COMP-1610

PROGRAMMING ENTERPRISE COMPONENTS

Author: Wajahat Ali Mughal - 001191074

Group No: 13

Module Leader: Dr. Rafael Martinez-Torres

Table of Contents

Part A

Design Documentation

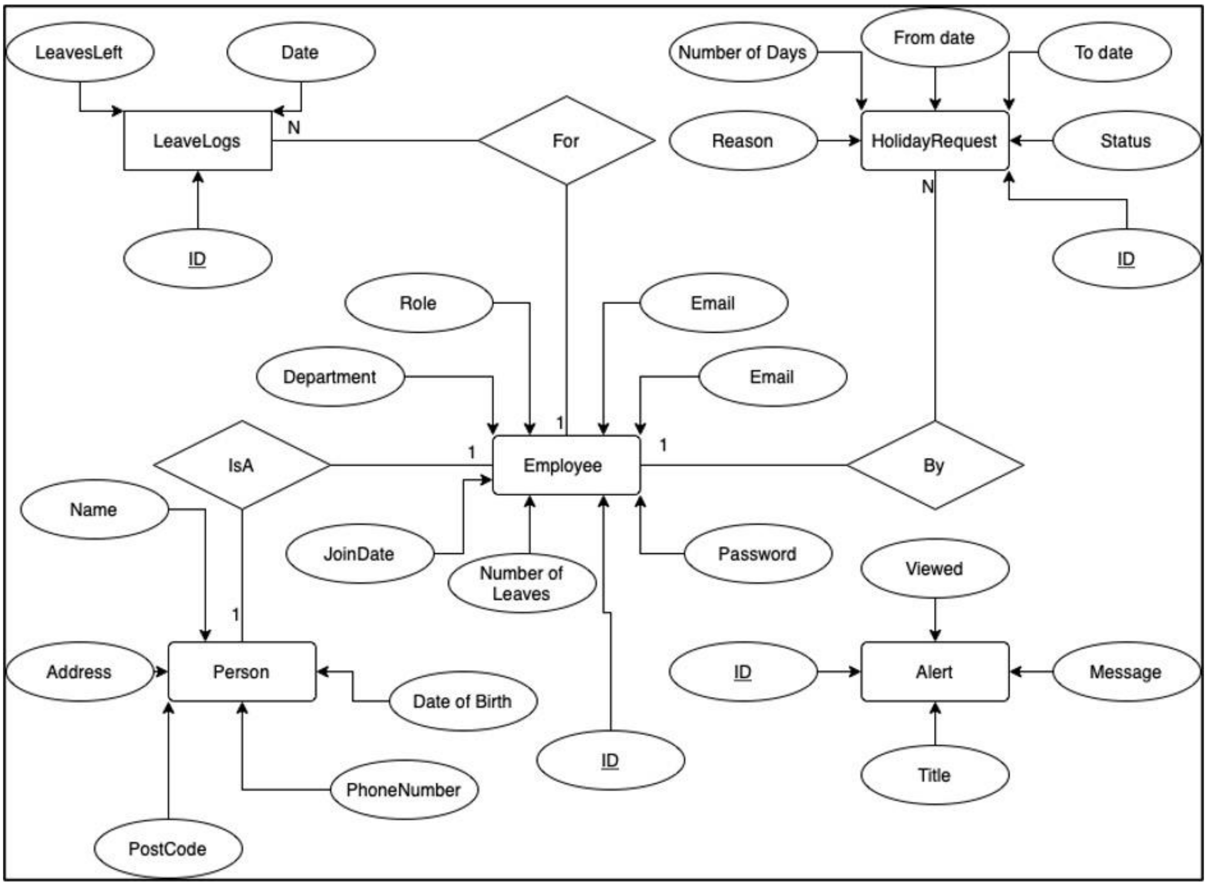
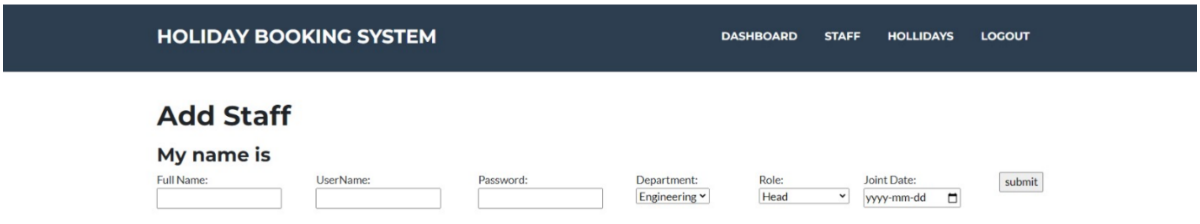


Fig1 : Entity Relationship Diagram for database structure

Screenshots of application



Add Staff

My name is

Full Name:

UserName:

Password:

Department: Engineering

Role: Head

Joint Date: yyyy-mm-dd

Fig2 : Holiday Booking system for adding staff

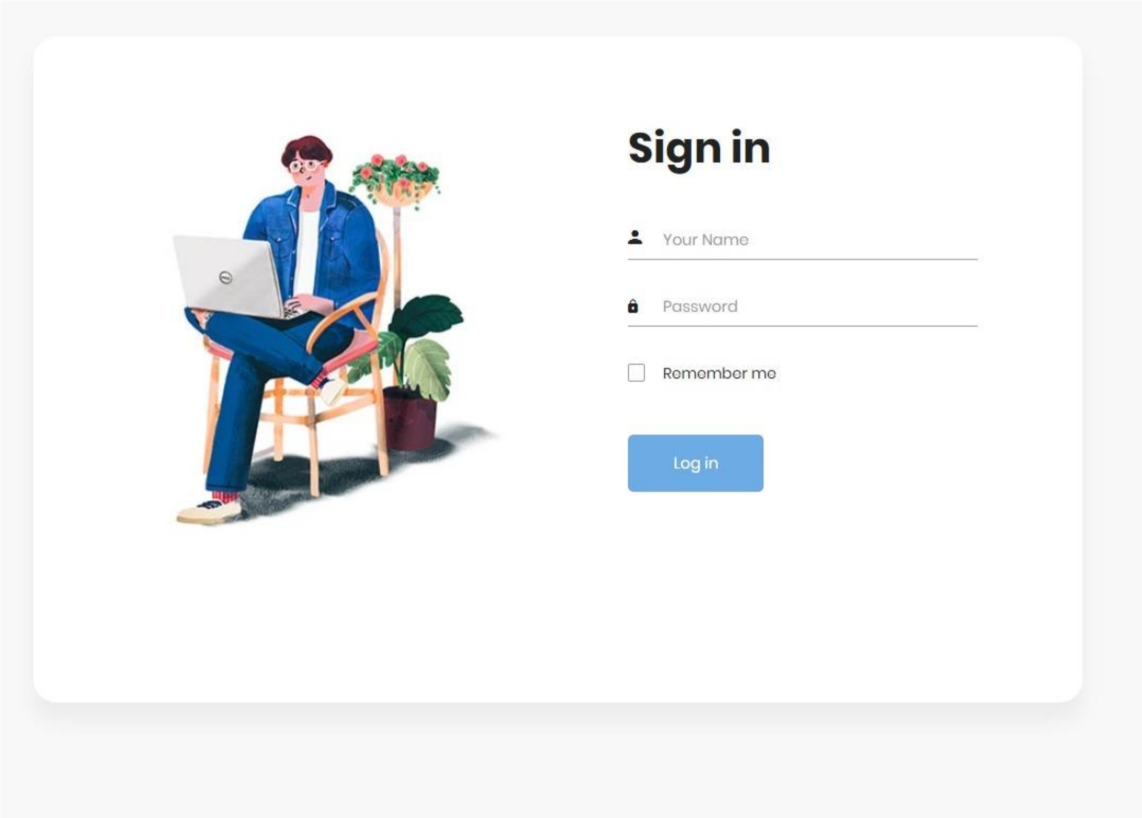
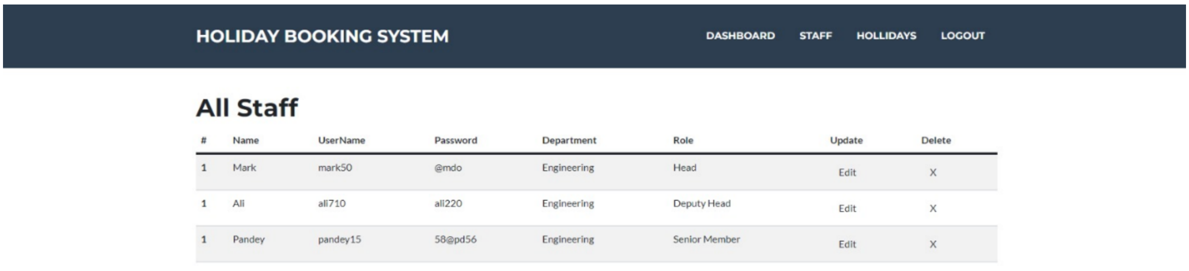


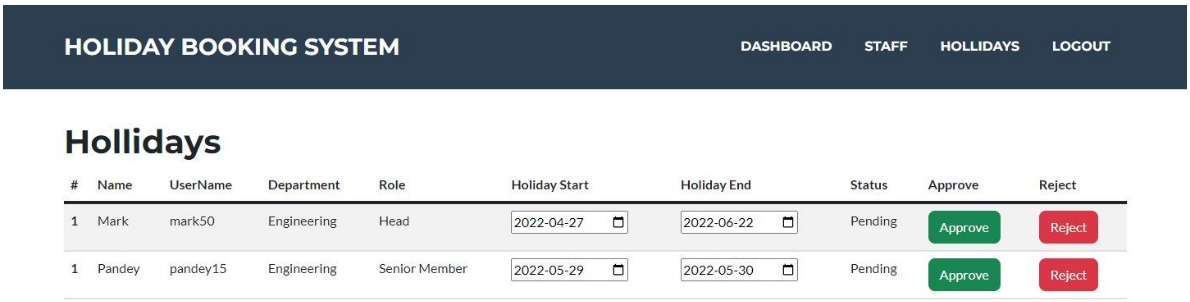
Fig 3: Employee Login



All Staff

#	Name	UserName	Password	Department	Role	Update	Delete
1	Mark	mark50	@mdo	Engineering	Head	Edit	X
1	Ali	ali710	ali220	Engineering	Deputy Head	Edit	X
1	Pandey	pandey15	58@pd56	Engineering	Senior Member	Edit	X

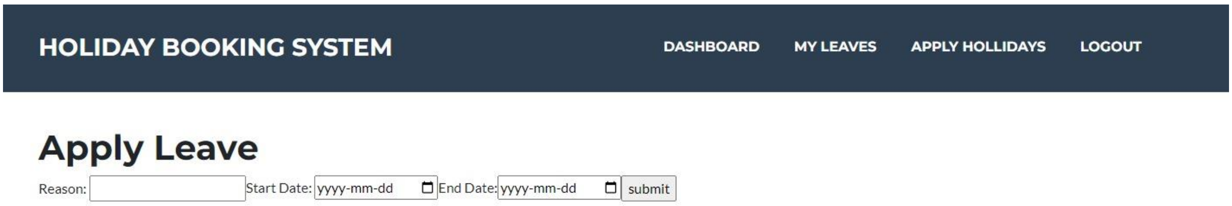
Fig 4: Viewing Employee details in database



Hollidays

#	Name	UserName	Department	Role	Holiday Start	Holiday End	Status	Approve	Reject
1	Mark	mark50	Engineering	Head	2022-04-27	2022-06-22	Pending	Approve	Reject
1	Pandey	pandey15	Engineering	Senior Member	2022-05-29	2022-05-30	Pending	Approve	Reject

Fig 5 : Admin page to approve or reject the holiday request



Apply Leave

Reason: Start Date: yyyy-mm-dd End Date: yyyy-mm-dd

Fig 6: Employee form to apply leave

Evaluation

We have implemented the application based on the requirements. Created the login form for the employees to see their leaves and they can view their outstanding leaves, rejected leaves, and approved leaves. The admin can access see the employee's requested leaves and have the access to accept or reject. When it comes to peak times, admin can prioritise based on their previous ones which they are requested, by outstanding leaves, by seeing the importance and can accept/reject. Database is designed to fetch the data from the database. Whenever the employee request for the leave, the admin will get the automated email with the employee details. This holiday request application will help in providing the leaves based on the priorities and will manage the staff to the workplace. Designing data base is the first and important part where we can get our application in a correct state to work based on the requested form.

As we have divided the tasks among us, it made easy to build entire project on time. While we are working together, we have faced few problems when it comes to planning. First, we have decided to go with the waterfall model. Here comes the issue as the code is very complex, and the requirements are more. So, we have unknowingly started working based on waterfall model as in starting the project looks like small one and easy to implement but later while started working on that we have realized this is not the small as expected and then decided to stop on that as a team. Later, we have decided to go with the agile model by communicating each other and started working on project. Now the team is going in a right track as we communicate each time while we are doing any changes or when we implement anything.

While defining relation between the entities it made us to think twice, and it took time as this is the main part to work on. If the relation is not set to the proper entity, then the whole project will not meet the requirements. So, finally we have by doing rough work together we have defined the relationship between the entities.

As this is the first project we are working on, fetching data from server to jsp is a new challenge to work on. By doing lot of research, we made it possible to get the data without any problems. In this challenge we have learnt by failing each time and improved code every time to get good result.

In evaluation phase while doing testing we have received unexpected results. At first, we have failed in getting the expected output, but we have monitored the logs and everything to check the issue we are facing. We have encountered the issue where it is causing and then we have improved the code to get the expected result.

It made possible as a team because of choosing the right path which is Agile methodology where scrum helped us to focus on our plans by continuously reviewing and updating each other.

Our final implementation is improved by doing regular changes whenever we felt like its not the better way. Changed database structure for three times. I would suggest designing database structure before started working on application. If we have done this before, we may save our time.

Self-Assessment Form

COMP1610 – Programming Enterprise Self-Assessment Report

Your name	Mudamala Roja	**Your Student ID**	001203480

Other members in the Group:	Wajahat Ali Mughal	**Their IDs**	001191074
	Ashishkumar S Pandey		001179060

1. Brief statement of features you have implemented

(THIS SECTION SHOULD BE THE SAME FOR ALL MEMBERS OF THE GROUP)

Feature	Status	Your Comments
Functionality A	Fully completed ☒ Partially completed ☐ Having bugs/Not working ☐ Not implemented ☐	Designed database, web application implemented using Jakarta EE and added User login functionality, access to admin to delete/create/edit employees
Functionality B	Fully completed ☒ Partially completed ☐ Having bugs/Not working ☐ Not implemented ☐	We have done setting up login for employees, submitting holiday request, viewing the holidays which are left, accepted, and rejected.
Functionality C	Fully completed ☒ Partially completed ☐ Having bugs/Not working ☐ Not implemented ☐	Provided access to view the list of requests, permission to accept/reject a request.
Functionality D	Fully completed ☐ Partially completed ☒ Having bugs/Not working ☐ Not implemented ☐	Few changes need to make to meet all the constraints
Functionality E	Fully completed ☐ Partially completed ☒ Having bugs/Not working ☐ Not implemented ☐	Employee app should be changed
Functionality F	Fully completed ☐ Partially completed ☒ Having bugs/Not working ☐ Not implemented ☐	Automated email should be improved
Functionality G	Fully completed ☒ Partially completed ☐	Prioritisation of request is implemented during peak times based on outstanding requests.

	Having bugs/Not working ☐ Not implemented ☐	

Part B

Research on Jakarta EE and other technologies

Jakarta EE assists every one of the engineers with making solid, versatile, enormous scope, multitiered and secure organization applications which we additionally call as big business applications. These applications help in tackling the issues which are experienced by huge endeavors. These applications are valuable for both huge enterprises and individual engineers.

With every one of the highlights it will be complicated when security and dependability included. Be that as it may, by giving API, advancement model and runtime climate, Jakarta EE helps in diminishing intricacy for the engineers.

As the capacity of the application has levels which is known as capacity regions. Multitiered application contains a center level where Jakarta EE focuses more to make application simpler, safer, and more hearty. To make solicitations to center level, we have client level which speaks with the client program. To deal with the solicitations, store in a super durable information store, and to handle application information, we have center level which is isolated into a business level and web level.

Jakarta EE server goes about as a server application to execute Jakarta stage APIs and offers types of assistance. It will go about as an application server to serve information to clients, and server website pages to programs. It additionally offers types of assistance to the parts as a compartment.

Web Container go about as connection point between we server and the parts. The compartment deals with the solicitations to applications. The EJB Container go about as connection point between big business beans to give business rationale in Jakarta EE server and Jakarta EE application.

The application client compartment go about as a connection point between Jakarta EE server and Jakarta EE application clients. This sudden spike in demand for client machine to give door among server and application.

With these highlights, Jakarta EE made us to make and run venture application on Jakarta EE.

We likewise have Spring MVC, Spring Boot, Amazon Corretto and Apache Spark and numerous more.(Anon, n.d.)

In the event that we contrast Jakarta EE and Spring, both are having remarkable highlights. A few engineers will use to go on with Jakarta EE when they have solid programming language.(Anon, n.d.) Some designers who are not more acquainted with programming language will go for spring. Spring system is excellent to go for, however for fledgling getting the hang of spring framework will be troublesome. It is additionally dependable, secure however when contrasted with speed Jakarta EE is quicker than spring.

The vast majority of the designers go for spring with regards to server-side turn of events. In this way, it will be hard to say Jakarta EE is the best when contrasted with spring as both are cutthroat.

Spring MVC is additionally a java web structure which is based on servlet API and remembered for system. The UI is great, and it is not difficult to arrangement. Be that as it may, it is convoluted in certain areas which should be improved to utilize. To configuration website pages, content, and information framework it is not difficult to utilize. As the underlying system and arrangement is simple, planning front pages will be more helpful, and the solidness and versatility is generally excellent. Adaptability should be further developed when contrasted with Jakarta EE and it won't MVC can't uphold all the web administrations and it's difficult to use.(Anon, n.d.)

Apache Spark is awesome in handling information by utilizing information blocks it will send everything to apache document in the information lake. By utilizing ML expectation calculation, AI will in all actuality do an investigation of some kind. As the memory handling is awesome, it will deal with everything incredibly quick. Here the downside is UI. Understanding the logs is extremely challenging. Whenever the execution falls flat, it will be hard to track down how it veered off-track. (Anon, n.d.)

There are advantages and disadvantages for every system. We should utilize relies upon our decision and I have picked Jakarta EE for my programming as I knew about the java EE and have the information on programming and it is exceptionally quick and secure.

Individual Reflection

My Primary job is project director, and BackEnd designer. As an undertaking director I have realized all the philosophy to choose the best one. For the first as we said we have dealt with cascade model yet later I have believed that this burning through the time as the code is mind boggling and can't meet the prerequisites in a given time. Thus, I have chosen to concentrate on every one of the models and picked nimble. By taking the choice on approach, the venture went in a smooth way since we update each other each time for doing any updates or checking for the options which can be preferable over the current one.

As a frontend engineer I have figured out how to code by gaining from fundamentals. I have acknowledged every one of the ideas from the group for the better enhancements and planned the UI.

I have figured out how to plan a code for client login and enrollment which can empower the administrator to erase/make/alter a worker. By utilizing HTML, CSS, Javascript, Bootstrap I have planned frontend part. I have likewise helped my group in planning backend by learning servlet, Jakarta EE and aided in planning the element relationship for data set structure. By including in each job, I have acquired information in all aspects of my venture.

With the group I lived it up in learning each job what they have planned and giving ideas for improved result. The best thing which occurred with the gathering is correspondence, while beginning the venture, we haven't involved much in one another's work however when we attempted to change to spry strategy made us to reconsider and took everybody's ideas.

Everything got worked on in some stage in the wake of doing the slip-ups lastly made conceivable to fabricate a task and satisfied working the application in light of the necessities.

Coursework Contribution (completed as a group)

In percentage, please indicate the work contribution of each member. This should be agreed by all group members**. The total of all members work must add to 100%**

Team member name	**Student ID**	**individual overall work contribution (%)**	**Signature**	**Note**
Student: **Wajahat Ali Mughal**	**001191074**	33.3	Wajahat Ali	
Student: **Ashishkumar S Pandey**	**001179060**	33.3	Ashishkumar	
Student: **Mudamala Roja**	**001203480**	33.4	Roja	
Student:				
Student:				
Student:				
		Total 100%	**100**	

Describe each task you performed	Student: Roja Mudamala work contribution	Student: Wajahat Ali work contribution	Student: Ashishkumar work contribution	Student : work contribution in %	Student: work contribution in %	Student: work contribution in %	Student: work contribution in %	**total**
Feature A:	45	30	25					100%
Feature B:	30	40	30					100%
Feature C:	45	35	30					100%
Feature D:	25	45	30					100%
Feature E:	30	35	35					100%
Feature F:	30	25	45					100%
Feature G:	30	30	40					100%

References

Anon (n.d.) Apache Spark Reviews, *https://www.peerspot.com/products/apache-spark-reviews*.

Anon (n.d.) Jakarta EE alternatives and competitors, *https://www.peerspot.com/products/jakarta-ee-alternatives-and-competitors*.

Anon (n.d.) Jakarta EE vs Spring MVC, *https://www.peerspot.com/products/comparisons/jakarta-ee_vs_spring-mvc*.

Anon (n.d.) Spring Boot Reviews, *https://www.peerspot.com/products/spring-boot-reviews*.