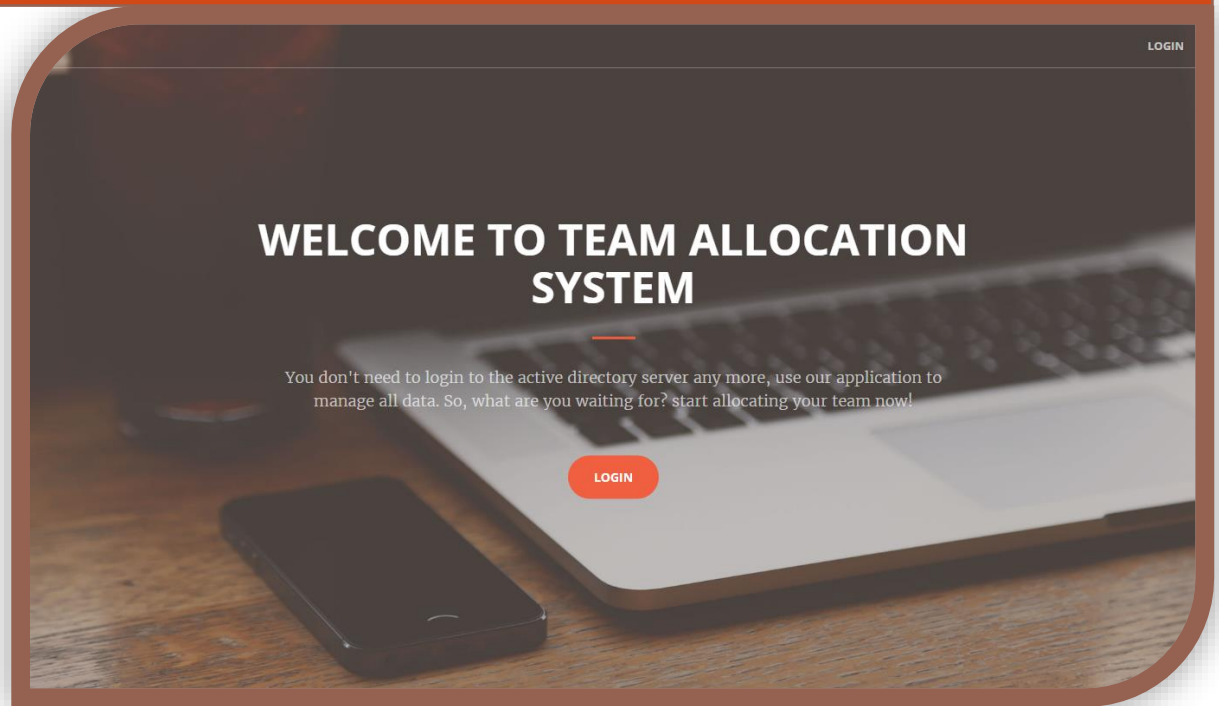


Team Allocation System



By:

Manal Lubbadah & Loay Zawyani

Supervised by:

Dr. Ashraf Armoush & Dr. Haya Sama'aneh

Engineering & Information Technology Faculty –Computer Engineering Department
An-Najah National University – Nablus .Palestine

Presented in partial fulfillment of the requirements for Bachelor degree in Computer
Engineering.



Acknowledgment

We thank our God (SWT), our parents for the big support, love and care they gave us all the time, and for always believing in us and trusting our judgment.

We also thank Dr. Ashraf Armoush and Dr. Haya Sama'aneh our supervisors, who gave us a big motivation every time we felt down. We would also like to thank our teachers and friends.

Special Thanks for ITG –Infinite Tiers Company for their support and supervision and keep updating us with the requirements.

Disclaimer Statement

This report was written by student(s) at the Computer Engineering Department, Faculty of Engineering & Information Technology, An-Najah National University. It has not been altered or corrected, other than editorial corrections, as a result of assessment and it may contain language as well as content errors. The views expressed in it together with any outcomes and recommendations are solely those of the student(s). An-Najah National University accepts no responsibility or liability for the consequences of this report being used for a purpose other than the purpose for which it was commissioned.

Table of Contents

Acknowledgment	1
Disclaimer Statement	2
Table of Figures	4
Nomenclature Symbols	6
Abstract	7
Chapter 1: Introduction	1
1.1 Statement of the problem:	1
1.2 Objectives and scope of work:.....	2
1.3 Significance of our work:.....	2
1.4 Organization of the report:.....	2
Chapter 2: Constraints	3
2.1 Constraints:	3
2.2 Standards/Codes:.....	3
Languages:	4
Plugins & Libraries:.....	4
2.3 Earlier course work:	4
Chapter 3: Literature Review	5
Chapter 4: Methodology	6
Chapter 5: Results and Discussion	11
5.1 Features in admin section:.....	12
5.1.1 Mapping:	13
5.1.2 Reports:.....	22
5.1.3 Custody Management:.....	25
5.1.4 Project Management:	28
5.2 Features in normal employee section:	36
5.3 Brief description of all features:	39
5.4 Database Class Diagram:.....	40
Chapter 6: Conclusion	41
References	42
Appendices	43
Appendix A:.....	43

Table of Figures

Figure 1: IP's Configuration.....	6
Figure 2: Start Page.....	11
Figure 3: Login Page.....	12
Figure 4: Admin Home Page	12
Figure 5: Add New Floor Form	13
Figure 6: Floor List	14
Figure 7: Floor Image	14
Figure 8: Zooming and Moving of Image.....	15
Figure 9: Add Multiple Seats	15
Figure 10: Save Added Seats	16
Figure 11: Assign Employee to a Seat	16
Figure 12: Auto Suggestions Feature.....	17
Figure 13: Assign Employee Successfully	17
Figure 14: Employee Information Sidebar	18
Figure 15: Multiple Delete of Employees from Floor Plan	18
Figure 16: Edit Employees Locations on Floor Plan	19
Figure 17: Search by City	19
Figure 18: Search by city result	20
Figure 19: Search by Department	20
Figure 20: Search by Username	21
Figure 21: Search for Mapped Employee	21
Figure 22: Search for Unmapped Employee.....	22
Figure 23: Reports Page.....	22
Figure 24: Department Report Table	23
Figure 25: Update A Field Value in Report Table.....	24
Figure 26: Change Employee's Department in Reports.....	24
Figure 27: PDF Report.....	25
Figure 28: Normal Assets Management	26
Figure 29: Computers Custody Management	26
Figure 30: Add New Asset.....	27
Figure 31: Edit Asset Information	27
Figure 32: Multiple Delete of Assets	28
Figure 33: Project Management.....	29
Figure 34: Display Department Projects.....	29
Figure 35: Add New Project	30
Figure 36: Add Tasks to Project	30

Figure 37: Project Tasks Status Chart.....	31
Figure 38: Assign Employee to a Task	31
Figure 39: Assign Employee With Tasks Less Than The Max.	32
Figure 40: Employee Can't Be Assigned to Task	32
Figure 41: Assign Free Employee.....	33
Figure 42: Delete Assigned Employee	33
Figure 43: Change Maximum Allowed Assigned Tasks	34
Figure 44: Generate Report Form	34
Figure 45: Generated Report Table.....	35
Figure 46: Generated Report Line Chart	35
Figure 47: Normal Employee Home Page	36
Figure 48: Normal Employee Information Viewing Privileges.....	37
Figure 49: Normal Employee Assets	37
Figure 50: Normal Employee Projects.....	38
Figure 51: Daily Log Form	39
Figure 52: Database Class Diagram.....	40
Figure 53: Determine the Path of Domain Code	43
Figure 54: Validate User Credentials Code	44
Figure 55: Check Login Code.....	45
Figure 56: Download PDF file Code	47

Nomenclature Symbols

LIST OF ABBREVIATIONS:

ITG Company: Infinite Tiers Group Company, Software Company in Nablus-Palestine.

JPEG: Joint Photographic Experts Group.

MVC: Model-View-Controller, Software design pattern.

PDF: Portable Document Format.

PNG: Portable Network Graphics.

VM: Virtual Machine.

Abstract

Many organizations have problems with team allocation and management. Usually, the admin needs to login to the server to be able to view and manage the staff information. Also, the employees can't see each other location in the building or general information about other users. Moreover, all companies needs a system which applies project and assets management. Our project solves these problems.

Team Allocation System was created using visual studio 2015 framework and windows server 2008 R2. The server was installed on a virtual machine and converted to a domain controller by activating active directory service on it to use it as the main source of employees' information.

Then, it was connected to the application by configuring the IP's of the server and the host PC. The project structure was build based on the concepts of ASP.NET/MVC-Model-View Controller applications. The main programming language used was C#. Some Libraries and plugins were also used to enhance the functionality of the project.

The website has two sources of data; Active directory server and SQL Database which differentiates it from other websites in its field since the active directory server is somehow hard to understand and deal with. The result was a secure and user friendly website with responsive design which matches the needs of many nowadays companies in the field of management.

Team Allocation System provides many useful features which are in short: Mapping, custody management, dynamic reports and project management.

Chapter 1: Introduction

1.1 Statement of the problem:

Companies nowadays trends towards using management systems to manage their information easily using user friendly applications. In general, these applications are specialized in specific fields. Majority of information management systems are specialized in project management and allocation of tasks to employees. And even in most mapping systems, the information needs to be inserted by the user despite that most of companies uses domain controllers to store it.

Administrators in companies are obliged to use servers-which are often domain controllers- to manage employees' information. These servers don't have user friendly design and it is complicated to deal and interact with them. So, we noticed that they need an application with user friendly design to view all the information and manage it in addition to mapping all the employees on the company's floor plans to facilitate reaching them.

Other side of the problem, the servers don't provide any reports about employees. However, reports are very useful in collecting and merging data and making management decisions. Also, any company would be very interested in managing its assets and keep track of them, which ones are in stock? Which ones are assigned to employees? What are the details about an asset? Or how many the company have of it? Answering all these questions is important to make decisions especially financial type of them.

In most outsourcing companies the employee logs his daily work on the system of the company he works with, but his original company doesn't have detailed information about his work and how much he works daily and his daily effort. These details are very important to calculate his progress and decide if he works under estimated periods of work or not.

Taking the side of normal employees in a company. In companies that don't have mapping systems, employees can't see their colleagues location in the company and there information. New concepts in work states that the communication between the company employees is very important. In large companies, it is hard for you to see other employees' locations or information so you may face difficulty in introducing to others especially if you are new to the company.

1.2 Objectives and scope of work:

Our objective is to produce a system which is a combination of solutions for all the previous problems in a secure and user friendly application. We aim to provide a system which satisfies most of information management needs.

Team Allocation system includes mapping of employees' to floor plans and viewing their information which retrieved from the domain controller. Also, these information can be viewed and edited in the form of dynamic reports. The application includes custody management in the scope of adding, editing, deleting and assigning these assets to employees but without being able to map them on the floor plans as like as employees. The project management part is not as wide as other systems which are specialized in this field, its scope of work is limited in adding projects, adding task to each project and assigning employees to these tasks based on some concepts. But, the application strongly resolves the problem that the admin now can view detailed information about the employee's progress during specified period using tables and charts.

1.3 Significance of our work:

Team Allocation system can be mainly used in the field of business. Actually, the basic idea of this system came from ITG-Infinite Tiers Group Company when they explained to us the difficulties they face in managing their employees information which we stated before in the problem statement section. This really proves the importance of Team Allocation system for these companies and its big role in facilitating many operations.

1.4 Organization of the report:

In this report, we illustrate the idea in its all details. Going through the report you can first see the constraints we faced during our work including any codes and earlier course work. Then, you can read about similar systems to get a background about the topic being discussed and what special features we have done upon other systems. After that, the methodology of our work is extensively explained. Next chapter states our results and a discussion to interpret and compare the results. Ending it up with the conclusion of the whole work and what is our vision for future to improve our work.

Chapter 2: Constraints

2.1 Constraints:

During working on our project, we faced a lot of constraints that we didn't expect. What are the constraints? And how we faced them?

Our first problem was downloading windows server 2008 R2 on our PCs and configuring the active directory server on it in order to convert it to a domain controller. We faced this problem when we started working on our project and it took from us nearly 2 week to solve it by reading and learning from different sources and tried a lot of times and finally the server was installed on a virtual machine successfully with the help we got from ITG Company.

Problems didn't stop to this line. The next constraint we faced was creating connection between the PCs and the domain controller. We searched about it on the internet but it wasn't really successful. So, we got help from network lab administrator Miss Tasneem and she helped us to do it by making some changes in the IPs of the VM and our PCs.

These challenges were the biggest criteria during our project because we didn't have any experience about them. After that, we started working using ASP.NET/MVC and somehow we have good experience in this field and it has a very wide documentation on internet. So, next challenges were not as big as the earlier ones.

2.2 Standards/Codes:

A lot of software engineers these days may use an already written code in their projects for saving time instead of writing the code from the beginning which is already written by others.

In our project we used some already written codes from the internet and ITG Company that helped us during our work and saved a lot of times and effort.

- **Connect the project with active directory:** we used an already existing code from the company to connect our application to the active directory that exists on our virtual machine. After that, we made some changes to the code to be applicable with our project and do exactly what we want. (refer to Appendix A)

- **Download information from a web page as PDF:** here we searched on internet and used an already existing code to be able to save some data as PDF using ITextsharp (iTextSharp, 2016) library. In this code, we made a lot of changes to have the format we want but we left the main code that is responsible for downloading the information as it is. We used this to download the report information from the Reports page as PDF file. (refer to Appendix A)

Languages:

C# /ASP.NET MVC, JavaScript, CSS, JQuery, Ajax, JSON, Bootstrap, MySQL.

Plugins & Libraries:

- **Tables:** DataTable (DataTables Table plug-in, 2007-2016), treefy (TreeFy Tables, 2012-2016)
- **Zooming:** PanZoom (Willison, 2016)
- **Charts:** CanvasJS (Charts, 2013), morris (Smith, 2013)
- **Start Page:** Bootstrap Start themes. (Startbootstrap, n.d.)
- **Download As PDF:** ITextsharp (iTextSharp, 2016)

2.3 Earlier course work:

Working on our project depends on many courses we learned from university such as Object Oriented Programming, Software, Web and Database. The course with biggest effect we learned from is the training course when we joined .NET team at ITG Company and learned so many things like Bootstrap, jQuery, AJAX, C#, ASP.Net, ASP.NET/MVC and the way to write good, clean and readable codes.

Chapter 3: Literature Review

StaffMap website has many features which are similar to our system. The main idea is to manage employees in the company's floorplans. It has many features, like dynamic floor plans which enables multi user simultaneous editing and plan employee movement scenarios. In addition, it is a seating planner and desk and equipment booking on floor plans. Also, it is mobile friendly, you can book office space from mobile devices. It has track remote assets which involves google maps integration to keep track of remote workers by their geolocation. Furthermore, the website gives you the opportunity to search for employees using the quick search tool to view their details which they present on floor plans including their contact information, records and some other custom details, these details are inserted by the website, they don't use a server to get the details of an employee. Finally, the last feature is fully customizable and resizable embedding of floor plans with iframe. (BrowseLab, 2016)

Bitrix24, is a project management and tasks system which is specialized in this field but doesn't contain the team mapping features. This system goes deeply into the field of project management and allocating tasks. It has Gantt chart and task dependencies. Furthermore, manages the employee's workload in tasks and generates monthly reports on time spent on tasks for each department and each employee like the daily log feature in Team Allocation System. It has a report builder which lets you collect the data fields needed and evaluate effectiveness, work efficiency, and total time spent on projects and tasks, and many other features. It's a strong website in the field of project management. (Bitrix, 2001-2016)

Most of systems which are in the field of mapping and management of information are specific either in mapping or project management and tasks allocation, but Team Allocation System is a combination of mapping, dynamic and downloadable reports, assets management and project management. Moreover, in most mapping and employees' information management systems the admin has to insert the information about each employee manually even very specific data like employee ID. Team Allocation System gets all these information from the domain controller-active directory server dynamically in a very user friendly and responsive design which we didn't find in other systems.

Chapter 4: Methodology

We used Microsoft Visual Studio 2015 Enterprise program for coding our project and used the VMware Workstation 12 Player to carry Windows Server 2008 R2 that includes the active directory. Also, we used ADPhotoEdit (AD PHOTO, 2012) program to manage images in active directory for each employee. In addition, we used Microsoft SQL Server Management Studio to manage our SQL Database. To convert the floor plans we took from ITG Company from PDF to a high resolution pictures in the format of PNG, Wondershare PDFelement (PDFEditor, 2007-2016) program was used.

First step in our project was downloading VMware Workstation 12 player. Then, we downloaded Windows Server 2008 R2 and installed it by going through a set of steps we took them from ITG Company (Francis, 2013). After installing the server we activated active directory feature on it to convert it to a domain controller (Shinder, 2009). We created a domain named “it.local” and an organizational unit named “Employees” and we created users in it and filled their information with real data. In order to integrate the server with the application, the PC should be connected to the domain we created. This was done by configuring the IP’s of the PC and the server.

Configuring the IP’s was as follows; on the PC side we set a static IP for the network adapter of the VMware, and on the VMware side we also gave it a static IP. Notice that they must have the same IP of the DNS server which is the same of the static IP of the server, the DNS feature should be activated on the server. This means that the PC which uses the website should be connected to the server, in C# code LDAP connection should be created.

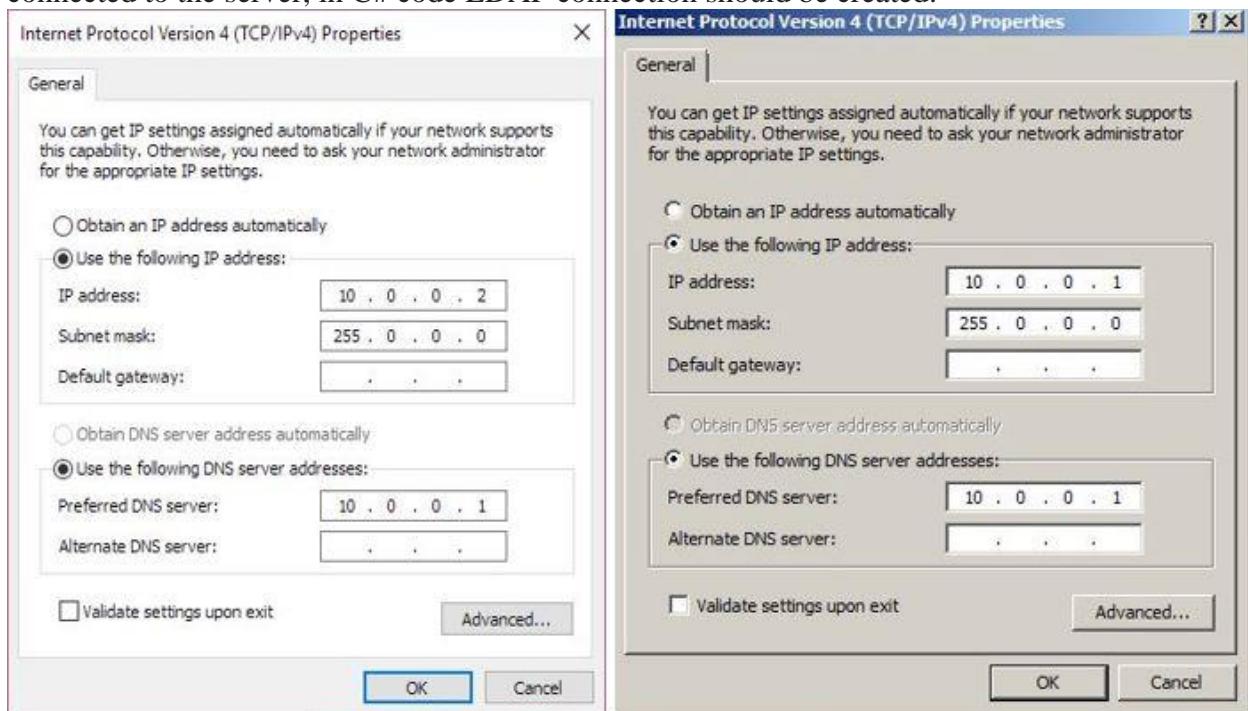


Figure 1: IP's Configuration

All previous work was done in two weeks of hard work. On the other side, we created a MVC project. All the pages were designed by us without using ready themes, we used Bootstrap3, HTML and CSS for design. And for retrieving data we used JSON. So that, there isn't any data sent using GET requests, you can't see any data in the URL of the page.

Then the flow was like this:

- We started by creating the login page which includes two fields for username and password. This page gives the ability to login as admin or as normal employee. Their credentials are validated from the server and any error in the connection to the server or the credentials is shown as an error message to the user.
- We created the SQL Database using Microsoft SQL Server Management Studio.
- Next we started with admin section by creating the admin page which includes too many fields:
 - First thing we did at admin page was creating the design for the page and creating Add Floor Tab which enables the admin to name the floor and upload the image for it in PNG or JPEG format and finally save these information.
 - Next we wrote the code for displaying the existing floors names in a list which gives the floor image when clicking on it. Also, we added zoom in and out to the image using two buttons and the ability to change the image location to any other location using drag and drop. We used Panzoom library for this feature
 - We added a new tab that is responsible of adding chairs at any location on the floor plan where the employees can be assigned.
 - After that, we worked on the search tab. We made three filters to search based on them which are username, city and department. We worked on auto complete while typing in the search box by showing some suggested data from the active directory.
 - Next, we worked on assigning an employee to a chair that exists on the floor plan by typing the name of the employee in the box and it will give auto suggested names from the active directory with button to book him to the chair.
 - Back to the search functionality. We edited the username filter that when the admin selects a username the website shows his position on the floor plan marked with red and a side bar with all of his information retrieved from the active directory and his assets.
 - Next we added another tab to edit employees and chairs that exist on the floor images. This tab provides the ability to change their locations using drag and drop and also delete multiple chairs and multiple employees at once.

Note that the floors and the employees' positions are stored in the database.

Next Section we worked on in the admin side was reports:

- First, we created the design of the page as one big table and inside it collapsible tables. But, this design wasn't efficient when we decided to make the reports dynamic.
- The design was changed to be more effective by dividing it into sections each section for a department and each department shows a table of the chosen department employees with all their information with the leader for each department.
- We added the edit functionality to the reports which enables the user to edit any field in the report just by clicking on it and typing the new value and then submit it. These updates are reflected directly on the server.
- ITG Company asked us to add more details about each employee. These details were not standard fields in the active directory. So, we searched about that and found that the active directory provides a functionality for adding custom fields. We learned the way of making them and added all the required fields (Morgan, 2013).
- Another functionality was added later to the reports. Any department information can be downloaded as a PDF file. We used ITextsharp for this.

Note that all the information in this section are retrieved from active directory even the departments. Which means that if a new department is added to the active directory, this page will dynamically add a tab for it.

Moving to the custody management section which is also accessible by an admin:

- We created this page with two tabs. One for the general assets like mouse, keyboard, chair...etc. The second tab is for computers since their details varies from other assets. These information are interpreted using Datatable plugin which provides search, sorting of data and pagination.
- The information for all assets were stored in the database.
- Here we faced a problem with adding the features of edit, add and delete records from Datatable while keeping it dynamic without the need to refresh the page. The Datatable website has another plugin called Editortable which provides these features. But, we decided to make our editable Datatable. We used a partial view in each tab and added add and edit buttons. And check boxes for multiple delete. Fortunately, it works properly in a well-designed way.
- When the asset is added it can be assigned to a user or if it is not assigned it will take a default value which indicates that it is still in stock and it can be assigned later to any user by editing it.

In parallel to this we were working on the normal user side:

- We started by making the main page for a normal user which interprets the floor plan when the user chooses one. All booked seats and users by the admin are shown. And the user can see other users' information but not as like as the information provided to the admin.
- Second step was the user's assets tab which we interpreted in it the assets for the logged in user in a Datatable.

At this point we ended all the features provided in the proposal provided at the beginning. But, there was a month and half remaining to the end of semester. So, we started thinking about new features to add. We consulted about that our supervisors and ITG Company and their employees. After thinking and searching during two weeks we decided to add Project Management feature with daily log for the user using the database to store the information.

- At first we constructed the idea which was that a user fills a form of his daily work by creating a task and filling its all information and how much he worked on it. And on the admin side, the admin can generate a report about any user to see his logs. This scenario was not really effective as our supervisors told us. So, we modified it.
- At the admin side, we created a page which shows all departments and for each department a table that contains all its projects represented by a Datatable.
- We made the add project feature to add a project and specify its start and end date.
- After that, we made add task functionality. By this functionality the admin can add tasks to the chosen project and specify its start and end date within the boundaries of the start and end dates of that project, this was done using JQuery Datepicker.
- Next, we made assign employees button. When it is clicked, a modal is shown so that the admin chooses a task from a select list which contains the tasks which are not finished yet. Then, we made a text box for the admin to type in it the username of the employee he wants to assign to that task, while typing suggestions of related names are shown.
- Beside each name from the suggestions we made a button named "Show table", and when it is clicked a table is shown. This table shows the tasks assigned for the chosen employee and their start and end dates which enables the admin to determine where this employee has an empty slot of time for the task and if he decided to assign the employee, we made a button for that, when it is clicked the assigned employee name is shown with a cross icon if the admin changed his mind. In this way the admin can assign multiple employees to one task at once.
- It's very important to note that the table of the chosen employee is not shown as long as he is already assigned to the maximum number of tasks an employee could be assigned to. This number is not hard coded, it can be specified by the admin for each department. We made that using a text field of type "number".

- We added a chart for each project. It is drawn using Morris library. We chose the doughnut shape to illustrate the number of started, in progress and finished tasks for the project in a user friendly and understandable way.

Moving to the daily log part:

- At the user side in the section of his project management. We made a table which shows the employee the projects he is assigned to with their start and end dates and the corresponding department.
- Next step in the user side, we made the “Dailylog” tab. At this tab we constructed a form. In this form the user chooses a project then chooses a task from a list, this list shows the tasks which are not finished yet. Then he chooses the date which should be the current date. And he fills the period he worked on this task either in minutes or hours but it is stored as minutes in the database. Then, he modifies the status of the task either to “In Progress” or “Finished”.
- Note that if the user forgot to log his work he can log it later. Since these logs are sorted before shown to the admin.
- At the admin side, in the Project management section we added a tab named “Generate Reports”. We made a form for the admin to fill in it the username of course with the functionality of dynamic suggestions. And a period defined by a start and end dates.
- When submitting the previous form. We made a page which illustrates the chosen user logs during the period chosen. We interpreted this data using treefy library to make the table collapsible. We colored the task which is not finished on its estimated time with red color and all the logs from the user which are after the end date of that tasks are also bold and red.
- Finally, we were not content with the table. We made a line chart using CanvasJS library which illustrates the data of a chosen project. This chart shows each task and its logs as a line. This way is clearer and user friendly for the data to facilitate understanding and analyzing it.

At the end we noticed that the websites needs a unique start page to forward the user to the login page. We used in designing this page Bootstrap Start theme.

The security concepts we have are:

1. Retrieve data using JSON. Nothing is sent in the URL.
2. The user is authenticated using his credentials from active directory server.
3. The session is validated on each page.
4. AntiForgeryToken is used to protect the application against cross-site request forgery.it creates a hidden field that is validated when the form is submitted.

This was a detailed description of all the steps we went through building our project and all tools we used. This work was finished on Thursday 1st of December 2016.

Chapter 5: Results and Discussion

We made a lot of things in our application that gave us a very good results at the end of the project with good friendly design for the website with respect to all functionality we used and all features we made.

Starting our results from the **start page** of Team Allocation system. This page forwards the user to the login page.

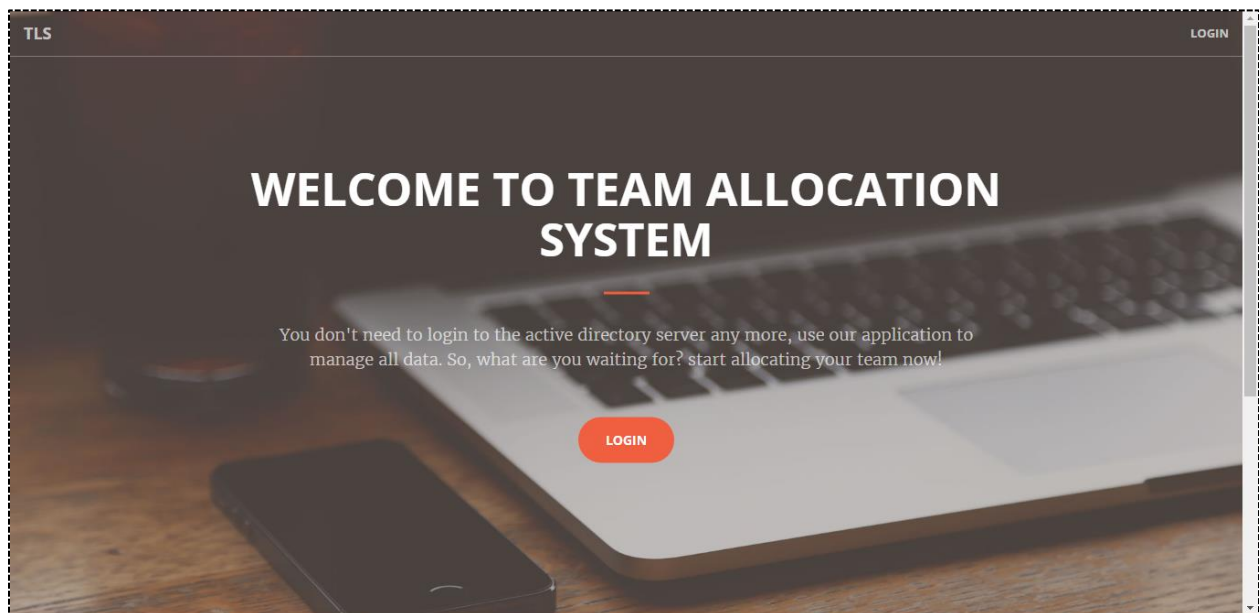


Figure 2: Start Page

A simple design for the login page was made. This page checks the connection to the server and validates the user depending on his credentials which are taken from the active directory. It produces an error message if any error detected in this procedure. The user is either an admin or a normal employee.

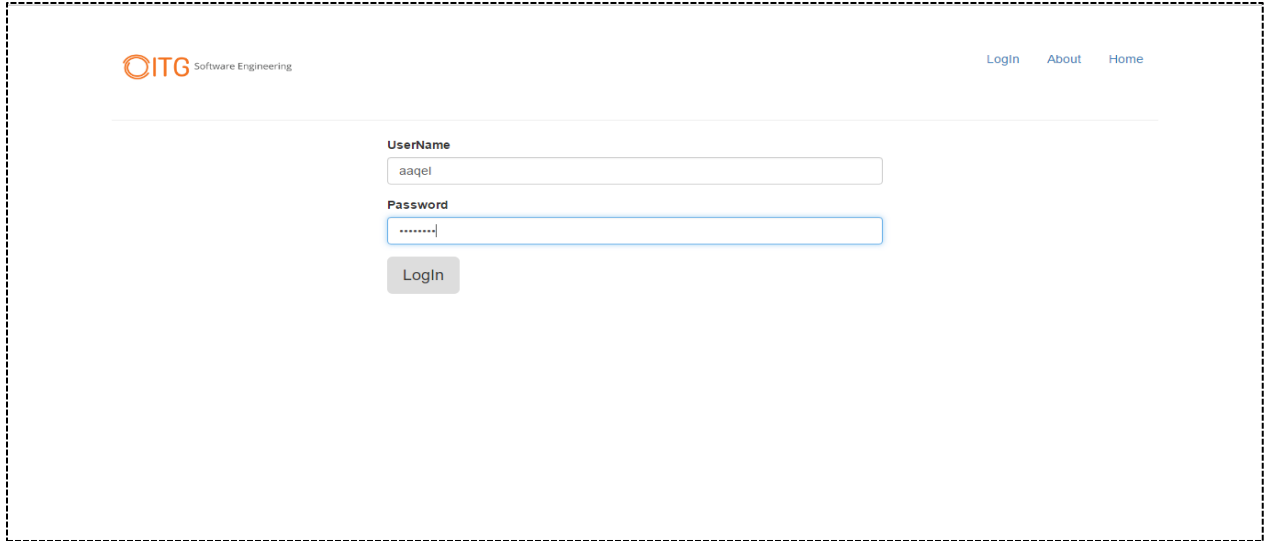


Figure 3: Login Page

5.1 Features in admin section:

When the user login as an admin he will be forwarded to the admin page as shown below.

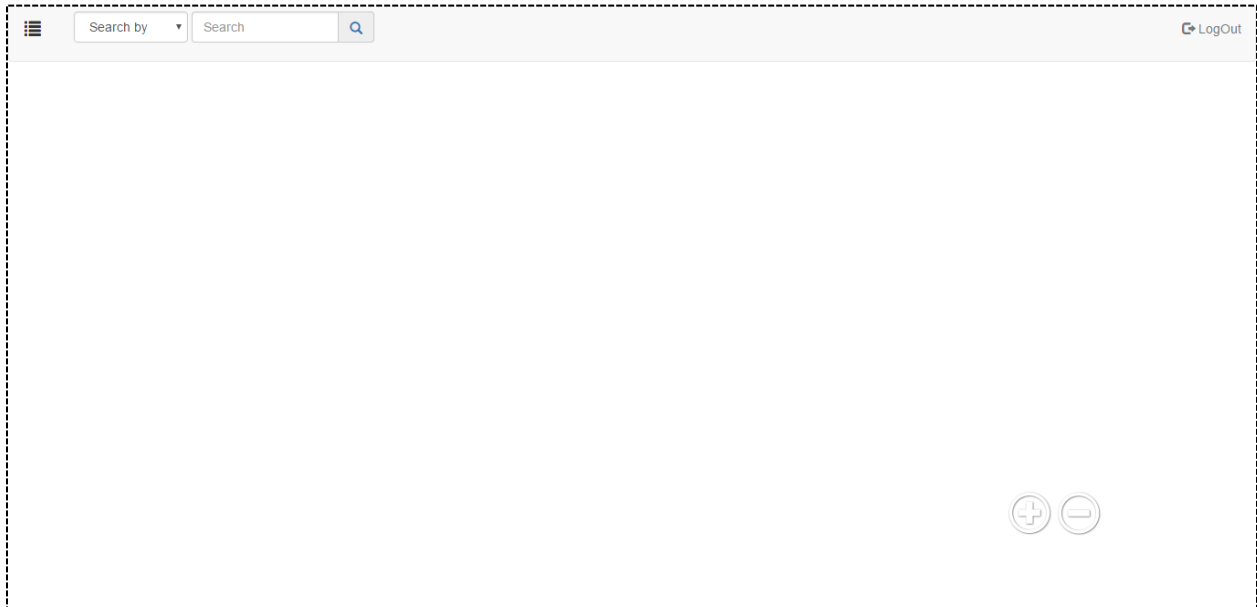


Figure 4: Admin Home Page

5.1.1 Mapping:

We added a lot of features and functionalities that help the admin to map the employees with respect to their current locations in the company.

- **Add New Floor:** the admin has the ability to add a floor name and to upload the floor image as PNG or JPEG format by clicking on the add floor tab which will display a window to input the information and then save these data.

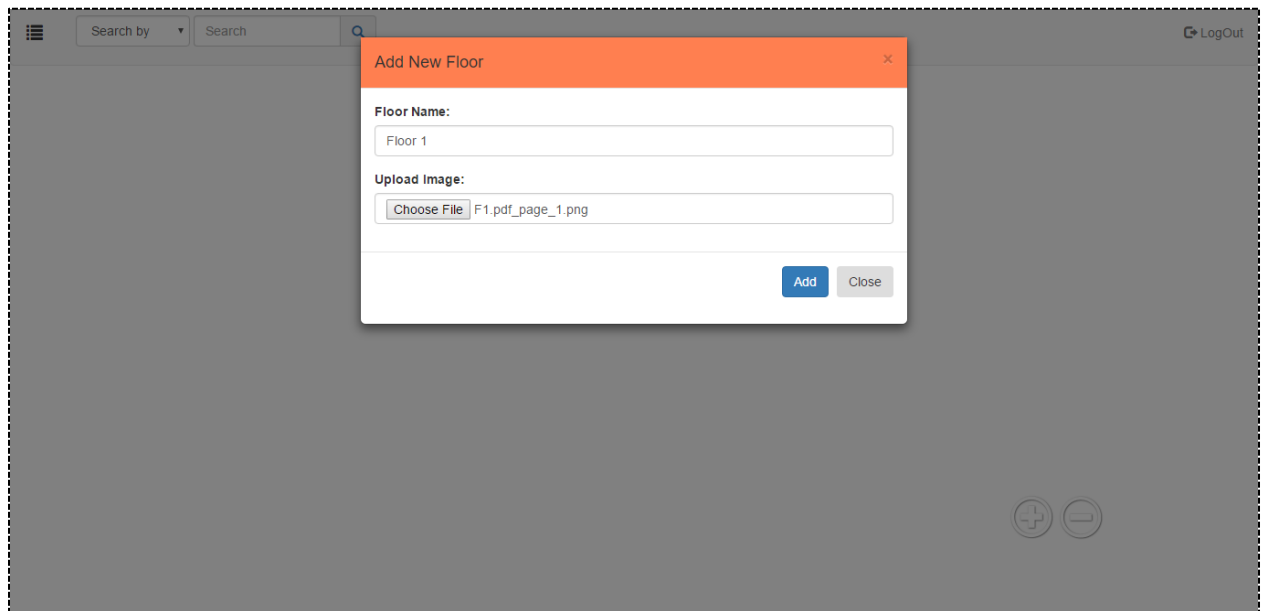


Figure 5: Add New Floor Form

- **Select one of saved floors:** after saving floor information, he can choose one of saved floors by clicking on its name which appears in list then the floor image will be displayed.

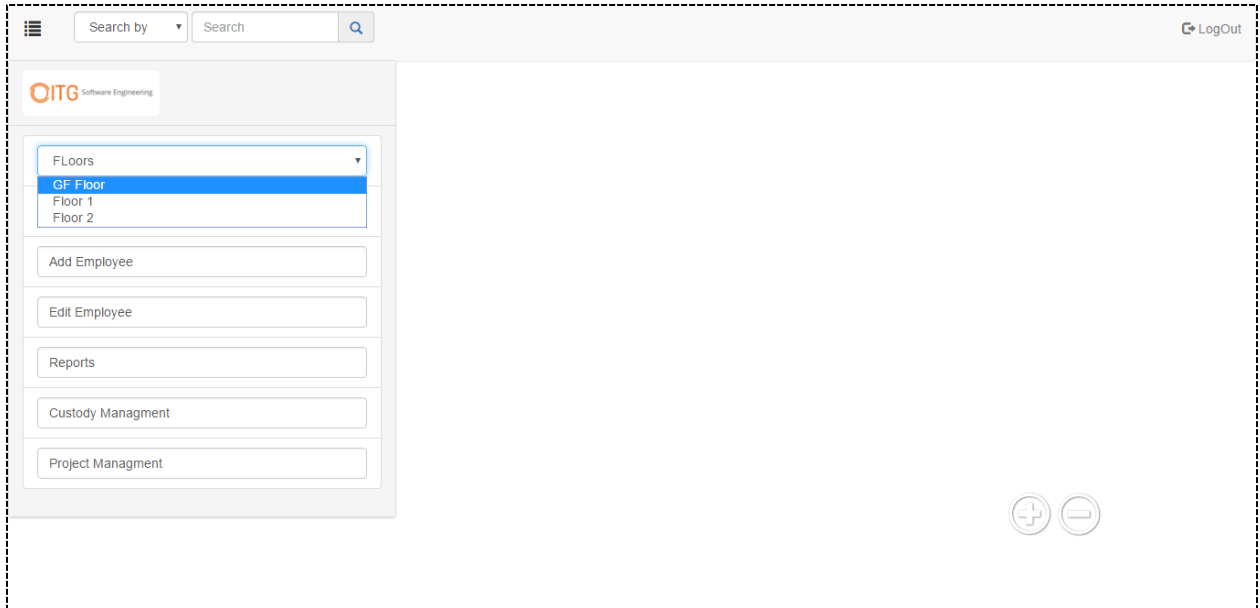


Figure 6: Floor List

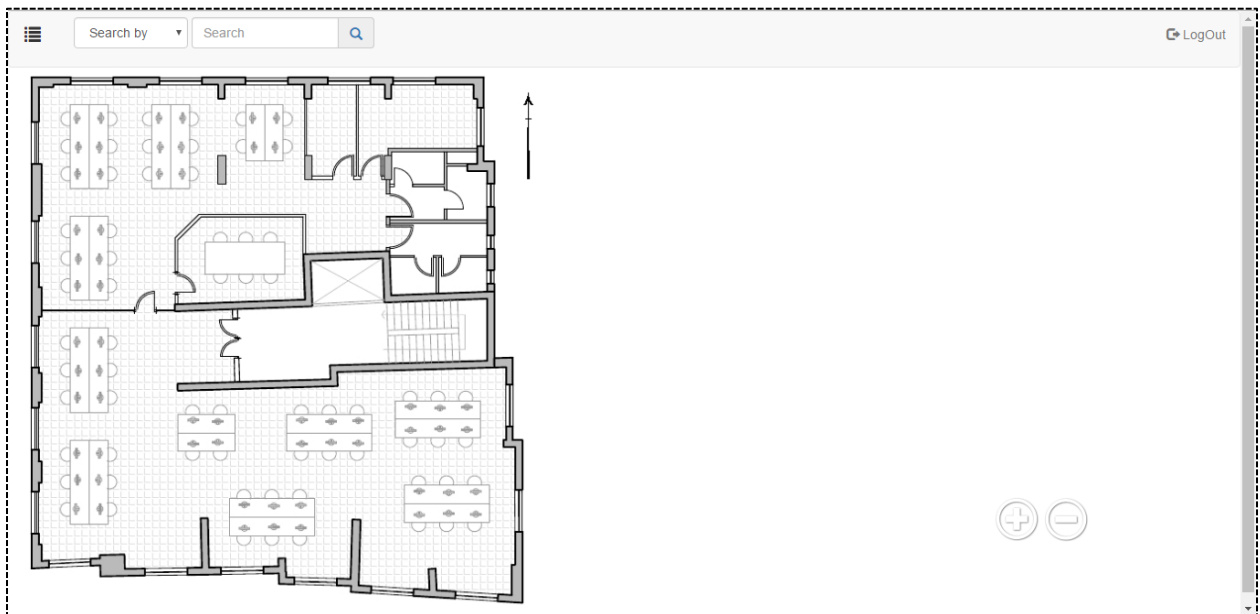


Figure 7: Floor Image

- **Moving the image, zoom in and zoom out:** one of the best features we have is the ability to move the image from its current location to any other location by just dragging and dropping the image. Also, the admin can change the size of the image by clicking on the zoom in or zoom out buttons at the bottom right of the website.

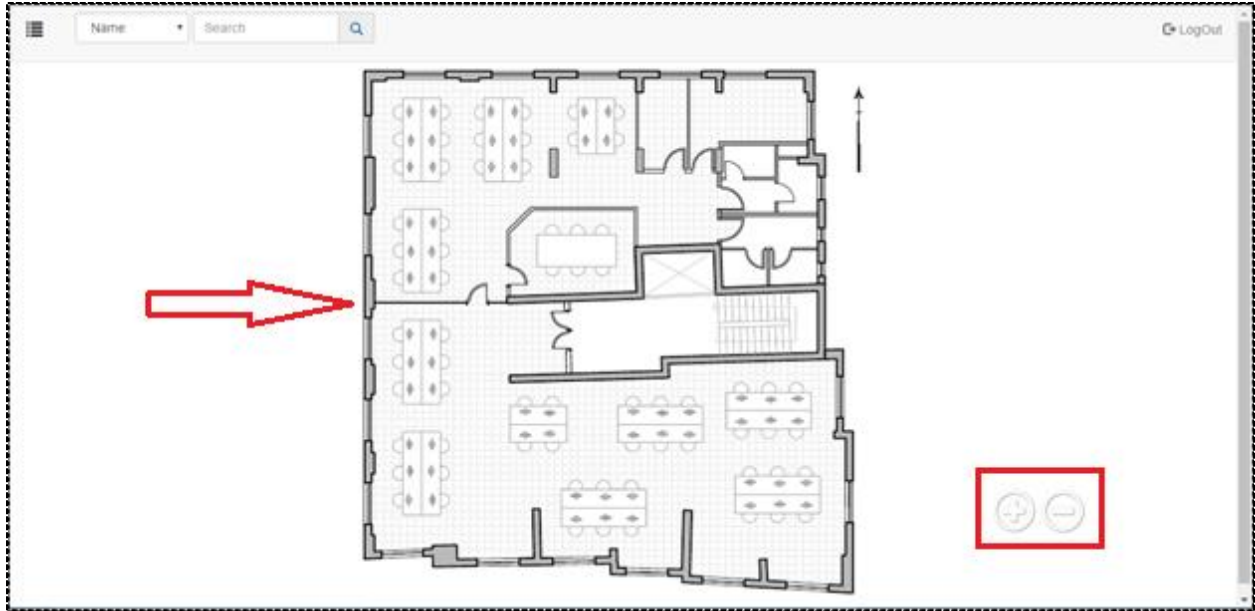


Figure 8: Zooming and Moving of Image

- **Adding new seat that will appear on the image:** after the floor image is displayed, the admin can add a new seat on the image by choosing the add employee tab and click on the add button causing the seat to appear at the left top of the image. After that, he can drag and drop the seat to any place on the image. He can add multiple seats at a time and then save all added seats.



Figure 9: Add Multiple Seats

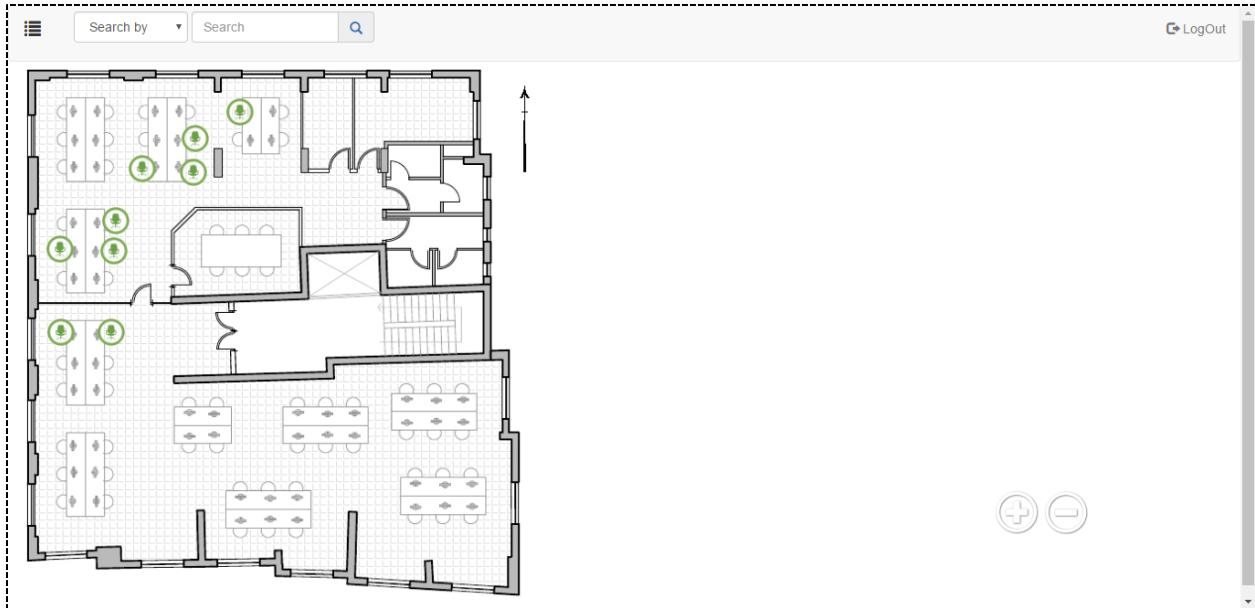


Figure 10: Save Added Seats

- **Assign Employee to a seat:** when the admin clicks on the seat, a window will be displayed with a search box to search for any employee using his name.

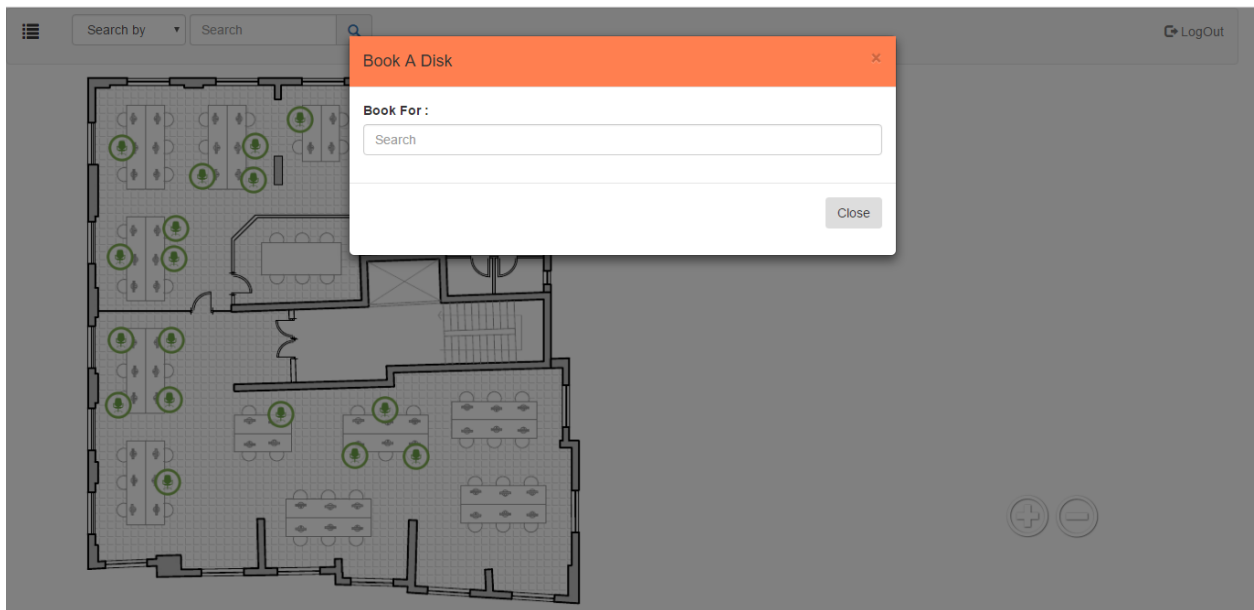


Figure 11: Assign Employee to a Seat

While writing the name our application will display auto suggested employees to him from active directory with add button to select.

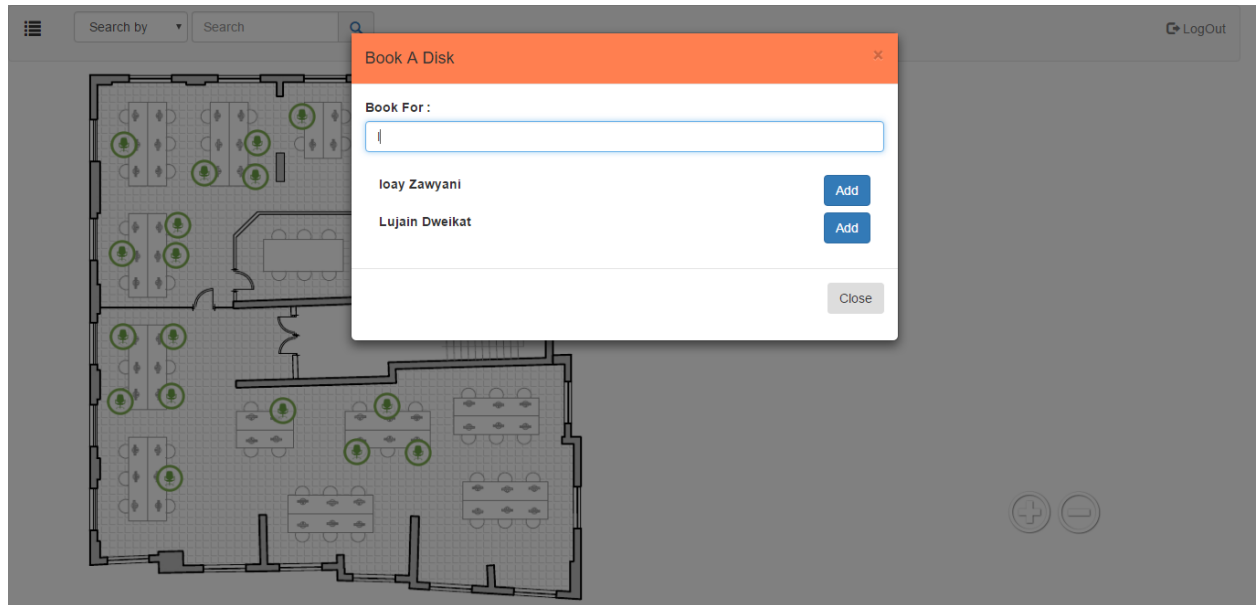


Figure 12: Auto Suggestions Feature

Here we have two situation the first one if the selected employee is already assigned to a seat and the admin selects him, a message will be displayed that this employee is already assigned.

The second situation when the selected employee is not assigned then the adding will work successfully and the assigned employee picture will appear instead of the chair picture.

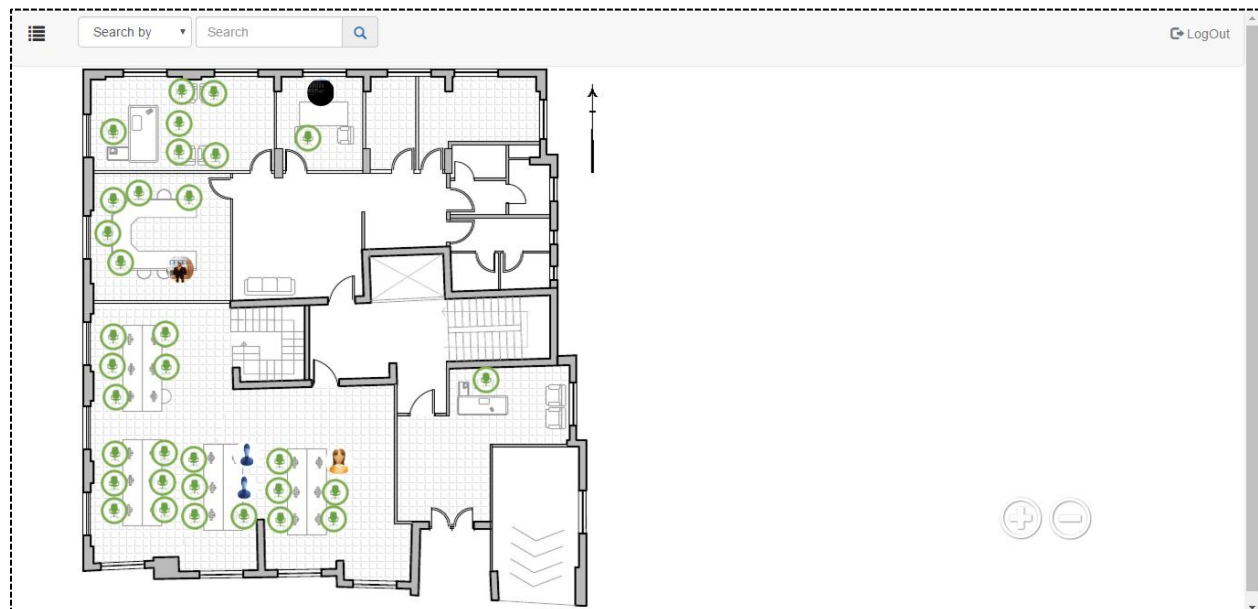


Figure 13: Assign Employee Successfully

- **Display the information of the employee:** when the employee that exists on the image is clicked, some information about him including his image from active directory and the floor where he placed will appear at the left side of the website.

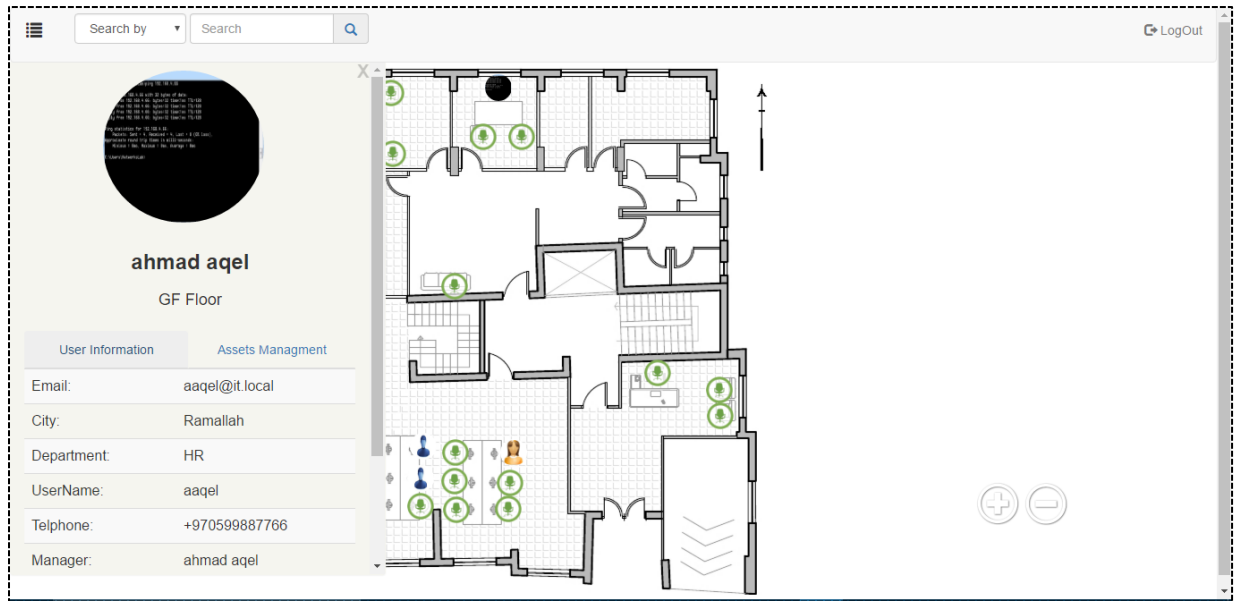


Figure 14: Employee Information Sidebar

- **Delete employees or move their locations:** our project gives the admin the ability to delete employees or seats by clicking on edit employee tab and select any employee to delete by clicking on him. He can also delete multiple employees at a time by clicking on them and press delete button.



Figure 15: Multiple Delete of Employees from Floor Plan

Also, he can change their locations by dragging and dropping the seats or employees to any location he wants on the image and then clicks on save button to save the new locations.



Figure 16: Edit Employees Locations on Floor Plan

- **Search Feature:** here the admin can search for employees using multiple search filters with auto complete from active directory on all filters.
 - 1) The first filter is searching using the city filter, auto suggested cities are retrieved from active directory. A window will be displayed containing the employees that lives in the selected city.

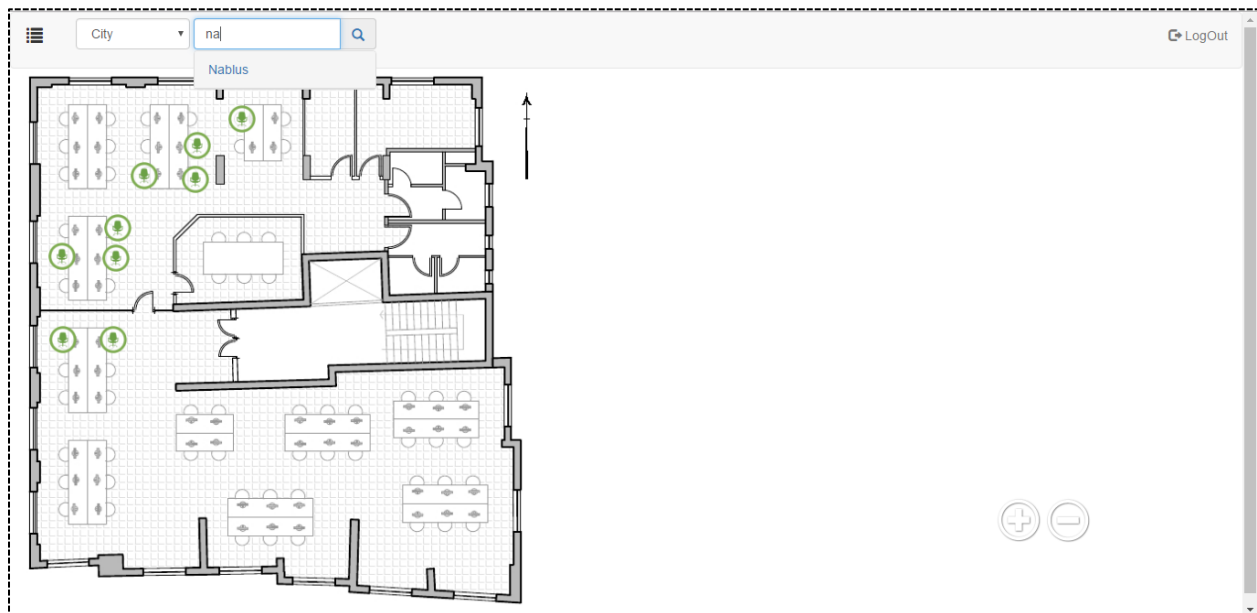


Figure 17: Search by City

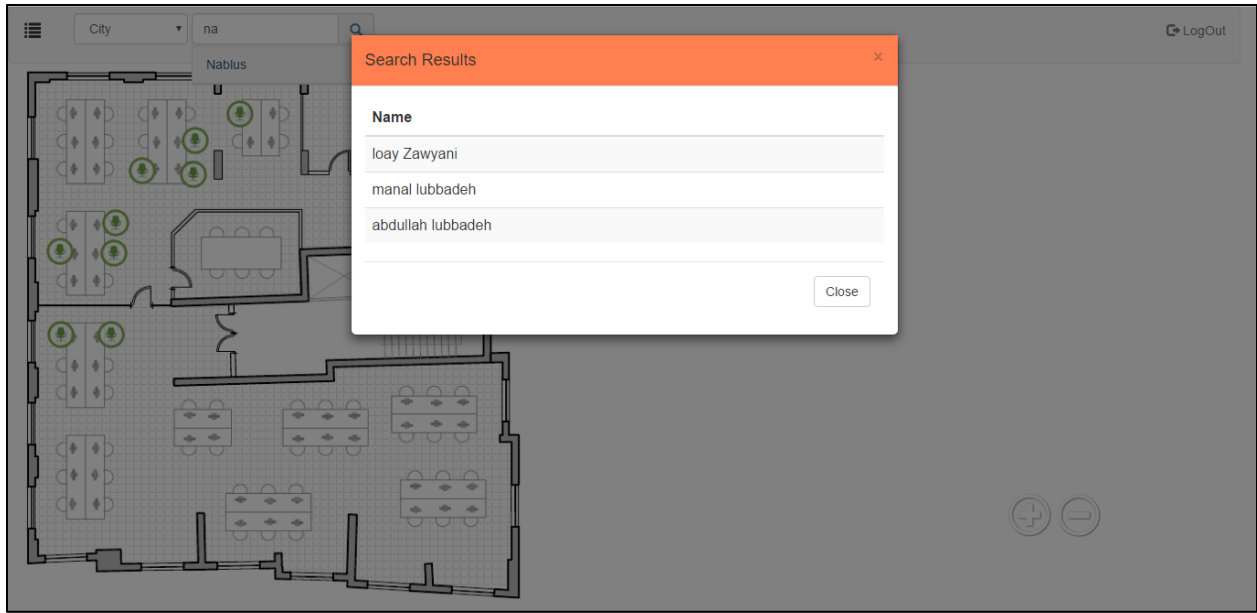


Figure 18: Search by city result

- 2) The second filter is searching using department filter with auto complete features from active directory. When he selects the department a window will be displayed containing the names of all employees in this department. The result is shown same as search by city result.

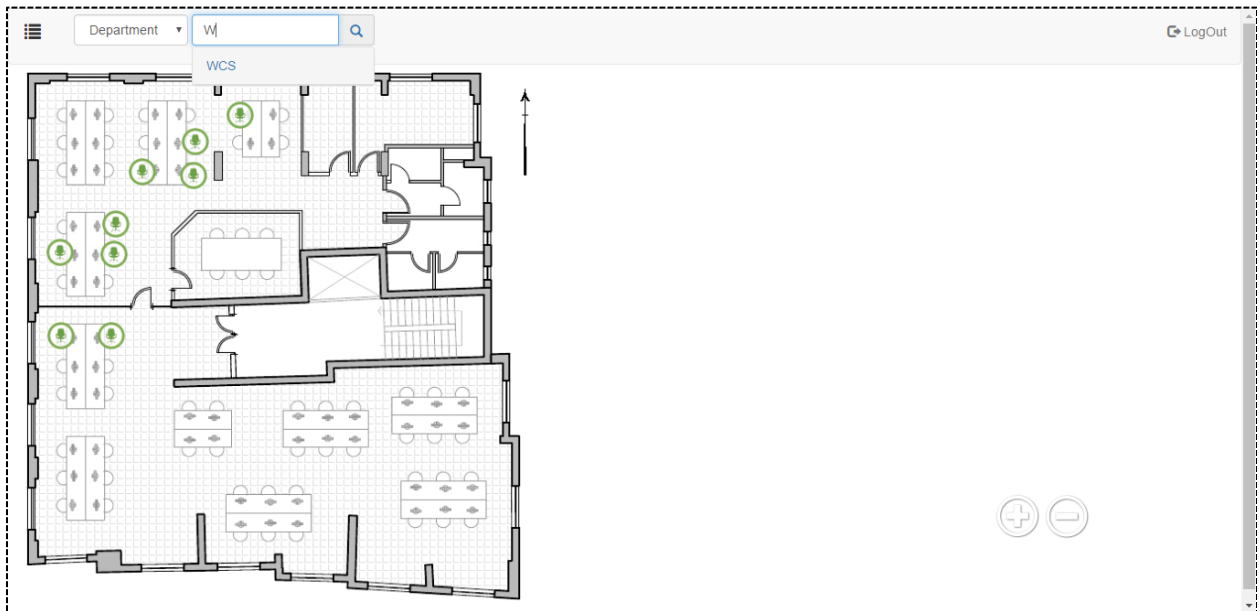


Figure 19: Search by Department

3) The final filter is searching using username with auto complete from active directory.

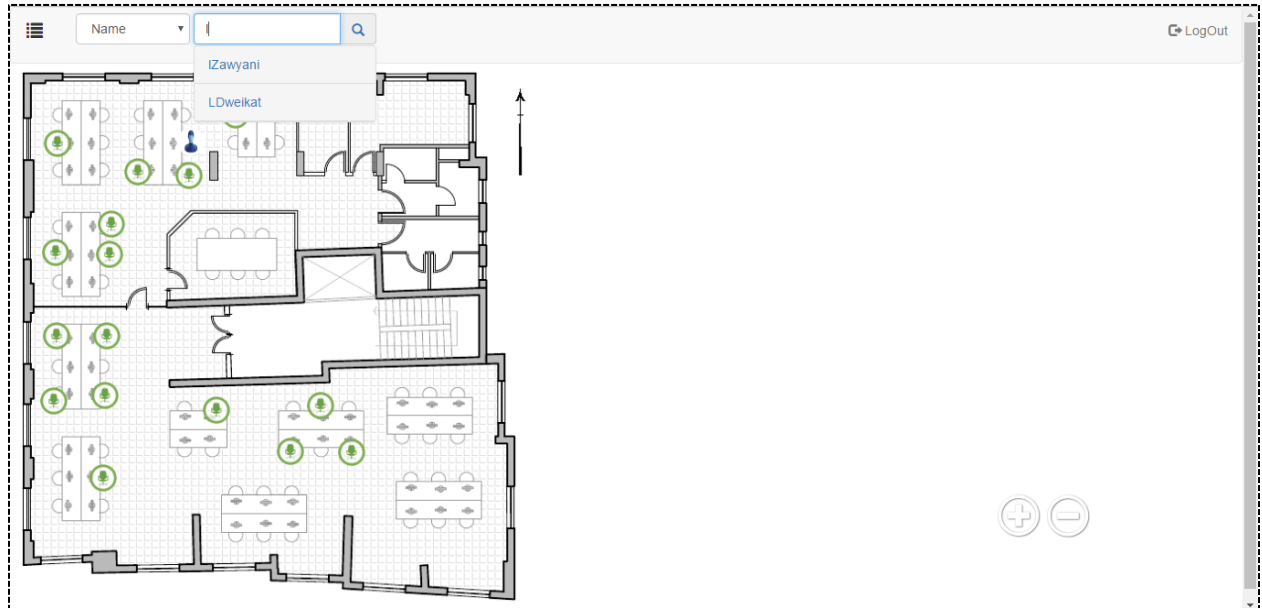


Figure 20: Search by Username

When the admin chooses the username for the employee, he will have two situations. The first one if the selected employee who has a location in one of the floors, the application will automatically open the floor where selected employee exists and marks him with red color.

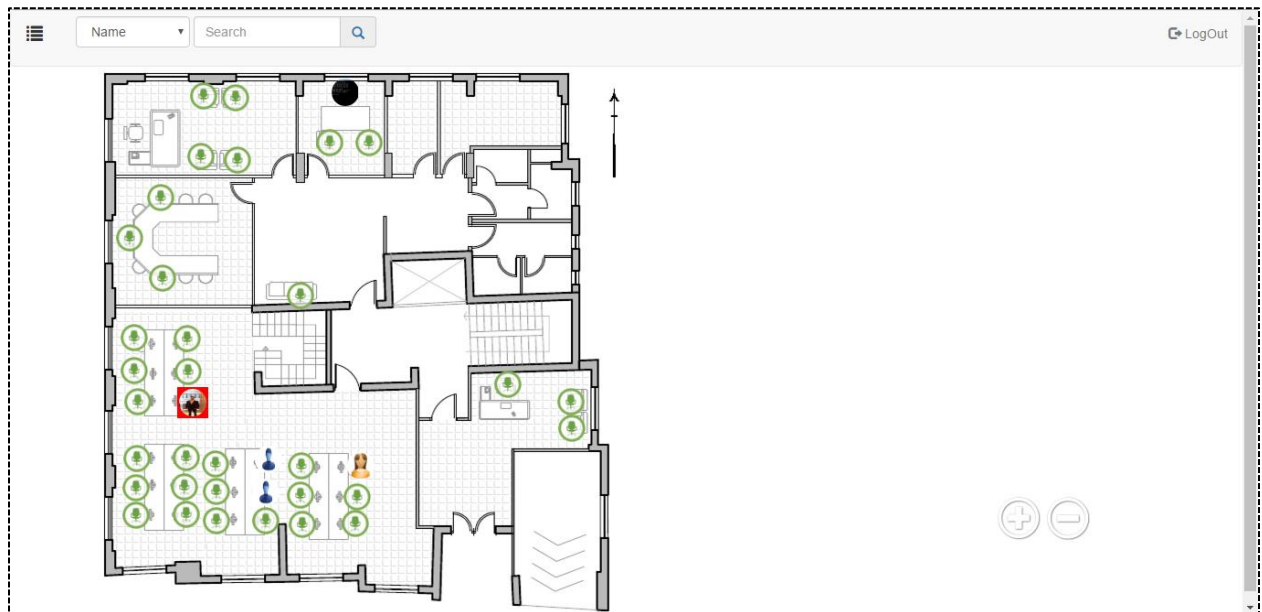


Figure 21: Search for Mapped Employee

The second situation is when the employee is not assigned to any floor. In this case, his information will be displayed showing that this employee place is not ready.

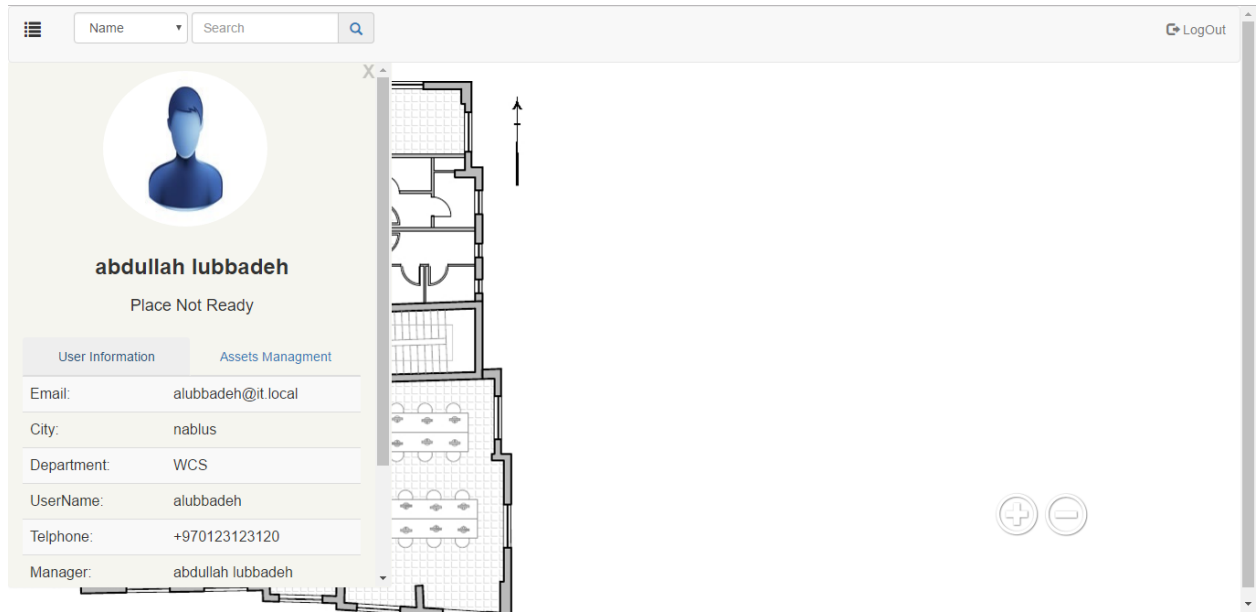


Figure 22: Search for Unmapped Employee

5.1.2 Reports:

When the admin clicks on “**Reports**” tab, he will be forwarded to the report page that includes:

- **Departments exist in the company:** when the admin is forwarded to the report page he will see all departments that exist in the company. So that, any department added in active directory, it will dynamically have a report in this page.



Figure 23: Reports Page

- **Departments Data:** after the departments are displayed, the employees that exist in each department and the leader of the department are shown with their information from active directory by clicking on the department name. The information will be displayed in a data table with ability to search for any field in the table. Also, the data table has paging feature and ability to control number of rows to appear in the table.

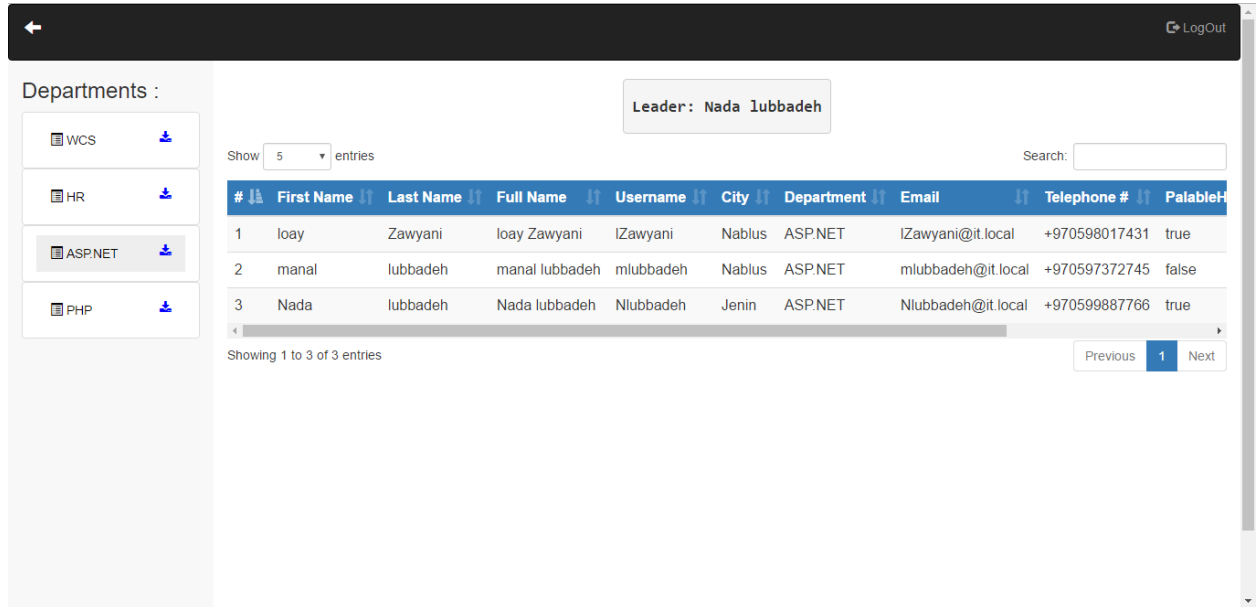


Figure 24: Department Report Table

- **Doing some changes to table fields:** after the data table displayed, the admin can do some changes on the employee's data by double click on the field and change it then when he leaves the field the changes will be saved and reflected on the active directory automatically.

The admin can change the city, telephone or palable hour's information by double click on its field and doing his changes then save.

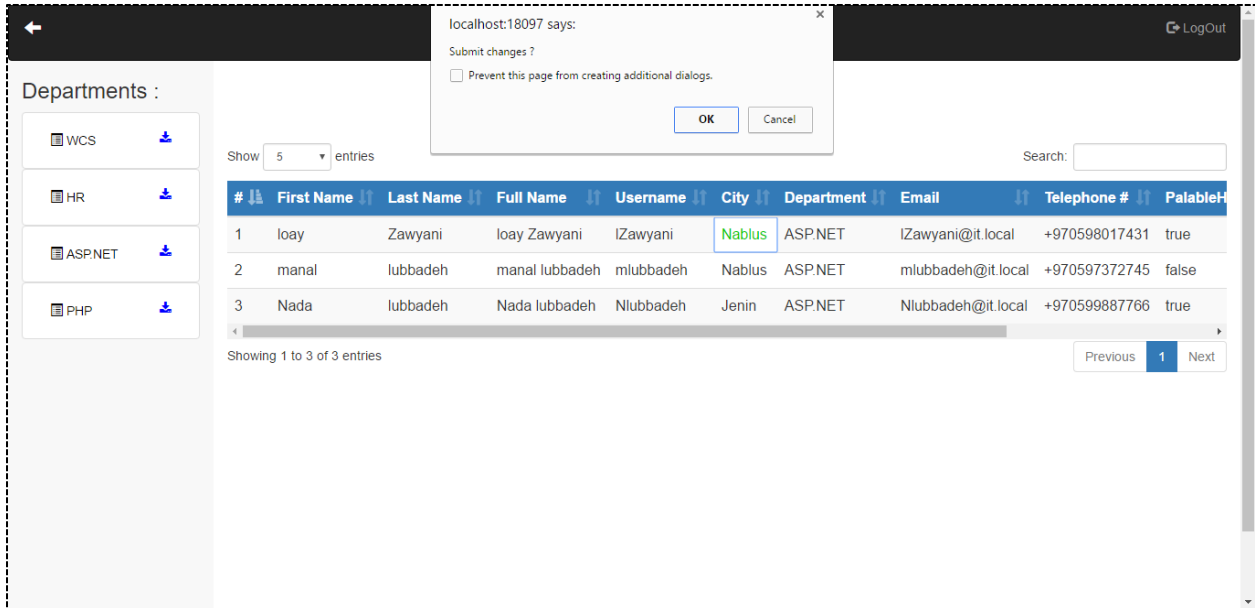


Figure 25: Update A Field Value in Report Table

He can also change the last name of the employee but here when he changes it then the full name, username and email will be changed automatically because in active directory the last name, username and email depend on the first name and the last name and this will be reflected on the active directory also.

In addition, the admin can change the department for each employee by changing the department field. By a double click on it, a list of the departments that exist in the company from the active directory is shown, then he selects any department and save changes. After changing the department the table will be refreshed automatically.

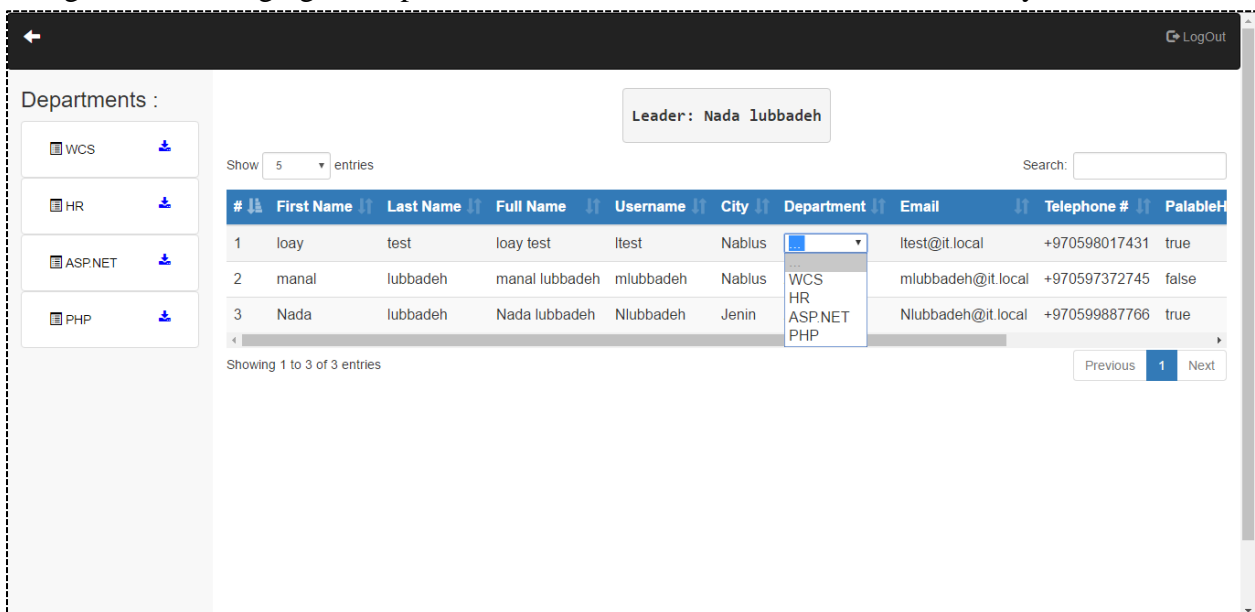


Figure 26: Change Employee's Department in Reports

- **Download information as PDF:** the admin can save reports information by clicking on the icon that appears beside the department name, clicking on the icon will download the data and it will be saved as PDF file.

Infinite Tiers Company				
Department: ASP.NET			Date: 12/3/2016 11:52:37 PM	
Leader: Nada lubbadeh				
FullName	Username	City	Email	Telephone#
loay test	ltest	Nablus	ltest@it.local	+970598017431
manal lubbadeh	mlubbadeh	Nablus	mlubbadeh@it.local	+970597372745
Nada lubbadeh	Nlubbadeh	Jenin	Nlubbadeh@it.local	+970599887766

Figure 27: PDF Report

5.1.3 Custody Management:

We mean by custody management the tools that any employee can use from the company for a period of time and then return these tools to the company; like chairs, laptops, pens, desktop P and any other tool available from company. When the admin clicks on the custody management tab, he will be forwarded to a page that includes all features about the custody management.

- **Displaying all tools available in the company:** when the page is opened, two tabs will appear; one for normal tools and the other for the computers. All tools and computers are displayed using data table with search, paging and selectable number of shown rows features.

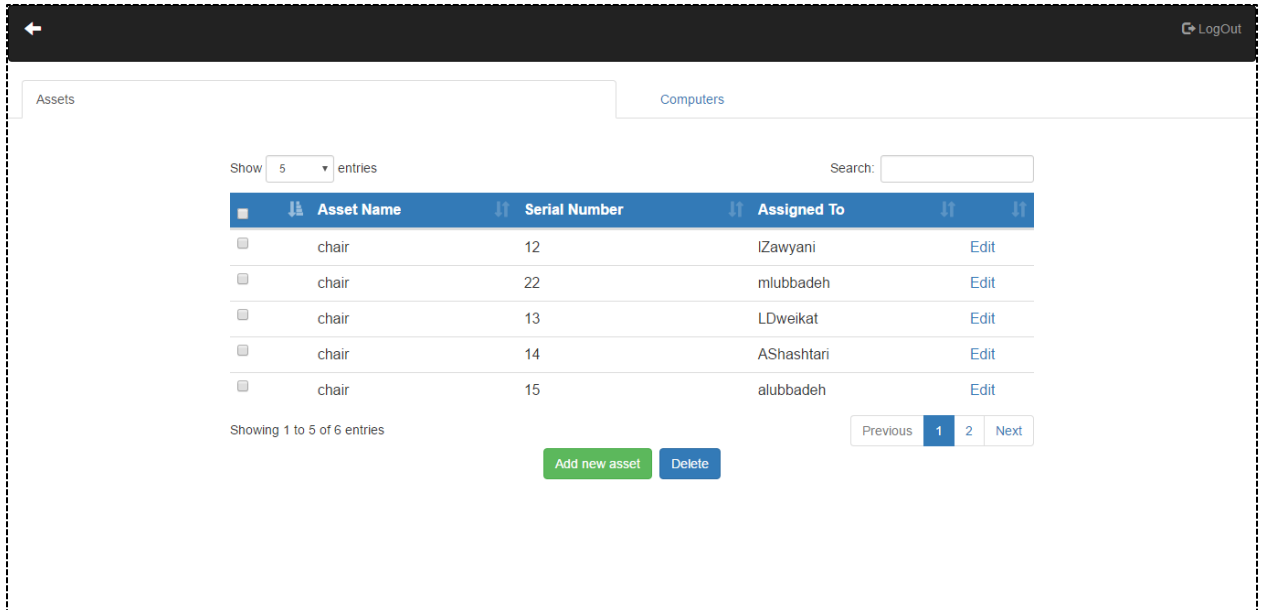


Figure 28: Normal Assets Management

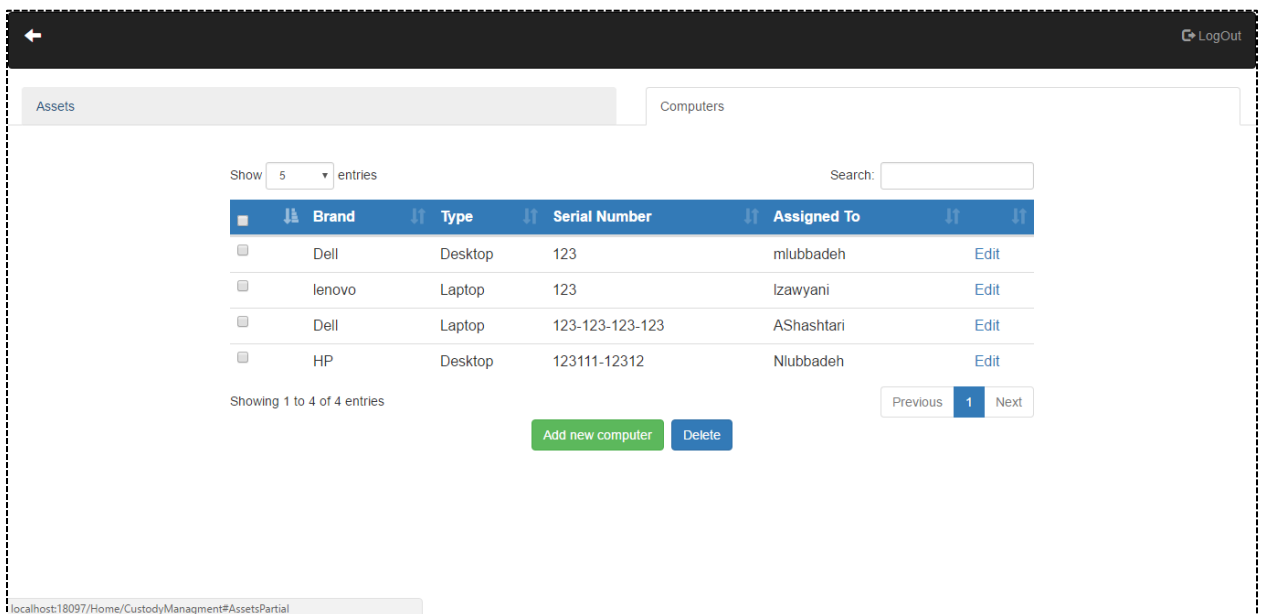


Figure 29: Computers Custody Management

- **Add new tool:** when the data table is shown, the admin can add a new asset or add a new computer by clicking on add button, then a window will be shown to enter the required data with search box with auto suggested employees from active directory to select which employee will has this tool or just adding the tool without assigning it to any employee as indication that it is still in stock.

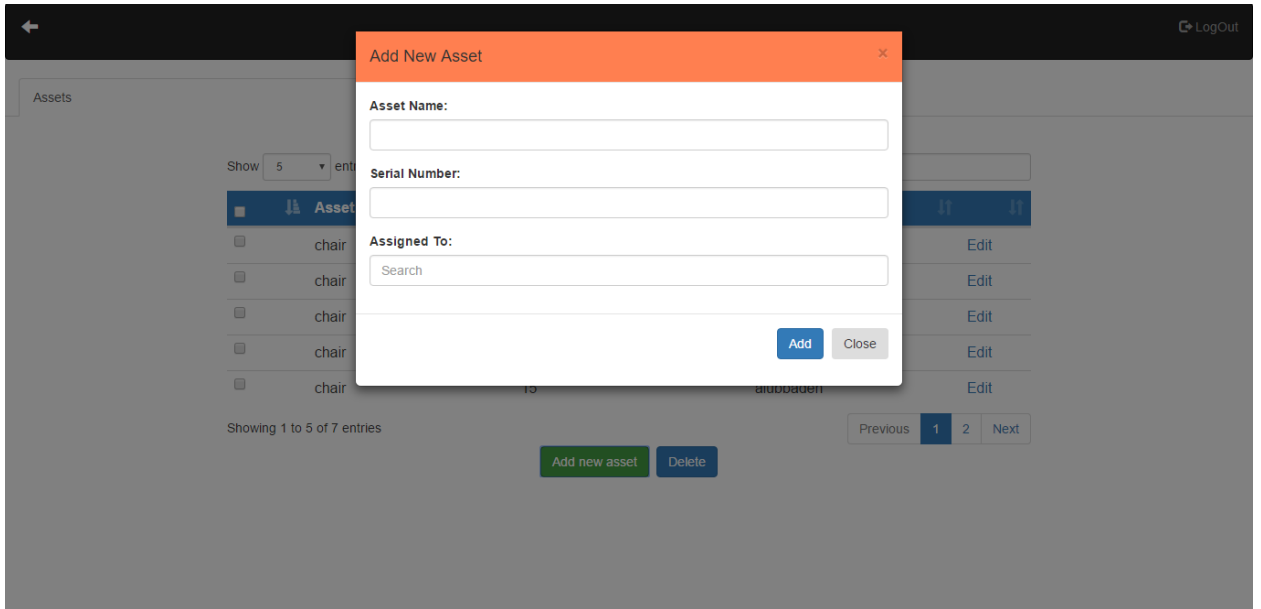


Figure 30: Add New Asset

- **Edit any selected tool:** the admin is able to edit any tool information by clicking on the edit button then a window will be shown which includes the tool's previous data, then he changes any data he wants and clicks edit to save new changes.

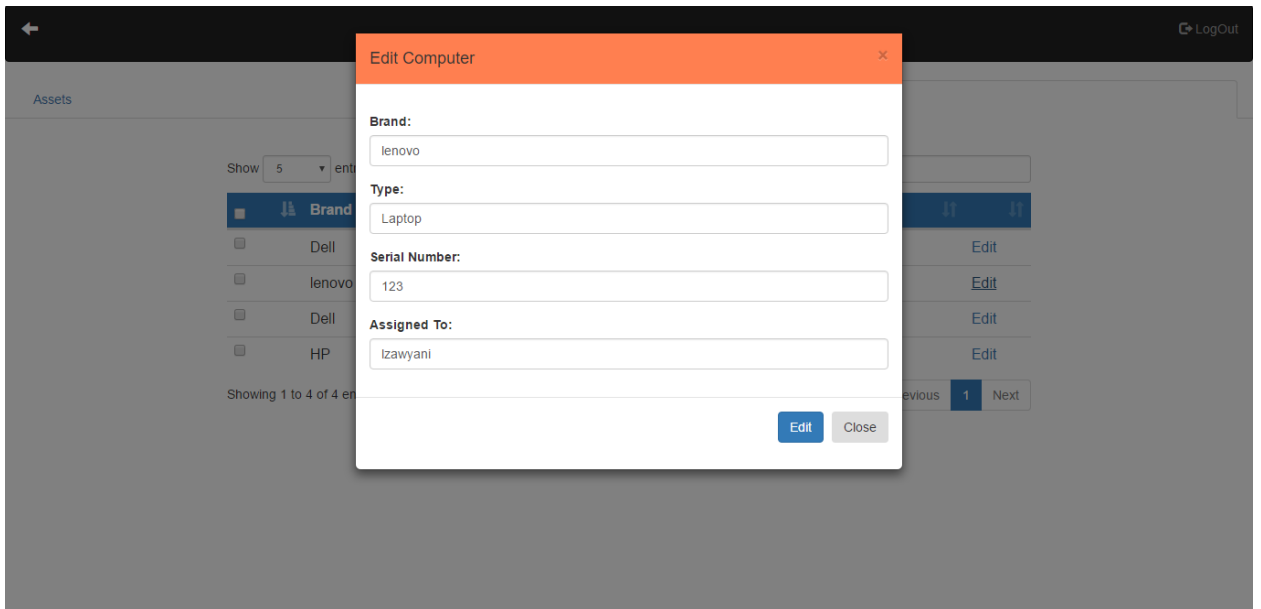


Figure 31: Edit Asset Information

- **Delete multiple tools at a time:** the admin can delete one tool or multiple tools at a time by clicking on the check box that appears beside the tool and then clicking on delete button, the table will be refreshed automatically.

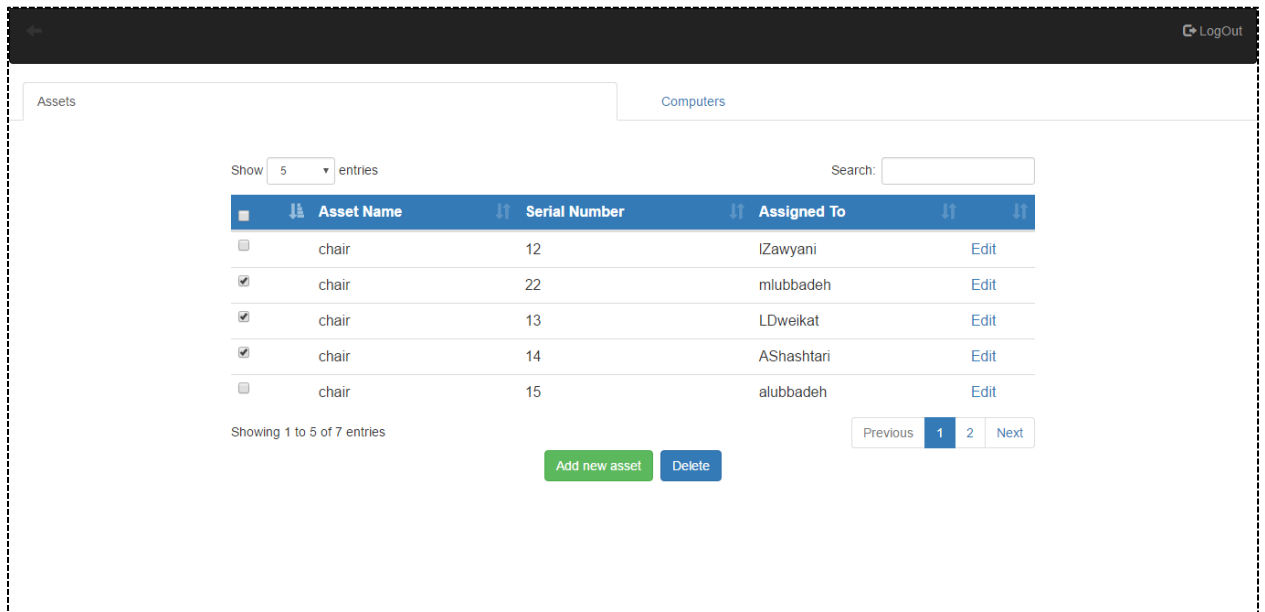


Figure 32: Multiple Delete of Assets

5.1.4 Project Management:

Each department has many projects to manage by adding the projects or assigning employees to each project and also watching each employee's how many hours he worked on each project. When the admin clicks on the project management tab he will be forwarded to its page which includes all departments in the company from active directory and will be allowed to manage each department projects and tasks.

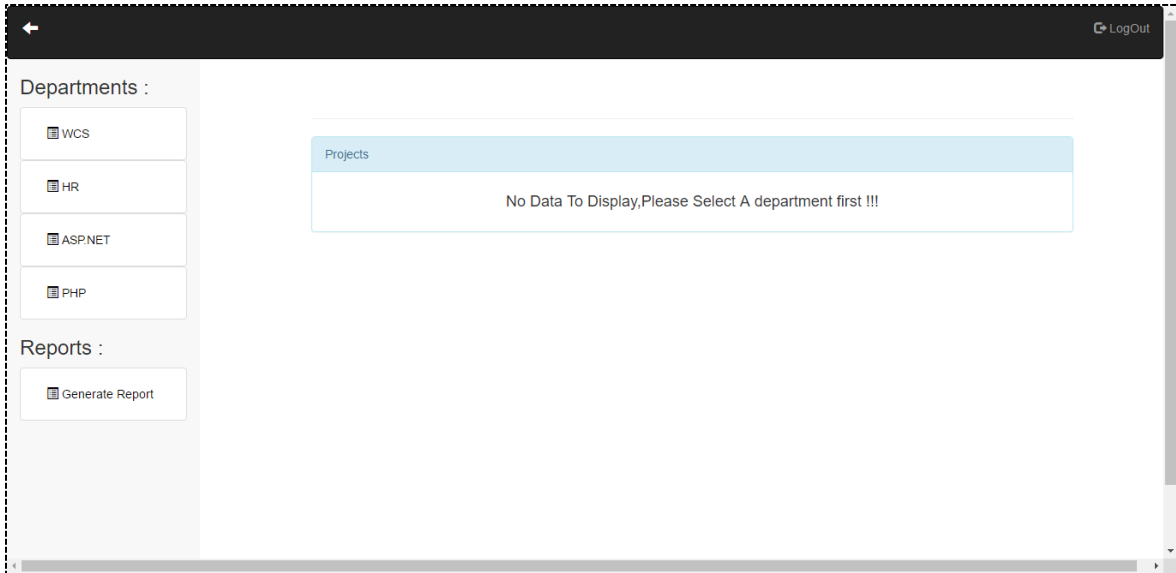


Figure 33: Project Management

- **See each department projects:** the admin can see each department projects by clicking on the department name then the projects will be displayed in a data table.

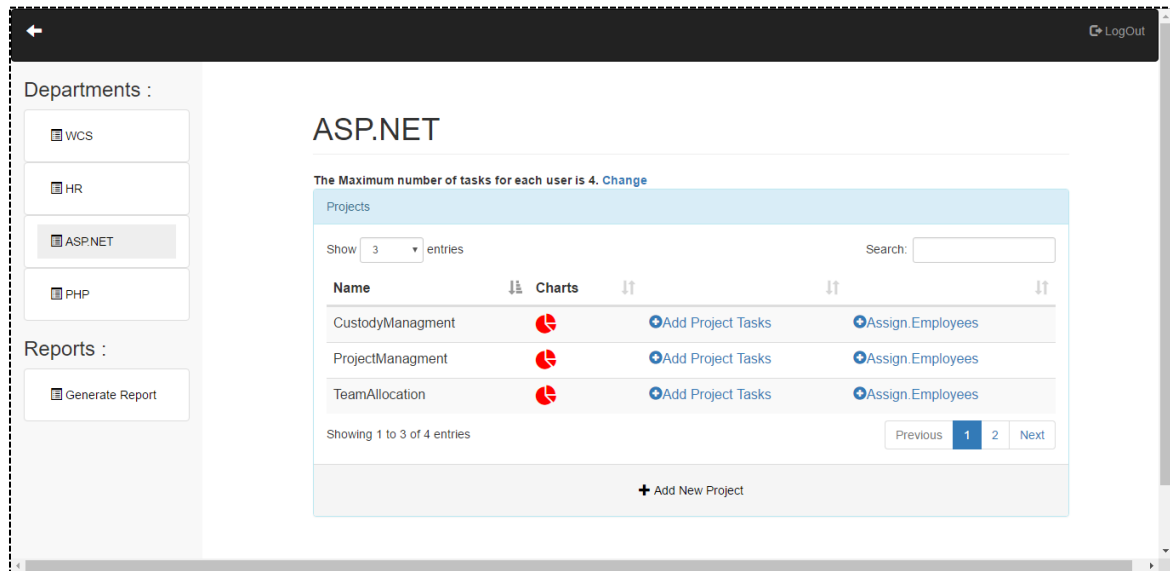


Figure 34: Display Department Projects

- **Add project:** the admin can add project to the selected department by clicking on “add new project” button below the table and filling the project required information about the project, then he clicks on add button to save.

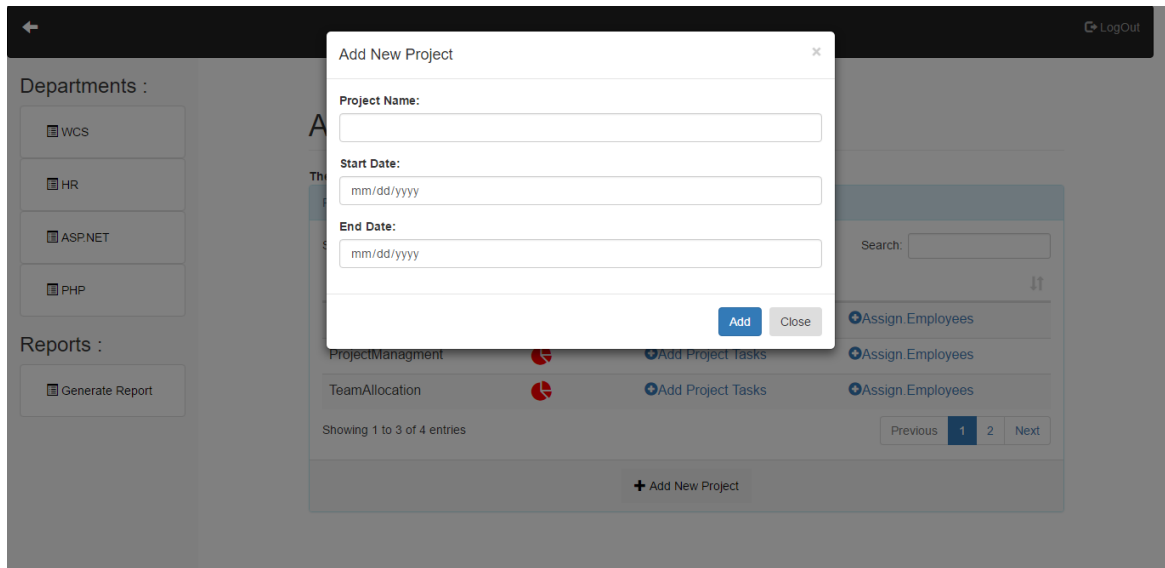


Figure 35: Add New Project

- **Add Tasks to Project:** After the project was added, the admin can add tasks to this project by clicking on “add project task” button, a new window will be shown to fill the task required information. The start and end dates of the task are restricted within the start and end dates of the selected project.

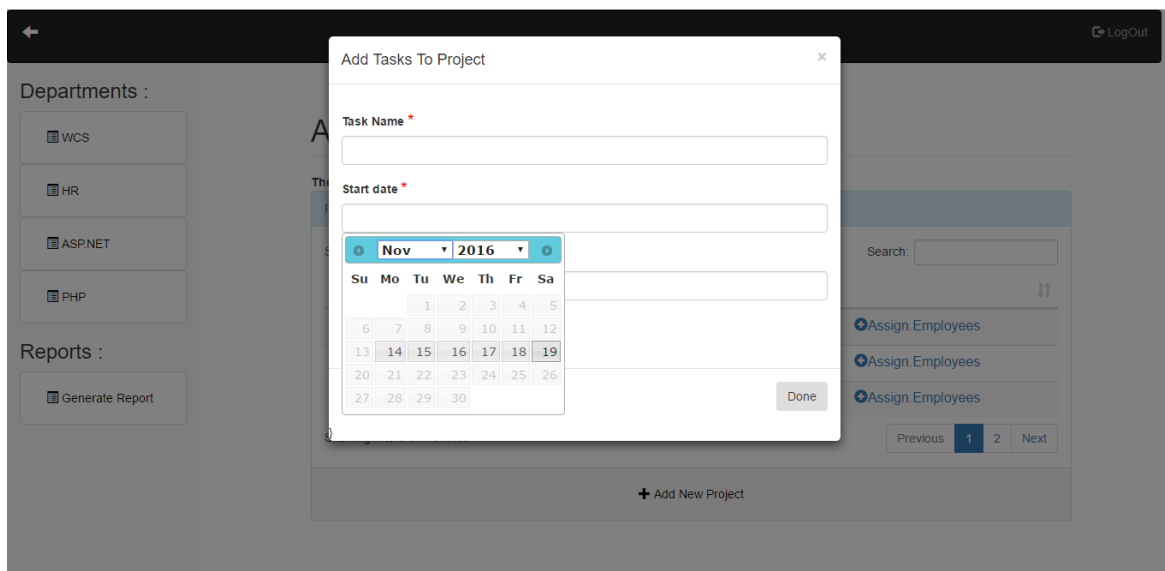


Figure 36: Add Tasks to Project

- **Show the tasks status using a chart:** the admin can see each project tasks status displayed as numbers in a chart by clicking on the chart icon beside the project name. When the mouse is over the red part, the number of started tasks will be shown, when it is over the yellow, the number of tasks that are in progress will be shown and finally when it is over the green part, the number of finished tasks will be shown.

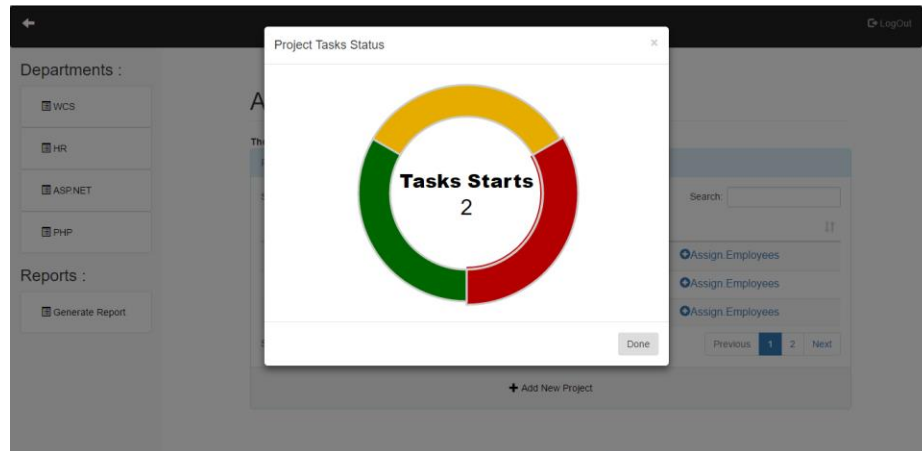


Figure 37: Project Tasks Status Chart

- **Assign employee to a task:** after the project and the tasks for the project added, the admin has the ability to assign employees to any task for selected project by selecting the task name and searching for the employee by name from active directory with auto suggested names.

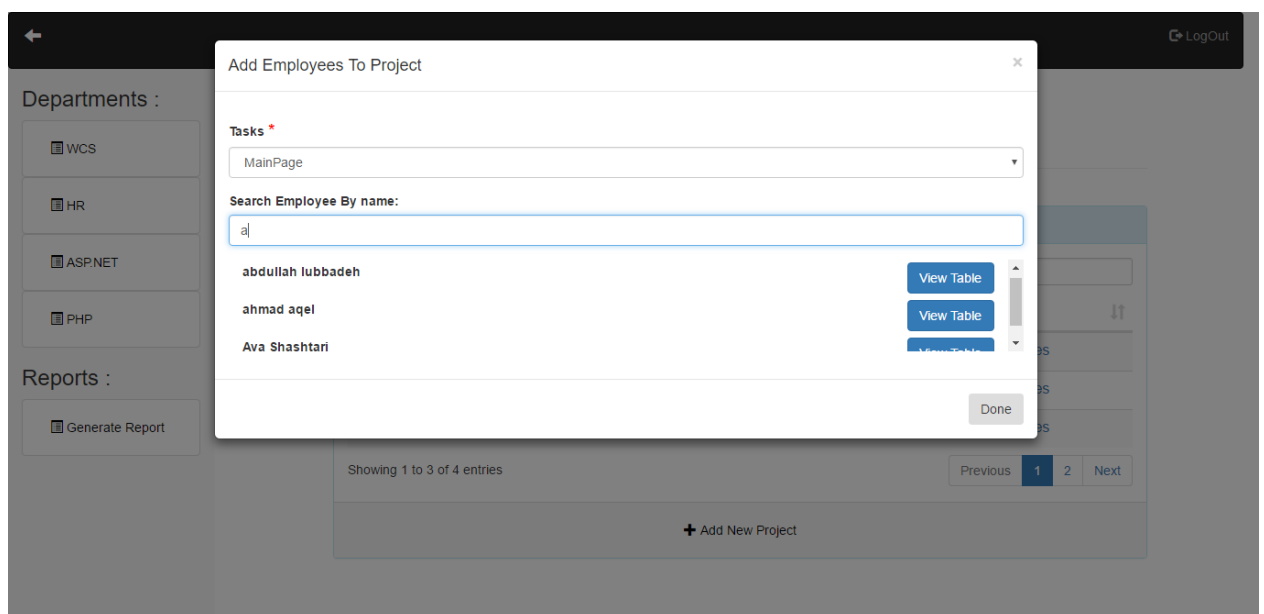


Figure 38: Assign Employee to a Task

After the admin searched for an employee, he clicks on “view table” button to see how many tasks this employee has before assigning this task to him. In this case, we will have 3 situation:

- 1) If he had a number of tasks that is below the default maximum number of tasks, the table of tasks will be shown and includes the tasks that are not finished yet with assign button to assign this employee to the selected task.

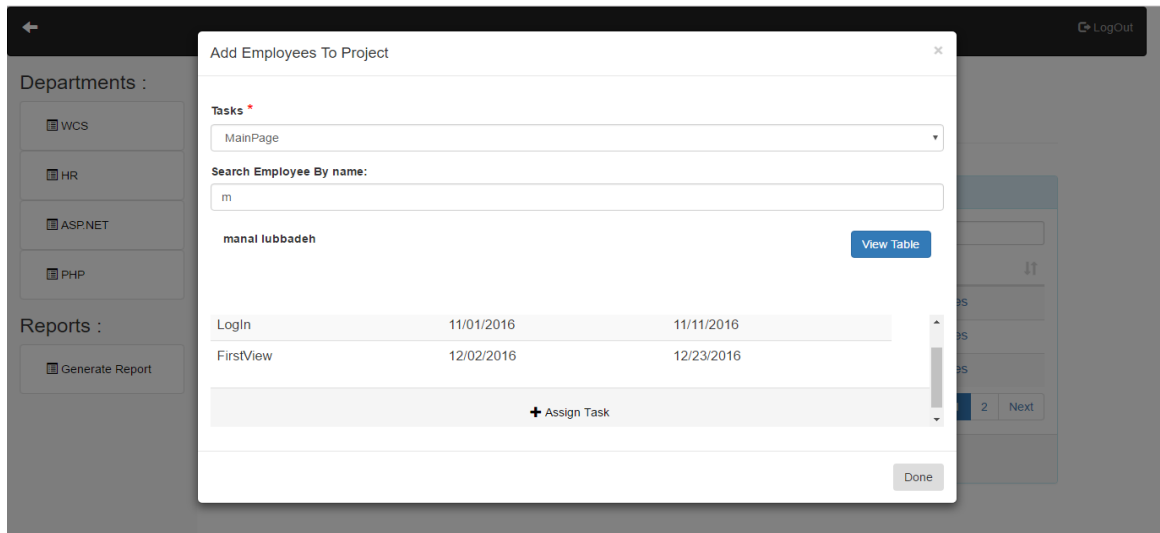


Figure 39: Assign Employee With Tasks Less Than The Max.

- 2) If the selected employee has a number of tasks that is greater than the maximum number, then the admin can't assign him to the selected task.

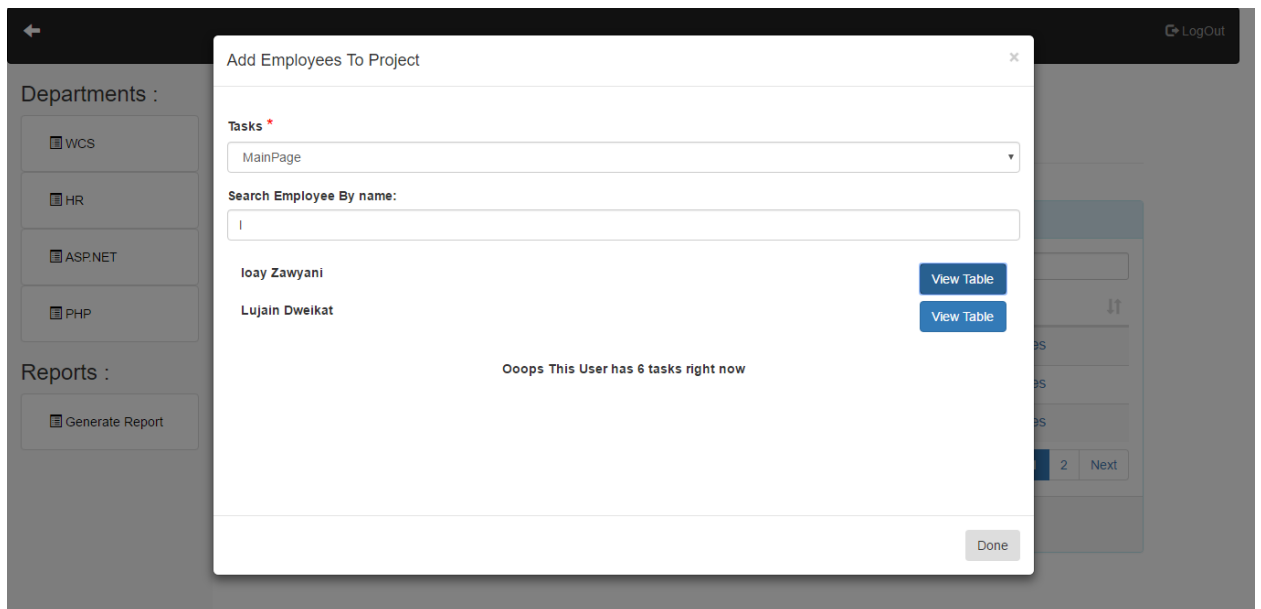


Figure 40: Employee Can't Be Assigned to Task

3) If the selected employee has no tasks, then the assign task button will be shown.

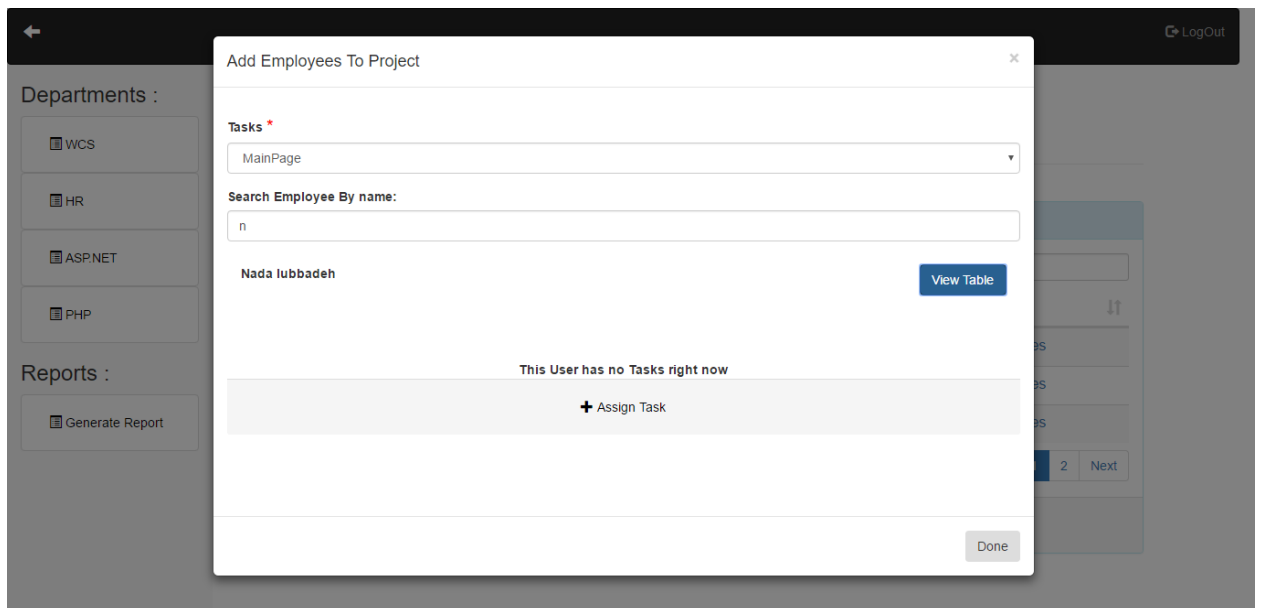


Figure 41: Assign Free Employee

- **Delete the assigned employees:** when the admin assigns number of employees to the task, they will be shown with **x** icon beside them which enables him to delete anyone of them before finishing.

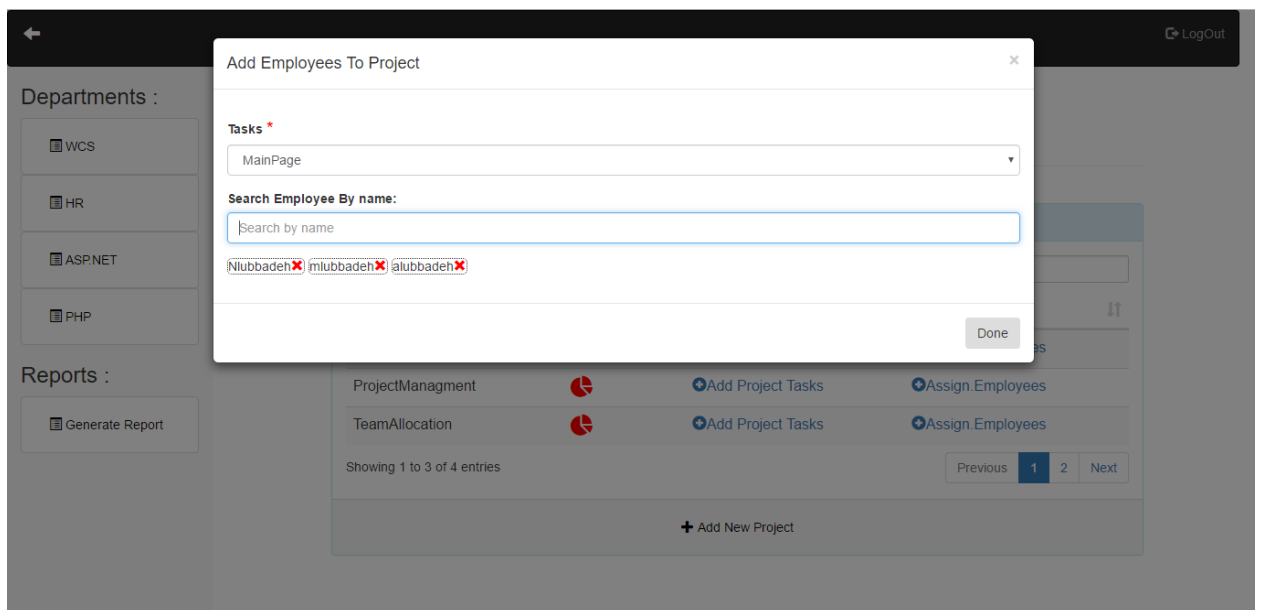


Figure 42: Delete Assigned Employee

- **Changing the default maximum number of tasks:** the default maximum number of tasks allowed for each user can be changed for each department by clicking on “Change” word above the project’s table and save the new value.

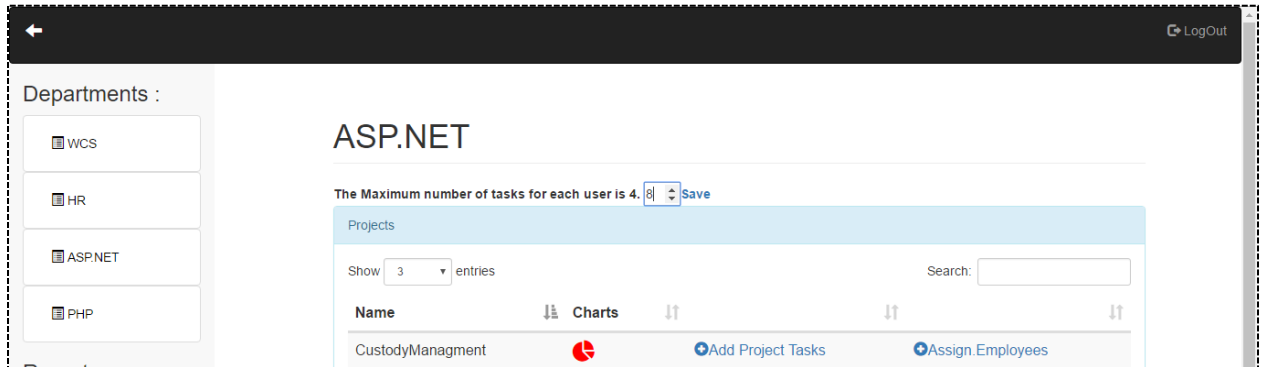


Figure 43: Change Maximum Allowed Assigned Tasks

- **Generate report:** the admin can watch how many hours a specific employee works on each project tasks in a specific period and if the employee worked on a task after its estimated end date ended. The data is displayed using a table and a chart. When he clicks on generate report tab he will be forwarded to a form to enter the username of the employee and the period he wants to see.

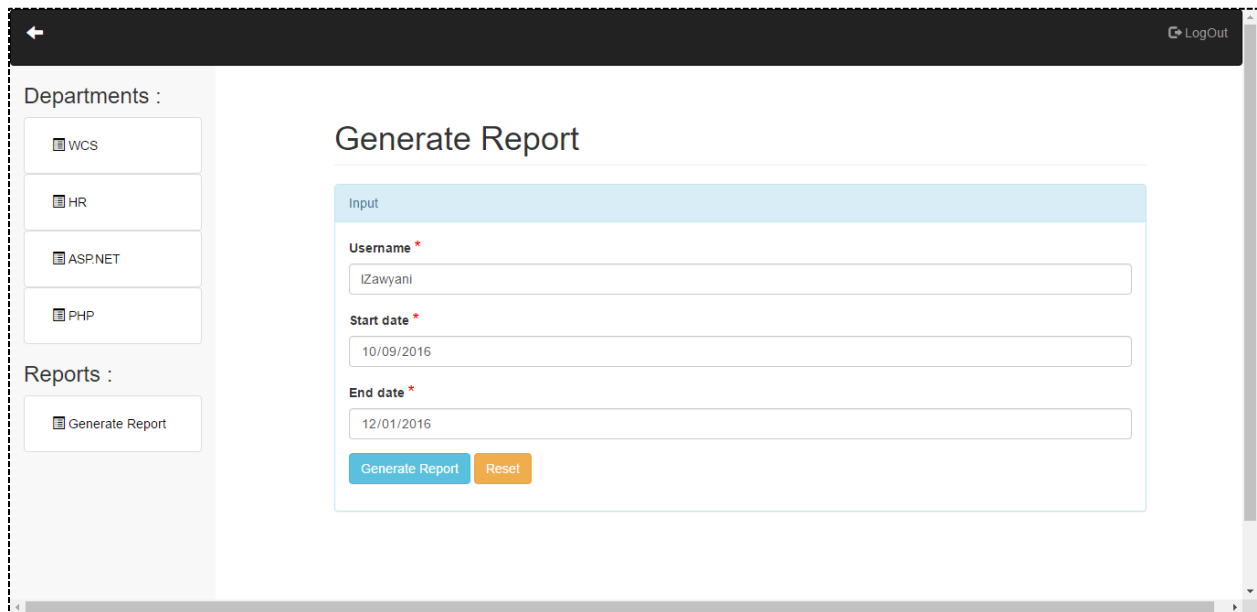


Figure 44: Generate Report Form

After he clicked on “generate report” button, he will be forwarded to a page that has a table which includes the projects that the selected employee worked on them within the selected period and the tasks he worked on it for each project. It also includes the dates he worked and how many hours the employee worked in that date. If the employee worked in a date that is after the task estimated end date, then the task and that date will be in red which means that this task has late worked dates.

Generate Report

*Tasks with red color are out of their estimated dates.

Report For loay Zawyani From 10/9/2016 12:00:00 AM To 12/1/2016 12:00:00 AM

Project Name	Tasks	9/10	10/10	11/10	12/10	13/10	14/10	15/10	16/10	17/10	18/10	19/10	20/10	21/10	22/10	23/10	24/10
<div style="display: flex; justify-content: space-between; align-items: center;"> Project Management 📊 </div>																	
	LogIn	-	-	-	-	-	-	-	-	-	-	-	-	-	3	-	-
	Logout	-	-	-	-	-	-	-	-	-	-	-	4	-	-	-	-
	Mapping	2	-3	-1	-5	-	-	-	-	-	-	-	-	-	-	-	-
<div style="display: flex; justify-content: space-between; align-items: center;"> Team Allocation 📊 </div>																	
	LogIn	-	-	-	-	-	-	-	-	-	-	3	-	-	-	-3	-4
	logout	-	-	-	-	-	-	-	-	-	-	-	3	-	-	-	-
	MainPage	-	-	-	5	-6	-	-3	-	-	-	-	-	-	-	-	-
Total work																	

Figure 45: Generated Report Table

Each project data can be shown as a chart that includes the tasks of this project, the dates when the employee worked on that task and how many hours he worked. It can be generated by clicking on the chart icon that is located beside the project name.

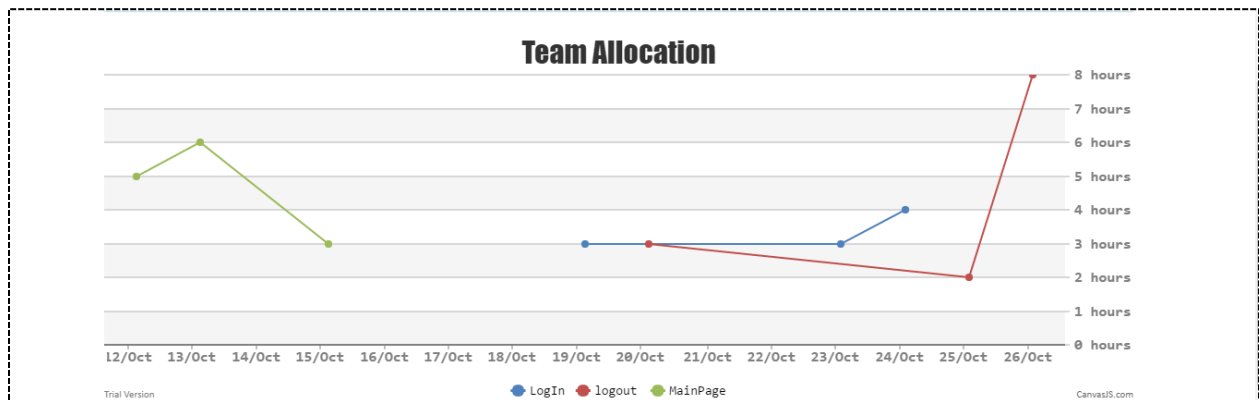


Figure 46: Generated Report Line Chart

5.2 Features in normal employee section:

The normal employee can log in using his name and password. Then he will be forwarded to the normal user page that includes the floors list for the company, his assets tab and his project management tab.

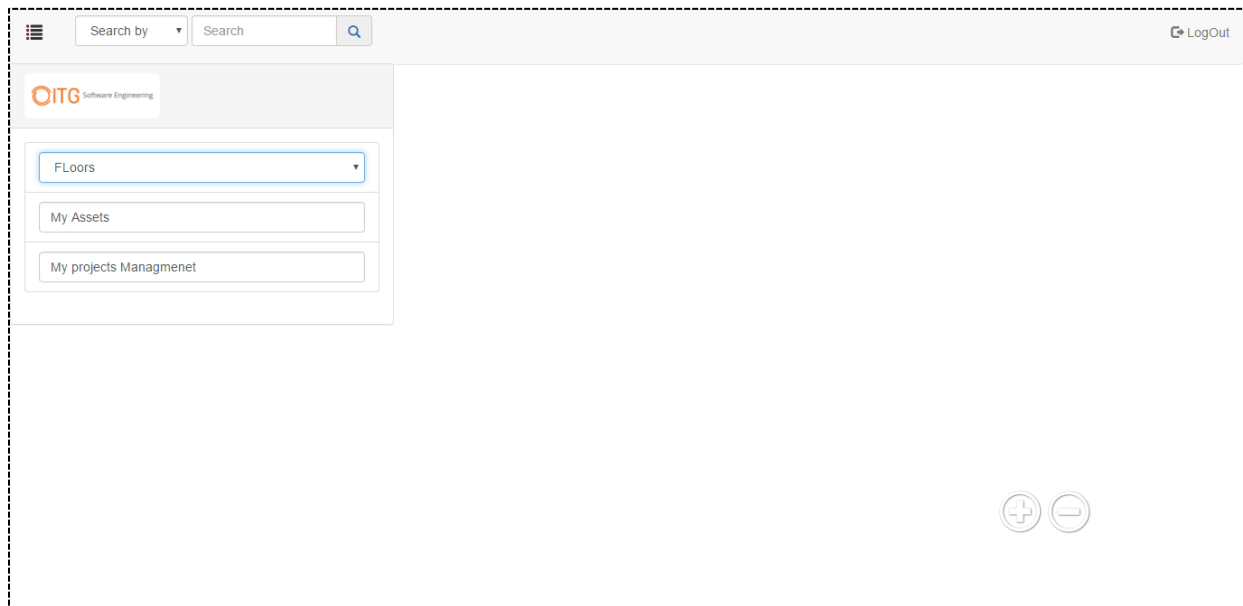


Figure 47: Normal Employee Home Page

- Choosing the floor, Zooming of image with drag and drop, Search and viewing other employees' information:** All these features are the same as the admin section but with small difference that the normal employee can't see all the data of his colleagues.

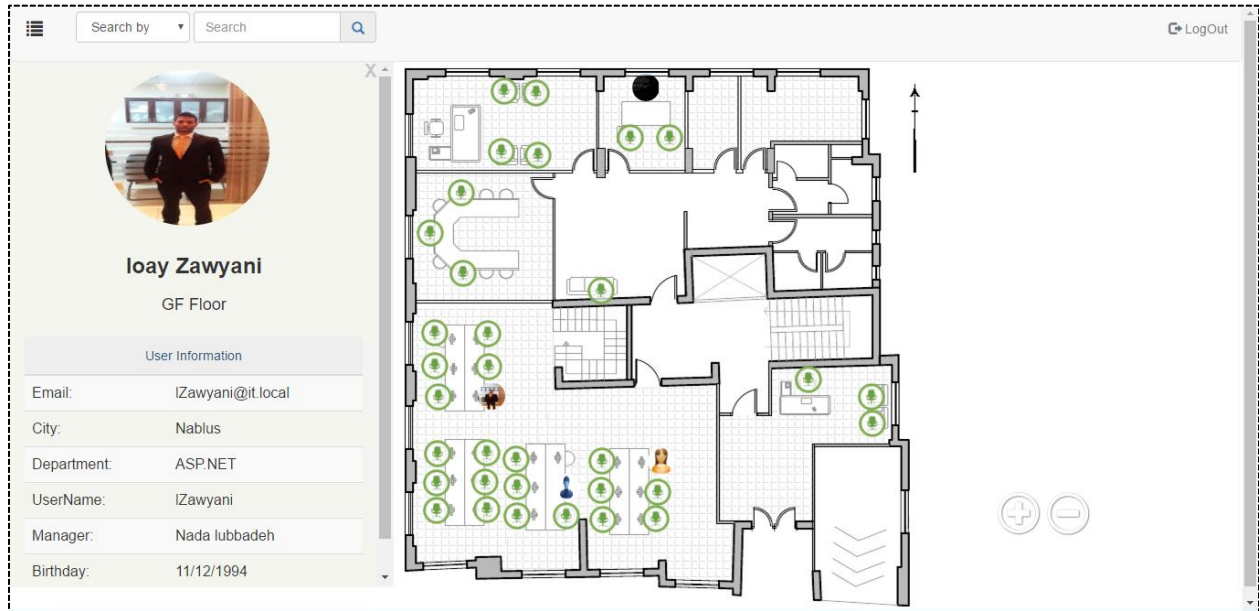


Figure 48: Normal Employee Information Viewing Privileges

- The employee assets:** the employee can see his assets by clicking on “My Assets” tab. After that, he will be forwarded to a page that includes two tabs one for normal assets and one for computers. Each tab will generate a data table that includes the assets or computers he had from the company.

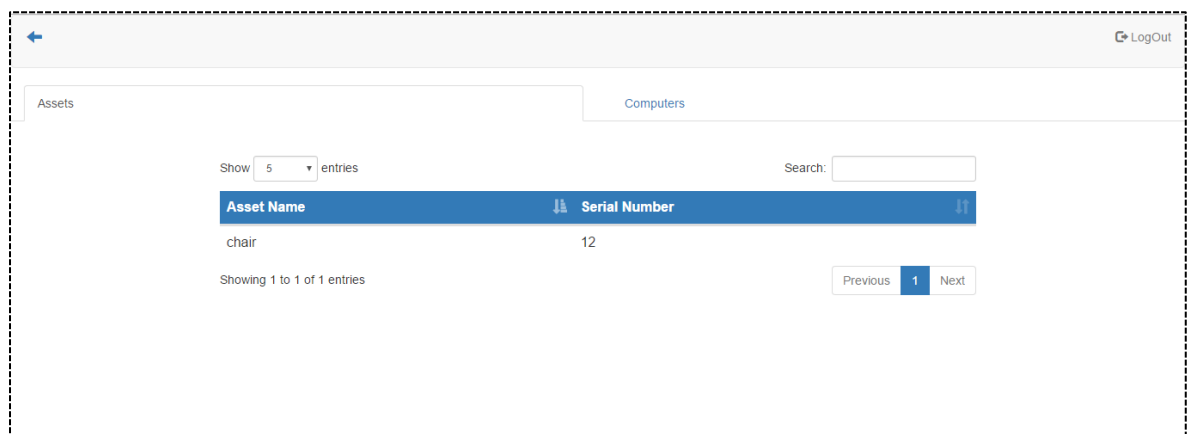


Figure 49: Normal Employee Assets

- **Project management:** when the employee clicks on “my project management” tab he will be forwarded to a page that includes two tabs; one for the projects he is working on and the other for his daily log.
 - When he clicks on “Projects” a data table will be shown that includes his current projects, the start date, end date and the department where this project belongs to.

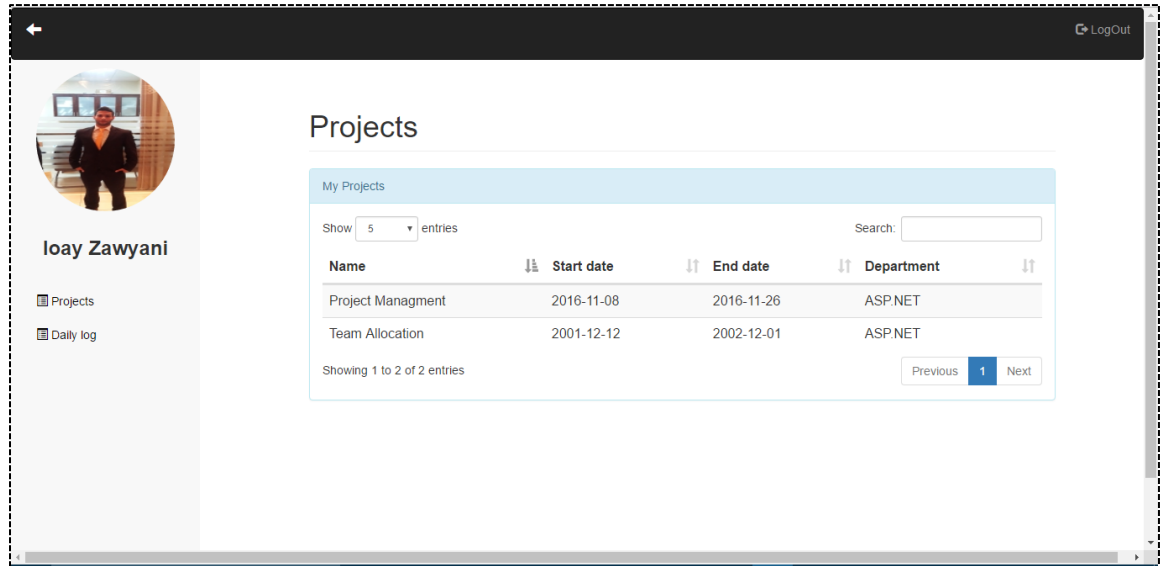


Figure 50: Normal Employee Projects

- The employee must fill the daily log form every day he works on a project task by clicking on the daily log tab and filling the required data. He selects the project and then selects the task he worked on it, enter the log date, fill the working time in hours or minutes and finally set the status for the task if it is still in progress or finished.

Figure 51: Daily Log Form

5.3 Brief description of all features:

A user can login to the website and see or edit information depending on his privileges. Stating the main functionalities of Team Allocation System:

- **User friendly design for the website:** A simple, responsive and a user friendly design that is easy to use, understandable and comfortable for the user.
- **Mapping:**
 - The admin can manage staff locations. He can add floors by uploading the floor plan as a high resolution picture. Then, he adds a seat and books an employee to it depending on his real location in the building and this position can be edited or deleted later. He can also do search based on some filters.
 - The normal user can search other employees and view their position in the company and some information about them.
- **Reports:** Dynamic reports about each department and its employees. These reports can be viewed and edited by the admin in addition to downloading them as PDF.

- **Custody management:** The admin manages the assets in the company by adding asset, editing existing one, delete it or assign it to an employee. Each employee can view his assets from his account.
- **Project Management:** The admin has the permission to manage each department projects by adding projects to it and add tasks for each project then assign employees to them taking into account their load of work which he can determine the policy of it, these tasks are presented by a chart which clarifies the progress of project based on the tasks status. The employee can see his projects, also he fills his daily work at the end of each day. In this way the admin is able to see his progress during a period of time presented by a table and a chart.

5.4 Database Class Diagram:

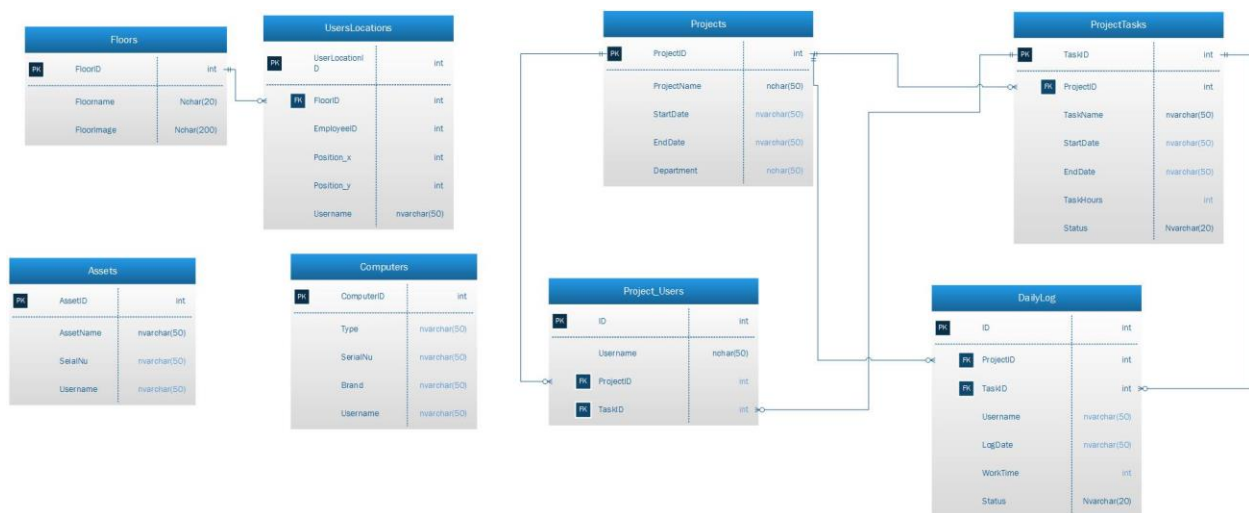


Figure 52: Database Class Diagram

Chapter 6: Conclusion

Fortunately, we were able to achieve all our goals that were written in the first proposal. We were also able to add more features after finishing the main features.

Concluding our results, Team Allocation System is a management website which provides many features. Mapping and dynamic reports are special features in the website since they interpret and manage the active directory information in a user friendly design. Other feature is custody management which is very important to manage the assets in a company. Moreover, project management is a basis in any company especially the daily log feature in it. The daily log is important in the case of outsourcing companies since the employees' logs their daily work details on the company system they are working for outside their original company. By daily log the admin will be able to view these details to track any employee's work by generating a report about it.

During our work we learned many new concepts in programming and building applications. We learned how to think from the view of the user to design a user friendly design. On the other hand, we learned to apply some of the security basics in the website.

However, we were limited by the time set by the university. But, this will not prevent us from keep working on the project with ITG Company and add new features to it. We hope to add an attendance feature which means that the employee will be shown online while he is in the company or as an enhancement to this, he will be shown online while he is interacting with his computer. Another feature to add is meeting rooms booking which means that the admin will be able to book a meeting room for a period of time and the room will be shown busy to other users.

After all, we can say that Team Allocation System will play an important role in the field of information management since all its features are widely required by nowadays companies

References

- AD PHOTO, E. (2012). */Software/ADPhotoEdit/Info.html*. Retrieved from cjwdev: <http://www.cjwdev.co.uk>
- Bitrix. (2001-2016). Retrieved from bitrix24: <https://www.bitrix24.com/>
- BrowseLab, I. (2016). */features.htm*. Retrieved from staffmap: <http://www.staffmap.com/>
- Charts, J. (2013). */javascript-charts/*. Retrieved from canvasjs: <http://canvasjs.com>
- DataTables Table plug-in, j. (2007-2016). */download/index*. Retrieved from datatables: <https://datatables.net/>
- Francis, D. M. (2013). */2011/03/step-by-step-guide-to-setup-active-directory-windows-server-2008/*. Retrieved from rebeladmin: <http://www.rebeladmin.com>
- iTextSharp, a. .. (2016). */projects/itextsharp/*. Retrieved from sourceforge Web site: <https://sourceforge.net/projects/itextsharp/>
- ITGCompany. (2016). Connecting to active directory. Nablus, Palestine.
- Morgan. (2013, August 21). */2013/08/how-to-create-custom-attribute-in.html*. Retrieved from morgantechspace: <http://www.morgantechspace.com>
- PDFEditor. (2007-2016). */ad/pdf-editor/?gclid=CMfty7vm4dACFYQW0wodBzoOOQ*. Retrieved from wondershare: <http://www.wondershare.net>
- Shinder, T. (2009, September 24). */articles-tutorials/windows-server-2008/Running-Windows-Server-2008-R2-Installing-Creating-Lab-Domain-Controller-Part1.html*. Retrieved from windownetworking: <http://www.windownetworking.com>
- Smith, O. (2013). */morris.js/*. Retrieved from morrisjs: <http://morrisjs.github.io>
- Startbootstrap. (n.d.). */startbootstrap-creative/*. Retrieved from blackrockdigital: <https://blackrockdigital.github.io>
- TreeFy Tables, B. -j. (2012-2016). */table/jQuery-Plugin-To-Create-Collapsible-Bootstrap-Tables-Bootstrap-TreeFy.html*. Retrieved from jqueryscript: <http://www.jqueryscript.net>
- Willison, T. (2016). */timmywil/jquery.panzoom*. Retrieved from github: <https://github.com>

Appendices

Appendix A:

- **Connect the project with active directory:** (ITGCompany, 2016)
 - The first part of the code is written in the **web.config** file. It is used to connect the application with the domain controller and to define which OU we want to use and connect with it.

```
<appSettings>  
  <add key="webpages:Version" value="3.0.0.0" />  
  <add key="webpages:Enabled" value="false" />  
  <add key="ClientValidationEnabled" value="true" />  
  <add key="UnobtrusiveJavaScriptEnabled" value="true" />  
  <add key="Domain" value="it.local" />  
  <add key="Path" value="LDAP://it.local/OU=Employees,DC=it,DC=local" />  
</appSettings>
```

Figure 53: Determine the Path of Domain Code

- The second part is a function inside a class called **ActiveDirectoryManager**. This function is used to get the domain from the **web.config**, create credentials and create the connection to check if the credentials exists.

```

public static bool ValidateCredentials(string userName, string password)
{
    //getting the domain from web.config
    var domain = ConfigurationManager.AppSettings["Domain"];
    loginUsername = userName;
    loginPassword = password;
    //create credentials
    NetworkCredential credentials
        = new NetworkCredential(userName, password, domain);

    // create the connection to check if the credentials exists
    LdapDirectoryIdentifier id = new LdapDirectoryIdentifier(domain);
    using (LdapConnection connection = new LdapConnection(id, credentials, AuthType.Kerberos))
    {
        connection.SessionOptions.Sealing = true;
        connection.SessionOptions.Signing = true;
        try
        {
            connection.Bind();
        }
        catch (LdapException lEx)
        {
            .
        }
    }
}

```

Figure 54: Validate User Credentials Code

- The third part is inside the controller to check if the username and password is valid and exists inside the active directory, we did some modifications on this part of code to be applicable and work with our application.

```

public ActionResult CheckLogIn(LogInData login, string returnUrl)
{
    if (ModelState.IsValid)
    {
        try
        {
            if (ActiveDirectoryManager.ValidateCredentials(login.UserName, login.Password))
            {
                .
            }
        }
    }
}

```

```

//ActiveDirectoryManager.GetFullNamebyUserName(login.UserName);
//FormsAuthentication.SetAuthCookie(login.UserName, false);
string[] userRoles = ActiveDirectoryManager.GetUserRoles(login.UserName);
FormsAuthenticationTicket authTicket =
    new FormsAuthenticationTicket
    (1,
    login.UserName,
    DateTime.Now,
    DateTime.Now.AddHours(1),
    false,
    JsonConvert.SerializeObject(new User
    {
        UserName = login.UserName,
        Fullname = ActiveDirectoryManager.GetFullNamebyUserName(login.UserName)
    }));

string encTicket = FormsAuthentication.Encrypt(authTicket);

HttpCookie faCookie = new HttpCookie(FormsAuthentication.FormsCookieName, encTicket);

Response.Cookies.Add(faCookie);
Session["username"] = login.UserName;
if (userRoles.Length == 0)
{
    return RedirectToAction("NormalUser");
}
else if (userRoles[0] == "Administrators")
{
    return RedirectToAction("Admin");
}
}
catch (LdapException lEx)
{
    ModelState.AddModelError("", "Check your connection to the server, please!");
    return View("Index", login);
}
}
ModelState.AddModelError("", "Invalid User Name Or Password");
return View("Index", login);
}
}

```

Figure 55: Check Login Code

- **Download information from a web page as PDF:** (iTextSharp, 2016)

```
using iTextSharp.text;  
using iTextSharp.text.html.simpleparser;  
using iTextSharp.text.pdf;
```

```
public void GenerateInvoicePDF(string departmentname)  
{  
    ReportsData data = new ReportsData();  
    data = ActiveDirectoryManager.GetDepartmentData(departmentname);  
    string companyName = "Infinite Tiers Company";  
    using (StringWriter sw = new StringWriter())  
    {  
        using (HtmlTextWriter hw = new HtmlTextWriter(sw))  
        {  
            StringBuilder sb = new StringBuilder();  
            sb.Append("<table width='100%' cellspacing='0' cellpadding='2'>");  
            sb.Append("<tr><td align='center' style='background-color: #18B5F0' colspan = '2'><b>"  
                + "" + companyName + "</b></td></tr>");  
            sb.Append("<tr><td colspan = '2'></td></tr>");  
            sb.Append("<tr><td><b>Department: </b>");  
            sb.Append(departmentname);  
            sb.Append("<td align = 'right'><b>Date: </b>");  
            sb.Append(DateTime.Now);  
            sb.Append(" </td></tr>");  
            sb.Append("<tr><td align='center' style='background-color: #18B5F0' colspan = '2'><b>"  
                + "|Leader:" + data.employees[0].manager + "</b></td></tr>");  
            sb.Append("<tr><td colspan = '2'></td></tr>");  
        }  
    }  
}
```

```

sb.Append("</table>");
sb.Append("<br />");
////////////////////////////////////

sb.Append("<table border='1'>");
sb.Append("<tr style='background-color:Blue;'>");
//sb.Append("FirstName</th>");
//sb.Append("<th>LastName</th>");
sb.Append("<th>FullName</th>");
sb.Append("<th>Username</th>");
sb.Append("<th>City</th>");
sb.Append("<th>Email</th>");
sb.Append("<th>Telephone#</th>");
sb.Append("</tr>");
for (var i = 0; i < data.employees.Count; i++)
{
    sb.Append("<tr>");
    sb.Append("<td>");
    sb.Append(data.employees[i].fullName);
    sb.Append("</td><td>");
    sb.Append(data.employees[i].username);
    sb.Append("</td><td>");

    sb.Append(data.employees[i].city);
    sb.Append("</td><td>");
    sb.Append(data.employees[i].email);
    sb.Append("</td><td>");
    sb.Append(data.employees[i].telephoneNumber);
    sb.Append("</td>");
    sb.Append("</tr>");
}

sb.Append("</table>");

//Export HTML String as PDF.
StringReader sr = new StringReader(sb.ToString());
Document pdfDoc = new Document(PageSize.A4, 10f, 10f, 10f, 0f);
HTMLWorker htmlparser = new HTMLWorker(pdfDoc);
PdfWriter writer = PdfWriter.GetInstance(pdfDoc, Response.OutputStream);
pdfDoc.Open();
htmlparser.Parse(sr);
pdfDoc.Close();
Response.ContentType = "application/pdf";
Response.AddHeader("content-disposition", "attachment;filename=" + departmentname + ".pdf");
Response.Cache.SetCacheability(HttpCacheability.NoCache);

```

Figure 56: Download PDF file Code