

User Requirements Specification

Software solution for “Media Bazaar”
GROUP 1

Members: Bohdan Tymofieienko, Stelian Rumenov, Andreea Sindrilaru, Andrei Sava.

Tutor: Pencheva, Sabina S.

Date: 07/03/2021

Version: 0.3

Table of contents

Agreements with client	3
Sides:	3
Agreements:	3
Functional requirements	4
Windows application (Manager and Administrator side):	4
Website (Employee side):	5
Prioritizing table MosCow	6
Use cases	9
Wireframes	11

Agreements with client

Sides:

Client: Media Bazaar represented by Andre Postma, "*Client*" later in this document.

Contractor: Group 1 (name TREE), "*Contractor*" later in this document.

Agreements:

- 1) Contractor does the problem analysis.
- 2) Contractor prepares the problem solution in form of information system.
- 3) Contractor implements the solution in form of client-server software application for Windows operating system.
- 4) Team presents the product to the Client.
- 5) Project must be completed within the period of 6 weeks.

Functional requirements

All the requirements were discussed with client. *“User” later in the document is an abstract alias to describe the functionality.*

Windows application (Administrator side):

1. **FR-A-01** User can login to the system with personal credentials.
2. **FR-A-02** User can change the password.
3. **FR-A-03** User can logout from the system.
4. **FR-A-04** User can view list of employees.
5. **FR-A-05** User can add new employee to the system.
6. **FR-A-06** User can add new administrator to the system.
7. **FR-A-07** User can add new manager to the system.
8. **FR-A-08** User can view a list of accounts.
9. **FR-A-09** User can remove administrator's or manager's account.
10. **FR-A-10** User can update employee information.
11. **FR-A-11** User can modify employee status
 - iii) Active / Inactive
 - ii) Started / Not started
12. **FR-A-12** User can view employee schedule.
13. **FR-A-13** User can assign one of three shifts to employee.
14. **FR-A-14** User can update information concerning employee's schedule.
15. **FR-A-15** System can automatically organize employee schedule
 - i) Following employee preference if possible.
 - ii) Based on work contract.
 - iii) According to work regulation.
16. **FR-A-16** User can cancel shift assignment for the employee.

Windows application (Manager side):

1. **FR-M-01** User can login to the system with personal credentials.
2. **FR-M-02** User can change the password.
3. **FR-M-03** User can logout from the system.
4. **FR-M-04** User can view employee list.
5. **FR-M-05** User can view employee's details.
6. **FR-M-06** User can view statistical information.
(Employee average attendance, employee average salaries, ratio of employee per department).
7. **FR-M-07** User can view employee schedule.

Website (Employee side):

1. **FR-E-01** User can login to the system with personal credentials.
2. **FR-E-02** User can change the password.
3. **FR-E-03** User can logout from the system.
4. **FR-E-04** User can view the work schedule.
5. **FR-E-05** User can choose preferred shift.
6. **FR-E-06** User can cancel the schedule.
7. **FR-E-07** User can send message(complaint, suggestion etc.) to a manager.
8. **FR-E-08** User can view list of items in stock.
9. **FR-E-09** User can send restoke request.


Naming convention:




{FR-(Abbreviation from user's type name)-(count in two-digit format)}

Prioritizing table MoSCow

MoSCoW table represents four levels of task prioritizing. The abbreviation stands for *Must*, *Should*, *Could* and *Won't*.

We consider it crucially important to define priorities before implementation phase, so both developers and client have realistic expectations on project results.

Feature	Must have	Should Have	Could have	Won't have
Store employee data (name, number, address etc.)				
Store stock data (product name, number, quantity etc.)				
Assign workshifts to employees				
View employee and stock statistics				
Extend application features				
Add or terminate employees				
Store history of previous employees				

Login/logout to unique profile				
Restrict or grant access to certain permissions and features based on role				
View employee attendance				
Register as a new user				
Track employee activity/availability				
Schedule shifts to employees				
View departments and their information				
View products and their information				
Add/remove a department				
Add/remove products				

Access system from home computer				
Employee can edit their work-related information (such as salary and role)				
State shift preference				
Website connected to same database as application				
Employee can login/logout of website				
Employee can register in website				
Employee can view personal information in website				
Employee can view schedule in website				
Employee can contact manager through website				
Employee can state a shift preference				

Use cases

Use case: Administrator adds new employee

Actors: Administrator

Main Success Scenario:

1. Administrator clicks "Add new employee".
2. New form is open.
3. Administrator inputs employee information.
4. Administrator press "Done."
5. System shows that employee has been successfully added.

Extension:

1. Field is empty
 - a. Notify user.

Use case: Administrator updates employee information

Actors: Administrator

Main Success Scenario:

1. Administrator clicks "Update employee information"
2. New form is open.
3. Administrator may change employee information.
4. Administrator press "Done."
5. System shows that information has been successfully updated.

Extension:

1. Field is empty
 - a. Notify user.

Use case: Administrator updates employee status

Actors: Administrator

Main Success Scenario:

1. Administrator clicks "Update employee information"
2. New form is open.
3. Administrator may change employee status (1 out of 4).
4. Administrator press "Done."
5. System shows that status has been successfully updated.

Extension:

1. Field is empty
 - a. Notify user.

Use case: Administrator assigns a shift to employee

Actors: Administrator

Main Success Scenario:

1. Administrator opens a week scheduler.
2. Administrator clicks on an empty slot.
3. Administrator chooses employee from the list of potentially available employees.
4. System shows that employee was successfully assigned.

Use case: Administrator adds new manager to the system

Actors: Administrator

Main Success Scenario:

1. Administrator opens an account section.
2. Administrator clicks on "Add new user".
3. "Add new user" form is open.
4. Administrator chooses "manager" from dropbox "Role".
5. Administrator inserts name and email in the textboxes.
6. Administrator clicks on "Add".
7. System shows that user was successfully added.
Password is shown on the screen.

Use case: Manager logs in first time

Actors: Manager

Main Success Scenario:

1. Manager opens an application.
2. Login form is opened.
3. Manager logs in with credentials received from administrator.
4. Manager opens MyAccount module.
5. Manager enters current password and new password.
6. System shows that password was successfully changed.

Use case: Manager views the employee statistics

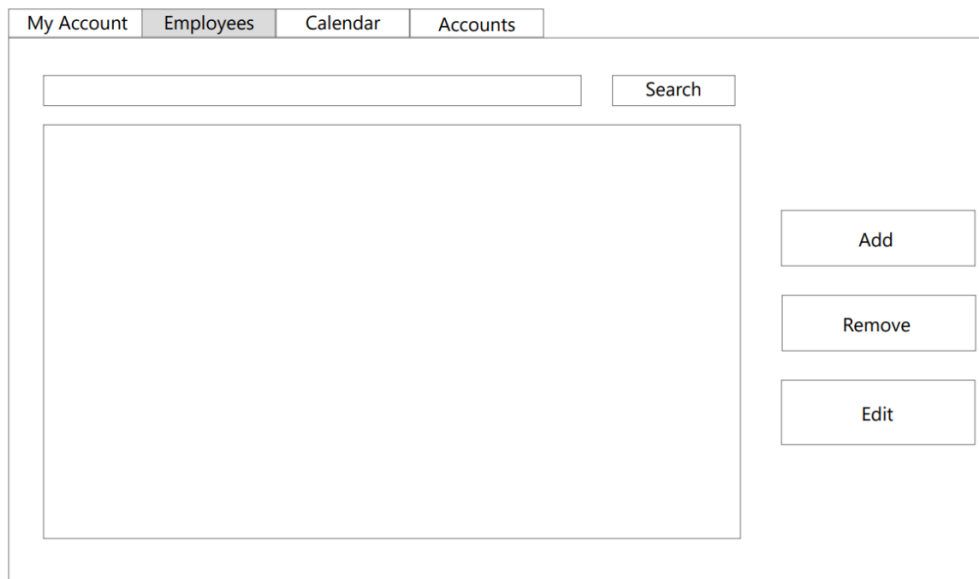
Actors: Manager

Main Success Scenario:

1. Manager opens a statistics section.
2. Manager clicks on "Employees".
3. Manager may go through the employee list.
4. Manager may search for certain employee.
5. Manager may see the graph with average attendance.
6. Manager may filter the departments to see attendance of specific department.
7. Manager closes the form.

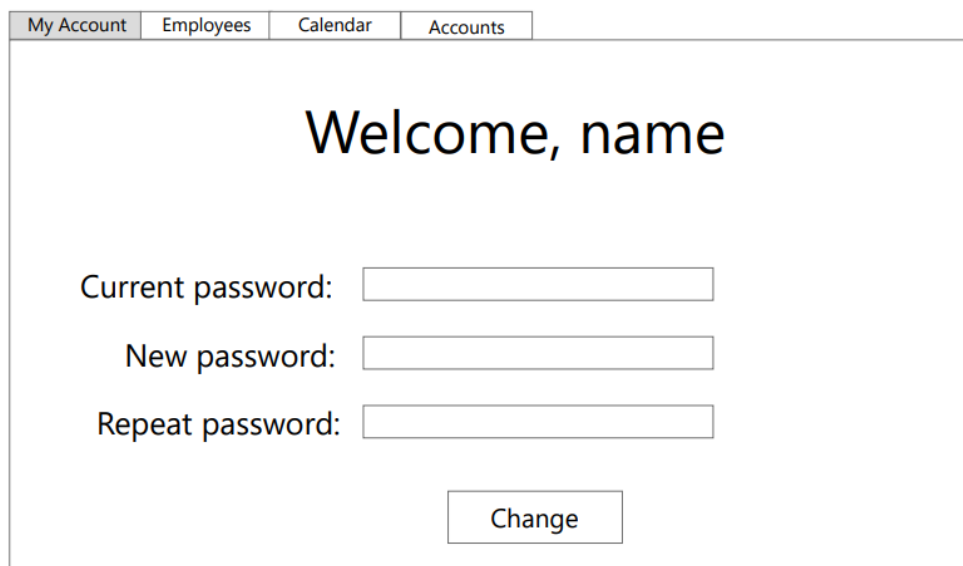
Wireframes

Employee section (list of employees with search bar and button referencing to dialog windows)



This wireframe shows the 'Employees' section of a web application. It features a top navigation bar with four tabs: 'My Account', 'Employees' (which is highlighted), 'Calendar', and 'Accounts'. Below the navigation bar, there is a search bar with a 'Search' button to its right. A large, empty rectangular box occupies the main content area, intended for a list of employees. To the right of this box, there are three vertically stacked buttons: 'Add', 'Remove', and 'Edit'.

MyAccount section (user may change the password).



This wireframe shows the 'My Account' section of a web application. It features a top navigation bar with four tabs: 'My Account' (which is highlighted), 'Employees', 'Calendar', and 'Accounts'. Below the navigation bar, the text 'Welcome, name' is displayed in a large font. Underneath, there are three password input fields, each preceded by a label: 'Current password:', 'New password:', and 'Repeat password:'. At the bottom center, there is a 'Change' button.

Example of dialogue window with all forms needed to add new employee.

First name	<input type="text"/>	Hourly wage	<input type="text"/>
Last name	<input type="text"/>	Account number	<input type="text"/>
Address	<input type="text"/>	Department	<input type="text"/> ▼
Email	<input type="text"/>	Role	<input type="text"/> ▼
Date of birth	<input type="text"/>	Contract	<input type="text"/> ▼
Date of first working day	<input type="text"/>	Shifts	
		Shift A	<input type="checkbox"/>
		Shift B	<input type="checkbox"/>
		Shift C	<input type="checkbox"/>

Add employee

More wireframes can be found in our GIT repository:

<https://git.fhict.nl/l454066/prj-cb07-group1/-/tree/master/Documentation/Wireframes.pdf>