|  |
| --- |
| Fontys Hogescholen |
| User Requirements Specifications |
| Project MB |

|  |
| --- |
| IHDY  Supervisor: Mieke van Vucht  Eindhoven, 2-17-2020 |

Document Change Record

|  |  |  |  |
| --- | --- | --- | --- |
| *Date* | *Version* | *Author* | *Comments* |
| 17-02-2020 | 1.0 | Hadzhikolev, Kristian | Beginning of Requirements |
| 18-02-2020 | 1.1 | Mureseanu,Gabriel | Background, Stakeholders and User Analysis |
| 19-02-2020 | 1.2 | Hadzhikolev, Kristian | Functional and Non-Functional Requirements update |
| 28-02-2020 | 1.3 | Hadzhikolev, Kristian | Detailed Use Cases |
| 02-04-2020 | 2.0 | Hadzhikolev, Kristian | Functional Requirements for the C# Application Update(Iterative Phase) |

# Definitions, Acronyms and Abbreviations

|  |  |
| --- | --- |
| *Term* | *Description* |
| MoSCoW | Stands for must, should, could and would |
|  |  |
|  |  |

Table of Contents

[Definitions, Acronyms and Abbreviations 3](#_Toc32142994)

[1 Introduction 5](#_Toc32142995)

[1.1 Document Purpose 5](#_Toc32142996)

[1.2 Document Overview 5](#_Toc32142997)

[2 Background 6](#_Toc32142998)

[2.1 Scope and Objectives 6](#_Toc32142999)

[3 Stakeholder and User Analysis 7](#_Toc32143000)

[4 User Requirements 8](#_Toc32143001)

[4.1 Functional Requirements 8](#_Toc32143002)

[4.2 Non-Functional Requirements 8](#_Toc32143003)

[5 Assumptions/Constraints 9](#_Toc32143004)

[6 Use Case Models 10](#_Toc32143005)

[7 GUI 11](#_Toc32143006)

[8 Website Wireframes (if not applicable, remove section) 12](#_Toc32143007)

# Introduction

## Document Purpose

This document is the definitive specification of the user requirements for project MB.

[Describe the purpose of the document; e.g. This document is the definitive specification of the user requirements for the “X” project. It is a primary input to the technical development and the primary specification for the acceptance criteria for those evaluating the outcome after the development has been finalized. This document is intended to be read by all responsible for the management of the development]

## Document Overview

[Give an overview of the chapters following; e.g. Section 2 provides a general description of the product(s) and the factors that affect their requirements]

# Background

Project MB is the solution to Media Bazaar’s presented problem. The solution has to be tailored to the requirements and expectations of our client, such as design, functionality , etc.

## Scope and Objectives The objective of the project is to deliver the best solution to the problems presented to us by Media Bazaar.

# Stakeholder and User Analysis

The stakeholders in this project are only the group members and the client.  
The users of the solution will be the employees at Media Bazaar.  
As the client has mentioned in the first interview, he does not expect a lot of traffic in the first weeks, so the system won’t be frequently used in that timeframe, that also means that the website won’t have a lot of orders.

# User Requirements

## Functional Requirements

|  |  |  |
| --- | --- | --- |
| *ID* | *Name* | *Priority (MoSCoW)* |
| FR-01 | Manager should be able to Login and Logout | M |
| FR-02 | Manager must be able to export information | M |
| FR-03 | Manager must be able to see statistics for employees, stock and departments | M |
| FR-04 | Depot Management should be implemented | M |
| FR-05 | Depot Manager should be able add new product | M |
| FR-06 | Depot Manager should be able remove existing product | M |
| FR-07 | Depot Manager should be able to search for specific product | M |
| FR-08 | Depot Manager should be able to update information for a product | M |
| FR-09 | Depot Manager should be able to see shelf restock requests as well as available stock(either by department or all departments at once) | M |
| FR-10 | Employee Management should be implemented | M |
| FR-11 | Employee Manager should be able to add new employees | M |
| FR-12 | Employee Manager should be able to remove employees(in case they are fired) | M |
| FR-13 | Employee Manager should be able to reassign employees | M |
| FR-14 | Employee Manager should be able to edit employee information(salary, working hours and etc.) | M |
| FR-15 | Employee Manager should be able to update schedule | M |
| FR-16 | Employee Manager must be able to assign roles(Department Manager, Depot Manager, Employee Manager or Employee) | M |
| FR-17 | Employee Manager should be able to see overview of all employee shifts | S |
| FR-18 | Employee Manager should be able to set automatically selected shift | W |
| FR-19 | Algorithm for shift replacement due to illness must be available | W |
| FR-20 | Department Management should be implemented | M |
| FR-21 | Department Manager should be able to add new department | S |
| FR-22 | Department Manager should be able to update existing department | S |
| FR-23 | Department Manager should be able to remove existing department | S |
| FR-24 | Department Manager should be able to search for existing department | S |
| FR-25 | Department Manager should be able to see all departments | S |
| FR-26 | Employee should be able to access the website to change various personal information | M |
| FR-27 | Employee should be able to view their own schedule in the website | M |
| FR-28 | Employee should be able to login and register in the website | M |

## Non-Functional Requirements

[All non-functional requirements]

|  |  |  |
| --- | --- | --- |
| *ID* | *Name* | *Priority (MoSCoW)* |
| N-FR-01 | Site/Program should be updated often | M |
| N-FR-02 | Site/Program should protect manager data | M |
| N-FR-03 | Information should be backed up daily | M |
| N-FR-04 | Site/Program should sustain high traffic | S |
| N-FR-05 | Site/Program should be responsive | M |

# Assumptions/Constraints

## At the beginning there won’t be much traffic in the software/website. However, it will increase in the future.

## The software solution and the website for the employees will be highly reliable.

## It will work fast enough to be effective.

# Use Case Models

1. UC: Manager login

|  |  |
| --- | --- |
| Use case: | Manager should be able to login |
| Actor: | Manager |
| Pre-condition: | Manager should not be already logged in |
| Trigger: | Login button is clicked |
| Main Success Scenario: | 1. Fields for username and password are entered by the manager 2. Submit is clicked 3. System establishes connection to the database 4. System checks if there is such username 5. System checks if the passwords matches 6. Manager is logged in |
| Extensions: | 1a: field(s) are empty  .1 System displays a not all fields are completed message  .2End of use case  3a: Can’t connect to database  .1 System tries to connect again, if not unsuccessful connection message displayed  .2 End of use case  4a Username does not exist  .1 Not existing username message is displayed  .2 End of use case  5a Wrong password  .1 Wrong password message is displayed  .2 End of use case |

1. UC: Manager register

|  |  |
| --- | --- |
| Use case: | Manager should be able to register |
| Actor: | Manager |
| Pre-condition: | Manager should not be already logged in |
| Trigger: | Register button is clicked |
| Main Success Scenario: | 1. Manager enters details such as name, email, department and etc. 2. The information is submitted 3. System connects to the database 4. System checks if the user is already registered 5. System adds user to database |
| Extensions: | 2a Missing/Incorrect Information  .1 System displays error message  .2 End of use case  3a: Can’t connect to database  .1 System tries to connect again, if not unsuccessful connection message displayed  .2 End of use case  4a Already Registered user  .1 System displays error message  .2 End of use case |

1. UC: Setting/Changing Product information

|  |  |
| --- | --- |
| Use case: | Manager should be able to set/change information for a product |
| Actor: | Manager |
| Pre-condition: | Manager logged in and is in the right department |
| Main Success Scenario: | 1. Manager selects a certain product 2. Manager clicks on edit button 3. Manager edits information for the selected product 4. Manager submits the edited information 5. System connects to the database 6. System saves the new information |
| Extensions: | 2a: No product is selected  .1 System displays an error message  .2 End of use case  4a: No information is changed  .1 System displays an error message  .2 End of use case  4b: Invalid information  .1 System displays an error message  .2 End of Use Case  5a: Can’t connect to database  .1 System tries to connect again, if not unsuccessful connection message displayed  .2 End of use case |

1. UC: Add Employee

|  |  |
| --- | --- |
| Use case: | Manager should be able to add new employees |
| Actor: | Manager |
| Pre-condition: | Logged in as Manager |
| Trigger: | Add Employee button is clicked |
| Main Success Scenario: | 1. Manager adds the required information 2. Manager clicks confirm button 3. System connects to database 4. System adds the information to the database |
| Extensions: | 2a: Empty fields  .1 System displays an error message  .2 End of use case  2b: Invalid information  .1 System displays an error message  .2 End of Use Case  3a: Can’t connect to database  .1 System tries to connect again, if not unsuccessful connection message displayed  .2 End of use case |

1. UC: View Employee Information

|  |  |
| --- | --- |
| Use case: | Manager should be able to view employee information(salary, working hours and etc.) |
| Actor: | Manager |
| Pre-condition: | Logged in as Manager |
| Trigger: | Employee selected from the list |
| Main Success Scenario: | 1. System connects to database 2. System extracts information from database 3. Information is displayed |
| Extensions: | 1a: Can’t connect to database  .1 System tries to connect again, if not unsuccessful connection message displayed  .2 End of use case |

1. UC: Update Employee Information

|  |  |
| --- | --- |
| Use case: | Manager should be able to update employee information(salary, working hours and etc.) |
| Actor: | Manager |
| Pre-condition: | Logged in as Manager |
| Trigger: | Employee is selected and update button is clicked |
| Main Success Scenario: | 1. Manager changes information 2. Manager clicks confirm button 3. System connects to database 4. System updates the information in database |
| Extensions: | 2a: No information is changed  .1 System displays an error message  .2 End of use case  2b: Invalid information  .1 System displays an error message  .2 End of Use Case  3a: Can’t connect to database  .1 System tries to connect again, if not unsuccessful connection message displayed  .2 End of use case |

1. UC: Remove Employee

|  |  |
| --- | --- |
| Use case: | Manager should be able to remove employees(in case they are fired) |
| Actor: | Manager |
| Pre-condition: | Logged in as Manager |
| Trigger: | Employee is selected and remove button is clicked |
| Main Success Scenario: | 1. System connects to database 2. System extracts employee information 3. System removes the information about the employee |
| Extensions: | 1a: Can’t connect to database  .1 System tries to connect again, if not unsuccessful connection message displayed  .2 End of use case  2a: Wrong Department  .1 System displays an error message  .2 End of use case  2b: Another Manager or person in higher position  .1 System displays an error message  .2 End of Use Case |

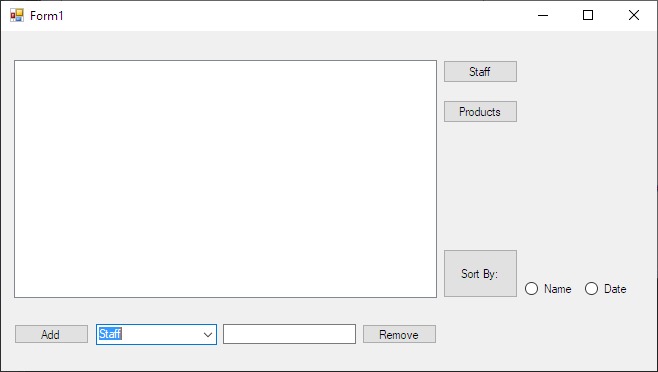
1. UC: Employee Access to Website

|  |  |
| --- | --- |
| Use case: | Employee should be able to access the website to change various personal information |
| Actor: | Employee |
| Pre-condition: | Logged in as Employee |
| Main Success Scenario: | 1. System connects to the database 2. System extracts the required information 3. System displays the information 4. Employee edits the information 5. Employee clicks confirm button 6. System adds the new information to the database |
| Extensions: | 1a: Can’t connect to database  .1 System tries to connect again, if not unsuccessful connection message displayed  .2 End of use case  5a: Invalid information  .1 System displays an error message  .2 End of use case  5b: No changes have been made  .1 System displays an error message  .2 End of Use Case |

1. UC: Last Depot Workers

|  |  |
| --- | --- |
| Use case: | Last depot workers should be able to see restock requests as well as available stock and update them |
| Actor: | Last depot worker |
| Pre-condition: | Logged in as last depot worker |
| Trigger: | Inventory or request button is clicked |
| Main Success Scenario: | 1. System connects to database 2. System extracts the required information 3. System displays the information 4. Last depot worker changes stock information 5. Last depot worker clicks confirm button 6. System removes requests that has been completed 7. System updates information in database |
| Extensions: | 1a: Can’t connect to database  .1 System tries to connect again, if not unsuccessful connection message displayed  .2 End of use case  5a: Invalid Information  .1 System displays an error message  .2 End of use case |

# GUI



# Website Wireframes

