

Specs & Design

Assignment Number	1
Version	01
Print Date	16/5/2021
Page	Page 1 of 104

Software Engineering project

[PICK ME UP]



Assignment Number	1
Version	01
Print Date	16/5/2021
Page	Page 2 of 104

Table of Content

1. Customer Statement of Requirements	Pg 3
2. Glossary of Terms	Pg 4
3. System Requirements	Pg 5
4. Functional Requirements Specification	Pg 11
5. Effort Estimation using Case Points	Pg 30
6. Domain Analysis	Pg 33
7. Interaction Diagrams	Pg 44
8. Class Diagram and Interface Specification	Pg 50
9. System Architecture and System Design	Pg 56
10. Algorithms and Data Structures	Pg 61
11. User Interface Design and Implementation	Pg 63
12. Design of Tests	Pg 95
13. History of Work	Pg 100
14. References	Pg 104

Assignment Number	1
Version	01
Print Date	16/5/2021
Page	Page 3 of 104

1. Customer Statement of Requirements

Ingrid - UB student

“My concerns have always been safety. The idea of holding taxi drivers responsible or linked to whom they are picking up or dropping off is a great idea; it provides accountability in case of a bad situation. My understanding of the application is that all the drivers have to provide valid credentials in order to be accepted as a pick me up driver; this provides comfort knowing that it's not a complete stranger who will pick me up ”.

Anivar - Visiting Tourist

“My concerns are knowing who to trust. The rating system would ensure that these drivers provide a level of expertise where they will be ranked accordingly. The centralized database along with its ratings will ensure that people have a better idea who to pick and why for future trips ”.

Bernard - GOB Employee

“As a former UB student and now a GOB employee, my concern has always been having an application that ensures the availability of the taxi operator. Since the pick me up app will show if the driver is indeed available as he/she claims to be will ensure customer satisfaction since being left waiting on a false promise of “am close by, I can reach in 10 minutes” and never showing is nothing nice!

Assignment Number	1
Version	01
Print Date	16/5/2021
Page	Page 4 of 104

2. Glossary of Terms

Client – anyone who would wants service from the taxi (taxi's clients)

GOB – acronym for Government of Belize

Live Tracking – real-time/current location of the Taxis

Taxi States – the current state of the driver, if driver is available for accepting requests, if the driver is busy or is carrying a client(s), or the driver is offline, the driver cannot/isn't available for requests (could be breakdowns, lunch, etc.)

Drop-off Location – the client's destination.

Map – a map of the current town/city of the Driver and the Client

Assignment Number	1
Version	01
Print Date	16/5/2021
Page	Page 5 of 104

3. System Requirements

Priority Weight	Description
5	Very Important
4	Important
3	Neutral
2	Low importance
1	No Importance

Functional Requirement

Identifier	Priority level (1-5)	Requirement
REQ1	1	The System Shall Show Available Taxis on the Map.
REQ2	1	The System Shall Live Track All Taxis.
REQ3	4	The System Shall Show the Shortest route from the Current Position to the Pick-up Location or Drop-off Location for the Taxi Driver.
REQ4	3	The System Shall Allow Administrators to Manage Taxis Driver Accounts.
REQ5	3	The System Shall Allow Administrators, clients and Drivers to login
REQ6	5	The System Shall Allow a Taxi Driver to Accept Pick-up Requests.
REQ7	4	The System Shall List Clients Requesting Pick-up to the Taxi Drivers.
REQ8	2	The System Shall Allow Drives to switch states: Available, Busy and offline.
REQ9	4	The System Shall Allow Client Pick-up Request to be Canceled.
REQ10	4	The System Shall Allow Clients to Pick a Drop-off Location, and Read their Current Location when Doing a Request.

Assignment Number	1
Version	01
Print Date	16/5/2021
Page	Page 6 of 104

Comments:

When reviewing the requirements of the application, the priority of Administrators adding/deleting drivers wasn't a high priority to the functionality of the software. So requirement 4 & priority was lowered to a 3 (Netural).

Enumerated NonFunctional Requirement

Identifier	Priority level	Requirement
REQ11	5	The system shall be at least 90% of the time up. (Performance)
REQ12	5	The system shall automatically provide the location of the client (pick-up location) when doing a request. (Functionality)
REQ13	2	The system shall provide the option to rate the driver. (Functionality)
REQ14	1	The system shall show all active drivers on the map when the client requests, saving the data of the client. (Performance)
REQ15	2	The system shall have a simplistic user interface design. (Usability)
REQ15	1	Drivers which are inactive or busy shall not be shown on the map. (Functionality)
REQ16	2	The system shall be supported on any Operating System. (Supportability)
REQ17	5	The system shall prevent 2 or more drivers from accepting the same client. (Functionality)
REQ18	4	The system shall notify the taxi drivers of new requests within 5 seconds (dependent on the data/internet speed). (Performance)
REQ19	2	The system shall request GPS to be turned on when opening app (Functionality)
REQ20	4	The system shall require Administrators and Drivers to login to modify the system (Security)

Specs & Design

Assignment Number	1
Version	01
Print Date	16/5/2021
Page	Page 7 of 104

REQ21	3	The system shall require clients wanting to request to sign in using his/her google account (Security).
REQ22	4	All data shall be encrypted for the safety of user data (Security).

Comments:

The NonFunctional requirement touches on all key importance of a software but security. To correct this, 3 security functions were added, highlighting the importance of Login credentials to change data and the importance of clients being logged into a valid google account to navigate the application. User data will also be encrypted for additional security.

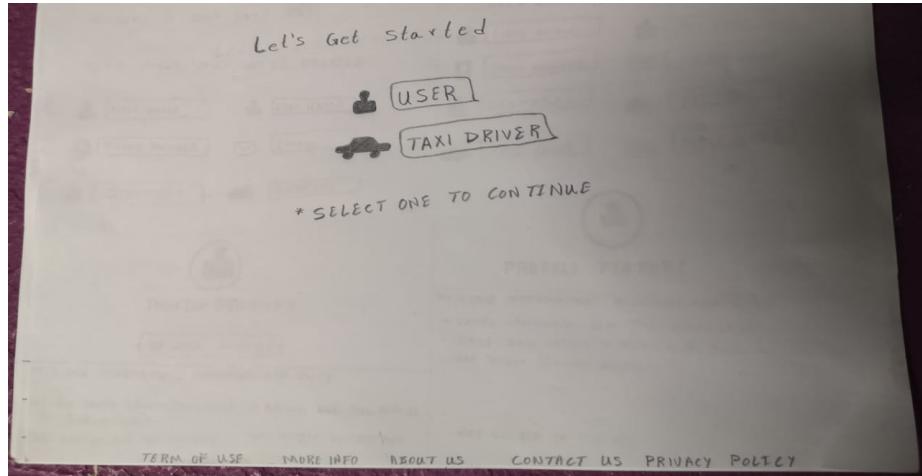
On-Screen Appearance Requirement

Identifier	Priority Level	Description
ONSREQ1	5	Map With Ability to move and zoom.
ONSREQ2	5	Map with ability to show taxis.
ONSREQ3	4	Information on Specific Taxis.
ONSREQ4	4	Toggle Between Different Driver Mode.
ONSREQ5	4	Rating System.

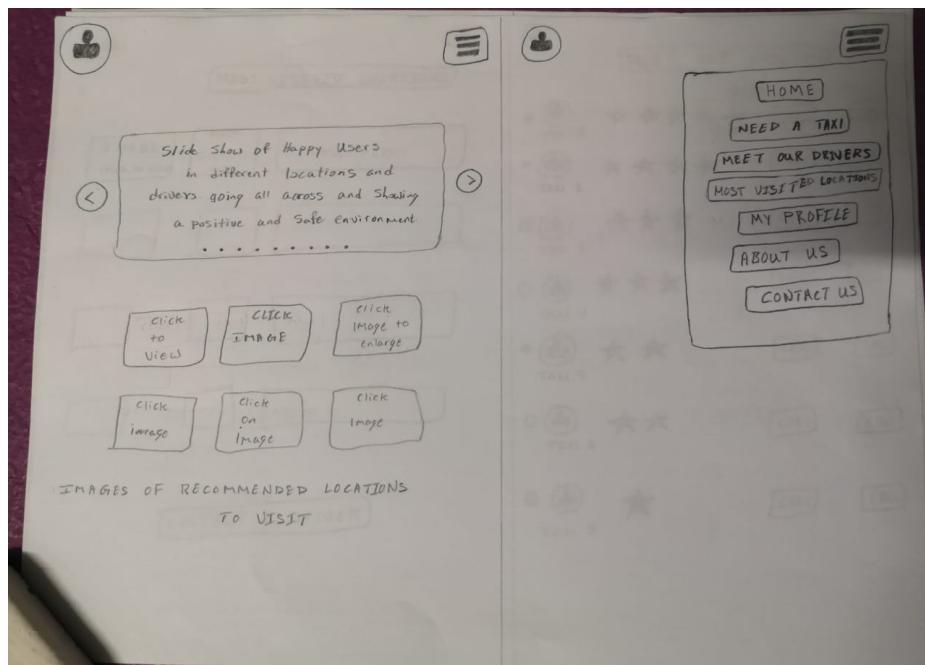
Specs & Design

Assignment Number	1
Version	01
Print Date	16/5/2021
Page	Page 8 of 104

- Create an account either a user or taxi driver



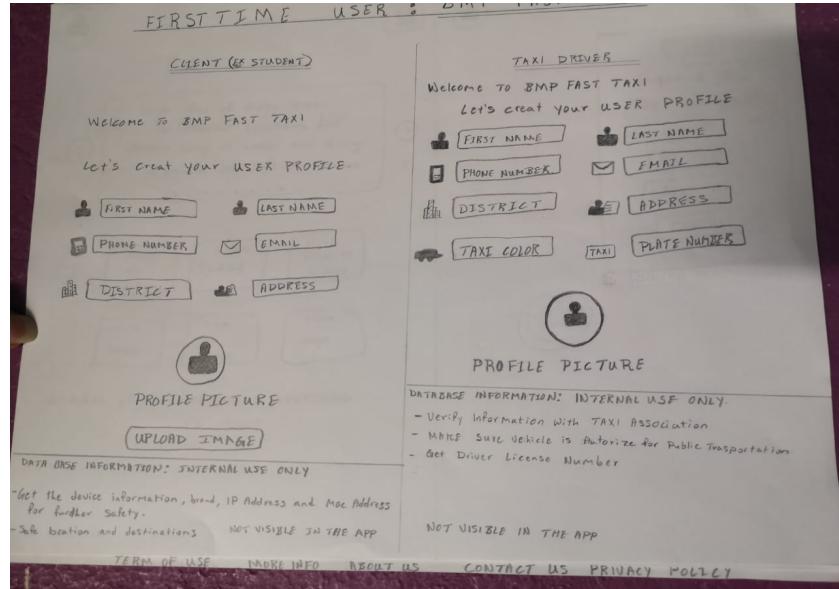
- After a user is logged in this page would provide a background of the past users comments and services pick me up offers.



- Expected platform for user or driver registration and information required to register to the taxi app.

Specs & Design

Assignment Number	1
Version	01
Print Date	16/5/2021
Page	Page 9 of 104

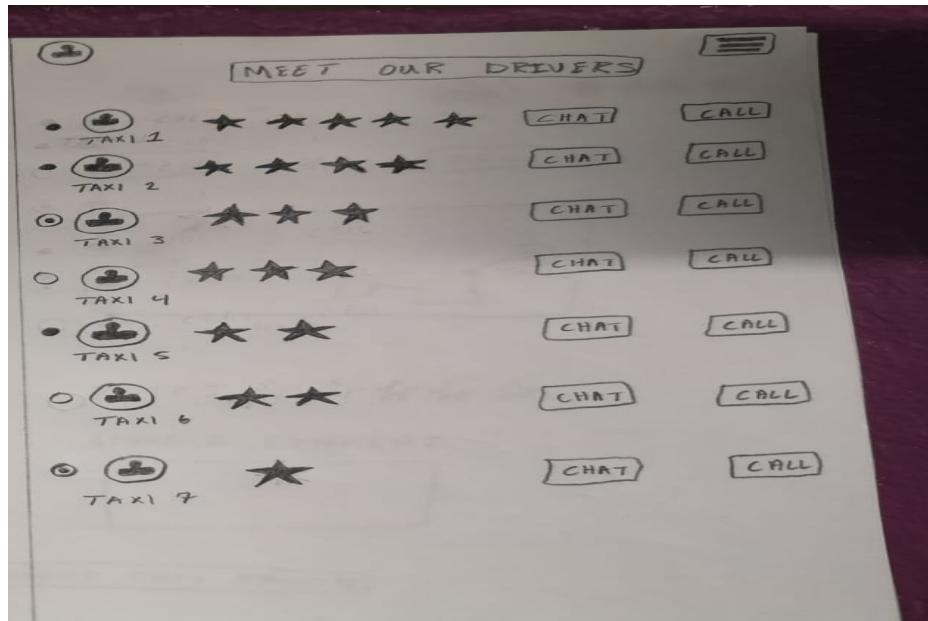


- User required information along with contact details in case assistance is needed

- A representation of ranking according to taxi drivers and background information.

Specs & Design

Assignment Number	1
Version	01
Print Date	16/5/2021
Page	Page 10 of 104



This selected interface would let the user or taxi driver create an account. Once logged in the users can view the taxi drivers along with their ratings. Users comments can also be displayed as to their experience using our services which should motivate future customers. After selecting the taxi needed once available the FIFO method would be applied, once the driver is notified of your location he/she can then proceed to picking the passenger up.

Assignment Number	1
Version	01
Print Date	16/5/2021
Page	Page 11 of 104

4. Functional Requirements Specification

a. Stakeholders

- Association for taxi drivers
- Taxi drivers
- Clients (anyone who will be requesting pickups)
- Developers

b. Actors and Goals

Actor	Roles	Types	Goals
Admin	<ul style="list-style-type: none"> ● Add Driver Information ● Update Driver Information ● Delete Driver Information 	<<initiate>>	Modify database
Driver	<ul style="list-style-type: none"> ● Accept client requests ● Change Status 	<<initiate>>	Get clients
Client	<ul style="list-style-type: none"> ● Request Pick-up ● Cancel Pick-up 	<<initiate>>	Get to his/her destination
Map API	<ul style="list-style-type: none"> ● Show current Location ● Show routes 	<<participating>>	Show routes to current client location or destination.

Assignment Number	1
Version	01
Print Date	16/5/2021
Page	Page 12 of 104

Comments:

Our Map API actor was added because it was needed to show the interaction of our application with the API we were using to implement the map. This actor was labeled as a participating actor because its only use is when an initiating actor calls a function that needs to display and update the map.

c. Use Cases**i. Casual Description**

Use case	Description	Requirement
UC-1: AddDriver	<ul style="list-style-type: none"> Allows the administrator to add new taxi drivers into the database. 	REQ4, REQ5
UC-2: DeleteDriver	<ul style="list-style-type: none"> Allows the administrator to remove existing taxi drivers from the database. 	REQ4, REQ5
UC-3: UpdateDriver	<ul style="list-style-type: none"> Allows the administrator to edit existing information of taxi drivers into the database. 	REQ4, REQ5
UC-4: Login	<ul style="list-style-type: none"> Allows the Admin, Clients and Drivers to login to their respective accounts. 	REQ5
UC-5: ViewTaxiInfo	<ul style="list-style-type: none"> Allow the Administrator to view the information about a specific driver. 	REQ5
UC-6: RequestPickup	<ul style="list-style-type: none"> Allows the commuter to request for a taxi at a given location. 	REQ12
UC-7: CancelPickup	<ul style="list-style-type: none"> Allow the commuter to cancel the request for a taxi at a given location. 	REQ11
UC-8: RateDriver	<ul style="list-style-type: none"> Allows the client to rate the taxi after a successful transaction has been completed. 	REQ10

Specs & Design

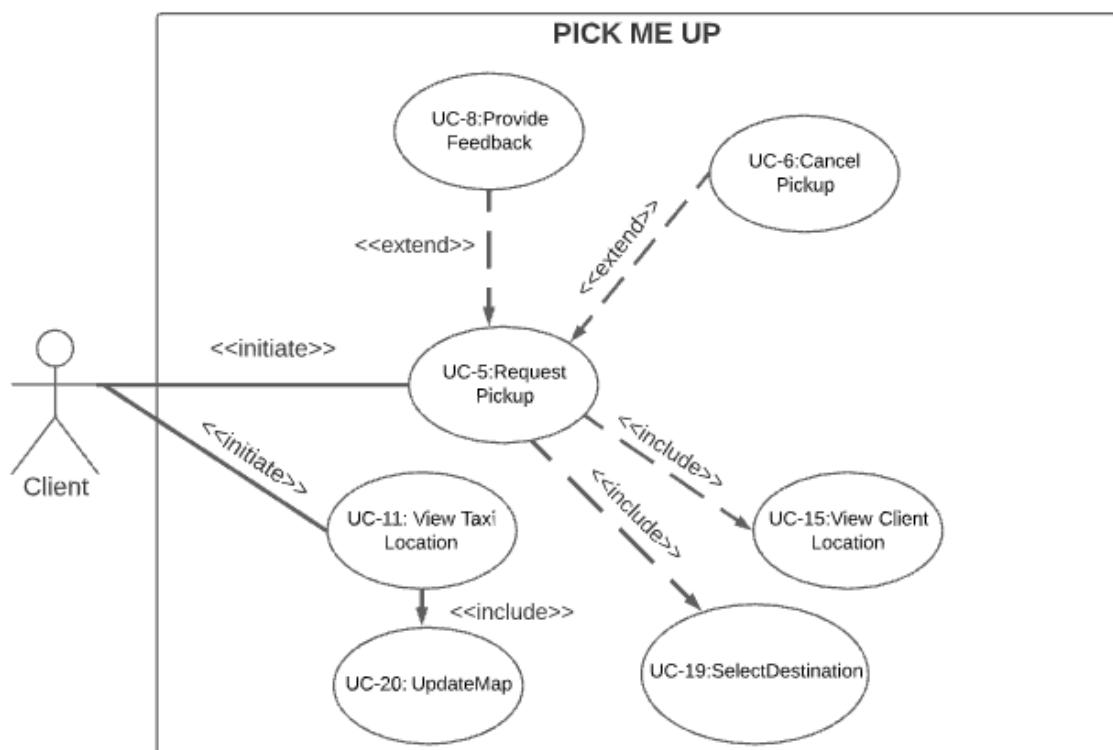
Assignment Number	1
Version	01
Print Date	16/5/2021
Page	Page 13 of 104

UC-9: ActivateTaxi	<ul style="list-style-type: none"> Allows the taxi Driver location to be seen on the map. 	REQ8, REQ5
UC-10: DeActivateTaxi	<ul style="list-style-type: none"> Allows the taxi Driver location be hidden on the map. 	REQ8, REQ5
UC-11: ViewTaxi Location	<ul style="list-style-type: none"> View all locations of nearby taxis. 	REQ1, REQ2
UC-12: ViewTaxi Information	<ul style="list-style-type: none"> Allow the client to see information about the driver such as vehicle color, licence plate and name of the driver. 	REQ9
UC-13: viewPickup	<ul style="list-style-type: none"> See the location of the requested pickup. 	REQ12
UC-14: selectDestination	<ul style="list-style-type: none"> Let the client select his/her destination. 	REQ12
UC-15: viewRating	<ul style="list-style-type: none"> Allows the taxi driver to view the ratings they have received. 	REQ10
UC-16: viewPath	<ul style="list-style-type: none"> create/show the shortest part to the client's destination 	REQ3, REQ12
UC-17: listRequest	<ul style="list-style-type: none"> List a request of waiting clients needing a pickup to the driver. 	REQ7, REQ5
UC-18: AcceptRequest	<ul style="list-style-type: none"> Accepts the request the client submitted 	REQ6, REQ5
UC-19: updateMap	<ul style="list-style-type: none"> Updates the map to show the location of the taxis. 	REQ1, REQ2
UC-20: Get Location	<ul style="list-style-type: none"> Retrieves the pickup location for the current user. 	REQ12

Assignment Number	1
Version	01
Print Date	16/5/2021
Page	Page 14 of 104

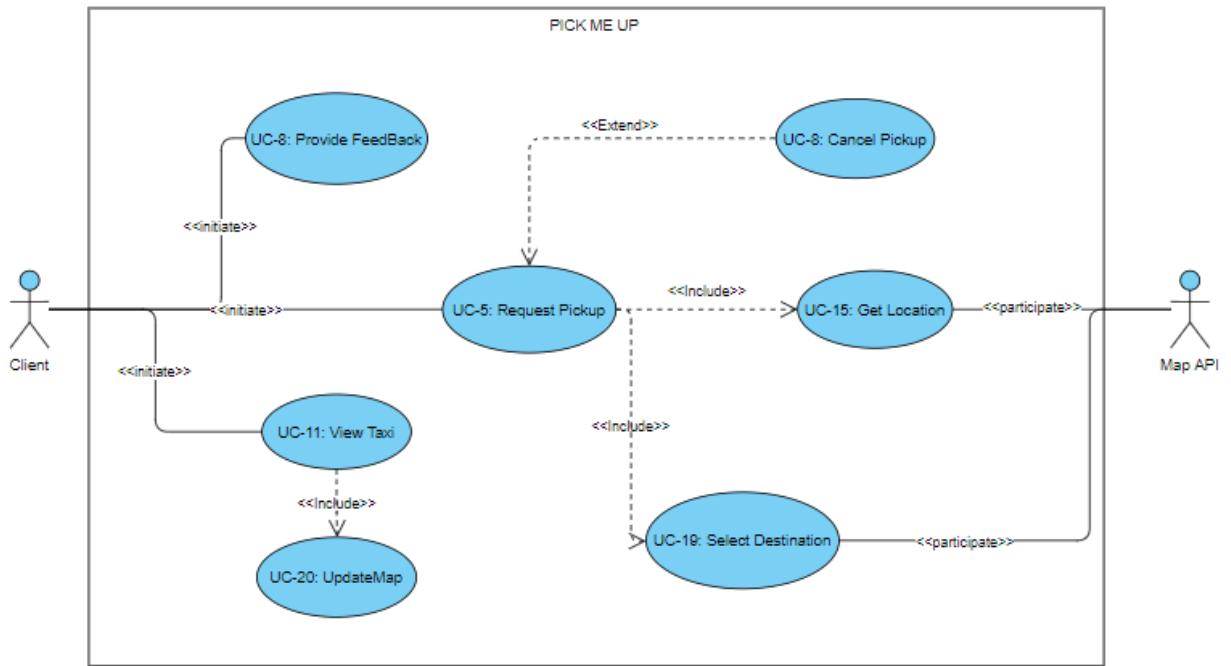
Comment:

Use case 20 was added to assist in pinpointing the location of the client when a Pick me up request is initiated. This assists in keeping track of the client location so that drivers can easily go to the pick up location without wasting any valuable time.

ii. Use Case Diagram**1) Client Use case Diagram****Original**

Assignment Number	1
Version	01
Print Date	16/5/2021
Page	Page 15 of 104

Revised



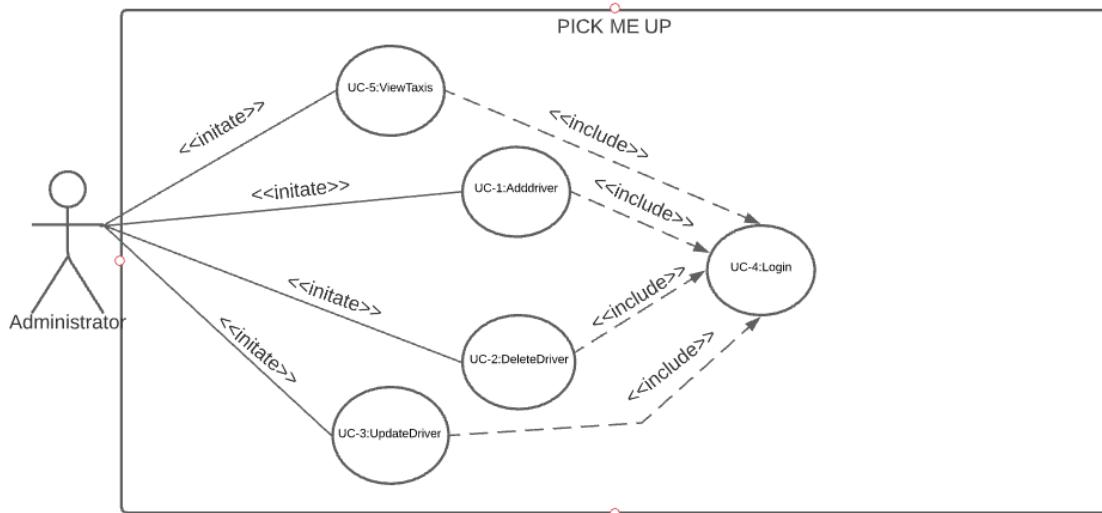
Comment:

After revision of the Client use case diagram, it was decided that the client is not required to request a Pickup to submit feedback on drivers. The Client can now initiate a feedback once opening the application. A Get Location use case was also added, so when the Client sends a Pick me up Request, the client pickup location can be saved into the request which will allow drivers to know where to pick up the client.

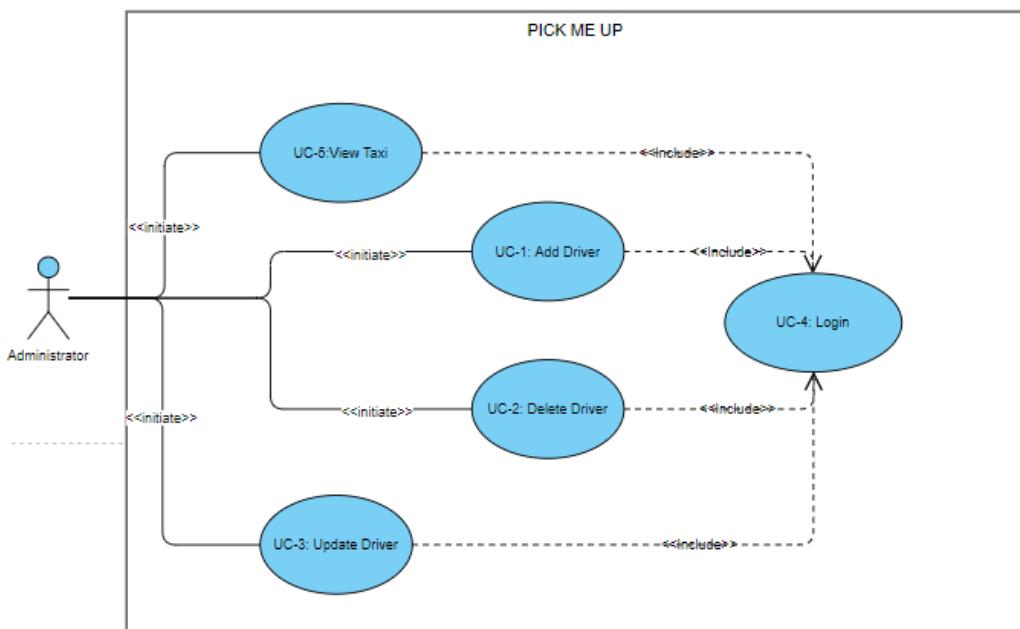
Assignment Number	1
Version	01
Print Date	16/5/2021
Page	Page 16 of 104

2) Administrator Use Case Diagram

Original



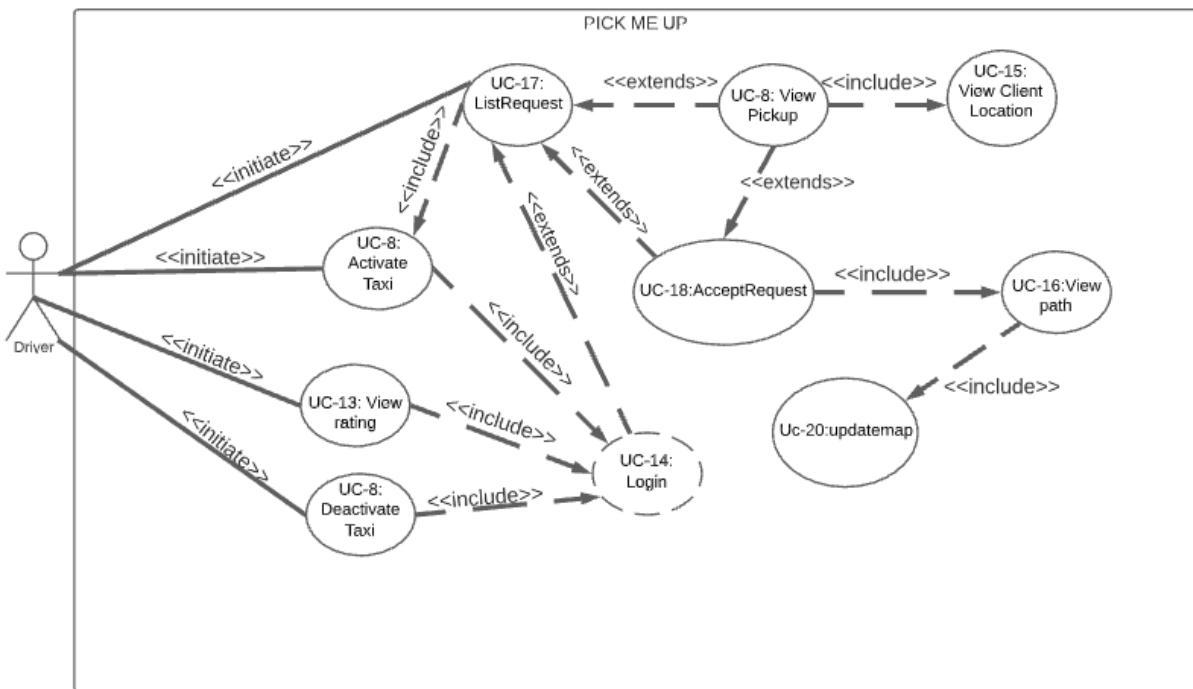
Revised



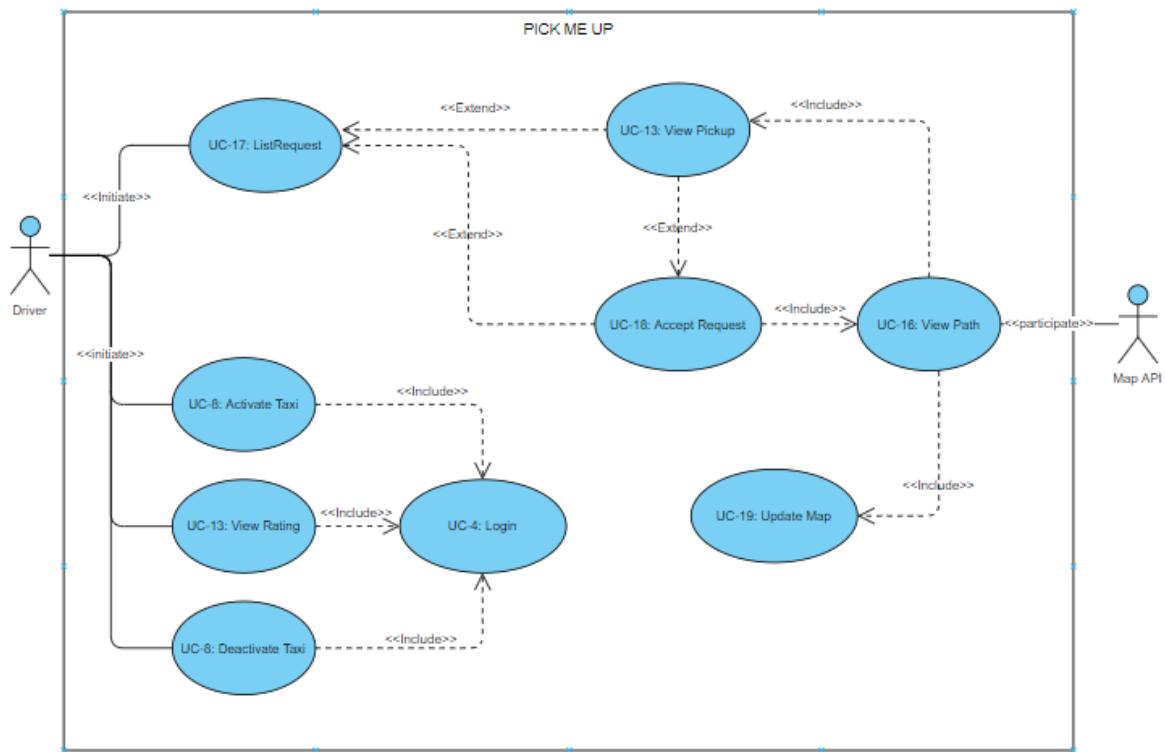
Assignment Number	1
Version	01
Print Date	16/5/2021
Page	Page 17 of 104

Comment:

After revision, we decided that the administrator use case diagram will remain the same. The original use case diagram shows all the functionalities that the administrator should have access to.

3) Driver Use Case**Original**

Assignment Number	1
Version	01
Print Date	16/5/2021
Page	Page 18 of 104

Revised**Comment:**

The original Driver Use Case diagram was a complete mess, there were too many use cases connected to each other which hinder the flow of the usage. So many use cases were disconnected and the use case view client location was removed. We decided to add in a Map API as a participating actor, this will allow us to demonstrate how the application interacts with a Map API to feed itself information.

Assignment Number	1
Version	01
Print Date	16/5/2021
Page	Page 19 of 104

iii. Traceability Matrix

Requirements	P W	U C4	UC 5	UC 6	UC 7	UC 8	UC 9	UC 10	UC 11	UC 12	UC 13	UC 14	UC 15	UC 16	UC 17	UC 18	UC 19	UC 20
REQ1	1								X								X	
REQ2	1									X							X	
REQ3	4														X			
REQ4	3																	
REQ5	4	X	X				X	X								X		
REQ6	5															X	X	
REQ7	4															X		
REQ8	2						X	X										
REQ9	3									X								
REQ10	3					X									X			
REQ11	4				X													
REQ12	4			X								X	X		X			X
Total Weight	4	4	4	4	3	6	6	2	3	4	4	3	8	9	9	2	4	

Comments:

Removed UC1, UC2, UC3 from the table. UC20 Added to table.

Assignment Number	1
Version	01
Print Date	16/5/2021
Page	Page 20 of 104

iv. Fully-Dressed Description

Use Case 1 - Add Driver	
Related Requirements	REQ4, REQ5
Initiating Actor	Administrator
Actor's Goal	To add a driver into the database.
Participating Actor	Database, system
Preconditions	Wifi or Data is turned on Admin has already logged in - in home page
Postconditions	Driver is in the database and can now serve the people
Flow of events for main success scenario	
← ← →	<ol style="list-style-type: none"> 1. Admin fills the form with valid information and submits. 2. a) System adds the driver into the database b) Responds with a confirmation
Flow of events for extensions(Alternate Scenario)	
← → →	<ol style="list-style-type: none"> 3a. Admin Clicks on cancel. Admin cancels. Redirects to the home page. 3b. Form not completed. Highlight fields missing. 4a. Data not added into the database due to no internet connection/connection lost.

Assignment Number	1
Version	01
Print Date	16/5/2021
Page	Page 21 of 104

Use Case 5 - View Taxi	
Related Requirements	REQ5
Initiating Actor	Administrator
Actor's Goal	To view the driver's information
Participating Actor	personal device(phone, PC)
Preconditions	Wifi or Data is turned on Admin has already logged in - in home page
Postconditions	See all of the driver's information.
Flow of events for main success scenario	
← → ← →	<ol style="list-style-type: none"> 1. Admin clicks on the “View Drivers” button 2. List of Drivers shows up. 3. Admin Selects the driver 4. System shows driver details including status
Flow of events for extensions(Alternate Scenario)	
	2a. List doesn't show due to no internet

Assignment Number	1
Version	01
Print Date	16/5/2021
Page	Page 22 of 104

Use Case 6 - Request Pick-up	
Related Requirements	REQ12
Initiating Actor	Client
Actor's Goal	To request service.
Participating Actor	personal device(phone)
Preconditions	GPS and Wifi or Data is turned on
Postconditions	Requests are seen by all drivers.
Flow of events for main success scenario	
← ← → ← →	<ol style="list-style-type: none"> 1. Client clicks on “Request pick up” 2. System gets location of the Client 3. System loads map for drop-off location (optional) 4. Client selects location and confirms 5. a) System signals drivers b) redirects client to home page
Flow of events for extensions(Alternate Scenario)	
	3a. Client cancels

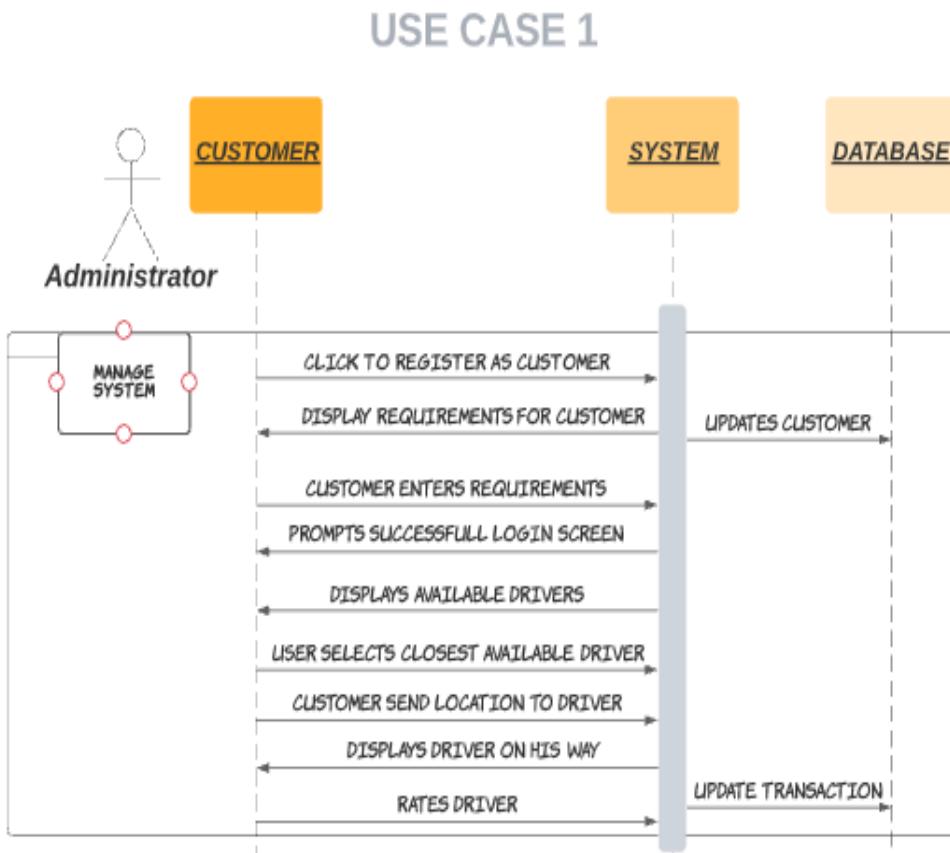
Assignment Number	1
Version	01
Print Date	16/5/2021
Page	Page 23 of 104

Use Case 18 - Accept Request	
Related Requirements	REQ5, REQ6
Initiating Actor	Driver
Actor's Goal	To give service to client
Participating Actor	personal device(phone)
Preconditions	GPS and Wifi or Data is turned on Driver is active Client has requested service
Postconditions	Driver heads to the client's location
Flow of events for main success scenario	
← → ← → ← →	<ol style="list-style-type: none"> 1. Driver clicks on “Requests” 2. System loads a “listRequest” 3. Driver clicks on a request 4. System loads page with the client’s current and drop-off location 5. Driver accepts the client’s request 6. a) Client’s request is removed from the list <ol style="list-style-type: none"> b) System sets the driver as busy c) System notifies client of acceptance d) System presents to client the driver’s car and name e) Shows direction to client
Flow of events for extensions(Alternate Scenario)	
→	<ol style="list-style-type: none"> 1a) Driver closes list 2a) List is empty, no requests at the moment. 4a) Driver clicks on “close” 6a) Request was accepted seconds earlier System replies with “Sorry, request has been accepted”

Assignment Number	1
Version	01
Print Date	16/5/2021
Page	Page 24 of 104

Comments on Use Cases:

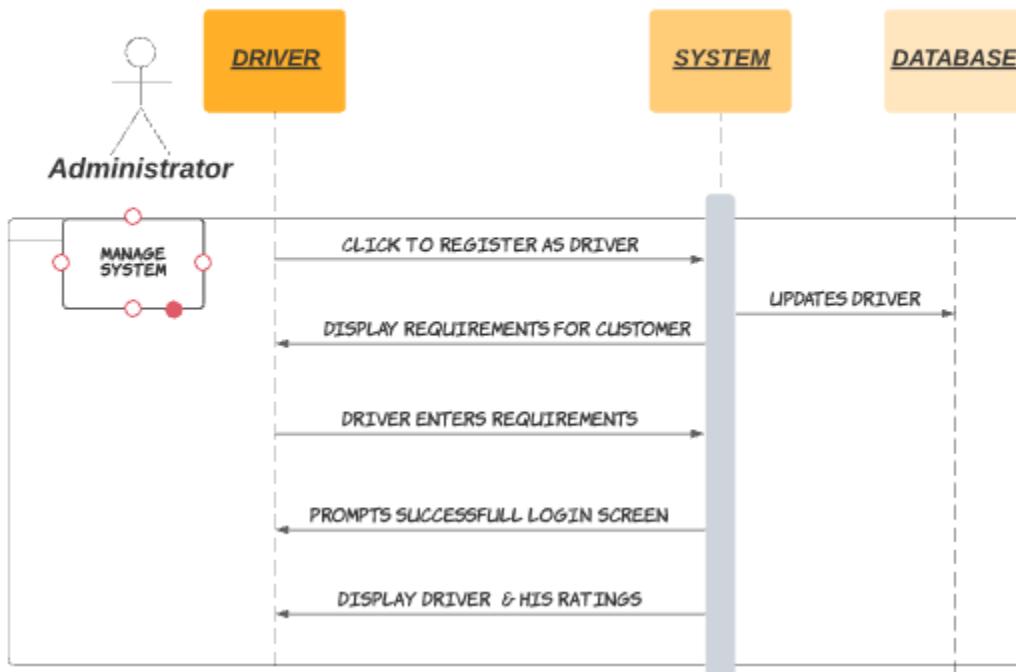
Many Use Case had unnecessary steps that weren't needed to show the flow of the scenario. Many of them were removed, some steps were rewritten to demonstrate how the events would call upon Use case to complete a task. The revised use case is still very similar in content to the original version, because it demonstrated our Functionality in a straightforward manner.

v. System Sequence Diagram**System Sequence Diagram: Customer**

Assignment Number	1
Version	01
Print Date	16/5/2021
Page	Page 25 of 104

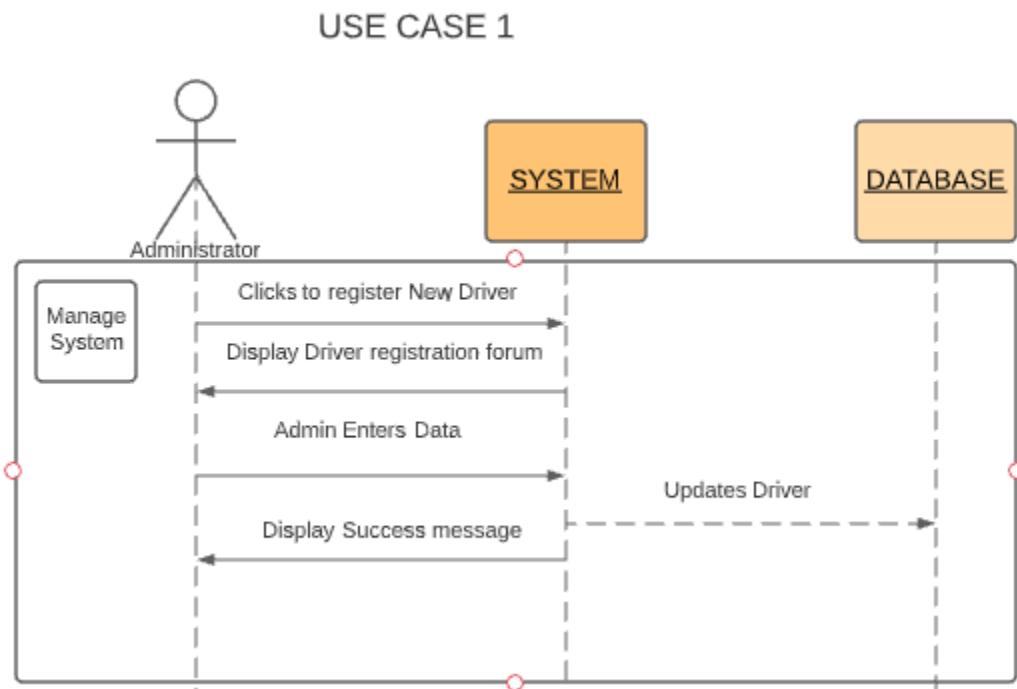
System Sequence Diagram: Driver

USE CASE 2



Assignment Number	1
Version	01
Print Date	16/5/2021
Page	Page 26 of 104

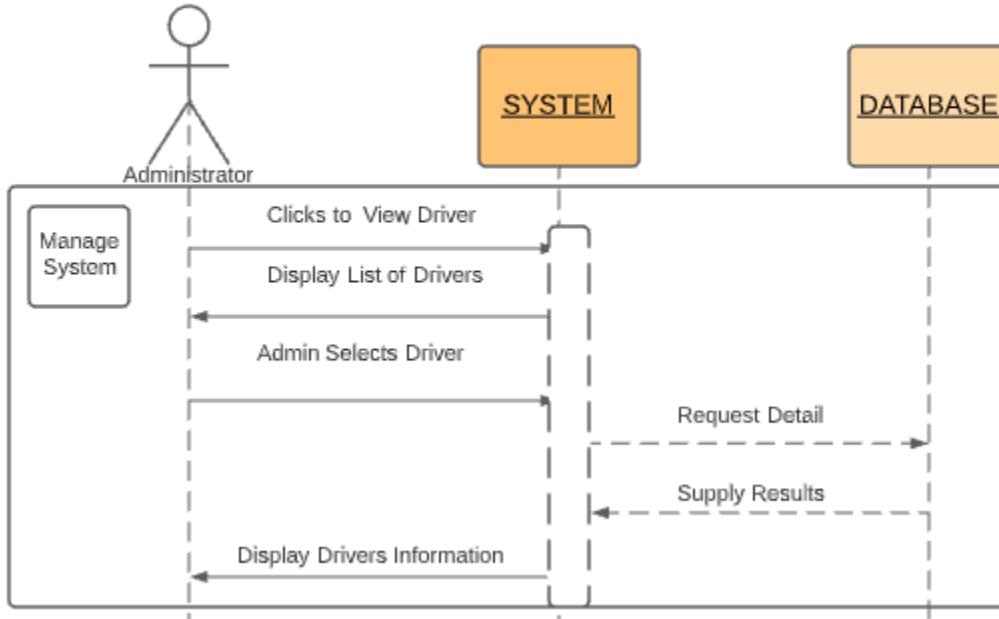
System Sequence Diagram: Use Case 1 Add driver



Assignment Number	1
Version	01
Print Date	16/5/2021
Page	Page 27 of 104

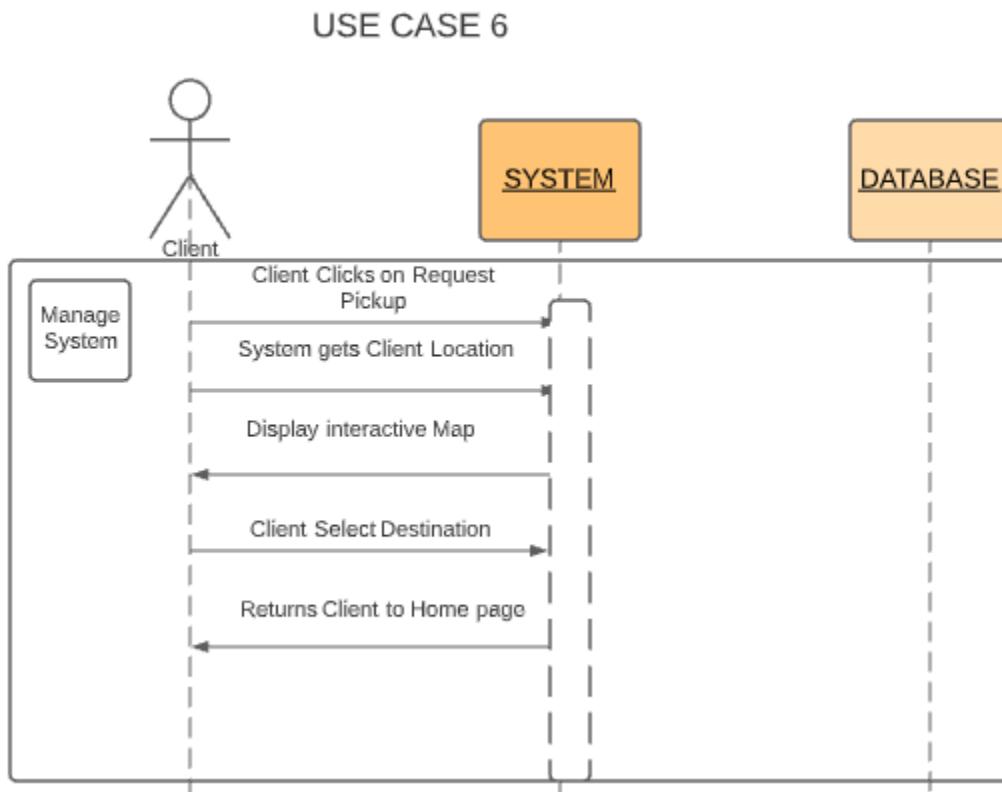
System Sequence Diagram: Use Case 5 View Taxi

USE CASE 5



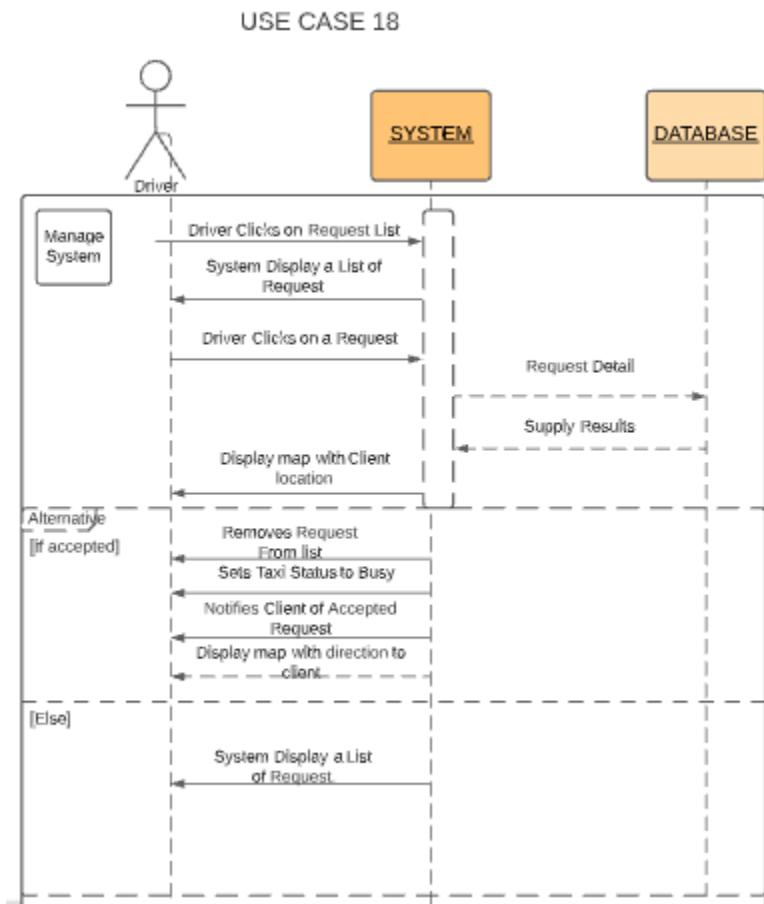
Assignment Number	1
Version	01
Print Date	16/5/2021
Page	Page 28 of 104

System Sequence Diagram: Use Case 6 Request Pickup



Assignment Number	1
Version	01
Print Date	16/5/2021
Page	Page 29 of 104

System Sequence Diagram: Use Case 18 Accept Request



Assignment Number	1
Version	01
Print Date	16/5/2021
Page	Page 30 of 104

5. Effort Estimation using Case Points

Actor	Complexity	Weight
Admin	Complex	3
Driver	Complex	3
Client	Complex	3
Map API	Simple	1
Total Unadjusted Actor Weight		10

Use Cases	Complexity	Weight
Add Driver	Simple	5
Delete Driver	Simple	5
Update Driver	Average	10
Login	Simple	5
View Taxi Info	Simple	5
Request Pickup	Average	10
Cancel Request	Simple	5
Rate Driver	Simple	5
Activate Taxi	Simple	5
View Taxi Information	Simple	5
View Pickup	Simple	5

Specs & Design

Assignment Number	1
Version	01
Print Date	16/5/2021
Page	Page 31 of 104

Select Destination	Simple	5
List Requests	Simple	5
Accept Requests	Average	10
Get Location	Simple	5
Total Unadjusted Use Case Weight		90

Technical Factor	Weight	Perceived Complexity	Calculated Factor
Security is important	1	4	4
Availability on all Operating Systems	2	4	8
Easy to use	1	4	4
Ease of install	0.5	2	1
Fast to use (limited steps)	0.5	4	2
Usable on low end devices	2	3	6
Multiple (concurrent) users	1	3	3
Total Technical Complexity Factor		28	

Specs & Design

Assignment Number	1
Version	01
Print Date	16/5/2021
Page	Page 32 of 104

Environmental Factor	Weight	Perceived Impact	Calculated Factor
Familiar with project	1.5	3	4.5
OOP experience	1	2	2
Experience with web development	0.5	3	1.5
Part time members	-1	0	0
Angular (bit difficult) will be used	-1	3	-3
Motivation	1	3	3
Total Environmental Factor			8

$$\begin{aligned}
 UUCP &= (UAW + UUCW) \\
 &= 10 + 90 \\
 &= \mathbf{100}
 \end{aligned}$$

$$\begin{aligned}
 TCF &= (C1 + (C2 * TF)) \\
 &= 0.6 + (0.01 * 28) \\
 &= \mathbf{0.88}
 \end{aligned}$$

$$\begin{aligned}
 ECF &= (C1 + (C2 * EF)) \\
 &= 1.4 + (-0.03 * 8) \\
 &= \mathbf{1.16}
 \end{aligned}$$

$$\begin{aligned}
 UCP &= UUCP * TCF * ECF \\
 &= 100 * 0.88 * 1.16 \\
 &= \mathbf{\underline{102.08}}
 \end{aligned}$$

Assignment Number	1
Version	01
Print Date	16/5/2021
Page	Page 33 of 104

6. Domain Analysis

- Concept Definitions

Use Case 1: addDriver

Responsibility Description	Type	Concept Name
Coordinate actions of concepts associated with this use case and delegate the work to other concepts	D	Controller
Form with driver's details to be saved	K	Driver details
Check that all fields are filled and has correct format	D	Information Checker
Prepare database, to save driver's details	D	Database Connection
Prepare pop up/dialogue box, informing that the driver has been added	D	Pop-up Maker

Use Case 5: viewTaxiInfo

Responsibility Description	Type	Concept Name
Coordinate actions of concepts associated with this use case and delegate the work to other concepts	D	Controller
Container for the driver's licence ID	K	Search Key
Prepare database query for the Admin's request	D	Database Connection
Container for the driver details	K	Driver details
Render the received details into an HTML page, to be passed to the administrator's browsers for display	D	Page Maker

Specs & Design

Assignment Number	1
Version	01
Print Date	16/5/2021
Page	Page 34 of 104

Use Case 6: requestPickup

Responsibility Description	Type	Concept Name
Coordinate actions of concepts associated with this use case and delegate the work to other concepts	D	Controller
Get client's current location	K	Current Location
Get client's destination location	K	Destination Location
Prepare map for client to select destination	D	Map Maker
Prepare database for client request	D	Database Connection
Prepare pop up/dialogue box, informing of driver added	D	Pop-up Maker

Use Case 9: activateTaxi

Responsibility Description	Type	Concept Name
Coordinate actions of concepts associated with this use case and delegate the work to other concepts	D	Controller
Get the driver's current state	K	Store state
Get the driver's new state	K	Store new state
Prepare pop up/dialogue box with the possible states of availability (busy, available, unavailable)	D	Selection Maker
Save state to database	D	Database Connection

Specs & Design

Assignment Number	1
Version	01
Print Date	16/5/2021
Page	Page 35 of 104

Use Case 17: listRequest

Responsibility Description	Type	Concept Name
Coordinate actions of concepts associated with this use case and delegate the work to other concepts	D	Controller
Prepares driver's query for the list of available pickup requests	D	Database Connection
Container for list of available requests	K	Request list
Render the received details into an HTML page, to be passed to the driver's browsers for display	D	Page Maker
Prepare map for route preview	D	Map Maker

Use Case 18: AcceptRequest

Responsibility Description	Type	Concept Name
Coordinate actions of concepts associated with this use case and delegate the work to other concepts	D	Controller
Get identification of the driver accepting	K	Driver tracker
Get the client information of the request accepted	K	Client tracker
Check whether the service request is still available	D	Check availability
Updates the database, removes the request selected	D	Remove Request
Render map route	D	Map Maker
Notify the client that his/her request has been accepted	D	Notifier

Specs & Design

Assignment Number	1
Version	01
Print Date	16/5/2021
Page	Page 36 of 104

- Association Definitions

Use Case 1: addDriver

Concept pair	Association description	Association name
Controller ↔ Driver Details	Stores the driver's data entered	Store data
Driver Details ↔ Information Checker	Check that the information in the form is complete and check the format	Check data
Controller ↔ Database Connection	Controller passes the query to the database	queries data
Database Connection ↔ Pop-up Maker	Database Connections passes the message that the data has been saved in the database	Provides response
Pop-up Maker ↔ Page Interface	The pop-up is displayed to the page of the administrator's browser	Displays

Use Case 5: viewTaxiInfo

Concept pair	Association description	Association name
Controller ↔ Search Key	Controller passes/stores the driver's license	Store data
Controller ↔ Database Connection	Controller passes the query to the database to search for the driver	Queries data
Database Connection ↔ Driver Details	Stores the results from the query passed by the database	Query results
Driver Details ↔ Page Maker	Results are passed to the page maker to display the data of the driver searched	Provide results

Specs & Design

Assignment Number	1
Version	01
Print Date	16/5/2021
Page	Page 37 of 104

Use Case 6: requestPickup

Concept pair	Association description	Association name
Controller ↔ Current Location	Gets the current location of the client	Provide Location
Controller ↔ Destination Location	Gets the destination location of the client	Provide Location
Controller ↔ Map Maker	Controller creates a mini map of the client to select his/her destination	Render Map
Controller ↔ Database Connection	Controller passes service query to the database	Queries
Database Connection ↔ Pop-up Maker	Database Connection passes a message to the client that his/her request has been placed	Provides response
Pop-up Maker ↔ Page Interface	The pop-up is displayed to the page of the administrator's browser	Displays

Use Case 9: activateTaxi

Concept pair	Association description	Association name
Controller ↔ Store state	Stores the current state of the driver	Store data
Store state ↔ Selection Maker	Selection Maker uses the current state to create the pop-up with the states except the current state	Passes data
Selection Maker ↔ Store new state	Saves the selected state	Provide selection
Selection Maker ↔ Database Connection	Saves the new state to the database	Save state
Selection Maker ↔ Page Interface	Displays the pop-up display with the states	Displays

Specs & Design

Assignment Number	1
Version	01
Print Date	16/5/2021
Page	Page 38 of 104

Use Case 17: listRequest

Concept pair	Association description	Association name
Controller ↔ Database Connection	Controller passes the query to the database	Queries
Database Connection ↔ Request list	Request list holds the data passed by the database	Database results
Request list ↔ Page Maker	Request list passes the results to the Page Maker to display it to the driver	Provide results
Request list ↔ Map Maker	Render a map preview of the route	Render Map
Page Interface ↔ Map Maker	Page maker displays the mini map	Displays

Use Case 18: AcceptRequest

Concept pair	Association description	Association name
Controller ↔ Driver tracker	Controller saves/holds the identification of the driver accepting the request	Provide identification
Controller ↔ Client tracker	Controller save/holds the request selected by the driver	Provide identification
Controller ↔ Check availability	Controller passes identification of the driver and the client to see if the selected request is available, or has been taken	Provide availability
Controller ↔ Remove Request	Controller passes the identification of the driver and the client request to be removed	Update list
Remove Request ↔ Map Maker	Database provides the route to be taken	Render map
Controller ↔ Notifier	Controller calls the notifier of the client, letting him/her of accepted request	Notify client

Assignment Number	1
Version	01
Print Date	16/5/2021
Page	Page 39 of 104

Attribute Definitions

Concept	Attributes	Attribute Description
Controller	NA.	
Driver details	First name Last name Address Contact Number Driver's license num. Vehicle License Plate	Driver's first name Driver's last name Driver's home address Driver's phone number Driver's driving license number Driver's vehicle license plate number
Information checker	First name Last name Address Contact Number Driver's license num. Vehicle License Plate	Driver's first name Driver's last name Driver's home address Driver's phone number Driver's driving license number Driver's vehicle license plate number
Database connection	NA.	
Pop-up maker	NA.	
Search key	Driver's license num.	Driver's driving license number
Page maker	NA.	
Current Location	NA.	
Destination Location	NA.	
Map Maker	Current Location Destination Location	The current GPS location of the client The destination location of the client
Store state	NA.	
Store new state	NA.	

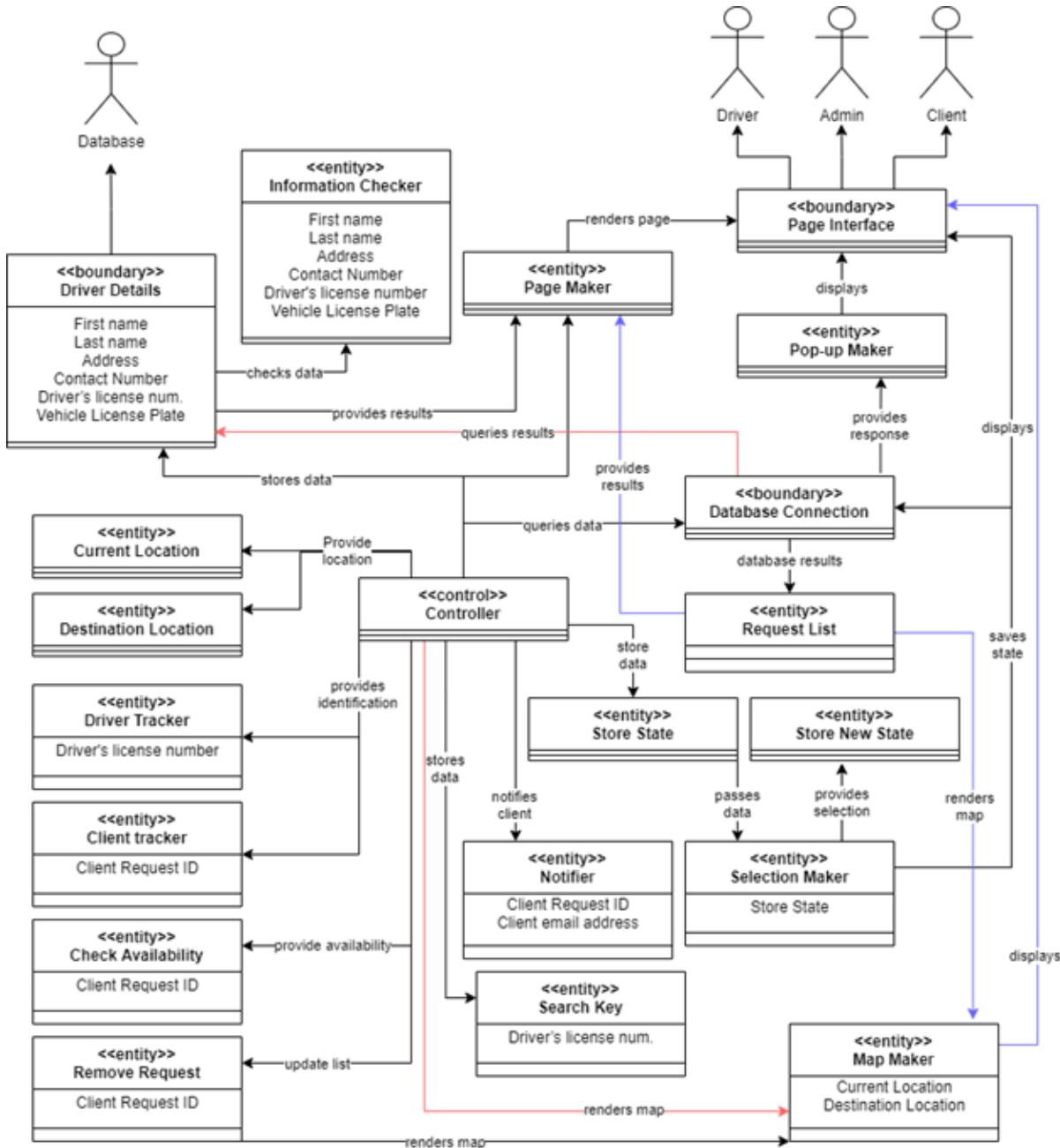
Specs & Design

Assignment Number	1
Version	01
Print Date	16/5/2021
Page	Page 40 of 104

Selection Maker	Store state	The current state that the driver is in
Driver tracker	Driver's license num.	Driver's driving license number
Client tracker	Client request ID	The identification of the request made by the client
Check availability	Client request ID	The identification of the request made by the client
Remove Request	Client request ID	The identification of the request made by the client
Notifier	Client request ID Client email address	The identification of the request made by the client Client's email address, used to notify client of acceptance

Specs & Design

Assignment Number	1
Version	01
Print Date	16/5/2021
Page	Page 41 of 104



Assignment Number	1
Version	01
Print Date	16/5/2021
Page	Page 42 of 104

Traceability Matrix

REQ	UC-1	UC-5	UC-6	UC-9	UC-17	UC-18
PW	8	4	4	6	9	9
Controller	X	X	X	X	X	X
Driver details	X	X				
Information checker	X					
Database connection	X	X	X	X	X	
Dialogue maker	X		X			
Search key		X				
Page maker		X			X	
Current Location			X			
Destination Location			X			
Map Maker			X		X	X
Store state				X		
Store new state				X		
Selection Maker				X		
Driver tracker						X
Request list					X	
Client tracker						X
Check availability						X
Remove Request						X
Notifier						X

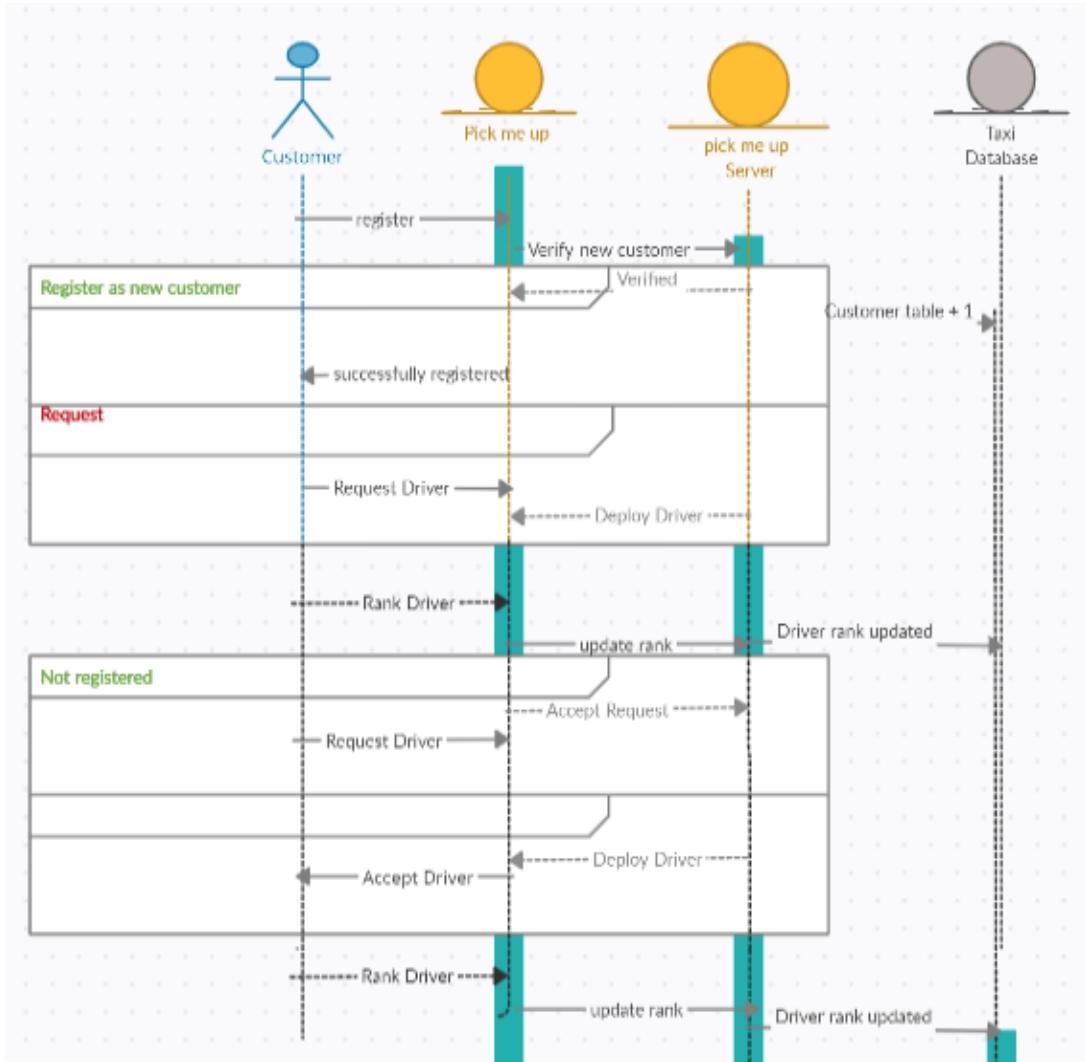
Specs & Design

Assignment Number	1
Version	01
Print Date	16/5/2021
Page	Page 43 of 104

Above is the traceability matrix that matches the use cases with the most important use cases. For example, it shows the controller being a very important aspect to the project, as it is used by all necessary use cases. On the other hand, the Notifier is not very important as it only applies to one use case. Although, it is still more important than the Search Key, which also has one use case, but is a lower priority use case.

Assignment Number	1
Version	01
Print Date	16/5/2021
Page	Page 44 of 104

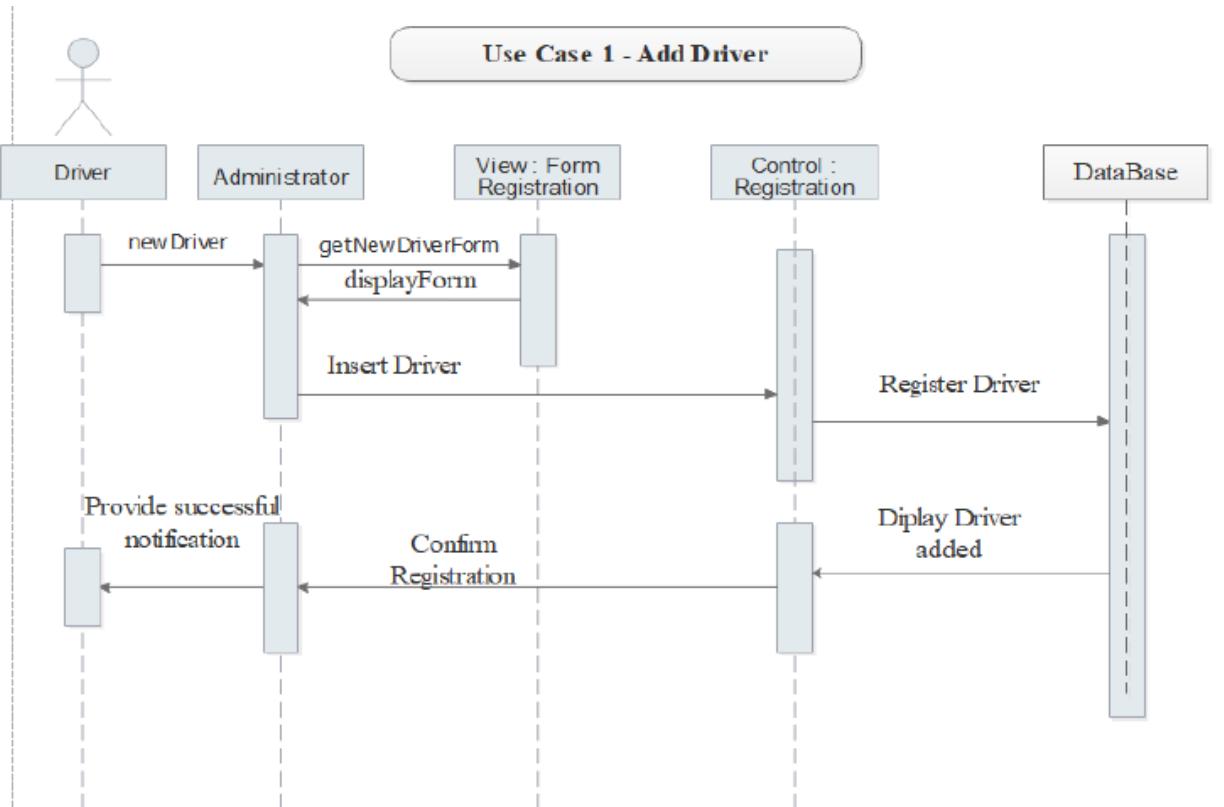
7. Interaction Diagrams



The above system shows the interaction between the customer, the pick me up app and the database . A customer can either register as a customer or simple request a driver without being registered. Note that a customer can request a driver even if not registered for simplicity reasons, since someone can request a driver for a one time only, specially visiting customer who only wants to use the pick me up service for one time only while they are in the country or passing by. Overall, all customers who request a driver from the pick me up app will be able to rank the driver if he or she so wishes, since their input is of importance to both the Administrators, taxi drivers and future customers.

Assignment Number	1
Version	01
Print Date	16/5/2021
Page	Page 45 of 104

1 . Add Driver

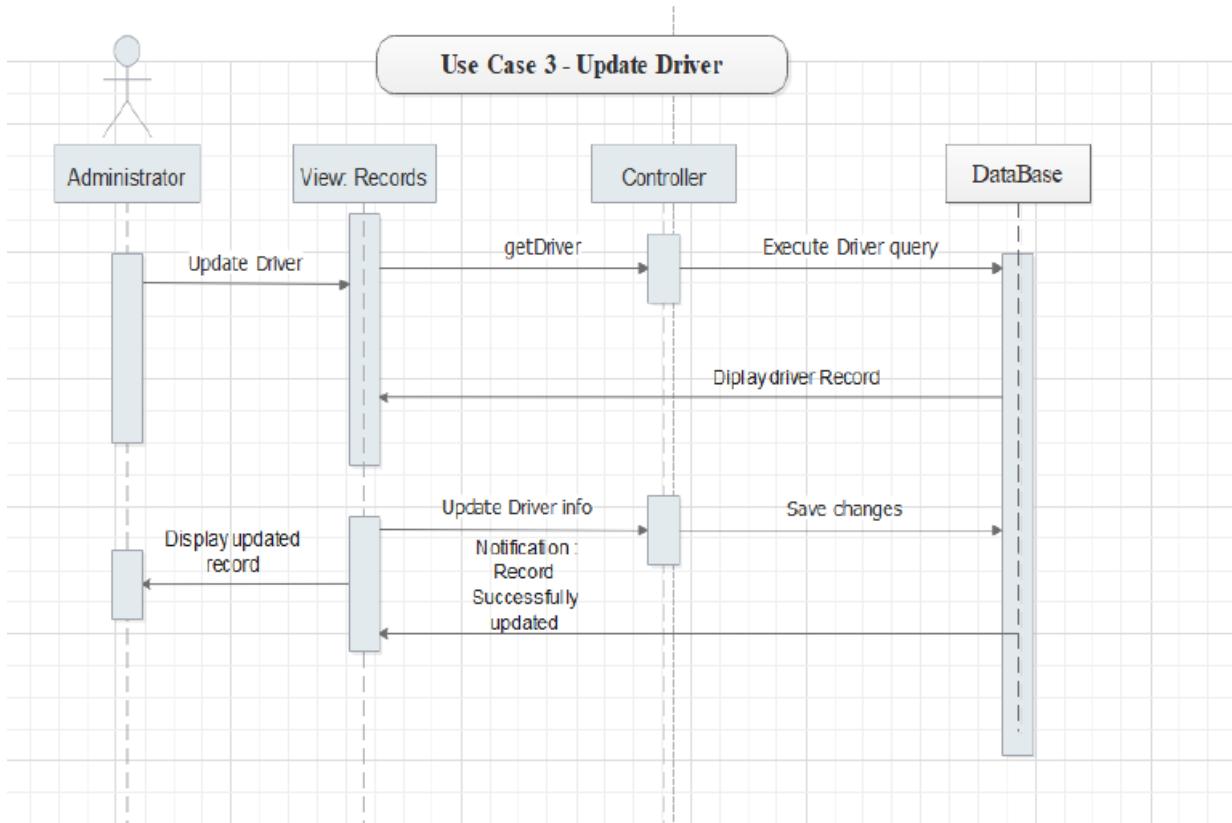


- Once a taxi driver wants to register to the “Pick me up App” the admin first proceeds to open the new driver form from the registration section, once the form meets all the requirements it is then sent to the controller which verifies the input and passes the register driver command to the database.

- Once the database registers the new driver it then sends a confirmation to the admin of the successful registration and how a new driver has been added to the Pick me up taxi group.

Assignment Number	1
Version	01
Print Date	16/5/2021
Page	Page 46 of 104

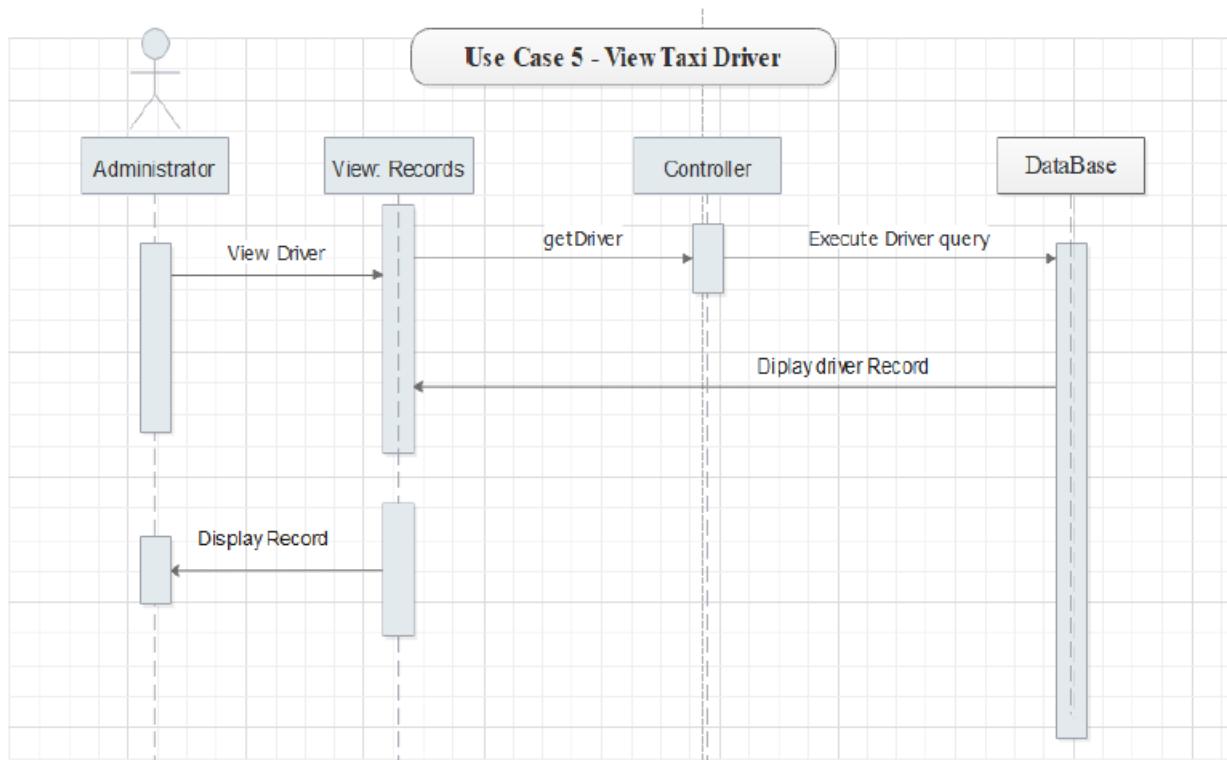
2. Update Driver



After a taxi driver either changes vehicle or licence type he then asks the administrator to make the necessary changes to reflect his record. The administrator then proceeds to call the drivers record from the database, once the update has been done the controller then issues the save changes command to the database which then notifies the admin about the record being updated successfully.

Assignment Number	1
Version	01
Print Date	16/5/2021
Page	Page 47 of 104

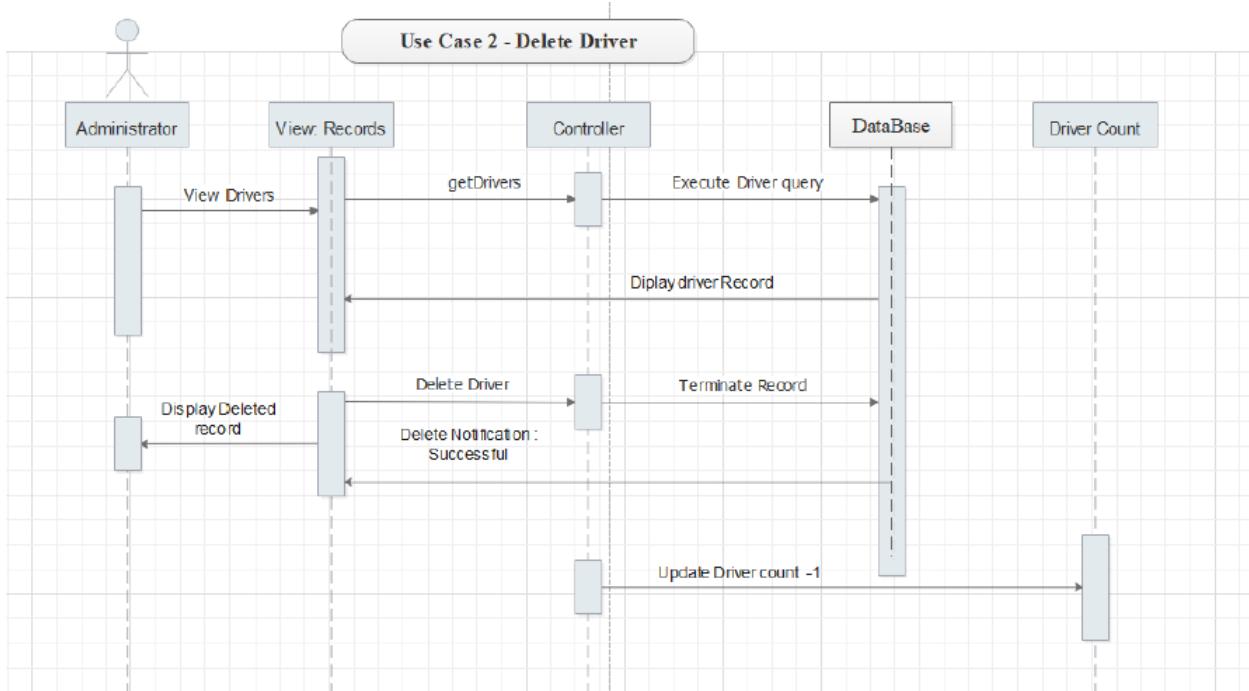
3. View Taxi Driver



- Once the administrator wants to view a driver he first checks the records list, then provides a command to the controller to getDrivers list. Once the database gets the execute driver query from the database the database displays the driver record to the administrator.

Assignment Number	1
Version	01
Print Date	16/5/2021
Page	Page 48 of 104

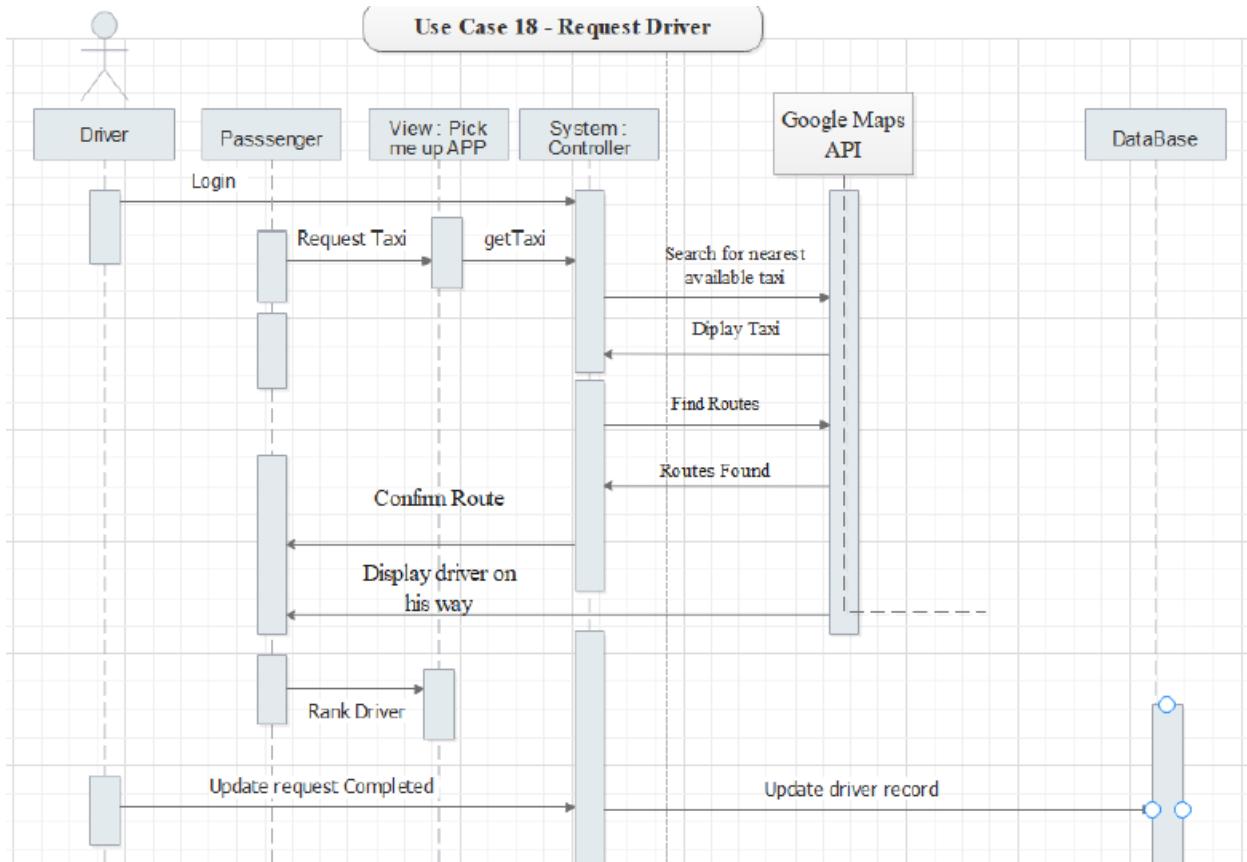
4. Delete Driver



When deleting a taxi driver, the admin first clicks the view record profile, once the record is clicked the command goes to the controller to initiate the database. The database then provides the list of drivers, the admin selects which record he/she wants to delete and passess the message to the controller to delete record X. Once record X has been deleted the database notifies the admin with a delete notification followed by the controller which removed 1 from total driver count.

Assignment Number	1
Version	01
Print Date	16/5/2021
Page	Page 49 of 104

5. Request Driver

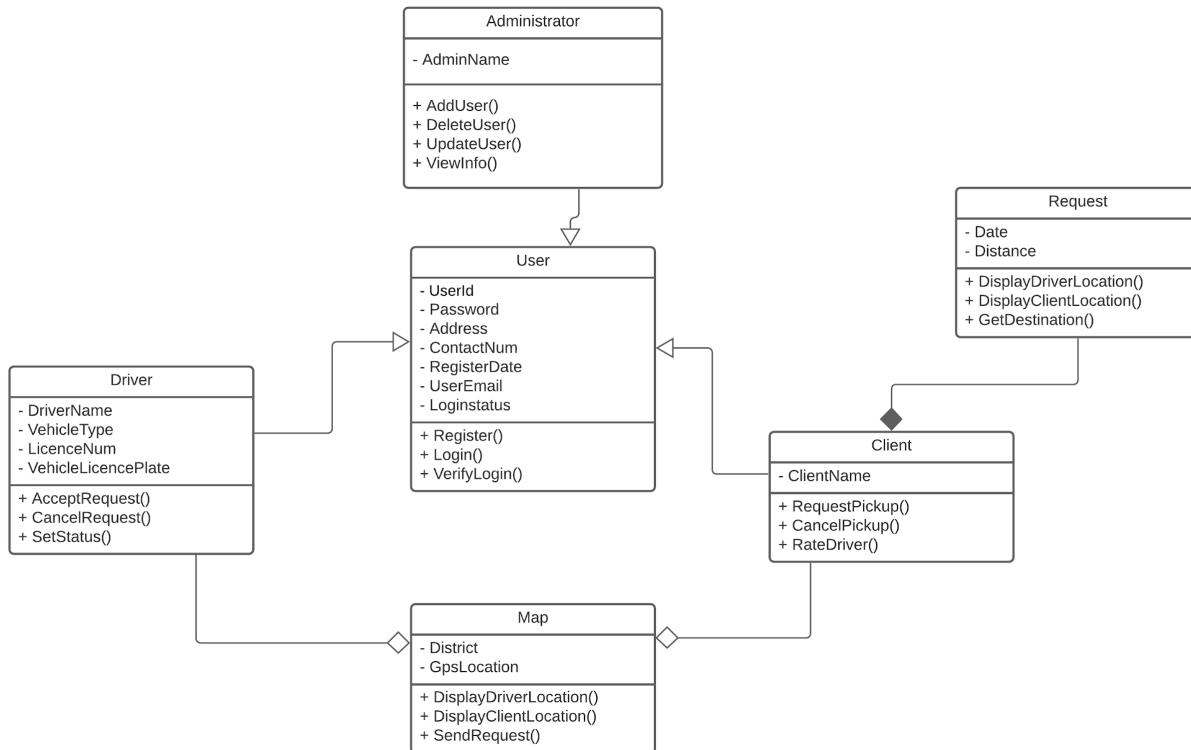


Once the driver is logged in to the pick me up app, he can then set his status to: available, busy or offline. Once the passenger clicks on the pick me up app they can then proceed to request for a taxi, the app then notifies the system controller to get an available taxi. The controller proceeds to search for the nearest available taxi in google map. Once the nearest available taxi is identified then the controller finds and identifies the nearest route to the passenger, once the passenger confirms the route google maps displays that the driver is on his way. After the passenger has been dropped off, the passenger proceeds to rank the driver where the controller then accepts the ranking information and updates the database to show the quality of service that was provided by the driver.

Assignment Number	1
Version	01
Print Date	16/5/2021
Page	Page 50 of 104

8. Class Diagram and Interface Specification

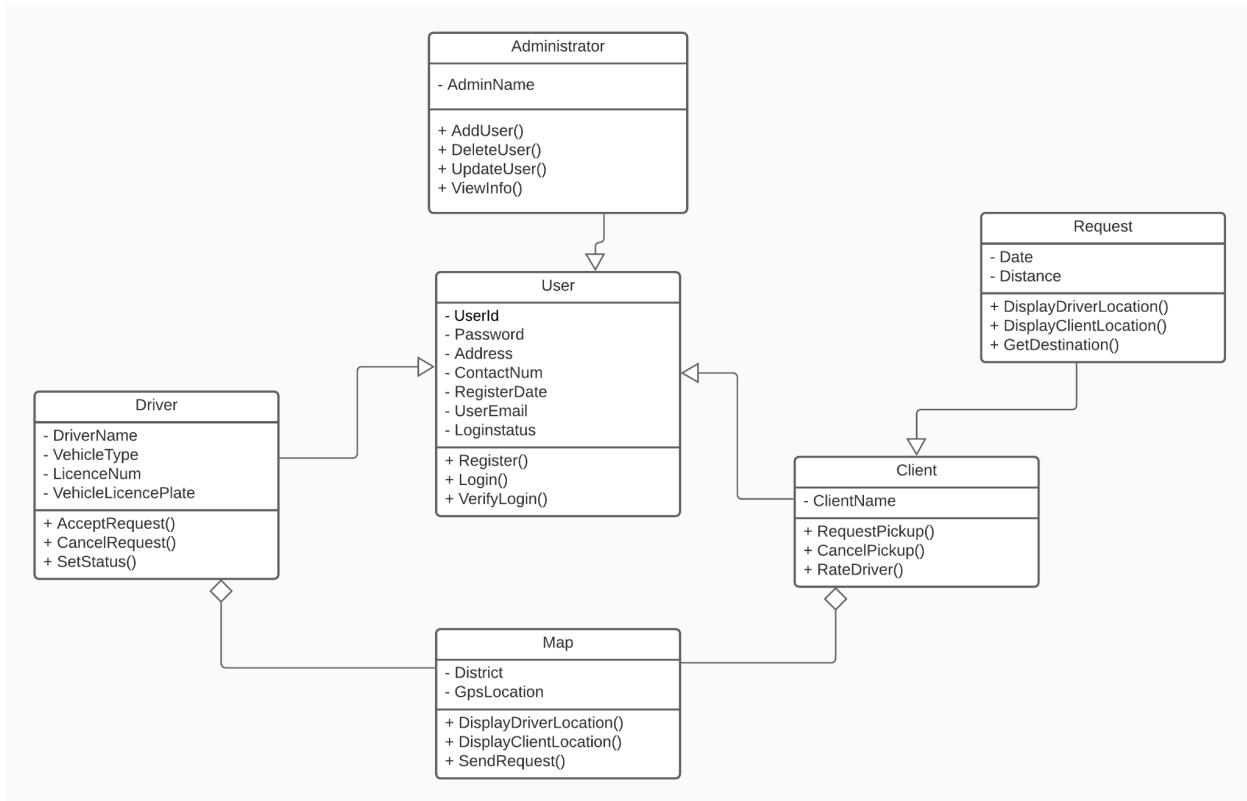
Original



Specs & Design

Assignment Number	1
Version	01
Print Date	16/5/2021
Page	Page 51 of 104

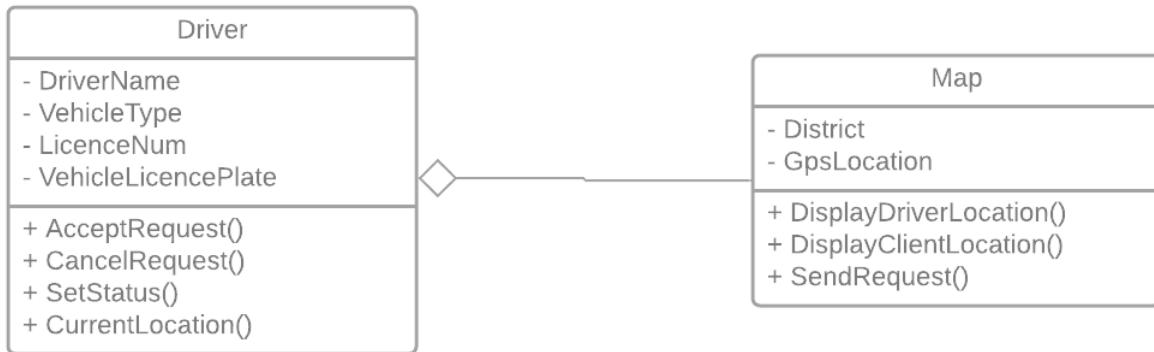
Revised



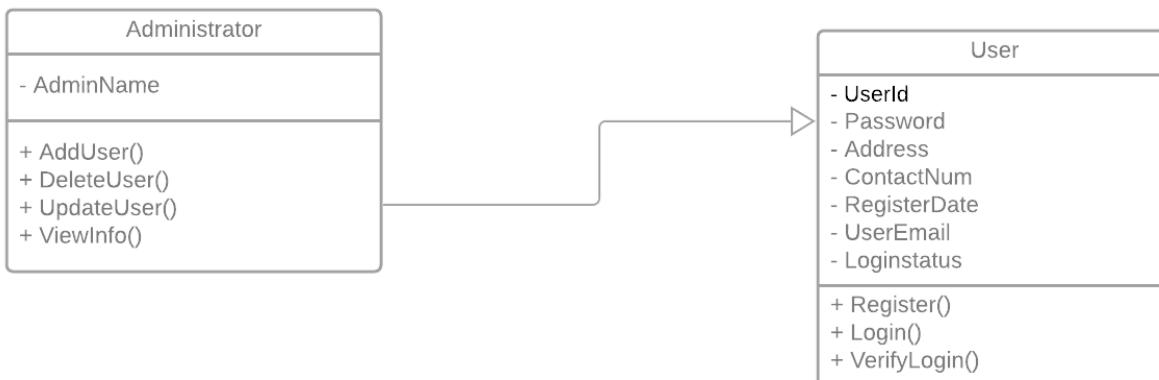
Comments: Fixed errors with diagram syntax accuracy.

Assignment Number	1
Version	01
Print Date	16/5/2021
Page	Page 52 of 104

Data Types and Operation Signatures



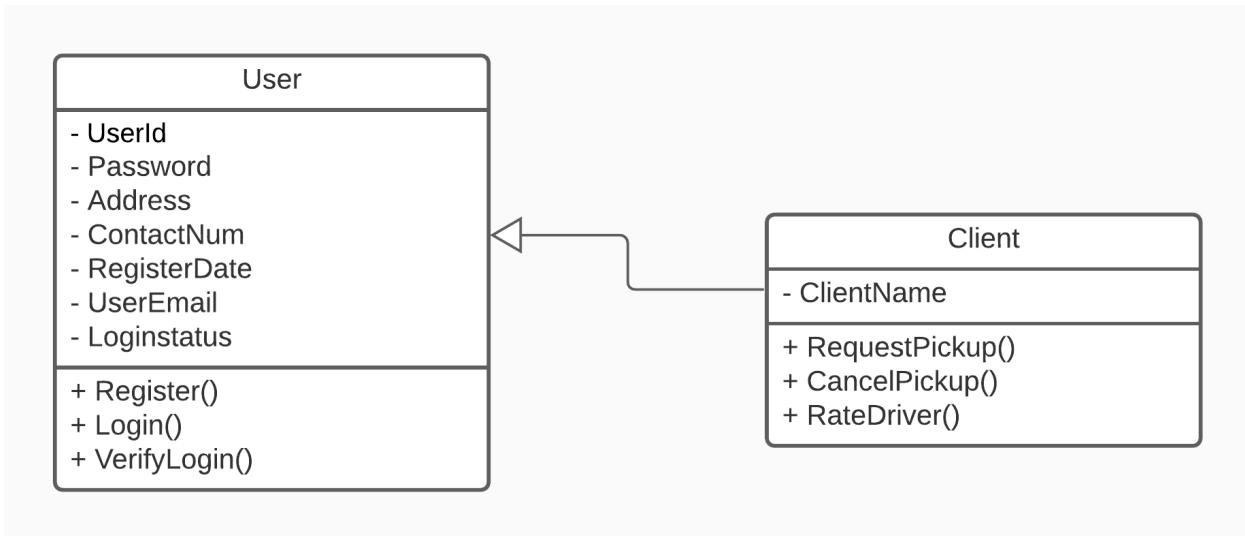
- Driver users have access to the map and may accept/cancel requests they've received. In addition they may also set their current status to "busy" "active" or "inactive"
- A Driver's location must be seen on the map by clients, however will not be shown if they are set to status "inactive"



Specs & Design

Assignment Number	1
Version	01
Print Date	16/5/2021
Page	Page 53 of 104

- Administrators inherit from the User class, and they may modify user information that includes both driver and clients.
- The Administrator is responsible for adding, updating and removing users, using the similarly named operations/methods; and users being both drivers AND clients. ViewInfo() is for admins to view information of users.



- Clients inherited the user class and are able to request and cancel a request, as well as give the option to rate the driver once a drive is over.
- Clients can request a pickup, thus notifying the driver. They may also cancel this request. At the end of a drive the client has the option to rate the driver.

Assignment Number	1
Version	01
Print Date	16/5/2021
Page	Page 54 of 104

Traceability Matrix Evolved

REQ	Driver	Admin	User	Map	Client	Requests
PW	8	4	4	6	9	9
Controller	X	X	X	X	X	X
Driver details	X	X	X			
Information checker	X	X	X		X	
Database connection	X	X	X	X	X	X
Dialogue maker		X				
Search key	X	X				
Page maker		X				
Current Location	X				X	
Destination Location				X	X	
Map Maker			X		X	X
Store state	X					
Store new state	X			X		
Selection Maker				X	X	
Driver tracker				X		
Request list						X
Client tracker						
Check availability				X		X
Remove Request					X	X
Notifier				X	X	X

Specs & Design

Assignment Number	1
Version	01
Print Date	16/5/2021
Page	Page 55 of 104

Important Contracts

Invariants: SetStatus(), GpsLocation

SetStatus Driver::SetStatus(Active, Inactive, Busy):string

Pre : result = string

Post : result = string

Request GetDestination::SetStatus(d:distance):int

Pre : User = Login

Post : result = distance

Assignment Number	1
Version	01
Print Date	16/5/2021
Page	Page 56 of 104

9. System Architecture and System Design

I. Identifying Subsystems

The system will be compatible with both mobile and desktop applications, such as Windows.

Due to this, the primary subsystems revolve around functionality of the system. We will implement a web app solution for the front end, a server side solution for the main backend, and we will make use of Google Maps to provide most of the primary mapping technology. We will also be using a database in order to store and retrieve information. The clients, drivers and administrators will all be performing actions on the front end view, while all calculations and management will be done by the server.

The overall package flow of the project can be seen below.

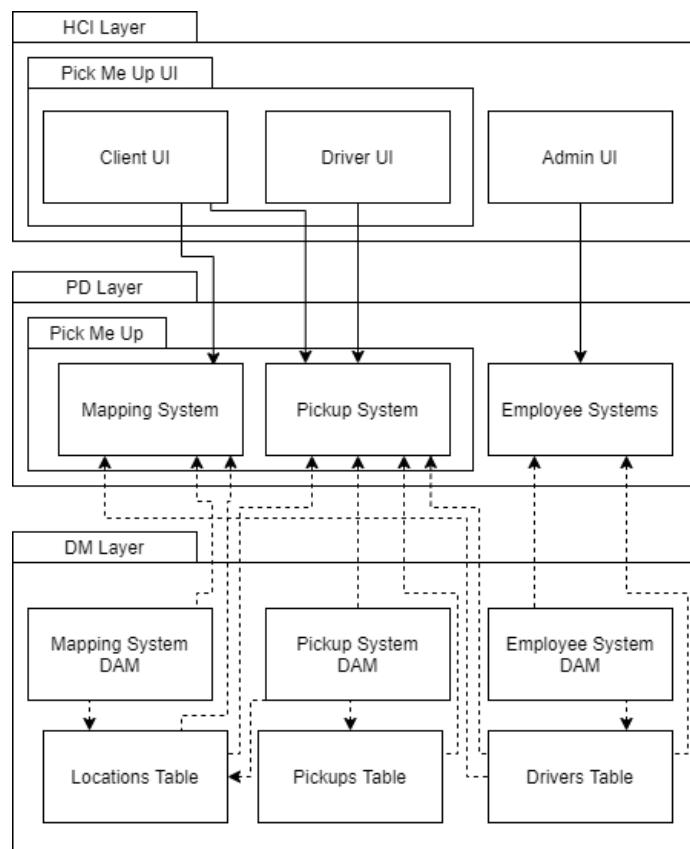


Diagram Showing Simplified Overview of the Package Diagram System

Assignment Number	1
Version	01
Print Date	16/5/2021
Page	Page 57 of 104

Our primary focus will be on the management of locations and pickups within the Pick Me Up system. A separate, driver management oriented system will be required in order to impact the drivers table. This would most likely be handled by the party affiliated with the administration of the drivers. Particularly, a taxi driver association member that would focus on regulating the admission, removal, and updating of taxi information.

Inside Pick Me Up, the main services revolve around the Map Location Viewing System and the Pickups System. In order for further clarification to be provided on how these systems would interact with each other, as well as be built, view the diagram below.

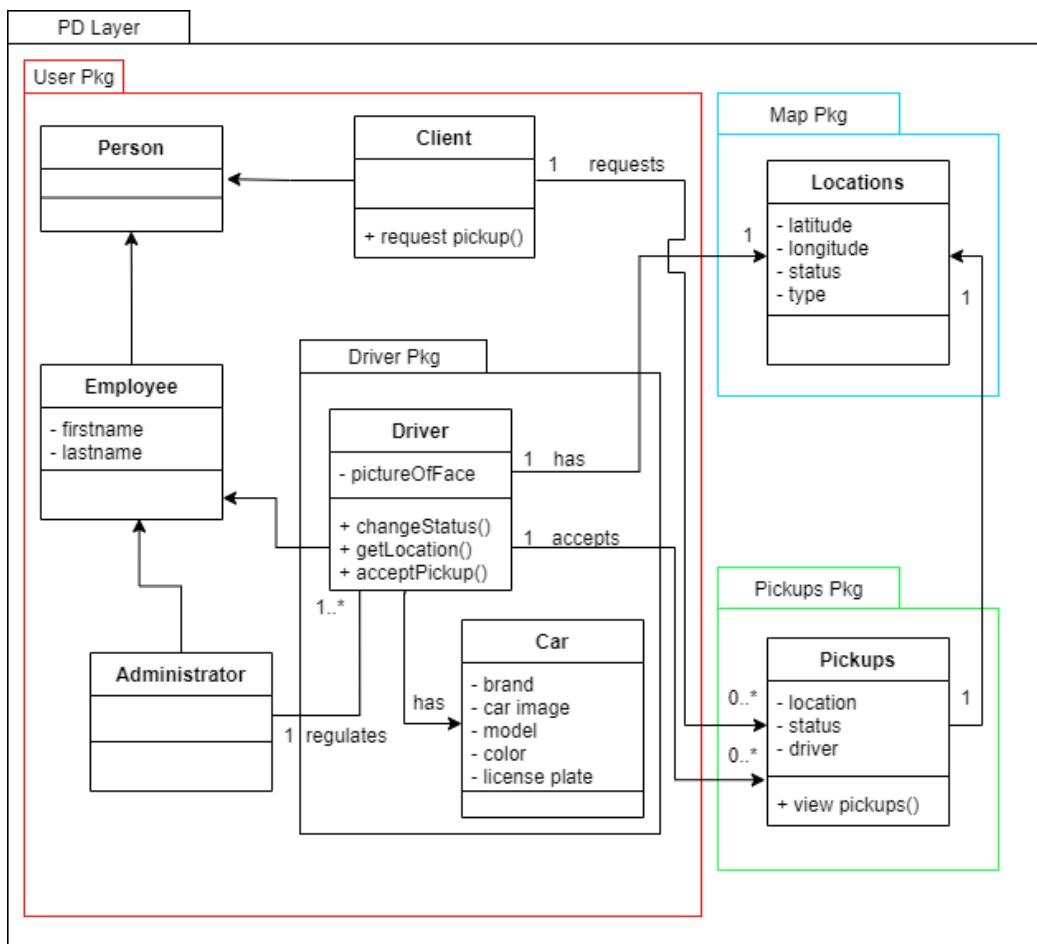


Diagram Showing UML Package Diagram of the Interactions Between the Pickup, Map, and the Systems Users

Assignment Number	1
Version	01
Print Date	16/5/2021
Page	Page 58 of 104

Architecture Styles

In the case of most web applications, the Pick Me Up system will implement a Server & Client Architecture, also known as the Master & Slave Architecture. The client will primarily focus on rendering information regarding the maps and locations on the maps. The client will also be responsible for retrieving the location of the client end hardware to be used within the computational aspects of processes. The server will primarily focus on storing the information of drivers, locations of all drivers currently being managed, as well as retrieving and interacting with the data being sent by the active users. The server/client architecture's selection was primarily due to the accessibility of information. Since most information regarding the locations and pickups will be the same, it would be more efficient, secure, and reliable to have the information at a single, primary system.

The Pick Me Up system will also be using the MVC architecture (Model, View, Controller) as its framework for building the project. The models would be separated into the subsystems shown in the identification of subsystems, such as the Pickups model, the User model, and the Locations model. The views will be the interfaces that both clients and drivers will be using in the final product. Isolating the view allows for modification of the view at any point in time to allow for tweaking and improvement throughout the project. The controllers will be the flow controllers of the system, such as in the Pick Me Up section of the PD layer in the Simplified Package Diagram. These are the Mapping System and Pickup System, who will both interact within their respective domains to accomplish their objectives while also talking to each other in order to provide a smooth experience.

Assignment Number	1
Version	01
Print Date	16/5/2021
Page	Page 59 of 104

Mapping Subsystems to Hardware

The system's nature, being a web-based system will require the system to be split between different machines. The subsystems are: the Client and the Server.

The Client

The client will be the subsystem that ordinary users will interact with through a traditional web browser. These include Chrome, Firefox, Opera, etc. Android devices will also be acting as clients within the system. This area will primarily be HTML, CSS, and JS. Since these do not provide much functionality and are instead for use with display, most of the processing will be done at the server side. JS will be used for basic display optimization and effectiveness, and managing the incoming and outgoing requests from the server.

The Server

The Server will contain the web server as well as the database server. The server will be responsible for handling all incoming and outgoing requests. Incoming requests involve the accepting of pickups, the sending of locations, etc. Outgoing requests involve sending the web displays to different clients. The database server will also be hosted on the server, allowing for efficient communication between the web server and the database.

Connectors and Network Protocols

The connectors used in the systems web application will be HTTPS, as it is a secure way of transferring information over the internet. Since it is likely the database will be hosted externally but its primary focus will be for this system, a connector will be required to connect the database to the system. It will be necessary to allow for communication between the client and server. Following the CRUD standard, requests will use GET, POST, PUT, and DELETE. Due to the database requirements, data on the client end will be formatted as a JSON file.

Assignment Number	1
Version	01
Print Date	16/5/2021
Page	Page 60 of 104

Global Control Flow

Execution Orderliness

The Pick Me Up System will be an event-driven system. This is because a user can do almost any action at any point in time. For example, even if there are no active taxi drivers, a client can still request a pickup. Since none of the tasks present require much follow up, it is easy for any action to be completed independently.

Time Dependency

The system will be real-time. This is because the live tracking requirement of the system requires constant monitoring and updating of the databases in order to get accurate and fluid functionality from the system. As having the https requests happening constantly could both be taxing on the client end, as well as require much processing on the server end, the real-time tasks will be done at short intervals, for example, a 10 second wait period between location gathering.

Hardware Requirements

Windows

Windows 7, Windows 8, Windows 8.1. Windows 10 or later

An intel pentium 4 processor or later that's SSE3 capable.

Linux

64-bit Ubuntu 14.04+, Debian 8+, openSUSE 13.3+, or Fedora Linux 24+

An intel pentium 4 processor or later that's SSE3 capable.

Android

Android Lollipop 4.0 or higher

Screen of at least 426dp x 320 dp

Assignment Number	1
Version	01
Print Date	16/5/2021
Page	Page 61 of 104

10. Algorithms and Data Structures

The system primarily consists of a pathing algorithm in order to provide the shortest path between two locations on the map. Namely, location of the driver and location of the client. The algorithm will be responsible for identifying this shortest path in order to ensure efficiency of travel. This algorithm will take place using an external API, namely, Google Directions. The system will be in charge of receiving both locations from the two independent parties, then sending it to the API. When the system retrieves the API response, the system will then show the Driver the shortest route between the two locations.

Algorithm:

Driver requests shortest route to client:

```
While “requesting_shortest_route” is true
    Set “Driver Location” to Current Location
    If cannot get Current Location
        Inform user that Locations services is disabled or Internet is not available
        continue
    End if
    Send Current Location and Client Location to Directions API
    Fetch Shortest Route from API
    Display Shortest Route on map view
End While
```

According to Crovari in an article from 2019, Google Directions uses Dijkstra's Algorithm to find the shortest route.

Assignment Number	1
Version	01
Print Date	16/5/2021
Page	Page 62 of 104

Algorithm Dijkstra(W[1..n, 1..n])

// Shows Dijkstra's Algorithm to find the shortest route in a weighted graph. According to GeeksforGeeks

// Input: A weighted graph W

// Output: Shortest Path Tree

Create empty set S that will hold shortest path tree

Assign all vertices distance values of INF

Assign source vertex distance value of 0

While S does not have all vertices

a.) Pick unrouted vertex U

b.) Include U in S

c.) Update shortest path tree with shortest path to vertex U

End While

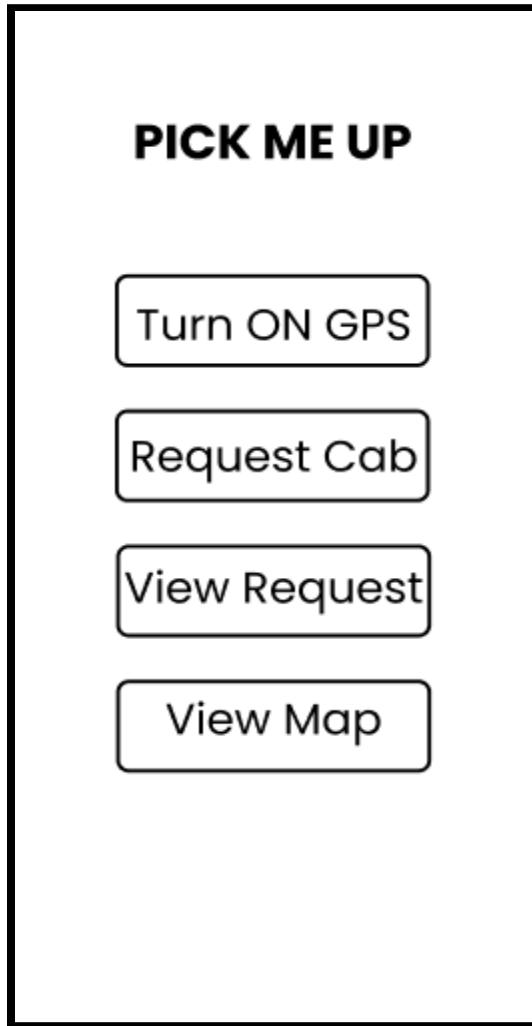
Return S

Using this shortest path tree, the Directions API returns the shortest path between two locations.

Assignment Number	1
Version	01
Print Date	16/5/2021
Page	Page 63 of 104

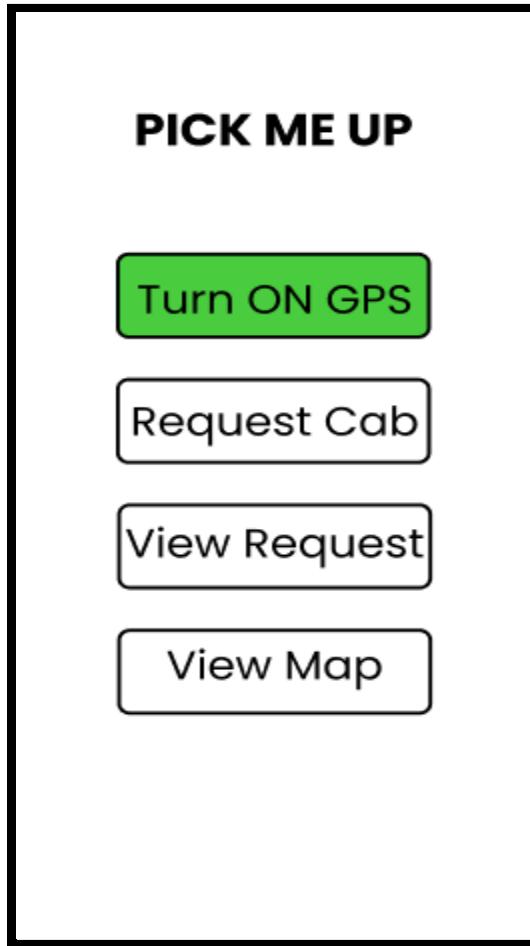
11. User Interface Design and Implementation

After extensive review of the first report, it is decided that the extensive review of the first report, it is decided that the interface will not receive any changes in its design and user effort. The interface will stay the same because it is easy to learn and understand with its simplistic design.



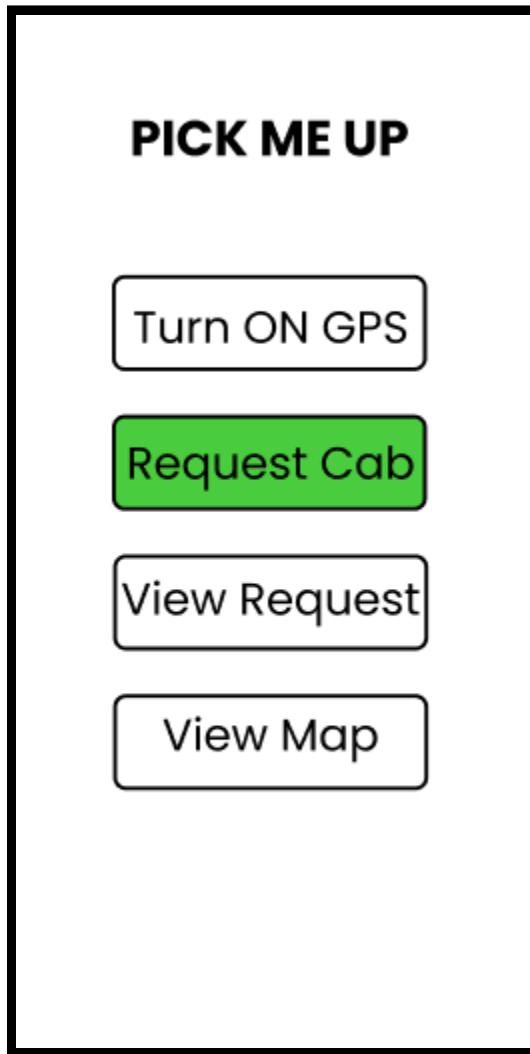
This diagram shows the client homepage after login to the system. The System will be interactive through touch.

Assignment Number	1
Version	01
Print Date	16/5/2021
Page	Page 64 of 104



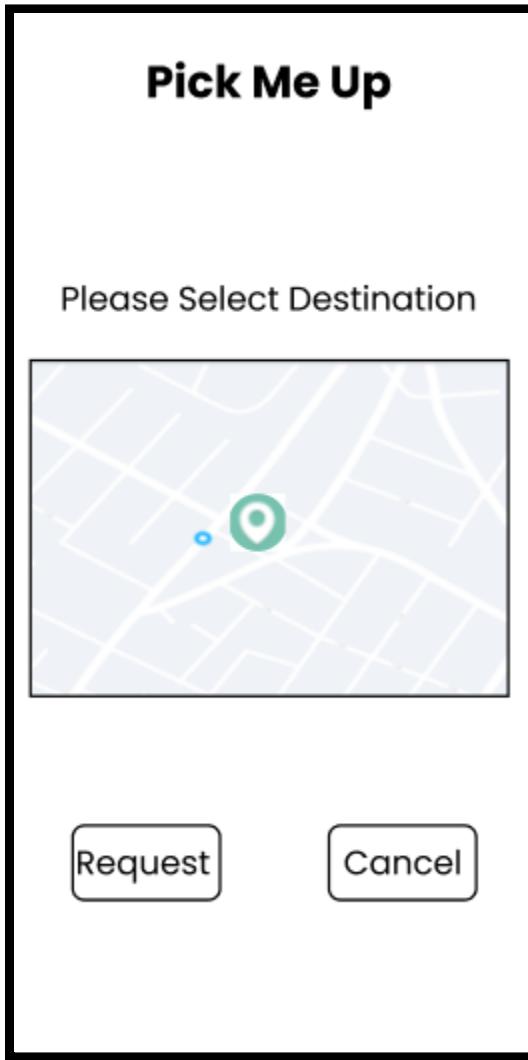
The Green highlights in the diagram will be used to show selection of different options available. The current option highlighted shows that the user GPS is currently on.

Assignment Number	1
Version	01
Print Date	16/5/2021
Page	Page 65 of 104



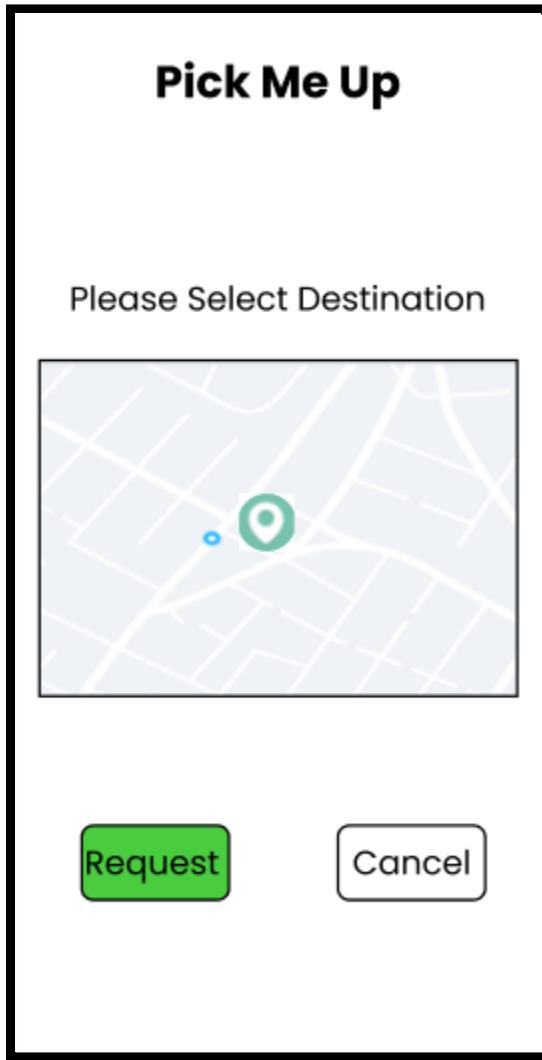
The diagram shows that you can click on the “Request Cab” option, which will bring up another page to confirm the service.

Assignment Number	1
Version	01
Print Date	16/5/2021
Page	Page 66 of 104



The Diagram shows an interactive map that contains your location and allows you to choose your destination. The client can Cancel or confirm the Pickup service on this page.

Assignment Number	1
Version	01
Print Date	16/5/2021
Page	Page 67 of 104



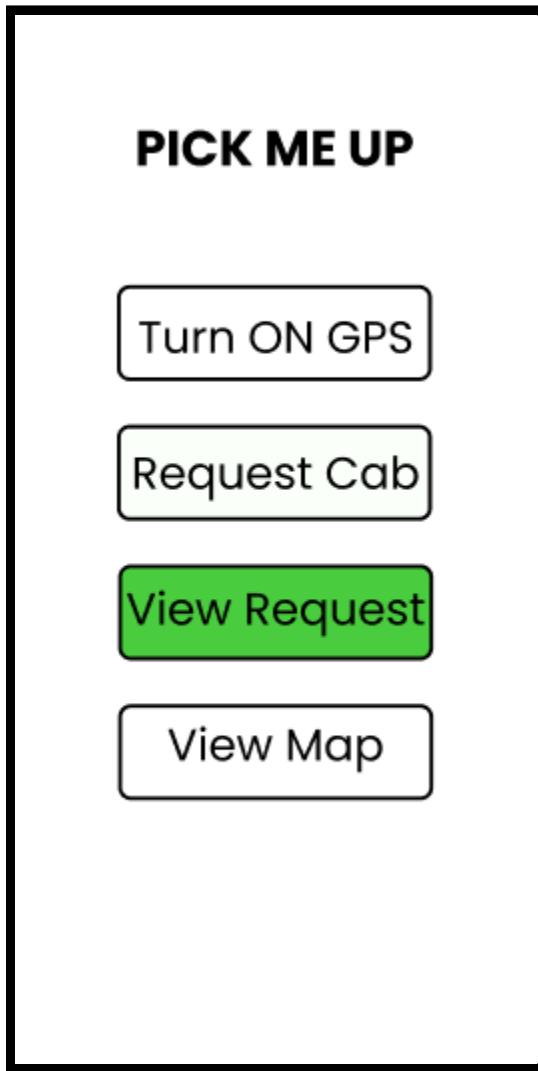
The Diagram shows you that to finish the confirmation, the client must click the request option. This will bring up a message showing you that your Pickup Request has been confirmed.

Assignment Number	1
Version	01
Print Date	16/5/2021
Page	Page 68 of 104



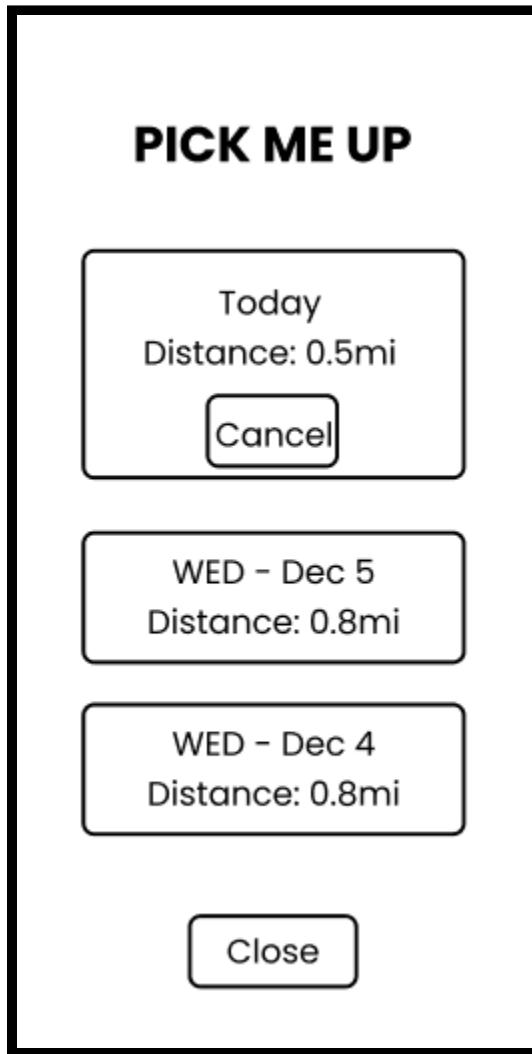
This diagram shows the successful Request Pick up notification message to the customer after choosing a destination and clicking on the Request option. After this notification, the system redirects the user to the homepage.

Assignment Number	1
Version	01
Print Date	16/5/2021
Page	Page 69 of 104



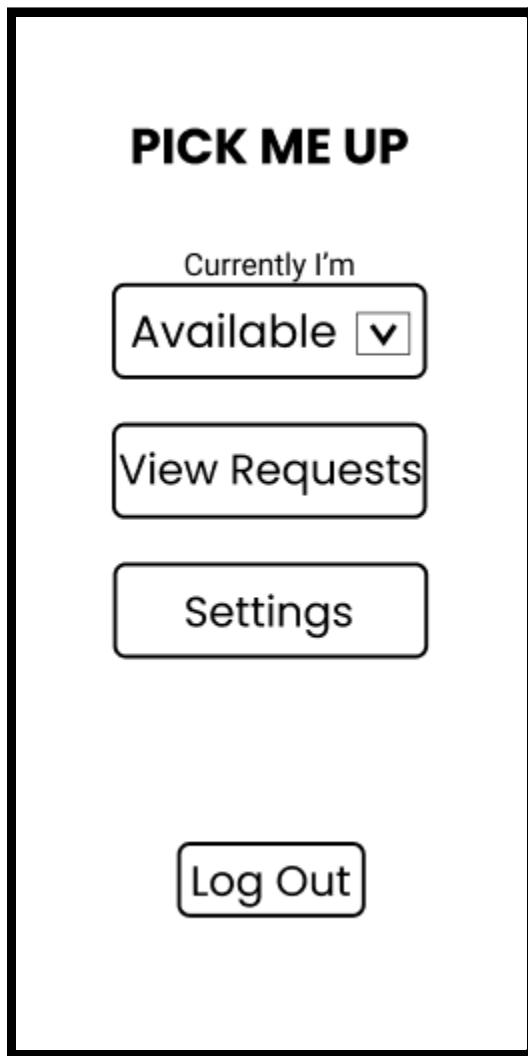
The Diagram shows that the View Request option is being selected, which will bring up another page with the User Pick up Requests.

Assignment Number	1
Version	01
Print Date	16/5/2021
Page	Page 70 of 104



This diagram shows the user all their past Requests and also the user's ongoing request. There are 2 options given, one is the close option which will return the user to the homepage and the other option is to cancel the ongoing Pickup request.

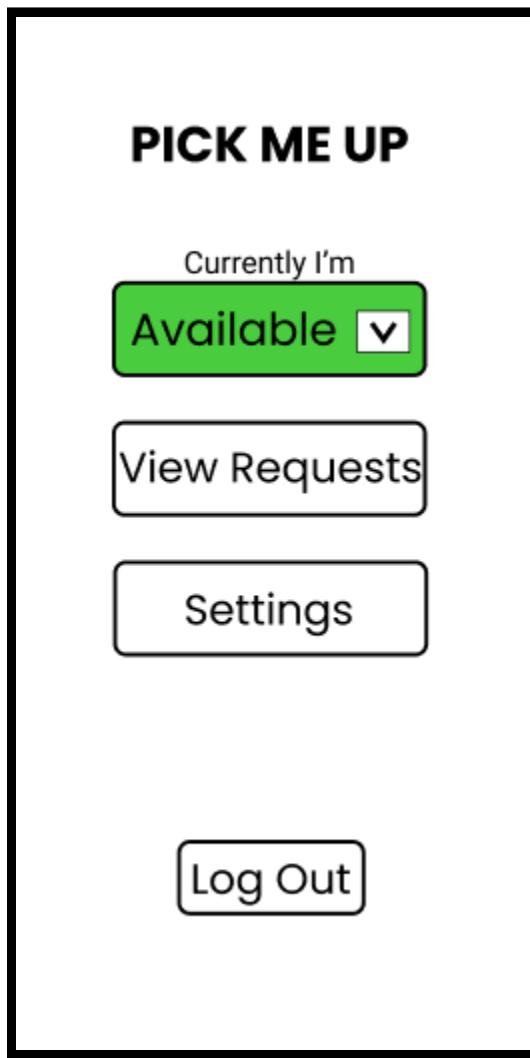
Assignment Number	1
Version	01
Print Date	16/5/2021
Page	Page 71 of 104



This Diagram shows the homepage for the Driver. There are 4 options available for the driver to carry out his task.

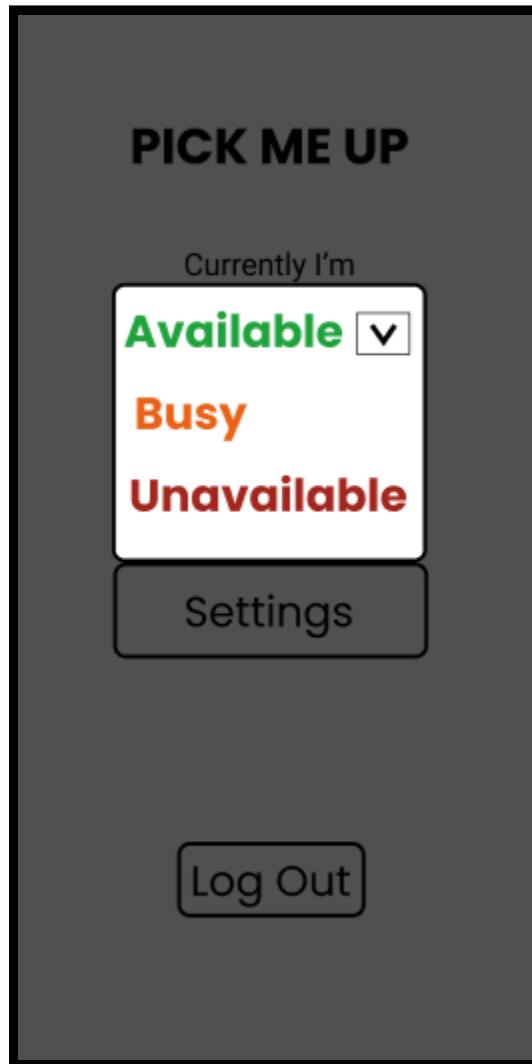
Specs & Design

Assignment Number	1
Version	01
Print Date	16/5/2021
Page	Page 72 of 104



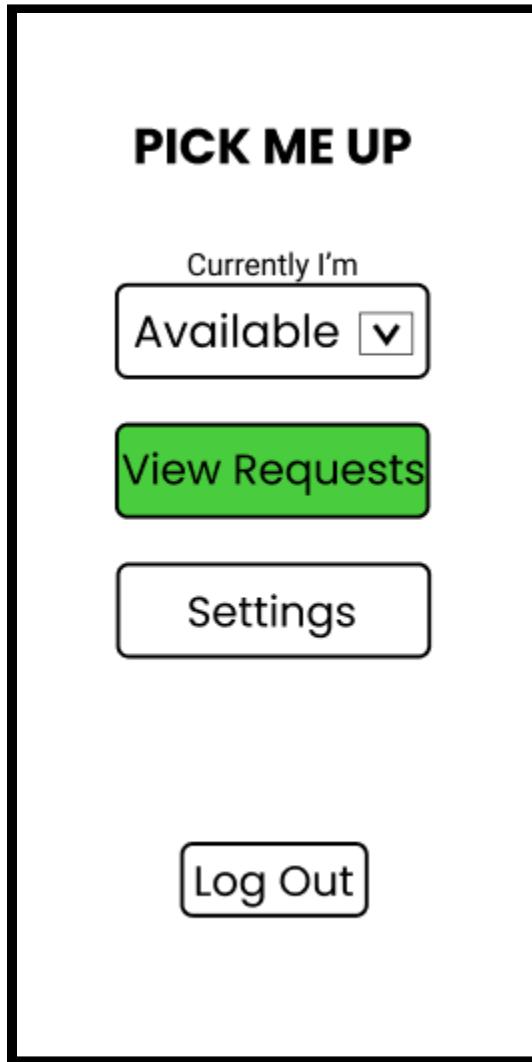
This diagram shows how the user can click on the status options and this will allow the user to set if they're busy, available or not.

Assignment Number	1
Version	01
Print Date	16/5/2021
Page	Page 73 of 104



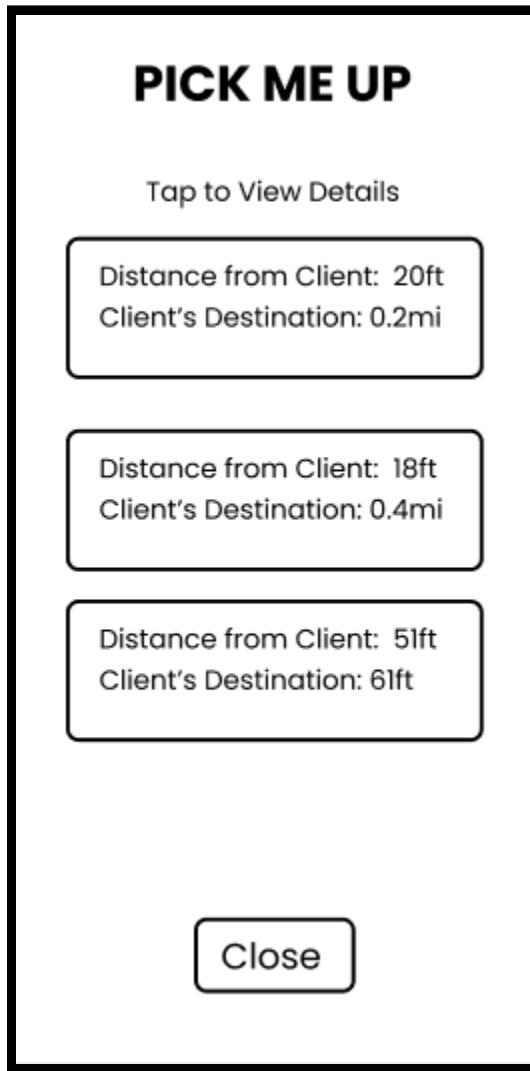
This diagram shows the 3 options available when the user clicks on their status and decides if they are available to work or not.

Assignment Number	1
Version	01
Print Date	16/5/2021
Page	Page 74 of 104



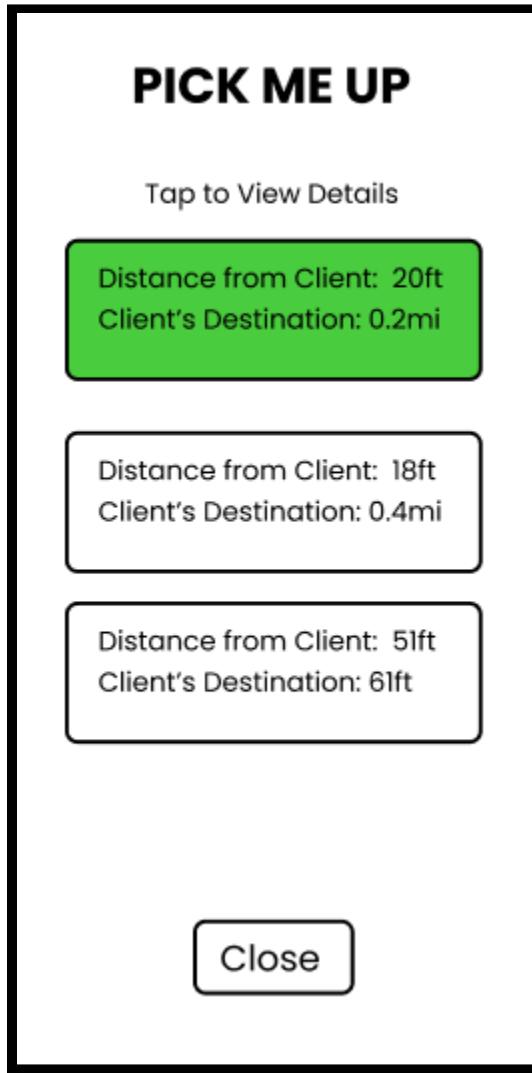
This diagram shows you that the option view Request can be clicked on. This will carry the user to another page showing all of the available requests.

Assignment Number	1
Version	01
Print Date	16/5/2021
Page	Page 75 of 104



This diagram shows the user all of the available Pick up requests. The user can then choose a request and get a more detailed view of the request or go back to the home page.

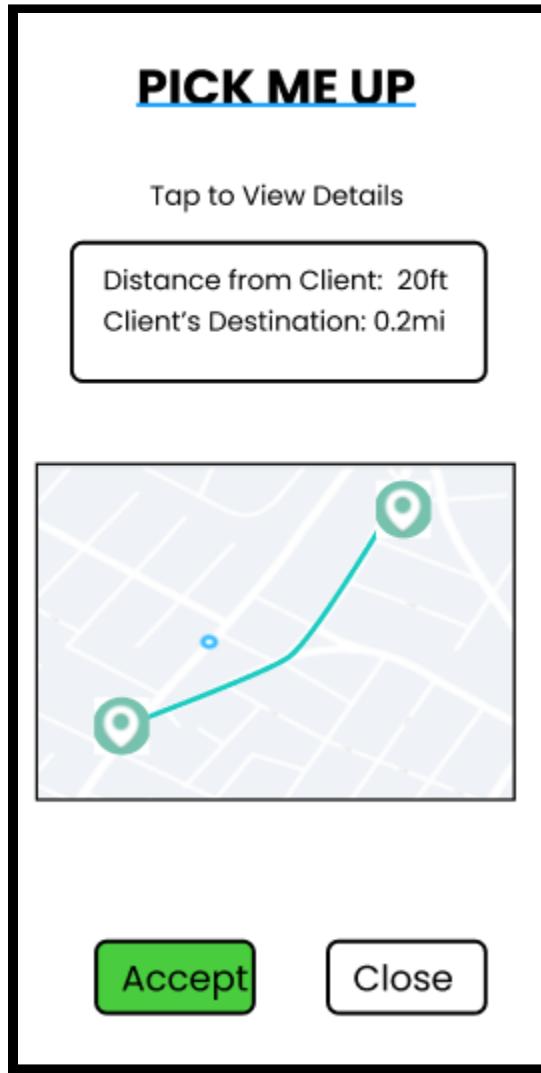
Assignment Number	1
Version	01
Print Date	16/5/2021
Page	Page 76 of 104



The diagram shows that a Pick up request can be clicked on to see more information about the Request.

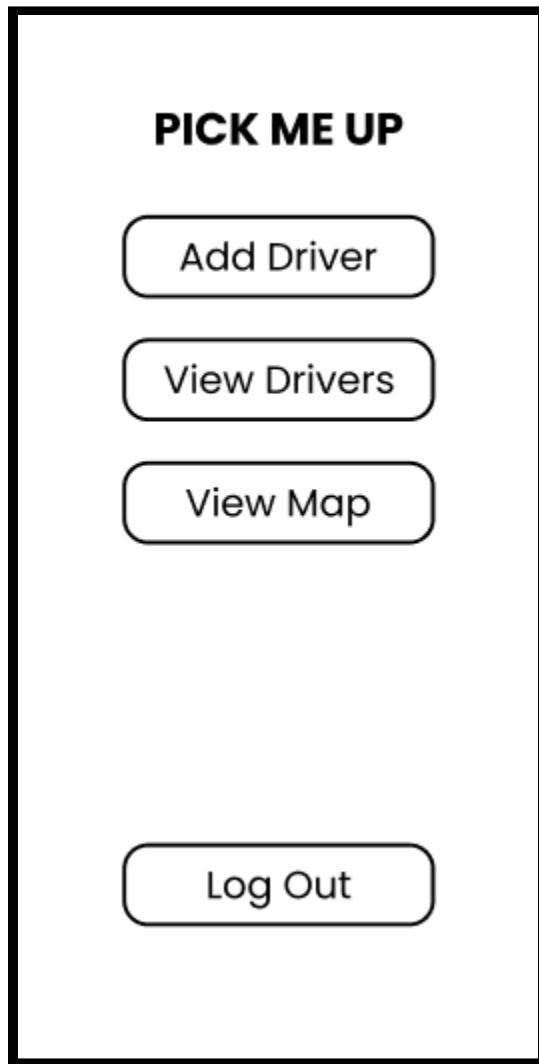
Clicking on a Pick up Request option will show another page with more information and a map.

Assignment Number	1
Version	01
Print Date	16/5/2021
Page	Page 77 of 104



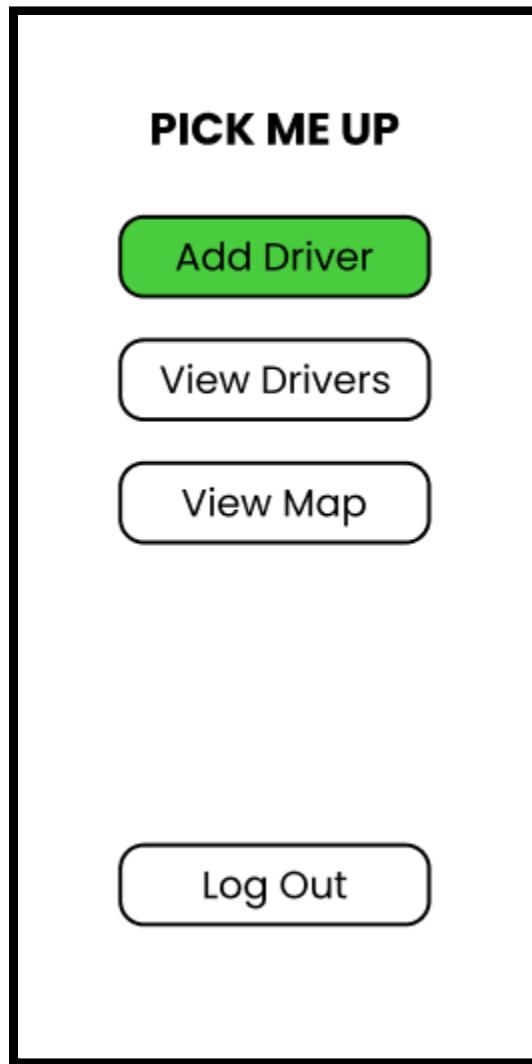
This Diagram shows the page after a user decides to view the Request list and chooses a request. This map will show the client distance from the current user location and there is an option for the user to click to accept this request.

Assignment Number	1
Version	01
Print Date	16/5/2021
Page	Page 78 of 104



This Diagram shows the interface of the homepage of the administrator. The administrator's main job is to manage the Drivers information on the database.

Assignment Number	1
Version	01
Print Date	16/5/2021
Page	Page 79 of 104



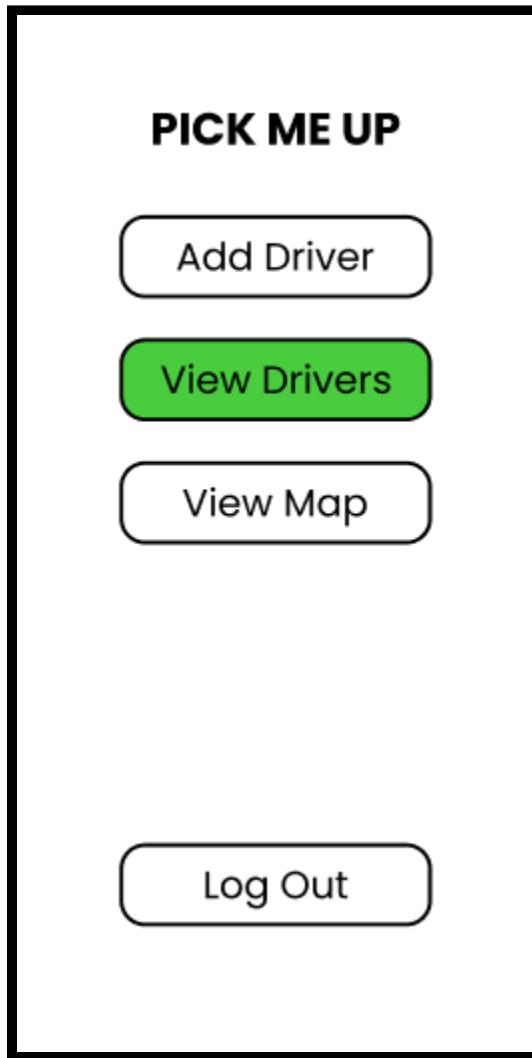
This Diagram shows the selection of the option add drivers, which will allow the administrator to enter another driver into the database. Choosing this option will carry the user to another page.

Assignment Number	1
Version	01
Print Date	16/5/2021
Page	Page 80 of 104

Add New Driver

This diagram shows what information is needed to enter a new driver into the database. Clicking on the Add button will complete this action, you can also click cancel and it will return the user to the homepage.

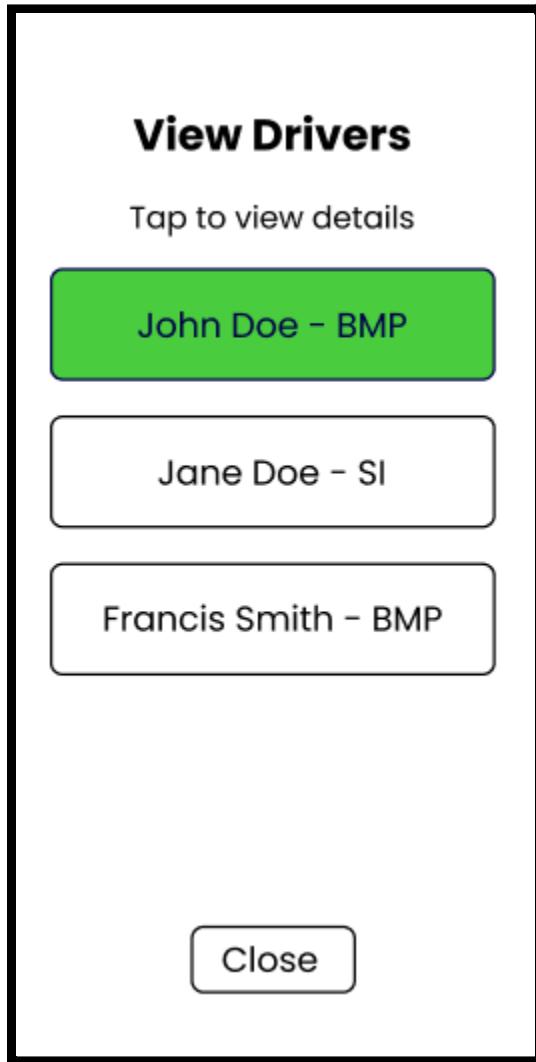
Assignment Number	1
Version	01
Print Date	16/5/2021
Page	Page 81 of 104



This Diagram shows the option View Drivers being selected to view all the Drivers in the database.

Clicking on this option will bring up another page to view all drivers in the database.

Assignment Number	1
Version	01
Print Date	16/5/2021
Page	Page 82 of 104



This Diagram shows the page after clicking the view driver option on the homepage. All drivers in the database are shown onto this list. You can click on a driver option to view their information on another page.

Assignment Number	1
Version	01
Print Date	16/5/2021
Page	Page 83 of 104

The diagram illustrates a mobile-style interface titled "Driver Details". At the top left is a back arrow icon, and at the top center is the title "Driver Details". Below the title are six input fields containing driver information: "John" (first name), "Doe" (last name), "16 Street, Bmp" (address), "667 2109" (phone number), "CY-12548" (license plate), and "C-54932" (another identifier). In the center of the screen below these fields is a rating section displaying "Rating: 4.5" followed by a yellow star icon. At the bottom of the screen are two buttons: "Edit" on the left and "Delete" on the right.

This Diagram shows specific driver information after clicking on them in the View Driver list page. You are given 3 options to either edit the driver, delete the driver or go back to the homepage. The Driver rating is also shown on this page.

Assignment Number	1
Version	01
Print Date	16/5/2021
Page	Page 84 of 104

The diagram illustrates a mobile application interface titled "Driver Details". At the top left is a back arrow icon labeled "Back". The title "Driver Details" is centered at the top. Below the title are six input fields containing driver information: "John", "Doe", "16 Street, Bmp", "667 2109", "CY-12548", and "C-54932". Below these fields is a rating section displaying "Rating: 4.5" followed by a yellow star icon. At the bottom are two buttons: a green "Edit" button and a white "Delete" button.

This diagram shows that the edit option is being clicked. This will bring up another page to edit the driver information.

Specs & Design

Assignment Number	1
Version	01
Print Date	16/5/2021
Page	Page 85 of 104

Driver Details

John

Doe

16 Street, Bmp

667 2109

CY-12548

Please Fill Field

Rating: 4.5 ★

Save ✓ Cancel

This diagram shows the interface of how a driver information is edited. Every field must be filled out with valid data or the system will prompt an error message. You can then save the newly updated information or just cancel the editing.

Assignment Number	1
Version	01
Print Date	16/5/2021
Page	Page 86 of 104

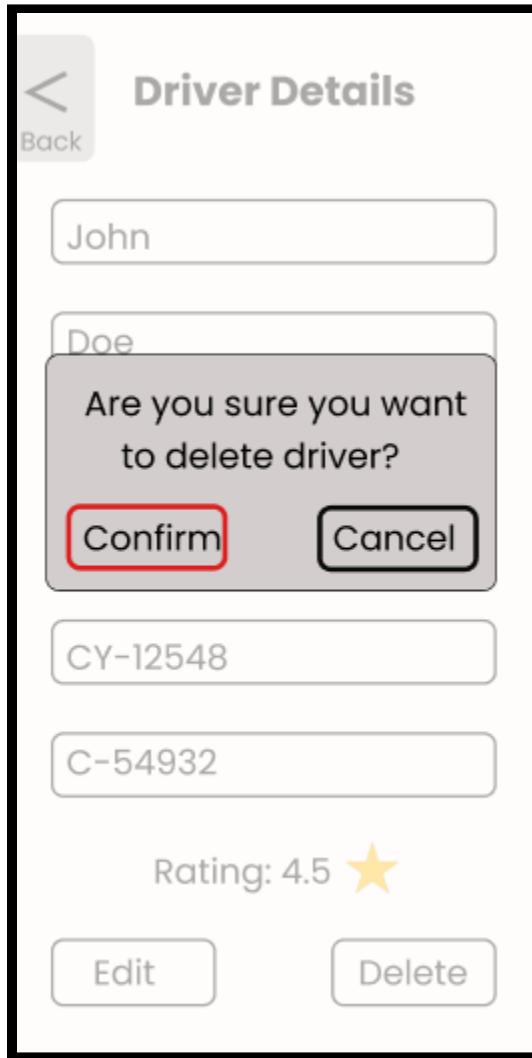
The image shows a mobile-style application interface titled "Driver Details". At the top left is a back arrow icon and the title "Driver Details". Below the title is a "Back" button. The main area contains six input fields, each with rounded corners and a thin black border. The fields contain the following text:

- John
- Doe
- 16 Street, Bmp
- 667 2109
- CY-12548
- C-54932

Below these fields is a rating section with the text "Rating: 4.5" followed by a yellow star icon. At the bottom are two buttons: a white "Edit" button on the left and a green "Delete" button on the right.

This diagram shows the delete option being selected. This will bring up a message box to confirm if the user wants to delete the driver information from the database.

Assignment Number	1
Version	01
Print Date	16/5/2021
Page	Page 87 of 104



This Diagram shows the message box prompting the user after an attempt to delete the driver information from the database. The user can click confirm to complete this action or just press cancel to abort it.

When designing the system, we focus mostly on the ease of use aspect of a system. We incorporated touch interaction, where navigating the system is so straight forward because the system only has big clickable

Specs & Design

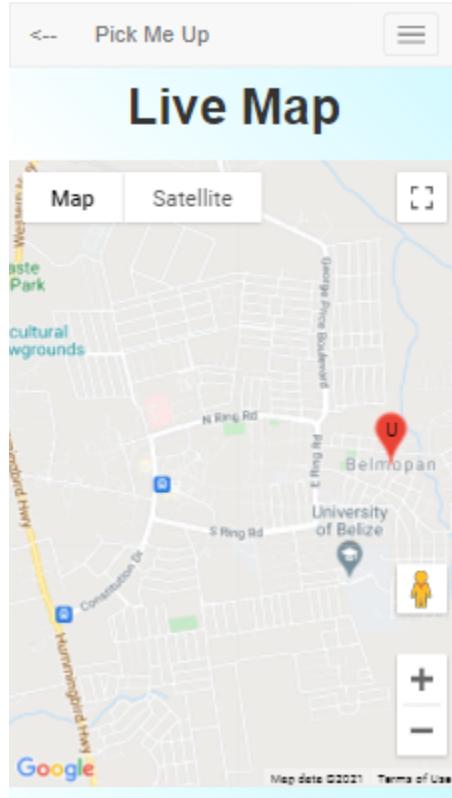
Assignment Number	1
Version	01
Print Date	16/5/2021
Page	Page 88 of 104

buttons with very descriptive labels on them. For example, to order a Pickup, it only takes a click of the Request button and a click on the map to decide your destination. We decided that signing up is a big issue in privacy and is also time consuming, so the minimum requirement to use our system is to leave on your GPS. The other main users

we focus on are the drivers, the drivers can pick up any client they choose to pick up from a list or requests and we also provide the driver with basic information like the pickup spot, destination and a shortest path similar to services of google maps.

Assignment Number	1
Version	01
Print Date	16/5/2021
Page	Page 89 of 104

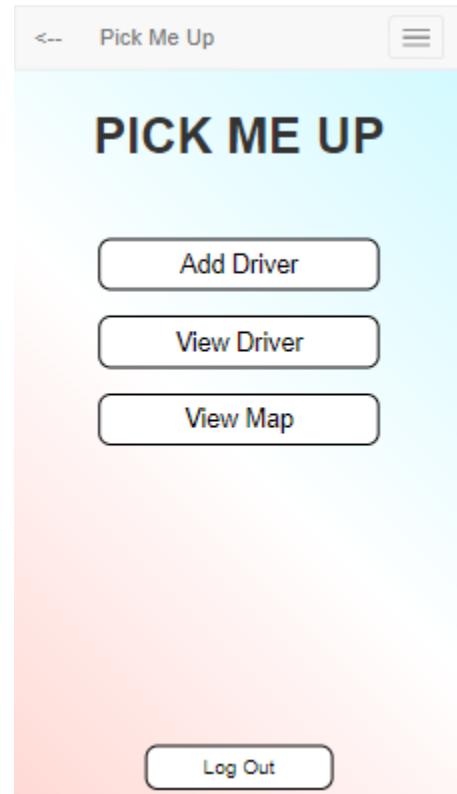
Final Implementation Screenshots



This is a screenshot of the final implementation live map. Containing the map as well as a navigation bar for page traversal. The “U” represents the current user location, also, as it can be seen in the screenshot, no taxis are currently available.

Specs & Design

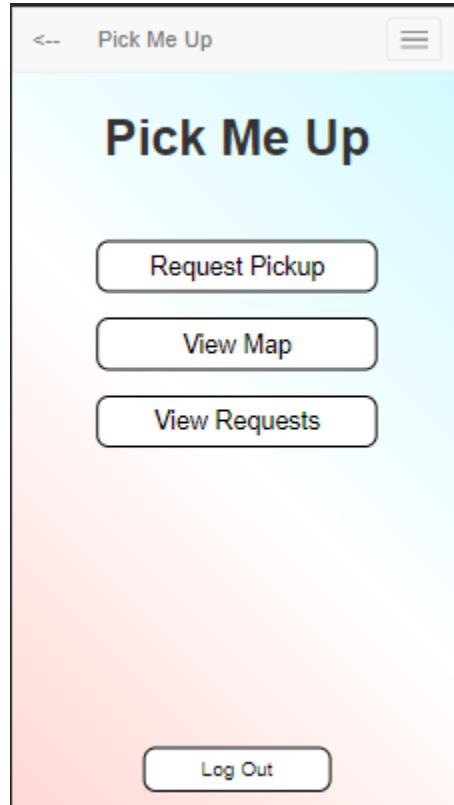
Assignment Number	1
Version	01
Print Date	16/5/2021
Page	Page 90 of 104



Final Implementation screenshot showing the initial drivers page. Also contains a navigation bar, but remains true to the initial idea, as it simply displays the information necessary.

Specs & Design

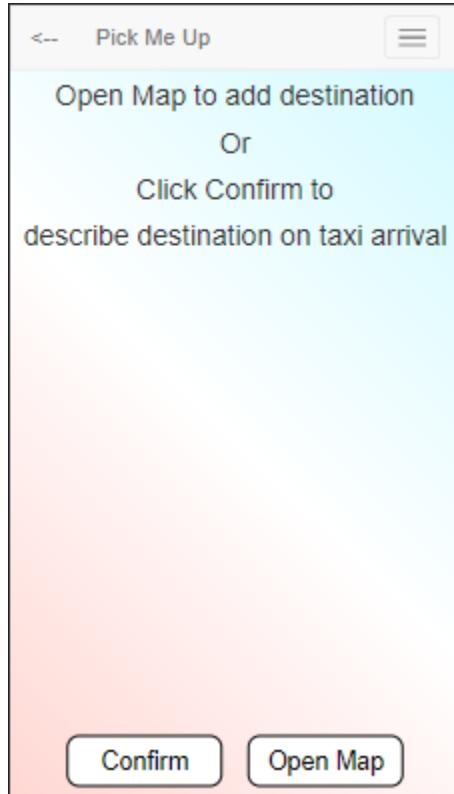
Assignment Number	1
Version	01
Print Date	16/5/2021
Page	Page 91 of 104



Final Implementation of initial pickups page. Once again, remaining true to the initial concept.

Specs & Design

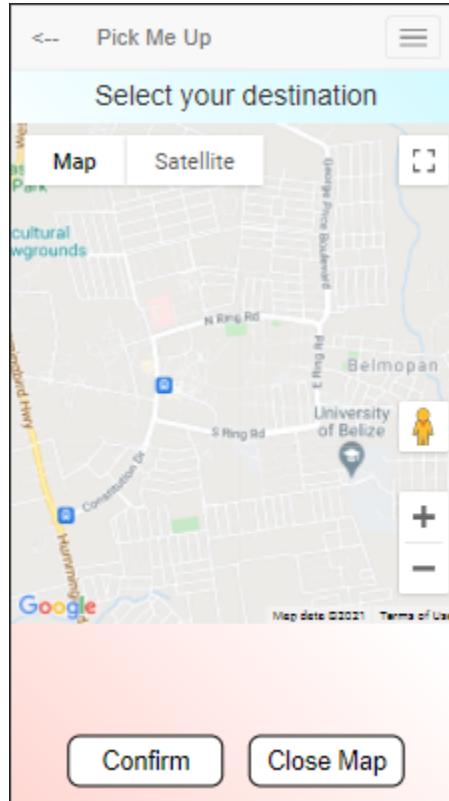
Assignment Number	1
Version	01
Print Date	16/5/2021
Page	Page 92 of 104



Final Implementation of the initial creation of a pickup. Though it requires more effort on the part of the user, considering the target audience, Belizeans. We decided to have the map closed initially, and have the user open it to select their final destination. The selection of a destination is optional due to some users preferring for their final destination to not be logged in testing.

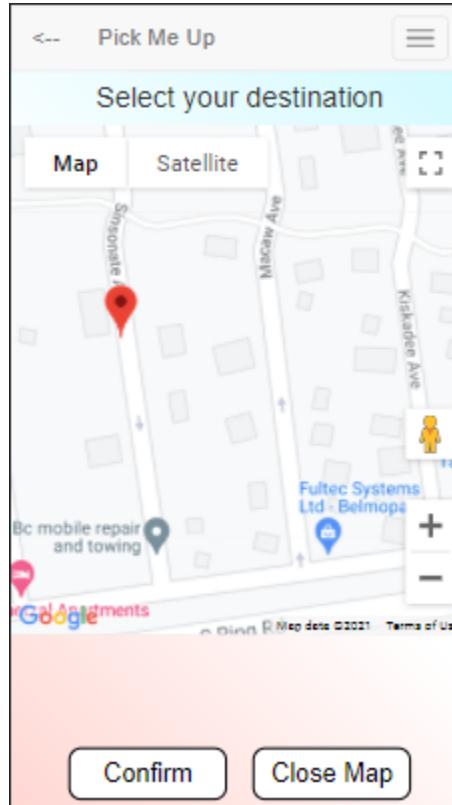
Specs & Design

Assignment Number	1
Version	01
Print Date	16/5/2021
Page	Page 93 of 104



Final implementation of the select destination segment, before a destination is chosen (when the “Open Map” button is clicked/tapped).

Assignment Number	1
Version	01
Print Date	16/5/2021
Page	Page 94 of 104



Final implementation of the destination segment. The map is currently zoomed in with a location chosen.

Comments:

It was found that the initial concept was easy to learn and understand, however, had limitations when considering traversal between different major aspects of the app. As a result, a navigation bar was added to more closely follow traditional navigation as well as provide additional functionality in terms of navigation. Aside from this, the initial concept was followed as closely as possible as can be seen in the above final implementation of the design.

Assignment Number	1
Version	01
Print Date	16/5/2021
Page	Page 95 of 104

12. Design of Tests

Test Cases 1 , 2 , 3 and 4 will be responsible to test the ability of the administrator to add drivers (UC-1),update drivers information (UC-3), view Taxi information (UC-5) and delete drivers information (UC-2) from the database.

Test Case 1	TC-1
Use case being used:	UC-1- Add Driver
Criteria for Success/Fail:	Test is a success if the administrator can add a new driver with all required information.
Input Data:	Text input, Date, integer(valid) Char/double(invalid)
Test Procedure:	<p>Expected Result:</p> <p>Create a connection with database</p> <p>Step 1: Call function “add_driver(fullname, dateofbirth, socialsecurity, licenseplate)” with invalid fullname data.</p> <p>Step 2: Call function “add_driver(fullname, dateofbirth, socialsecurity, licenseplate)” with invalid dateofbirth data.</p> <p>Step 3: Call function “add_driver(fullname, dateofbirth, socialsecurity, licenseplate)” with invalid socialsecurity data.</p> <p>Step 4: Call function “add_driver(fullname, dateofbirth, socialsecurity, licenseplate)” with invalid licenseplate data.</p> <p>Step 5: Call function “add_driver(fullname, dateofbirth, socialsecurity, licenseplate)” with valid data for all parameters.</p>
	<p>Success</p> <p>Fail - Display an error message for invalid data: Prompts the user for another input</p> <p>Fail - Display an error message for invalid data: Prompts the user for another input.</p> <p>Fail - Display an error message for invalid data: Prompts the user for another input.</p> <p>Fail - Display an error message for invalid data: Prompts the user for another input.</p> <p>Success- A new driver is added to the database.</p>

Specs & Design

Assignment Number	1
Version	01
Print Date	16/5/2021
Page	Page 96 of 104

Test Case 2	TC-2
Use case being used:	UC-2-Delete Driver
Criteria for Success/Fail:	The test is a success if the administrator can remove driver information from the database.
Input Data:	none
Test Procedure:	Expected Result:
Create a connection with database	Success
Step 1: Call function “deleteDriver()”	Success- Selected driver is deleted from the database

Test Case 3	TC-3
Use case being used:	UC-3- Update Driver
Criteria for Success/Fail:	The test is a success if the administrator can update the driver information from the database.
Input Data:	Text input, Date, integer(valid) , Char/double(invalid)
Test Procedure:	Expected Result:
Create a connection with database	Success
Step 1: Call function “updateDriver(fullname, dateofbirth, socialsecurity, licenseplate)” with invalid fullname data.	Fail - Display an error message for invalid data: Prompts the user for another input
Step 2: Call function “updateDriver(fullname, dateofbirth, socialsecurity, licenseplate)” with invalid dateofbirth data.	Fail - Display an error message for invalid data: Prompts the user for valid input data.
Step 3: Call function “updateDriver(fullname,	

Specs & Design

Assignment Number	1
Version	01
Print Date	16/5/2021
Page	Page 97 of 104

<p>dateofbirth, socialsecurity, licenseplate)" with invalid socialsecurity data.</p> <p>Step 4: Call function "updateDriver(fullname, dateofbirth, socialsecurity, licenseplate)" with invalid licenseplate data.</p> <p>Step 5: Call function "updateDriver(fullname, dateofbirth, socialsecurity, licenseplate)" with valid data in all parameters.</p>	<p>Fail - Display an error message for invalid data: Prompts the user for valid input data.</p> <p>Fail - Display an error message for invalid data: Prompts the user for valid input data.</p> <p>Success- Driver information is updated in the database.</p>
---	---

Test Case 4	TC-4
Use case being used:	UC-5 view Taxi
Criteria for Success/Fail:	The test is a success if the administrator can view the taxi driver information.
Input Data:	Text input (valid), char/integer (invalid)
Test Procedure:	Expected Result:
Create a connection with database Step 1: Call function "viewTaxi (drivername)" with empty parameters. Step 2: Call function "viewTaxi(drivername)" with invalid drivername data. Step 3: Call function "viewTaxi(drivername)" with valid drivername data.	<p>Success</p> <p>Success- Displays a list of all Driver information.</p> <p>Fail- Display an empty list. Prompts the user to input valid data.</p> <p>Success- Displays the Driver information</p>

Specs & Design

Assignment Number	1
Version	01
Print Date	16/5/2021
Page	Page 98 of 104

Test case 5 will test the precision of location tracking of the client and the availability of service when the client requests a “Pickup”. (UC-6) Request Pick-up does this with the help of (UC-13) viewPickup

Test Case 5	TC-5
Use case being used:	UC-6 & UC-13 Request Pick-up & viewPickup
Criteria for Success/Fail:	The test is a success if the user can request a service.
Input Data:	double(valid), integer/char (invalid)
Test Procedure:	Expected Result:
Step 1: Call function “RequestPickup()” Step 2: Call function “viewPickup()”.	Success- A pickup order is created for the client. Success- The client pickup location has been set to the order.

Test Case 6 will test the functionality of a driver selecting and accepting to provide services to a client. (UC-18) accept Request does this with the help of (UC-17) List Request and (UC-13) View Pick Up.

Test Case 6	TC-6
Use case being used:	UC-18 & UC-17& UC-13-Accept Request & ListRequest & viewPickUp
Criteria for Success/Fail:	The test is a success if the driver can accept a Pickup Request.
Input Data:	Touch, Integer(valid), Char(invalid)
Test Procedure:	Expected Result:
Step 1: Call function “ListRequest()”.	Success- A list of Pickup Requests is shown. Success- Displays the location of the selected

Specs & Design

Assignment Number	1
Version	01
Print Date	16/5/2021
Page	Page 99 of 104

Step 2: Call function “viewPickUp()”.	client.
Step 3: ”Call function “AcceptRequest()”.	Success- The selected Request is marked as taken.

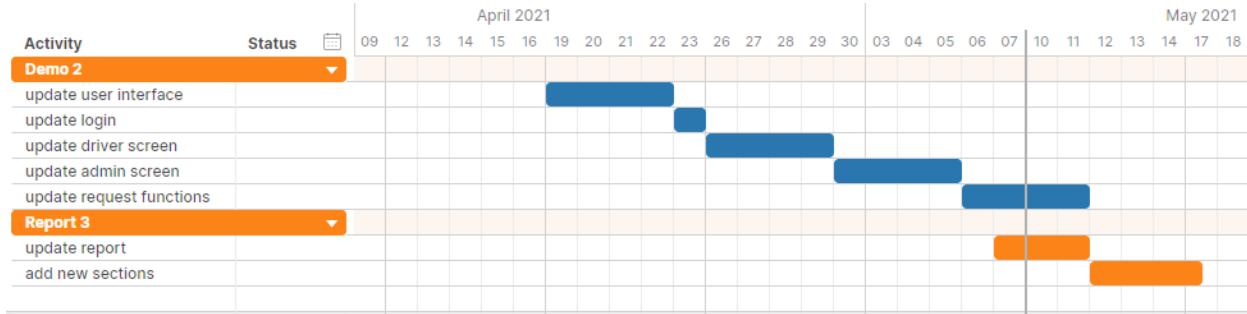
Test Case 7 will test the functionality of the user to sign into the system and Display an interface depending on the authority of the account. (UC-4) Login is incharge of this Test case.

Test Case 7	TC-7
Use case being used:	UC-4 Login
Criteria for Success/Fail:	The test is a success if the user can login to their respective home page.
Input Data:	Text Input (valid)
Test Procedure:	Expected Result:
Step 1: Call function “login(Username, password)” with invalid username data.	Fail- Display an error message for invalid input data. Prompts the user to input valid data.
Step 2: Call function “login(Username, password)” with invalid password data.	Fail- Display an error message for invalid input data. Prompts the user to input valid data.
Step 3: Call function “login(username, password)” with valid data for all parameters.	Success- Display Homepage based on credentials.

Assignment Number	1
Version	01
Print Date	16/5/2021
Page	Page 100 of 104

13. History of Works

Final Phase:



Key Accomplishments:

- Client can do requests and optionally select their drop off location
- A driver can claim and unclaim a pickup request
- Admin can add, update and remove drivers
- Anyone can see the live tracking of taxi drivers
- The system and show the path - from driver initial to pickup
- The system and show the path - from pickup to destination
- Anyone who wants to do requests needs to sign in with their Google Account
- Optional Rating System
- System automatically updates drivers ratings

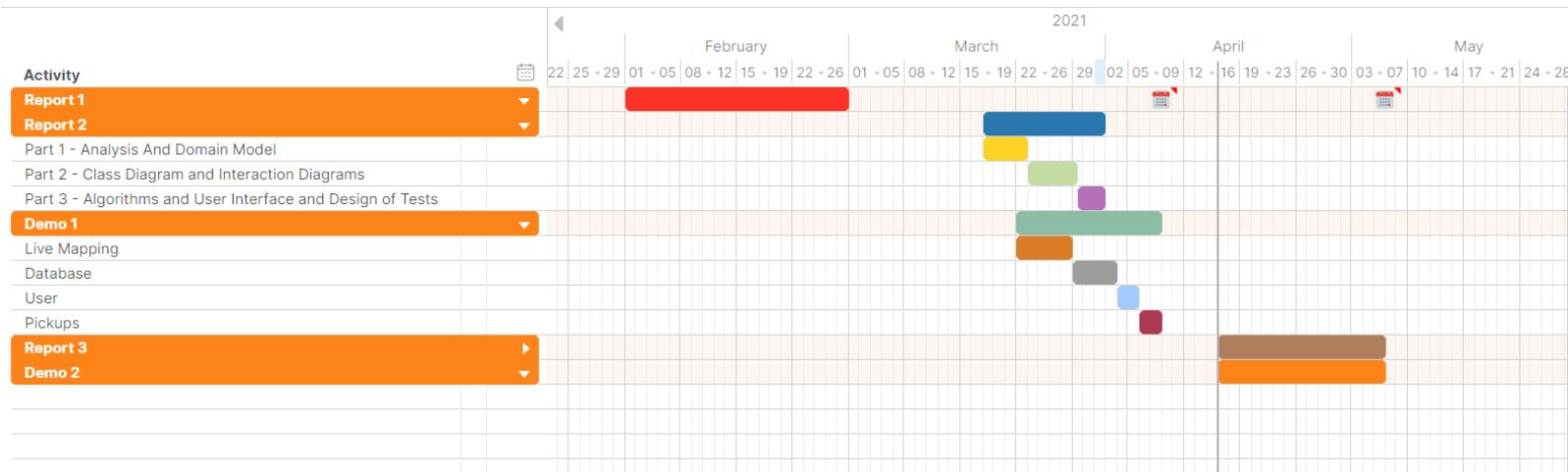
Future Works:

- Detect and deny spams - track those that do requests and never appear for their pickup.
- Implement sms services to inform the clients that their request has been approved
- Add restrictions - prevent/limit request from places known for robbery
- Add a service to charge the client with their credit card
- Display picture and license plate of the driver's vehicle
- Have manipulate capacity for a pick up
- Have a report driver or car
- Road block - update map during
- favorite a taxi
- book a taxi
-

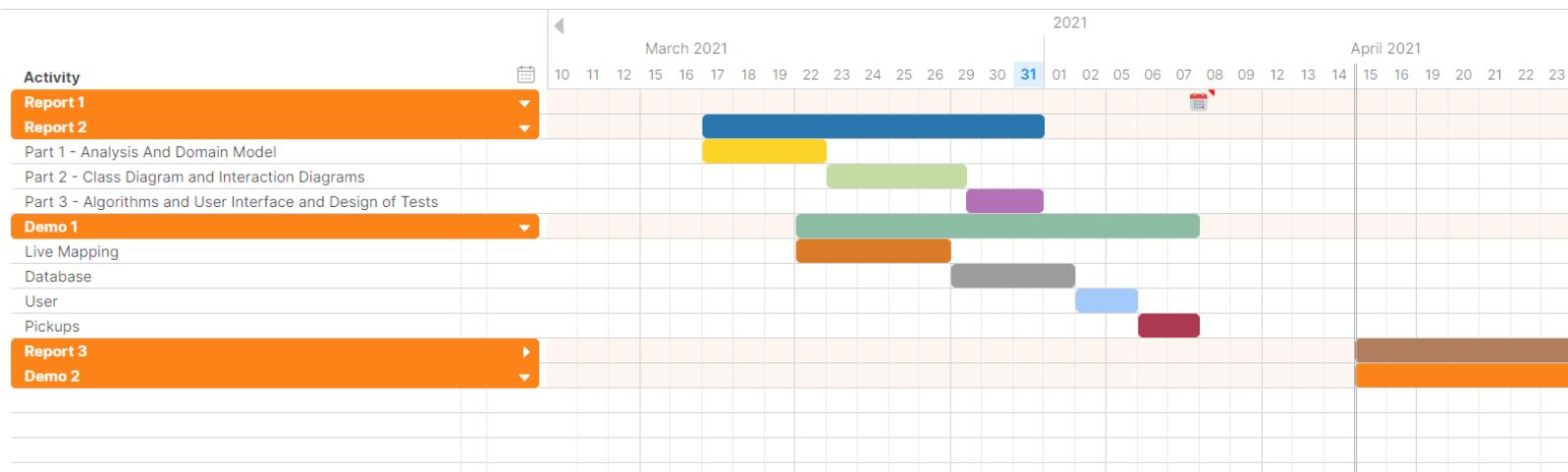
Specs & Design

Assignment Number	1
Version	01
Print Date	16/5/2021
Page	Page 101 of 104

Initial Plan of Work



Showing Weekly Plan of Work



Showing Daily Plan of Work between 10 March and April 23.

Can be seen dynamically at <https://plan.tomsplanner.com/public/se-pick-me-up-group-one-plan-of-word>.

Assignment Number	1
Version	01
Print Date	16/5/2021
Page	Page 102 of 104

Effort Breakdown Table

	Team Member Names				
	Pablo Cawich	Hector Castellanos	Osmer Escarraga	Austin Shaw	Michael Sanchez
Customer Statement of Requirements	0%	0%	0%	0%	100%
Glossary of Terms	0%	0%	0%	0%	100%
System Requirements	20%	30%	30%	10%	10%
Functional Requirements Specification	10%	46%	30%	7%	7%
Effort Estimation	0%	0%	100%	0%	0%
Domain Analysis	0%	0%	100%	0%	0%
Interaction Diagrams	0%	50%	0%	0%	50%
Class Diagram and Interface Specification	0%	20%	0%	80%	0%
System Architecture and System Design	60%	20%	0%	20%	0%
Algorithms and Data Structures	90%	10%	0%	0%	0%
User Interface Design and Implementation	0%	80%	20%	0%	0%

Specs & Design

Assignment Number	1
Version	01
Print Date	16/5/2021
Page	Page 103 of 104

Design of Tests	0%	100%	0%	0%	0%
History of Work, Current Status, and Future Work	0%	0%	100%	0%	0%
References	50%	0%	50%	0%	0%
Project Management	33%	33%	33%	0%	0%

Assignment Number	1
Version	01
Print Date	16/5/2021
Page	Page 104 of 104

14. References

Lurette H. (2020, August 19). How does Google Maps find the shortest path? Retrieved from
<https://blog.devgenius.io/how-does-google-maps-find-the-shortest-path-44be4b1c74c6?gi=f1545c88fd0b>

<https://www.geeksforgeeks.org/unified-modeling-language-uml-sequence-diagrams/#:~:text=A%20sequence%20diagram%20simply%20depicts,objects%20in%20a%20system%20function.>

<https://www.omg.org/spec/OCL/2.0/PDF>