# Software Requirement and Design Specifications

***Employee Task Management System***

***Version: [1.0]***

| *Course Code* | CS 3004 |
| --- | --- |
| *Instructor* | *Mrs Syedab Rubab Jaffar* |
| *Project Team* | *Saifullah 21k4900*  *Umar hashmi 21k-3270*  *Rosin Kumar 21k-3184* |
| *Submission Date* | 03/12/2023 |

## [Instructions]

###### No section of template should be deleted. You can write ‘Not applicable’ if a section is not applicable to your project. But all sections must exist in the final document.

* *All comments/examples mentioned in square brackets ([]) are in the template for explanation purposes and must be replaced / removed in final document.*

###### This’ Instruction’ section should also be removed in final document.

*Table of Contents*

1. [*INTRODUCTION 5*](#_gjdgxs)
   1. [Purpose of Document 5](#_30j0zll)
   2. [Intended Audience 5](#_1fob9te)
2. [*OVERALL SYSTEM DESCRIPTION 6*](#_3znysh7)
   1. [Project Background 6](#_2et92p0)
   2. [Project Scope 6](#_tyjcwt)
   3. [Not In Scope 6](#_3dy6vkm)
   4. [Project Objectives 6](#_1t3h5sf)
   5. [Stakeholders 6](#_4d34og8)
   6. [Operating Environment 6](#_2s8eyo1)
   7. [System Constraints 6](#_17dp8vu)
   8. [Assumptions & Dependencies 6](#_3rdcrjn)
3. [*EXTERNAL INTERFACE REQUIREMENTS 7*](#_26in1rg)
   1. [Hardware Interfaces 7](#_lnxbz9)
   2. [Software Interfaces 7](#_35nkun2)
   3. [Communications Interfaces 7](#_32hioqz)
4. [*FUNCTIONAL REQUIREMENTS 8*](#_1ksv4uv)
   1. [*FUNCTIONAL HIERARCHY 8*](#_44sinio)
   2. [Use Cases 8](#_2jxsxqh)
      1. [[Title of use case] 8](#_z337ya)
5. [*NON-FUNCTIONAL REQUIREMENTS 9*](#_3j2qqm3)
   1. [Performance Requirements 9](#_1y810tw)
   2. [Safety Requirements 9](#_4i7ojhp)
   3. [Security Requirements 9](#_2xcytpi)
   4. [User Documentation 9](#_1ci93xb)

[*SDS 10*](#_3whwml4)

1. [***SYSTEM ARCHITECTURE 11***](#_2bn6wsx)
   1. [*SYSTEM LEVEL ARCHITECTURE 11*](#_qsh70q)
   2. [*SOFTWARE ARCHITECTURE 11*](#_3as4poj)
2. [***DESIGN STRATEGY 12***](#_1pxezwc)
3. [***DETAILED SYSTEM DESIGN 13***](#_49x2ik5)
   1. [*DATABASE DESIGN 13*](#_2p2csry)
4. [*APPLICATION DESIGN 15*](#_147n2zr)
5. [*REFERENCES 15*](#_3o7alnk)
6. [*APPENDICES 17*](#_ihv636)

## Introduction

##### Purpose of Document

##### The purpose of this document is to provide a detailed view of the project we have been working on.

This Software Requirements Specification will describe the processes and functions of the Task Management System

##### Intended Audience

The intended audience for this document will be the designers of this system, Mrs.Syeda Rubab Jaffar and her teaching assistant. The major portion of the product will be described within this documentation, with a possible upgrade to the system available in the future which may include a points reward system redeemable for avatars and avatar upgrades.

***Definition of Terms, Acronyms and Abbreviations***

*[This section should provide the definitions of all terms, acronyms, and abbreviations required to interpret the terms used in the document properly. ]*

| ***Term*** | ***Description*** |
| --- | --- |
| *ASP* | *Active Server Pages* |
| *DD* | *Design Specification* |
| *UCD* | *USE CASE DIAGRAM* |
| *URL* | *UNIFORM RESOURCE LOCATOR* |
| *API* | *Application Programming interface* |
| *SQL* | *Structured Query Language* |
| *GUI* | *Graphical User Interface* |
| *DBA* | *Data Base Administrator* |

##### Document Convention

Font style: Arial.

Font size: 10

Font decoration: none.

*For headings: (As given)*

Font style: Arial.

Font size: 12

Font decoration: Italic, and Bold.

## Overall System Description

##### Project Background

##### The idea behind this project is to allow Create an efficient task management system to enhance productivity and streamline task management processes.

##### Project Scope

##### The Task Management System provides an interface for both managers and workersto use to track their daily tasks. Each user, manager or worker, will have a different user interface. The manager will have a traditional manager user interface, with the worker having a traditional worker interface. The manager interface will allow for entering of new tasks, editing tasks, deleting tasks, and searching of tasks. The manager interface will also allow for creating and deleting of users. Lastly, this interface will have the permission to reset passwords. The worker interface will allow them to search tasks and mark tasks that they have completed. The system will also provide a history of previous tasks assigned or completed.

##### Not In Scope

##### Not Applicable.

##### Project Objectives

##### Efficient Task Management:

##### Enable employees to create, update, and delete tasks, ensuring a streamlined process for managing work assignments.

##### Task Assignment and Tracking:

##### Facilitate task assignment from administrators or managers to employees, allowing for clear ownership and progress tracking.

##### Task Prioritization:

##### Provide a mechanism for assigning priorities to tasks, helping employees focus on high-priority assignments and meet deadlines effectively.

##### Employee Collaboration:

##### Foster collaboration among employees by allowing them to view each other's tasks, facilitating teamwork and resource sharing.

##### Task Status Monitoring:

##### Allow administrators and employees to monitor the status of tasks, providing visibility into ongoing work and completed assignments.

##### Task History Tracking:

##### Record and maintain a history of tasks, including updates and changes, for reference and auditing purposes.

##### Review and Feedback:

##### Enable employees to add reviews and feedback for completed tasks, promoting continuous improvement and communication.

##### Departmental Organization:

##### Organize tasks by departments to enhance department-level collaboration and management.

##### User Authentication and Authorization:

##### Implement secure user authentication to ensure that only authorized individuals can access and modify tasks. Define roles and permissions to control access levels.

##### User-Friendly Interface:

##### Design an intuitive and user-friendly interface to enhance the overall user experience, making task management accessible and efficient.

##### Notification System:

##### Implement a notification system to alert users about task assignments, updates, and approaching deadlines.

##### Security Measures:

##### Incorporate security measures to protect sensitive data, ensuring the confidentiality, integrity, and availability of task-related information.

##### Reporting and Analytics:

##### Provide reporting and analytics features to generate insights into task performance, workload distribution, and overall productivity.

##### Scalability:

##### Design the system to scale effectively, accommodating the growing number of users and tasks without compromising performance.

##### Stakeholders

##### The admin, who can perform various operations that no one else is authorized to do.

##### The user who can login and use the system anytime once he has signed up on the website by the amdin. The user can view, complete tasks and also add their remarks. The admin can also track the taks, once the task is placed.

##### Developer who developed the project, for further releases and new versions (Both front end, and back end).

##### Internal database engineer who manages the database working.

##### Software quality assurance engineer, who is responsible to test the environment for bugs, and errors.

##### Operating Environment

***Hardware platform:*** It needs pretty basic hardware requirements. Any average system will do the work. The hardware we used to run this is mentioned below:

* Core i5 6th Generation
* 8gb ram (hardly uses 1 or 2gb(s) of ram for all the softwares we used)
* Less than 200mb of disk space for project, and related data.

***Operating System:*** Windows 10 any variant (Recommended)

Network Environment: Should have a decent internet connection.

***Applications:***

* Visual Studio Code
* XAMPP
* MySql Server

##### System Constraints

##### Software constraints:

This system doesn’t require any software constraints so far.

##### Hardware constraints:

It needs pretty basic hardware requirements. Any average system will do the work. The hardware we used to run this is mentioned below:

* Core i5 8th Generation
* 16gb ram (hardly uses 1 or 2gb(s) of ram for all the softwares we used)
* Less than 200mb of disk space for project and its dependencies

##### Cultural constraints:

##### The person should know how to read, and understand basic English language.

##### Legal constraints:

##### Not applicable.

##### Environmental constraints:

* There are no environmental constraints. The system can be used anywhere anytime.
* The project is developed for people of all ages and ethnicities

##### User constraints:

##### No specific age constraints. Users of all ages can use the webpage.

##### Assumptions & Dependencies

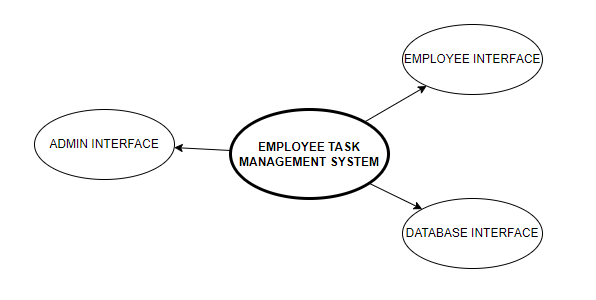
##### Assumptions:

##### Data is saved incase of any exception.

1. Employees are dependent on the admin

## External Interface Requirements

The Employee task management system seamlessly connects with essential components, including the MySQL database, standard web browsers, and Xampp server, ensuring efficient data management, user interaction, and feature integration. The main overview of the project can be understood using the context diagram below:



##### Hardware Interfaces

##### The system is perfectly supported by desktop computers. It cannot be supported on mobiles currently as it hasn’t been deployed to a domain.

##### The type of data we are taking from the user are login credentials, delivery information, and we’re saving the information for future orders.

Processor: Intel

Installed Memory: 2 GB or Higher

Speed: 1.40 GHz or Higher

Operating System: 32/64-bit Operating System

##### Software Interfaces

**Operating System:** Windows 10

**Database:** MySQL Workbench

**Webserver**: XAMPP

**Web Technologies**: HTMLL/ CSS/ JS/ FLASK

**Ide and tools:** *Visual Studio Code*

##### Communications Interfaces

##### User email for signup should be valid.

##### No encryption usage, or data transfer or synchronization issues.

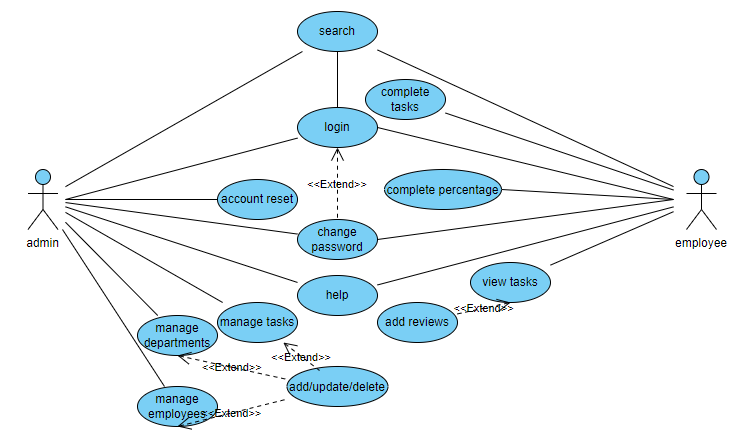
## Functional Requirements

##### Functional Hierarchy

*[This section will give a big picture of overall system functionality. The main modules/features of system and their sub-functions will be described here in the form of a functional hierarchy so that, before getting into the use case, audience could grab the idea of overall system functions.]*

##### Use Cases

***4.2.1. [Employee Task Management system]***



| ***Use Case Description*** | |
| --- | --- |
| ***Use Case name:*** | Employee task management system. |
| ***Use Case Description:*** Efficiently manage tasks and enhance collaboration: Employees can view, complete tasks, provide feedback, and access a user-friendly system for streamlined task management whereas Employers are the admins. | |
| ***Primary actor:*** Admin | ***Other actors:*** Employee |
| ***Stakeholders:*** user, admin, software developer, software designer, DBA |  |
| ***Relationships***   * ***Includes:***   search,manage tasks, manage employes, manage departments, account reset, help, change password, view tasks, complete tasks, add reviews/feedbacks   * ***Extends:***   login extends from search  change password extends from login  add/update/delete extends from manage employees, manage departments and manage tasks | |
| ***Pre-conditions:***  ▪ User must have an adequate internet connection, and a computer to run the webpage. | |
| ***Flow of Events:***  **Admin Use Cases:**   1. Login: 2. System verifies credentials. 3. Admin can select the "Manage Tasks, “Manage employees”, “add task”, “view tasks” options. 4. System guides the user accordingly and displays pages.   **Employee Use Cases:**   1. Login 2. System verifies credentials. 3. If valid, the system grants access; otherwise, an error message is displayed. 4. Employee can select the "View Tasks", “Complete Tasks:”, “Add Reviews/Feedbacks”, “Search”, “Help”, “Change Password” options. 5. System guides the user accordingly and displays pages. | |
| ***Alternative and exceptional flows:***  1. If login user doesn’t match, the information is taken again.  2. if the employee doesn’t exit properly, and he logs out, his task updation are lost.  3. If task is scheduled for future, it can't be tracked. | |
| ***Post-conditions:***  - Admin registered.  - Employee registered.  - Departments registered. | |

## Non-functional Requirements

##### Performance Requirements

1. Performance wise, our project is built to be very responsive and fast. The transitions between the interfaces take no time.
2. Capacity wise, our system is very storage friendly, hardly requires some Mbs of data.
3. Safety wise, the data of users, interacting with our system stays safe and can only be accessed by the system owner.
4. The software is reliable in a sense that it fulfills all the needs that it is promised to fulfill. It was tested for any sort of bugs/ issues and was fixed by the developers eventually.

##### Safety Requirements

1. We took extra care that our system must not cause any damage on the machine on which the user is running our system.
2. The only thing that the user should take care of is the entry of dummy (fake) results in the database. Dummy data must be deleted from the database.
3. Data Integrity: Implement measures to prevent data corruption and ensure the accuracy of order information.
4. User Authentication: Safeguard user data through secure login processes, protecting against unauthorized access.
5. Transaction Security: Employ encryption protocols for secure payment transactions, mitigating risks associated with financial information.

##### Security Requirements

1. External users such as someone out of the organization must not be given access to the system’s Admin panel. Login ensures this.
2. Only the Stakeholders should have access to the system.
3. The data of the user stays safe and untouchable. So, privacy is maintained.

##### User Documentation

User manuals, will be provided alongside and they will be explained how to use the system, first they are asked to sign-in if not already and then login to continue using the system.

# SDS

## System Architecture

## The architecture embodies the major static and dynamic aspects of a system. It is a view of the whole system highlighting the important characteristics and ignoring unnecessary details. In the context of our approach, architecture is primarily specified in terms of views of tier architecture which is a client-server architecture in which the presentation, the application processing and data management are logically separate processes.

**Components:**

Frontend:

* Admin Interface
* Employee Interface

Backend:

* Application Server

Admin Module

Employee Module

Task Management Module

* Database Server

Admin Manager Table

Department Table

Employee Table

Task Table

Task Status Table

**Interactions:**

Admin Interface:

Allows the admin to perform operations such as adding, updating, or deleting employees, departments, and tasks.

Provides access to view task status, view task history, and logout functionality.

Employee Interface:

Allows employees to view their tasks, add reviews for tasks, and edit their passwords.

Admin Module (Backend):

Handles requests from the Admin Interface.

Interacts with the Database to perform CRUD (Create, Read, Update, Delete) operations on Admin Manager, Department, Employee, Task, and Task Status tables.

Employee Module (Backend):

Handles requests from the Employee Interface.

Interacts with the Database to fetch and update employee-related information.

Task Management Module (Backend):

Manages tasks and their statuses.

Interacts with the Database to create, update, and delete tasks.

Handles requests for viewing task status and history.

Database Server:

Stores data in the Admin Manager, Department, Employee, Task, and Task Status tables.

Enforces relationships between tables using foreign keys.

**Flow of Data:**

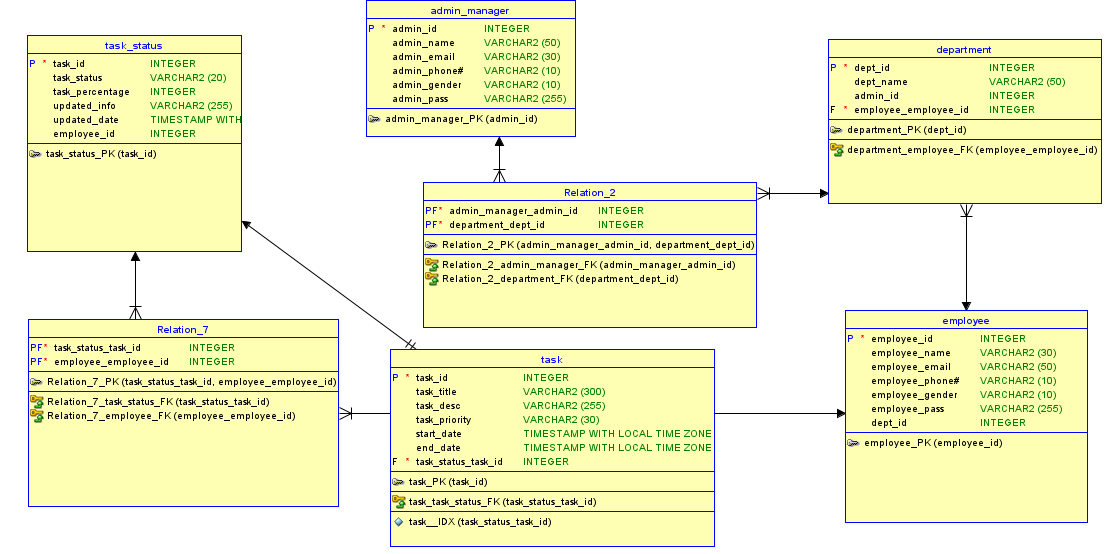
* Admin or Employee Interface sends a request to the corresponding module in the Application Server.
* Admin or Employee Module processes the request, which may involve interacting with the Database Server.
* Database Server performs CRUD operations based on the request.
* The result or response is sent back to the requesting Interface.

**Data Flow Example:**

Admin adds a new employee:

* Admin Interface sends a request to the Admin Module.
* Admin Module interacts with the Database to insert a new record into the Employee table.
* Database Server updates the Employee table.
* The result is sent back to the Admin Interface.

## Domain Model:

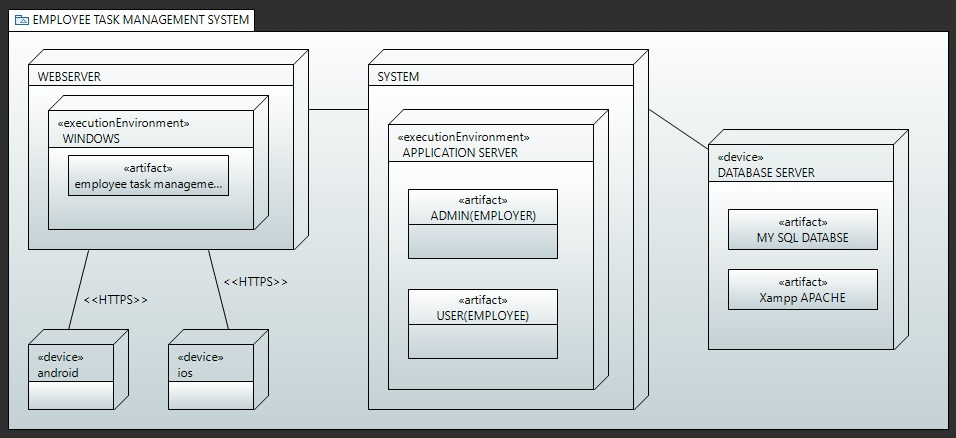


### *System Level Architecture*

### *COMPONENT DIAGRAM*

### 

### *DEPLOYMENT DIAGRAM*



### *Software Architecture*

### *User Interface Layer:*

* Allows users to interact with the system for order placement.

### *Middle tier:*

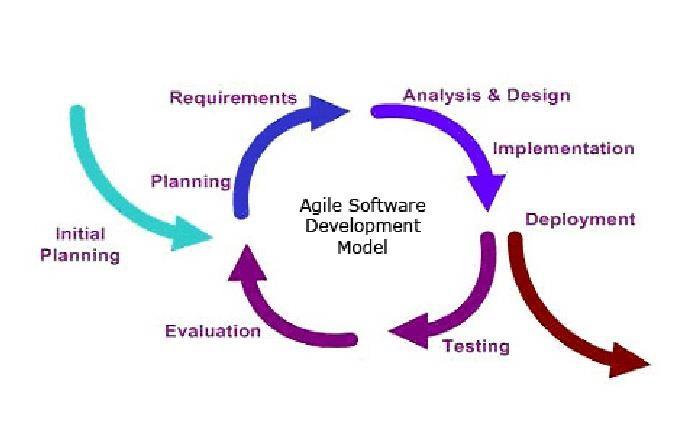
* Manages order processing and administrator functions.

### *Data access Layer:*

* Facilitates interaction with the MySQL database for data storage and retrieval

## Design Strategy

The project uses an agile approach as its design base. This agile approach accelerates adaptation to changing requirements, ensuring the system remains responsive to customer needs and market dynamics in the fast-paced food industry.



## System Reuse:

## The webpage may be deployed on a world wide web, allowing users from all around the world to access the website.

## Future system extension or enhancement:

## We aim to add additional features to the webpage, such as adding chat bot using Machine learning, and deep learning, as well as add real time order tracking (on a live map) as well as integrate an ai.

## User interface paradigms:

## The program prompts the user to enter their information at multiple stages: while signing up, logging in, as well as during checkout. Moreover, the UI also displays the user information at multiple pages, such as displaying the order summary, or menu items.

## Concurrency and synchronization:

## Only one user is allowed to use the system at a time.

## Detailed System Design

## Class Diagram:

## 

## Functions:

## 1. Login:

## Admin enters username and password.

## System verifies credentials.

## If valid, the system grants access; otherwise, an error message is displayed.

## 2. Manage Tasks:

## Admin selects the "Manage Tasks" option.

## System presents a list of existing tasks.

## Admin chooses to add, update, or delete a task.

## System prompts for task details (for add or update).

## Admin provides task details and confirms the action.

## System updates the task database accordingly.

## 3. Manage Employees:

## Admin selects the "Manage Employees" option.

## System presents a list of existing employees.

## Admin chooses to add, update, or delete an employee.

## System prompts for employee details (for add or update).

## Admin provides employee details and confirms the action.

## System updates the employee database accordingly.

## 4. Manage Departments:

## Admin selects the "Manage Departments" option.

## System presents a list of existing departments.

## Admin chooses to add, update, or delete a department.

## System prompts for department details (for add or update).

## Admin provides department details and confirms the action.

## System updates the department database accordingly.

## 5. Search:

## Admin selects the "Search" option.

## System presents search options for tasks, employees, or departments.

## Admin enters search criteria.

## System displays search results.

## 6. Password Reset:

## Admin selects the "Password Reset" option.

## System verifies admin's identity (e.g., through security questions).

## Admin provides a new password.

## System updates the password in the database.

## 7. Help:

## Admin selects the "Help" option.

## System provides relevant help information or redirects to documentation.

## 8. Change Password:

## Admin selects the "Change Password" option.

## Admin provides the current password and a new password.

## System updates the password in the database.

## 9. Employee Login:

## Employee enters username and password.

## System verifies credentials.

## If valid, the system grants access; otherwise, an error message is displayed.

## 10. View Tasks:

## Employee selects the "View Tasks" option.

## System displays a list of assigned tasks for the employee.

## .11. Complete Tasks:

## Employee selects a task to mark as completed.

## System updates the task status to "Completed."

## 12. Add Reviews/Feedbacks:

## Employee selects the "Add Reviews/Feedbacks" option.

## System prompts for the task for which feedback is provided.

## Employee provides feedback, and the system updates the database.

## 13. Search:

## Employee selects the "Search" option.

## System presents search options for tasks.

## Employee enters search criteria.

## System displays search results.

## 14. Help:

## Employee selects the "Help" option.

## System provides relevant help information or redirects to documentation.

## 15. Change Password:

## Employee selects the "Change Password" option.

## Employee provides the current password and a new password.

## System updates the password in the database.

## 

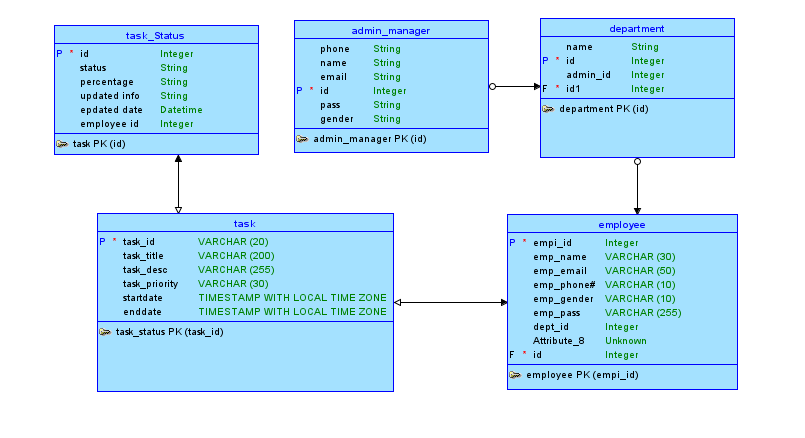
### *Database Design*

#### ER Diagram<Barker notation>

#### 

#### 

#### ER Diagram <Bachman notation>



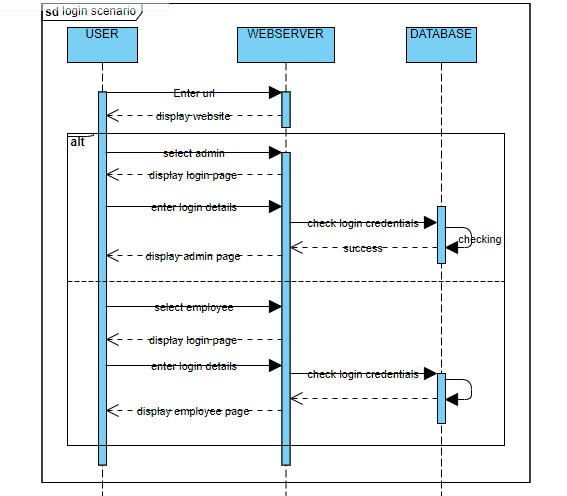
#### Data Dictionary

Not Applicable

### *Application Design*

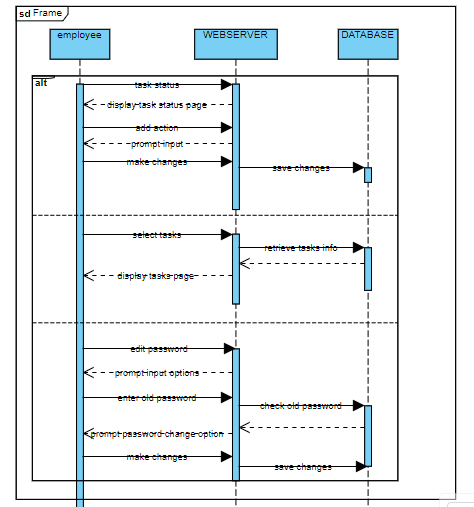
#### Sequence Diagram

* + - 1. ***<Sequence Diagram 1>***



***Explanation:***This sequence diagram illustrates the working on Login feature. The user chooses required button, and enters his credentials, which is checked: if the validation is successful, the system displayed the required admin/employee page. if validation fails, error message is shown and rollback happens

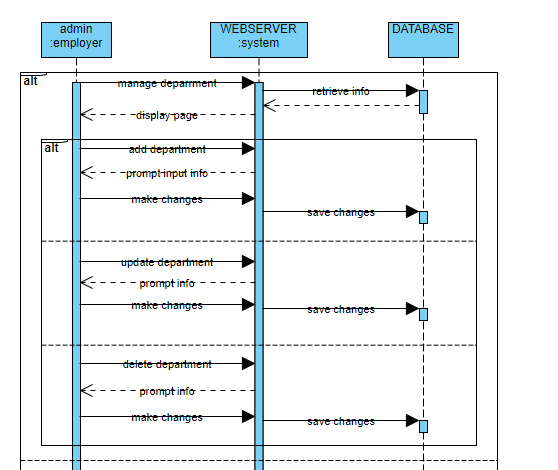
* + - 1. ***<Sequence Diagram 2>***

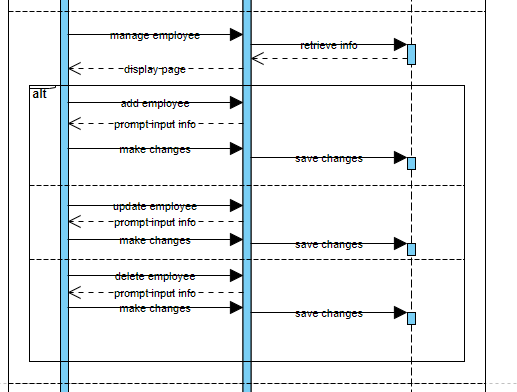


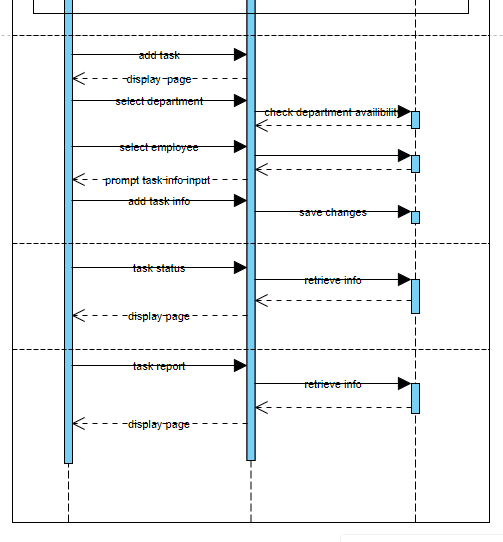
Explanation: this sequence diagram is for the employee functions.

*.*

* + - 1. ***<Sequence Diagram 3>***



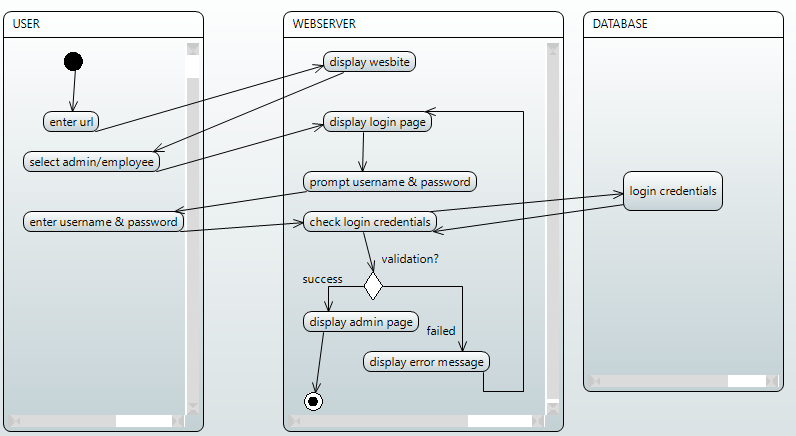




Explanation: this sequence diagram is for the admin page and its features.

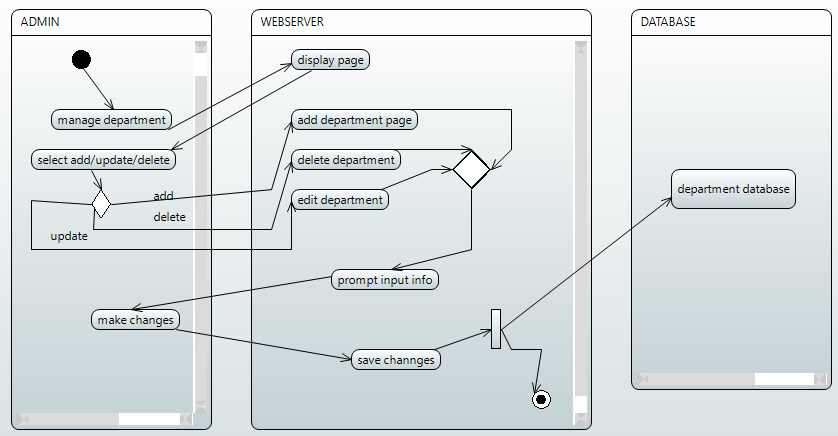
#### Activity Diagram

* + - 1. ***<Activity Diagram 1>***



Explanation: This activity diagram illustrates the working on Login feature for both admin/employee. The user chooses required button, and enters his credentials, which is checked: if the validation is successful, the system displayed the required admin/employee page. if validation fails, error message is shown and rollback happens.

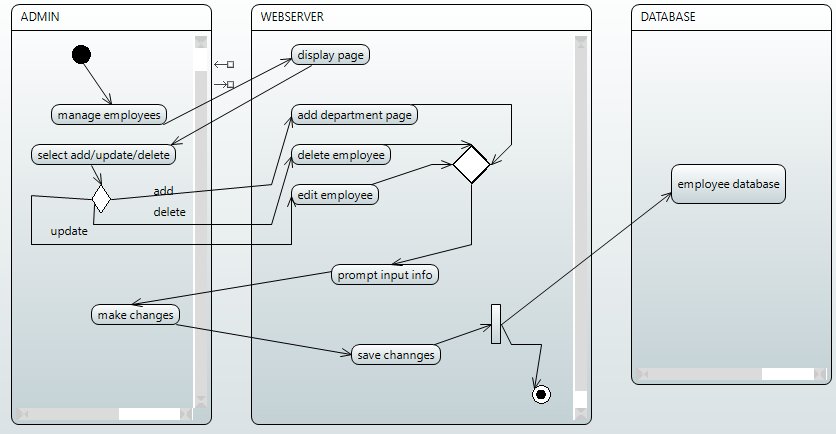
* + - 1. ***<Activity Diagram 2>***



Explanation: this activity diagram is for the delivery management feature. the page has 3 options, add, update or delete departments. it also shows all the saved departments and their info. once the admin selects any option, he can make changes and save them, hence ending the activity diagram.

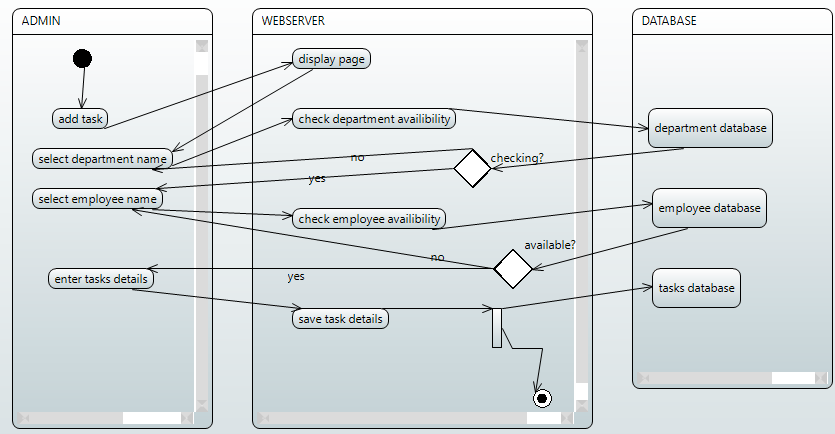
.

***9.1.3.3. <Activity Diagram 3>***



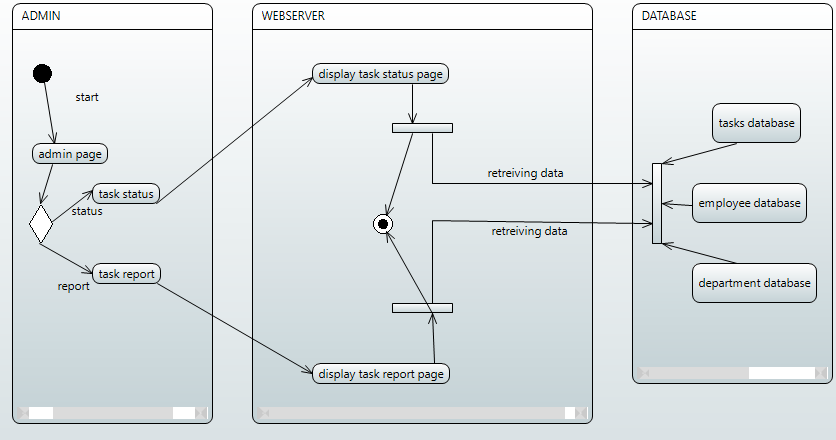
Explanation: this activity diagram is for the employee management feature. The page has 3 options, add, update or delete employees. It also shows all the saved employees and their info. Once the admin selects any option, he can make changes and save them, hence ending the activity diagram. NOTE: an employer can only us this project if he/she is added by the employer.

***9.1.3.4. <Activity Diagram 3>***



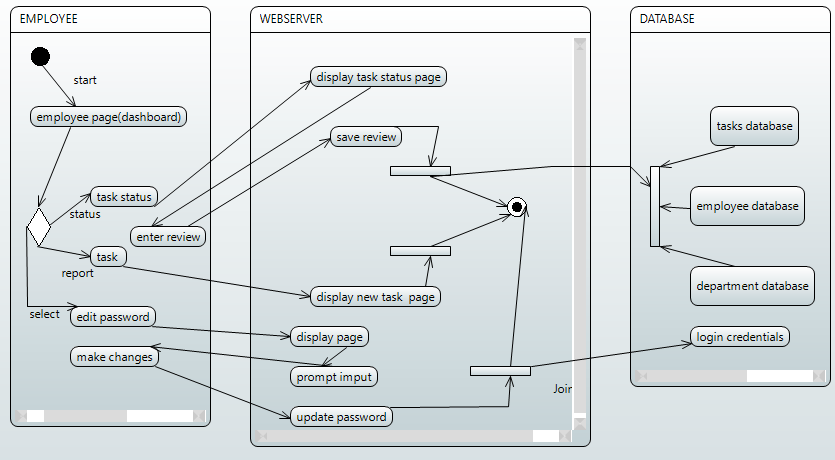
Explanation: this activity diagram is for the add task feature. the add task feature works a bit differently, as compared to the other features. here, to add the admin has to make sure that the required department and employee are added beforehand before adding their task and assigning them to it. once it is added, admin can then add task details and move forward. The page also has update or delete tasks feature.

***9.1.3.5. <Activity Diagram 3>***



Explanation: this activity diagram is for the admin page features. The page has 2 options view task status and view task report. The information is fetched from the database and displayed on their display pages.

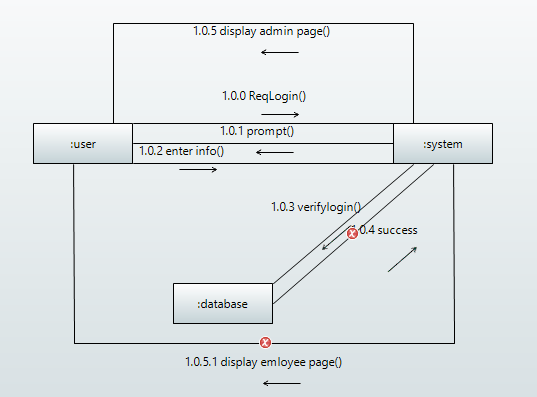
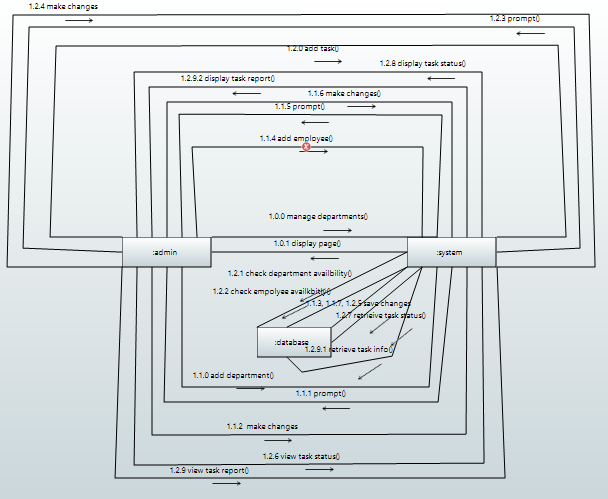
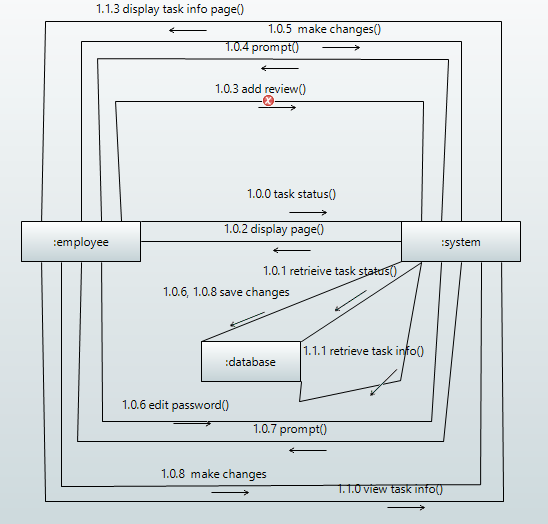
***9.1.3.6. <Activity Diagram 3>***



Explanation: this activity diagram is for the Employee page and its features. The page has 3 options,

task status, tasks, and edit password options. Task status displays the task status page, where employee can thier reviews as well. Tasks page displays new task pages and their info. edit password option allows to edit their password after the admin creates their account.

* + 1. ***Collaboration Diagram***

1. 
2. 
3. 

## References

Ronacher, A., 2017. Flask documentation. *Retrieved August*, *15*, p.2018.

Grinberg, M., 2021. Flask-socketio documentation. *línea]. Disponible en: https://flask-socketio. readthedocs. org/en/latest/.[Último acceso: 2015]*.

https://jinja.palletsprojects.com/en/3.0.x/

https://getbootstrap.com/docs/3.3/getting-started/

<https://flask.palletsprojects.com/en/2.0.x>

Silbio Peroni et al, “Research Articles in Simplified HTML: A Web-first format for HTML-based scholarly articles,” PeerJ Computer Science 3(194):e132. 2017. [Online] Available: https://www.researchgate.net/publication/320171394\_Research\_Articles\_in\_Simplifie d\_HTML\_A\_Web-first\_format\_for\_HTML-based\_scholarly\_articles [Accessed 2020].

Sorana D. Bolboaca, “SQL by Example. 2. PHP and MySQL Web Application based on Tanner-Whitehouse Standard,” Leonardo Electronic Journal of Practices and

Technologies 2(2). 2003. [Online] Available:

https://www.researchgate.net/publication/26440510\_SQL\_by\_Example\_2\_PHP\_and\_ MySQL\_Web\_Application\_based\_on\_Tanner-Whitehouse\_Standard. [Accessed 2020].

Mrs. Gunjan Behl and Mr Nripesh Kumar Nrip. “A STUDY OF AGILE SOFTWARE

DEVELOPMENT MODEL,” International Conference on Current Trends &

Challenges in Management, Engineering, Computer Application & Technology-2012.

[Online] Available:

https://www.researchgate.net/publication/264626696\_A\_STUDY\_OF\_AGILE\_SOFT WARE\_DEVELOPMENT\_MODEL. [Accessed 2020].

Sheetal Sharma et al. “Agile Processes and Methodologies: A Conceptual Study,” International Journal on Computer Science and Engineering (IJCSE). 2012. [Online] Available:

https://www.researchgate.net/publication/267706023\_Agile\_Processes\_and\_Methodo logies\_A\_Conceptual\_Study. [Accessed 2020].

## Appendices

Not Applicable.