**ToDo List / Task Management**

**A PROJECT REPORT**

***Submitted by***

**Baraiya Nikhilkumar Rajnikantbhai**

**(23030501003)**

***In partial fulfilment for the award of the degree of***

**Master of Computer Application (MCA)**

***at***

**DARSHAN INSTITUTE OF COMPUTER APPLICATION (DICA),**

**RAJKOT**



**Darshan University, Rajkot**

**March/April – 2025**

****

**Darshan Institute of Computer Application, Rajkot**

Rajkot - Morbi highway, Rajkot-363650, Gujarat

**CERTIFICATE**

This is to certify that the project report submitted along with the project entitled “ToDo List / Task Management**”** has been carried out by “**Baraiya Nikhilkumar Rajnikantbhai**” under my guidance in partial fulfilment for the degree of Master of Computer Application in ***Department of Computer Science***, 4th Semester of Darshan University, Rajkot during the academic year 2024-25.

|  |  |  |
| --- | --- | --- |
| Internal Guide  Prof. Hemang Chath  CSE Department,  Darshan University |  | Dean-DICA  Dr. Gopi Sanghani  CSE Department, Darshan University |

****

**Darshan Institute of Computer Application, Rajkot**

Rajkot - Morbi highway, Rajkot-363650, Gujarat

**DECLARATION**

We hereby declare that the Project report submitted along with the Project entitled **ToDo List / Task Management** submitted in partial fulfilment for the degree of Master of Computer Application at **Darshan Institute of Computer Application,** **Darshan University, Rajkot**, is a bonafide record of original project work carried out by me at Darshan Institute of Computer Application under the supervision of **Prof. Hemang Chath** and that no part of this report has been directly copied from any students’ reports or taken from any other source, without providing due reference.

|  |  |
| --- | --- |
| Baraiya Nikhilkumar Rajnikantbhai  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |

**ACKNOWLEDGEMENT**

I wish to express my sincere gratitude to my project guide Prof. **Hemang Chath** and all the faculty members for helping me through my project by giving me the necessary suggestions and advices along with their valuable co- ordination in completing this work.

I also thank my parents, friends and all the members of the family for their precious support and encouragement which they had provided in completion of my work. In addition to that, I would also like to mention the college personals who gave me the permission to use and experience the valuable resources required for the project from the college premises.

Thus, in conclusion to the above said, I once again thank the faculties and members of **Darshan Institute of Computer Application** for their valuable support in completion of the project.

**With Sincere Regards**

Baraiya Nikhilkumar Rajnikantbhai

**ABSTRACT**

**ToDo List / Task Management**

The To-Do List system is designed as a modern, automated solution to streamline task management by replacing traditional paper-based methods with a digital platform. It enables users to create, view, update, and delete tasks with ease while offering advanced features like task prioritization, deadline settings, and collaboration capabilities. By allowing tasks to be categorized and marked as complete, the system addresses common productivity challenges such as forgetfulness and inefficient time management, ensuring that daily activities are effectively tracked and organized.

Emphasizing simplicity and accessibility, the system features a user-friendly interface that caters to individuals of all technical skill levels. Its focus on automation minimizes manual effort and delivers a seamless task management experience, making it an ideal tool for both personal organization and team collaboration. Ultimately, the To-Do List system not only enhances productivity but also transforms the way users manage their daily tasks by providing a structured, intuitive, and efficient digital environment.

**LIST OF FIGURES**

|  |  |  |
| --- | --- | --- |
| **Figure No** | **Figure Description** | **Page No** |
| 3.1.1.1 | Context diagram of task management system | 5 |
| 3.1.2.1 | Use case diagram of task management system | 6 |
| 3.1.4.1 | Activity diagram of task management system | 9 |
| 3.1.5.1 | Sequence diagram of task management system | 10 |
| 3.1.6.1 | State diagram of task management system | 10 |
| 3.1.7.1 | Class diagram of task management system | 11 |

**LIST OF TABLES**

|  |  |  |
| --- | --- | --- |
| **Table No** | **Table Description** | **Page No** |
| 3.2.1 | User table | 12 |
| 3.2.2 | Task table | 12 |
| 3.2.3 | Notification table | 12 |
|  |  |  |
|  |  |  |

**TABLE OF CONTENT**

|  |  |  |  |
| --- | --- | --- | --- |
| **Chapter No** | **Chapter Title** | | **Page No** |
|  | Acknowledgement | | i |
|  | Abstract | | ii |
|  | List of Figures | | iii |
|  | List of Tables | | iv |
|  | Table of Contents | | vi |
| **Chapter 1** | **INTRODUCTION** | | 1 |
|  | 1.1 | Motivation | 1 |
|  | 1.2 | Project Scope | 1 |
|  | 1.3 | Project Stakeholders | 1 |
|  | 1.4 | Project Scheduling | 1 |
| **Chapter 2** | **SYSTEM ANALYSIS** | | 2 |
|  | 2.1 | Introduction | 2 |
|  | 2.2 | Requirement Analysis | 2 |
|  |  | 2.2.1 Functional Requirement  2.2.2 Non-Functional Requirement  2.2.3 Software & Hardware Requirements | 2  2  3 |
|  | 2.3 | Design and Implementation Constraints | 3 |
|  | 2.4 | Assumptions and Dependencies | 3 |
| **Chapter 3** | **SYSTEM DESIGN** | | 4 |
|  | 3.1 | UML Diagram / Data-Flow Diagram | 4 |
|  |  | 3.1.1 DFD 0 Diagram | 4s |
|  |  | 3.1.2 Use case Diagram | 5 |
|  |  | 3.1.3 Use case Scenario | 6 |
|  |  | 3.1.4 Activity Diagram | 8 |
|  |  | 3.1.5 Sequence Diagram | 9 |
|  |  | 3.1.6 State Diagram | 9 |
|  |  | 3.1.7 Class Diagram | 10 |
|  | 3.2 | Database Design | 11 |
|  |  | 3.2.1 Database Schema  3.2.2 E-R Diagrams | 11  12 |
| **Chapter 4** | **IMPLEMENTATION** | | 13 |
|  | 4.1 | Module Design and Specification | 13 |
|  | 4.2 | Testing | 16 |
| **Chapter 5** | **CONCLUSION AND FUTURE ENHANCEMENT** | | 17 |
|  | 5.1 | Conclusion | 17 |
|  | 5.2 | Future Enhancement | 17 |
|  | **APPENDICES** | | 18 |
|  | **REFERENCES** | | 19 |

**CHAPTER 1**

**INTRODUCTION**

* 1. **Motivation**

Managing tasks efficiently is crucial in today’s busy world, where missed deadlines and disorganization can lead to stress and reduced productivity. A well-structured task management system helps users organize, prioritize, and track their tasks, ensuring better workflow and time management.

With the growing demand for digital solutions, a smart and intuitive to-do list application will empower users to stay organized, set reminders, and collaborate seamlessly. By offering a simple yet powerful tool, this project aims to enhance productivity and efficiency for individuals and teams alike.

* 1. **Project Scope**

The To-Do List or Task Management application will be a web-based designed to help users efficiently create, organize, and track their tasks. It will support features such as task creation, categorization, prioritization, due dates, reminders, and progress tracking. Users will be able to collaborate by sharing tasks with team members, assigning responsibilities, and monitoring real-time updates. The system will offer a user-friendly interface with cloud synchronization, ensuring accessibility across multiple devices

* 1. **Project Stakeholders**

The project stakeholders for this document are:

* Software Project Managers
* Software Engineers
* Software Developers
* Task/schedule Organizers
  1. **Project Scheduling**

|  |  |  |
| --- | --- | --- |
| Phase | Tasks | Duration |
| Week 1-2 | Requirement Analysis & Feasibility Study | 2 Weeks |
| Week 3 | System Design (ERD, DFD, UML Diagrams) | 1 Week |
| Week 4-6 | Database Design & Setup (MongoDB) | 3 Weeks |
| Week 7-10 | Frontend Development (HTML, CSS, JavaScript, React) | 4 Weeks |
| Week 11-14 | Backend Development (Nodejs/Express) | 4 Weeks |
| Week 15-16 | Integration & Testing | 2 Weeks |
| Week 17 | Deployment & Documentation | 1 Week |
| Week 18 | Final Review & Submission | 1 Week |

**CHAPTER 2**

**SYSTEM ANALYSIS**

* 1. **Introduction**

### Introduction

The To-Do List or Task Management application is a digital solution designed to help users efficiently organize, prioritize, and track their tasks. It will provide features such as task creation, categorization, due dates, reminders, and collaboration tools, ensuring seamless task management for individuals and teams. With a user-friendly interface and cloud synchronization, the application will enhance productivity by making task tracking accessible anytime, anywhere.

* 1. **Requirement Analysis**
     1. **Functional Requirement**

**The Individual**

* **Register** : User can register in to the system.
* **Login** : User can login in to the system.
* **Forget Password** : User can forget password.
* **Add Task**: New task must be entered in database.
* **Task Deadlines** : User can set task deadlines.
* **Task Priority** : User can set task priority.
* **Task Progress Tracking** : User can set task as completed or pending.
* **Update Task**: Any changes in task should be updated in case of update.
* **Delete Task** : Wrong entry must be removed from system.
* **Search Task** : User can search any Task.
* **Share Task** : User can share task to other.
  + 1. **Non-Functional Requirement**

**Performance Requirements**

* The system should handle at least 100 concurrent users without performance degradation.
* Task search functionality should return results within 2 seconds.
* Response time for issuing or returning a task should be less than 3 seconds.

**Scalability**

* The system should support the addition of new users, task , and transactions without requiring major changes.
* The database should be capable of storing at least 100 000 tasks.

**Security**

* User authentication should be implemented using role-based access control (RBAC).
* Passwords should be stored using encryption techniques (e.g., bcrypt, SHA-256).
* Only authorized users can add or remove task.
  + 1. **Software & Hardware Requirements**

|  |  |  |  |
| --- | --- | --- | --- |
| **User** | **Particulars** | **Client System** | **Server System** |
| Individual  Or  Organization  (Windows  Application ) | Operating System | Windows | Windows Server |
| Processor | Dual core (Minimum) | Pentium 4.0 GHz or higher |
| Hard disk | 123 GB (Minimum) | 1 GB |
| RAM | 512 MB (Minimum) | 80 GB |

* 1. **Design and Implementation Constraints**

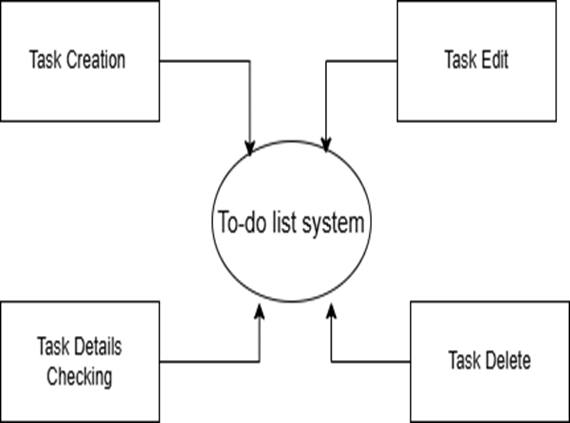
Individual or Organization must keep their password as confidential. Only Owner can manage resources like task. More over the user must have individual ID for getting account information.

* 1. **Assumptions and Dependencies**
* Internet must be needed for communication with server.
* Individual or organization must be use standalone application for manage resources.
* Each Member must have a unique ID for accessing Task resources.
* User need basic understanding of system.

**CHAPTER 3**

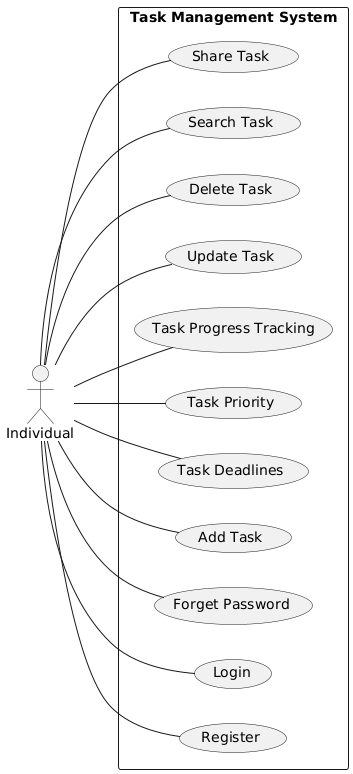
**SYSTEM DESIGN**

* 1. **UML Diagram**
     1. **Data-Flow Diagram**



*Figure 3.1.1.1 context diagram of task management system*

* + 1. **Use case Diagram**

****

*Figure 3.1.2.1 Use case diagram of task management*

* + 1. **Use-case Scenarios**

**Login**

|  |  |
| --- | --- |
| **Abbreviated Title** | Login |
| **Actors** | Individual |
| **Description** | To interact with the system, TMS will validate its registration with this system. It also defines the actions a user can perform in TMS |
| **Pre-Conditions** | User must have proper client installed on user terminal |
| **Task Sequence** | * System show Login Screen * User Fill in required information. Enter user name and password * System acknowledge entry |
| **Post Conditions** | System transfer control to user main screen to proceed further actions |
| **Exception** | If in step 3 no user found then system display Invalid user name password error message . |

**Search Task**

|  |  |
| --- | --- |
| **Abbreviated Title** | Search Task |
| **Actors** | Individual |
| **Description** | Search task makes it easy to search for task on TMS. With this search companion, user can specify several search criteria. For example, type task name, etc. |
| **Pre-Conditions** | User must be login |
| **Task Sequence** | * System will show searching screen * User enter required information. It can be task name, task description etc * By pressing search button system will list down all searching results |
| **Post Conditions** | * User can view his desire results * Request task |
| **Exception** | If in step 3 no task found then system display no task found message |

**Add Task**

|  |  |
| --- | --- |
| **Abbreviated Title** | Add task |
| **Actors** | Individual |
| **Description** | System will add new task to system for individual or collab task |
| **Pre-Conditions** | User must be login to the system |
| **Task Sequence** | System will display add task form  It will add task in to database  And show task in screen |
| **Post Conditions** | User must add all necessary fields |
| **Exception** | - |

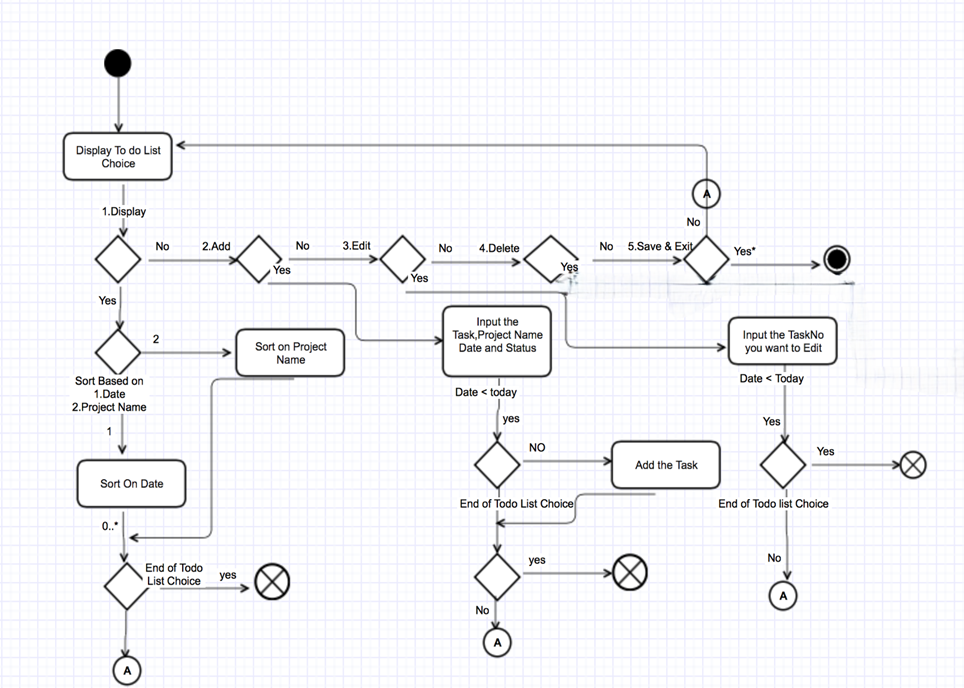
**Delete Task**

|  |  |
| --- | --- |
| **Abbreviated Title** | Delete task |
| **Actors** | Individual |
| **Description** | System will delete the unnecessary or not needed task |
| **Pre-Conditions** | User must be login with its account |
| **Task Sequence** | System will show delete button on down side of task  On press it will delete task |
| **Post Conditions** | System can have updated system inventory |
| **Exception** | - |

**Share task**

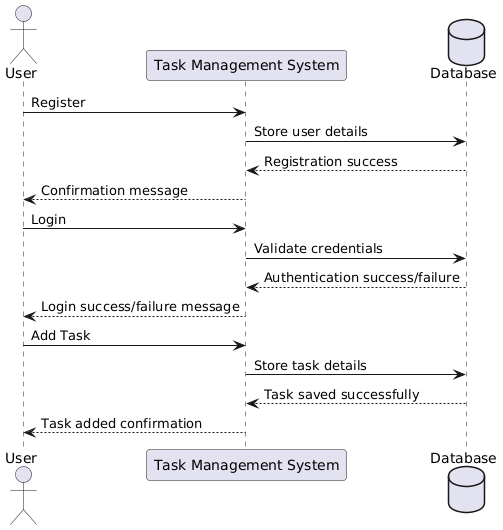
|  |  |
| --- | --- |
| **Abbreviated Title** | Share task |
| **Actors** | Individual |
| **Description** | User can share task through share button on email |
| **Pre-Conditions** | Member must be logged on with its account |
| **Task Sequence** | * System show share button below task * On press textbox pop up to enter email * On share system share task to that email |
| **Post Conditions** | Member will be restricted or granted |
| **Exception** | If email does not exist it will shoe error else show task shared successfully |

* + 1. **Activity Diagram**

****

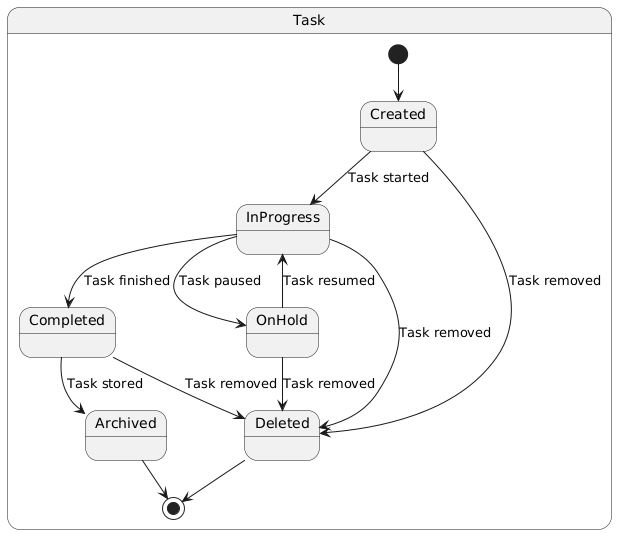
*Figure 3.1.4.1 Activity diagram for TMS*

* + 1. **Sequence Diagram**

****

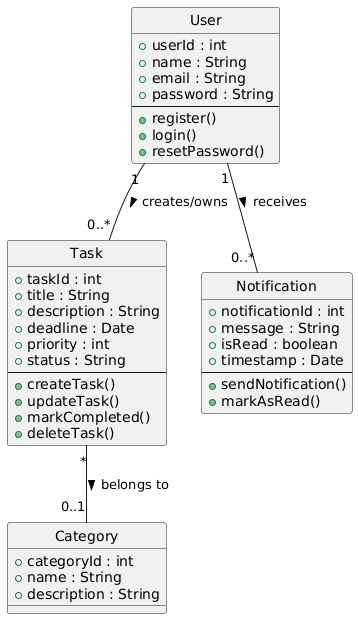
*Figure 3.1.5.1 Sequence diagram for task add*

* + 1. **State Diagram**

****

*Figure 3.1.6.1 State diagram of task*

* + 1. **Class Diagram**

****

*Figure 3.1.7.1 Class diagram of TMS*

* 1. **Database Design**
     1. **Database Schema**

**Table 1: User**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Column | Data Type | Null | Keys & Constrains | Default Value & Description |
| \_id | ObjectId | NN | PK (Auto Increment) |  |
| name | String | NN |  |  |
| email | String | NN |  |  |
| password | String | NN |  |  |
| createdAt | Date | AN |  |  |

*Table 3.9.1 Table: Book*

**Table 2: Task**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Column | Data Type | Null | Keys & Constrains | Default Value & Description |
| \_id | ObjectId | NN | PK (Auto Increment) |  |
| title | String | NN |  |  |
| description | String | NN |  |  |
| deadline | Date | AN |  |  |
| priority | Number | NN |  |  |
| status | int | NN |  |  |
| createdAt | Date | AN |  |  |

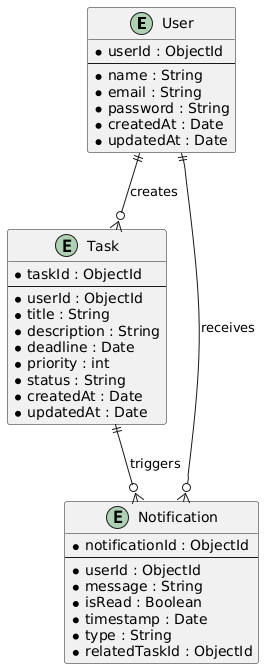
*Table 3.9.2 Table: Task*

**Table 3: Notification**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Column | Data Type | Null | Keys & Constrains | Default Value & Description |
| \_id | ObjectId | NN | PK (Auto Increment) |  |
| message | String | AN |  |  |
| isRead | Boolean | AN |  |  |
| timestamp | Date | AN |  |  |

*Table 3.9.3 Table: notification*

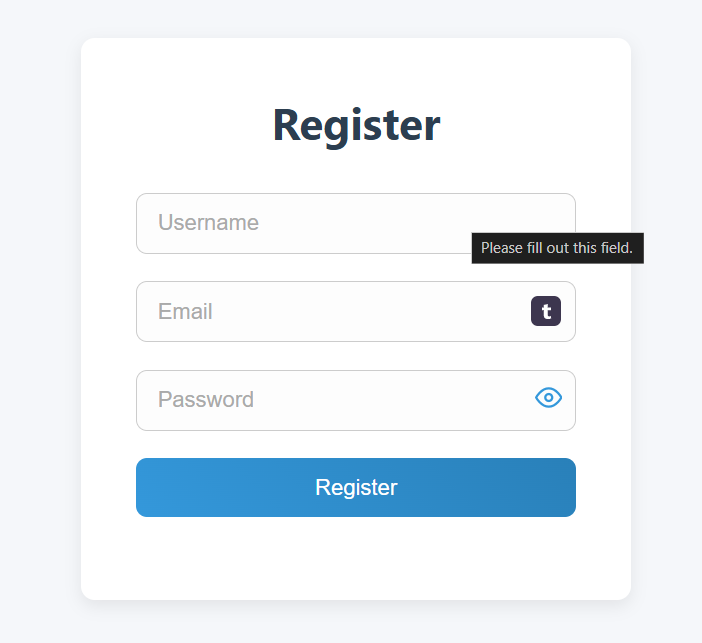
* + 1. **E-R Diagram**

****

**CHAPTER 4**

**IMPLEMENTATION**

* 1. **Module Design and Specification**
     1. **Screen-1: Registration Form**

****

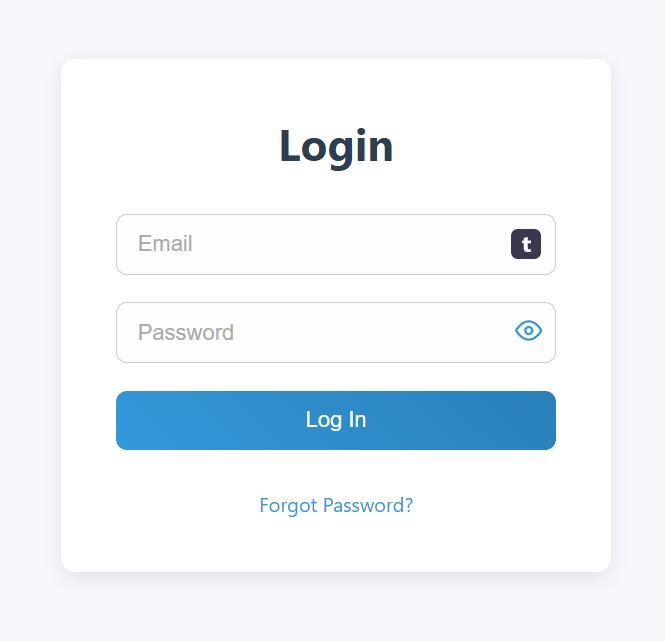
*Figure 4.1 1 Screen-1: Registration Form*

**Purpose**: This form will allow the target end-users to register in the system. To register, the following information will be encoded in the system.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Sr. | Screen Element | Input Type | O/M | Description |
| 1 | Username | Textbox | M | Username field should be editable and accept the Username. |
| 2 | Email | Textbox | M | Email field should be editable and accept the email with proper format. |
| 3 | Password | Password | M | Password field should be editable and accept the password and display as star or dot. |
| 4 | ShowPassword | Button | M | Show password of the password textbox |
| 5 | Register | Button | ------ | Register is a button for store the entered data into database. |

*Table 4.1 1 Screen element of Registration form*

**Screen-2: Login Form**

****

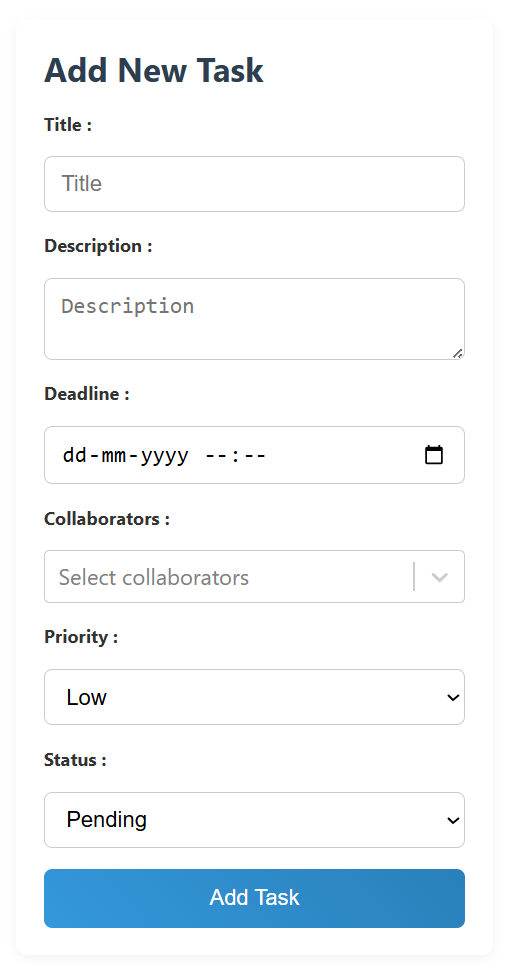
*Figure 4.1.2 Screen-2: Login Form*

**Purpose**: This form will be used by the system’s users to access records and features of the system. The users will input the correct combination of their username and password to be able to login to the system.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Sr. | Screen Element | Input Type | O/M | Description |
| 1 | Email | Textbox | M | Email field should be editable and accept the email. |
| 2 | Password | Password | M | Password field should be editable and accept the password and display as star or dot. |
| 3 | ShowPassword | Button | ----- | Show password of the password textbox |
| 4 | forgot password ? | Link | ------ | Link for navigate to Forgot password page for allows users to recover password. |
| 5 | Log in | Button | ------ | Login button navigates to another page even if valid login credentials. |

*Table 4.1.2 Screen element of Login form*

* + 1. **Screen-3: Add Task**

****

*Figure 4.1.3 Screen-3: Add task*

**Purpose**: This module will allow the system administrator to add, edit, update or delete task.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Sr. | Screen Element | Input Type | O/M | Description |
| 1 | Title | Textbox | M | Title of the task should be filled in textbox |
| 2 | Description | Textbox | M | Description field should be editable and accept Description. |
| 3 | Deadline | Date | M | Due date field should be editable and accept only date of the due date. |
| 4 | collaborators | Dropdown | O | Collaborators field should be editable and accept option. |
| 5 | Priority | Dropdown | M | Priority field should be editable and accept only option given. |
| 6 | Status | Dropdown | M | Status field should be editable and accept only given option. |
| 7 | Add Task | Button | ----- | Add task is a button for store the entered data into database. |

*Table 4.1.3 Screen element of Add Task*

* 1. **Testing**
     1. **Functional Testing**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test Case ID | Test Scenario | Test Steps | Expected Result | Status |
| TC001 | User Login | 1. Open Login Page 2. Enter valid username & password 3. Click "Login" | User is successfully logged in | Pass |
| TC002 | Invalid Login | 1. Enter incorrect credentials 2. Click "Login" | Error message displayed: "Invalid Credentials" | Pass |
| TC003 | Add New Task | 1. User logs in 2. Navigates to "Add Task" 3. Enters task details 4. Clicks "Add Task" | Task is successfully added | Pass |
| TC004 | Search for a Task | 1. User enters task name in search bar 2. Clicks "Search" | Matching task displayed | Pass |
| TC005 | Edit task | 1. User logs in 2. Selects a task to edit 3. Edit box open for task 4. Clicks “ Save” | Task is updated successfully | Pass |
| TC006 | Delete task | 1.User logs in  2.Clicks "Delete" | Task deleted successfully | Pass |
| TC007 | Share task | 1. User logs in 2. Click “Share” 3. Enter email 4. Click hare task | Task shared successfully | Pass |
| TC008 | Logout | Click on "Logout" button | User is logged out and redirected to login page | Pass |

* + 1. **Non-Functional Testing**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test Case ID | Test Scenario | Test Steps | Expected Result | Status |
| TC008 | Load Testing | 1. Simulate 100+ concurrent users 2. Perform task searches & transactions | System should not crash, response time ≤ 2s | Pass |
| TC009 | Security Testing | Attempt document Injection in login field | System should prevent document Injection | Pass |
| TC010 | Usability Testing | 1. Test UI for ease of use 2. Check accessibility for users | Users can navigate easily, UI is clear | Pass |

**CHAPTER 5**

**CONCLUSION AND FUTURE ENHANCEMENT**

* 1. **Conclusion**

The **Task Management System** provides an efficient and user-friendly platform for individuals to organize, track, and manage their tasks effectively. With core features such as task creation, prioritization, deadline management, progress tracking, and notifications, the system enhances productivity and ensures that users stay on top of their responsibilities. The integration of MongoDB as the database solution allows for scalability and flexibility, making the system adaptable to growing user needs. By offering a structured workflow and intuitive interface, this system serves as a valuable tool for personal task management and collaborative work.

* 1. **Future Enhancement**

While the current system covers the fundamental aspects of Task management, several enhancements can be made to improve its functionality and user experience:

1. **Collaborative Task Management** – Introduce team-based task sharing, allowing multiple users to collaborate on projects with assigned roles and permissions.
2. **AI-Based Task Prioritization** – Implement artificial intelligence to suggest task priorities based on deadlines, user behaviour, and workload.
3. **Mobile Application** – Develop a mobile version of the system for seamless task management on the go.
4. **Integration with External Tools** – Enable integration with popular productivity tools such as Google Calendar, Slack, and Trello for a seamless workflow.
5. **Dark Mode and UI Customization** – Enhance the user experience with customizable themes and accessibility features.
6. **Offline Mode Support** – Allow users to create and update tasks offline, syncing data when reconnected to the internet.

**APPENDICES**

**Appendix A: Acronyms and Abbreviations**

|  |  |
| --- | --- |
| Acronym | Full Form |
| TMS | Task Management System |
| DBMS | Database Management System |
| UI | User Interface |
| Context | A broad level diagram of the project showing a basic overview |
| Use Case | summarizes some of the relationships between Cases, user, and systems. |
| UML | Unified Modelling Language |

**REFERENCES**

[1] Kapil B. Patil, Software Requirement Specification Document, LinkedIn Corporation, Sunnyvale USA,2018, www.slideshare.net /ANASNAIN/17337071 srslibrarymanagementsystem

[2] Karl E. Wiegers, IEEE Standard Of SRS Document, IEEE, web.cs.dal.ca/~hawkey/3130/srs\_template-ieee.doc

[3] Roger S. Pressman, Software Engineering, McGraw-Hill, IA New York,2010. www.processimpact.com/process\_assets/srs\_template.doc.