INTERNSHIP

An INTERNSHIP REPORT

Submitted by

Mitesh Dave
181260107013

In partial fulfilment for the award of the degree of

BACHELOR OF ENGINEERING
In
Computer Engineering
SAL Engineering & Technical Institute , Ahmedabad

Gujarat Technological University, Ahmedabad
[April, 2021-2022]

SAL Engineering & Technical Institute

Opp. Science City, Sola
Ahmedabad , Gujarat 380060

CERTIFICATE

This is to certify that the project report submitted along with the project entitled **Internship** has been carried out by **Mitesh Dave** under my guidance in partial fulfilment for the degree of Bachelor of Engineering in **Computer Engineering**, 8th Semester of Gujarat Technological University, Ahmadabad during the academic year 2021-22.

Prof. Harshit Vora
Internal Guide

Prof. Sudha Patel
Head of the Department

SAL Engineering & Technical Institute

Opp. Science City, Sola
Ahmedabad , Gujarat 380060

DECLARATION

We hereby declare that the Internship report submitted along with the Internship entitled **Internship** submitted in partial fulfilment for the degree of Bachelor of Engineering in **Computer Engineering** to Gujarat Technological University, Ahmedabad, is a bonafide record of original project work carried out by me **Azilen Technologies Pvt. Ltd.** at under the supervision of **Prof. _____** and that no part of this report has been directly copied from any students' reports or taken from any other source, without providing due reference.

Name of the Student	Sign of Student
1 **Mitesh Dave**	**181260107013**

ACKNOWLEDGEMENT

- It gives me great pleasure in expressing thanks and profound gratitude to **"Mr.Arshil Shaikh"** (Tech Lead, Azilen Technologies, Ahmedabad) for his continued support throughout the Internship.

- I would also like to thank my Faculty Guide, **"Prof.Harshit Vora"**, Department of Computer Engineering, SAL Engineering & Technical Institute, Ahmedabad for his valuable guidance and continual encouragement throughout the Internship.

- I am also thankful to **"Dr. Tejas Patalia"**, Head of Department, Computer Engineering Department, VVP Engineering, Rajkot for his continual kind words of encouragement and motivation throughout the Internship.

- I am thankful to all faculty members of the Computer Engineering Department, SAL Engineering & Technical Institute, Ahmedabad for their special attention and suggestions towards the Internship.

- I am also thankful to the employees of Azilen Technologies Pvt. Ltd. for support and encourage me toward the training.

List of Figures

Fig 4.1 Use case Diagram……..……..…24
Fig 4.2 Activity Diagram……27
Fig 4.3 Sequence Diagram…….29
Fig 4.4 Class Diagram……30
Fig 4.4 E-R Diagram……..32
Fig 5.1 Login page……..33
Fig 5.2 Home Screen of HR……...33
Fig 5.3 Create Cycle……...34
Fig 5.4 Edit Cycle……..35
Fig 5.5 Delete cycle……36
Fig 5.6 Create Iteration……...36
Fig 5.7 Edit Iteration……..37
Fig 5.8 Delete Iteration……...38
Fig 5.9 Employees and Assigning Project Manager……..40
Fig 5.10 Project Manager Screen……...41
Fig 5.11 ADD KRAs……..41
Fig 5.12 Delete KRA……..42
Fig 5.13 Give Comments and Ratings……44
Fig 5.14 Self-evaluation……46
Fig 5.15 Create Profile……...48

Table of Contents

Acknowledgement ……i
List of Figures ……ii
Table of Contents……iii
Chapter 1 Introduction……1
1.1 About the Company ……1
1.2 About the Internship……2
Chapter 2 Learning and knowledge gain……...3
2.1 Different role in software company……..……3
2.2 Different architecture……...5
2.3 SDLC models……...6
Chapter 3 PMS(Performance management system)……9
3.1 Project Definition……9
3.2 Feasibility study ……12
3.3Requirement engineering……12
3.4 System requirement study……..13
Chapter 4 UML Diagrams……21
4.1 Use case diagram……21
4.2 Activity diagram……22
4.3 Sequence diagram……...25
4.4 Class diagram……28
4.5 E-R diagram ……..28
Chapter 5 User manual ……31
5.1 Login……..…31
5.2 Home screen of HR……31
5.3 Create cycle……32
3.4 Edit cycle……32
5.5 Delete cycle……....33
5.6 Create Iteration……...34
5.7 Edit Iteration……..34
5.8 Delete Iteration…….....35
5.9 Employees and Assigning Project Manager……...36
5.10 Project Manager Screen……38
5.11 Add KRAs……39
5.12 Delete KRA……..39
5.13 Give ratings and comments…….....40
5.14 Self Evaluation ……42

5.15 Create Profile ……...45
Chapter 6 Testing……...47
6.1 Testing Phase…….……47
6.2 Testing types……..48
Chapter 7 Future Enhancement…….…..50
Conclusion……...51
Appendix……52
References……...53

1. INTRODUCTION

1.1 ABOUT THE COMPANY

1.1.1 Introduction of the Company

- 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, Artificial Intelligence, Internet of Things, Big Data Analytics and HR-Tech.

1.1.2 Communication

- **Company Address**
  Azilen Technlogies Pvt. Ltd.
  12th & 13th Floor, B Square, 1,
  Ambli - Bopal Rd,
  Ahmedabad, Gujarat 380054
- **Phone:** 079 4009 3121
- **Company Website Address**
  https://www.azilen.com/

1.1.3 Resources

1.1.3.1 Fix resources

- The Head office is at Ahmedabad and other office is in Mumbai and Indore.
- Our other branches are also there in different countries like USA, Switzerland and Belgium..
- Database Servers having very high storage capacity.
- Data Warehouse Server
- More than 400 Computers with high configuration

1.1.3.2 Human Resources

- More than 350 employees.

1.2 ABOUT THE INTERNSHIP

1.2.1 Work Done in Internship

- During my internship first I have gain some knowledge of company work and some new technology
- After that I have work on One project named PMS. In that project I have drawn some diagram like Activity diagram, Class diagram, Sequence Diagram, Dataflow diagram, E-R diagram etc.
- After that I have assign one live project named Azicoin. In that project I have built user module.

2. LEARING AND KNOWLEDGE GAIN

2.1 DIFFERENT ROLE IN SOFTWARE COMPANY

2.1.1 Software Developer

- The task of a software developer depends on the needs of the company, organization, or team they are on. Some build and maintain systems that run devices and networks. Others develop applications that make it possible for people to perform specific tasks on computers, cellphones, or other devices.
- Modifying software to fix errors, adapt it to new hardware, improve its performance, or upgrade interfaces.
- Directing system testing and validation procedures.
- Directing software programming and documentation development.
- Consulting with departments or customers on project status and proposals.
- Working with customers or departments on technical issues including software system design and maintenance.
- Analysing information to recommend and plan the installation of new systems or modifications of an existing system.
- Consulting with engineering staff to evaluate software hardware interfaces and develop specifications and performance requirements.
- Designing and developing software systems using scientific analysis and mathematical models to predict and measure outcomes and design consequences.
- Preparing reports on programming project specifications, activities, or status.
- Conferring with project managers to obtain information on limitations or capabilities.

2.1. 2 Quality assurance (QA)

- Documenting and reporting product or service quality levels
- Developing and implementing standards for inspection
- Developing a workflow for product inspection
- Developing plans to help a company manage waste
- Communicating with other team members to solve problems

- Following up with the appropriate channels when mistakes are found
- Training other quality insurance members on all inspection processes

2.1.3 Chief Architect

- Drive project/program execution.
- Provide architectural and technical guidance to the engineering organizations.
- Deliver quality software products on schedule and within cost.
- Be responsible for Full life cycle systems/applications development to achieve corporate goals and clients' requirements.
- Lead the design, development, integration, and debugging of systems and applications.
- Do requirements analysis, system/software design and analysis, development/software metric creation and analysis, system risk analysis, integration, and test metrics.
- Develop and maintain detailed program schedules and software development plans.
- Translate product requirements into engineering specifications.
- Manage and oversee software development tasks and overall engineering schedule.
- Provide leadership to the software engineering team and enforce standard company policies and procedures that adhere to software development standards.
- Work closely with the CTO to define product direction and design.
- Anticipate future system needs and be proactive in identifying solutions.
- Be responsible for project planning and management.
- Evaluate and identify new technologies for implementation.
- Support, maintain, and document software functionality.
- Be responsible for Object Oriented Design and Analysis (OOD & OOA)

2.1.4 Project Manager

- One of the chief responsibilities of a software project manager is to define the scope of the project. Software development won't be successful without a

clear delineation of timelines, schedules, project styles to use, projections, available resources, and overall goals.

- It's the responsibility of a project manager, with the help of a capable team, to set these clear terms at the beginning, at regular intervals, and to revisit the terms at the end of the product development life cycle.
- A project manager is expected to be able to coordinate company resources and that of third parties to ensure that the project is carried out perfectly
- To successfully manage resources, a project manager has to be able to delegate resources efficiently to maximize output while minimizing waste. Part of the responsibilities of a project manager in software development is expert resource allocation.
- And of course, a project manager is expected to be able to execute a project. It's the combined responsibilities of a software project manager, which is to ensure that a project comes to a successful close.

2.2 DIFFERENT ARCHITECTURE

2.2.1 Layered

- As the name suggests, components(code) in this pattern are separated into layers of subtasks and they are arranged one above another.
- Each layer has unique tasks to do and all the layers are independent of one another. Since each layer is independent, one can modify the code inside a layer without affecting others.
- t is the most commonly used pattern for designing the majority of software. This layer is also known as 'N-tier architecture'.

2.2.2 Client-Server

- The client-server pattern has two major entities. They are a server and multiple clients.
- Here the server has resources (data, files or services) and a client requests the server for a particular resource. Then the server processes the request and responds back accordingly.

2.2.3 Event-Driven

- Event-Driven Architecture is an agile approach in which services (operations) of the software are triggered by events.
- When a user takes action in the application built using the EDA approach, a state change happens and a reaction is generated that is called an event.

2.2.4 Microkernel

- Microkernel pattern has two major components. They are a core system and plug-in modules.
- The core system handles the fundamental and minimal operations of the application.
- The plug-in modules handle the extended functionalities (like extra features) and customized processing.

2.2.5 Micro Services

- The collection of small services that are combined to form the actual application is the concept of micro services pattern. Instead of building a bigger application, small programs are built for every service (function) of an application independently. And those small programs are bundled together to be a full-fledged application.
- So adding new features and modifying existing micro services without affecting other micro services are no longer a challenge when an application is built in a micro services pattern.
- Modules in the application of micro services patterns are loosely coupled. So they are easily understandable, modifiable and scalable.

2.3 SDLC MODELS

2.3.1 Waterfall Model

- The waterfall is a universally accepted SDLC model. In this method, the whole process of software development is divided into various phases.
- The waterfall model is a continuous software development model in which development is seen as flowing steadily downwards (like a waterfall) through the

steps of requirements analysis, design, implementation, testing (validation), integration, and maintenance.

- Linear ordering of activities has some significant consequences. First, to identify the end of a phase and the beginning of the next, some certification techniques have to be employed at the end of each step.
- Some verification and validation usually do this mean that will ensure that the output of the stage is consistent with its input (which is the output of the previous step), and that the output of the stage is consistent with the overall requirements of the system.

2.3.2 RAD Model

- RAD is a linear sequential software development process model that emphasizes a concise development cycle using an element based construction approach. If the requirements are well understood and described, and the project scope is a constraint, the RAD process enables a development team to create a fully functional system within a concise time period.
- RAD (Rapid Application Development) is a concept that products can be developed faster and of higher quality through:
    - Gathering requirements using workshops or focus groups
    - Prototyping and early, reiterative user testing of designs
    - The re-use of software components
    - A rigidly paced schedule that refers design improvements to the next product version
    - Less formality in reviews and other team communication

2.3.3 Spiral Model

- The spiral model is a **risk-driven process model**. This SDLC model helps the group to adopt elements of one or more process models like a waterfall, incremental, waterfall, etc. The spiral technique is a combination of rapid prototyping and concurrency in design and development activities.
- Each cycle in the spiral begins with the identification of objectives for that cycle, the different alternatives that are possible for achieving the goals, and the constraints that exist. This is the first quadrant of the cycle (upper-left quadrant).

- The next step in the cycle is to evaluate these different alternatives based on the objectives and constraints. The focus of evaluation in this step is based on the risk perception for the project.
- The next step is to develop strategies that solve uncertainties and risks. This step may involve activities such as benchmarking, simulation, and prototyping.

2.3.4 Agile Model

- Agile methodology is a practice which promotes continues interaction of development and testing during the SDLC process of any project.
- In the Agile method, the entire project is divided into small incremental builds. All of these builds are provided in iterations, and each iteration lasts from one to three weeks.
- Any agile software phase is characterized in a manner that addresses several key assumptions about the bulk of software projects:
    - It is difficult to think in advance which software requirements will persist and which will change. It is equally difficult to predict how user priorities will change as the project proceeds.
    - For many types of software, design and development are interleaved. That is, both activities should be performed in tandem so that design models are proven as they are created. It is difficult to think about how much design is necessary before construction is used to test the configuration.
    - Analysis, design, development, and testing are not as predictable (from a planning point of view) as we might like.

2.3.5 Incremental Model

- The incremental model is not a separate model. It is necessarily a series of waterfall cycles. The requirements are divided into groups at the start of the project.
- For each group, the SDLC model is followed to develop software. The SDLC process is repeated, with each release adding more functionality until all requirements are met. In this method, each cycle act as the maintenance phase for the previous software release.
- Modification to the incremental model allows development cycles to overlap. After that subsequent cycle may begin before the previous cycle is complete.

3.PMS(PERFORMANCE MANAGEMENT SYSTEM)

3.1 PROJECT DEFENATION

3.1.1 Definition

- 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.

3.1.2 Purpose and objectives

- **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.

3.1.3 Scope of Project

- 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 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.

3.1.4 Proposed system

- In PMS we have implemented Micro service architecture, here the system is divided into small individual services.
- Each and every service can work on its own and is not dependent on any other service. So, if 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.
- We have followed Domain Driven Design (DDD); the richness is in the way we approached the requirement understanding and suctioning.
- In DDD there is a fixed set of parameters or steps to understand a domain. We have followed those steps before and throughout during the design process.
- In this system HR will create Appraisal cycles for different years. HR will create multiple iterations for the same employee.
- HR can create iterations for custom time range also. HR will be able to create profiles for different designations where he/she will set some predefined KRAs.
- KRA is abbreviation of Key Result Area refers to the common areas of results or outcomes for any roles which is responsible.
- Also, the people of organization or say individual or group of is responsible for the outcomes logically.
- Here, the Project Manager will either import the profiles created by HR or he/she can set different custom made KRAs for different employees and rate them against these KRAs.
- Here the Employees can also perform self-evaluation. The employee KRAs and ratings will be visible to corresponding Project Manager and the Reporting Manager of that particular employee. There will be a functionality to import employee data from another system (existing HRMS system).
- All the employee evaluation related data will be encrypted before storing in the database for the protection of the data.

3.1.5 Benefits 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.

3.2 FEASIBILITY STUDY

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

3.2.1 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.

3.2.2 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 mango db for the database.
- Knowledge of java and angular is essential to carry out this project.

3.2.3 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.

3.3 REQUIREMENT ENGINEERING

3.3.1 Facts-Finding Techniques

3.3.1.1 Interview

▪ Most of the communication is done through face to face. Here we used to collect information from the Project lead and the CTO. The communication used to be very clear and accurate as no interruptions were caused.

3.3.1.2 Presentation and Questionnaires

▪ After the end of every sprint we used to show our work to project lead and CTO.

▪ At that time only we would ask out the questions for the next sprint and discuss all the feasibility aspects.

3.3.1.3 Record Review

▪ There were documents which were shared by the project lead for our reference to the project.

▪ Also, the exchange of mails was done for guidance and problem solving.

3.3.1.4 Observation

▪ We learned a lot by observing our seniors work. We used to observe them so that we can improve our working style.

3.4 SYSTEM REQUIREMENT STUDY

3.4.1 Functional and non-functional requirements

- **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.

a. 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.

b. 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.

c. 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.

a. Add Employee:

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

b. 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.

7. View Comments and Ratings:

- **Description:** This functionality allows Project Manager and Reporting Manager to see the all employees' comments and ratings. Also, employees can see comments given by Project Manager.

a. View comments and ratings for Project Manager and Reporting Manager:

- **Input:** Project Manager and Reporting Manager will log in.
- **Output:** View comments and ratings of Employees.

b. View Comments for Employees:

- **Input:** Employee Login.
- **Output:** View comment.

8. Logout:

- **Description:** This functionality allows log out of login mode in particular page.
- **Precondition:** User can be on any page.
- **Input:** User will select log out button from their own profile.
- **Output:** Successfully logout and will navigate to login page.
- **Processing:** System will logout of that application.

- **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 any browser only. It will be access through an internal link of an organization and only the members of an organization can access it.

3.4.2 User Characteristics

- The project Performance Management System has a total of four users in it:
  - HR
  - Project Manager
  - Employee
  - Reporting Manager
- The HR will add all employees into the system including Project Manager and Reporting Manager. This panel will have all the controls except giving weightage, comments and rating.
- Secondly, there is user who is Project Manager who will evaluate employees which are allocated to him/her. Evaluation includes adding various types of KRAs along with its weightage and will

add ratings afterwards. If, the evaluation is going to be done by Project Manager then Project Manager will add comments too.

- Third, there is user who is employee. Employees can see to whom they are allocated to, in which cycle they are allocated and in which iteration they are allocated. Also, employees can self-evaluate themselves if HR has given such permission (control) to them. In such case, employee has to add only comments against the KRAs.

3.4.3 Need of Self-evaluation

- Self-evaluation of employees is very important for the organization, as self-evaluation is the self-realization about the performance for their employee.
- It helps employees in setting up new goals for their carrier and also for themselves.
- This self-evaluation also helps Project Manager or Reporting manger or the other employer to analyse and understand each employee and also their mind-set.
- By self-evaluation process employees can know or can explore their own talent and also allows employee to grow their talents in front of management.

3.4.4 Technologies used in project (detail)

- **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.
    - One more reason for implementing Keycloak is suppose one of the employee has to occasionally access one HR based resource on situational basis, then we just have to provide the scope corresponding

to that particular resource and we don't have to give hi HR role for that matter, as it would be breaking the organization hierarchy.

- Other things implemented using keycloak is that the APIs that are available will be scope based unlike role based, as it was in the requirement.

- **ELK stash**
  - It has been used for storing the logs in a central space following the concept of centralized logging. It helps in finding the errors and issues when multiple servers are connected at a same time as all the logs will be at one place. In future we can use this for analytical processes also.

4. UML DIAGRAMS

4.1 USECASE DIAGRAM

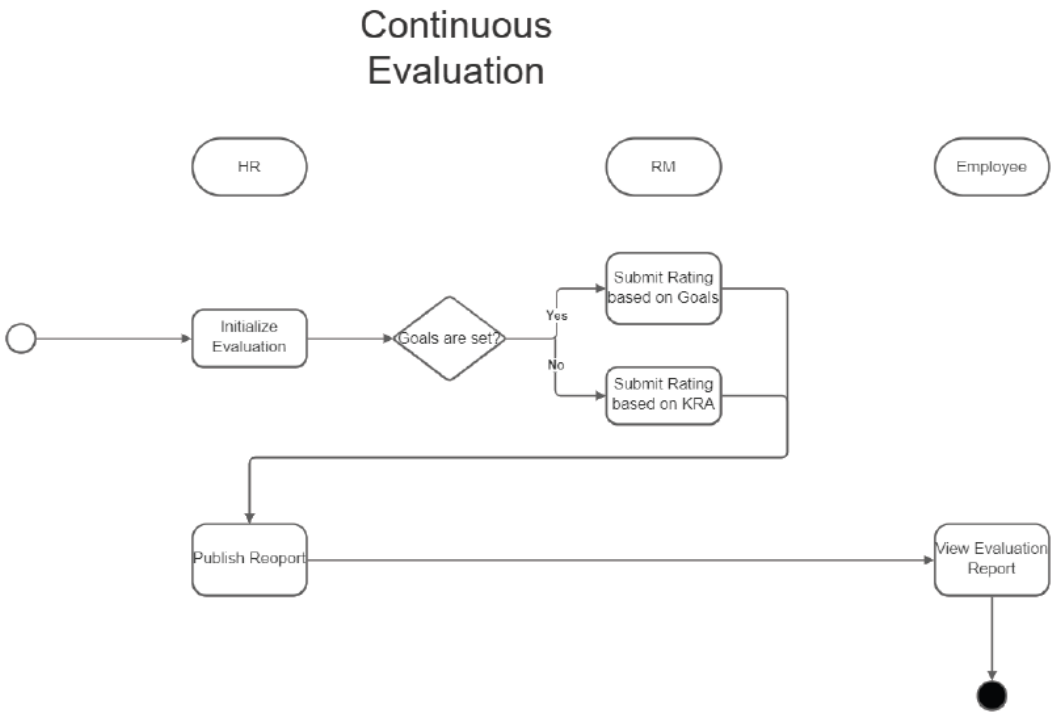
- A use case diagram is used to represent the dynamic behavior of a system. It encapsulates the system's functionality by incorporating use cases, actors, and their relationships. It models the tasks, services, and functions required by a system/subsystem of an application. It depicts the high-level 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.
- Once both the actors and use cases are enlisted, the relation between the actor and use case/ system is inspected. It identifies the no of times an actor communicates with the system. Basically, an actor can interact multiple times with a use case or system at a particular instance of time.

Fig4.1 Use case diagram

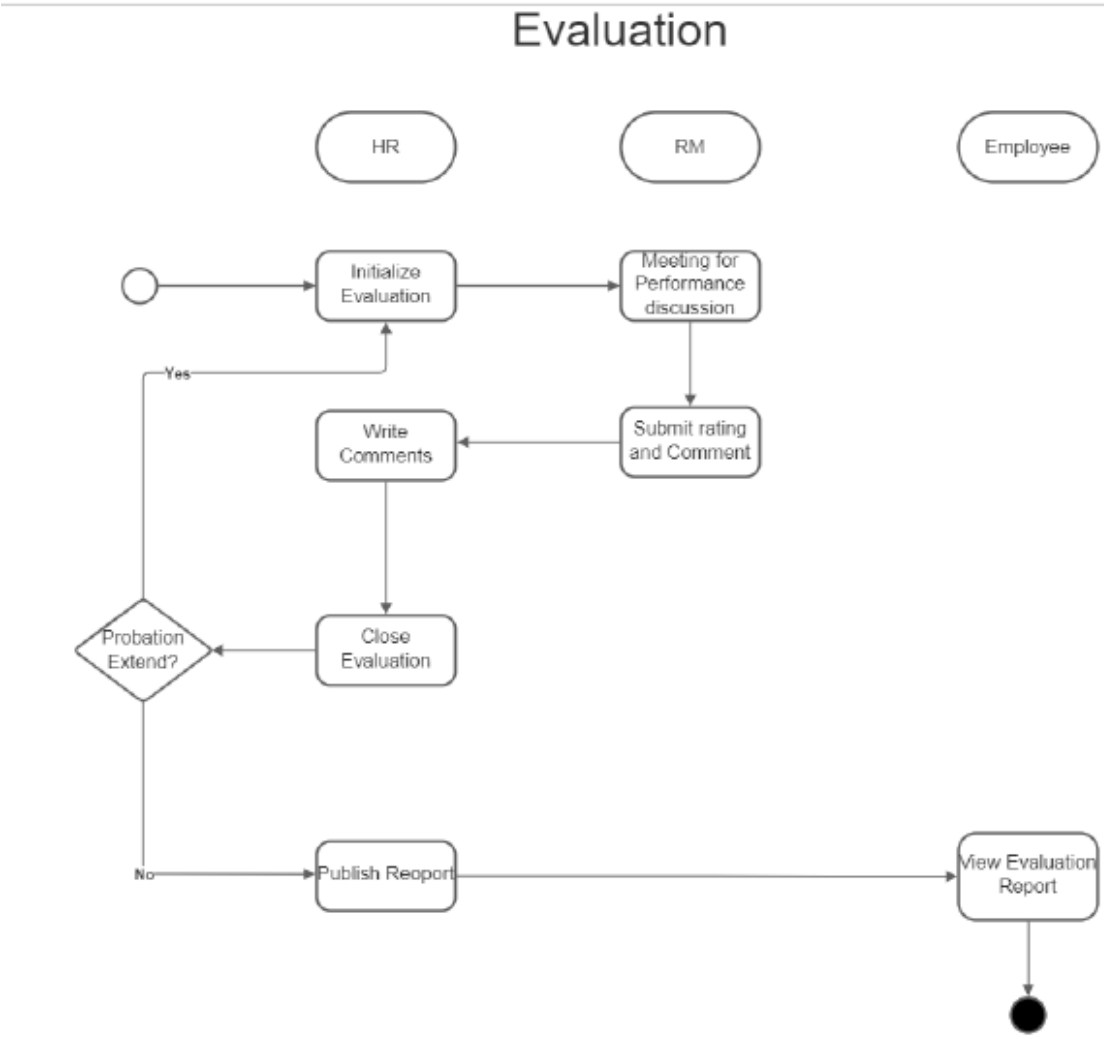
4.2 ACTIVITY DIAGRAM

- 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.

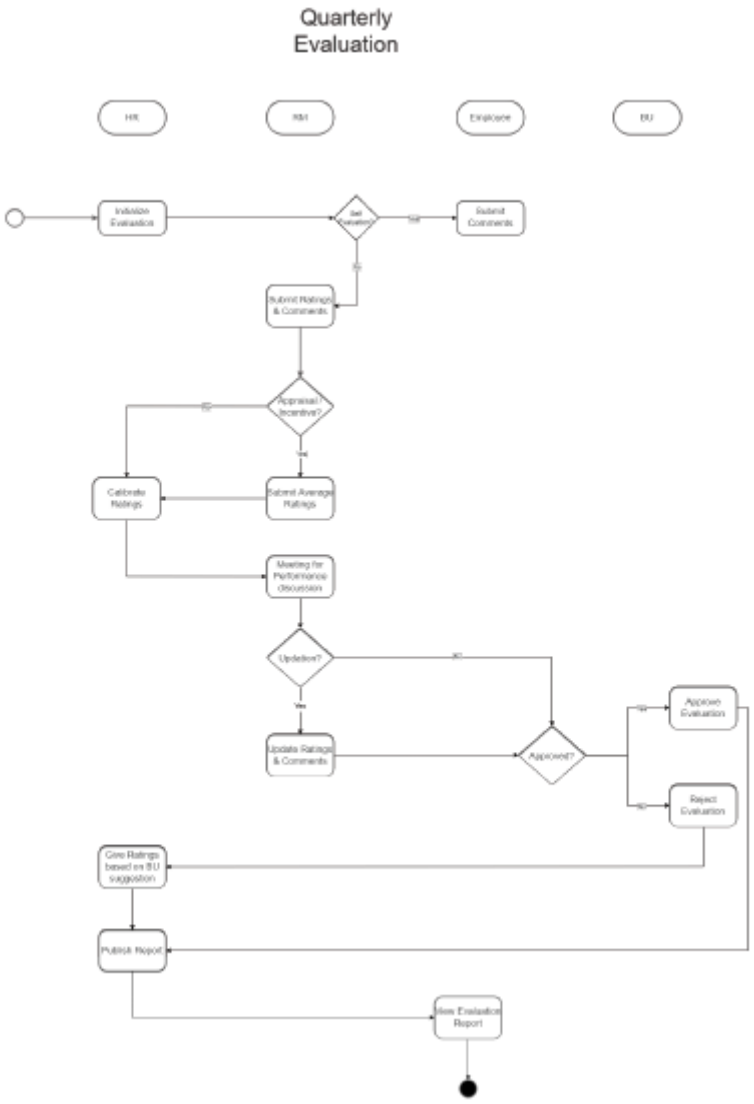
- It does not show any message flow from one activity to another. Activity diagram is sometimes considered as the flowchart. Although the diagrams look like a flowchart, they are not. It shows different flows such as parallel, branched, concurrent, and single.
- Activity diagrams are mainly used as a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


Evaluation
HR
RM
Employee
Initialize Evaluation
Meeting for Performance discussion
Yes
Write Comments
Submit rating and Comment
Probation Extend?
Close Evaluation
No
Publish Reoport
View Evaluation Report

Fig(b)



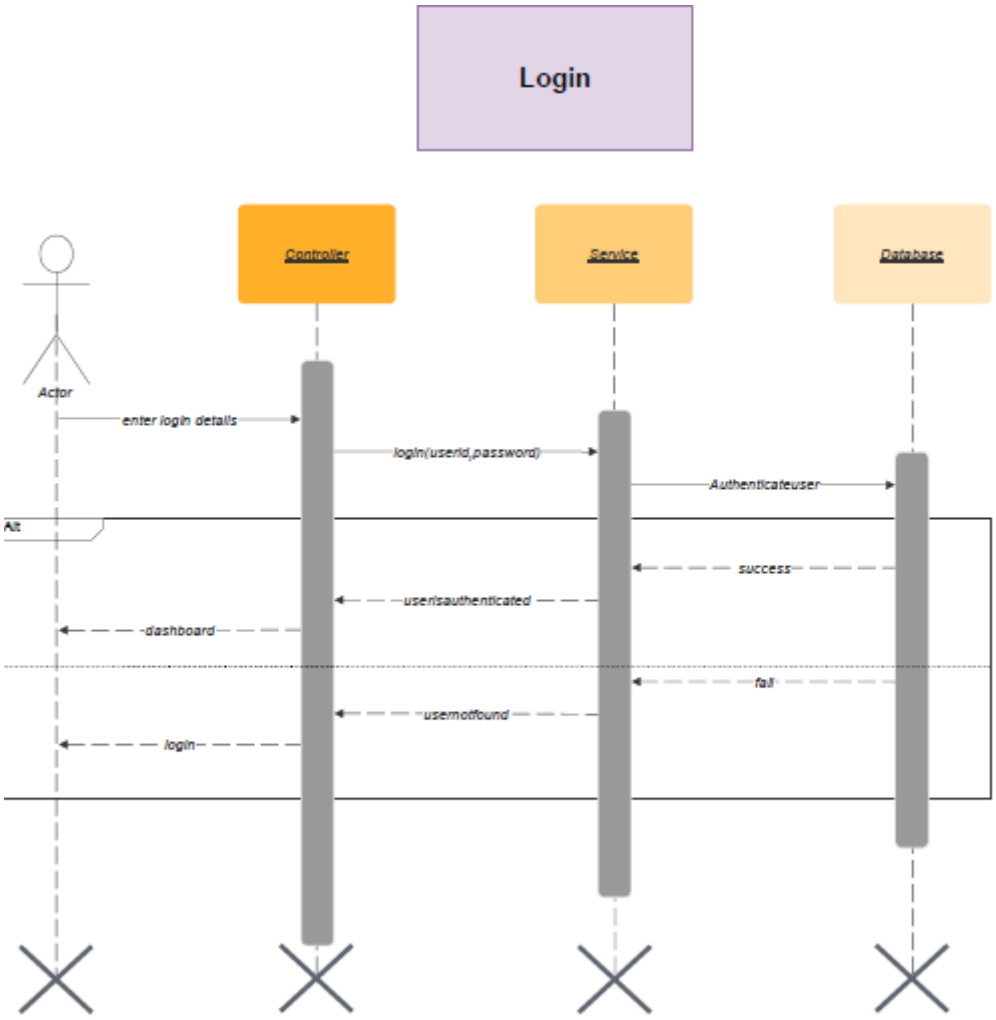
Fig(c)

4.2Activity Diagram

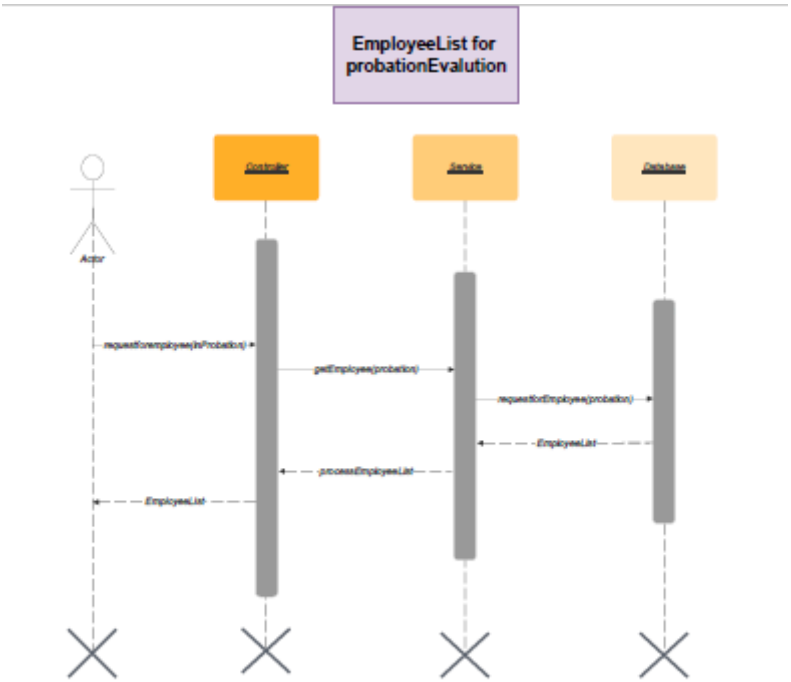
4.3 SEQUENCE DIAGRAM

- 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)

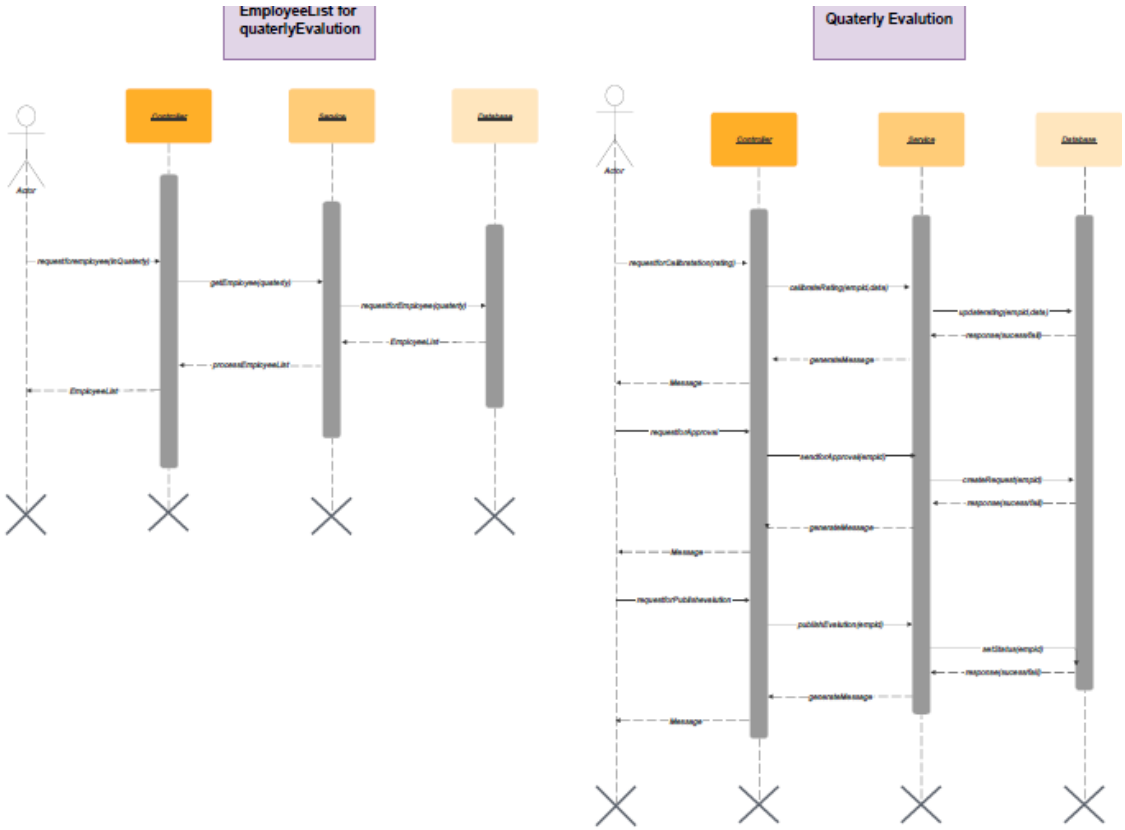
- high-level interactions between user of the system and the system, between the system and other systems, or between subsystems (sometimes known as system sequence diagrams)



Fig(a)



Fig(b)

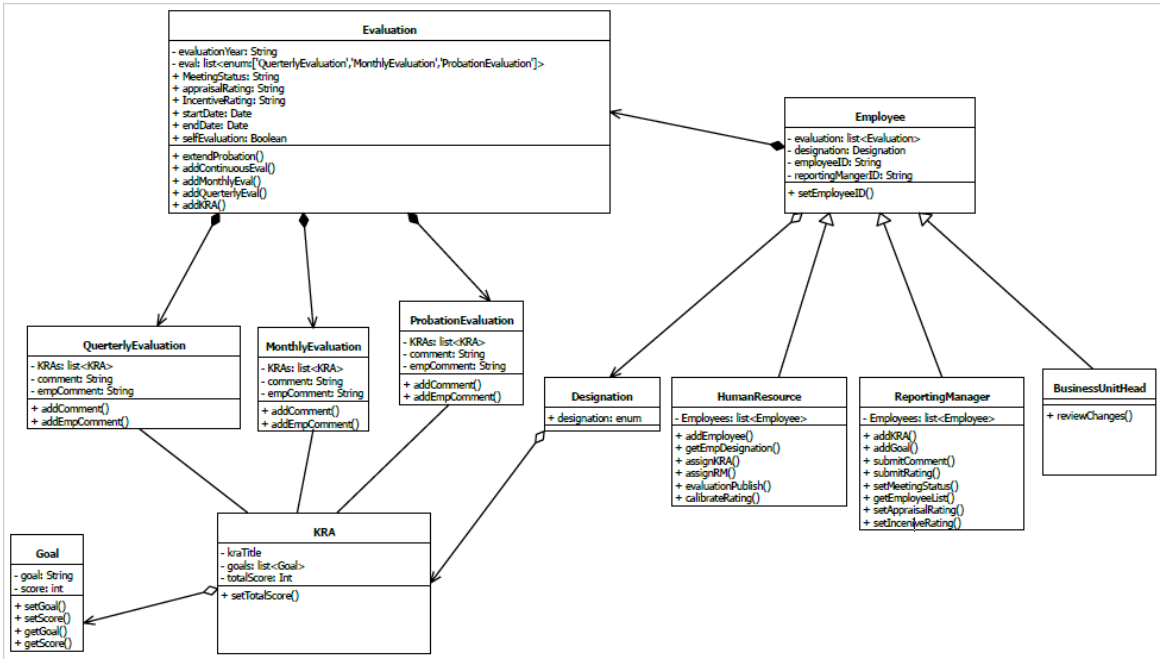


Fig(c)

4.3 Sequence Diagram

4.4 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 modeling 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.



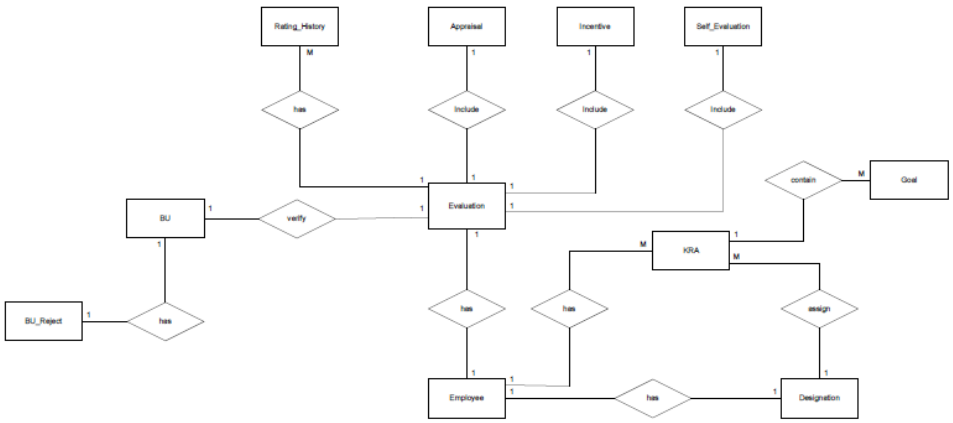
4.4 Class Diagram

4.4 E-R DIAGRAM

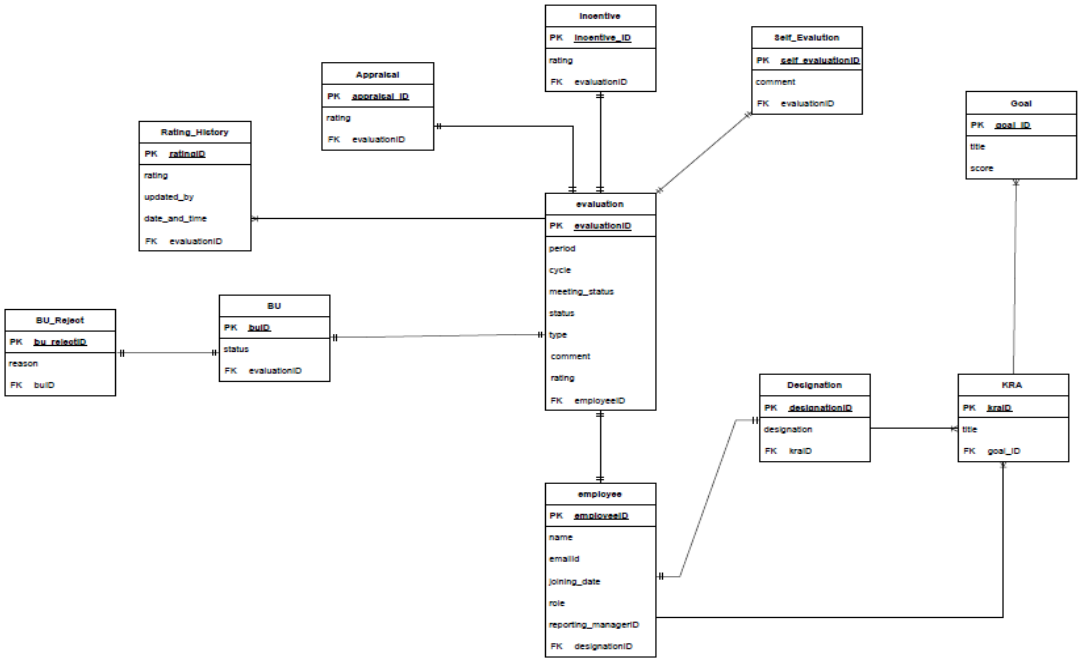
- An entity relationship diagram (ERD), also known as an entity relationship model, is a graphical representation that depicts relationships among people, objects, places, concepts or events within an information technology (IT) system. An ERD

uses data modeling 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(a)



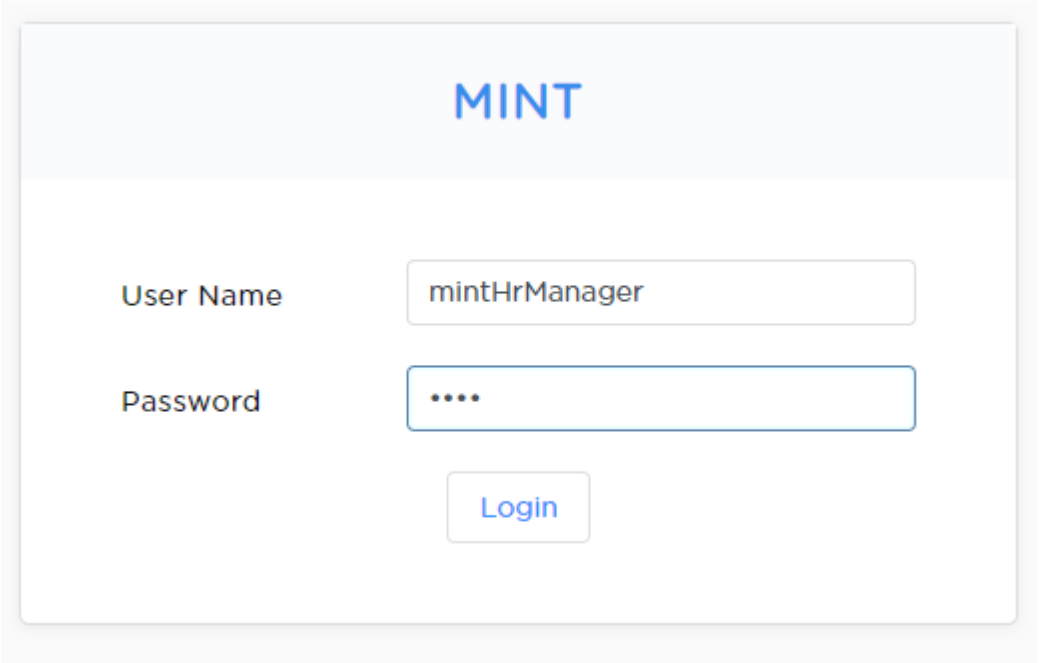
Fig(b)

4.5E-R Diagram

5. USER MANUAL

5.1 Login

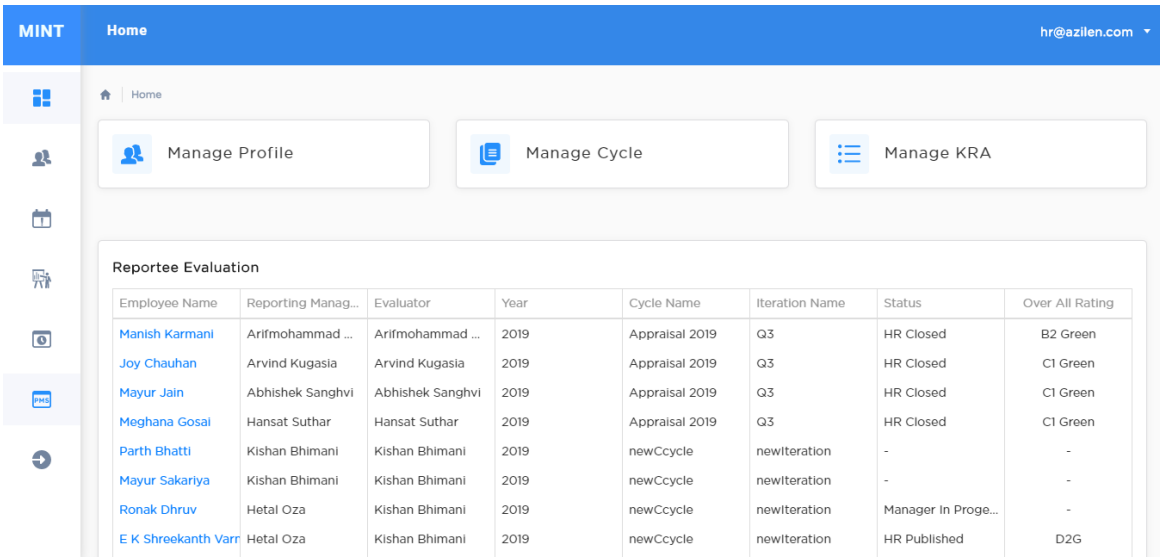
- Login Page, where username and password will be validated.



5.1 Login Page

5.2 Home screen of HR

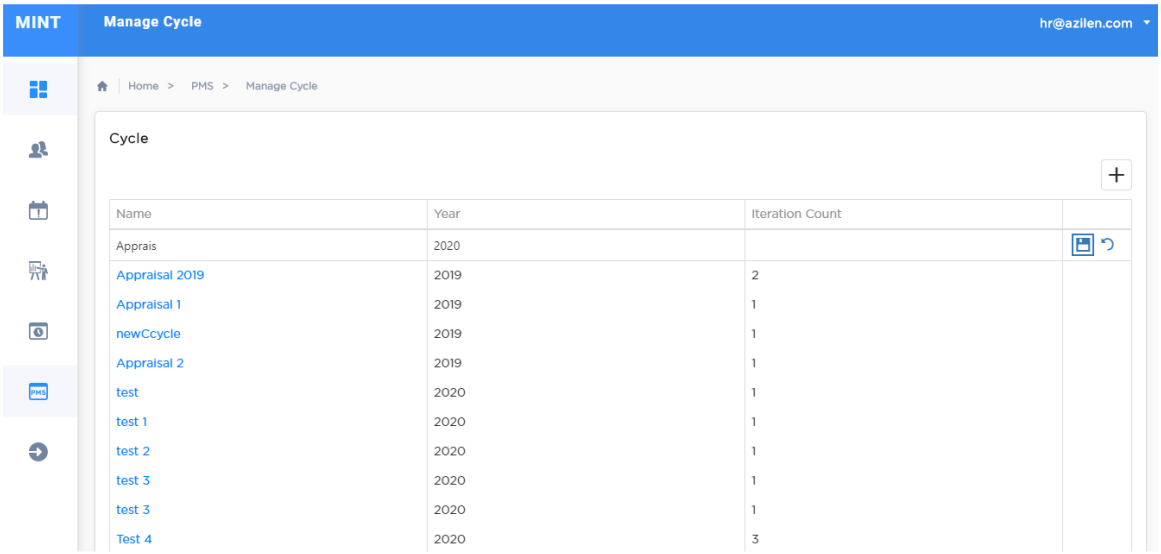
- After login page, the user will be redirected to their dashboard where, HR will be able to see the list employees of Repartee Evaluation.



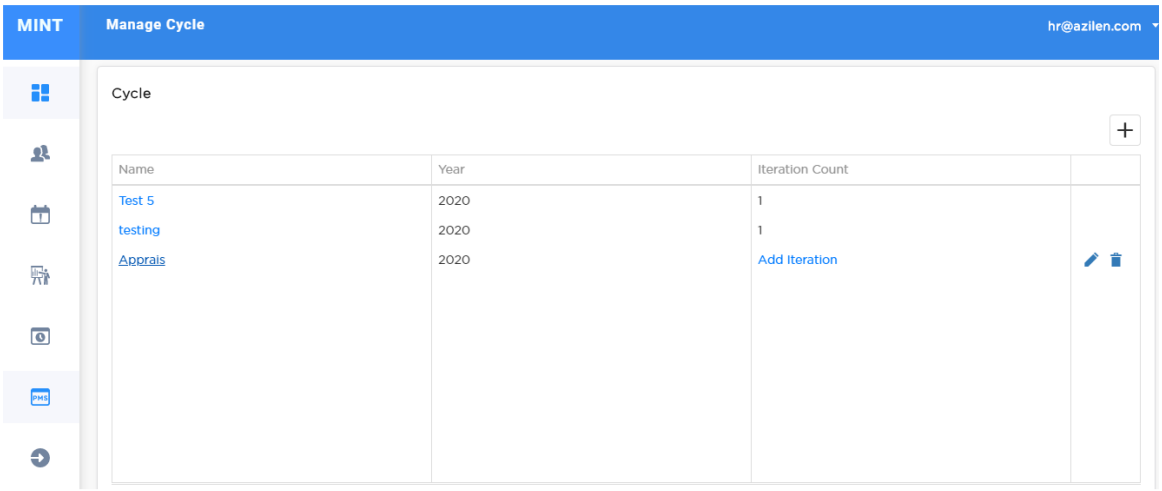
5.2 Home Screen of HR

5.3 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(a)

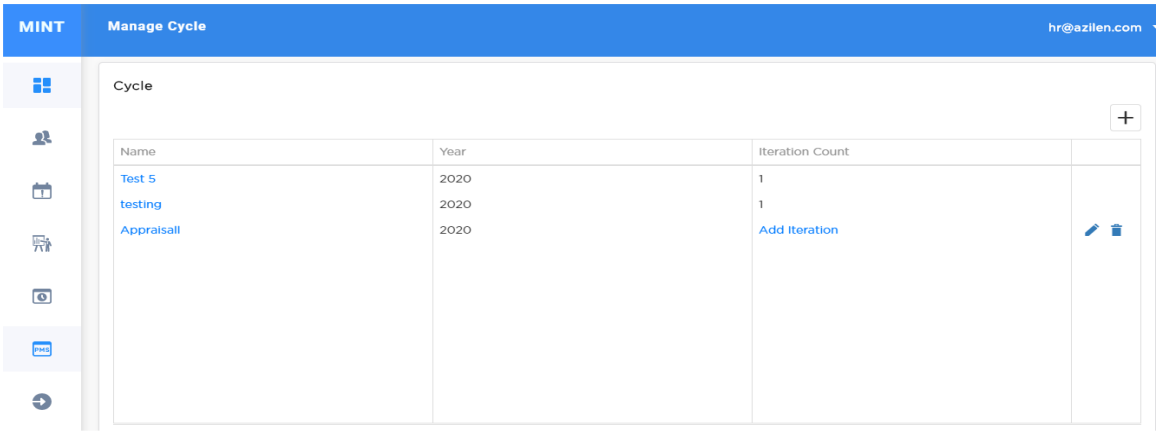


Fig(b)

5.3 Create Cycle

5.4 Edit Cycle

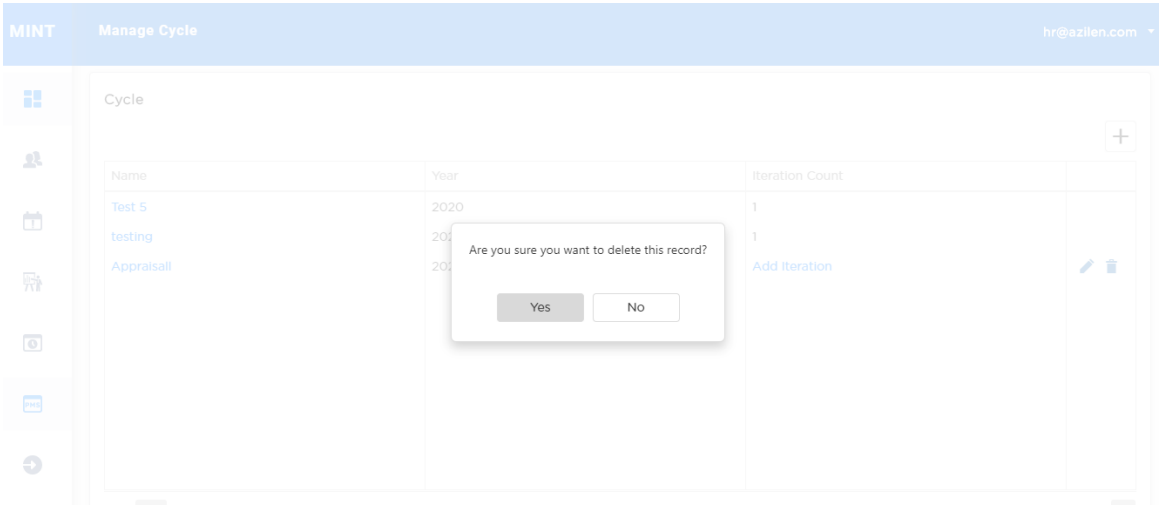
- There is an option of Edit (update) using which you can make changes in that.
- After clicking on Edit sign, cycle will become editable, so that you can make changes.



5.4 Edit Cycle

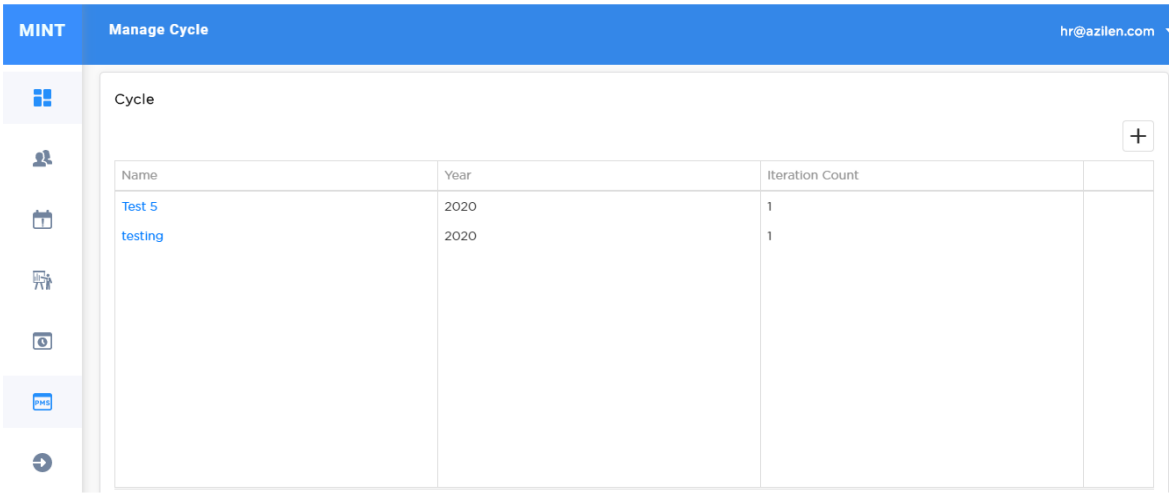
5.5 Delete Cycle

- 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(a)

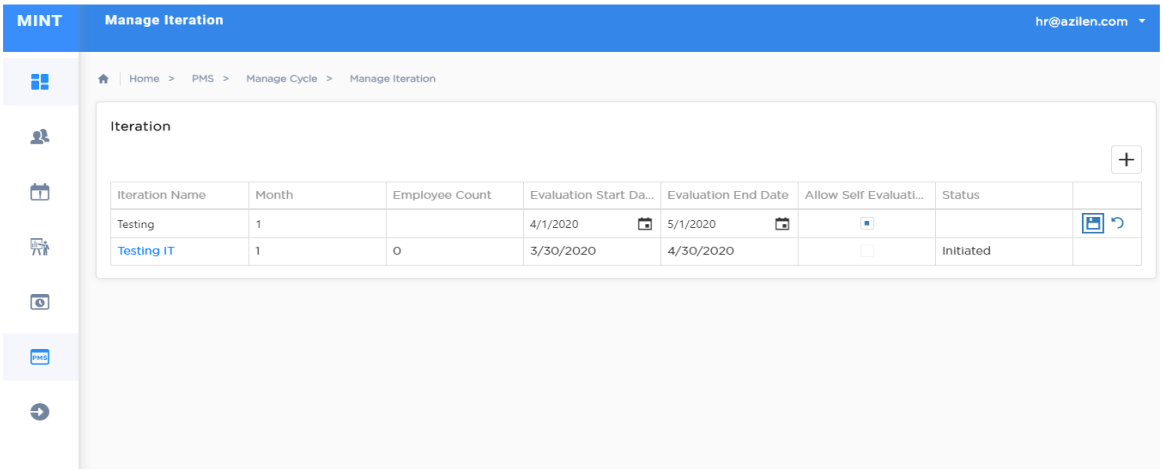
- If you click yes, then the changes will be reflected as seen below.



Fig(b)
5.5 Delete cycle

5.6 Create Iteration

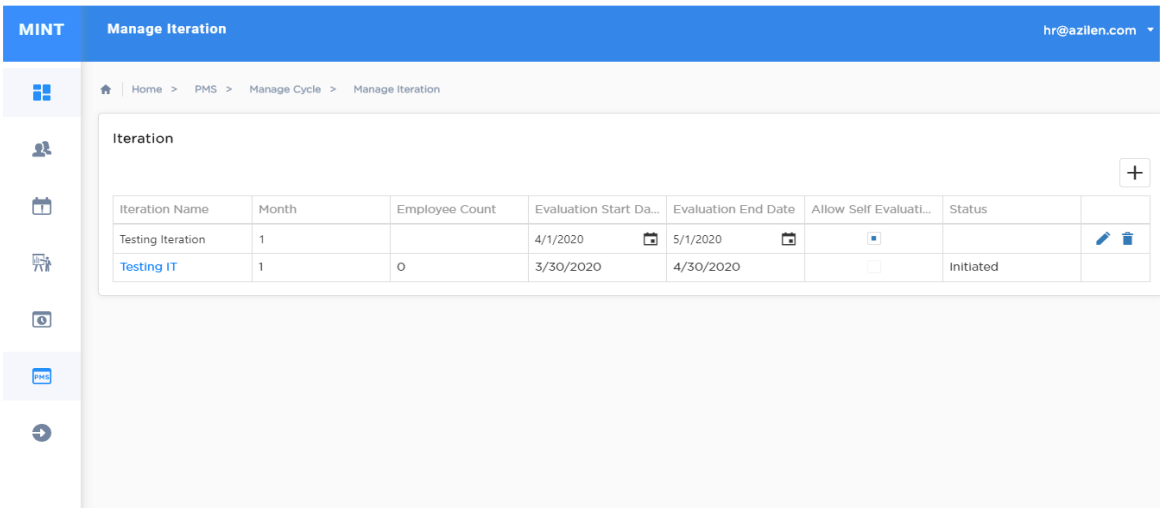
- By clicking on ‘Add Iteration’ user will be redirected to this page.



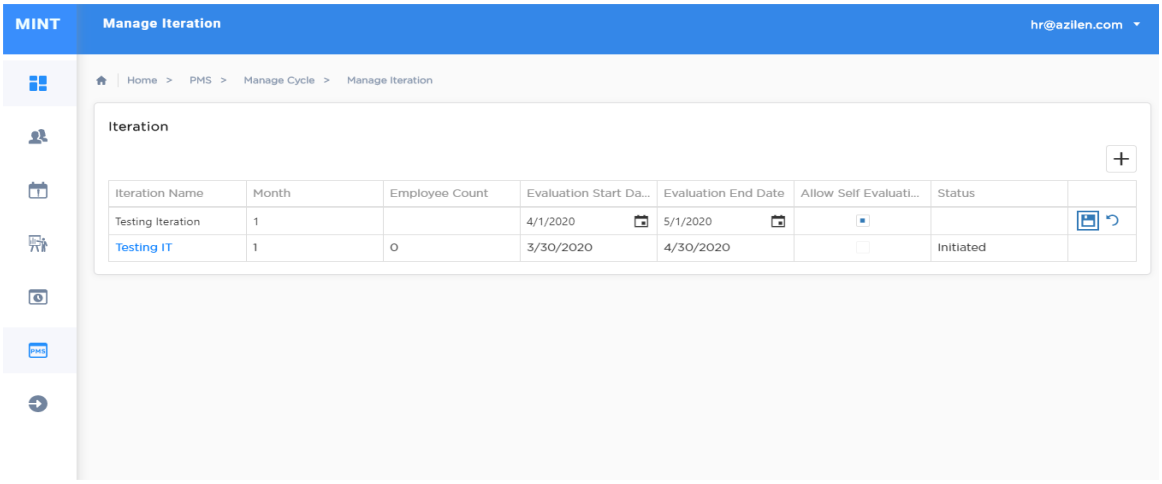
5.6 Create Iteration

5.7 Edit Iteration

- User can edit the filed over here.



Fig(a)

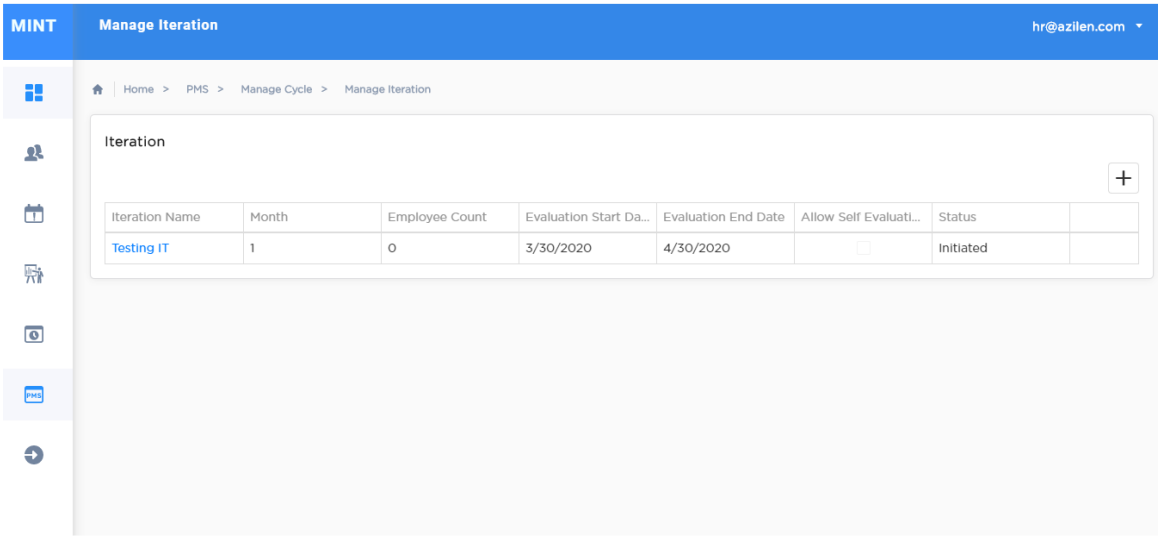


Fig(b)

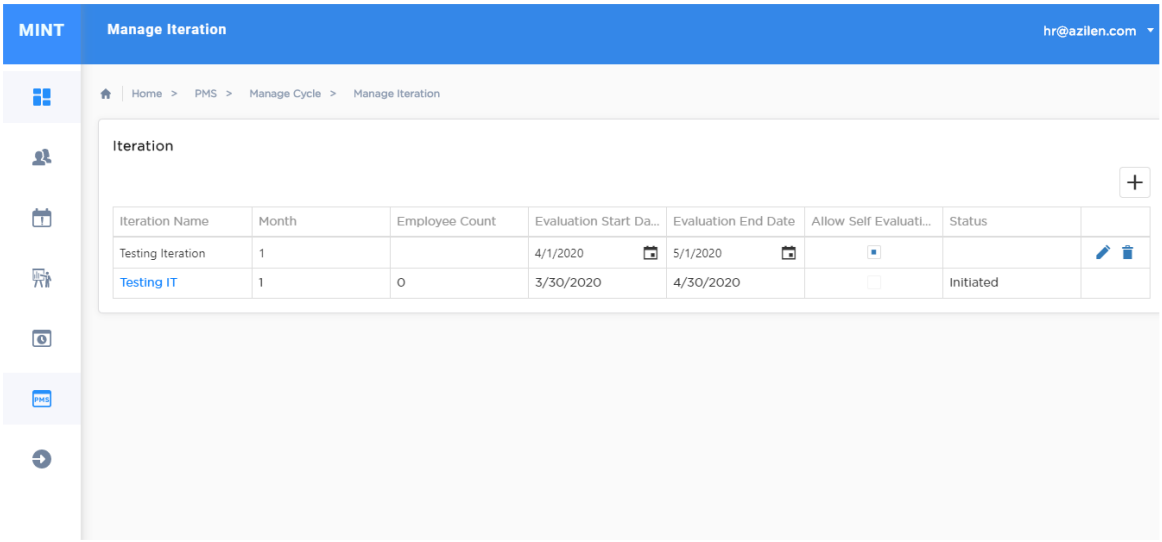
5.7 Edit Iteration

5.8 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(a)

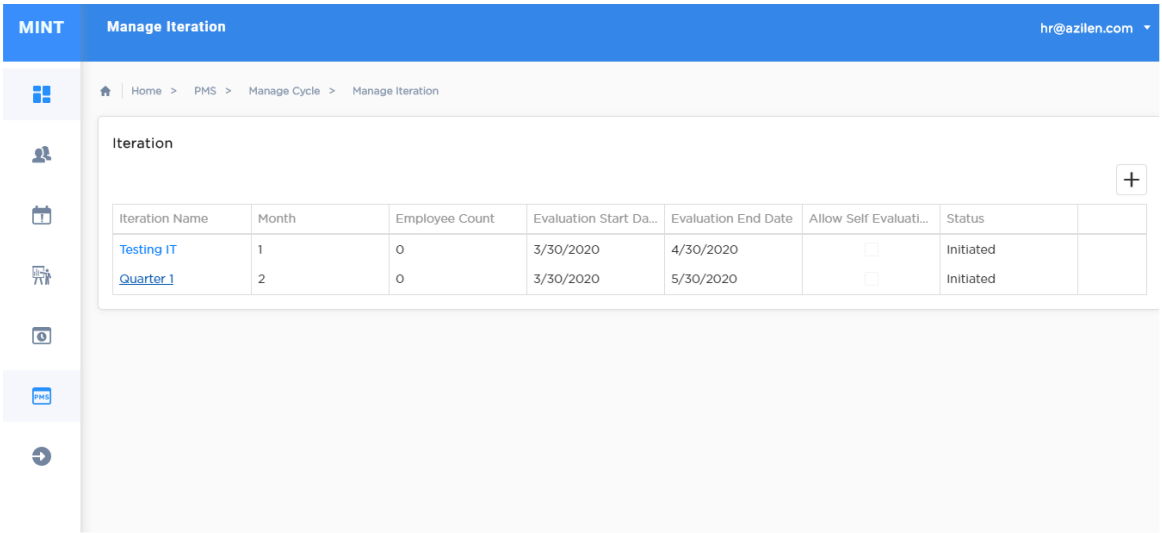


Fig(b)

5.8 Delete Iteration

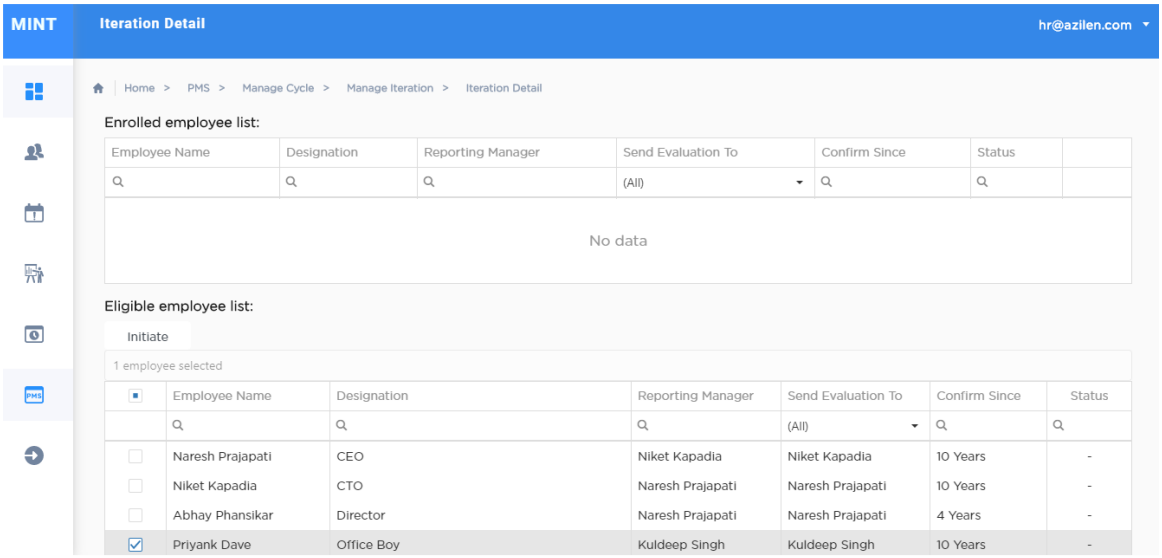
5.9 Employees and Assigning Project Manager

- Below image shows the new iteration added and in that employees are added and assigned Project Manager.

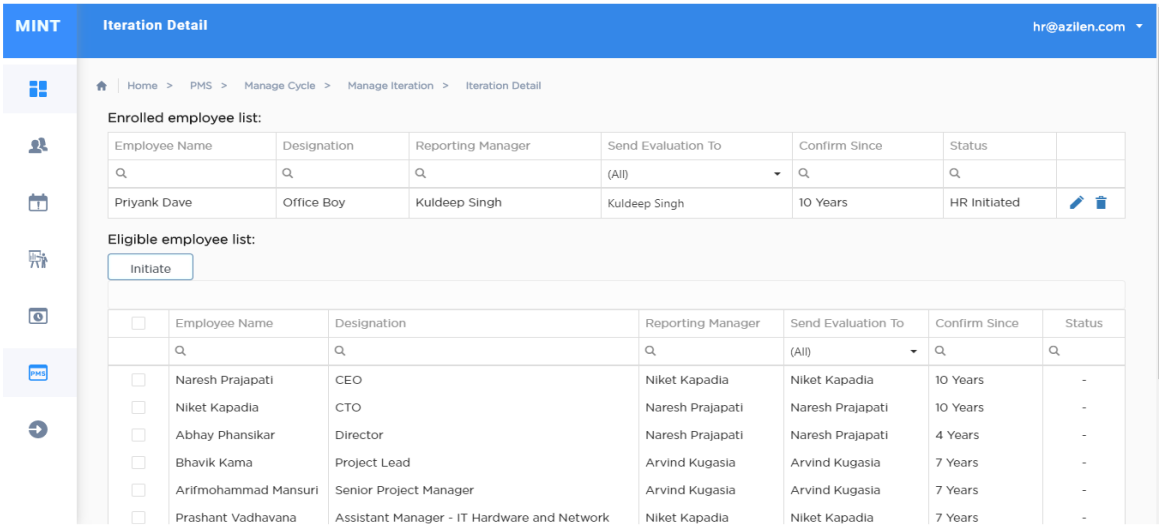


Fig(a)

▪ In the below two images employees are added.

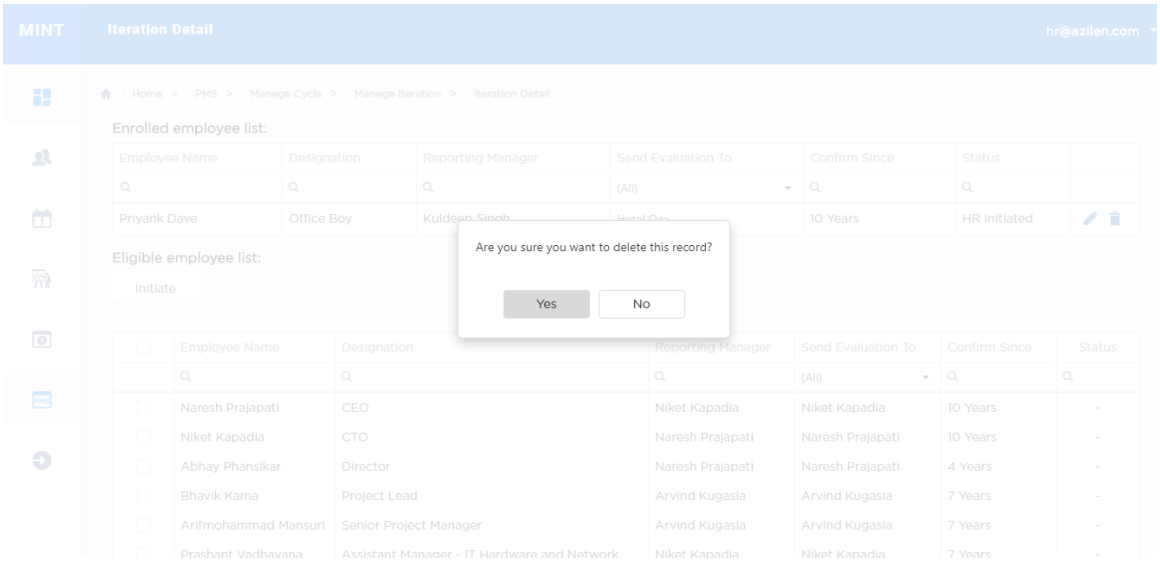


Fig(b)

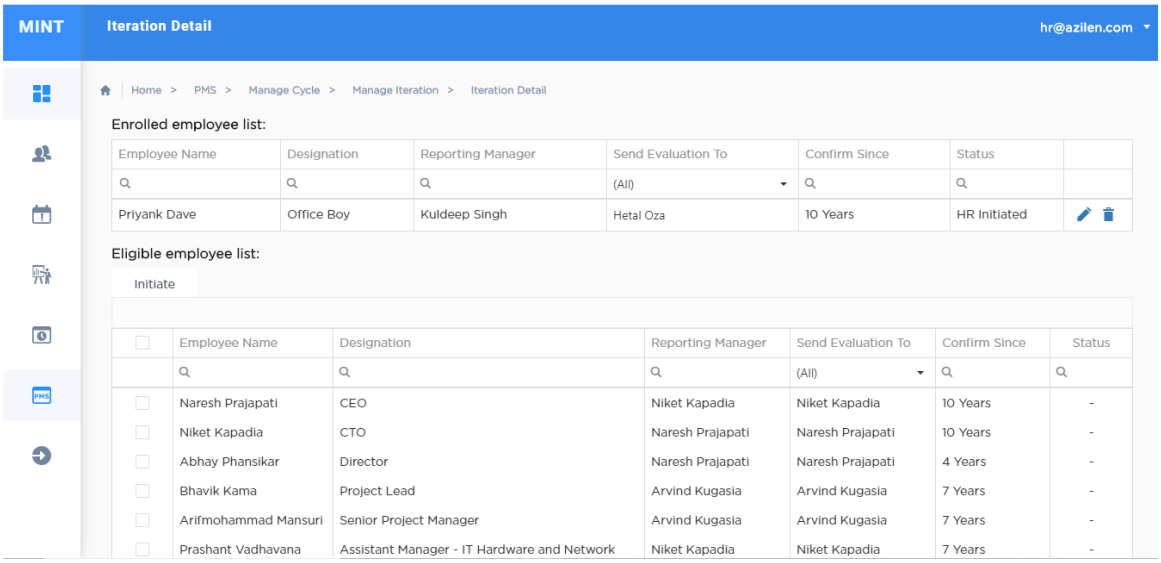


Fig(c)

- As shown in below images, we can edit Project Manager of an employee and also can delete that employee from the iteration.



Fig(d)

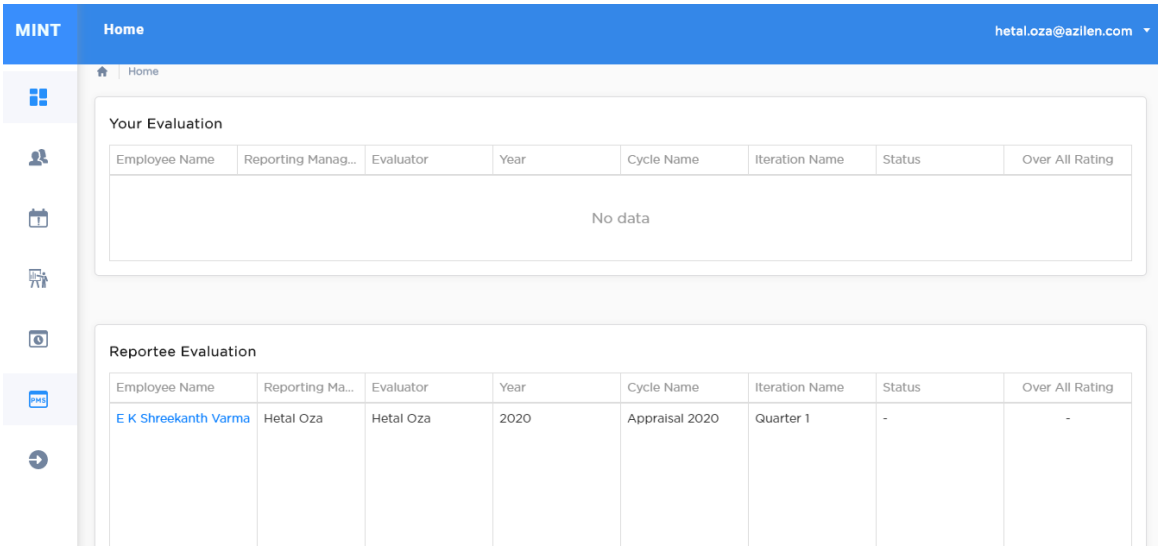


Fig(e)

5.9 Employees and Assigning Project Manager

5.10 Project Manager Screen

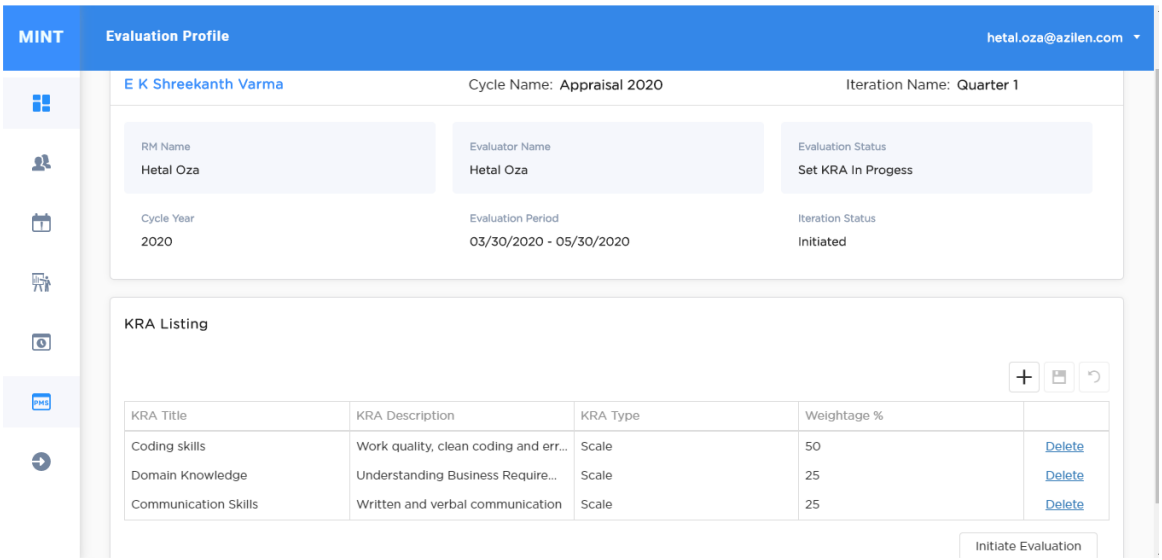
- In the below image list of employees are displayed which are either evaluated or to be evaluated.



5.10 Project Manager Screen

5.11 ADD KRAs

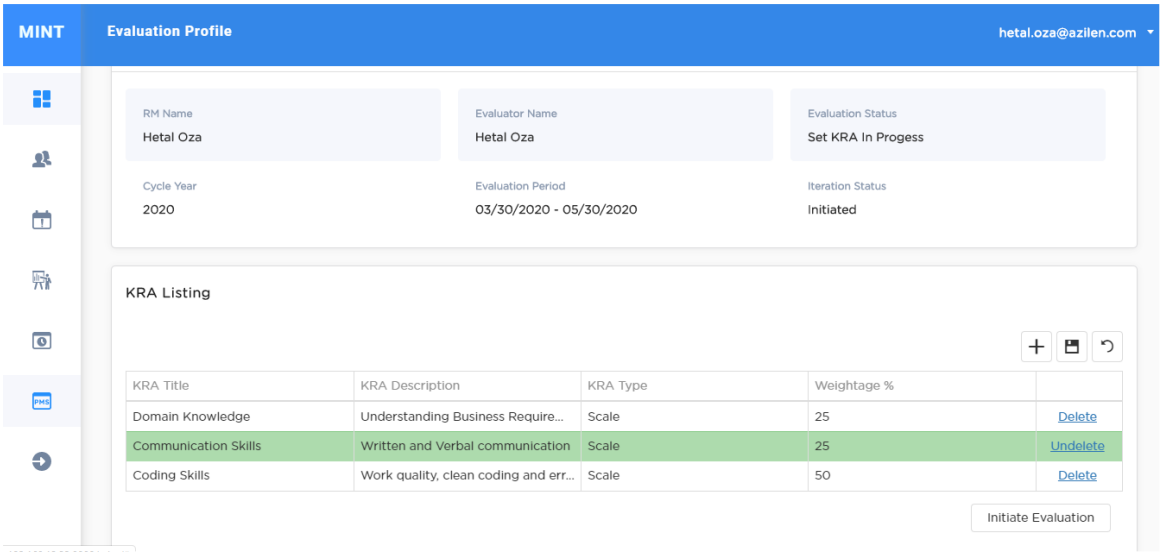
- When Project Manager clicks on *E K Shreekanth Varma* then the below screen will be displayed and can add in it.



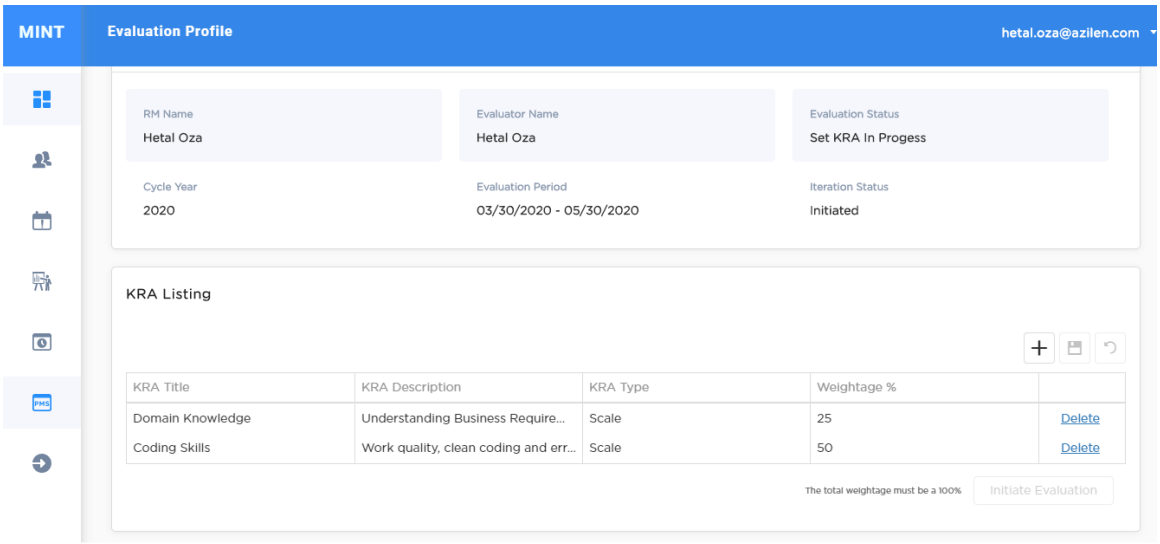
5.11 ADD KRAs

5.12 Delete KRA

- By clicking on *Delete* link we can delete KRA.



Fig(a)

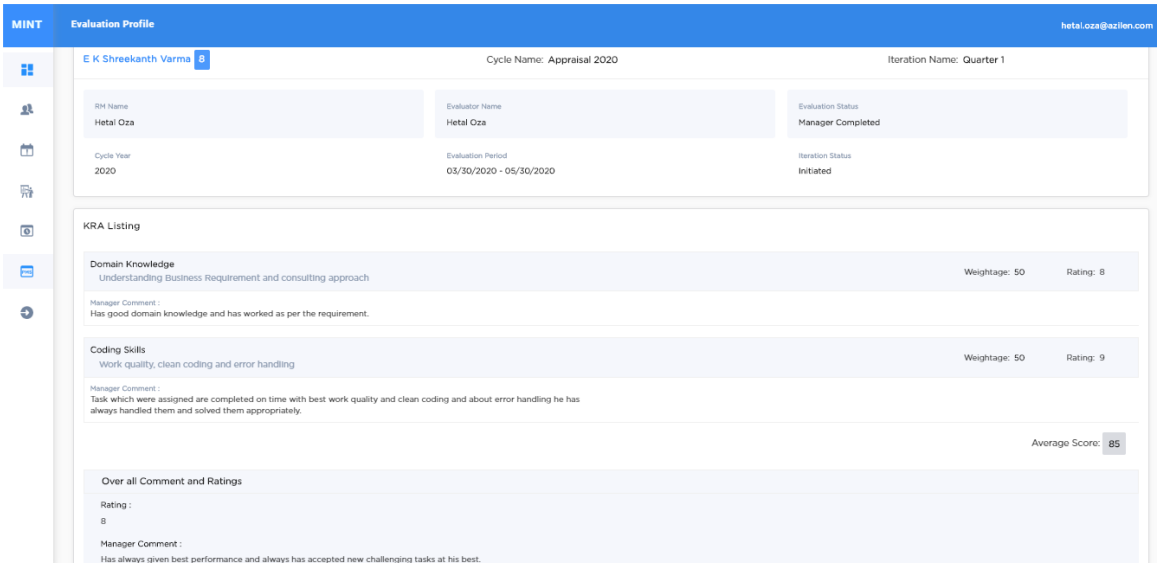


Fig(b)

5.12 Delete KRA

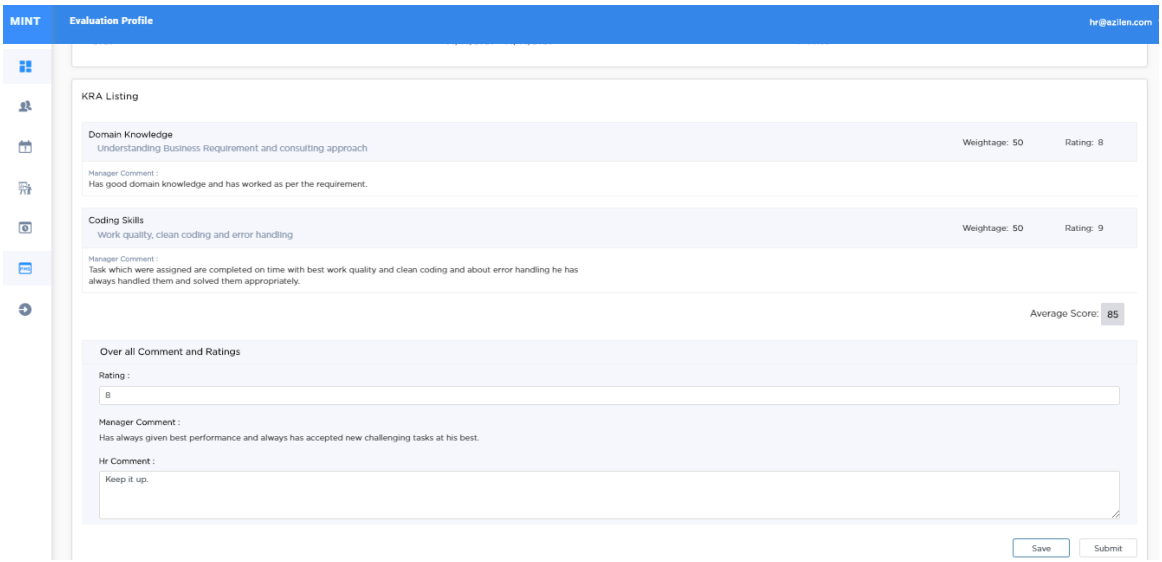
5.13 Give Comments and Ratings

- After adding comments and ratings HR can able to see it. (Refer the image below the below image)



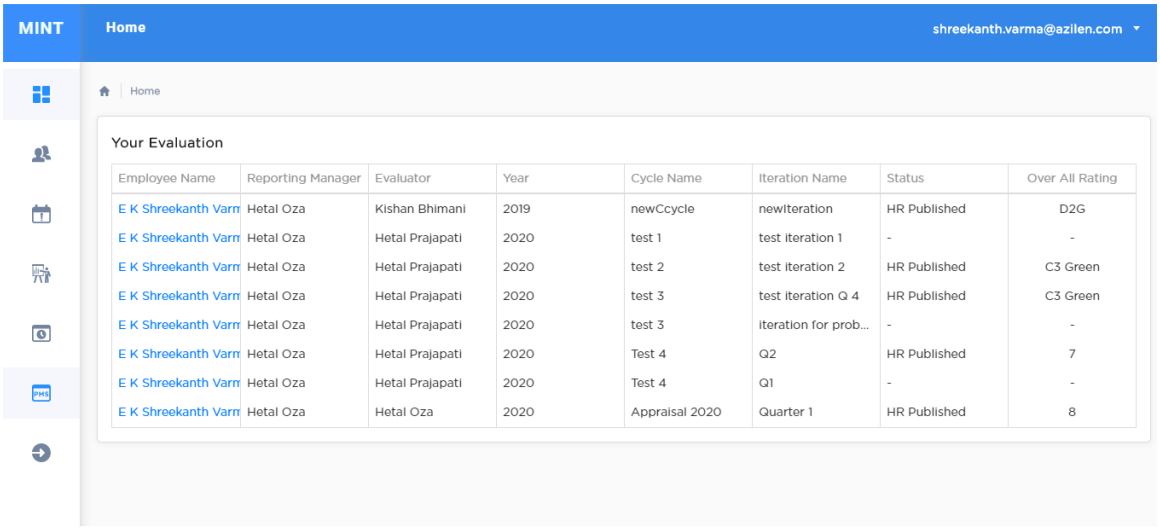
Fig(a)

- HR can see the comments and ratings and will also give comment as shown in below image.



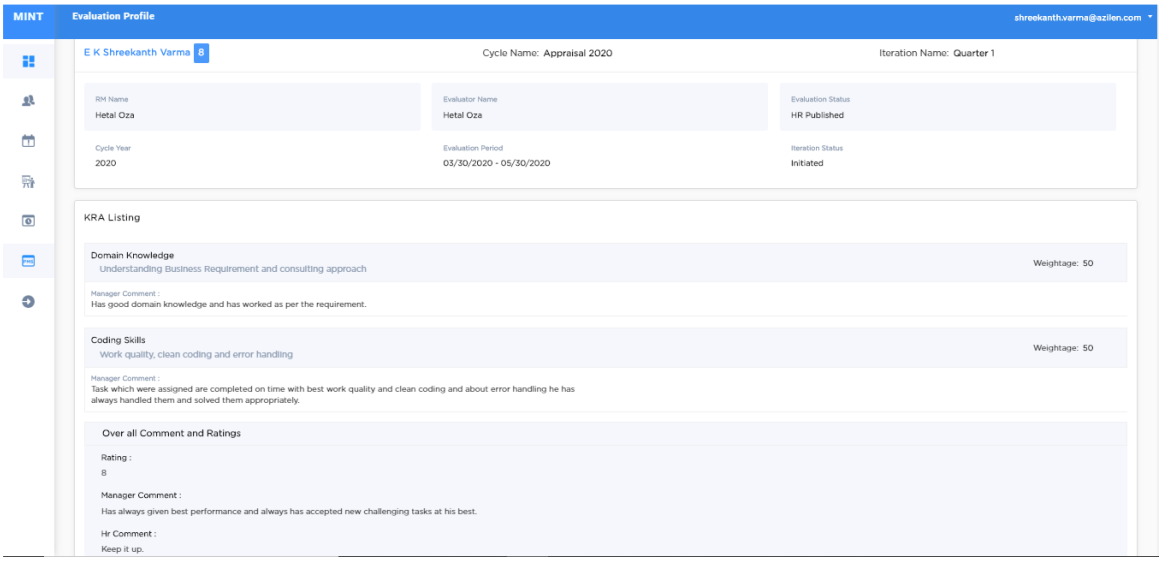
Fig(b)

- After submitting it employees will able to see their lists evaluation as shown in below image (Appraisal 2020).



Fig(c)

- By Clicking on *E K Shreekanth Varma (Appraisal 2020)* the whole detail will be displayed.

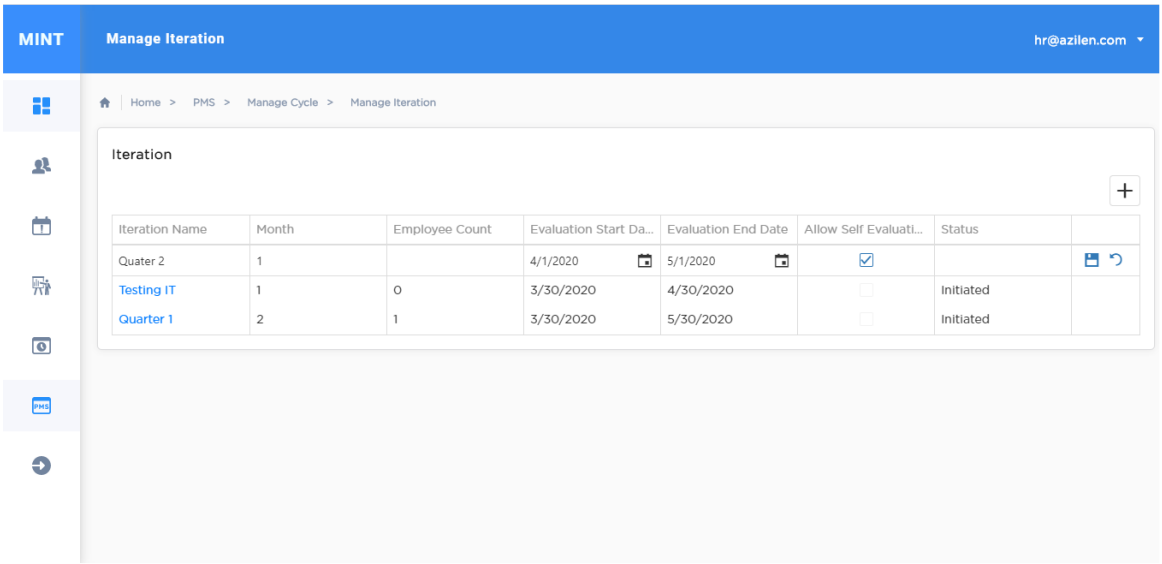


Fig(d)

5.13 Give Comments and Ratings

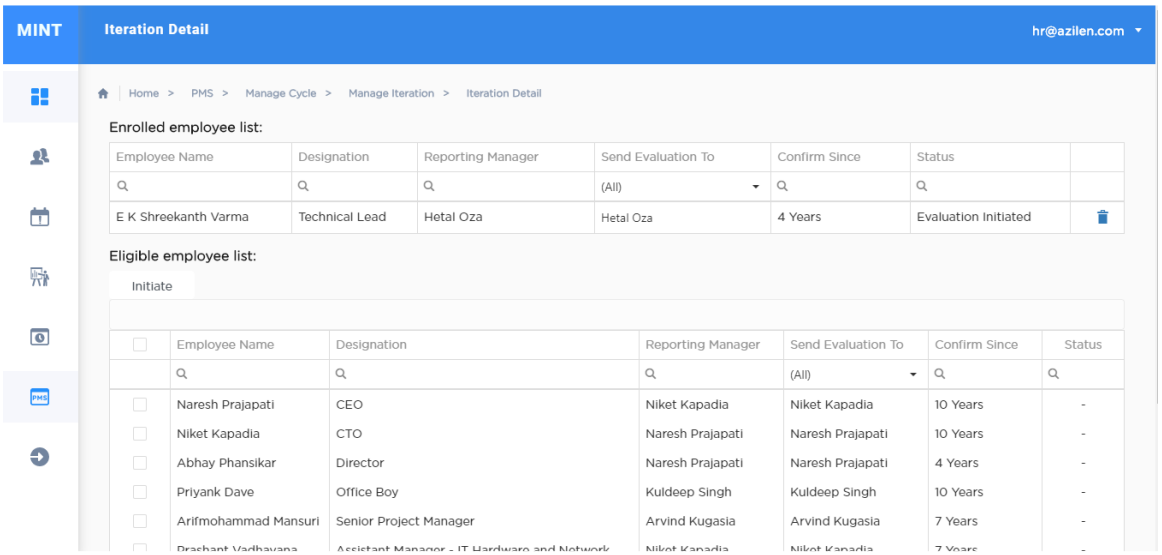
5.14 Self-evaluation

- In below image HR has added created new iteration.
- *Iteration name: Quarter 2* and as checked *"Allow Self Evaluation"* checkbox.



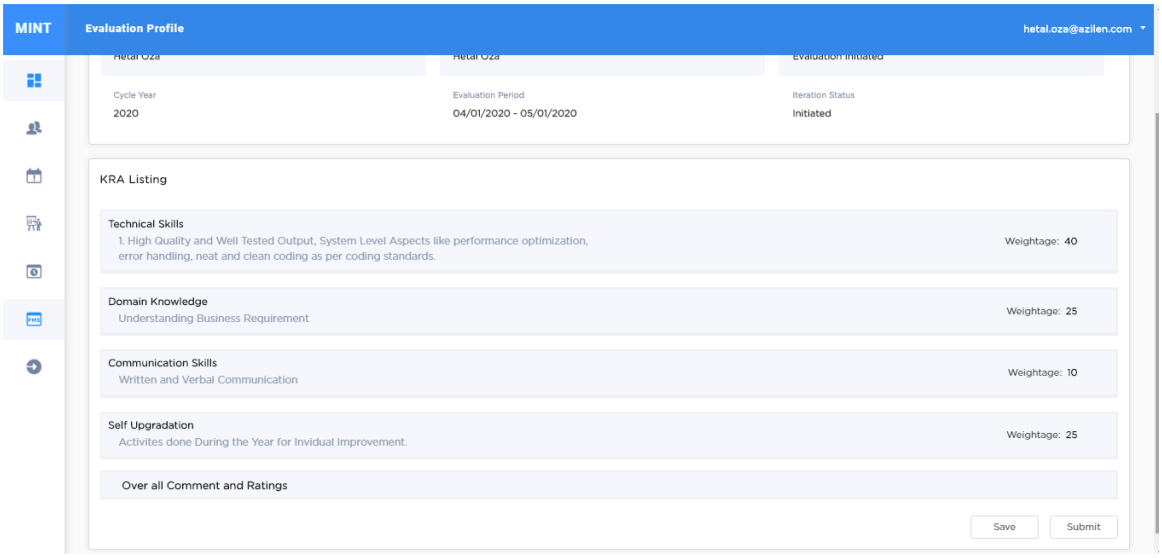
Fig(a)

- Here, employee is added in that iteration.



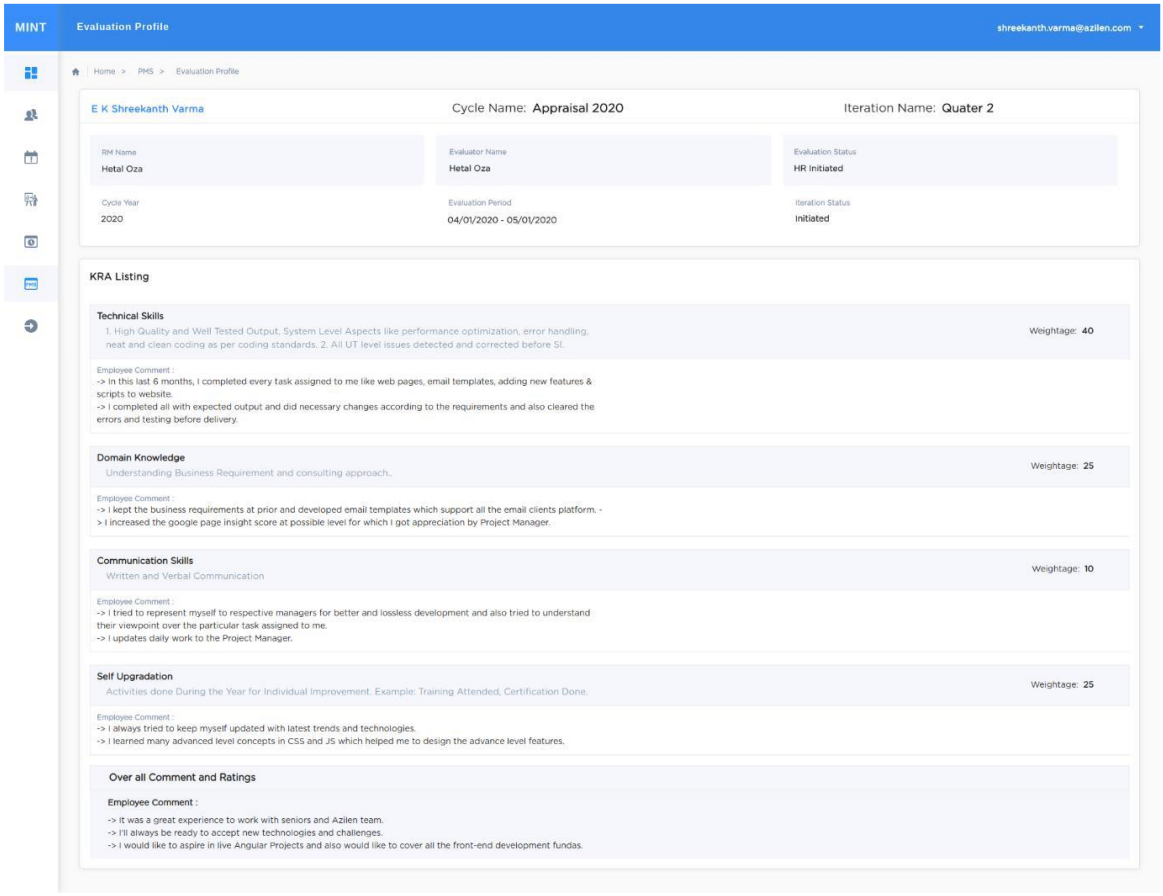
Fig(b)

- Project Manager has added KRAs for the employee.



Fig(c)

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

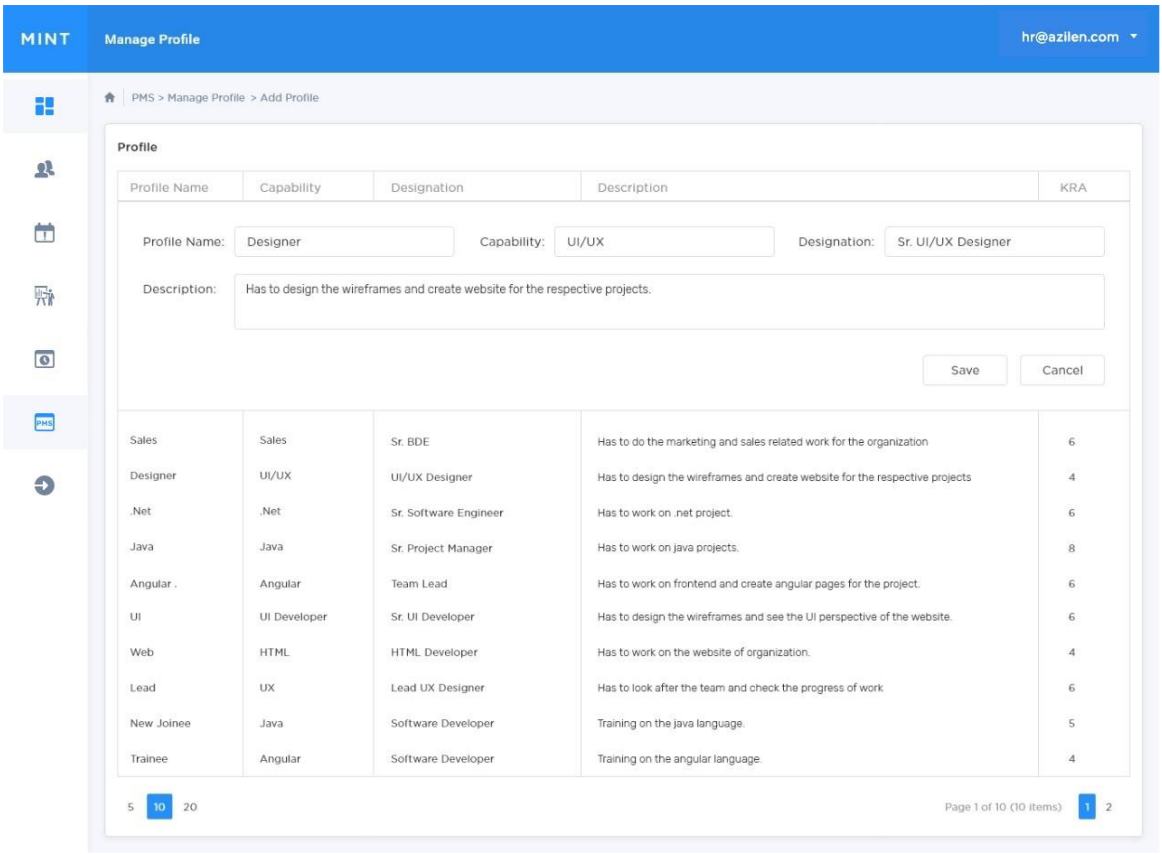


Fig(d)

5.14 Self-evaluation

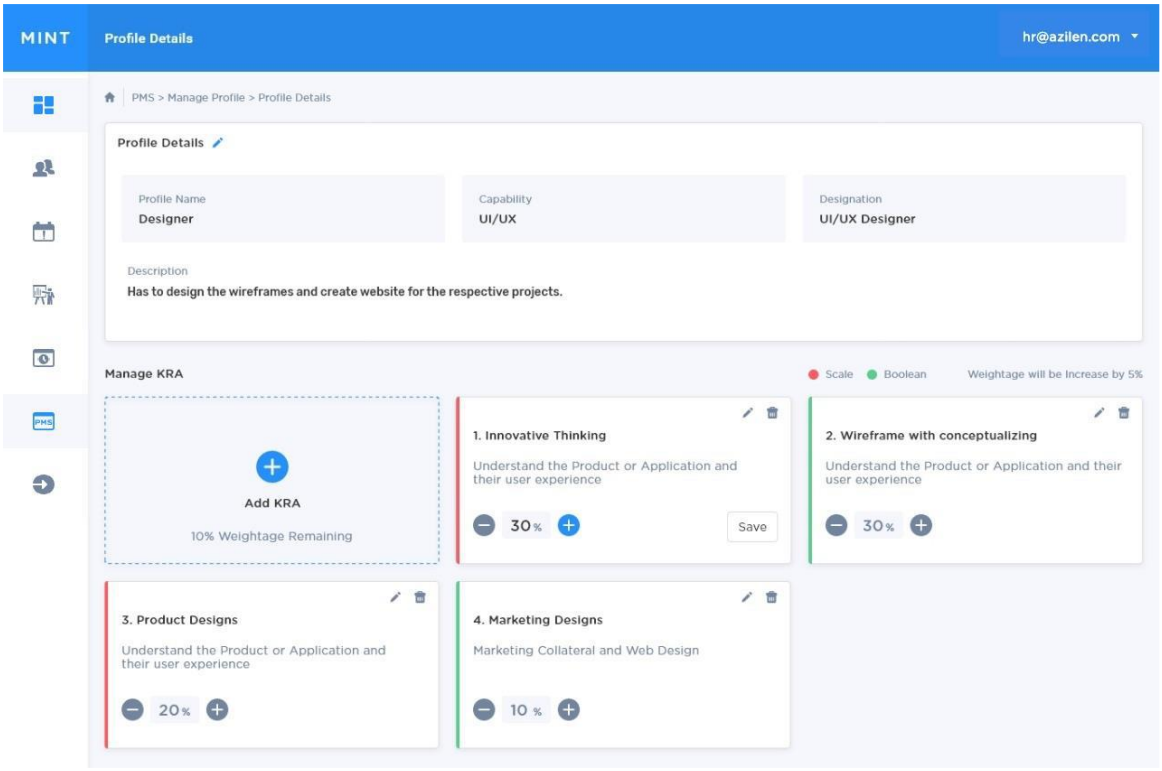
5.15 Create Profile

- We can Create Profile as shown in below screen.



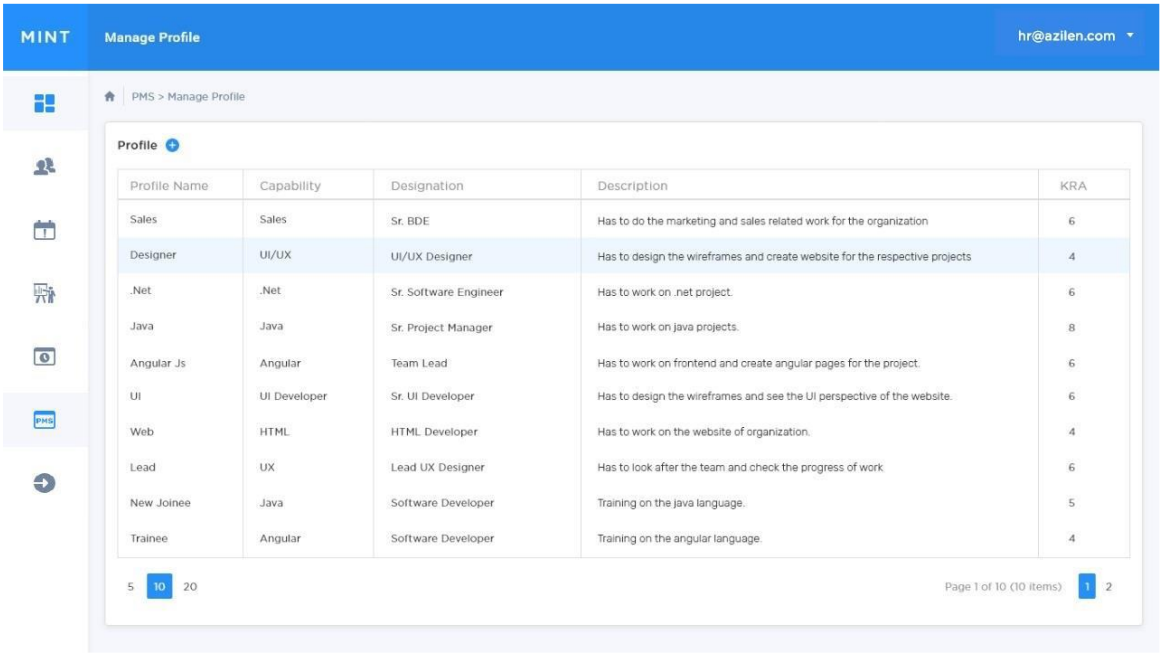
Fig(a)

- Here, we can add KRAs as shown below.



Fig(b)

- Below image shows the that KRA has been added.



Fig(c)

5.15 Create Profile

6. TESTING

6.1 TESTING PHASE

6.1.1 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.

6.1.2 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,Add ratings

6.2 TESTING TYPES:

- The following types of testing are for this project:

1. **Unit Testing:**
   - In this testing following are the few tested things which were tested:
   - Testing if user is able to login successfully.
   - Checked if HR is able to see the dashboard having Manage Profile, Manage Cycle, 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 that iterations and assign Project Mangers.
   - Checked if employees are able to see their profiles.

2. **Integration Testing:**
   - In this system following are the few tested things which were tested:
   - 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 their evaluation done by Project Manager.
   - Checked if Project Manager is able to navigate to employee profile by clicking on it to add KRAs.

3. **System Testing:**

- Following are the few tested things which were tested:
    a. Checked if the HR is able to create cycle and iteration with adding employees and assigning those employees Project Manager.
    b. Checked if those employees are displayed on the dashboard of HR for evaluation.
    c. Checked if Project Managers is able to see the employees which are allocated to them.
    d. Checked if Project Managers are able to add KRAs to that employee and also evaluate them.
    e. Checked if employees are able to view their evaluation which done by Project Managers.

7. FUTURE ENHANCEMENT

- ▪ 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.

8. CONCLUSION

- 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.

9. APPENDIX

9.1 TOOLS USED

1. Eclipse

- Eclipse is an integrated development environment used in computer programming. It includes a base workspace and an extensible plug-in system to optimize the environment

2.Mongo db.

- MongoDB is a cross-platform document-oriented database program. MongoDB is developed by MongoDB Inc. and licensed under the Server-Side Public. Classified as a NoSQL database program, MongoDB uses JSON-like documents with schema.

3.Visual Studio code

- Visual Studio Code is a source code editor developed by Microsoft, Windows, Linux and macOs.
- This includes support for embedded Git and debugging, syntax highlighting, intelligent code completion, snippets and code refactoring.

4. draw.io, Visual Studio Paradigm, Smart Draw

- Above tools were used to draw the diagrams.

10. REFERENCES

- Following are the sites that are used in Internship:

  - https://www.pluralsight.com/
  - https:/www.javatpoint.com/spring-boot-tutorial
  - https://www.baeldung.com/spring-boot
  - https://angular.io/
  - https://angular.io/tutorial/toh-pt6
  - https://js.devexpress.com/
  - https://stackoverflow.com/
  - https://www.freelancinggig.com/blog/2018/11/02/what-is-the-difference-between-api-and-rest-api/
  - https://dzone.com/articles/microservices-communication-zuul-api-gateway-1
  - https://app.pluralsight.com/library/courses/springmvc-intro/table-of-contents
  - https://www.tutorialspoint.com/spring/index.htm
  - https://www.n-ix.com/microservices-vs-monolith-which-architecture-best-choice-your-business/
  - https://insights.daffodilsw.com/blog/monolithic-vs-microservices-which-is-the-better-architecture-for-ecommerce-app-development
  - https://www.petrikainulainen.net/programming/spring-framework/spring-data-jpa-tutorial-part-nine-conclusions/
  - https://www.javacodemonk.com/spring-cloud-and-its-advantages-3ac60b2c
  - https://www.youtube.com/watch?v=P1mlC8Ar0_k
  - https://www.baeldung.com/spring-boot-keycloak
  - https://www.keycloak.org/docs/latest/securing_apps/
  - https://medium.com/@techgeek628/easily-secure-your-spring-boot-applications-with-keycloak-41e09acc88fd