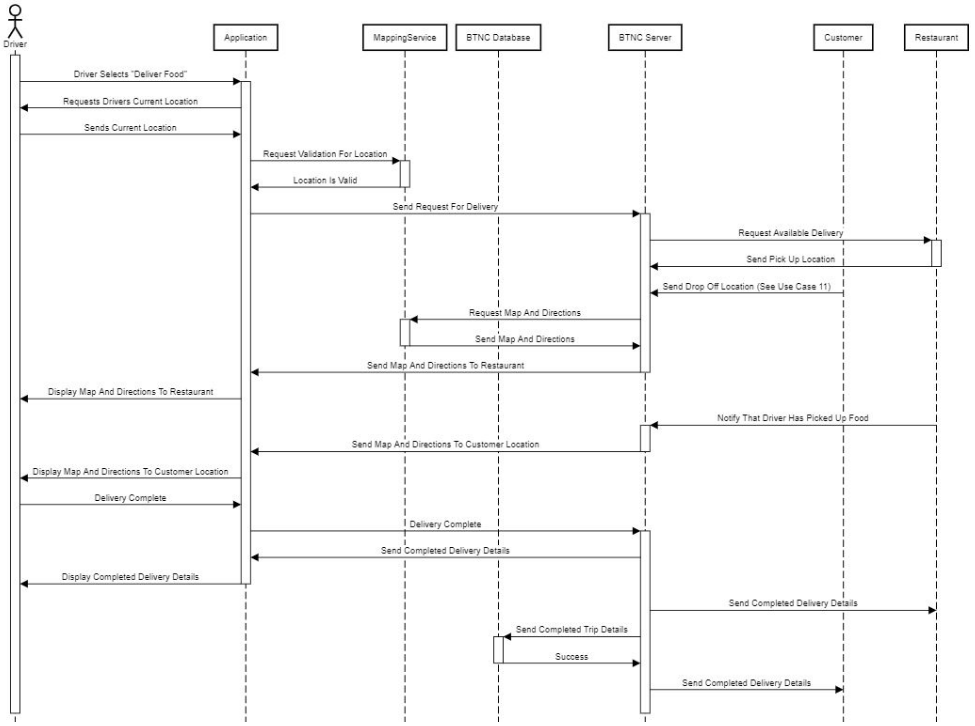
Pick Up Bike


User
Application
MappingService
BTNC Database
BTNC Server
BTNC Bike Rack
Payment Process - see Use Case 21
User Selects "Pick Up Bike"
Prompts User To Input Location
User Inputs Location
Requests Validation For Location
Location Is Valid
Request Nearest Stocked Bike Rack
Send User Location
Returns Route Information To Bike Rack
Send Route Information And Pick Up Details
Prompt User To Confirm Pick Up
User Confirms
Direct User To Bike Rack And Display Pick Up Details
Enters Bike Lock Passcode
Notify System That Bike Has Been Picked Up
Pick Up Successful
Pick Up Successful
Process Payment
Payment Successful
Update Database For Users Account

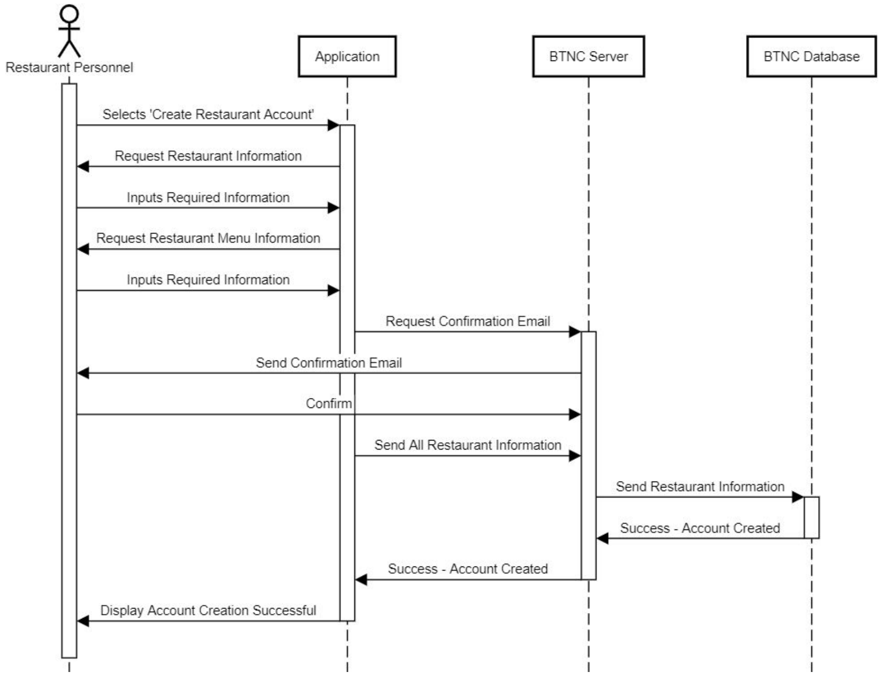
Pickup Passenger/Start Trip


Driver
Application
MappingService
BTNC Database
BTNC Server
Passenger
Driver Selects "Pick Up A Passenger"
Requests Drivers Current Location
Sends Current Location
Requests Validation For Location
Location Is Valid
Send Request For Passenger
Request Available Passenger
Send Pick Up And Drop Off Location
Request Map And Directions
Send Map And Directions
Send Map And Directions
Display Map And Directions To Pick Up Location
Passenger Has Been Picked Up
Display Map And Directions To Drop Off Location
Trip Complete
Trip Complete
Send Completed Trip Details
Display Completed Trip Details
Send Completed Trip Details
Send Completed Trip Details
Success

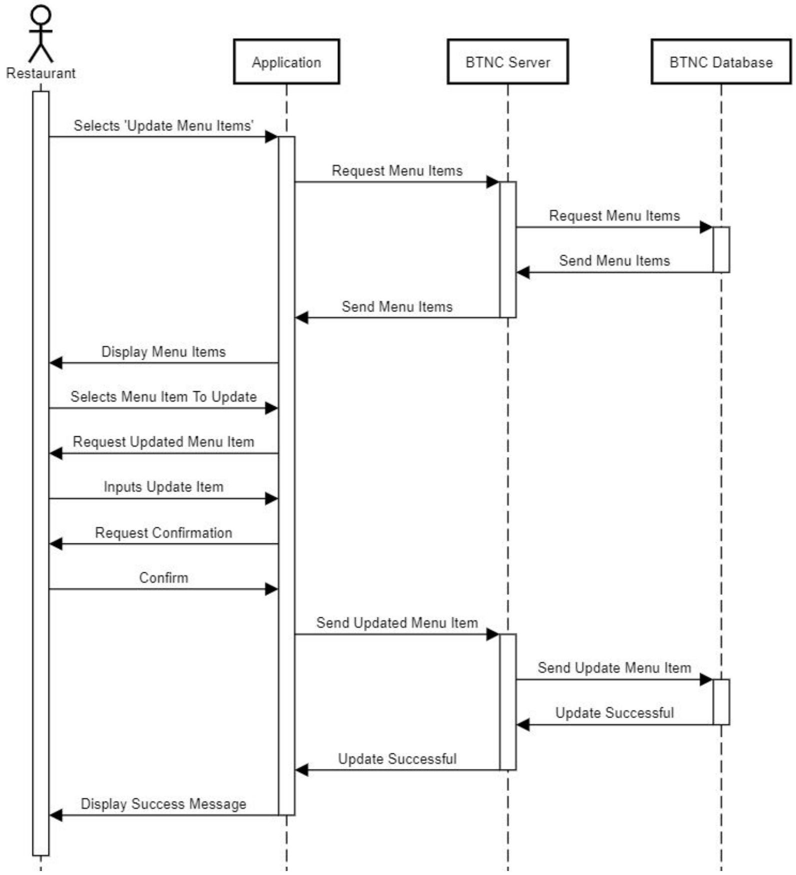
Pick Up/Drop Off Food


Driver
Application
MappingService
BTNC Database
BTNC Server
Customer
Restaurant
Driver Selects "Deliver Food"
Requests Drivers Current Location
Sends Current Location
Request Validation For Location
Location Is Valid
Send Request For Delivery
Request Available Delivery
Send Pick Up Location
Send Drop Off Location (See Use Case 11)
Request Map And Directions
Send Map And Directions
Send Map And Directions To Restaurant
Display Map And Directions To Restaurant
Notify That Driver Has Picked Up Food
Send Map And Directions To Customer Location
Display Map And Directions To Customer Location
Delivery Complete
Delivery Complete
Send Completed Delivery Details
Display Completed Delivery Details
Send Completed Delivery Details
Send Completed Trip Details
Success
Send Completed Delivery Details

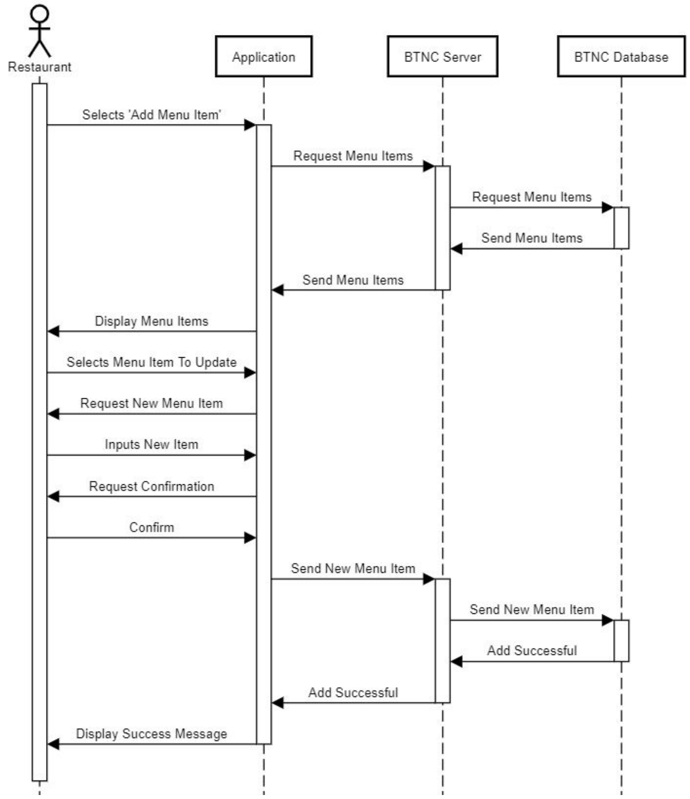
Creating Restaurant Account


Restaurant Personnel
Application
BTNC Server
BTNC Database
Selects 'Create Restaurant Account'
Request Restaurant Information
Inputs Required Information
Request Restaurant Menu Information
Inputs Required Information
Request Confirmation Email
Send Confirmation Email
Confirm
Send All Restaurant Information
Send Restaurant Information
Success - Account Created
Success - Account Created
Display Account Creation Successful

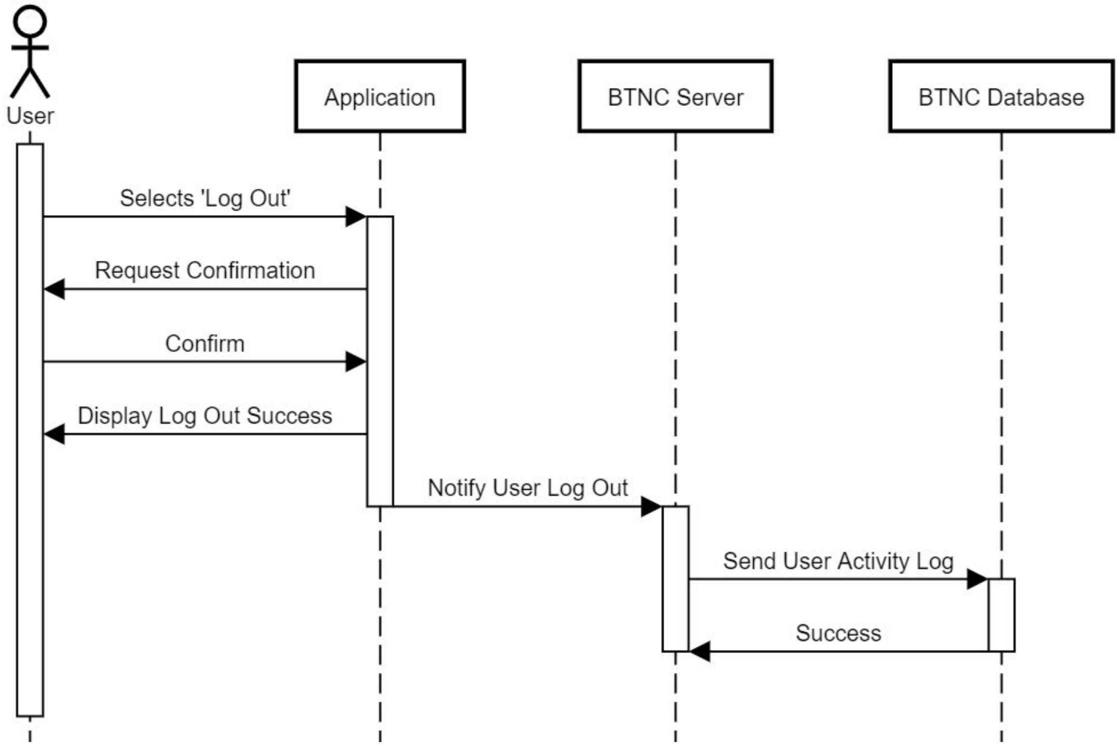
Update Menu Items



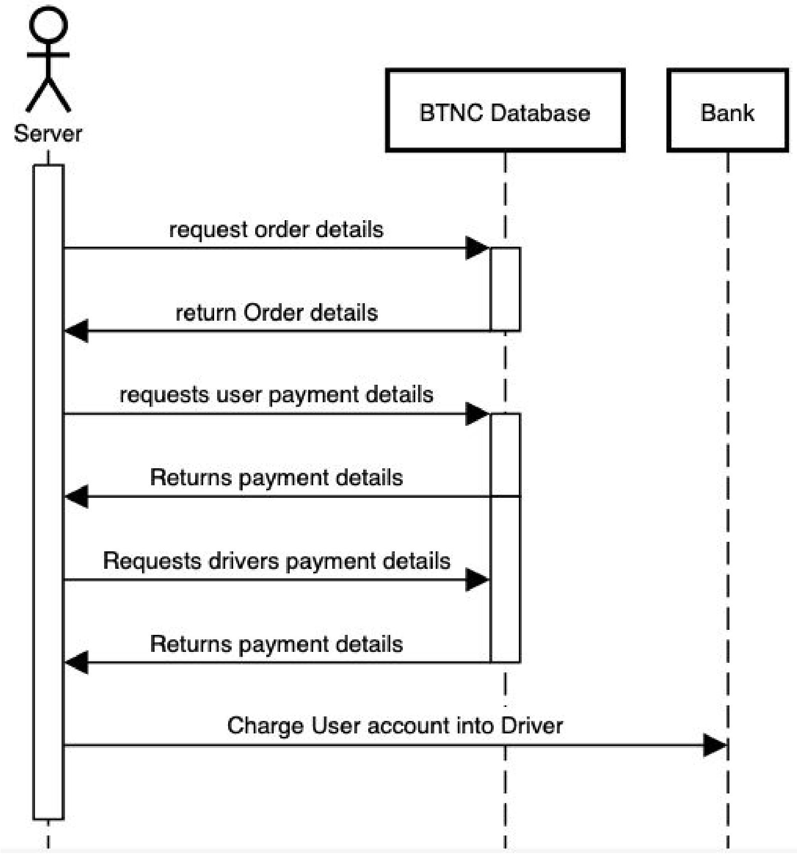
Add Menu Item



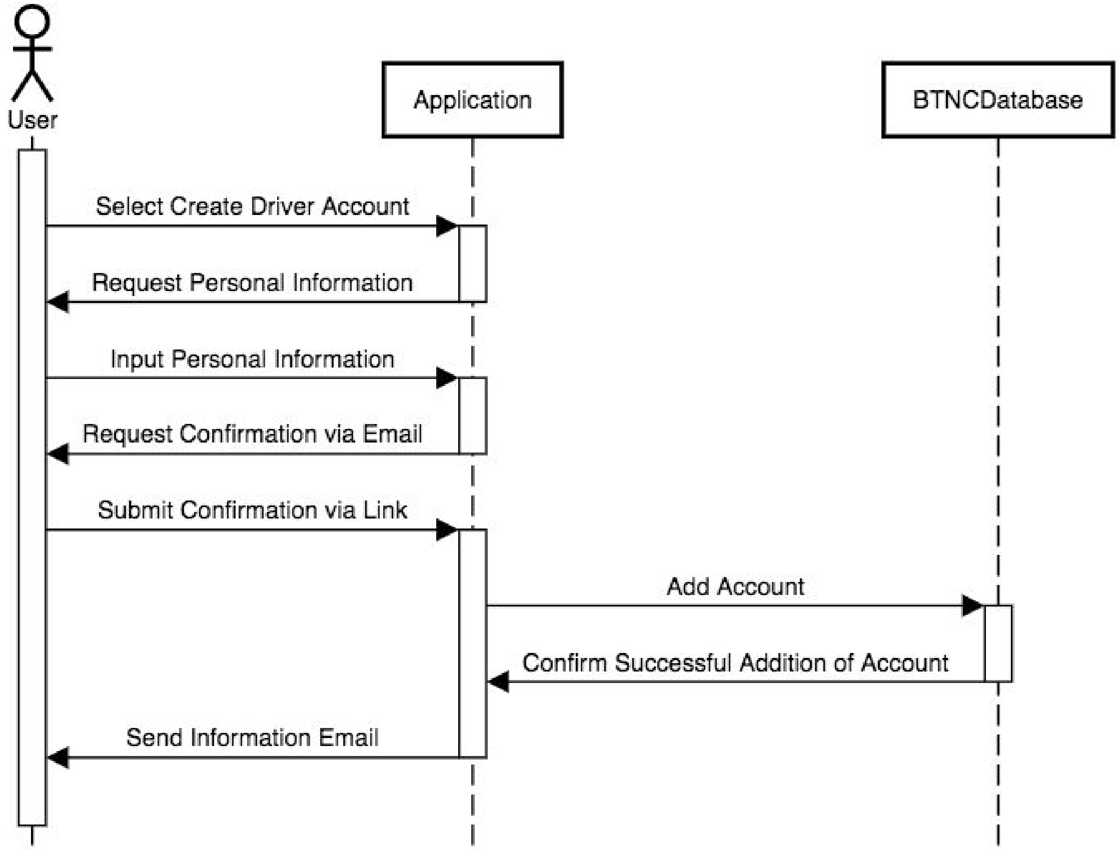
Log Out



Payment Processing

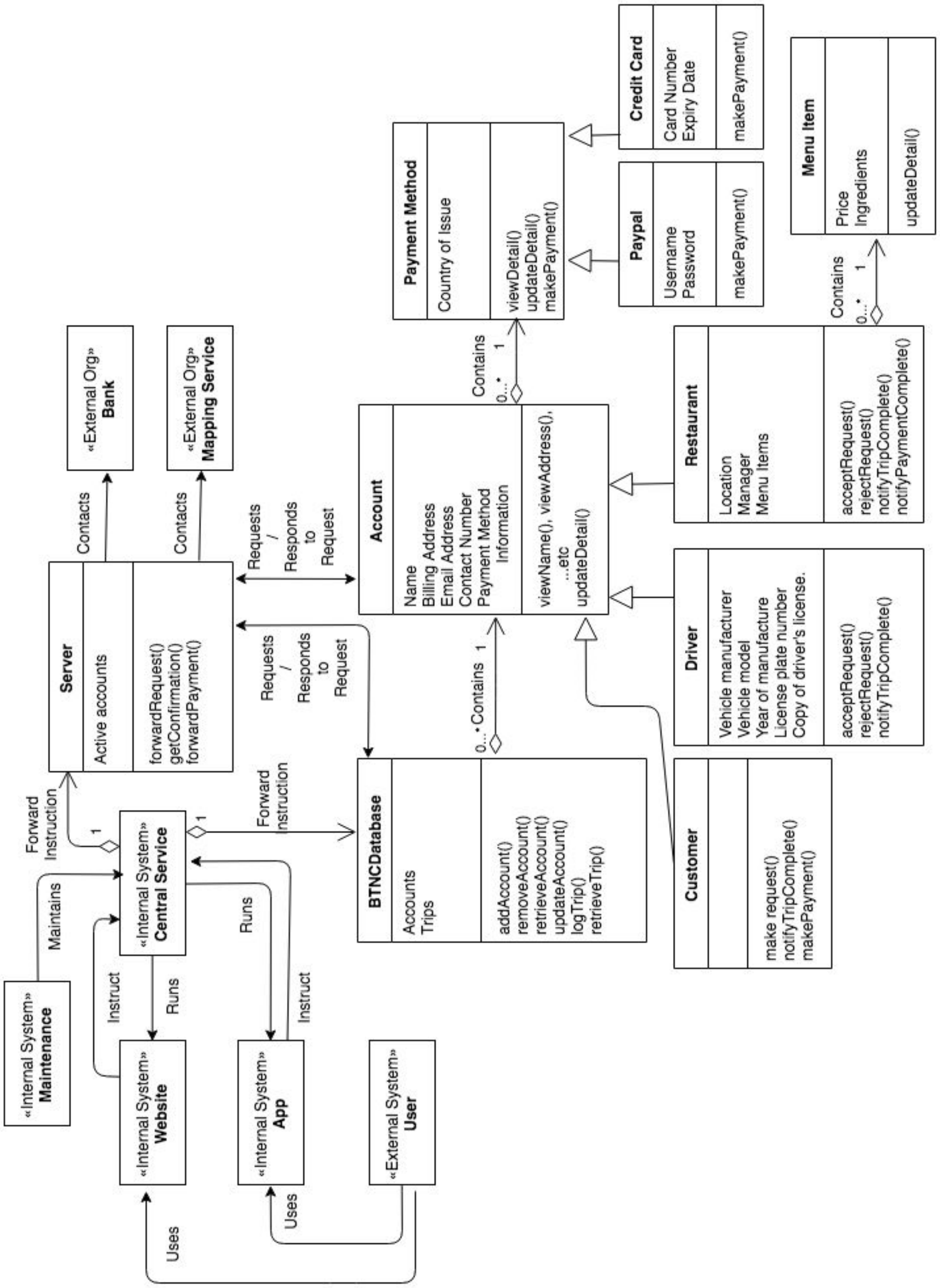


Create Driver Account

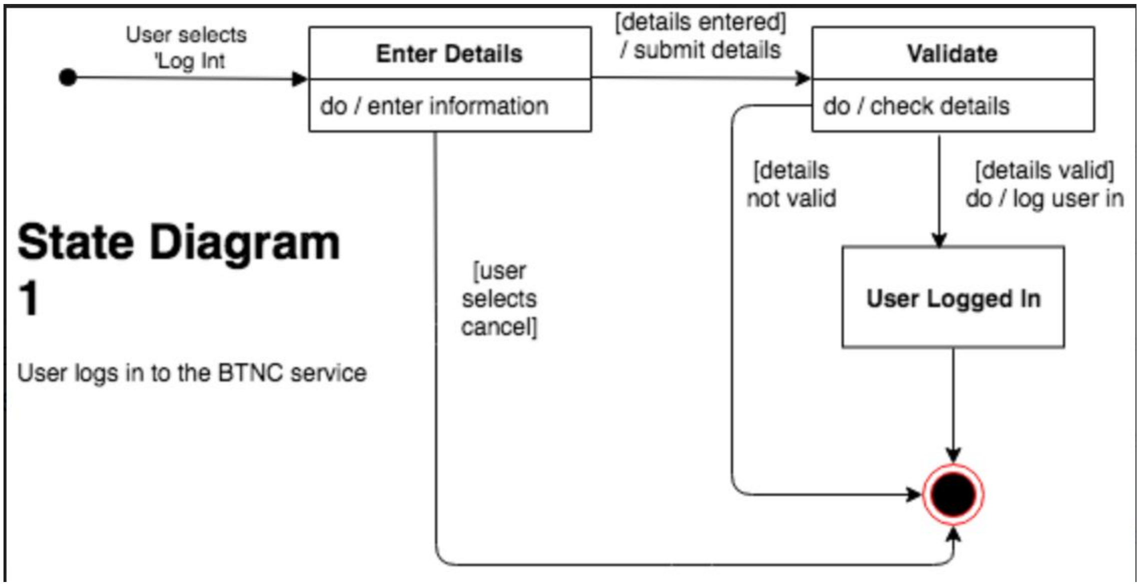

User
Application
BTNCDatabase
Select Create Driver Account
Request Personal Information
Input Personal Information
Request Confirmation via Email
Submit Confirmation via Link
Add Account
Confirm Successful Addition of Account
Send Information Email

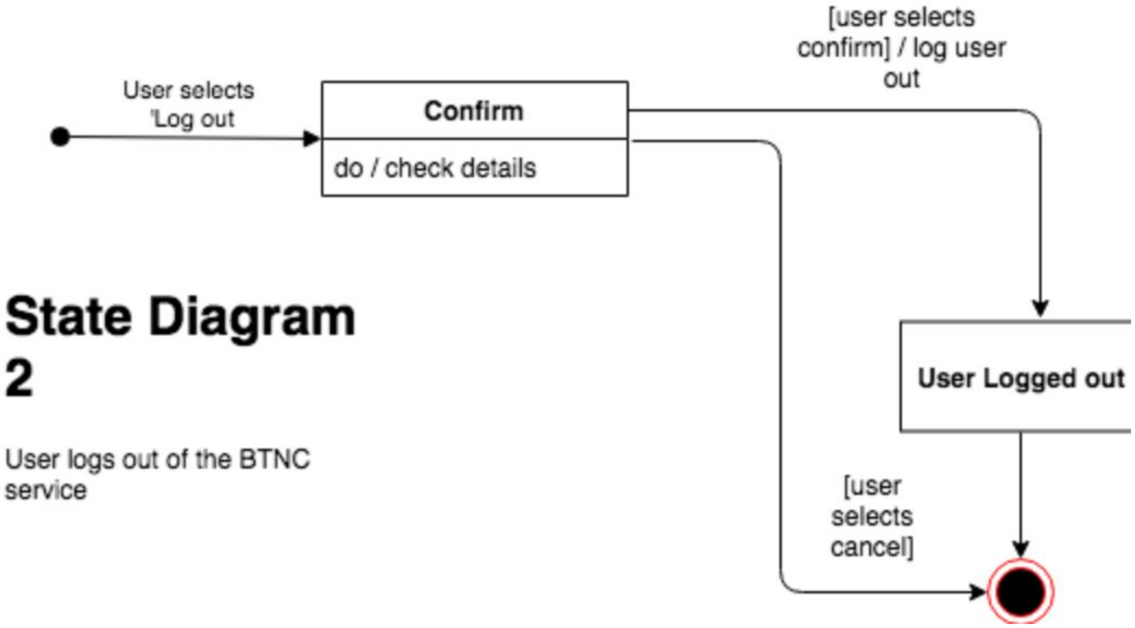
9.5 Appendix - Application Structure

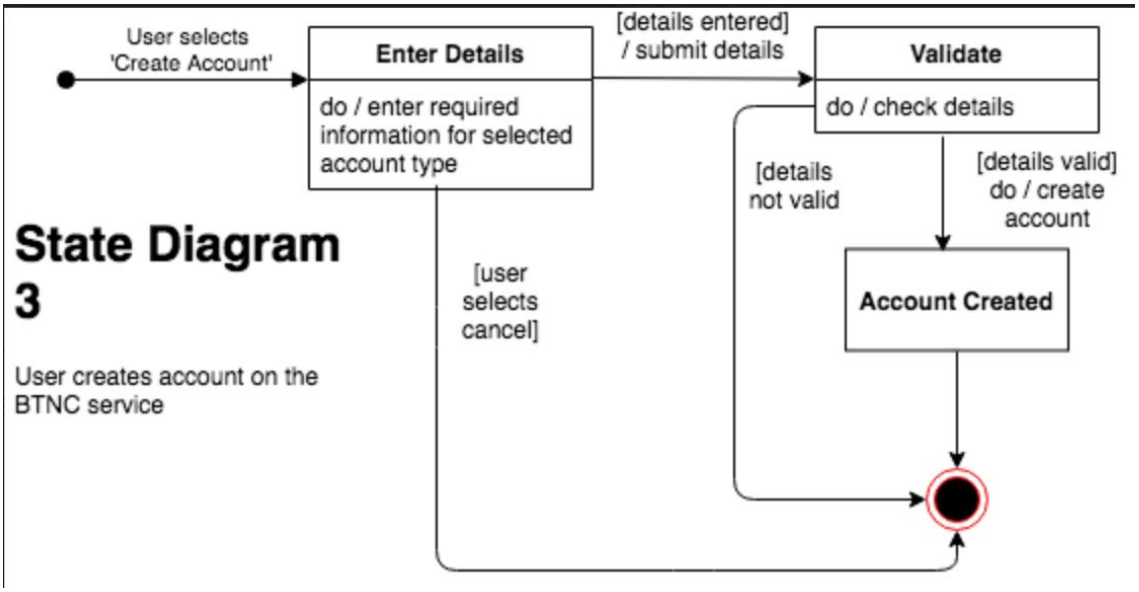
9.5.1 UML Class



9.5.2 UML State Diagrams



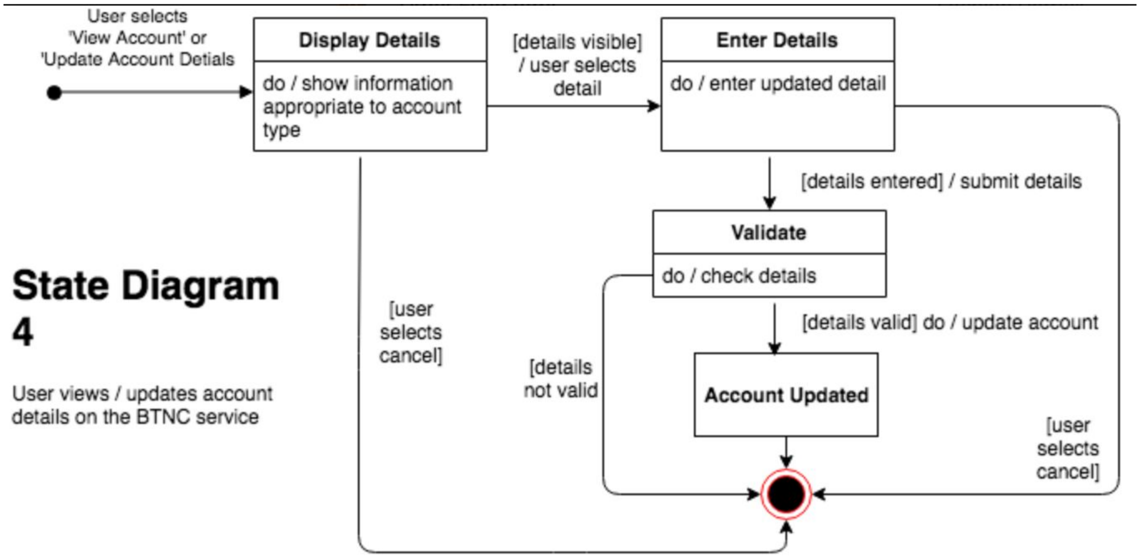




State Diagram 3

User creates account on the BTNC service

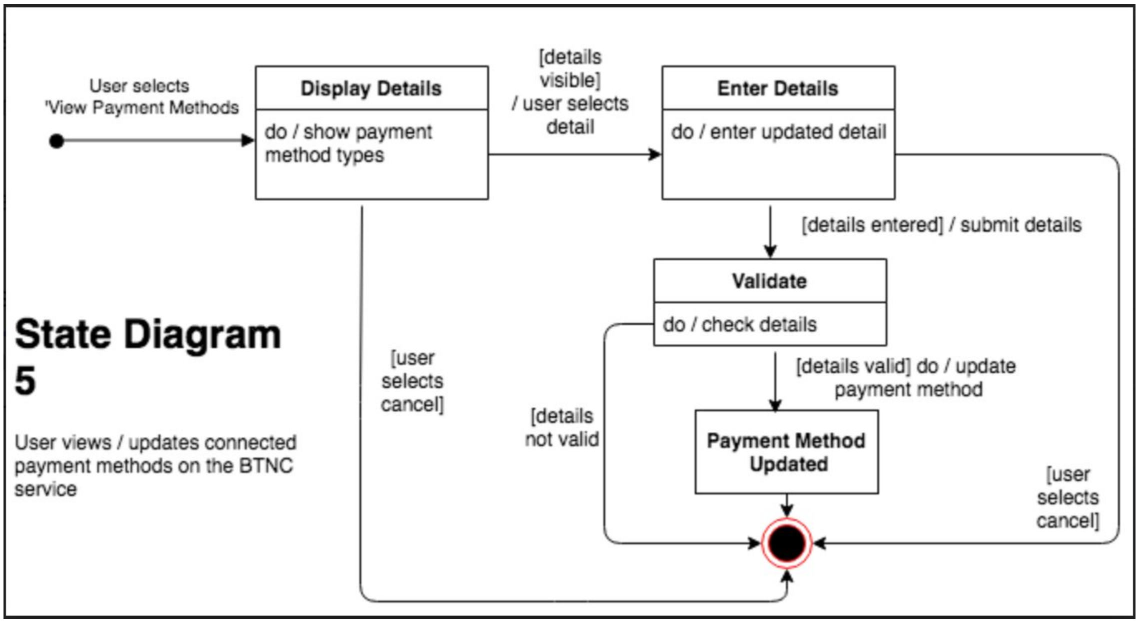
User selects 'Create Account'
Enter Details
do / enter required information for selected account type
[details entered] / submit details
Validate
do / check details
[details not valid
[details valid] do / create account
Account Created
[user selects cancel]



State Diagram 4

User views / updates account details on the BTNC service

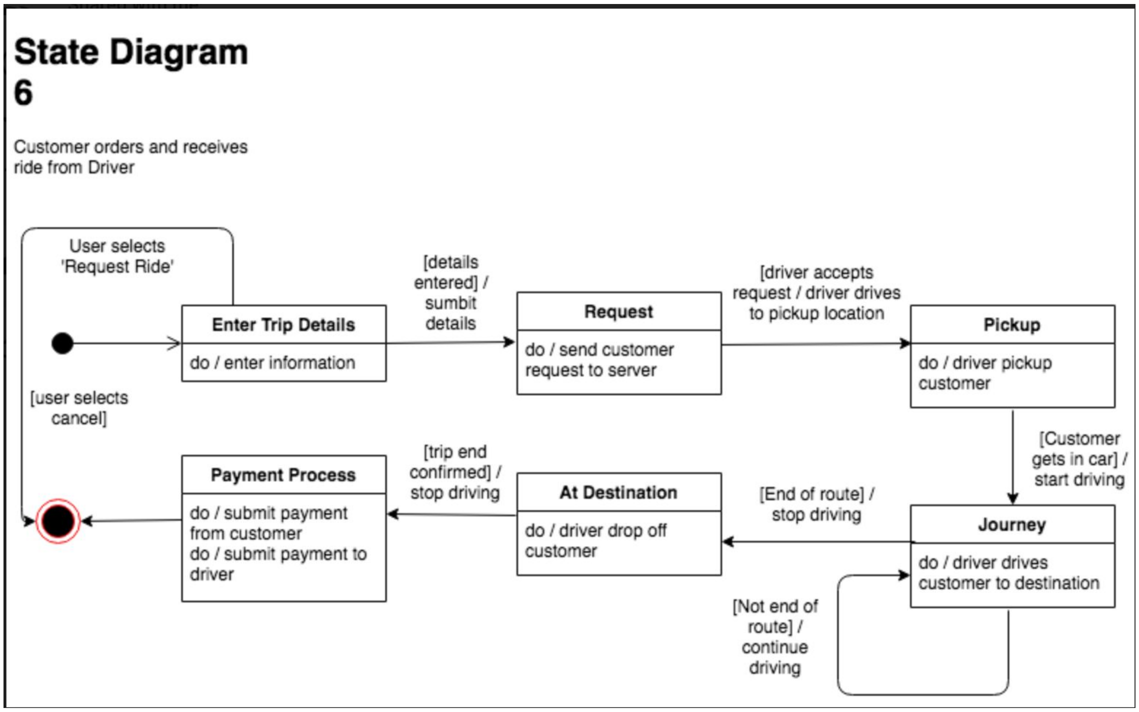
User selects 'View Account' or 'Update Account Detials
Display Details
do / show information appropriate to account type
[details visible] / user selects detail
Enter Details
do / enter updated detail
[details entered] / submit details
Validate
do / check details
[details valid] do / update account
Account Updated
[details not valid
[user selects cancel]
[user selects cancel]



State Diagram 5

User views / updates connected payment methods on the BTNC service

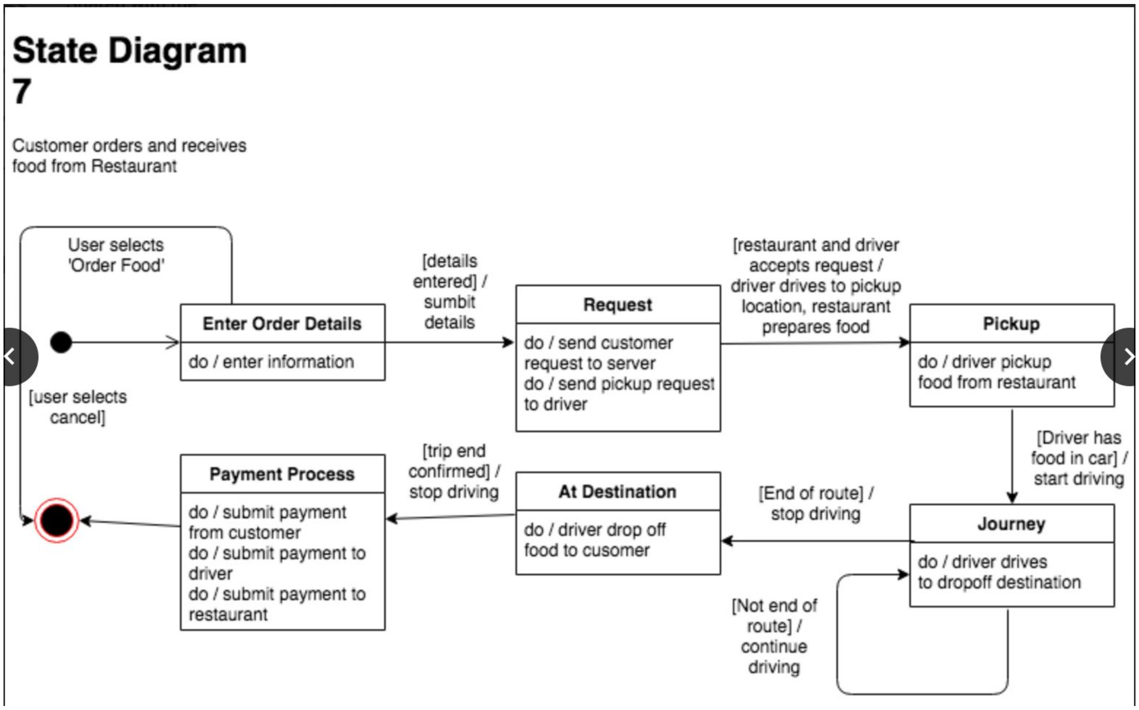
User selects
'View Payment Methods
Display Details
do / show payment method types
[details visible] / user selects detail
Enter Details
do / enter updated detail
[details entered] / submit details
Validate
do / check details
[details valid] do / update payment method
Payment Method Updated
[user selects cancel]
[details not valid
[user selects cancel]



State Diagram 6

Customer orders and receives ride from Driver

User selects
'Request Ride'
Enter Trip Details
do / enter information
[details entered] / sumbit details
Request
do / send customer request to server
[driver accepts request / driver drives to pickup location
Pickup
do / driver pickup customer
[Customer gets in car] / start driving
Journey
do / driver drives customer to destination
[Not end of route] / continue driving
[End of route] / stop driving
At Destination
do / driver drop off customer
[trip end confirmed] / stop driving
Payment Process
do / submit payment from customer
do / submit payment to driver
[user selects cancel]



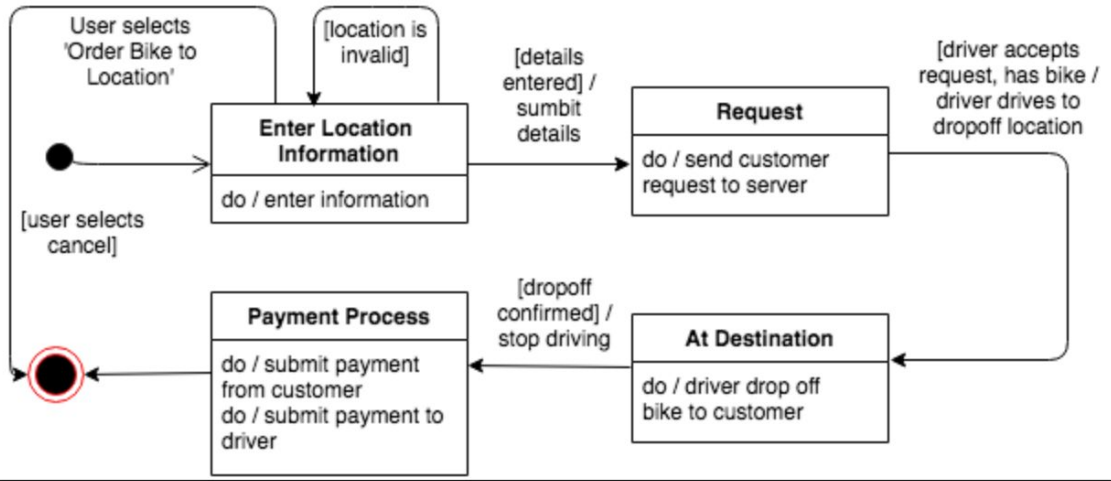
State Diagram 7

Customer orders and receives food from Restaurant

User selects 'Order Food'
[user selects cancel]
Enter Order Details
do / enter information
[details entered] / sumbit details
Request
do / send customer request to server
do / send pickup request to driver
[restaurant and driver accepts request / driver drives to pickup location, restaurant prepares food
Pickup
do / driver pickup food from restaurant
[Driver has food in car] / start driving
Journey
do / driver drives to dropoff destination
[Not end of route] / continue driving
[End of route] / stop driving
At Destination
do / driver drop off food to cusomer
[trip end confirmed] / stop driving
Payment Process
do / submit payment from customer
do / submit payment to driver
do / submit payment to restaurant

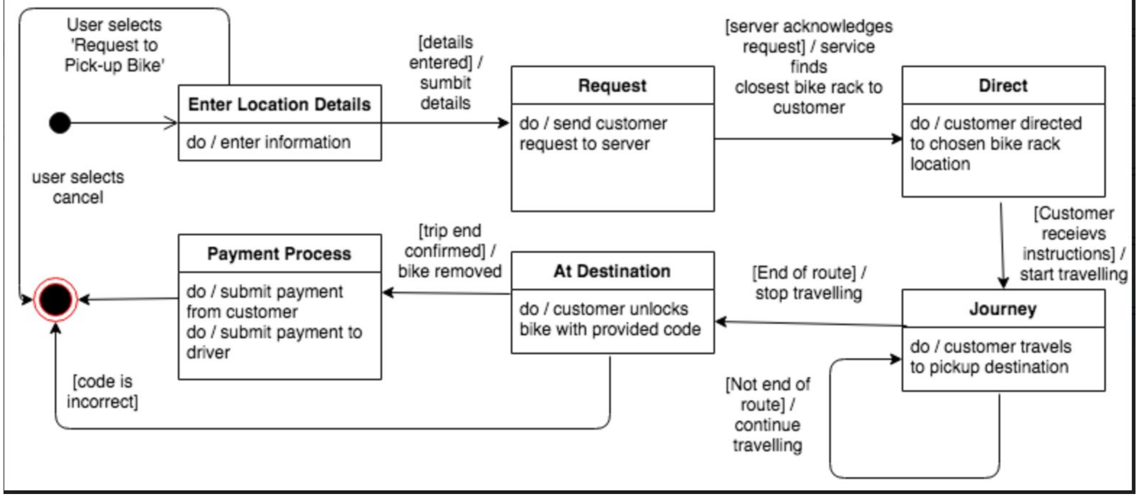
State Diagram 8

Customer orders bike to their location to be dropped off by driver


User selects 'Order Bike to Location'
[location is invalid]
Enter Location Information
do / enter information
[details entered] / sumbit details
Request
do / send customer request to server
[driver accepts request, has bike / driver drives to dropoff location
[user selects cancel]
Payment Process
do / submit payment from customer
do / submit payment to driver
[dropoff confirmed] / stop driving
At Destination
do / driver drop off bike to customer

State Diagram 9

Customer orders and picks up bike from bike rack


User selects 'Request to Pick-up Bike'
Enter Location Details
do / enter information
[details entered] / sumbit details
Request
do / send customer request to server
[server acknowledges request] / service finds closest bike rack to customer
Direct
do / customer directed to chosen bike rack location
user selects cancel
[Customer receievs instructions] / start travelling
Payment Process
do / submit payment from customer
do / submit payment to driver
[trip end confirmed] / bike removed
At Destination
do / customer unlocks bike with provided code
[End of route] / stop travelling
Journey
do / customer travels to pickup destination
[code is incorrect]
[Not end of route] / continue travelling

9.6 Appendix - Requirement Validation

9.6.1 Requirement Tracing

Requirement tracing was some to ensure the project had no failures and it met the expectations of the stakeholders. We then traced each requirement we gathered to the stakeholders to validate that we had both met all the stakeholders to the requirements and vice versa.

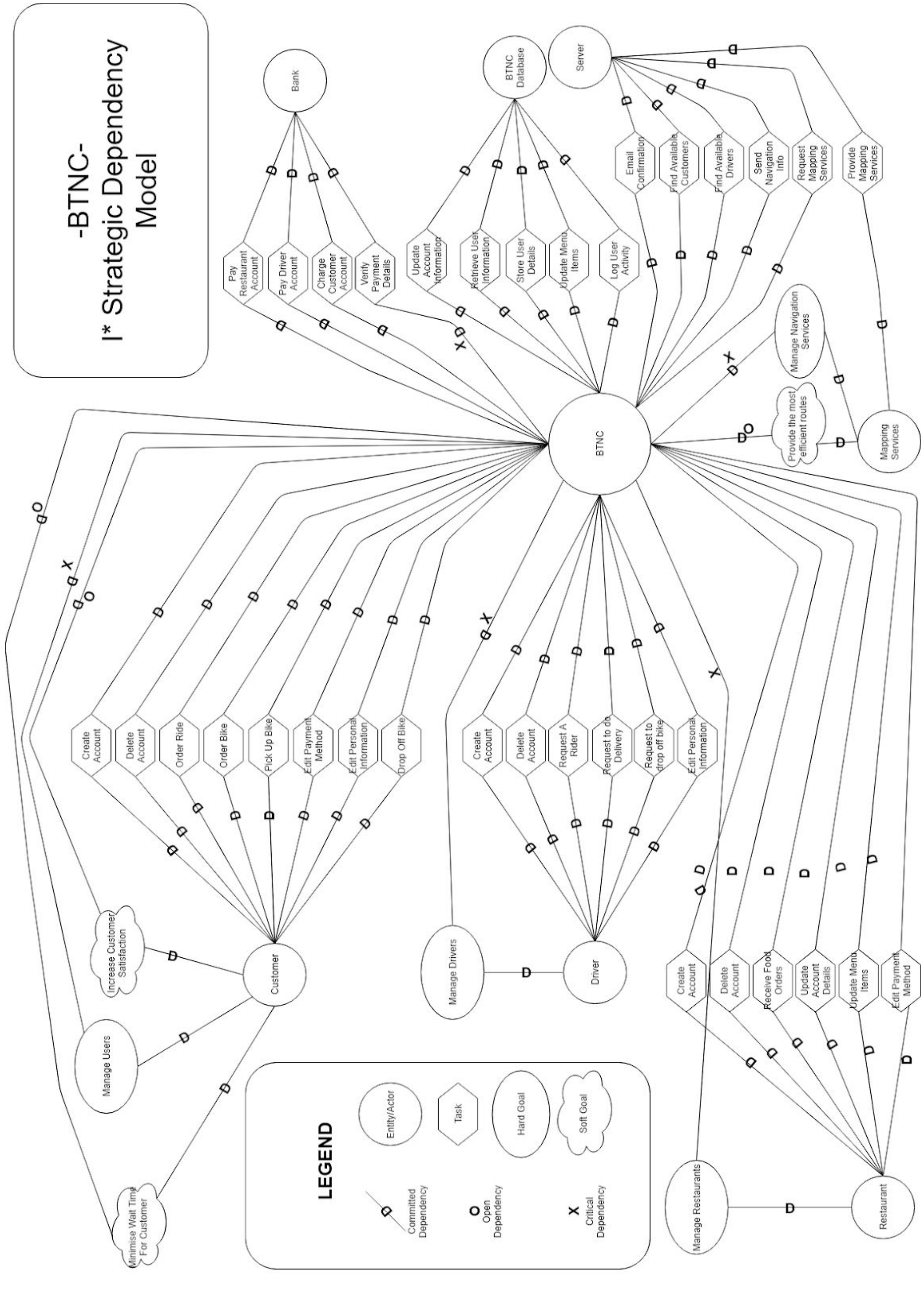
Require nt #	Requirement	Stakeholder Relation (where it comes from)
1	Thee application must be developed using version control so that code is maintainable and a timeline of the code history ca be seen and reverted to.	BTNC Developer Interview
2	Any application faults with code must be maintained and fixe with a software update within 24 hours of it occurring/report to the company.	Stakeholder meeting, BTNC developer and Managers
3	There must be support for the product 24/7 with an inhour te to aid any faults.	Stakeholder meeting BTNC Management
4	Database must undergo a continuous incremental backup wi daily full backup to ensure, at any moment there will be no lo of business or user data.	BTNC Developer BTNC Manager with the meeting
5	Database and its full backup must be stored in different state limit loss data if a natural disaster was to occur.	BTNC interview
6	Incremental backups must be stored in different buildings suc that if a building caught on fire, or something that wasn't expected no data will be lost.	BTNC Developer Interview
7	Database must be secured by lock and key and must ensure 24/7 surveillance to deter and stop theft or intentional damag	BTNC Stakeholder meeting
7	User account passwords must require capital letter, longer th 8 characters and contain 2 numbers. Ensure passwords are h to guess.	Application Specifications - From Stakeholder meeting
9	Data sent from device to server/database must be encrypted end to end using the DES standard to ensure that middle ma attacks and spying on packets can't occur.	BTNC Developer Interview

10	Payment and credit card details must be encrypted and store with users personal detail, however the 3/4 digit CVN numbe must never be stored or sent to the database. Must be entere every use of the card.	BTNC Stakeholder meeting
11	The application must require the user to login once every mo on their mobile phones.	BTNC Stakeholder meeting
12	User passwords must be changed once a year.	BTNC Stakeholder meeting
13	Security team must have employees hired to monitor attacks the database 24/7 to detect and recover any hacks or unauthorized access into the system.	BTNC Stakeholder meeting
14	The application must display a walk through visual helper on first login or creation of a new account in the application. This feature must also be accessible from settings menu inside the application.	Application Specifications
15	The application must be easy to use, with 90% of users being able to book a service after watching a walk through within 2 minutes of opening the application.	BTNC Stakeholder meeting
16	The application must work both on iOS and android devices.	Application Specifications
17	Create an account specifying which accounts they wish to create (driver, rider or restaurant) with their details	Application Specifications
18	Allow user to login into the app with the option of logging int their restaurant, driver or rider account.	Application specifications
19	View and change account details through the application	Application Specifications
20	Add a credit/debit card to pay for trips with the option to use apple pay or google pay (this must be a hard constraint on the system and you cant create an app without including card details)	Application Specifications
21	Order a ride from a specific location, whether its current or at specified location	Application Specifications
22	Order food from a chosen restaurant and have it delivered to current or specified location	Application Specifications

22	Lorem ipsum dolor sit amet, consectetur adipiscing elit. Proin sollicitudin lacus non tortor semper porta at vitae lorem. Fusce id accumsan nisl.	BTNC Stakeholder Meeting
23	Pick up a bike from a bike rack nearby, and once paid for the have the bike unlock from the rack	Application Specifications
24	Order a driver to drop off a bike to your current location	Application Specifications
25	View quotes and cost of trips or food delivery before the user follow through with a purchase	BTNC Stakeholder meeting
26	Show on a map the location of a driver when the driver has accepted a request with eta to the customer	BTNC Stakeholder meeting
27	Switch between the 3 different account modes within the app Driver, Rider and Restaurant	Uber Driver interview
28	Driver mode allows drivers to turn their status to waiting	Uber Driver interview
29	Allow driver to view income and past trips with all details and ratings/feedback	Uber Driver interview
30	When driver is in waiting mode, notify a driver if a customer request a rideshare service, OR a food delivery is available OR bike delivery	Uber Driver interview
31	Allow a driver to accept or reject a customer request (this mea the driver doesn’t want to do this request)	BTNC Stakeholder Meeting
32	Allow a driver to drop off a bike to a user's location	Application Specifications
33	Notify the users when a driver is near by their specified locati after a request has been made	Application Specifications
34	Charge customers pay method when they have requested a service	BTNC Manager Interview
35	Allow a restaurant account owner to open or close their store food delivery orders	Restaurant Owner: BTNC Interview
36	Allow restaurant owner to add or remove menu items	Application Specifications
37	Allow restaurant to change and view their public information	Application Specifications
38	Allow restaurant to have access to a drivers information so th can verify pickups	Restaurant Owner: BTNC Interview
39	Allow a user interface for restaurants to see how much they have sold	Restaurant Owner: BTNC Interview

40	Allow for site administrators access to data collected and maintain servers and user accounts if they contact with issues	BTNC Developer/Manager Interview
41	For drivers and restaurants, allow them to change bank detail for their income to be added to	Application Specification

9.6.2 i* Goal Modelling Diagram



9.6.3 Board Member Feedback

After Modelling the key stakeholders requirements, we proposed a consultation meeting with the BTNC board to discuss our models and to validate whether or not the requirements were met. An agenda was created to retrieve answers to validate our requirements:

- Does the system meet the requirements of the driver?
- Does the system meet the requirements of the user?
- Does the system meet the requirements of the restaurant owners?
- Does the system meet the requirements of the developers?
- Does the System meet the requirements of the management?

During the meeting we recorded feedback and how successful we were at determining the requirements. All of the feedback was positive with 100% of the proposed requirements being accepted by the stakeholders. The only concern the stakeholders had was with the security of the user data, which we then informed them that we added a non-functional requirement that asserts that the CVN credit card number will not be stored on our databases. Another problem was brought up about the SLA of our application with a requirement being added to have a guaranteed uptime of 98%, so users can rely on our services.

At a later date, we reviewed the recording of the consultation to identify any requirements that we may have missed. We reviewed and updated a non functional requirement to now guarantee DES encryption of the data being transferred between the database and the user.

9.8 Appendix - Software Requirements Specification

9.8.1 Introduction

9.8.1.1 Purpose

The following Software Requirements Specification outlines the requirements of the Beta Transport Network Company (BTNC) system which is to be used by our inhouse software engineering team for implementation. The intended design of the system allows users to order food to a location, request ride-sharing services and hire bicycles through one application, with these functionalities provided by each driver.

9.8.1.2Document Conventions

This document follows the recommended practices for Software Requirement Specifications as outlined in the IEEE Std 830 - 1998, created by the IEEE-SA Standards Board.

Term Definitions:

- The term ‘SRS’ is an abbreviation for Software Requirement Specifications.
- The term ‘BTNC’ is an abbreviation for ‘Beta Transportation Network Company.
- The term ‘user’ is used to refer to any individual that makes use of the system in some way.
- The term ‘customer’ refers specifically to those users that will either request and receive transportation, or order and receive food.
- The term ‘driver’ refers specifically to those users that perform the delivery of these services to customers.
- The term ‘app’ is used to refer to the mobile application that will allow users to engage and interact with the system.

9.8.1.3 Intended Audience

The intended audience of this SRS document is to outline the system design such that is relevant for any stakeholders involved in the implementation and life cycle of the project, at a technical, managerial or administrative level.

Reader Title	**Relevance to Project**
Developer	Responsible for the design and implementation of the project in accordance with the SRS
Project Manager	Managing the development of the project over the course of its life cycle
Focus Groups	Provides feedback on proposed functionality
Testers	Responsible for ensuring developed software meets the requirements and specifications of the project
Human Resources	Providing new and the best talent available to work on the project in the areas above whilst making sure that the talent have the skills or training needed to complete tasks.

9.8.1.4 Project Scope

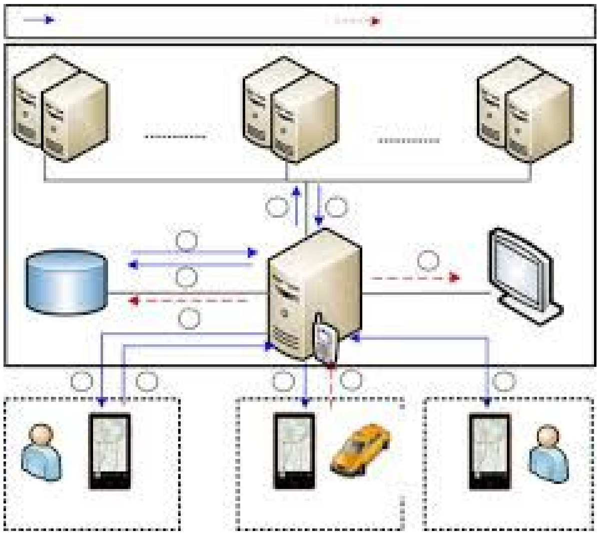
The scope this project involves the design and implementation of an application that allows customers to order or pickup a bike, order food to a location or order a ride-sharing service. Where customers can pick up pikes from any nearby bike rack if there are any, or they can order a bike to their location, which will be dropped off by a driver. This includes the scope of connections to interfaces defined in the external interface requirements section 3.1.

9.8.2 Overall Description

9.8.2.1 Product Perspective

The BTNC system is a combination system which seeks to combine and reinvent the way users order food, rideshare and hire bicycles, drivers to earn money through aiding these services and restaurant owners to sell food, with a new centralised app that combines the functionality into one. This product is similar to others out there, however it will have centralised architecture into one app that will allow versilitilty to the user if they wish to be both a driver or a rider. This system does not have any existing products and architecture to be implemented with, rather it will be build similar to current ride sharing applications (example architecture below), however since there are more stakeholders potentially involved there will more interfaces involved. Interfaces involved are covered.

As shown below in the image, the app will be hosted on a remote server, communicating with the database to store and retrieve customer information and ride information. Drivers, riders and restaurant owners will have access to their respective user interfaces through the one app after logging into the server when they open the app. The System admin will inherently have access to the server and database for maintenance.



9.8.2.2 Product Functions

BTNC must include the following features:

- Create an account specifying which accounts they wish to create (driver, rider or restaurant) with their details
- Allow user to login into the app with the option of logging into their restaurant, driver or rider account.
- View and change account details through the application
- Add a credit/debit card to pay for trips with the option to use apple pay or google pay (this must be a hard constraint on the system and you cant create an app without including card details)
- Order a ride from a specific location, whether its current or at a specified location
- Order food from a chosen restaurant and have it delivered to a current or specified location
- Pick up a bike from a bike rack nearby, and once paid for the hire, have the bike unlock from the rack
- Order a driver to drop off a bike to your current location
- View quotes and cost of trips or food delivery before the user follow through with a purchase
- Show on a map the location of a driver when the driver has accepted a request with eta to the customer
- Switch between the 3 different account modes within the app, Driver, Rider and Restaurant
- Driver mode allows drivers to turn their status to waiting
- Allow driver to view income and past trips with all details and ratings/feedback
- When driver is in waiting mode, notify a driver if a customer request a rideshare service, OR a food delivery is available OR bike delivery
- Allow a driver to accept or reject a customer request (this means the driver doesn't want to do this request)
- Allow a driver to drop off a bike to a user's location
- Notify the users when a driver is near by their specified location after a request has been made
- Charge customers pay method when they have requested a service
- Allow a restaurant account owner to open or close their store for food delivery orders
- Allow restaurant owner to add or remove menu items
- Allow restaurant to change and view their public information
- Allow restaurant to have access to a drivers information so they can verify pickups
- Allow a user interface for restaurants to see how much they have sold
- Allow for site administrators access to data collected and maintain servers and user accounts if they contact with issues

- For drivers and restaurants, allow them to change bank details for their income to be added to

9.8.2.3 User Classes and Characteristics

Customers	Customers will on an average case use this application multiple times a week, with their only access through the app. Customers are expected to be of all ages, with people under 16 require the accompaniment of someone over 18, with all genders expected to use equally. All customers will have regular patterns of travel allowing the app and push notification to be tailored to the users needs week by week. After loading of the app, the customer can order food, rideshare or a bike with in 3 interactions at minimum. The main features the customer can interact with is ordering a request of service (food,ride, bike), review quotes, review and change their details (including payment) and contact for customer support from the app.
Drivers	Drivers will be of the age of 20 + as you must have a full drivers licence to register with the expected use of the system is on a daily basis for their main source or side income. Gender will have no influence on the interaction with the system. Interactions are kept to a minimal, within 2 interactions of the system the driver can register themselves as waiting such that they can take requests. The main features the driver can interact with is accepting requests, review rides, review and change their details (including payment),contact for customer support from the app and see their income in a stats page..
Restaurant Owner	The restaurant owners will have an average use case of everyday their stores are open for business. The main

	usage of this application will maintain their orderable menu and their contact details, with the ability within a few interactions to request and change payment details for their orders.

9.8.2.4 Operating Environment

- The BTNC application will be able to run on all iOS and android operating systems only, with backward compatibility available to most OS for both platforms.
- The payment system will be mainly through paypal with users also having the ability to use apple pay and google pay. It will also have direct debit from credit/debit cards.
- All details and information regarding all parties involved will be stored in a secure, monitored database that will remain in a secure location in the inhouse offices of BTNC.
- The servers to process traffic will completed through an outsource company.

9.8.2.5 Design and Implementation Constraints

- Server and database must be able to handle high levels of parallel traffic for the location updates and ordering. This will have a constraint on the way it is designed and managed in the backend of the application.
- The applications language will cater for english and a few other international languages, however it will limit a constraint on the user.
- All backend will be completed in best practice and standards of the development for AWS (back end) and Kotlin for front end of the application.
- Data must be encrypted with personal data being stored in parallel of updates from any parties using the application.
- Application must maintain a 98% SLA with the developers and maintainers.
- Any communication with the database and server must be fully encrypted with the latest standard.

- User data should be stored in a database local to the state its relevance for the most time.

9.8.2.6 User Documentation

- An online tutorial after first install will run to help user get to know the user interface and is available inside setting app to toggle on if the user accidently missed it.
- Inside settings there will be a tab to allow all parties to read up about their specific rights and privacy concerns in regards to use of the application.
- There will be a FAQ section inside settings to allow users to read up about possible problems they may have
- At a last resort there is a contact number, email and live chat for any party of the application to contact for support in regards to the app.

9.8.2.7 Assumptions and Dependencies

- Due to the reliance of Paypal and commonwealth bank for purchases, it is assumed that those services will have almost 100% uptime as it is needed to complete any customer requests, charge them and distribute the payments to drivers or restaurants.
- It is assumed that the database will be distributed across australia such that users that use the app will have access to the data within 2 seconds of request. This will rely on AWS to distribute the data across the split databases every month.
- All parties are assumed to have devices that can run the app.
- When a driver starts a ride, it will open an interacted google maps inside the app to start directions, this assumes the uptime of google maps to be a 98% SLA at no cost for integration.
- All external help for the user relies on a customer support team which will be available monday to friday and is to be organised and ran by HR of BTNC.

9.8.3 Specific Requirements

9.8.3.1 External Interface Requirements

9.8.3.1.1 User Interfaces

Mobile Application

- Information will be displayed on a mobile application for the user to be able to input to the application with their fingers with ease of use. The aim of the UI is to make it clean and simple such that there will be no need for continuous customer training.
- The UI will display a series of help directions on first login to show where each of the menus are and how to order the services.
- The user interface will allow ease of access to all the services for any of the customers.
- The interface will be interchangeable depending on user options and the type of customer they are. (Driver, ride, restaurant)

9.8.3.1.2 Software Interfaces

Database

- This interface will allow users to input their data, and for it to be stored using SQL in a database
- The software interface will allow users to access and edit their data that is stored in an SQL database through the user interface.
- Allow users to input and store securely their accounts into a secure database.

Android/iOS (Mobile)

- The software interface of the application will meet the user interface on both android and iOS software interfaces.

Administration

- Allow the software interfaces to be maintained using internal resources
- Allow administrators to access and modify the database without accessing directly to the secure payment details of the user

Banking

- Allow the application to utilize any shortcuts to access banking mediums through the other applications on their phones ie Paypal, Apple Pay

Map

- Allow the applications software interface connect with installed map applications, so you can utilize maps that the user trusts.
- Redirect the app to and from the map app and use the maps API to use it inside BTNC app. Allow the user to choose.

9.8.3.1.3 Hardware Interfaces

Application

- Make use, request use, of the physical capabilities of the users phone, connect to WiFi, mobile data and bluetooth.
- Allow use to the users phone’s storage medium to store both the application itself and it’ data and security information.

Administration

- Allow admin to be able to physically connect their computer to the server and database for maintainability.

9.8.3.1.4 Communication Interfaces

- Server and application will send text messages to the account on file, of any impacts or updates on their services using free of charge text messages.
- Communication interface will send out push notifications to the users phones that have the app installed of any major sale or event to do with marketing of the application.
- Application will allow push notifications to be turned on or off using the UI.
- Application will allow notifications to the user if any of their banking details are expiring or have expired, allowing them to update them using the UI.
- Communication between hardware and internet will be done through TCP/IP and HTTPS communication interfaces, along with push notification communication.

9.8.3.1.5 Banking interface

- The application will allow, using the UI, users to add a payment method, and for the application to contact the required banking interface using the apps internet connection to verify account details.
- The application will be able to charge the registered account details using the connected banking interface to the account.

9.8.4.2 System Features

Below is a list of system features, for each of the following a use case and sequence diagrams can be found at Appendix 9.4:

- Create user account
- Delete account
- Login
- View account details
- Update account details
- View payment methods
- Update payment methods
- Add payment method
- Order ride
- Order food
- Order bike
- Pick up bike

- Pickup passenger/start trip
- Pick up food/ drop off food
- Create restaurant account
- Update menu item
- Add menu item
- Logout
- Create driver account
- Payment processing

9.8.4.3 Software System Attributes (Non-Functional Requirements)

9.8.4.3.1 Performance Requirement

The below NFR relates to the performance expected in an efficient and reliable way to ensure users receive a high performance of the application:

- NFR: The application must load within 3 seconds on any android or iOS phone for 90% of users. (some users may have older OS, so may have to use older APIs)
- NFR: The login screen must be displayed for users who aren't logged in already within 2 seconds of the application loading.
- NFR: The application must determine the users location and display it on the opening may screen within 3 seconds of the app loading. (assumed use is logged in).
- NFR: The application must notify service receivers (driver's/restaurants) within 5 seconds of a customer ordering a specific services.
- NFR: The application must be available and usable 98% of the time, with down time of the application being no greater and once a month.

9.8.4.3.2 Maintainability Requirement

The below NFR relates to the maintainability of the application expected for software engineers:

- NFR: The application must be developed using version control so that code is maintainable and a timeline of the code history can be seen and reverted to.

- NFR: Any application faults with code must be maintained and fixed with a software update within 24 hours of it occurring/reported to the company.

9.8.4.3.3 Safety Requirement

The below NFR relates to the reliability and measures to ensure user and business data is safely stored:

- NFR: Database must undergo a continuous incremental backup with a daily full backup to ensure, at any moment there will be no loss of business or user data.
- NFR: Database and its full backup must be stored in different states to limit loss data if a natural disaster was to occur.
- NFR: Incremental backups must be stored in different buildings such that if a building caught on fire, or something that wasn't expected no data will be lost.

9.8.3.4 Security Requirement

The below NFR relates to the security, confidentiality and authentication of user data and business data:

- NFR: Database must be secured by lock and key and must ensure 24/7 surveillance to deter and stop theft or intentional damage.
- NFR: User account passwords must require capital letter, longer than 8 characters and contain 2 numbers. Ensure passwords are hard to guess.
- NFR: Data sent from device to server/database must be encrypted end to end using the DES standard to ensure that middle man attacks and spying on packets can’t occur.
- NFR: Payment and credit card details must be encrypted and stored with users personal detail, however the 3/4 digit CVN number must never be stored or sent to the database. Must be entered every use of the card.
- NFR: The application must require the user to login once every month on their mobile phones.
- NFR: User passwords must be changed once a year.
- NFR: Security team must have employees hired to monitor attacks on the database 24/7 to detect and recover any hacks or unauthorized access into the system.

9.8.3.5 Software Quality Requirement

The below NFR relates to the usability, reliability and quality of the application, to ensure that the application is easy to use and understand from a user level:

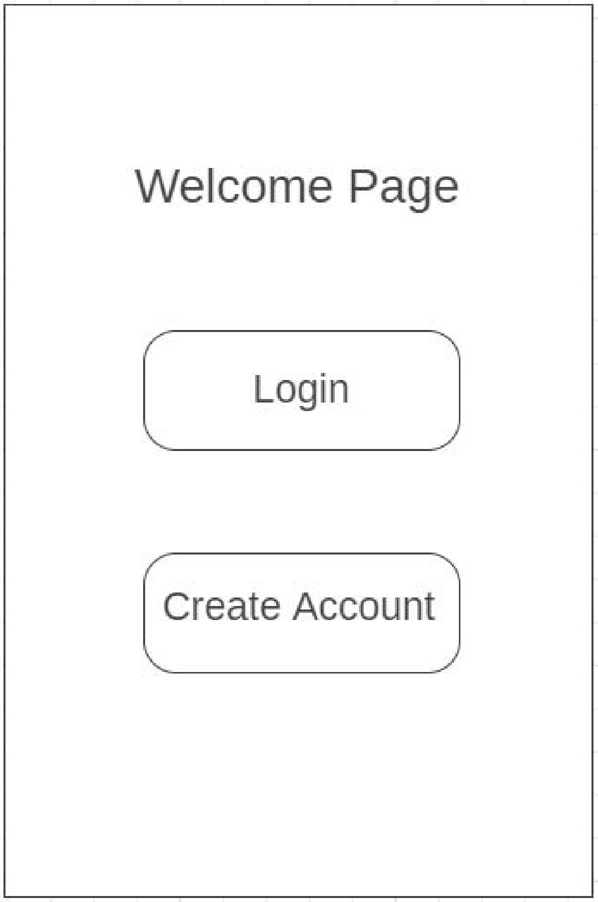
- NFR: The application must display a walk through visual helper on the first login or creation of a new account in the application. This feature must also be accessible from settings menu inside the application.

- NFR: The website must be easy to use, with 90% of users being able to book a service after watching a walk through within 2 minutes of opening the application.
- NFR: The application must work both on iOS and android devices.

9.9 Wireframe Prototypes

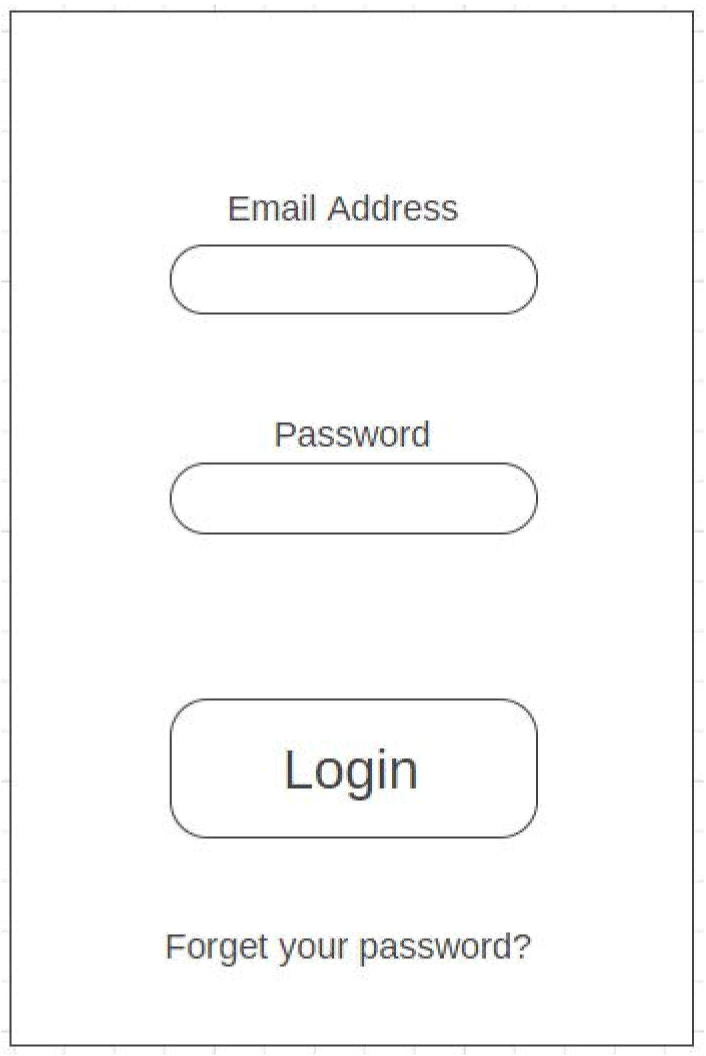
Welcome Page

- Allows user to either login or create an account if they’re a new user.
- Create account will take you to the create account page
- Login will take you to the login page



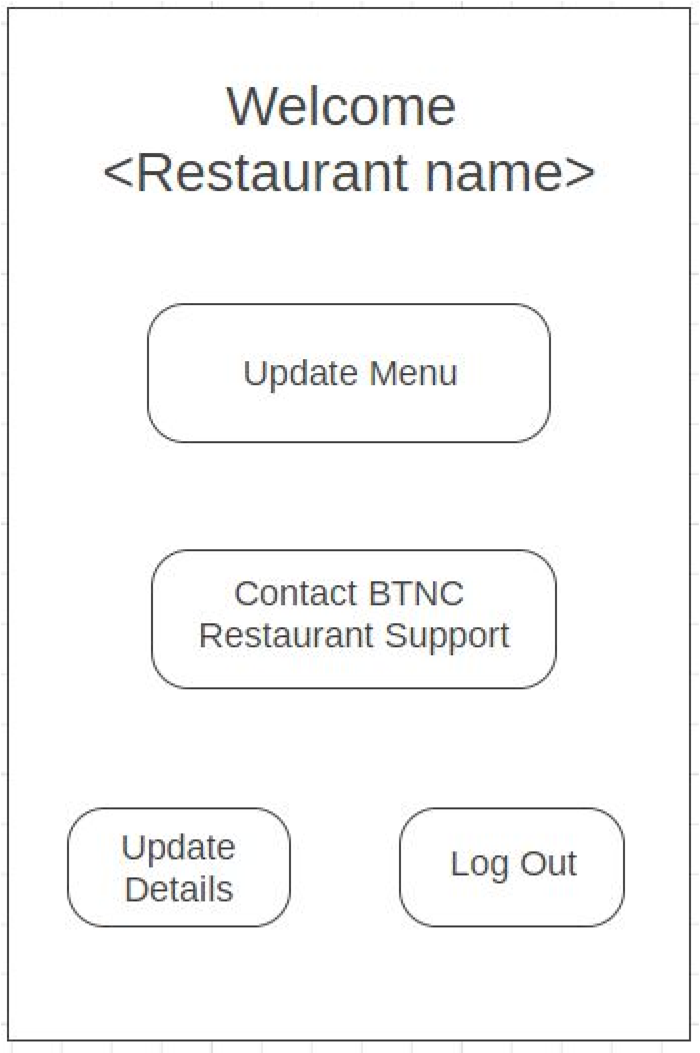
Login Screen

- Allow a user to enter their email address and password so that they can then login into their account and order a service.
- Forget your password button will send an email for the user to reset their application account password



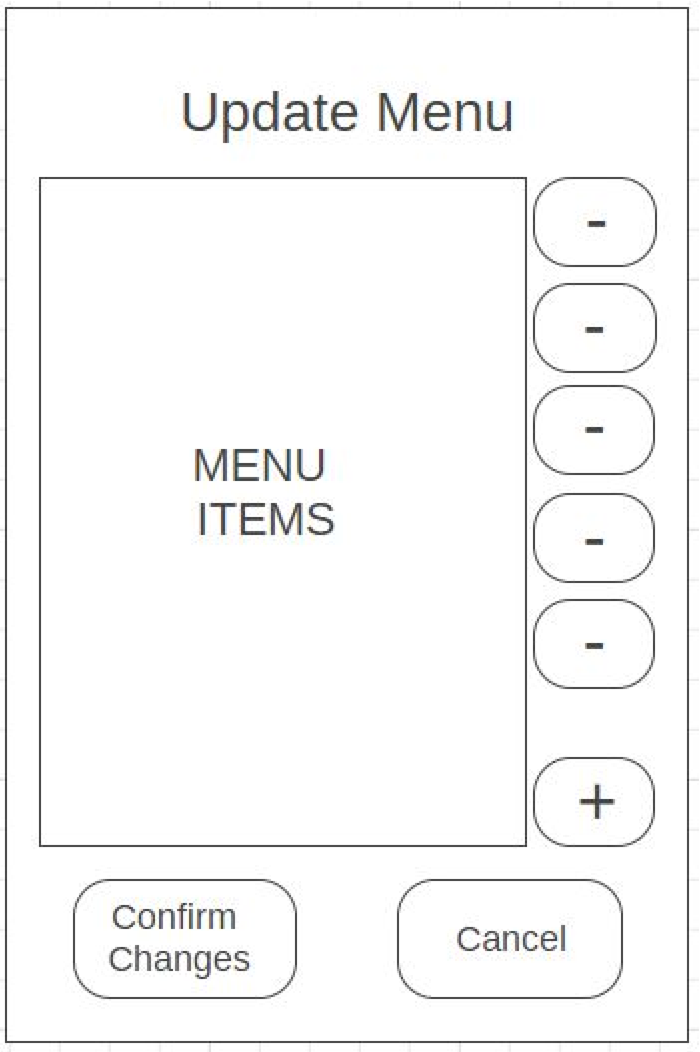
Welcome screen for restaurants

- Update the menu takes you to the update menu screen, allows them to update the menu that is online for sale
- Contact support page will call a sale rep who can assist in questions
- Update details will take you to the update details page for the restaurant account
- Log out button will log you out of the screen



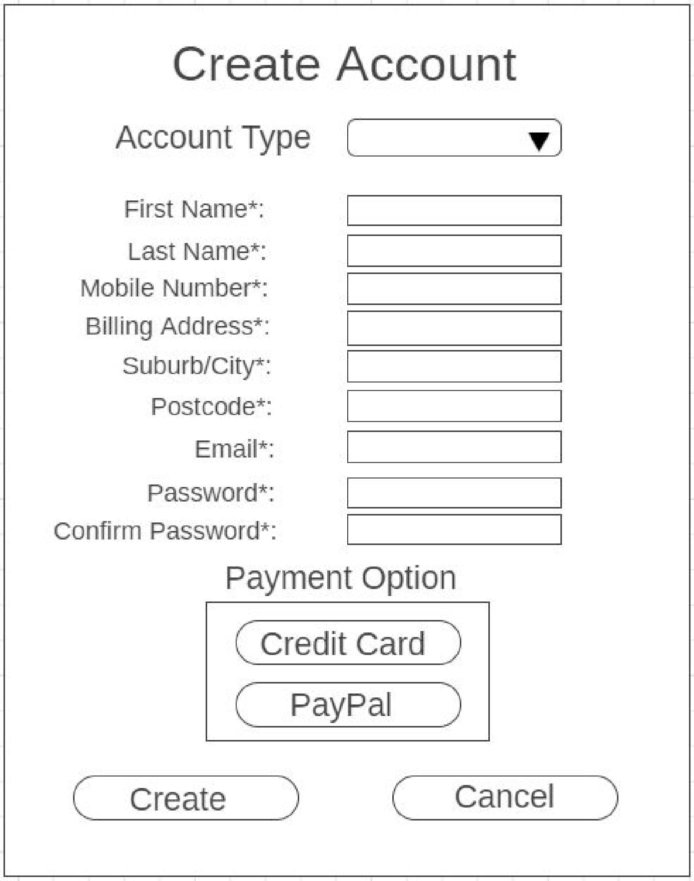
Update Menu

- Allows restaurant to add or take menu items from the online store
- Confirm changes will update the database of the changes and all people using the application will be pushed an updated menu
- Cancel will take them back to the restaurant page



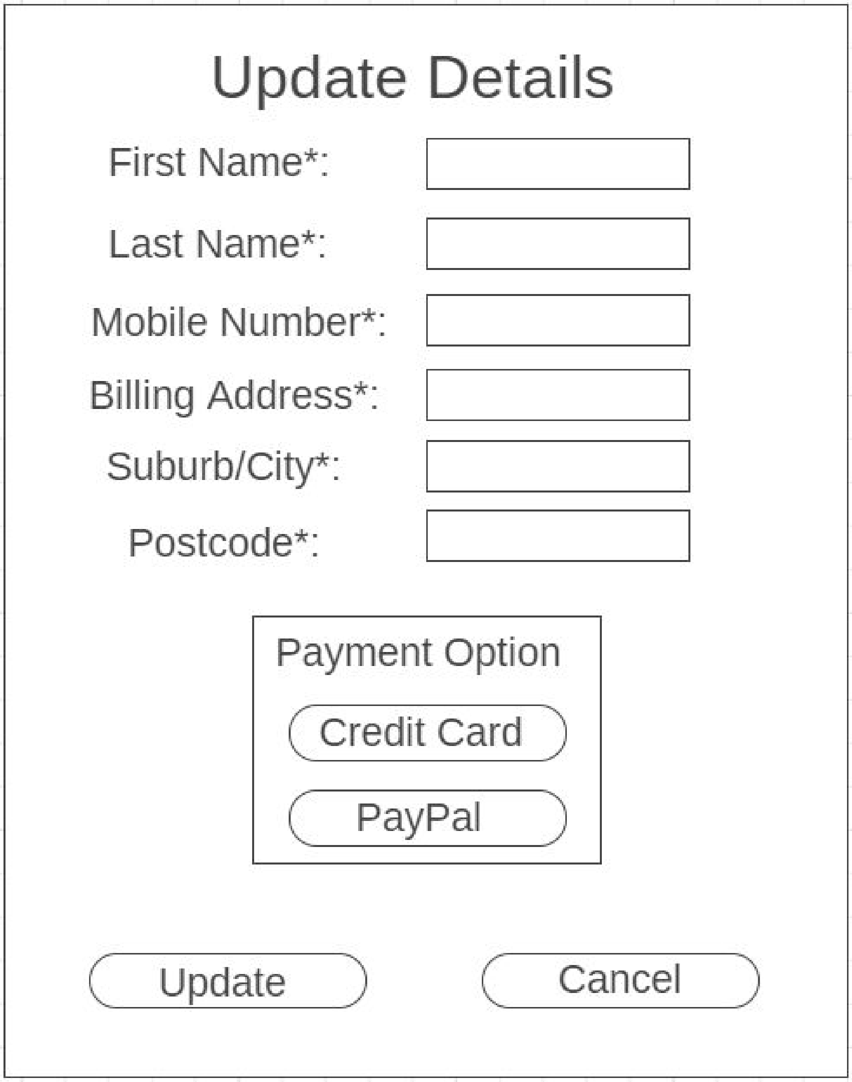
Create account

- Allow user to select the account type for creation which will change the details needed to be entered
- Allow users to enter details
- Allow users to input a payment option
- Credit card will opt users to enter the needed fields
- Paypal will redirect to external login
- Create will send request to database and server to validate details
- Cancel will take them to login screen



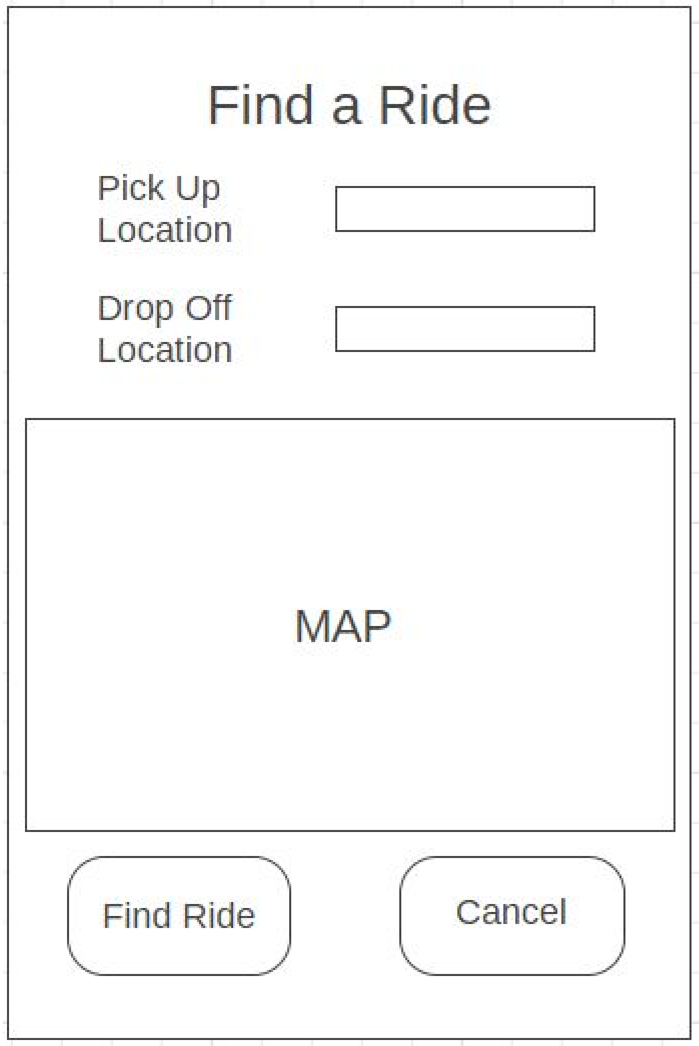
Update details

- Allows user to enter specific details they wish to update
- Allows user to update their credit card and paypal details, which like before redirect to respective sites.
- Update will confirm details are valid and account exists to update



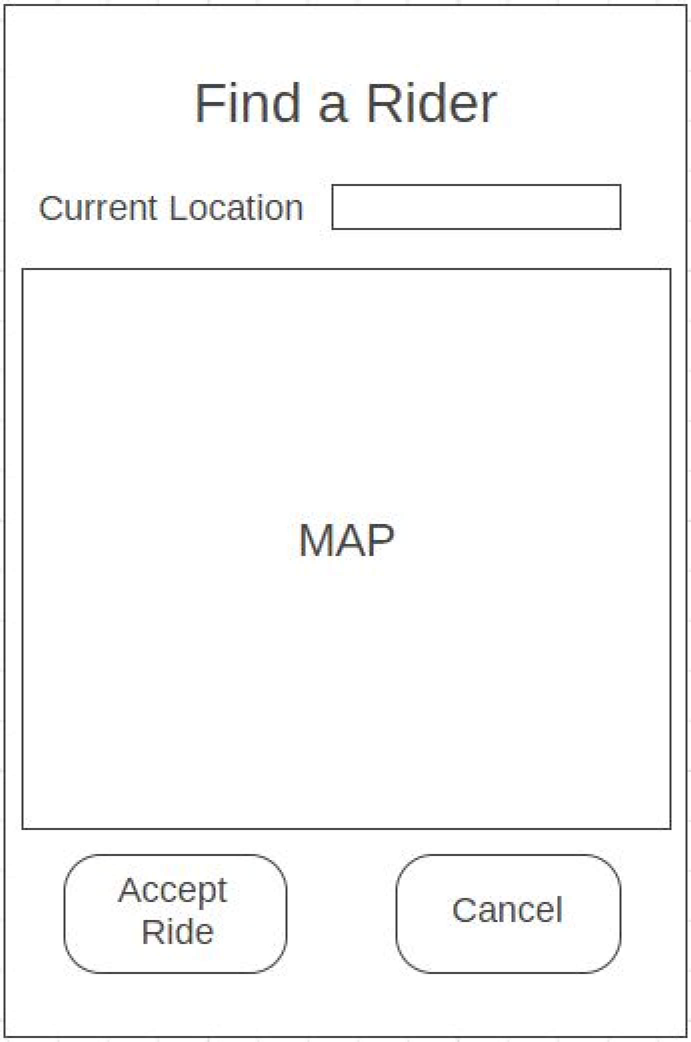
Find a ride

- Allows users to make requests for a ride.
- Shows users nearby drivers on a visual map interface.
- Allows users to specify a pick up location.
- Allows users to specify a drop off location.
- Find a ride will submit the user's entered pickup and drop off location details along with their request for a ride to the server and database for validation.
- Cancel will take them to login screen



Find a rider

- Allows driver to have one site to find a rider
- Shows the drivers current location
- Shows on a map the service that wants to be completed
- Allows driver to accept the ride, which will redirect to direction in the map section
- Cancel will cause the rider to go offline and go back to the home screen



9.10 Appendix - Group Meeting Minutes

9.10.1 Meeting 1

Beta Transport Network Company Meeting Minutes (Internal Meeting)

Date: Sunday 25th September 2018
Time: 12:00pm-1:00pm
Location: Building 314, Curtin University
Meeting Type: Assignment Specification Discussion
Attendees: Dakota Epton, Simeon Leatherland, Lachlan Derrick

Agenda Items:

- Select problem statement to be undertake
  - Problem statement 2 (Beta Transportation Network Company) chosen
- Allocate sections for initial groundwork
  - Lachlan: Stakeholders
  - Simeon: Scoping & Document Formatting
  - Dakota: Requirements elicitation

Action Items:

- Work on sections individually, combine to construct a forumlative understanding of the Problem Domain from which to proceed.
- Document formatting and the structure
- Discussed specification details
- As a group went through Uber, Uber Eats and JumpBike applications and noted requirements
- Decided on a SRS template to use

9.10.1 Meeting 2

Beta Transport Network Company Meeting Minutes (Internal Meeting)

Date: Wednesday 3rd October 2018
Time: 11:00am-12:00pm
Location: Building 314, Curtin University
Meeting Type: Assignment Planning and SRS Discussion
Attendees: Dakota Epton, Simeon Leatherland, Lachlan Derrick

Agenda Items:

- Determining the end goal of the product
  - Provide a service for customer that allows them to:
    - order a ride, order a bike drop off, order food delivery and also pick up a bike
  - Combine the functionality of Uber Eats, Uber and JumpBike into one Application
- Introduce Trello and card system for Planning and Management
- Complete Introduction for our SRS

Action Items:

- Create questionnaires
- Finalised end goal of product
- Plan elicitation interviews
- Populate trello with tasks and delegate

9.10.1 Meeting 3

Beta Transport Network Company Meeting Minutes (Internal Meeting)

Date: Wednesday 10th October 2018
Time: 12:00am-2:00pm
Location: Building 314, Curtin University
Meeting Type: Report Discussion
Attendees: Dakota Epton, Simeon Leatherland, Lachlan Derrick

Agenda Items:

- Discuss diagram creation and how to move ahead with it
- Create the report and decide on format
- Move ahead with elicitation

Action Items:

- Created report
- Brainstormed elicitation interview

9.10.1 Meeting 4

Beta Transport Network Company Meeting Minutes (Internal Meeting)

Date: Sunday 14th October 2018
Time: 12:00am-2:00pm
Location: Building 314, Curtin University
Meeting Type: Finalise Information
Attendees: Dakota Epton, Simeon Leatherland, Lachlan Derrick

Topics Discussed:

- **Report Components:**
  - Analysis Process
  - Difficulties Encountered
  - Appendix
  - Conclusion
- **System Requirements:**
  - Functional
  - Non-Functional
- **Diagrams:**
  - UML
  - State Diagram
  - Sequence Diagram
  - Use Case Diagram
  - i* Diagram

Action Items:

- Requirements finalised
- Use Case diagrams completed
- Sequence diagrams completed

9.11 References

Weigers, Karl. 2002. *Software Requirements Specification Template. https://lms.curtin.edu.au/bbcswebdav/pid-2321522-dt-content-rid-21346606_1/xid-21346606_1*

Dr Computing, 2014, Architecture Cloud. https://www.computer.org/csdl/trans/tk/2015/07/06847170.pdf)

The Institute of Electrical and Electronics Engineers, 1998

IEEE Recommended Practice for Software Requirements Specifications