1. **INTRODUCTION**

**1. INTRODUCTION:**

### ASP .NET MVC:

The Model-View-Controller (MVC) architectural pattern separates an application into three main components: the model, the view, and the controller. The ASP.NET MVC framework provides an alternative to the ASP.NET Web Forms pattern for creating Web applications. The ASP.NET MVC framework is a lightweight, highly testable presentation framework that (as with Web Forms-based applications) is integrated with existing ASP.NET features, such as master pages and membership-based authentication.

### MODEL:

 Model objects are the parts of the application that implement the logic for the application's data domain. Often, model objects store and retrieve model state in a database. For example, a Product object might retrieve information from a database, operate on it, and then write updated information back to a Products table in a SQL Server database.

In small applications, the model is often a conceptual separation instead of a physical one. For example, if the application only reads a dataset and sends it to the view, the application does not have a physical model layer and associated classes. In that case, the dataset takes on the role of a model object.

### VIEW:

  Views are the components that display the application's user interface (UI). Typically, this UI is created from the model data. A view can be any output representation of information, such as a chart or a diagram. Multiple views of the same information are possible, such as a bar chart for management and a tabular view for accountants. An example would be an edit view of a Products table that displays text boxes, drop-down lists, and check boxes based on the current state of a Product object.

### CONTROLLER:

The controller accepts input and converts it to commands for the model or view.  These are the components that handle user interaction, work with the model, and ultimately select a view to render that displays UI. In an MVC application, the view only displays information; the controller handles and responds to user input and interaction. For example, the controller handles query-string values, and passes these values to the model, which in turn might use these values to query the database.

## LOCATION MAPPING/RESTACKING TOOL:

## This tool provides a single window view to help real estate management. It helps to strategize the move globally, across facilities to make optimum and precise usage of available space. It helps senior managers to identify the location needs and plan accordingly. It also makes them to review existing resource or group or an organisation unit, current location. This tool will also be a great help during emergency relocation of teams. Region based administration roles can be created and region based location can be maintained separately. Major restacking can be done with minimal effort. Details/Extract of this application can be provided to other application as services. This tool deals with Digital Workplace and Mysource to explore where this tool best can be applied for enterprise usage.

**3. PROBLEM STATEMENT**

* 1. **PROBLEM DEFINITION :**

The location mapping or restacking of employees/teams within the organization was currently maintained using Excel files.  As it is manual process, there was problem in effectively managing the available space and time consuming. In addition, there was no central control or reporting capabilities. In order to overcome these, we created service for mapping location and restacking using MVC .NET framework.

**4. SYSTEM CONFIGURATION**

**4.1 HARDWARE REQUIREMENTS:**

The minimum hardware configuration of the system on which the project was developed is as follows:

* + - Processor : Intel(R) Core(TM) 3.2 Ghz
    - CPU clock : 450 Mhz
    - RAM : 8 GB
    - HardDisk : 40 GB
    - Keyboard : Any type
    - Mouse : Any type
    - Monitor : Any type/Color Monitor preferable

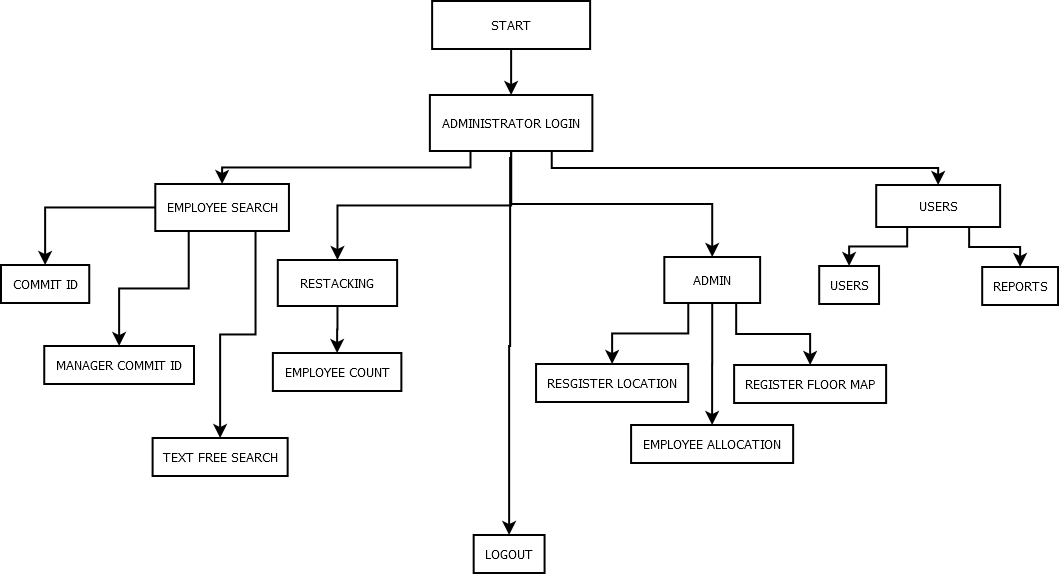
### 4.2 SOFTWARE REQUIREMENTS:

* Operating System : Windows 7
* Language Used : C sharp(Visual Studio 2013)

**5. METHODOLOGY**

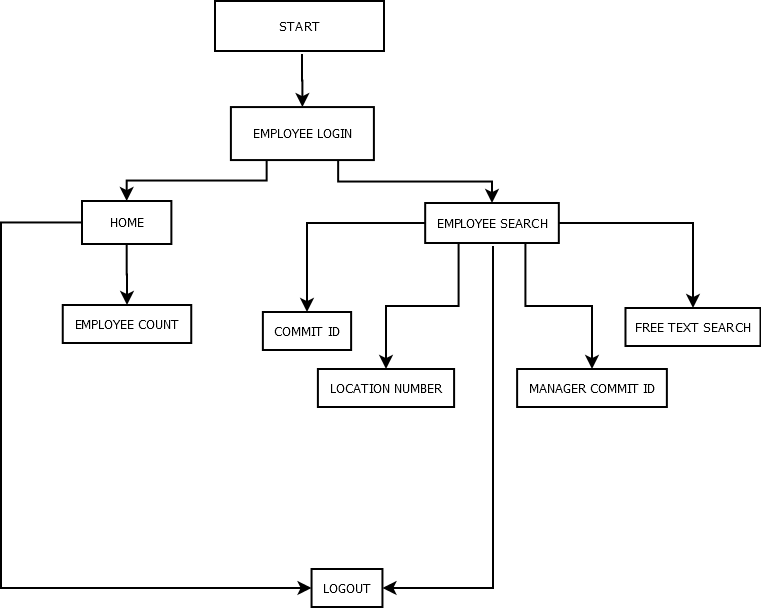
**5.1 BLOCK DIAGRAM:**

**5.1.1 ADMINISTRATOR LOGIN:**

****

**Figure 5.1**

**5.1.2 EMPLOYEE LOGIN:**

****

**Figure 5.2**

**5.2 MODULES:**

* Login module
* Administrator Module
* Employee Module
  1. **MODULE DESCRIPTION:**

### 5.3.1 LOGIN:

### Login in the iLoc tool is based on Membership. ASP.NET provides built in controls to manage Membership in Web Applications. All these controls use ASP.NET providers that are loaded via web.config file. Membership provider and Role provider allow a complete system to maintain user’s information, authenticate and authorize the users.

### 5.3.2 ADMINISTRATOR MODULE:

### Register Floor Map-Option to add new location map into the tool. Floor map image file can be any bit map file`

### Register Location-Option to add new location on the newly added map. Tool has the option to insert consecutive location numbers by a single Click

### Employee Allocation-Option to allocate location to the Employee by clicking the unallocated location and provide employee commit id.

### Employee Search- Search employee’s location by providing commit id.

### Employee, Group Search-Search employee’s or group/Division by location by manager id

### Restacking-Based on count of employees, Tool will provide the best available locations and provides a single view.

### User Administration-To create application level users and assign roles

### 5.3.3 EMPLOYEE MODULE:

### Employee Search- Search employee’s location by providing commit id.

### Employee, Group Search-Search employee’s or group/Division by location by manager id

* 1. **PROPOSED SYSTEM:**

1. In the iLoc tool, there are types of Login: Administrator Login and Employee Login.
2. If the Employee gets logged in, he/she can only view the other employee’s information along with the location of them.
3. If an administrator gets logged in, he/she can do prioritized works such as

* Register Location
* Register Floor Map
* Add other employee’s location
* Delete other employee’s location
* Modify other employee’s location
* Reallocate employees location

1. The proposed system is in such a way that input, processing and the output is separated by means of Model, View and Controller.
2. A Controller returns a Json or a View. A View basically displays an interface to the user such that it may be a list or particular details.
3. A Model is similar to a class but there exists a DAL (Data Access Layer) where it interacts with the database and displays the result in the view.
4. In addition to DAL, we can have BLL (Business Logic Layer) that encodes the real-world business rules which determine how data can be created, stored and changed.
5. We can have entity-framework where we don’t need to have the code to get interacted with the database. Instead everything is automated using LINQ (Language integrated Query) – Uniform programming model for any kind of data.

### BENEFITS OF NEW SYSTEM:

* Generic to enable using the tool for any kind of facility Offices, Parking lots, etc.
* Enhance operational efficiency.
* Eliminates/Minimize UDT usage and leverage optimization of available space.
* Major restacking can be done with very minimal effort.
* Region based administrators roles can be created and region based location can be maintain separately
* Availability reports to review and strategize resource movement across cross facilities.
* Details/Extract can be provided to other application as services.

1. **RESULTS AND DISCUSSION**

**6.1 SCREENSHOTS:**

The below figure depicts the Login Page.

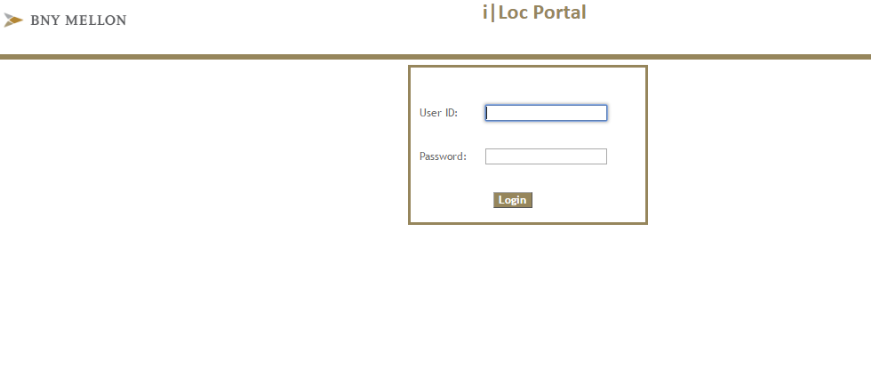


Figure 6.1



Figure 6.2

This figure shows the home page with the count of employees in a specified location.

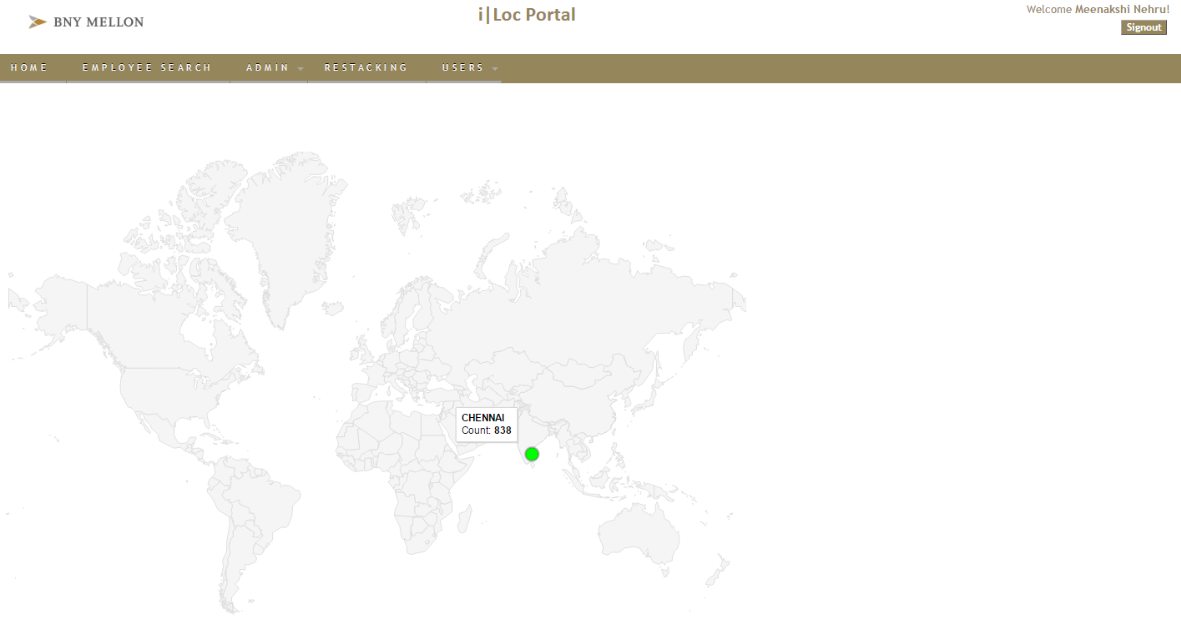


Figure 6.3

This figure shows that the employee search can be done by means of Commit ID, Location Number, Manager Commit ID or Department

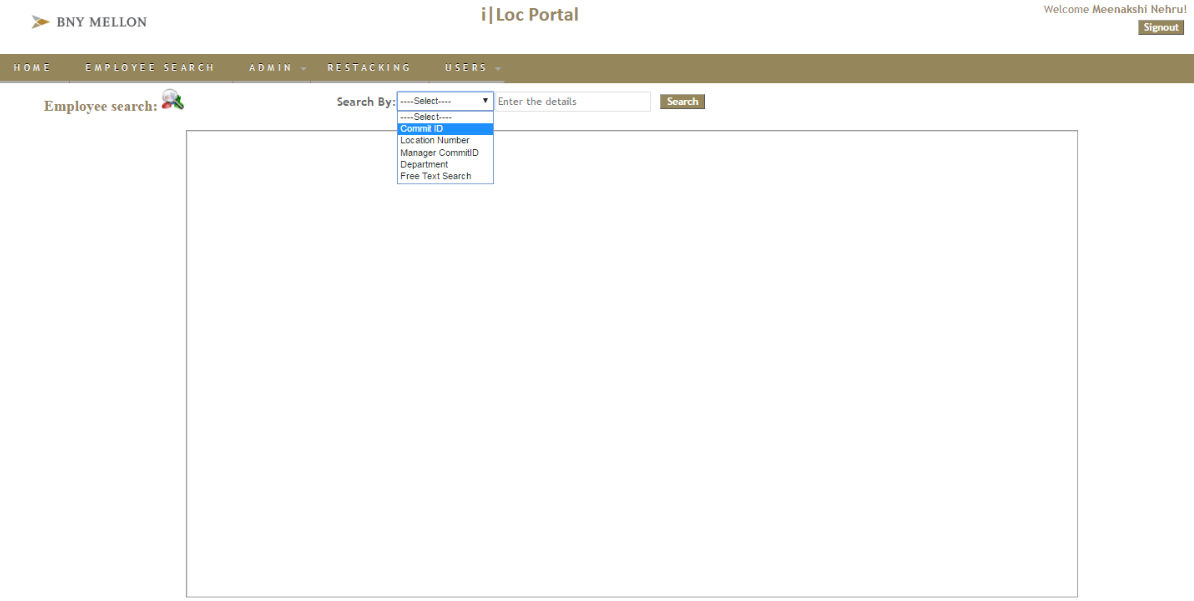


Figure 6.4

The below figure indicates the employee search by means of Commit ID.

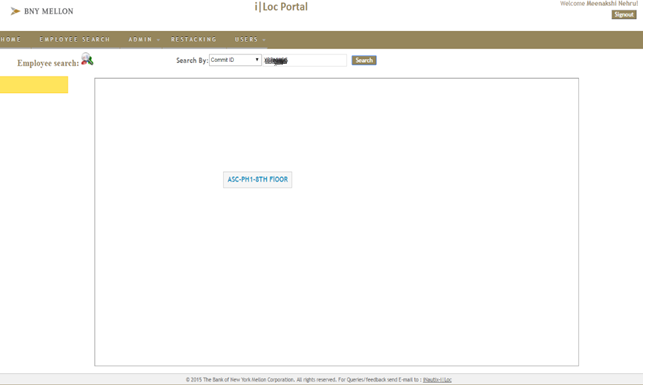
****

Figure 6.5

The below figure shows the employee’s location details.

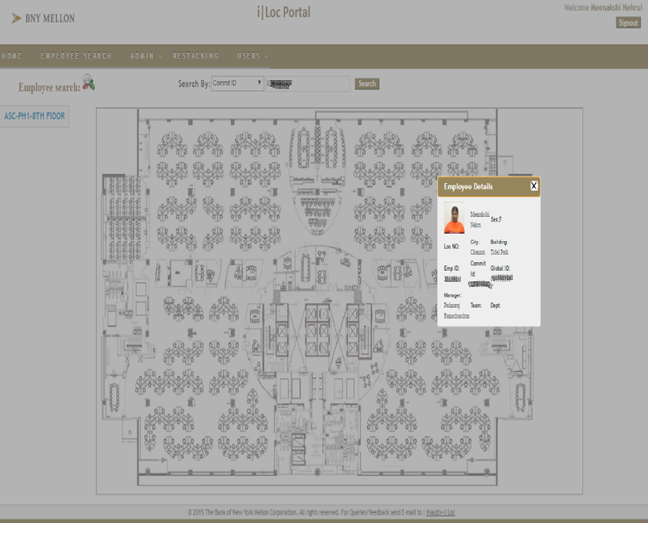
****

Figure 6.6

The below figure shows the employee search through Manager Commit ID.

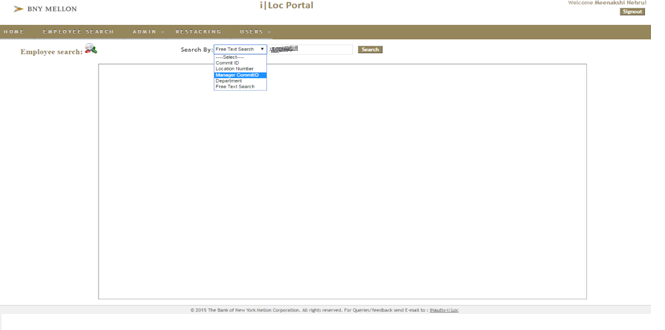


Figure 6.7

This figure shows the employees allotted under the mentioned manager.

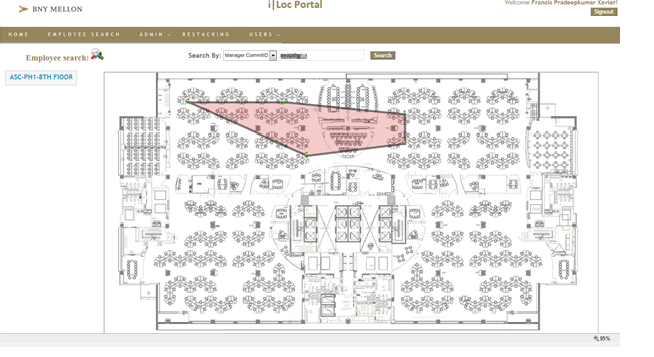
****

Figure 6.8

This figure shows the administrator’s role i.e. Register Floor Map, Register Location and Employee Allocation.

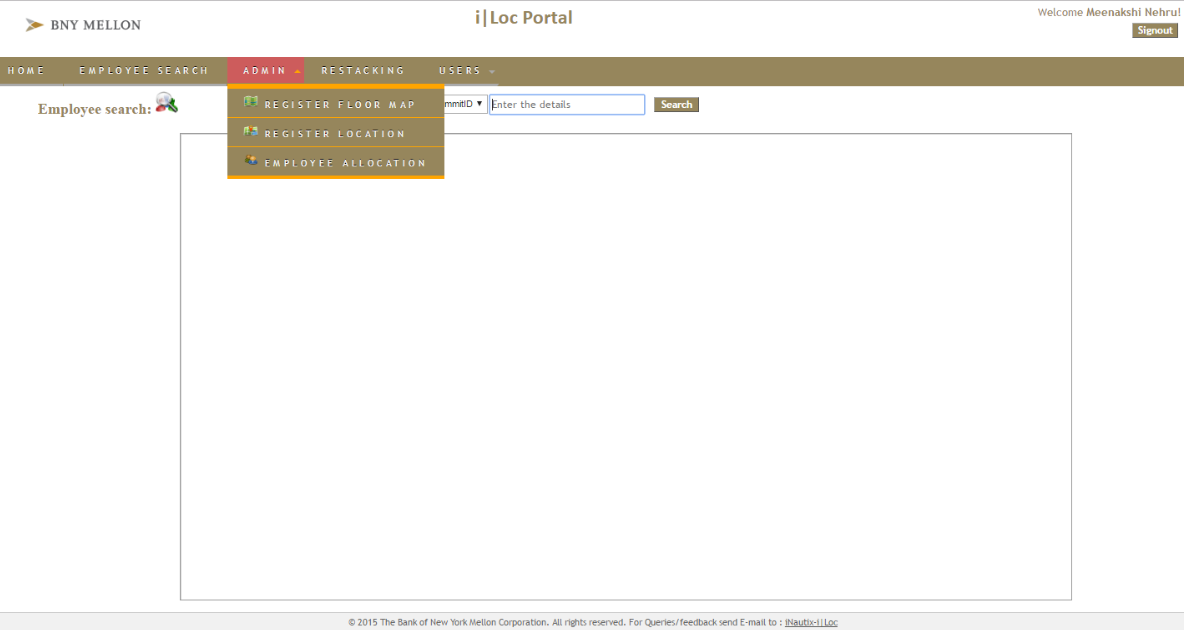


Figure 6.9

The below figure shows the Floor Map registration.

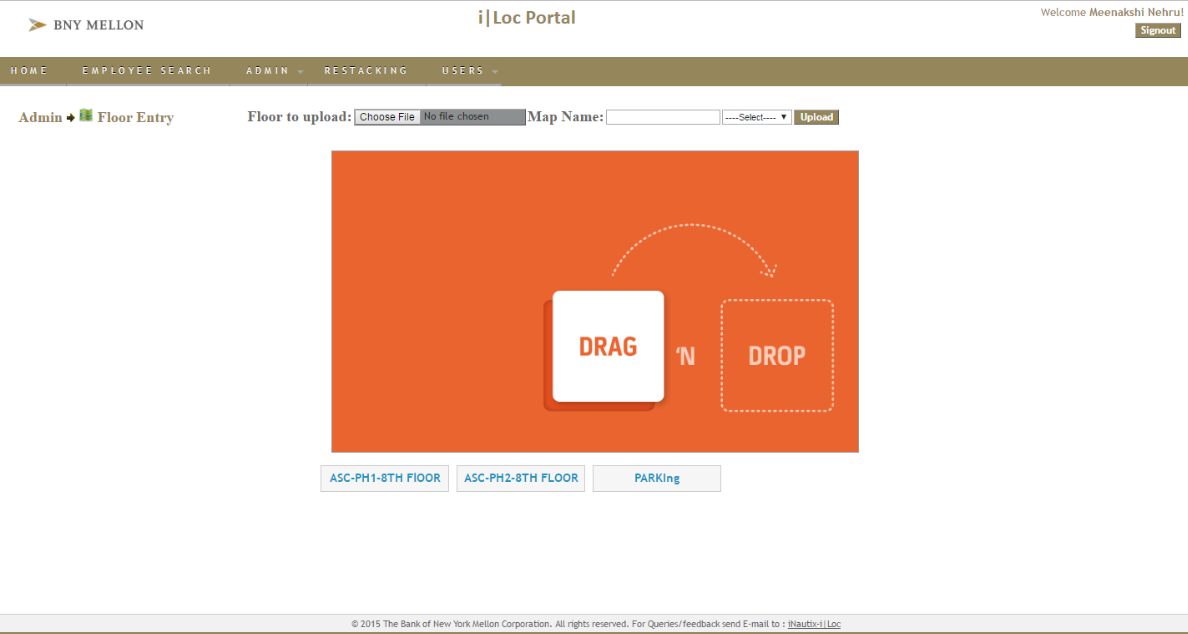


Figure 6.10

This figure depicts the location registration.

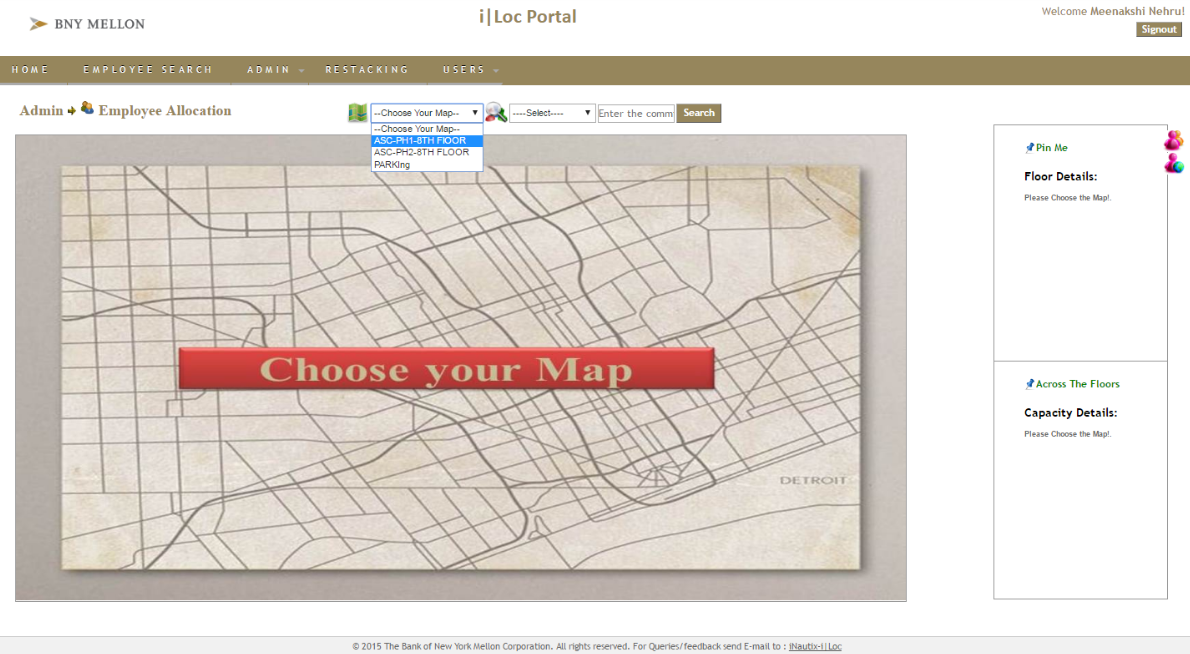


Figure 6.11

This figure depicts the employee allocation.

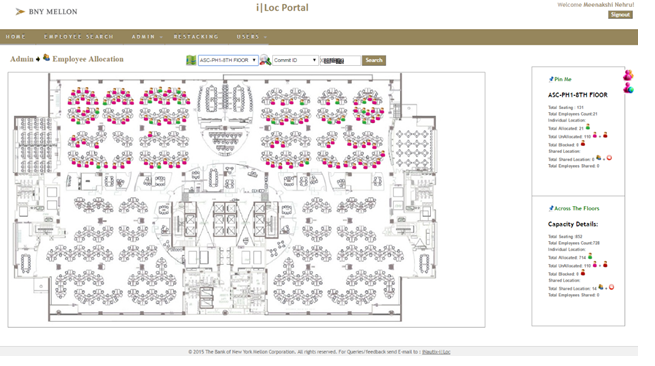
****

Figure 6.12

The below figure shows that the employee is successfully allocated and we can also reallocate them.

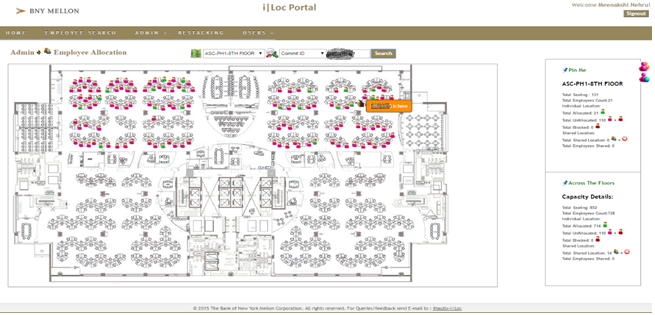


Figure 6.13

The below figure shows the available space where the employees can be newly allocated or reallocated.

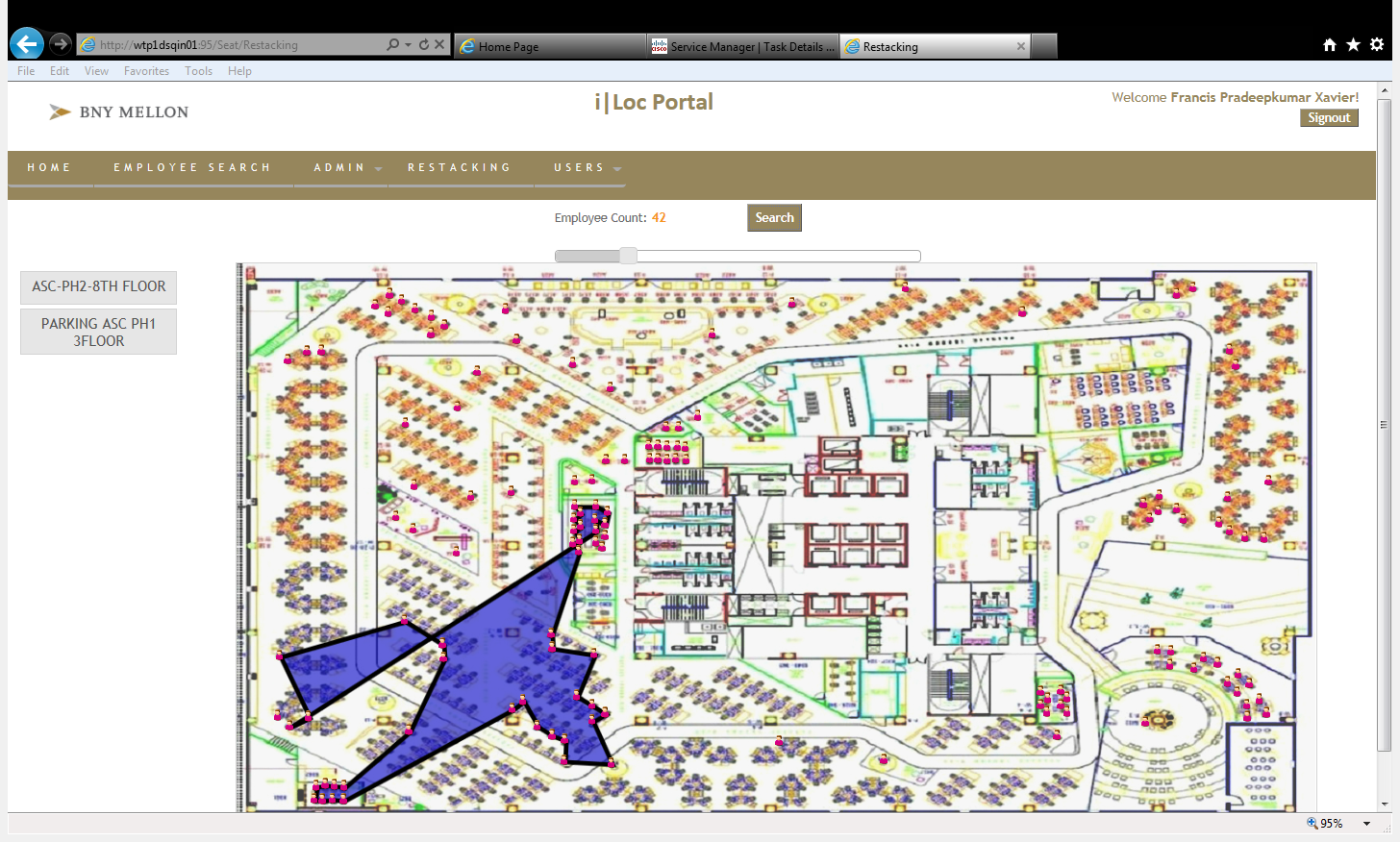


Figure 6.14

The below figure shows the roles of each employee which can be newly added, edited or deleted.

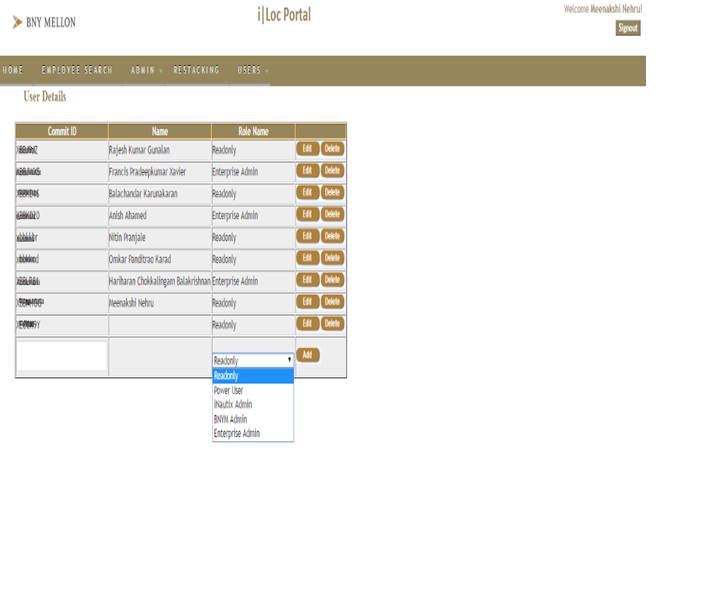


Figure 6.15

* 1. **CONCLUSION AND FUTURE WORK:**

This tool has proposed a web service for location mapping/ restacking any employee of the organization. It provides an efficient manner of locating an employee within a single window view. The administrator uploads the floor map, assigns locations and then allocates the employees to their particular locations using their Commit ID. The employees can view the location of any other employee or can also locate the whole team using the employee’s manager ID. This tool helps in emergency relocation of teams whenever there is a new project allocation.

In future, the tool is going to be expanded for the Global Cash Flow Management for BNY Mellon Company. And also collaborating with the Digital Pulse team, this tool can be explored and put to enterprise usage.

1. **REFERENCES**

**REFERENCES:**

1. Dasgupta K, Mandal B, Dutta P, Mandal JK, Dam S. A Genetic Algorithm (GA) based load balancing strategy for cloud computing. Proc Technol 2013;10:340–7.

2. Pop F, Dobre C, Cristea V. Genetic algorithm for DAG scheduling in grid environments. In: 5th IEEE int conf intell comput commun process; 2009. p. 299 305

3. Ge Y, Wei G. GA-based task scheduler for the cloud computing systems. In: Proc int conf web inf syst min, vol. 2; 2010. p. 181–6. http://dx.doi.org/10.1109/WISM.2010.87

4. Zheng Z, Wang R, Zhong H, Zhang X. An approach for cloud resource scheduling based on parallel genetic algorithm. In: 3rd IEEE int conf comput res dev. IEEE; 2011. p. 444–7. <http://dx>. doi.org/10.1109/ICCRD.2011.5764170

5. Zhao C, Zhang S, Liu Q, Xie J, Hu J. Independent tasks scheduling based on genetic algorithm in cloud computing. In:5th int conf wirel commun netw mob comput; 2009. p. 1–4.http://dx.doi.org/10.1109/WICOM.2009.5301850.

6. Ghorbannia Delavar A, Aryan Y. HSGA: a hybrid heuristic algorithm for workflow scheduling in cloud systems. ClusterComput 2014;17:129–37. http://dx.doi.org/10.1007/s10586-013-0275-6.

7. Joseph CT, Chandrasekaran K, Cyriac R. A novel family genetic approach for virtual machine allocation. Proc Computer Science

8.Gu J, Hu J, Zhao T, Sun G. A new resource scheduling strategy based on genetic algorithm in cloud computing environment.J Comput 2012;7:42–52.

9. Kaur S, Verma A. An efficient approach to genetic algorithm for task scheduling in cloud computing environment. Int J InfTechnol Comput Science 2012;4:74–9.

10. Wang T, Liu Z, Chen Y, Xu Y, Dai X. Load balancing task scheduling based on genetic algorithm in cloud computing. In: IEEE 12th int conf dependable auton secur comput; 2014. p.146–52.

11. Zhu K, Song H, Liu L, Gao J, Cheng G. Hybrid genetic algorithm for cloud computing applications. In: IEEE Asia-Pacific serv comput conf; 2011. p. 182–7.

12. Tao F, Feng Y, Zhang L, Liao TW. CLPS-GA: a case library and Pareto solution-based hybrid genetic algorithm for energyaware cloud service scheduling.

13. Xiong A, Xu C. Energy efficient multiresource allocation of virtual machine based on PSO in cloud data center. Math Probl Eng 2014:2014.

14. Bahman Keshanchi a , Alireza Souri a , Nima Jafari Navimipour b , ∗ An

improved genetic algorithm for task scheduling in the cloud environments

using the priority queues: Formal verification, simulation, and statistical

testing, The Journal of Systems and Software, *124July 17 (2017) 1–21.*

15. Saima Gulzar Ahmada, Chee Sun Liewa, Ehsan Ullah Munirb, Ang Tan

Fonga,Samee U. Khanc,∗ , A hybrid genetic algorithm for optimization of

scheduling workflow applications in heterogeneous computing systems,Journal of Parallel Distribution Computation.

16. ZHANG Yan-huaa, Feng Leia, Yang Zhia, a\*,Optimization of Cloud

Database Route Scheduling Based on Combination of Genetic Algorithm and

Ant Colony Algorithm, Advanced in Control Engineering and Information

Science, *15 ( 2011 ) 3341 – 3345.*

17.Israel Casasa,b,∗, Javid Taheric, Rajiv Ranjanb,e, Lizhe Wangd, Albert

Y. Zomayaa, GA-ETI: An enhanced genetic algorithm for the scheduling of

scientificworkflows in cloud environments, Journal of Computational Science,

JOCS-540, 14, August 2016.

18. HAO-CHIN CHANG, YEH-PENG CHEN, TUNG-KUAN LIU, AND

JYH-HORNG CHOU, Solving the Flexible Job Shop Scheduling Problem

With Makespan Optimization by Using a Hybrid Taguchi-Genetic Algorithm,

Solving the FJS Scheduling Problem, VOLUME 3, 2015.

19. JASRAJ MEENA, MALAY KUMAR, AND MANU VARDHAN, Cost

Effective Genetic Algorithm for Workflow Scheduling in Cloud Under Deadline Constraint, CEGA for Workflow Scheduling in Cloud, VOLUME 4, (5065-5080) 2016.

20. Saima Gulzar Ahmada, Chee Sun Liewa, Ehsan Ullah Munirb, Tan Fong

Anga, Samee U. Khanc A Hybrid Genetic Algorithm for Optimization of

Scheduling Workflow Applications in Heterogeneous Computing Systems,

Journal Parallel Distribution Computation, S0743-7315(15)00186-0, 9 October 2015.

21. Saima Gulzar Ahmada, Chee Sun Liewa, Ehsan Ullah Munirb, Tan Fong

Anga, Samee U. Khanc A Hybrid Genetic Algorithm for Optimization of

Scheduling Workflow Applications in Heterogeneous Computing Systems,

Journal Parallel Distribution Computation, S0743-7315(15)00186-0, 9

October 2015.

22. HAO-CHIN CHANG, YEH-PENG CHEN, TUNG-KUAN LIU, AND

JYH-HORNG CHOU, Solving the Flexible Job Shop Scheduling Problem

With Makespan Optimization by Using a HybridTaguchi-Genetic Algorithm,

Solving the FJS Scheduling Problem, VOLUME 3, 2015.

23. JASRAJ MEENA, MALAY KUMAR, AND MANU VARDHAN, Cost

Effective Genetic Algorithm for Workflow Scheduling in Cloud Under Deadline Constraint, CEGA for Workflow Scheduling in Cloud, VOLUME 4,

(5065-5080) 2016.