# **URS**



Location:

Members: Kwangjin Lee, Evaldas Drasutis, Emilian Alexandru, Florin Deleanu

Tutor: Stan S. van Hartingsveldt

Date: 03/09/2020

Table of Contents:

1. Title page
2. Agreements with client
3. Functional requirements
   1. What should the system be able to do?
   2. Group them by application/website (if applicable)
   3. Prioritize them using MoSCoW (must, should, could, won’t)
4. Use Cases
5. GUI
   1. Images of the various views.
   2. A brief description per view how it works
6. Website wireframe (when applicable)
   1. The structure of your website with a brief description of its intent.

Agreements with Client:

1. Employee administration is the most important part of the program, followed by stock
2. The following will be the information on employees:

* First name
* Surname
* Username (account)
* Password (account)
* Phone number
* Address
* Email address
* Spouse (and contact details)
* Date of birth
* BSN (Social security number)
* Functions (at least one, possibly more functions the employee can serve as e.g. sales representative)

1. Employees will manage their password as well as information that is subject to change( example: Spouse, address, etc…) They will be able to also see the unchangeable information like BSN. //list them in detail, to make it less unambiguous
2. In order to search for employees you can use either their ID or name
3. Application will be build for Windows
4. Employees will check in when arriving and out when leaving. Tracking break time is not needed
5. An employee can have 1 or 2 shifts per day. In the latter case they must be consecutive
6. All employees will use the website but they will only see their information and schedule
7. Website pages should have a clear focus on the topic and have an intuitive to use interface
8. The main statistics tracked are the number of employees, the average number of shifts, the number of hours compared to their contract, and the illness rates. The client would like to have the total number of employees and products. About stock, he would like to see the stock levels, avg time between replenishment of actions and stock.

Functional requirement:

1. Administration should be able to add new employees.(and change their data)
2. Administration should be able to remove employees.(button should be harder to reach than the deactivate button)
3. Administration should be able to dismiss an employee. //not fully save the reason why fired)
4. Administration should be able to change employees data.
5. Administration should be able to assign an employee to a department. //not high priority
6. Administration should be able to schedule employees.
7. Administration should be able to update the employees schedule. // add also mean that you unschedule them
8. Administration should be able to remove employees from the schedule.
9. Administration should be able to mark employees' attendance.
10. Administration should be able to view employees absent days
11. Administration should be able to view employees statistics.
12. Administration should be able to view employees' information. (idk if we need it)
13. Administration should be able to search for specific employee.(idk about this)
14. Sales representative should be able to request stock from the depot.
15. Sales representative should be able to view requested status information.
16. Sales representative should be able to accept incoming stock.
17. Sales representative should be able to edit shelf stock.
18. Sales representative should be able to view shelf stock information.
19. Sales representative should be able to view production statistics.//not high priority
20. Sales representative should be able to disable stock.
21. Depot should be able to view data on stock.
22. Depot should be able to add new stock.
23. Depot should be able to remove stock items.
24. Depot should be able to deactivate stock.(maybe first they need to deactivate the stock and then to remove it to make it harder to get the stock removed?)
25. Depot should be able to accept stock refill request.(send the stock with a processing ticket which is accepted by the sales representative when it arrives in the store)
26. Depot should be able to view stock request status.
27. Depot should be able to respond to stock requests.
28. User should be able to log in into the application.
29. User should be able to log into the website.(web)
30. User should be able to edit profile information.(web)
31. User should be able to view working shifts.(web)
32. User should be able to logout of the application
33. User should be able to logout of the website

UC done for: FR-1, FR-2, FR-3, FR-4, FR-13, FR-14, FR-20, FR-21, FR-22, FR-24, FR-25, FR-28, FR-29, FR-32, FR-33

UC remaining for: FR-5, FR-6, FR-7, FR-8, FR-9, FR-10, FR-11 , F-12, FR-15, FR-16, FR-18, FR-19, FR-23, FR-26, FR-27, FR-30, FR-31

Moscow prioritization table

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Administration** | | | | |
| **Functional Requirements** | **M** | **S** | **C** | **W** |
| Create new employee account | X |  |  |  |
| Schedule employees | X |  |  |  |
| Edit Employee account information | X |  |  |  |
| Search for employee | X |  |  |  |
| Update employees schedule | X |  |  |  |
| assign an employee to a department |  | X |  |  |
| View inactive/removed employee’s information |  |  | X |  |
| Delete employee account |  |  | X |  |
| Deactivate active employee account |  |  | X |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Sales representative** | | | | |
| **Functional Requirements** | **M** | **S** | **C** | **W** |
| Request stock | X |  |  |  |
| View request status | X |  |  |  |
| View shelf stock information | X |  |  |  |
| Delete items from stock | X |  |  |  |
| View statistics regarding stock changes | X |  |  |  |
| View product statistics |  | X |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Depo** | | | | |
| **Functional Requirements** | **M** | **S** | **C** | **W** |
| View stock data | X |  |  |  |
| Add new stock | X |  |  |  |
| See incoming shelf stock |  | X |  |  |
| Deactivate stock item | X |  |  |  |
| Remove deactivated product | X |  |  |  |
| Accept stock refill request |  | X |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **User** | | | | |
| **Functional Requirements** | **M** | **S** | **C** | **W** |
| Login(application,website) | X |  |  |  |
| Edit profile info(website) | X |  |  |  |
| View working shifts(web) | X |  |  |  |
| Mark employees attendance(application) | X |  |  |  |

Use Cases:

Potential end users are company’s staff that maintain and view the information in the software and website.

In the application the following actors should be able to:

Actor:Administration()

Actor: Sales representative ()

Actor: Depot()

Use case: View data of stock

Actor: Depot

Main Success Scenario:

1. Employee selects the refresh
2. System returns the most recent data.
3. Data displayed in the application

Use case: Add employee

Actor: Administrator

Main Success Scenario:

1. User goes to a button to add a new employee.
2. System requests for employees information.
3. Administrator fills in employees personal information (listed in agreements with client number 2)
4. System shows a message box of confirmation to add a new employee.

Extensions:

1. Not all mandatory fields are filled in.
   1. System displays not all fields filled in the message.
   2. End of use case.

Use case: Login to the application or website

Actor: Any User

Pre-Condition:

1. Employee starts the application
2. Employee logs out

Main Success Scenario:

1. Employee fills in credentials and confirms
2. System confirms the credentials exist in the database
3. System redirects the employee to the main page

Extension:

2a: System rejects the credentials

1. System displays an error message informing the user of the wrong provided information.
2. End of use case

Use Case:Adding new products in stock

Actor: Depo

Main Success Scenario:

1. Employee goes to the button labeled add stock
2. Employee fills in the new product information, including the amount in stock and confirms
3. System creates a new product, adding it to the list of available products

Extension:

2a: Employee fills in information not matching the systems criteria

1. System displays a wrong information message, informing the employee of the correctness and location of the problem
2. Return to MMS step 2

2b: Employee fills in already existing information

1. System displays an already existent product message
2. The employee is returned to the stock overview page
3. End of use case

Use Case: Making a product unavailable

Actor: Sales representative

Pre-condition: User must modify an existing product

Main Success Scenario:

1. User clicks a radio button in order to make product unavailable
2. User confirms the changes
3. System displays a confirmation message and mark the product as unavailable

Use Case: Modify a product

Actor: Depo worker

Pre-condition: User must select an item from the product list

Main Success Scenario:

1. User goes to the modification tab by pressing a button
2. System displays the current product information
3. User makes changes to the product and confirms
4. System displays a confirmation message and saves the changes

Extension:

4a: Filled in information does not match the system criteria

1. System displays an error message informing the user of the problem
2. Return to MMS step 3

Use Case: Make a stock refill request

Actor: Sales representative

Main Success Scenario:

1. User selects a product from the list and confirms by pressing a button
2. System process the order and displays a confirmation message

Extension:

2a: Stock is already at max capacity

1. System displays an error message and cancels the order
2. End of use case

Use Case: Permanently removing an employee

Pre-condition: Administration must search for an employee and select them from a list

Actor:Administration

Main Success Scenario:

1. User goes to additional options by pressing a button in the employee management form
2. User presses the permanently remove button
3. System opens a new window and asks if they want to proceed
4. User confirms
5. System permanently removes the employee

Extension:

1a: No employee selected

1. System displays a message informing the user of their problem
2. End of use case

Use Case: Assigning shifts //In-progress

Actor: Administration

Pre-Condition: Administration must search for an employee and select them from a list

Main success scenario:

1. User chooses a day from the selected week
2. User assigns the shift by pressing one of the 3 buttons
3. User presses the “Assign selected employee button”
4. System assigns the selected user to the chosen date/shift

Extension :

1a :User does not select the date or shift

1. System display error and inform the user of the missing selection
2. Return to MMS step 1
3. End of use case

Use Case:Searching for an employee

Actor :Administration

Main success scenario :

1. User enters the name or id then confirms
2. System displays a message that a match was found along with the employee information(First name, Surname)

Extension

2a: System finds multiple matches

1. System informs user that multiple matches were found showcasing them inside a list
2. User selects one of the returned employees and confirms
3. End of use case

Use case : Administration should be able to dismiss and employee

Pre-Condition:Administration must search for an employee and select them from a list

Actor :Administration

Main success scenario:

1. User presses the dismiss button
2. System displays a new window asking the user to provide a reason
3. User provides the reason and confirms

Use case : Changing employee data

Pre-Condition : Administration must search for an employee and select them from a list

Actor:Administration

Main success scenario :

1. User goes to the modification tab by pressing a button
2. System displays the current employee information
3. User makes changes to the employee data with the help of text inputs and confirms
4. System displays a confirmation message and saves the changes

Extension:

4a: Filled in information does not match the system criteria

1. System displays an error message informing the user of the problem
2. Return to MMS step 3

Use case: Accepting stock refill request

Actor: Depot Worker

Main success scenario:

1. The user navigates to the requests tab
2. System displays a list of requests tickets
3. The user selects a ticket from the list and accepts it
4. The system marks the ticket as processed

Use case: Viewing workishifs on the website(do we need an use case if the schedule is on the homepage?)

Actor: Any user

Pre-condition: User must be logged in

Main success scenario:

Use Case: Log out clients website

Actor: Any user

Main success scenario:

1. User presses the logo-off button
2. System logs-off the user returning to the homepage

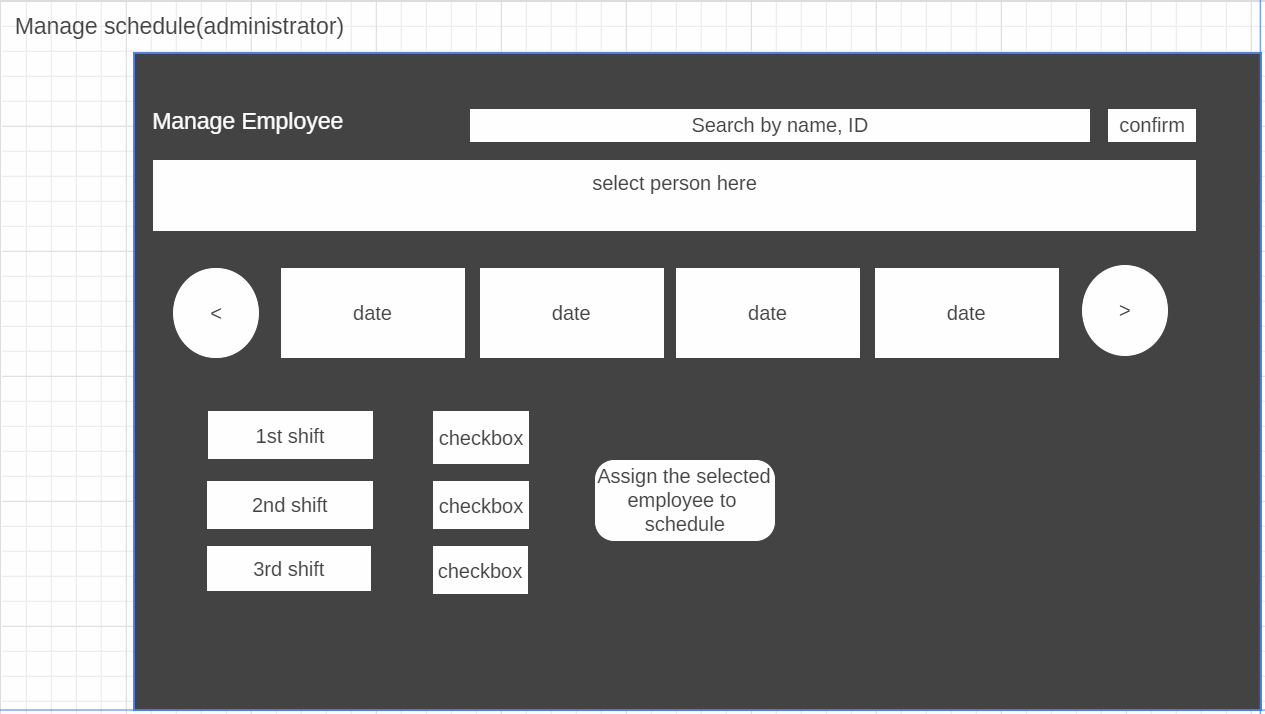
Use case: Log out clients application

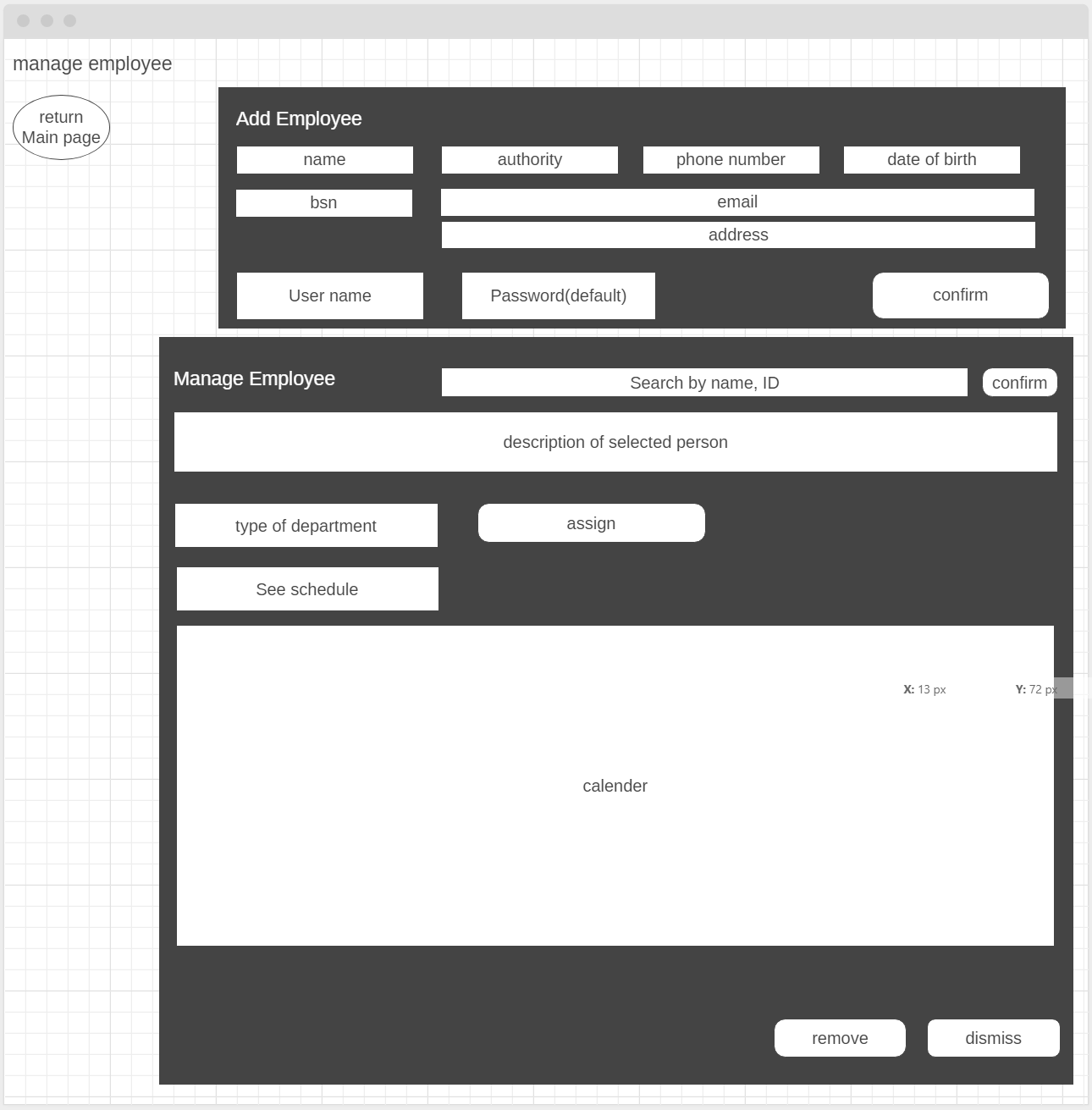
Actor: Any user

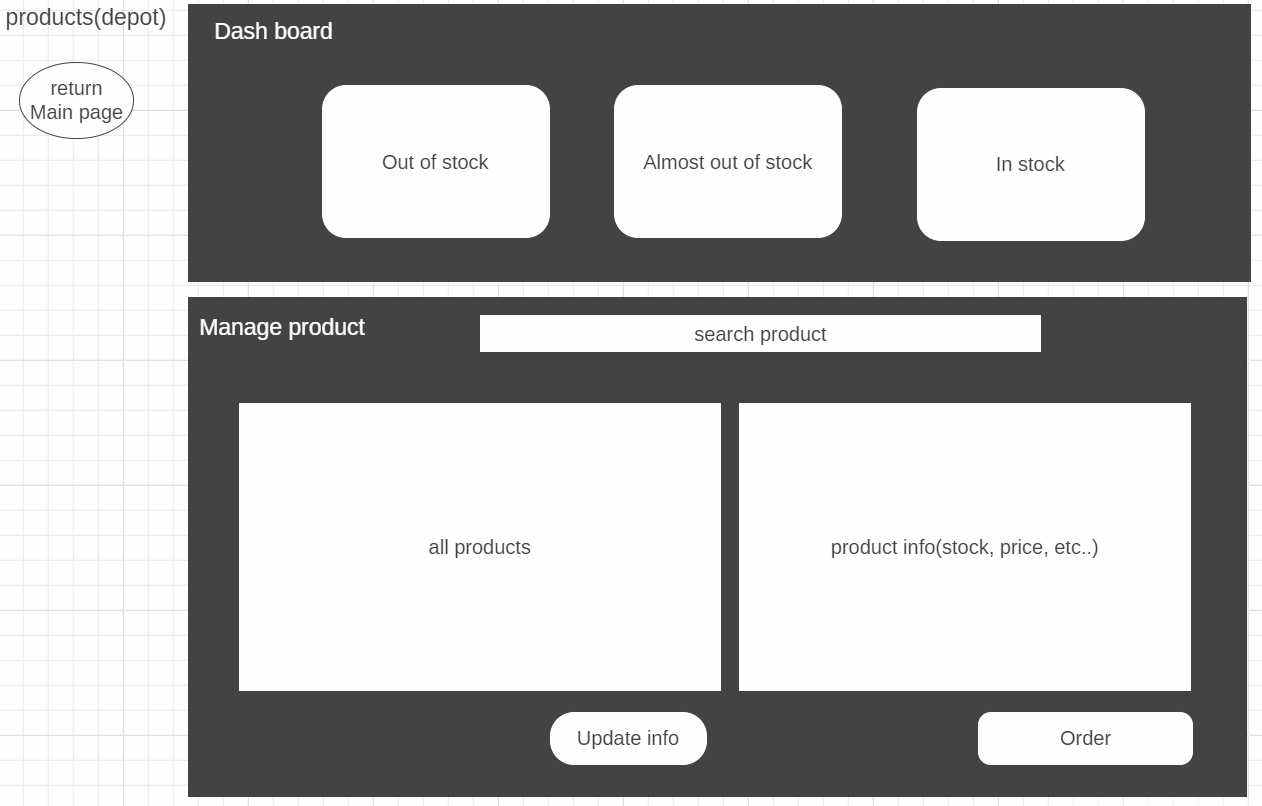
Main success scenario:

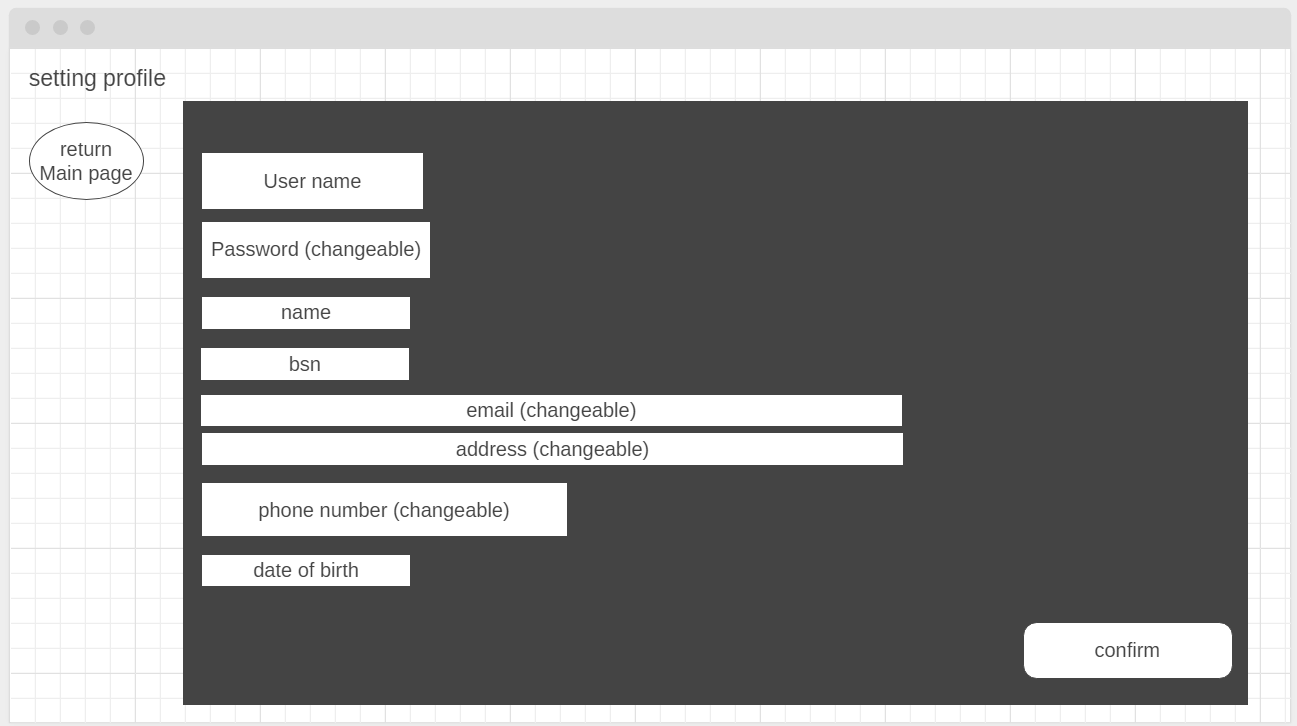
1. User presses the log out button
2. System logs out user out closing the current window and returning the user to the login window

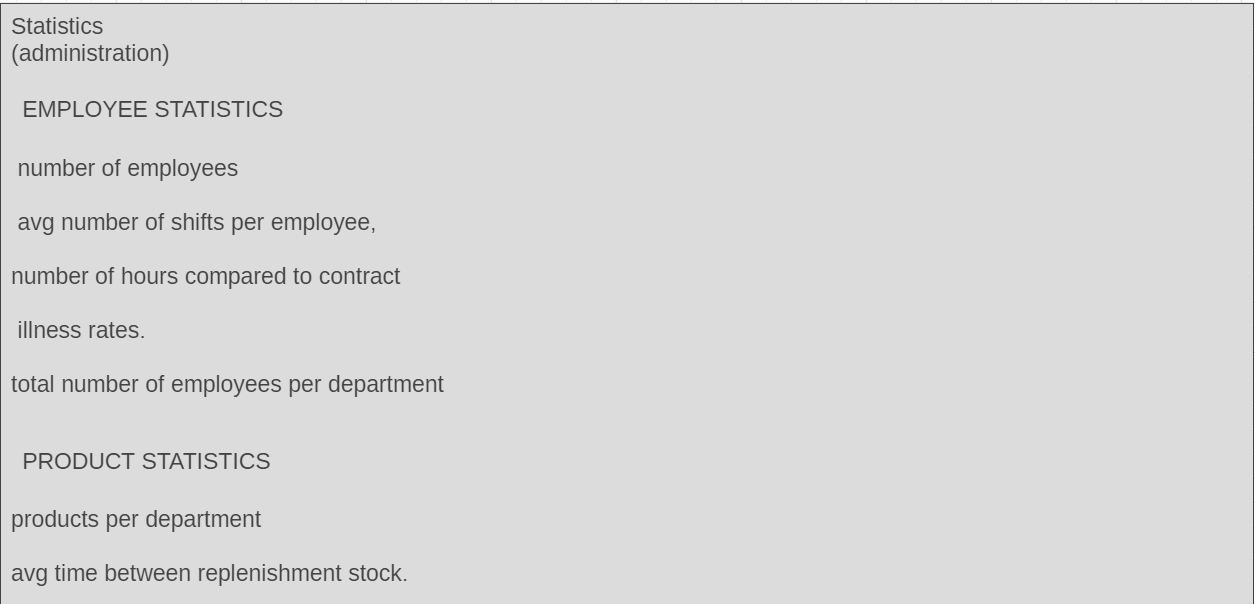
GUI:



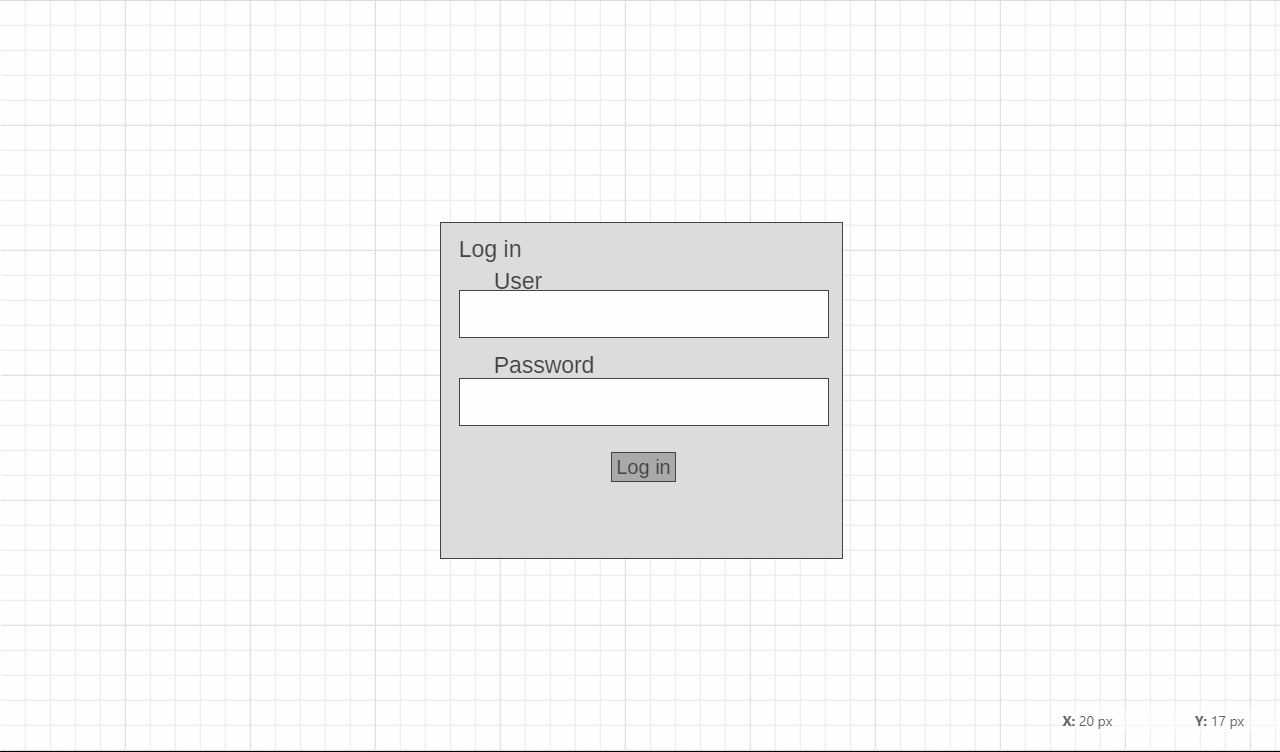








Website wireframe:

Main url to log in

Schedule page

