

# Moving Terps Project Plan

**Team Members:** Krishik Nataraj Gowda, Raghul Pandi, Kai-Hsiang Lin, Jingru Liu

**Client:** The Department of Transportation Services (DOTS), The University of Maryland, College Park (UMD)

<b>Executive Summary</b>	<b>1</b>
<b>Introduction</b>	<b>2</b>
<b>System Planning Phase Report</b>	<b>2</b>
1.0 Introduction of the client	2
1.1 Opportunity	2
1.2 Project Objective	3
1.3 Project Scope	3
2.0 Project Methodology	4
2.1 Justification	4
3.0 Project Plan	4
<b>System Analysis Phase Report</b>	<b>6</b>
4.0 Requirements	6
4.1 Business requirements	6
4.2 User requirements	6
4.3 Functional requirements	7
Process	7
Information	7
4.4 Non-functional requirements	8
Operational	8
Performance	8
Security	8
Cultural and political	8
5.0 Data Flow Diagrams	8
5.1 Context Diagram	8
5.2 Level 0 Diagram	10
5.3 Level 1 Process Tables	11
5.3.1 Account Management	11
5.3.2 Parking Service	11
5.3.3 Shuttle Bus Service	12
5.3.4 Notification	13
5.3.5 Third-party online payment system	13
5.3.6 Add to User Favorite	13
5.4 Entity Relationship Diagram	14
5.5 Swimlane Processes Diagrams	15

5.5.1 Account Management System	15
5.5.2 Parking Service	15
Parking Lot and Route Searching	15
Vehicle Information	16
Permit Purchase	16
Citation Payment	16
5.5.3 Shuttle Bus Service	16
Search a Route	16
Near-by Stations	17
Route Planning	17
Track your ride	18
5.5.4 Notification	18
5.5.5 Online Payment System	18
5.5.6 Add to User Favorites	19
<b>System Designing Phase</b>	<b>20</b>
6.0 Physical ERD	20
7.0 User Interface	20
8.0 System Design Architecture	22
9.0 Testing Plan/Documentation	23
9.1 Stub testing	23
9.2 Unit testing	23
9.3 System testing	25
9.4 Acceptance testing	25
9.5 Integration testing	25
10.0 Implementation Plan	25
10.1 Objectives	25
10.2 Details and Deliverables	25
10.3 Implementation Plan Schedule	26
11.0 Conclusion	27

## Executive Summary

*Moving Terps* is the proposed mobile application outlined in this proposal. It is aimed to serve as the integration of the current Next Bus shuttle service mobile application and the parking portal offered by the Department of Transportation Services (DOTS) of the University of Maryland, College Park (UMD.) The five major modules of this application will be account management, parking service, shuttle bus service, add to favorites, and notification. The application will also be integrated with third-party online payment systems that allows users to pay for parking permits and citations and a mobile identity verification system that enables users to read shuttle bus information and store ride history information.

The team will be using a waterfall and prototype mixed methodology due to the fact that the service provided by DOTS is relatively static; thus, the requirements of the application are well-defined which can be fulfilled in phases. In addition, since we are trying to replace the legacy application and website provided by DOTS, it is crucial to collect feedback from students and faculties so the team can identify changes and refine before the final implementation. All components of this mobile application will be tested before implementation.

## Introduction

This project aims to integrate transportation services provided by the Department of Transportation Services (DOTS) of the University of Maryland, College Park (UMD) and develop a comprehensive mobile application, *Moving Terps*. It will serve as the main tool for assisting students, faculty, and staff to commute to and travel around the campus.

## System Planning Phase Report

### 1.0 Introduction of the client

The Department of Transport Services (DOTS) is an integral part of University of Maryland (UMD). DOTS is committed to providing safe, cost effective and innovative services that anticipate the needs of our growing community of more than 50,000 students, faculty and staff in the city of College Park and beyond.

The various services provided by DOTS include a free shuttle program for its users, an on- demand NITE ride facility for users living in the vicinity of the campus, Charter for long haul travel, parking system in the university, and sustainable transport facilities like carpool, carshare, electric vehicle charging stations.

### 1.1 Opportunities

The main goal of a university transportation department is to provide cost effective and hassle-free transportation to its users. DOTS UMD has many such facilities to achieve this goal but the main problem that it faces is that this information is spread around multiple applications and websites with no single repository of information which makes it difficult for users to take advantage of this facility.

So, this project gives us an opportunity to integrate all the functions and facilities provided by DOTS into a single mobile application. We also aim to make these facilities more interactive and user friendly for all users by adding new functionalities in the application for shuttle service and for parking.

## 1.2 Project Objectives

The main objective of the project is to present UMD students, faculties, and staff with a transportation service mobile application that is easy to use. It will not only make up for the deficiencies of the current system, but it will also increase the user satisfaction rate by enhancing the user experience. Therefore, it is time for our app, *Moving Terps*, to come to offer by truly connecting with our users and understanding their needs and constraints.

## 1.3 Project Scope

*Moving Terps* is a mobile application which aims to make daily commute and parking facility around the campus more convenient and efficient. The app transforms the most frequently used services provided by the UMD Department of Transportation Services into functions on the interface for easier access. Enrolled students, faculty, and staff will have access to the application by logging into their directory id. As a user-based instead of office-based system, *Moving Terps* display transportation information in a more dynamic and smart way by allowing users to interact with the application and receive recommendations. The deployment of *Moving Terps* should increase users' satisfaction rate and provide much more intuitive and accurate transportation information.

### Functions

1. Accounts: users should be able to log into the application with UMD directory id. The system should be able to identify account types (student/staff/faculty.)
2. Shuttle: users can enter starting location and destination, and the application will recommend bus routes along with real-time bus tracking information (location, arriving time and capacity) on an interactive map. Users can also check nearby stations and the routes passing by.
3. Parking: users should be able to enter starting location and destination, and an interactive map will show parking lot recommendations based on user type together with the detailed parking lot restrictions. Then, by clicking on a recommended parking lot, the system will show the direction to get to that selected parking lot. Additional features under this function include: update personal and vehicle information, pay for permits, citations, and temporary parking, appeal against citations
4. Notification: Notify any impact schedule and current status (specific shuttle and parking lot availability during special events) through "push notification." Users don't need to check email or the impact calendar manually.
5. User Favorites: users are able to add their frequently used bus routes, bus stops, and parking lots as their favorite items.

### Integrations

1. Third-party online payment system to allow students to pay for parking permits, citations, and temporary parking.
2. Mobile identity verification system to allow users utilize mobile devices to scan QR code/tap on an NFC (near-field communication) terminal so that their identity can be verified and the

trip information can be stored in their accounts. An NFC terminal will need to be installed on each bus.

## 2.0 Project Methodology

### 2.1 Justification

Of the various system development methods, a hybrid of waterfall and system prototyping best fits our goal.

This project has well defined requirements that don't change substantially during the development cycle. Additionally, there is privilege of time as most of the services, although not integrated and easy to use, are available across applications and DOTs website. Finally, incorporating system prototype in the design phase, allows for feedback cycles from users to identify changes and refine real requirements before implementation.

## 3.0 Project Plan

WBS	NAME	Duration	Start Date	End date
1	<b>Project</b>	187	2/9/22	8/19/22
1.1	<b>Initiation</b>	31	2/9/22	3/12/22
1.1.1	Preliminary plan	10	2/10/22	2/20/22
1.1.2	Detailed project plan	10	2/20/22	3/2/22
1.1.3	Objective identification	10	3/2/22	3/12/22
1.1.4	Plan finalization	1	3/12/22	3/12/22
1.2	<b>Analysis and Interviews</b>	35	3/13/22	4/17/22
1.2.1	Collect requirement from students	10	3/13/22	3/23/22
1.2.2	Collect requirement from faculty	10	3/23/22	4/2/22
1.2.3	Collect requirement from transportation	10	4/2/22	4/12/22

	office			
1.2.4	Requirement Determination	5	4/12/22	4/17/22
<b>1.3</b>	<b>Design</b>	65	4/18/22	6/22/22
1.3.1	Design plan brainstorm	10	4/18/22	4/28/22
1.3.2	Prototype	20	4/29/22	5/9/22
1.3.3	Database design	10	5/10/22	5/19/22
1.3.4	Interface design	10	5/20/22	5/29/22
1.3.5	Back-End design	10	5/30/22	6/9/22
1.3.6	Documentation	5	6/10/22	6/22/22
<b>1.4</b>	<b>Implementation</b>	39	6/23/22	8/1/22
1.4.1	Evaluate design plan	5	6/23/22	6/27/22
1.4.2	Unit implementation	20	6/28/22	6/18/22
1.4.3	Acquire system access	2	6/19/22	6/20/22
1.4.4	Integrated with different systems	10	6/21/22	6/30/22
1.4.5	Documentation	2	6/30/22	8/1/22
<b>1.5</b>	<b>Test</b>	17	8/2/22	8/19/22
1.5.1	Function Test	5	8/2/22	8/6/22
1.5.2	Requirement Check	5	8/7/22	8/11/22
1.5.3	User Feedback Analysis	5	8/12/22	8/16/22
1.4.4	Documentation	2	8/17/22	8/19/22

# System Analysis Phase Report

## 4.0 Requirements

### 4.1 Business requirements

- Must be a mobile application
- Provide search capabilities for shuttle bus and parking service
- Show optimized search results in an interactive map
- Provide push notification and offer options for signing up
- Integrate with third-party online payment systems
- Capture and store ride information with NFC technology

### 4.2 User requirements

- Users will be able to create an account and login with UMD Directory ID
- Users will be able to update personal information
- Users will be able to sign up for push notifications
- Users will be able to check selected favorite bus routes, stops, and parking lots
- Shuttle Service:
  - Users will be able to search for a specific bus routes
  - Users will be able to search for near-by bus stations
  - Users will be able to enter starting and ending locations to plan for a route
  - Users will be able to save frequently-used bus routes and stops as favorites
  - Users will be able to record ride history by tapping their mobile device on the NFC terminals on the buses
  - Users will be able to review ride history
- Parking Service:
  - Users will be able to search for available parking lots and areas based on input data (user type, arrival time of day, starting location, and destination)
  - Users will be able to add and update vehicle information
  - Users will be able to pay for parking permits and citations
  - Users will be able to search and appeal against citations
  - Users will be able to save frequently used parking lots as favorites

### 4.3 Functional requirements

- Process
  - Software should allow users to search for specific bus routes and display route number & name, traveling direction, starting/terminal stops, all stops of that route, and operational status (running/offline, current location) in an interactive map
  - Software should display search results based on user location in an interactive map
  - Software should allow users enter starting/ending locations and departure/arrival by time, and the software must display optimized route by shortest travel time in an interactive map

- Software should add the selected bus routes or stops and selected parking lots into the “UserFavorite” database
  - Software must automatically estimate the arrival time of each bus at the next stop
  - Software should allow users to search for available parking lots and areas with search filter inputs (user type, arrival time of day, destination) and display search results in an interactive map
  - Software should display detailed information about parking lots when users click on each available parking lots
  - Software should generate an optimized route from the user’s location to the selected parking lot when user click on “direction” in each available parking lot
  - Software should allow users to add and update vehicle information
  - Software should allow users to pay for parking permits and citations
  - Software should allow users to search for parking permits and citation history
  - Software should allow users to review and write appeals on citations
  - Software should allow users to sign up for notifications
  - Software should allow users to edit personal information
  - Software must capture and store ride information into ride history database when users tap their mobile devices on a NFC terminal
  - Software should allow users to review ride history
- Information
  - Software must store all user accounts information
  - Software must capture all bus routes, bus stops, and ride schedule data
  - Software must capture location information of all operating buses
  - Software must store all detailed information about parking lots, parking restrictions, and impact schedule
  - Software must store routes, bus stops, and parking lots marked as favorites by users
  - Software must have an interactive map that display all data about parking lots and spaces, shuttle bus stops, and optimized routes
  - Software must store ride history
  - Software must capture user information
  - Software must store information about purchased permits, vehicle, and citations
  - Software must store citation appeal letters posted by users
  - Software must store comments of appeal letters posted by system administrators

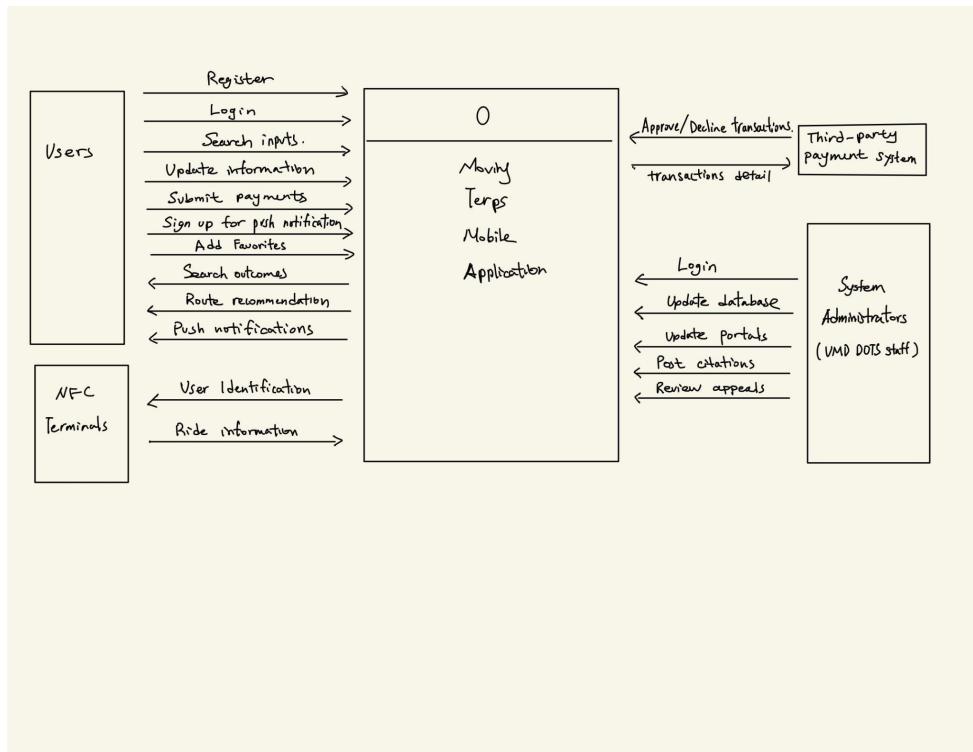
#### 4.4 Non-functional requirements

- Operational
  - The software will run on IOS and Android devices
  - The software should integrate with third-party payment systems
  - The software should enable NFC technology to be able to read and write information from a NFC terminal on the buses
- Performance
  - Any interaction between the user and the system should not exceed 2 seconds
  - The software should update real-time bus location and arrival times every 30 seconds

- The software should be available for use for 24 hours per day, 365 days per year.
- The software should support 5000 simultaneous users from 8-11 am and 3-7 pm; 3000 simultaneous users at all other times
- Security
  - Only people with active UMD Directory ID can register for an account
  - Users should log into the system with UMD Directory ID
  - System administrators will have access to update shuttle bus service and parking service information
  - System should not store user credit/debit card information
  - Personal information should not be shared with third-party systems
  - The system should include all available safeguards from malware
- Cultural and political
  - System should ask consent from users for push notifications
  - Personal information should be protected in compliance with the Data Protection Act

## 5.0 Data Flow Diagrams

### 5.1 Context Diagram



#### External Entities:

- **Users**: represent the active/current UMD students and faculty & staff
- **System Administrators**: represent authorized UMD Department of Transportation Service (DOTS) staff that have access to the datastores

- **Third-party Payment System:** represents the system that gives the Moving Terps mobile application the ability of making transactions without check or cash
- **NFC Terminal:** represents the machines install on the UMD shuttle buses that read user information and send ride information to user accounts

System Inputs:

*From Users*

- Account information
- Search inputs
- Edit/update information
- Submit payments
- Sign up for push notifications
- Add to Favorites

*From System Administrators*

- Account information
- Add/update databases
- Update portals
- Post citations
- Review appeal letters

*From Third-party Payment System:*

- Approve/decline transactions

*From NFC Terminal*

- Ride information

System Outputs:

*To Users*

- Search outcomes
- Route recommendations
- Push notifications

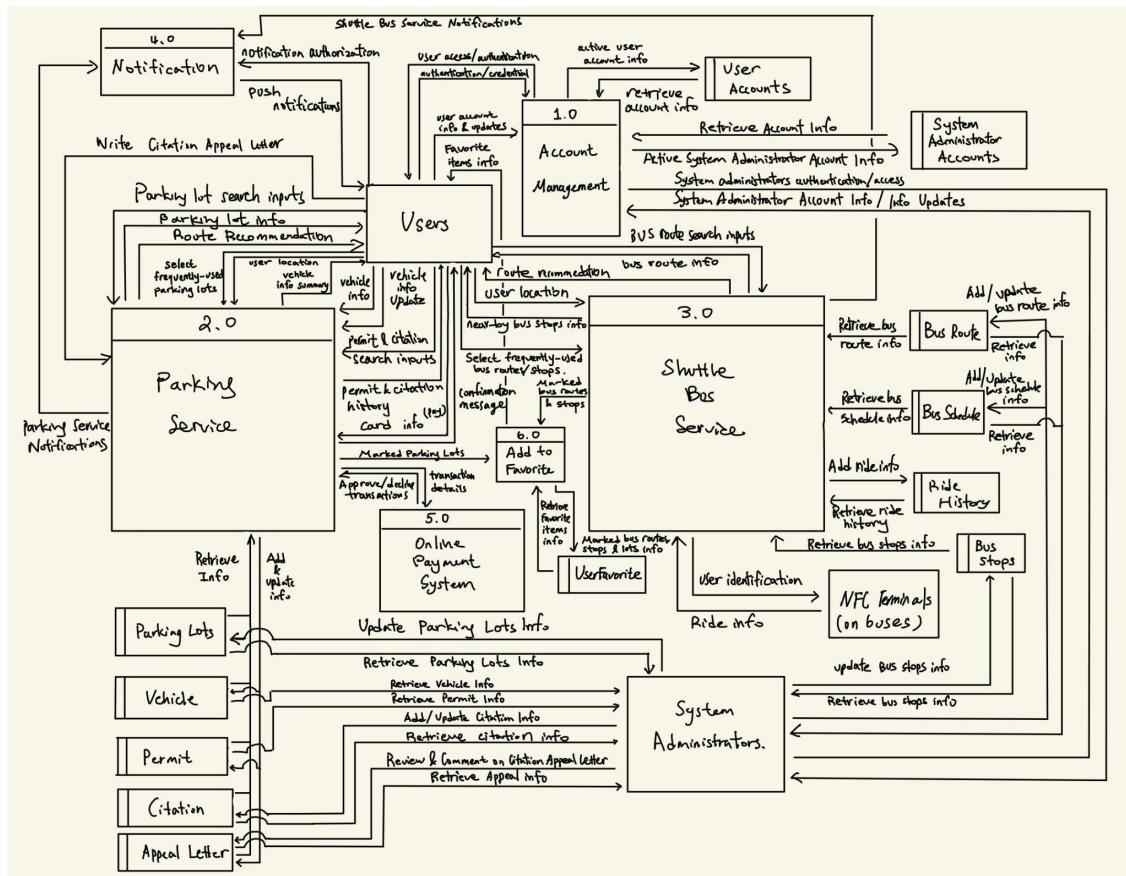
*To Third-party Payment System*

- Transaction detail

*To NFC Terminal*

- Ride information

## 5.2 Level 0 Diagram



## 5.3 Level 1 Process Tables

### 5.3.1 Account Management

Inputs	Outputs
<b>From Users:</b> <ol style="list-style-type: none"> <li>1. Account information (UID, password, name, user type, address, phone number, email address)</li> <li>2. Authentication/credential - password</li> </ol>	<b>To Users:</b> <ol style="list-style-type: none"> <li>1. User authentication/access</li> </ol>
<b>From User Accounts datastore:</b> <ol style="list-style-type: none"> <li>1. Retrieve active user account information</li> </ol>	<b>To User Accounts datastore:</b> <ol style="list-style-type: none"> <li>1. Active user account information</li> </ol>
<b>From System Administrators:</b> <ol style="list-style-type: none"> <li>1. Account information (UID, password, name, phone number, email address)</li> </ol>	<b>To System Administrators:</b> <ol style="list-style-type: none"> <li>1. System administrators authentication/access</li> </ol>

From System Administrator Accounts datastore: 1. Retrieve active system administrator account information	To System Administrator Accounts datastore: 1. Active system administrator account information
--	---

### 5.3.2 Parking Service

Inputs	Outputs
i) Parking Map From User: 1. Parking lot search 2. Frequently used parking lot 3. User location From Parking Lot datastore: 1. Retrieve parking lots information	To Users: 1. Parking lot information 2. Route recommendation 3. Marked parking lot
ii) Vehicle From User: 1. Vehicle information updated 2. Vehicle information added From Vehicle datastore: 1. Retrieve vehicle information	To Users: 1. Vehicle information summary To Vehicle datastore: 1. Add/update vehicle information
iii) Permit From Users: 1. Permit search inputs 2. Card information From Permit datastore: 1. Retrieve permit information From Online Payment System: 1. Approve/Decline transaction	To Users: 1. Permit history 2. Purchase confirmation message To Permit datastore: 1. Add/update permit information To Online Payment System: 1. Transaction details
iv) Citation From Users: 1. Citation appeal letter 2. Citation search inputs 3. Card information From Citation datastore: 1. Retrieve citation information From Appeal Letter datastore: 1. Retrieve appeal letter information From Online Payment System: 1. Transaction approval/decline	To Users: 1. Citation history 2. Payment confirmation message To Appeal Letter datastore: 1. Add/update appeal letter information To Online Payment System: 1. Transaction details

### 5.3.3 Shuttle Bus Service

Inputs	Outputs
i) Search a Route From User:	To User: 1. Routes, stops, and schedule detail

<p>1. Search input 2. Select frequently-used bus routes and stops</p> <p>From Bus Route datastore: 1. Routes detail</p> <p>From Bus Schedule datastore: 1. Schedule detail</p> <p>From Bus Stops datastore: 1. Stops detail</p>	<p>To Shuttle Bus Service function: 1. Marked bus routes and stops</p>
<p>ii) Near-by Stations</p> <p>From User: 1. User location 2. Select frequently used bus routes and stops</p> <p>From Bus Route datastore: 1. Routes detail</p> <p>From Bus Schedule datastore: 1. Schedule detail</p> <p>From Bus Stops datastore: 1. Stops detail</p>	<p>To User: 1. Nearby bus stops detail</p> <p>To Shuttle Bus Service function: 1. Marked bus routes and stops</p>
<p>iii) Route Planning</p> <p>From User: 1. Starting location 2. Destination Location 3. Departure time / Arrival by time</p> <p>From Bus Route datastore: 1. Routes detail</p> <p>From Bus Schedule datastore: 1. Schedule detail</p> <p>From Bus Stops datastore: 1. Stops detail</p>	<p>To User: 1. Available route options sorted by shortest time 2. Nearest route stop from the user starting location and distance 3. Bus arrival time at nearby stops.</p>
<p>iv) Track Your Ride</p> <p>From User: 1. User identification (user information)</p>	<p>To User: 1. Ride history information and tracking (from NFC terminals on buses)</p>

#### 5.3.4 Notification

Inputs	Outputs
<p>From Users: 1. Notification preference authorization selection/update</p> <p>From Parking Service Function: 1. Parking Information according to preference setting</p> <p>From Shuttle Bus Service function:</p>	<p>To Users: 1. Push notifications</p>

1. Shuttle Bus Information according to preference setting	
--	--

### 5.3.5 Third-party online payment system

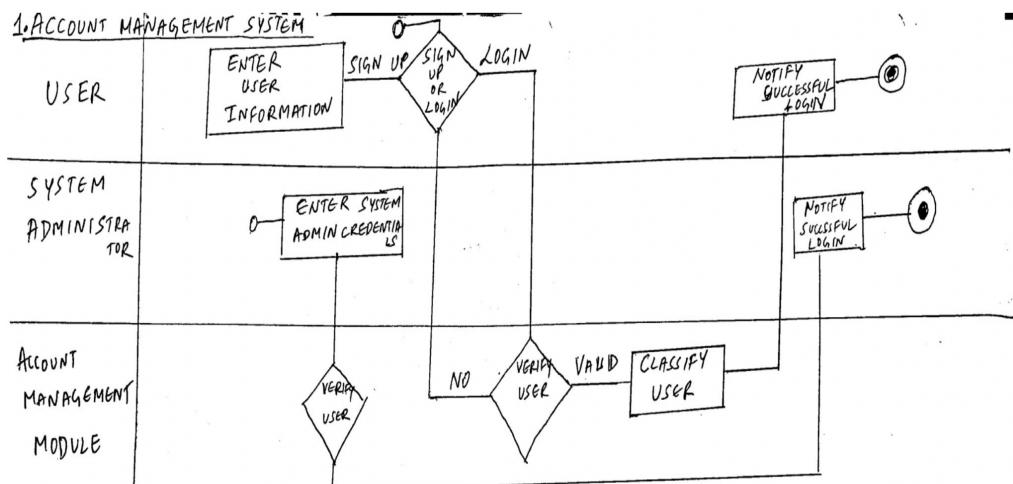
Inputs	Outputs
From Parking Service function: 1. Transaction details	To Parking Service function: 1. Transaction approval/decline

### 5.3.6 Add to User Favorite

Inputs	Outputs
From Parking Service function: 1. Marked parking lots	To User Favorite datastore: 1. Marked parking lots information
From Shuttle Bus Service function: 1. Marked bus routes and stops	To User Favorite datastore: 1. Marked bus routes and stops information
From User Favorite datastore: 1. Retrieve favorite items information	To Users: 1. Favorite items information

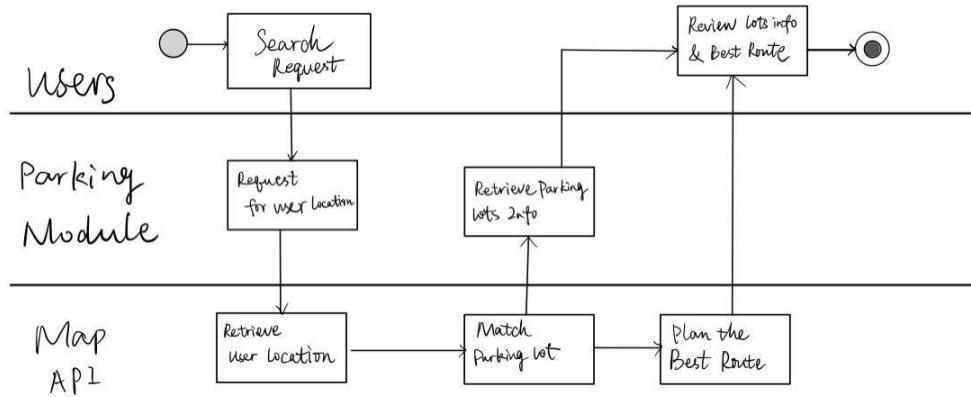
## 6.0 Swimlane Processes Diagrams

### 6.1 Account Management System

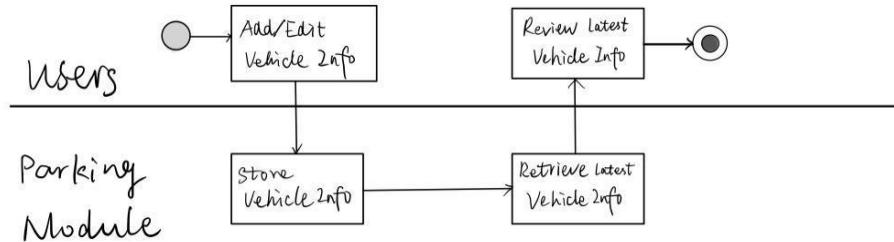


## 6.2 Parking Service

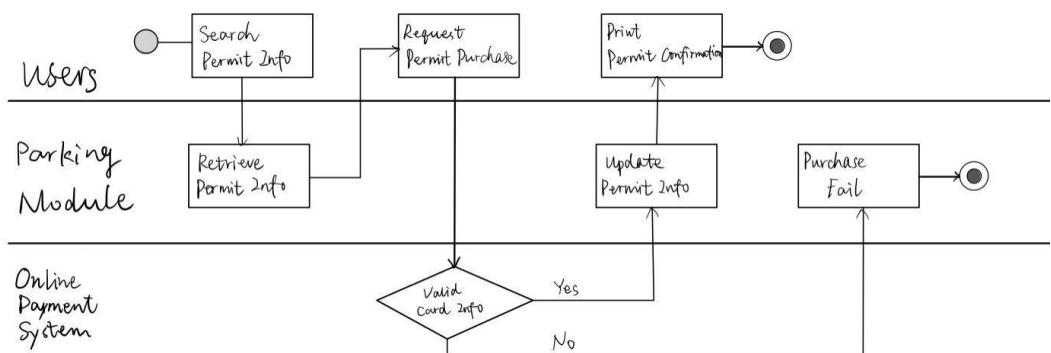
### I. Parking Lot and Route Searching



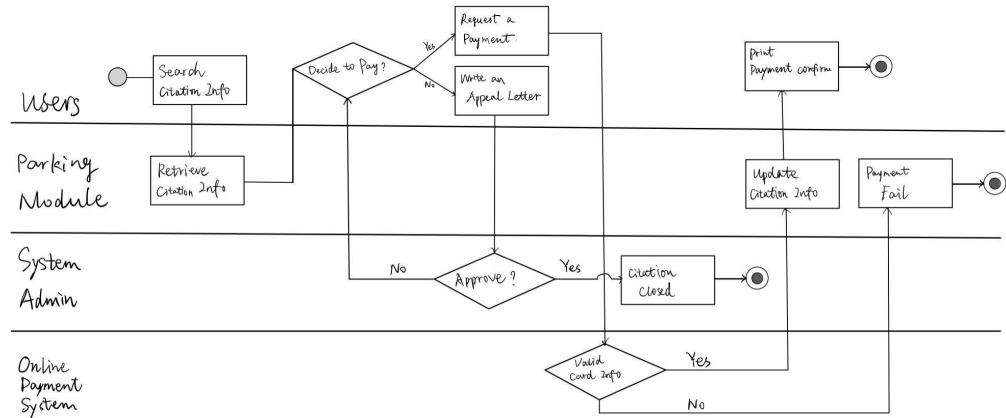
### II. Vehicle Information



### III. Permit Purchase

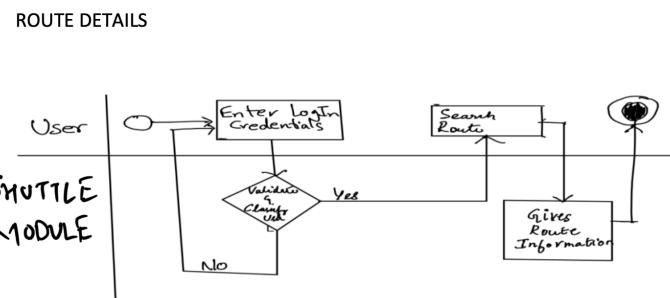


#### IV. Citation Payment

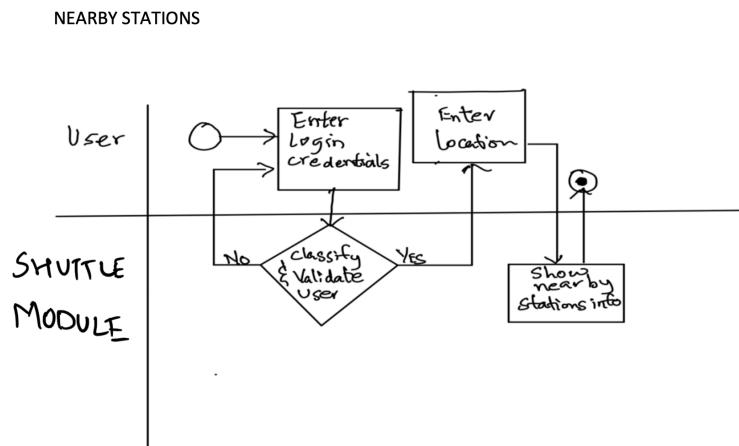


#### 6.3 Shuttle Bus Service

##### I. Search a Route

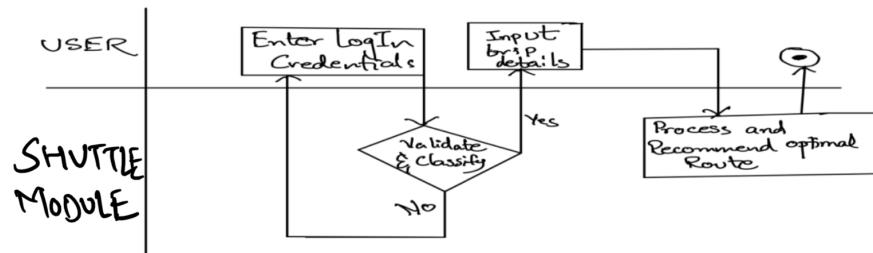


##### II. Near-by Stations



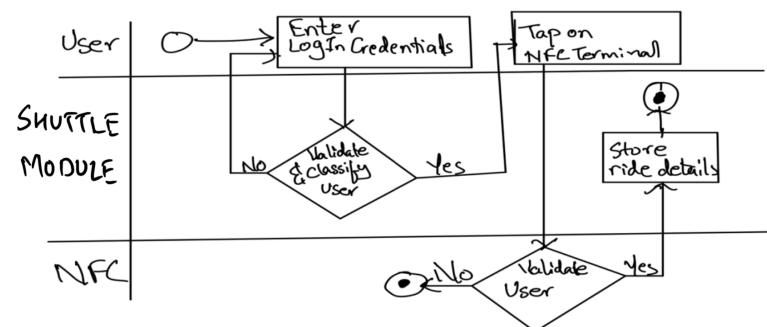
### III. Route Planning

#### ROUTE PLANNING

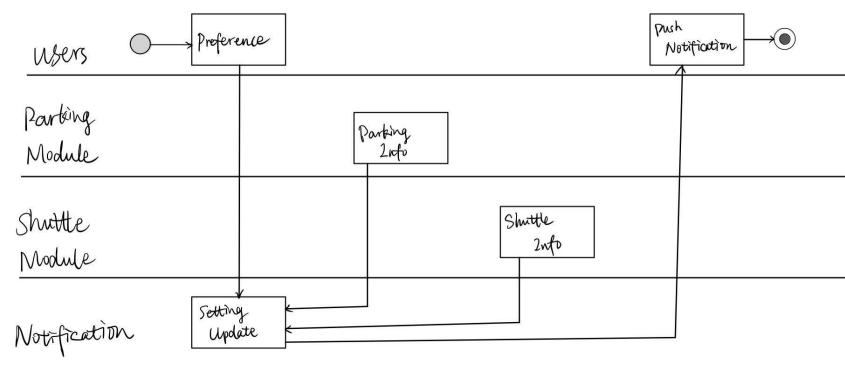


### IV. Track Your Ride

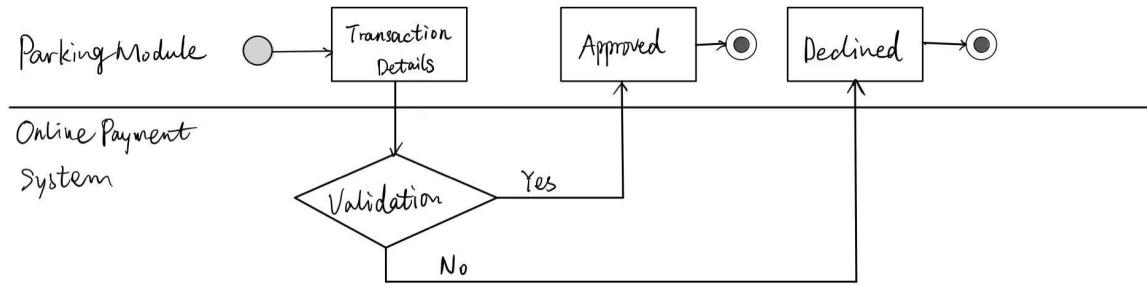
#### NFC RIDE DETAILS



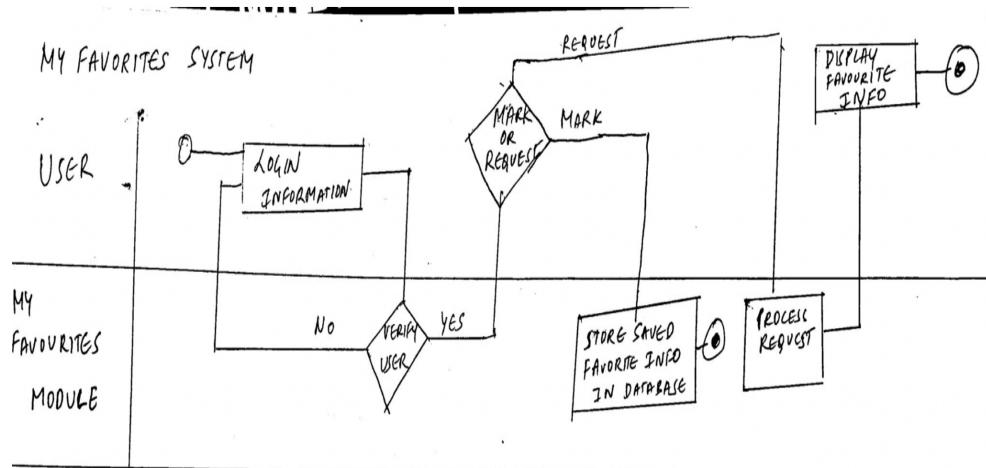
### 6.4 Notification



## 6.5 Online Payment System

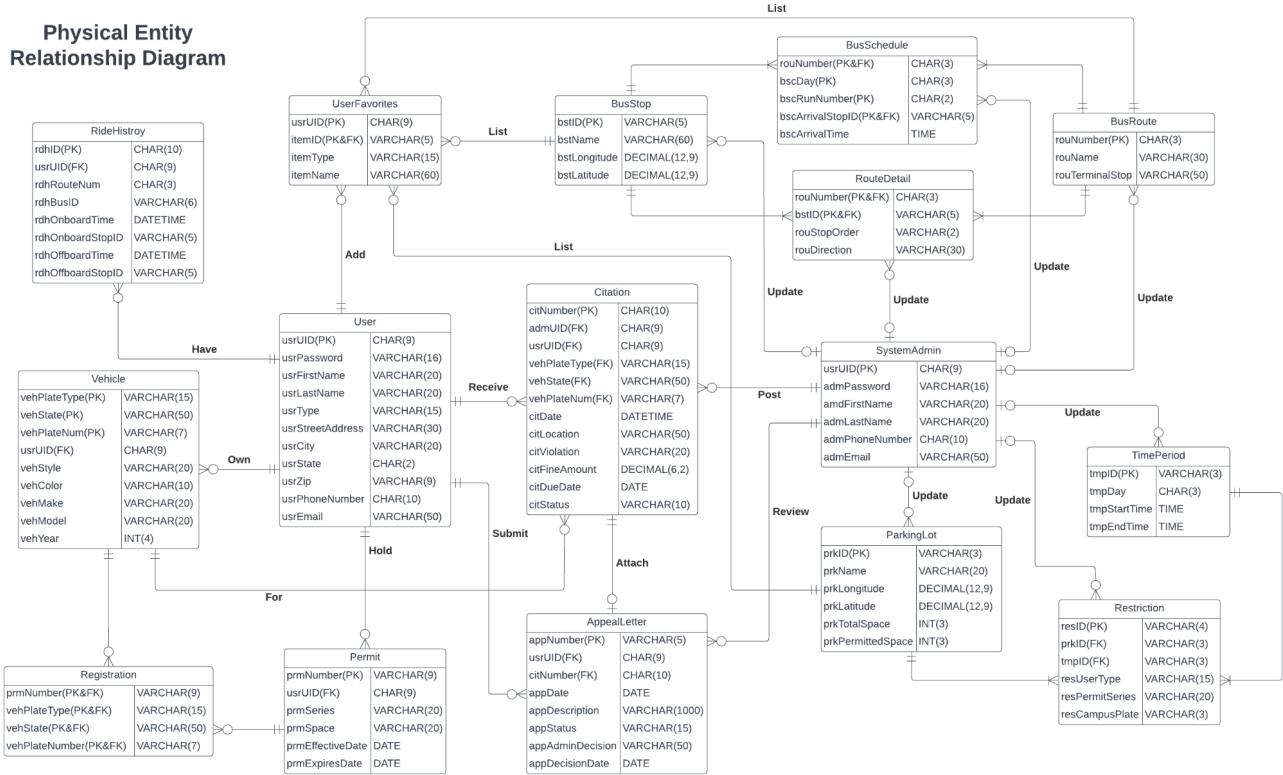


## 6.6 Add to User Favorites

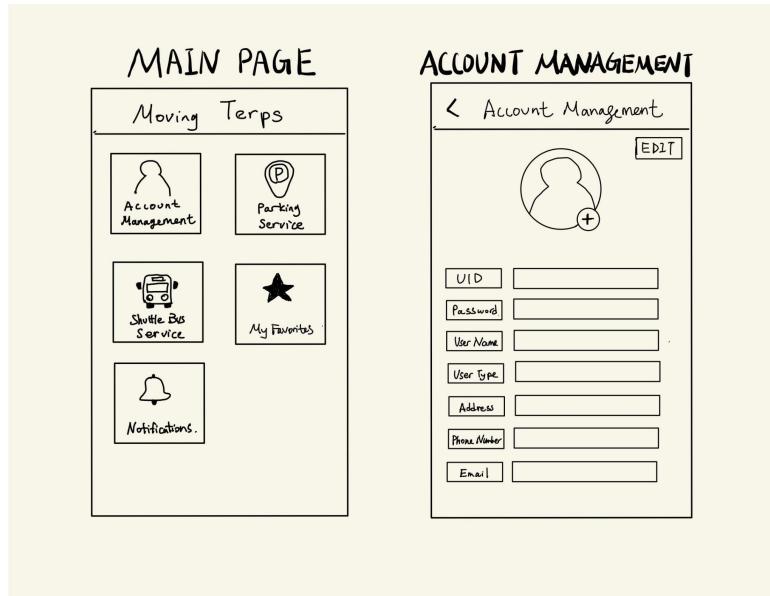


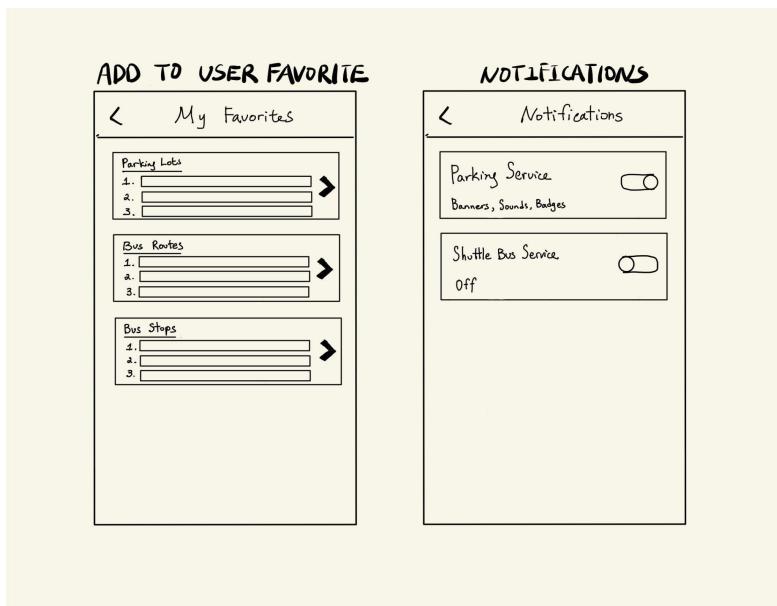
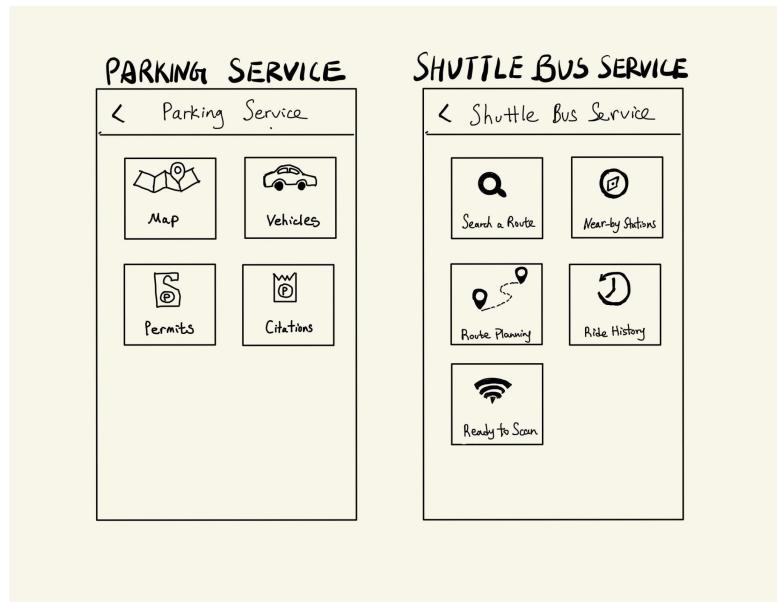
## System Designing Phase

## 7.0 Physical ERD



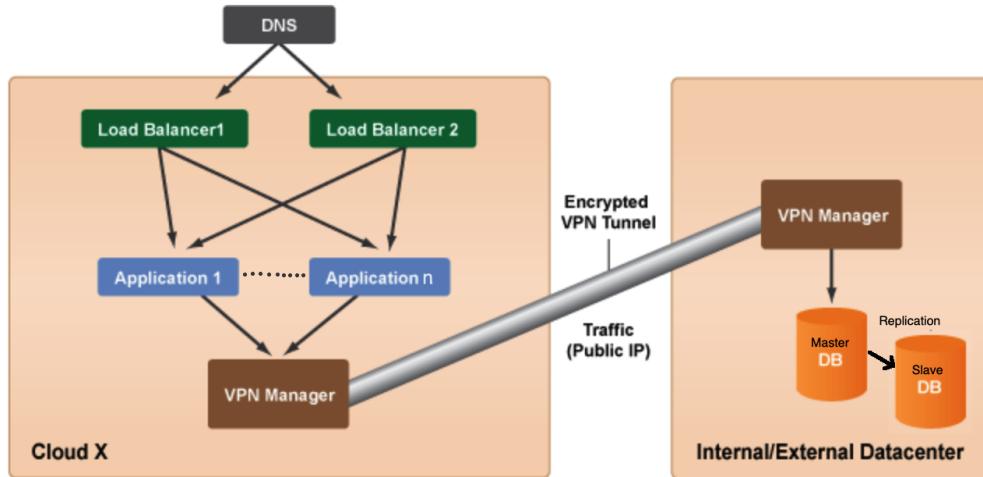
## 8.0 User Interface





## 9.0 System Design Architecture

For our system we would be opting for Cloud and dedicated hosting architecture.



### Architecture:

1. Users would access the system using the mobile application.
2. The mobile application is connected to the VPC via the internet. Domain Name System Server will resolve the URL for a particular web resource to the TCP-IP address of the system or service which can deliver that resource.
3. A virtual private cloud (VPC) is a secure, isolated private cloud hosted within a public cloud. VPC customers can run code, store data, host applications, and do anything else they could do in an ordinary private cloud, but the private cloud is hosted remotely by a public cloud provider. VPCs combine the scalability and convenience of public cloud computing with the data isolation of private cloud computing.
4. The Load Balancer present in the VPC is used to distribute the network or application traffic across several servers. They improve the overall performance of applications by decreasing the burden on servers associated with managing and maintaining application and network sessions, as well as by performing application-specific tasks especially during the peak hour when many users are logging and using the application at the same time.
5. We will also use auto-scaling functionality of the cloud. In case the load on the server increases, for example, during match day the number of people using the application tends to increase, the cloud will have the provision to spin up a new application server to handle the additional workload.
6. The data for students is stored in a private cloud within the VPC as student data is sensitive and it is also legally binding to ensure privacy of students.
7. The datacenter consists of a master node and a slave node. This ensures replication of data and helps in disaster management and data recovery.

8. The cloud and the datacenter interact with each other using an encrypted VPN tunnel.

## 10.0 Testing Plan/Documentation

### 10.1 Stub Testing

Test Name	Test Description	Inputs	Outputs	Expected Results	Potential Errors
Check_parking_direction_stub1	Stubbing recommended parking lot details to check if system shows optimized direction	Login credentials, stubbing for recommended routes	Map is displayed with direction to the chosen route to the chosen parking lot.	Displays the most optimized route to the chosen parking lot.	1.Route not found 2.Displays unoptimized route
Check_notifications_direction_stub2	Check push notifications by stubbing notification information content.	Login credentials, notification information.	Notifications to be displayed for all users.	Notification information displayed when the app is closed.	Notification doesn't show up if the app is closed, displays only for certain users.
Check_account_admin_stub3	Check access permission for admin, by stubbing information for different kinds of users.	Login credentials for different users	Tickets closed by admin users.	Access to admin users while other users should not have access to closing tickets.	Students and faculty are able to close tickets.
Check_nearest_stops_stub4	Check nearest stops are populated correctly by stubbing bus and route details.	Start and End location, bus and route details.	Display Distance from nearby stops.	Distance from nearby stops are displayed correctly	1. Distance not displayed correctly 2. Stops from farther distance is displayed instead of nearby stops

Check_bus_recommendation_ordering_stub5	Verify bus recommendations ordered by shortest duration of trip by stubbing route, bus and schedule details	Start and End location, bus and route details	Recommendations are ordered by shortest duration	All Recommendations for nearby stations are ordered by shortest duration of the trip	Trip details of the longest trip are displayed first.
Check_bus_recommendation_stub6	Check bus recommendation by stubbing dynamic display of route and bus details on map	Start and End location, map and bus details.	Recommendations for different buses are displayed.	Recommendation functionality should work fine	1.Error 'no route found' 2. Buses for different routes are displayed
Check_payment_redirection_stub7	Stub information from a third party payment app to check if the redirection to payment app is working fine.	Payment details, otp, address, input from payment app.	Transaction successful.	Page should redirect to payment app and back	1.Transaction reveals payment details of the user. 2.Transaction not successful in entering the right details.
Check_payment_redirection_stub8	Check nearest parking lots are recommended by stubbing the real time parking lots information	Login details, Location, vehicle details	Nearest available parking lots are displayed	Nearest available parking lots should be displayed to the user.	1.Unavailable lots are shown as available. 2.Parking lots of farther location is shown when there are other nearby available lots.
Check_payment_timeout_stub9	Check time out when there is no response from gateway or user by stubbing response from payment gateway	Login details, payment details, OTP, address, input from payment app, mobile number	Page timeout error is displayed	Page should time out and ask the user to start the process again.	No timeout message and the indefinitely keeps loading.
Check_rideHistor	Stub the NFC	User	Trip	Response	1.Wrong trip

y_NFC_stub10	response to check storing of ride history in the app	information, NFC response as stub.	information is available in the history of it.	from NFC should be stored in the app and the user should be able to view it.	details stored. 2.Error in accepting response from NFC.
--------------	--	------------------------------------	--	--	--

## 10.2 Unit Testing

Test Name	Test Description	Inputs	Outputs	Expected Results	Potential Errors
Modify account information_ Unit test1	Checking modifications of user accounts	Modified user account information	Modification successful	New information displayed in account profile	1. Unsuccessful modification 2. Invalid inputs 3. New information not displayed in profile
Search for a parking lot_ Unit test2	Checking search for parking lots	Search inputs: user type, arrival time of day, destination	1. Recommended parking lots based on user inputs 2. Detail information about each recommended parking lots	Parking lots displayed as clickable markers on the interactive map interface	1. Required field missing 2. Outputs not displayed 3. Inaccurate outputs displayed 4. Users do not turn on location service
Add or update vehicle information_ Unit test3	Checking add vehicle information	Required vehicle information (plate type, plate state, plate number) and optional fields	Successfully add a new vehicle's information	New vehicle information is added and displayed in the vehicle function	1. Required field missing 2. Values for the required field are not in the correct format 3. Information not added or displayed in the vehicle list
Pay for parking permits_ Unit test4	Checking pay for parking permits	1. Permit selection 2. Vehicle to register under the permit 3. Payment	1. Users purchase the selected permit item without error 2. Payment	Users successfully purchased permits and the item is displayed in	1. Selected item does not match with user type 2. Payment is declined by the online payment

		method & corresponding payment credential (card holder name, card number, expire date, and code)	credential approved by third-party online payment system	the permit list function under the parking service module	system 3. Purchased item is not displayed
Appeal against citations_ Unit test5	Checking appeal against citations	1. Citation number 2. Appeal letter text 3. User personal information (name, contact information, address)	Appeal letter submission confirmation message	User receive a submission confirmation message, submission and status displayed in the citation module	1. Required fields missing 2. Citation number not found 3. Length of the appeal letter exceeds maximum limit 4. Submission failed 5. Submission not displayed
Search for a bus route_ Unit test6	Checking search for a bus route	Bus route number	1. Route information based on search inputs 2. Route details (stops and schedule)	Search results are displayed as clickable marker on an interactive map	1. Invalid route number 2. Results are not displayed on the map 3. Users are not able to interact with the map
Search for a nearby bus stop_ Unit test7	Checking search for nearby bus stops	User location (longitude & latitude) by enabling location service	Nearby bus stops and corresponding detailed stops information	Bus stops displayed as clickable markers on the interactive map	1. Location service not functioning 2. Users don't enable location services 3. Search outputs not displayed
Plan a route_ Unit test8	Checking plan a route	1. User starting location (either manually entered or by enabling location service) 2. Destination (manually)	1. Available route options sorted by shortest time 2. Nearest route stops from the user starting location and distance 3. Bus arrival	Search results and corresponding search details displayed as clickable markers or items on the interactive	1. Required fields missing 2. Search function unusable 3. Search inputs (time) exceed limitation 4. Search inputs not found 5. Outputs not

		entered) 3. Departure and/or arrival time	time at nearby stops	map	displayed on the interactive map
Opt in push notifications_Unit test9	Checking push notification	1. Notification preference authorization from users 2. Information from parking and shuttle bus service modules	Push notifications containing information from the parking and shuttle bus service modules	Push notifications displayed based on user preference (lock screen, banner, an/or notification center)	1. Notification not displayed 2. Notifications displayed when user disable the function 3. Inaccurate information
Add items to User Favorites_Unit test10	Checking add items to User Favorites	Selected items from other modules (bus routes, bus stops, and parking lots)	Selected items successfully added to the User Favorites list	Selected items added to the User Favorites item list and displayed in order	1. Unable to select items from other modules 2. Selected items not displayed

### 10.3 System Testing

Test Id	Test Description	Inputs	Outputs	Expected Results	Potential Errors
Operating on different system_System test1	Applications can launch normally on different operating system (IOS and Android)	System initiated request to open the application	Application initiated and operates successfully	Application runs normally without error	1. System bugs 2. Outdated operating system 3. Network/server malfunction
Send to and receive information from NFC terminals_System test2	Users can use the integrated NFC function to send user identification information to the NFC terminals and read and store ride information from terminals	Send user identification by tapping the mobile phone on the NFC terminals on shuttle buses	Bus and ride history information are stored to the shuttle bus service module	Detailed ride information is stored and listed in the ride history function under the shuttle bus service module	1. NFC terminals are not able to detect user mobile device 2. Inaccurate information is being stored 3. Ride information is not stored at all

Log in/out and modify information System test3	Users can log into and out of the application and modify personal information	1. User UMD directory ID and password 2. Edit request	1. Users log into and out of the application successfully 2. Users have the access to modify personal information	1. Users are led to the account management module 2. Old information is replaced by the new information	1. Inaccurate authentication 2. Correct authentication information being entered but application not responding 3. Modified information isn't stored
Shuttle bus service System test4	Users can use the shuttle bus service module to search for shuttle bus route, locate the nearest stop, or plan a route	1. Bus route number 2. User current location, destination, and planned departure/arrival time	1. Bus stops for each route 2. Nearest bus stops 3. Optimized travel options	Routes, stops, or travel options are displayed as clickable markers on an interactive map	1. No result found 2. Users don't enable location service 3. Entered destination exceeds service range 4. Preferred departure time doesn't fall within reasonable time range
Parking service System test5	Users can use the application to search for parking lots, modify information, pay for permits and citations, and appeal against citations	1. Preferred destination, user type, and arrival date and time 2. Vehicle information 3. Citation number	1. Search results (parking lots) 2. New vehicle information 3. Detailed information about purchased permits and citations	1. Parking lots (search results) are displayed as clickable markers on the interactive map 2. Information listed in each function	1. Required search fields missing 2. Inaccurate payment information 3. Citation number not found 4. Search results are not displayed on the interactive map 5. Stored data (ex. purchased permits) not displaying
Check historical transaction records System test6	Users can check historical transaction records for purchasing permits and paying for citations	User click on the parking service module and select functions accordingly	History payment information (date, item name, payment method, payment amount, and billing address)	Historical payment records are arranged by date and displayed in each function	1. Missing transaction records 2. Inaccurate transaction information
Check	Users can check	1. System	Marked items	The	1. Marked items are

favorite items_System test7	and modify item listed in the Add to User Favorites module	initiated request and edit request 2. New selected items from other modules	add to the module	application accurately displayed the marked items in the module	not displayed in the module 2. Users are not able to select items from other modules
Notification setting_System test8	Users can turn on and off the notification setting for different modules and functions	Request to turn on or off a notification setting	Notification setting being turned on or off	Application responds to the request and turn on or off the notification setting without errors	Users are not able to turn on or off a notification setting (application not responding)
Peak time performance_System test9	The application can operate under the high volume of simultaneous users during peak time	Simulate 5000 users login and use application at the same time	1. system response time 2. error rate 3. throughput – amount of data transmitted per second	Application should be able to run without major breakdown throughout the whole testing period	1. System response time not meeting business requirements 2. Wrong load-balancing model 3. High probability of errors occurring
Online payment security check_System test10	Checking if the application will store payment credentials when users are placing an order and completing an transaction	Dummy payment credentials	No new data entries to the cloud database that contains any piece of information about the payment credentials	Card information is not stored in the database	Data is stored automatically without user permission

#### 10.4 Acceptance Testing

Test Id	Test Description	Inputs	Outputs	Expected Results	Potential Errors
Account Creation_Acceptance Test 1	Check if a new user is created after entering UMD ID,	UMD ID, Password, Verification code	User account information should be added to the	Notification of account created. Login Page to re-login with the	User account not added to database, preventing re-login for the customer

	password and verifying user.		database.	created credentials.	
User Login_Acceptance Test 2	Check if a returning user is able to login into an existing account using correct credentials	UMD ID and password	Successful databases fetches from database for user	User Home page should be displayed upon successful login	1. Inaccurate authentication 2. Correct authentication information provided but not stored in database
Bus Route Details_Acc eptance Test 3	Check if a user is able to see the bus route details of selected route	Bus route number	Query and fetch data regarding the selected bus route	Details of selected bus route should be displayed	1. Incorrect information is fetched from the database. 2. System taking too long to display the results
Bus Stop_Acceptance Test 4	Check if a user can see the nearest bus stop	1) Bus route number 2) User location	Select the nearest bus stop for the user based on his current location and given bus route	Nearest bus stop and route to the bus stop from the user's current location should be displayed in a map.	1. No bus stop found 2. Bus not serving nearest bus stop 3. User location not enabled
Search Parking_Acceptance Test 5	Check if nearby parking spots available for user are shown upon searching the location	UMD ID, location	Based on users' location select all available nearby parking allowed for him	Interactive map displaying all nearby locations for parking to that user with the option to select.	1. Unable to interact with the map. 2. Map not rendered correctly on screen. 3. Map displaying parking not accessible to user
Show Ride History_Acceptance Test 6	Check if a user can access their ride history	UMD ID	Query and fetch data about ride history	Display ride history on the basis of UMD ID	1. User UMD ID not linked to database 2. Incorrect ride history being fetched from database
Display Notifications_Acceptance Test 7	Provide push notifications to the user	Permissio n of user to enable push notifications	Allow push notifications to be enabled on user's phone	Display various updates on parking and bus route changes	1. Notifications not being displayed 2. Inaccurate data being displayed in notifications 3. Blank/test notifications being displayed
Payment_Acceptance	Check if the payment made	Amount, UMD ID,	Check if the amount is	Display payment successful	1. Payment successful but amount not reflected in the

Test 8	via third party systems on app is successful	payment details, authorization code	reflected in the UMD bank account.	notification for successful payment,	bank account. 2. Response message Not being displayed to the user.
NFC_Acceptance Test 9	Check if the ride data is being stored in the database on NFC tap	Tap on NFC terminal using mobile app of user	User information should be stored in the database	Correct Ride info(Bus number, time of boarding, route number) is stored in the database.	1.Terminal not able to read the NFC via device. 2. Incorrect data getting stored in the database.
Parking Info_Acceptance Test 10	Check if new parking information is added after entering vehicle number and permit number	Vehicle Number, Permit Number	Parking information should be added to the database	Display information added in the user's account	1. Information not being stored in database 2. Inaccurate information being retrieved from database 3. Values of required field not in correct format 4. Important fields missing

## 10.5 Integration Testing

Test Name	Test Description	Inputs	Outputs	Expected Results	Potential Errors
Ready to Scan and Track Your Ride_Integration test1	Users use the NFC system integrated with the mobile application to send identification information and read ride information from NFC terminals on buses	Identification number of the NFC sensors of the user mobile device and terminals on buses	User information are sent to the NFC terminals and bus/ride information is sent to and stored in the application	User mobile device and the NFC terminals exchange information successfully without an error	1. User's mobile device does not send out identification information 2. Users do not turn on the NFC option on mobile device 3. NFC terminals on buses are not working
Making online payment_Integration test2	Users conduct transactions with third-party online payment systems	Payment credentials (card holder name, card number, expiration date, and	1. Payment acceptance or denial message 2. Receipt/invoice if the payment	Transactions are approved by the online payment system and the order is	1. Invalid payment credential 2. Payment credential is compromised 3. Account for the payment is with insufficient funds 4. Transactions are

		SEC code)	is accepted	successfully placed	submitted but do not go through the payment system 5. Transactions are approved but the users have not been charged
Search function with map API_ Integration test3	Users search for a parking lot based on search inputs and an interactive mapping API processes the query and displays the results	Search inputs (user type, starting location, destination, and arrival time of day)	1. Search results (parking lots) and their corresponding detailed information (name, restrictions, and available spaces) 2. Optimized route to the destination, distance, and travel time	Search results are displayed as interactable markers on the map API	1. The map API does not receive any information from the search parameters 2. The map API cannot display the search results 3. Users are not able to interact with the map API
Cloud-based application and cloud-based database_ Integration test4	Cloud-based mobile application Moving Terps will retrieve data and information from a cloud-based database hosted on a virtual private cloud	1. Request for data (user to server) 2. Database query (server to database)	1. Query results in a table format (database to server) 2. Information displayed on the user interface when users click on each module and functions with minimized waiting time and without any errors	Information/ data is displayed on the user interface when users click on each module and functions with minimized waiting time and without any errors	1. Server cannot receive request for data from users 2. Server cannot process the request and convert it into a database query 3. Database cannot execute the query and generate results 4. Database cannot return the query results to the server 5. Server cannot process the database response and transmit it to the users

## 11.0 Implementation Plan

### 11.1 Objectives

Primary Objective: Successful completion and deployment of the application at mobile end link with UMD DOTS system.

Supporting Objective: Training end-users and setting up maintenance rules.

## 11.2 Details and Deliverables

Tasks	Deliverables	Developed By	Approved By
<b>User Interface -</b> Includes interface for students, employers and faculty based on usability.	Deliver interfaces that meet the business needs, and all requirements include: <ul style="list-style-type: none"> <li>- Create Account View</li> <li>- Create Search History</li> <li>- Create Edit/update information</li> <li>- Create Payment Method</li> <li>- Create Favorites Route</li> <li>- Create Nearby Routes with times to arrive</li> <li>- Create push notifications for whom signed up</li> <li>- Enable users to search the route, nearby station, and parking lots</li> <li>- Enable users to track their ride while they are on the shuttle bus</li> </ul>	Development Team	System Analyst Project Manager Project Sponsor
<b>Integration -</b> Includes data and functionality linking between DOTS and integration with supporting NFC terminal.	<ul style="list-style-type: none"> <li>- Integration with DOTS and NFC system</li> <li>- Integration with students' enrolled information to check if users are enrolled student at UMD</li> <li>- Integration with UMD staff/ faculty system to check if users are currently hired by UMD</li> <li>- Integration with third-party payment entity to check if the payment methods are fraud</li> </ul>	Development Team	System Analyst Project Manager
<b>Developer/User Documentation -</b> Includes all technical documentation delivered during the project; all documentation necessary to effectively operate and maintain the system based on user types	<ul style="list-style-type: none"> <li>- Provide users with infrastructure setup, daily process information that includes performance assessment</li> <li>- Includes simulations and knowledge-based reports</li> <li>- The software could be run on IOS and Android devices</li> <li>- The software should</li> </ul>	Development Team	System Analyst Project Manager

	<p>be used 24/7</p> <ul style="list-style-type: none"> <li>- The software should support 5000 simultaneous users from 8-11 am and 3-7 pm; 3000 simultaneous users at all other times</li> </ul>		
<b>Readiness Document -</b> Consolidate information regarding the current status of the parking lot and shuttles for users to make Go/No Go decisions. Should include checklists for users preparing to apply for the service.	<ul style="list-style-type: none"> <li>- Provide information necessary for users to decide like which parking lot to park or which shuttles bus to take</li> <li>- Affirm achievement of all deliverable acceptance criteria</li> </ul>	System Analyst Project Manager	Project Sponsor
<b>License Structure -</b> Primary contract document between UMD and NFC	<ul style="list-style-type: none"> <li>- Identity ways that the system can be used by UMD</li> <li>- Ensure NFC's obligation for UMD's access and support</li> <li>- Ensure all students with valid UID can access to the account, including creating accounts and submit application</li> <li>- Ensure software won't store and share any private information of users registered in the system</li> <li>- Personal information should be protected in compliance with the Data Protection Act</li> <li>- Third-party payment system won't have the right to sell or use users' payment method for uses other than paying for the parking service for UMD</li> <li>- System should not store user's credit/ debit card information</li> </ul>	System Analyst Project Manager	Project Sponsor
<b>Post-Implementation review Report-</b> summarizes the assessment of implementation activities at the end of the implementation phase	<ul style="list-style-type: none"> <li>- Summarize assessment of implementation activities</li> <li>- Evaluate the effectiveness of the system after the system has been in production</li> <li>- Determine if the</li> </ul>	System Analyst Project Manager	Project Sponsor

	<p>system does what it was designed to do</p> <ul style="list-style-type: none"> <li>- Determine if all users who applied for the parking have received quotation within acceptable time frame with all the information on the citation is correct</li> </ul>		
<b>Standard Operating Procedures</b> - define in detail on how the system team will perform the business process related to the operations and maintenance of the system. Whereas the User Guide is focused on the use of the system specifically, the SOP addresses all related business processes of our app.	<ul style="list-style-type: none"> <li>- Provide detailed instructions for future business process</li> <li>- Ensure consistent execution of business process</li> <li>- Drive performance improvement and improve organization results</li> <li>- Provide detailed information on all external entities together building this system with all citations on transaction</li> </ul>	Development Team	System Analyst Project Manager
<b>Training Plan</b> -includes full schedule of end-user training for Students and Faculties	<p>Set notification plans to end-users to ensure all system participants can operate systems based on business rules.</p> <ul style="list-style-type: none"> <li>- Student Training Plan</li> <li>- Faculty Training Plan</li> </ul>	System Analyst/Project Manager	Project Sponsor
<b>Infrastructure</b> -Overall system Performance as intended based on business needs.	<ul style="list-style-type: none"> <li>- Load Requirements Assessment: Ensure the system can function with full user load.</li> <li>- Infrastructure Setup-Up Assessment: Ensures process application functionality</li> </ul>	Development Team	System Analyst/Project Manager — Project Sponsor
<b>Rollout Plan</b> -Schedule and methods for actual deployment of MovingTerps for users.	<ul style="list-style-type: none"> <li>- UMD Faculty Notifications</li> <li>- UMD Student Notifications</li> <li>- Deployment Schedule</li> <li>- Development Response Plan: ensure optimal IT support response for initial rollout.</li> </ul>	Project Manager	Project Sponsor

### 11.3 Implementation Plan Schedule

<b>Task Name</b>	<b>Owner</b>	<b>Date</b>	<b>Status</b>
<b>Development Activities</b>			
Database Design	Development team	10-May	In Progress
Data Migration Plan	Development team	10-May	In Progress
User Interface Design	Development team	20-May	In Progress
Integrations	Development team	21-Jun	In Progress
Reports	System Analyst	30-Jun	In Progress
<b>Testing Activities</b>			
Testing Planning	System Analyst Project Manager	30-Jun	In Progress
Unit Testing	Development team	5-Jul	Pending
Integration Testing	Development team	5-Jul	Pending
Acceptance Testing	Development team	8-Jul	Pending
<b>Documentation Prep</b>			
Developer Documentation	Development team	12-Jul	Pending
Standard Operating Procedures	System Analyst Project Manager	12-Jul	Pending
User Documentation	Development team	12-Jul	Pending
<b>Training</b>			
Training Planning	System Analyst	12-Jul	Pending
User Training Development	System Analyst	12-Jul	Pending
User Training Review	System Analyst	18-Jul	Pending
User Training Rollout	Development team	18-Jul	Pending
<b>Rollout</b>			
Architecture Planning	Development team	19-Jul	Pending
Infrastructure Prep	Development team	19-Jul	Pending
License Structure	System Analyst Project Manager	19-Jul	Pending

Deployment	Development team	25-Jul	Pending
Data Migration Steps	Development team	25-Jul	Pending
User Setup	Development team	29-Jul	Pending
User Training	Development team	29-Jul	Pending
Pilot Rollout Plan/Phase Rollout Plan	System Analyst Project Manager	1-Aug	Pending
Rollback Steps	Development team	4-Aug	Pending
Readiness Document	System Analyst Project Manager	5-Aug	Pending
Day 0 Deployment	Development team	8-Aug	Pending
Support Setup	Development team	8-Aug	Pending
Week 1 Support Plan	System Analyst Project Manager	15-Aug	Pending
Week 1 Daily Review Meeting	Development team	15-Aug	Pending
Post Implementation Review Report	System Analyst Project Manager	19-Aug	Pending

## 12.0 Conclusion

The requirements are collected through interviewing a wide range of actual users of the service provided by the DOTS. Interviews are also conducted on the system administrators to recognize the flaws of the current system. User feedback is the fundamental component of designing our mobile application and all perspectives are being considered to meet the business requirements set. Further integrations with systems like third-party vehicle or bike rental systems can be performed once the mobile application is successfully implemented and the operation of the application is proven to be up to standard and approved by the users.

## Appendix - Feedback from Previous Meetings

### 1.0 Feedback from Meeting 1

- System must be able to filter search results based on user types/search inputs
- System must allow users to sign up push notifications

## Project Scope - Functions (Cont.)

- Parking:** Users are able to enter starting location and destination, and an interactive map will show parking lot recommendations based on user type together with the detailed parking lot restrictions. Then, by clicking on a recommended parking lot, the system will show the direction to get to that selected parking lot. Users can update personal and vehicle information, pay for permits, citations, and temporary parking, appeal against citations.
- Notification:** Notify any impact schedule and current status (specific shuttle and parking lot availability during special events) through “push notification.” Users don’t need to check email or the impact calendar manually.



Sindhu Rajan  
Not all users can use all parking lots, make sure u accomodate for that

Sindhu Rajan  
User must sign up for this

## 2.0 Feedback from Meeting 2

- Add primary keys, foreign keys, and fields to the entity relationship diagram (ERD) to convert it into a physical entity relationship diagram (physical ERD.)
- Should keep the same id name across different tables (ex. usrUID)

