|  |
| --- |
| **Fontys - University of Applied Sciences - ICT** |
| Individual track project: Online inventory system |
| Individual track Project Description |

|  |
| --- |
| Martin Georgiev  Student number: 3782484  Eindhoven, Tuesday, September 8, 2020 |

Document Change Record

|  |  |  |  |
| --- | --- | --- | --- |
| *Date* | *Version* | *Author* | *Comments* |
| 08-09-2020 | 1.0 | Martin Georgiev | Added the initial body structure and text of the project description/plan |
|  |  |  |  |

Table of contents

[1 Introduction 4](#_Toc50476880)

[1.1 Document Purpose 4](#_Toc50476881)

[1.2 Document Overview 4](#_Toc50476882)

[2 Project Description 5](#_Toc50476883)

[2.1 Project Overview 5](#_Toc50476884)

[2.2 Project Scope 5](#_Toc50476885)

[2.3 Project Limitations 5](#_Toc50476886)

[3 Project Requirements 6](#_Toc50476887)

[3.1 Functional Requirements 6](#_Toc50476888)

[3.2 Non-functional Requirements 6](#_Toc50476889)

[4 First domain model sketch 7](#_Toc50476890)

# Introduction

## Document Purpose

The purpose of this document is to provide a short and concise overview of the project I am planning to develop as part of the individual track. The aim is to give more information on the project topic, project scope and requirements that have been considered as first steps that will eventually lead to the final product’s completion.

## Document Overview

This document will go through all sections, considered important to all partaking stakeholders (in this case – teacher/mentor and student). Namely, these sections are:

* **Project Description:** Description of the initial concept of the project with details on planned features, the scope and possible limitations.
* **Project Requirements:** An in-depth look at the project’s requirements
* **First domain model sketch:** A general model sketch of the planned relationships and communication between core structures of the project.

# Project Description

## Project Overview

The planned product is an online inventory system for businesses that are in a need of a monitoring system to track incoming and outgoing warehouse items. The end product of this project is a web-based application that intends to combine a JavaScript-based front-end and a Java-based backend (including, but not limited to database operations) using an API to perform the communication between the two. The intended user group is separated into two roles: warehouse managers and warehouse workers.

## Project Scope

The intention is to meet all base requirements for this project:

* User authentication (login and registration functionality).
* User authorization (the ability of logged-in users to add, edit/update and delete items into/from the system).
* Retrieval of detailed/aggregated data from the database (a statistics page where users will have the ability to monitor the activity of the target warehouse).
* Search functionality using filters (the ability of logged-in users to find items using a set number of criteria).

One of the main goals is to implement a user-friendly, yet robust and secure application. To accomplish a simplistic, minimalistic and modern user interface, I plan to use a JavaScript framework. While the choice of the framework is not final, the initial plan is to use React.js as I believe it will be the most suitable one for my needs.

## Project Limitations

Two main limitations restrict the scope of the project – time and knowledge. The latter is dependent on the former. All base requirements will be prioritized during the development period. Depending on the initial progress, the scope of this project can either be extended or decreased. This will be done under the teacher/mentor’s feedback and suggestions.

# Project Requirements

Using the given information on the individual track assignment, the following project requirements have been processed and grouped when appropriate. They are divided into two main groups: functional and non-functional.

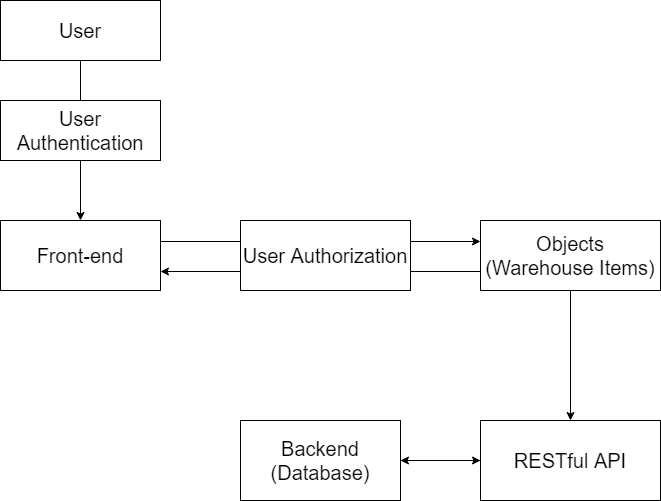
## Functional Requirements

* **User Authentication:** Users should be able to log in and out (or register) of the front-end section of the application. Access to the backend should be access restricted to only be accessible by the system.
* **User Authorization:** Onlylogged-in users should be able to perform the main CRUD functionalities (add, edit and delete items from the warehouse inventory) through the front-end part of the application. In addition to that, said users should only be able to view and manipulate the inventory of the warehouse they are assigned to.
* **Specific data retrieval from the database:** Logged-in users should be able to retrieve specific datasets from the database through the abovementioned interface.
* **Retrieval of aggregated data from the database:** Logged-in users should be able to view aggregated data in the form of a statistics page/tab.
* **Search functionality with filters:** Logged-in users should be able to view specific items through a search system with toggleable filters.

## Non-functional Requirements

* **Scalability:** The solution should work seamlessly no matter the size of the warehouse.
* **Security:** Users should not be able to access backend systems that are intended only for the system.
* **Privacy:** Users should only be able to access information, corresponding to their role.
* **Manageability:** The solution should have a user-friendly interface and a well-structured codebase for easier further work on the project.

# First domain model sketch

The following sketch aims to provide a very basic understanding of how I foresee that the relationship and communication between the user and the system (front-end and backend) will go. Retrieval of data from the backend will go through a RESTful API that will collect the data and format it appropriately for the front-end to use. While it is not present in the sketch, there is a relationship between the user authentication/authorization and the API. Furthermore, the user authentication and authorization sections are part of the front-end. However, I decided to separate them to better visualize the communication process between the core structures of the application.