Human Resources Data Warehousing Management System

CSED 2020 graduation project

Project Advisor : Prof. Dr. Yousry Taha

Team Members

Ashraqat Sheta

Fatma Ibrahim

Marina Zakaria

Abdelrahman Said

Mohamed Gaber

Mohamed Said

1. Project Scope & Motivation

1.1 Project Scope

• Web Application provides several services for HR(s) and Managers to help them manage all aspects of the workforce efficiently.

• provide interfaces for employees to be more involved with Hr and Managers transparently.

• use data warehousing to provide better performance and data analytics for decision making and prediction .

1.2 Project Motivation

The workforce is the basic element of any corporation .

if data was accurately analyzed and managed , generating decisions and predictions that would save resources and achieve massive profits in case analytics were accurate enough to read the future.

1.3 Problem statement

- The amount of data is massive , highly variant and distributed over many databases that are not connected or mapped for extracting valuable information .
- That leads to difficulty for HR employees to draw accurate purposeful analytics which might lead later on to taking wrong decisions that would cost the corporation an unnecessary resources or waste of already existed resources .

1.4 Solution proposed by the project

This data must be gathered in a Data-Warehouse that connects all variant aspects of the corporation where it can be managed dynamically, viewed, updated and analyzed applying data mining techniques with trusted predictions for better decision making and less HR employees.

2. Project Objectives

2.1 Employee self service

- Each employee is provided with a control panel to execute all eligite operations according to his position.
- Employee’s panel gives him all access to his personal data , attendance, logging records , progress and performance charts.

2.2 Attendance

- Each employee can view his attendance records with notes of (holidays , absence , presence and permissions).
- Each employee can get his attendance percentage records to monitor his progress over time.
- Hr employee can enter a permission , holiday or official vacation to be taken into account during calculating the employees’ attendance percentage calculation.
- Hr and managers can view the attendance reporting over time for specific employees , departments or positions to monitor any drop in attendance or any out of regular observation.

2.3 Feedback

At the end of every task , all members that worked on the task and the team leader must give feedback on the members skills.
This feedback is taken into account on calculating both project and employee’s performance.

2.4 Time tracking and task scheduling

- Each employee can see his active tasks and the required task not started yet.
- Each employee can choose to start a task but only the team leader can end a task when he thinks it's over.
- If the deadline of the task is passed and it’s not finished yet , the system would alert the employee.
- Each team leader can create a new project then create project tasks and assign employees to tasks which fit in slots of their time schedule in order to the time schedule to be consistent between tasks and training so that no 2 tasks or task and training can overlap.
- Each team leader would end the project when it’s over.

2.5 Talent Management

Each team leader would have a list of suitable employees for the task he’s creating ready for him based on the skill level the task needs and both time management and skill rank of other employees in a specific department or position to choose from.

2.6 Bonus and promotion

- HR can enter criteria of the bonus then view a list of employees that earn the bonus then grant the bonus.
- The manager can enter criteria of the promotion and view list of employees who would fit the promotion then he can choose one of them and promote him to the new position

2.7 Need for training

- Hr can create new training , choose employees from a recommended employees list by the system of employees who need the training based on the skill rank and enroll them in training after system inspecting if their time schedule is suitable for the training time.
- Each employee would view the list of training that he is currently enrolled in as a part of his time schedule.

2.8 Performance evaluation

- Each employee can keep track of his skill rank and his performance progress through time.
- Hr employees can view employee's performance progress reports by employee , position or department.
- The manager can view collective progress reports for employees , projects and the company as whole.

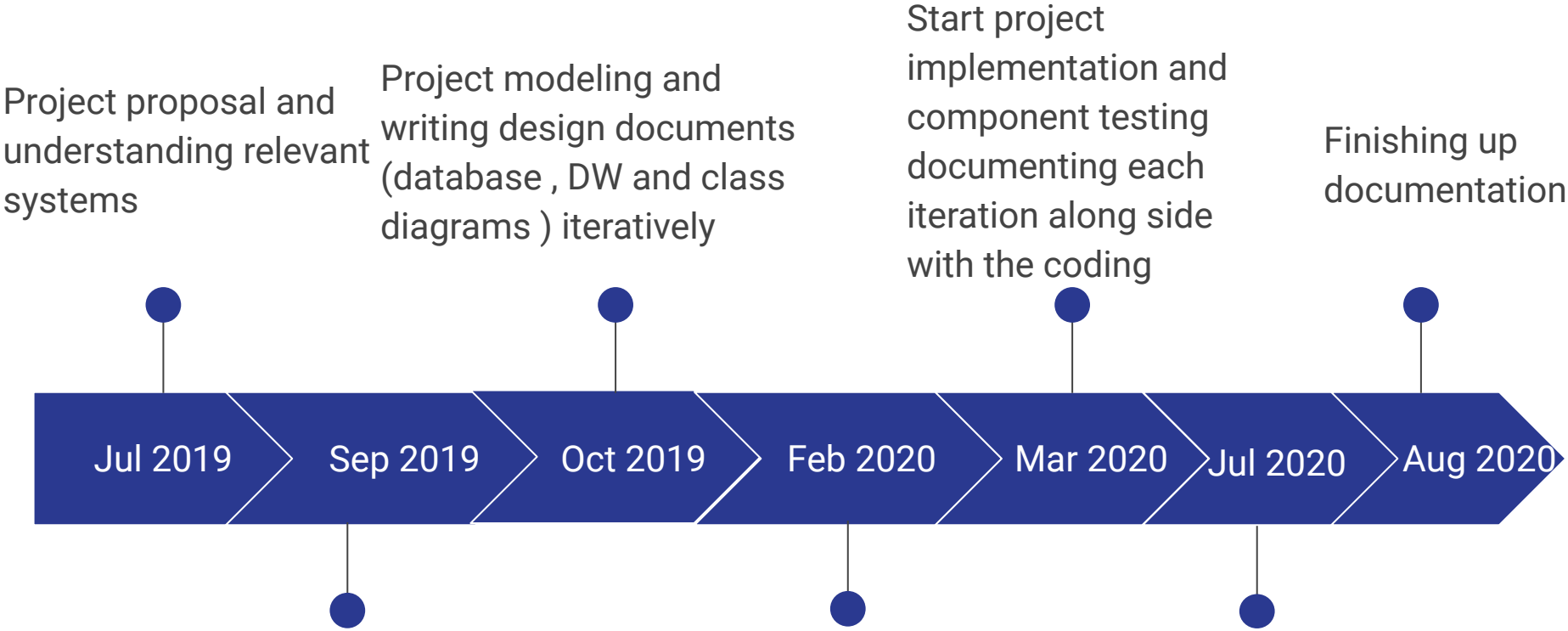
2.9 Strength and weakness fields analysis

- Manager can view the departments of his company in order from strongest down and the most powerful skills in each department based on projects of this department and relative skills.
- Manager can view the departments of his company in order from weakest up and the weakest skills in each department based on projects of this department and relative skills.

2.10 Recruitment

- Applicants can fill in the form with required data needed for the initial filtration process.
- Hr can view lists of recommended applicants based on job requirements they specified and the stored applications.
- Hr can hire new employees , register their data and give them their username and initial password.

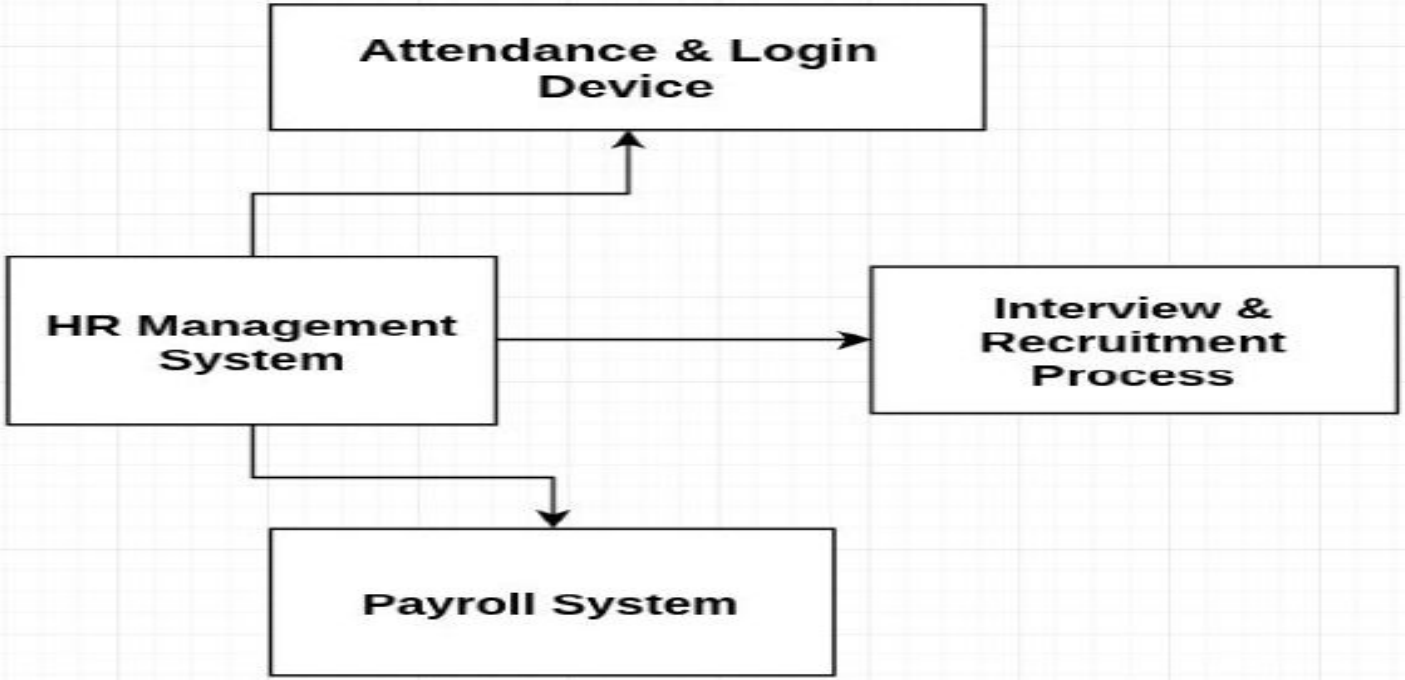
3. Time Plan


Project proposal and understanding relevant systems
Project modeling and writing design documents (database , DW and class diagrams) iteratively
Start project implementation and component testing documenting each iteration along side with the coding
Finishing up documentation
Jul 2019
Sep 2019
Oct 2019
Feb 2020
Mar 2020
Jul 2020
Aug 2020
Deciding project objectives and writing initial requirement document
Setting up and learning the frameworks (.Net visual studio , SQL DBMS and SSIS)
Overall system and performance testing

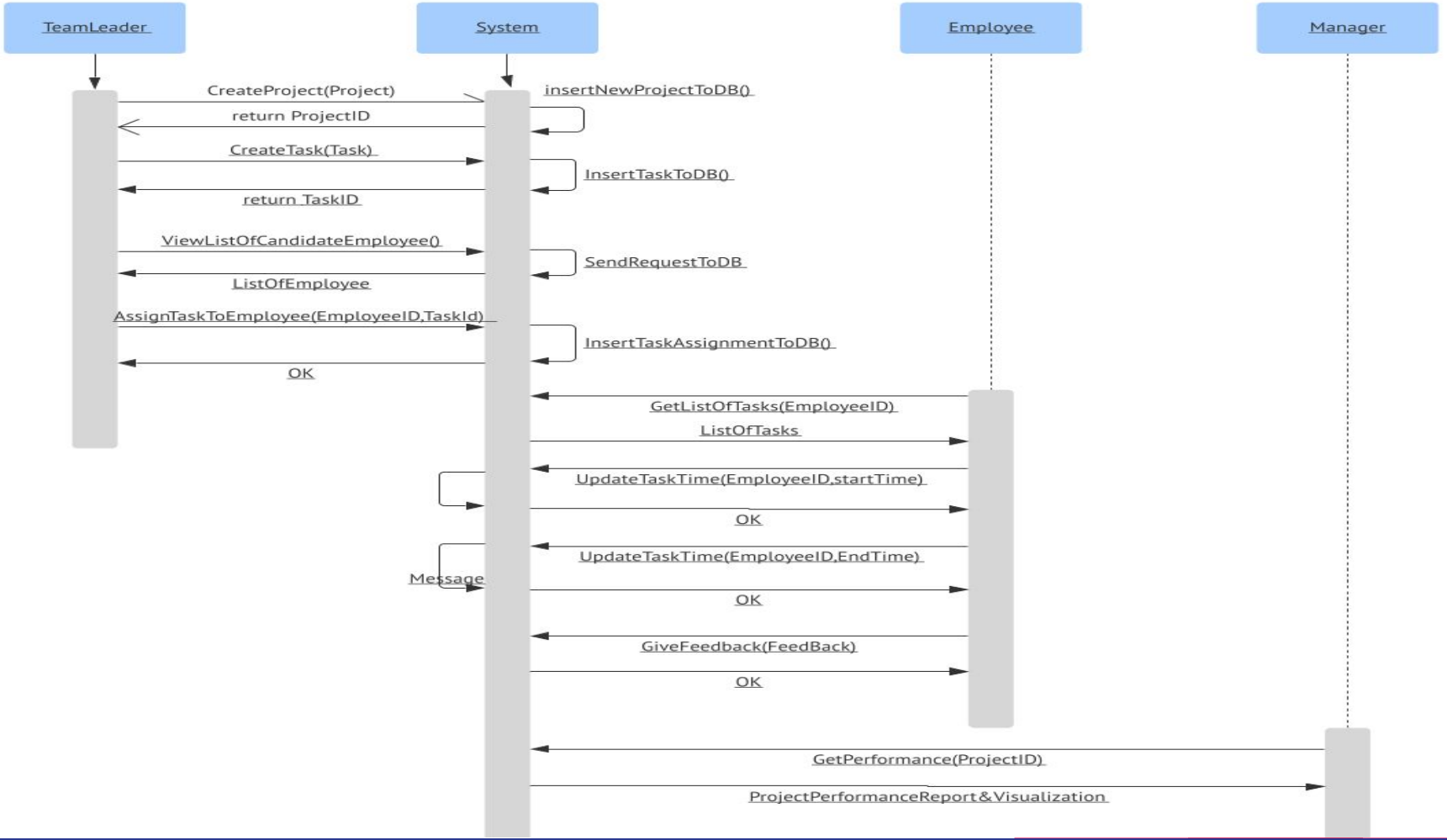
4. Design & Modeling

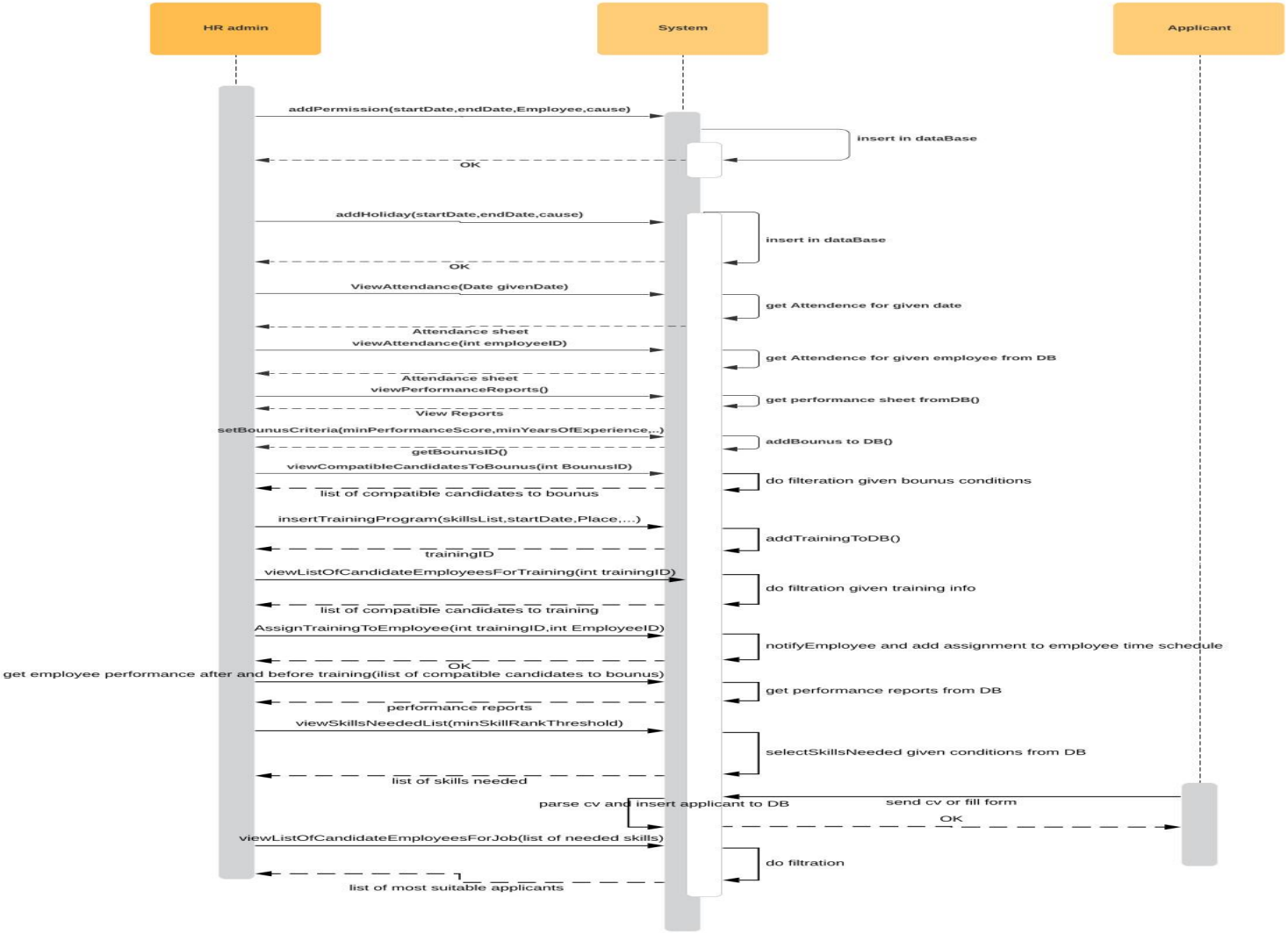
4.1 Context Modeling
4.2 Interactional Modeling
4.3 Structural Modeling
4.4 Architectural Modeling

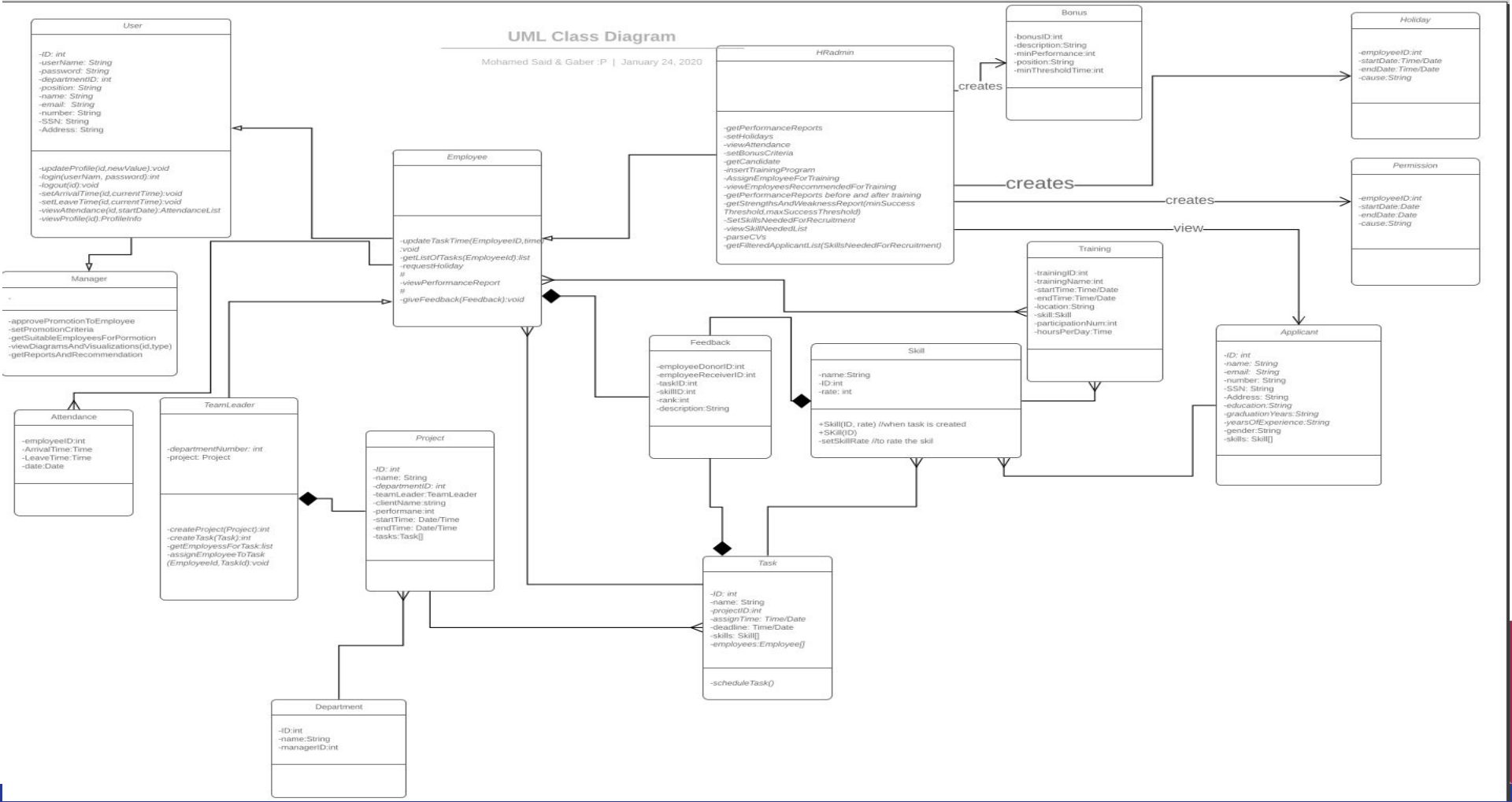
4.1 Context Modeling

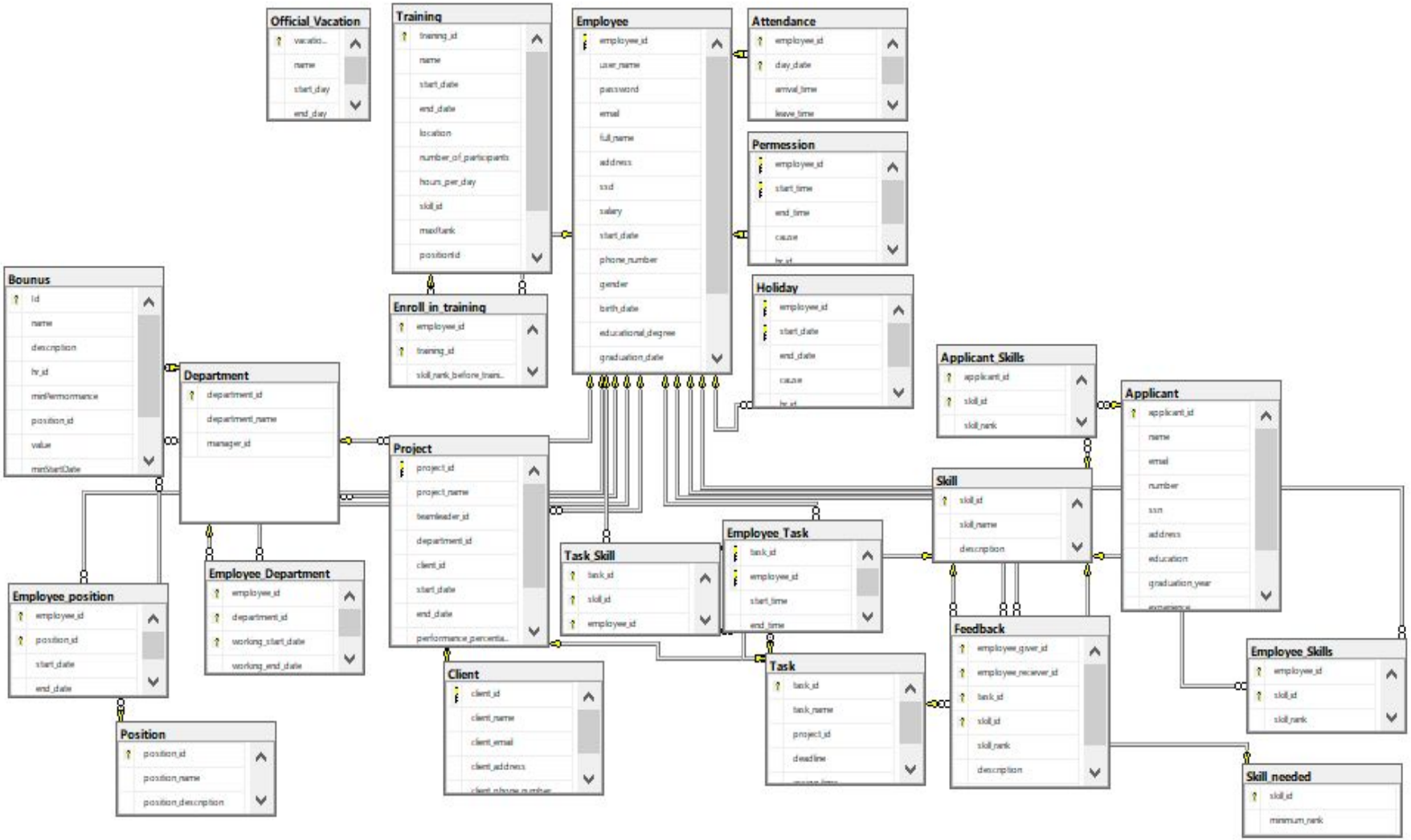


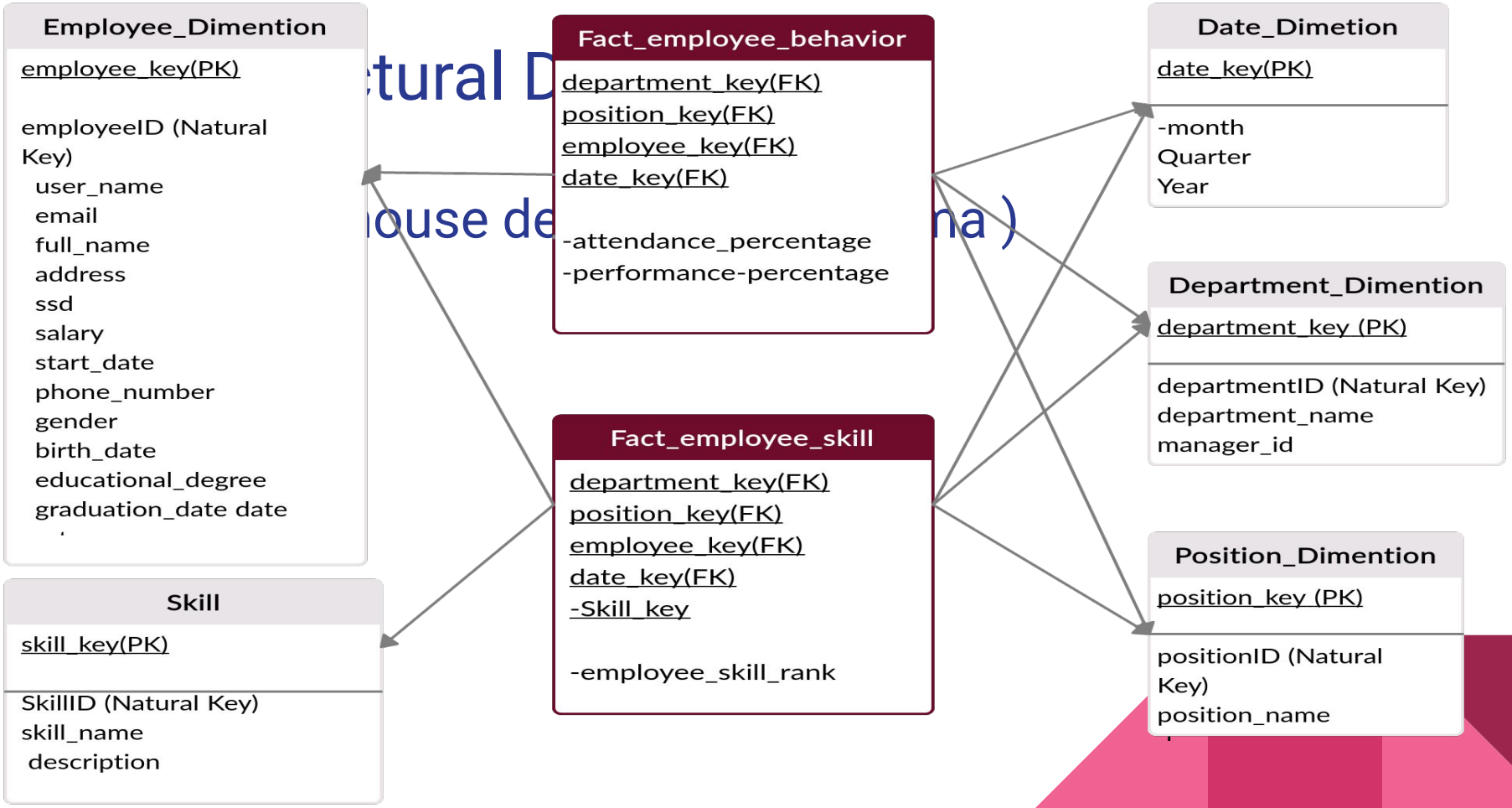
Employee	• Sign into the system and logout from the system. • View future and present scheduled tasks and trainings . • View performance and attendance progress report.
Team Leader	Create a new project and tasks within. Follow up project progress and deadlines and mark them finished . Assign tasks to employees.
HR Admin	• Set employee vacation ,permission and official holidays. • View attendance and progress reports for employees , department and positions. • Insert training program and get recommended available employees and enroll them. • Enter job requirements and get applicants recommandations • Set bonus criteria and get employees that earn the bonus then grant bonus
Manager	• Get performance progress reports for employees , projects departments and positions. • Get Strength and Weakness Fields reports. • Set promotion criteria and get recommended employees for promotion. • View attendance reports for employees , departments and positions. • Update the data warehouse to add the new months records and update performance and skill ranks .

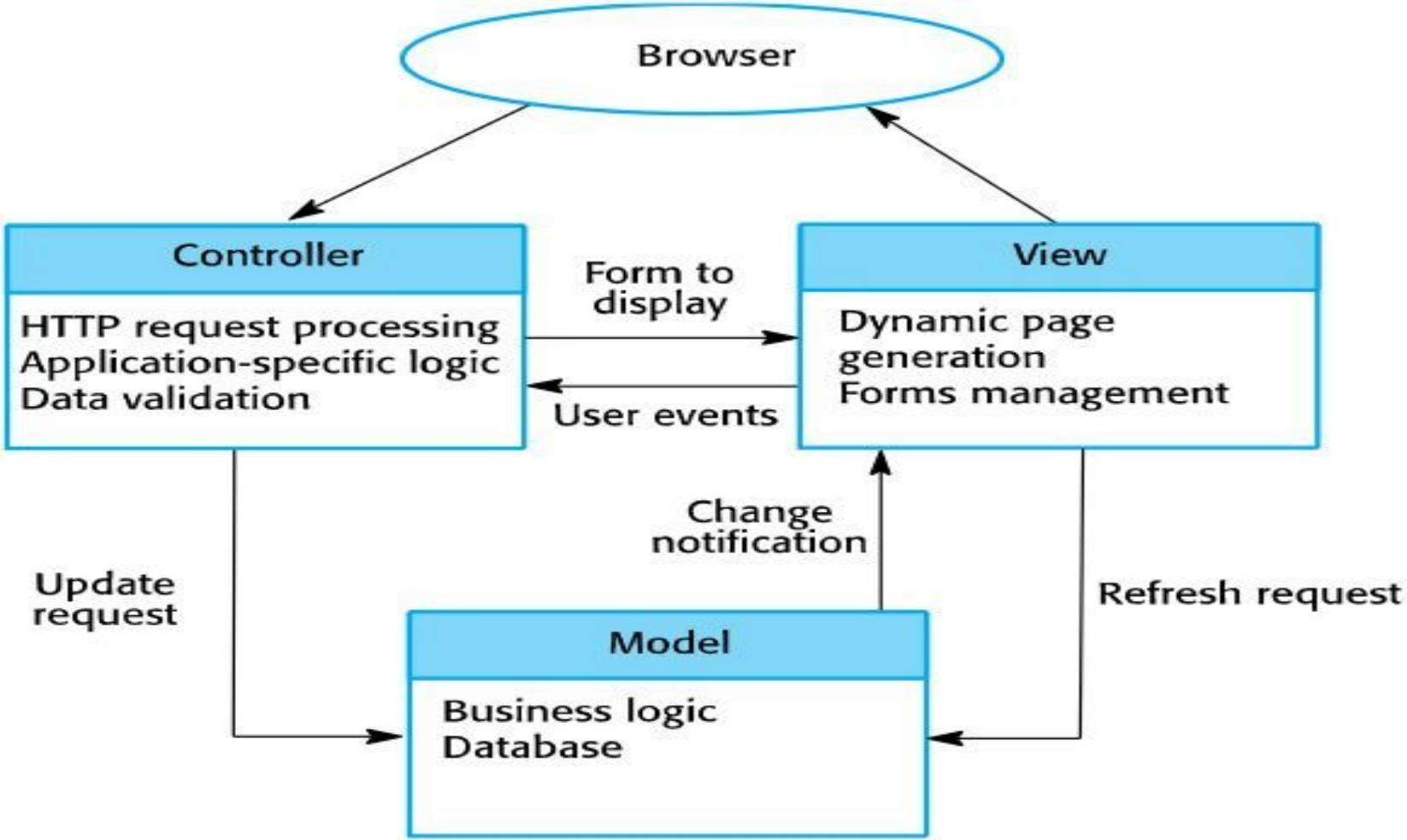

TeamLeader
System
Employee
Manager
CreateProject(Project)
insertNewProjectToDB()
return ProjectID
CreateTask(Task)
InsertTaskToDB()
return TaskID
ViewListOfCandidateEmployee()
SendRequestToDB
ListOfEmployee
AssignTaskToEmployee(EmployeeID,TaskId)
InsertTaskAssignmentToDB()
OK
GetListOfTasks(EmployeeID)
ListOfTasks
UpdateTaskTime(EmployeeID,startTime)
OK
UpdateTaskTime(EmployeeID,EndTime)
Message
OK
GiveFeedback(FeedBack)
OK
GetPerformance(ProjectID)
ProjectPerformanceReport&Visualization


HR admin
System
Applicant
addPermission(startDate,endDate,Employee,cause)
insert in dataBase
OK
addHoliday(startDate,endDate,cause)
insert in dataBase
OK
ViewAttendance(Date givenDate)
get Attendence for given date
Attendance sheet
viewAttendance(int employeeID)
get Attendence for given employee from DB
Attendance sheet
viewPerformanceReports()
get performance sheet fromDB()
View Reports
setBounusCriteria(minPerformanceScore,minYearsOfExperience,...)
addBounus to DB()
getBounusID()
viewCompatibleCandidatesToBounus(int BounusID)
do filteration given bounus conditions
list of compatible candidates to bounus
insertTrainingProgram(skillsList,startDate,Place,...)
addTrainingToDB()
trainingID
viewListOfCandidateEmployeesForTraining(int trainingID)
do filtration given training info
list of compatible candidates to training
AssignTrainingToEmployee(int trainingID,int EmployeeID)
notifyEmployee and add assignment to employee time schedule
OK
get employee performance after and before training(ilist of compatible candidates to bounus)
get performance reports from DB
performance reports
viewSkillsNeededList(minSkillRankThreshold)
selectSkillsNeeded given conditions from DB
list of skills needed
parse cv and insert applicant to DB
send cv or fill form
OK
viewListOfCandidateEmployeesForJob(list of needed skills)
do filtration
list of most suitable applicants


UML Class Diagram
Mohamed Said & Gaber :P | January 24, 2020
User
-ID: int
-userName: String
-password: String
-departmentID: int
-position: String
-name: String
-email: String
-number: String
-SSN: String
-Address: String
-updateProfile(id,newValue):void
-login(userNam, password):int
-logout(id):void
-setArrivalTime(id,currentTime):void
-setLeaveTime(id,currentTime):void
-viewAttendance(id,startDate):AttendanceList
-viewProfile(id):ProfileInfo
Manager
-approvePromotionToEmployee
-setPromotionCriteria
-getSuitableEmployeesForPormotion
-viewDiagramsAndVisualizations(id,type)
-getReportsAndRecommendation
Attendance
-employeeID:int
-ArrivalTime:Time
-LeaveTime:Time
-date:Date
TeamLeader
-departmentNumber: int
-project: Project
-createProject(Project):int
-createTask(Task):int
-getEmployeesForTask:list
-assignEmployeeToTask
(EmployeeId,TaskId):void
Project
-ID: int
-name: String
-departmentID: int
-teamLeader:TeamLeader
-clientName:string
-performane:int
-startTime: Date/Time
-endTime: Date/Time
-tasks:Task[]
Department
-ID:int
-name:String
-managerID:int
Employee
-updateTaskTime(EmployeeID,time)
:void
-getListOfTasks(EmployeeId):list
-requestHoliday
#
-viewPerformanceReport
#
-giveFeedback(Feedback):void
HRadmin
-getPerformanceReports
-setHolidays
-viewAttendance
-setBonusCriteria
-getCandidate
-insertTrainingProgram
-AssignEmployeeForTraining
-viewEmployeesRecommendedForTraining
-getPerformanceReports before and after training
-getStrengthsAndWeaknessReport(minSuccess
Threshold,maxSuccessThreshold)
-SetSkillsNeededForRecruitment
-viewSkillNeededList
-parseCVs
-getFilteredApplicantList(SkillsNeededForRecruitment)
Bonus
-bonusID:int
-description:String
-minPerformance:int
-position:String
-minThresholdTime:int
creates
Holiday
-employeeID:int
-startDate:Time/Date
-endDate:Time/Date
-cause:String
creates
Permission
-employeeID:int
-startDate:Date
-endDate:Date
-cause:String
creates
view
Training
-trainingID:int
-trainingName:int
-startTime:Time/Date
-endTime:Time/Date
-location:String
-skill:Skill
-participationNum:int
-hoursPerDay:Time
Applicant
-ID: int
-name: String
-email: String
-number: String
-SSN: String
-Address: String
-education:String
-graduationYears:String
-yearsOfExperience:String
-gender:String
-skills: Skill[]
Feedback
-employeeDonorID:int
-employeeReceiverID:int
-taskID:int
-skillID:int
-rank:int
-description:String
Skill
-name:String
-ID:int
-rate: int
+Skill(ID, rate) //when task is created
+SKill(ID)
-setSkillRate //to rate the skil
Task
-ID: int
-name: String
-projectID:int
-assignTime: Time/Date
-deadline: Time/Date
-skills: Skill[]
-employees:Employee[]
-scheduleTask()


Official_Vacation
vacatio...
name
start_day
end_day
Training
training_id
name
start_date
end_date
location
number_of_participants
hours_per_day
skill_id
maxRank
positionId
Employee
employee_id
user_name
password
email
full_name
address
ssd
salary
start_date
phone_number
gender
birth_date
educational_degree
graduation_date
Attendance
employee_id
day_date
arrival_time
leave_time
Permession
employee_id
start_time
end_time
cause
hr_id
Holiday
employee_id
start_date
end_date
cause
hr_id
Enroll_in_training
employee_id
training_id
skill_rank_before_trains...
Bounus
Id
name
description
hr_id
manPermormance
position_id
value
manStartDate
Department
department_id
department_name
manager_id
Project
project_id
project_name
teamleader_id
department_id
client_id
start_date
end_date
performance_percentia...
Applicant_Skills
applicant_id
skill_id
skill_rank
Applicant
applicant_id
name
email
number
ssn
address
education
graduation_year
Skill
skill_id
skill_name
description
Employee_Task
task_id
employee_id
start_time
end_time
Task_Skill
task_id
skill_id
employee_id
Employee_position
employee_id
position_id
start_date
end_date
Employee_Department
employee_id
department_id
working_start_date
working_end_date
Client
client_id
client_name
client_email
client_address
client_phone_number
Task
task_id
task_name
project_id
deadline
Feedback
employee_giver_id
employee_receiver_id
task_id
skill_id
skill_rank
description
Employee_Skills
employee_id
skill_id
skill_rank
Position
position_id
position_name
position_description
Skill_needed
skill_id
minimum_rank


Employee_Dimention
employee_key(PK)
employeeID (Natural Key)
user_name
email
full_name
address
ssd
salary
start_date
phone_number
gender
birth_date
educational_degree
graduation_date date
Skill
skill_key(PK)
SkillID (Natural Key)
skill_name
description
Fact_employee_behavior
department_key(FK)
position_key(FK)
employee_key(FK)
date_key(FK)
-attendance_percentage
-performance-percentage
Fact_employee_skill
department_key(FK)
position_key(FK)
employee_key(FK)
date_key(FK)
-Skill_key
-employee_skill_rank
Date_Dimetion
date_key(PK)
-month
Quarter
Year
Department_Dimention
department_key (PK)
departmentID (Natural Key)
department_name
manager_id
Position_Dimention
position_key (PK)
positionID (Natural Key)
position_name


Browser
Controller
HTTP request processing
Application-specific logic
Data validation
Form to display
User events
View
Dynamic page generation
Forms management
Change notification
Update request
Refresh request
Model
Business logic
Database

5. Implementation

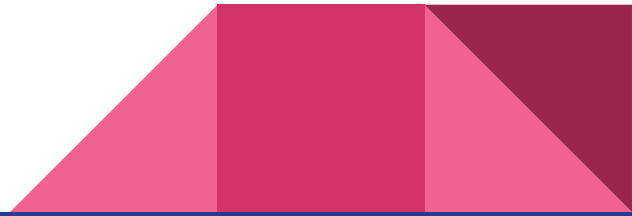
Agility for work and time management

We made use of the agile environment practice in managing the work which made it easier to work in parallel (front end , back end and database) as all phases (requirements identification , design and implementation) were done in iterations and incrementally which made the system embrace the constantly changing requirements and design decisions and also made it maintainable and ready for new additions as additional increments with the minimum modifications.

XP Practices for coding

We used some of the XP (extreme programming) practices such as :

Pair programming
Repeated refactoring



Iteration 1 (2 weeks)

Build the infrastructure

Database tables and relations built

The classes of the model part are generated and methods ready to be implemented

Iteration 2(1 Week)

Implementing Employee self service

3 web pages :

Login
Hire employee
View employee personal information

Iteration 3(2 Weeks)

Implementing Need for Training

3 web pages.
Add training
Training board
View Training Information

Iteration 4 (2 weeks)

Build the Data warehouse tables and data flow

- ETL (extract , transform, load)process done from the database to data warehouse
- Data flow for each dimension table is implemented
- Data flow for each fact table is implemented and queries for measure wait to be determined.

Iteration 5 (2 Weeks)

Implementing Employee Attendance

4 web pages :

Employee attendance
Attendance Reporting progress
Attendance reporting comparison
Insert permissions and vacations

Iteration 6(3 Weeks)

Implementing Time tracking and task scheduling alongside with talent management

5 web pages.

View Projects
View Tasks by team leader
New Task
View Task by team leader
Task Scheduling for employees

Iteration 7(1 week)

Implementing Give Feedback

2 pages:

Feedback

Give Feedback

Iteration 8 (2 Weeks)

Implementing Employee performance & bonus and promotion

4 pages :

Performance Reporting progress

Performance reporting comparison

Insert bonus

Insert Promotion

Iteration 9(2 Weeks)

Implementing Project and overall company performance & weakness and strength fields

3 web pages.

Performance reporting comparison

Overall Company Performance

Weakness and strength fields

Iteration 10(1 week)

Implementing recruitment

2 pages:

Applicant form

HR recruitment

6. Tools Used

Visual studio 2019

Support web application development for MVC architecture and database connections and manipulations and SSIS for data warehouse analysis in an efficient easy way which matches our project scope

Microsoft SQL Server Management 2019

Database Management System that manages database implementation ,query manipulation and transaction and scripts execution.

Compatible with the visual studio setting for more consistent implementation

SSIS

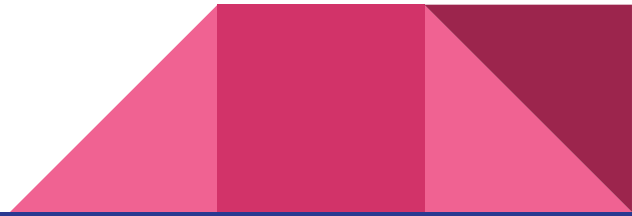
SQL Server Integration tool that supports data flows and implementing the data warehouse star schema design .

Also compatible with using SQL server DBMS and visual studio.

7. Testing

Testing strategy

1) Unit and component testing were used during the implementation phase (test driven technique using pre written test cases)
2) System testing was performed in 2 directions :
    - Requirement based testing
        - Functional requirements validation
        - Non functional requirements validation



Usability	the system's interfaces are easy to use by any average employee with training period that won't exceed 2 weeks due to descriptive messages , warnings and interfaces.
Performance	the system has fast response to analysis and reporting and listing commands due to the efficient implementation of the DW and database transactions (we only tested the system on local host so the performance might differ when deployed on a host server due to network and other technical complications)
Maintainability	the system is very highly maintainable as we lied the bases in the architecture , design , database procedures and backend implementation for very wide range of additions and modifications that can easily be added even when new functionalities are going to be introduced , the system would show great flexibility , readability and simplicity through requirement changes as we made sure to site a detailed system modeling and implementation documentation.
Security	security is partially achieved through password log in mechanism but there would be threats as authorization and token authentication mechanisms weren't applied as they were out of our interest scope that included DW design and implementation in real life application.
Reliability	the system wouldn't fail in the normal conditions and operations but in case of unpredicted input , the system behavior would be unpredictable and there's risk of system failure as input validation and testing wasn't thoroughly examined.

8. Conclusion and Future Work

8.1 Conclusion

The system is a very clear illustration to the implementation of a DW based application that can still be extended and lots of functionalities are realisable given the architectural design and the infrastructure (DB ,DW and model) implementation

8.2 Future Work

1- Some extra functional requirements can be added (training follow up , further data analysis and predictions ,...etc).
2- Security can be increased through token authentication.
3- More excessive testing can take place to make the system.
4- More reliable and increases availability and failure mutiny.
5- Input validation for non predictable inputs and behavior can be done.
6- predictions using data mining and deep learning models can easily be introduced based on statistics and analytics provided by the system would be a great asset to add value to the current system
7-Using SSAS tools for data warehousing analytics(perform analysis and generate automated reports)

9-Demo

Thank You