

User Requirement Specifications



Members:

Ivan Ovcharov | 4090993

Anton Yordanov | 4095995

Hitesh Manglani | 3880753

Anatolie Dunai | 4328809

Tutor:

Koehorst, Michiel M.W.

GIT repository: https://git.fhict.nl/I470558/prj_cb01_01

Table of content

Table of content	2
Agreements with client	3
Functional requirements	4
Use cases	7
Use case #1: Assign workshift to employee	7
Use case #2: View statistics of employees	7
Use case #3: View different statistics of products	8
Use case #4: Add employee	8
Use case #5: View personal information	9
Use case #6: Change information of the specific user	9
Use case #7: Deleting an employee	9
Use case #8: Search employee by id	10
Use case #9: Update employee personal information	10
Use case #10: Search employee by name	10
Use case #11: Add product	11
Use case #12: Search product by brand name	11
Use case #13: Update product	11
Use case #14: Search product by id	12
Use case #15: Delete product by id	12
GUI	13
Login page:	13
Personal employee page:	13
Personal admin page	14
Manage employees page	14
Manage products page	15

Agreements with client

- A working software solution will be delivered on week 6.
- We have agreed with the client to work on the functionalities that are urgent (employee management and).
- We have not agreed on a specific budget.
- We have agreed to deliver a blank database that can be filled with the user's information (attributes of products, employee information, etc.).
- We have agreed to not deliver any hardware.
- We have agreed to have designated meetings, arranged by both our team and the client.
- We have agreed to provide a secure application, keeping all of the client's private information safe.

Functional requirements

All Users:

- FR-001. Must be able to login to the system
- FR-002. Must be able to view personal information
- FR-003. Could be able to change personal information.
- FR-004. Should be able to log out from the system.
- FR-005. Should be able to see the workshifts assigned to him.

Employee:

- FR-006. Must be able to see and complete move requests for stock
- FR-20. Must be able to view personal information
- FR-21. Must be able to change personal information
- FR-22. Must be able to call in sick
- FR-23. Being able to choose work days of preference
- FR-24. Must be able to apply for vacation days

Manager:

- FR-007. Should be able to check statistics.
- FR-008. Must be able to create move requests for stock.
- FR-009. Must be able to assign shifts for employees
- FR-011. Must be able to add products in the system.
- FR-012. Must be able to change information about specific products.
- FR-018. Must be able to delete shifts
- FR-019. Must be able to edit shifts

FR-020. Must be able to remove products

Admin:

FR-010. Must be able to add new employees or managers.

FR-012. Must be able to change information about specific products.

FR-013. Should be able to change information about employees.

FR-014. Must be able to remove employees or managers.

FR-015. Should be able to view employee or manager information

FR-016. Should be able search for employees or managers

FR-017. Could be able to remove old products

Must	Should	Could	Won't
All Users: FR-001. Must be able to login to the system	All Users: FR-004. Should be able to log out from the system.	Admin: FR-017. Could be able to remove old products	Deliver hardware
All Users: FR-002. Must be able to view personal information	All Users: FR-005. Should be able to see the workshifts assigned to him.	All Users: FR-003. Could be able to change personal information.	Deliver Database with information about employees and products
Employee:	Manager:	Manager:	-----

FR-006. Must be able to see and complete move requests for stock	FR-007. Should be able to check statistics.	FR-008. Could be able to create move requests for stock.	
Employee: FR-006. Must be able to see and complete move requests for stock	Admin: FR-013. Should be able to change information about employees.	Manager: FR-017. Could view different statistics of each product	-----
Manager: FR-009. Must be able to assign shifts for employees	Admin: FR-015. Should be able to view employee or manager information	-----	-----
Admin: FR-010. Must be able to add new employees or managers.	Admin: FR-016. Should be able search for employees or managers	-----	-----
Admin: FR-011. Must be able to add products in the system.	-----	-----	-----

Admin: FR-012. Must be able to change information about specific products.	-----	-----	-----
Admin: FR-014. Must be able to remove employees or managers.	-----	-----	-----
Manager: FR-018. Must be able to delete shifts	-----	-----	-----
Manager: FR-019. Must be able to edit shifts	-----	-----	-----
Employee: FR-20. Must be able to view personal information	-----	-----	-----
Employee:	-----	-----	-----

FR-21. Must be able to change personal information			
Employee: FR-22. Must be able to call in sick	-----	-----	-----
Employee: FR-23. Being able to choose work days of preference	-----	-----	-----
Employee: FR-24. Must be able to apply for vacation days	-----	-----	-----

Use cases

Use case #1: Assign workshift to employee

Actor : Manager

Main Success Scenario:

1. Manager opens the scheduling section
2. Manager selects employee based on unique identifier and name
3. System opens calendar on the current month
4. Manager chooses the work week

5. System displays calendar
6. Manager assigns the work shifts
7. Manager confirms the selected work shifts
8. The system notifies the employee

Use case #2: View statistics of employees

Actor: Manager

Main Success Scenario:

1. Manager opens the statistics section for employees
2. Manager selects which employee they want to see
3. System displays the requested information

Extension:

- 2a. Manager selects to view the statistics for all employees
- 2b. Manager selects to view the statistics for a selected department

Use case #3: View different statistics of products

Actor: Manager

Main Success Scenario:

1. Manager opens the statistics section for products
2. Manager selects which product they want to see
3. System displays the requested information

Extension:

- 2a. Manager selects to view the statistics for all products
- 2b. Manager selects to view the statistics for a specific category

Use case #4: Add employee

Actor: Administrator

Main Success Scenario:

1. Admin opens the add employee section
2. Admin fills in name, email, password, department, address and salary of each employee
3. Admin adds the employee
4. System displays whether it was successful or not

Extension:

2a. If key information is missing (employee number, address, etc.) the system will notify and add will not be successful and the administrator has to fill the needed information.

Use case #5: View personal information

Actor: Employee

Main Success Scenario:

1. Employee opens the personal account section
2. Employees can check personal information, such as name, address, salary, etc.

Use case #6: Change information of the specific user

Actor: All Users

Main Success Scenario:

1. User opens the personal account section.
2. User selects what piece of information they want to manipulate
3. The field becomes editable
4. User edits selected data
5. User saves the changes
6. System displays the new (edited) information

Extensions:

- 2a. Some information can only be viewed, but not manipulated
- 3b. Upon changing password, the previous one will not be shown

Use case #7: Deleting an employee

Actor: Admin

Main Success Scenario:

1. Admin opens the employee management section
2. Admin selects the desired employee
3. Admin selects the desired employee
4. Admin clicks on the delete button

Extensions:

- 2a. If no employee is found, nothing happens

Use case #8: Search employee by department

Actor: Admin

Main Success Scenario:

5. Admin opens the employee management section
6. Admin enters the desired employee's department
7. Employee is now visible

Extensions:

- 2a. If no employee's department is found, nothing displayed

Use case #9: Update employee personal information

Actor: Admin

Main Success Scenario:

8. Admin opens the employee management section
9. Admin enters the desired change
10. Admin clicks the button labeled "Update"
11. Admin gets a visible confirmation of confirmed change

Extensions:

- 2a. If information is is not valid, the admin will get a visible cue of what needs to be fixed

Use case #10: Search employee by name

Actor: Admin

Main Success Scenario:

12. Admin opens the employee management section
13. Admin enters the desired employee's name
14. Employee is now visible

Extensions:

- 2a. If no employee is found, nothing displayed

Use case #11: Add product

Actor: Manager

Main Success Scenario:

1. Manager opens the add Product section
2. Manager fills in needed information for the product.
3. Manager adds the product
4. System displays whether it was successful or not

Extension:

2a. If key information is missing (product name, brand, etc.) the system will notify and add will not be successful and the manager has to fill the needed information.

Use case #12: Search product by brand name

Actor: Manager

Main Success Scenario

1. Manager opens the product section
2. Manager fills in the desired brand name
3. Manager clicks the search by brand button
4. System displays whether it was successful or not

Extension:

2a. If the input is empty, nothing is displayed.

Use case #13: Update product

Actor: Manager

Main Success Scenario

1. Manager opens the product section
2. Manager selects one of the already existing products
3. The information of the selected product gets filled into the text fields
4. Manager clicks on the text field with the information he wants changed and updates it
5. Manager clicks on the update button
6. System displays whether it was successful or not

Use case #14: Search product by department

Actor: Manager

Main Success Scenario

1. Manager opens the product section
2. Manager fills in the desired department
3. Manager clicks the search by department button
4. System displays whether it was successful or not

Extension:

2a. If the input is empty, nothing is displayed.

Use case #15: Delete product

Actor: Manager

Main Success Scenario

1. Manager opens the product section
2. Manager clicks on the desired product
3. Manager clicks the Delete button
4. System displays whether it was successful or not

Extension:

2a. If no product is clicked, nothing will be deleted.

Use case #16: Delete shifts

Actor: Manager

Main Success Scenario

1. Manager selects a shift
2. Manager presses the delete button
3. System gives confirmation that a shift is removed

Extension:

3a. If no shift is selected the system will ask for one to be selected and nothing will be deleted.

Use case #17: Edit shifts

Actor: Manager

Main Success Scenario

1. Manager selects a shift
2. Manager edits
3. Manager presses the edit button
4. System gives confirmation that a shift is edited

Extension:

- 4a. If no shift is selected the system will ask for one to be selected and nothing will be changed.
- 4b. if the shift isn't changed the system won't change anything.

Use case #17: Add department

Actor: Admin

Main Success Scenario

1. Admin opens the employee section
2. Admin clicks on the "Manage department"
3. Admin enters department name
4. Admin clicks on "Add department"
5. System displays the added department

Extension:

- 4a. If the input field for the department is empty, the message "Please fill in a department" will display.

Use case #18: Delete department

Actor: Admin

Main Success Scenario

1. Admin opens the employee section
2. Admin clicks on the "Manage department"
3. Admin clicks on desired department name
4. Admin clicks on "Delete department"

5. System deletes selected department

Use case #19: Viewing employee information

Actor: Employee

Main Success Scenario

1. Employee opens the information section
2. Employee clicks selects on the desired field
3. Employee is able to see his personal information

Use case #20: Changing employee information

Actor: Employee

Main Success Scenario

1. Employee opens the information section
2. Employee clicks selects on the desired field
3. Employee is able to change his personal information

Use case #21: Employee can call in sick

Actors: Employee and Manager

Main Success Scenario

1. Employee opens the shifts section
2. Employee selects the days he will be absent
3. Employee specifies the reason behind his absence
4. Employee confirms by pressing the confirm button
5. Managers are informed of his absence and can confirm it

Use case #21: Employee can apply for vacation

Actors: Employee and Manager

Main Success Scenario

1. Employee opens the shifts section
2. Employee selects the days he desires
3. Employee specifies the reason
4. Employee confirms by pressing the confirm button
5. Managers are informed of his application and can confirm it

Use case #22: Employee can specify work days preference

Actors: Employee and Manager

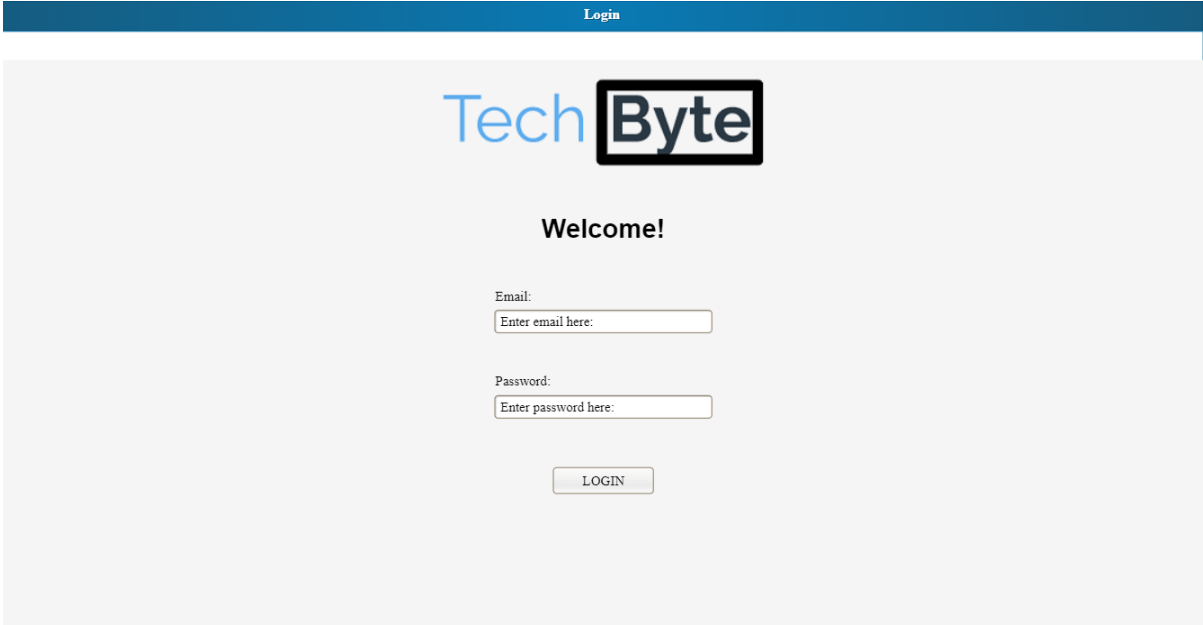
Main Success Scenario

1. Employee opens the shifts section
2. Employee selects the days he desires to work
3. Employee confirms by pressing the confirm button
4. Managers are informed of his application and can confirm it

GUI

Login page:

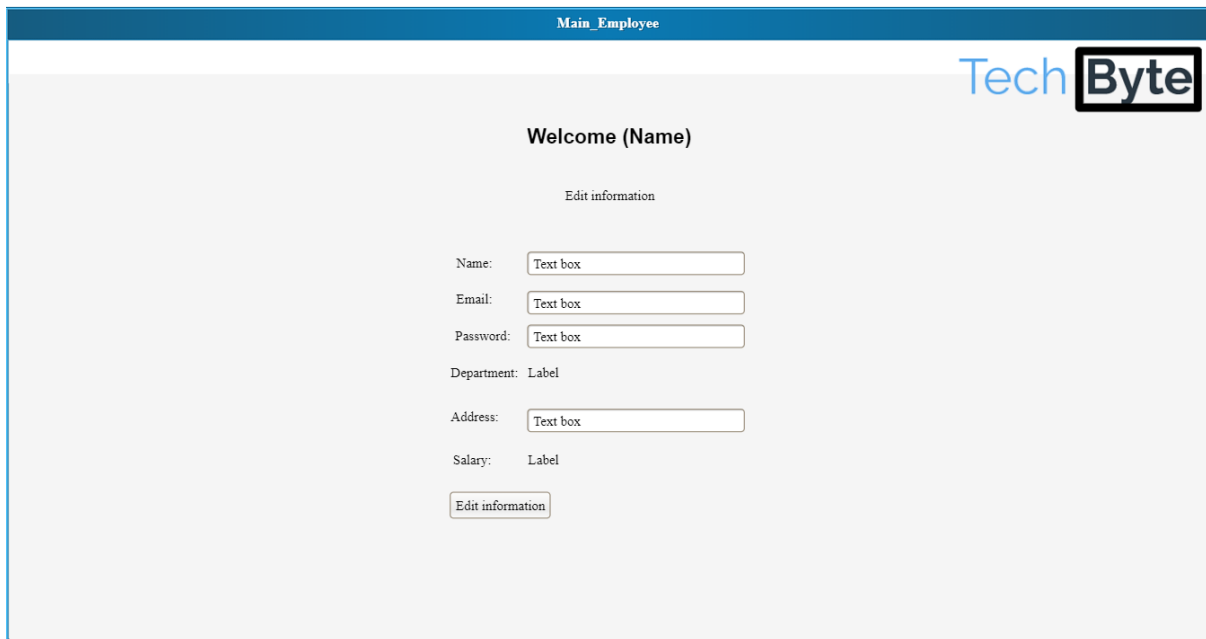
Users enter their information and enter the application



The screenshot shows a web application's login page. At the top, there is a dark blue header bar with the word "Login" in white. Below the header, the page has a light gray background. In the center, the "TechByte" logo is displayed, with "Tech" in blue and "Byte" in black with a thick black border. Below the logo, the text "Welcome!" is centered. Underneath, there are two input fields: one for "Email:" with the placeholder text "Enter email here:" and one for "Password:" with the placeholder text "Enter password here:". Below these fields is a "LOGIN" button.

Personal employee page:

Here employees can see and edit their information



Main_Employee

Tech **Byte**

Welcome (Name)

Edit information

Name:

Email:

Password:

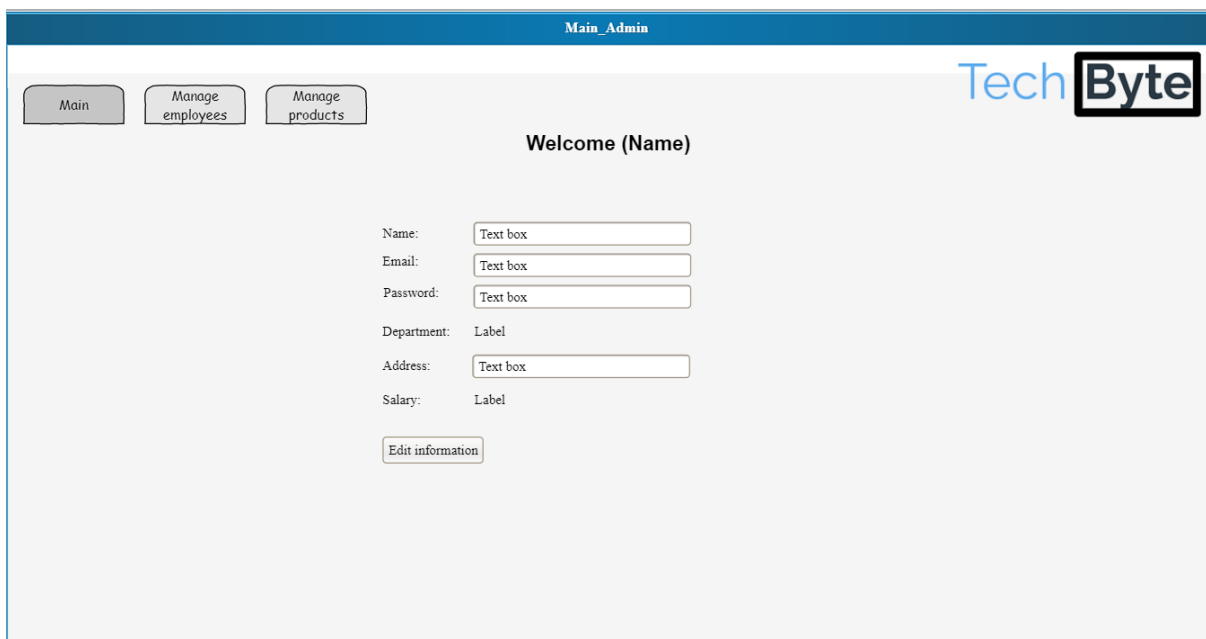
Department: Label

Address:

Salary: Label

Personal admin page

Here admins can see and edit their information



Main_Admin

Tech **Byte**

Main Manage employees Manage products

Welcome (Name)

Name:

Email:

Password:

Department: Label

Address:

Salary: Label

Manage employees page

Here admins can edit information about employees

The screenshot shows the 'Employees_Admin' web application. At the top, there is a blue header bar with the text 'Employees_Admin'. Below the header, there are three navigation buttons: 'Main', 'Manage employees', and 'Manage products'. The 'Manage employees' button is highlighted. On the right side, there is a 'TechByte' logo. The main content area has a 'Welcome (Name)' message. Below this, there are three sections: 'List of all employees:' with a large empty box, 'Select employee' with a form for Employee ID and a 'Select' button, and 'Search by name:' with a form for Employee name and a 'Search' button. On the right, there is an 'Employee information' section with a form for Name, Email, Password, Department, Address, Salary, and an 'Update' button.

Manage products page

Here admins can edit information about products

The screenshot shows the 'Products_Employee' web application. At the top, there is a blue header bar with the text 'Products_Employee'. Below the header, there are three navigation buttons: 'Main', 'Manage employees', and 'Manage products'. The 'Manage products' button is highlighted. On the right side, there is a 'TechByte' logo. The main content area has a 'Welcome (Name)' message. Below this, there are three sections: 'List of all product' with a large empty box, 'Select product' with a form for Product ID and a 'Select' button, and 'Search by product:' with a form for Product name and a 'Search' button. On the right, there is a 'Product information' section with a form for Description, Model, Brand, Retail Price, Amount, Price, Dimension, Category, Shelf, and an 'Update' button.

Future pages:

Information/statistics of all employees and products

Iteration 3

Duration: 2 weeks

Planned deliveries:

1. Stock requests
2. Hosting of website
3. Merge websites
4. Shift availability
5. Fixing bugs

Potential:

1. Automatic shifts
2. Fixing design

Must	Should	Could	Won't
Stock requests - Ivan	Log out - Hitesh		
Fix Departments -Ivan	Design application (HR, Depot) - Hitesh		
Host website - Hitesh	Auto Shifts - Anton		
Merge website - Hitesh			
Design application (Shifts) -Anton			
Documentation - Everyone			

Iteration 4

Duration: 2 weeks

Planned deliveries:

1. Fixing bugs
2. Updating documentation
3. Completion of auto-shifts
4. Completion of stock requests
5. Completion of statistics
6. Better usability
7. Better deliverables

Must	Should	Could	Won't
Completion of statistics			
Updating documentation			
Better usability			
Updating documentation			
Fixing bugs			
Automatic shifts			
Better deliverables			