BusPlanner	Version: 1.0
Design Description	Date: 2016-11-11

Distributed **S**oftware **D**evelopment:
BusPlanner
Design Description

Revision History

Date	Version	Description	Author
2016-11-11	1.0	Initial draft	Team

Contents

1 INTRODUCTION

1.1 Purpose of this document

1.2 Document organization

1.3 Intended Audience

1.4 Scope

1.5 Definitions and acronyms

1.5.1 Definitions

Keyword	Definitions
User	A person who requests for bus by being from bus stop.
Fleet Manager	Who owns the buses. He/she wants to know the utilization of buses and scheduling of buses.
User Request	Information generated with timestamps for the scheduling purpose.
Algorithm	A method used to enhance the scheduling process which is static as well as dynamic.
Sprint	A repeatable work cycle which is also known as iteration.

1.5.2 Acronyms and abbreviations

Acronym/abbreviation	Definitions
UI	User Interface
GUI	Graphical User Interface
MDH	Mlardalens Hgskola, Vsters, Sweden
POLIMI	Politecnico di Milano, Milan, Italy
QA	Quality Assurance
DSD	Distributed Software Development

2 BACKGROUND AND OBJECTIVES

2.1 Overview

The purpose of this project is to help the city of Johannesburg with the bus planning process which, as of now, lacks of efficiency and effectiveness.

For more information: http://www.fer.unizg.hr/_download/repository/Project_Plan%5B12%5D.pdf

2.2 High level description of the functionalities

The BusPlanner project is based on an algorithm that aims to reduce the number of buses per route and the time needed to do a scheduling. The users' waiting time will thus be dropped from hours to minutes.

Users will interact with the system sending requests for a bus, related to a specific position. They will also be able to view the buses' location in the city, thanks to the mapping service the system will make use of.

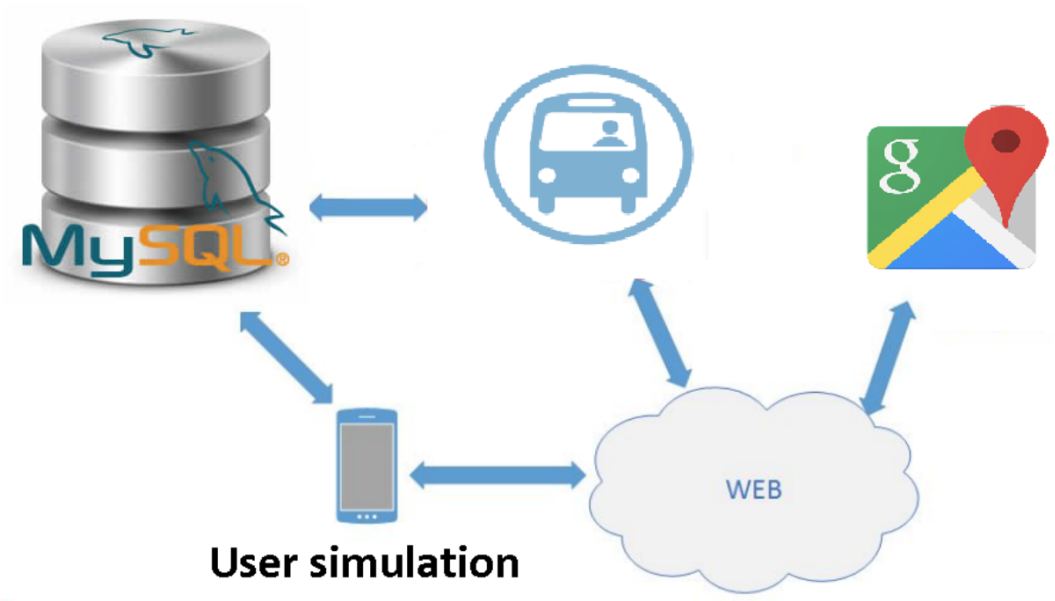
On the other hand, bus drivers will receive a notification for each user request, along with all the related information.

Fleet managers' duty will be to assign each bus to a specific route, in order to satisfy the maximum number of requests in a minimum period of time.

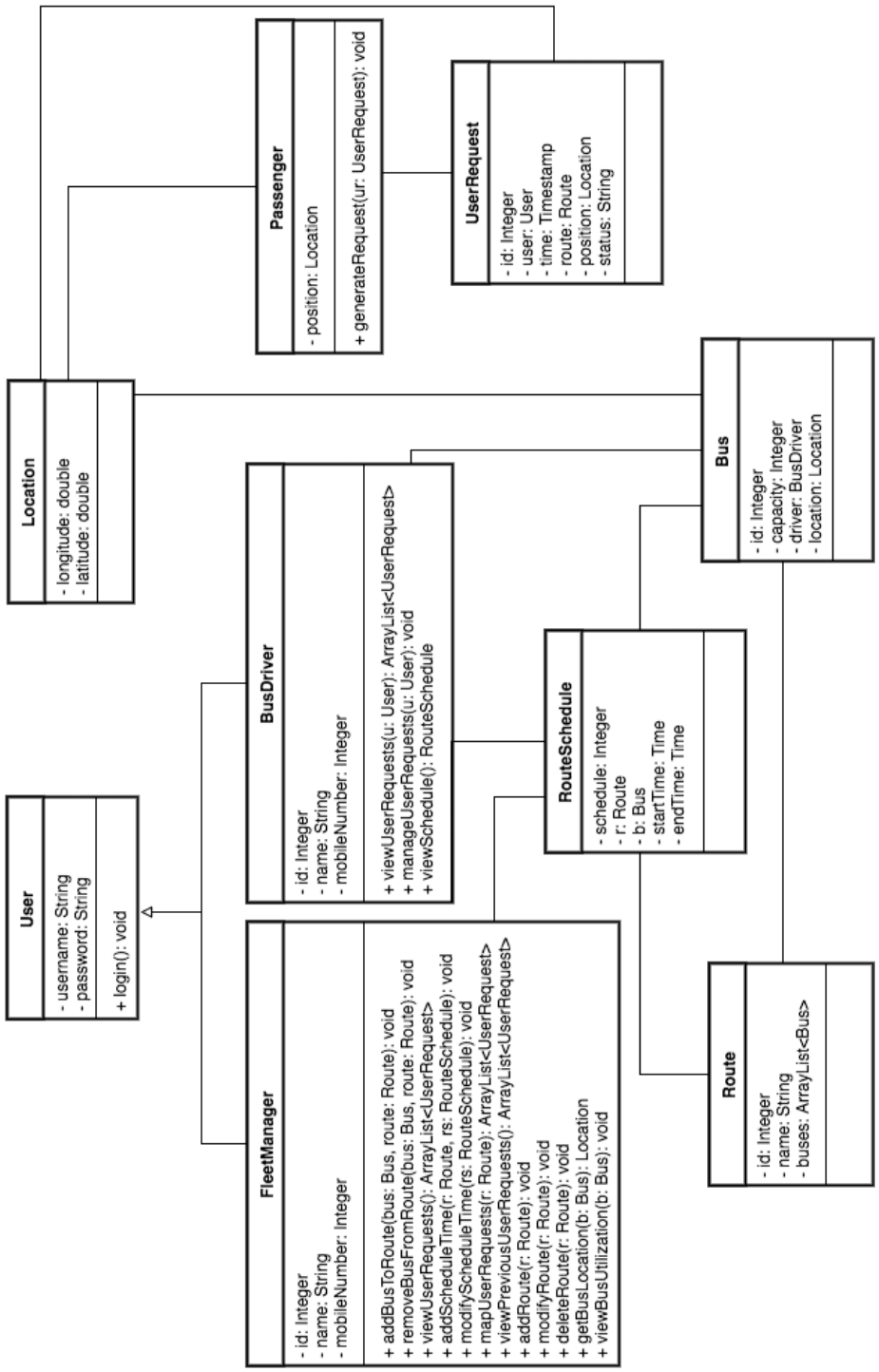
3 HIGH LEVEL SYSTEM ARCHITECTURE

The system's main components will be:

- **MySQL**, for the storage of all the information related to both the BusPlanner application and the users' requests.
- **BusPlanner application**, through which all kind of users (fleet managers, bus drivers, passengers) will interact with the system and will be able to use it over the WEB.
- **Mapping service**, to keep track of the position of both buses and users.
- **User simulation**, to simulate users' requests from a certain position.



3.1 Class diagram



4 SOFTWARE ARCHITECTURE

4.1 System architecture layers

The application consists of three layers:

- **Presentation Layer**: contains the components that implement and display the user interface and manage user interaction.
- **Business Logic Layer**: is used as an intermediary for data exchange between the presentation layer and the data access layer. Business entities, or business objects, encapsulate the business logic and data necessary to represent real world elements. The business layer's goal is to minimize the complexity by separating tasks into different areas of concern.
- **Data Access Layer**: includes methods which can interact with the database. When methods are called, it will connect to the database by using query and will return the corresponding results.

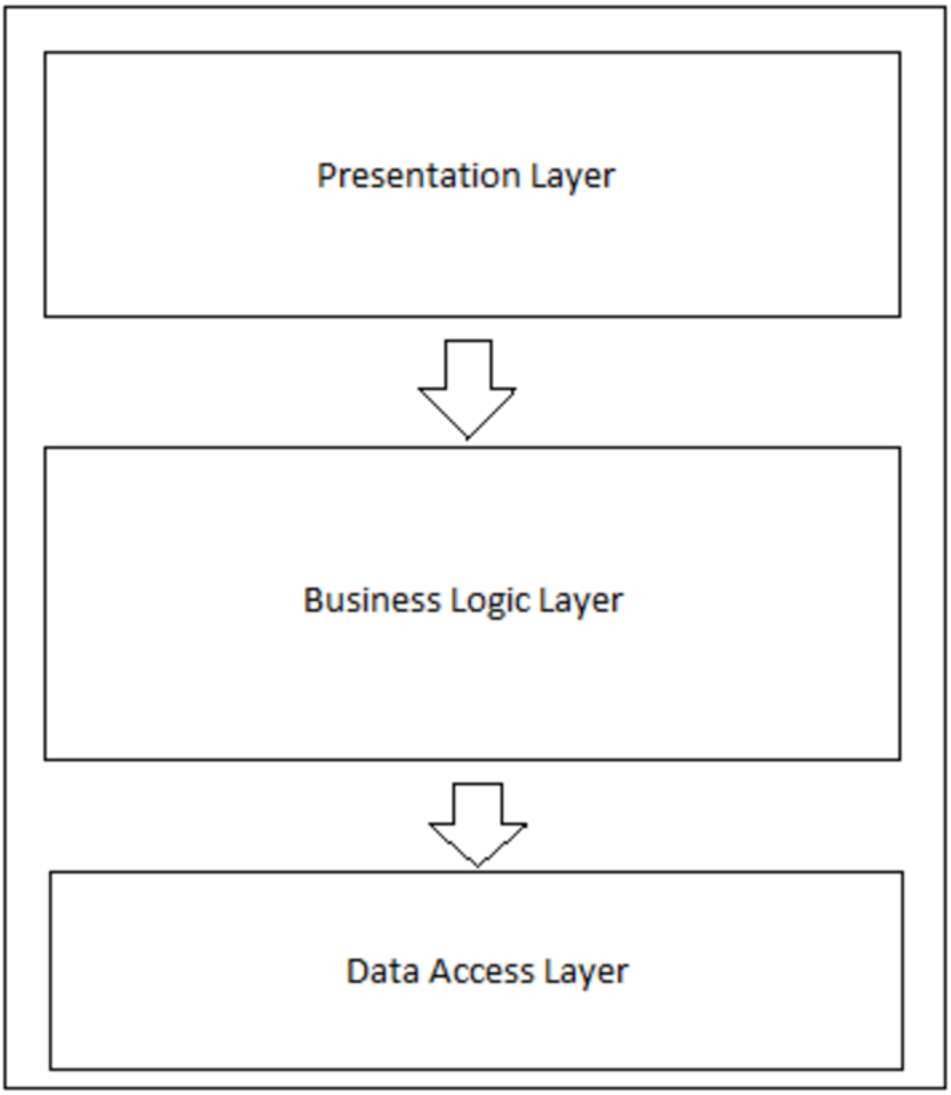


Figure 1: Three-layered architecture

With a three-layered architecture, improving the modularity of the application will be simple and this can make it easier for developers to extend features in the future. Distinguishing among the different layers allows the development team to program according to the interfaces, thus allowing an easier distribution of the work.

4.2 Technologies used

- **Back-end and database**: PHP will be used to build the back-end of the application and the database used in this application will make use of MySQL.
- **Front-end**: the application will use HTML, CSS and Bootstrap for front-end.

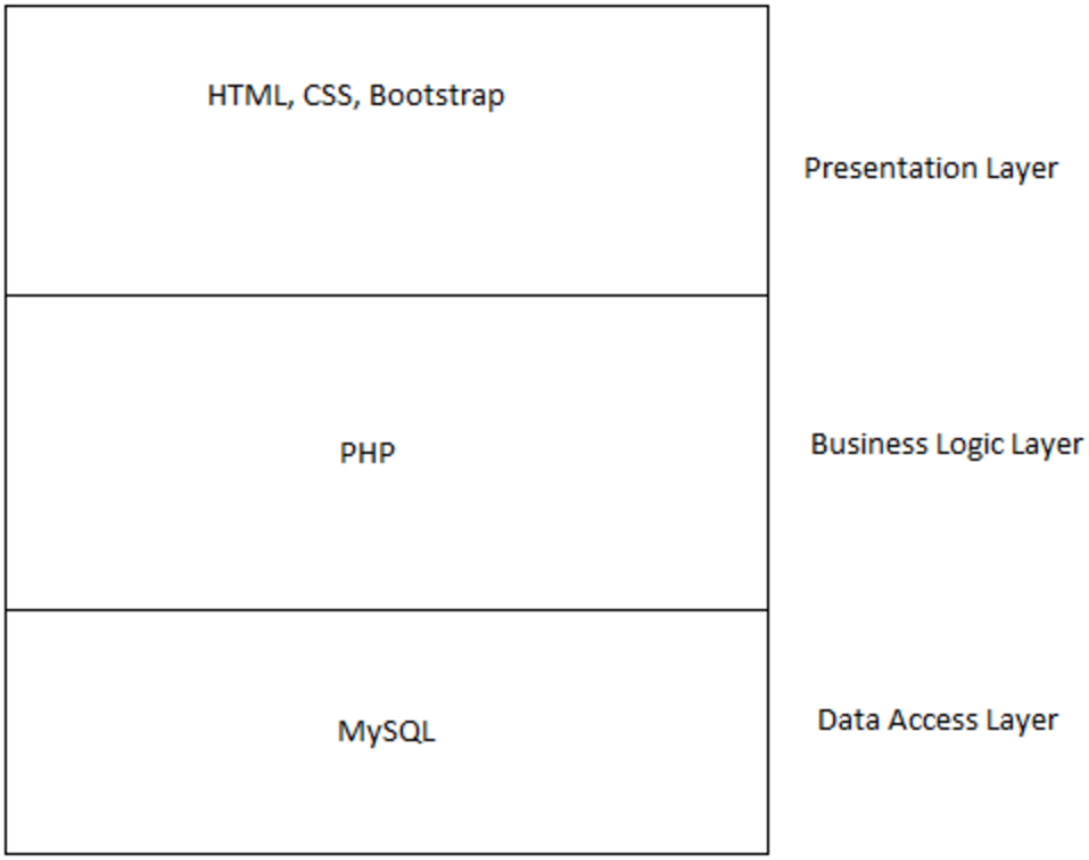
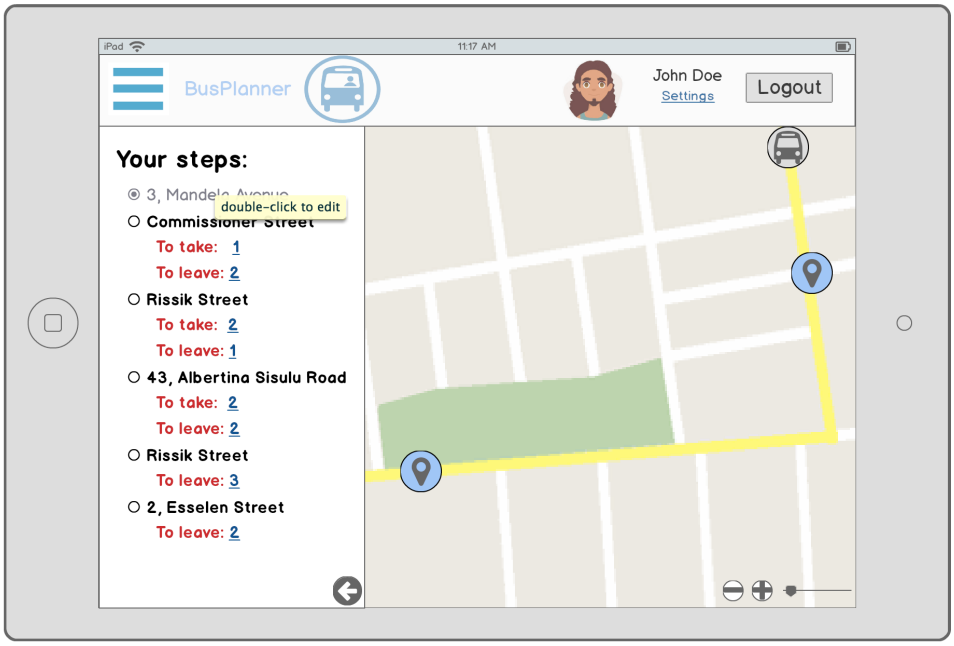


Figure 2: The technologies in the three-layered architecture of the application

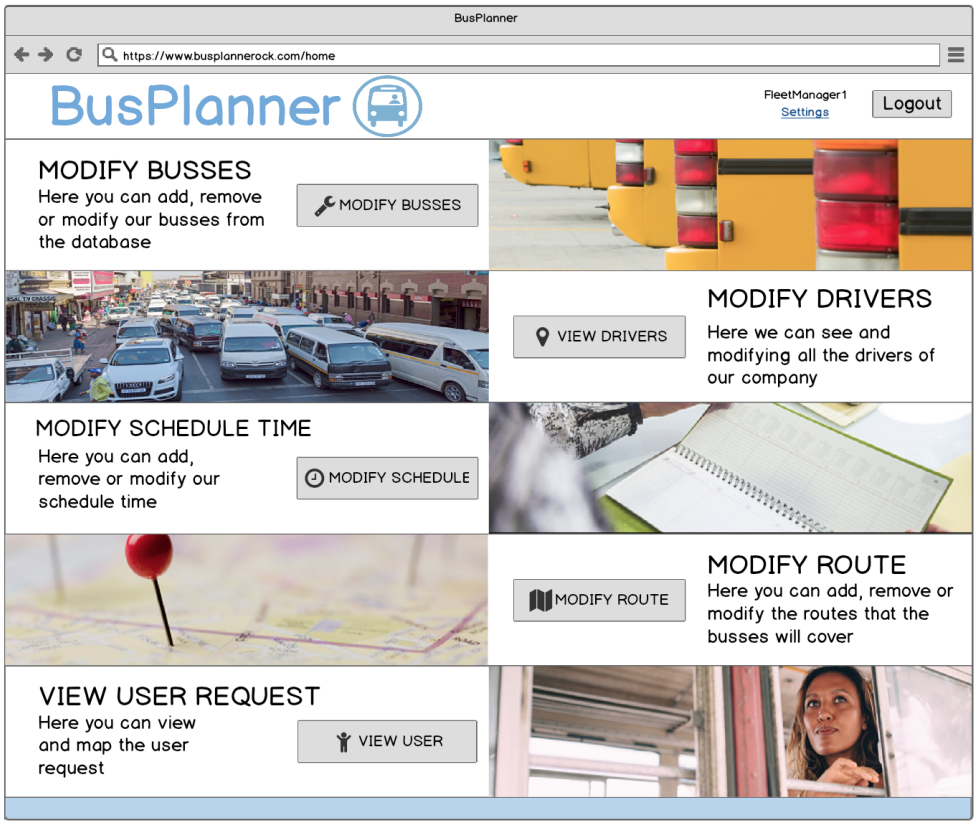
5 GRAPHICAL USER INTERFACE

- Driver's login:

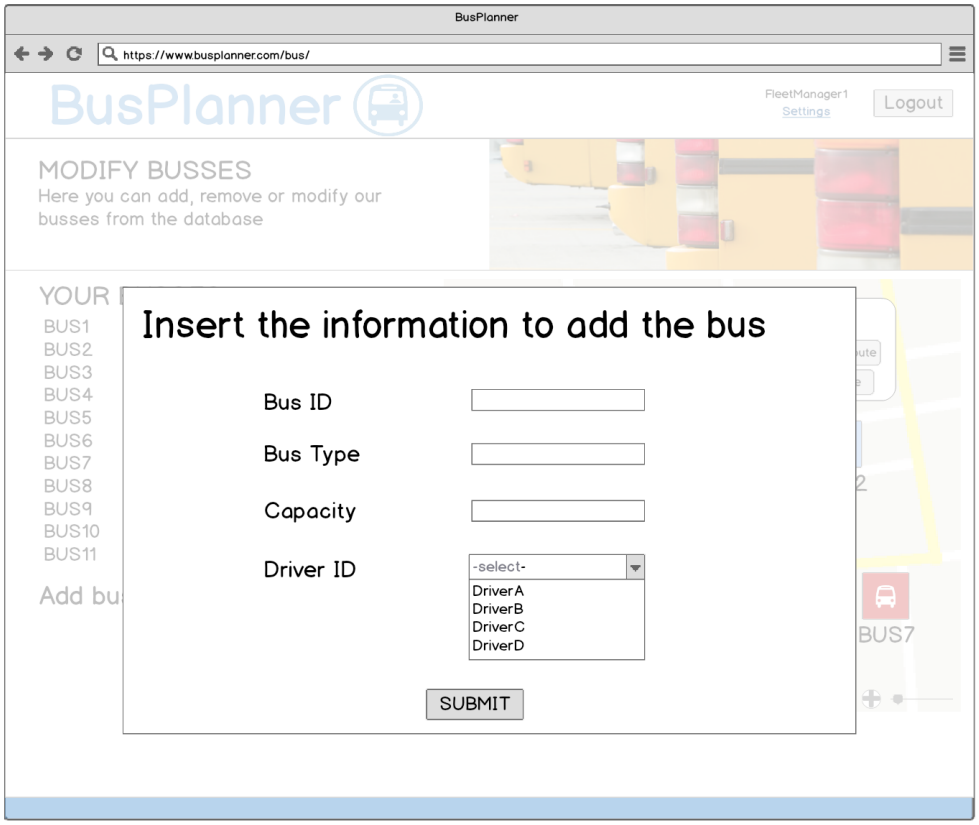
- Driver's homepage:



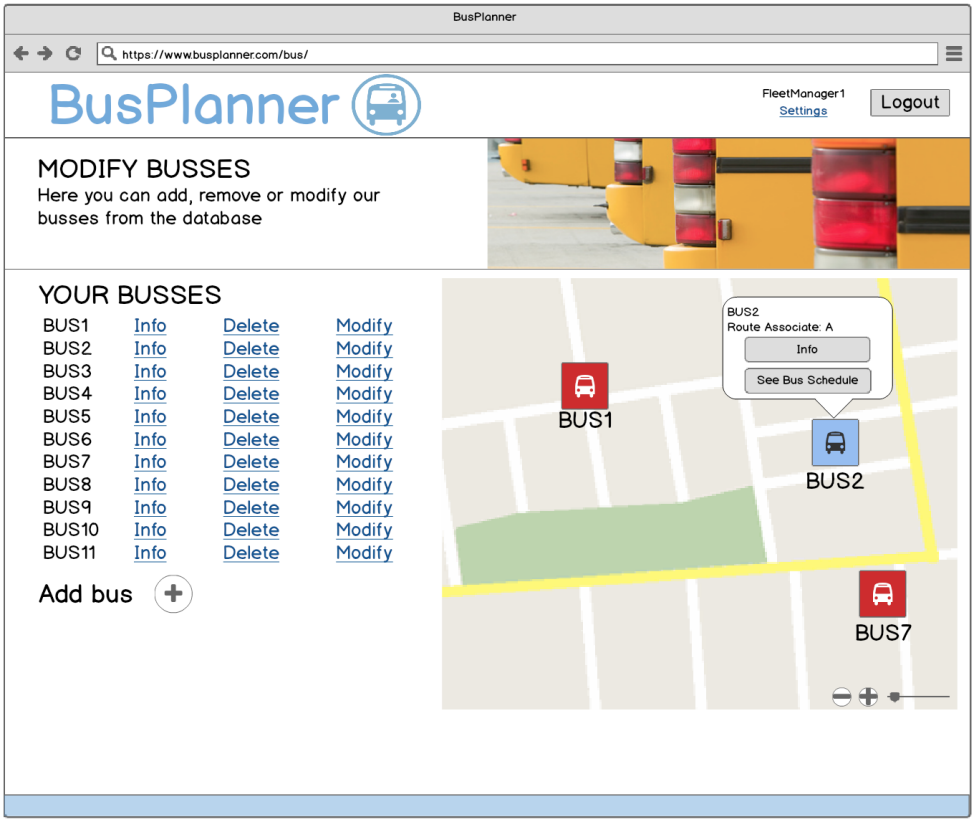
- Fleet manager's homepage:



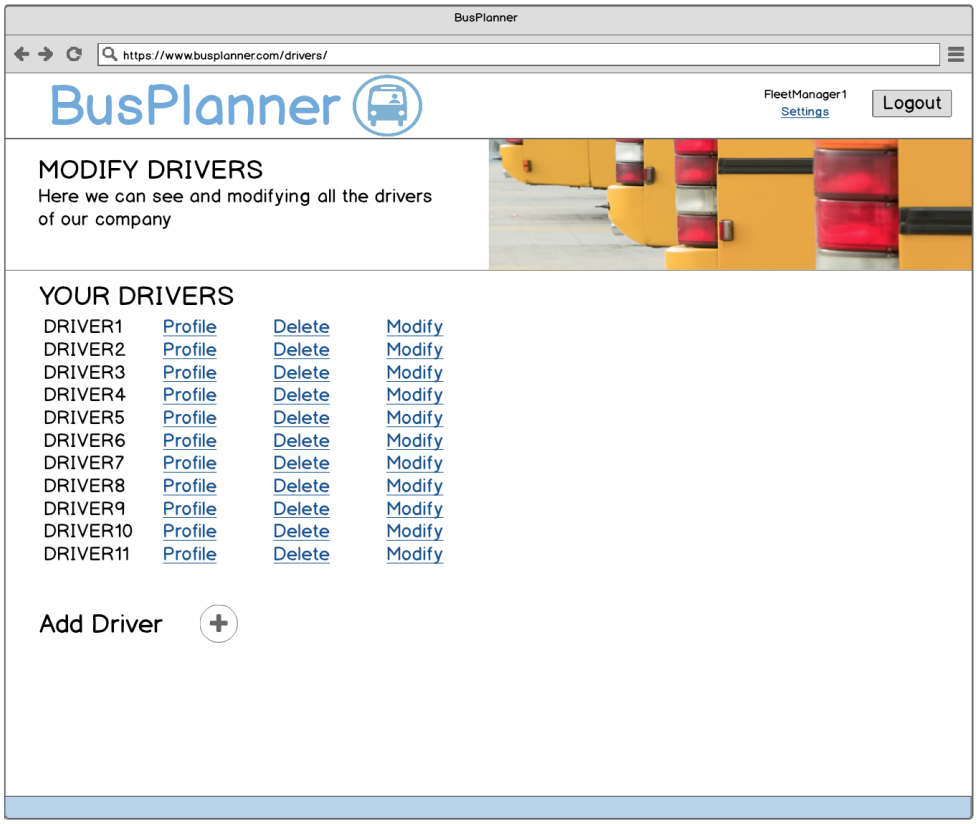
- Adding a bus:


BusPlanner
https://www.busplanner.com/bus/
BusPlanner
FleetManager1
Settings
Logout
MODIFY BUSSES
Here you can add, remove or modify our busses from the database
BUS1
BUS2
BUS3
BUS4
BUS5
BUS6
BUS7
BUS8
BUS9
BUS10
BUS11
Insert the information to add the bus
Bus ID
Bus Type
Capacity
Driver ID
-select-
DriverA
DriverB
DriverC
DriverD
SUBMIT

- Buses' list:


BusPlanner
https://www.busplanner.com/bus/
BusPlanner
FleetManager1
Settings
Logout
MODIFY BUSSES
Here you can add, remove or modify our busses from the database
YOUR BUSSES
BUS1 Info Delete Modify
BUS2 Info Delete Modify
BUS3 Info Delete Modify
BUS4 Info Delete Modify
BUS5 Info Delete Modify
BUS6 Info Delete Modify
BUS7 Info Delete Modify
BUS8 Info Delete Modify
BUS9 Info Delete Modify
BUS10 Info Delete Modify
BUS11 Info Delete Modify
Add bus
BUS2
Route Associate: A
Info
See Bus Schedule
BUS1
BUS2
BUS7

- Modify drivers:



6 DATABASE MODEL

Route

Route_id	integer	PK
Route_name	varchar	

Login

Login_id	varchar	PK
User_name	varchar	
Password	varchar	
User_type_id	integer	

UserType

User_type_id	integer	PK
User_type	varchar	
Login_id	integer	FK

UserRequest

User_id	integer	PK
Timestamp	timestamp	
Route_id	integer	FK
Latitude	float	
Longitude	float	
Status	varchar	

Bus

Bus_id	integer	PK
Bus_type	varchar	
Driver_id	integer	FK
Bus_capacity	integer	
Longitude	float	
Latitude	float	

Driver

Driver_id	integer	PK
Driver_name	varchar	
Mobile_number	varchar	

FleetManager

Manager_id	integer	PK
Manager_name	varchar	
Mobile_number	varchar	

UserSchedule

User_schedule_id	integer	PK
User_id	integer	FK
Driver_id	integer	FK
Bus_id	integer	FK
Route_id	integer	FK
Status	varchar	

RouteSchedule

Route_schedule_id	integer	PK
Start_time	timestamp	
Finish_time	timestamp	
Bus_id	integer	FK
Route_id	integer	FK