**USER REQUIREMENTS SPECIFICATIONS**

**PRJ-CB04**

GROUP 3



23rd February 2021

**Members :**

Joran van de Moosdjik (400000) - [j.vandemoosdijk@student.fontys.nl](file:///C:\Users\shanessa\OneDrive%20-%20Office%20365%20Fontys\Semester%202%20-%20Software\PRJ\s-cb04-s2-g3-prj\j.vandemoosdijk@student.fontys.nl)

Jordy Rutjens (3946975) - [j.rutjens@student.fontys.nl](file:///C:\Users\shanessa\OneDrive%20-%20Office%20365%20Fontys\Semester%202%20-%20Software\PRJ\s-cb04-s2-g3-prj\j.rutjens@student.fontys.nl)

Shanessa Kostaman (450080) - [genoveva.shanessakostaman@student.fontys.nl](file:///C:\Users\shanessa\OneDrive%20-%20Office%20365%20Fontys\Semester%202%20-%20Software\PRJ\s-cb04-s2-g3-prj\genoveva.shanessakostaman@student.fontys.nl)

Tobias Agyasta Halomoan (4252861) - [t.halomoan@student.fontys.nl](file:///C:\Users\shanessa\OneDrive%20-%20Office%20365%20Fontys\Semester%202%20-%20Software\PRJ\s-cb04-s2-g3-prj\t.halomoan@student.fontys.nl)

**Tutors :**

Kalina Petrova (as a mentor) - [k.petrova@fontys.nl](file:///C:\Users\shanessa\OneDrive%20-%20Office%20365%20Fontys\Semester%202%20-%20Software\PRJ\s-cb04-s2-g3-prj\k.petrova@fontys.nl)

Frank de Lepper (as a client) - [f.delepper@fontys.nl](file:///C:\Users\shanessa\OneDrive%20-%20Office%20365%20Fontys\Semester%202%20-%20Software\PRJ\s-cb04-s2-g3-prj\f.delepper@fontys.nl)

**Fontys University of Applied Sciences**

**Eindhoven – Netherland**

**2020-2021**

Table of Contents

[**Agreements with Client** 3](#_Toc65227606)

[**End Users** 4](#_Toc65227607)

[**Functional Requirements** 5](#_Toc65227608)

[**Current User Cases** 6](#_Toc65227609)

[**GUI** 10](#_Toc65227610)

[**Website Wireframe** 11](#_Toc65227611)

# **Agreements with Client**

This section constitute the agreements made between the client (Mr.Frank de Lepper) and the developers regarding the end-products that will be delivered during the course of development. All these were agreed upon and documented during the first meeting with the client on Wednesday, 10th February 2021.

The client agrees that such items will be provided :

Documentations

* GUI Wireframe

Schematics of the user interface for the application will be provided to the client for input and clarity, keeping in mind that these will be subject to change with further developments of the application. All changes will be reported and documented.

* Meeting agenda and minutes

To the client’s request, meeting agendas and minutes may be shared as to receive updates of the development’s progress and milestones.

Deliverables

* Desktop Application

An executable program shall be delivered with the functionalities and preferences agreed upon with the client.

* Web Application

A working website application will be delivered for the client.

* Database

The applications will be provided with database implementation that ensures easy and safe storing and updating of data

* Early Iterations of Application

Each iteration of the application during development, corresponding with each completed milestone, will be delivered throughout the process for input and clarity.

This application is made for people working on the depot, receiving goods from suppliers, stock the products, receive request from sales reps to move product to shop floor.

There’re few section of department. People who working on the shop department is sales reps, cashiers, customer support. People who working to manage the stores is HR, shift managers.

On this project the highest priority is the app to store data on keeping track of employee work hours, making sure employee information correct (first name, last name, account name, phone number, address, email address, relationship status, DOB, BSN, roles in the company), assigning shifts to employee (shift managers). Beside that we also need to make an app to keeping track of products (Warehouse Manager).

Based on client respond, financial information is not needed. It’s not priority at this moment. But maybe at the future they would want an integrated application for the product.

Every products should be categorize by the department. Televisions, audio sets, kitchen appliances, car audio, computers it’s goes to shop departments. In every department there’s employees are assigned for each department in the future.

The application is protected with the password and all the info about employee already confirmed by HR.

# **End Users**

As per the client agreements, the end users that serve as the subject of the application consists of:

|  |  |
| --- | --- |
| End Users | Information |
| Administrators (HR Department) | Administrators are responsible for the management of human resources. Responsibilities may range from recruiting or firing employees, to checking and updating employee background information |
| Floor Managers | Floor managers are responsible for employee management and the day-to-day activities in the shop floor. Responsibilities may range from managing employee workshifts, to assessing shop floor stock and creating stock requests to the warehouse. |
| Employees | Employees are responsible to the sales and customer service of the shop. They are responsible for running the shop floor during opening hours. |
| Warehouse Manager | Warehouse managers are responsible for the management of the warehouse and its products. Responsibilities may range from adding/removing products into/out of stock, to processing received stock requests. |
| Warehouse Employee | Warehouse employees are responsible for the daily operations of the warehouse. Responsibilities may range from keeping track of warehouse inventory, and moving product from the warehouse floor to the shop floor. |

# **Functional Requirements**

In this section, functionalities inside the application will be documented alongside their corresponding priorities during development.

First Iteration (Week 1 – Week 6)

|  |  |  |
| --- | --- | --- |
| **ID** | **Requirements** | **Priority** |
| FR-01 | Administrators can use CRUD operations to add employees | MUST |
| FR-02 | Administrators can use CRUD operations to view employees background information | MUST |
| FR-03 | Administrators can use CRUD operations to update employees background information | MUST |
| FR-04 | Administrators can use CRUD operations to remove employees | MUST |
| FR-05 | Floor Managers can assign employees to their respective workshifts | MUST |
| FR-06 | Users can access the application through the use of a login page, provided the username and password | MUST |
| FR-07 | Floor Managers can view employee information and keep track of work hours | SHOULD |
| FR-08 | Employees and Floor Managers can post announcements and updates for Floor Managers and other Employees to see | SHOULD |
| FR-09 | Floor Managers can automatically assign workshifts to employees | COULD |
| FR-09a | Warehouse Managers can receive and process stock requests from Floor Managers | COULD |
| FR-09b | Warehouse Managers can accept or reject stock requests | COULD |
| FR-10 | Warehouse Managers can use CRUD operations to add products in stock | COULD |
| FR-11 | Warehouse Managers can use CRUD operations to view product information | COULD |
| FR-12 | Warehouse Managers can use CRUD operations to update product information | COULD |
| FR-13 | Warehouse Managers can use CRUD operations to remove products out of stock | COULD |
| FR-14 | Warehouse Managers can keep track of and update products in stock | COULD |
| FR-15 | Warehouse employees can view accepted stock requests | COULD |
| FR-11 | Administrators can use CRUD operations to add new departments | COULD |
| FR-12 | Administrators can use CRUD operations to view department information | COULD |
| FR-13 | Administrators can use CRUD operations to update department information | COULD |
| FR-14 | Administrators can use CRUD operations to remove existing departments | COULD |

# **Current User Cases**

**User Login:**

**Actor**: All roles

**Summary** : Accessing the application through a login page.

**Main Success Scenario**:

1. Actor enters the application
2. The Actor enters username and password
3. The system checks and confirms the log-in credentials
4. Actor logs in and assumes the role assigned to them alongside the functionalities they possess

**Extensions:** 3a1. Incorrect log-in credentials will show an error message

3a2. System reverts back to **2**

**Add Employee :**

**Actor**: Administrator

**Summary**: The process of adding an employee

**Description**:

1. The system loads an interface to access the “Employee Roster” page
2. The Administrator enters the employee’s background information and assign their role
3. Administrator clicks the “Add” button
4. The system saves the added employee to the database

**Extensions:** 3a. Administrator inputs wrong information

1. System reports “Inaccurate information” when essential information (name, ID, BSN, etc.) was not filled or not in the proper format.
2. Actor fills in correct information
3. System checks input and reverts back to **2**

**Remove Employee :**

**Actor**: Administrator

**Summary**: The process of removing an employee

**Main Success Scenario**:

1. The system loads an interface to access the “Employee Roster” page
2. The system shows database with list of employees
3. Administrator selects employee to remove
4. Administrator clicks the “Remove” button
5. Employee status is updated to “fired” and removed from roster.

**Update Employee Information :**

**Actor**: Administrator

**Summary**: Updates the employee’s background information

**Description**:

1. Administrator visits the “Employee Roster” page
2. Administrator can select an employee
3. System opens up the employees “Account” page
4. Administrator can change the details on the page
5. System saves the changes in the database, updating them.

**Work Shift Check:**

**Actor:**Employee

**Summary:** the employee sees when and where he has to work

**Main Success Scenario:**

1. Employee visits the Shift screen
2. The employee looks through the timetable and sees their assigned shift alongside the location

**Extensions:** 3a: The employee wants to see his work shift for a different time

3a1: The employee select a different week in the timetable

3a2: the program shows the schedule for the new week

**Work Shift Assignment:**

**Actor:**Floor manager

**Summary:** Assigning an employee an allotted time slot

**Main Success Scenario:**

1. The system loads the Shift page.
2. Floor Manager selects the employee from the list
3. Floor Manager chooses a work shift
4. Floor Manager saves the work shift
5. The system updates the time table for the employee

**Extensions:**  2a:Floor Manager doesn’t know the employee’s contract and available work hours information

2a1: Floor manager goes to the employee roster

2a2: Floor manager assesses the work hours left in their week

2a3: Continue to **3**

**Updating Announcement Table:**

**Actor:**Floor manager, Employee

**Summary:** Putting a message in the announcement table

**Main Success Scenario:**

* 1. The actor opens the Announcements page
  2. The actor adds a message to the page
  3. Actor clicks on the post message button
  4. System posts the message and updates the announcement table alongside the sender’s name and time sent for all floor managers and employee to see

**Extensions:** 4a: Floor manager has the option of removing messages

4a1: Floor manager selects a message in the announcement table

4a2: Floor manager clicks “Remove message” button

**Creating Stock Requests**

**Actor:**Floor Manager

**Summary:** Creating a stock request to be sent to the warehouse manager

**Main Success Scenario:**

1. Floor Manager opens stock request interface
2. Floor Manager chooses product to be stocked and the quantity to be re-stocked
3. Floor Manager selects the department where the product is to be re-stocked
4. Floor Manager submits the request
5. System sends message to the Warehouse Manager in a different interface

**Assessing Stock Requests**

**Actor:** Warehouse Manager

**Summary:** Warehouse Manager views stock requests and assesses whether to accept or reject requests

**Main Success Scenario:**

1. Warehouse Manager opens Requests interface
2. Warehouse Manager selects a specific request and click it to access further information
3. **a.** Warehouse Manager accepts the request
4. **b.** Warehouse Manager rejects the request
5. Warehouse Manager submits the request
6. System updates the status of the request in the Floor Manager requests interface

**Extensions:**  3b1. Floor manager inputs the reason for rejection, then proceed to **4**

**Add Product :**

**Actor**: Warehouse Manager

**Summary**: The process of adding a product in stock

**Main Success Scenario**:

1. The system loads an interface to access the “Product inventory” page
2. Warehouse Manager enters the product specifications and signify their amount
3. Warehouse Manager clicks the “Add” button
4. The system saves the added product to the inventory

**Extensions:** 3a. Warehouse Manager inputs wrong information

1. System reports “Inaccurate information” when essential information (name, Barcode ID, measurements, etc.) was not filled or not in the proper format.
2. Warehouse Manager fills in correct information
3. System checks input and reverts back to **2**

**Remove Product :**

**Actor**: Warehouse Manager

**Summary**: The process of removing a product out of stock

**Main Success Scenario**:

1. The system loads an interface to access the “Product Inventory” page
2. The system shows products out of stock (zero amount)
3. Warehouse Manager selects the products to remove
4. Warehouse Manager clicks the “Remove” button
5. System updates product as “Out of stock” and removed from available inventory

**View Accepted Stock Requests :**

**Actor**: Warehouse Employee

**Summary**: View pending stock requests that has been accepted by the warehouse manager and hasn’t been acted upon.

**Main Success Scenario**:

1. The system loads an interface to access the “Pending Requests” page
2. The system shows pending requests sorted by elapsed time since being processed
3. Warehouse Employee selects a specific request
4. Warehouse Manager clicks the “Accept request”
5. System updates request as “Delivered” in both warehouse manager and floor manager Requests interfaces

**Add Department :**

**Actor**: Administrator

**Summary**: The process of adding a department

**Main Success Scenario**:

1. The system loads an interface to access the “Departments” page
2. The Administrator enters the Department information and assigns products to be sold in it
3. Administrator clicks the “Add” button
4. The system saves the added department to the database

**Extensions:** 3a. Administrator inputs wrong information

1. System reports “Inaccurate information” when a similar department has already existed or essential information was not filled in the correct format
2. Actor fills in correct information
3. System checks input and reverts back to **2**

**Remove Department :**

**Actor**: Administrator

**Summary**: The process of removing a department

**Main Success Scenario**:

1. The system loads an interface to access the “Departments” page
2. The system shows database with list of Departments
3. Administrator selects Department to remove
4. Administrator clicks the “Remove” button
5. System removes department from the list

**Update Department Information :**

**Actor**: Administrator

**Summary**: Updates the department information

**Main Success Scenario**:

1. Administrator visits the “Departments” page
2. Administrator can select a department from the database
3. System opens up the “Department Information” page
4. **a.** Administrator changes details of the department (Name, specifications, etc)
5. **b.** Administrator changes the products assigned to a specific department
6. System updates the department in the database

# **GUI**

Graphical user interface

Description automatically generated

Graphical user interface

Description automatically generated with medium confidence

Green Line : For Manager and Empliyee

Red Line : For manager and Administrator only

Blue Line : For Employee only

# **Website Wireframe**

**Manager Page**

**Graphical user interface, website

Description automatically generated**

**Graphical user interface, text

Description automatically generated with medium confidence**

**Graphical user interface, application

Description automatically generatedGraphical user interface, text, application, email

Description automatically generatedGraphical user interface

Description automatically generated with medium confidence**

**Graphical user interface

Description automatically generated**

**Employee Page**

**Graphical user interface, website

Description automatically generated**

**Graphical user interface, application

Description automatically generated**