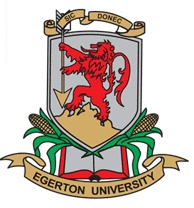
**EGERTON UNIVERSITY**



**SYSTEM DESIGN**

**DOCUMENT**

**FOR**

**EGERTON UNIVERSITY TRANSPORT**

**MANAGEMENT SYSTEM**

**PREPARED BY: PAUL CYRIL OYUNGU**

**REG NO: S13/21403/14**

**PROJECT SUPERVISOR: KIMANI NJOROGE**

**PROJECT COORDINATOR: DR. WILFRED GIKARU**

**DATE: 18/7/2018**

**VERSION 1.0**

***OVERVIEW***

**TABLE OF CONTENTS**

Contents

[1. INTRODUCTION 5](#_Toc520122415)

[1.1Purpose and Scope 5](#_Toc520122416)

[1.2Project Executive Summary 5](#_Toc520122417)

[1.2.1 System Overview 5](#_Toc520122418)

[1.2.2Design Constraints 8](#_Toc520122419)

[1.2.3Future Contingencies 8](#_Toc520122420)

[1.3Document Organization 9](#_Toc520122421)

[1.4Project References 9](#_Toc520122422)

[1.5Glossary 9](#_Toc520122423)

[2.SYSTEM ARCHITECTURE 10](#_Toc520122424)

[2.1. System Hardware Architecture 10](#_Toc520122425)

[2.2System Software Architecture 11](#_Toc520122426)

[2.3Internal Communications Architecture 12](#_Toc520122427)

[3.FILE AND DATABASE DESIGN 13](#_Toc520122428)

[3.1Database Management System Files 13](#_Toc520122429)

[3.2. Non-Database Management System Files 17](#_Toc520122430)

[4.HUMAN-MACHINE INTERFACE 17](#_Toc520122431)

[4.1. Inputs 18](#_Toc520122432)

[4.2. Outputs 24](#_Toc520122433)

[5. DETAILED DESIGN 24](#_Toc520122434)

[5.1 Hardware Detailed Design 24](#_Toc520122435)

[5.2 Software Detailed Design 24](#_Toc520122436)

[5.3 Internal Communications Detailed Design 25](#_Toc520122437)

[6. EXTERNAL INTERFACES 25](#_Toc520122438)

[6.1. Interface Architecture 25](#_Toc520122439)

[7.SYSTEM INTEGRITY CONTROLS 25](#_Toc520122440)

## 1. INTRODUCTION

## 1.1Purpose and Scope

This software design document describes the architecture and system design of Egerton University Transport Management System(EUTMS)

This document contains a complete description of the design and architecture of the EUTMS. The basic architecture is a web based application that users will access the system through the web. The description of the architecture and design will mainly be on the:

* Login and registration process
* Booking process
* Database architecture
* Inventory and staff management process

## 1.2Project Executive Summary

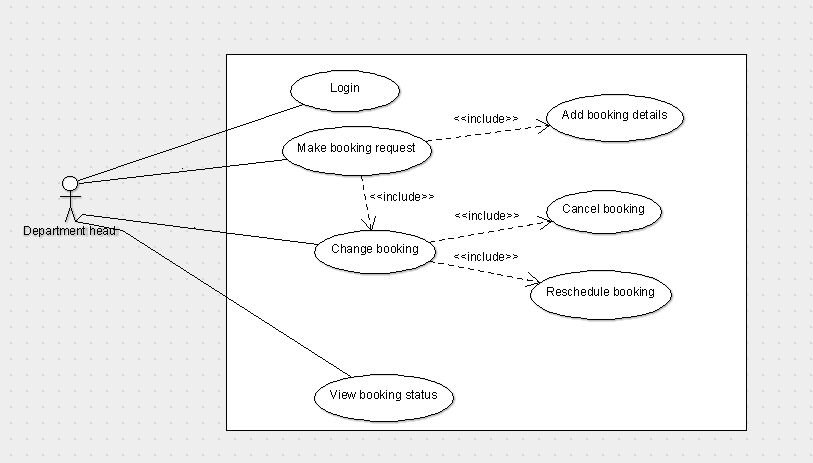
## 1.2.1 System Overview

EUTMS is responsible for managing most if not all activities done by the transport department of the university. The system will be broken down into several parts or modules to take care of the problem the system is trying to solve such as conflicting allocations, booking process, management of staff and inventory management

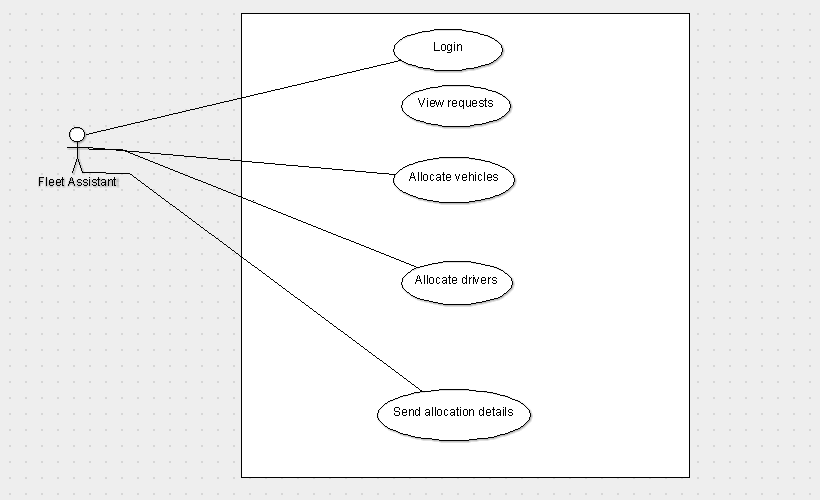
First off, there will be a login and registration section where users (transport manager, fleet assistant, mechanic head and head of department) of the system will login in or register if they don’t have an account. Login will need a staff id and a password that will be authenticated by the system and allowed into the system depending on the role. If the user has no account, the user will have to register with a staff id, an email address and a password

After login, each user will be redirected to their respective page depending on their role. A head of department would be allowed to make bookings, change a booking request by either rescheduling or canceling the booking request and later view the booking status. The fleet assistant would view requests, then allocated vehicle and driver depending on the availability of a vehicle or driver, after which a report would be sent to the transport manager on the allocation where there might be an allocation or none. The mechanic manager would use the system to add and allocate mechanics to vehicles, record replaced parts, provide vehicle status and make a request on equipment needed. The transport manager would then use the system to manage all users of the system, notify departments on allocation, provide requisition on equipment and add or remove drivers and vehicles.

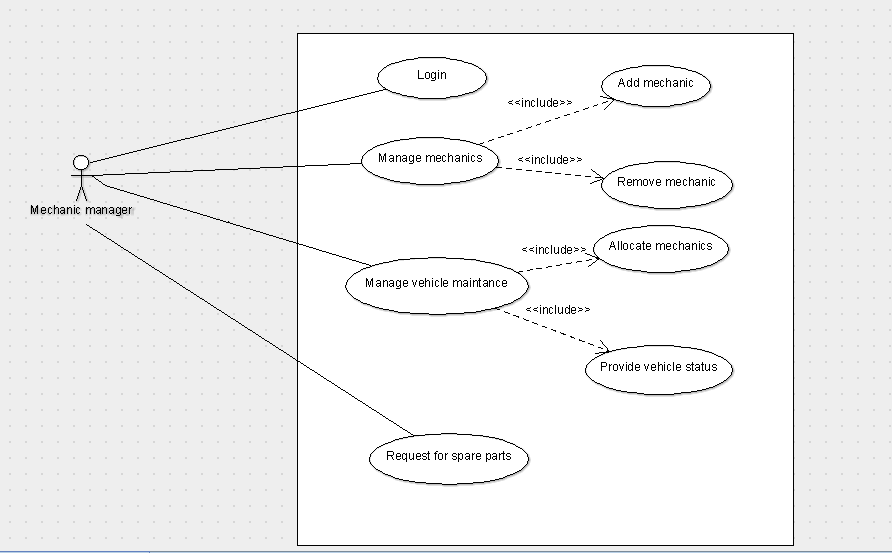
The system ids going to automate this process by storing all data, coming up with a schedule, fix the conflicting allocations problem and show what spare parts are available



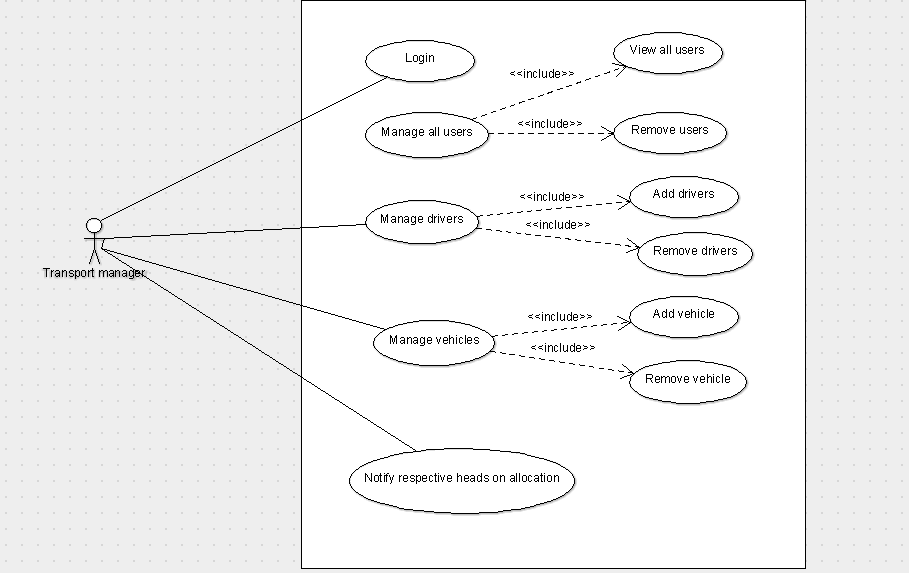
**Figure 1: *Department head uses case***

****

**Figure 2: *Fleet Assistant use case***

****

**Figure 3: *Mechanic manager use case***

**Figure 4: *Transport manager use case***

## 1.2.2Design Constraints

There are several design constraints and limitations that come up when coming up with the design such us:

* Flexibility – the design has to be able to deal with the changes in a module that might arise. Coming up with a design that can be easy to deal with errors in a module
* Acceptability - an assumption had to be made on the visual design being liked or not. Also the flow of activities in the system had to be assumed that it would be liked by the user
* Integration – the design of the system integrating with other external systems was a problem since it would affect the system overall working

## 1.2.3Future Contingencies

Here are some of the likely circumstance that might lead to changes in the system plan:

* Requirements – there might be changes in requirements where features might need to be added or removed in the system depending on the departments needed I that particular time

## 1.3Document Organization

The current document is organized as follows:

* System architecture – this presents the system architecture of the EUTMS project including its subsystems by viewing the system from various perspectives such as the hardware architecture, software architecture and the internal communication architecture
* File and database design – this presents the system’s file and database organization and design. Gives the full and final design of the system’s database management system files including non-database management system files
* Human machine interface – this presents the design of system’s and subsystem’s inputs and outputs related to the users in details
* Detailed design – this presents information on hardware design, software design, and internal communication design that will be integrated together into the system
* External interfaces – presents information on the systems that are not within the scope of the EUTMS
* System integrity controls – presents information on the security and level of access to some information on the system

## 1.4Project References

SDD sys\_design\_doc.pdf by Dr. Wilfred Gikaru

## 1.5Glossary

EUTMS- Egerton University Transport Management System

SDD – System Design Document

Dr. – Doctor

TCP/IP - Transfer Control Protocol

CSRF - Cross-Site Request Forgery

MVT – Model View Template

MVC – Model View Controller

HTTP – Hyper Text Transfer Protocol

URL – Uniform Resource Locator

SQL – Structured Query Language

## 2.SYSTEM ARCHITECTURE

## 2.1. System Hardware Architecture

The EUTMS will be hosted in a single server since the system will have a handful of users. The system may undergo horizontal scaling if users become more. The server has main memory, secondary memory and the central processing unit each with their functionality. All may application data will reside in a PostgreSQL database. The system applications will be written in Django-flavored Python which will be served up in server. Since the system is web based, there will be a browser which will communicate with a server using HTTP and the server will communicate with the Django application using a socket as show in the figure below

Single server

HTTP

Application

(Django)

Database

(PostgreSQL)

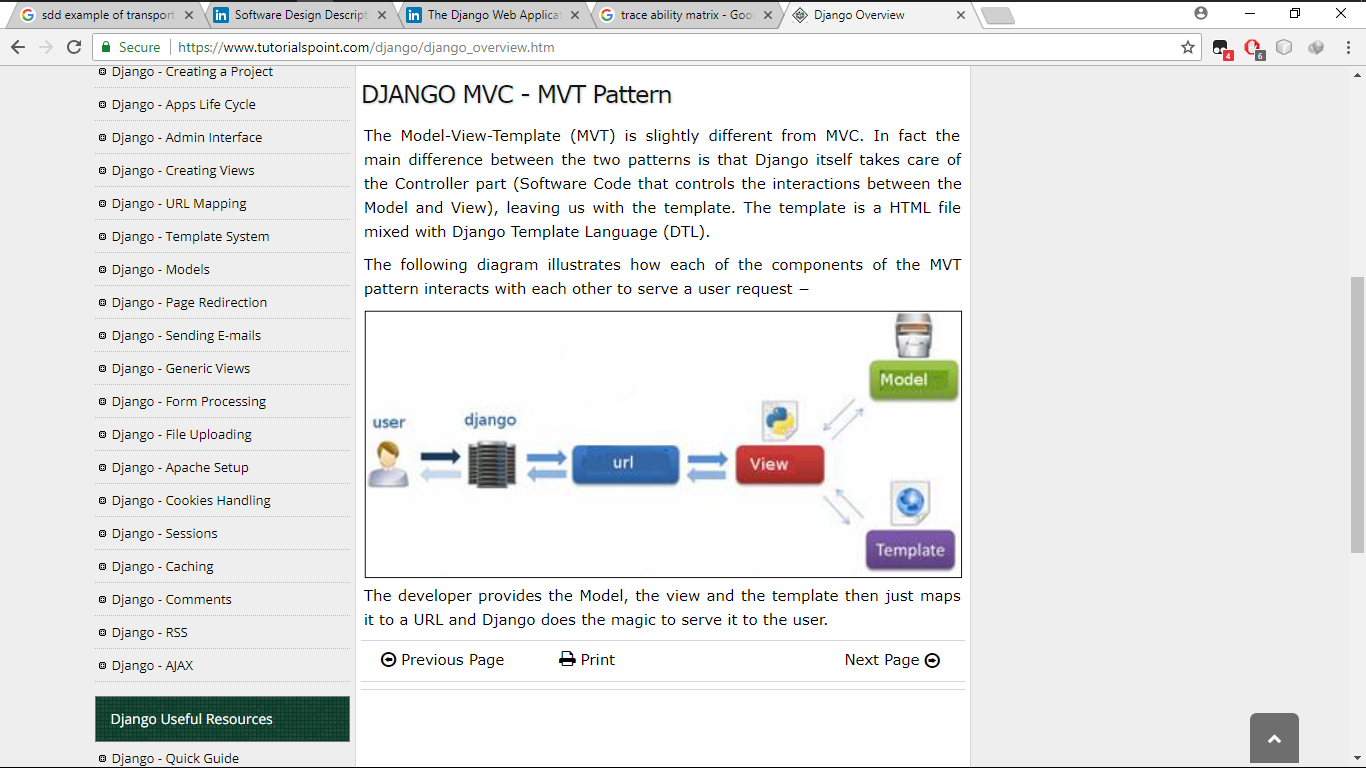
Webserver

Browser

## 2.2System Software Architecture

The system is going to be developed in Python programming language using the Django web development framework. Django is organized using the MVT architecture which is similar to the MVC architecture as shown below:

* URL- They are used to process every single URL via a single function. Based on URL requests, URL mapper is used to redirect HTTP requests to the appropriate views
* Models- They are objects that define structure of the data of an application and provide a mechanism to manage and query records in a database
* Views – is a request handler function that receives HTTP requests and then returns a HTTP response they access data needed using the models and then delegate response formatting to templates
* Template – is a text file that defines the structure and layout of a file such as a html page with placeholders used to represent the actual content



***Django framework architecture***

Browser

URL dispacher

Template

Views

Models

Database

The following software will be used:

* **VS Code** – this is an IDE that will be used to coding and development of the system
* **PostgreSQL server**- this is a database management server that will be used in the management of all database files and functions
* **Django server** – Django has its own built in lightweight and standalone server that will be used in development and testing of the system
* **NginX server** – this is an application server that can be used to execute python based applications and has the capability to serve the static content of Django directly. This will be used in the production setting of the system
* **Web Browser** – this will be used to access the system when the system’s server is on. Examples include google chrome, Firefox, Opera mini and Safari

## 2.3Internal Communications Architecture

Communication between devices or computers using the system will only be possible using the internet. A user will use a browser to access the system by making requests to the web server using HTTP protocol requests.

## 3.FILE AND DATABASE DESIGN

## 3.1Database Management System Files

Egerton University Transport management system will store its data in a database that will have seven tables:

* Users
* Staff
* Drivers
* Vehicles
* Mechanics
* Spare Parts
* Requests
* Bus Allocation

**STAFF**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **column** | **Constraints** | **Data type** | **size** | **Description** |
| 1.StaffId | Primary key | Character varying | 15 | Unique identifier of staff |
| 2.Contact |  | Character varying | 15 | Contact of staff member |
| 3.Email |  | Character varying | 100 | Email of staff member |
| 4.Name |  | Character varying | 200 | Name of staff member |
| 5.Department |  | Character varying | 100 | Department of staff member |

**USERS TABLE**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Column** | **constraints** | **Data type** | **size** | **Description** |
| 1.StaffId | Primary key, Foreign key | Character varying | 15 | Staff id of the user |
| 2.Password |  | Character varying | 100 | Password of user |
| 3.Role |  | Small integer | 2 | Role of the user |

**DRIVERS TABLE**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Column** | | **constraints** | | **Data type** | | **size** | | **Description** |
| 1.StaffId | | Primary key, Foreign key | | Character varying | | 15 | | Staff id of the user |
| 2.Availability | |  | | Boolean | |  | | Availability of the driver |
| 3.Date |  | | Date | |  | | Date the driver is available | |
|  | |  | |  | |  | |  |

**MECHANICS TABLE**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Column** | | **constraints** | | **Data type** | | **size** | | **Description** |
| 1.StaffId | | Primary key, Foreign key | | Character varying | | 15 | | Staff id of the mechanics |
| 2.Availability | |  | | Boolean | |  | | Availability of the mechanic |
| 3.Date |  | | Date | |  | | Date the mechanic is available | |

**VEHICLES TABLE**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Column** | **constraints** | **Data type** | **size** | **Description** |
| 1. Number\_plate | Primary key | Character varying | 11 | Unique identifier to vehicle |
| 2.Vehicle\_type |  | Character varying | 15 | The type of vehicle |
| 3.Engine\_capacity |  | Character varying | 10 | The capacity of the engine |
| 4.Capacity |  | Integer | 11 | Capacity of vehicle |
| 5.Driver\_id | Foreign key |  | 15 | driver Identifier |
| 6.Mechanic\_id | Foreign key |  | 11 | mechanic Identifier |

**SPARE PARTS TABLE**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Column** | **constraints** | **Data type** | **size** | **Description** |
| 1.Spare\_parts\_id | Primary key | Integer | 11 | Unique identifier to spare part |
| 2.Amount |  | Integer | 11 | Quantity of spare part |
| 3.Name |  | Character varying | 100 | Name of the spare part |
| 4.Cost |  | Double | 11 | Cost of the spare part |
| 5. Description |  | Character varying | 100 | Description of the spare part |

**REQUESTS TABLE**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **column** | **constraints** | **Data type** | **size** | **Description** |
| 1.Request\_id | Primary key | integer | 11 | Unique identifier to request |
| 2.Deptrequesting |  | Character varying | 100 | Department making the request |
| 3.Reason |  | Character varying | 100 | Reason for requesting |
| 4.Travel\_date |  | Date | 10 | Day of travelling |
| 5.Return\_date |  | Date | 11 | Day of travelling back |
| 6.Destination |  | Character varying | 40 | Destination |
| 7.Travellers\_desc |  | SmallInt | 1 | Type of travelers i.e. staff |
| 8.Capacity |  | Integer | 11 | Number of travelers |
| 9.User\_id | Foreign key |  | 15 | User Identifier |
| 10.Confirm\_status |  | Boolean(default=false) |  | Shows if request is confirmed or denied |

**BUS ALLOCATION TABLE**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **column** | **Constraints** | **Data type** | **size** | **Description** |
| 1.REquest\_id | Primary key, Foreign key | Integer | 11 | Unique identifier to booking made |
| 2.Driver fee |  | Integer | 11 | Fee paid to driver |
| 3.Fuel money |  | Integer | 11 | Fueling cost |
| 4.Vehicle\_id | Foreign key |  | 11 | Vehicle Identifier |
| 5Estimated distance |  | Integer | 11 | Estimated distance of travel |
| 7.Travel\_date | Foreign key |  |  | Foreign key of the request date |
| 6.Driver\_id | Foreign key |  | 15 | Driver Identifier |

**1**

**STAFF**

**1**

Profile info

Profile info

**1**

Manages

**1**

**N 1** **1**

Profile info

**MECHANICS**

**USERS**

**N** **1**

Work on

Provide requisition

**1** **1**

**N**

Booking request

**DRIVERS**

**N N**

**SPARE PARTS**

Drives

**N 1 N**

**VEHICLES**

**REQUESTS**

Driver info

**1 1**

Allocation

Vehicle info

**BUS ALLOCATION**

**N**

**N**

## 3.2. Non-Database Management System Files

## 4.HUMAN-MACHINE INTERFACE

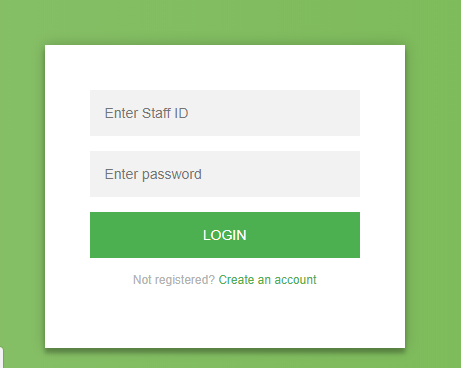
The user interface of this system has five major interfaces, four of which are of the users the system. There will be the login and registration pages that will prompt users for some given data in order to login or register. There will also be an interface for the transport manger with several options for him/her to carry out activities or manage the transport department. There is also an interface for the fleet assistant with section to enable manage requests. Department heads will also have an interface to make bookings and view the status. Mechanic manager will also have an interface to manage mechanic and spare parts

## 4.1. Inputs

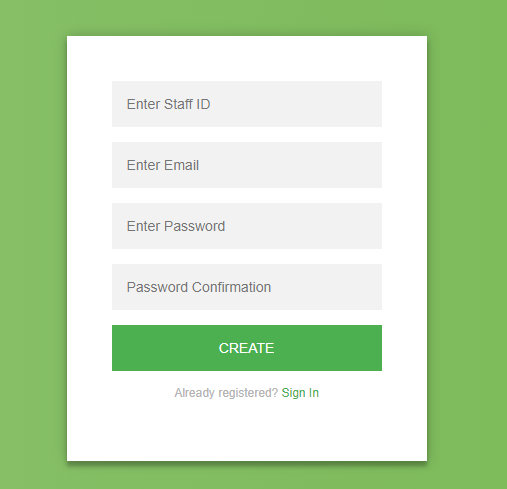
This section describes input media used by the operators of the system for providing information to the system. Users of the system are required to input data depending on the form they are trying to fill. Each input has its own data type and restrictions when a user is trying to input data. For instance, to input an email address during registration, restrictions have been put in place to ensure email format match the format of an email filed as specified by the system. If it’s a wrong format, the user will be prompted for a correct email format. Since all operators of the system will register and login with a staff id, the system has a regular expression to capture the correct format else sends an error message that the format is wrong

All data entered will be stored in a database right after going a verification and authentication process for the case of login and registration. A Cross-Site Request Forgery (CSRF) token will be used to ensure only users who are authorized and are part of the system can access it

The login process will need a staff id and password as inputs. Staff id should be in the correct format and exist in the system and the password should have a length of at least eight. Registration process needs a staff id, email and a password with its confirmation. Below are the images of the login and registration page

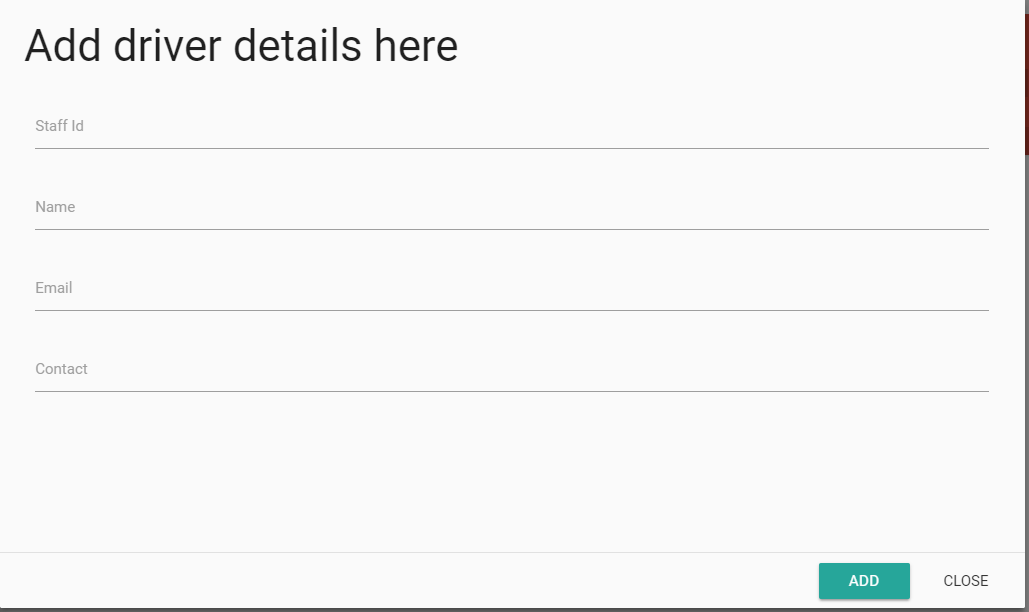


***Login form***

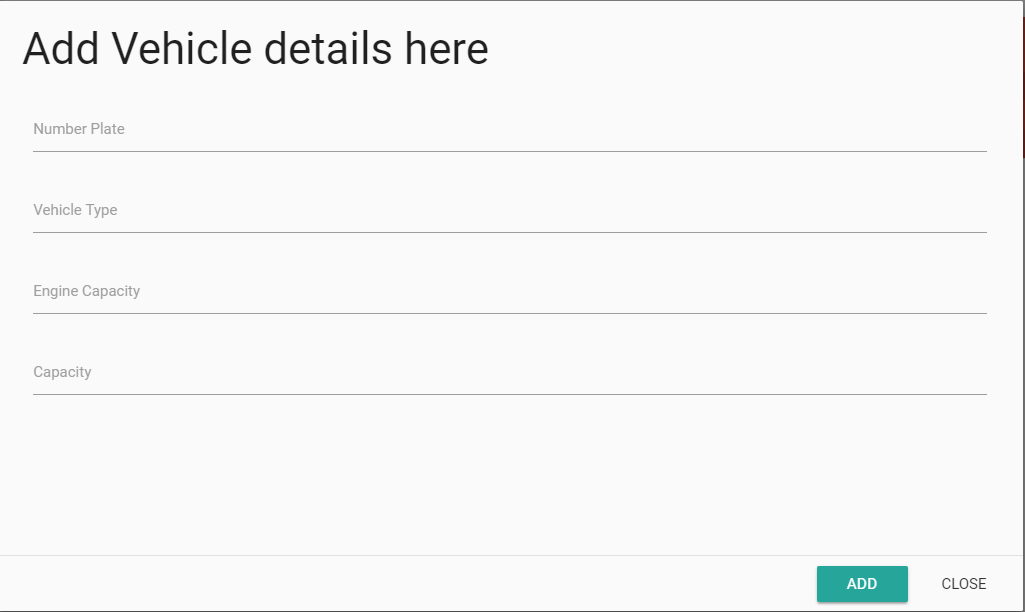


***Registration form***

Once logged in the transport manager will have two input form: adding drivers and adding vehicles. Driver inputs include staff id, name, email and contact. Vehicle inputs will include the followingNumber plate, engine capacity, vehicle type and the capacity

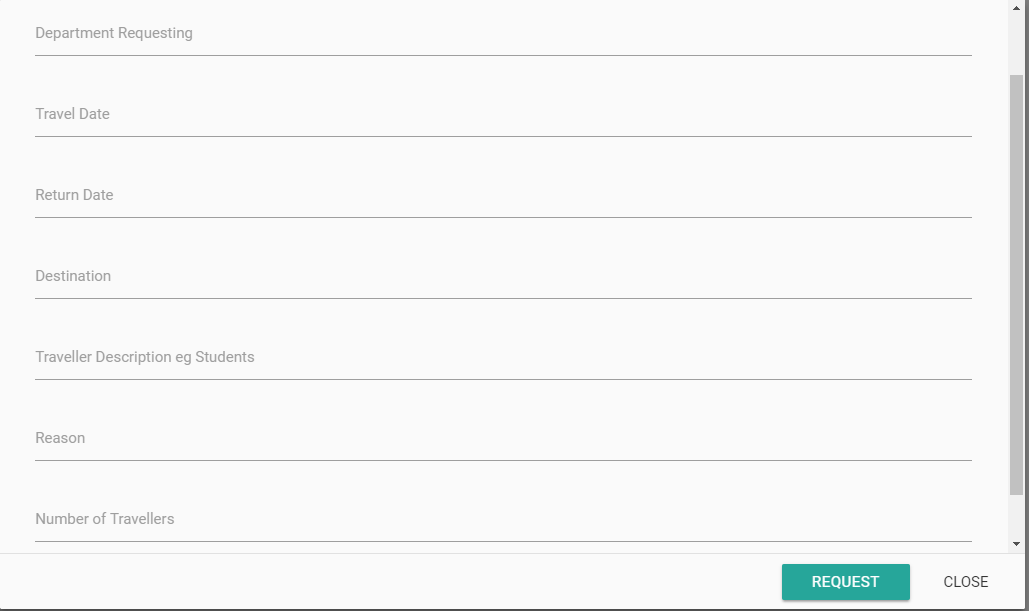


***Driver form***



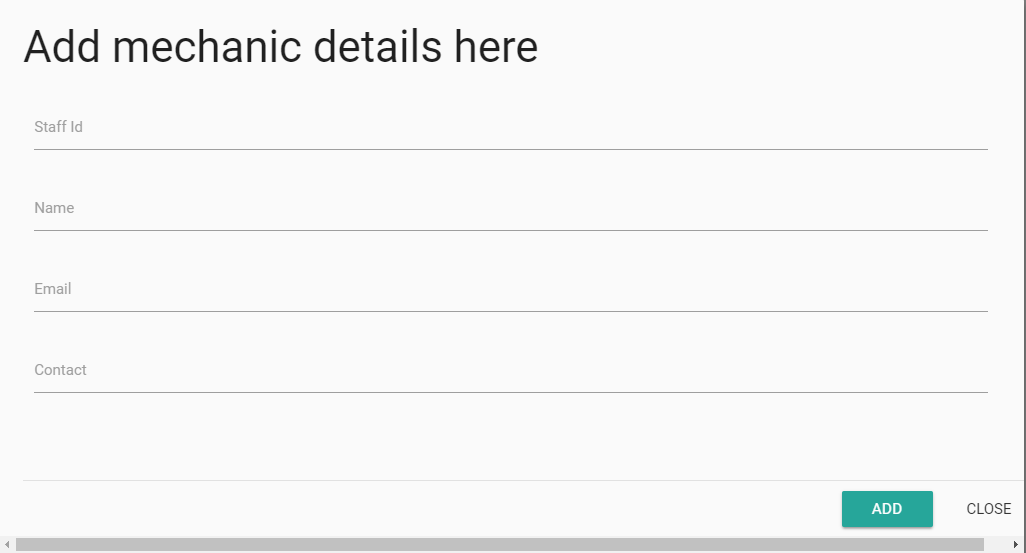
***Vehicle form***

Department heads will also have an input form to make booking requests. The following are the inputs: department requesting, travel date, return date, destination, travelers description, reason and the number of travelers

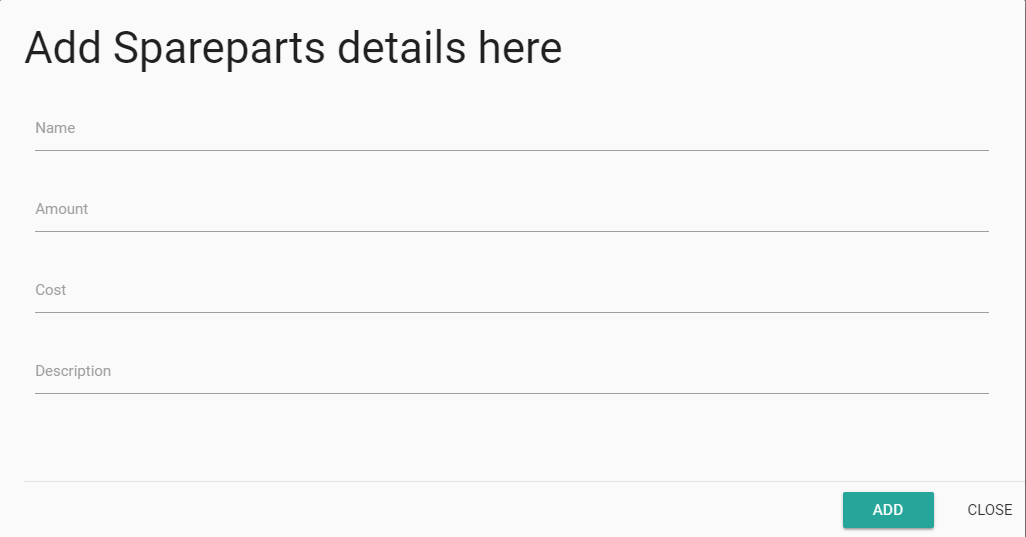


***Booking requests form***

Mechanic manager will also have two input forms: mechanics and spare parts. Mechanics form will have the following inputs: staff id, name, email, and contact. The spare parts form will have the following inputs: name, amount, cost, description



***Mechanic form***



***Spare parts form***

## 4.2. Outputs

The system has four dashboards for the four categories of users where data they need to view and are authorized to view will be displayed. Fleet assistant will view the requests made by different departments, allocations made and a list of available drivers and vehicles. Department heads will be able to view reports on the booking requests they have made and the booking status. Mechanic manager will have reports on mechanics and the spare parts in stock including those that have been used. The transport manager will have reports on booking allocations, all users of the system, drivers, vehicles and equipment requisitions

## 5. DETAILED DESIGN

## 5.1 Hardware Detailed Design

For this system to work, there is need of a web server that will help in hosting this web based system, an operator’s PC or device that can have a browser to access the system and a switch, router or hub to establish a connection between the devices

## 5.2 Software Detailed Design

There are several modules that will be used in this system in order to take care of all the functionalities and purpose of the system. They are as follows:

* **Registration –** this involves users of the system registering in order to have access of the system. The details used to register will be used an identification of the user to the system and will be used in the login module
* **Login –** this involves the users of the system entering their login details (staff id and password) in order to access the services. A user must have registered in the registration module in order to be able login to the system. A Cross-Site Request Forgery (CSRF) token will be used in this module to ensure only users who are authorized and are part of the system can access it. Once logged in the users will be redirected to their respective pages depending on their role using a decorator
* **Requests –** this module will involve a user making a booking request and the details will be stored in the database. The details can also be viewed by the user
* **Allocations –** this module involves the allocation of drivers and vehicle to a given booking request. Provides a report on allocations made that will be used to generate a final bookings schedule
* **Management –** this module’s main function is to manage all users of the system and the staff and inventory where there will be a functionality to either add or remove a given staff member, remove an operator of the system and also keep track of all the inventory (what is available and what is not)

## 5.3 Internal Communications Detailed Design

Since the system will have few users, a single server will be used at the beginning which may later undergo horizontal scaling where other servers may be added when the system grows or the number of users increases. HTTP protocol will be used to facilitate communication between the web server in which the web application will reside and the user of the system using a browser. The system will also link with the PostgreSQL database in order to have a access to the data that is essential to the system

## 6. EXTERNAL INTERFACES

## 6.1. Interface Architecture

The system being web based, there is need to use the internet hence the TCP/IP protocol will be used to have an interface between the PCs used, the server and the database used

## 7.SYSTEM INTEGRITY CONTROLS

The system takes into consideration the security of data and the system as a whole using several mechanisms:

* **Authorization** – There is restriction on what data a given user is allowed to view or access. Each user is in a category or has a role where each category of users will be given restriction on what data in the database the can access
* **Integrity** - ensures that data stored in the database cannot be violated or altered by any unauthorized parties in any way whatsoever.
* **Validation** - there will be standard tables to be used or requested for validating data fields where a given type of data can only be added to a field that will be specified by the data type in the table. A CSRF token will be used in the system login and registration to ensure only users who are authorized and are part of the system can access it and avoid an attacker forging credentials and accessing the system
* **Database control** - there will be control in what data a given user can add, delete or update in a database. The administrator will set rules or access restrictions on all the categories of users
* **Verification** – the system will ensure that the credentials of a user match during login before they can gain access to the system