# Table of Contents

1. Introduction x
   1. Purpose x
   2. Project Scope x
      1. Out of Scope x
      2. Project Motivation x
      3. Project Objectives x
      4. Project Expectations x
      5. Project Restrictions, Risks and Contingency Plan x
   3. Definitions, Acronyms, and Abbreviations x
2. User Requirements Definition x
3. Requirements Specification x
   1. Functional Requirements x
      1. Use Case Diagram x
         1. Requirement 1: User access to main page x
         2. Requirement 2: User login x
         3. Requirement 3: Real-time calendar view x
         4. Requirement 4 : Submit leave request x
         5. Requirement 5 : View leave balances x
         6. Requirement 6: Approve/ decline leave requests x
         7. Requirement 7: User notifications x
         8. Requirement 8: View company leave policies x
   2. Non-Functional Requirements x
      1. Performance/ Response time requirement x
      2. Availability Requirement x
      3. Recover Requirement x
      4. Robustness Requirement x
      5. Reliability Requirement x
      6. Security Requirement x
      7. Maintainability Requirement x
      8. Portability Requirement x
      9. Extendibility Requirement x
4. Interface Requirement x
   1. Graphical User Interface (GUI) x
   2. Application Programming Interface (API) x
5. System Architecture x
6. System Evolution x

# 1. Introduction

## 1.1. Purpose

The purpose of this document is to outline the requirements for the development of the Take A Day Off (T.A.D.O.) Leave Management web application.The web application will be developed with the aim to streamline and automate the leave management process in organisations and promote efficiency, transparency and compliance with the organisations’ policies and regulations. It serves as a centralised platform where employees can submit leave requests, receive approvals and updates, track leave balances along with employers/ HR teams to be able to manage the employee leaves.

## 1.2. Project Scope

The list below outlines the project scope elements for the development of the T.A.D.O. Leave Management web application:

* The web application will facilitate an employee in managing their leaves from requesting to tracking leave balances and view their daily schedules.
* The web application will facilitate an employer/ HR personnel/ administrator in managing leave requests, schedules and gathering relevant and historical data.

### 1.2.1. Out of Scope

This section outlines a list of what is outside of the scope for this web application:

* This web application will not support payroll processing such as managing the calculations for employee wages based on approved leaves or incorporating payroll systems
* This web application will not support any actions unrelated to leave management such as adding or updating new employee profiles, employee performance evaluations, employee work time tracking, workload scheduling, employee conflict resolutions, health and safety, etc.
* This web application will not be supported as a mobile application (The implementation of this is reliant on timeframe restrictions)

**1.2.2. Project Motivation**

The motivation for this project is to provide a streamlined and automated leave management process that allows organisations to manage leaves efficiently and effectively.

### 1.2.3. Project Objectives

The list of project objects are outlined below.

* Employees should be able to:
  + Log into their profile to manage leaves
  + Submit leave requests with the option to select the type of leave required
  + Receive notifications on their leave request status, upcoming leave and leave balances reaching a threshold through email or in-app alerts
  + Track their leave balances
  + View their schedules and leaves through a real-time calendar
  + View the organisation’s rules and regulations
* Employers/ HR personnel/ IT personnel/ Administrators should be able to:
  + Log in to their admin profiles
  + Manage the approval of leave requests
  + Customise policies in accordance to their organisation’s rules an regulations
  + View employee schedules through a real-time calendar
  + Gather any relevant and historical data for leave approval decision making

### 1.2.4. Project Expectations

The outcome of this project will be a user-friendly web application that automates and streamlines the entire leave management process of an organisation. Employees will be able to easily submit leave requests, view their schedules and leave balances while the management team can manage all approvals effectively.

### 1.2.5. Project Restrictions, Risks and Contingency Plan

Defining the project restrictions and risks help outline the limitations that guide the development process along with developing a backup strategy to ensure a well developed web application. The customisation of the web application is restricted to the features implemented directly for the purpose of the leave management process. Any further customisations unrelated to leave management may not be supported or feasible. The main restriction and risk for this project is the short defined timeline, completing and implementing all features along with any changes to the project scope may not be feasible within the allocated time frame. To ensure the completion of this project, certain objectives may be subjected to adjustments to align with the project deadlines.

## 1.3 Definitions, Acronyms, and Abbreviations

*LMS* - Leave Management System

*T.A.D.O* - Take A Day Off (Web Application Name)

*User* - The term user refers to the main actor/ actors using the system

*Admin/ Manager* - Refers to the user that has special control or access to modify requests or information in the system

**2. User Requirements Definition**

This section outlines all the user requirements needed to access the web application.

* Device Compatibility: The users require a device that is compatible with the web application. This can be a laptop, desktop computer, smartphone or a tablet.
* Internet Connection: Users must be connected to the internet in order to access the web application.
* Web Browser Compatibility: Users must access the web application through a supported web browser such as Google Chrome, Safari, Mozilla Firefox.
* Login Credentials: Users must have login credentials such as an email/ username and password to be able to access the web application.

# 3. Requirements Specification

# In this section, the functional and non-functional requirements of the web application will be described in detail with the aid of use case scenarios and use case diagrams.

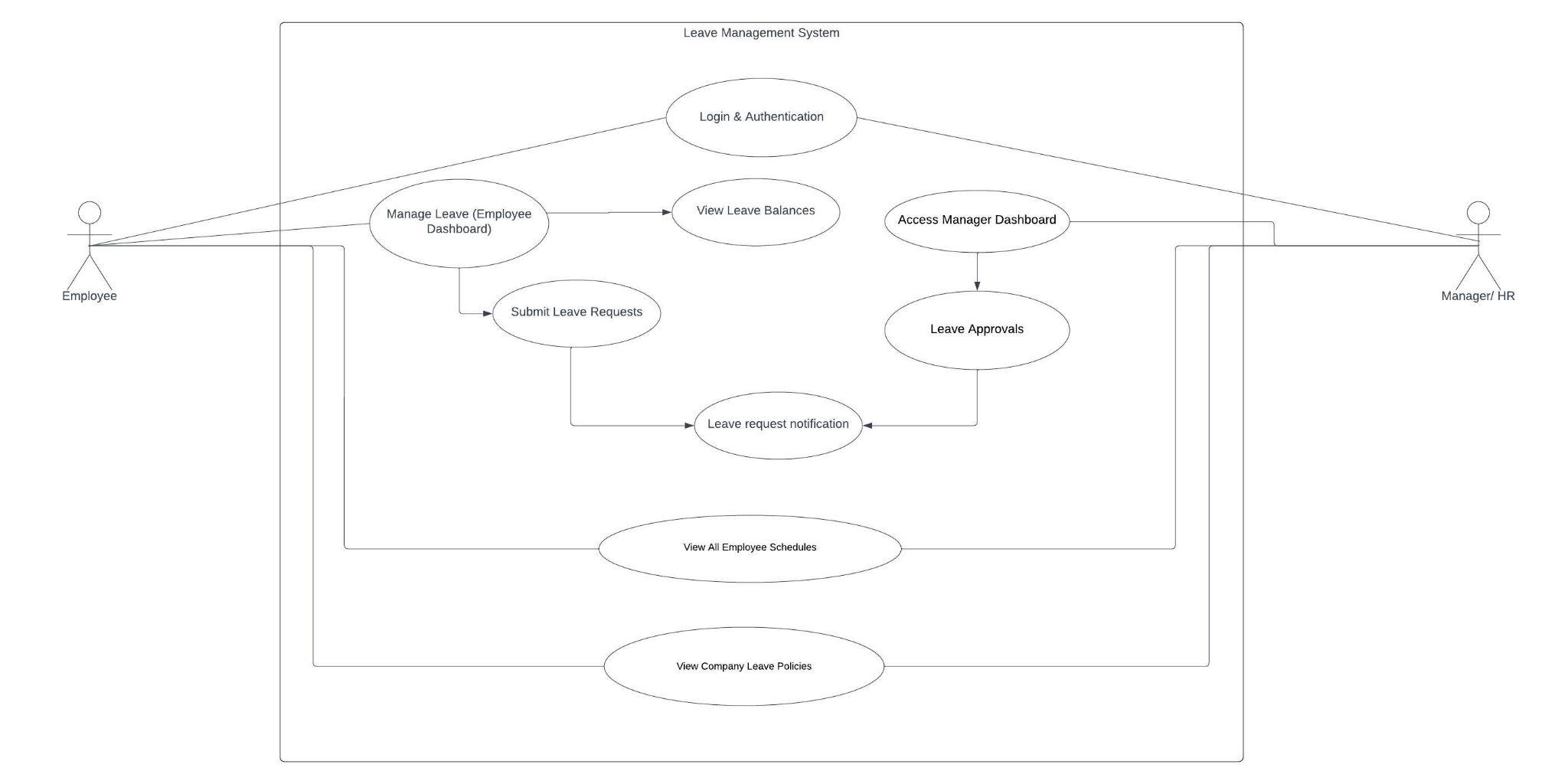
## 3.1. Functional Requirements

## The functional requirements outline the leave management system’s objectives and what it needs to achieve. The list of functional requirements are as follows:

1. Users are able to view the main page of the web application where they have the choice to log in, learn about the T.A.D.O. web application background, navigate through the web application through clicking onto different feature selections.
2. Users are able to login to the web application.
3. Users are able to view the real-time calendar where they can view their and co-workers schedules and availability.
4. Users are able to submit their leave request while selecting the type of leave and provide required and optional details such as start and end dates, reasons and comments. Users are also able to provide an attachment if desired.
5. Users are able to view their leave balances for the different types of leaves they have.
6. Users are able to approve or decline leave requests. Users are able to provide a reason for any leave requests declined.
7. Users are able to acknowledge or dismiss notifications regarding upcoming leaves, leave approvals, leave balance reaching a threshold.
8. Users are able to view the list of company policies and regulations.

### 3.1.1. Use Case Diagram

The use case diagram below outlines the actors and how they interact with the leave management system (Fig. 3.1.1.)



*Fig 3.1.1.: T.A.D.O. Use Case Diagram*

#### 3.1.1.1. Requirement 1: User access to main page

| **Use Case Name** | Accessing the main page |
| --- | --- |
| **Description** | This use case describes the process of users accessing the main page of the T.A.D.O. web application |
| **Actors** | Employees, Managers, HR and IT personnel, Public access to all |
| **Use Case Goal** | The user will be able to access the web application’s main page successfully and navigate through the options in the main page |
| **Preconditions** | The user has the necessary device, web browser and correct URL. The system is running and accessible |
| **Postconditions** | The main page is successfully displayed with the relevant features and information |
| **Activation/ Triggers** | The user navigates through the correct URL |
| **Flow Description** | |
| **Basic Flow** | 1. The user opens the web browsers and inputs the valid URL for the web application 2. The main page displays all relevant information and features available for the user to access such as login, company LMS sign-up form, information on features and benefits of the LMS, etc. |
| **Alternative Flow** | 1. If the user has issues loading the contents of the main page, the system will provide an error message with trouble-shooting steps |
| **Exceptional Flow** | 1. If the user inputs the incorrect URL, an error message will be shown - “This site can’t be reached” |

#### 

#### 3.1.1.2. Requirement 2: User Login

|  |  |
| --- | --- |
| **Use Case Name** | User Login |
| **Description** | This use case describes the process of employees accessing the T.A.D.O. web application to manage leave requests and view any leave-related details |
| **Actors** | All types of employees (including managers, HR/IT personnel) |
| **Use Case Goal** | The user will be able to login successfully and securely to the web application |
| **Preconditions** | The user should have a valid username/ email address and password |
| **Postconditions** | The user is successfully logged into the T.A.D.O. dashboard altered to their role type |
| **Activation/ Triggers** | The user navigates through the T.A.D.O. login page |
| **Flow Description** | |
| **Basic Flow** | 1. The user inputs their username and password on the appropriate boxes in the login page 2. The system validates the user credentials 3. If the user credentials are valid, the user is directed into their dashboard altered to their role type |
| **Alternative Flow** | 1. The user enters invalid user credentials 2. An error message will show stating “Invalid username or password. Please try again.” 3. The user will be able to enter their username and password again in the user inputs |
| **Exceptional Flow** | 1. After 10 consecutive retries, the user will be locked out of their account 2. The user can click on the “Forgot Password?” or “Recover account” button 3. The system will send a reset link to their registered email where they can reset their password   Note: The implementation of this process is subject to the allowed time provided for the project submission. If not implemented by the deadline, it shall be implemented for the future before the web application is fully released for commercial use |

#### 

#### 3.1.1.3. Requirement 3: Real-time calendar view

|  |  |
| --- | --- |
| **Use Case Name** | View calendar/ schedule |
| **Description** | This use case describes the process of employees accessing their individual and all employees schedules |
| **Actors** | Employees, Managers, HR and IT personnel |
| **Use Case Goal** | The user will have a view of all employees schedules in real-time |
| **Preconditions** | The user is logged into the system and can navigate the calendar through their employee dashboard |
| **Postconditions** | The calendar is displayed with all employee schedules and approved leaves |
| **Activation/ Triggers** | The user selects the “View All Schedules” in their dashboard |
| **Flow Description** | |
| **Basic Flow** | 1. The user clicks on the “View All Schedules” button 2. The system retrieves all schedules and approved leave information 3. The system displays the calendar containing all the details, colour-coded according to the employee availability 4. The user can navigate through days and months and is able to filter by employee name and date |
| **Alternative Flow** | 1. The user is able to filter through different team leads, managers or roles |
| **Exceptional Flow** | 1. The user receives an error message if there are issues with loading the calendar data |

#### 

#### 3.1.1.4. Requirement 4: Submit Leave Request

|  |  |
| --- | --- |
| **Use Case Name** | Submit Leave Requests |
| **Description** | This use case describes the process of employees submitting their leave requests in the web application |
| **Actors** | Employee |
| **Use Case Goal** | To successfully submit a leave request with the relevant leave type, dates and information |
| **Preconditions** | The employee is logged into the web application and have access to their dashboard |
| **Postconditions** | The leave request has been submitted for approval with all relevant information. The user receives a confirmation post the submission |
| **Activation/ Triggers** | The employee initiates the leave request process |
| **Flow Description** | |
| **Basic Flow** | 1. The user will login to the system 2. The user selects the “Apply Leave” option in their dashboard 3. The user fills in the leave request details such as start and end dates, leave types, comments and attachments 4. The user submits the request 5. The system sends a notification to the manager/ HR for leave approval 6. The user receives a confirmation message that the request has been submitted |
| **Alternative Flow** | 1. If the entered dates are invalid, an error message will be prompted for the user to correct the dates 2. If the leave balance for a specific type of leave is 0 or insufficient for the requested dates, the user will get an error message to correct the date range or type of leave |
| **Exceptional Flow** | 1. The user will receive an email or notification to request for alternative dates if the manager/ HR declines the leave request |

#### 

#### 3.1.1.5. Requirement 5: View Leave Balances

|  |  |
| --- | --- |
| **Use Case Name** | View Leave Balances |
| **Description** | This use case describes the process of retrieving employee leave balances |
| **Actors** | Employee, System |
| **Use Case Goal** | The employee is successfully able to view and review their leave balances for the different types of leaves |
| **Preconditions** | The employee is logged into the web application and have access to their dashboard |
| **Postconditions** | The leave balances for different types of leaves are displayed |
| **Activation/ Triggers** | The user selects the “View Leave Balances” option in their dashboard |
| **Flow Description** | |
| **Basic Flow** | 1. The user will login to the system 2. The user selects the “View Leave Balances” option in their dashboard 3. The system retrieves the balance details and displays all leave balances for different types of leaves |
| **Alternative Flow** | No alternative flow |
| **Exceptional Flow** | 1. If the user faces an issue with the system retrieving the leave details, an error message will be shown |

#### 

#### 3.1.1.6. Requirement 6: Approve/ Decline Leave Requests

|  |  |
| --- | --- |
| **Use Case Name** | Approving or Declining Leave Requests |
| **Description** | This use case describes the actions taken by managers/ HR personnel when approving or declining leave requests |
| **Actors** | Manager, System |
| **Use Case Goal** | To efficiently manage employee leave requests and provide a response |
| **Preconditions** | The manager is logged into the system. There are employee leave requests submitted and available for review |
| **Postconditions** | The manager approves or declines the leave request and the system sends a notification/ response to the employee whilst updating all information |
| **Activation/ Triggers** | The manager navigates through the “Leave Approvals” section in their dashboard or receives a notification regarding a pending leave request |
| **Flow Description** | |
| **Basic Flow** | 1. The manager will login to the system 2. The manager selects the “Leave Approvals” option in their dashboard 3. The manager views the list of pending requests 4. The manager selects a leave request, a dropdown will be shown with all the information and the manager reviews all the details 5. The manager selects “Approve” or “Decline” 6. The system updates the leave request status 7. The employee receives a notification with the updates |
| **Alternative Flow** | 1. If there are no pending requests, an error message will be shown - “No pending requests” |
| **Exceptional Flow** | 1. An error message will be shown if the manager faces an error when approving or declining the leave requests |

#### 

#### 3.1.1.7. Requirement 7: User Notifications

|  |  |
| --- | --- |
| **Use Case Name** | User Notifications |
| **Description** | This use case describes the process of users receiving notifications regarding leave requests, upcoming leaves and leave balances reaching a threshold |
| **Actors** | Employees, Managers, System |
| **Use Case Goal** | To have users informed and updated about their leave statuses |
| **Preconditions** | The user must be logged into the system |
| **Postconditions** | The user receives and views notifications |
| **Activation/ Triggers** | * The user submits a leave request * The request has been actioned by the manager * The user's leave balances are reaching a threshold |
| **Flow Description** | |
| **Basic Flow** | 1. The user logs into the system 2. The system checks for any leave requests approvals, upcoming leaves or leave balances reaching a threshold 3. The system generates a notification for any relevant information 4. The system sends a notification in the notification bar or email to the user containing the relevant events 5. The user receives and views the notifications or email and is able to acknowledge |
| **Alternative Flow** | 1. No notifications will be generated by the system if the user has no upcoming leaves, leave requests |
| **Exceptional Flow** | 1. If there are issues or errors with the system retrieving relevant information, the user will not receive any notifications. The user w3.1.be able to manually check their leave information by navigating through their dashboard |

#### 

#### 3.1.1.8. Requirement 8: View Company Leave Policies

|  |  |
| --- | --- |
| **Use Case Name** | View Company Leave Policies |
| **Description** | This use case outlines how the users can view the company's rules and regulations on leaves |
| **Actors** | Employees, System |
| **Use Case Goal** | The user views all rules and regulations related to leaves |
| **Preconditions** | The user must be logged into the system and have access to their dashboard |
| **Postconditions** | The user views all leave related company rules and regulations |
| **Activation/ Triggers** | The user selects the “View Company Leave Policies” option in their dashboard |
| **Flow Description** | |
| **Basic Flow** | 1. The user logs into the web application 2. The user selects the “View Company Leave Policies” button in their dashboard 3. The system retrieves all leave related company rules and regulations and displays it 4. The user views and reads all information 5. the user can navigate back to their dashboard |
| **Alternative Flow** | 1. If there is an error in retrieving the data, an error message will be shown and the user can try again later |
| **Exceptional Flow** | N/A |

## 

## 3.2. Non-Functional Requirements

### 3.2.1. Performance/ Response time requirement

The system response time for submitting a leave request, approval of request and sending notifications should not take longer than 2 to 3 seconds to ensure a prompt and efficient user experience.

### 3.2.2. Availability Requirement

The LMS should ensure consistent service and functionality for users while minimising downtime or interruptions. The users should be able to easily access the application through various compatible devices and web browsers.

### 3.2.3. Recover Requirement

In the event of system failure, issues or loss of data, the system should be able to recover and restore any critical data. To mitigate this, regular data backups are to be maintained.

### 3.2.4. Robustness Requirement

The web application should be able to handle any invalid inputs, unexpected user behaviours and errors by providing meaningful/ descriptive error messages.

### 3.2.5. Reliability Requirement

The system should be able to handle multiple users and user requests at the same time while running seamlessly and continuously without any issues and maintaining data.

### 3.2.6. Security Requirement

The web application will follow secure implementation in accessing sensitive information such as role-based access/ controls, user authentication and employee details.

### 3.2.7. Maintainability Requirement

The web application should be well-documented and structured to allow for future modifications.

### 3.2.8. Portability Requirement

The web application should be easily accessed through different devices (e.g. laptop, desktop, tablet) and web browsers (e.g. Google Chrome, Safari).

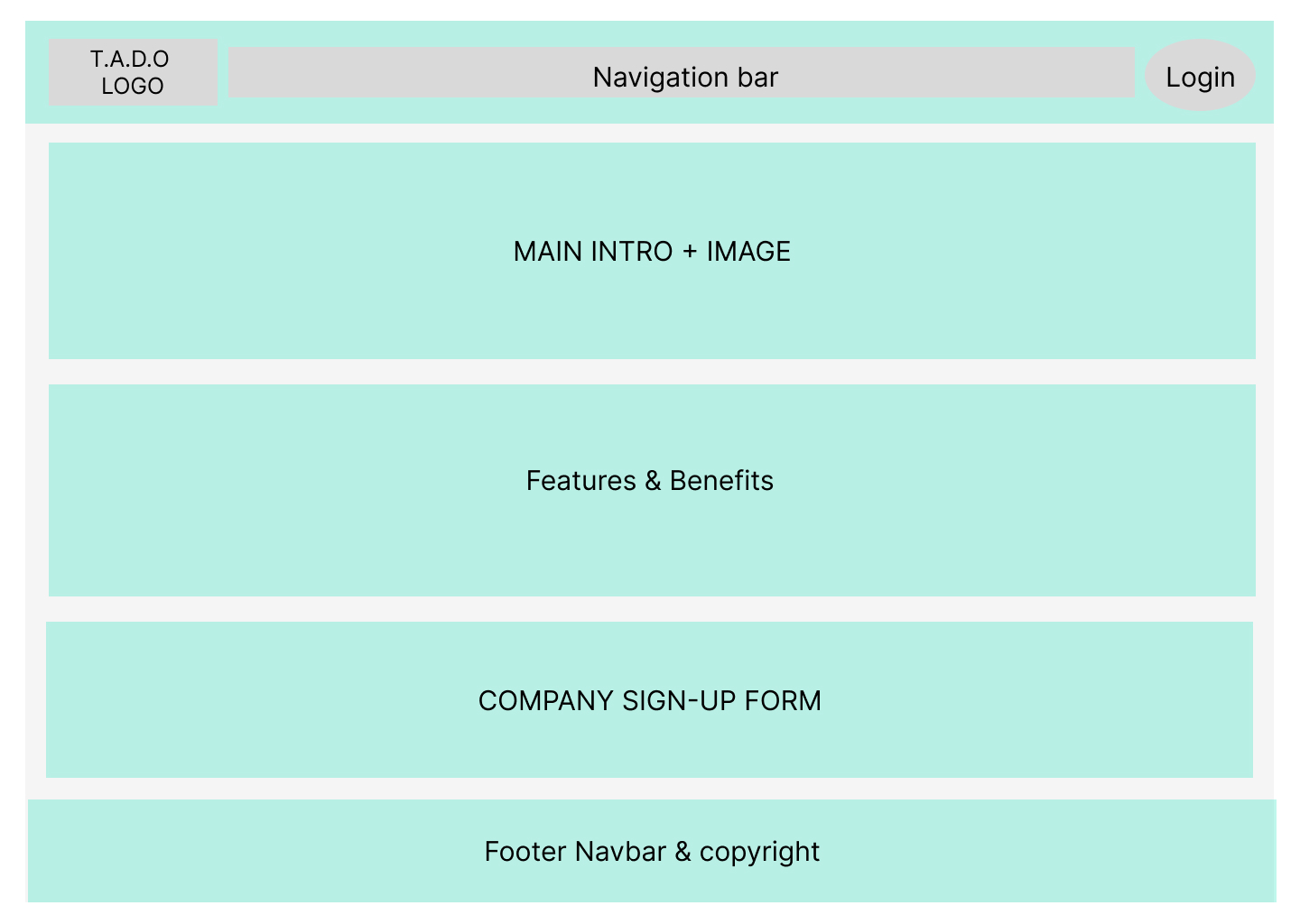
### 3.2.9. Extendibility Requirement

The web application should allow for extendibility and be able to accommodate future enhancements or extensions post deployment.

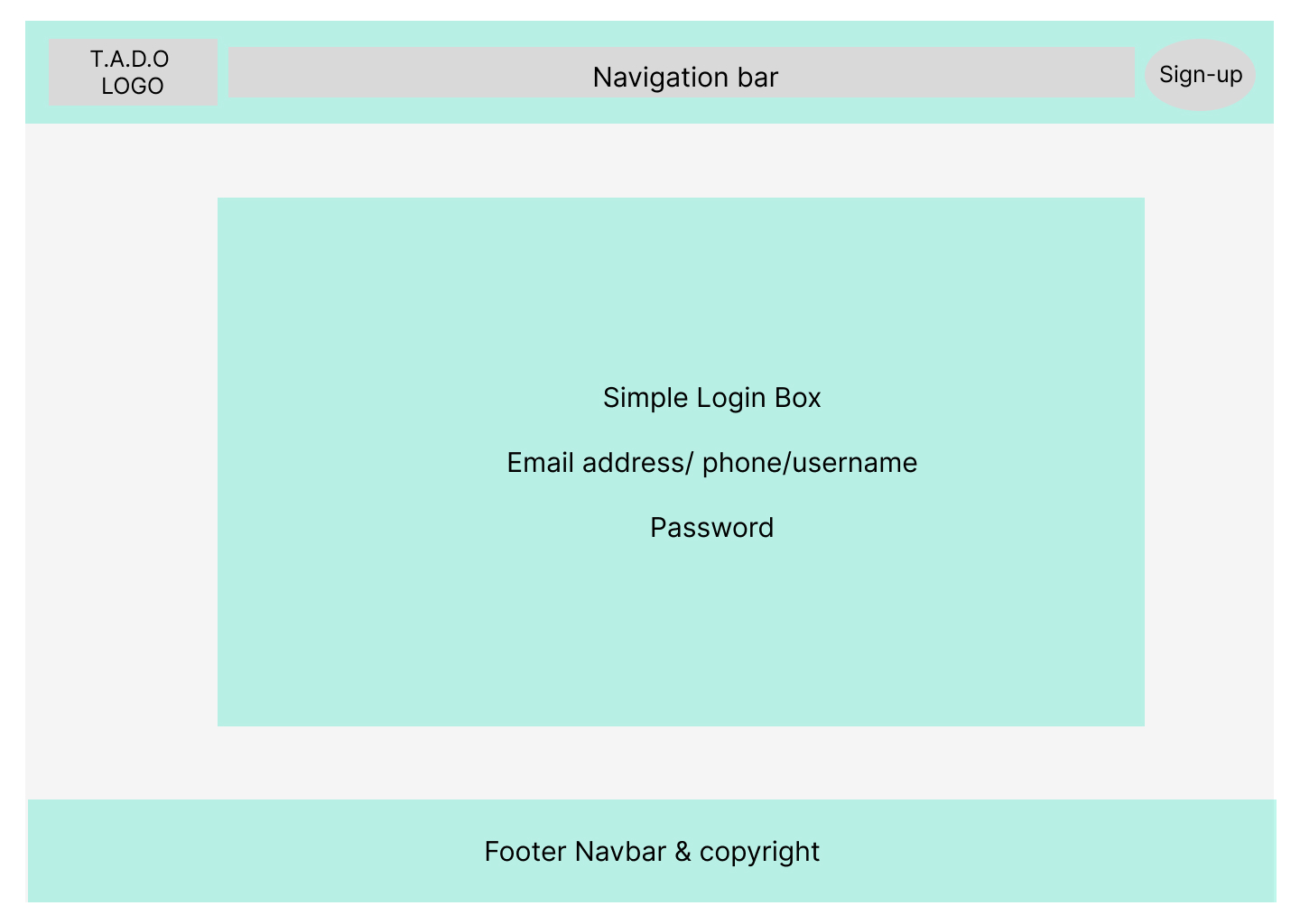
# 4. Interface Requirement

### 4.1. Graphical User Interface (GUI)

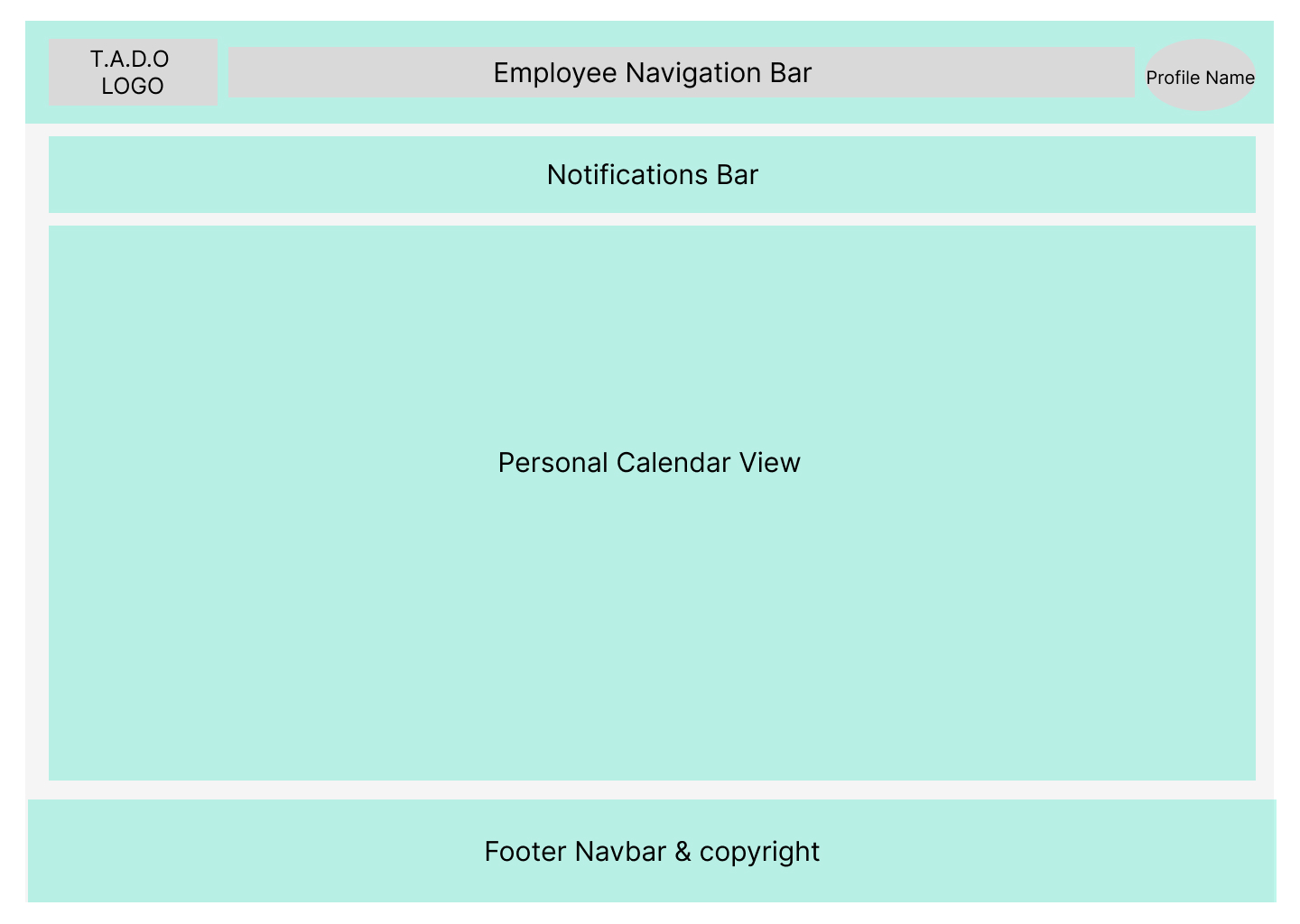
A simple GUI will be integrated for this project to allow for simplicity and user-friendliness. In the use case User Access to the main page, the GUI will look similar to the image presented in *Fig 4.1.1.* The user will be able to login by clicking the login button in *Fig 4.1.1.* and providing user credentials in *Fig 4.1.2.* The user will be able to view their employee dashboard as shown in *Fig 4.1.3.* where the employee can submit leave requests through a form as shown in *Fig 4.1.4.* Managers are able to view their dashboards similar to *Fig 4.1.5.* where they are notified of any leave approvals pending through the notification bar. *Fig 4.1.6.* shows how the approval page for leave requests will look. All users are able to view a real-time calendar showing all employee schedules in a similar layout to *Fig 4.1.7.* The GUI shown below are subject to change depending on the progress of the project and the given timeframe.



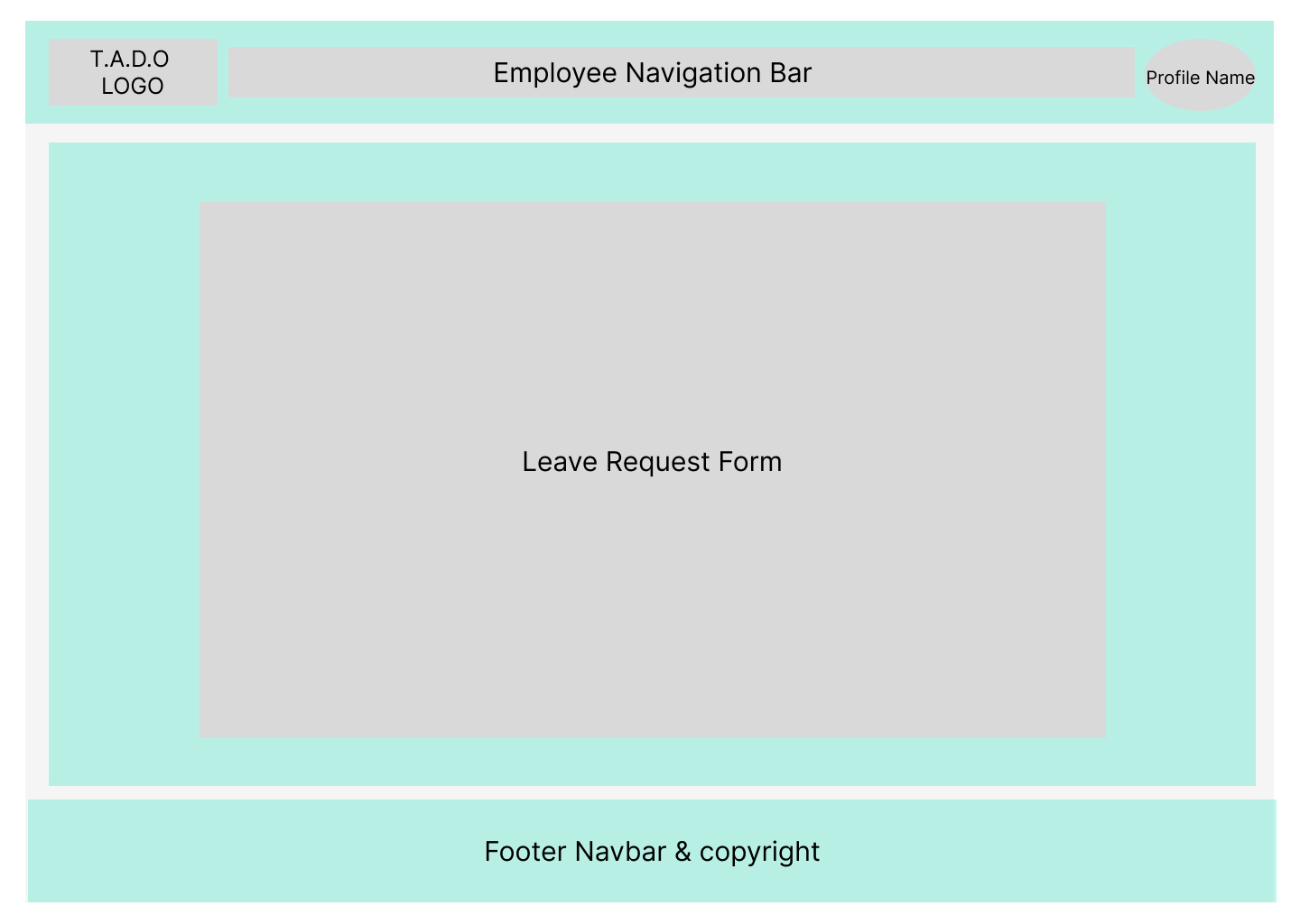
*Fig 4.1.1.: T.A.D.O. Main Page*

**

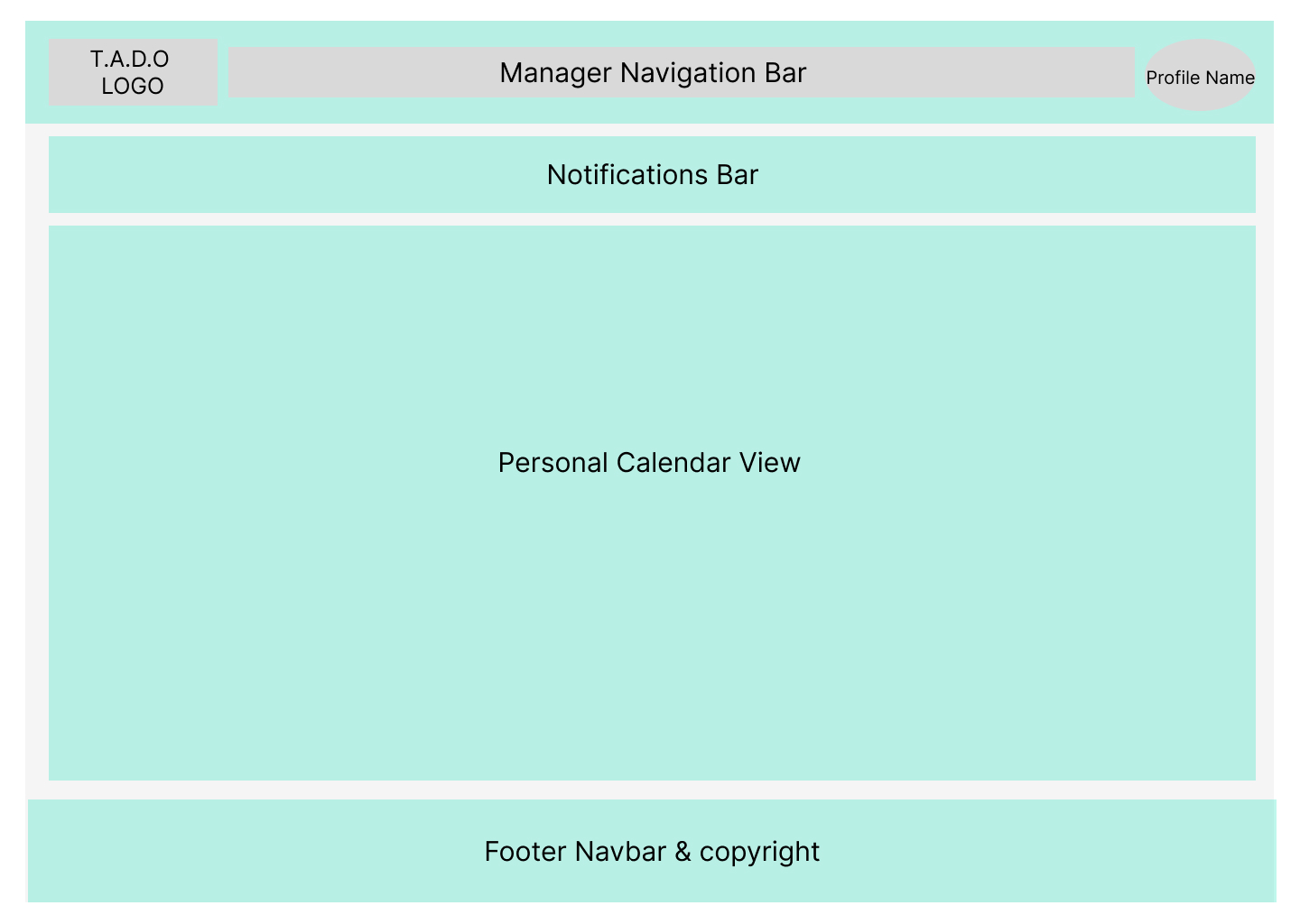
*Fig 4.1.2.: Login Page*

**

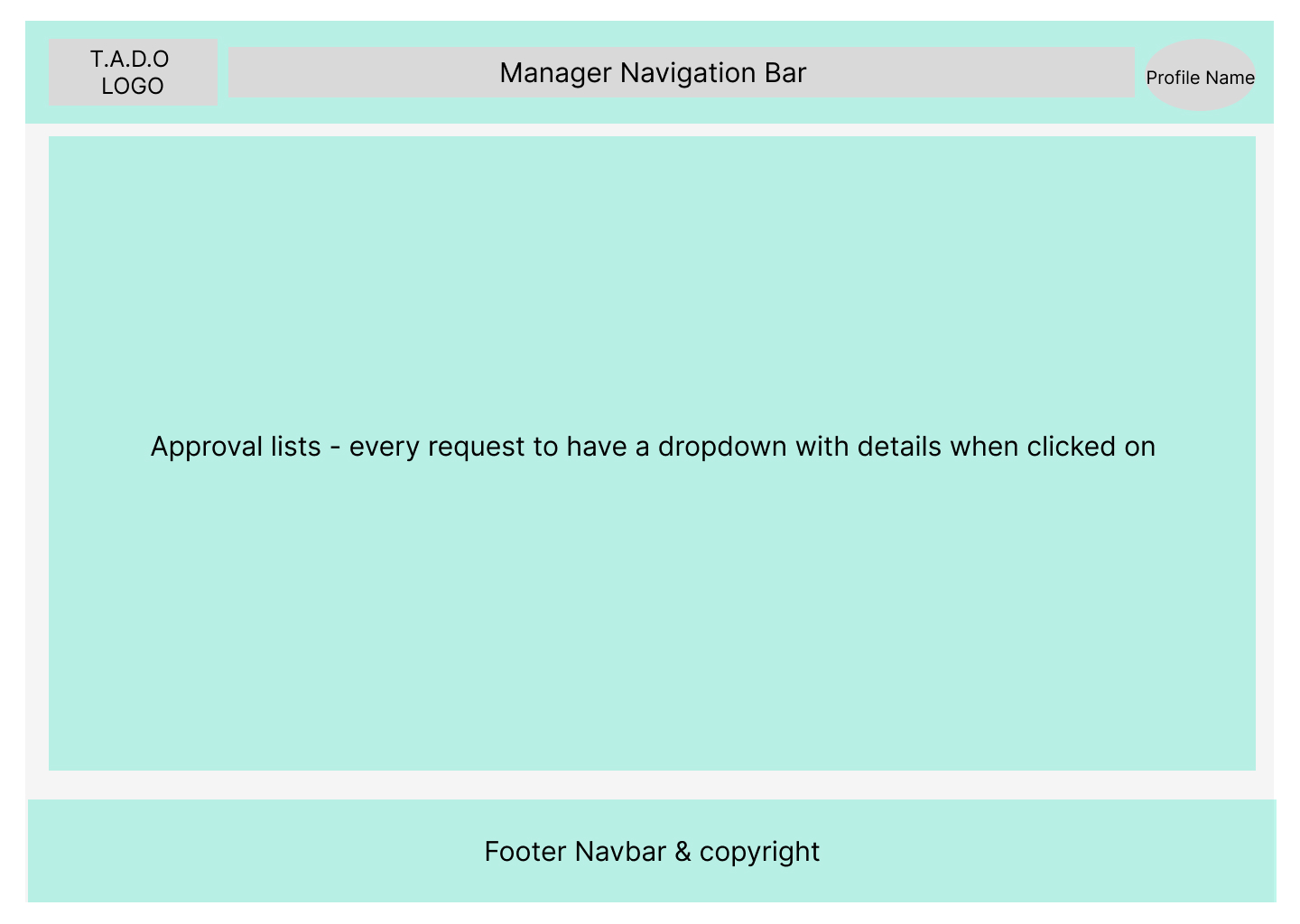
*Fig 4.1.3.: Employee Dashboard Page*

**

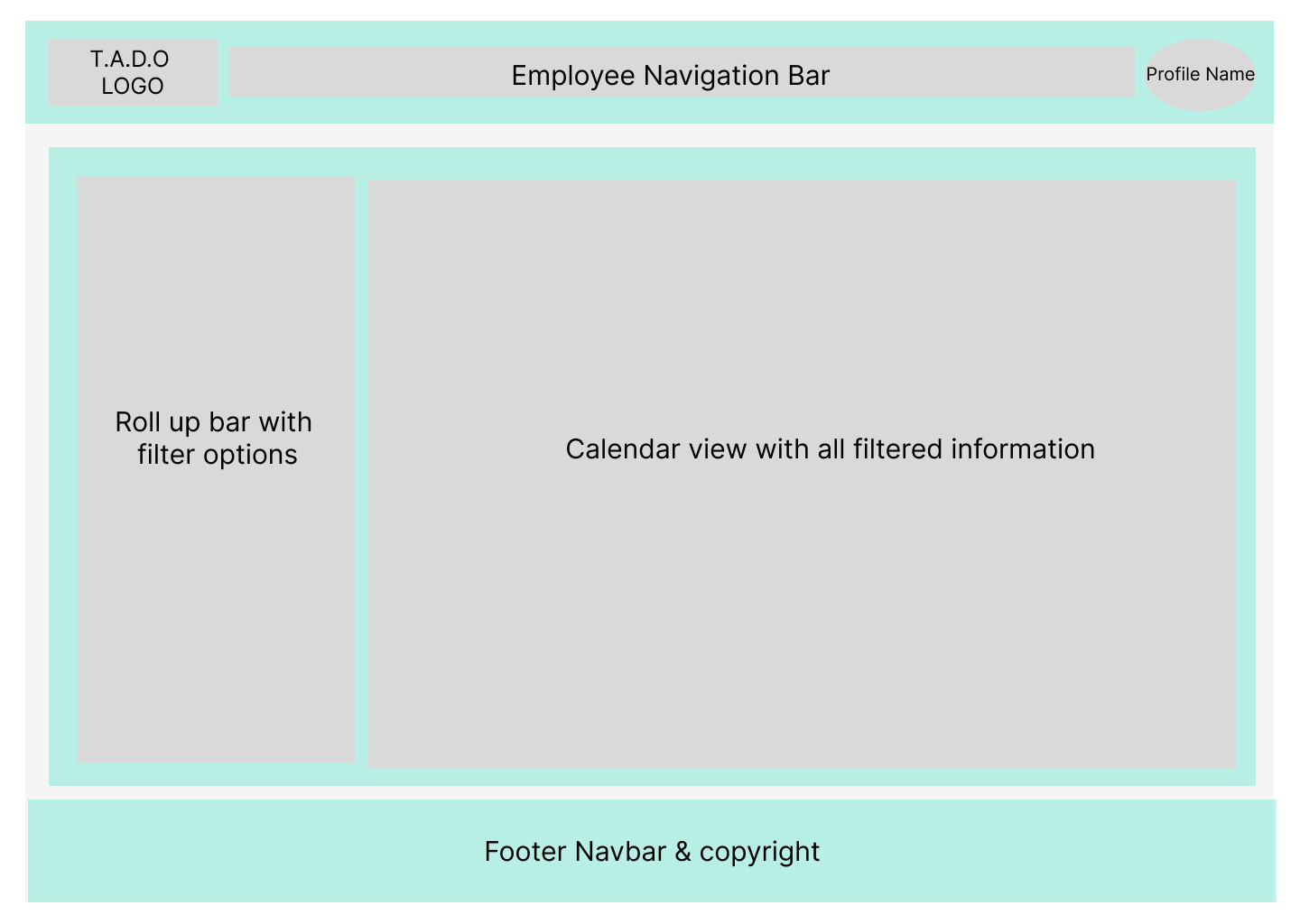
*Fig 4.1.4.: Leave Request Page*

**

*Fig 4.1.5.: Manager Dashboard Page*

**

*Fig 4.1.6.: Approval Page*

**

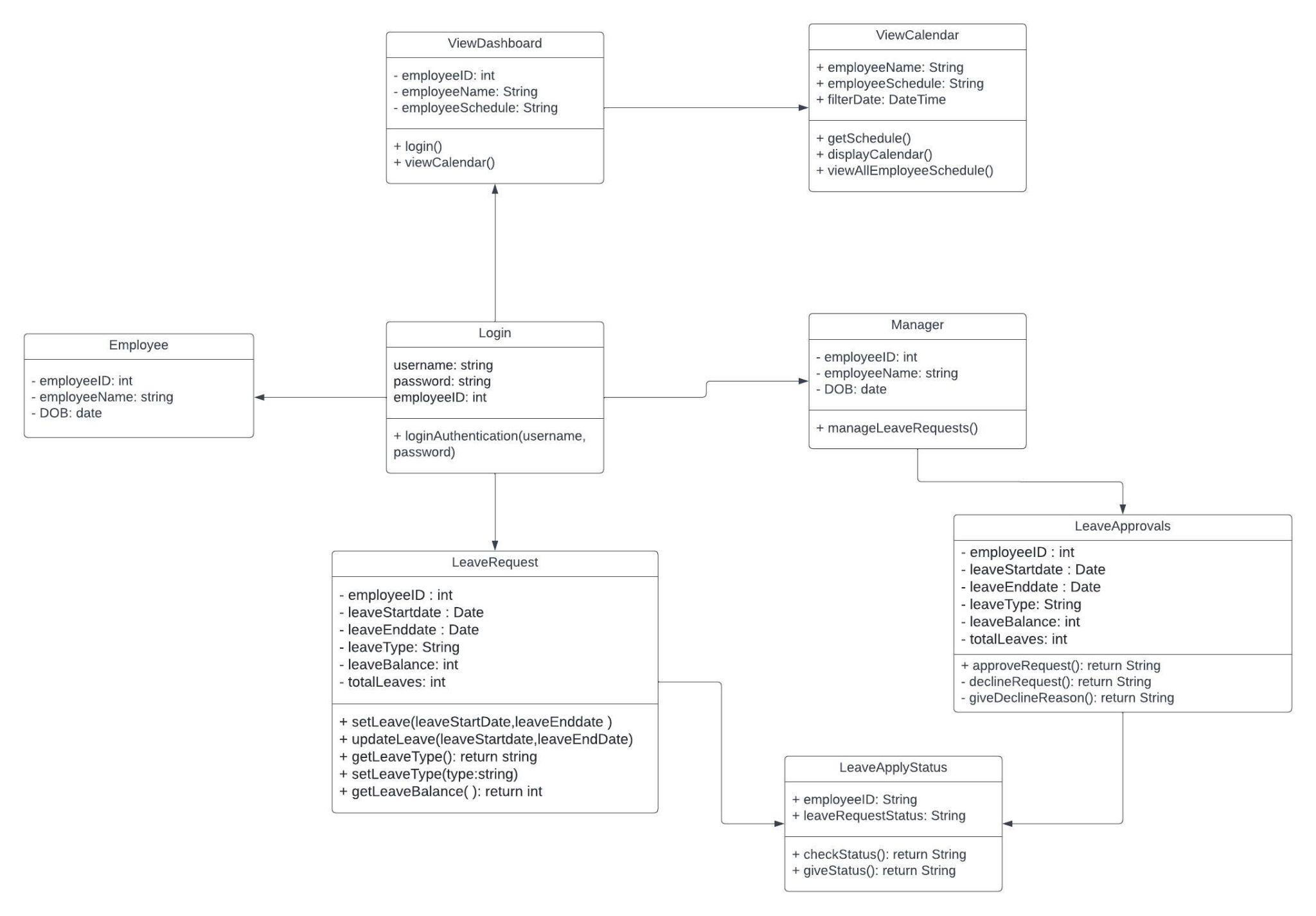
*Fig 4.1.7.: Calendar View Page - All Employee Schedules*

### 4.2. Application Programming Interface (API)

There are a few use cases that the API will need to support in this project and this includes the leave request submission, leave request approval and retrieving the leave balances. The RESTful principles will be followed in designing the API which include the HTTP methods (GET, POST, PUT, DELETE). The structure of the request and response data will be needed to be considered along with defining the API endpoints. As for the request and response formats, I am considering using JSON as the data format for readability and simplicity purposes. This part of the project is still under review and can change as the project progresses.

# 5. System Architecture

The class diagram (Fig. 5.1.) below outlines the initial design structure for the T.A.D.O. web application. This structure may change as the project progresses.



*Fig 5.1.: T.A.D.O. Leave Management System Class Diagram*

# 6. System Evolution

The T.A.D.O. LMS web application is designed for the purpose of this project and is open to further improvements and revision for future commercial use.

1. Login to include mobile or email authentication
2. Real-time calendar view to include filters based on different supervisors, roles, departments, location, etc.
3. Leave requests and approvals to automatically consider all employee availability and headcount requirement for each day
4. Add HR capabilities such as adding new employees, etc.