

horizontal line

|  |
| --- |
| Fontys Hogescholen |
| URS KILR TECH |
|  |
|  |

|  |
| --- |
|  |

1 **Introduction**

1.1 Document Purpose

The purpose of this document is to contain all of the user requirements and to list all of them in a consumable way for the client.

1.2 Document Overview

Section №1: Introduces the purpose and the overview of the document.

Section №2: Provides a background and gives the scopes and objectives of the project.

Section №3: Gives more detailed information about the stakeholders and user analysis.

Section №4: Lists the user requirements in MOSCOW

Section №5: A look into the assumptions and constrains that might be faced.

Section №6: The use case models describing every possible interaction of the application.

Section №7: The design of the application.

2 **Background**

Our application will help the company “Jupiter” with keeping track of stock and the handling of the company’s employees. With our application the managers that are part of the company will be able to assign shifts and manage all the employees, including hiring, firing and adjusting wages.

2.1 Scope and Objectives

We will create an application that performs all requirements mentioned in the MOSCOW table. The objective is to satisfy the client’s needs with the app. The scope of this project will provide the hiring company “Jupiter” with an application that will fix all of the managing problems that they are experiencing. It will consists.

3 **Stakeholders and User Analysis**

*Formal Client*

The formal client is Mieke van Vucht. She has requested that the group comes up with a software solution for successfully managing her employees and stock.

*Tutor*

The tutor of the group is Mrs. Roopali Gupta. She is the person that follows the team’s progress and deadlines.

*Project Leader*

Radoslav Karaganchev (team representative)

*Team*

Ivan Marinchev (technical role)

Kristian Lachev (technical role)

Lukas Rimavičius (technical role)

4 **User requirements (MOSCOW)**

**Must:**

* Employee management - CRUD
* Roles of users (manager, depot worker, employee)

**Should:**

* Department management
* Statistics page - CRUD
* Stock management

**Could:**

* Website for employees to view and edit their details

**Won’t:**

None

5 **Assumptions and constraints**

* Hardware and technology constraints

We lack the hardware that would make work faster.

* Time/Deadline constraint

We must complete the project in a certain amount of time.

1. **Use Case Models**

**Employee Administration**

UC: Viewing info for Employee

|  |  |
| --- | --- |
| Use case: | Viewing the employee’s information |
| Actor: | Administrator |
| Trigger: | Clicking the “Employee information” button |
| Main Success Scenario: | 1. System requests a an employee number 2. User provides the system with the employee number. 3. System searches, finds the employee and opens a window with all the information about the employee. |
| Extensions: | 1a. The User does not exist  -System displays a message.  -End of Use Case.  2a. User leaves the field empty  -System displays a message.  -End of Use Case. |

|  |  |
| --- | --- |
| Use case: | Add or Update Employee |
| Actor: | Administrator |
| Trigger: | Clicking the “Add/Update Employee” button |
| Main Success Scenario: | 1. System requests – First Name, Last Name, Department, Position, Adress, Email and Shift 2. User fills in the required information 3. System confirms that the information is correct and adds/updates the employee. |
| Extensions: | 1a. The employee already exists  -System displays a message  -End of Use Case.  2a. The Information given is not correct  -System displays an error message  -End of Use Case.  3a. The User leaves an empty field.  -System displays a message.  -End of Use Case. |

Removing an Employee

|  |  |
| --- | --- |
| Use case: | Removing an Employee |
| Actor: | Administrator |
| Trigger: | Clicking the “Remove Employee” button |
| Main Success Scenario: | 1. System requests a an employee number 2. User provides the system with the employee number. 3. System searches, finds the employee and dislays a “Confirm” Message. 4. Users click on “Confirm” message. 5. System removes the employee. |
| Extensions: | 1a. The User does not exist  -System displays a message.  -End of Use Case.  2a. User leaves the field empty  -System displays a message.  -End of Use Case. |

Add an Employee

|  |  |
| --- | --- |
| Use case: | Add or Update Employee |
| Actor: | Administrator |
| Trigger: | Clicking the “Add/Update Employee” button |
| Main Success Scenario: | 1. System requests – First Name, Last Name, Department, Position, Adress, Email and Shift 2. User fills in the required information 3. System confirms that the information is correct and adds/updates the employee. |
| Extensions: | 1a. The employee already exists  -System displays a message  -End of Use Case.  2a. The Information given is not correct  -System displays an error message  -End of Use Case.  3a. The User leaves an empty field.  -System displays a message.  -End of Use Case. |

Update an Employee

|  |  |
| --- | --- |
| Use case: | Update an employee’s details |
| Actor: | Administrator |
| Trigger: | Clicking the “Update Employee details” button |
| Main Success Scenario: | System requests an employee number  User provides the system with an employee number  System searches, finds and displays all the details about the employee in the corresponding fields.  User clicks on the “Add/Update” button.  System Updates the details of the employee. |
| Extensions: | 1a. The User does not exist  -System displays a message.  -End of Use Case.  2a. User leaves the field empty  -System displays a message.  -End of Use Case. |

Logging in.

|  |  |
| --- | --- |
| Use case: | Logging in. |
| Actor: | User |
| Trigger: | Adding email to textbox and clicking the “login” button |
| Main Success Scenario: | 1. System requests email 2. User provides a valid email 3. User provides a valid password 4. System confirms email 5. System confirms password 6. System allows user to view past login screen |
| Extensions: | 1a. The User does not exist  -System displays a message.  -End of Use Case.  2a. User leaves the field empty  -System displays a message.  -End of Use Case. |

**Handling of Stock**

Adding stock

|  |  |
| --- | --- |
| Use case: | Adding stock |
| Actor: | Administrator |
| Trigger: | Pressing the “Add” button the in Stock tab |
| Main Success Scenario: | 1. The system requires the fields for Name, Maximum Stock and Minimum Stock to be filled. 2. The administrator fills the required information. 3. The new stock is added to the system. |
| Extensions: | 1a. Some of the required information is missing.   * The system returns an appropriate message. * End of use case   2a. The stock already exists   * The system returns an appropriate message. * End of use case |

Marking stock as active/inactive

|  |  |
| --- | --- |
| Use case: | Setting stock as active/inactive |
| Actor: | Administrator |
| Trigger: | Pressing the “Active/Inactive Stock” button |
| Main Success Scenario: | 1. The system requires the ID of the stock. 2. The administrator inputs the ID. 3. The system looks for the specific stock. 4. The stock is set as Active/Inactive. |
| Extensions: | 1a. The ID of the stock is incorrect.   * The system returns an appropriate message. * End of use case. |

Changing Stock

|  |  |
| --- | --- |
| Use case: | Changing Stock |
| Actor: | Administrator |
| Trigger: | Pressing the “Change Stock” button |
| Main Success Scenario: | 1. The system requires the ID of the stock. 2. The administrator inputs the ID. 3. The system looks for the specific stock. 4. A different form opens where the administrator can change the details of the stock . |
| Extensions: | 1a. The ID of the stock is incorrect.   * The system returns an appropriate message. * End of use case. |

Accessing a certain stock’s information

|  |  |
| --- | --- |
| Use case: | Accessing stock information |
| Actor: | Administrator |
| Trigger: | Clicking the “Stock Information” button. |
| Main Success Scenario: | 1. The system requires the ID of the stock. 2. The administrator inputs the ID. 3. The system looks for the specific stock. 4. A new form opens with the all information about the stock. |
| Extensions: | 1a. The ID of the stock is incorrect.   * The system returns an appropriate message. * End of use case. |

**Department**

Removing a department

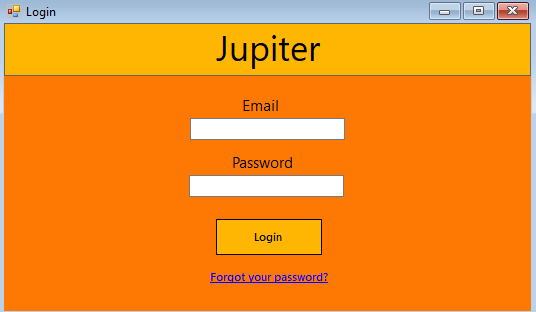
|  |  |
| --- | --- |
| Use case: | Removing a department |
| Actor: | Administrator |
| Trigger: | Pressing the “Remove Department” button |
| Main Success Scenario: | 1. The system requires the name of the department. 2. The administrator provides the name of the department 3. The system looks through the datebase for a department with the same name as inputed in the required field. 4. The suggested department is deleted. |
| Extensions: | 1a. The department with the inputted name does not exist   1. The system returns an appropriate message. 2. End of use case.   2a. The department still has employees assigned to it  1. The system returns an error message.  2. End of use case. |

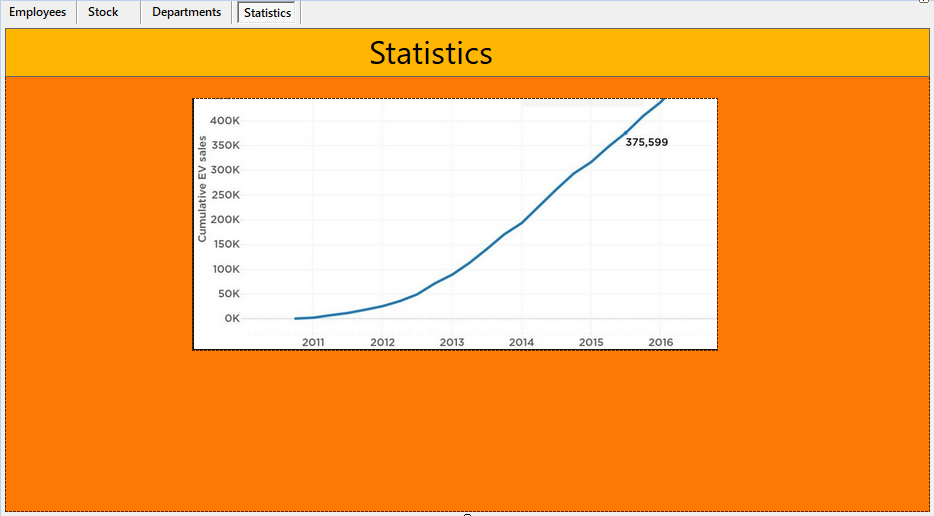
Updating the information of a department

|  |  |
| --- | --- |
| Use case: | Updating the information of a certain department |
| Actor: | Administrator |
| Trigger: | Pressing the “Update Department Details” button |
| Main Success Scenario: | 1. The system requires the name of the department. 2. The administrator provides the name of the department 3. The system looks through the datebase for a department with the same name as inputed in the required field. 4. A second form opens with the corresponding fields regarding the details of the department. |
| Extensions: | 1a. The department with the inputted name does not exist   * The system returns an appropriate message. * End of use case. |

Accessing a certain department’s information

|  |  |
| --- | --- |
| Use case: | Accessing department information |
| Actor: | Administrator |
| Trigger: | Pressing the “Department Information” button |
| Main Success Scenario: | 1. The system requires the name of the department. 2. The administrator provides the name of the department 3. The system looks through the datebase for a department with the same name as inputed in the required field. 4. A second form is opened with all the details of the particular department. |
| Extensions: | 1a. The department with the inputted name does not exist   * The system returns an appropriate message. * End of use case. |

7 **Main** **GUI**



Картина, която съдържа екранна снимка

Описанието е генерирано автоматично

