



PROGRESS REPORT – WEEKS #6, 7 (PCT 104)

Student Information

Student Name:	Maimunah Mohammed Hammad AL-Harbi	ID Number:	2180001941
Company Name:	T2 Company for Business Research and Development		

Progress Report Information: Brief description of activities, assignments, projects and type of training you were involved during each week of training, and problems faced with the resources used (Individuals, Books, and websites).

Tasks Done	Problems Faced	Resources Used
1- Implement server-side processing on jQuery Datatable. 2- Implement a side menu. 3- Attending a lecture about communication skills. 4- Writing System Requirement Specification (SRS) document for an HR System. 5- Attending a lecture about resumes and interviews.	None.	1- Microsoft Visual Studio 2019 2- Murugan, M. (2021, January 7). <i>JQuery Datatable in ASP.NET Core - Server-Side Processing</i> . Code With Mukesh. https://codewithmukesh.com/blog/jquery-datatable-in-aspnet-core/ 3- <i>How to Write a System Specification / Verasseti</i> . (n.d.). Verasseti. Retrieved July 7, 2021, from https://www.verasseti.com/insights/how-to-write-a-system-specification

Student MUST send this report to COOP supervisor and E-mail: training.ccsit@uod.edu.sa or Fax: 013-333-0257

1- Implement server-side processing of jQuery Datatable.

The jQuery Datatable is a plug-in for the jQuery JavaScript library that helps add advanced interaction controls to the HTML table. By implementing the server-side processing on jQuery Datatable, we will manage pagination, sorting, and searching the table. As shown in the figure below. Notice that the comments in green color explain the code in detail.

```
[HttpPost]
0 references
public IActionResult GetCourses()
{
    try
    {
        var draw = Request.Form["draw"].FirstOrDefault(); //The Number of times the API is called for the current datatable.
        var start = Request.Form["start"].FirstOrDefault(); //The count of records to skip. This will be used while Paging in EFCore
        var length = Request.Form["length"].FirstOrDefault(); //The page size. See the Top Dropdown in the JQuery Datatable that says, 'Showing n entries'.
        var sortColumn = Request.Form["columns[" + Request.Form["order[0][column]"].FirstOrDefault() + "][name]"].FirstOrDefault(); //The Column that is set for sorting
        var sortColumnDirection = Request.Form["order[0][dir]"].FirstOrDefault(); //Ascending / Descending
        var searchValue = Request.Form["search[value]"].FirstOrDefault(); //The Value from the Search Box
        int pageSize = length != null ? Convert.ToInt32(length) : 0;
        int skip = start != null ? Convert.ToInt32(start) : 0;
        int recordsTotal = 0;
        var courseData = (from tempcourse in context.Courses select tempcourse); //Gets an IQueryable of the DataSource
        if (!string.IsNullOrEmpty(sortColumn) && string.IsNullOrEmpty(sortColumnDirection)) //Sorting
        {
            courseData = courseData.OrderBy(sortColumn + " " + sortColumnDirection);
        } //Sorting
        if (!string.IsNullOrEmpty(searchValue)) //Searching. Here we will search through each column.
        {
            courseData = courseData.Where(c => c.Name.Contains(searchValue)
                || c.CreditHours.ToString().Contains(searchValue));
        } //Searching. Here we will search through each column.
        recordsTotal = courseData.Count(); //Gets the total count of the Records.
        var data = courseData.Skip(skip).Take(pageSize).ToList(); //Performs paging using EFCore
        var jsonData = new { draw = draw, recordsFiltered = recordsTotal, recordsTotal = recordsTotal, data = data }; //Sets the data in the required format and returns it.
        return Ok(jsonData); //Sets the data in the required format and returns it.
    }
    catch (Exception ex)
    {
        throw ex;
    }
}
```

After implementing the server-side processing in the controller part, we need to call the function that we implemented from the view part, specifically, in the JavaScript file that implements jQuery Datatable as circled in red in the figure below.

Also, there are many options as shown in the figure below such as “processing” to enable processing, “serverSide” to enable Side-server processing, “filter” to enable search box that we will see in the web interface later, “targets” which indicate that the number of target columns that we want to apply some options to them, in this case, we applied “visible” which indicate the visibility of column and “searchable” which indicate the ability of search on them using the search box, Finally, the “columns” indicates the columns of the table and we can add some options that are self-explanatory as shown in the figure below.

```

$(document).ready(function () {
    $('#CoursesTable').dataTable({
        "processing": true,
        "serverSide": true,
        "filter": true,
        "ajax": {
            "url": "/Courses/GetCourses",
            "type": "POST",
            "datatype": "json"
        },
        "columnDefs": [{
            "targets": target,
            "visible": false,
            "searchable": false
        }],
        "columns": [
            { "data": "id", "name": "ID", "autoWidth": true },
            { "data": "name", "name": "Name", "autoWidth": true },
            { "data": "creditHours", "name": "Credit Hours", "autoWidth": true, "orderable": false },
            {
                "render": function (data, type, row) {
                    return "<button onclick='location.href = \"/Courses/Edit/\" + row.id + \"'\" class='btn btn-outline-primary'>Edit</button>"
                        + " <button onclick='location.href = \"/Courses/Delete/\" + row.id + \"'\" class='btn btn-outline-danger'> Delete</button> ";
                },
                "searchable": false, "orderable": false
            }
        ]
    });
});

```

So now let's see the table on the website.

Courses

Create New

Show 10 entries

Search:

Name	Credit Hours	Action
Data Security	3	<button>Edit</button> <button>Delete</button>
English	7	<button>Edit</button> <button>Delete</button>
Electronics	4	<button>Edit</button> <button>Delete</button>
Economic System in Islam	2	<button>Edit</button> <button>Delete</button>
Social System in Islam	2	<button>Edit</button> <button>Delete</button>
Data Mining	3	<button>Edit</button> <button>Delete</button>
Physics	4	<button>Edit</button> <button>Delete</button>

We can sort in ascending or descending according to "Name" column.
In the case of sorting ascending, the result is as shown in the figure below.

Courses

Create New

Show 10 entries

Search:

Name	Credit Hours	Action
Biology	4	Edit Delete
Computer Organization	3	Edit Delete
Data Mining	3	Edit Delete
Data Security	3	Edit Delete
Data Structure	2	Edit Delete
Economic System in Islam	2	Edit Delete
Electronics	4	Edit Delete

And vise versa.

Courses

Create New

Show 10 entries

Search:

Name	Credit Hours	Action
Data Mining	3	Edit Delete
Data Security	3	Edit Delete
Data Structure	2	Edit Delete

Also, we can search for any value of any column to filter only those rows that contain that value. For example, in the figure below we searched about the names of the courses that contain “data”.

Courses

localhost:44353/Courses/Index

Sign out

Courses

Create New

Show 10 entries

Search: data

Name	Credit Hours	Action
Data Structure	2	Edit Delete
Data Security	3	Edit Delete
Data Mining	3	Edit Delete

Showing 1 to 3 of 3 entries

Previous 1 Next

Copyright © 2021, School

And in the figure below, we searched about the courses with 2 credit hours.

Courses

localhost:44353/Courses/Index

Sign out

Courses

Create New

Show 10 entries

Search: 2

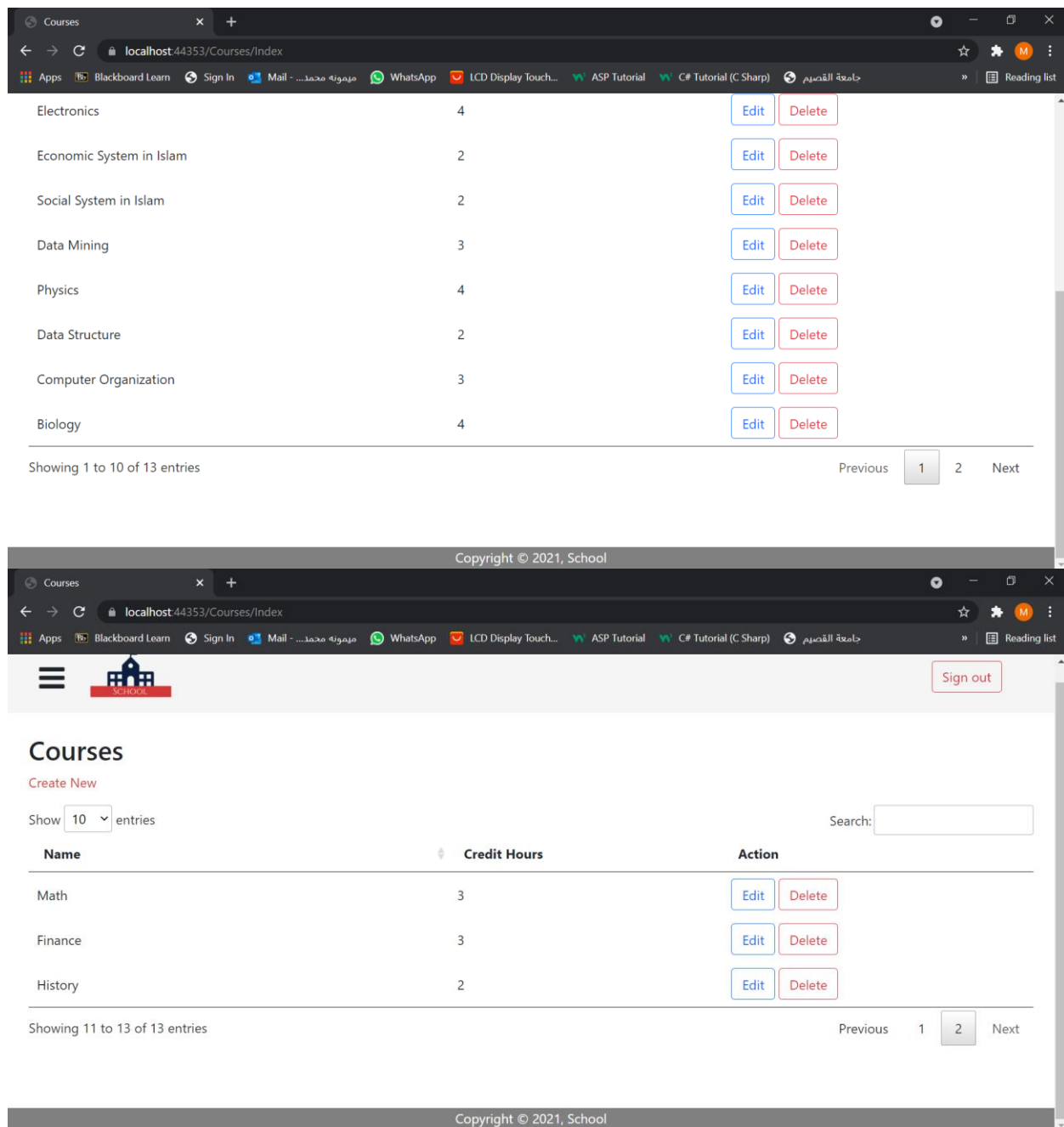
Name	Credit Hours	Action
Economic System in Islam	2	Edit Delete
Social System in Islam	2	Edit Delete
Data Structure	2	Edit Delete
History	2	Edit Delete

Showing 1 to 4 of 4 entries

Previous 1 Next

Copyright © 2021, School

And we can page from one set of data to another as shown in the figures below.



2- Implement a side menu.

We implemented a side menu instead of a top menu, and to make it open and close at any point while navigation the website, we should tell all the pages about the status of the side menu recently whether it was opened or closed, and we did that using cookies as circled in red in the figure below.

```

function setCookie(cname, cvalue, exdays) {
    var d = new Date();
    d.setTime(d.getTime() + (exdays * 24 * 60 * 60 * 1000));
    var expires = "expires=" + d.toUTCString();
    document.cookie = cname + "=" + cvalue + ";" + expires + ";path=/";
}

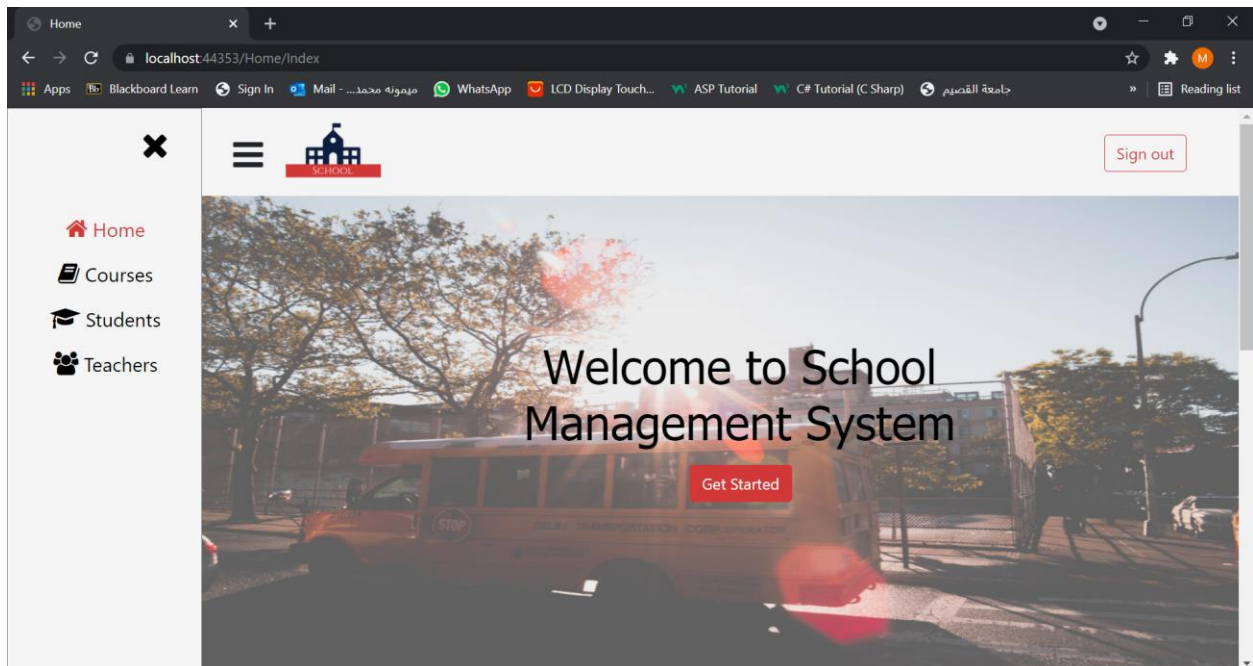
function getCookie(cname) {
    var name = cname + "=";
    var ca = document.cookie.split(';');
    for (var i = 0; i < ca.length; i++) {
        var c = ca[i];
        while (c.charAt(0) == ' ') {
            c = c.substring(1);
        }
        if (c.indexOf(name) == 0) {
            return c.substring(name.length, c.length);
        }
    }
    return "";
}

function openMenu() {
    setCookie("sidebar", "opened", 30);
    location.reload();
}

function closeMenu() {
    setCookie("sidebar", "closed", 30);
    location.reload();
}

```

So, we can navigate between pages smoothly as shown in the figures below.



Courses

localhost:44353/Courses/Index

AppsBlackboard LearnSign InMail - ميمونه محمد...WhatsAppLCD Display Touch...ASP TutorialC# Tutorial (C Sharp)جامعة القصيمReading list

×

Home

Courses

Students

Teachers

×

SCHOOL

Sign out

Courses

Create New

Show 10 entries

Search:

Name	Credit Hours	Action
Data Security	3	Edit Delete
English	7	Edit Delete
Electronics	4	Edit Delete
Economic System in Islam	2	Edit Delete
Social System in Islam	2	Edit Delete

Students

localhost:44353/Students/Index

AppsBlackboard LearnSign InMail - ميمونه محمد...WhatsAppLCD Display Touch...ASP TutorialC# Tutorial (C Sharp)جامعة القصيمReading list

×

Home

Courses

Students

Teachers

×

SCHOOL

Sign out

Students

Create New

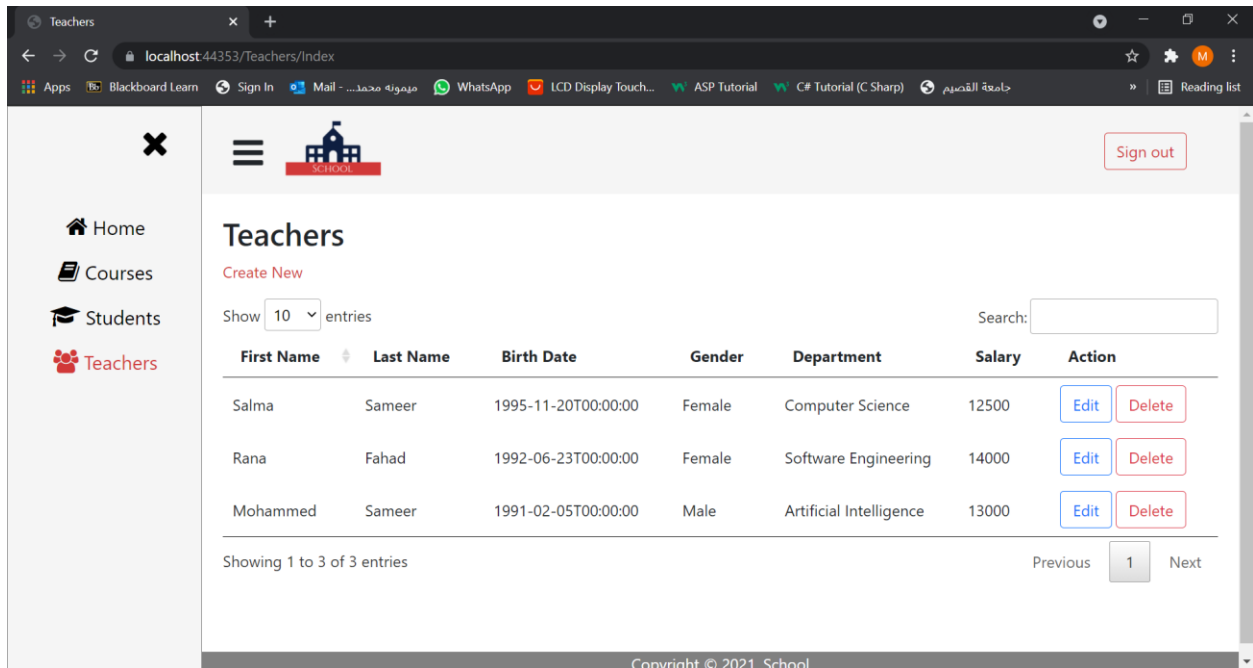
Show 10 entries

Search:

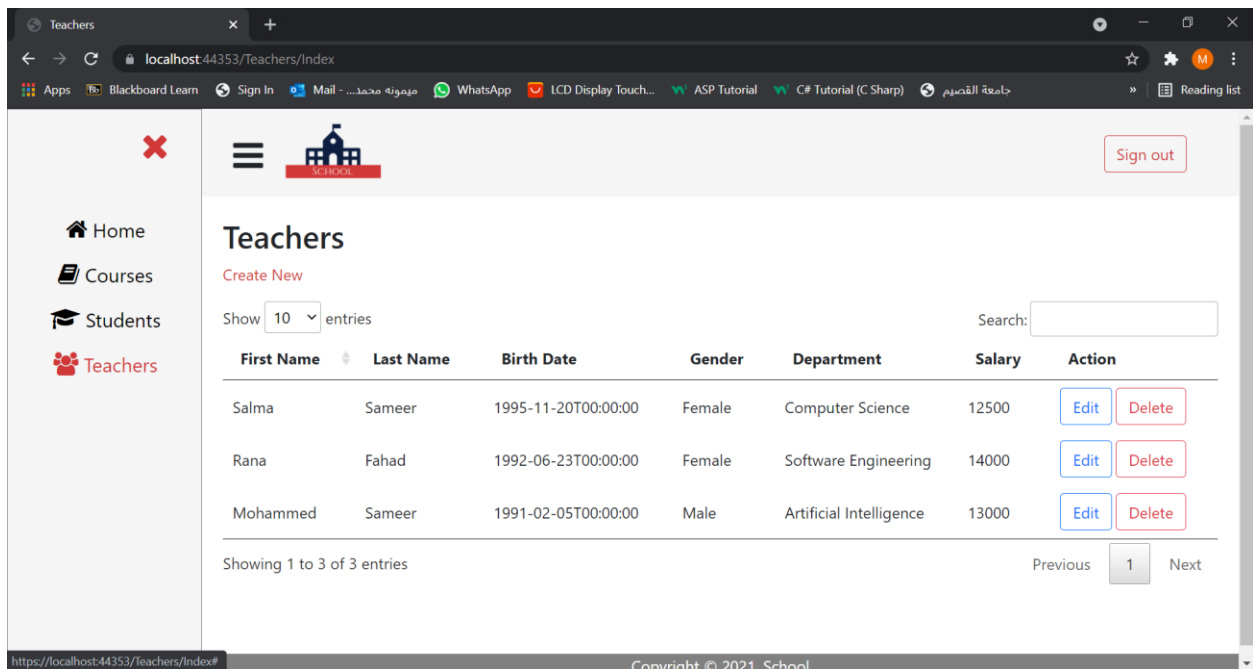
First Name	Last Name	Birth Date	Gender	Enrollment Date	Major	Status	GPA	Action
Noura	Mohammed	1999-08-18T00:00:00	Female	2017-06-06T00:00:00	Computer Science	Active	4.52	Edit Delete
Salma	Sameer	2000-12-19T00:00:00	Female	2017-06-06T00:00:00	Software Engineering	Active	4.32	Edit Delete

Showing 1 to 2 of 2 entries

Previous1Next



And now, we can close it and move to the home page by clicking on the website logo while it's still closed as shown in the figures below.



Teachers

localhost:44353/Teachers/Index

Sign out

Teachers

Create New

Show 10 entries

Search:

First Name	Last Name	Birth Date	Gender	Department	Salary	Action
Salma	Sameer	1995-11-20T00:00:00	Female	Computer Science	12500	Edit Delete
Rana	Fahad	1992-06-23T00:00:00	Female	Software Engineering	14000	Edit Delete
Mohammed	Sameer	1991-02-05T00:00:00	Male	Artificial Intelligence	13000	Edit Delete

Showing 1 to 3 of 3 entries

Previous 1 Next

https://localhost:44353/Home/Index

Home

localhost:44353/Home/Index

Sign out

Welcome to School Management System

Get Started

3- Writing System Requirement Specification (SRS) document for an HR System.

[Click Here](#) to move to the SRS document.

System Requirement Specification

HR System

Introduction

This HR system is responsible for managing the vacations and leaves requested by the employees of an organization, it will automate the operations of submitting and responding to vacation requests which will contribute to reducing the costs of papers consumed for these kinds of requests, and reducing the time required to submitting these requests and receiving the responses to it.

Description

Users of this system will be the following:

- 1- Employee
- 2- Manager

In addition, a database server is needed to implement this system.

Furthermore, this system has some constraints which are the following:

- 1- All users have to log in to the system.
- 2- The employee can submit his vacation request to his manager only.
- 3- When the employee requests a vacation, the following data fields are required:

3.2- Vacation Type, where the accepted types are the following:

- 3.2-1. Annual
- 3.2-2. Sick
- 3.2-3. Leave
- 3.2-4. Exceptional

3.3- The duration of the vacation

In addition, the “attachment” field is optional.

4. The manager can add a new employee who is under his management only. Similarly, he can preview the list of the employees who are under his management only.
5. When the manager adds a new employee, the following data fields are required:

- 5.1- Name
- 5.2- Mobile Number
- 5.3- Email Address
- 5.4- Job Title

6. The manager can preview the list of vacation requests for employees who are under his management only.
7. When the manager rejects a pending vacation request, he must enter a reason for rejection.

User Interfaces

1- Shared Interfaces:

1.1- Login Page

2- Employees Interfaces:

2.1- The employee information page

2.2- The vacation request page

2.3- The vacation requests list page

3- Manager Interfaces:

3.1- User Management:

3.1-1. Adding new employee page

3.1-2. The employees' list page

3.2- Vacation Request Management:

3.2-1. The vacation requests list page

Functional Requirements

1- Shared Functional Requirements:

1.1- Logging in to the system

2- Employees' Functional Requirements:

2.1- The ability of the employee to review his information such as his name, mobile number, email, and his manager information.

2.2- The ability of the employee to request a new vacation.

2.3- The ability of the employee to save the vacation request as a draft to edit it later or to submit it to his manager.

2.4- The ability of the employee to review all his requested vacations with their statuses, and filter on a specific vacation.

3- Managers' Functional Requirements:

3.1- User Management

- 3.1-1. The ability of the manager to add a new employee.
- 3.1-2. The ability of the manager to review the list of the employees.

3.2- Vacation's Request Management

- 3.2-1. The ability of the manager to review the list of the vacation's requests, and filter on specific vacation by date, employee name, duration, or type.
- 3.2-2. The ability of the manager to approve or reject the pending vacation's requests.

Non-Functional Requirements

1- Security

All users should log in to the system to be allowed to perform only the operations they are authorized to.

2- Integrity

Whenever some users update some data, it will be updated to all users interact with the same data. For example, when the manager updates the status of the vacation request of an employee, the employee will preview his request with updated status by the manager.

3- Usability

The user interfaces are designed to be familiar to the knowledgeable intermittent users who use other different systems which is the minimum expected classification for users of such a system. So, the time expected to be spent learning this system will not exceed one hour.

4- Availability

In normal circumstances, the system must be available to use 24/7

5- Robustness

In case of receiving any unacceptable inputs, the system will handle it by showing a message error to the user prompting to enter a correct input.

Arwa Alruhaili

