University of Maryland

DEPARTMENT OF
TRANSPORTATION SERVICES

DOTS SmartParking Information System Plan

Client: Department of Transportation Services

BUDT723 Business Process Analysis

Professor Sindhu Rajan

May 8th, 2022

Table of Contents

Executive Summary 4

System Planning Phase Report 4

1.0 Introduction 5

2.0 Project Justification

2.1 **Client Background**

2.2 **Opportunity**

2.3 **Project Objective**

3.0 Project Scope 6

4.0 Project Methodology 8

5.0 Project Plan 9

System Analysis Phase Report 10

6.0 Requirements 10

6.1 Business Requirements 10

6.2 User Requirements 10

6.3 Functional Requirements 11

6.4 Non-Functional Requirements 12

7.0 Data Model - Context Diagram 13

8.0 Level 0 Diagram 14

8.0.1 Level 0 Table 16

9 Swimlane Diagrams 17

9.0 Payment Processing Process 17

9.1 Nearest Bus Route Request process 18

System Design Phase Report 19

10 Physical ER Diagram 19

10.0 Entities 19

11 Design Architecture 21

12 User Interface 23

12.0 Signup and Login Pages 24

12.1.1 Create account page

12.1.2 Sign up page

12.1.3 Log in page

12.1 Payment Page 26

12.2 Analytics Page 27

12.3 Parking Availability Space 28

13 Structure Chart 29

14 Testing Plans 30

14.0 Unit Testing 30

14.1 Stress Testing 31

14.2 Integration Testing 32

14.3 Systems Testing 33

14.4 Acceptance Testing 35

15 Implementation Plan 38

15.0 Objectives 38

15.1 Implementation Plan Schedule 39

16 Conclusion 40

Department of Transportation Parking System

Client: The University of Maryland, College Park (UMD)
Team:13

Executive Summary

This report describes the features of a Parking web portal information system developed for the Department of Transportation Services at the University of Maryland. The system's major capabilities will comprise a secure database to record live parking space data, a notification center for visitors, an analytic system for data intelligence and parking analytics, and integration with an external payment gateway. The team will employ both prototype and agile project techniques. Finally, this project is projected to take 88 days to finish.

System Planning Phase Report

1.0 Introduction

A diversified community of more than 50,000 students, teachers, and staff members in the City of College Park are served by a full range of parking and transportation services from the University of Maryland Department of Transportation Services (DOTS). In addition, they are also dedicated to creating and fostering a vibrant, diversified workforce that offers both full-time career options for adults and educational opportunities for students.

Planning, educating, and enforcing are just a few of the services that DOTS is committed to offering to the campus community. On the College Park campus, DOTS is the main organization in charge of managing the parking and transportation management programs. DOTS adopted the LPR (license plate recognition technology) early on. In 2019, DOTS partnered with Veo to test a bike and electric scooter (e-scooter) program over the course of a year. At 24 locations, the fleet consisted of 150 e-bikes, 70 pedal cycles, and 70 e-scooters. In 2022, DOTS launched the Transit app integrated with all the Shuttle-UM information, which is the University of Maryland's official shuttle service. Therefore it's quite evident that DOTS has been quite adaptable to accommodate new technologies as needed.

However, due to the increasing number of admissions every year, lately, there has been a situation of overcrowded parking lots during the peak hours in the morning when students and faculty rush to classes. To tackle this problem, DOTS can yet again leverage a Smart Parking system that includes installing a new sensor technology integrated with a web portal to provide real-time information on the currently available parking spaces in a given lot at any given time. In addition, implementing this new technology also aligns with the key mission of DOTS to maintain a sustainable environment and work to embrace alternative transportation systems and programs that promote social good; in this case, significantly saving time for the drivers and reducing the emissions from vehicles as they circle around to find a parking spot.

2.0 Design Architecture

2.1 Client Background

University of Maryland (UMD), College Park, is a public university located in College Park, Maryland. There are more than 41,300 students and 14,000 faculty and staff members. According to the university's Department of Transportation Services, there are approximately 17,000 parking spaces available on campus, spread across more than 80 parking lots and garages. The university offers several parking options, including visitor parking, student parking, and faculty/staff parking, but parking permits are required to park on campus for most lots. Some lots are reserved for specific groups of people, and parking restrictions and fees vary depending on the lot and the time of day.

2.2 Opportunity

Although many students have to purchase a parking permit to park during busy hours on campus, it is still a nightmare to find a space to park. The parking permit does not assign a specific parking space and students still need to drive around the crowded parking lots to find an empty space. There are also reports saying DOTS is overselling the permits, allowing more vehicles entering than the lots can handle. To get a parking permit, however, students have to request long before they can actually purchase one, even for those who live far from campus.

Few modern technologies have been deployed in the parking space, thus both motorists and DOTS cannot know very well about the situation of the parking lots. Motorists can not know where they can find a space to park and thus have to drive around for a long time. There is no real-time indicator which informs motorists of essential parking information such as number of empty spaces and temporary parking restrictions. The administrators have to patrol around the parking lot to check if there is any parking violation.

Information technologies can surely improve the current situation of parking lots in UMD, College Park.

2.3 Project Objective

The main objective of the Smart Parking system is to help Department of Transportation Services better manage the campus visitor parking lots. The system can gather the information of vehicles in the parking lots, manage reservations and violations, and disperse vehicles to free parking spaces.

3.0 Project Scope

- Parking Management System:
  The first point in the scope of the project would be to develop a comprehensive parking management system that automates and digitizes the entire parking process. This system will include features such as real-time parking availability updates, online booking of parking spots, and automated payment processing. It will also provide an interface for students, faculty, and visitors to easily find and reserve parking spots on-campus, reducing the likelihood of overcrowding.

- Data Management and Analytics:
  Another essential point in the scope of the project would be to design a data management system that can store all parking-related data, such as occupancy rates, payment history, and parking violations. This data will be used to generate a live dashboard for the department of transportation, providing them with an overview of parking activity on-campus. Additionally, data analytics will be employed to predict parking trends, helping to plan for future parking requirements and identifying areas for improvement.

- Integration with Existing Systems:
  To ensure seamless integration with the existing university infrastructure, the parking system will be integrated with other systems, such as the university's student and faculty databases. This integration will enable automated verification of student and faculty parking permits, eliminating the need for manual validation.

- Robust Security and Privacy:
  Finally, it is important to ensure that the parking system is secure and respects user privacy. The system will implement industry-standard security measures, such as SSL encryption, two-factor authentication, and regular vulnerability assessments. Additionally, user data will be protected under the university's privacy policy, ensuring that personal data is not shared with any third-party service without explicit consent.

- Automatic Penalty Assessment and Payment:
  The parking system will have a mechanism to automatically assess and charge penalty fees for parking violations such as overstaying, occupying an unauthorized parking spot, or not paying

for parking. The system will send notifications to the vehicle owner and deduct the penalty fee from their registered payment method.

- Real-Time Notifications and Alerts:
  The parking system will provide real-time notifications and alerts to users, informing them of parking availability, their parking spot location, and when their parking time is about to expire. This will ensure that users are always informed and can manage their parking experience seamlessly.

- Integration with Public Transportation:
  The parking system will integrate with public transportation systems, such as university shuttles, to offer users an alternative mode of transportation from remote parking areas. This will help reduce traffic congestion and provide users with convenient options to reach their destination on-campus

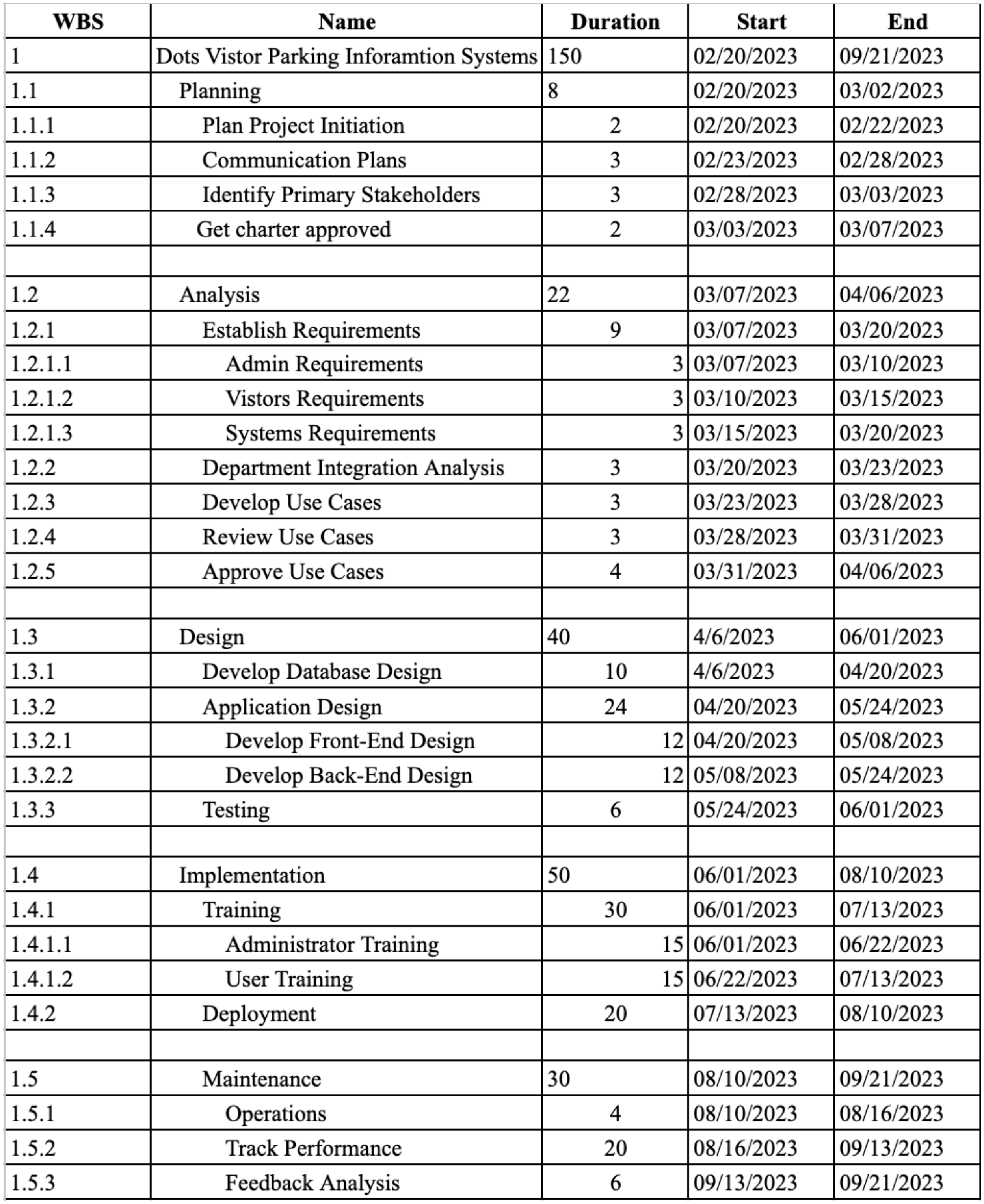
4.0 Project Methodology

Of the various system development methods, Hybrid of system prototyping and agile methodology best fits our goal.

Analysis: This project has evolving user requirements, requires consistent customer/user feedback, builds off of an existing model, and has a short time constraint. Comparison to other development models is as follows:

Ability to develop systems with	**Waterfall**	**Agile Development**	**System Prototyping**
Unclear User Requirements	Poor	Excellent	Excellent
Customer Feedback	Poor	Excellent	Excellent
Existing Model	Poor	Average	Excellent
A Short Time Constraint	Poor	Excellent	Excellent
Change Management	Poor	Excellent	Average
On-going Maintainance	Good	Excellent	Excellent

5.0 Project Plan



WBS	Name	Duration	Start	End
1	Dots Vistor Parking Inforamtion Systems	150	02/20/2023	09/21/2023
1.1	Planning	8	02/20/2023	03/02/2023
1.1.1	Plan Project Initiation	2	02/20/2023	02/22/2023
1.1.2	Communication Plans	3	02/23/2023	02/28/2023
1.1.3	Identify Primary Stakeholders	3	02/28/2023	03/03/2023
1.1.4	Get charter approved	2	03/03/2023	03/07/2023
1.2	Analysis	22	03/07/2023	04/06/2023
1.2.1	Establish Requirements	9	03/07/2023	03/20/2023
1.2.1.1	Admin Requirements	3	03/07/2023	03/10/2023
1.2.1.2	Vistors Requirements	3	03/10/2023	03/15/2023
1.2.1.3	Systems Requirements	3	03/15/2023	03/20/2023
1.2.2	Department Integration Analysis	3	03/20/2023	03/23/2023
1.2.3	Develop Use Cases	3	03/23/2023	03/28/2023
1.2.4	Review Use Cases	3	03/28/2023	03/31/2023
1.2.5	Approve Use Cases	4	03/31/2023	04/06/2023
1.3	Design	40	4/6/2023	06/01/2023
1.3.1	Develop Database Design	10	4/6/2023	04/20/2023
1.3.2	Application Design	24	04/20/2023	05/24/2023
1.3.2.1	Develop Front-End Design	12	04/20/2023	05/08/2023
1.3.2.2	Develop Back-End Design	12	05/08/2023	05/24/2023
1.3.3	Testing	6	05/24/2023	06/01/2023
1.4	Implementation	50	06/01/2023	08/10/2023
1.4.1	Training	30	06/01/2023	07/13/2023
1.4.1.1	Administrator Training	15	06/01/2023	06/22/2023
1.4.1.2	User Training	15	06/22/2023	07/13/2023
1.4.2	Deployment	20	07/13/2023	08/10/2023
1.5	Maintenance	30	08/10/2023	09/21/2023
1.5.1	Operations	4	08/10/2023	08/16/2023
1.5.2	Track Performance	20	08/16/2023	09/13/2023
1.5.3	Feedback Analysis	6	09/13/2023	09/21/2023

6.0 Requirements

- **6.1 Business Requirements (what the business needs):**
  - Provide an effective parking tracking system that is web-based.
  - Provide real-time information on the available parking spaces.
  - Provide analytical reports of frequently visiting motorists.
  - Provide an online reservation system to book parking in advance.
  - Provide parking notifications and alerts on customers’ mobile devices.
  - Provide a system that integrates with the existing databases seamlessly.
  - Provide capabilities to interact with the current transportation applications.
- **6.2 User Requirements (what the users need to do):**
  - Motorists
    - Should be able to access the automated parking system through auser-friendly web portal
    - Should be able to make parking reservations
    - Should be able to easily find available parking spots on-campus.
    - Should be able to pay for parking using a variety of payment methods.
    - Should be able to receive real-time notifications and alerts about parking availability, parking time expiration, and penalty assessments.
  - Facilities Staff
    - Should be able to create and manage parking permits for all motorists
    - Should be able to manage parking enforcement operations.
    - Should be able to manage and optimize parking areas based on usage patterns
  - IT Department
    - Should be able to maintain and troubleshoot the parking system's technology infrastructure.
    - Should be able to integrate the parking system with other university systems, such as security and transportation systems, to ensure seamless operation.
    - Should be able to upgrade and scale the parking system's technology infrastructure as needed.
    - Should be able to provide technical support to other users.
  - Analytics Department
    - Should be able to access comprehensive parking data and analytics, including occupancy rates, trends, and patterns.

    - Should be able to access detailed reports on parking usage and occupancy for different parking areas.
    - Should be able to conduct advanced data analysis and predictive modeling to optimize parking availability and usage.
- Senior Management
    - Should be able to view comprehensive reports on parking usage and occupancy across different areas of the university.
    - Should be able to monitor and assess the impact of the parking system on transport operations and revenue.

6.3 Functional Requirements (what the software should do):

- PROCESS
    - The system should be integrated to process payments
    - The system must only allow authorized customers to access data on the website
    - The system must be able to process reservation requests
    - The system must be able to show garage parking level both online and on site
    - The system should be integrated with university shuttles and public shuttle to provide alternative routes
- INFORMATION
    - Should store User data
    - Should store guidance information
    - Should store parking data Information
    - Should store payment information
    - Should store sensor information
    - Should store login information

6.4 Nonfunctional requirements (characteristics the system should have):

- OPERATIONAL
    - The web pages can be accessed on computers, tablets and mobiles.
    - The notifications should work even under adverse weather conditions.
- PERFORMANCE
    - The system should be accessible 24 hours a day 365 days per year
    - The server updates the parking information to every indicator and corresponding web pages every second.
    - The server receives the signal from sensors and cameras after the space is occupied or unoccupied within 30 seconds.
    - The system should be able to allow 300 concurrent users.
    - The web pages can handle the data flow that 10,000 motorists on campus are accessing at the same time.
    - The server updates the public transportation information every time it is changed.

- The system can process parking payments, spot reservations, and violation payments in 5 seconds.
- Mails or texts should be sent to vehicle owners or registered drivers within 5 mins if parking violations are captured.
- Notifications should be sent to registered drivers within 5 seconds if there’s a parking alert or notification involving him or her.

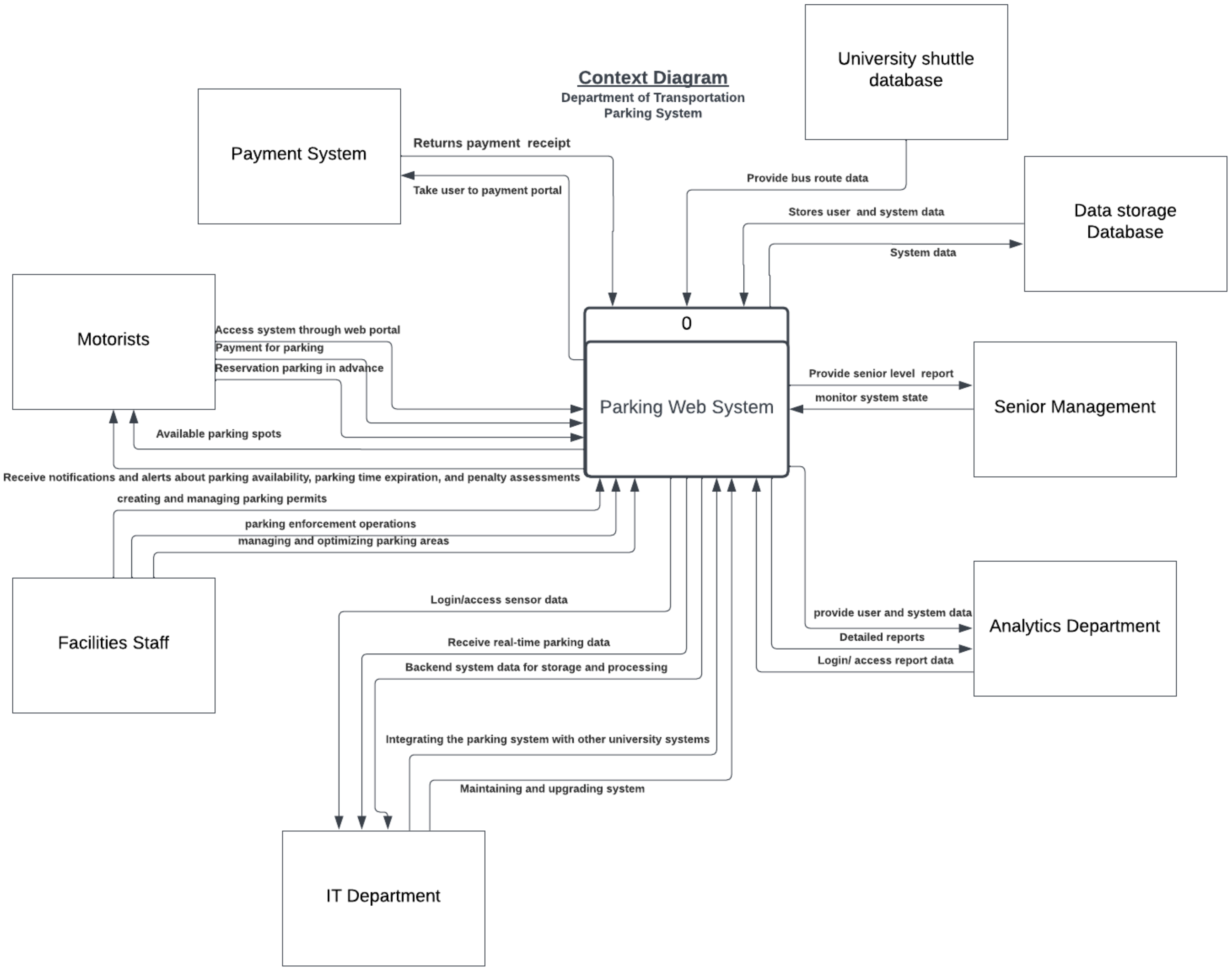
➢ SECURITY

- The system should use the latest encryption to protect the confidentiality of data.
- Only essential motorist information will be collected, such as name and driving license.
- All UMD related transactions will be processed by UMD billing system.
- The system will implement industry-standard security measures, such as SSL encryption, two-factor authentication, and regular vulnerability assessments.

➢ CULTURAL & POLITICAL

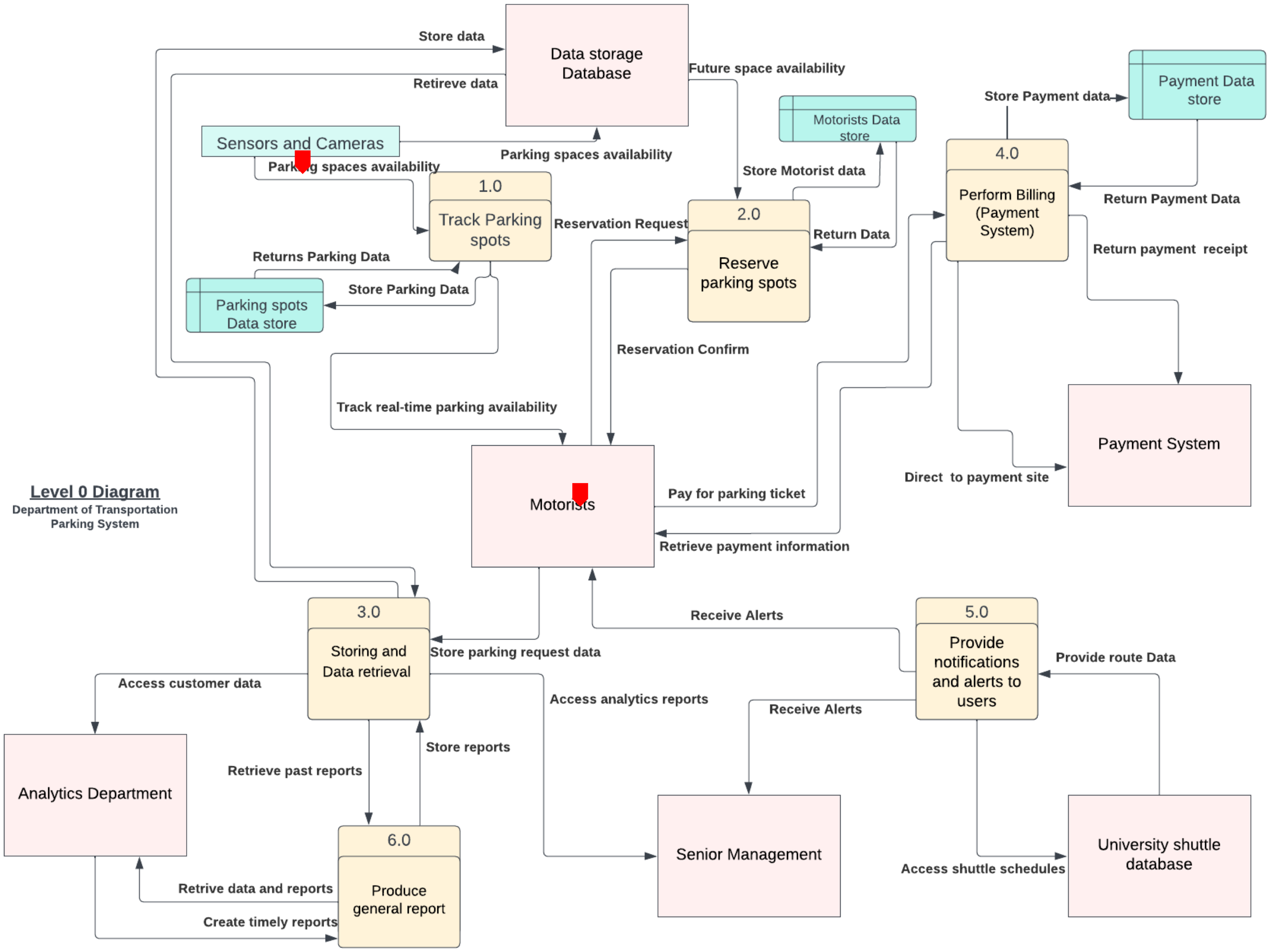
- The system should follow regulations on data privacy and Data Protection Act, USA.
- The System should support English and Spanish and have the potential to support more languages. Other languages should be supported if there’s a large group of other-language-speaking motorists who do not use current languages.
- User data will be protected under the university's privacy policy.

7.0 Data Model- Context Diagram


Context Diagram
Department of Transportation
Parking System
University shuttle database
Payment System
Returns payment receipt
Take user to payment portal
Provide bus route data
Stores user and system data
System data
Data storage Database
0
Parking Web System
Motorists
Access system through web portal
Payment for parking
Reservation parking in advance
Available parking spots
Receive notifications and alerts about parking availability, parking time expiration, and penalty assessments
creating and managing parking permits
parking enforcement operations
managing and optimizing parking areas
Provide senior level report
monitor system state
Senior Management
Facilities Staff
Login/access sensor data
Receive real-time parking data
Backend system data for storage and processing
Integrating the parking system with other university systems
Maintaining and upgrading system
provide user and system data
Detailed reports
Login/ access report data
Analytics Department
IT Department

Description: There are 8 external entities namely the Payment system, University Shuttle Database, Data storage database, Senior Management, Analytics Department, IT Department, Facilities staff, and Motorists which exchange information through the system, and the actions are performed outside the system. Here, the context diagram is used to define and clarify the boundaries of the system, clearly depicting the flow between the parking web system and external entities.

8.0 Level 0 Diagram



Description: There are a total of 6 processes in the level 0 diagram, namely Track Parking spots, Reserve parking spots, Storing and data retrieval, Perform Billing (Payment system), Provide notifications and alerts, and, Produce general reports, on which actions are performed within the system. Speaking of external entities, there are 6, namely Data storage database, Motorists, Payment System, Analytics Department, Senior Management, and, University shuttle database which communicate with the processes and data stores cohesively. Finally, the system comprises of 4 data stores, including ones for Parking spots, sensors & cameras, Motorists data, and Payment data.

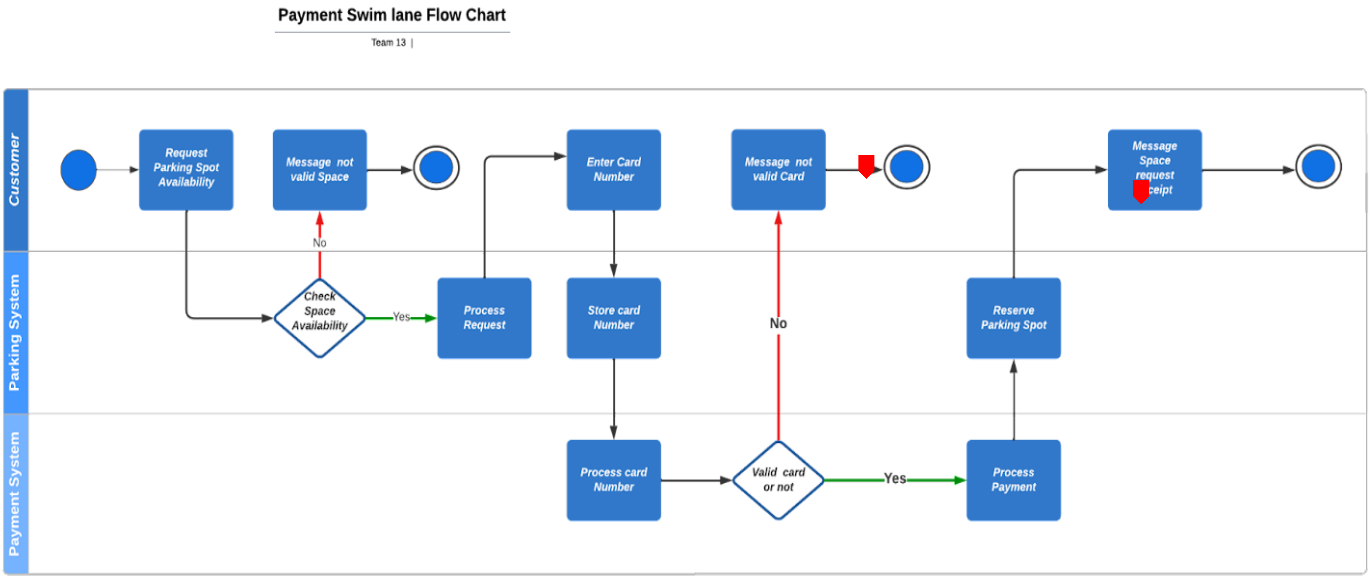
8.0.1 Level 0 Table

Process ID	Process	Input	Output
1.0	Track Parking spots	● Parking spaces availability ● Returns parking data	● Store Parking Data ● Track real-time parking availability
2.0	Reserve parking spots	● Reservation request ● Future space availability ● Return Data	● Reservation confirm ● Store motorist data
3.0	Storing and data retrieval	● Retrieve data ● Store parking request data ● Store reports	● Store data ● Access customer data ● Retrieve past reports ● Access analytics reports
4.0	Perform Billing	● Return Payment Data ● Pay for parking ticket	● Store Payment data ● Retrieve

			payment information ● Direct to payment site ● Return payment receipt
5.0	Provide notifications and alerts to users	● Provide route data	● Receive alerts ● Access shuttle schedules ● Receive alerts
6.0	Produce general reports	● Retrieve past reports ● Create timely reports	● Retrieve data and reports ● Store reports

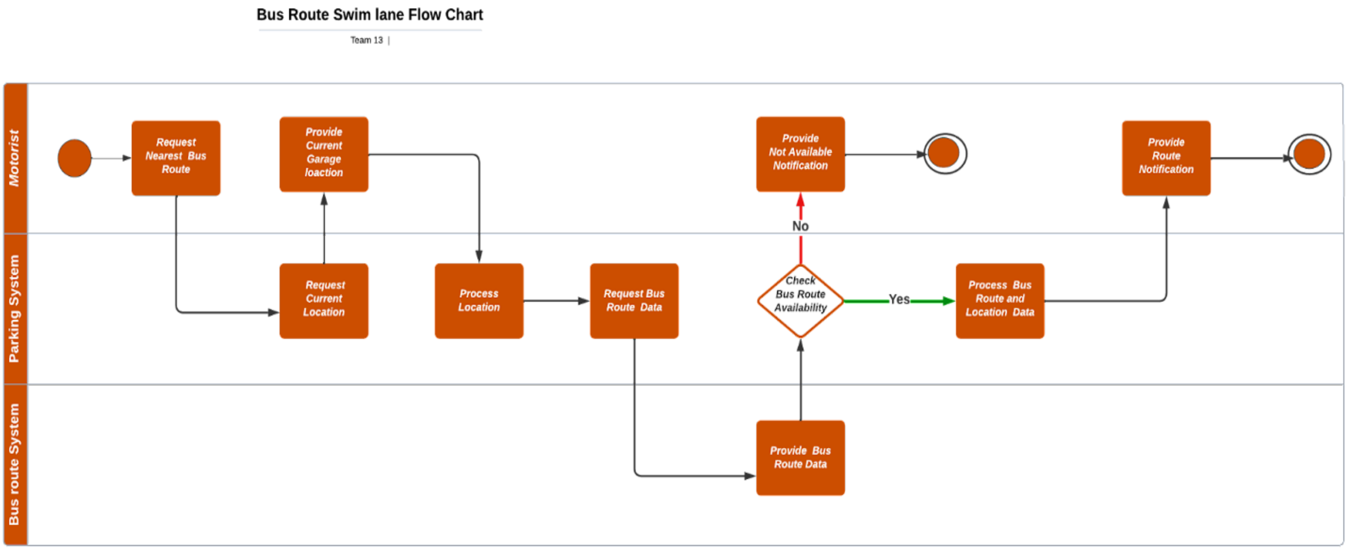
9.0 Swimlane Diagrams

9.1 Payment processing process



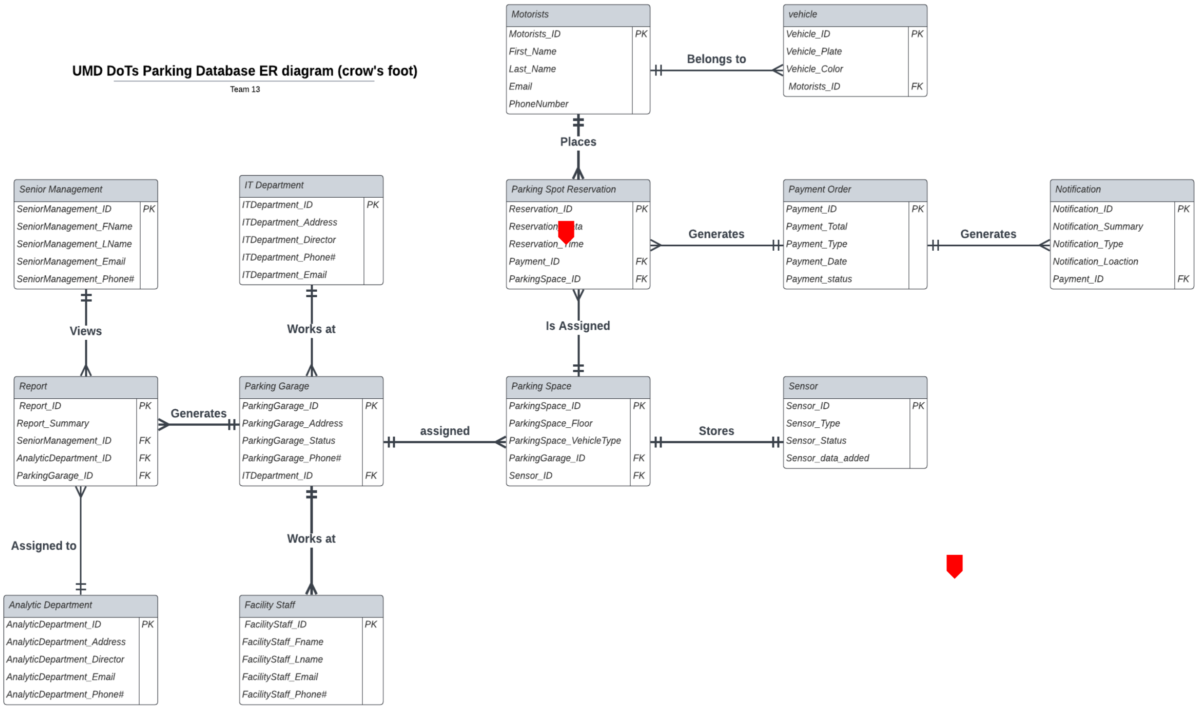
Description: This swimlane diagram is showing the flow of the process of checking and paying for reserving parking spots using the proposed smart parking system. From requesting availability of spots to making a successful reservation after validation of identity and payment to the final responses for each case. This process involves customers, the parking system and its interface with the payment system.

9.2 Nearest Bus Route Request Process


Bus Route Swim lane Flow Chart
Team 13 |
Motorist
Parking System
Bus route System
Request Nearest Bus Route
Provide Current Garage loaction
Request Current Location
Process Location
Request Bus Route Data
Provide Bus Route Data
Check Bus Route Availability
No
Yes
Provide Not Available Notification
Process Bus Route and Location Data
Provide Route Notification

Description: This diagram follows the process of requesting closest bus route info based on current location. It involved motorists, the parking system and the university bus route system.

10.0 Physical ER Diagram



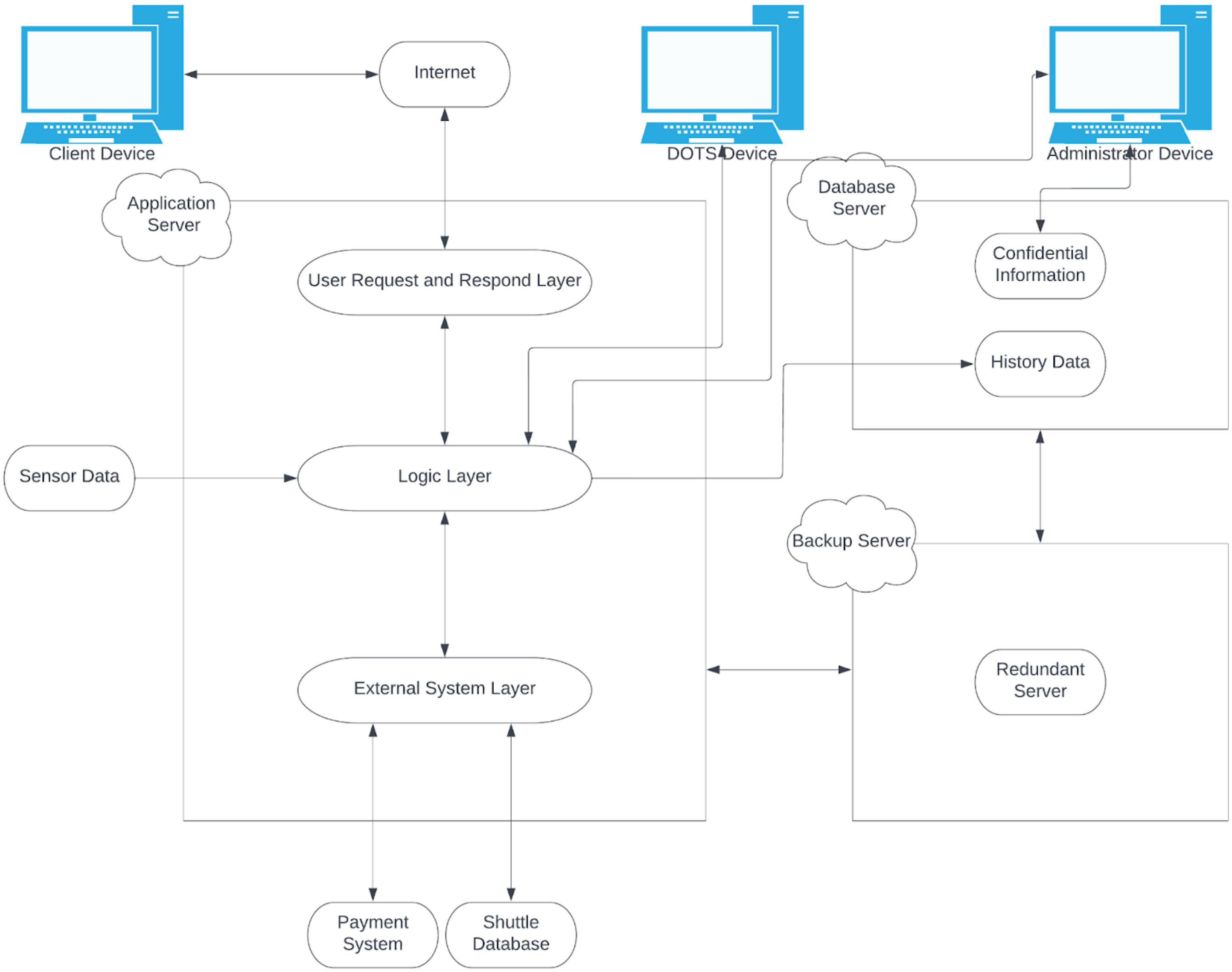
Description: There are a total of 13 entities in the physical Er diagram, namely Senior Management, IT Department, Report, Parking Garage, Analytics Department, Facilities Staff, Parking Space, Parking Spot Reservation, Motorists, Vehicle, Payment Order, Sensor, Notification. The relationship between all the entities is indicated by the cardinality

10.1 Entities

- Senior Management (Primary Key = SeniorManagement_ID): Contains information regarding the Senior Management personnel such as their First Name, Last Name, their email and phone number.
- IT Department (Primary Key = ITDepartment_ID): Contains information regarding the IT Department such as its Address, director name of the department, phone number and email associated with the department.
- Report (Primary Key = Report_ID): Contains information regarding the report summary.

- Parking Garage (Primary Key = ParkingGarage_ID): Contains information about the garage such as its location(Address), availability status, phone number associated with garage.
- Analytics Department (Primary Key = AnalyticsDepartment_ID): Contains information about the analytics department such as the its director, Address of the department, its contact information such as email, and phone number.
- Facilities Staff (Primary Key = FacilityStaff_ID): contains information about the facilities staff such as their Firstname, Last name, email and phone number.
- Parking Space (Primary Key = ParkingSpace_ID): Contains the information regarding the floor where there is availability of parking space, the type of vehicle the parking spot accomodates.
- Parking Spot Reservation (Primary Key = Reservation_ID): Contains information about the reservations, teh time the reservations are made.
- Motorists (Primary Key = Motorists_ID): Contains information about the motorists such as their First name, Last name, and contact information as email and phone number.
- Vehicle (Primary Key = Vehicle_ID): Contains information regarding the vehicle as its color and vehicle plate number.
- Payment Order (Primary Key = Payment_ID): Contains information regarding the payments type, payment status, date of payment made and the total amount of payment.
- Sensor (Primary Key = Sensor_ID): Contains information about the sensor type, sensor status and data collected by sensor.
- Notification (Primary Key = Notification_ID): Contains information regarding notifications type, notification messages summary, payments id, and notification location.

11.0 Design Architecture



1. Three-tiered client-server architecture

1.1 Servers are located on campus to reduce delay and enhance sensor and internet connection.
1.2 Backup servers are set for redundancy.
1.3 Industry-standard security features will be implemented, such as SSL encryption, two-factor authentication, and regular vulnerability assessments.

2. Client devices

Users can access the system via the Internet, on common browsers.

3. Application server

3.1 Application server handles all real-time parking services.
3.2 Application server collects all sensor data.
3.3 User request and respond layer handles all demands from users and sends notifications.

3.4 Logic layer processes all real-time information.
3.5 Staff can access the logic layer.
3.6 The payment system and university shuttle database is connected to the logic layer through the external system layer.

4. Database server

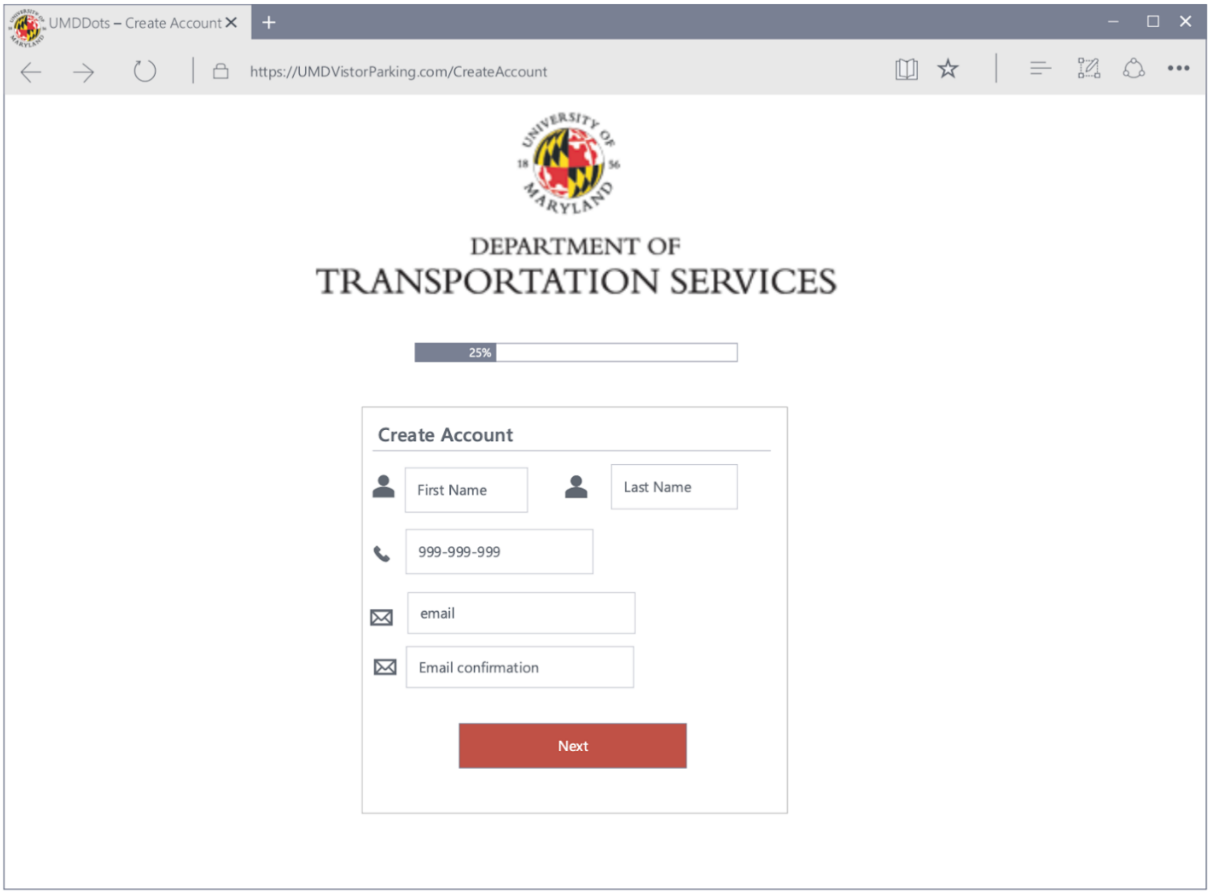
4.1 Only administrator staff can have access to confidential information such as motorist personal information and edit it.
4.2 Database server stores history data created by logic layer from application server.

12. User Interface

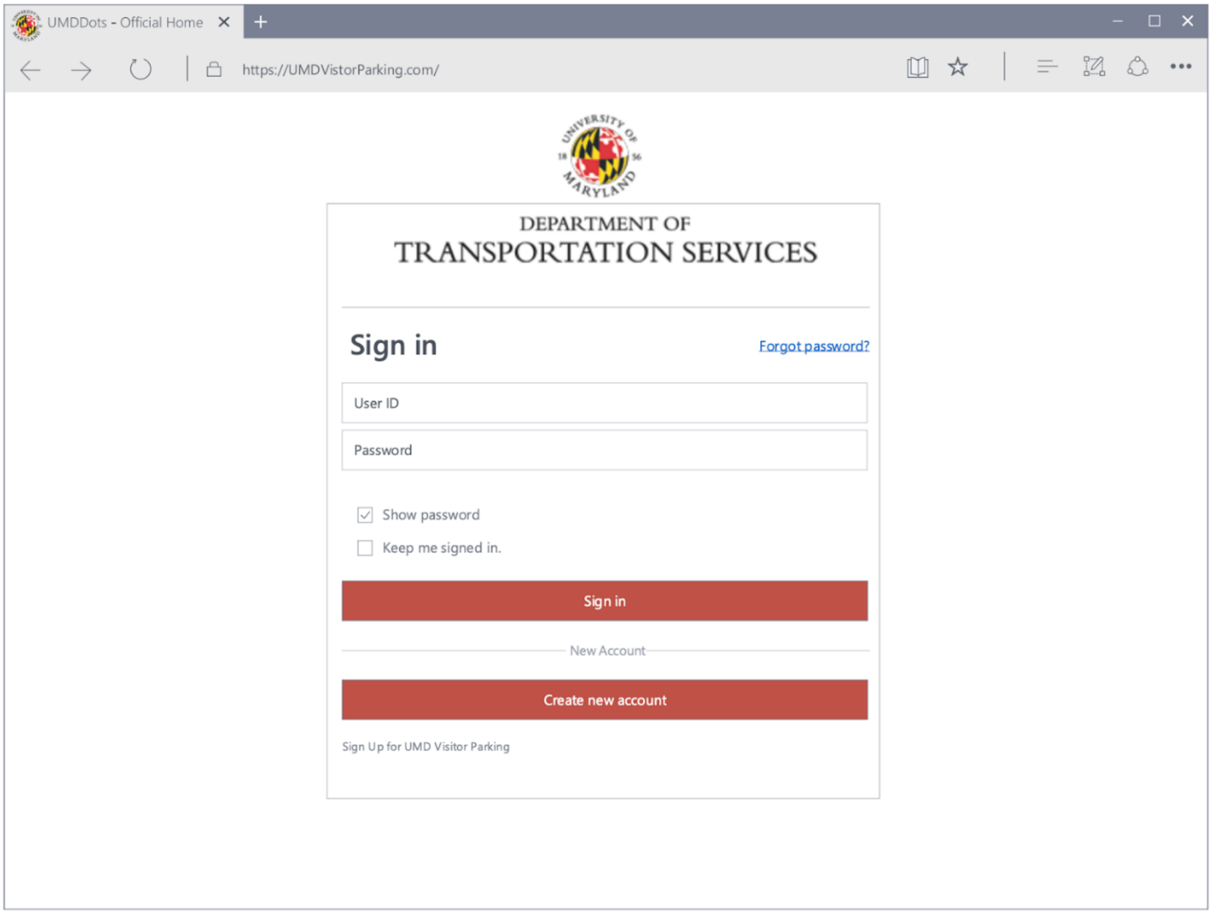
12.1 Signup and Login Pages

The following screens are the Signup and Login pages which will be available to both the motorists and facilities staff.

12.1.1 Create account page:



12.1.2 Sign up page:



UMDDots - Official Home

https://UMDVistorParking.com/

DEPARTMENT OF
TRANSPORTATION SERVICES

Sign in

Forgot password?

User ID

Password

☑ Show password

□ Keep me signed in.

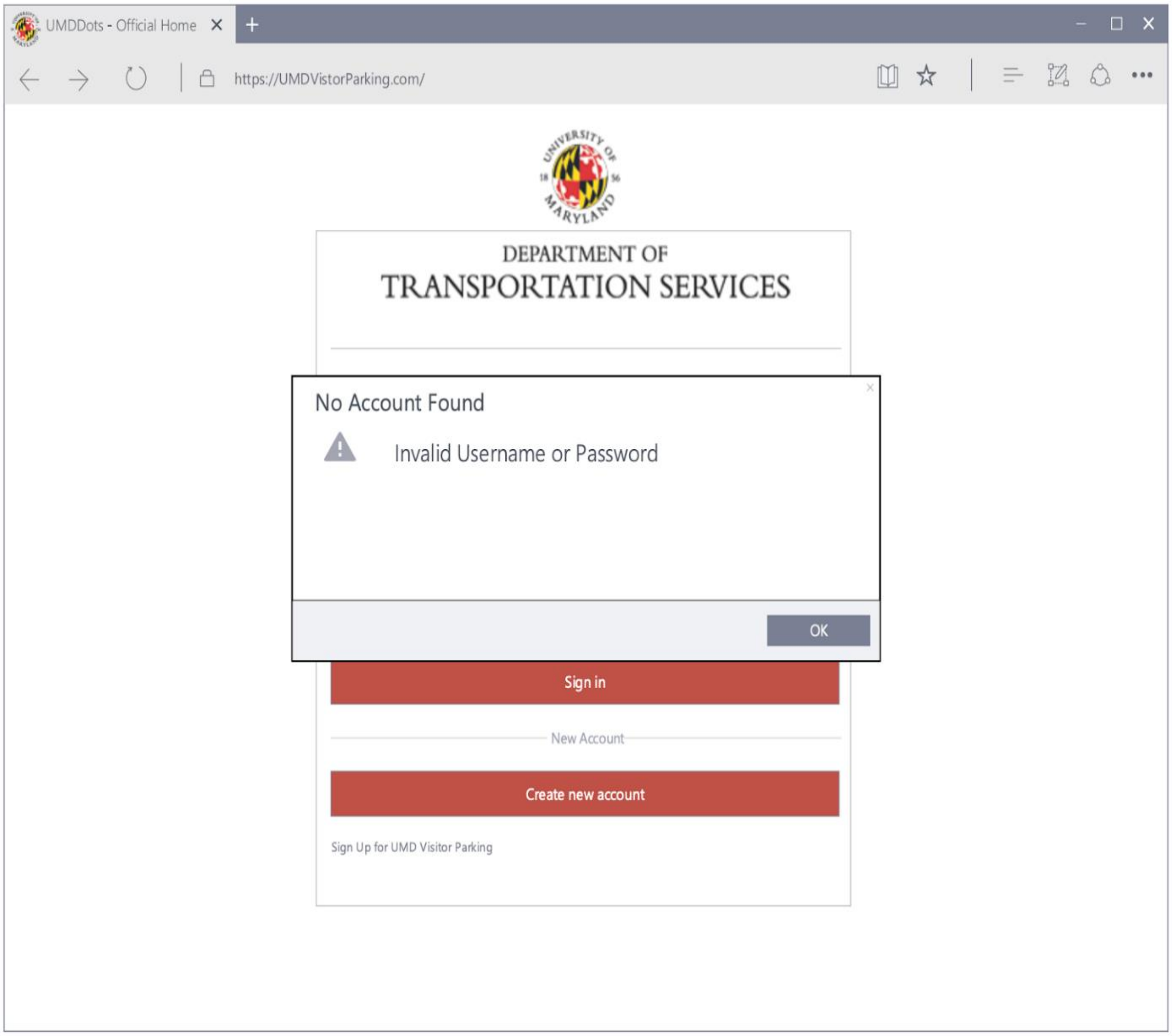
Sign in

New Account

Create new account

Sign Up for UMD Visitor Parking

12.1.3 Log in page:

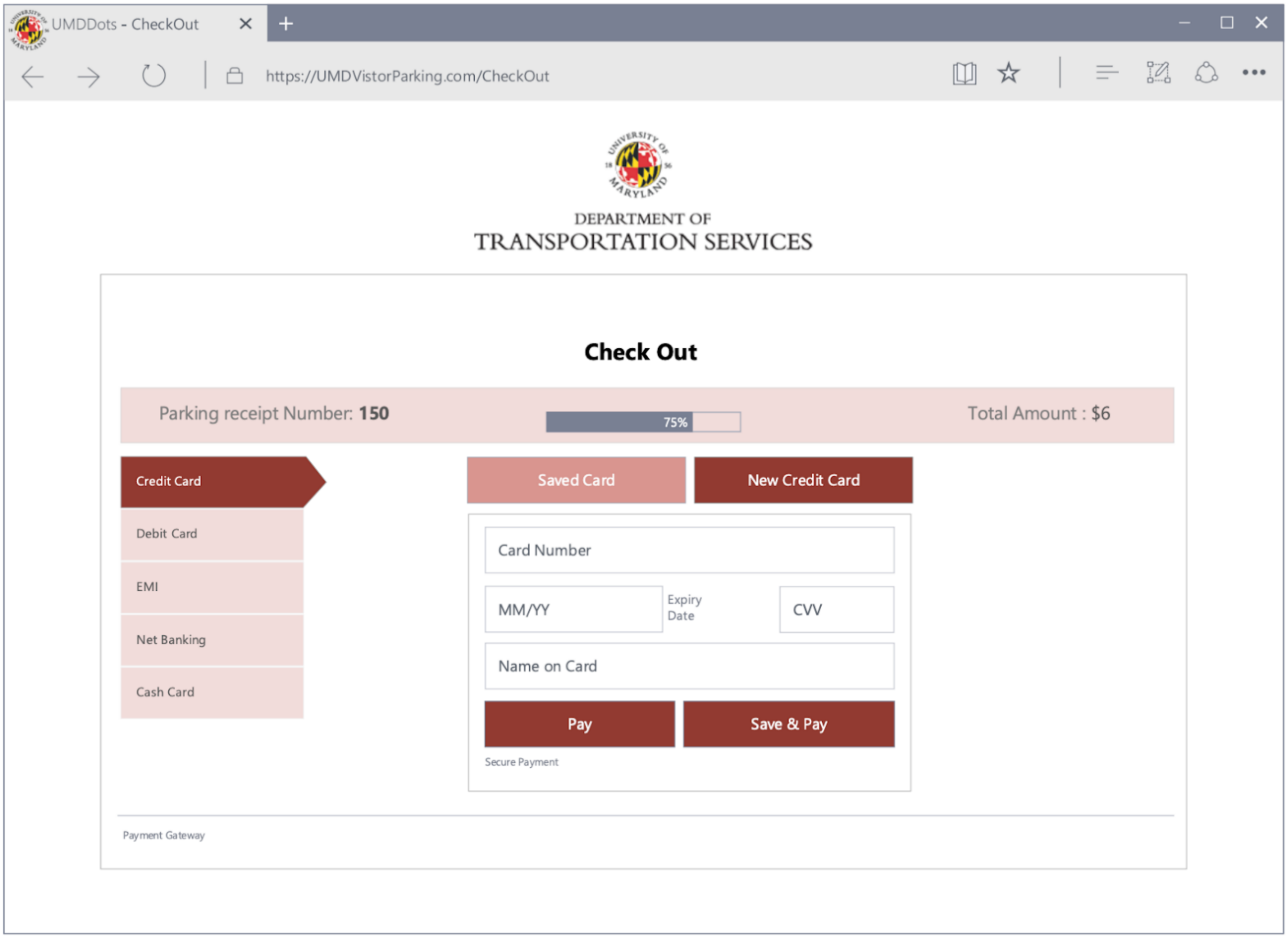


No Account Found

Invalid Username or Password

12.2 Payment Page

This page acquires the payment information from users.



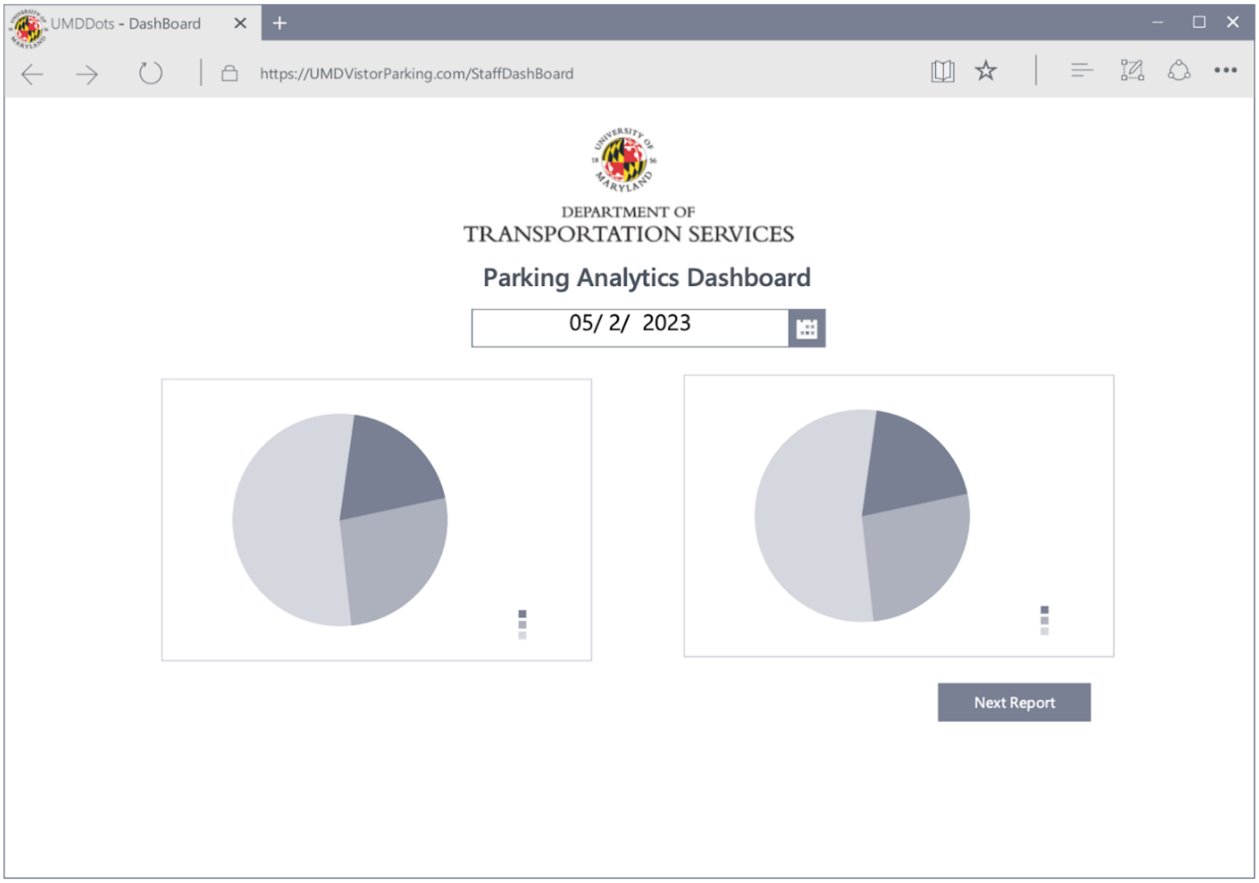
https://UMDVistorParking.com/CheckOut

Parking receipt Number: **150**

Total Amount : $6

12.3 Analytics Page

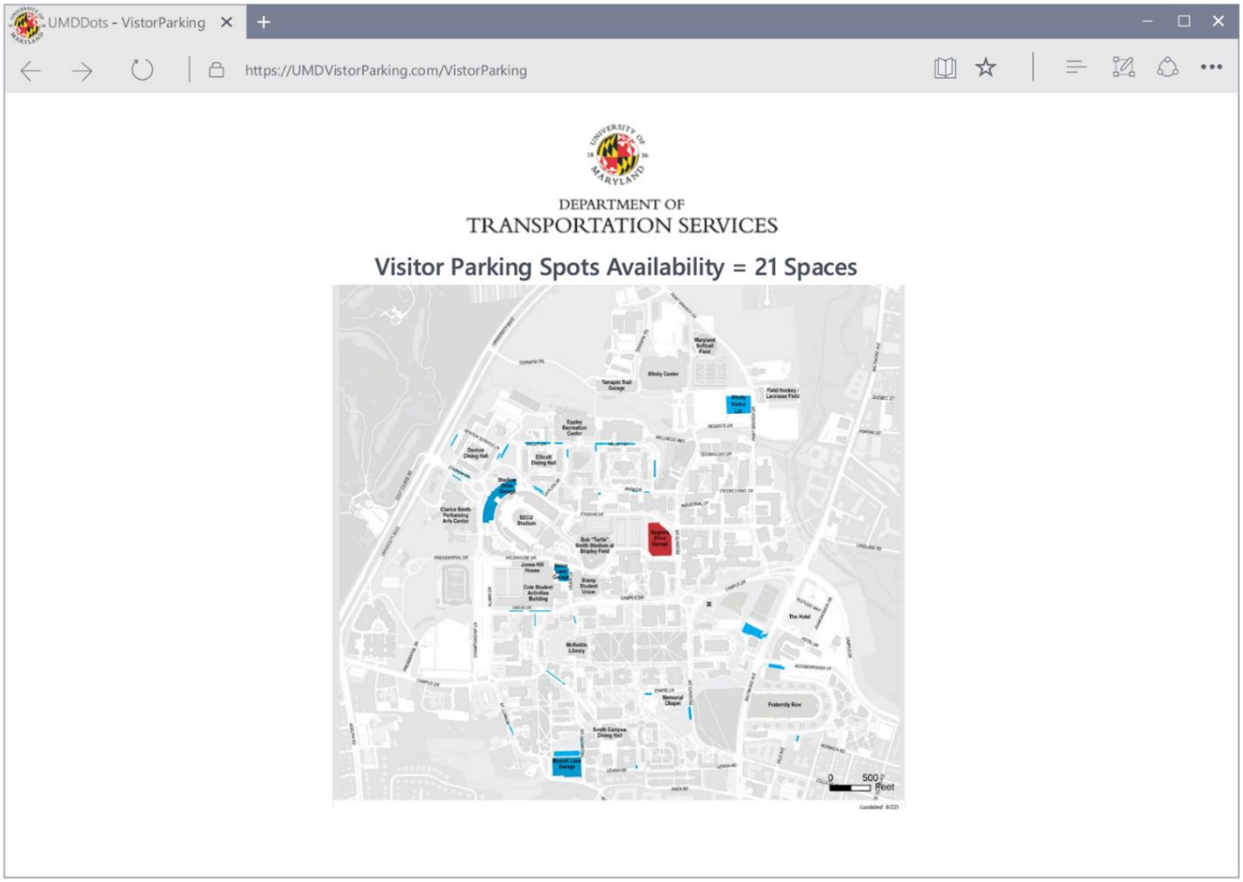
This page generates a brief analytics report about the parking situation on campus. Users can select dates and information type based on their needs.



UMDDots - DashBoard
https://UMDVistorParking.com/StaffDashBoard
DEPARTMENT OF
TRANSPORTATION SERVICES
Parking Analytics Dashboard
05/ 2/ 2023
Next Report

12.4 Parking Availability Space

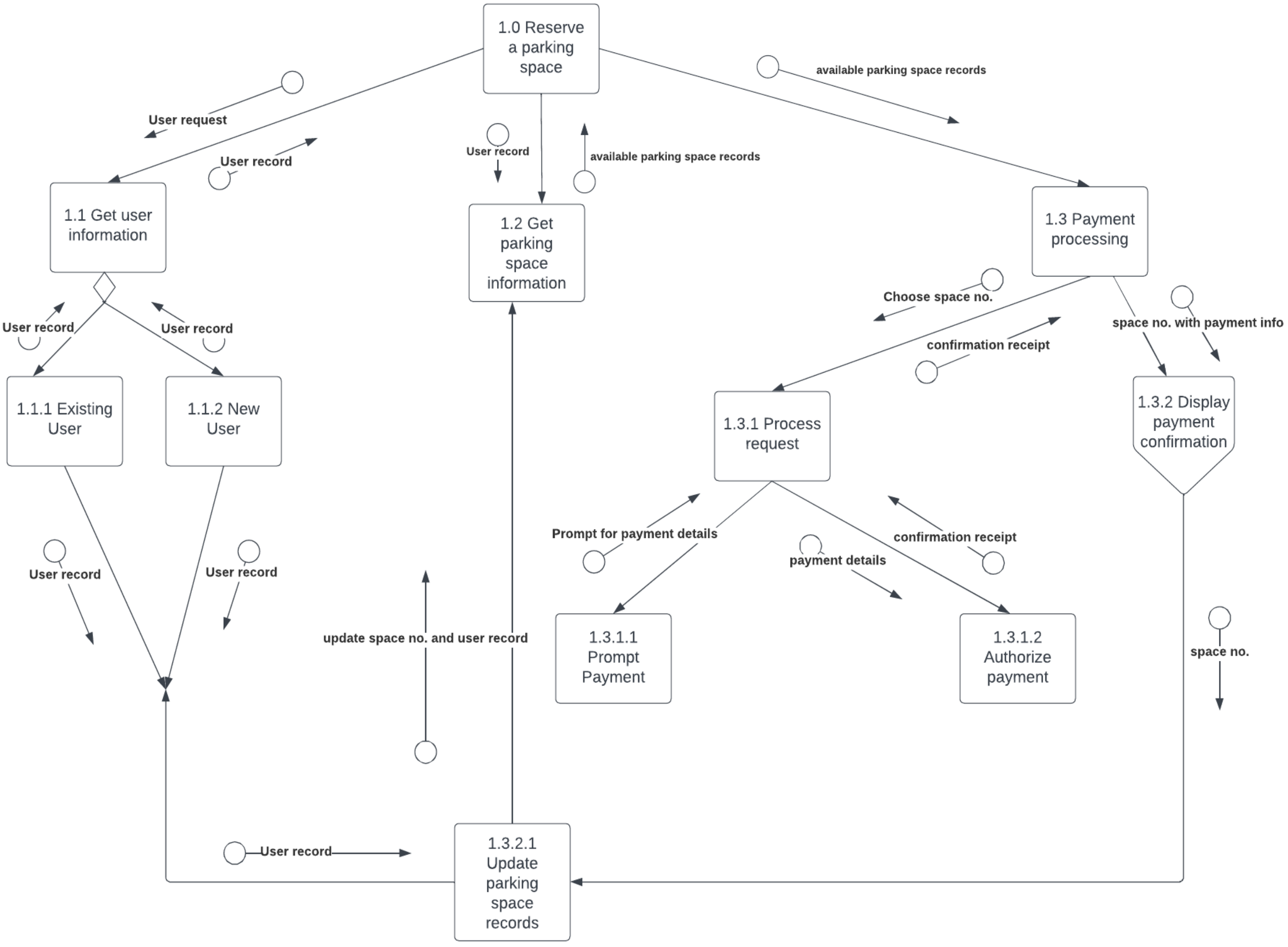
The page provides users with the current available parking spots on campus so that motorists can find them easily.



Visitor Parking Spots Availability = 21 Spaces

13. Structure Chart

Description: The structure chart below perfectly summarizes how different program modules that focus on searching and updating user information, fetching parking space information and processing payments cohesively undergo coupling with each other, all controlled by the single module 'Reserve a parking space'.


1.0 Reserve a parking space
User request
User record
User record
available parking space records
available parking space records
1.1 Get user information
1.2 Get parking space information
1.3 Payment processing
User record
User record
Choose space no.
space no. with payment info
confirmation receipt
1.1.1 Existing User
1.1.2 New User
1.3.1 Process request
1.3.2 Display payment confirmation
Prompt for payment details
confirmation receipt
payment details
User record
User record
update space no. and user record
1.3.1.1 Prompt Payment
1.3.1.2 Authorize payment
space no.
User record
1.3.2.1 Update parking space records

14. Testing Plans

14.1 Unit Testing

During unit testing, we will be testing individual units of each function or process to ensure that they are working as expected and that there are no issues present. The goal is to test each individual unit or component of the website separately in order to detect if there are any flaws in the code during the early stages of development. Unit testing is an important step in the development process, because if done correctly, it can help detect early flaws in code which may be more difficult to find in later testing stages.

Unit Test Scenarios:

Test Case #	Test Case Name	Test Case Description	Inputs	Expected Output
1	Account Creation	Checking creation of user accounts	Valid user email id, valid user Password	The user account must be successfully created
2	Account Login	Checking login of existing user accounts	Valid existing user email id, valid existing user password	User must be able to log into their account successfully
3	Reservation Process	Verify that the system can successfully process parking reservation requests from authorized customers.	appropriate parking slot selection	The user must be able to navigate to payments page
4	Payment Processing	Verify that the system can process payments for parking reservations and penalties in 5 seconds.	penalities fee request if exceeding time, parking requests from previous page	The user must be able to successfully make a parking slot reservation and reservation is visible on the parking spot indicators and the corresponding web pages The user must receive a penality notice for exceeding time in 5 seconds.
5	Notification Delivery	Verify that the system can send parking notifications and alerts to registered drivers within 5 seconds.	notification/alert requests for exceeding time, reminding of booking, payments	The user must receive parking notifications and alerts within 5 seconds through email or mobile
6	Access Control	Verify that only authorized customers can access data on the website, and that the system uses the latest encryption to protect the confidentiality of data.	username and password	Only authorised users should be able to login

7	sensor module	Verify that the sensor module is working correctly	sensors appropriate installed, sensor data	1. Sensors should detect the presence or absence of vehicles accurately. 2. Updated sensor data is in the system's database 3. System notifications to registered drivers if a sensor malfunctions or if there's an issue with a parking spot

14.2 Stress Testing

Stress testing is used to test software or hardware to check whether its performance is satisfying and is used to nullify the errors during every testing. Here, we have performed stress testing for different test cases to check if they are facing any errors due to process loading, underlocking and overlocking. Stress testing is commonly used for error handling. Any errors will be fixed during the testing so that the users will have satisfactory experience of the web portal for parking services.

Stress Test Scenarios:

Test Case #	Test Case Name	Test Case Description	Pre-condition	Expected Output
1	Account Creation	Checking creation of user accounts	login username, password	● Maximum number of accounts that can be created. ● Maximum number of accounts that can be created once. ● Maximum number of accounts that can be created per category (Admins, Drivers,vehicle Owners).
2	Account Login	Checking login of existing user accounts	Valid username, Valid password.	● Maximum number of logins that can be made per account. ● Maximum number of logins that can be made at once.
3	Maximum User Capacity	The system should be able to handle up to 300 concurrent users.	300 users accessing the system at the same time	The system should be able to handle 300 users accessing the system simultaneously.

4	Maximum Data Flow	The web pages should be able to handle the data flow that 10,000 motorists on campus are accessing at the same time.	10,000 motorists accessing the system at the same time.	The web pages should be able to handle the traffic without any significant delays or errors.
5	Reservation Requests	The system must be able to process reservation requests.	10,000 reservation requests at the same time.	The system should be able to handle the traffic without any significant delays or errors.
6	Payment Processing	The system should be able to process parking payments, spot reservations, and violation payments in 5 seconds	10,000 payment requests at the same time.	The system should be able to process the payments without any significant delays or errors.
7	Sensor Data Processing	The server receives the signal from sensors and cameras after the space is occupied or unoccupied within 30 seconds.	5000 sensor data signals at the same time	The system should be able to process the data signals without any significant delays or errors.
8	Notifications	Mails or texts should be sent to vehicle owners or registered drivers within 5 mins if parking violations are captured. Notifications should be sent to registered drivers within 5 seconds if there’s a parking alert or notification involving him or her.	5000 notifications at the same time.	The system should be able to send the notifications within the expected time frame without any significant delays or errors.
9	Real-time Information	The server updates the parking information to every indicator and corresponding web pages every second. The server updates the public transportation information every time it is changed.	5000 updates at the same time.	The system should be able to update the information in real-time without any significant delays or errors.

14.3 Integration Testing

For the following processes, we will evaluate the experience of the integrations associated with this new information system to ensure that each integration is working successfully and that there are no issues present. The goal is to test each interface separately and test the navigation between all interfaces to ensure that all accounts and the payment portal connect successfully with the main portal. Any common errors will need to be fixed in an update so that the users can have a successful experience with the integrations offered by this portal.

Integration/User Scenario Test Scenarios:

Test Case#	Test Case Name	Test case	Test Case Description	Pre-condition	Expected Output
1	Payment processing	Verify payment processing	Make a parking payment through the parking system	The parking system and UMD	The parking payment is processed correctly and

	integration	integration between parking system and UMD billing system	and verify that the payment is processed correctly through the UMD billing system.	billing system are integrated.	reflected in the UMD billing system.
2	Parking spot availability integration	Verify parking spot availability integration between parking system and sensors	Verify that the parking system correctly receives and displays real-time parking spot availability information from the sensors.	The parking system and sensors are integrated.	The parking system correctly displays real-time parking spot availability information.
3	Reservation processing integration	Verify reservation processing integration between parking system and reservation system	Make a parking reservation through the parking system and verify that the reservation is correctly processed through the reservation system.	The parking system and reservation system are integrated.	The parking reservation is processed correctly and reflected in the reservation system.
4	Notification integration	Verify notification integration between parking system and mobile devices	Verify that the parking system sends real-time notifications and alerts to customers' mobile devices correctly.	The parking system and mobile devices are integrated.	The parking system correctly sends real-time notifications and alerts to customers' mobile devices.
5	Shuttle integration	Verify shuttle integration between parking system and transportation applications	Verify that the parking system integrates with transportation applications correctly to provide alternative shuttle routes for parking customers.	The parking system and transportation applications are integrated.	The parking system correctly integrates with transportation applications to provide alternative shuttle routes for parking customers.
6	Analytics integration	Verify analytics integration between parking system and analytics department	Verify that the parking system provides comprehensive parking data and analytics to the analytics department correctly.	The parking system and analytics department are integrated.	The parking system correctly provides comprehensive parking data and analytics to the analytics department.
7	Security integration	Verify security integration between parking system and security systems	Verify that the parking system integrates with security systems correctly to ensure the security of parking customers and their vehicles.	The parking system and security systems are integrated.	The parking system correctly integrates with security systems to ensure the security of parking customers and their vehicles.

14.4 Systems Testing

During systems testing, we intend to make sure the entire system functions as desired; as opposed to testing individual functionalities like before, we aim to ensure that successful use of one functionality prepares the user to use the next functionality. This testing is done by the development team before launching the system for acceptance testing to be done by end users. These tests need to be done on both the drivers or user side, as well as the admin/support and transportation staff side to ensure functionalities work well with each other.

Systems Test Scenarios:

Test Case #	Test Case Name	Test Case	Test Case Description	Inputs	Expected Output
1	Login Functionality Test	Verify that the login functionality is working as expected	Attempt to login with valid and invalid credentials and verify that the system responds accordingly	The user must have an account registered with the system	1. If the user enters valid credentials, the system should grant access to the user's account 2. If the user enters invalid credentials, the system should display an error message and prevent access to the account
2	Reservation Functionality Test	Verify that the reservation functionality is working as expected	Attempt to make a parking reservation and verify that the system responds accordingly	The user must have a valid account and payment method registered with the system	1. If the user makes a valid reservation, the system should confirm the reservation and display the reservation details 2. If the user attempts to make an invalid reservation, the system should display an error message and prevent the reservation from being made
3	Payment Functionality Test	Verify that the payment functionality is working as expected	Attempt to make a payment for parking and verify that the system responds accordingly	The user must have a valid reservation and payment method registered with the system	1. If the user makes a valid payment, the system should confirm the payment and display the payment details 2. If the user attempts to make an invalid payment, the system should display an error message and prevent the payment from being made
4	Sensor Functionality Test	Verify that the sensor functionality is working as expected	Test the sensor system by simulating the occupancy and vacancy of parking spaces and verifying that the system responds accordingly	The sensor system must be connected and functional	1. If a parking space is occupied, the system should update the occupancy status of the space in real-time 2. If a parking space is vacated, the system should update the occupancy status of the space in real-time

5	Notification Functionality Test	Verify that the notification functionality is working as expected	Test the notification system by sending test notifications and verifying that the system responds accordingly	The user must have a registered mobile device with the system	1. If a parking reservation is made, the system should send a notification to the user's registered mobile device with reservation details 2. If a parking time is about to expire, the system should send a notification to the user's registered mobile device with a reminder to extend parking time or vacate the space 3. If a parking violation occurs, the system should send a notification to the user's registered mobile device with violation details and penalty assessments

14.5 Acceptance Testing

For the following processes, we will evaluate the experience of a random subset of users to ensure they are satisfied with the user interface and that the system meets their requirements regarding functionality and performance. These tests will need to be conducted a second time in the presence of outside users. Any common errors and concerns brought up by the end-users will need to be fixed in an update so that the users are able to understand the portal's functions and correct any user errors they may be making (understanding error messages).

Acceptance Test Scenarios:

Test Case #	Test Case Name	Test Case Description	Inputs	Expected Output
1	User-friendly Web Portal Access	Test whether the web portal is user-friendly and easily accessible to motorists.	The system is deployed and accessible to the testers. Steps: 1. Open the web portal. 2. Check if the layout and design are easy to understand. 3. Test if the links and buttons are working properly. 4. Verify if the information provided is accurate and up-to-date.	The web portal is user-friendly and accessible to all types of users.
2	Reservation	Test whether the	The system is deployed and	The reservation system works

	System Testing	reservation system works as expected.	accessible to the testers. Steps: 1. Create a new account or log in as an existing user. 2. Navigate to the reservation system. Search for available parking spots on-campus. 3. Make a reservation for a specific parking spot and date/time. 4. Verify if the reservation is confirmed and a confirmation email is sent.	seamlessly, and users can make reservations without any issues.
3	Payment Processing Testing	Test whether the payment processing system works accurately and securely.	The system is deployed and accessible to the testers, and users have valid payment methods linked to their accounts. Steps: 1. Create a new account or log in as an existing user. 2. Navigate to the payment processing system. 3. Add a new payment method or select an existing one. 4. Process a payment for parking or a violation fee. 5. Verify if the payment is processed accurately, and the payment information is encrypted and secure.	The payment processing system works accurately and securely, and users can process payments without any issues.
4	Notifications and Alerts Testing	Test whether the notifications and alerts system works as expected.	The system is deployed and accessible to the testers, and users have valid mobile devices registered to their accounts. Steps: 1. Create a new account or log in as an existing user. 2. Navigate to the notifications and alerts system. 3. Configure the notification settings for parking availability, expiration, and penalties. 4. Verify if notifications and alerts are sent to registered mobile devices accurately and in a timely manner.	The notifications and alerts system works seamlessly, and users receive timely and accurate notifications and alerts.
5	Analytics and Reporting	Test whether the analytics and	The system is deployed and accessible to the testers, and users	The analytics and reporting system works accurately and efficiently, and

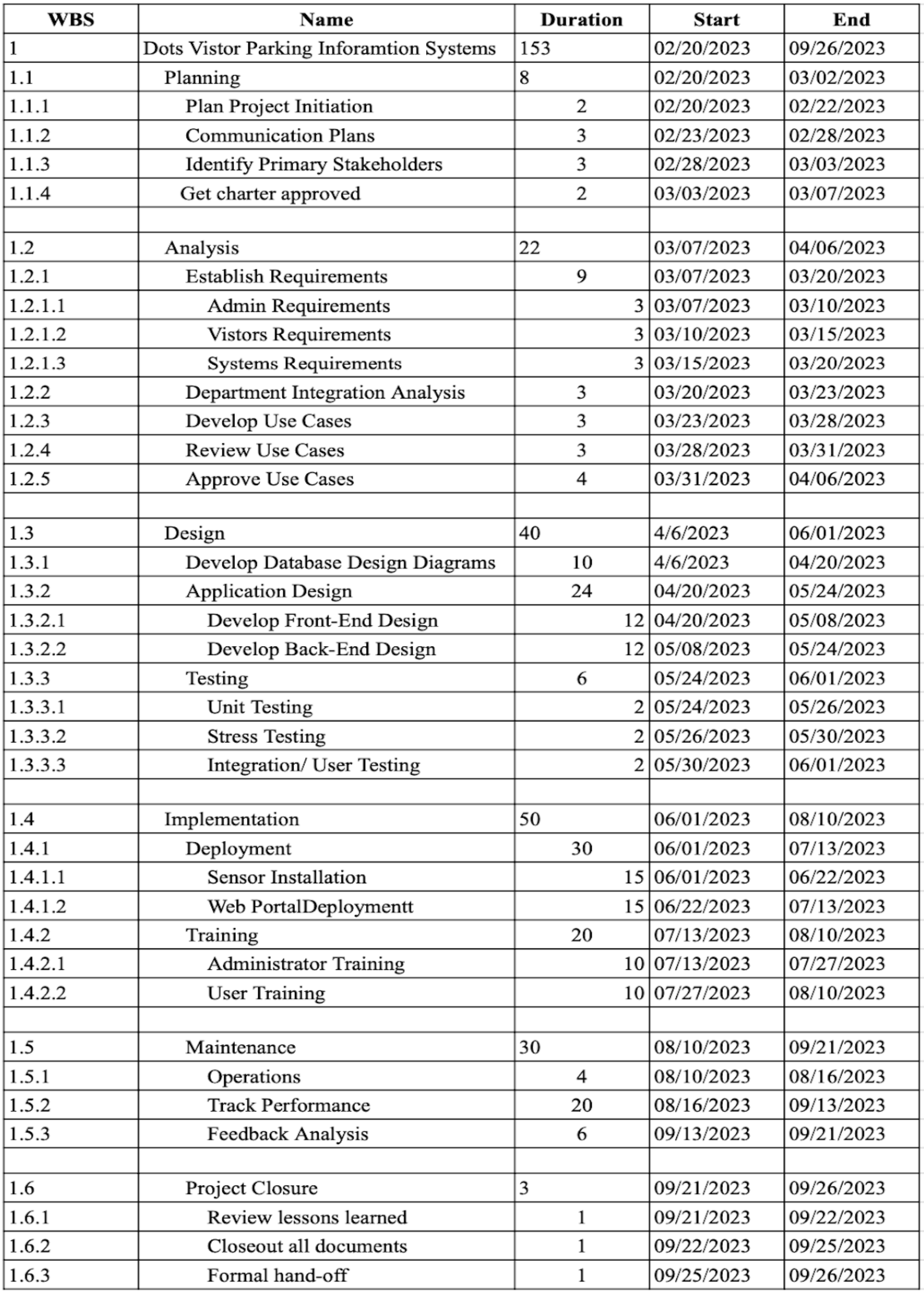
	Testing	reporting system works as expected.	have the necessary privileges to access the analytics and reporting system. Steps: Create a new account or log in as an existing user with analytics and reporting privileges. 1. Navigate to the analytics and reporting system. 2. Generate different types of reports on parking usage, occupancy, and trends. 3. Verify if the reports are accurate, comprehensive, and easily understandable.	users can generate and access comprehensive reports on parking usage and occupancy.

15. Implementation Plan

15.0 Objectives

Primary: DOTS Smart Parking system is successfully implemented and deployed.
Secondary Objectives: Secondary objective will be to ensure proper training is provided to the current users to switch to the new system, to provide maintenance support and finally review the project requirements via project closure.

15.1 Implementation Plan Schedule:



WBS	Name	Duration	Start	End
1	Dots Vistor Parking Inforamtion Systems	153	02/20/2023	09/26/2023
1.1	Planning	8	02/20/2023	03/02/2023
1.1.1	Plan Project Initiation	2	02/20/2023	02/22/2023
1.1.2	Communication Plans	3	02/23/2023	02/28/2023
1.1.3	Identify Primary Stakeholders	3	02/28/2023	03/03/2023
1.1.4	Get charter approved	2	03/03/2023	03/07/2023
1.2	Analysis	22	03/07/2023	04/06/2023
1.2.1	Establish Requirements	9	03/07/2023	03/20/2023
1.2.1.1	Admin Requirements	3	03/07/2023	03/10/2023
1.2.1.2	Vistors Requirements	3	03/10/2023	03/15/2023
1.2.1.3	Systems Requirements	3	03/15/2023	03/20/2023
1.2.2	Department Integration Analysis	3	03/20/2023	03/23/2023
1.2.3	Develop Use Cases	3	03/23/2023	03/28/2023
1.2.4	Review Use Cases	3	03/28/2023	03/31/2023
1.2.5	Approve Use Cases	4	03/31/2023	04/06/2023
1.3	Design	40	4/6/2023	06/01/2023
1.3.1	Develop Database Design Diagrams	10	4/6/2023	04/20/2023
1.3.2	Application Design	24	04/20/2023	05/24/2023
1.3.2.1	Develop Front-End Design	12	04/20/2023	05/08/2023
1.3.2.2	Develop Back-End Design	12	05/08/2023	05/24/2023
1.3.3	Testing	6	05/24/2023	06/01/2023
1.3.3.1	Unit Testing	2	05/24/2023	05/26/2023
1.3.3.2	Stress Testing	2	05/26/2023	05/30/2023
1.3.3.3	Integration/ User Testing	2	05/30/2023	06/01/2023
1.4	Implementation	50	06/01/2023	08/10/2023
1.4.1	Deployment	30	06/01/2023	07/13/2023
1.4.1.1	Sensor Installation	15	06/01/2023	06/22/2023
1.4.1.2	Web PortalDeploymentt	15	06/22/2023	07/13/2023
1.4.2	Training	20	07/13/2023	08/10/2023
1.4.2.1	Administrator Training	10	07/13/2023	07/27/2023
1.4.2.2	User Training	10	07/27/2023	08/10/2023
1.5	Maintenance	30	08/10/2023	09/21/2023
1.5.1	Operations	4	08/10/2023	08/16/2023
1.5.2	Track Performance	20	08/16/2023	09/13/2023
1.5.3	Feedback Analysis	6	09/13/2023	09/21/2023
1.6	Project Closure	3	09/21/2023	09/26/2023
1.6.1	Review lessons learned	1	09/21/2023	09/22/2023
1.6.2	Closeout all documents	1	09/22/2023	09/25/2023
1.6.3	Formal hand-off	1	09/25/2023	09/26/2023

16.0 Conclusion

Project background and opportunities are carefully investigated and it is concluded that the parking lot is crowded and inconvenient. A project objective about building a parking system is proposed, along with specific and feasible project scopes.Requirements from users, including motorists, facilities staff, IT department, analytics department and senior management, are identified to create a parking system aiming to relieve parking spot condition.

The system will enable motorists to check available parking spots and make reservations. Analysts and other administrators can review the report and analyze current or previous parking site information. Other systems such as the university billing system and shuttle database are connected to the system to provide further services and more information. Real-time notifications and alerts are also provided to inform motorists about essential parking information.

The team will employ both prototype and agile project techniques, which are suitable for user feedback and changing requirements. The project is estimated to have a total duration of 88 days, from planning to maintenance. Unit tests and integration tests will be conducted to ensure that all functions are implemented successfully. Stress tests will be conducted to ensure the system and servers can handle severe situations and huge data flow. System tests and acceptance tests will be conducted to ensure that users can use the system without errors and have a good experience. Security and redundancy are also enhanced by the implementation of encryption and redundant server.