ASHESI

ASHESI UNIVERSITY

WANDALUTE: AN EFFICIENT ONLINE APPLICATION FOR STAFF-TRANSPORT LOGISTICS MANAGEMENT

Applied Project

B.Sc. Computer Science

Nkosinathi Mzembe

2021

ASHESI UNIVERSITY

Wandalute: An efficient online application for staff-transport logistics management.

APPLIED PROJECT

Applied Project submitted to the Department of Computer Science, Ashesi University in partial fulfilment of the requirements for the award of Bachelor of Science degree in Computer Science

Nkosinathi Mzembe

April 2021

Declaration

I hereby declare that this applied project is the result of my own original work and that no part of it has been presented for another degree in this university or elsewhere.

Candidate's Signature: Mzembe

……………………………………………………………………………………………

Candidate's Name:

NKOSINATHI MZEMBE

……………………………………………………………………………………………

Date:

13 MAY 2021

……………………………………………………………………………………………

I hereby declare that preparation and presentation of this applied project were supervised in accordance with the guidelines on supervision of applied project laid down by Ashesi University.

Supervisor's Signature:

……………………………………………………………………………………………

Supervisor's Name: **DR DAVID EBO ADJEPON-YAMOAH**

……………………………………………………………………………………………

Date: **13TH MAY 2021**

……………………………………………………………………………………………

Acknowledgements

If not for God, I sincerely do not know how I would have pulled through. He divinely situated a support system of people who ensured that the set goals were achieved.

I want to express my sincere gratitude to my supervisor Dr David Ebo Adjepon-Yamoah for his invaluable guidance, comments, and suggestions. I am also sincerely grateful for how he motivated me, especially through this tiring pandemic period filled with despair.

Special thanks also go to Hope Bowa for being readily available to provide feedback and input during our check-in and testing sessions. Using an Agile Development approach, the project highly valued customer collaboration, of which Hope was fully involved throughout.

As an individual who is more inclined to frontend development, the online community of developers was very helpful in overcoming programming stumbling blocks. Besides online support, I should also appreciate all the people I reached out to regarding the best approach to adopt for this project, mainly my internship colleagues and classmates.

In addition, I would like to thank my family for believing in me, the Mastercard Foundation for funding my undergraduate studies, and Ashesi University for equipping me.

Abstract

Although we have significantly advanced in technology over the decades, there is still a reliance on conventional approaches to handling logistics. With a focus on the transportation of staff members out-of-office, this paper proposes an online application called '*Wandalute*' to ease the process a transportation logistics officer undergoes when assigning drivers to staff members who have requested office-owned transportation in fulfilling office duties. 'Wandalute' is a Chitumbuka (Malawi native) word which means '*They haven't left*. It is time for PLAN Malawi's staff members to request office transportation void of a cumbersome process that previously existed, which involved back and forth phone calls. This work intends to create a web-based application that would manage transportation requests made. The goal is also to address Sustainable Development Goals 8 and 9, which entail economic growth, and industry and infrastructure, respectively, through utilizing the company's transport resources.

Table of Contents

Declaration......iii

Acknowledgements......iv

Abstract......v

Chapter 1: Introduction......1

1.1 Background......1

1.2 Significance of the problem......1

1.3 Proposed solution......1

1.4 Related work......1

1.5 Data collection......2

1.6 The Wandalute web application......2

Chapter 2: Requirement Analysis......3

2.1 Requirements gathering......3

2.2 System scope......3

2.3 Users......3

2.4 Functional requirements......4

2.5 Non-functional requirements......4

2.6 Domain requirements......4

2.7 Use case diagram......5

2.8 System actors diagram......5

Chapter 3: Architectural Design......7

3.1 Model-View-Controller (MVC)......7

3.2 Database design......7

3.2.1 Entity Relationship Diagram (ERD)......8

3.3 Layered architecture......9

3.4 Client-server architecture..10

Chapter 4: Implementation...11

4.1 Technologies used in development..11

4.2 Implementation technique..11

4.3 System components..13

4.5 Business rules...17

Chapter 5: Testing and Results..19

5.1 Development testing..19

5.1.1 Unit testing..19

5.1.2 Component testing...21

5.1.3 System testing..21

5.2 User testing...22

5.2.1 Acceptance testing..22

5.3 Other testing..23

5.4 Test results analysis...24

Chapter 6: Conclusion...25

6.1 Challenges faced...25

6.2 Recommendations and future work...25

6.3 Conclusion...25

References...27

Appendix A..28

Chapter 1: Introduction

1.1 Background

PLAN Malawi has had an issue with the management of its transport resources. When Hope was employed as the company's Logistics Officer, he realized that although his primary role was procurement, he was also in charge of handling staff transportation. A cumbersome approach was adopted, and that involved back and forth phone calls. He created a simple Excel file to track the various bookings staff members would make in curbing this. Although this new approach helped keep records and prevent the assignment of multiple staff members to the same driver, it still has not stopped the back and forth phone calls. Hence the need for a timely system that would enable the staff members to book transportation at their own convenience.

1.2 Significance of the problem

Without real-time assignment and notification, there is the wastage of the organization's resources as certain procedures are redundant and full of collisions not easily identified prior. Aside from this, the back and forth calls require credit, hence an extra cost to the company.

1.3 Proposed solution

This work intends to create a web application that would manage transportation and the assignment of persons (i.e., staff and drivers). In addition, the project addresses Sustainable Development Goals (SDG) 8 and 9. SDG 8, Promote sustained, inclusive and sustainable economic growth, full and productive employment and decent work for all[3]. SDG 9, Build resilient infrastructure, promote inclusive and sustainable industrialization and foster innovation[4]. This would, in turn, help enhance and utilize the company's transport resources.

1.4 Related work

Development of Internet-Based Applications for Fleet Management and Logistics. This application utilized an effectively developed Content Storage Management [1]. Further, a flexible platform for intermodal transportation and integrated logistics; A real-time ICT platform facilitating logistics incorporating cloud-based services in its logistics ecosystem [2]. The work found out that neither of these systems deals with the transportation of staff members. Although several other transportation systems exist, most dwell on fleet management, which is a public resource and often for commercial purposes. In this case, the work is considering a tailored solution. This is so that as an organization, its internal vehicle can be easily managed.

1.5 Data collection

Through researching secondary sources of literature and other studies would inform the requirements needed in bridging gaps identified. Secondly, conducting an interview – with the end-user to collect data that would inform the needs of the systems and how it ought to function in achieving their organizational goal.

1.6 The Wandalute web application

An online system that allows the booking of office-owned transportation by staff members. Wandalute is a web-based application that would manage the process in which Office Drivers are booked and re-assigned to Staff Members that have requested to use office-owned transport.

Chapter 2: Requirement Analysis

This chapter of the project focuses on the means by which the requirements were gathered and how the findings have constituted towards the functional and non-functional requirements of the system. The use case diagram further portrays the expected interaction.

2.1 Requirements gathering

Although having briefly spoken with the user of the system over the summer vacation of 2020. There was the need to gather an in-depth understanding of what precisely he needed, hence a survey. A phone-call-interview session was conducted upon supervisor approval of the question format. Hope Bowa, the Logistics Officer at PLAN, through this interview, provided responses that genuinely reflect the kind of system he needs in his field of work. (*The interview session was also recorded for future reference*).

The purpose of the interview was to gather an in-depth understanding of PLAN Malawi's major problem concerning the management of its transportation resources. The benefit is that PLAN Malawi would be delivered a system based on the requirements specified.

2.2 System scope

Wandalute will mainly focus on allowing office staff members to book transportation based on driver availability. They can further edit their profiles if they wish to. As for the drivers, they indicate their availability based on days and time of day of which they can attend to another staff member. Lastly, the Administrator (the logistics officer in this case) manages the whole system and can re-assign staff members to drivers in some instances where need be.

2.3 Users

The Wandalute system would have three typical users: (1) Logistics Officer (Administrator), (2) Office Drivers, and (3) Staff Members. These three users would have different interfaces based on their roles and tasks to be achieved.

2.4 Functional requirements

FR01: The booking component of the system should enable staff members to book a driver for a particular day and time.

FR02: Drivers should be assigned to the staff member once the booking is successful.

FR03: Staff members who wish to drive an office vehicle should be authorized to do so.

FR04: The notification component of the system should provide pop-up alerts on the application.

FR05: The system should have a calendar component that enables the selection of dates.

2.5 Non-functional requirements

NFR01: Reliability – The system should display drivers currently available.

NFR02: Availability – The system should be available at any time for use.

NFR03: Security – Only the Logistics Officer should have access to the admin dashboard.

NFR04: Data Integrity – Registered user profiles should not be tampered with.

NFR05: Usability – The application should have an appealing user interface.

2.6 Domain requirements

Wandalute should support all web browsers for both mobile devices and desktops.

2.7 Use case diagram

The use case diagram for Wandalute summarizes the application's functional requirements as shown below, giving an overview of the system users and some of the functionalities they can perform within the application.

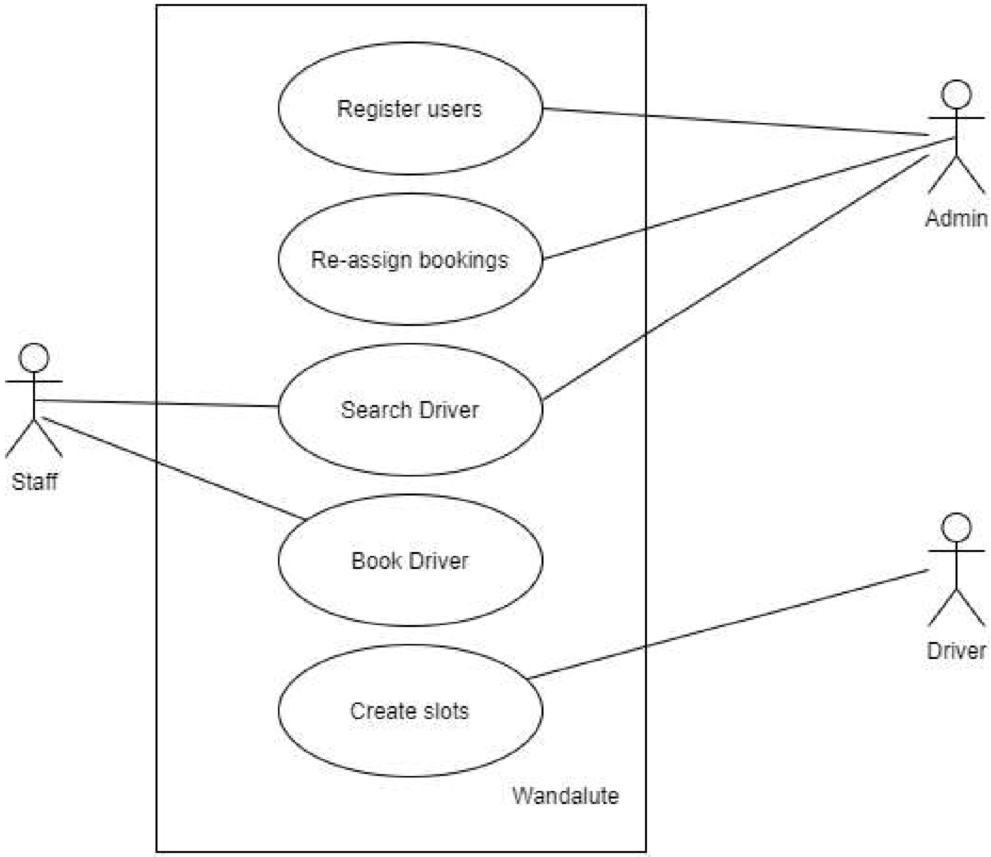


Figure 2.1: The use-case diagram

2.8 System actors diagram

A light-weight overview of how the users would interact with the system. See Figure 2.1 below.

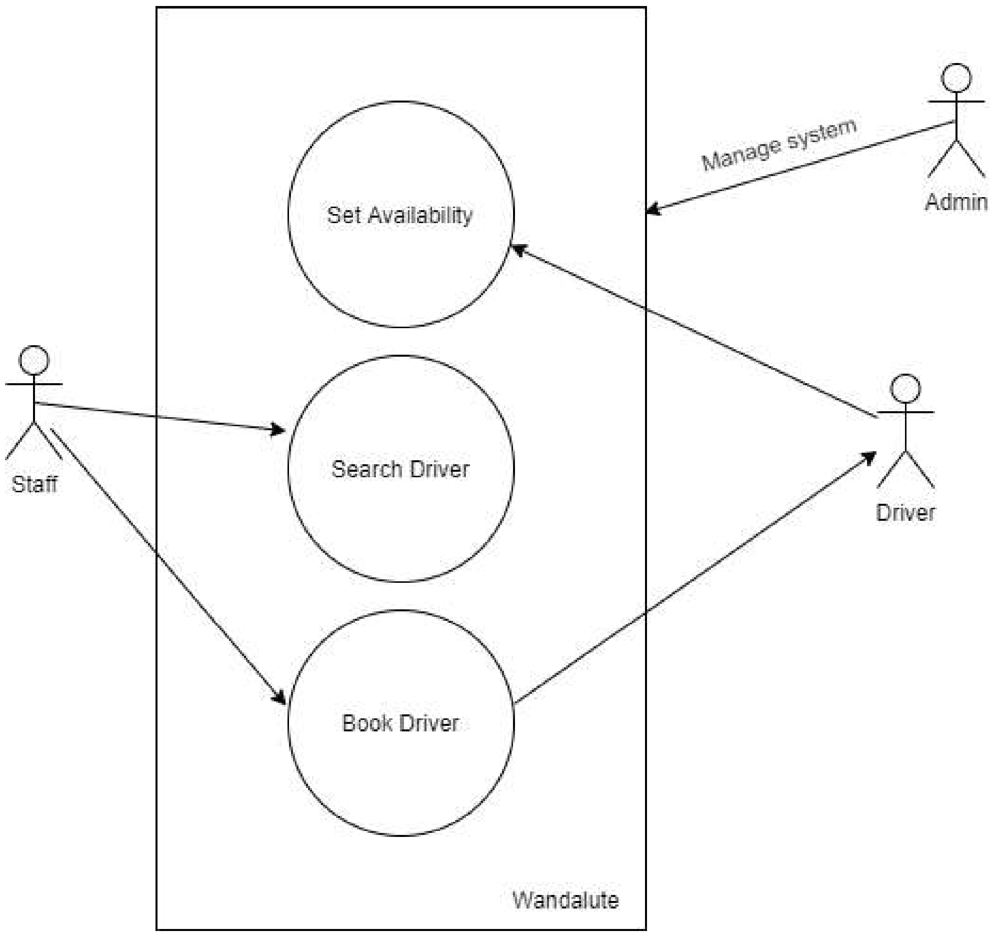


Figure 2.2: The system actors' diagram

Chapter 3: Architectural Design

The Logistics Management System is an interactive online application. Hence, its architecture can be best visualized using the following software architectural designs, which illustrate the control and communications amongst the various components of the system.

3.1 Model-View-Controller (MVC)

The work will use the Model-View-Controller architectural pattern to serve as a presentation layer in facilitating understanding of the system. In addition, the work makes use of a relational database in managing Wandalute's data. This architectural pattern divides the system into three distinct components: Refer to Figure 3.1 below.

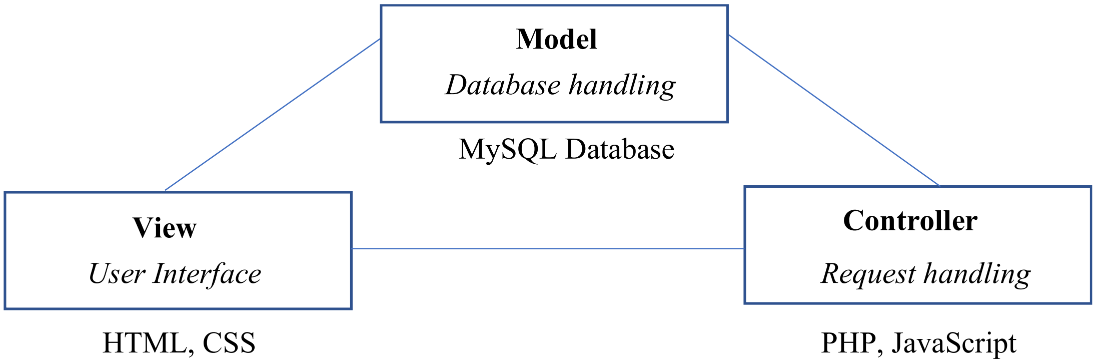


Figure 3.1: Model-View-Controller Diagram

3.2 Database design

Wandalute's database consists of nine (9) tables, 3 of which were included from the Laravel package. The table (Table 3.1) below shows the tables used by Wandalute and a description of each table.

Table 3.1 Database tables used by Wandalute

Table	Description
users	This table contains details of the different users of the system to grant access to the system as well as for identification.
password_resets	This table stores the token and email of the user who has requested to reset their password.
migrations	This table contains a record of all the tables created as development occurred.
failed_jobs	This table contains a record of all failed system tasks to be handled by the framework.
roles	This table stores the role ids for the ‘Admin’, ‘Driver’, and ‘Staff’.
schedules	This table contains the different schedules drivers have created for people to book.
times	This table stores the different time slots drivers have created as well as the status of those time slots.
Bookings	This table stores all booking records with the persons involved and the status to determine whether the event has happened.
Vehicles	This table stores data for the different vehicles available on-site.

3.2.1 Entity Relationship Diagram (ERD)

Wandalute’s database makes use of relationships. Each user has a role (either ‘Admin’, ‘Driver’ or ‘Staff’). Office drivers are assigned to internal vehicles. A driver creates an availability schedule, and that schedule is attached to a ‘times’ table that stores the various time slots. Once a booking has been made, it takes into account data stored in the other database tables. The Entity Relationship Diagram for the tables created is shown in figure 3.2 below.

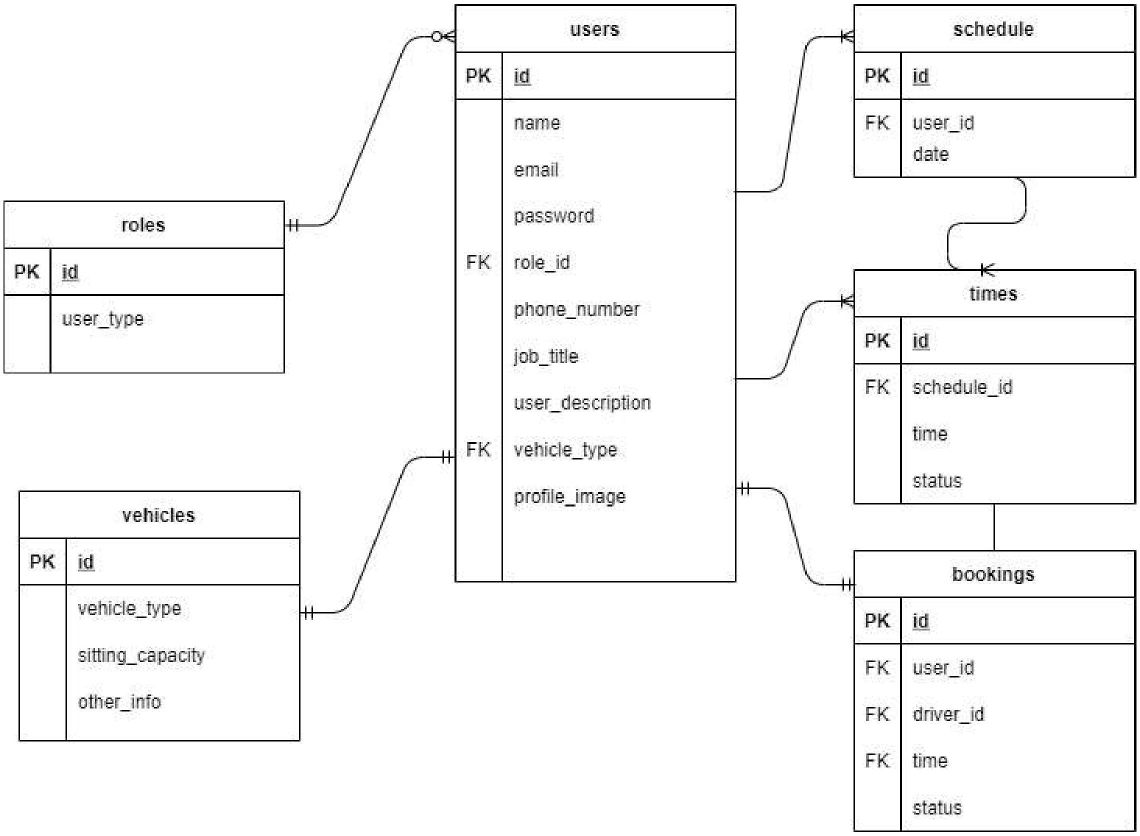


Figure 3.2: Entity Relationship Diagram for Wandalute

3.3 Layered architecture

The user interface layer would be independent of the other layers for easy modification of individual UI components that would foster a more appealing interaction by the user with the system. The authentication and authorization layer for login in security would be independent too. The component of the system that handles the re-assignment of drivers would be separate from the other layers as well. Lastly, the database to store all this data in real-time would also be catered for in another layer. The layers are all based on specific roles for easy modification. Refer to Figure 3.3 below.

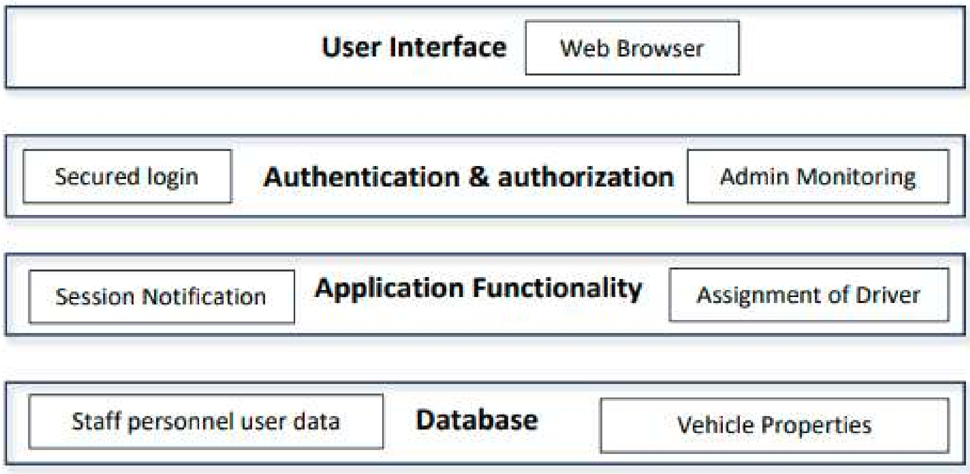


Figure 3.3: Layered Application Diagram for Wandalute

3.4 Client-server architecture

The application is online, having its services requested by various clients who may access the application from either their handheld mobile devices, laptop or even desktop, provided there is an internet connection to communicate with the server as one makes a request to the server. The Client-Server Architecture shows how services are requested between the two parties and the kind of services. Refer to Figure 3.4 below for an illustration of the client-server architecture.

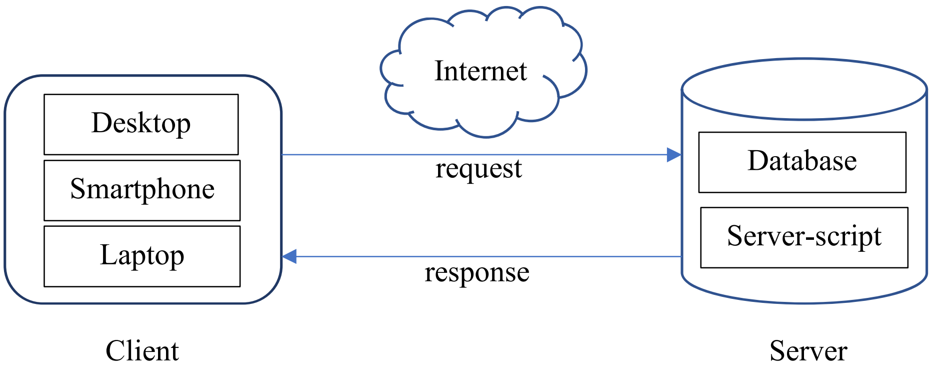


Figure 3.4: Client-Server Architecture

Chapter 4: Implementation

In developing Wandalute, a considerable number of decisions were made to deliver a system that works according to the requirements specified by the user.

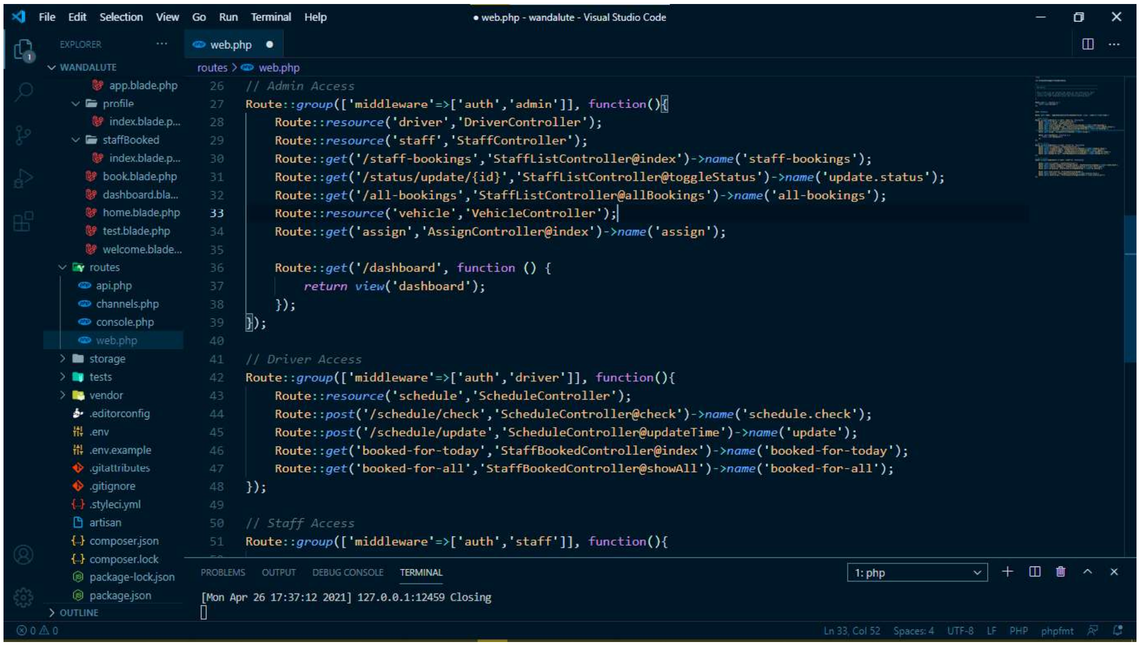
4.1 Technologies used in the development

This application was built using web technologies such as HTML to render it as a web application. PHP for the server-side scripting, leveraging on the PHP Laravel Framework (Laravel 8). The database management system used is MySQL. Additionally, the Bootstrap framework for elegant CSS rendering as well as mobile-responsiveness support. TailwindCSS is another tool used as it renders more customizable options. VueJS was also used for part of the front end. Finally, an AJAX function is implemented on returning drivers based on the date feature for booking. See section 4.3 for screenshots of the system.

4.2 Implementation technique

Using the MVC model offered by PHP Laravel, the development took an incremental approach getting component by component to work. The Model created objects such as the users, roles, times, bookings as well as schedules. The View is for the different pages that are being presented to the user. The Controller helped these two components communicate by also rendering defined functionalities that executed tasks such as which pages to return or actions to take when the user has taken particular measures. The work was carried in stages: firstly, creating the database and necessary tables with respective relationships(if any). Users and roles were then created in which users have different page authorization access. This was accomplished through authentication and route access permissions. Only the 'Admin' creates users for the system and views overall schedules and bookings made by everyone else. The 'Admin' is also capable of re-assigning drivers to staff members (if need be).

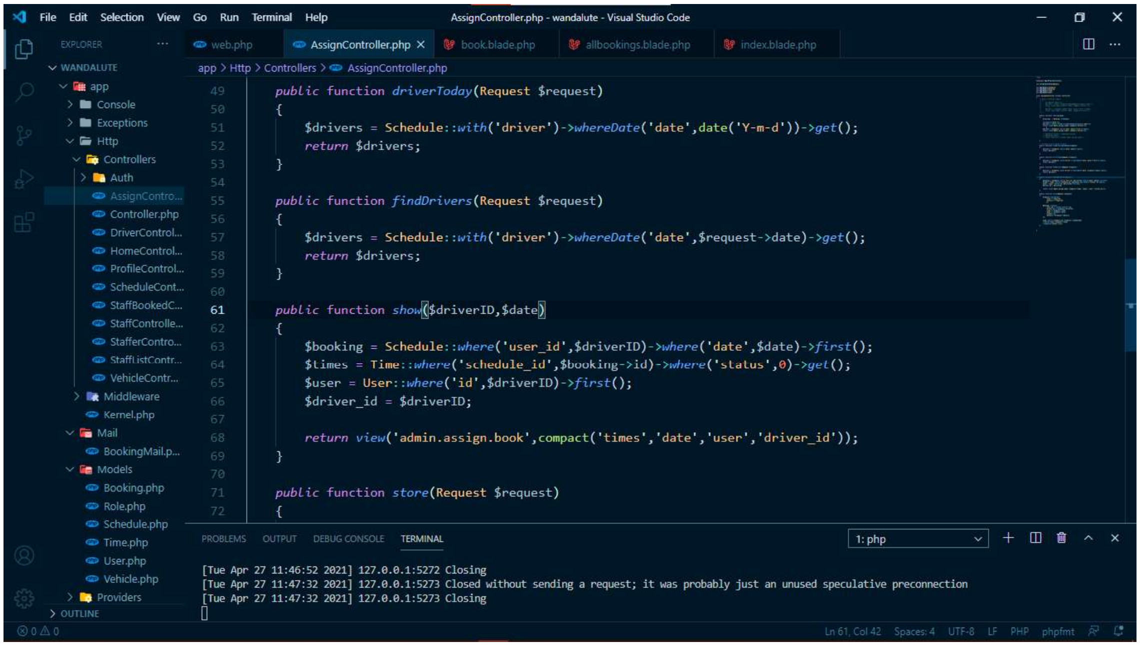
On the other hand, drivers declare their availability (time of day and on which dates). Lastly, the staff members can book drivers to take them to their desired destinations to fulfil office-related duties. Overall, the admin can monitor all these actions and make changes too. See below some code snippets.



```php
// Admin Access
Route::group(['middleware'=>['auth','admin']], function(){
    Route::resource('driver','DriverController');
    Route::resource('staff','StaffController');
    Route::get('/staff-bookings','StaffListController@index')->name('staff-bookings');
    Route::get('/status/update/{id}','StaffListController@toggleStatus')->name('update.status');
    Route::get('/all-bookings','StaffListController@allBookings')->name('all-bookings');
    Route::resource('vehicle','VehicleController');
    Route::get('assign','AssignController@index')->name('assign');

    Route::get('/dashboard', function () {
        return view('dashboard');
    });
});

// Driver Access
Route::group(['middleware'=>['auth','driver']], function(){
    Route::resource('schedule','ScheduleController');
    Route::post('/schedule/check','ScheduleController@check')->name('schedule.check');
    Route::post('/schedule/update','ScheduleController@updateTime')->name('update');
    Route::get('booked-for-today','StaffBookedController@index')->name('booked-for-today');
    Route::get('booked-for-all','StaffBookedController@showAll')->name('booked-for-all');
});

// Staff Access
Route::group(['middleware'=>['auth','staff']], function(){
```

```
[Mon Apr 26 17:37:12 2021] 127.0.0.1:12459 Closing
```

Figure 4.1: Code snippet for middleware authentication to access web routes


AssignController.php - wandalute - Visual Studio Code
app > Http > Controllers > AssignController.php

```php
    public function driverToday(Request $request)
    {
        $drivers = Schedule::with('driver')->whereDate('date',date('Y-m-d'))->get();
        return $drivers;
    }

    public function findDrivers(Request $request)
    {
        $drivers = Schedule::with('driver')->whereDate('date',$request->date)->get();
        return $drivers;
    }

    public function show($driverID,$date)
    {
        $booking = Schedule::where('user_id',$driverID)->where('date',$date)->first();
        $times = Time::where('schedule_id',$booking->id)->where('status',0)->get();
        $user = User::where('id',$driverID)->first();
        $driver_id = $driverID;

        return view('admin.assign.book',compact('times','date','user','driver_id'));
    }

    public function store(Request $request)
    {
```

```
[Tue Apr 27 11:46:52 2021] 127.0.0.1:5272 Closing
[Tue Apr 27 11:47:32 2021] 127.0.0.1:5273 Closed without sending a request; it was probably just an unused speculative preconnection
[Tue Apr 27 11:47:32 2021] 127.0.0.1:5273 Closing
```

Figure 4.2: Code snippet for driver re-assignment

4.3 System components

Wandalute is a responsive web-based application and hence renders well on both mobile and desktop browsers.

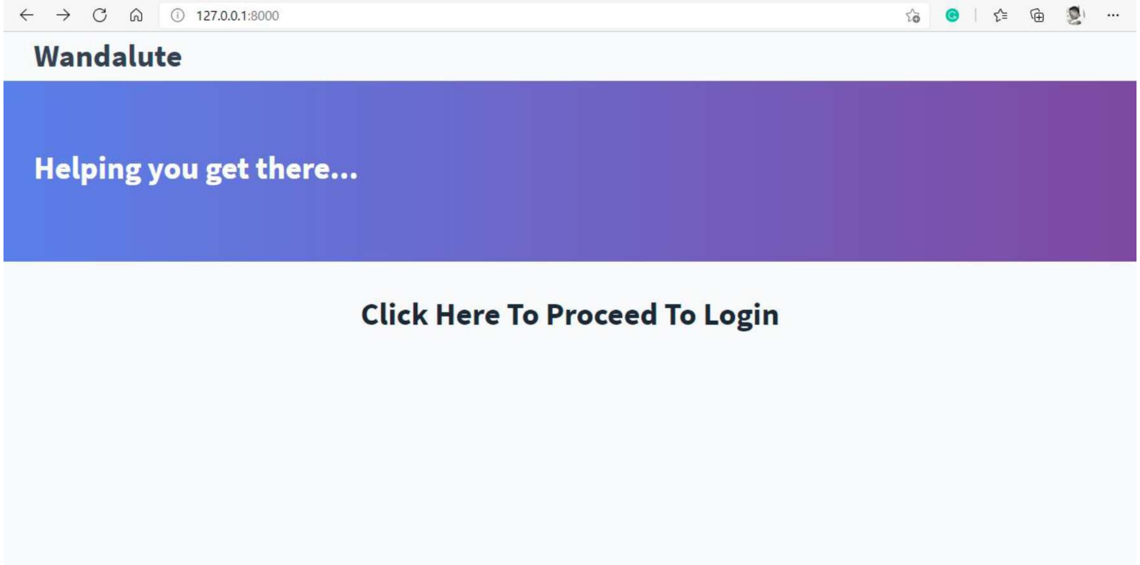


Figure 4.3: Screenshot of Wandalute landing page

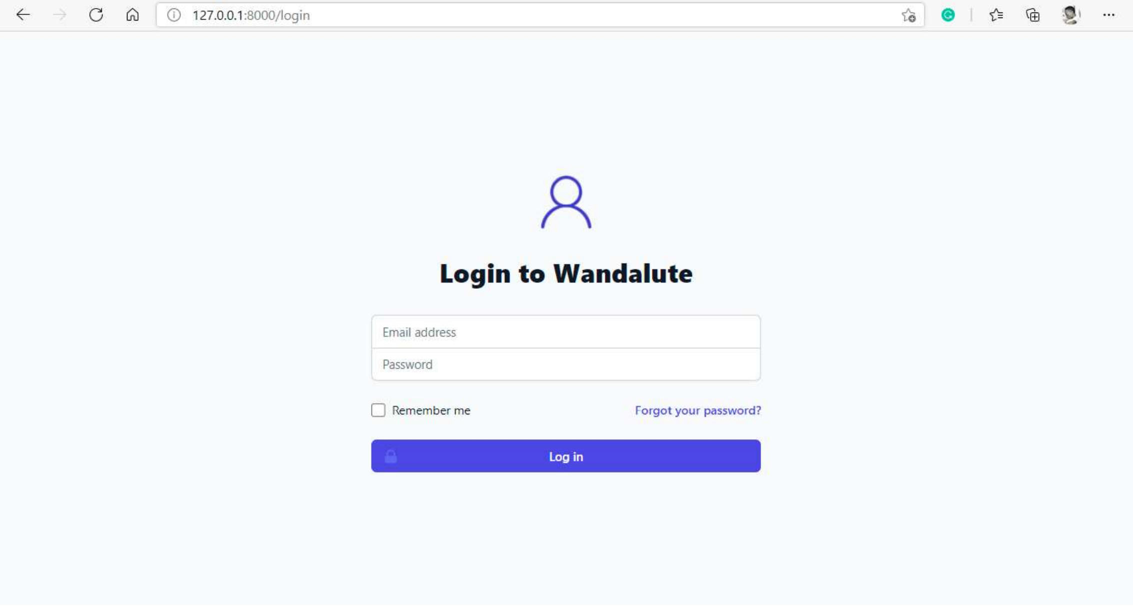
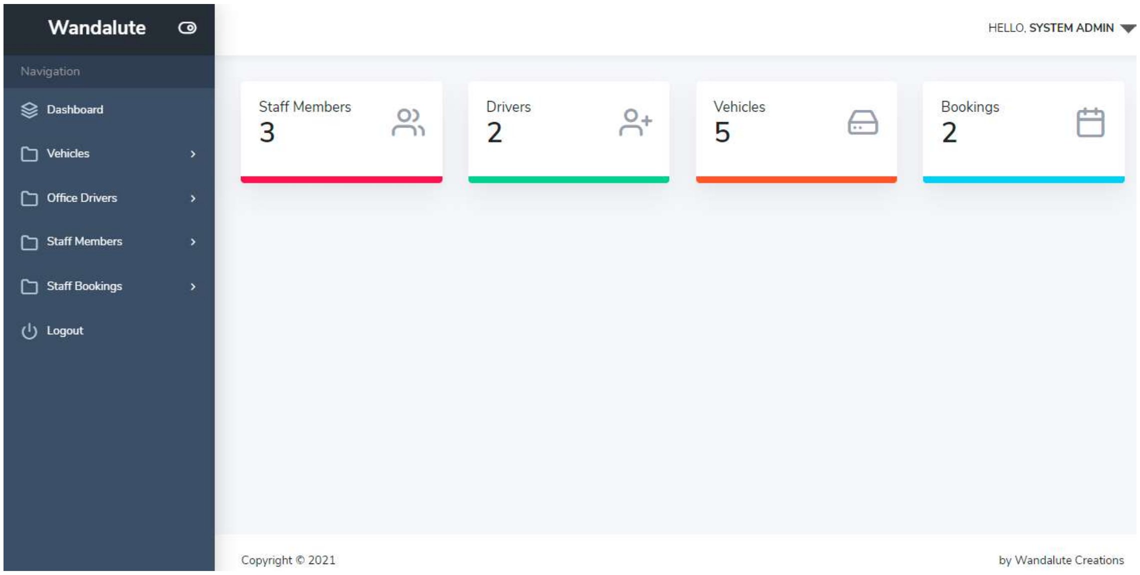


Figure 4.4: Screenshot of Wandalute login page



Staff Members 3

Drivers 2

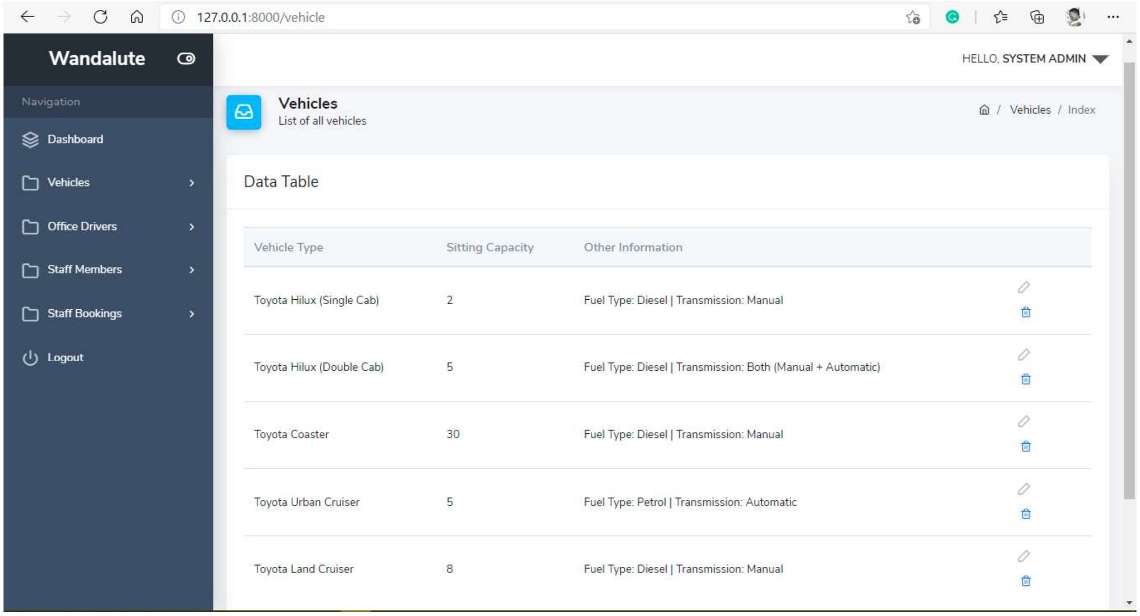
Vehicles 5

Bookings 2

Copyright © 2021

by Wandalute Creations

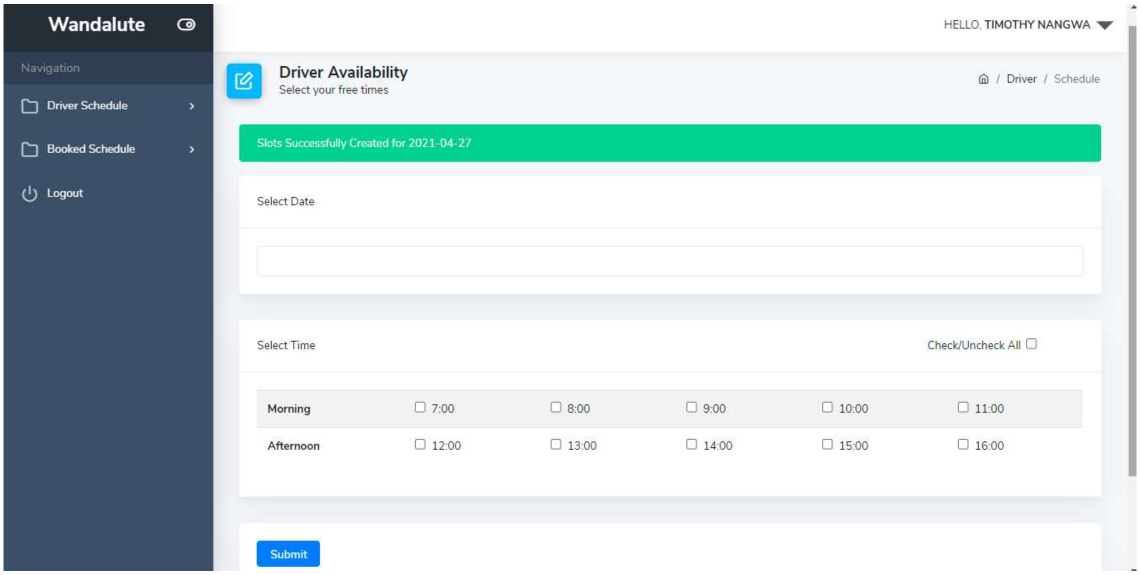
Figure 4.5: Screenshot of Wandalute Admin Dashboard



Data Table

Vehicle Type	Sitting Capacity	Other Information
Toyota Hilux (Single Cab)	2	Fuel Type: Diesel | Transmission: Manual
Toyota Hilux (Double Cab)	5	Fuel Type: Diesel | Transmission: Both (Manual + Automatic)
Toyota Coaster	30	Fuel Type: Diesel | Transmission: Manual
Toyota Urban Cruiser	5	Fuel Type: Petrol | Transmission: Automatic
Toyota Land Cruiser	8	Fuel Type: Diesel | Transmission: Manual

Figure 4.6: Screenshot of current office vehicles



Driver Availability

Select your free times

Slots Successfully Created for 2021-04-27

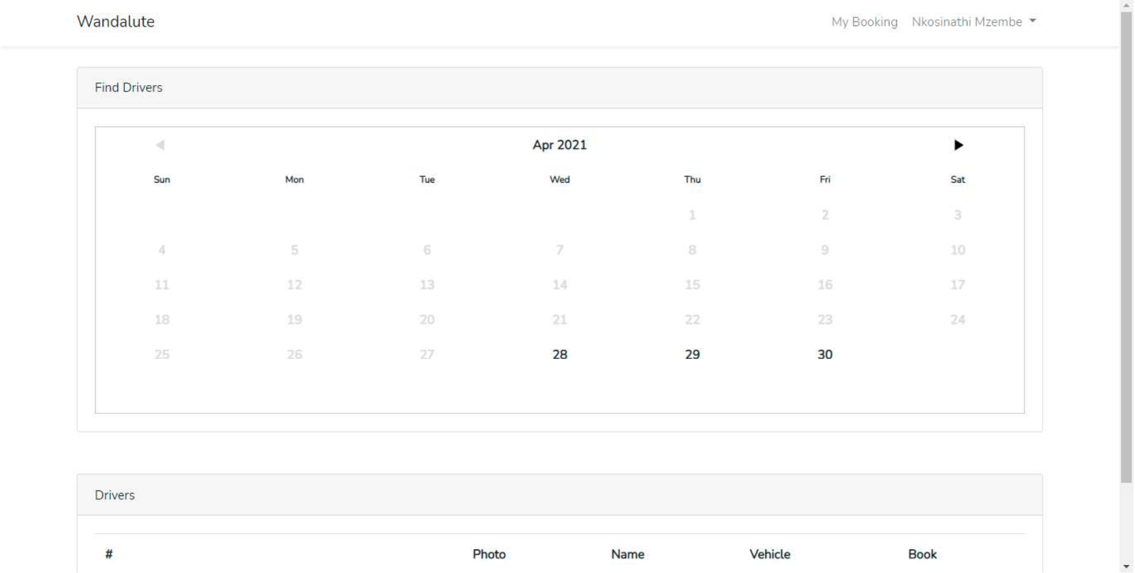
Select Date

Select Time

Check/Uncheck All □

Morning	□ 7:00	□ 8:00	□ 9:00	□ 10:00	□ 11:00
Afternoon	□ 12:00	□ 13:00	□ 14:00	□ 15:00	□ 16:00

Figure 4.7: Screenshot of the web page where driver indicates their availability



Find Drivers

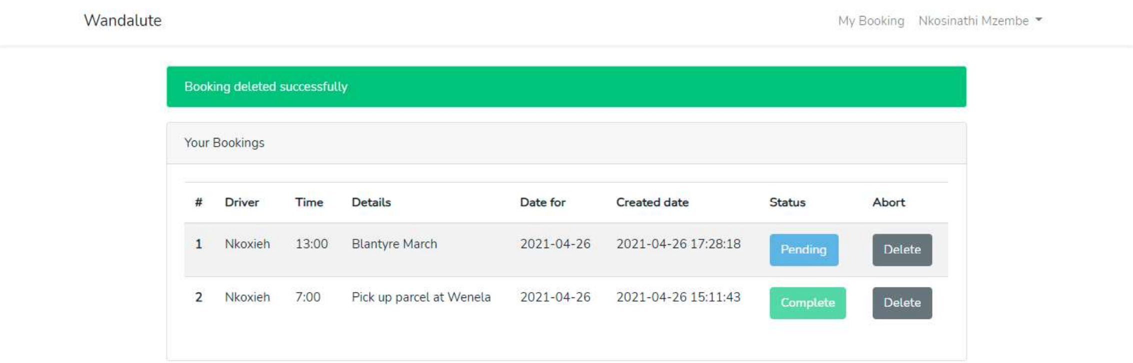
Apr 2021

Sun	Mon	Tue	Wed	Thu	Fri	Sat
				1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	

Drivers

#	Photo	Name	Vehicle	Book

Figure 4.8: Screenshot web page where staff finds a driver based on the date selected

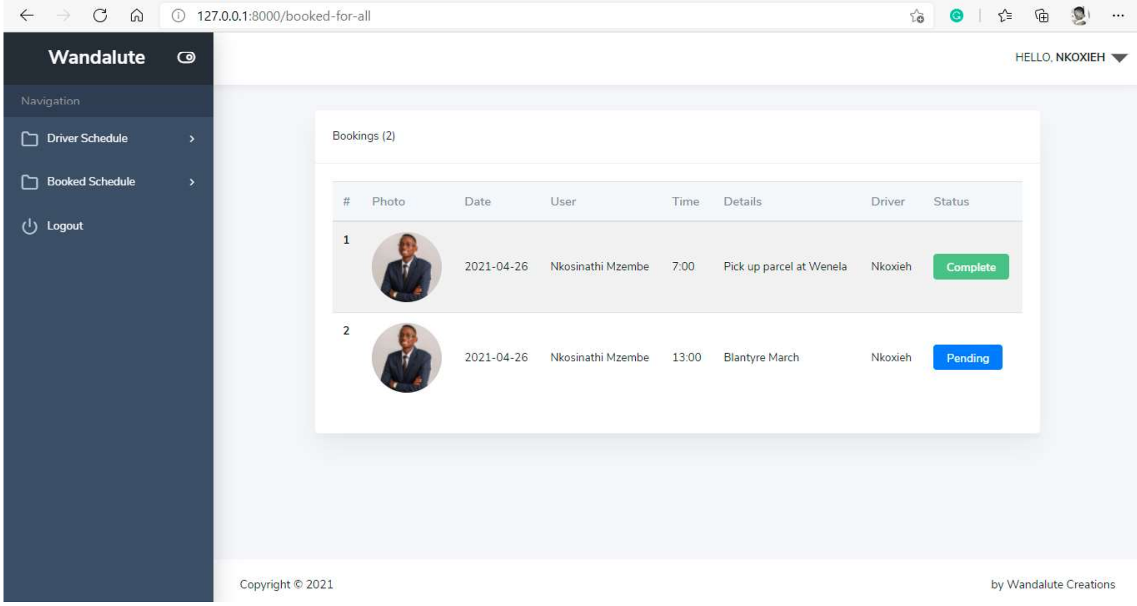


Booking deleted successfully

Your Bookings

#	Driver	Time	Details	Date for	Created date	Status	Abort
1	Nkoxieh	13:00	Blantyre March	2021-04-26	2021-04-26 17:28:18	Pending	Delete
2	Nkoxieh	7:00	Pick up parcel at Wenela	2021-04-26	2021-04-26 15:11:43	Complete	Delete

Figure 4.9: Screenshot web page where a staff member can view their bookings.



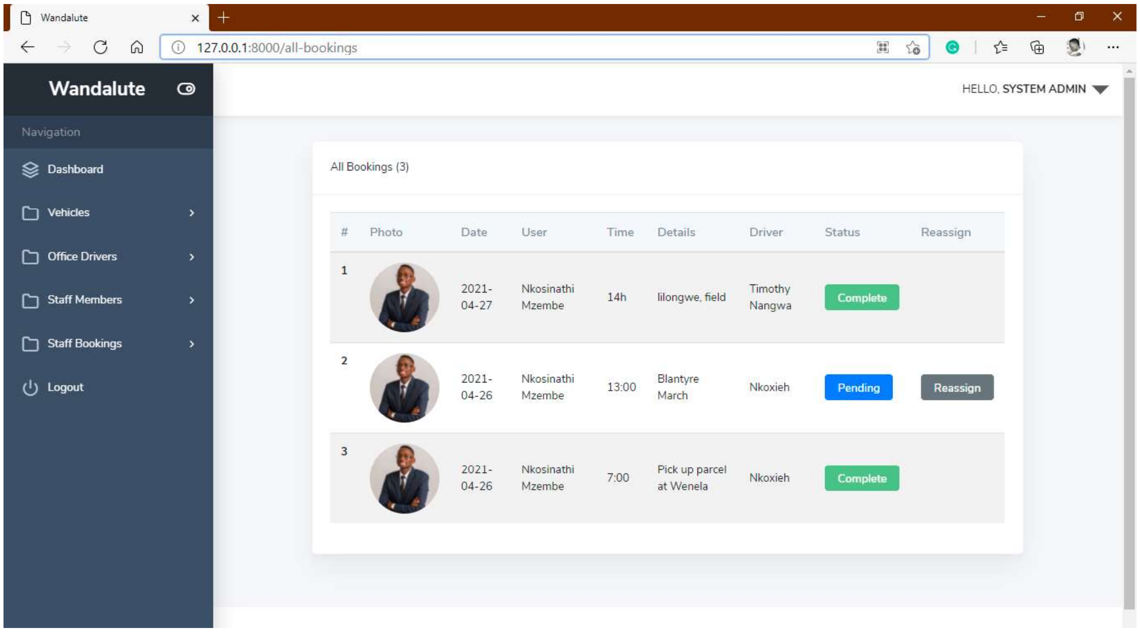
Bookings (2)

#	Photo	Date	User	Time	Details	Driver	Status
1		2021-04-26	Nkosinathi Mzembe	7:00	Pick up parcel at Wenela	Nkoxieh	Complete
2		2021-04-26	Nkosinathi Mzembe	13:00	Blantyre March	Nkoxieh	Pending

Copyright © 2021

by Wandalute Creations

Figure 4.10: Screenshot web page where the driver can change the status of the trip



All Bookings (3)

#	Photo	Date	User	Time	Details	Driver	Status	Reassign
1		2021-04-27	Nkosinathi Mzembe	14h	lilongwe, field	Timothy Nangwa	Complete	
2		2021-04-26	Nkosinathi Mzembe	13:00	Blantyre March	Nkoxieh	Pending	Reassign
3		2021-04-26	Nkosinathi Mzembe	7:00	Pick up parcel at Wenela	Nkoxieh	Complete	

Figure 4.11: Screenshot web page where the admin can re-assign a staff member (if need be)

4.4 Business rules of the system

There is controlled access achieved through middleware routing; hence all web pages cannot be accessed unless one has authorization. The staff member makes a booking via the integrated booking system feature in requesting a driver. The system has three main dashboards, the Admin Dashboard, Driver Dashboard, Staff Dashboard. Only the Administrator has the privilege of registering new users of the system. This was inspired by the way Ashesi registers the community by creating default accounts for everyone.

The system uses a 'First-Come-First-Serve' analogy such that the person who books first gets assigned the driver based on the time and day they wish to, and this serves the essence of encouraging people to book earlier.

The work gathered that incorporating prioritization would discourage people from booking transportation since others (lower ranked staff) would believe that their requests would never be granted due to the bias/discrimination in the system.

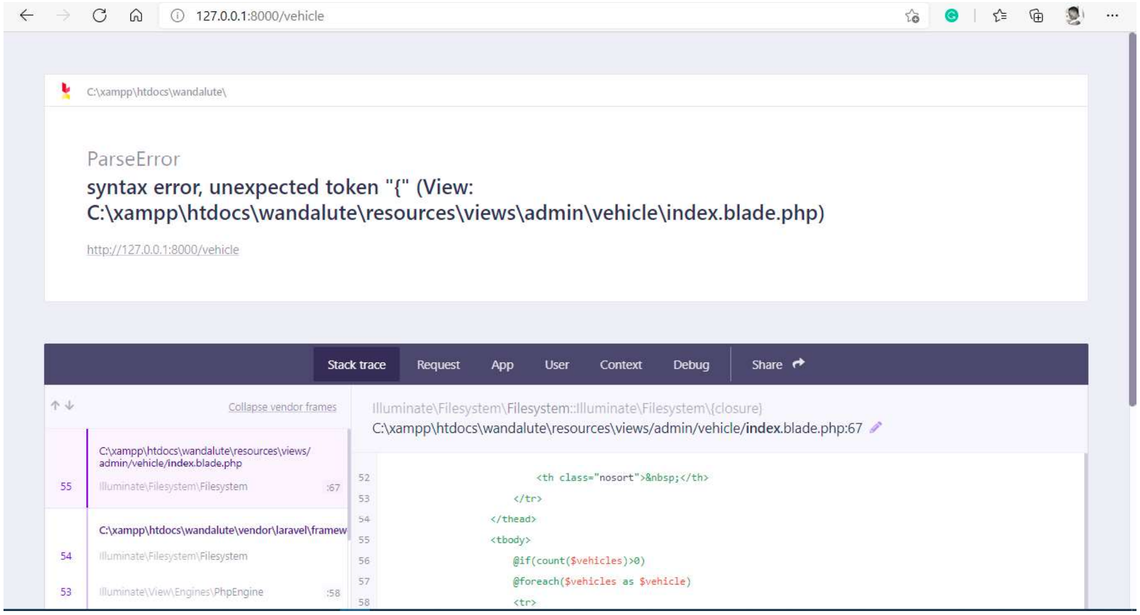
Lastly, every user of the system is unique, and this is with respect to both user id and email.

Chapter 5: Testing and Results

The work underwent a series of tests in validating whether the built product meets the requirements gathered during the early phases of the project. The project did not consider just one general testing approach because by conducting low levels of testing, the system is tested in isolation while considering its functionality and error-handling capabilities[5]. With respect to 'user testing', it took place on the video conferencing tool, Zoom, whereby the participant was granted remote access to control the screen since the application has not been deployed to a live server yet.

5.1 Development testing

As the project was being built, it was tested on every run for errors. PHP helped in fishing out some of those errors. See Figure 5.1 below.



C:\xampp\htdocs\wandalute\

ParseError
syntax error, unexpected token "{" (View: C:\xampp\htdocs\wandalute\resources\views\admin\vehicle\index.blade.php)

http://127.0.0.1:8000/vehicle

55 C:\xampp\htdocs\wandalute\resources\views/admin/vehicle/index.blade.php
Illuminate\Filesystem\Filesystem :67

54 C:\xampp\htdocs\wandalute\vendor\laravel\framew
Illuminate\Filesystem\Filesystem

53 Illuminate\View\Engines\PhpEngine :58

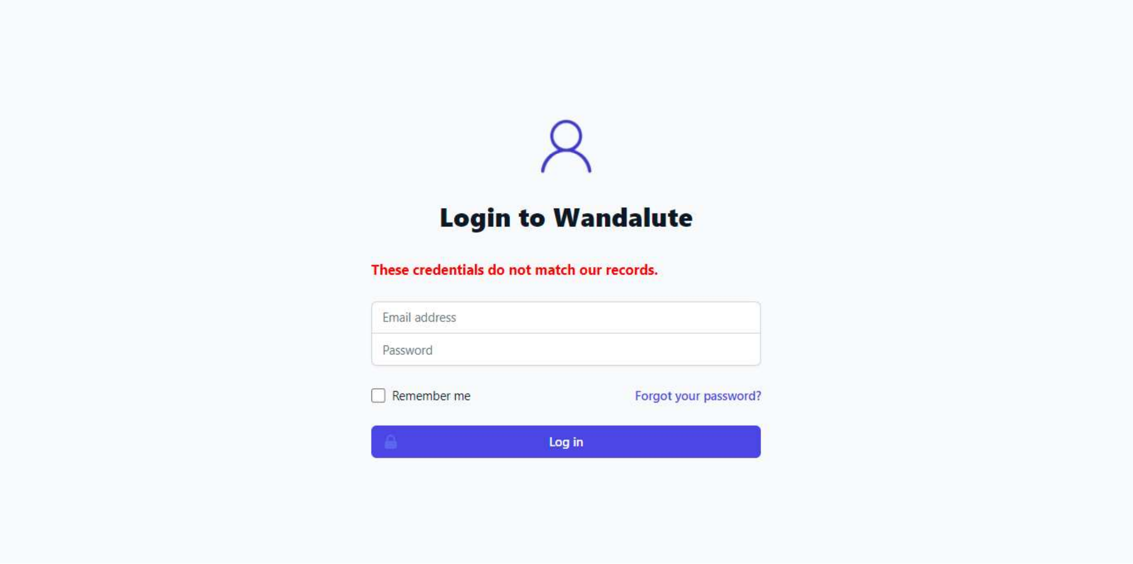
Illuminate\Filesystem\Filesystem::Illuminate\Filesystem\{closure}
C:\xampp\htdocs\wandalute\resources\views/admin/vehicle/index.blade.php:67

```
                    <th class="nosort">&nbsp;</th>
                </tr>
            </thead>
            <tbody>
                @if(count($vehicles)>0)
                @foreach($vehicles as $vehicle)
                <tr>
```

Figure 5.1: Screenshot of a syntax error caught

5.1.1 Unit testing

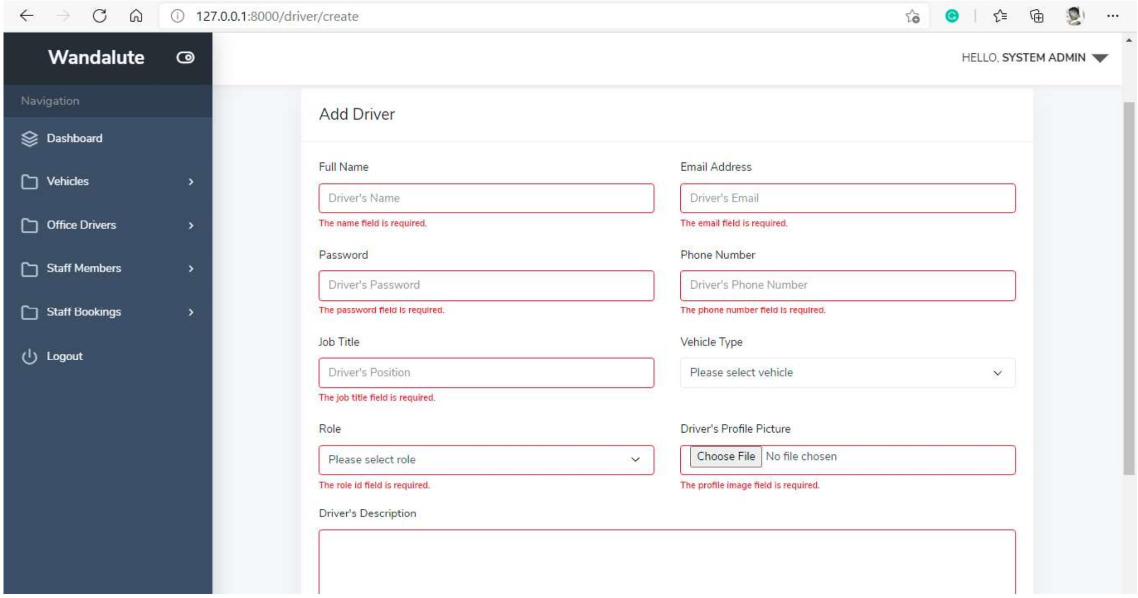
Independent units of the system were tested to check whether they work as they ought to. See figures below.



Login to Wandalute

These credentials do not match our records.

Figure 5.2: Screenshot of alert when invalid login credentials have been provided



127.0.0.1:8000/driver/create

Add Driver

Full Name: Driver's Name
The name field is required.

Email Address: Driver's Email
The email field is required.

Password: Driver's Password
The password field is required.

Phone Number: Driver's Phone Number
The phone number field is required.

Job Title: Driver's Position
The job title field is required.

Vehicle Type: Please select vehicle

Role: Please select role
The role id field is required.

Driver's Profile Picture: Choose File No file chosen
The profile image field is required.

Driver's Description

Figure 5.3: Screenshot of alert when Admin attempts to create a driver with incomplete data

5.1.2 Component testing

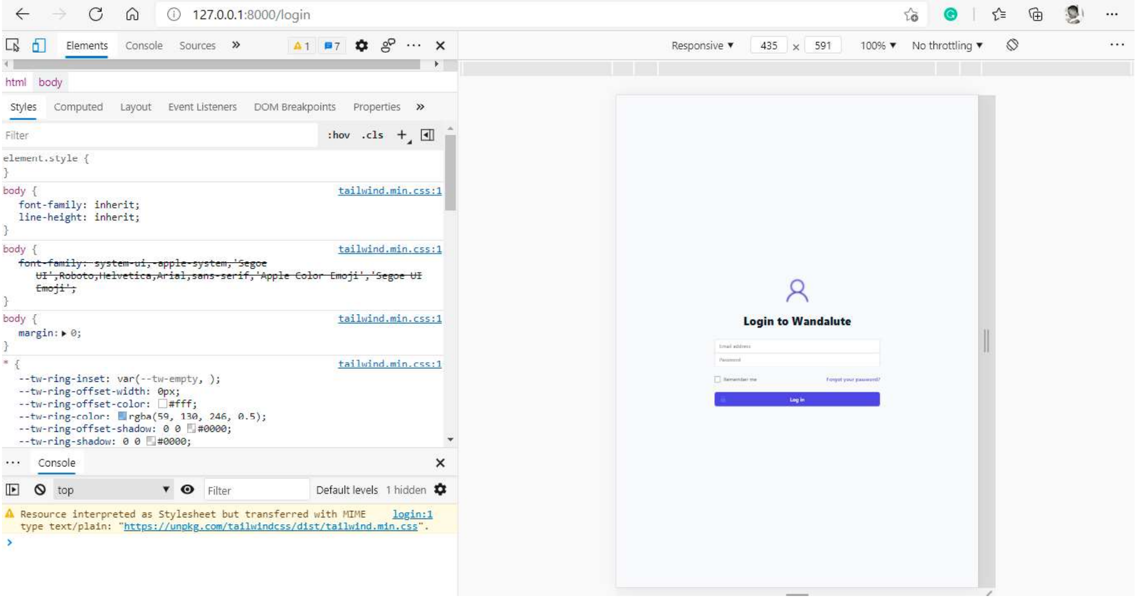
Ensuring that each component of the system works as it ought to upon implementing each component, it was verified as to whether it works as it ought to. For instance, ensuring that a form was getting the correct type of data and all the required data. Another crucial check was with button click actions and ensuring that the action either returns the right page or sends data to where the data is needed for another application functionality to be met.

Table 5.1.1: Component testing

Component Name	Purpose	Expected Input	Expected Output	Meets Requirement
Booking System	Enable staff members to book drivers available for the day they have selected	Date, time and details of the activity	A booking notification and record of the booking visible to the user, selected driver and system admin.	Yes, a staff member can make a booking, and that record is stored.
Driver-to-staff	A driver is assigned to a staff member	Based on booking	A driver is assigned to the staff member who made the request	Yes, drivers are assigned to staff members and re-assigned if need be
Notifications	Inform the user of an action status	Submission of a form and general selections	A pop-up on the screen notifying the user with either a success message or a warning	Yes, upon making any submission, the user is notified.
Calendar	For date selections	Date	Results of available drivers and schedule slots if user is a driver	Yes, and only valid selection dates are made available.

5.1.3 System testing

The application was tested on different browsers such as Brave, Mozilla Firefox, Chrome, and Microsoft Edge. In addition to this, the system was tested for responsiveness to different media sizes. See Figure 5.4 how the login of Wandalute appears on a smaller media device.



```
element.style {
}
body {                                                        tailwind.min.css:1
   font-family: inherit;
   line-height: inherit;
}
body {                                                        tailwind.min.css:1
   ~~font-family: system-ui,-apple-system,'Segoe
      UI',Roboto,Helvetica,Arial,sans-serif,'Apple Color Emoji','Segoe UI
      Emoji';~~
}
body {                                                        tailwind.min.css:1
   margin: ▶ 0;
}
* {                                                           tailwind.min.css:1
   --tw-ring-inset: var(--tw-empty, );
   --tw-ring-offset-width: 0px;
   --tw-ring-offset-color: □#fff;
   --tw-ring-color: ■rgba(59, 130, 246, 0.5);
   --tw-ring-offset-shadow: 0 0 □#0000;
   --tw-ring-shadow: 0 0 □#0000;
```

```
⚠ Resource interpreted as Stylesheet but transferred with MIME      login:1
  type text/plain: "https://unpkg.com/tailwindcss/dist/tailwind.min.css".
>
```

Figure 5.4: Screenshot of Wandalute on a smaller screen size rendered by the browser

5.2 User testing

The product to be adopted by the target user needs to be accepted by the user, and this can be accessed through Acceptance Testing. This test also helps highlight the user experience offered by the application or system based on how the participant/user tries to achieve a task.

5.2.1 Acceptance testing

Acceptance testing was conducted to help in demonstrating that required business functions were operating in a manner suited to the user requirements and real-world circumstances and usage. The acceptance tests and results are shown in Table 5.3

Table 5.2.1: Acceptance testing

Requirement/Use Case	Meets Expectation	Comments
Booking	Meets expectation	The participant was able to make a booking easily.
Registration of users	Meets expectations	The participant was able to register a new user (driver)
Available slots	Meets expectation	The participant was able to make there availability schedule
Re-assignment	Incomplete as it does not quite meet the expectation	The admin has to manually provide all the booking details again, which is quite inconvenient.

5.3 Other testing

This entails any other testing that was conducted to ensure the system works as it ought to.

Table 5.3.1: Other testing

Attribute	Purpose	Method	Expected Output	Meets Requirement
UI Testing	To find out whether the user interface is appealing	Asked the participant what they liked and did not like about the system	Appealing design	Somewhat, since the user did was not able to identify the 'Click here to login' on the landing page as a button.
UX Testing	To find out how the user felt while using the system.	Asked the participant how they found the application on an emotional level	Great feeling about the application and wanting to visit it more often.	Yes, the user felt comfortable using the application and nothing frustrating.
Mobile Friendliness	To find out whether the application renders well on mobile devices	Used the browsers in-built virtual devices	The application had a good screen to component ratio	Yes, but this can be better verified using an actual mobile device.
Browser Support	To find out whether the application works well on different web browsers.	Ran the application in different browsers (Microsoft Edge, Google Chome, Mozilla	Work perfectly fine.	Yes, the application was able to open and run in the different browsers tested.

		Firefox and Brave)		

5.4 Test results analysis

The application passed the component testing successfully as the various aspects of the system are working as they ought to and handles most errors properly wherever applicable.

Chapter 6: Conclusion

This chapter concludes the project by providing an overview of how the system has met the users' requirements. The chapter also highlights some of the challenges and insights gathered that might be useful for future works in making the system much more efficient.

6.1 Challenges faced

The work thought it would be nice to include Google Map features into the system. The plan was to have the user pin the location they intended on heading to instead of providing it as raw text input. The challenge came in with changes that Google maps have incorporated concerning the access of API keys which involves now billing. In addition to this, finding a Google Map Laravel Package that supports Laravel 8 was a challenge since finding such a package would make implementing the feature easier since the Google Maps JavaScript code would have been adequately wrapped in the Laravel 8 PHP framework.

Another challenge was testing its mobile-friendliness in its entirety. Unfortunately, this was not possible since Wandalute is currently being hosted locally but intends to be deployed soon.

6.2 Recommendations and future work

- Wandalute mobile application version to support push notifications.
- Map navigation system integration for accessible provision of directions.
- Chat feature that would enable the three users to share messages on the app.
- Incorporation of a function that overrides the system to cater for emergencies.

6.3 Conclusion

In conclusion, this work addresses the SDG's 8 and 9 that it intended to address. SDG 8, which entails decent work and economic growth, also aims at achieving higher levels of economic productivity through diversification, technological upgrading and innovation. Through Wandalute, the workers would be more efficient and have less to worry about with respect to booking office transportation with this new technology. On the other hand, SDG 9, which dwells on industry, innovation and infrastructure, with a focus on promoting inclusive and sustainable innovations. Wandalute contributes to this by providing a system that would be beneficial to the workforce whilst being long sustaining in nature. Wadalute serves to help Hope effectively execute his Logistics Office duties regarding managing the office's transport.

References

[1] A. Dorofeev, “Development of Internet-Based Applications for Fleet Management and Logistics,” 2013 IEEE 15th Conference on Business Informatics, Vienna, 2013, pp. 428-432, doi: 10.1109/CBI.2013.69.

[2] M. P. Fanti, G. Iacobellis, A. M. Mangini, I. Precchiazzi and W. Ukovich, “A flexible platform for intermodal transportation and integrated logistics,” 2017 IEEE International Conference on Service Operations and Logistics, and Informatics (SOLI), Bari, 2017, pp. 224-229, doi: 10.1109/SOLI.2017.8120998.

[3] United Nations. Department of Economic and Social Affairs: Sustainable development. Retrieved from https://sdgs.un.org/goals/goal8

[4] United Nations. Department of Economic and Social Affairs: Sustainable development. Retrieved from https://sdgs.un.org/goals/goal9

[5] Amir Ghahrai. 2008. Component Testing - Software Testing Life Cycle. Testing Excellence. Retrieved March 29, 2019 from https://www.testingexcellence.com/component-testing

Appendix

A.1 Interview Questions (Approved by supervisor)

- What are the resources involved in transportation logistics at your company?
- Do these resources have shared features or functionalities?
- Do all the vehicles have the same sitting capacities?
- Would you like for the system to have a booking feature?
- How would you like to access this application (e.g., on one’s phone, desktop etc.)?
- Do all stakeholders have access to the internet regularly?
- Would you like the system to provide notifications?
- If so, how would you like the notification alerts to be received?
- Would you like to have a calendar feature in the application?
- Which other persons would you like to have access to the admin dashboard?
- What other additional features would you want to see in this system?

A.2 Project Management Gantt Chart

Distribution of project tasks and activities (August 2020 to April 2021)

	Aug.	**Sep.**	**Oct.**	**Nov.**	**Dec.**	**Jan.**	**Feb.**	**Mar.**	**Apr.**
Problem Identification	■	■							
Requirement Analysis			■	■					
Architecture and Design				■	■				
Project Implementation						■	■	■	
Conducting Testing									■