Graduation Project:

Workflow Management System

Supervised by

Dr: Ahmed Gaber

Co-Supervision

Eng.Abderhman Sherif

Project No: 14

Academic Year :2023/2024

Graduation Project:

Workflow Management System

20200533-احمد محمد عبدالمحسن

20190798- خالد سيد قناوي

20181344- صلاح الدين عبدالغفور صالح

20193786- مازن حسين حسن

20192056- انس رفاعي احمد

20183280- سيف احمد سيف الدين

20184983- عمر طارق السيد

20201640- محمد خالد عطيه

Supervised by

Dr: Ahmed Gaber

Co-Supervision

Eng.Abderhman Sherif

Project No: 14

Academic Year :2023/2024

Acknowledgment

We would like to express our sincere gratitude and appreciation to the dean **Prof**. **Dr. Ahmed salama** for your invaluable guidance and support throughout the project. Your expertise, insights, and encouragement have been instrumental in the successful completion of this endeavor.

We all have special thanks to **Dr. Ahmed Gaber**

for his useful comments for us and support to us and his guidance in achieving theseresults and for providing us with all the necessary facilities.

We are truly grateful for the opportunity to work under your supervision. Your mentorship and leadership have enhanced our skills and knowledge. Your commitment to excellence and attention to detailhas served as a constant source of inspiration.

We would also like to acknowledge the time and effort you have invested in reviewing and providing feedback on our work. Your constructive criticism and valuable suggestions have significantly contributed to improving the quality and effectiveness of the project.

Special thanks go to **Eng. Abdelrahman sherif**
Our supervisor on this project to support us throughout the year. He was very patient with us and acted like a real guide when we needed him. We have greatly benefited from his knowledge.

With sincere appreciation,

Project Abstract

This project aims to develop a workflow management system for a Institute , which helps to streamline the communication and task management between different levels of the Institute staff, such as the Dean, Vice Dean, Heads of Department, doctors, and Assistant doctors.

The system includes four levels of users, with each level able to assign tasks to those of lower levels. A Secretary, who can communicate with the Dean only, and an Admin, who can view tasks but cannot see any details of the task to the security and can delete it, are also included.

Each user has a personalized dashboard that they can access, which displays all tasks assigned to them and tasks assigned to others.

This feature allows users to easily keep track of their tasks and responsibilities and can also help to improve accountability and communication among the Institute staff.

The system also includes a feature for requesting and approving vacation leave for staff members with different types like usual leave, Casual leave, Sick leave, Unpaid leave, Commission leave, and Permission.

The implementation of this system is done using [Flutter,Dart ,Firebase,MySQL,Visual Studio Code, and Figma], and its user interface is designed to be user-friendly

The system was tested and evaluated, and the results show that it significantly improves communication and task management within the Institute.

Contents

List of Figures

CHAPTER 1: INTRODUCTION

Chapter1: Introduction

1.1 Introduction:

The workflow Management System is a web-based platform designed to facilitate seamless communication, task, progress tracking, and feedback exchange between various roles within the faculty or department. The management of workflow between different levels of workers in a Institute is a complex and time-consuming task. In the past, the Institute relied on manual methods such as paper documents to manage tasks and vacation requests. This approach was prone to errors, lack of transparency, and inefficiency. The process was also not easily accessible and required a significant amount of manual effort. As a result, the Institute administration faced challenges in monitoring and controlling the workflow and there was a lack of collaboration and communication between different levels of workers. The proposed faculty or department use aims to streamline task management processes, improve collaboration, and enhance productivity. By providing a centralized platform for task creation, vacation, tracking, and feedback exchange, the system empowers users at different levels to efficiently manage their responsibilities and work together toward achieving common goals. With its user-friendly interface and comprehensive feature set, this system will be a valuable tool for educational institutions seeking to optimizetask management and coordination among faculty members.

1.2 Overview:

The system is a web-based application that helps manage the workflow between workers in a Institute. It is designed to simplify the task and tracking process and the vacation request and approval process. The system has four levels of users: Dean, Vice Dean, Heads of Department, and Doctors and Assistant Doctors. Each level has different roles and permissions.

The Dean can assign tasks to any level, while the Vice Dean, Heads of Department, and Doctors and Assistant Doctors can only assign tasks to levels below them. The Secretary can communicate with the Dean only, and no one can assign tasks to her except the Dean. The admin can view tasks and delete them but cannot edit or see details of tasks.

The system also includes a vacation request feature, where doctors and assistant doctors can submit requests for different types of leave. The requests then needto be approved by the Head of Department, and if accepted by the Dean or Vice Dean, do not require further approval from the Head of Department.

Each level of users has a dashboard that allows them to view all tasks assigned to them and tasks assigned to another person. The system also provides an easy-to-use user interface for all users and improves the security of sensitive information by using an electronic system.

Overall, the Task Management System serves as a centralized platform for efficient task creation, vacation, tracking, feedback exchange, and collaboration among faculty members, ultimately enhancing productivity and coordination within the educational institution.

1.3 Problem Definition:

In educational institutions, effective task management and coordination among faculty members are essential for ensuring smooth workflow, productivity, and collaboration. However, the current manual methods of managing the workflow in a Institute are inefficient and prone to errors. There is a need for a system that streamlines the task and tracking process and provides transparency and efficiency. The system should also facilitate the request and approval process for vacations. The previous reliance on paper-based systems has led to confusion anddelays, and a need for a more efficient and reliable solution. A system that addresses these issues would help the Institute administration to better monitor and control the workflow, improve communication and collaboration between different levels of workers, and enhance overall productivity.

1.4 Challenges:

- **User Adoption**: The system requires all users to learn and adapt to new processes and procedures, which can be a challenge.
- **Data Security**: Ensuring the security of sensitive information stored in thesystem is a crucial challenge. The system must protect against unauthorized access, data breaches, and other security threats.
- **System Integration**: The system must be able to integrate with existing systems and databases in the Institute, such as student information systems and payroll systems.
- **System Scalability**: The system must be able to handle a large number of users, tasks and be able to scale to meet the needs of the Institute as it grows.
- **Data Backup and Recovery**: The system must have a robust data backup and recovery plan in place to ensure that data is not lost in the event of a system failure.
- **User Authentication**: The system must have a secure and reliable user authentication process to ensure that only authorized users can access the system.

- **Technical Support**: The system must have adequate technical support in place to help users troubleshoot any issues they may encounter.
- **Maintenance and Upgrades**: The system must be maintained and upgraded regularly to ensure that it continues to function correctly and meet the needs of the Institute.

1.5 Project Objectives:

- Centralization: The Task Management System aims to centralize task management processes within educational institutions, providing a single platform for faculty members to manage tasks, collaborate, and communicate effectively.
- Efficient Task: The system seeks to streamline task by providing a structured and organized method, reducing misunderstandings, missed deadlines, and improving overall accountability.
- Enhanced Feedback Exchange: The purpose of the system is to facilitate a structured feedback mechanism, allowing faculty members to provide valuable insights, suggestions, and improvements to foster continuous learning and collaboration.
- Improved File Management: The system aims to enable easy file upload, storage, and processing, ensuring faculty members have access to relevant resources and promoting efficient documentation.
- Effective Progress Tracking: The system's goal is to enable transparent progress tracking, allowing faculty members and management to monitor task status, evaluate performance, identify bottlenecks, and make necessary adjustments.
- Hierarchical Management Support: The system seeks to support hierarchical management by allowing the Dean to assign tasks to the Vice Dean, manage branches, and provide effective feedback and guidance.
- Increased Visibility and Notifications: The system aims to provide visibilityinto task progress for both higher-level management and subordinates, ensuring effective coordination and timely completion. Users will be notified of any changes or updates to task details.
- User Account Management and Support: The system aims to provide efficient user account management, access control, password authorization, and prompt resolution of user issues through the System Administrator role.

The overall purpose of the Task Management System is to optimize task management, streamline collaboration, enhance communication, and improve overall productivity within educational institutions' faculty or department.

1.6 The technology stack:

- **Front-end Development**:
  - Dart: Programming Language Developed By Google.
  - Flutter: UI Tool Kit.

- **Back-end Development**:
  - Firebase: Cloud Computing Services

The front end of the system is built using Dart, and Flutter framework being utilized to create a dynamic and responsive user interface.The combination of these technologies allows for the creation of visually appealing and interactive web pages.

The back end of the system is built using Firebase framework, which provides a robust and secure platform for the development of server-sidelogic. And Fire Store which is a widely used and well-established relational database management system that provides reliable and efficient storage of data.

Together, the front-end and back-end technologies used in the system provide a flexible and scalable solution for the development of web-based applications. The use of these technologies ensures that the system can be easily maintained and updated as the needs of the users evolve.

CHAPTER 2: Literature Review

Chapter 2: Literature Review

2.1 Introduction:

The literature review aims to explore existing research and practical implementations related to task management systems in the context of an educational institution. It covers key areas such as task management system concepts, user authentication and access control, file upload and processing, feedback and collaboration mechanisms, notifications and real-time updates, and the role of system administrators. By examining relevant literature, the review aims to identify gaps, best practices, and insights that can inform the design and development of an efficient and user-friendly task management system.

2.2 Web-based service to workflow management system for the project

To create a web-based task management system with the specified requirements, you will need to develop a comprehensive workflow management system that includes user authentication, task creation, vacation, task tracking, file uploading, feedback mechanisms, and user roles and permissions. Here's an outline of the key components and functionalities you should consider when developing the system:

1- User Registration and Authentication:
- Users can create an account with appropriate credentials.
- User roles should be defined (e.g., Dean, Vice Dean, Heads of Department, Doctors, Teaching Assistants, Secretary, System Administrator).

2- User Management:
- The System Administrator (Admin) can create and manage user accounts.
- Admin has the authority to remove or control user access.
- Admin can authorize password changes and address user issues.

3- Task Creation:
- Users can create tasks using text or attachments.
- Tasks can be assigned to other users with appropriate permissions.
- Users can set deadlines for tasks or assign them immediately.
- Users can track the progress of tasks and mark them as completed.
- Users can determine if tasks are related or separate.

4- Task Hierarchy and Feedback:
- Dean can send tasks to the Vice Dean, Secretary, and manage the Vice Dean.
- Vice Dean can manage other branches (Heads of Department, Doctors, Teaching Assistants).
- Dean can give feedback to the Vice Dean.
- Vice Dean can give feedback to Doctors.
- Doctors can give feedback to Teaching Assistants.

5- File Upload and Processing:
- Users can upload files related to tasks for processing.
- Files should be stored securely and associated with the respective tasks.

6- Announcements:
- Users can create announcements for important deadlines, events, or meetings.
- Announcements should be visible to relevant users but not treated as tasks.
- Higher levels of management can view task progress of subordinates.
- Subordinates should be notified of any changes in task details.

7- Database Interaction:
- The system should be able to send and retrieve data from a database.
- Task and user data should be stored in the database securely.

8- User Interface and Accessibility:
- Design a user-friendly web interface for easy navigation and task management.
- Different user roles should have access to specific features based on their permissions.

9- Error Handling and Support:
- Implement error handling mechanisms for common user issues.
- Admin should be able to assist users with problems, such as deleting tasks if necessary.

2.3 System Components:

User Interface: This component provides an interface for the users to interact with the system. It includes features such as login, task, vacation request, and viewing dashboard.
Database: This component stores all the data related to the system, such as user information, task details, and vacation requests.
Business Logic: This component handles all the business logic of the system, such as task, vacation request approval, and user management.
Security: This component handles all the security measures of the system, such as user authentication, access control, and data encryption.
Communication: This component handles the communication between the different components of the system, such as sending notifications, emails, and SMS.
Reporting: This component generates reports and statistics based on the data stored in the system.

2.4 The scalability and performance:

Database design: The system's database has been designed to scale horizontally by sharing the data across multiple servers. This allows the system to handle a large number of concurrent users and requests without experiencing a significant drop in performance.
Caching: The system uses caching to temporarily store frequently used data, reducing the number of database queries and improving the system's overall performance.
Asynchronous processing: The system uses asynchronous processing tohandle a large number of requests simultaneously, reducing the time it takes for the system to respond to a user's request.
Monitoring and logging: The system has built-in monitoring and loggingcapabilities, which allow system administrators to track performance metrics and identify and troubleshoot performance bottlenecks.

Overall, the system has been designed with scalability and performance in mind and will be able to handle a large number of concurrent users and requests without experiencing a significant drop in performance.

2.5 Conclusion:

The review highlighted the benefits of web-based services and identified key features such as task, deadline management, and notifications. Limitations included scalability and data entry challenges. The findings will guide the development of an intuitive system that meets user requirements, and user research and testing will be important for refining the design. Overall, the review established a solid foundation for the development of a practical and user-centric web-based workflow management system based on tables.

CHAPTER 3: Methodology

Chapter 3 : Methodology

3.1 Introduction

This chapter presents the methodology used in the development of the Task Management System. It outlines the requirements, both functional and non-functional, that guided the system's design and implementation. Additionally, it highlights the software tools and technologies utilized throughout the development process.

3.2 System Requirements

The Diagrams and Flowcharts section provides a visual representation of the different processes and components of the system. It includes diagrams such as block diagrams, entity-relationship diagrams, and flowcharts that help to explain the system's architecture and functionality. This section is intended to provide a clear and detailed understanding of how the system works and how the different components interact with each other. The diagrams and flowcharts used in this section are designed to be easy to understand and provide a comprehensive overview of the system.

3.2.1 Functional Requirements

- The website:

  - The website can accept and save data from users.
  - Users can upload files for processing.
  - Users are required to log in to access certain features.
  - The website can send and retrieve data to and from the database.
  - Tasks can be created using text or attachments.
  - Tasks can be assigned to other users.
  - Tasks can be scheduled for a particular day with a deadline.
  - Users can track the progress of tasks.
  - Users can separate tasks and determine if they are related or not.
  - Users can choose who can complete the task.
  - Users can submit tasks.

- Users can give updates to each other through announcements.
- Higher levels of management can view the task progress of subordinates.
- Subordinates are notified of changes in task details.

- User Roles and Permissions:

  - The system supports multiple user roles, including Dean, Vice Dean, Heads of Department, Doctors, Teaching Assistants, and Secretary.
  - Dean can send tasks to Vice Dean.
  - Dean can manage the Vice Dean.
  - Vice Dean can manage other branches such as Heads of Department, Doctors, and Teaching Assistants.
  - Dean can give feedback to the Vice Dean.
  - Vice Dean can give feedback to Doctors.
  - Teaching Assistants can view tasks assigned by head of department but cannot give tasks.

- Account Management:

  - The System Administrator (Admin) is authorized by faculty executives tocreate an account to access the system.
  - Admin can manage user account information.
  - Admin can remove or control user access.
  - Admin can authorize a user account password change.
  - Admin can assist users with problem-solving, such as deleting tasks when a user cannot delete them.

- Data Storage and Security:

  - The system can accept and saving data from users.
  - Users can upload files for processing.
  - User data and files are securely stored in the database.
  - The website ensures data privacy and follows best practices for data protection.

- Usability:

  - The website provides a user-friendly interface for easy navigation and task management.
  - Users can view and track their tasks effectively.
  - Announcements are available for users to communicate important information.

- System Performance:

  - The system should be able to handle concurrent user requests and maintain acceptable performance levels.
  - Load testing and performance optimization techniques should be implemented to ensure smooth operation during peak usage periods.

- Reliability and Error Handling:

  - The system should be always reliable and available for users.
  - Proper error handling and exceptional management should be implemented to handle any unforeseen errors or issues.

- Scalability:

  - The system should be designed to accommodate future growth and additional features.
  - The architecture should support scalability to handle increasing user and data load.

- Accessibility:

  - The website should be designed to be accessible to users with disabilities, following relevant accessibility guidelines.

- Documentation and Support:

  - Comprehensive documentation should be provided to guide users on how to use the system effectively.
  - Adequate support channels should be available for users to seek assistance or report issues.

3.2.2 Non-Functional requirements

- Security:

  - User authentication and authorization mechanisms should be in place to ensure only authorized users can access sensitive data and perform specific actions.
  - User passwords should be securely stored and transmitted.
  - The system should implement measures to protect against common security threats, such as cross-site scripting (XSS) and SQL injection attacks.
  - File uploads should be properly validated and scanned for potential security risks.

- Integration:

  - The system should be able to integrate with other existing systems or services, such as email notifications or external databases if required.
  - APIs or web services should be provided to allow seamless integration with external applications.

- Compatibility:

  - The system should be compatible with a wide range of web browsers and devices to ensure a consistent user experience.
  - It should support different operating systems and screen sizes.

- Maintainability:

  - The system's codebase should be well-structured, modular, and maintainable, following best practices and coding standards.
  - Documentation should be provided to assist developers in understanding the system architecture, code, and configuration.
  - Updates and bug fixes should be easy to deploy without causing disruptions to the system.

- Compliance:

    - The system should comply with relevant legal and regulatory requirements, such as data protection and privacy laws.

It is essential to evaluate the non-functional aspects of the Task Management System to ensure its reliability, security, performance, and usability meet the desired standards and provide a robust and efficient solution for the intended users.

3.3 Software Used

1. **Lucidchart** - a web-based platform that allows users to collaborate on drawing, revising, and sharing charts and diagrams.
2. **Draw.io -** a web-based platform that allows users to collaborate on drawing, revising, and sharing charts and diagrams.
3. **Figma** - an open-source Arduino Software (IDE) makes it easy to write codeand upload it to the board.
4. **GitHub -** a provider of Internet hosting for software development and version control using Git.
5. **GIT -** Git is software for tracking changes in any set of files usually used forcoordinating work among programmers.
6. **Visual Studio Code -** a source-code editor made by Microsoft for Windows,Linux and macOS
7. **Atlassian -** Atlassian's team collaboration software like Jira, Confluence andTrello help teams organize, discuss, and complete shared work**.**

3.4 Conclusion

This chapter presents the methodology used in the development of the Task Management System. It outlines the requirements, both functional and non-functional, that guided the system's design and implementation. Additionally, it highlights the software tools and technologies utilized throughout the development process.

CHAPTER 4: System Analysis

Chapter 4: System Analysis

4.1 System Model

4.1.1 Data Flow Diagram (DFD)

Data Flow Diagram is a visual representation of how data moves within a system, using processes, data flows, data stores, and external entities. It helps analyze system requirements and identify data dependencies.

4.1.1.1 Context Diagram

Level 0 data flow diagrams, also known as context diagrams. show a single process node and its connections to external entities. The Diagram shown below illustrates the process with the flow of information between the Dean, Vice, head of department, doctors, assistant, and Secretary.

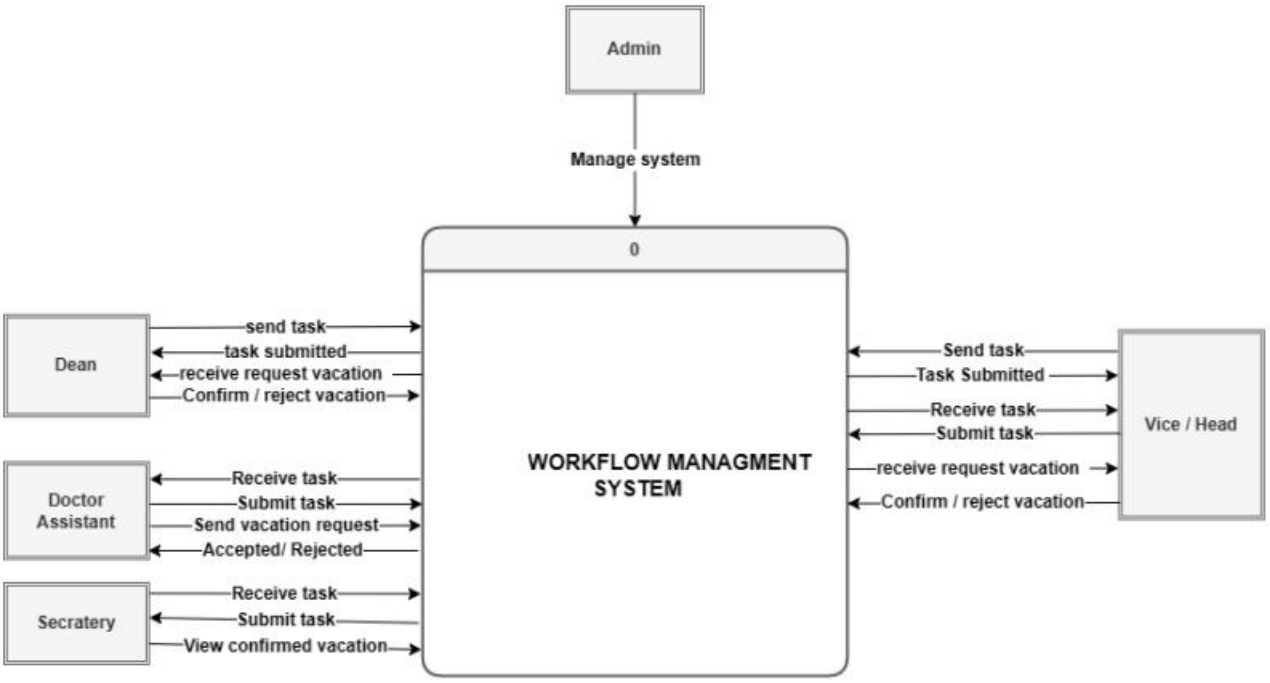


Figure 1 DFD (Level 0)

4.1.1.2 Level 1 Data flow diagram

Level 1 DFDs the single process node from the context diagram is broken down into sub-processes.

The Diagram shown below includes adding the additional task, receiving a task, submitting a task, vacation request, vacation response, managing the system, Viewing tasks of all users, Login processes to the workflow system, as well as data stores.

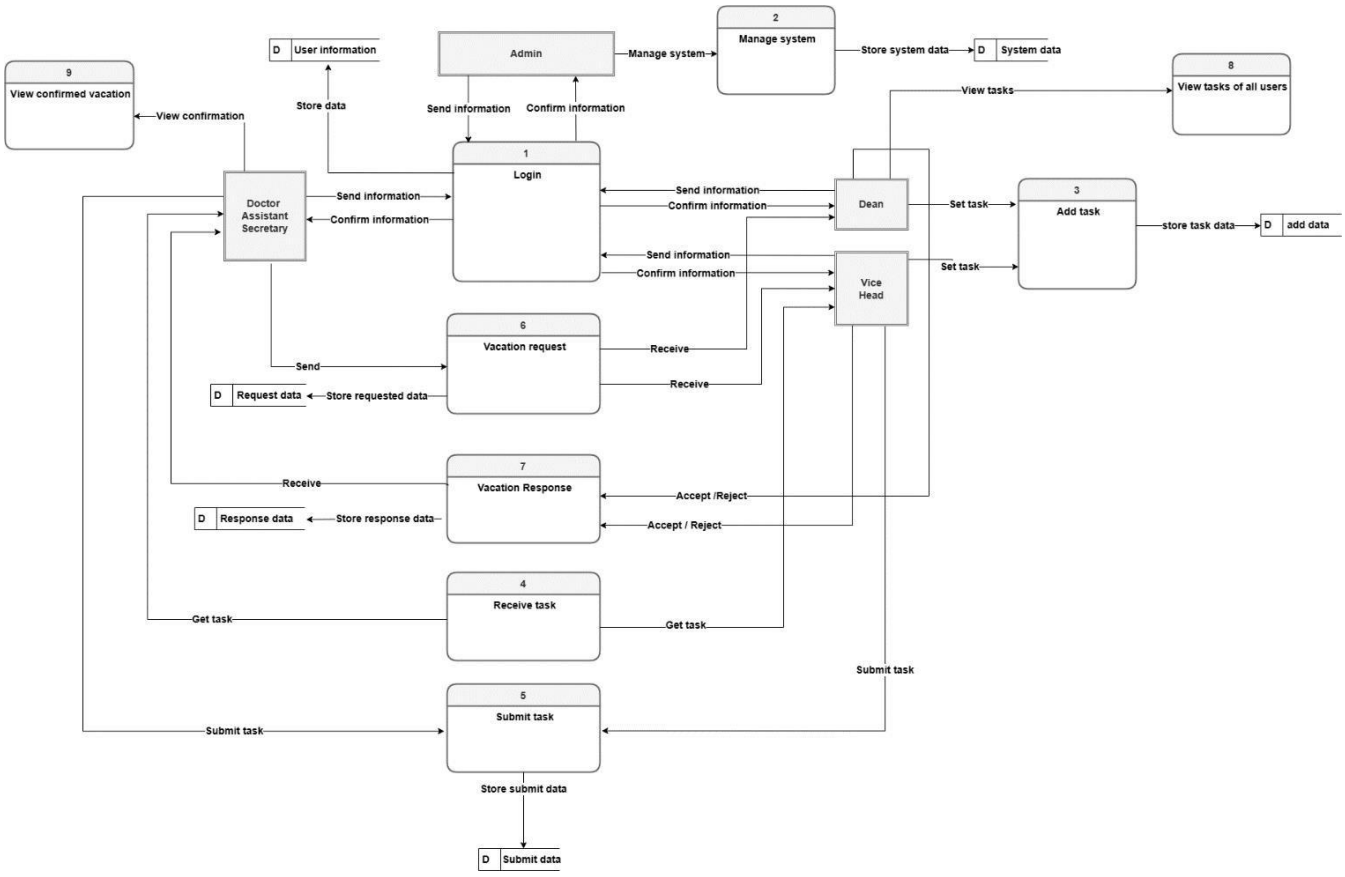


Figure 2 DFD (Level 1)

4.1.1.3 Level 2 Data flow diagram

Level 2+ DFDs simply break processes down into more detailed sub-processes.

The level 2 diagram below expands on the add task process to include more granular processes involved, such as setting deadlines, setting receivers, editing a task, and deleting task-connected data flows.

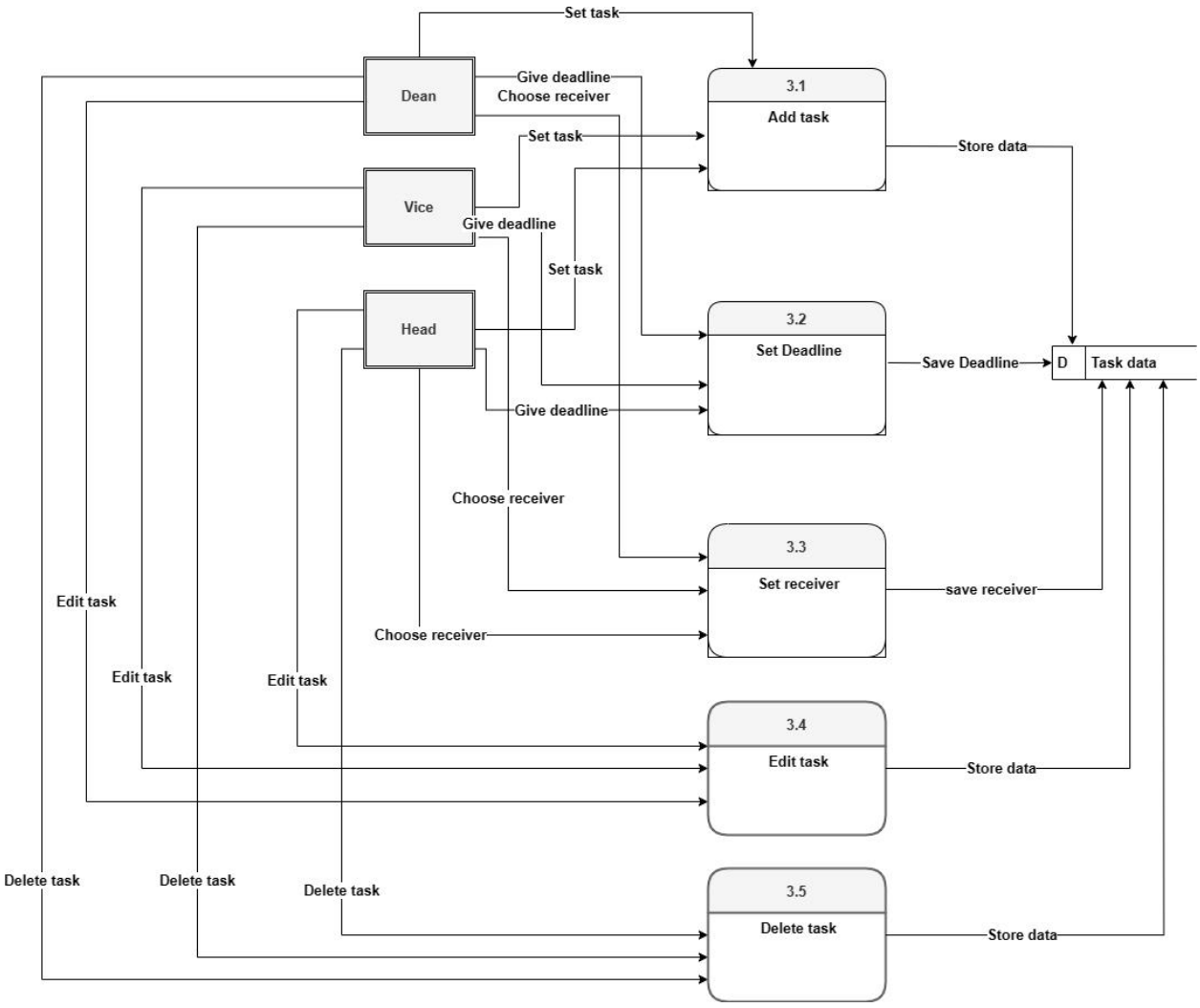
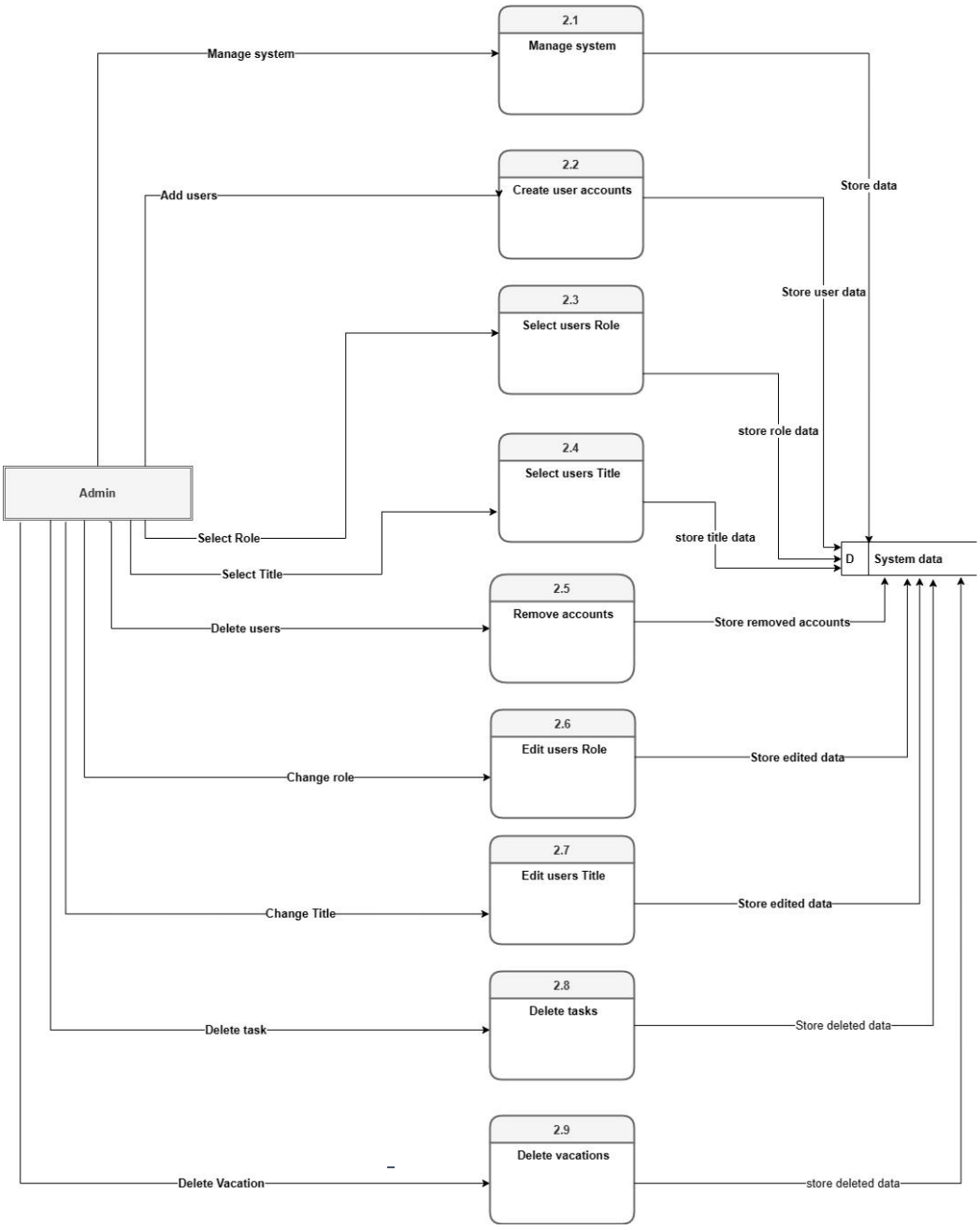


Figure 3 DFD (Level 2)

4.1.1.4 Level 3 Data flow diagram

Level 3 data flow diagrams are detailed enough that expands on the management system process to include more granular processes involved, such as creating user accounts, selecting users' levels, removing accounts, and changing user accounts connected data flows.



4.2 Interaction Perspective

4.2.1 Use-Case Diagram

Use case diagrams can summarize the details of your system's users and their interactions with the system.

The Diagram shown below illustrates Actors such as admin, dean, vice, head of the department, doctors, assistant, and secretary, also shows Use cases and Associations.

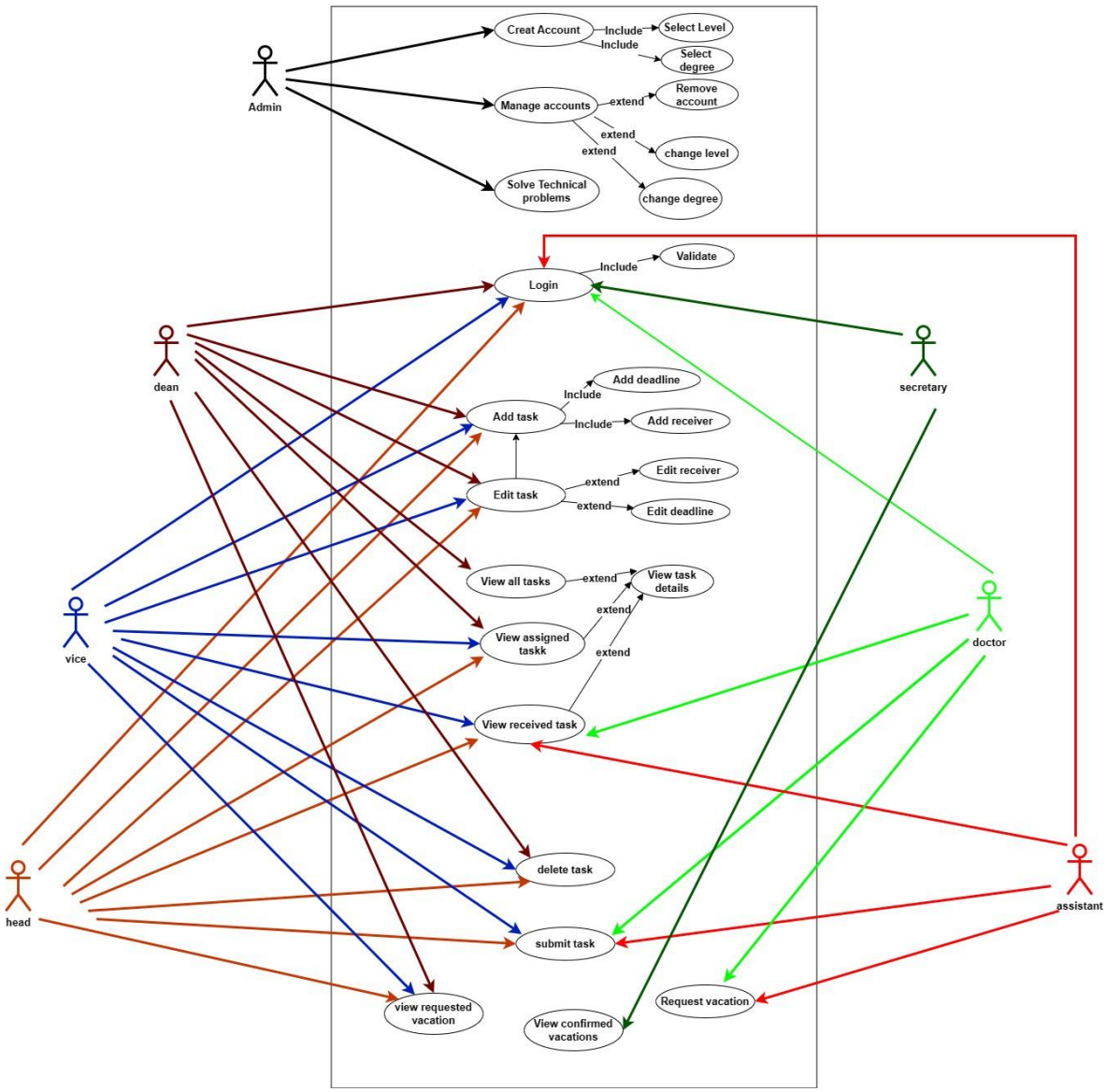


Figure 5 Use Case

4.2.2 Sequence Diagram

A sequence diagram is a type of interaction diagram because it describes how—and in what order—a group of objects works together.

4.2.2.1 Admin

The Diagram shown below illustrates the actor is the admin, and Lifeline is the user interface, system, and database. Also shows the Synchronous message and Asynchronous return message.

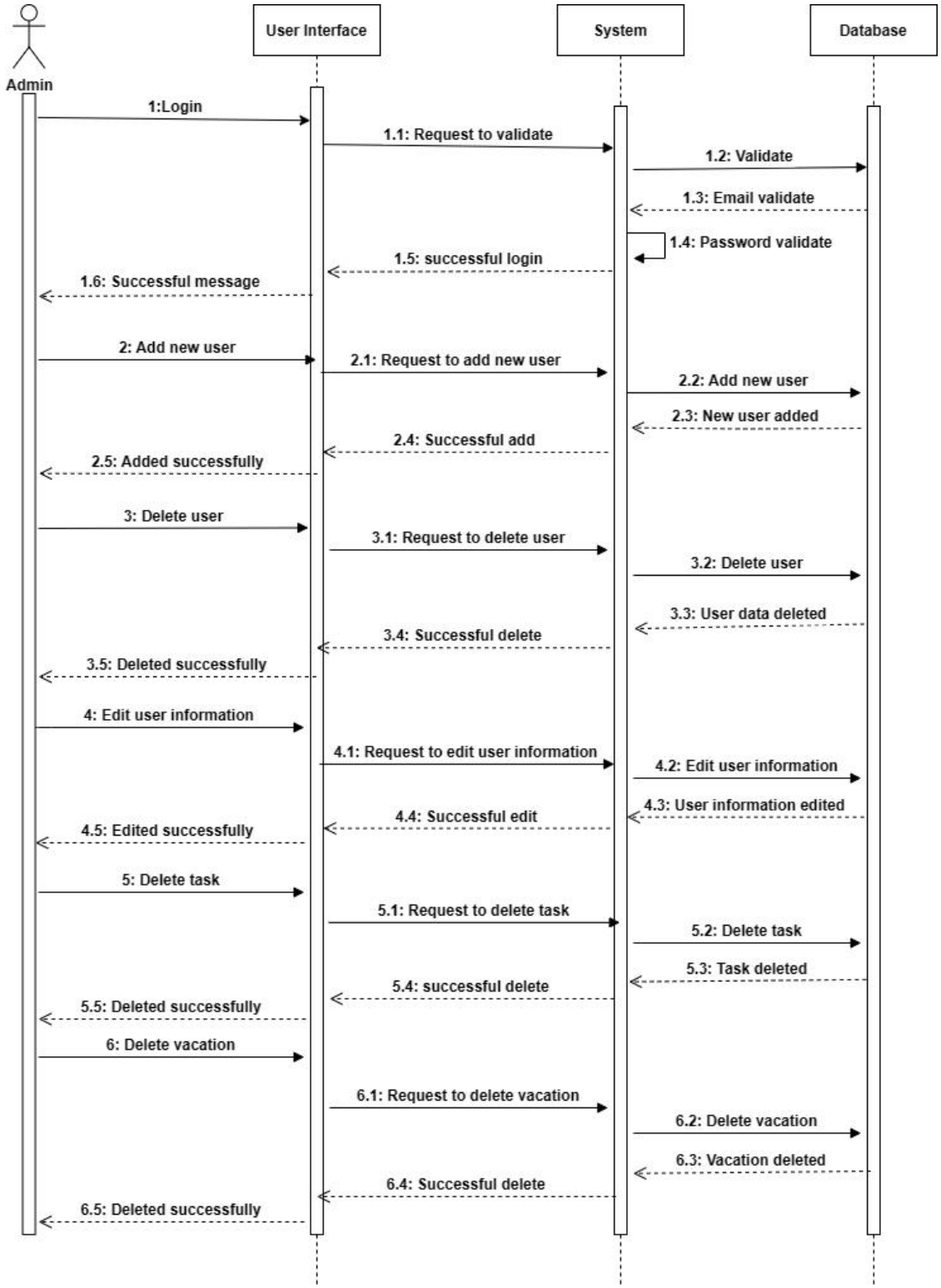


Figure 6 Sequence (Admin)

4.2.2.2 Dean

The Diagram below illustrates actor is the dean, and Lifeline is the user interface, system, and database. Also shows the Synchronous message and Asynchronous return message.

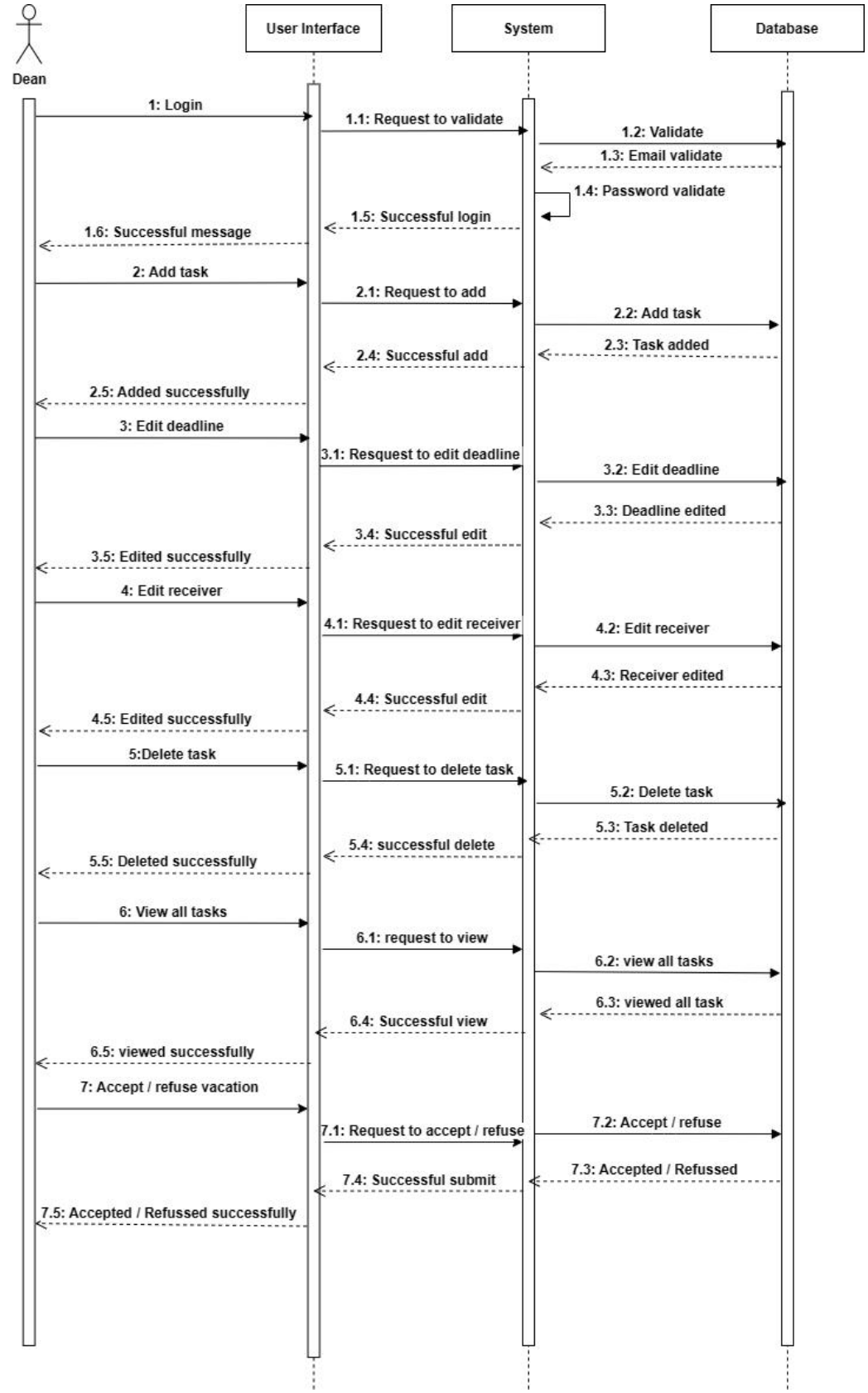


Figure 7 Sequence (Dean)

4.2.2.3 Vice & Head of Department sequence

The Diagram shown below illustrates actor is the vice & head of the department,and Lifeline is the user interface, system, and database. Also shows the Synchronous message and Asynchronous return message.

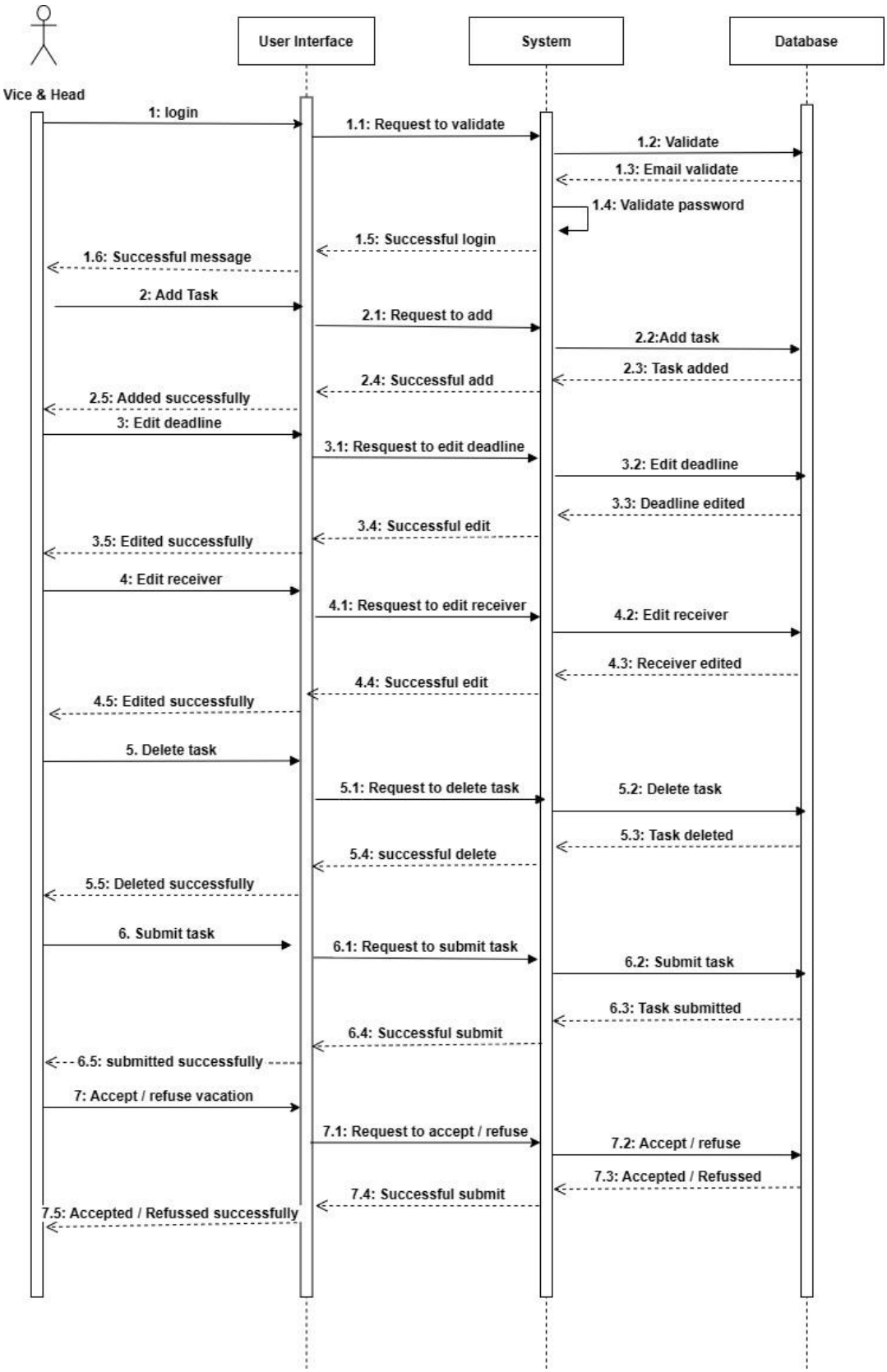


Figure 8 Sequence (Vice / Head)

4.2.2.4 Doctor & Assistant

The Diagram shown below illustrates actor is a doctor & assistant, and Lifeline is the user interface, system, and database. Also shows the Synchronous message and Asynchronous return message.

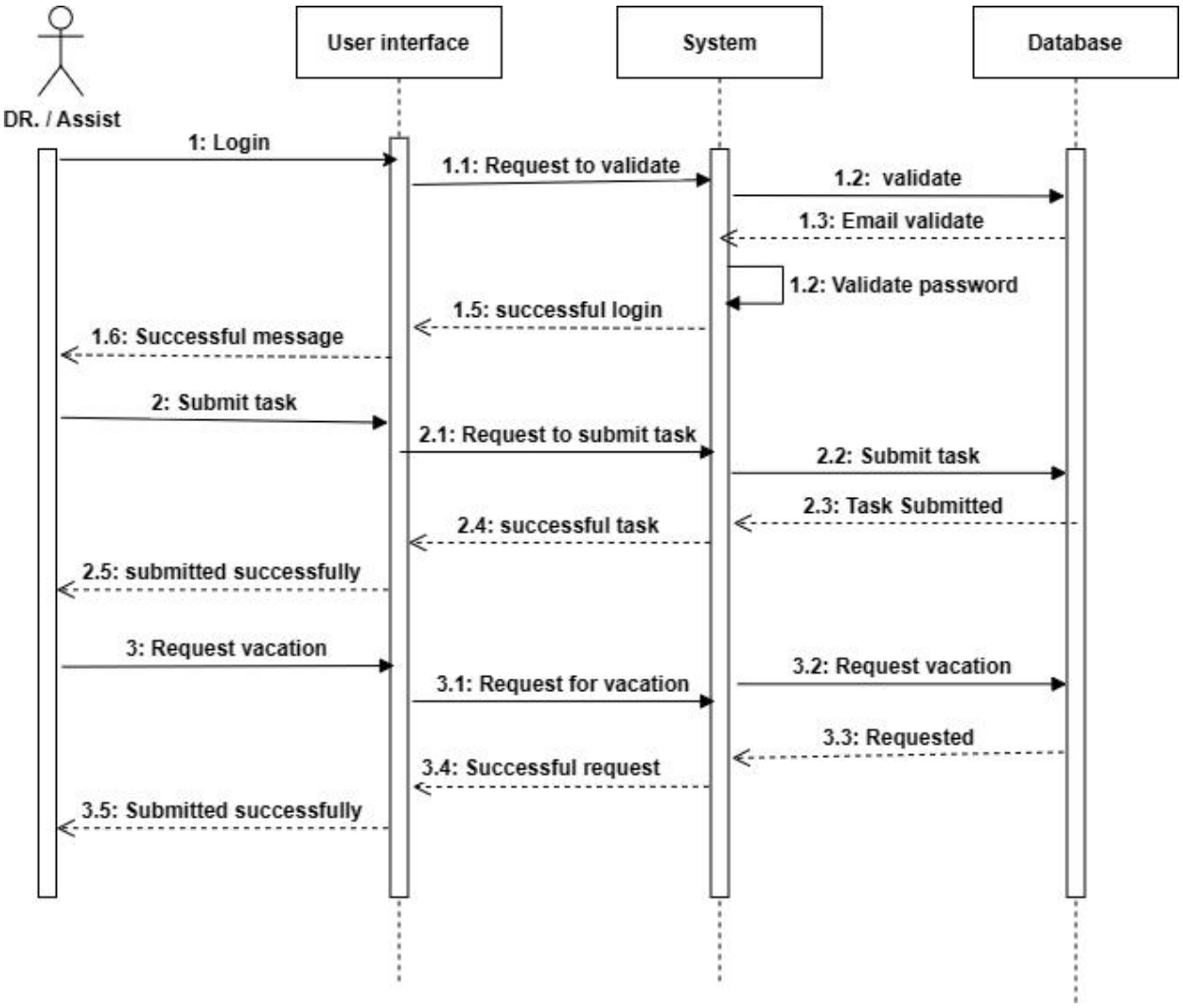


Figure 9 Sequence (Doctor / Assistant)

4.2.2.5 Secretary

The Diagram shown below illustrates actor is the secretary, and Lifeline is the user interface, system, and database. Also shows the Synchronous message and Asynchronous return message.

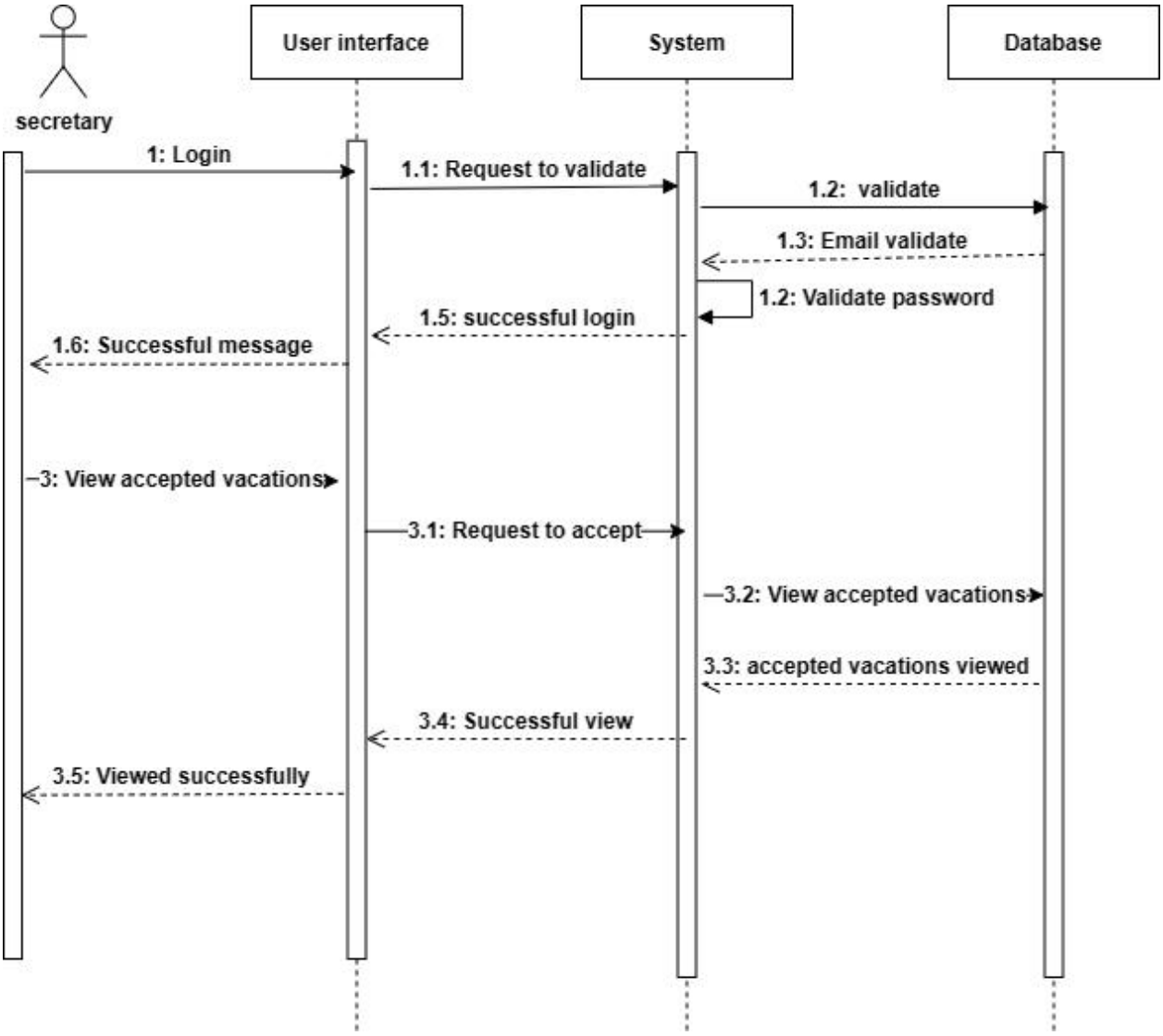


Figure 10 Sequence (Secretary)

4.3 Structural Perspective

4.3.1 Class Diagram

A class diagram is a visual representation in UML (Unified Modeling Language) that illustrates the structure of a system by showing the classes, their attributes, methods, and relationships. It provides a high-level overview of the entities and their interactions within the system. Class diagrams are used to design, analyze, and document object-oriented systems, helping to understand the structure, behavior, and relationships between classes.

The Diagram shown below illustrates:

- **Classes:** *A class represents an object or a set of objects that share a common structure and behavior. Shows Their names such as Admin, Dean, Vice/Head of the department, Doctor/Assistant, secretary, Vacation, and Task. Also shows Attributes and Methods.*
- **Inheritance/generalization:** *The process of a child or sub-class taking onthe functionality of a parent or superclass.*

  - Such as the class Doctor/Vice/Head would inherits the class users
  - The class Doctor/Assistant would inherits the class Doctor/Assistant.
  - The object Users would inherit all classes of Admin, Doctor/Vice/Head, and Secretary/Doctor/Assistant.

A class diagram is a visual representation in UML (Unified Modeling Language) that illustrates the structure of a system by showing the classes, their attributes, methods, and relationships. It provides a high-level overview of the entities and their interactions within the system. Class diagrams are used to design, analyze, and document object-oriented systems, helping to understand the structure, behavior, and relationships between classes.

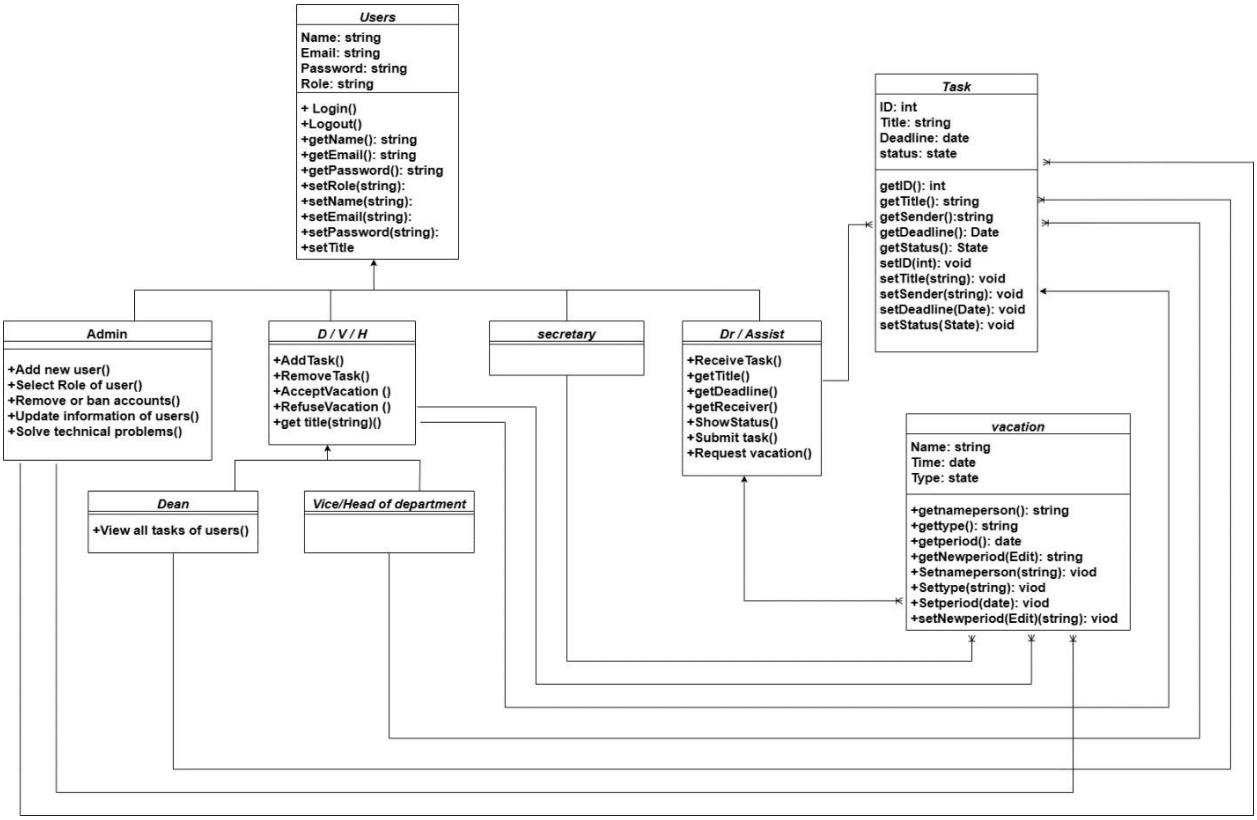


Figure 11 Class diagram

4.4 Behavioral Perspective

4.4.1 Activity Diagram

Is a type of diagram in UML (Unified Modeling Language) that depicts the flow ofactivities and actions within a system or process. It shows the sequence of tasks, decision points, and parallel activities involved in a process.

4.4.1.1 Admin

The Diagram shown below illustrates the start node, Activity indicates the activities that Admin makes, and the Connector symbol Shows the directional flow, of the activity than the end node.

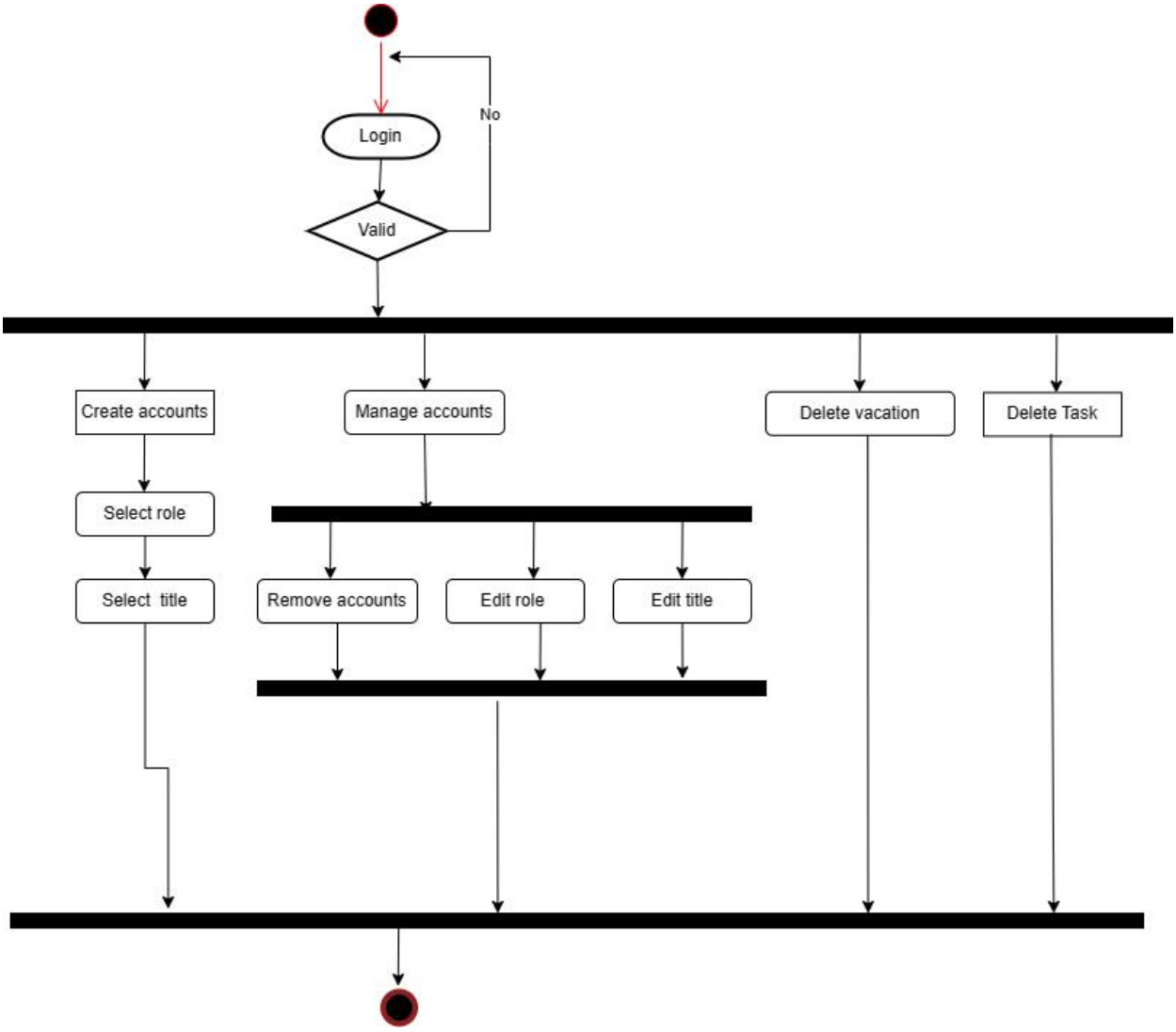


Figure 12 Activity (Admin)

4.4.1.2 Dean

The Diagram shown below illustrates the start node, Activity indicates the activities that Dean makes, and the Connector symbol Shows the directional flow,of the activity and then the end node.

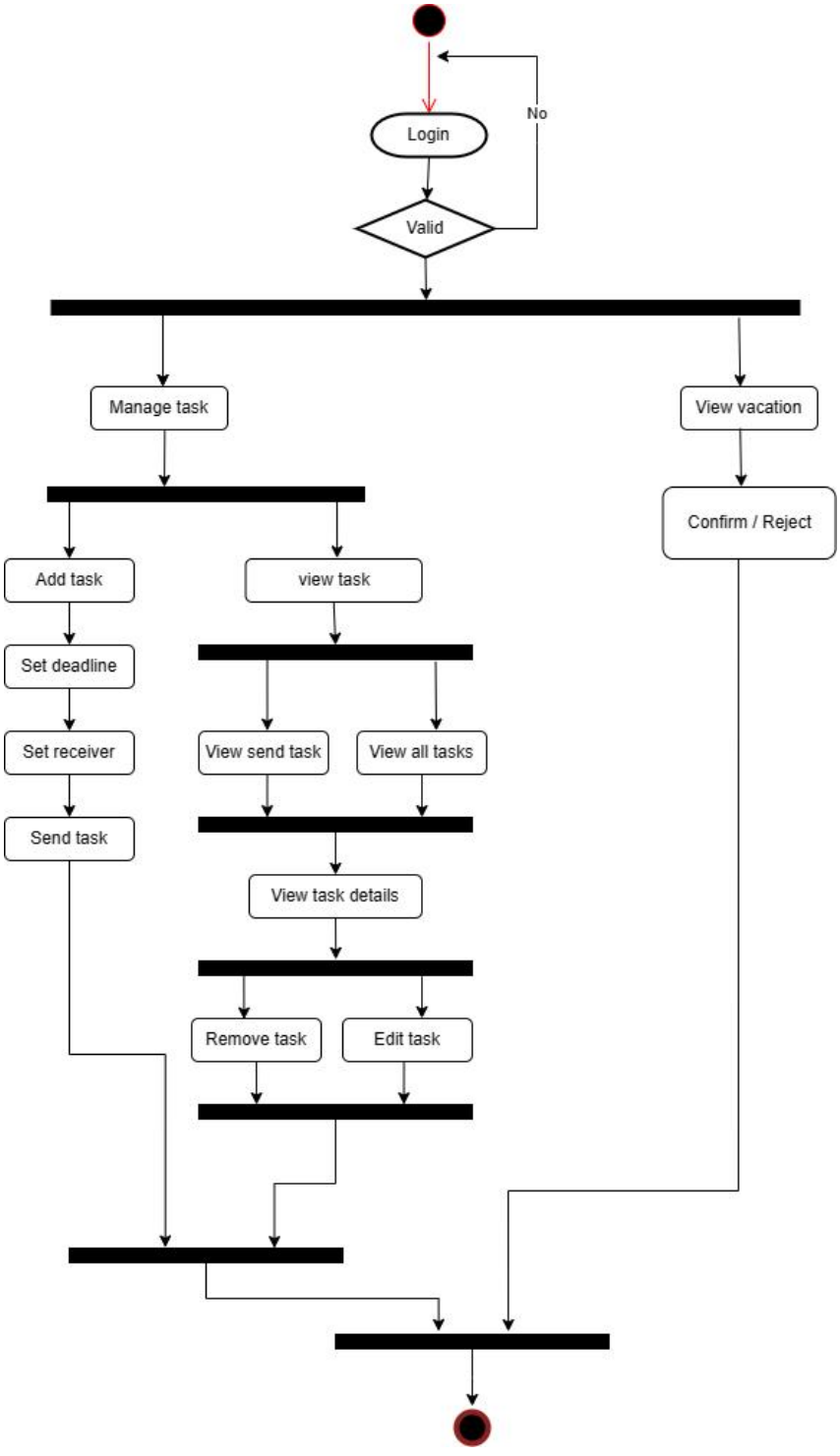


Figure 13 Activity (Dean)

4.4.1.3 Vice & Head of Department Sequence

The Diagram shown below illustrates the start node, the Activity indicates the activities that the Vice and head of the department make, and the Connector symbol Shows the directional flow, of the activity and then the end node.

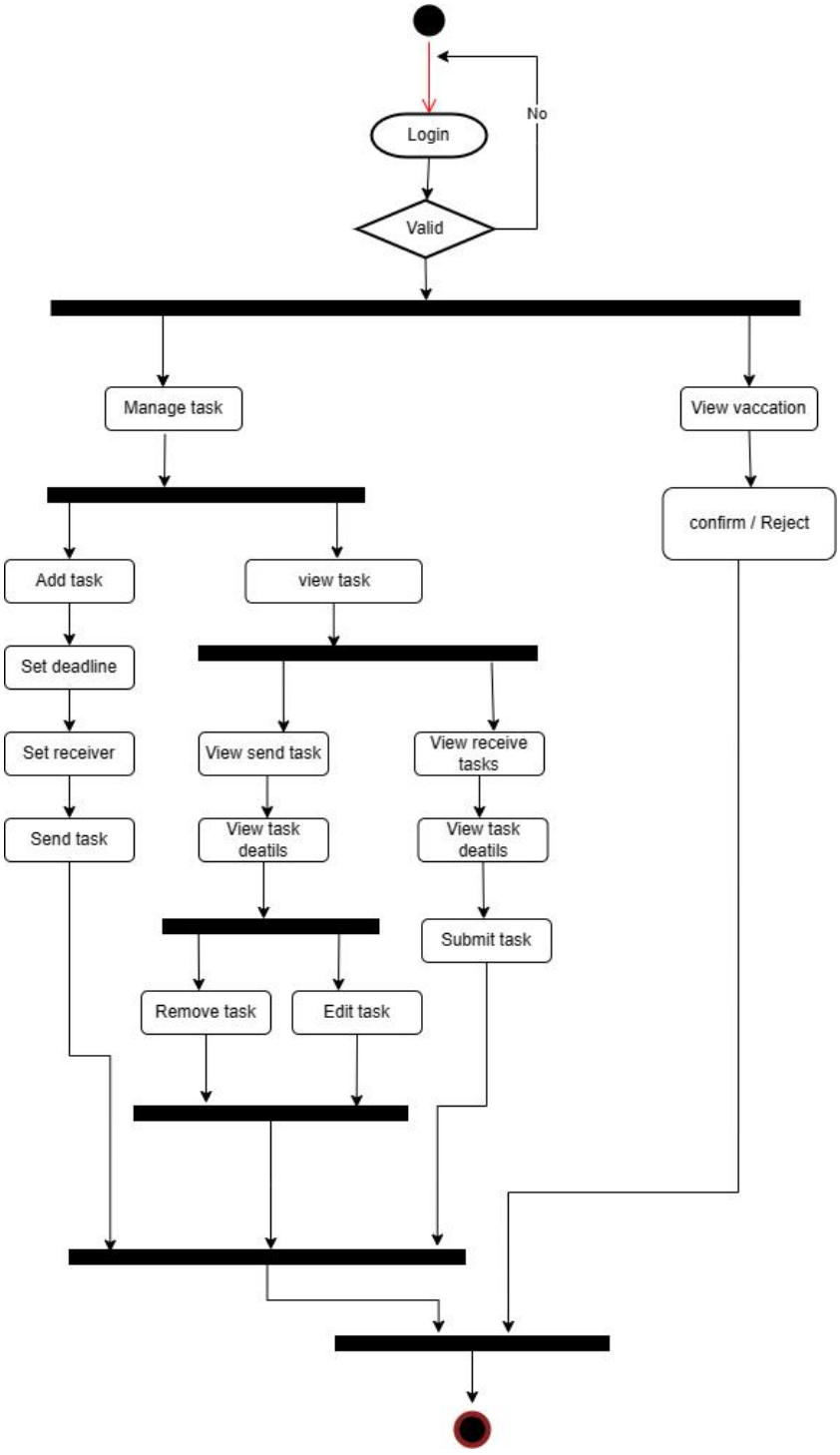


Figure 14 Activity (vice&head)

4.4.1.4 Doctor & Assistant

The Diagram shown below illustrates the start node, Activity indicates the activities that Doctor and Assistant make, and the Connector symbol Shows the directional flow, of the activity and then the end node.

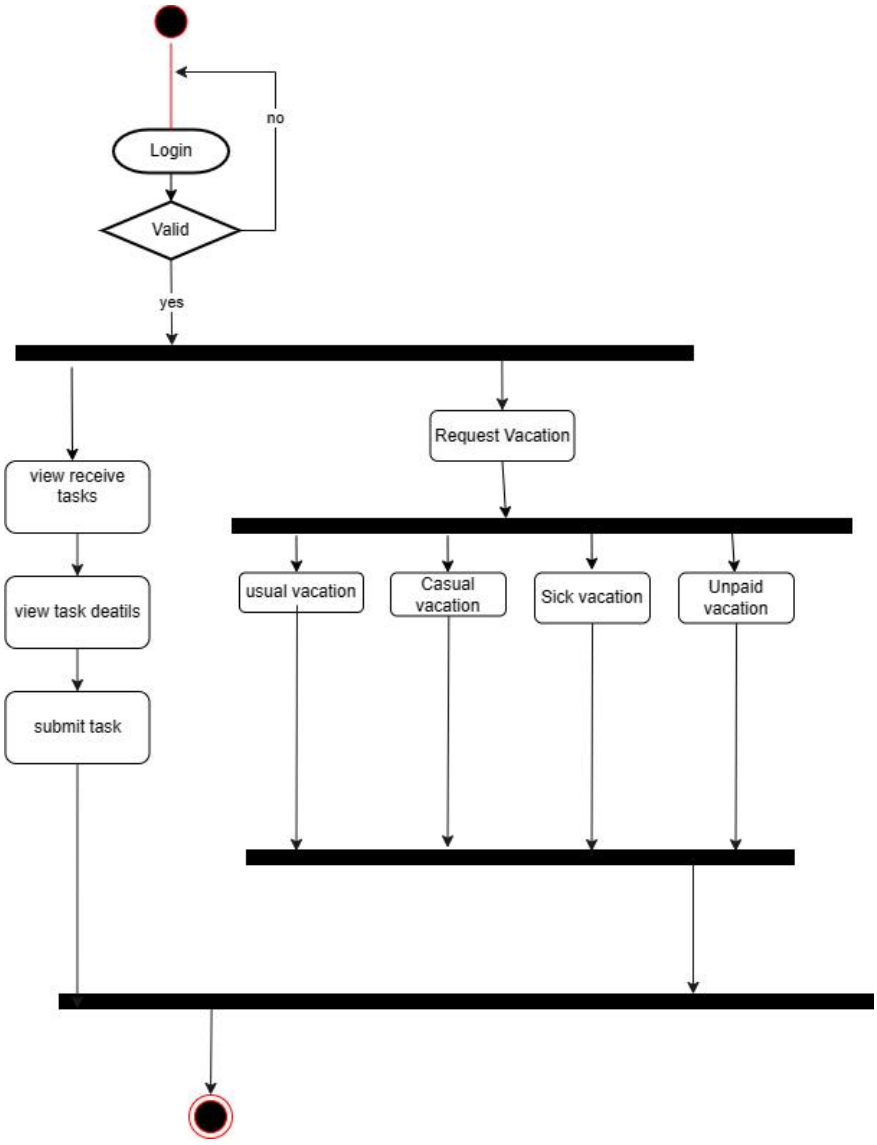


Figure 15 Activity (Doctor / Assistant)

4.4.1.5 Secretary

The Diagram shown below illustrates the start node, Activity indicates the activities that Secretary makes, and the Connector symbol Shows the directional flow, of the activity than the end node.

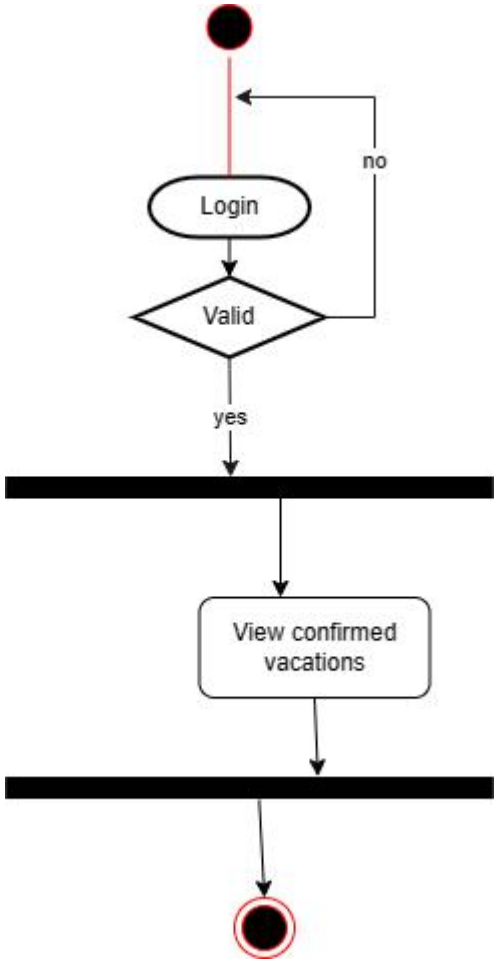


Figure 16 Activity(Secretary)

4.5 Entity Relationship Diagram (ERD)

An ER (Entity-Relationship) diagram is a visual representation that illustrates the structure of a database system. It shows the relationships between entities, the attributes associated with them, and how they interact. ER diagrams are used in database design and software development to depict the logical structure of a system and ensure accurate data representation.

The Diagram shown below illustrates entity a definable thing such as Admin, Dean, Vice, head of the department, doctors, assistant, Secretary, Task, and vacation. Also shows their attributes and the relationship between them.

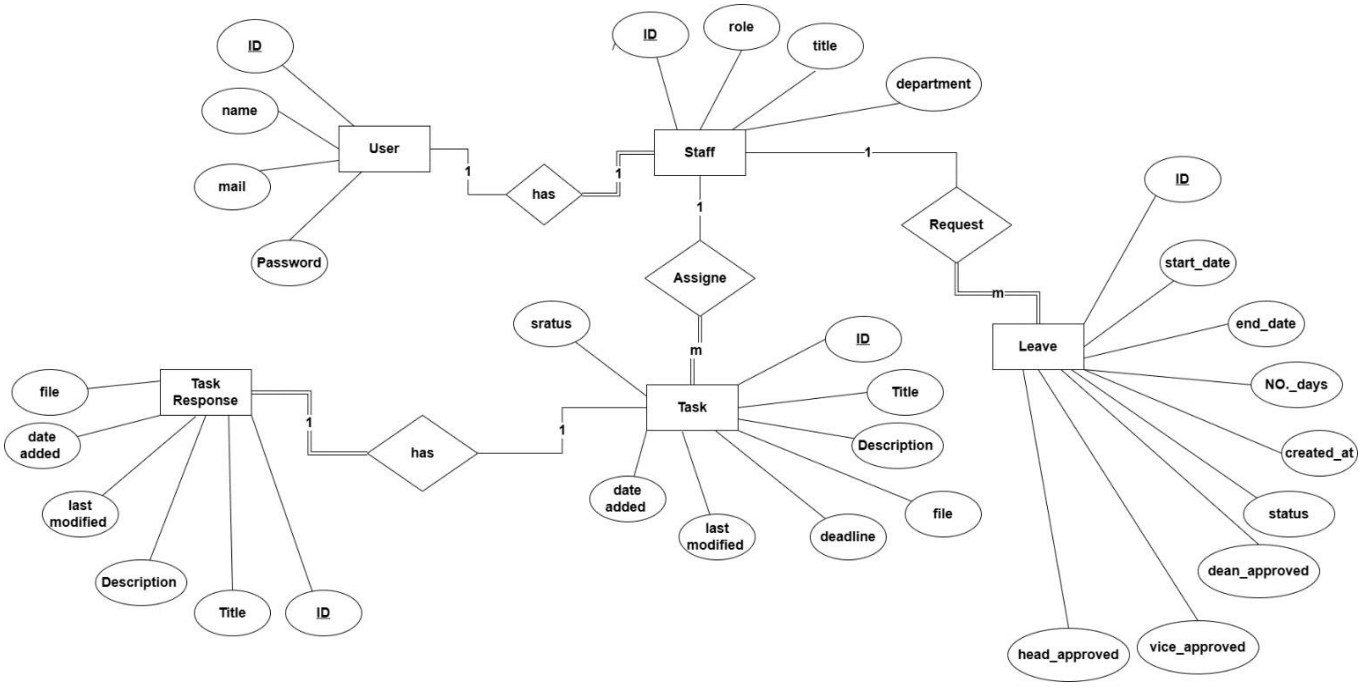


Figure 17 ERD

4.5.1 Relational Data Model (RDM)

The Relational Data Model (RDM) is a framework for organizing and representing data in a relational database management system (DBMS). It organizes data intotables composed of rows and columns.

The Relational Data Model (RDM) is a framework for organizing and representing data in a relational database management system (DBMS). It organizes data intotables composed of rows and columns.

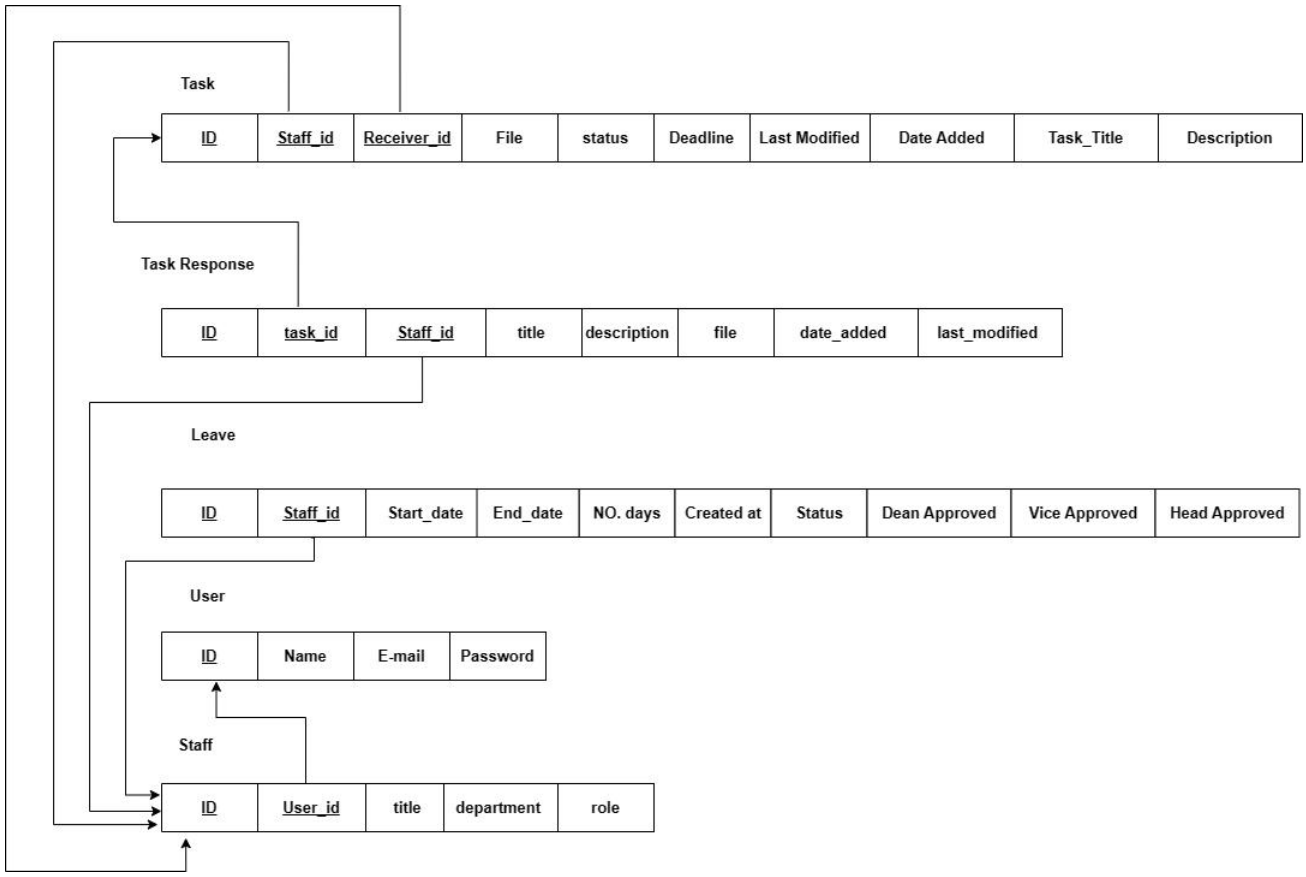


Figure 18 RDM

4.6 System Design

4.6.1 Scheme

A database schema defines the structure and organization of a relational database.It includes table definitions, relationships, constraints, and security permissions. It serves as a blueprint for creating and managing the database.

A database schema defines the structure and organization of a relational database.It includes table definitions, relationships, constraints, and security permissions. It serves as a blueprint for creating and managing the database.

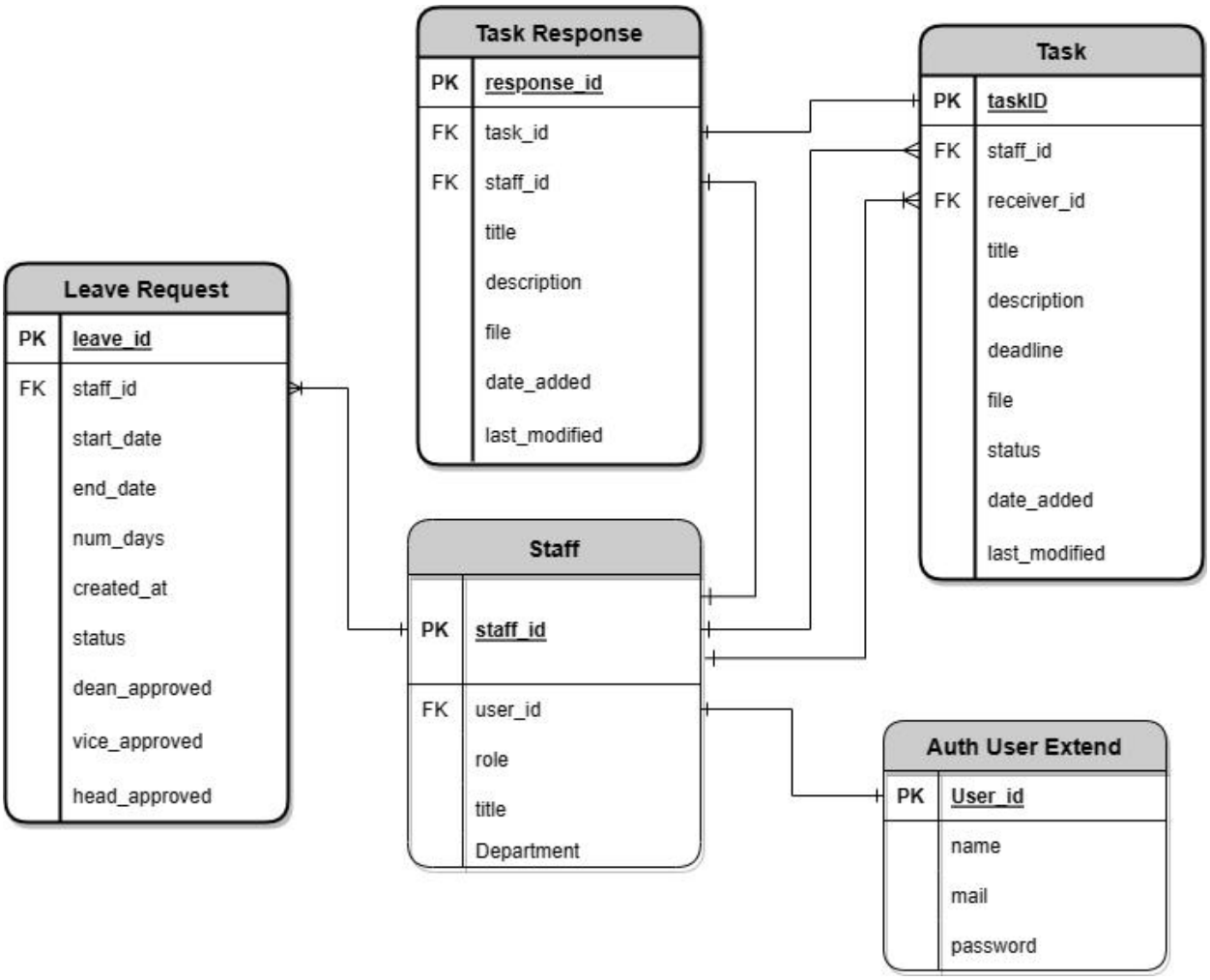


Figure 19 DB schema

CHAPTER 5: Implementation

Chapter 5: Implementation

5.1 The implementation phase is divided into 3 sub-phases:

1. Implementation of UI
2. Implementation of frontend
3. Implementation of backend

Each phase works parallel with the other phases except the parts depends on

previous work must be done first such as, coding of web site depends on UI (user Interface), developing the web site without implementation of APIs to connect the web site with the database and the server-side, sending data from site to Database.

At first front end of the web site pages are built, then we start to connect with the database and start retrieve, insert, update, and delete data from the databasethrough the created API. Finally, the implementation of the web site is developed.

Now we will explore the main functions of the application, whether the web and the implementation of the whole system.

5.2 User Interface (UI)

The user interface (UI) design requirements for a website differ from those for mobile devices and desktop computers. While mobile interfaces prioritize considerations such as smaller screen sizes and touch screen controls, web interfaces have their unique set of considerations to ensure usability, readability,and consistency across different devices.

- **Responsive Design**: Websites should be designed responsively to adapt to different screen sizes and resolutions. This ensures that the content is displayed appropriately on various devices, including desktops, laptops, tablets.

- **Consistency**: Consistency in UI elements, such as navigation menus, buttons, and typography, is crucial for a seamless user experience. Users should be able to easily recognize and understand the interface elements regardless of the page they are on within the website.

- **Visual Hierarchy**: Establishing a clear visual hierarchy helps users understand the importance and relationship between different elements on the webpage. Proper use of visual cues like size, color, and spacing can guide users' attention and make it easier to navigate and interact with the site.

5.2.1 login page

Figure 20 login page

5.2.2 Dean home page

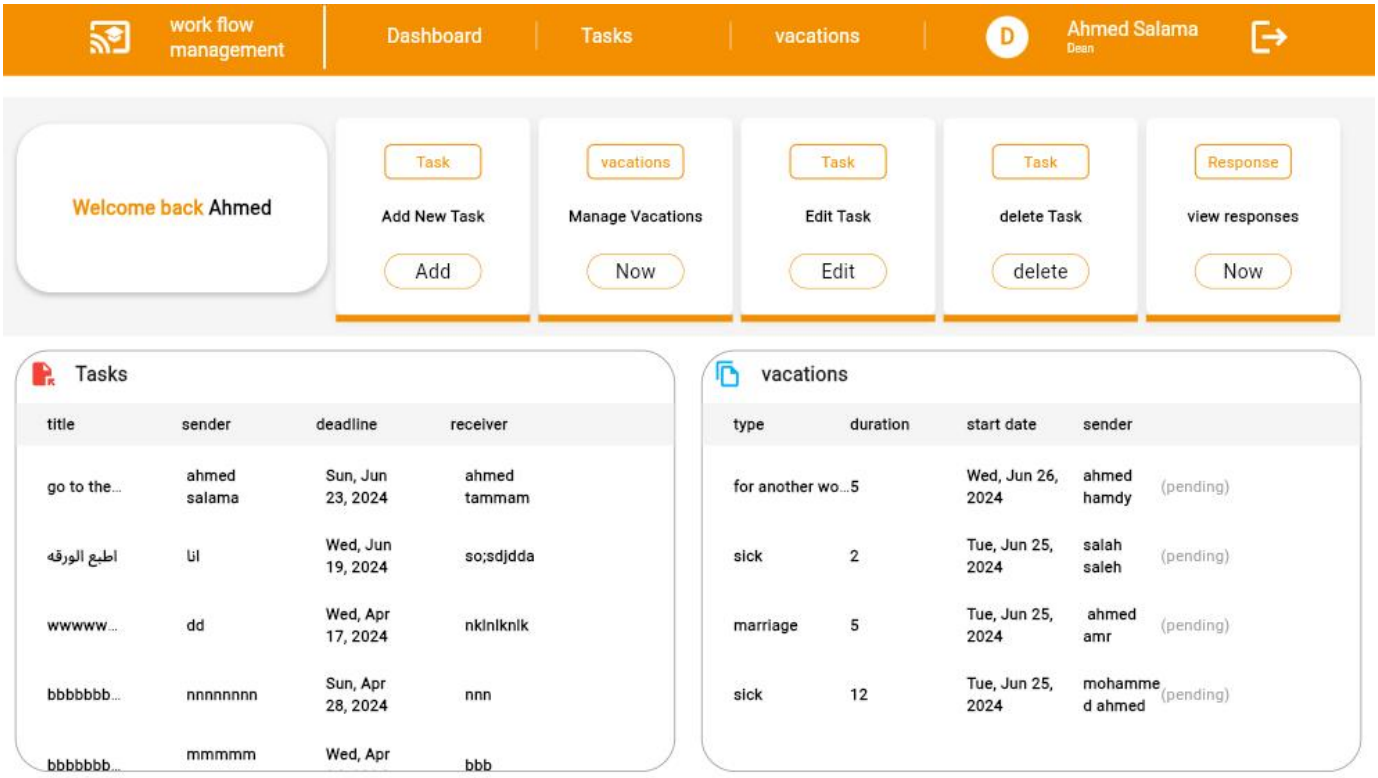


Figure 21 Dean home page

5.2.3 Vice, head home page

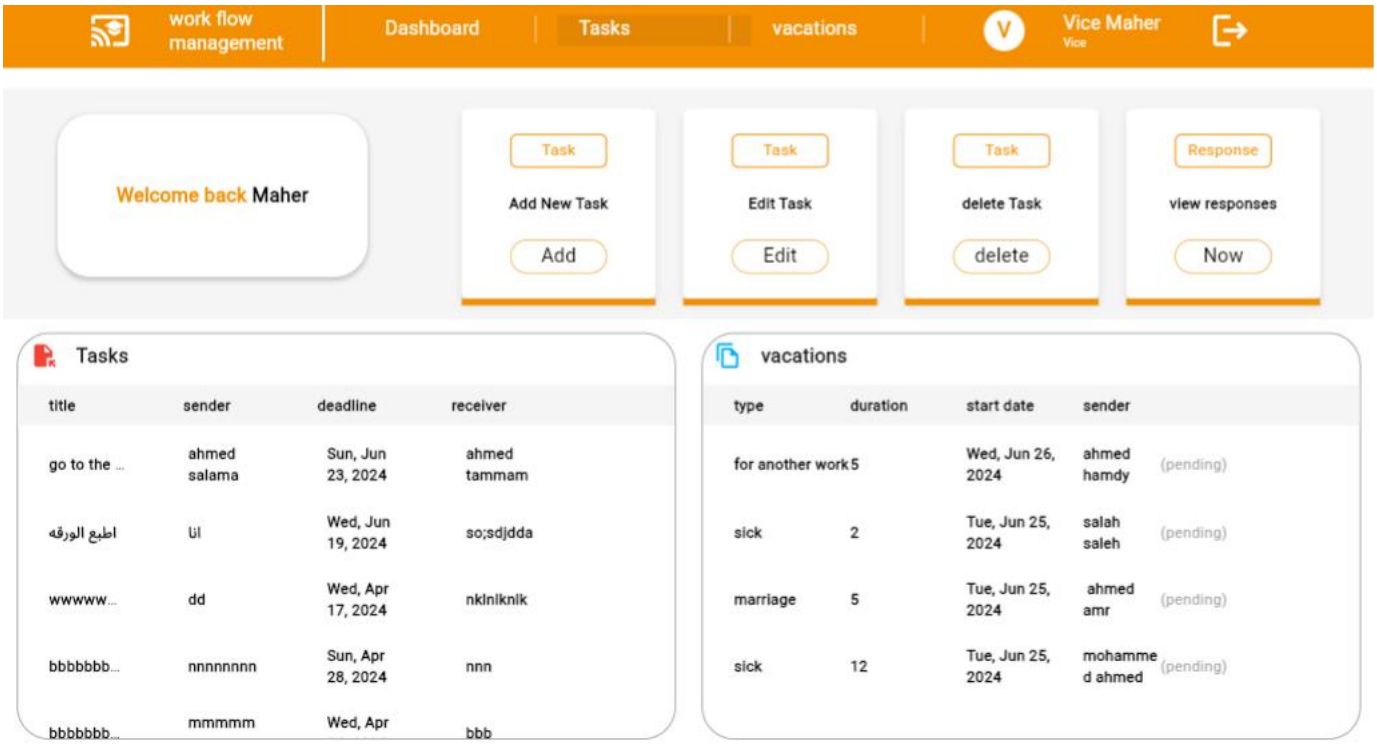


Figure 22 Vice, head home page

5.2.4 Doctor, Assistant home

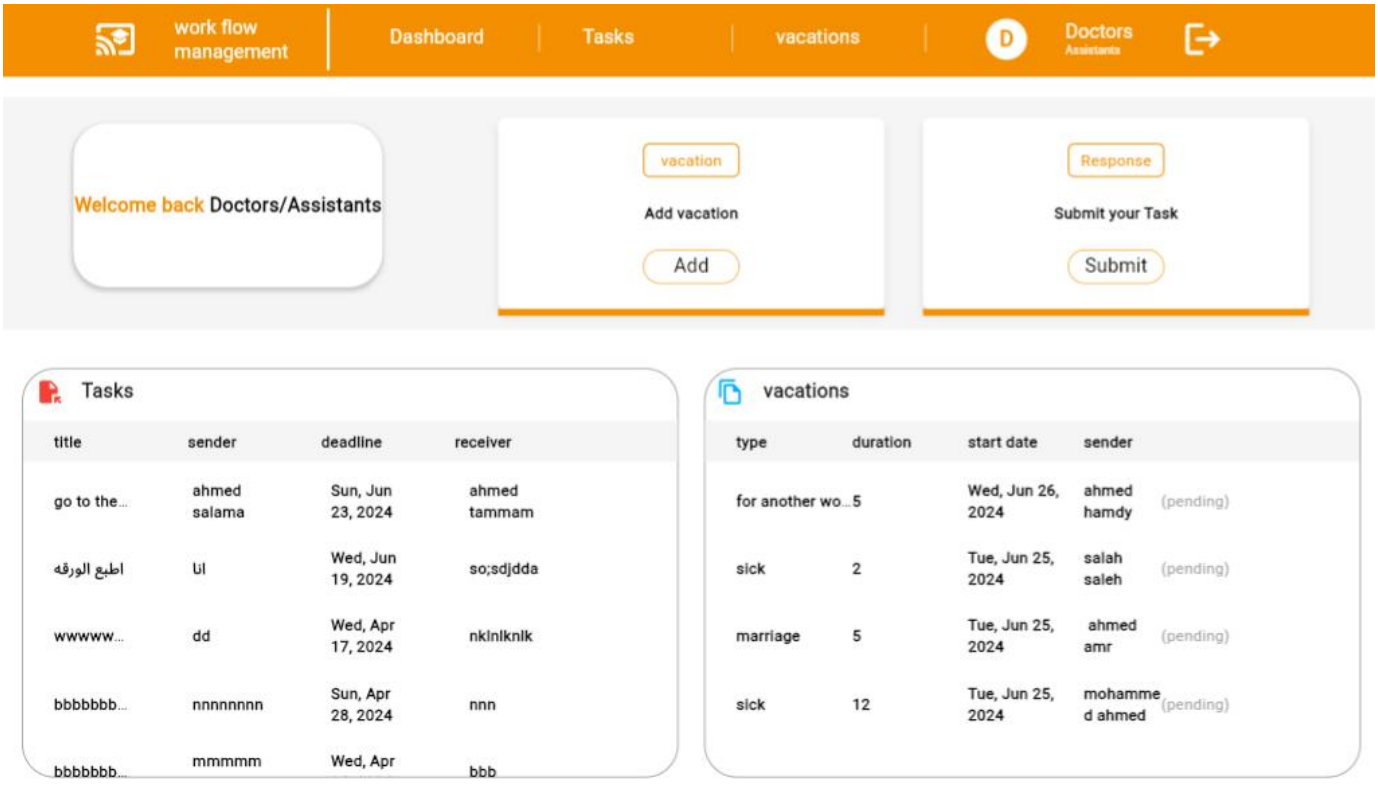


Figure 23 Doctor, Assistant home

5.2.5 Dean tasks

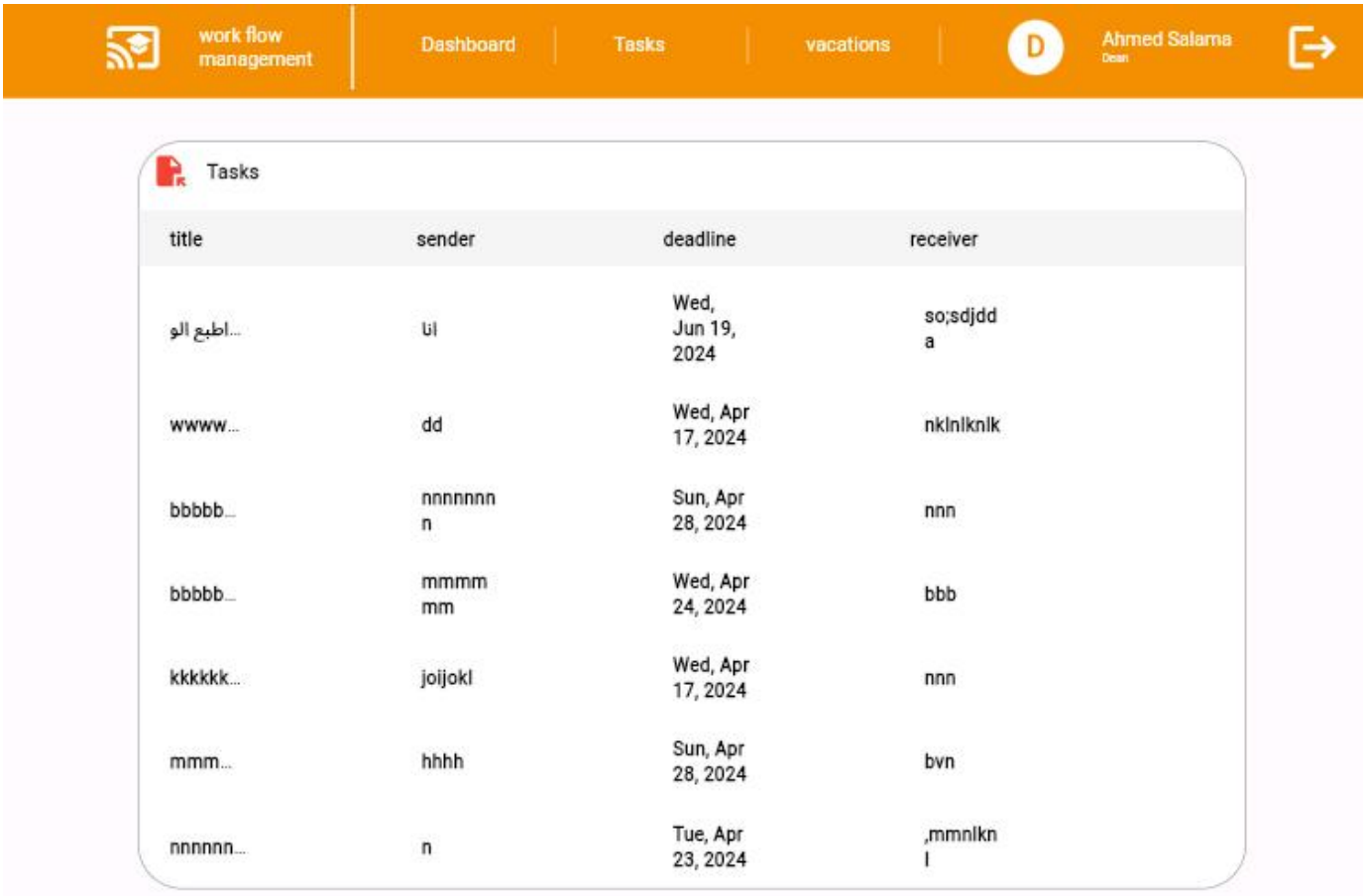


Figure 24 dean tasks

5.2.6 Head, vice tasks

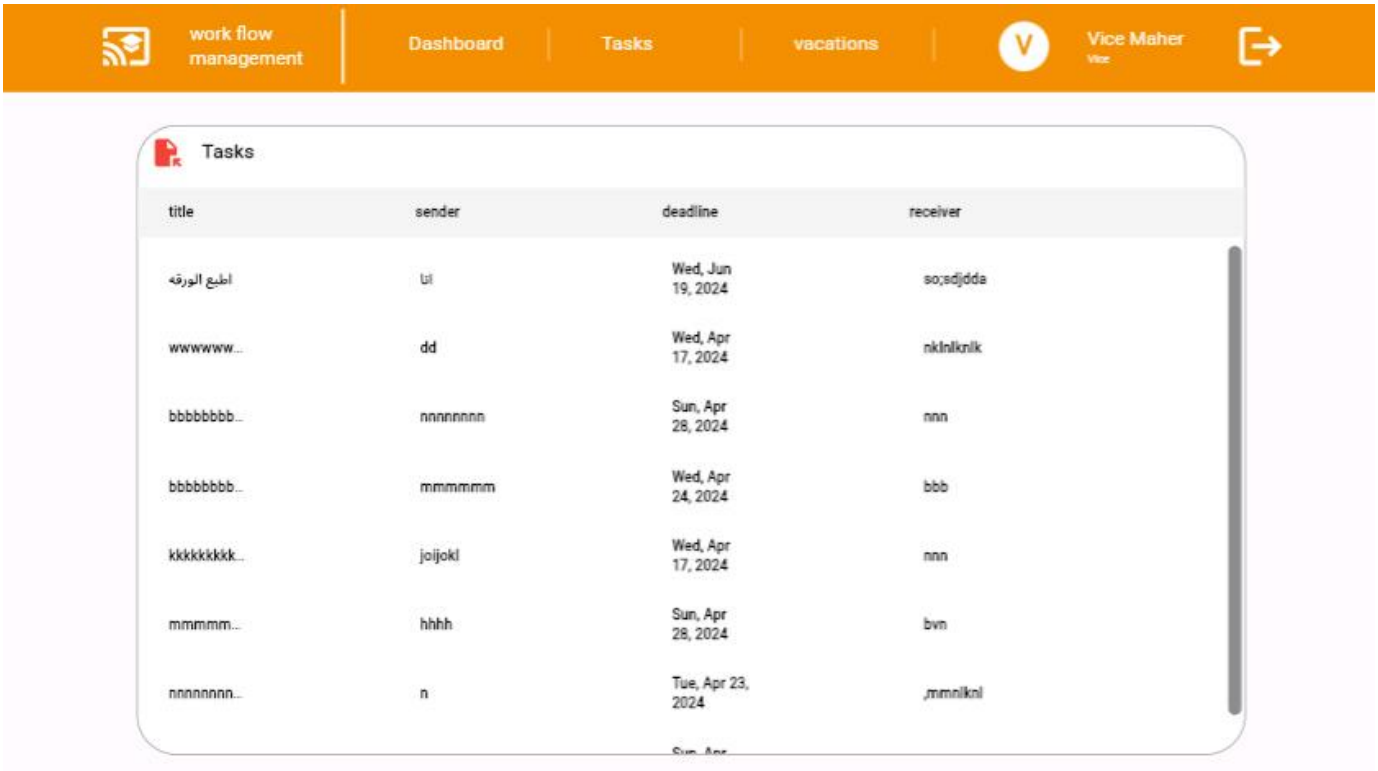


Figure 25 Head, vice tasks

5.2.7 Doctor, assistant tasks

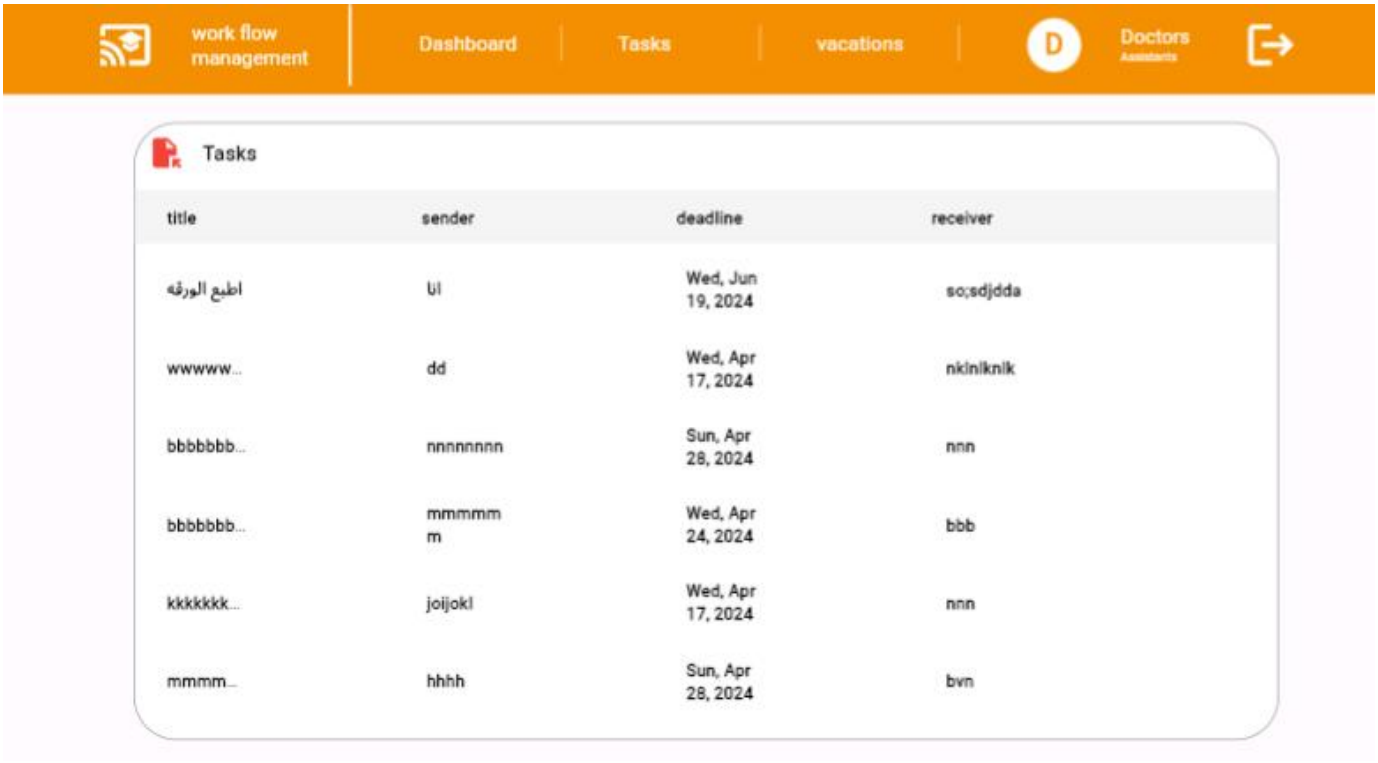


Figure 26 Doctor, Assistant tasks

5.2.8 Add task

Figure 27 Add task

5.2.9 Add response

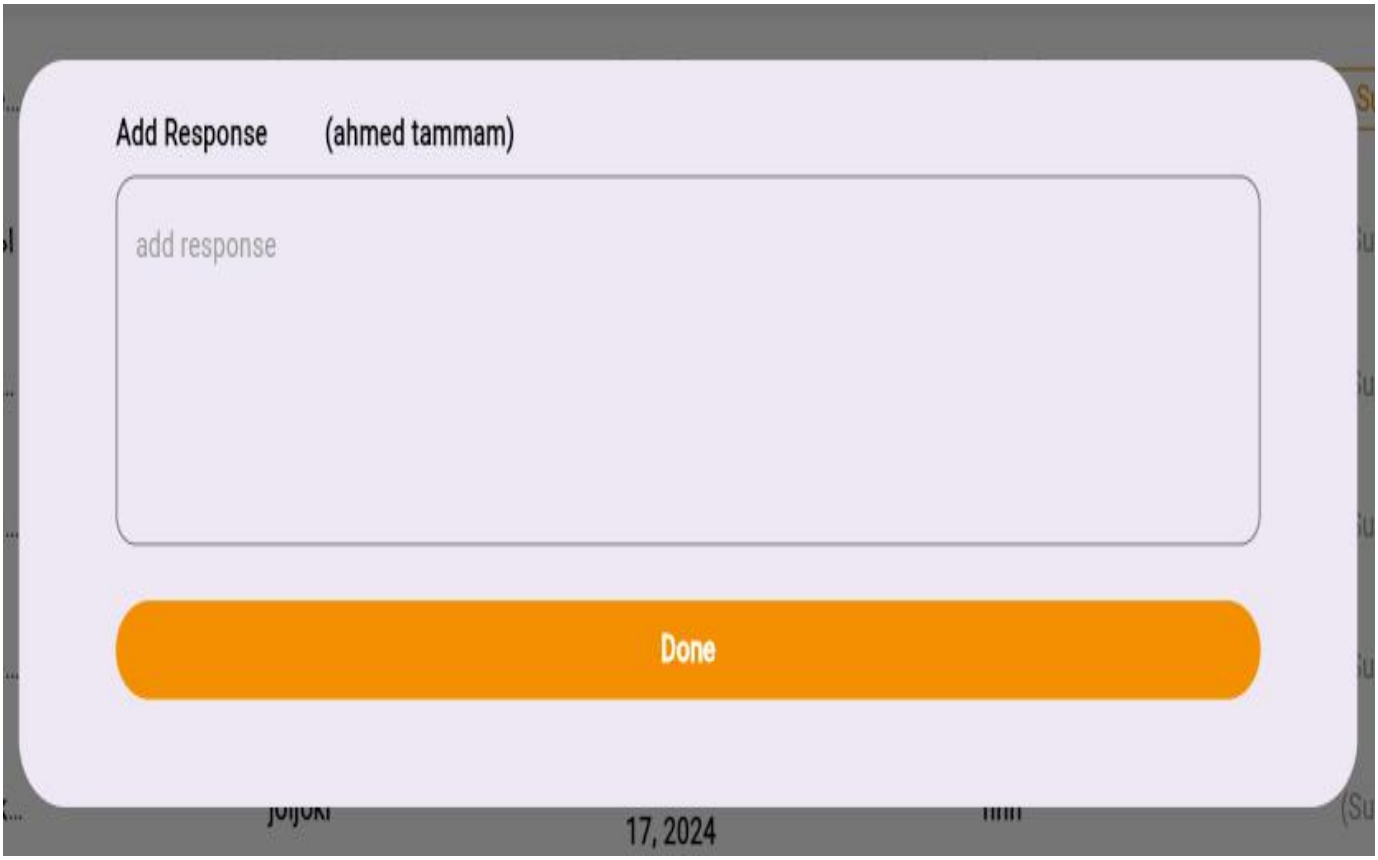


Figure 28 Add response

5.2.10 Task details

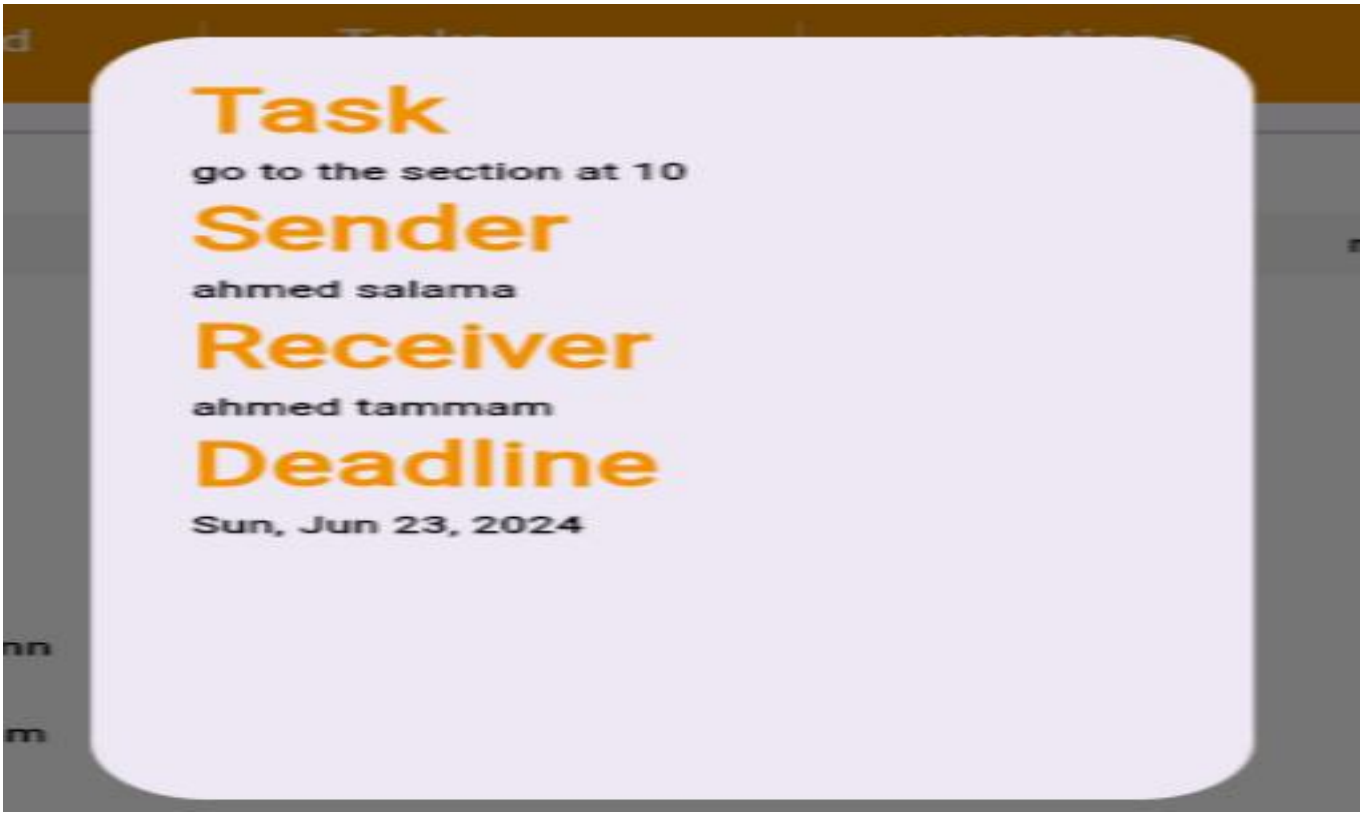


Figure 29 task details

5.2.11 Dean, head, vice vacation

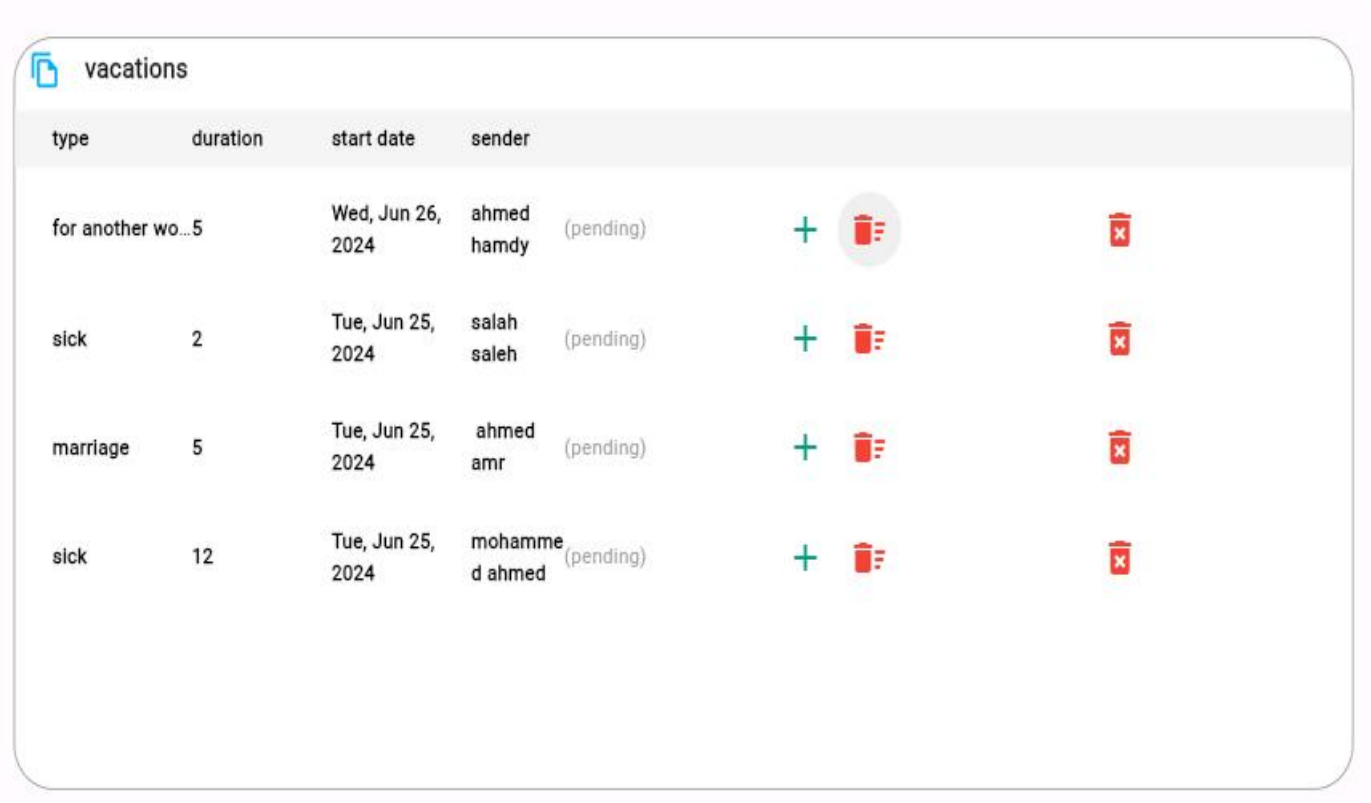


Figure 30 Dean, head, vice vacation

5.2.12 Doctor, assistant vacation

vacations

type	duration	start date	sender	
marriage	5	Tue, Jun 25, 2024	ahmed amr	(pending)
sick	12	Tue, Jun 25, 2024	mohamme d ahmed	(pending)
تعبانننن	12	Sat, Jun 15, 2024	saif ahmed	(accepted)

Figure 31 Doctor, assistant vacation

5.2.13 Add vacation

Figure 32 Add vacation

5.2.14 Action buttons

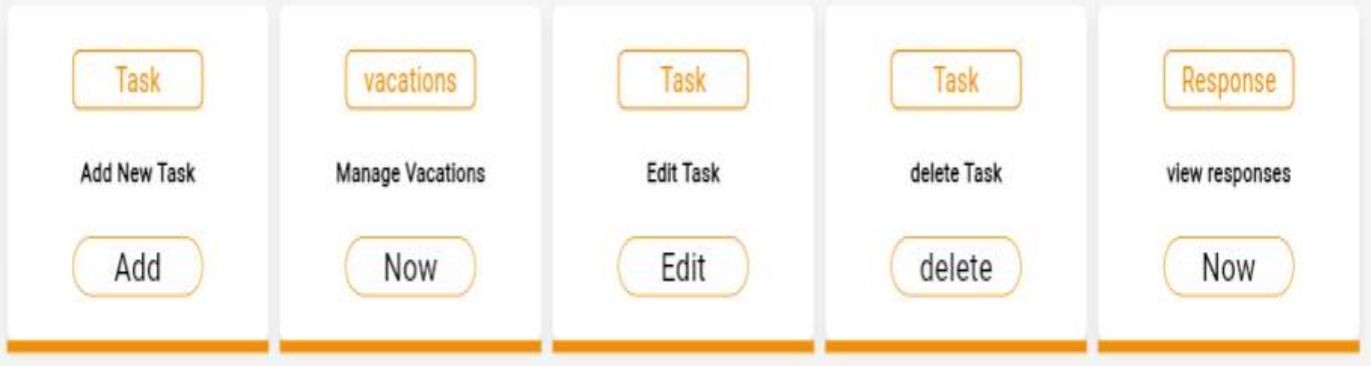


Figure 34 Action buttons

5.3 Implementation of software

github.com

https://github.com/salahsaleh1015/workflow_management_system

Conclusion

The system aims to improve the task management process at a Instituteby providing a more efficient and reliable platform for managing tasksand vacations. With the integration of advanced technologies such as [Flutter, Dart, Firebase, MySQL, Visual Studio Code, Figma]

The system promises to streamline the task management process and reduce errors caused by manual paper-based management. The use of a user-friendly interface, role-based dashboards, and secure data storage ensures that the system is easy to use and meets the requirements of the users.

Additionally, the system's scalability and performance have been carefully considered to ensure that it can accommodate increasing user numbers and handle the demands of large amounts of data. The design and development of the system have followed best practices and standards to ensure its long-term sustainability and effectiveness.

In conclusion, the system represents a major step forward in task management at the Institute and promises to deliver substantial improvements in terms of efficiency, accuracy, and user experience. With careful attention to detail and the use of the latest technologies,the system is poised to revolutionize the way tasks are managed and vacations are carried out at the Institute.

References

1. *Dart Programming language Developed by Google*

2. *Flutter Frame Work UI Tool Kit version 3.16.9*

3. *Firebase Cloud Computing services*

4. *GitHub Platform allow users to manage the code*

Abstract

يهدف هذا المشروع إلى تطوير نظام إدارة سير العمل للمعهد، مما يساعد على تبسيط التواصل وإدارة المهام بين المستويات المختلفة لموظفي المعهد، مثل العميد ونائب العميد ورؤساء الأقسام والدكاتره والدكاتره المساعدين وكذالك المعيدين .
يتضمن النظام أربعة مستويات من المستخدمين، حيث يستطيع كل مستوى إسناد المهام إلى المستويات الأدنى. يتم أيضًا تضمين السكرتير، الذي يمكنه التواصل مع العميد فقط، والمسؤول، الذي يمكنه عرض المهام ولكن لا يمكنه رؤية أي تفاصيل للمهمة للأمن ويمكنه حذفها.
يمتلك كل مستخدم لوحة تحكم مخصصة يمكنه الوصول إليها، والتي تعرض جميع المهام المخصصة له والمهام المخصصة للآخرين.
تسمح هذه الميزة للمستخدمين بتتبع مهامهم ومسؤولياتهم بسهولة ويمكن أن تساعد أيضًا في تحسين المساءلة والتواصل بين موظفي المعهد.
كما يتضمن النظام خاصية طلب واعتماد الإجازة للموظفين بأنواعها المختلفة مثل الإجازة الاعتيادية، العارضة، الإجازة المرضية، الإجازة بدون راتب، إجازة العمولة، والإذن.
يتم تنفيذ هذا النظام باستخدام [Flutter, Dart, Firebase, MySQL, Visual Studio Code, Figma]، كما تم تصميم واجهة المستخدم الخاصة به لتكون سهلة الاستخدام.
تم اختبار النظام وتقييمه، وأظهرت النتائج أنه يحسن بشكل كبير التواصل وإدارة المهام داخل المعهد.