Software Documentation for Web application

**“TruckGistic”**

Produced By:

Piotr Myszkowski

6500 25th Ave NE

Seattle, WA 98115

Email: [myszon01@gmail.com](mailto:myszon01@gmail.com)

Contents

[1. Introduction 3](#_Toc515824978)

[2. Subject Area Description 4](#_Toc515824979)

[3. Description of Technologies Used 5](#_Toc515824980)

[4. Database Design 6](#_Toc515824981)

[5. Application Structure and UI Design 7](#_Toc515824982)

[6. Algorithms Design 8](#_Toc515824983)

[7. Testing 9](#_Toc515824984)

[8. Conclusion 10](#_Toc515824985)

[9. References 11](#_Toc515824986)

# Introduction

The purpose of this document is to describe the implementation of the TruckGistic web application. The TruckGistic web application software is designed to help freight management businesses focused in USA territory to track theirs loads, drivers and customers.

This document describes the implementation details of the TruckGistic Web Application. The software will consist of multiple functions. That include create/delete/update loads, customers, drivers, facilities. We will also outline and describe specific components of the project.

When designing this software it was critical to know and deeply understand logistics problems that freight companies have to face every day. This application was consultant with group of people that work in the industry to help understand the best design and solution for the problems.

This application is design for multiple companies/users. Because of the design and lightweight framework that were used it can handle thousands of users making requests with no downtime. Thanks to newest technologies and design techniques used in this project it will me much easier to maintain, troubleshoot, add new features and scale this web application.

All technologies, tools that were used are free besides server that will host database and/or web application. That is really important point since I want this application to be free for all user and deliver to them the best quality of the freight managing web application available on the market for free.

# Subject Area Description

# Description of Technologies Used

I will divide this section on four parts. In first part we will talk about back-end then in second part I will focus on front-end. Next in third part we will talk about host machine and finally in forth part I will try to create a list with all technologies I used and where are they sitting in the application. Let start with first part.

Every good application needs to be scalable and have lightweight technologies that work great with each other. That was the reason I choose **Spring Boot** **framework** as a backend. This framework of Java is great if it comes to large, enterprise applications and have many API, Documentation, support that helps implement different technologies to it. Spring Boot has no XML based configurations at all. Very much simplified properties. The beans are initialized, configured and wired automatically. Spring is also lightweight container, can deployed on tomcat or jetty, easy to implement new features if your application getting bigger and bigger.

For database connectivity I choose **Hibernate** over JDBC because JDBC is complex when is used in large projects, it is harder to implement MVC concept, there is no encapsulation. Hibernate on other side has transparent persistence that ensures automatic connection between the application’s object with the database tables. This feature prevents developers from writing lines of connection code. Transparent persistence enables hibernate to reduce the development time and maintenance cost. Hibernate supports both first level and second level caching mechanism. The first level caching is associated with Session object which is used by default. The second level caching is associated with Session Factory Object. Through caching concept, Hibernate retains the objects in cache so as to reduce repeated hits to the database. This feature makes Hibernate highly scalable and optimizes the application’s performance. Hibernate supports optimistic locking through its version property feature. This functionality supports multiple transactions without affecting one another. For example, when two or more users try to alter a database entity at the same time, the version field avoids the conflict and gives preference to the user who commits the changes first. The other user will be prompted with an error message and will be asked to restart the process.

I integrate Hibernate with **Spring Data** so I will have repositories CRUD out of box, nice and easy ways to extend your repositories with query methods, to navigate between entities.

For authentication and authorization, I used **Spring Security.** It is a powerful and highly customizable authentication and access-control framework. It is the de-facto standard for securing Spring-based applications.

As a database I choose **MySQL** because is globally renowned for being the most secure and reliable database management system used in popular web applications. MySQL offers unmatched scalability to facilitate the management of deeply embedded apps using a smaller footprint even in massive warehouses that stack terabytes of data. On-demand flexibility is the star feature of MySQL. This open source solution allows complete customization to eCommerce businesses with unique database server requirements. MySQL features a distinct storage-engine framework that facilitates system administrators to configure the MySQL database server for a flawless performance. Whether it is an eCommerce website that receives a million queries every single day or a high-speed transactional processing system, MySQL is designed to meet even the most demanding applications while ensuring optimum speed, full-text indexes and unique memory caches for enhanced performance. All the fears and worries that arise in an open source solution can be brought to an end with My SQL’s round-the-clock support and enterprise indemnification. The secure processing and trusted software of MySQL combine to provide effective transactions for large volume projects. It makes maintenance, debugging and upgrades fast and easy while enhancing the end-user experience. MySQL is considered a very fast database program. This speed has been backed up by a large number of benchmark tests (though such tests -- regardless of the source -- should be considered with a good dose of skepticism). MySQL is really easy to use with hibernate. There is nothing too complicated to have those two technologies configure and up and running in less than 1 hour. MySQL also provides with very powerful tool – MySQL Workbench – that makes your development of database super easy and enjoyable. On top of that this tool has many other useful tools like data migration, database backup, UML creator, schema generator etc.

S

# Database Design

# Application Structure and UI Design

# Algorithms Design

# Testing

# Conclusion

# References