|  |
| --- |
| Fontys Hogescholen |
| User Requirements Specifications |
| Media Bazaar Project |

|  |
| --- |
| IBI Student Group  Supervisor: Roopali Gupta  Eindhoven, 19.02.2020 |

Document Change Record

|  |  |  |  |
| --- | --- | --- | --- |
| *Date* | *Version* | *Author* | *Comments* |
| 19.02.2020 | 1.0 | Kaloyan Aleksiev |  |
| 15.04.2020 | 2.0 | Kaloyan Aleksiev |  |

# Definitions, Acronyms and Abbreviations

|  |  |
| --- | --- |
| *Term* | *Description* |
| GUI | Graphic User Interface |
| Interface | Screen for user interaction |
|  |  |

Table of Contents

[Definitions, Acronyms and Abbreviations 3](#_Toc34640469)

[1 Introduction 5](#_Toc34640470)

[1.1 Document Purpose 5](#_Toc34640471)

[1.2 Document Overview 5](#_Toc34640472)

[2 Background 6](#_Toc34640473)

[2.1 Scope and Objectives 6](#_Toc34640474)

[3 Stakeholder and User Analysis 7](#_Toc34640475)

[4 User Requirements 8](#_Toc34640476)

[4.1 Functional Requirements 8](#_Toc34640477)

[4.2 Non-Functional Requirements 8](#_Toc34640478)

[5 Use Case Models 9](#_Toc34640479)

[6 GUI 12](#_Toc34640480)

[7 Website Wireframes (if not applicable, remove section) 13](#_Toc34640481)

# Introduction

## Document Purpose

This document is the definitive specification of the user requirements for the Media Bazaar 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

Section 2 provides a general description of the product and the factors that affect their requirements, including the objectives.

# Background

Media Bazaaris a new hardware store funded by the parent company Jupiter**.** They are opening a new shop in Eindhoven and want to be as prepared as possible when it comes to keeping track of management and stocks. Because the administration is the backbone of that manages all aspects of the company, management would like to hire a team of software developers to implement an administrative system.

## Scope and Objectives

The scope of the project includes developing a complete system which consists of a database, a website and a C# application, all of which should work together in order to let the client manage their stocks and employees.

# Stakeholder and User Analysis

The stakeholders in our project are the following: The parent company “Jupiter”, the store “Mediabazaar”, the IBI student group. The solution is intended to be used internally within “Mediabazaar” by the Administration and Management departments as well as the Depo-workers and the employees that work in the store. Each of the users will have access to a specific part of the application while everyone will be able to log into the website. The most frequent users would be the employees in the shop and the depo workers. That is because they are the ones that would need to check for stock availability and send restock requests. The Management would also use the application quite frequently in order to schedule employee’s shifts for each month and to resolve problems when it comes to an employee being unavailable or sick. Administration would check the statistics of the store, separate employees and stock a couple of times per month, which means they would use the system the least.

# User Requirements

## Functional Requirements

[All functional requirements; if multiple stand-alone apps, either desktop or web, make sure to separate the requirements per app

Make sure to order them on priority]

|  |  |  |
| --- | --- | --- |
| *ID* | *Name* | *Priority (MoSCoW)* |
| FR-01 | Adding employees to the system. | M |
| Assigning employees to different departments. |
| Removing employees from the system. |
| FR-02 | Schedule shifts for each employee. | M |
| FR-03 | Change employee information. | S |
| FR-04 | View statistics. | M |
| View employee information. |
| View stock information. |
| FR-05 | Send restock requests. | S |
| FR-06 | Have 4 different types of User Entities | C |
|  |  |  |

|  |  |  |
| --- | --- | --- |
| *ID* | *Name* | *Priority (MoSCoW)* |
| FR-01 | Everyone must be able to login into the website. | C |
| FR-02 | Everyone must be able to view their shifts. | C |
| FR-03 | Everyone must be able to change personal info. | S |
| FR-04 | Attendance could be recorded via the website. | S |
| FR-05 | People could switch shifts | C |
| FR-06 | People should be able to call in sick or unavailable through the website | C |
| FR-07 | Administrators should be able to add new departments | W |

## Non-Functional Requirements

[All non-functional requirements]

|  |  |  |
| --- | --- | --- |
| *ID* | *Name* | *Priority (MoSCoW)* |
| N-FR-01 |  |  |
| N-FR-02 |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

# Use Case Models

1. UC: View stock

|  |  |
| --- | --- |
| Use case: | View stock |
| Actor: | Depot workers |
| Pre-condition: | Depot worker must be logged in |
| Trigger: | Press ‘View Stock’ Button |
| Main Success Scenario: | 1. Worker presses the ‘View Stock’ button  2. The system returns a table with the whole stock |
| Extensions: | 1a: Worker needs to look for a specific item  .1: Worker uses the search bar to input the name of the product  .2: System displays the information about the product  2a: No such product  .1: System returns an error |

2. UC: Make restock requests

|  |  |
| --- | --- |
| Use case: | Make restock requests |
| Actor: | Depot workers |
| Pre-condition: | Depot worker must be logged in, an item needs to be low on stock |
| Trigger: | Press ‘Make Stock Request’ Button |
| Main Success Scenario: | 1. Worker inputs the ID of the item that needs to be restocked  2. Worker presses the ‘Make Stock Request’ button  3. The request gets sent to the database |

3. UC: View restock requests

|  |  |
| --- | --- |
| Use case: | View restock requests |
| Actor: | Administrator |
| Pre-condition: | Admin needs to be logged in, a stock request needs to have been made |
| Trigger: | Press ‘See restock requests’ Button |
| Main Success Scenario: | 1. Administrator presses the ‘See restock requests’ button  2. The database returns a list of all restock requests |
| Extensions: | 1a: Admin needs to look for a specific request  .1: Admin inputs the request ID in the search bar  .2: System displays the information about the request  2a: No such request  .1: System returns an error |

4. UC: Add/remove employees

|  |  |
| --- | --- |
| Use case: | Add/remove employees |
| Actor: | Administrator |
| Pre-condition: | Admin needs to be logged in |
| Trigger: | Press ‘Add employee’ / ‘Remove employee’ button |
| Main Success Scenario: | 1. Admin presses the respective button  2. Employee is added/removed from the system |
| Extensions: | 1a: Removing an employee by ID  .1: Admin inputs ID, presses the button, employee is removed |

5. UC: Assign employee to department

|  |  |
| --- | --- |
| Use case: | Assign employee to department |
| Actor: | Administrator |
| Pre-condition: | Admin is logged in, employee exists in the system |
| Trigger: | Press ‘Assign employee’ button |
| Main Success Scenario: | 1. Admin inputs the ID of the desired employee  2. Admin selects a department from a dropbox  3. Admin presses the ‘Assign employee’ button  4. Employee is assigned to desired department |

6. UC: See employee department, salary and position

|  |  |
| --- | --- |
| Use case: | See employee department, salary and position |
| Actor: | Administrator, Employee |
| Pre-condition: | Admin/Manager is logged in, employee is assigned |
| Trigger: | Press ‘See employee information’ Button |
| Main Success Scenario: | 1. Admin/Manager inputs the ID of the desired employee  2. Admin/Manager presses the ‘See employee information’ button  3. System returns information about the employee |

7. UC: Change personal info

|  |  |
| --- | --- |
| Use case: | Change personal info |
| Actor: | Employee |
| Pre-condition: | Employee is logged in and assigned |
| Trigger: | Press ‘Update personal information’ button |
| Main Success Scenario: | 1. Employee fills in the correct details in the textboxes  2. Employee presses the ‘Update personal information’ button  3. The info is sent to the database and updated |

8. UC: Call in sick

|  |  |
| --- | --- |
| Use case: | Call in sick |
| Actor: | Employee |
| Pre-condition: | Employee is logged in and has a shift assigned to them |
| Trigger: | Press ‘Sick leave’ button |
| Main Success Scenario: | 1. Employee selects the shift they want to skip due to being sick  2. Employee presses the ‘Sick leave’ button  3. Shift is marked as ‘Not taken’  4. Employee’s Sick Leave is registered in the system |

9. UC: See statistics

|  |  |
| --- | --- |
| Use case: | See statistics |
| Actor: | Manager |
| Pre-condition: | Manager must be logged in |
| Trigger: | Press “Statistics button” |
| Main Success Scenario: | 1. Manager inputs a name and selects department and clicks “View statistics”  2. The system returns info about the selected employee |
| Extensions: | 1a: No such employee  .1: System returns an error  1b: Manager selects a department and clicks “View department statistics”  .1: System returns information about the selected department  1c: Manager clicks the “View statistics overall button”  .1: System returns information about the whole store |

# GUI

Website

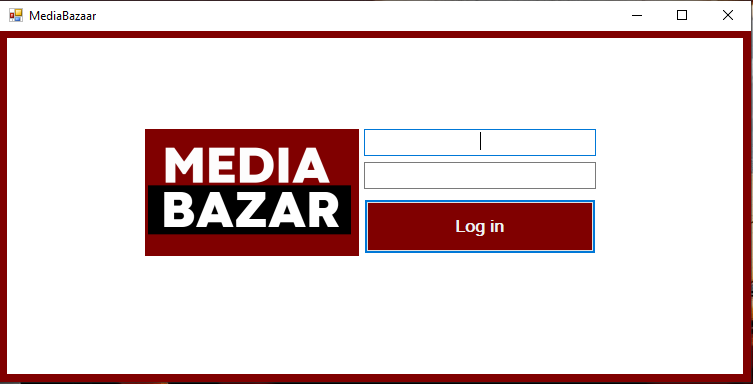
A picture containing drawing

Description automatically generated

A screenshot of a cell phone

Description automatically generated

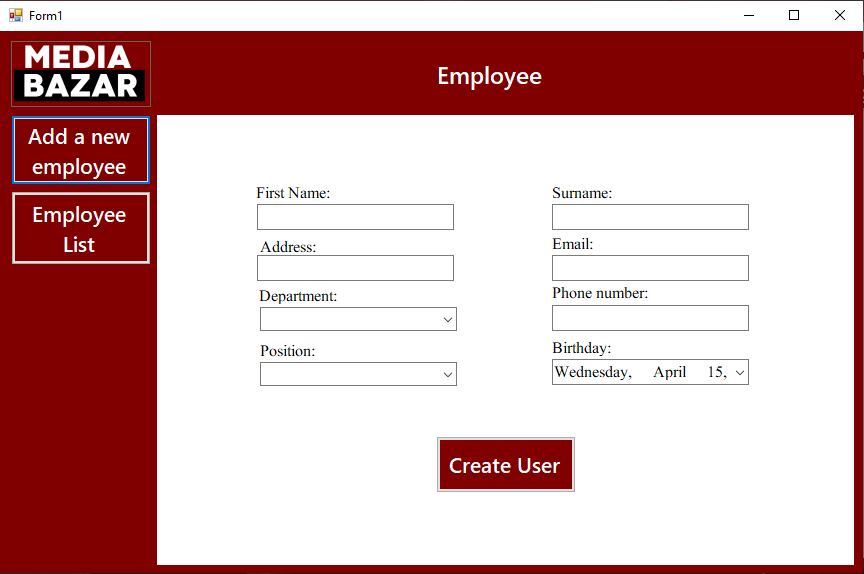
App Login



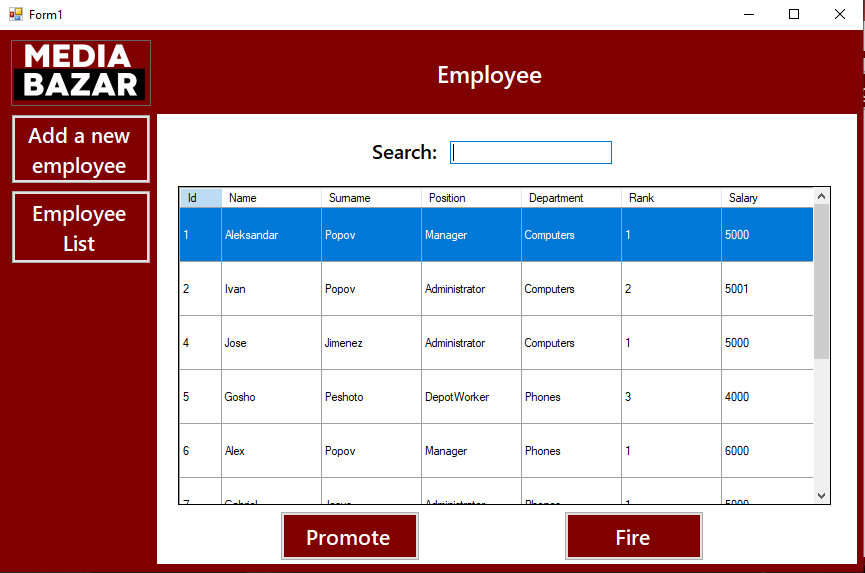
App Home Screen



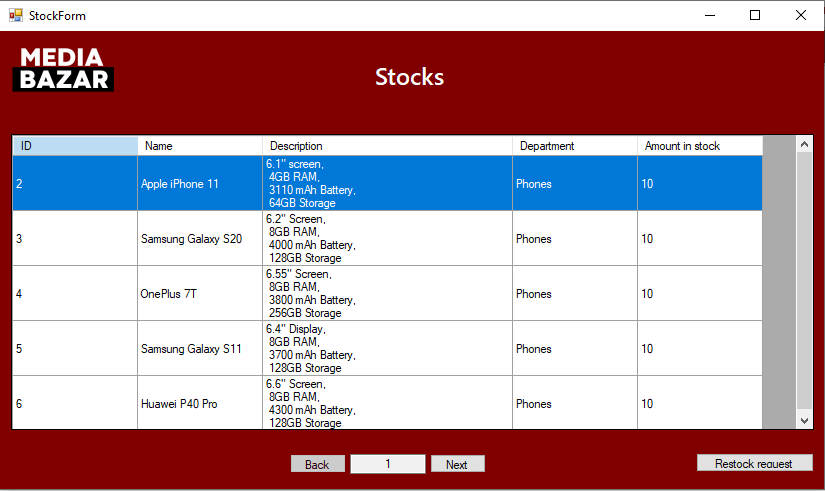
Adding an Employee



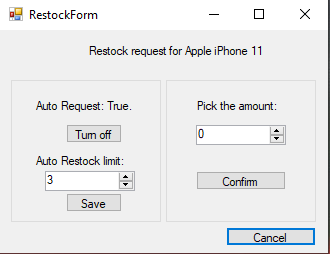
Employee List



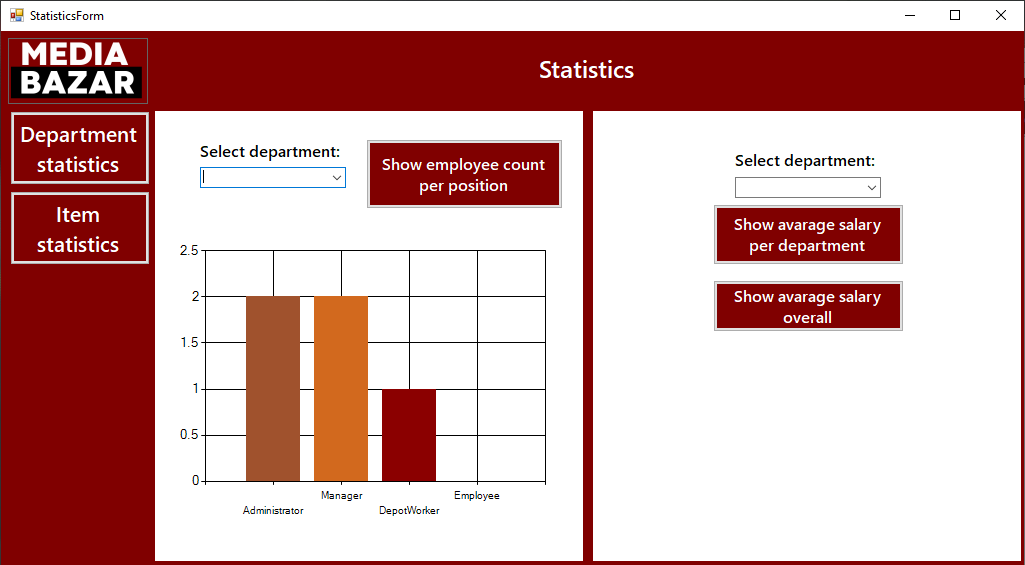
List of all the items in stock



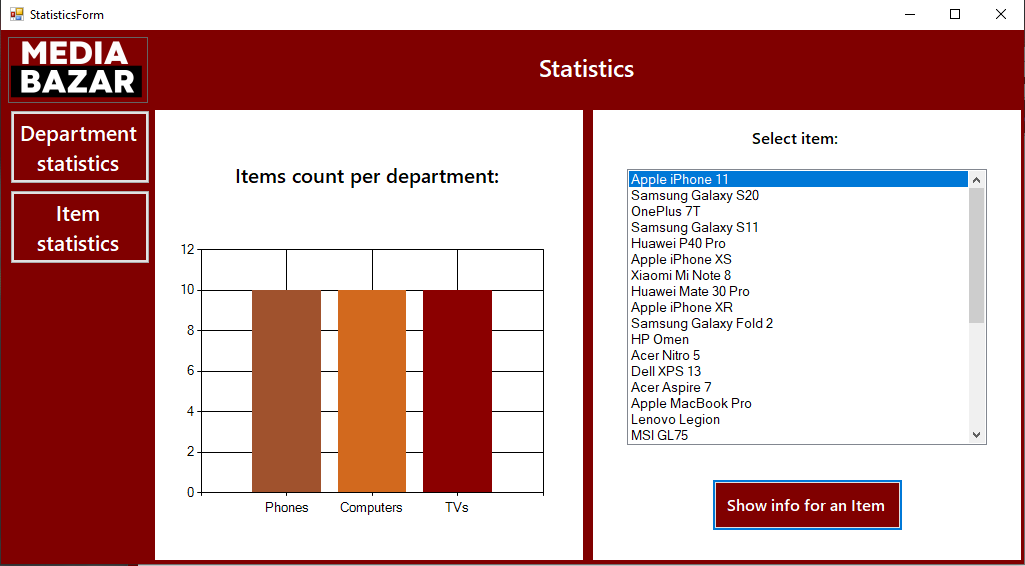
Restock Request Form



Department Statistics



Item Statistics



# Website Wireframes

Link to the wireframes in Adobe XD: <https://xd.adobe.com/view/63f1bf0a-430c-4ad4-761a-52c3852c2cb0-15c0/>