Tribhuvan University

Faculty of Humanities and Social Sciences

TASK ASSIGNING SYSTEM

A PROJECT REPORT

Submitted to

Department of Computer Application

Ratna Rajyalaxmi Campus, Kathmandu

In partial fulfillment of the requirements for the Bachelor in Computer Applications

Submitted by

Manish Tamang

October 2023

Under the Supervision of

Deepak Thakur

Tribhuvan University

Faculty of Humanities and Social Sciences

Ratna Rajyalaxmi Campus, Kathmandu

SUPERVISOR'S RECOMMENDATION

I hereby recommend that this project prepared under my supervision by **MANISH TAMANG** entitled "**TASK ASSIGNING SYSTEM**" in partial fulfillment of the requirements for the degree of Bachelor of Computer Application is recommended for the final evaluation.

……

SUPERVISOR

Deepak Thakur

Lecturer

Bachelor in Computer Applications

Ratna Rajyalaxmi Campus, Kathmandu

Tribhuvan University

Faculty of Humanities and Social Sciences

Ratna Rajyalaxmi Campus, Kathmandu

LETTER OF APPROVAL

This is to certify that this project prepared by **MANISH TAMANG** entitled "**TASK ASSIGNING SYSTEM**" in partial fulfillment of the requirements for the degree of Bachelor in Computer Applications has been evaluated. In our opinion it is satisfactory in the scope and quality as a project for the required degree.

Deepak Thakur, Lecturer BCA, Ratna Rajyalaxmi Campus	Bhupendra Ram Luhar, Coordinator BCA, Ratna Rajyalaxmi Campus
Internal Examiner	**External Examiner**

ACKNOWLEDGEMENT

Many people have lent a helping hand to make this project a reality. I would like to thank each one of them who is related to this project.

Primarily, I would like to thank the supervisor of this project, **Deepak Thakur**. His guidance, suggestions and supervision of this project were important in keeping the progress of the project in track. I would also like to express my gratitude to our lecturers **Er. Shree Krishna Maharjan, Bhoj Raj Joshi, Bijay Mishra and Kriti Nemkul** under whose guidance, I learned a lot about this project as they always helped me without hesitation. I won't forget to thank our Head of Department, **Mr. Bhupendra Ram Luhar** for his suggestion throughout the project. The suggestions and directions have helped a lot in the completion of this project.

Finally, I would like to thank all my lecturers and friends, who always supported me with their valuable suggestions and guidance and have been very helpful in various stages of project completion.

ABSTRACT

Task assigning is a very arduous and time-consuming task. Assigning task manually through email, memos, conversation, it takes lot of patience and man hours. In the project, the difficulties of assigning tasks is reduced by using the Hopcroft-Karp algorithm. By using this algorithm, the time required to assign the task to the employees is reduced. The system is based on waterfall methodology and incorporates features to facilitate effectiveness task management. The system allows admin/project manager to create and assign tasks, set deadlines, track progress. The system has the potential to improve task management in a variety of organizational settings and can be customized to meet specific user needs.

Results from a pilot study indicates the system improved team productivity and communication, resulting in faster task completion time and increased overall efficiency. Hence, the task is assigned to the employees automatically based on their skills with the aim of producing only reasonable results.

Keywords: ***Task assigning, Task management, Hopcroft-Karp Algorithm***

LIST OF ABBREVIATIONS

CSS	-	Cascading Style Sheet
HTML	-	Hypertext Markup Language
IDE	-	Integrated Development Environment
JS	-	JavaScript
MVC	-	Model View Controller
SDLC	-	Software Development Life Cycle
SSMS	-	SQL Server Management Studio
UI	-	User Interface

LIST OF FIGURES

LIST OF TABLES

Table of Contents

CHAPTER 1: INTRODUCTION

1.1 Introduction

Task assigning is a crucial aspect of project management as it ensures that tasks are assigned efficiently and effectively. A task assigning system is a tool for planning, tracking and coordinating the activities of a team or project. It usually consists of an online interface where project managers can create and assign tasks, track progress and collaborate with employees. A task assigning system can help to streamline the process and make it easier to manage multiple projects at once. The purpose of this project is to outline a task assigning system that is developed using ASP.NET MVC technology. This system allow project managers to assign tasks to the team members, track the progress of each task, and monitor the overall status of the project.

The task assigning system has a user-friendly interface that makes it easy for the project managers to create tasks, assign tasks, set deadlines, add required skills, and monitor the progress of each task. The system also provide team members with the ability to update the status of their tasks and communicate with the project manager.

By implementing a task assigning system, project managers can improve the efficiency and effectiveness of their projects and better meet the needs of their team members. The system provides a streamlined, centralized solution that help project managers to manage multiple projects more effectively.

1.2 Problem Statement

The problem that this project aims to address is the lack of an efficient and effective system for task assigning in project management. Currently, project managers often rely on manual methods such as email, spreadsheets, and sticky notes to assign tasks and track their progress. This can lead to confusion, missed deadlines, and a lack of accountability for team members. Furthermore, manual methods for task assigning do not provide a centralized solution for project management. This can make it difficult for project managers to get a complete picture of the status of multiple projects and to ensure that tasks are assigned efficiently.

1.3 Objectives

The main objectives of this system to overcome some of the problem regarding current system are given below:

- To allow admin (project manager) to automatically assign tasks.
- To use Hopcroft-Karp algorithm to assign tasks to employees.

1.4 Scope and Limitation

1.4.1 Scope

The project aims to develop a task assigning system that streamlines the process of assigning tasks to employees in an organization, thereby reducing the time and effort required for task management. The system utilizes advanced algorithm to allocate tasks to team members based on their skillset and availability. Additionally, it ensures that no employee is overburdened with tasks, and all tasks are completed within the specified deadlines. The system enhances the productivity and efficiency of the organization and provides a seamless task management for employees.

Further research can focus on incorporating machine learning techniques to predict task completion times and optimizing the task allocation process further.

1.4.2 Limitation

For a task assigning system, a limitation could be assuming that all tasks have the same priority. This assumption may not hold in real-world scenarios where some tasks may have higher priority or urgency than others. The system should ideally allow for the assignment of priorities to tasks based on their importance, deadlines, and other relevant factors. This would ensure that the system is flexible and adaptable to different organizational requirements.

Another limitation could be the assumption that all employees have the necessary experience to perform any given task. In reality, some employees may not have the required expertise or experience to perform certain tasks efficiently. This can lead to reduced efficiency and productivity if tasks are assigned to the wrong employees. Therefore, the system should include a feature that allows managers to identify and assign tasks to employees based on their skillset and experience level, ensuring that tasks are completed effectively and efficiently.

1.5 Development Methodology

Waterfall Model is used to develop this system. In waterfall model, each phase must be completed before next phase can begin and there is no overlapping in the phases. This means, output of previous phase works as input to another phase

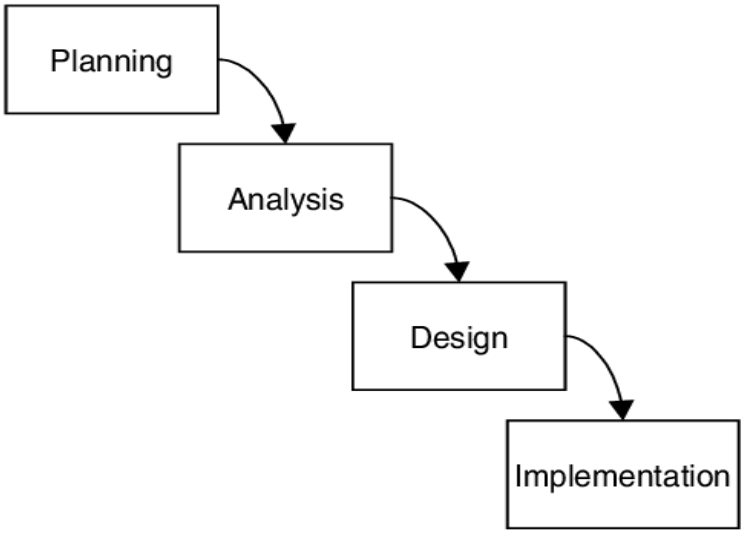


Figure 1.1: Waterfall Model for SDLC

Following steps (phases) of waterfall model has been followed while working on this project.

Planning: In this phase, planning of the structure and platform of the system was done. Ideas from the surroundings and nearby people were gathered. Also gathering of facts about the need of this system was done.

Analysis: Here, analysis of old projects, currently used system and what problems people are facing now are studied. With combined and processed facts, some requirements for the project were finalized.

Design: In this design phase, design of database schemas, interface designs, process modeling, etc. was done. In short, designing of everything from how the system will look alike to how it will work was done. In addition, workflows and architectural designs of the system has been designed.

Implementation: Finally, the coding and testing part started in this phase. Previous phase's outputs are used to input to this phase. That means, each studies and designs made earlier were implemented. Not only that, even testing of this system by preparing different test cases for unit and system testing is done.

1.6 Report Organization

This project report is divided into five chapters. The chapters are as follows:

Chapter 1 contains brief introduction about the project. It includes introduction, problem statement, objectives, scope and limitation, development methodology and report organization of the project.

Chapter 2 contains information that we acquired from background studies and literature reviews on similar projects.

Chapter 3 includes information on system analysis, system design and algorithm details of the project.

Chapter 4 contains the implementation details of the project as well as testing of the project.

Chapter 5 includes the summary of the project along with the improvements and recommendations for future projects.

CHAPTER 2: BACKGROUND STUDY AND LITERATURE REVIEW

2.1 Background Study

The task assigning system is designed to assign tasks to employees based on their skills and availability. This system aims to improve work efficiency and ensure that tasks are completed by the most suitable employee. The Hopcroft-Karp algorithm is a graph algorithm used to find the maximum matching in a bipartite graph. In context of task assigning, a bipartite graph can be created where one set of vertices represent the tasks and the other set represent the employees. The edges between the two sets of vertices represent the skills required for each task possessed by each employee.

The system first takes in input data such as a list of available tasks and employee information including their skills and qualifications. The relation between the input data is established by creating a bipartite graph where each task is connected to the employees who possess the required skills to complete the task. The system constraints are also taken into account such as the availability of employees at certain times and their workload. The Hopcroft-Karp algorithm is applied to find the maximum matching in the bipartite graph. That means that each task is assigned to the most suitable employee based on their skills and qualifications. The algorithm ensures that each employee is assigned to only one task and that each task is assigned to only one employee.

The task assigning system using the Hopcroft-Karp algorithm aims to optimize task assignment by ensuring that each task is assigned to the most suitable employee. This can lead to increased work efficiency and productivity, as well as job satisfaction for employees who are working on tasks that are well-suited to their skills and qualification.

2.2 Literature Review

Task assigning systems have been widely studied in project management, with various methods proposed to improve the efficiency and accuracy of task allocation. One approach involves using a genetic algorithm to optimize task assignment based on employee skillset, workload, and availability. Results have shown that this approach outperforms traditional methods [1]. Another study proposed a task assigning system that uses a fuzzy logic algorithm to assign tasks based on employee skills, availability, and preferences. The

system was tested in a healthcare organization and showed significant improvements in task completion time and employee satisfaction [2].

In the context of intelligent manufacturing, a task assigning system was proposed that uses a reinforcement learning algorithm to optimize task allocation based on factors such as task complexity, employee skills, and workload. The authors demonstrated that this system outperformed traditional methods [3]. Additionally, a multi-criteria decision-making approach for task assignment in a collaborative environment was proposed. The system considers factors such as task priority, employee skills, availability, and workload to assign tasks, resulting in increased productivity and reduced employee stress [4].

One study proposed a task assigning system that uses the Hopcroft-Karp algorithm to optimize task allocation in a distributed system. The system considers various factors, such as task complexity, resource availability, and resource constraints, to assign tasks optimally. The authors demonstrated that the proposed system outperformed traditional methods in terms of efficiency and scalability, making it a promising solution for large-scale distributed systems [5]. Another study proposed a task assigning system that uses a hybrid algorithm combining the Hopcroft-Karp algorithm with a heuristic algorithm to optimize task allocation in a cloud computing environment. The system takes into account various factors, including task deadlines, resource availability, and task dependencies, to assign tasks optimally. The authors demonstrated that the proposed system outperformed traditional methods in terms of efficiency and scalability, making it a promising solution for cloud-based task assigning systems [6].

These studies highlight the potential of using advanced algorithms, such as genetic algorithms, fuzzy logic, Hopcroft-Karp algorithm and reinforcement learning, to improve task assigning systems' performance and efficiency. Additionally, considering multiple criteria in task allocation can result in fair and effective task assignment.

CHAPTER 3: SYSTEM ANALYSIS AND DESIGN

3.1 System Analysis

3.1.1 Requirement Analysis

i) Functional Requirements

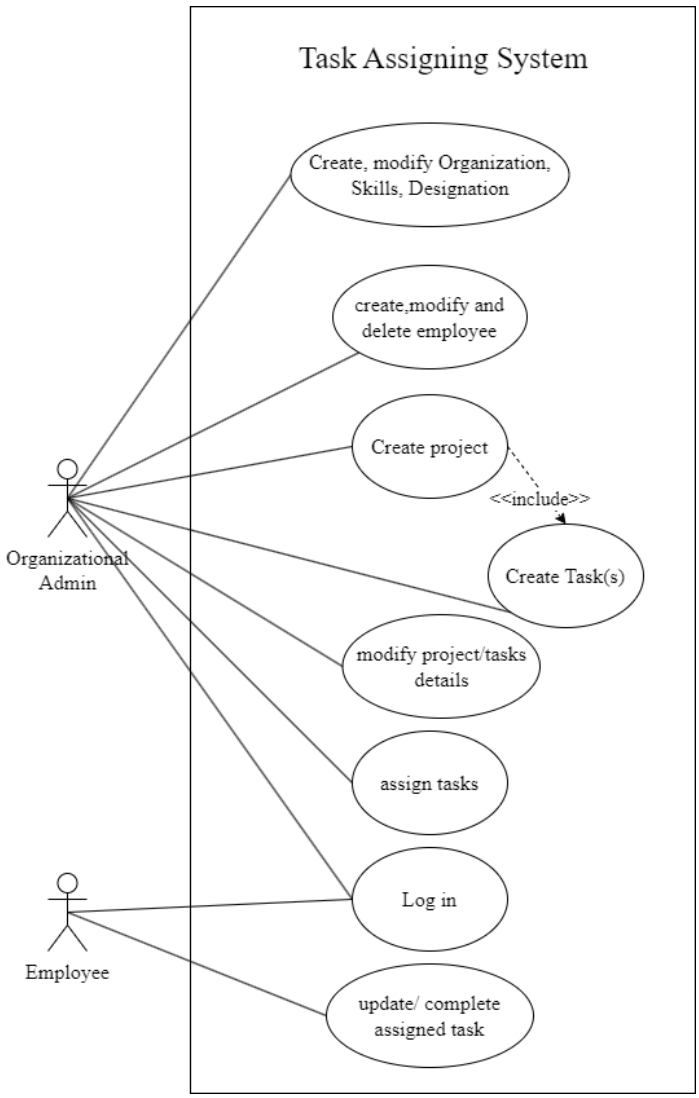


Figure 3.1: Use case Diagram of Task Assigning System

Figure 3.1 shows the use case diagram of task assigning system. It is clearly exhibited that the system consists of two actors-organizational admin, and employee. The organizational admin is also known as project manager. Admin can create, modify, and delete organization, skills, designation. Also can create, modify, and delete (deactivate) employee. Creation and modification of projects and tasks, and final assigning the task to

employee is also done by the organizational admin. Employee can login and update/complete the assigned task.

ii) Non-functional Requirements

Availability: The system is developed as a web application and can be used online.

Security: The system is secure and data isn't shared with third party. Likewise, only authorized entities have access to the data.

Performance: The system is designed for smooth performance with optimization.

Reliability: The system is reliable and free from errors and bugs.

Usability: The system is easy to use, and require minimal training for users.

Scalability: The system is scalable and able to accommodate future growth and expansion.

Compatibility: The system is compatible with different devices, operating systems, and web browsers.

3.1.2 Feasibility Analysis

i. Technical

The Task Assigning System is a technically feasible project as it uses open-source, free and readily available components and programming languages. It is built using HTML, CSS, JavaScript, C#, ASP.NET , Razor and SQL Server database.

ii. Operational

It is concerned with the operating capabilities of the system. For efficient operation, this system is useful for project-oriented organization. Managing the projects has been easier. Assigning the tasks to the employees automatically has also been very helpful as well.

iii. Economical

The purpose of economic feasibility is to determine the positive economic benefits. In context to the project work, the system developed is a web-based application; which requires all the hardware and software support as required by other applications. There is no cost since free resources has been used which are available in the internet and no cost for the manpower because the project is completed manually without any constable training.

iv. Schedule

This is one of the most important feasibility analyses as it helps an organization to estimate how much time the organization will take to complete the project and how much of it is on track to a given schedule. The scheduling of the project is performed as below:

GANTT chart:

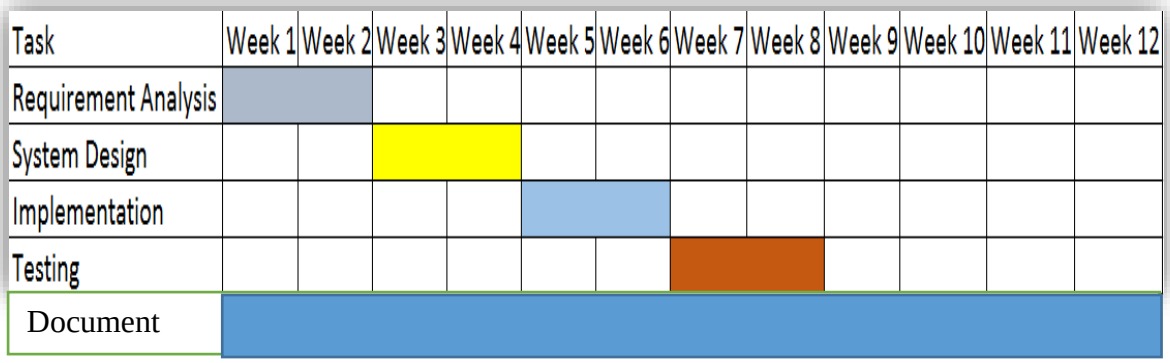


Figure 3.2: Gantt chart

3.1.3 Object Modeling using Class and Object Diagram

- **Class Diagram:**

  In the class diagram there are necessary attributes along with methods. It is used for general conceptual modeling of the structure of the system, and for detailed modeling, translating the models into programming code.

  Following are the various class used for the system:

  - Organization Class
  - Skill Class
  - Designation Class
  - Employee Class
  - Project Class
  - Task Class

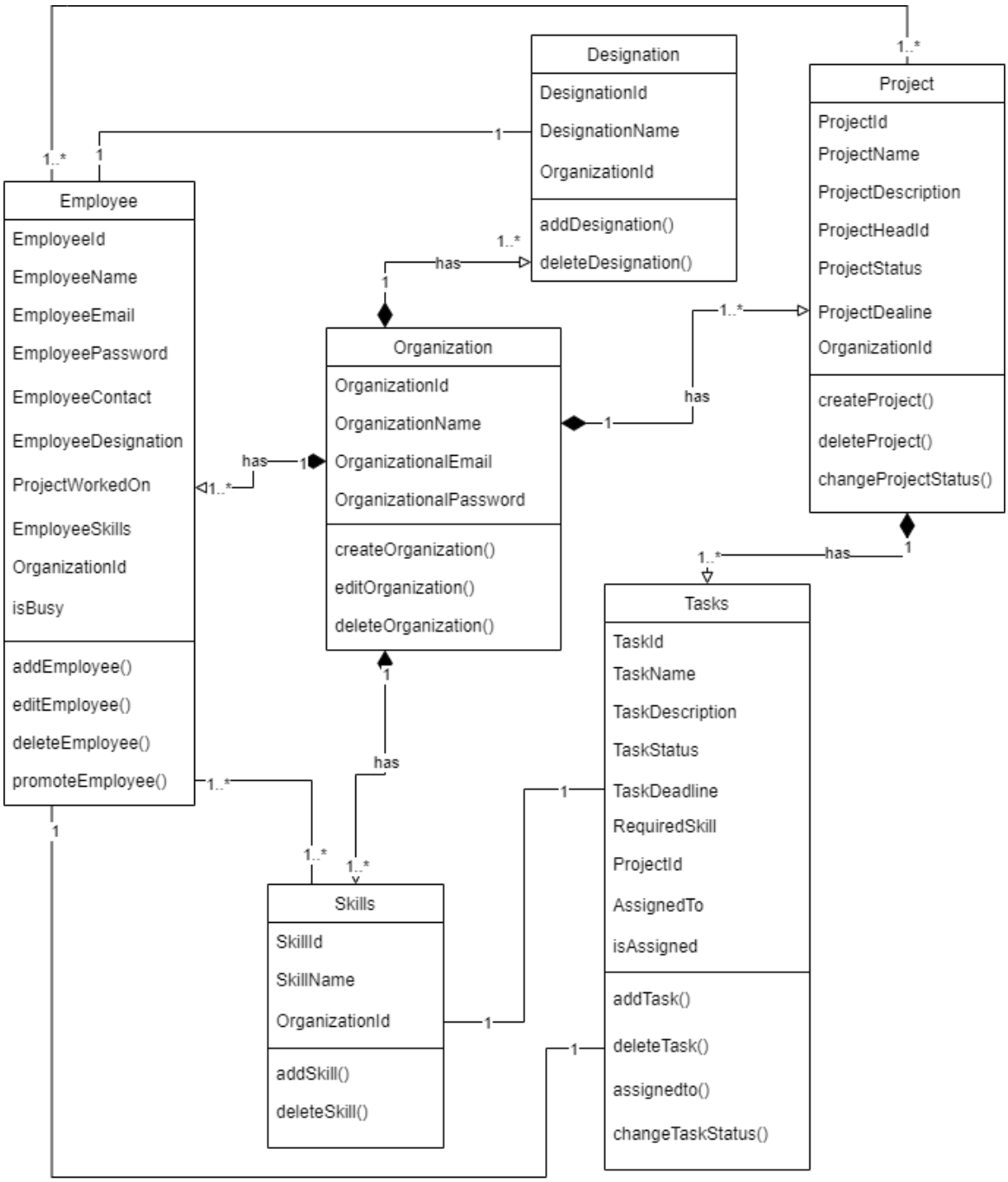


Figure 3.3: Class Diagram of Task Assigning System

Figure 3.3 shows Class Diagram of Task Assigning Booking System. It shows the working of class in the system.

- **Object Diagram:**

  Object diagram is used to show the instance of a class diagram. It shows a snapshot of the detailed state of a system at a point of time.

  Here, there is data set given for Organization, Employee, Project, Skill, Designation and Task classes.

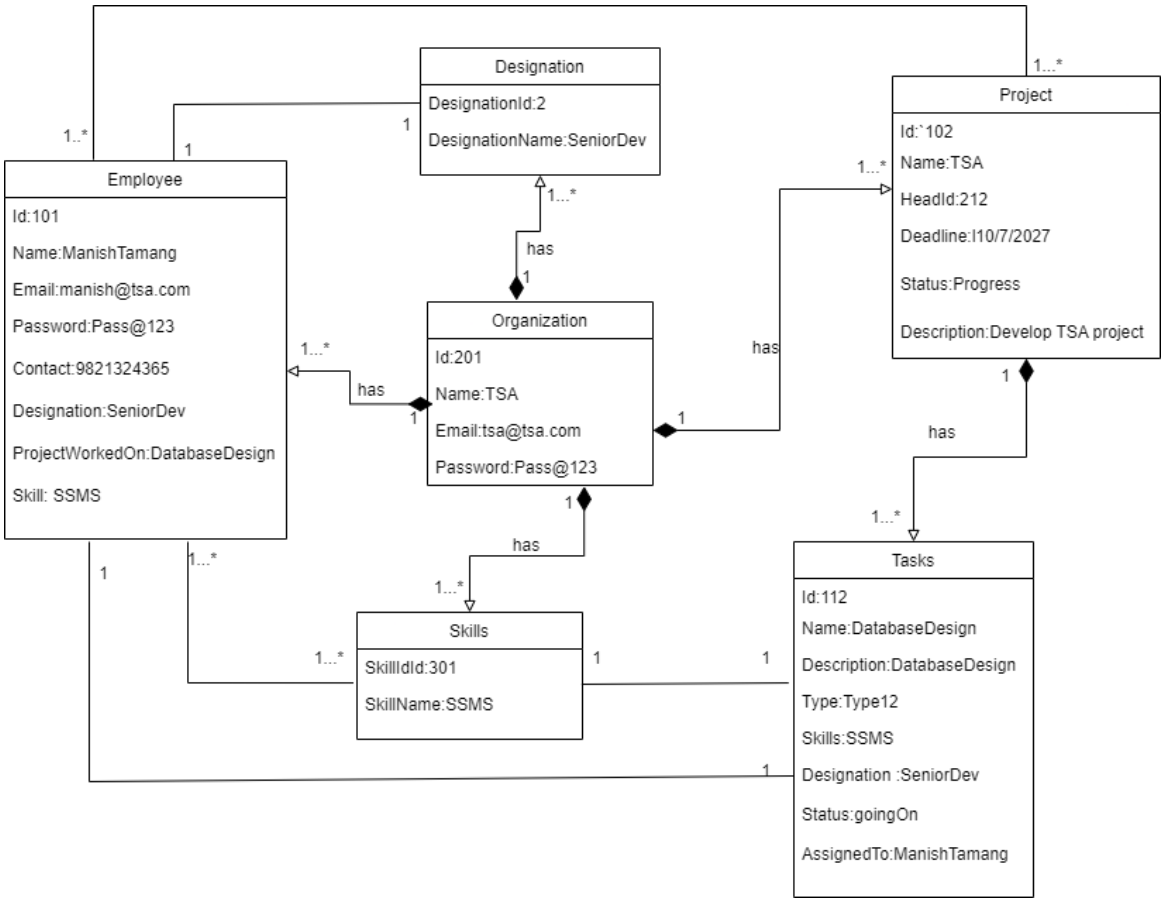


Figure 3.4: Object Diagram of Task Assigning System

3.1.4 Dynamic Modeling using State and Sequence Diagram

- **State Diagram:**

  State diagram shows a behavioral model consisting of states, state transitions and actions. Here, the figure depicts the finite number of every possible step of the system prescribed.

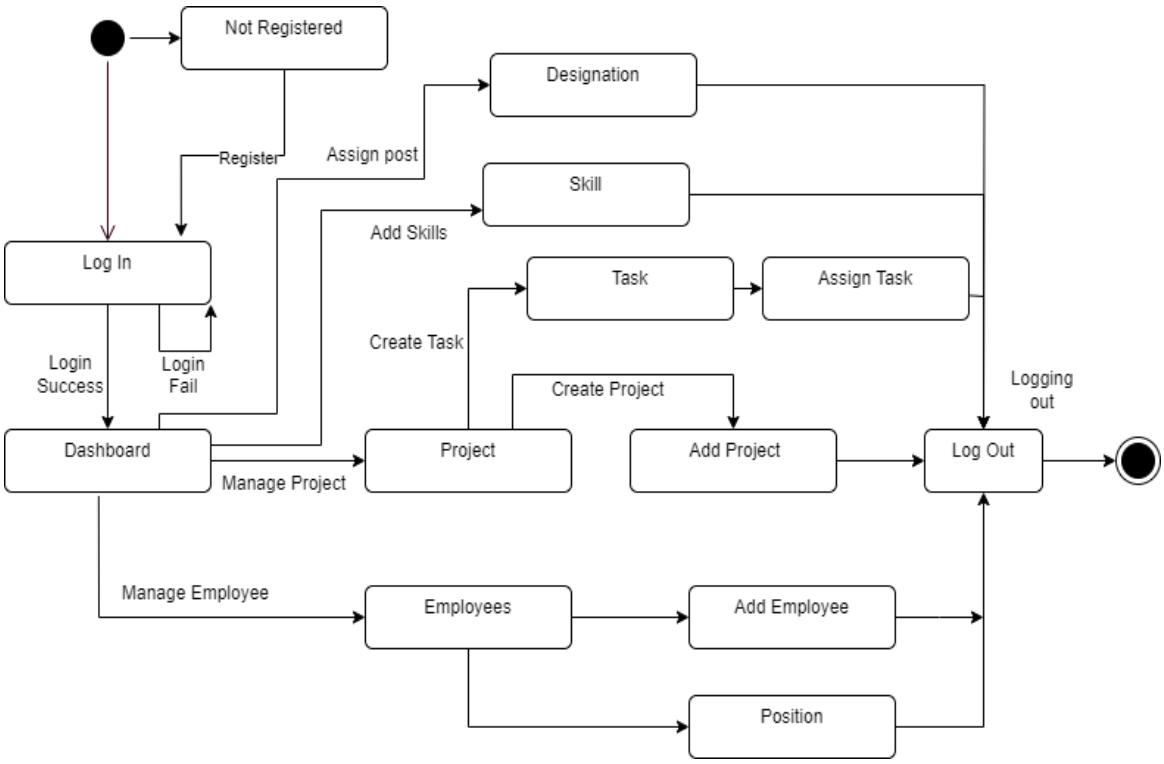


Figure 3.5: State Diagram of Task Assigning System

Figure 3.5 shows the state diagram of Task Assigning System. It shows the behavior of the system. It shows from creating the project to assigning the project to the employee.

- **Sequence Diagram:**

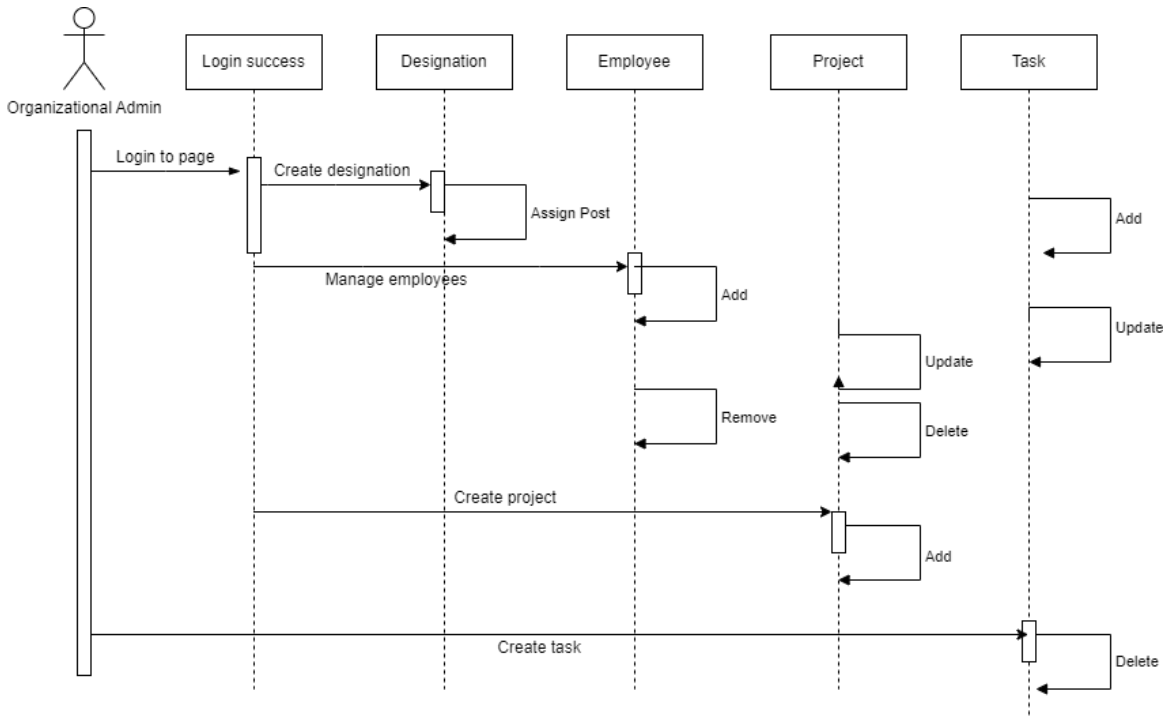


Figure 3.6: Sequence Diagram of Task Assigning System

Figure 3.6 shows the sequence diagram of Task Assigning System. It shows the sequence of operations performed by the admin (project manager) after login has been successful. It shows how the project manager creates project and assign to the employee.

3.1.5 Process Modeling using Activity Diagram

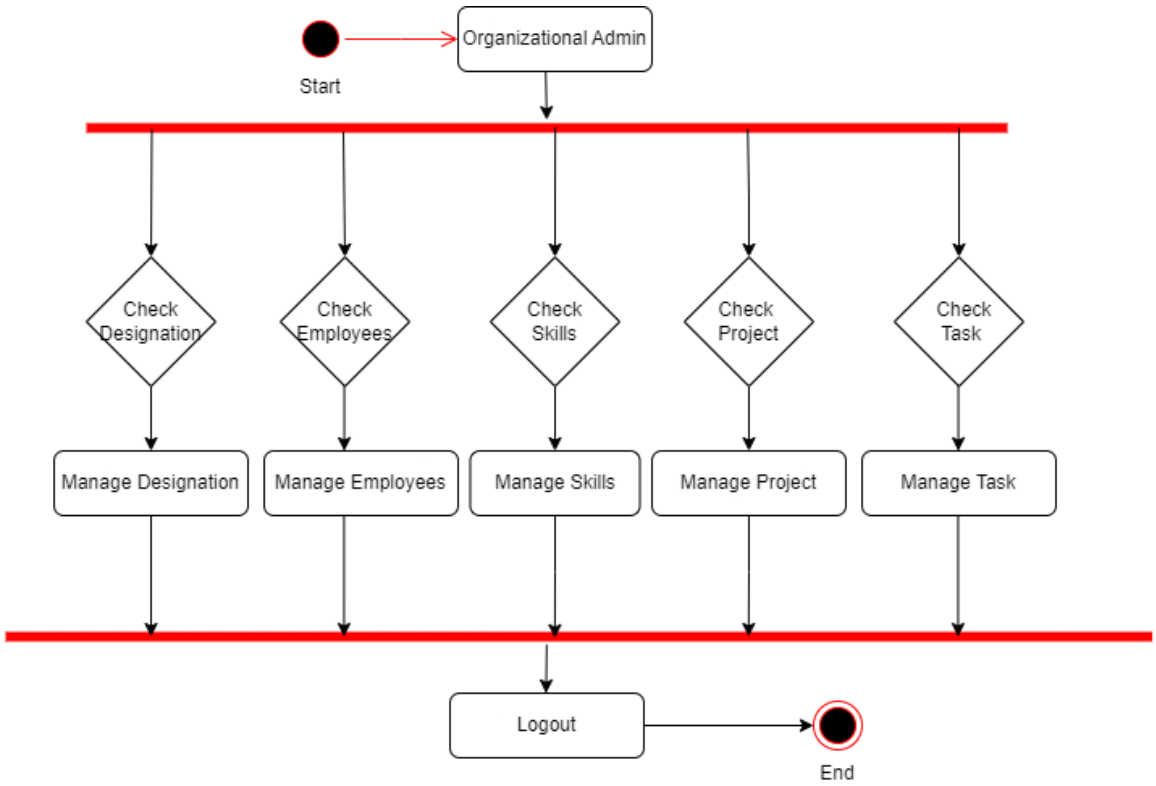


Figure 3.7: Activity Diagram of Task Assigning System

Figure 3.7 shows the activity diagram of Task Assigning System. It is a high-level activity diagram that illustrates the various processes the system goes through to generate algorithm. The diagram shows the flow between the activity of Organization, Employee, Project, Skill, Designation and Task.

3.2 System Design

System design is the phase that bridges the gap between problem domain and the existing system in manageable way. This phase focuses on the solution domain, i.e. “how to implement?” It is the phase where the SRS document is converted into a formatted that can be implemented and decides how the system will operate.

3.2.1 Refinement of Class and Object Diagram

Refinement of class and object diagrams involves adding more details and clarifying relationships between classes and objects. It is a process of elaborating on the initial

representations to provide a more comprehensive view of the system. Let's refine the previously provided class and object diagrams for task assigning system.

Figure 3.8: Refinement of Class and Object Diagram of Task Assigning System

3.2.2 Component Diagram

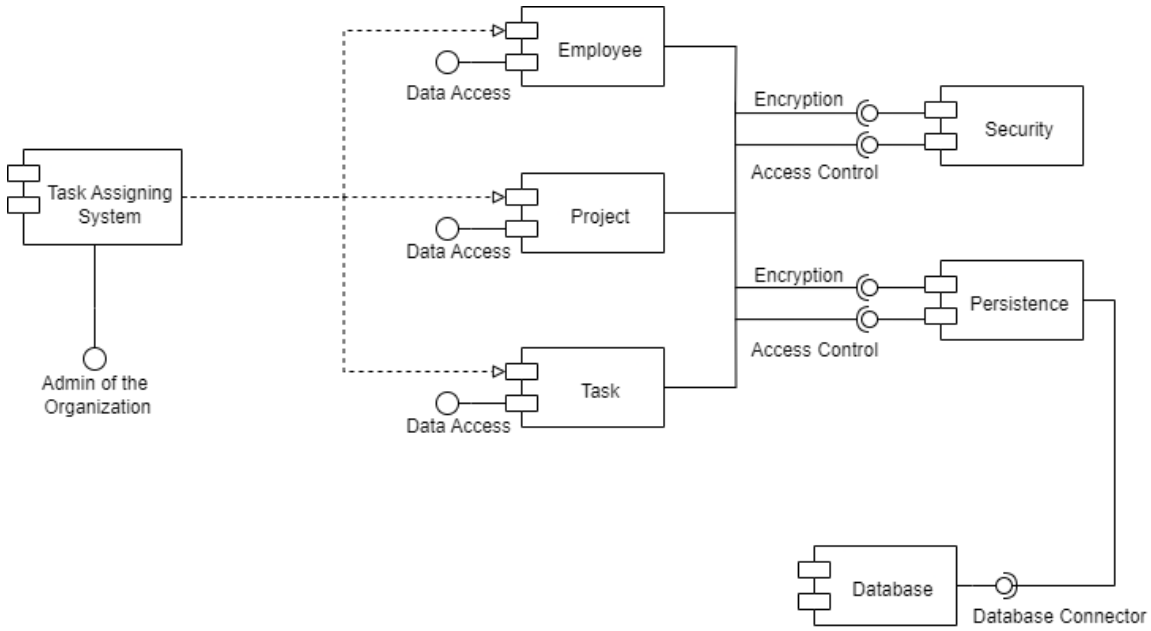


Figure 3.9: Component Diagram of Task Assigning System

Figure 3.9 shows the component diagram of Task Assigning System. This depicts the components, provided and required interfaces, relation between the components.

3.2.3 Deployment Diagram

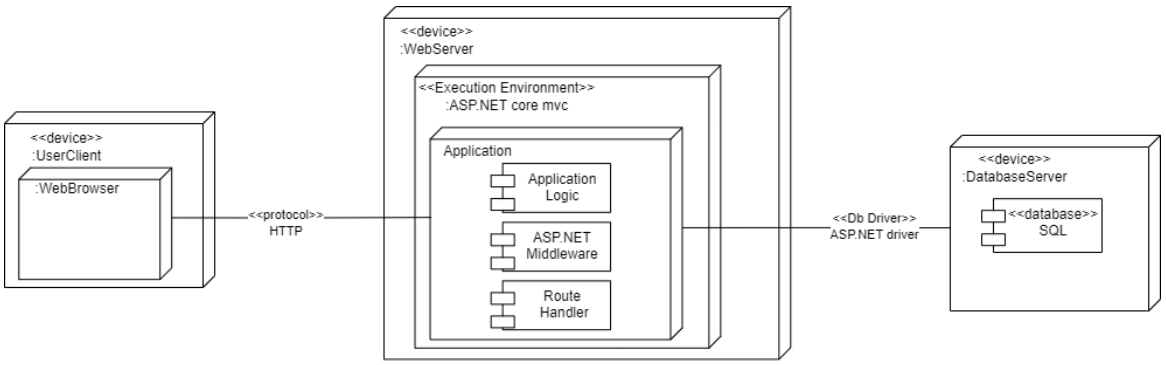


Figure 3.10: Deployment Diagram of Task Assigning System

Figure 3.7 shows the deployment diagram of Task Assigning System. It shows that the system is built in ASP.NET core MVC.

3.2 Algorithm Details

3.2.1 Hopcroft-Karp Algorithm

The Hopcroft-Karp algorithm is an algorithm to generate a maximum matching for a given bipartite graph. This was developed by John Hopcroft and Richard Karp.

The algorithm works in phases, where each phase is looking for augmenting paths of increasing lengths. An augmenting path is any path starting and ending on unmatched vertices, where edges within the path alternate between unmatched and matched. We will always have an odd number of edges in these paths, but they could be of size 1 or up.

Each path we find is then used to increase the matching that we have so far. We do this by adding all previously unmatched edges into our matching, whilst removing all previously matched edges from the matching. This will have the effect of increasing the size of the matching by 1 for every path found.

The overall algorithm starts with some general set up and then performs a number of searches grouped together in phases. During each phase, we perform a breadth-first search of the entire graph to discover any augmenting paths that exist in the graph. We then perform a number of depth-first searches, starting from each vertex in that is currently unmatched to build the appropriate augmenting paths and update the matching.

The steps that are used to implement the Hopcroft-Karp algorithm for the task assigning system are:

1. Build the bipartite graph as described above.
2. Initialize an empty matching between the employees and tasks.
3. While there exists an augmenting path in the graph, do the following:
    a. Initialize a set of visited vertices and a queue of the vertices to visit.
    b. For each employee vertex that is not already matched with a task, add it to the queue to be visited.
    c. For each visited employee vertex, iterate through its adjacent task vertices. If the task vertex is not already matched with an employee, add it to the matching and return.
    d. If the task vertex is already matched with an employee, mark the employee vertex as visited and add the task vertex's matching the employee to the queue to be visited.

e. Repeat steps b-d until there are no more vertices to visit or an augmenting path is found.

4. Return the matching between employees and tasks.

Sample code of the Hopcroft-Karp Algorithm

```
public Dictionary<Vertex, Vertex> HopcroftKarp(){
    var matching = new Dictionary<Vertex, Vertex>();
    while (BFS(matching)){
      var visited = new HashSet<Vertex>();
      foreach (var employeeNode in matching.Keys){
        if (DFS(employeeNode, visited, matching)){
          Console.WriteLine("Augmenting path found for Employee");
        }}}
    return matching;}
```

CHAPTER 4: IMPLEMENTATION AND TESTING

4.1 Implementation

4.1.1 Tools Used

Analysis and Design Tools

- Draw.io

Implementation Tools

- Visual Studio 2022

Frontend Tools

- HTML
- CSS
- JavaScript

Backend Tools

- C#
- ASP.NET Core mvc 6
- SQL Server Database
- Razor
- SSMS

4.1.2 Implementation Details of Modules

Employee Module:

The Employee module has been implemented with an "Employee" model that has properties for name, email, skills, and availability. The "EmployeesController" has been created with actions for creating, updating, listing, and deactivating employees. Views have been implemented to render the UI for these actions.

Project Module:

The Project module has been implemented with a "Project" model that has properties for name, description, and deadline. The "ProjectsController" has been created with actions for creating, updating, listing, and deleting projects. Views have been implemented to render the UI for these actions.

Task Module:

The Task module has been implemented with a "Tasks" model that has properties for name, description, and deadline. The "TasksController" has been created with actions for creating, updating, listing, and assigning employees to tasks. Views have been implemented to render the UI for these actions.

Hopcroft-Karp Algorithm Module:

The Hopcroft-Karp algorithm module has been implemented as a separate class or controller. The Task and Employee models have been used to create a bipartite graph, and the Hopcroft-Karp algorithm has been implemented to find the maximum matching between them. The dictionary data structure has been used to store the task-employee assignments, and it is returned as the output.

4.2 Testing

Testing is one of the most important processes that should be performed during the development process. Testing is a process that spans throughout the Software development life cycle (SDLC). Testing can be considered a set of activities that expose system and vulnerabilities and fix them.

4.2.1 Test Cases for Unit Testing

Testing is a process, to evaluate the functionality of a software application with an intent to find whether the developed software met the specified requirements or not and to identify the defects to ensure that the product is defect free in order to produce the quality product.

Mainly there are two types of testing. They are: manual testing and automation testing. The testing process runs parallel to software development. Testing separately is done just to make sure that there are no hidden bugs or issues left in the software. Hence, the testing phase is done to verify and validate this system. This activity includes programming, testing and integration of modules into a progressively more complete system. Implementation is the process of collect all the required parts and assembles them into a major product.

Table 4.1: Unit Testing for Employee Module

S.N.	Test Cases	Input Data	Expected Outcome	Result
1.	Check with valid data	EmployeeName: Manish Tamang EmployeeContact: 9821324365 EmployeeEmail: manish@tsa.com EmployeePassword: Pass@123 DesignationId:1	Employee object should be created successfully with all the input values	Pass
2.	Check with empty EmployeeName field	EmployeeName: EmployeeContact: 9821324365 EmployeeEmail: manish@tsa.com EmployeePassword: Pass@123 DesignationId:1	Validation error should occur for EmployeeName field with the message "The EmployeeName field is required."	The EmployeeName field is required
3.	Check with empty skill field	EmployeeName: Manish Tamang EmployeeContact: 9821324365 EmployeeEmail: manish@tsa.com EmployeePassword: Pass@123 DesignationId:1	Employee object should be created successfully with all the input values and the EmployeeSkills navigation property should be null	Pass
4.	Giving 11 numbers in contact	EmployeeName: Manish Tamang EmployeeContact: 98213243651 EmployeeEmail: manish@tsa.com EmployeePassword: Pass@123 DesignationId:1	Validation error should occur for EmployeeContact field with the message "Contact should be of 10 numbers."	Contact should be of 10 numbers
5.	Give already registered name and contact	EmployeeName: Manish Tamang EmployeeContact: 98213243651 EmployeeEmail: xyz@tsa.com EmployeePassword: Pass@12 DesignationId:2	Validation error should occur with the message "The EmployeeName and EmployeeContact combination already exist in the database."	The EmployeeName and EmployeeContact combination already exist in the database

4.2.2 Test Cases for System Testing

System testing is the overall testing of the system after integrating all the functions of the project. When all the functions of the Task Assigning System are integrated then the system testing is done. Some of the tests performed are mentioned below:

Table 4.2: System testing performed in the application

S.N.	Type of Test	Description
1.	Usability Testing	This test was mainly focused on the user's ease to use of the application, flexibility in handling controls, and ability of the system to meet its objectives.
2.	Load Testing	In this phase, tests were performed to know that a software solution performs well under real-life loads.
3.	Functional Testing	Also known as functional completeness testing, this test involved trying to think of any possible missing functions. Additional functionalities that a product could have to improve during functional testing and later implemented that were listed.
4.	Migration Testing	This test was done to ensure that the software can be moved from one system infrastructure to another system infrastructure without any issues. This application was migrated from our local development server to the production server as well.

4.3 Result Analysis

Hopcroft-Karp Algorithm

The Hopcroft-Karp algorithm is a graph algorithm used to solve the maximum cardinality matching problem in bipartite graphs. In the context of a project task assigning system, the algorithm could be used to optimize the assignment of tasks to resources (e.g., employees) based on certain criteria or constraints. To focus on time the time analysis of the Hopcroft-Karp algorithm in the system, a table is created below:

Table 4.3: Result Analysis of Hopcroft-Karp Algorithm

Test Case	Number of Tasks	Number of Employees	Matching Size	Execution Time (seconds)
1	10	10	5	2.1
2	8	8	4	1.5
3	6	6	3	1.2
Average				1.6

Execution Time (seconds):

Test Case 1: 2.1

Test case 2: 1.5

Test Case 3: 1.2

To calculate the average time, sum up these values and divide by number of test cases (in this case, 3):

Average Time = (2.1 + 1.5 + 1.2) / 3 = 4.8 / 3 = 1.6 seconds

So, the average execution time for the provided test cases is 1.6 seconds.

CHAPTER 5: CONCLUSION AND FUTURE RECOMMENDATIONS

5.1 Conclusion

Task Assigning System has been developed that help organizations efficiently manage their projects, tasks and employees. The system allows for the creation of tasks with details like name, description, and deadline, as well as the assignment of employees based on their skills and availability.

The Hopcroft-Karp algorithm is used to find the maximum matching between tasks and employees, ensuring that the right person is assigned to each task. The system also includes modules for managing employee data and project data, providing a comprehensive solution for task assignment and management. The use of ASP.NET MVC and Entity Framework allows for efficient data management and easy UI rendering.

5.2 Future Recommendation

For future enhancement of the Task Assigning System, adding features such as real-time progress tracking, automatic task prioritization based on deadlines and integration with project management tools such as ClickUp, Jira and Trello can be considered. Machine Learning Algorithm for better allocation of task to the employees based on their experience, skillsets and their ability to handle the task can also be used. Finally, gathering user feedback and conducting surveys can help to identify areas of improvement and make the system more efficient and user-friendly.

References

[1] H. Zhang et al., "Design and implementation of a task assigning system based on genetic algorithm," IEEE Access, vol. 5, pp. 6471-6480, 2017.

[2] Y. Wang, Y. Li, and Y. Zhang, "A fuzzy task assignment model based on employee skills, availability and preferences," Journal of Intelligent & Fuzzy Systems, vol. 35, no. 1, pp. 1-9, 2018.

[3] X. Zhang et al., "A novel task assignment method based on reinforcement learning for intelligent manufacturing," Journal of Intelligent Manufacturing, vol. 30, no. 2, pp. 803-812, 2019.

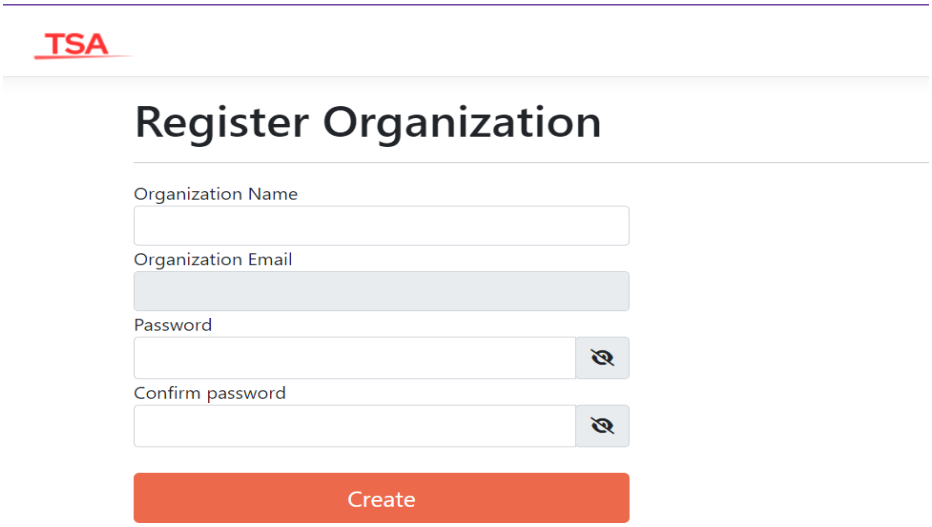
[4] P. T. de Carvalho et al., "A multi-criteria decision-making approach for task assignment in a collaborative environment," International Journal of Production Research, vol. 56, no. 12, pp. 4179-4195, 2018.

[5] K. Mishra and P. Sahoo, "Hopcroft-Karp Algorithm Based Task Assignment in Distributed Systems," 2018 IEEE International Conference on Advances in Computing, Communications and Informatics (ICACCI), Bangalore, 2018, pp. 1377-1381.

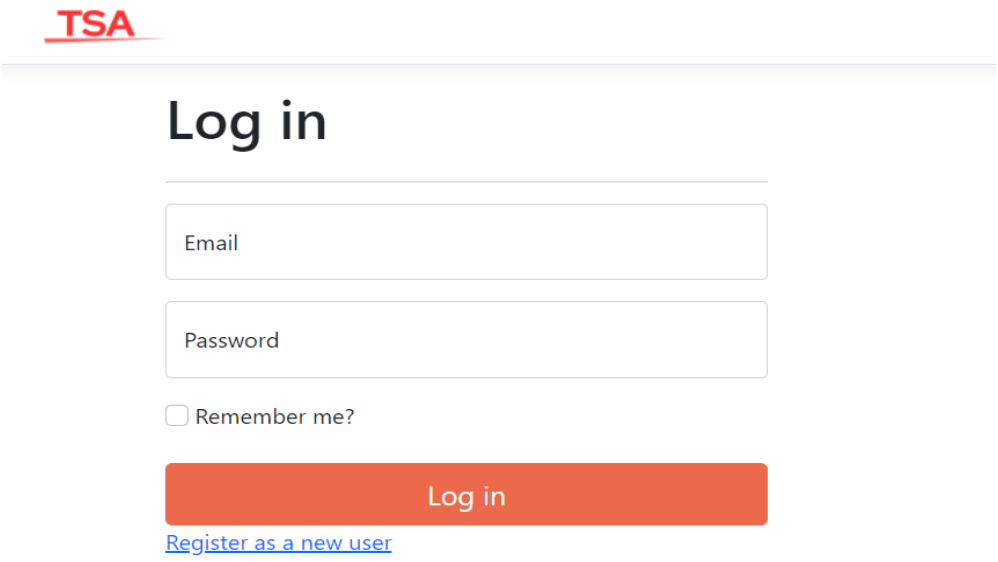
[6] A. Shrestha, J. Gao and D. Datta, "A Hybrid Hopcroft-Karp Algorithm for Efficient Task Allocation in Cloud Computing," 2019 IEEE International Conference on Services Computing (SCC), San Diego, CA, USA, 2019, pp. 1-8.

Appendices

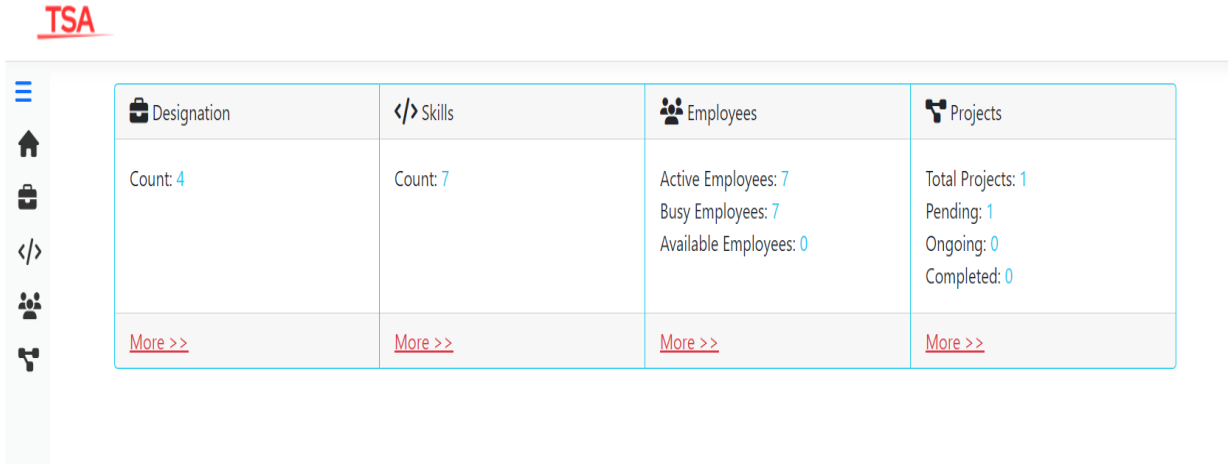
Screenshots



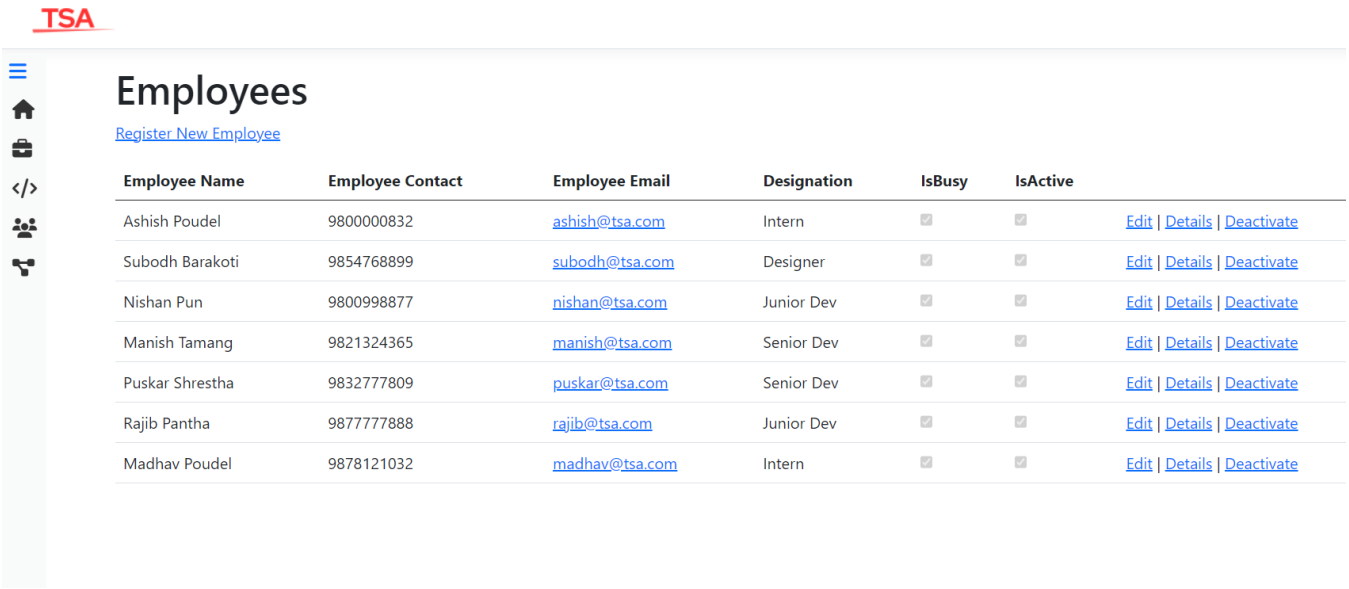
Screenshot of Registration Page



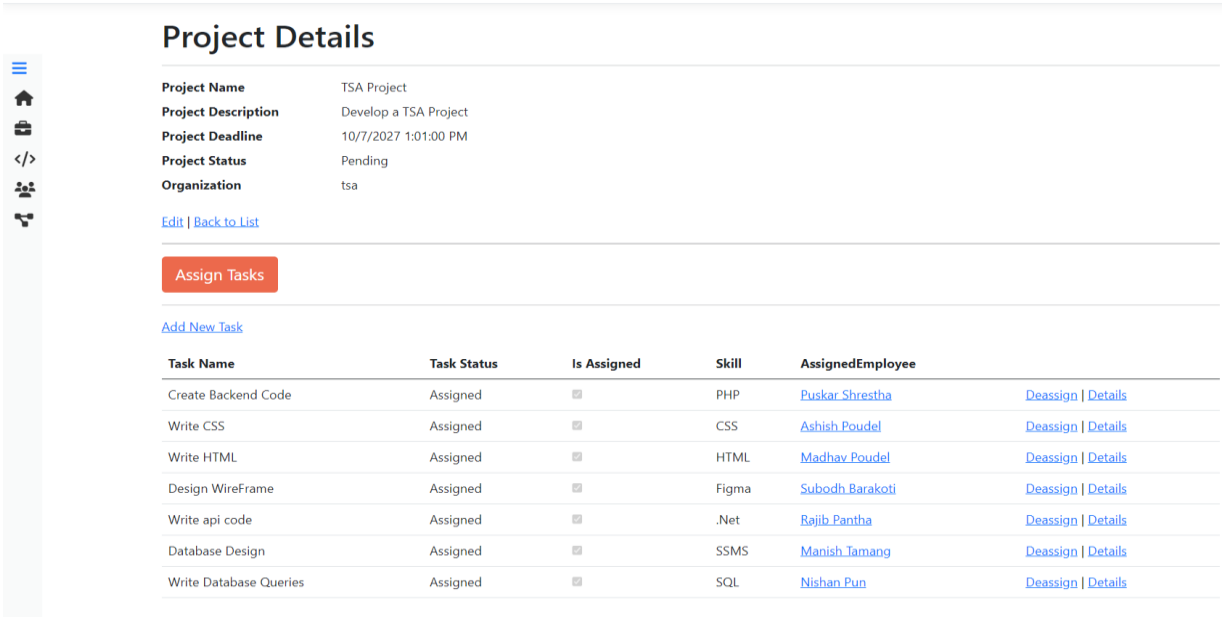
Screenshot of Login Page



Screenshot of Home Page of Project Manager



Screenshot of Employee Detail Page



Screenshot of Project Detail Page

Screenshot of assigned task to employee