Abstract

In past the performance of the employees was done by writing reports and submitting it to HR for each employee, which was a very tedious process. Every manager will have to follow the same process and it would waste their time as they cannot focus on more important work. HR will also face many difficulties as he will have to keep track of all the reports for each employee. And it becomes very inefficient for the top level to measure the performance for every employee.

To solve all these issues PMS (Performance Management System) was developed. Every employee including managers will be a part of it and all of their performance will be tracked during each iteration. Using some predefined methods and set of algorithms the employee will be evaluated.

We have implemented microservice architecture here, the PMS is divided into small individual services. Each and every service can work on its own and is not dependent on any other service. So even if one of the services is down it won't affect the whole system. And the services could be updated without any shutdown of the system.

So, this solves most of the issues that were there as managers will just have to comment and give some ratings against the work done by the employee and HR can get any information by some clicks only. The top level can easily measure the performance by simply looking at the ratings of the employee.

List of Figures

Sr. No	Title	Page No.
Fig. 1.3	Organization Chart	5
Fig. 2.2	Technical Specification	8
Fig. 4.5.1	Use Case Diagram	22
Fig. 4.5.2.1	Continuous Evaluation	24
Fig. 4.5.2.2	Evaluation	24
Fig. 4.5.3.1	Sequence Diagram	25
Fig. 4.5.3.2	Sequence Diagram	26
Fig. 4.5.3.3	Sequence Diagram	26
Fig. 4.5.4	Class Diagram	27
Fig. 4.5.5	E-R Diagram	28
Fig. 5.2	Database Tables	32
Fig. 5.3.1	Login Screen	33
Fig. 5.3.2	Home Screen of HR	33
Fig. 5.3.3.2	Create Cycle	34
Fig. 5.3.4	Edit Cycle	34
Fig. 5.3.5.1	Delete Cycle	35
Fig. 5.3.5.2	Delete Cycle	35
Fig. 5.3.6	Create Iteration	36
Fig. 5.3.7	Edit Iteration	36
Fig. 5.3.8.1	Delete Iteration	36
Fig. 5.3.8.2	Delete Iteration	37
Fig. 5.3.9.1	Self-Evaluation	37
Fig. 5.3.9.2	Self-Evaluation	38
Fig. 5.3.9.3	Self-Evaluation	39
Fig. 5.3.9.4	Self-Evaluation	39

List of Figures

Fig. 6.2.1	Results	40
Fig. 6.2.2	Results	41
Fig. 6.2.3	Results	41
Fig. 6.2.4	Results	41
Fig. 6.2.5	Results	42

List of Tables

Sr. No.	Title	Page No.
Table 3.6	Internship Scheduling	12
Table 7.2.1	Login	45
Table 7.2.2	Create Cycle	46
Table 7.2.3	Integration Testing	47
Table 7.2.4	Create Iteration	48

TABLE OF CONTENT

Sr. No.	Title	Page No.
	Acknowledgement	i
	Abstract	ii
	List Of Figures	iii
	List Of Tables	iv
	Table Of Contents	V
1	**Overview Of The Company**	1
	1.1 History	1
	1.2 Different Product	3
	1.3 Organization Chart	5
2	**Overview Of Different Department**	6
	2.1 Detailed Department	6
	2.2 Technical Specification	7
	2.3 Sequence Operations	9
3	**Introduction To Internship**	10
	3.1 Internship Summary	10
	3.2 Purpose	10
	3.3 Objective	10
	3.4 Scope	11
	3.5 Technology And Literature Review	11
	3.6 Internship / Project Scheduling	12
4	**System Analysis**	16
	4.1 Study Of Current System	16
	4.2 Problem And Weakness Of Current System	16
	4.3 Requirement Of New System	16
	4.4 System Feasibility	20

TABLE OF CONTENT

	4.5	Process In New System	21
	4.6	Features Of The System	29
	4.7	Main Modules	29
	4.8	Selection Of Software	30
5	**System Design**		31
	5.1	System Design & Methodology	31
	5.2	Database Design	32
	5.3	Interface Design	33
6	**Implementation**		40
	6.1	Implementation Environment	40
	6.2	Results	41
7	**Testing**		43
	7.1	Testing Phase	43
	7.2	Test Results & Analysis	44
8	**Conclusion And Discussion**		49
	8.1	Overall Analysis Of Internship	49
	8.2	Summary Of Internship	50
	8.4	Future Enhancement	50

1. OVERVIEW OF COMPANY

1.1. HISTORY

- Established in 2009, Azilen Technologies started with its own product in the field of Hospitality at global level. After gaining a lot of success and gaining complete product development experience, in 2011 Azilen started a separate division with a passion for consulting Start-ups, SMBs and Enterprise to build, support and transform a global product offering best consumer “experience”.
- At Azilen we believe in philosophy of “Adding Value” to our People’s Growth, Customer’s Growth and Growth in Adopting New Technology & Innovation which enable us to create and sustain world class experience and driving industries like never before.
- At Azilen we provide solutions in different kinds of Industries, Services and Business types. Its main aim is to provide a solution with Innovation, Best of Strategy and Adding Value to the product.
- And we provide solutions in Product Engineering, Digital Enterprise.
- People and Passion are the two strong pillars, which drives the success for Azilen.
- Our PEOPLE are not just our headcount, but our biggest assets and we believe in appreciating the smallest contributions, celebrating every small win, and building inspiring stories every day.
- The PASSION, our people bring along is our very secret ingredient to all the prominent solutions we have brought on the table.
- The diverse team at Azilen signifies a celebration of vibrant cultures, traditions, festivities, and our motto of respect for all.
- We believe that all work and no play make you a dull professional. Our office premises & monthly activities intertwined the culture of team spirit and healthy competition through sports. The generous appreciation, rewards & recognition, and transparency are what fuels the culture of excellence at Azilen

1.2 DIFFERENT PRODUCT

- **Product Consulting**

  - Strategic innovation with a blend of technology is all, that is needed for your business to be on its peak.
  - Azilen Consulting Expertise in business consulting, technology consulting Hassle-free and flexible at the same time, which Enhancing your processes through our expertise and Leverage our technology expertise to fuel your product's growth.
  - Azilen Team Delivering Excellence by Brainstorming Workshops, Strategic Analysis, Mind-Maps, Innovation Landscape, Key Initial Visualizations, SWOTS, Competition Analysis and Comparative Study.

- **UX Engineering**

  - Place where creativity meets technology. An experience carved with engaging customer journey
  - A great design is a combination of look and feel. The design approach and best practices we follow take into account, the Look: with a User Interface (UI) that's flawless | Feel: with seamless User Experience (UX). The outcome is a marvelous design driving a seamless customer journey.
  - Azilen pro experience approach achieved by Mind-Mapping, White-Boarding, Wire framing, Visual Designs.
  - At Azilen tools we use are InVision Studio, Adobe XD, Sketch, Figma, Photoshop, illustrator, Zaplin etc.

- **Architecture Engineering**

  - Strong technical foundation, flexible and adaptable to ever changing business needs.

  - A curated strategy, future ready for your product's foundations & vision.
  - We adapt to architecture model that is flexible enough to adapt to agile environment with changing technology landscape and business need.
  - Our Architecture Approach by Architecture Audit & Assessment, Architecture Planning & Execution, Tailored Implementation Strategy.

  - With these services we aim to deliver the best in class architecture design Scalable architecture, Scalable architecture, Scalable architecture, Scalable architecture, Scalable architecture are Expected.

- **Types of Architecture.**
  - Microservices Architecture
  - Event-driven Architecture
  - Object-oriented Architecture
  - Micro-kernel Architecture
  - Layered (n-tier) Architecture

- **Software Development**

  - Technology enablement for sustainable product development.

  - Software Development Competence. Our software development services are aimed at designing, engineering, supporting, and evolving enterprise software products. We manage, monitor and control these product lifecycles with our proven processes and cross functional teams blended in a collaborative development approach.

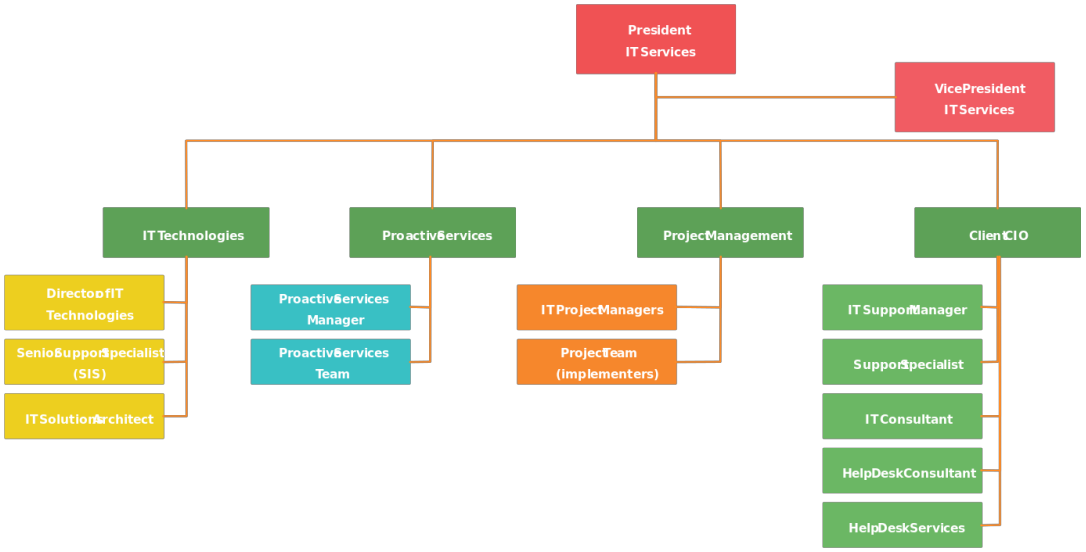
  - **Front End Development**

    - Micro front ends & atomic design
    - PWA (Progressive Web Apps)
    - Responsive interfaces

  - **Backend Development**

    - API and web service development
    - Smooth front-end synchronization
    - Data Engineering: Database migration & integration | DB administration.

1.3 ORGANIZATION CHART



[Fig. 1.3 Organization Chart]

2. OVERVIEW OF DIFFERENT DEPARTMENT

2.1 DETAILED DEPARTMENT

2.1.1 Software Development

- Front End Development
- Backend Development
- Mobile Development
- Cloud Engineering
- Software Integration.
- Support & Enhancement.

2.1.2 DevOps Implementation

- Automated CI / CD pipeline
- Managed cloud services
- IT infrastructure management
- DevSecOps - Security integrated
- Infrastructure as Code (IAC)
- DevOps maturity model
- ASR considerations: Deployability | Modifiability | Testability | Monitor-ability

2.1.3 Cloud Engineering

- Cloud Consulting
- Architecture Design
- Cloud Security &Testing
- Cloud Deployment
- Cloud Migration
- Maintenance and Support
-

2.1.4 Artificial Intelligence

- Enterprise AI Consulting.
  - o Strategy Formulation & Process Modelling
  - o AI Readiness & Maturity Assessment

- Data Understanding & Discovery
- Pilot Development
- Execution Roadmap

- Natural Language Processing

  - Natural Language Understanding
  - Contextual Information Analysis
  - Sentiment & Syntactic Analysis
  - Topic Discovery & Modelling
  - Intent Classification
  - Intelligent Chat Bots
  - Audio Analyzer and Speech Insights

- Data Engineering
  - Machine Learning
  - Data Mining
  - Deep Learning

- Computer Vision
  - Object Detection & Tracking
  - Image & Video Processing
  - Facial Recognition
  - Optical Design Modellin

2.1.5 Internet of Things

- IoT Consulting and Audit
- IoT Engineering
- IoT Insights
- IoT Managed Services

2.2 TECHNICAL SPECIFICATION

[Fig 2.2 Technical Specification]

2.2.3 Software Development

- Frontend Development: ReactJS, AngularJS, VueJS, UI/UX design & development, NextJS.
- Backend Development: Python (Flask, Django, FastAPI), NodeJS, .Net, ROR.
- Mobile Development: Android, Flutter
- AI/ML, Chatbot development, IOT, Blockchain AR/VR

2.2.4 Cloud Development

- AWS, Google Cloud Platform, Microsoft Azure.

2.2.5 DevOps

- Heroku, AWS, CI/CD, Kubernetes, Jenkins, Jira, BitBucket, SourceTree etc..

2.3 SEQUENCE OPERATIONS

- At Azilen we are use various SDLC model as per project requirement, andclient's requirement. Some of SDLC models are as bellows.
  - Waterfall
  - V-model (Validation and Verification model)
  - Incremental and Iterative model
  - Spiral model
  - The Rational Unified Process (RUP)

- Development using Agile is costly used in Azilen Technologies. Agile Methodology follows below step to develop complete product
  - Requirement Gathering
  - Design Solution
  - Develop Designed Solution
  - Test Solution
  - Deploy Solution

- All these steps are repeated until final product has been developed. To achieve agile model, we need to use some of these methods.
  - Scrum
  - Crystal
  - Dynamic Software Development Method (DSDM)
  - Feature Driven Development (FDD)
  - Lean Software Development
  - Extreme Programming (XP)

3. INTRODUCTION TO INTERNSHIP

3.1 INTERNSHIP SUMMARY

- During the Internship I have learned about the Multiple Technologies and I have also created one project Performance Management System Using That Technologies.
- These Are The Various Technologies That I Have Learned During Internship For Frontend: Angular Framework, HTML and CSS, Typescript , Bootstrap For Backend Java, Spring Boot And For Database : Mongo Db, For API Testing Postman is used. Also 3rd party services like Eureka, Zuul API, Keycloak etc.. are used as per requirement.

3.2 PURPOSE

- Performance Management System (PMS) is a web-based application which will be used by the organizations to evaluate their employee's performance from the work they have done in their respective projects.
- Generally, in every organization, there exists a procedure to measure the performance of each employee based on certain predefined set of parameters.
- This tool will help the organization to automate this evaluation process with ease.

3.3 OBJECTIVE

- Evaluate Performance: Main objective of this system is to evaluate performance of employees and to get proper and unbiased feedback of employees.
- Ease of work: It saves time and effort of the HR and Managers as most of the work will be done by using clicks only.
- Automated Work: This system helps in automate the process of evaluation of employees.

3.4 Scope

- The manager should be able to set a list of expected responsibilities for an employee based on the designation of the employees.
- The system should be able to set various parameters and corresponding ratings against which employees can be evaluated.
- The system should be able to set various parameters and corresponding ratings against which employees can be evaluated.
- The employees should be evaluated and rated against these responsibilities by different level of users based on their efforts and contribution for their respective projects.
- Finally, the system should be able to generate and calculate overall rating based on these parameters.

3.5 TECHNOLOGY AND LITERATURE REVIEW

- **Eureka service discovery**
    - The main use of using service discovery is that it helps in locating the services.
    - We have used Eureka service discovery as our project is based on micro services and their REST is used and Eureka supports REST.
    - In this, every micro services when started will come and register themselves by stating their Ip address and port number, so when a client requests for access using API at that time API will come to service discovery to get the IP address and port number of the particular service. It works like a Phonebook.

- **Zuul API Gateway and CORS**
    - Zuul works like an API gateway, it redirects the requests that come from UI to the appropriate microservices. So basically, the client can contact to Zuul only and then Zuul will fulfil the request by transferring the API to its microservice.

- The main advantage we get is that we will fulfil some important aspects such as CORS, authentication and security. In our system two authentication is done i.e. one by keycloak and another by Zuul.
- CORS is Cross Origin Resource Sharing. It is an internet protocol which lets us share/transfer data or resources which are restricted from one.
- domain to another. We need this as Angular is in different domain and spring is in different domain, so to let them share resources we use this.
- The other advantage of using Zuul is that it does load balancing, there is an internal ribbon server inside Zuul API Gateway that acts as the load balancer.

- **Keycloak**
  - We used keycloak to fulfil multiple functionalities that is to successfully implement the LDAP (Lightweight Directory Access Protocol) login, which is a protocol for global login. In simple words it means we can use a same set of credentials for logging into multiple applications.
  - It is implemented to bifurcate the users as per their roles and permission. So, for example an employee won't get access to all functionalities of the system whereas the HR will get access to all the functionality. So as per the roles and permission of the user the keycloak will give access of information.

3.6 INTERNSHIP / PROJECT SCHEDULING

Week 1	09-1-23 To 13-01-23	o OOPs Basics. o Creative Thinking. o Programming basics (Frontend). o UX Basics
Week 2	16-01-23 To 20-01-23	o Creative Thinking. o Programming basics (Backend). o QA Basics. o Unix Basics. o DevOps Basics. o Working in a teams

Week 3	23-01-23 To 27-01-23	o Introduction to Wireframes. o Introduction to Whiteboard activity. o Requirement Gathering. o High level design.
Week 4	30-01-23 To 03-02-23	o Introduction to Performance Management System. o Wireframe Activity on PMS System. o User Story Mapping on PMS System. o UML Diagram. o Basics of API. o Basics of Design Pattern.
Week 5	06-02-23 To 10-02-23	o Low Level Design. o Design Pattern for PMS system. o Class diagram for PMS system. o Sequence diagram for PMS system. o Basics cloud computing. o Basics of server architecture.
Week 6	13-02-23 To 17-02-23	o Basics of DBMS. o Clean code introduction. o Version control introduction. o In house project code review.
Week 7	20-02-23 To 24-02-23	o Design pattern Tactical DDD. o Design pattern CQRS. o CSR and SSR. o Cloud computing.
Week 8	27-02-23 To 03-03-23	o 1. Presentation of PMS. o Changes in wireframes. o Changes in HLD.

Week 9	06-03-23 To 10-03-23	o Requirement Gathering for Login Functionality. o Coding For HR Login Functionality. o Coding For Employee Login Functionality. o Coding For Reporting Manager / Project Manager Login Functionality.
Week 10	13-03-23 To 17-03-23	o Requirement Gathering For Profile Management. o Code Review of Login Functionality. o Clean Code for Login Functionality. o Coding Profile Management.
Week 11	20-03-23 To 24-03-23	o Requirement Gathering For Cycle Management. o Explore Cycle Information. o Types Of Cycle And In Which Who Is Include Understanding. o Coding For Cycle Management.
Week 12	27-03-23 To 31-03-23	o Requirement Gathering For Cycle Management. o Types Of Cycle And In Which Who Is Include Understanding o Requirement Gathering For Iteration. o Learn about Self Evaluation. o Requirement Gathering Rating Grade. o Coding For Created Cycle and Delete Cycle

Week 13 03-04-23 To 07-04-23	o Requirement Gathering For Keycloak Management. o Testing the application. o Presenting this project to our BU head, taking their feedbacks and moving further as per their requirements.
Week 14,15 10-04-23 To 21-04-23	o Preparing the documentation for the Performance Management System o Final Testing Phase of the system and error solving and gathering.

[Table 3.6 Internship Scheduling]

4. SYSTEM ANALYSIS

4.1 STUDY OF CURRENT SYSTEM

- In current evaluation system all the process of maintaining data and evaluation process is done manually but by using PMS system it become easy and make automate the evaluation process of any employee become easy.

4.2 PROBLEM AND WEAKNESS OF CURRENT SYSTEM

- In current system the evaluation of any employee in organization is done by manually. So there are chances of bias and the evaluation of employee is not done proper way.
- Due to this it lead to loss to both organization and Employee. The Organization will suffer because if evaluation is not done properly the feedback given for employee might be some times wrong. Due to this good employee may leave company.

4.3 REQUIREMENT OF NEW SYSTEM

- Functional requirements :

1. **Login**
   - Description: This functionality provides users having different roles to login to the system. Depending on their role, they are redirected to appropriate pages.

HR Login :

- Precondition: User needs to be on the Login page.
- Input: User will enter valid username and password which should belong to HR.
- Output: User is redirected to HR Homepage.
- Processing: Username and password will be verified and should match to those in database.

Employee Login :

- Precondition: User needs to be on the Login page.
- Input: User will enter valid username and password which should belong to a student.
- Output: User is redirected to employee homepage.
- Processing: User username and password will be verified and should match those in database.

Project Manager/Reporting Manager Login :

- Precondition: User needs to be on the Login page.
- Input: User will enter valid username and password which should belong to an organization.
- Output: User is redirected to their homepage.
- Processing: User username and password will be verified and should match those of Database.

2. **Create Profile :**
   - Description: This functionality allows the HR to create employee profile and access the system.
   - Precondition: HR needs to be on the Create profile page.
   - Input: HR will enter the required details for all Employees.
   - Output: HR can go to manage profile that created profile for employees.
   - Processing: System will validate the details entered by HR and display Manage profile page.
3. **Manage Cycle :**
   - Description: This functionality allows an organization to manage cycle.

Add Employee:

- Precondition: User needs to be on the Manage Cycle Page.
- Input: User will select the Add Employee and provide the necessary details.

Assign Project Manager/Reporting manager:

- Precondition: User needs to be on the Manage Profile Page.
- Input: User will select and assign the new project manager.
- Output: New Project Manager will be added for particular that project.
- Processing: System will add the Project Manager to the employee.
- Output: New Employee will be added and will be available for the Project.
- Processing: System will validate the details entered for the Employee and if valid, a new employee will be added and available for project.

4. **Add KRA:**

- Description: This functionality allows the Project Manager to add KRAs for all employees.
- Precondition: Project Manager should be on the manage cycle page
- Input: Project Manager will add KRAs like communication skills, domain knowledge, etc.
- Output: Add the KRAs in employee profile.

5. **Give Comments**:

- Description: This functionality provides has two ways; one is self- evaluation and the other is without self-evaluation. If HR selects the self-evaluation then employees will give comments and in another case Project Manager will give comments.

a. **Self-evaluation :**

- **Precondition:** Employee needs to be on KRA Page.
- **Input: E**mployee will log in and give comment.
- **Output:** Save comment.

b. No-Self-evaluation:

- Precondition: Project Manager needs to be on the KRA Page.
- Input: Project Manager login and navigation to KRA page give comments.
- Output: Save comments.

6. Give Ratings :

- Description: This functionality provides the Project Manager to give ratings.
- **Precondition:** Project Manager needs to have completed evaluation of that employee.
- **Input:** Project Manager will login and go to give ratings.
- **Output:** Save that rating of employee.

- **Non-functional requirements :**

1. Reliability :

- A System shall be reliable i.e. in the case of server crashes a backup server will be there to work which will be maintained continuously.

2. Availability :

- This system is web based and it is in house production of an organization so it will always available of an organization through an internal link and only organization's members can access it.

3. Performance :

- This system is faster so users doesn't have to wait for loading it

4. Security :

- The system will be made secure by assigning all users with separate login ids and password i.e. every user will be responsible for their assigned credentials. Also, All the employee evaluation related data is encrypted before storing in the database for the protection of the data.

5. Accessibility :

- User will only be able to access this web-based system through browser only. It will be access through an internal link of an organization and the members of an organization can access it.

4.4 SYSTEM FEASIBILITY

- A feasibility study is performed by a company when they want to know whether the proposed framework is possible or not.

Operational feasibility

- This system brings better achievement for the operations by providing unbiased and accurate evaluation.
- The system will help ease the work of the HR and Manager. The problem of keeping record will also be sorted as all the old evaluations will be there.
- Creating and retrieving output is much easier.
- Issue of security won't be there as only HR and the Project Manager will know what evaluation is given to the employee.

Technical feasibility

- Here we have used postman for developing the APIs as it has better interface and easy to use. We used to write the angular code in visual studio and use mongoDB for the database.
- Knowledge of java and angular is essential to carry out this project.

Financial and Economical feasibility

- Yes, this feasibility possible as mainly it's an in-house project which will owned by the company.
- There will be some initial cost but later on we can sell it in the market also and generate revenue also.

- And the benefits from this system will surely outweigh the cost that is to be put in creating this system.
- The overall costs will mostly revolve around the buying of the services required for the system that's it.

4.5 PROCESS IN NEW SYSTEM

4.5.1 Use case Diagram

- A use case diagram is used to represent the dynamic behavior of a system.
- It encapsulates the system's functionality by incorporating use cases, actors, and their relationships. It models the tasks, services, and functions required by a system/subsystem of an application. It depicts the high-level functionality of a system and also tells how the user handles a system.
- The main purpose of a use case diagram is to portray the dynamic aspect of a system. It accumulates the system's requirement, which includes both internal as well as external influences. It invokes persons, use cases, and several things that invoke the actors and elements accountable for the implementation of use case diagrams. It represents how an entity from the external environment can interact with a part of the system.
- It is essential to analyse the whole system before starting with drawing a use case diagram, and then the system's functionalities are found. And once every single functionality is identified, they are then transformed into the use cases to be used in the use case diagram.
- After that, we will enlist the actors that will interact with the system. The actors are the person or a thing that invokes the functionality of a system. It may be a system or a private entity, such that it requires an entity to be pertinent to the functionalities of the system to which it is going to interact.

[Fig. 4.5.1. Use Case Diagram]

4.5.2 Activity Diagram

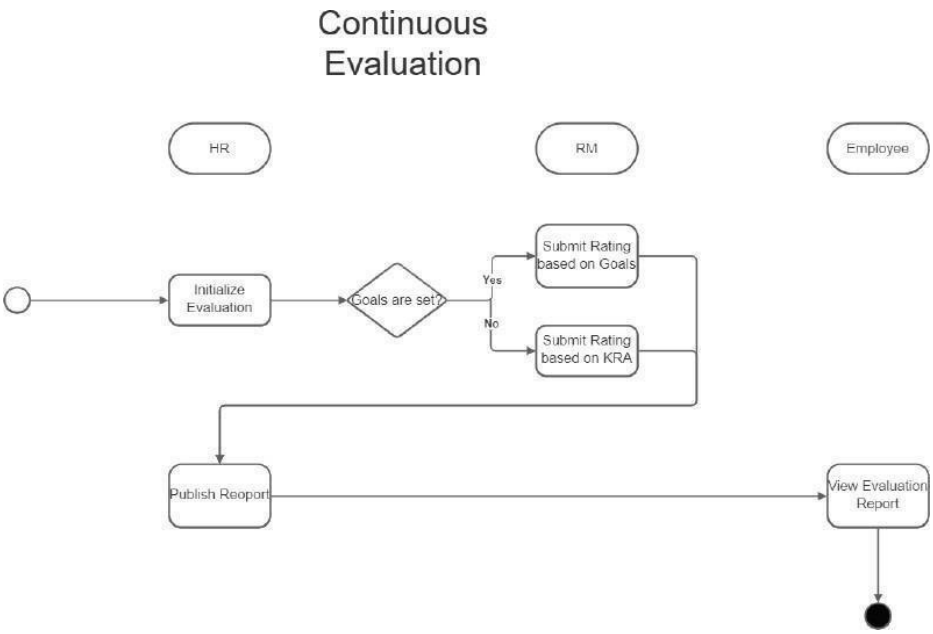
- The basic purposes of activity diagrams is similar to other four diagrams. It captures the dynamic behavior of the system. Other four diagrams are used to show the message flow from one object to another but activity diagram is used to show message flow from one activity to another.

- Activity is a particular operation of the system. Activity diagrams are not only used for visualizing the dynamic nature of a system, but they are also used to construct the executable system by using forward and reverse engineering techniques. The only missing thing in the activity diagram is the message part.

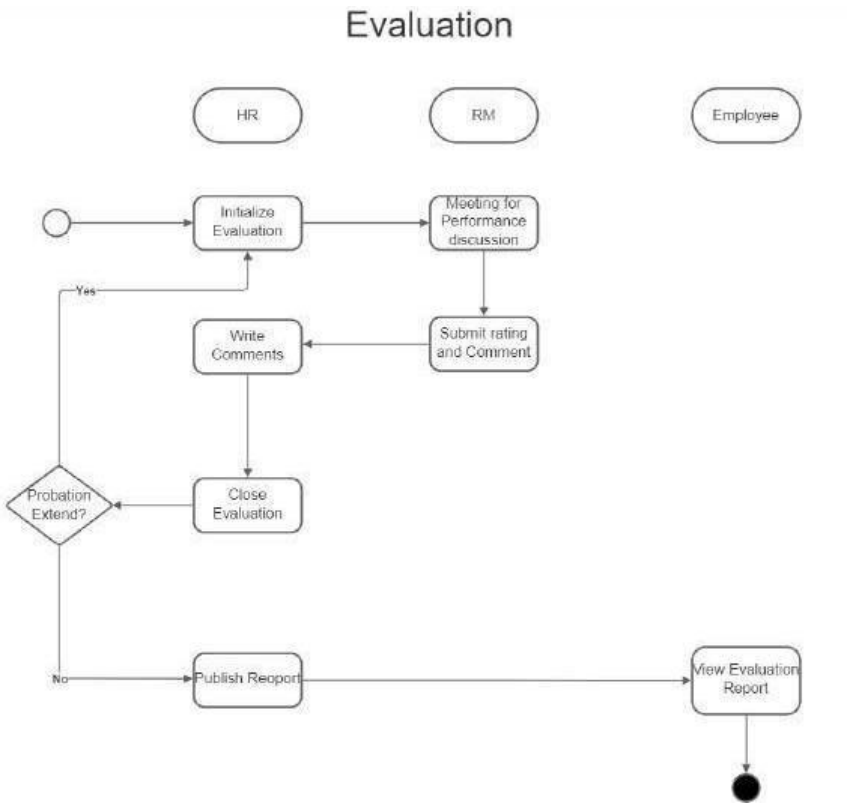
- It does not show any message flow from one activity to another. Activity diagram is sometimes considered as the flowchart. Although the diagrams look like a flowchart, they are not. It shows different flows such as parallel, branched, concurrent, and single.

- Activity diagrams are mainly used as a flowchart that consists of activities performed by the system. Activity diagrams are not exactly flowcharts as they have some additional capabilities. These additional capabilities include branching, parallel flow, swimlane, etc..

- Before drawing an activity diagram, we must have a clear understanding about the elements used in activity diagram. The main element of an activity diagram is the activity itself. An activity is a function performed by the system. After identifying the activities, we need to understand how they are associated with constraints and conditions.



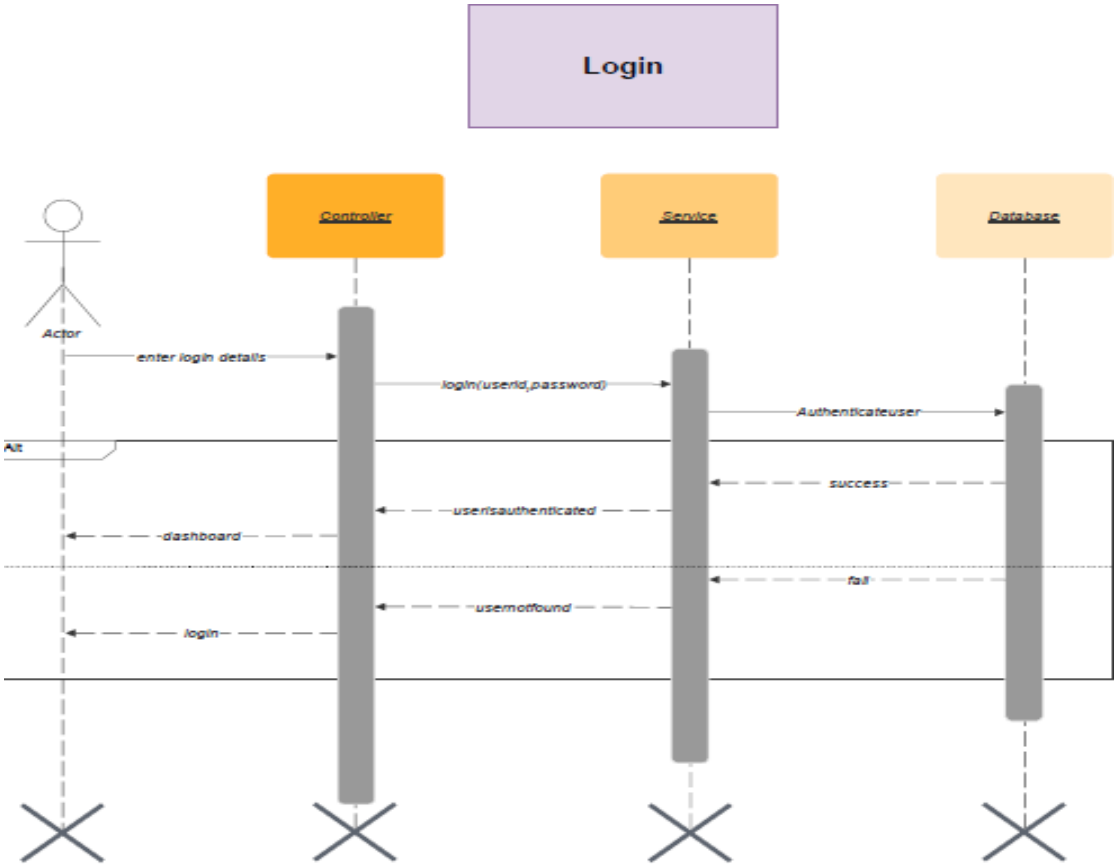
[Fig 4.5.2.1 Continuous Evaluation]



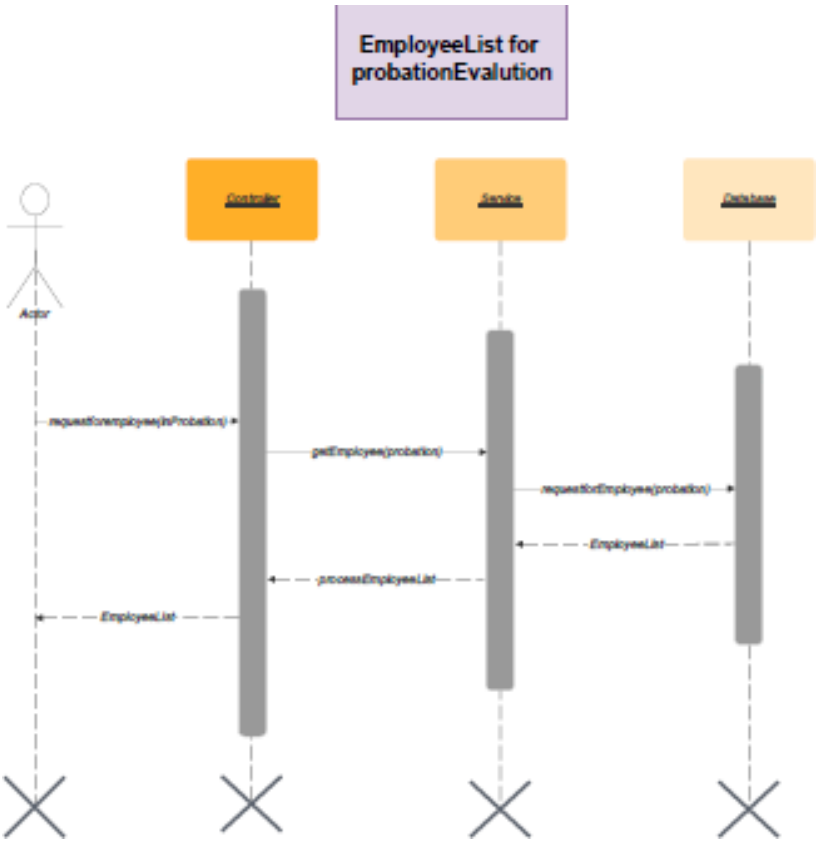
[Fig 4.5.2.2 Evaluation]

4.5.3 Sequence Diagram

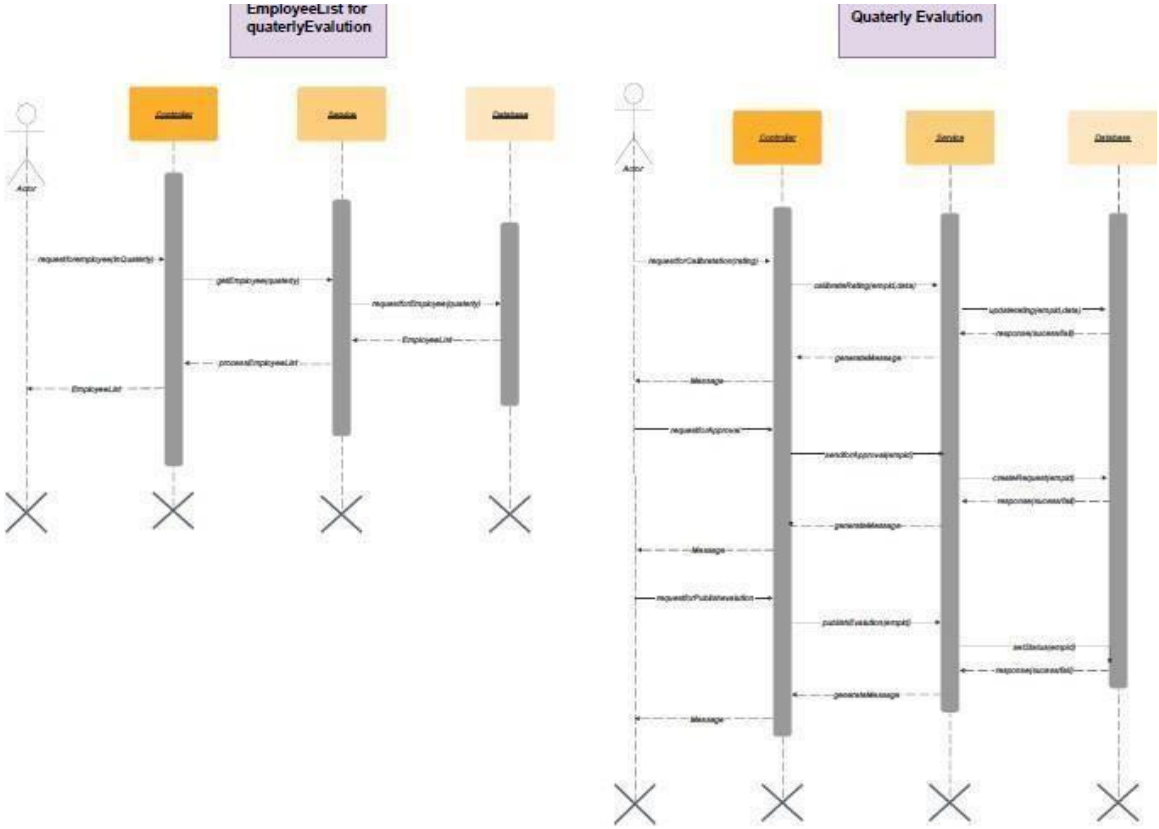
- UML sequence diagrams are interaction diagrams that detail how operations are carried out. They capture the interaction between objects in the context of a collaboration. Sequence diagrams are time focus and they show the order of the interaction visually by using the vertical axis of the diagram to represent time what messages are sent and when.
- The interaction that takes place in a collaboration that either realizes a use case or an operation (instance diagrams or generic diagrams)
- High-level interactions between user of the system and the system, between the system and other systems, or between subsystems (sometimes known as system sequence diagrams).



[Fig. 4.5.3.1 Sequence Diagram]



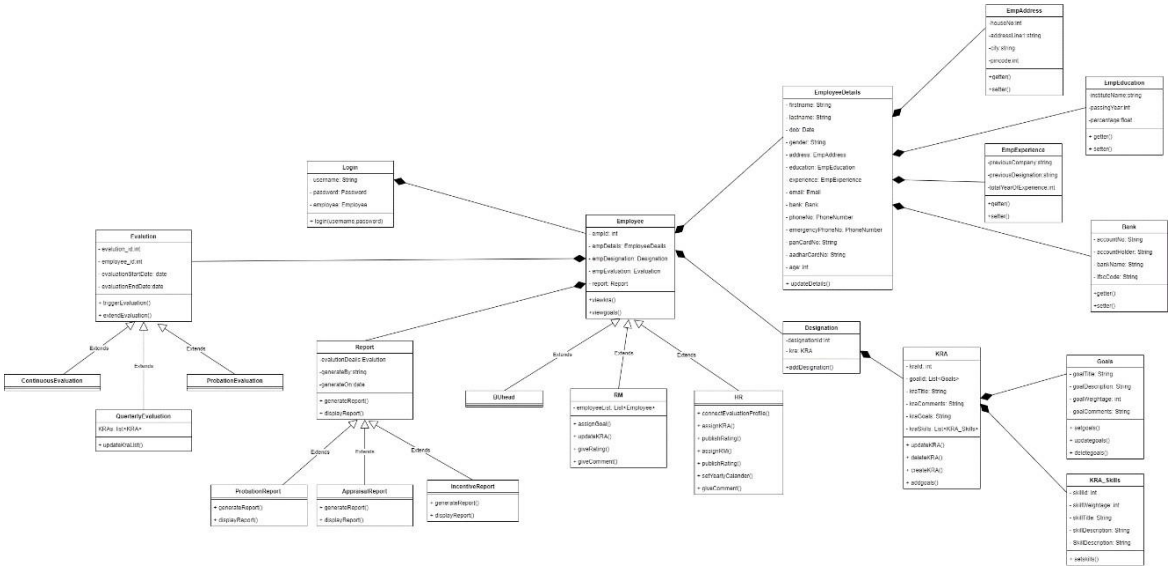
[Fig. 4.5.3.2 Sequence Diagram]



[Fig. 4.5.3.3 Sequence Diagram]

4.5.4 Class Diagram

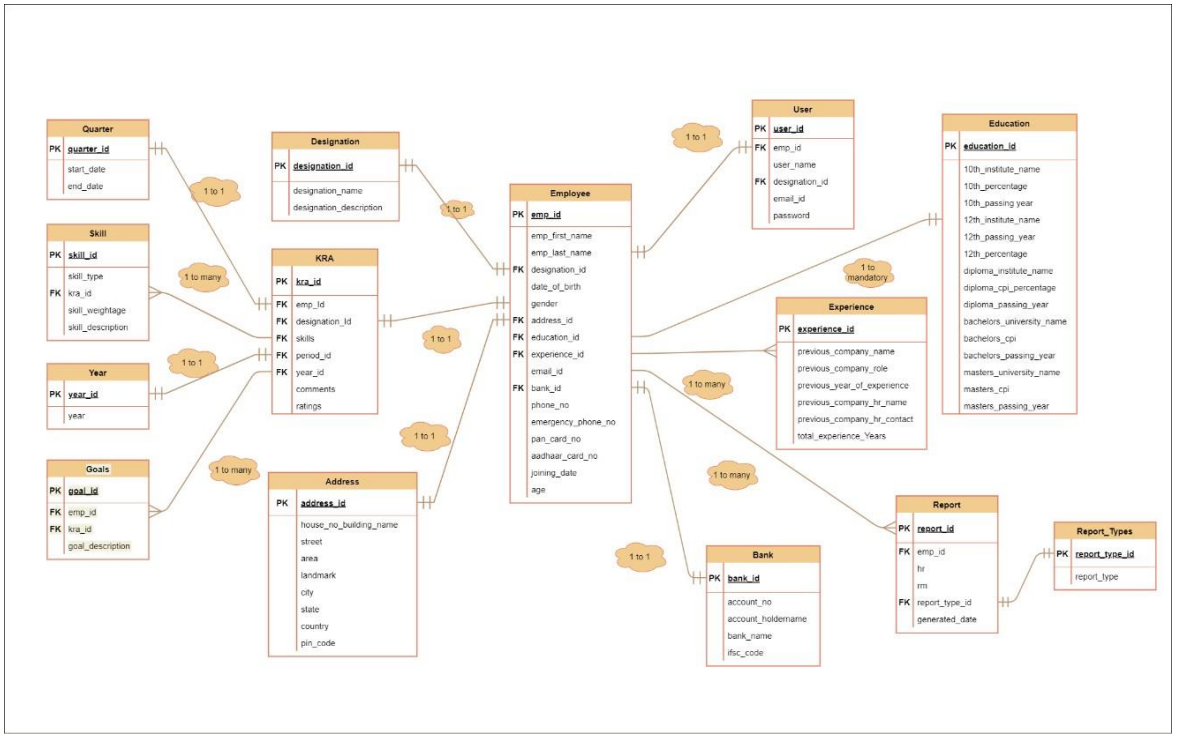
- Class diagram is a static diagram. It represents the static view of an application. Class diagram is not only used for visualizing, describing, and documenting different aspects of a system but also for constructing executable code of the software application.
- Class diagram describes the attributes and operations of a class and also the constraints imposed on the system. The class diagrams are widely used in the modelling of object-oriented systems because they are the only UML diagrams, which can be mapped directly with object-oriented languages.
- Class diagram shows a collection of classes, interfaces, associations, collaborations, and constraints. It is also known as a structural diagram.
- The purpose of class diagram is to model the static view of an application. Class diagrams are the only diagrams which can be directly mapped with object-oriented languages and thus widely used at the time of construction.



[Fig. 4.5.4 Class Diagram]

4.5.5 E-R Diagram

- An entity relationship diagram (ERD), also known as an entity relationship model, is a graphical representation that depicts relationships among people, objects, places, concepts or events within an information technology (IT) system. An ERD uses data modelling techniques that can help define business processes and serve as the foundation for a relational database.
- Entity relationship diagrams provide a visual starting point for database design that can also be used to help determine information system requirements throughout an organization. After a relational database is rolled out, an ERD can still serve as a reference point, should any debugging or business process re-engineering be needed later.
- However, while an ERD can be useful for organizing data that can be represented by a relational structure, it can't sufficiently represent semi-structured or unstructured data. It's also unlikely to be helpful on its own in integrating data into a pre-existing information system.



[Fig. 4.5.5 E-R Diagram]

4.6 FEATURES OF THE SYSTEM

- Its important for an organization to understand the benefits of Performance Management System, in the current global environment in which the market is growing very rapidly.
- So, management of employee's performance is the need of an organization as the employees are considers as an asset by the organization.
- The following are the various benefits of Performance Management System:
- In Micro service architecture each service is a deployment unit that is each service will have to be deployed separately. This adds to the advantage that suppose a change comes in a particular service then we only have to deploy the latest jar of that service. We won't have to deploy each service again. All these micro services communicate with each other using REST API.
- It motivates the employees to face new challenges and innovate through structure process.
- New opportunities are provided to employees for their development
  And growth in their careers.
- Employees will be able to provide good results due to the clarity on their performance goals.
- Employees who do not perform as expected or who is under performer, can be identified through performance evaluation and can increase their skills level. It determines learning needs through individual development plans or performance improvement plans.

4.7 MAIN MODULES

- Set KRA
- Employee On boarding
- Set Goals
- Evaluation of Probation Period.
- Give Ratings by HR,PM,DM
- Report Generation.

4.8 SELECTION OF SOFTWARE

- In this we have used IntellijIdea and Eclipse for coding purpose.
- For Testing API we have used Postman.
- For Test cases we have used Keyclock.

5. SYSTEM DESIGN

5.1 SYSTEM DESIGN & METHODOLOGY

- System design is the phase that bridges the gap between problem domain and the existing system in a manageable way. This phase focuses on the solution domain, i.e. “how to implement?”
- It is the phase where the SRS document is converted into a format that can be implemented and decides how the system will operate.
- In this phase, the complex activity of system development is divided into several smaller sub- activities, which coordinate with each other to achieve the main objective of system development. Inputs to System Design

1) System design takes the following inputs −
2) Statement of work
3) Requirement determination plan
4) Current situation analysis
5) Proposed system requirements including a conceptual data model, modified DFDs, and Metadata (data about data).

- Outputs for System Design

1) System design gives the following outputs −
2) Infrastructure and organizational changes for the proposed system.
3) A data schema, often a relational schema.
4) Metadata to define the tables/files and columns/data-items.
5) A function hierarchy diagram or web page map that graphically describes the program structure.
6) Actual or pseudocode for each module in the program.
7) A prototype for the proposed system.

5.2 DATABASE TABLES

Field Name	Datatype	Descriptions
_id	Int	Employee id
firstName	String	Employee first name
middleName	String	Employee middle name
lastName	String	Employee last name
Username	String	Employee username
Email	String	Employee's email id
Phone	Int	Employee's phone number
professionalDetail{}	Embedded document	It is a document which holds professional details.
Status	Enum	Status of the employee

[Fig 5.2.1 Database Table – Employee Table]

Field Name	Datatype	Descriptions
id	Int	Evaluation id
Name	String	Name of the Evaluation
Year	Int	Evaluation year
cycleType	String	Defines type of cycle
Active	Bool	Status of evaluation

[Fig 5.2.2 Database Table – Employee Iterarion Table]

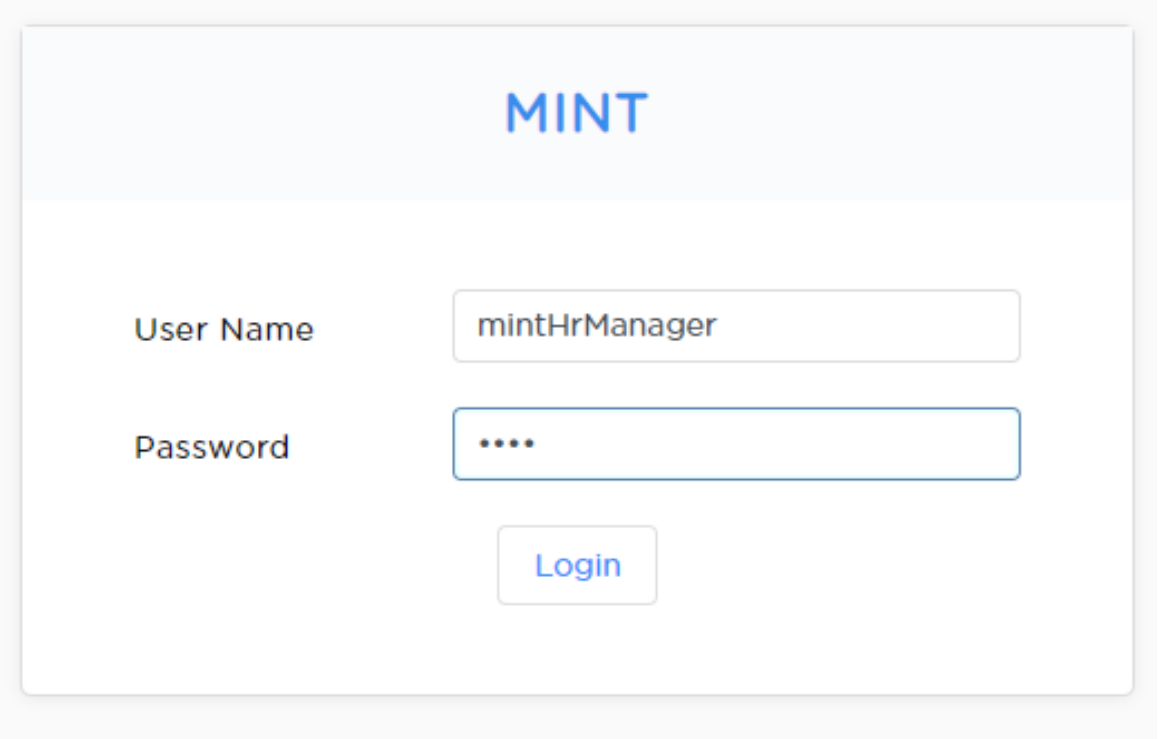
Field Names	Datatype	Description
_id	Int	PMS Evaluation id
empId	Int	Employee id
firstName	String	Employee first name
lastName	String	Employee last name
Email	String	Employee's email id
designation	String	Employee's Designation
cycleDetail{}	Embedded document	It has details of the cycle
iterationDetail{}	Embedded document	It has details of the iteration
kraList{}	Embedded document	It has all the fields related to KRA

[Fig 5.2.2 Database Table – Employee Evaluation Table]

5.3 INTERFACE DESIGN

5.3.1 Login

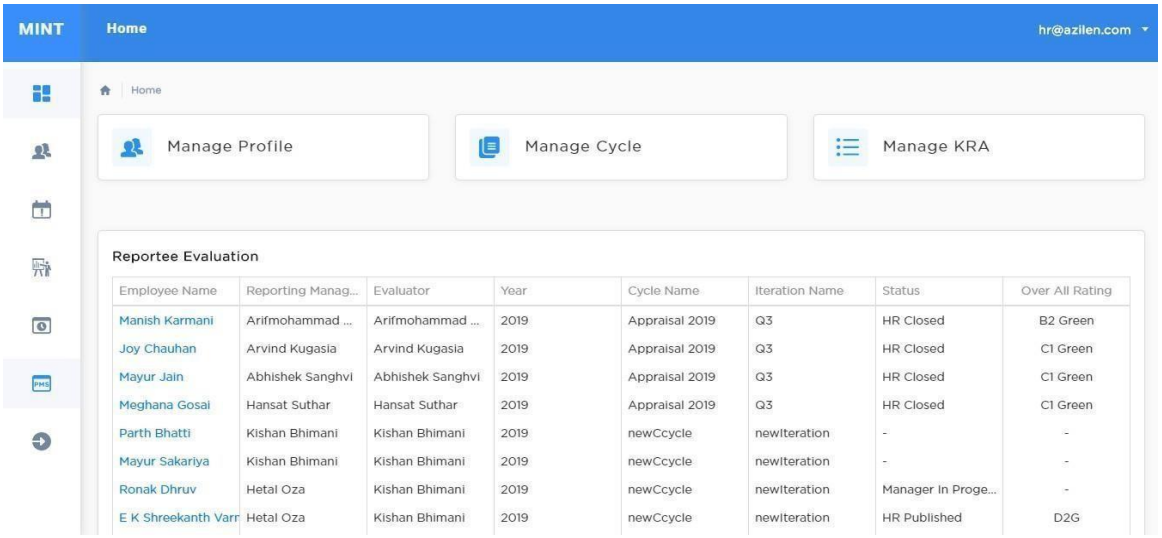
- Login Page, where username and password will be validated.



[**Fig 5.3.1 Login Screen – HR can login into the system**]

5.3.2 HOME SCREEN OF HR

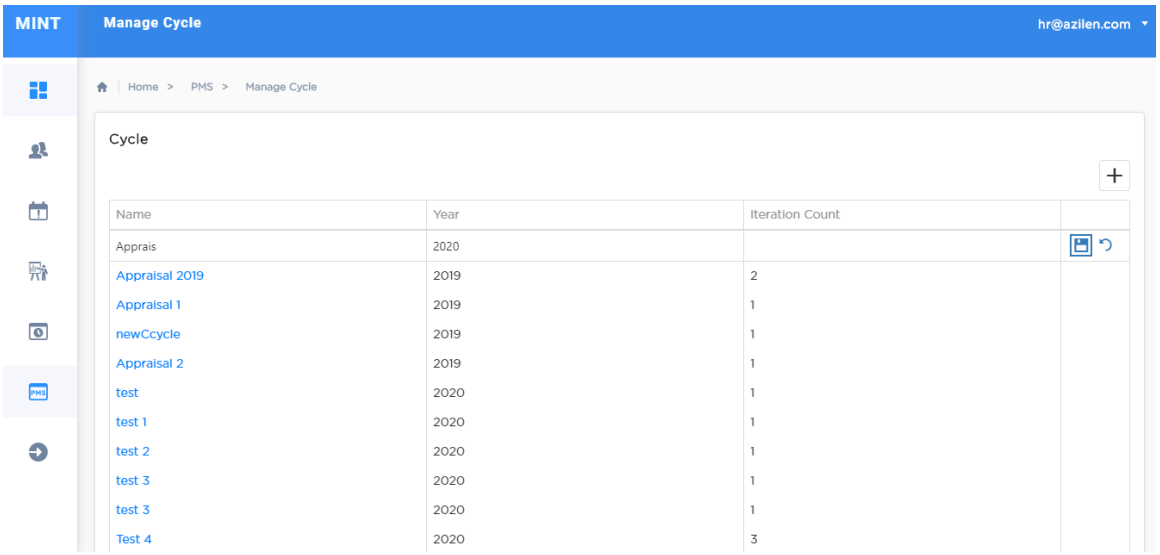
- After login page, the user will be redirected to their dashboard where, HR will be able to see the list employees of Repartee Evaluation.



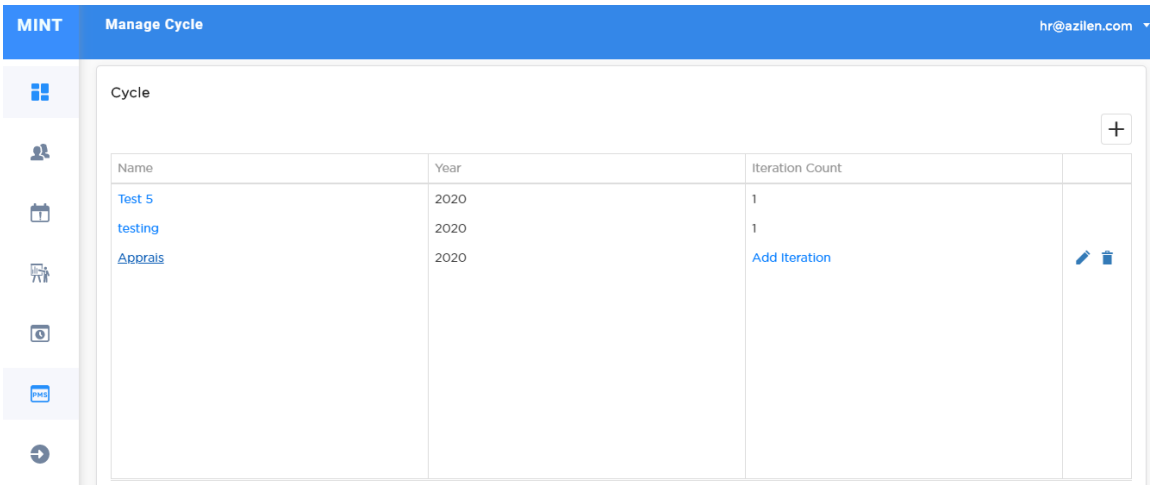
[**Fig. 5.3.2 Home Screen of HR- HR can see this screen whenever HR can login with valid Credentials**]

5.3.3 CREATE CYCLE

- After clicking on 'Manage Cycle', user will be redirected to this page. Now clicking on the '+' sign on the right side of screen will create a new cycle.



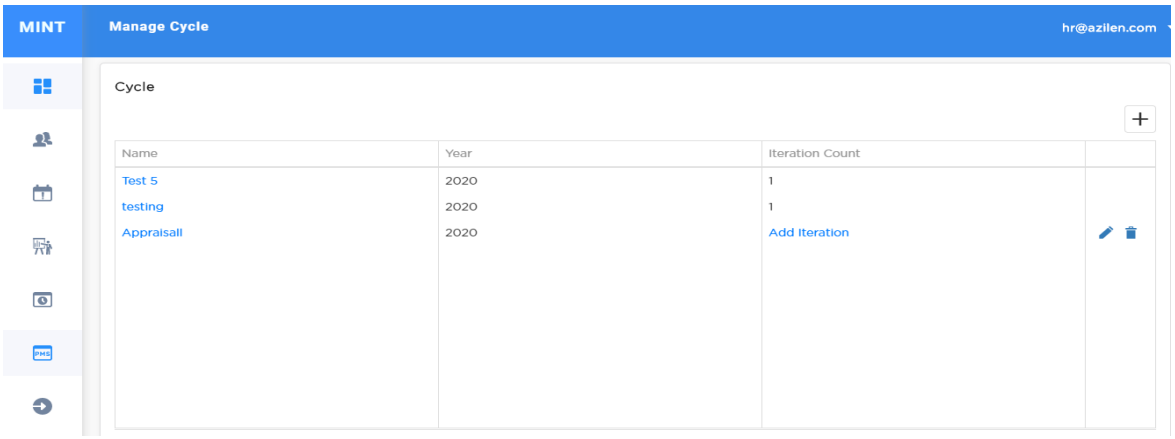
[Fig. 5.3.3.1 Create Cycle – HR create a cycle for a employee in this screen]



[Fig. 5.3.3.2 Create Cycle- Whenever HR click on the create new cycle]

5.3.4 Edit Cycle

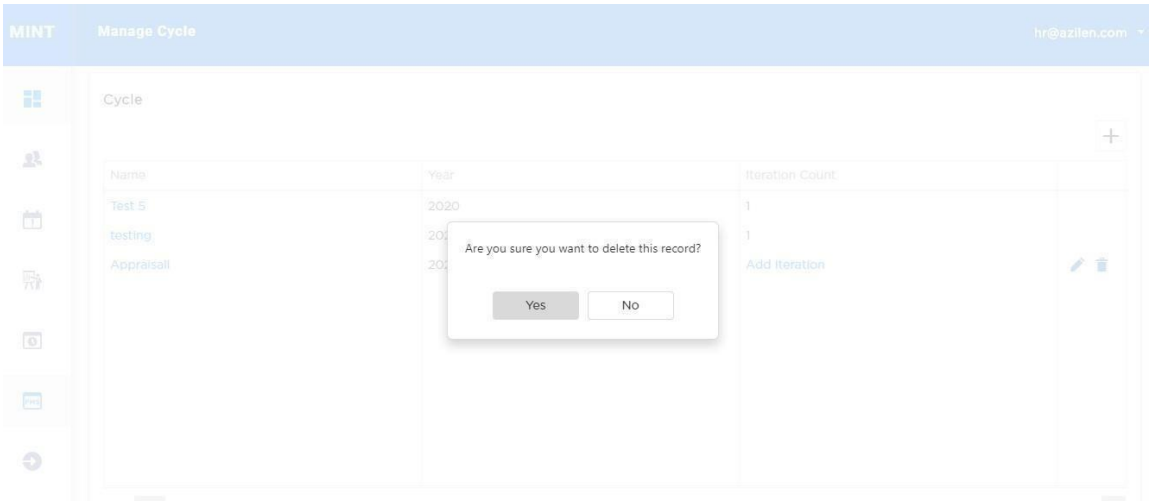
- There is an option of Edit (update) using which you can make changes in that.
- After clicking on Edit sign, cycle will become editable, so that you can makechanges.



[Fig. 5.3.4 Edit Cycle – HR can edit previously created cycle for employee]

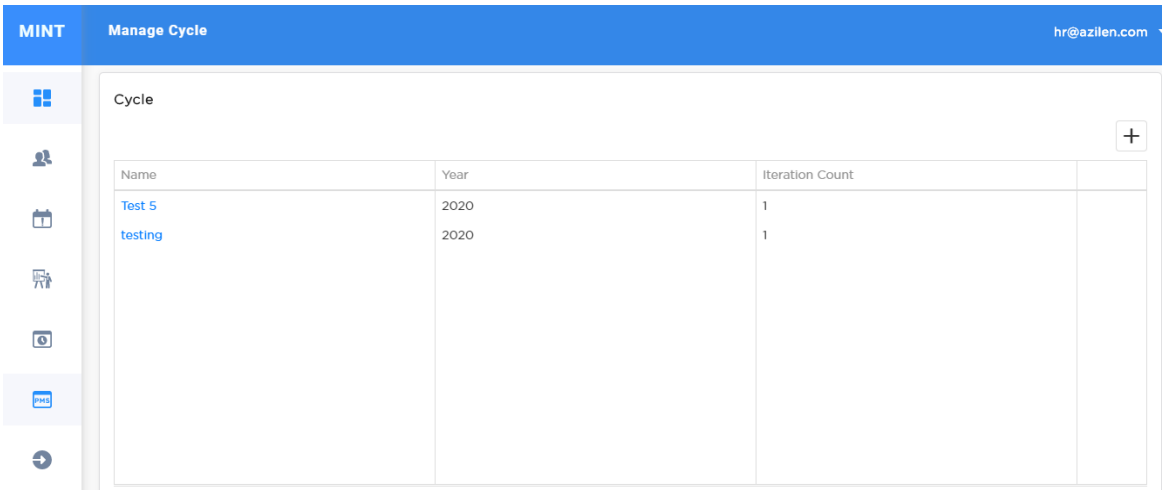
5.3.5 Delete Cycle

- By clicking on the Delete sign you can delete the cycle also.
- A pop-up will come confirming whether you want to delete or not.



[Fig. 5.3.5.1 Delete Cycle – HR delete unwanted cycle]

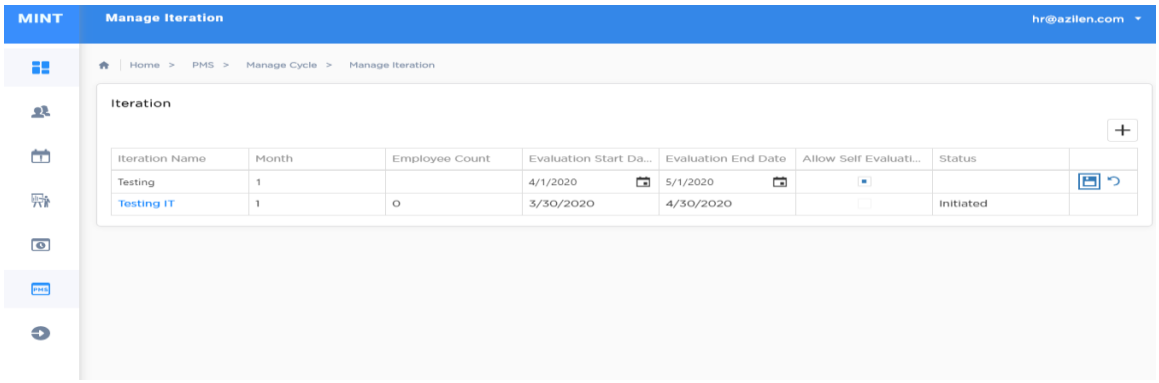
- If you click yes, then the changes will be reflected as seen below.



[Fig. 5.3.5.2 Delete Cycle – After HR will delete cycle , cycle will remove from list]

5.3.6 Create Iteration

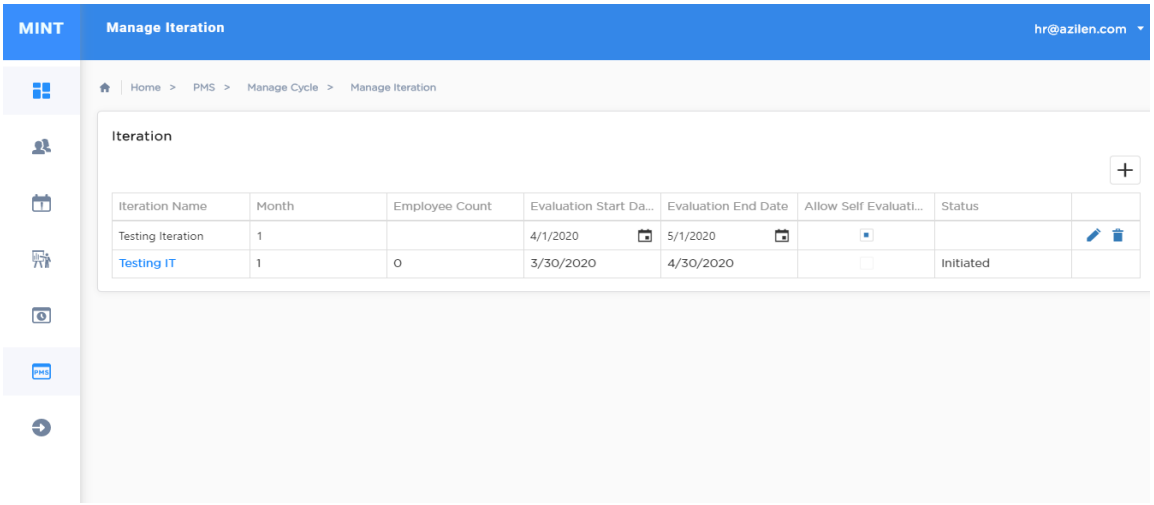
- By clicking on 'Add Iteration' user will be redirected to this page.



[Fig. 5.3.6 Create Iteration – HR can create Iteration from this screen]

5.3.7 Edit Iteration

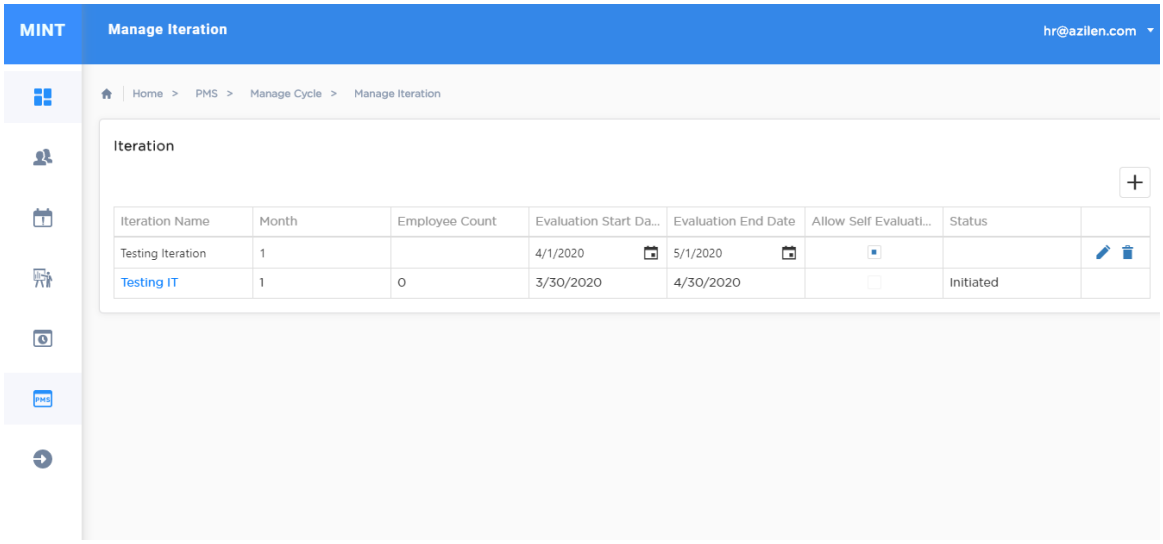
- User can edit the filed over here.



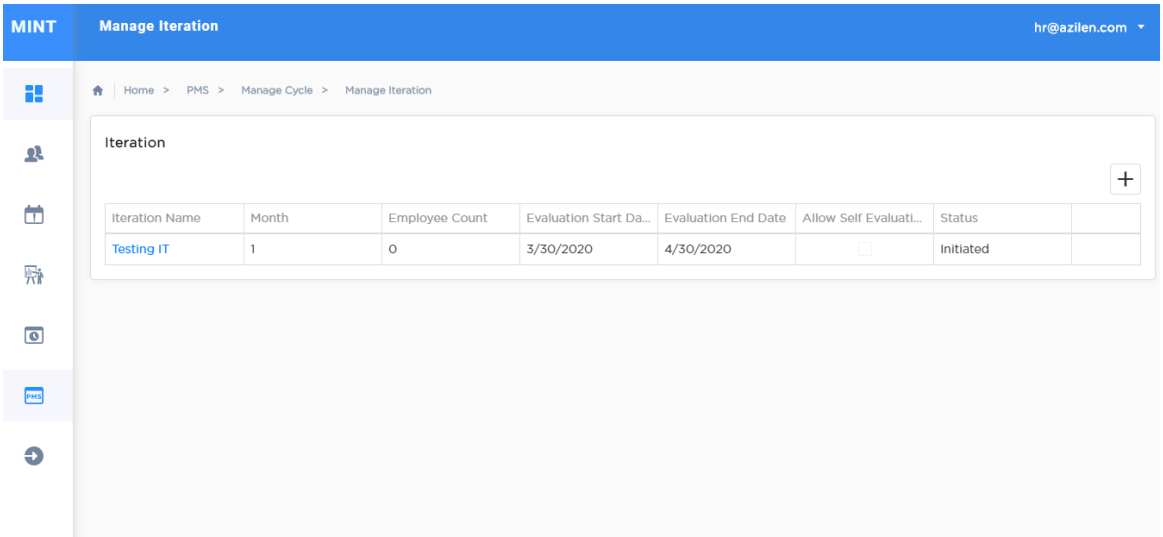
[Fig. 5.3.7 Edit Iteration – HR can edit previously created iteration]

5.3.8 Delete Iteration

- By clicking on the Delete sign you can delete the iteration also and below image shows that iteration created has been deleted.



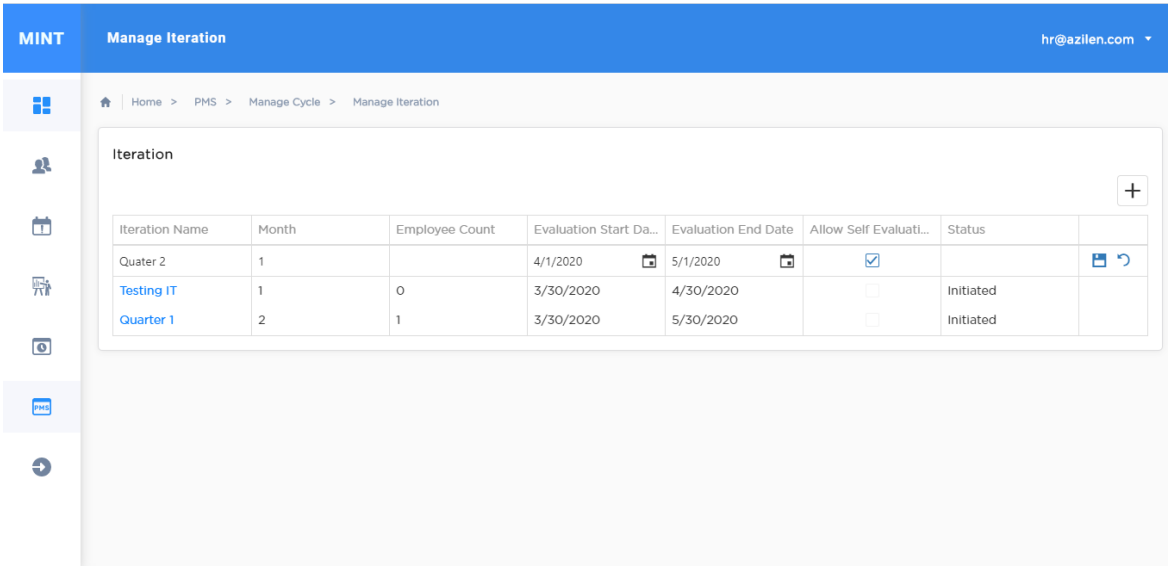
[Fig. 5.3.8.1 Delete Iteration – HR can delete iteration by clicking on delete]



[Fig. 5.3.8.2 Delete Iteration – Successfully deleted one iteration]

5.3.9 SELF-EVALUATION

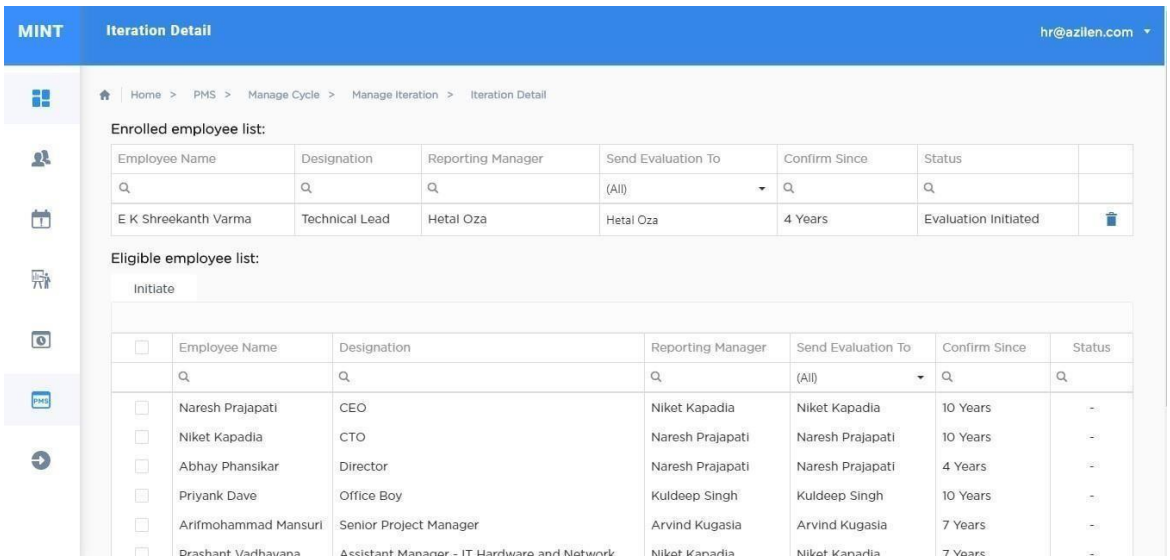
- In below image HR has added created new iteration.



- Iteration name: Quarter 2 and as checked “Allow Self Evaluation” checkbox.

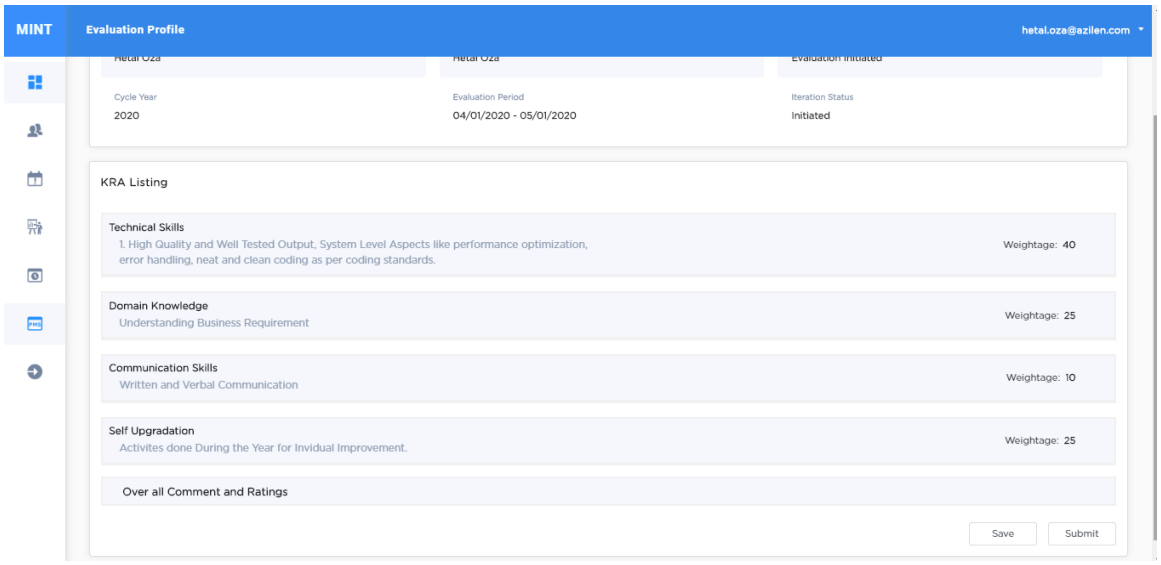
[Fig. 5.3.9.1 Self-Evaluation – HR can edit self evaluation section for every quarter]

- Here, employee is added in that iteration.



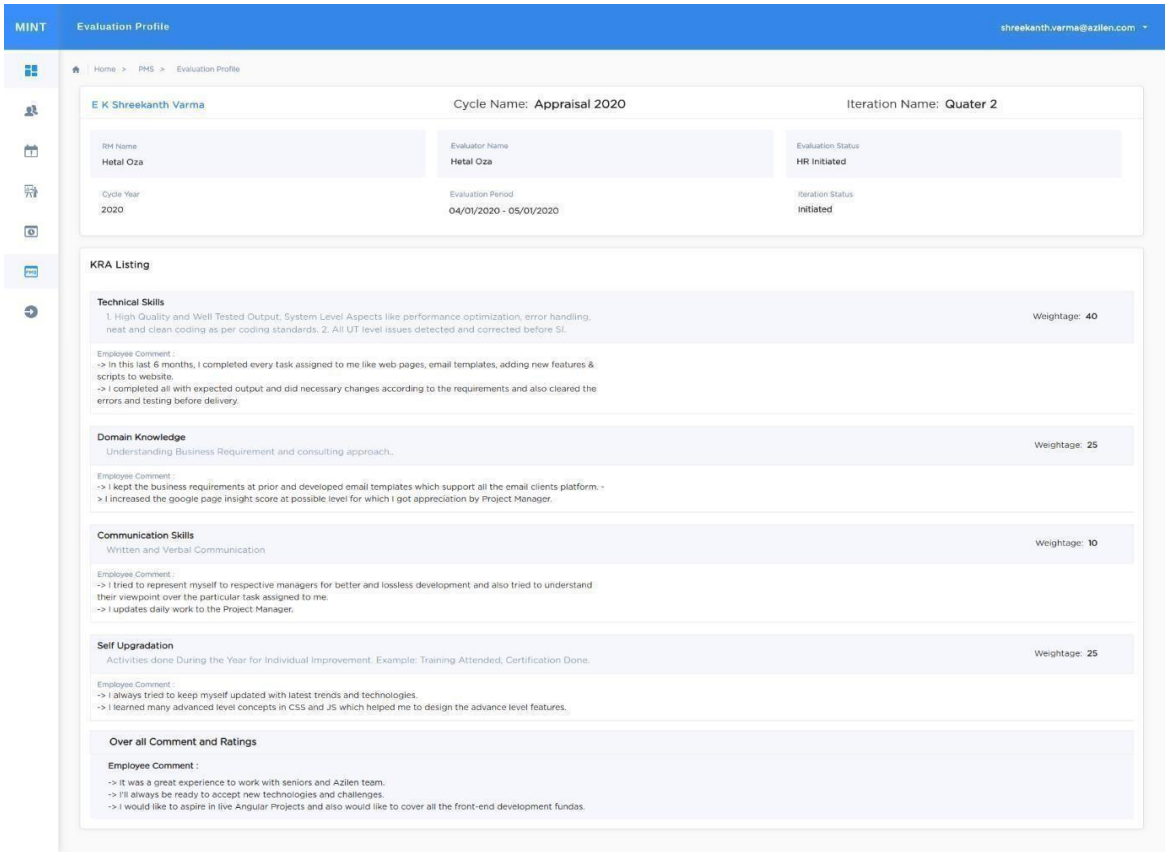
[Fig. 5.3.9.2 Self-Evaluation – employees can be part of self-evaluation process]

- Project Manager has added KRAs for the employee.



[Fig. 5.3.9.3 Self-Evaluation – by employee on basis of KRA]

- Here, employee has entered their comments for self-evaluation.



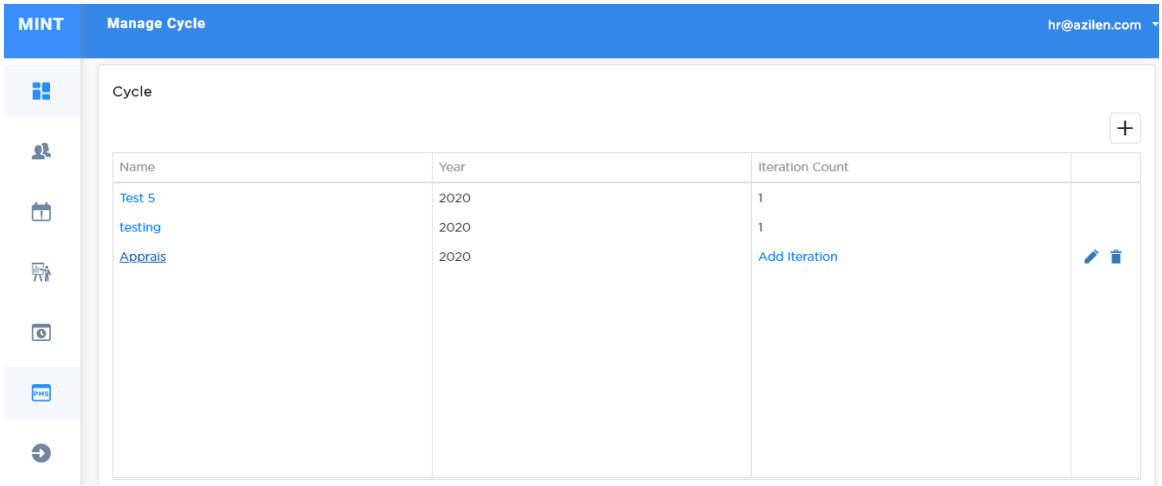
[Fig. 5.3.9.4 Self-Evaluation – Employee can add their comments on self evaluation]

6. IMPLEMENTATION

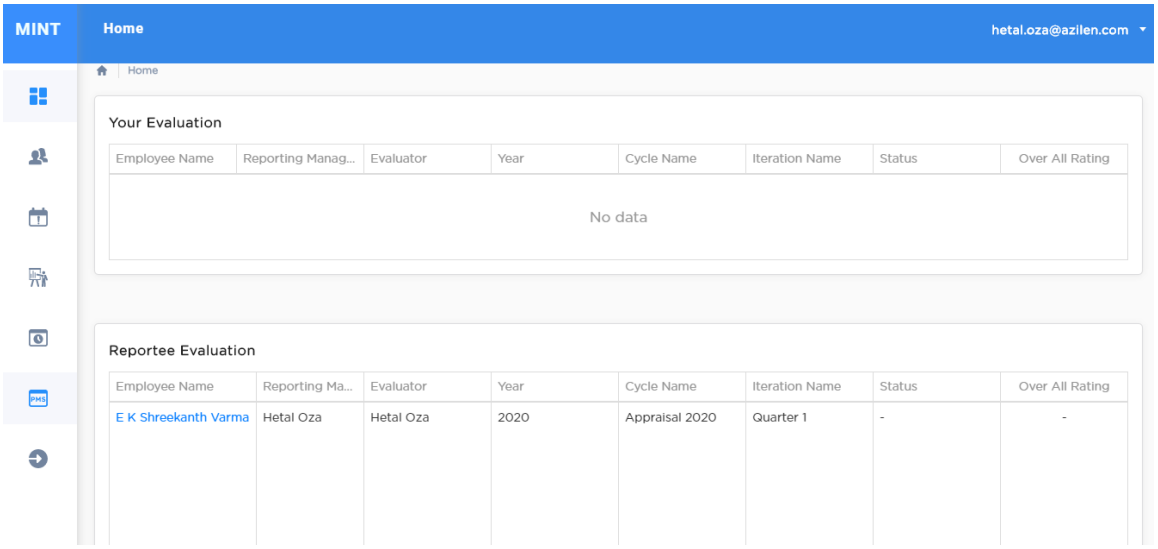
6.1 IMPLEMENTATION ENVIRONMENT

1) Our project is suitable to all type of users like single and multi-users.
2) Multi users are allowed to operate the System at the same time.
3) We provide the interface which is user friendly.
4) One user at a time and also multi users can access the System at the same time and use all the services.
5) So, for the more security and performance we have to use the Keycloak.
6) For Authorization and authentication purpose we have used Keycloak middleware.

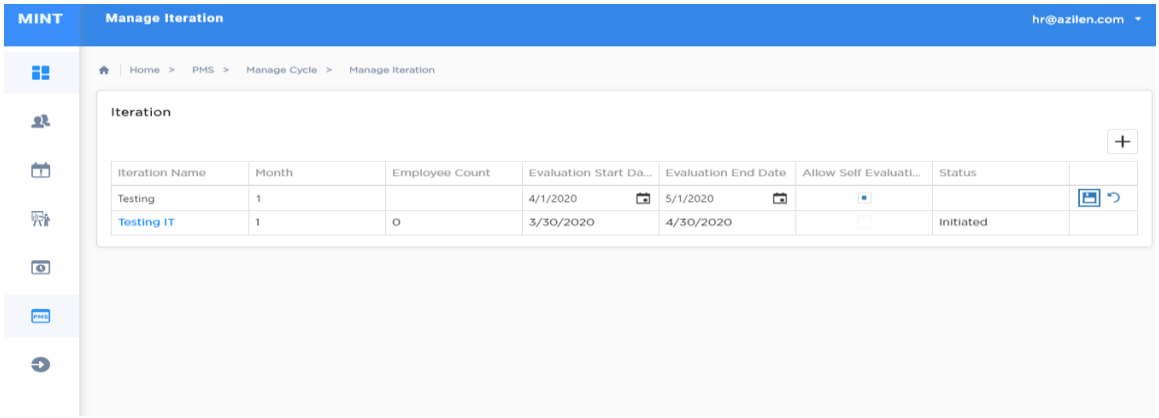
6.2 RESULTS



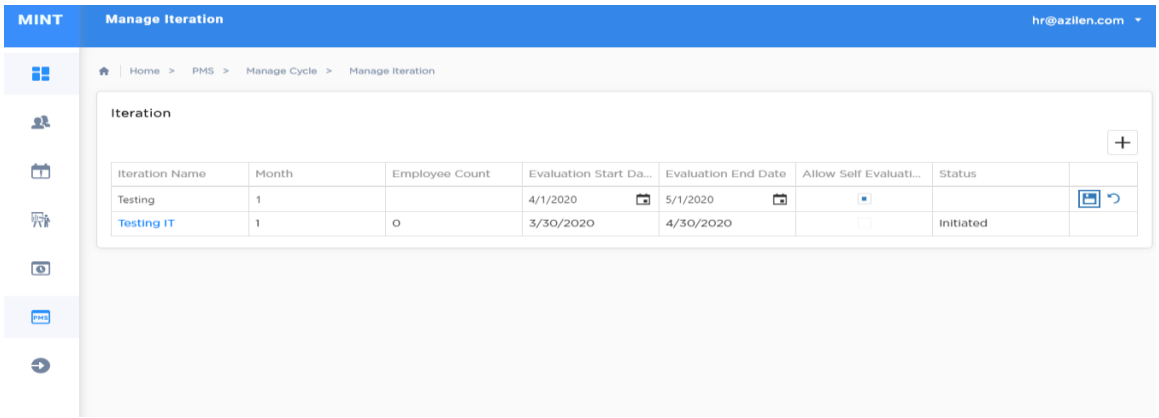
[Fig. 6.2.1 HR created performance cycle for employee]



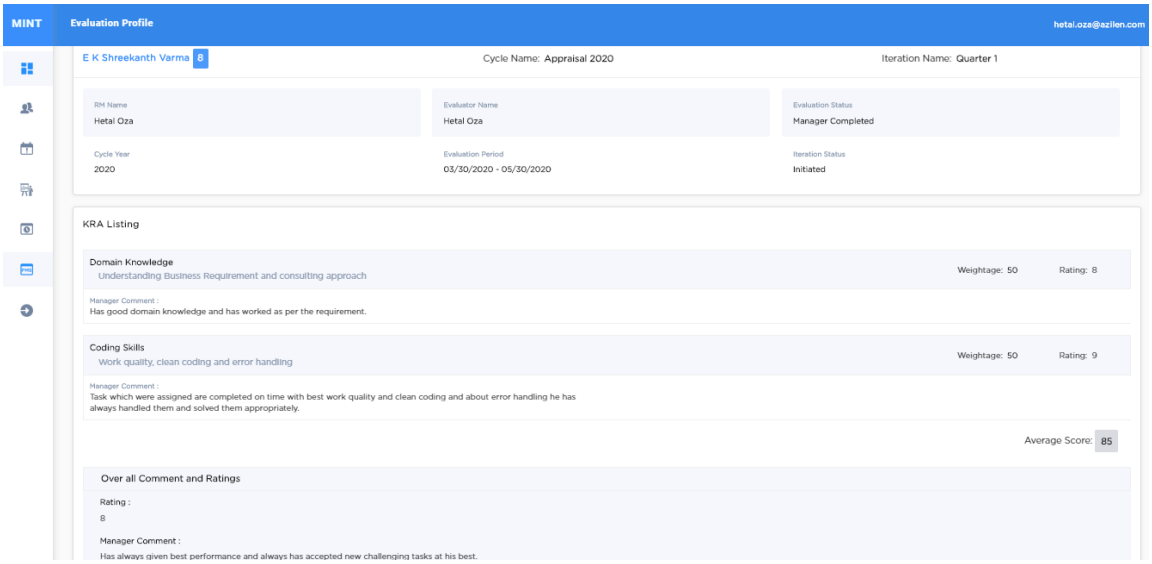
[Fig. 6.2.2 Employee side evaluation screen]



[Fig. 6.2.3 Iteration cycle dashboard]



[Fig. 6.2.4 HR Can Edit in iteration cycles]



[Fig. 6.2.5 Employee can comment in self evaluation process]

7. TESTING

7.1 TESTING PHASE

7.1.1 Testing Requirement :

- User should be able to log into the system by providing his/her username
- If user is HR then HR should be able to see dashboard where there are components: Manage Profile, Manage Cycle and Manage KRA. Also, HR should be able to the list of employees who are which HR has added them for evaluation.
- HR should be able to create cycle, iteration and add assign the employees to the Project Manager successfully.
- If the user is employee, then should be able to see their pervious evaluations and if Self-evaluation is enabled then employee should also be self-evaluating themselves successfully.
- If user is Project Manager, then he/she should be able to see the employees which are assign to them.
- Also, Project manager should be able to add KRAs, weightage, comments and ratings to the employees successfully.
- If the user is Reporting Manager, then he/she should be able to view given comments and ratings of the employees.

7.1.2 Test Planning:

- Testing should be done on different browsers such as Internet Explorer, Google Chrome, Mozilla Firefox, Safari, etc.
- **Features to be tested:**
  - Login
  - Create Profile
  - Create Cycle
  - Create Iteration
  - Add KRAs
  - Add comments

7.1.3 Testing Types :

- The following types of testing are for this project :

1. Unit Testing :

- Testing if user is able to login successfully.
- Checked if HR is able to see the dashboard having Manage Profile, Manage Cycle, and Manage KRA.
- Checked if cycle and iteration pages are giving correct output when HR creates cycle/s or iteration/s.
- Checked if HR is able to search and employees and add those employees to the iterations and assign Project Mangers.
- Checked if employees are able to see their profiles.

2. Integration Testing :

- Checked if user is able to navigate to their respective home screens successfully.
- Checked if HR is able to navigate to Manage Cycle screen by clicking on it from dashboard.
- Checked if HR is able to see the list of employees which are to be evaluated and which were added in iteration.
- Checked if HR is able to create iteration/s in the cycle/s and those functionalities working correct after that.
- Checked if employees are able to view evaluation done by Project Manager.
- Checked if Project Manager is able to navigate to employee profile by clickingon it to add KRAs.

3. System Testing :

- Checked if the HR is able to create cycle and iteration with adding employees and assigning those employees Project Manager.
- Checked if those employees are displayed on the dashboard of HR for evaluation.
- Checked if Project Managers is able to see the employees which are allocated to them.

- Checked if Project Managers are able to add KRAs to that employee and also evaluate them. Checked if employees are able to view their evaluation which done by Project Managers.

7.2 TEST RESULTS & ANALYSIS

7.2.1 Login

TC Id	Test Title	Test Data	Expected Result	Actual Result	Pass/Fail
LTC1	Check username field with valid credentials	Username: mintHrManager	User should be successfully logged in	User is able to successfully logged in	Pass
LTC2	Check behavior by keeping username filed empty.	password: test	User should shown error saying “username is required.”	When user keeps username field empty then it is showing the error.	Pass
LTC3	Check behavior by keeping password field empty	Username: mintHrManager	User should shown error saying “password is required”.	When user keeps username field empty then it is showing the error.	Pass

[Table 7.2.1 Login]

7.2.2 Create Cycle

TC Id	Test Title	Test Data	Expected Result	Actual Result	Pass/Fail
CTC1	Check if user is able to add cycle details by clicking on '+' sign		When user clicks n'+' sign then user should be able to add cycle details.	User is able to add cycle details by clicking on '+' sign.	Pass
CTC2	Check behavior of the system by keeping cycle name empty.		When user keeps cycle name empty then error should be shown.	Error is shown when cycle name is kept empty.	Pass
CTC3	Check behavior of the system by keeping year field empty.		When user keeps year field empty then error should be shown.	Error is shown when year filed is kept empty	Pass
CTC4	Check if cycle is created by clicking on save button	Cycle name: Appraisal Year: 2020	When user clicks on save button Then cycle should be created.	Cycle is Created when clicks on save button.	Pass

[Table 7.2.2 Create Cycle]

7.1.1 Integration Testing

TC Id	Test Title	Test Data	Expected Result	Actual Result	Pass/Fail
GTC1	Check if user is able to add employees in the cycle and iteration.	E K Shreekanth Varma	User should be able to add employee/s in the cycle and iteration.	User is able to add employee/s in cycle and iteration.	Pass
GTC2	Check if Project Manager is to see the employee/s list allocated to them.		Project Manager should be able to see to the list of employee/s allocated to them.	Project Manager is able to see to the list of employee/s allocated to them.	Pass
GTC3	Check if HR is able to see the evaluation done by Project manager.		When Project Manager does evaluation, the HR should be able to see that.	HR is able to see the evaluation done by Project Manager.	Pass
GTC4	Check if Employee is able to see the evaluation done by Project Manager.		When Project Manager does evaluation, the employee should be able to see that.	Employee is Able to see the evaluation done by Project Manager.	Pass
GTC5	Check if Employee is bale to add comments after Project Manager initiate the evaluation.		When Project Manager is initiate the evaluation, Employee should be able to add comments.	Employees are able to add comments.	Pass
GTC6	Check if Employees is able to view HR's comment in their evaluation.		When HR adds any comment in employee's evaluation then Employees should be able to see it.	Employees are able to the comment given by HR.	Pass

[Table 7.2.3 Integration Testing]

7.2.3 Create Iteration

TC Id	Test Title	Test Data	Expected Result	Actual Result	Pass/Fail
ITC1	Check if user is able to add iteration Details by clicking on '+' sign		When user clicks n'+' sign then user should be able to add iteration details.	User is able to add iteration details by clicking on '+' sign.	Pass
ITC2	Check behavior of the system by keeping iteration name empty.	Month: 2 Start Date: 4/1/2020 End Date: 4/1/2020	When user keeps iteration name empty then error should be shown.	Error is shown when iteration name is kept empty.	Pass
ITC3	Check Behavior of the system by keeping month field empty.	Iteration Name:Q1 Start Date: 4/1/2020 End Date: 4/1/2020	When user keeps month field empty then error should be shown.	Error is shown when month filed is kept empty.	Pass
ITC4	Check behavior of thesystem by keeping start date and end date empty.	Iteration Name:Q1 Month: 2	When user keeps start dateand end date empty then error should be shown.	Error is shown when start date and end date is kept empty.	Pass
ICT5	Check if self-Evaluation allows to check or uncheck the checkbox.	Iteration Name:Q1 Month:2 Start Date: 4/1/2020 End Date: 4/1/2020	When user wants to allow self-evaluation then user should be able to check the box and if user doesn't want to do so then user should be able to uncheck it.	User is able to check or uncheck the checkbox.	Pass

[Table 7.2.4 Create Iteration]

8. CONCLUSION AND DISCUSSION

8.1 OVERALL ANALYSIS OF INTERNSHIP

- In this internship I have learned many new things and Technology. By This Internship I came to know how the Projects are designed at Business level, how the project is managed by the managers.

- Through this internship I also learned how the client interacts with various developers and how developers try to solve problem of client.

- In this internship I learned about Spring Boot framework, Maven, Hibernate, Axon Framework, Event handling. I got a Great experience will working with this project.

- It was a complete immersive experience in learning new things and technologies in this internship.

- Found the senior developers over there so much helpful. We were provided with mentors over there.

- Learned the backend side of things in the internship. Clearly interacted with the CTO of the company. Made this Performance Management System project under his guidance.

8.2 SUMMARY OF INTERNSHIP

- PMS Performance Management System is a system through which employees could be easily evaluated based on their performance. It is quite easy to use and understand the system flow.
- Through this we can reduce the work of HR and managers both. Even the problem of storing data is not there as the ratings and evaluations are automatically getting stored in the system.
- The main aim was to give a better environment to the organization where the people can work more effectively and efficiently.

8.3 FUTURE ENHANCMENT

- In future we are planning to implement a "Cost functioning'" module, which will help the organization in many ways as it would ease a lot of processes done by Sales Team and Top- level management.

- The Cost functioning module is divided into two parts, one part is where it will help the Top-level management in deciding the appraisals for all the employees as per their performance. There will be different algorithms set using which this would be done. It will take time in implementing as it has many security issues as well as it is much complex considering the diversified departments in the organization.

- Another part is where the Sales team and Accounting team will be benefitted. Here all the income and expense related detailing work will be done. All the accounting of money will be done through this module.

- After the successful implementation of PMS, we will monitor it for six months and observe for any defects or enhancements requirements and if needed will do the same.

References

Following are the sites that are used in Project Development:

[1] https://www.pluralsight.com/

[2] https://www.javatpoint.com/spring-boot-tutorial

[3] https://www.baeldung.com/spring-boot

[4] https://angular.io/

[5] https://angular.io/tutorial/toh-pt6

[6] https://js.devexpress.com/

[7] https://stackoverflow.com/

[8] https://dzone.com/articles/microservices-communication-zuul-api-gateway-1

[9] https://app.pluralsight.com/library/courses/springmvc-intro/table-of-contents

[10] https://www.tutorialspoint.com/spring/index.htm

[11] https://www.javacodemonk.com/spring-cloud-and-its-advantages-3ac60b2c

[12] https://www.youtube.com/watch?v=P1mlC8Ar0_k

[13] https://www.baeldung.com/spring-boot-keycloak

[14] https://www.keycloak.org/docs/latest/securing_apps/