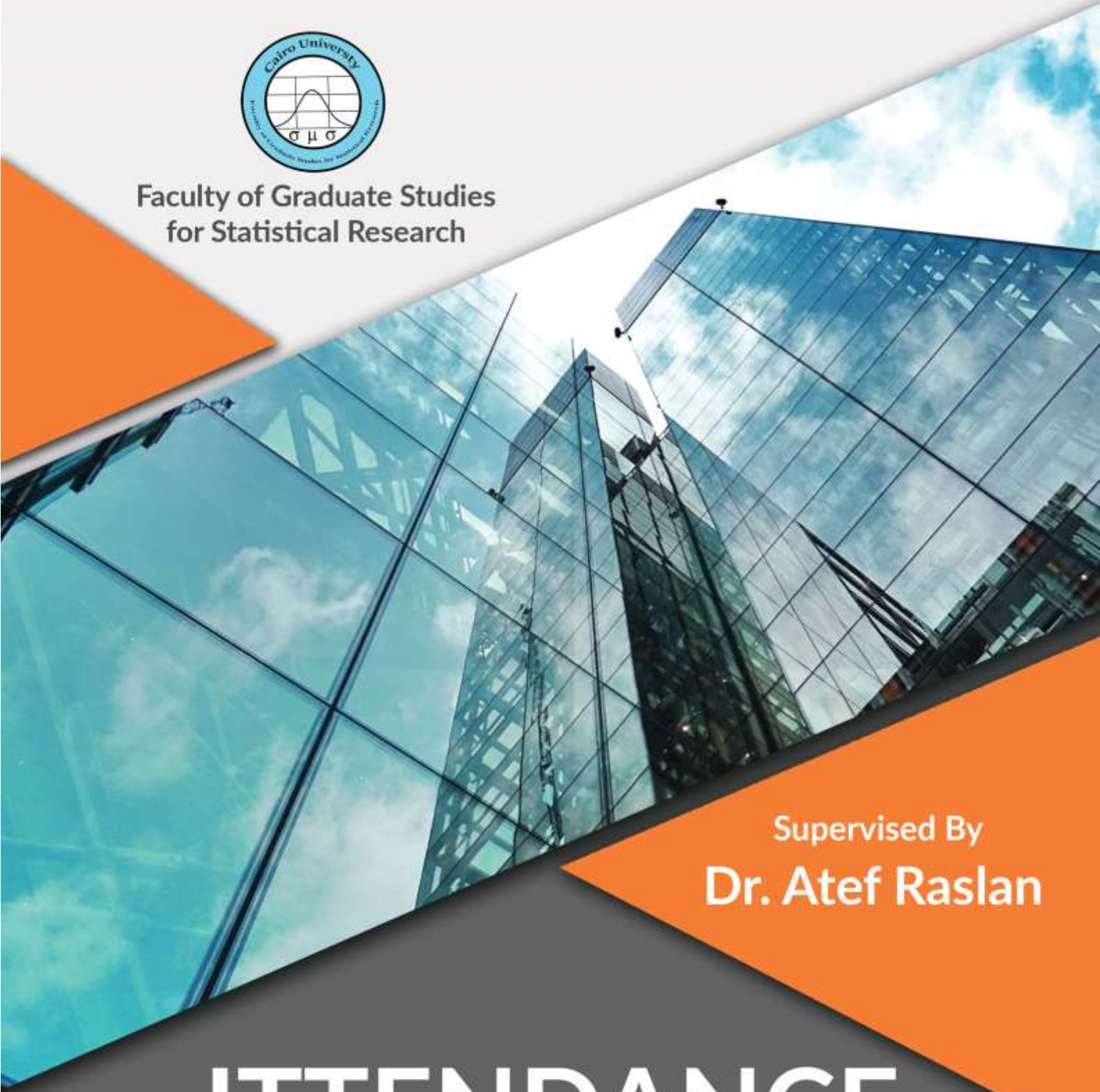


Faculty of Graduate Studies
for Statistical Research

Supervised By

Dr. Atef Raslan

ITTENDANCE

Attendance Management System

Submitted By

Abdullah Ezzat
Esraa Abdelmaksoud
Mostafa Zaher

Index

Abstract

The world has been witnessing a great break-through in uses of technology. This led to changes in how daily processes and tasks are performed in all fields of life. For businesses, one of the greatest advantages is replacing paper work with computerized programs to get over human errors and manipulation. Besides, businesses usually aim for reducing labor costs, and this can be achieved by having a software that perform the same task as humans but with higher quality.

Introduction

ITTENDANCE is a flexible, customized, and easy-to-use cloud-based system that is developed to help with handling the majority of tasks related to Human Resources department. This includes, but not limited to, attendance and exit timings, permits, absences, and permissions.

The main purpose for creating ITTENDANCE is to replace all paper work related to attendance of employees and decline the time spent to handle HR related tasks. The name ITTENDANCE was chosen to represent an abbreviation of intelligence and attendance. Therefore, it is an intelligent system that is used for and by the employees in an organization.

SCRUM, an agile methodology, has been used for developing this project. Embracing change and flexibility are the main reasons for selecting it. These as well as having a distributed team of only three members made SCRUM the best choice for ITTENDANCE.

Project Background

During the last few months, many organizations faced problems with saving the attendance timings of employees due to Coronavirus pandemic. It is a respiratory disease, so employees were not supposed to leave their fingerprint or signature for their safety. In addition to this, many requests like absence, early exits and permits were preferred to be submitted online to decline face-to-face interactions. These along with other requirements were the reason why ITTENDANCE system is coming to life.

Problem Statement

Many companies face the following problems with their human resources department:

- Paper documents occupy large spaces.
- Reporting is difficult when paper documents are used.
- Immediate reporting is not possible without an electronic system.
- Calculation of previously used balance of several types of vacations for all employees require more employees and takes a long time.
- Sharing desktop tools and touching attendance register book may spread diseases.
- Non-computerized work is vulnerable to manipulation and human errors.
- Documents cannot be accessed remotely or at the same time by more than one employee.
- Signature can be falsified in attendance sheets.
- Employees data is not secured.
- Documents are subject to loss.

Objective

- Designing an easy-to-use system as an electronic attendance register (eAR).
- Decreasing face-to-face interactions with HR employees.
- Having a web-based system to be mainly used on desktop devices.
- Immediate tracking of employees who leave early without permission.
- Tracking of absent and attending employees.
- Tracking of number of permits and delayed employees.
- Avoiding any touches of desk tools to register attendance.
- Providing reports on demand immediately.
- Declining the need for HR employees.
- Reducing time taken by HR employees used for performing daily tasks.
- Preserving attendance data for all employees except those who work on time-shift basis.
- Maintaining security of employees' attendance and insurance information.

Stakeholders

Business Rules

There is a set of business rules that control this project; they are as follows:

- A large number of employees can be added.
- An employee can be moved to another department.
- The job type of an employee can be changed. This is like changing a part-time employee to a full-time employee.
- An “Employee Code” is chosen manually by human resources department.
- Personal information of users can be changed.
- Insurance information can be changed.
- An employee can be moved from a region to another one.
- The job level of an employee can be changed.
- The dependents of each employee can be added to his information.
- For each employee, there is a balance for each type of vacations.
- For each department, there are check in and check out times.
- For each department, there is minimum time for overtime.
- For each department, there is delay exception time.
- Vacation requests should be approved or rejected.

Limitations

There are a few limitations to the system, and they are as follows:

- There are five types of users that are basic, employee, supervisor, HR, and admin.
- Employees can access their dashboard data and their vacation requests only.
- Supervisors can access their dashboard data, vacation requests and attendance events including attendance of employees of their department at other departments.
- HR employees can access the dashboard, vacation requests, attendance events, settings, and employees' data of all departments.
- Admins can access everything on the system.

Requirements

1. A large number of employees can be registered.
2. The system must cloud-based.
3. The system will be upgraded later, thus the backend must be separate from the frontend.
4. The system must be in English language.
5. Each employee gets an employee code once hired to be used in registration.
6. Users must log into the system to see content and use it.
7. Employees with foreign nationalities can be registered.
8. Dependents of employees can be added.
9. Insurance details of employees and their dependents can be added.
10. A QR code scanner to be used for signature.
11. A barcode scanner can be integrated to be used for signature too.
12. Integration of face recognition can be an advantage.
13. Vacation requests must be submitted through the system.
14. Requesting permits must be done through the system.
15. Calculation of available vacations balance is needed for HR decision makers.
16. Approval of permits must be done through the system.
17. Approval of vacation requests must be done through the system.
18. A dashboard that shows the number of attending employees, absent employees, delays, permits, and early exists is needed.
19. Passwords must be hashed for security purposes.

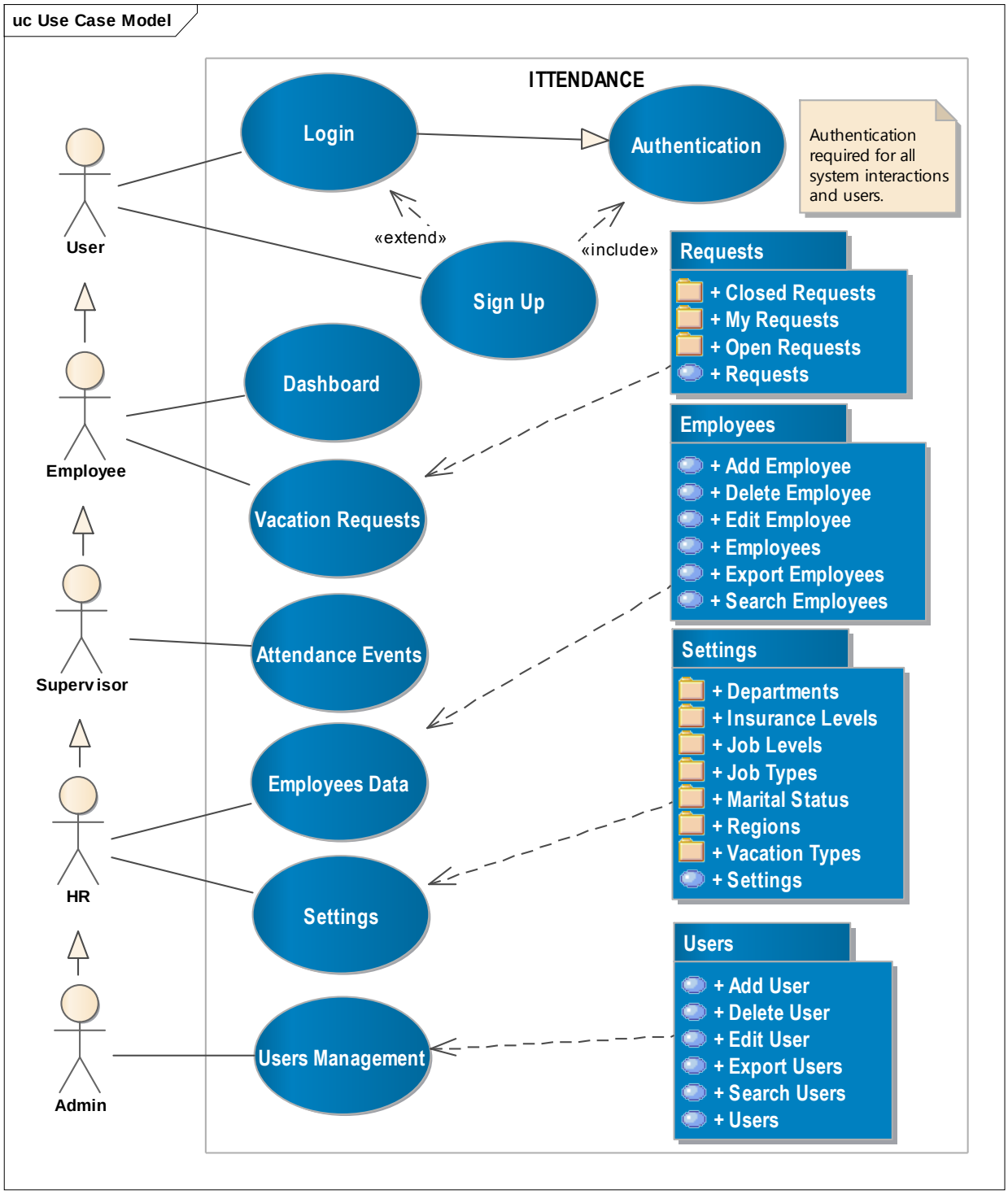
20. Enabling export of reports in PDF format is an advantage.
21. Exporting reports in Excel format is required.
22. Exporting reports related to employees' data, users' data, and vacation requests, settings is needed.
23. Supporting Arabic language is an advantage.
24. Adding, updating, and deleting users' data is required.
25. Adding, updating, and deleting employees' data is required.
26. Adding, updating, and deleting vacation requests is required.
27. Adding, updating, and deleting departments is required.
28. Adding, updating, and deleting insurance data is required.
29. Adding, updating, and deleting job types is required.
30. Adding, updating, and deleting job levels of employees is required.
31. Adding, updating, and deleting vacation types is required.
32. Adding, updating, and deleting regions is required.
33. Adding, updating, and deleting marital statuses might be needed.
34. Users must be able to change their passwords.
35. The system must not allow having a password that is typical to the username or less than eight characters in length.
36. For all employees, there should be "Regular", "Sick leave", "Religious observance", "Parental leave", "Paid leave", and more can be added or edited.
37. For each department, there should be a check-in time, checkout time, delay time limit, and minimum limit for overtime.

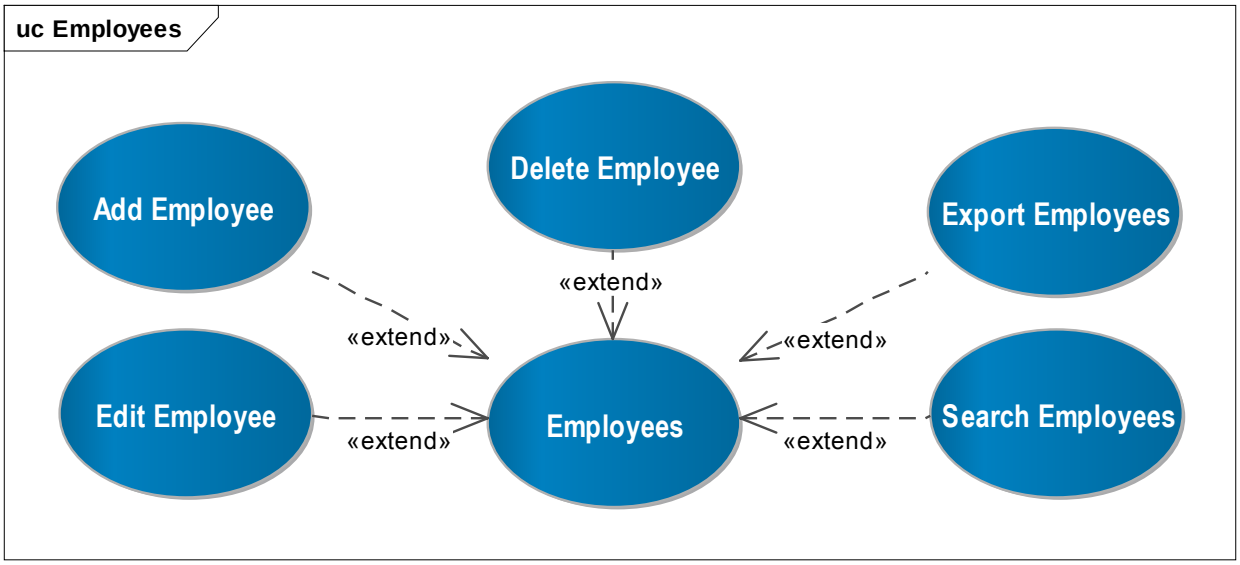
Required Reports

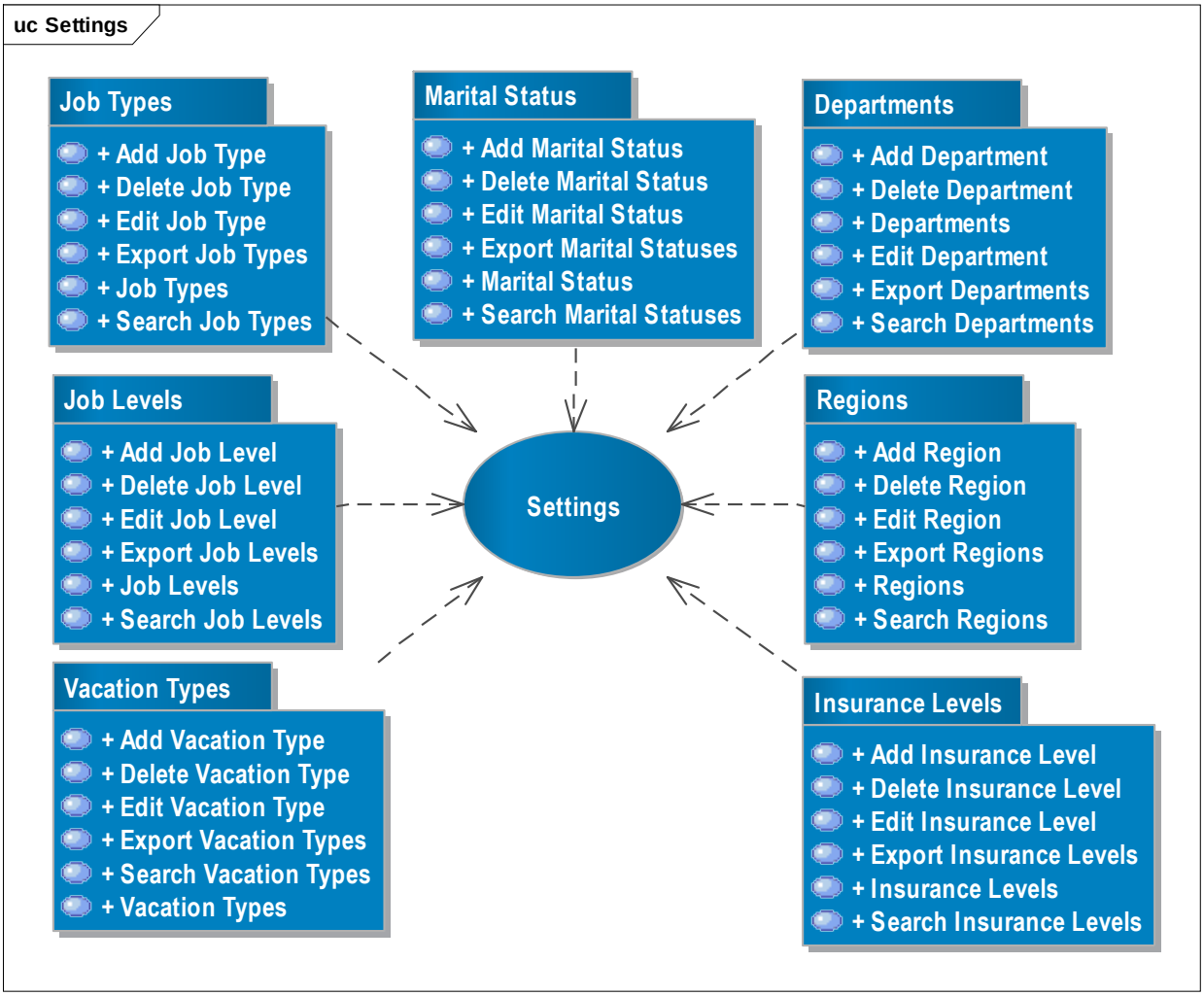
The following reports are needed to be exported in Excel format and optionally in PDF:

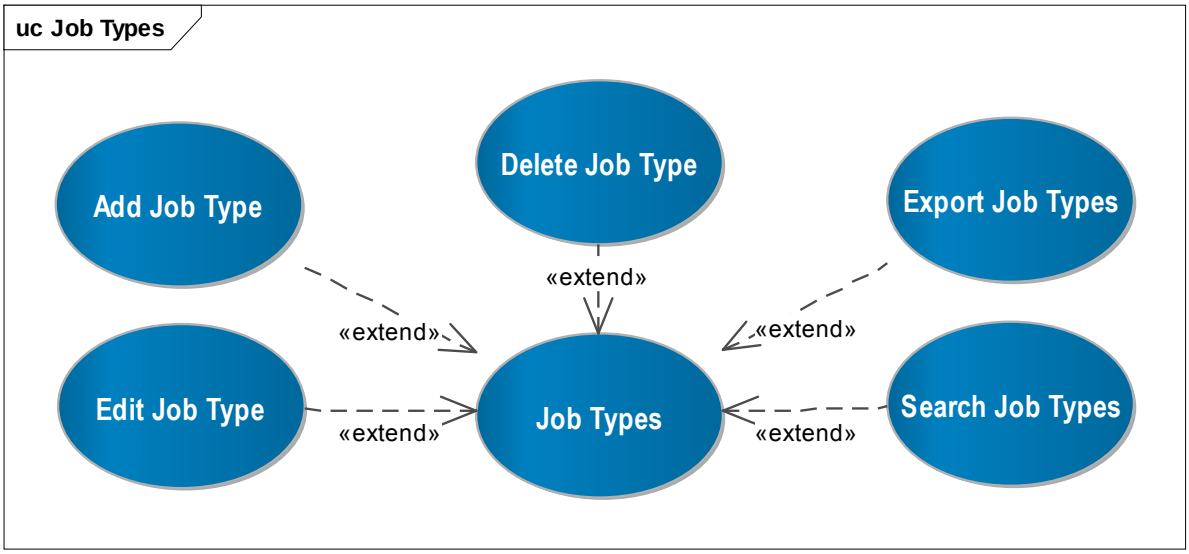
- Open requests report
- Closed requests report
- My requests report
- Employees Data report
- Users Data report

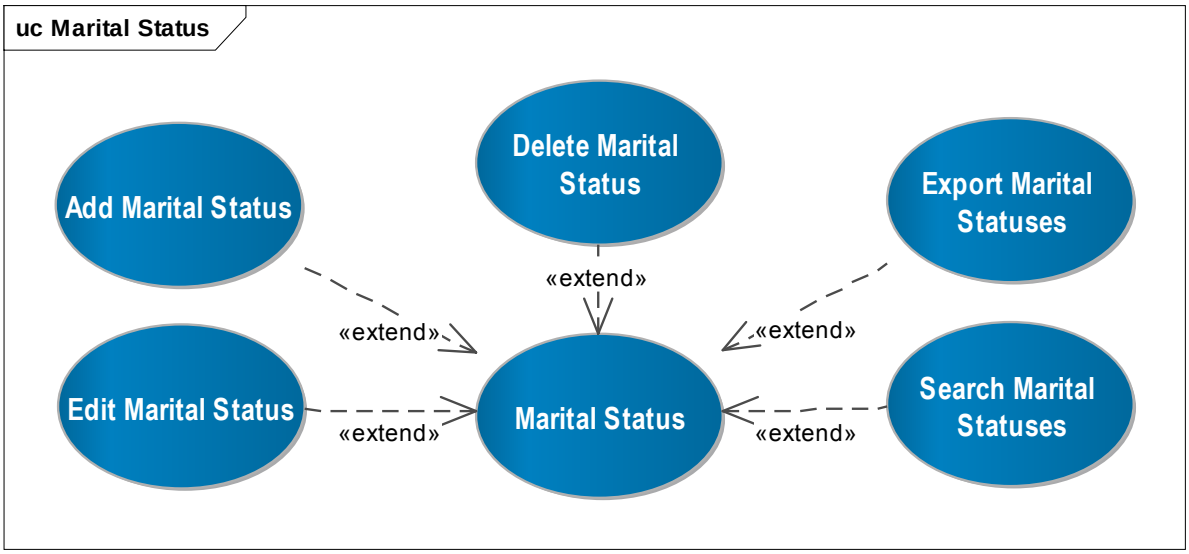
Use Case Diagram

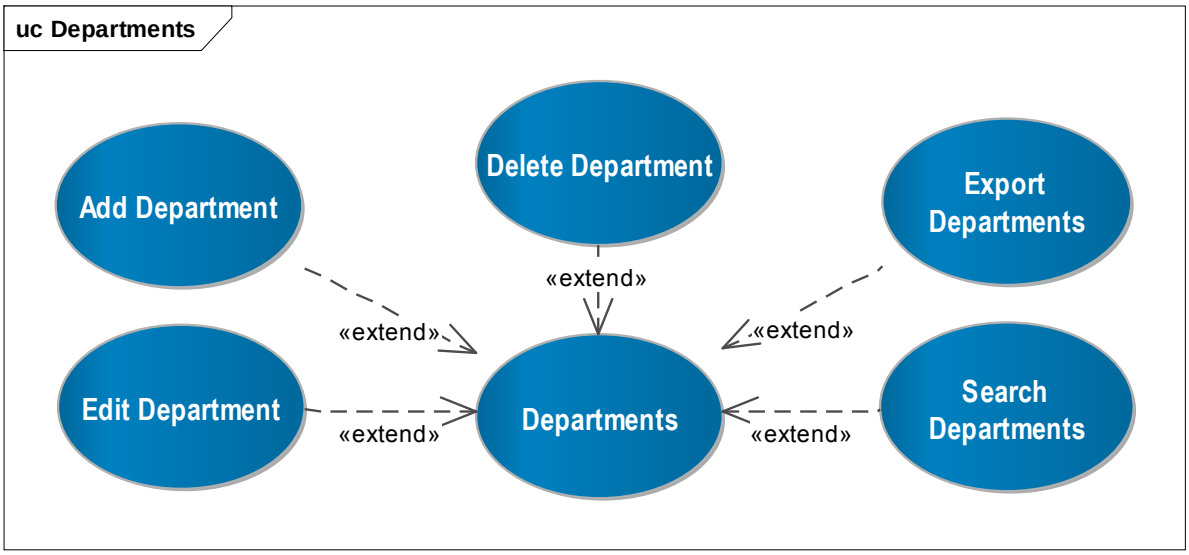


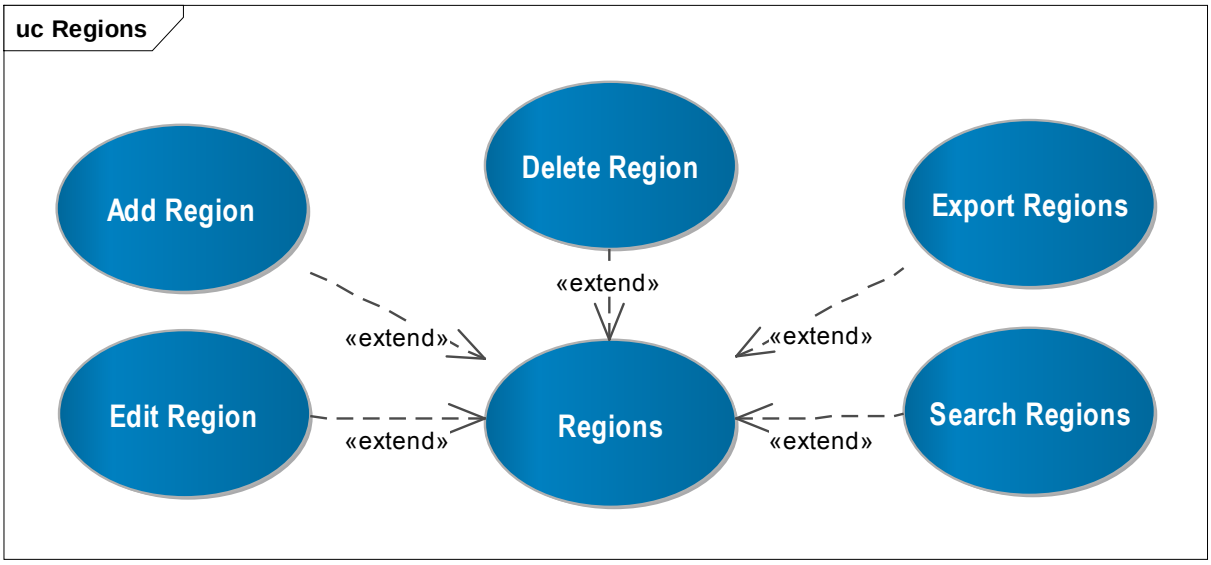

uc Employees
Add Employee
Delete Employee
Export Employees
«extend»
«extend»
«extend»
Edit Employee
Employees
Search Employees
«extend»
«extend»

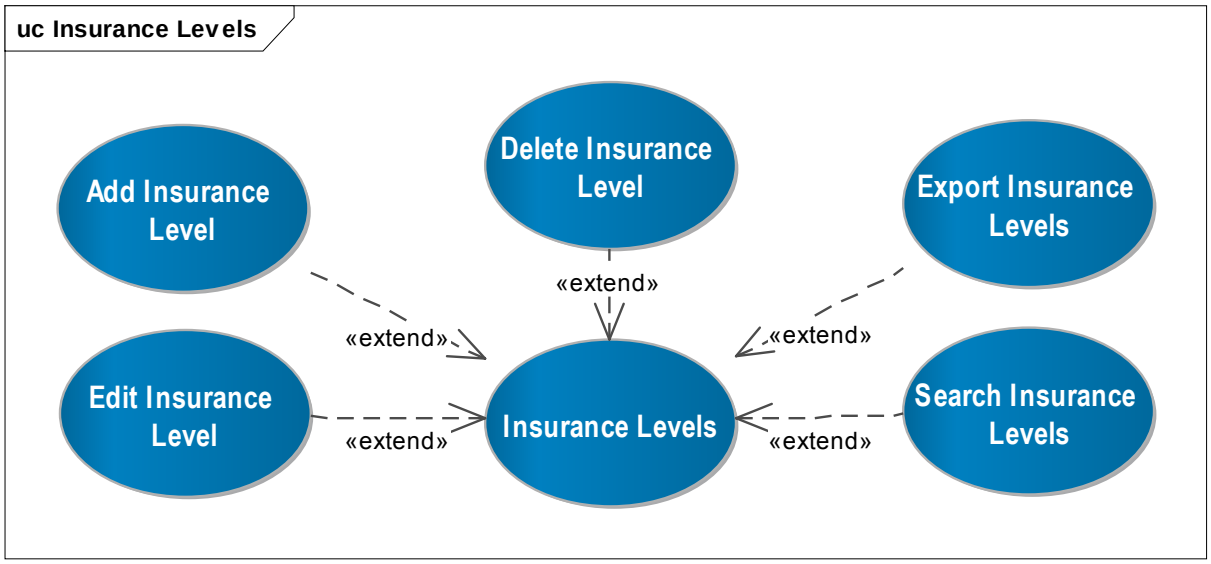

uc Settings
Job Types
+ Add Job Type
+ Delete Job Type
+ Edit Job Type
+ Export Job Types
+ Job Types
+ Search Job Types
Marital Status
+ Add Marital Status
+ Delete Marital Status
+ Edit Marital Status
+ Export Marital Statuses
+ Marital Status
+ Search Marital Statuses
Departments
+ Add Department
+ Delete Department
+ Departments
+ Edit Department
+ Export Departments
+ Search Departments
Job Levels
+ Add Job Level
+ Delete Job Level
+ Edit Job Level
+ Export Job Levels
+ Job Levels
+ Search Job Levels
Settings
Regions
+ Add Region
+ Delete Region
+ Edit Region
+ Export Regions
+ Regions
+ Search Regions
Vacation Types
+ Add Vacation Type
+ Delete Vacation Type
+ Edit Vacation Type
+ Export Vacation Types
+ Search Vacation Types
+ Vacation Types
Insurance Levels
+ Add Insurance Level
+ Delete Insurance Level
+ Edit Insurance Level
+ Export Insurance Levels
+ Insurance Levels
+ Search Insurance Levels

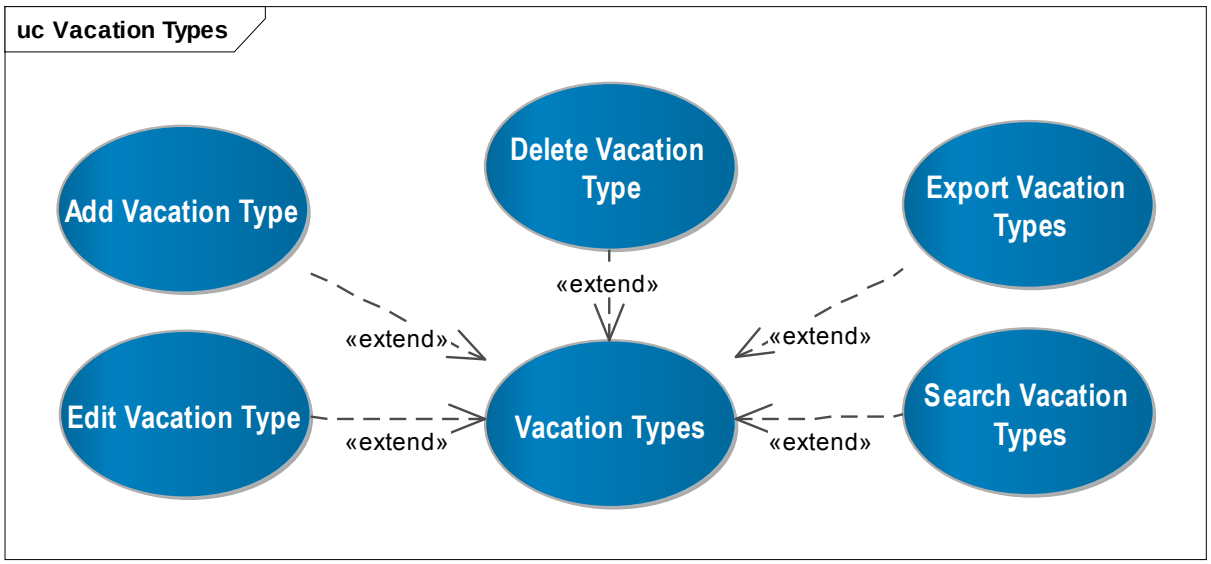

uc Job Types
Add Job Type
Delete Job Type
Export Job Types
«extend»
«extend»
«extend»
Edit Job Type
Job Types
Search Job Types
«extend»
«extend»

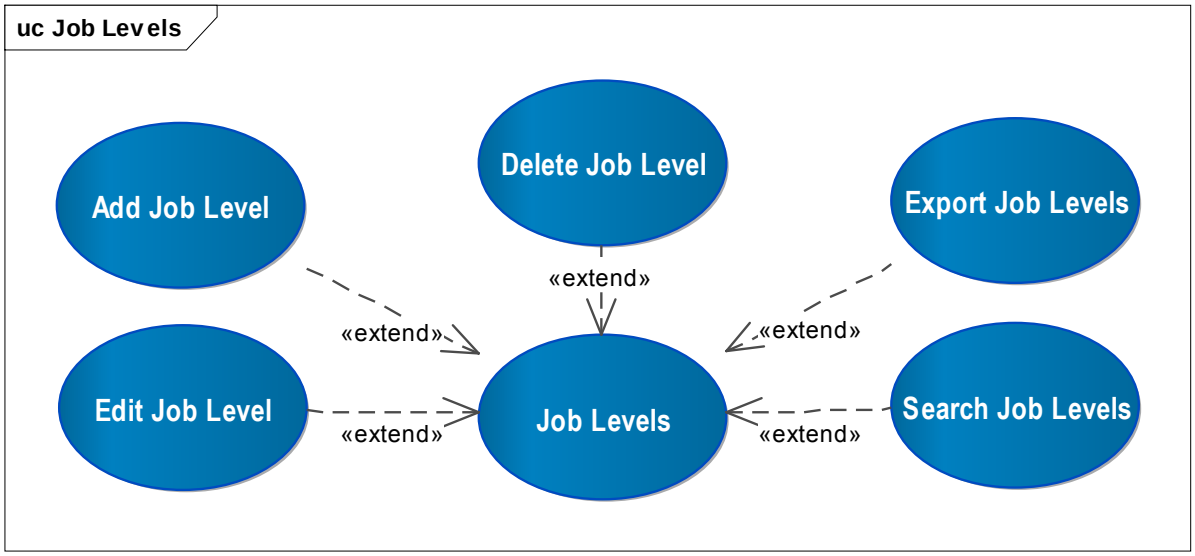

uc Marital Status
Add Marital Status
Delete Marital Status
Export Marital Statuses
«extend»
«extend»
«extend»
Edit Marital Status
Marital Status
Search Marital Statuses
«extend»
«extend»

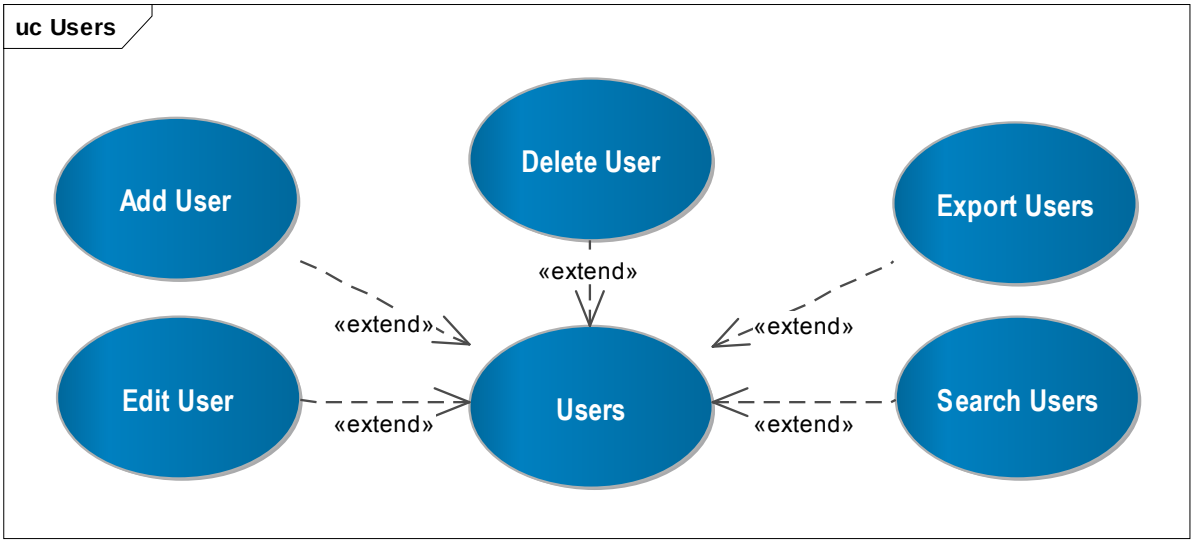

uc Departments
Add Department
Delete Department
Export Departments
«extend»
«extend»
«extend»
Edit Department
Departments
Search Departments
«extend»
«extend»

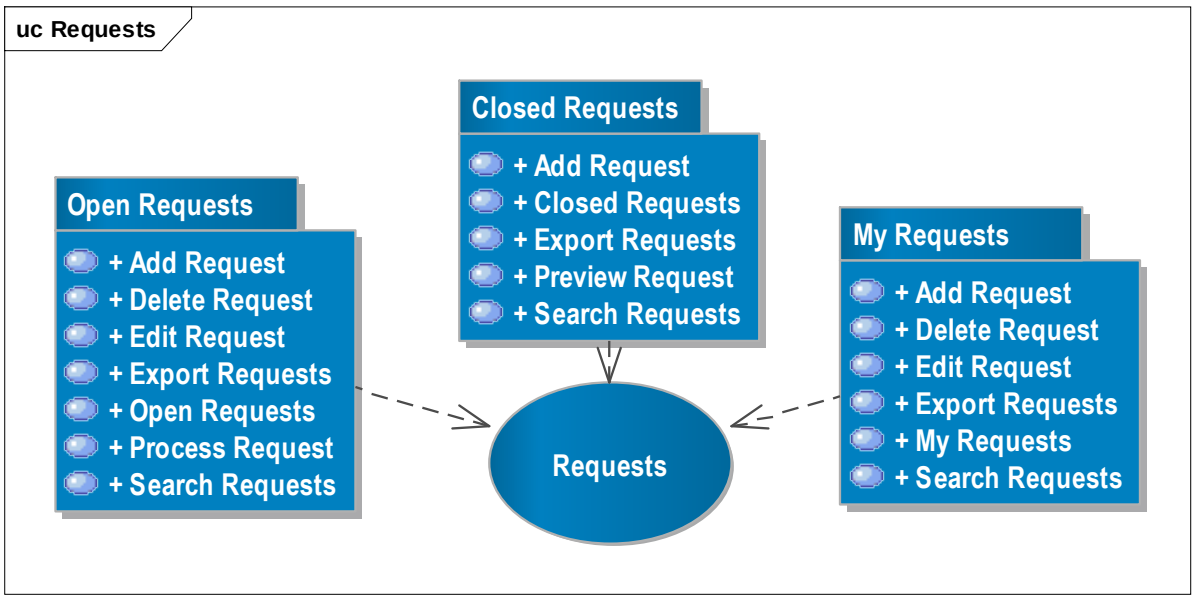

uc Regions
Add Region
Delete Region
Export Regions
«extend»
«extend»
«extend»
Edit Region
Regions
Search Regions
«extend»
«extend»

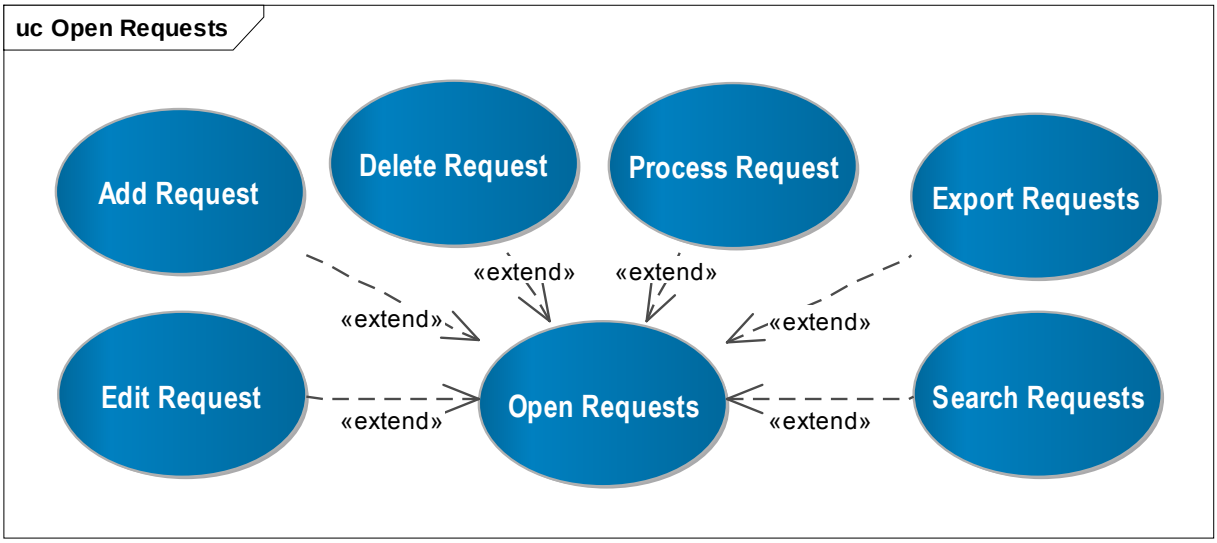

uc Insurance Levels
Add Insurance Level
Delete Insurance Level
Export Insurance Levels
«extend»
«extend»
«extend»
Edit Insurance Level
Insurance Levels
Search Insurance Levels
«extend»
«extend»

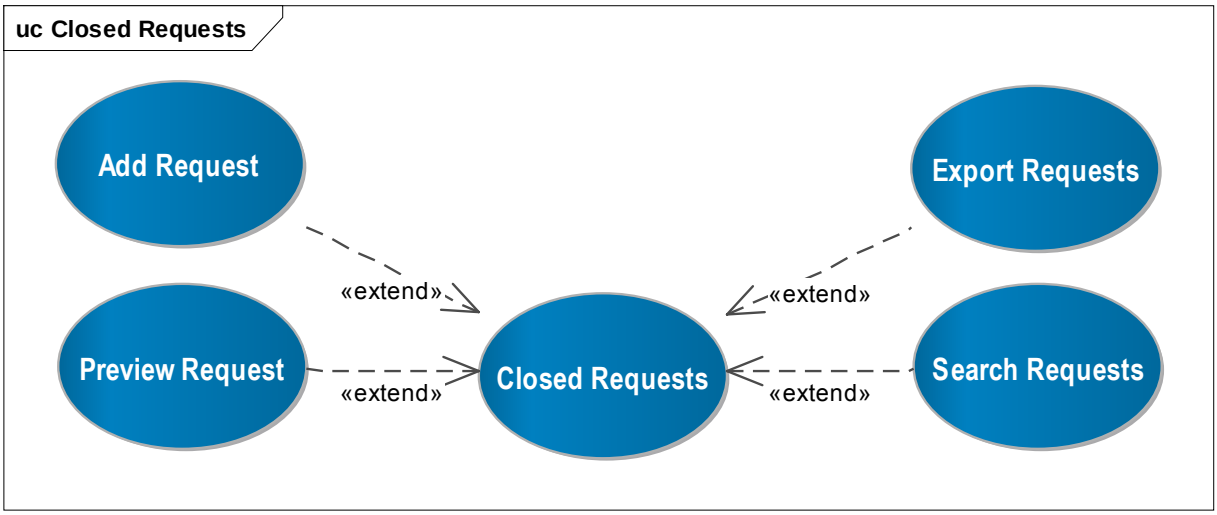

uc Vacation Types
Add Vacation Type
Delete Vacation Type
Export Vacation Types
«extend»
«extend»
«extend»
Edit Vacation Type
Vacation Types
Search Vacation Types
«extend»
«extend»

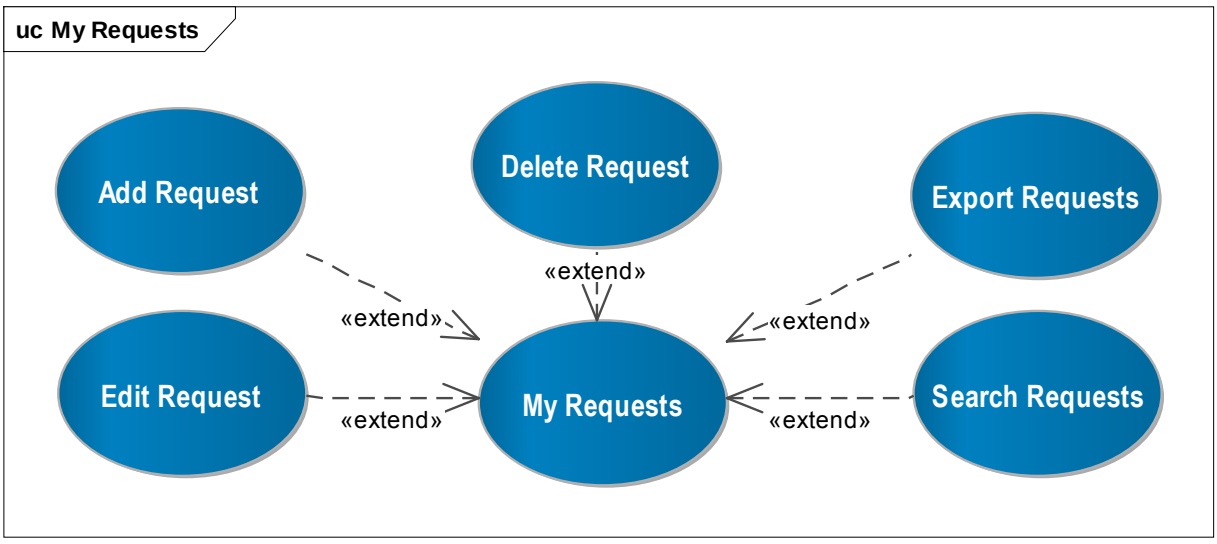

uc Job Levels
Add Job Level
Delete Job Level
Export Job Levels
«extend»
«extend»
«extend»
Edit Job Level
Job Levels
Search Job Levels
«extend»
«extend»


uc Users
Add User
Delete User
Export Users
«extend»
«extend»
«extend»
Edit User
Users
Search Users
«extend»
«extend»

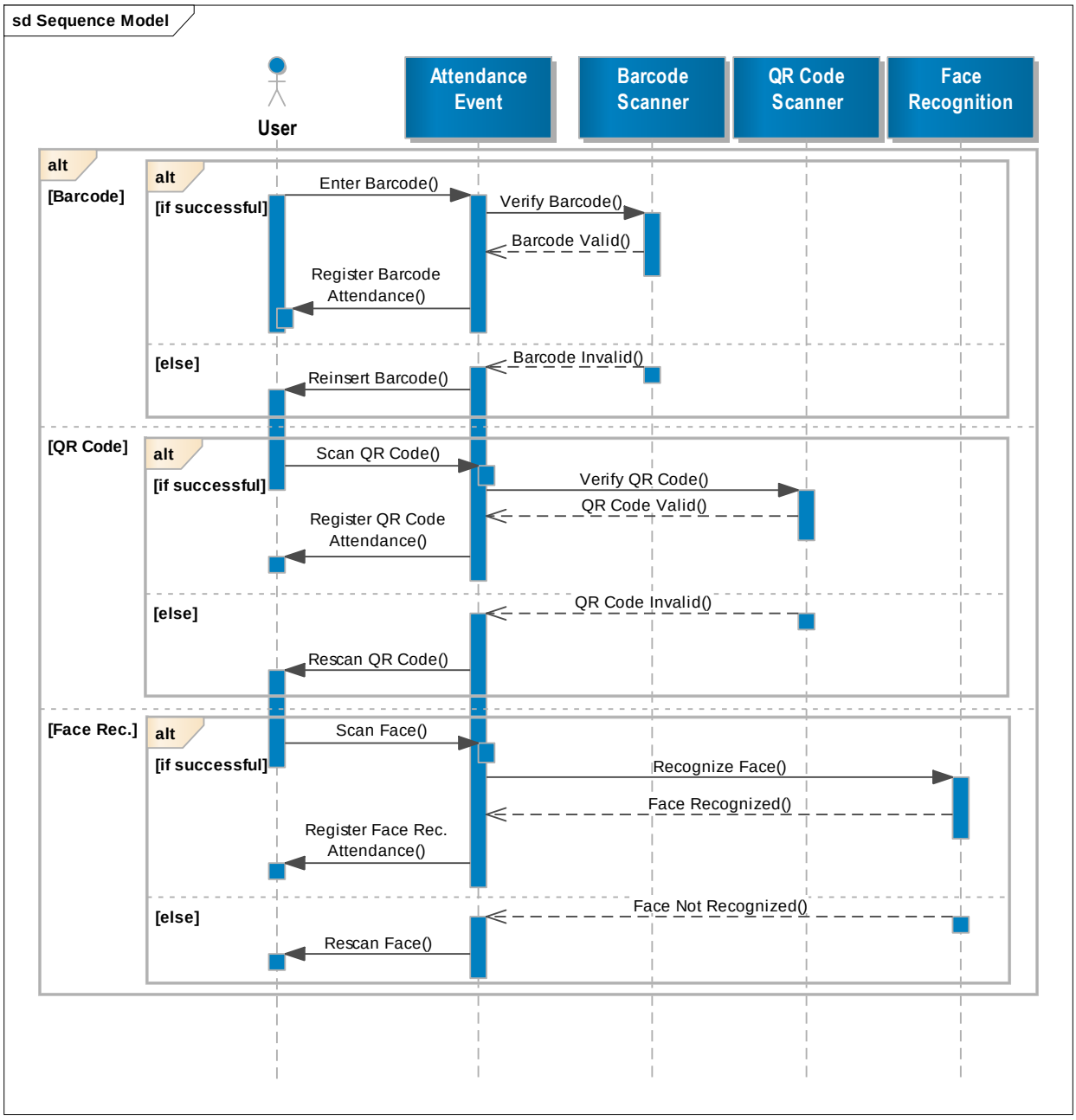

uc Requests
Closed Requests
+ Add Request
+ Closed Requests
+ Export Requests
+ Preview Request
+ Search Requests
Open Requests
+ Add Request
+ Delete Request
+ Edit Request
+ Export Requests
+ Open Requests
+ Process Request
+ Search Requests
My Requests
+ Add Request
+ Delete Request
+ Edit Request
+ Export Requests
+ My Requests
+ Search Requests
Requests

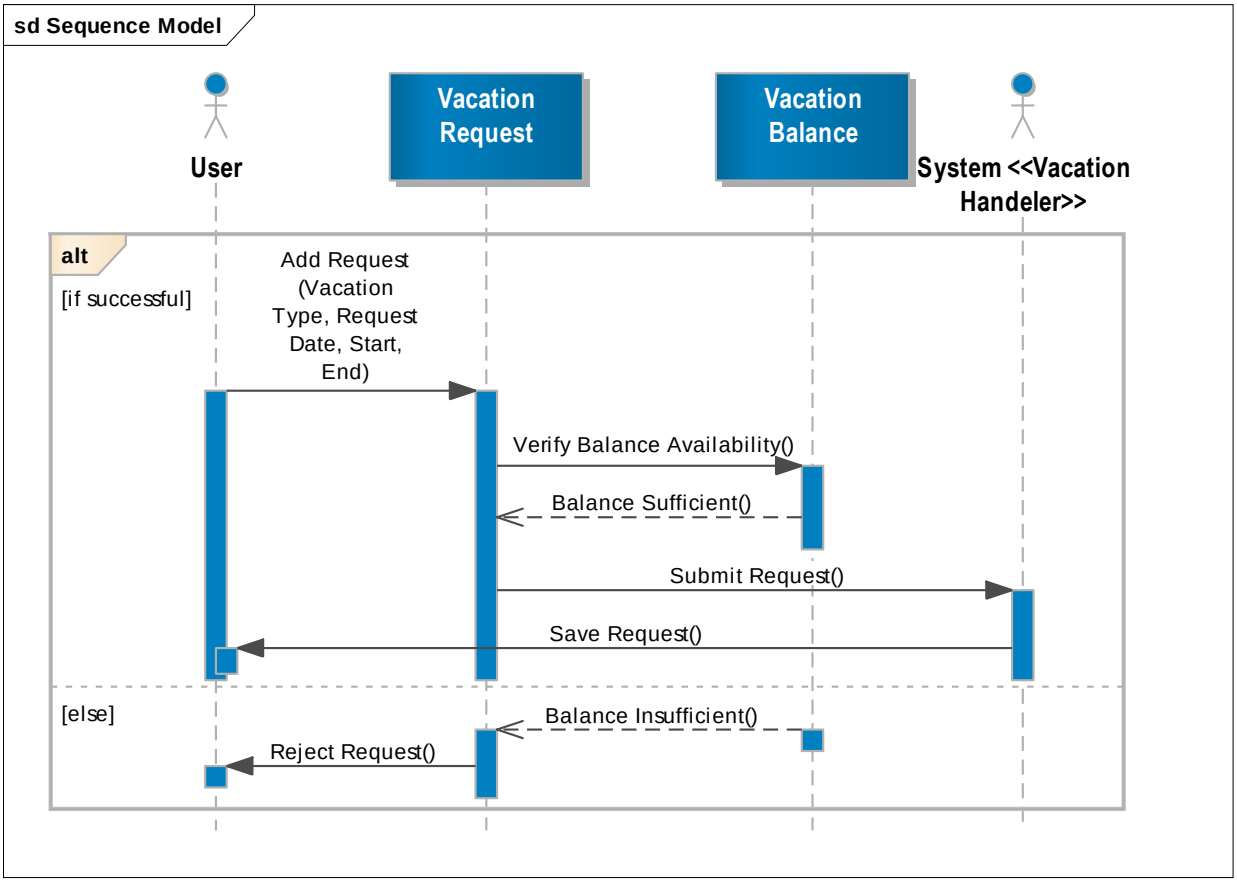

uc Open Requests
Add Request
Delete Request
Process Request
Export Requests
«extend»
«extend»
«extend»
«extend»
Edit Request
Open Requests
Search Requests
«extend»
«extend»


uc Closed Requests
Add Request
Export Requests
«extend»
«extend»
Preview Request
Closed Requests
Search Requests
«extend»
«extend»


uc My Requests
Add Request
Delete Request
Export Requests
«extend»
«extend»
«extend»
Edit Request
My Requests
Search Requests
«extend»
«extend»

Sequence Diagram




sd Sequence Model
User
Vacation Request
Vacation Balance
System <<Vacation Handeler>>
alt
[if successful]
Add Request (Vacation Type, Request Date, Start, End)
Verify Balance Availability()
Balance Sufficient()
Submit Request()
Save Request()
[else]
Balance Insufficient()
Reject Request()

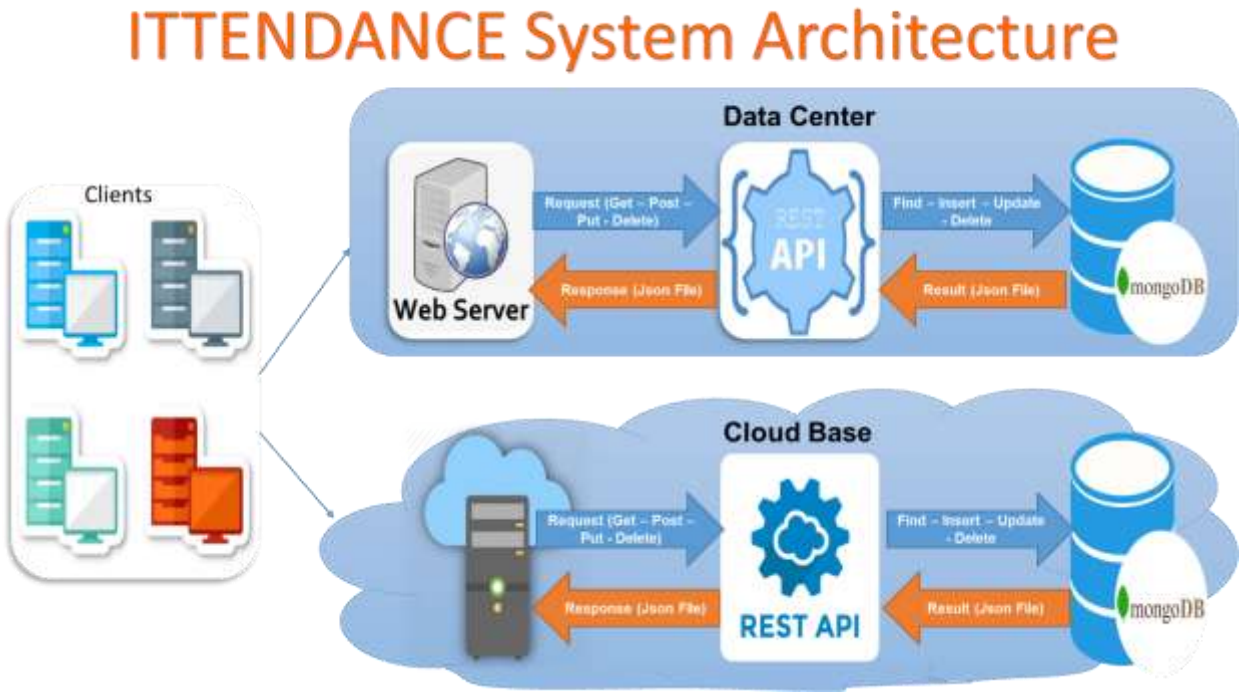
Development Technologies

ITTENDANCE has been developed using MEAN STACK that includes the following technologies:

- MongoDB has been used as the Database.
- Express JS has been used as the Web Application Framework.
- Angular has been used as the Frontend Framework.
- Node JS has been used as the Backend Programming Language.

These technologies were all selected as well as SCRUM methodology to develop and manage ITTENDANCE.

How It Works



Option 1

If the system uses a Data Center:

A get, post, put, or delete request is sent by a client through the web server to RESTFUL API in the **data center** to process a find, insert, update, or delete selected data in Mongo database. The result is returned in a JSON file to RESTFUL API; then, the response is sent in a JSON file too through the web server.

Option 2

If the system uses a Cloud Base:

A get, post, put, or delete request is sent by a client through the web server to RESTFUL API in a **cloud base** to process a find, insert, update, or delete selected data in Mongo database. The result is returned in a JSON file to RESTFUL API; then, the response is sent in a JSON file too through the web server.

Methodology

There are several reasons behind choosing “Scrum Methodology” for this project, and they are as follows:

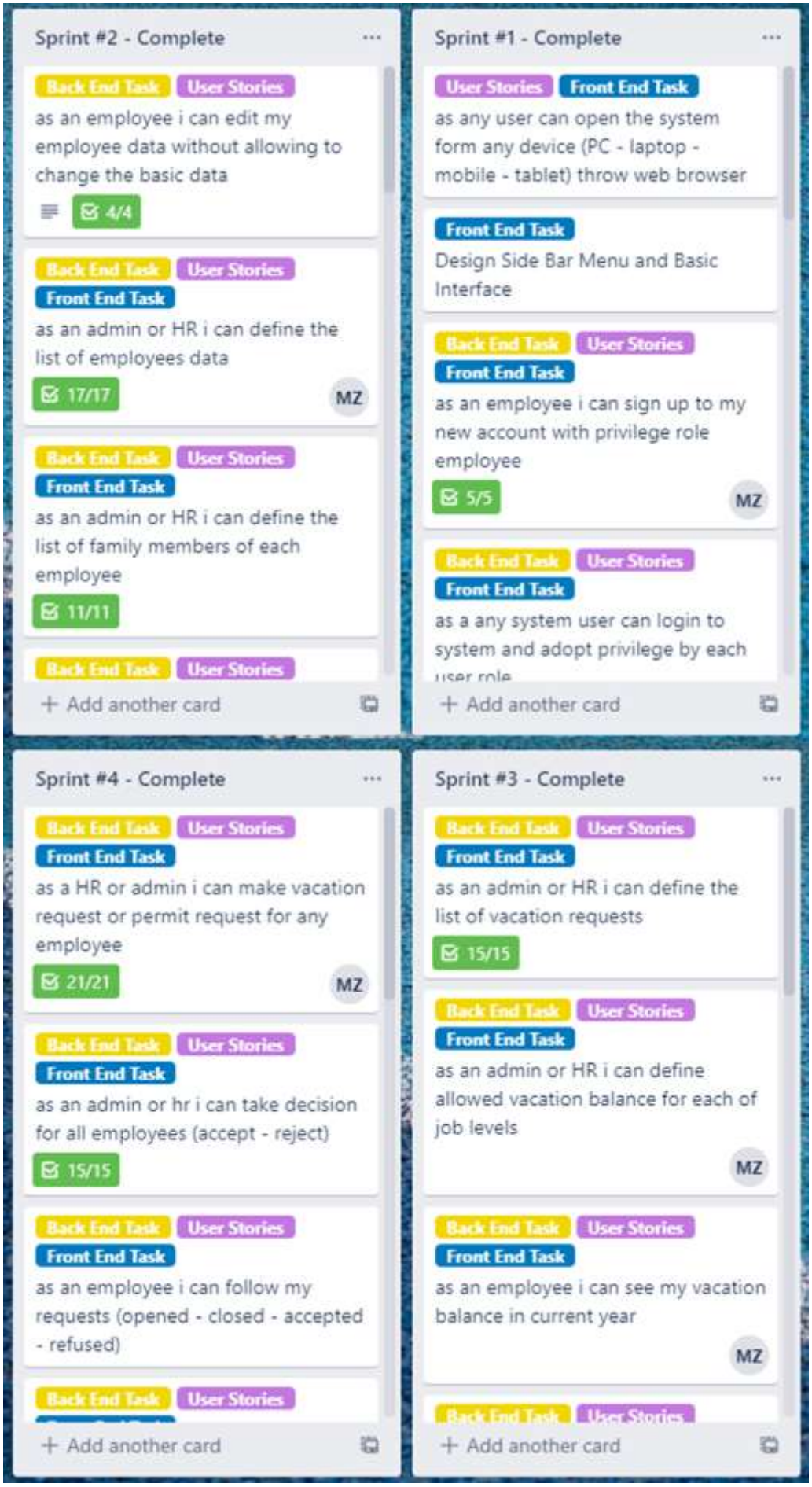
1. The project team “Smart Falcon Team” consists of only three members.
2. Team members are working remotely.
3. Team members need to take a few minutes per day to talk about any problems or ideas they have.
4. Ease of dealing with changes in requirements is needed.
5. Delivering value frequently is highly required. This is due to unexpected changes in final delivery time.
6. Continuous improvements require the flexibility of Scrum.

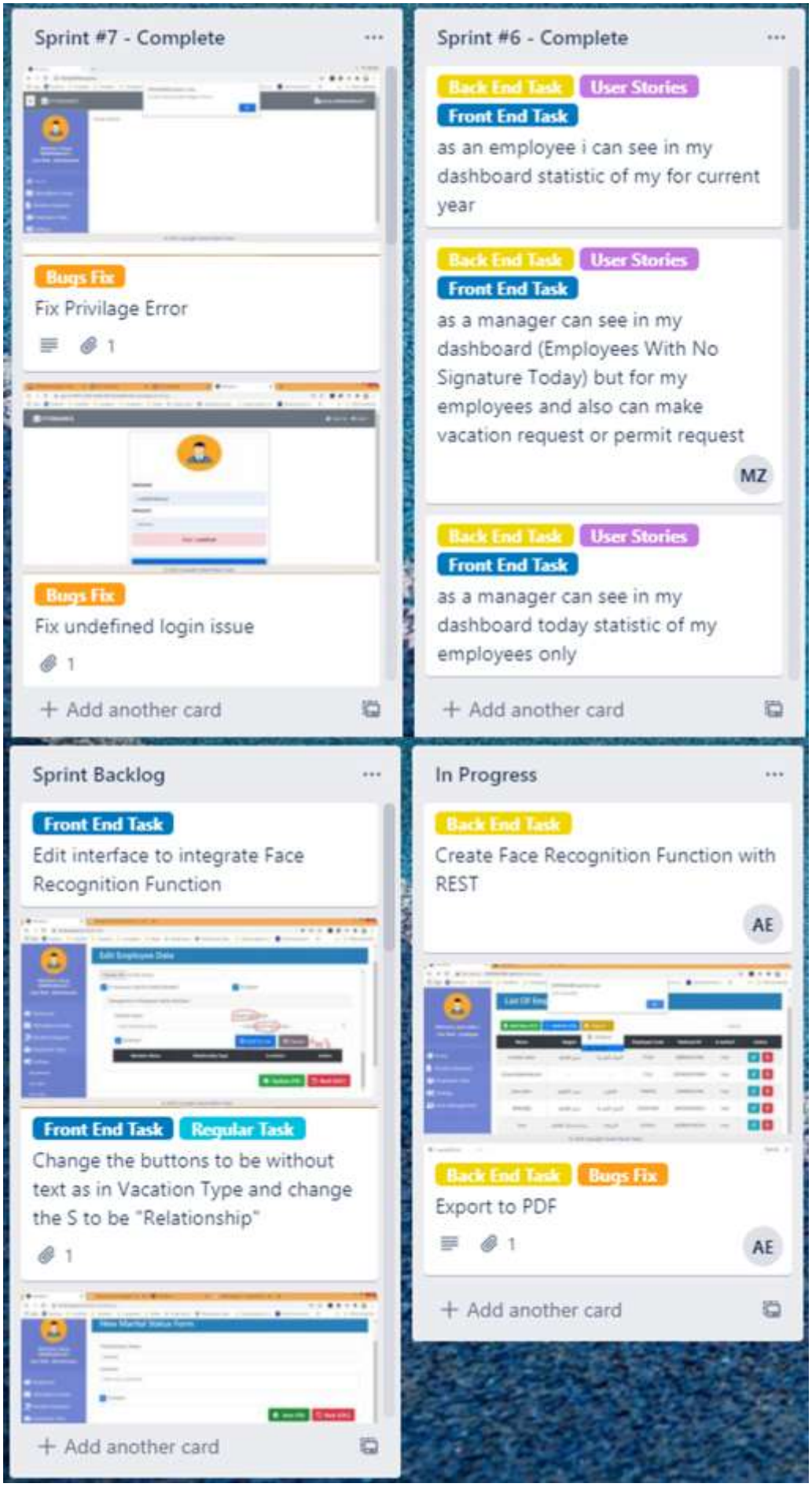
For this project, each sprint will last for fifteen days, and the whole project is expected to last for six to eight sprints. An agile tool is needed to run this project, and “Trello” platform is one of the best choices to go for.

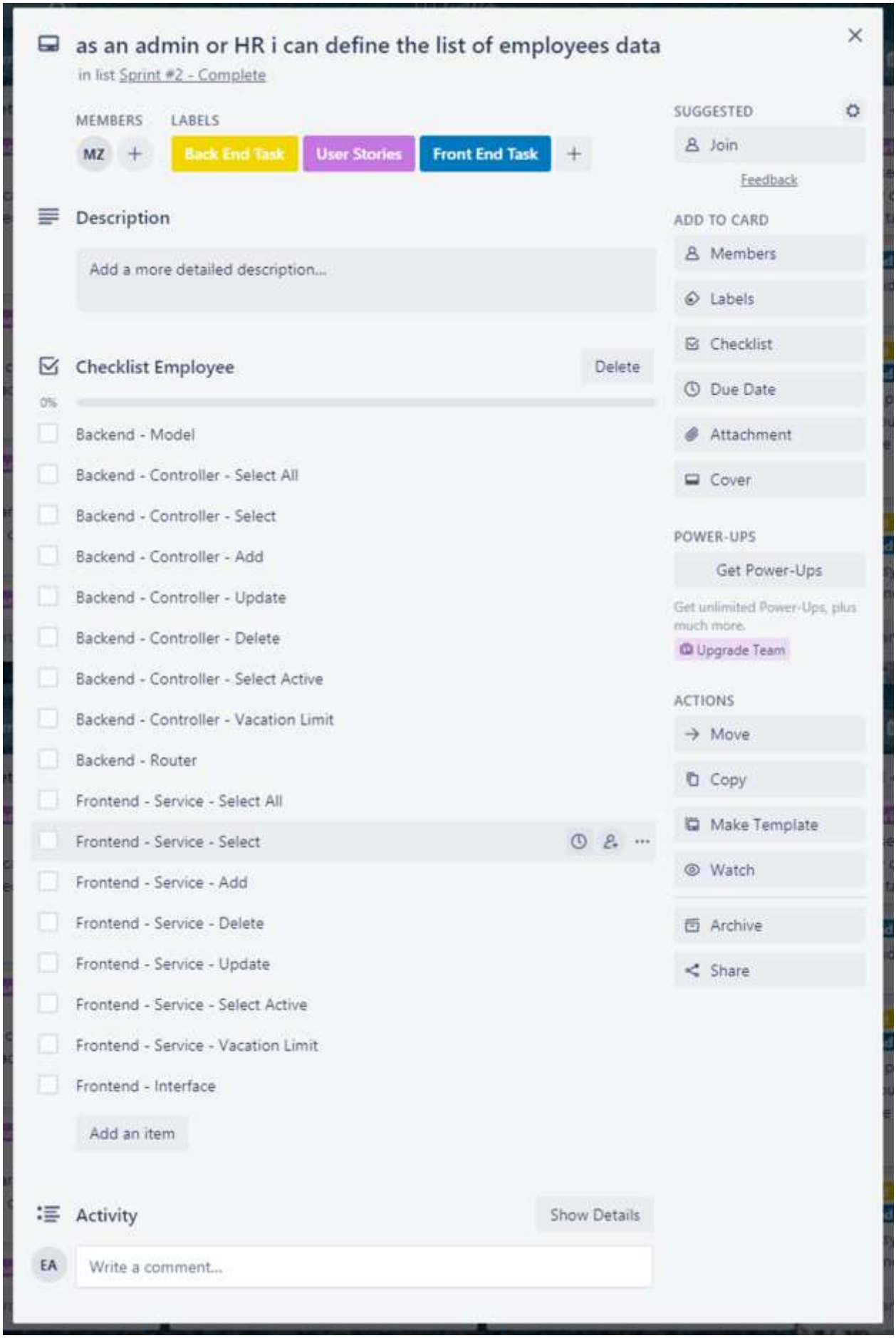
“Trello” allows creating a board or more with lists for backlog, sprint backlog, in progress items, sprint complete items, and any other required list. In these lists, card added to write headlines. In addition to this, the following functions are available for each card:

1. Description: A description about the card can be written to provide more details about the problem.
2. Label: Labeling the card helps in identifying the task to be performed. For example, frontend tasks are labeled in blue, backend tasks are labeled in yellow, bug tasks are labeled in orange, and documentation tasks are labeled in pink.
3. Attachment: An attachment or more can be added to each card to give more details about the task.
4. Members: A member or more can be added to each card to mark that they are responsible for performing the task of the card.
5. Checklist: A checklist in a card can be used to divide it into parts that should be performed together and cannot be in different cards.
6. Due date: Setting a due date helps when a task should be performed early after the end of a sprint to use its output in the same sprint for another task.
7. Comment: Comments can be added to explain more about activities and provide more details for other team members.

Below are some screenshots from Trello. Click here to visit the board.

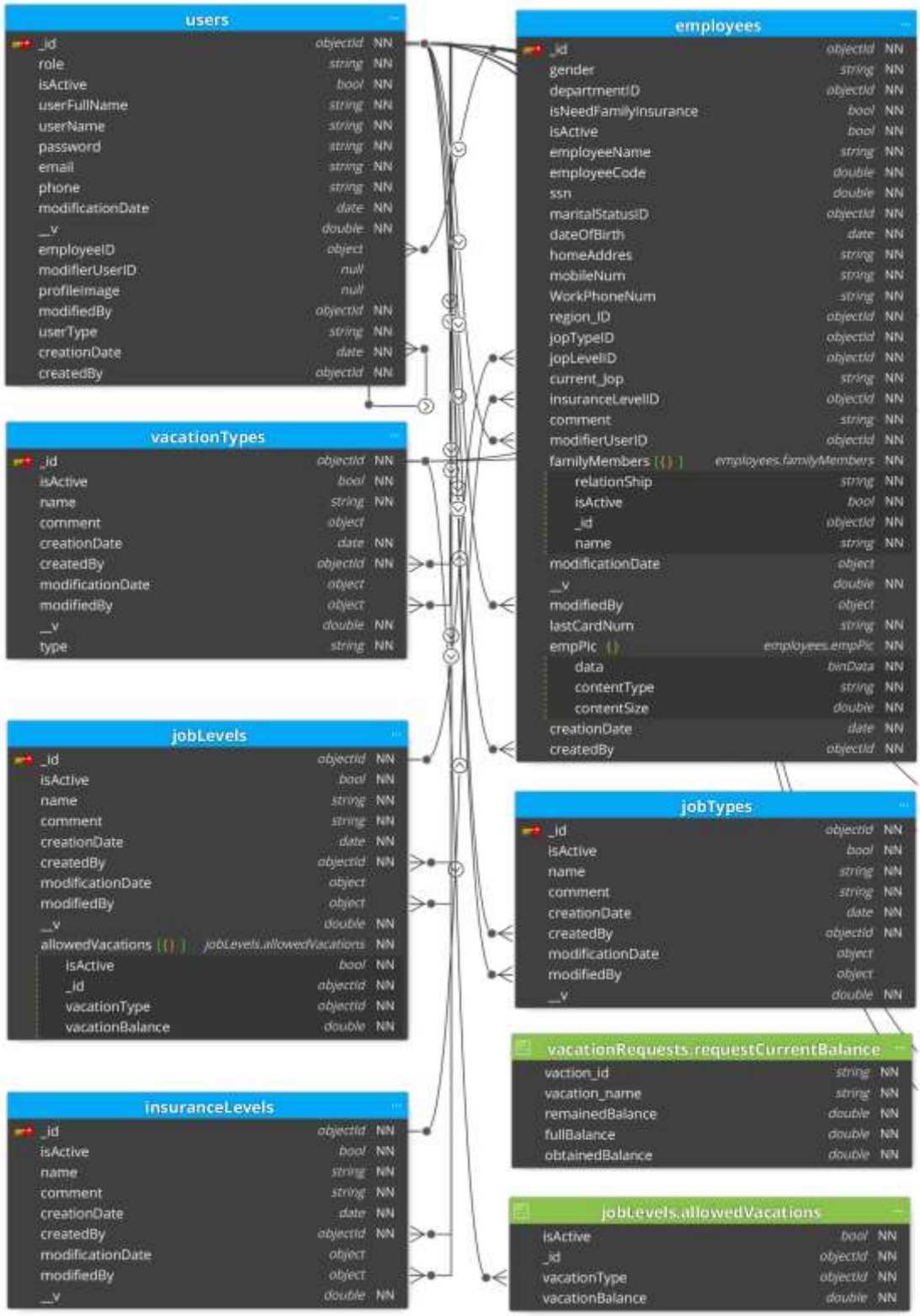

Sprint #2 - Complete
Back End Task
User Stories
as an employee i can edit my employee data without allowing to change the basic data
4/4
Back End Task
User Stories
Front End Task
as an admin or HR i can define the list of employees data
17/17
MZ
Back End Task
User Stories
Front End Task
as an admin or HR i can define the list of family members of each employee
11/11
Back End Task
User Stories
Add another card
Sprint #1 - Complete
User Stories
Front End Task
as any user can open the system form any device (PC - laptop - mobile - tablet) throw web browser
Front End Task
Design Side Bar Menu and Basic Interface
Back End Task
User Stories
Front End Task
as an employee i can sign up to my new account with privilege role employee
5/5
MZ
Back End Task
User Stories
Front End Task
as a any system user can login to system and adopt privilege by each
Add another card
Sprint #4 - Complete
Back End Task
User Stories
Front End Task
as a HR or admin i can make vacation request or permit request for any employee
21/21
MZ
Back End Task
User Stories
Front End Task
as an admin or hr i can take decision for all employees (accept - reject)
15/15
Back End Task
User Stories
Front End Task
as an employee i can follow my requests (opened - closed - accepted - refused)
Back End Task
User Stories
Add another card
Sprint #3 - Complete
Back End Task
User Stories
Front End Task
as an admin or HR i can define the list of vacation requests
15/15
Back End Task
User Stories
Front End Task
as an admin or HR i can define allowed vacation balance for each of job levels
MZ
Back End Task
User Stories
Front End Task
as an employee i can see my vacation balance in current year
MZ
Back End Task
User Stories
Add another card

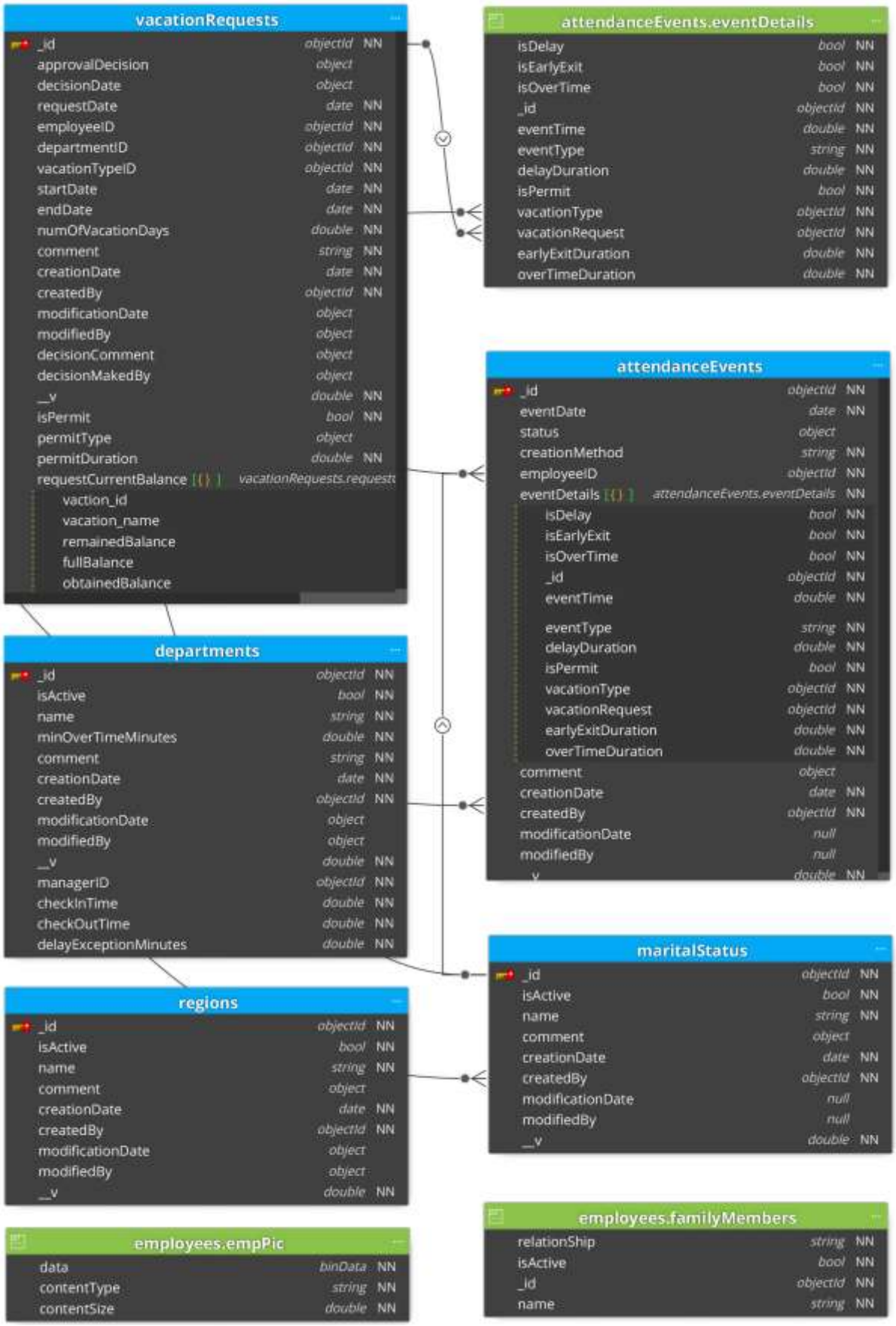

Sprint #7 - Complete
Bugs Fix
Fix Privilage Error
1
Bugs Fix
Fix undefined login issue
1
Add another card
Sprint #6 - Complete
Back End Task
User Stories
Front End Task
as an employee i can see in my dashboard statistic of my for current year
Back End Task
User Stories
Front End Task
as a manager can see in my dashboard (Employees With No Signature Today) but for my employees and also can make vacation request or permit request
MZ
Back End Task
User Stories
Front End Task
as a manager can see in my dashboard today statistic of my employees only
Add another card
Sprint Backlog
Front End Task
Edit interface to integrate Face Recognition Function
Front End Task
Regular Task
Change the buttons to be without text as in Vacation Type and change the S to be "Relationship"
1
Add another card
In Progress
Back End Task
Create Face Recognition Function with REST
AE
Back End Task
Bugs Fix
Export to PDF
1
AE
Add another card


as an admin or HR i can define the list of employees data
in list Sprint #2 - Complete
MEMBERS
LABELS
MZ
Back End Task
User Stories
Front End Task
Description
Add a more detailed description...
Checklist Employee
Delete
0%
Backend - Model
Backend - Controller - Select All
Backend - Controller - Select
Backend - Controller - Add
Backend - Controller - Update
Backend - Controller - Delete
Backend - Controller - Select Active
Backend - Controller - Vacation Limit
Backend - Router
Frontend - Service - Select All
Frontend - Service - Select
Frontend - Service - Add
Frontend - Service - Delete
Frontend - Service - Update
Frontend - Service - Select Active
Frontend - Service - Vacation Limit
Frontend - Interface
Add an item
Activity
Show Details
EA
Write a comment...
SUGGESTED
Join
Feedback
ADD TO CARD
Members
Labels
Checklist
Due Date
Attachment
Cover
POWER-UPS
Get Power-Ups
Get unlimited Power-Ups, plus much more.
Upgrade Team
ACTIONS
Move
Copy
Make Template
Watch
Archive
Share

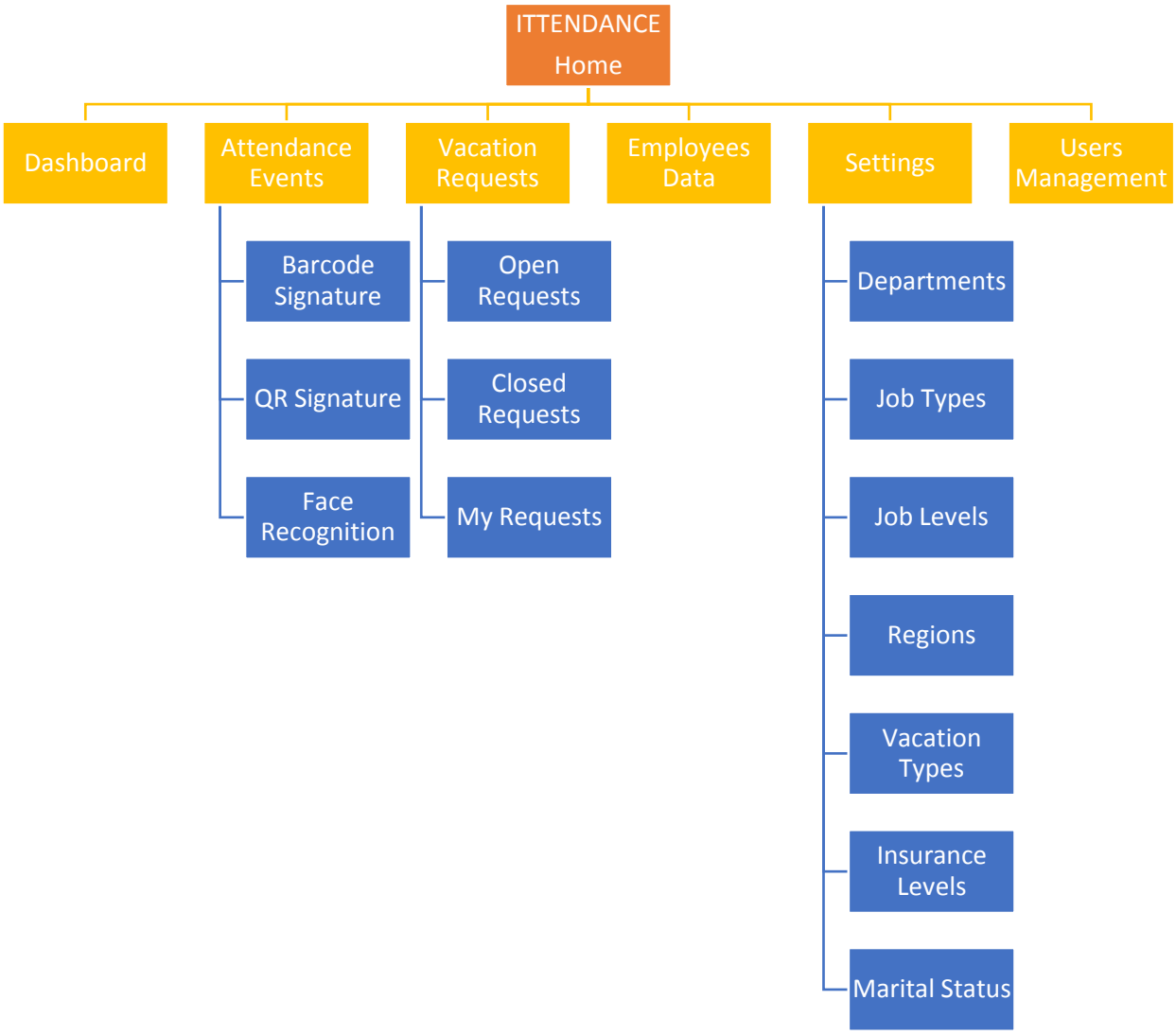
Database

MongoDB is the database program used for building the document database in this project. The following below was created by Moon Modeler, a database modeling tool used for MariaDB, GraphQL, and MongoDB. Click here to view the full model.




vacationRequests
_id objectId NN
approvalDecision object
decisionDate object
requestDate date NN
employeeID objectId NN
departmentID objectId NN
vacationTypeID objectId NN
startDate date NN
endDate date NN
numOfVacationDays double NN
comment string NN
creationDate date NN
createdBy objectId NN
modificationDate object
modifiedBy object
decisionComment object
decisionMakedBy object
__v double NN
isPermit bool NN
permitType object
permitDuration double NN
requestCurrentBalance [{}] vacationRequests.requestC
vaction_id
vacation_name
remainedBalance
fullBalance
obtainedBalance
attendanceEvents.eventDetails
isDelay bool NN
isEarlyExit bool NN
isOverTime bool NN
_id objectId NN
eventTime double NN
eventType string NN
delayDuration double NN
isPermit bool NN
vacationType objectId NN
vacationRequest objectId NN
earlyExitDuration double NN
overTimeDuration double NN
attendanceEvents
_id objectId NN
eventDate date NN
status object
creationMethod string NN
employeeID objectId NN
eventDetails [{}] attendanceEvents.eventDetails NN
isDelay bool NN
isEarlyExit bool NN
isOverTime bool NN
_id objectId NN
eventTime double NN
eventType string NN
delayDuration double NN
isPermit bool NN
vacationType objectId NN
vacationRequest objectId NN
earlyExitDuration double NN
overTimeDuration double NN
comment object
creationDate date NN
createdBy objectId NN
modificationDate null
modifiedBy null
__v double NN
departments
_id objectId NN
isActive bool NN
name string NN
minOverTimeMinutes double NN
comment string NN
creationDate date NN
createdBy objectId NN
modificationDate object
modifiedBy object
__v double NN
managerID objectId NN
checkInTime double NN
checkOutTime double NN
delayExceptionMinutes double NN
maritalStatus
_id objectId NN
isActive bool NN
name string NN
comment object
creationDate date NN
createdBy objectId NN
modificationDate null
modifiedBy null
__v double NN
regions
_id objectId NN
isActive bool NN
name string NN
comment object
creationDate date NN
createdBy objectId NN
modificationDate object
modifiedBy object
__v double NN
employees.empPic
data binData NN
contentType string NN
contentSize double NN
employees.familyMembers
relationShip string NN
isActive bool NN
_id objectId NN
name string NN

Site Map


ITTENDANCE
Home
Dashboard
Attendance Events
Vacation Requests
Employees Data
Settings
Users Management
Barcode Signature
QR Signature
Face Recognition
Open Requests
Closed Requests
My Requests
Departments
Job Types
Job Levels
Regions
Vacation Types
Insurance Levels
Marital Status

Software Quality - Review Checklist

Requirements

- ☐ Is ITTENDANCE capable of handling attendance operations in a timely manner?
- ☐ Is ITTENDANCE a cloud-based system with separate backend from the frontend?
- ☐ Is ITTENDANCE in English?
- ☐ Does ITTENDANCE ban signing up without “Employee Code”, “National ID”, ”Region”, and ”Job Level”?
- ☐ Does ITTENDANCE allow registering information of foreign employees?
- ☐ Does ITTENDANCE allow adding insurance information?
- ☐ Does ITTENDANCE allow adding dependents information?
- ☐ Does ITTENDANCE have a QR code scanner?
- ☐ Does ITTENDANCE have a barcode scanner?
- ☐ Does ITTENDANCE support signature by face recognition?
- ☐ Does ITTENDANCE allow submitting vacation requests?
- ☐ Does ITTENDANCE calculate vacations balance per user?
- ☐ Does ITTENDANCE allow requesting permits?
- ☐ Does ITTENDANCE allow approving vacation requests and permits by HR employees and Admins?
- ☐ Does ITTENDANCE have a dashboard to show attending employees, absent employees, delays, permits, and early exists?
- ☐ Does ITTENDANCE allow employees to access their “Dashboard” data and their “Vacation Requests” only?
- ☐ Does ITTENDANCE allow supervisors to access their “Dashboard” data, “Vacation Requests” and “Attendance Events” including attendance of employees of their department at other departments?
- ☐ Does ITTENDANCE allow HR employees to access “Dashboard”, “Vacation Requests”, “Attendance Events”, “Settings”, and “Employees Data” of all departments?
- ☐ Does ITTENDANCE allow Admins to access everything on the system?
- ☐ Does ITTENDANCE allow exporting reports in PDF format?
- ☐ Does ITTENDANCE allow exporting reports in Excel format?
- ☐ Does ITTENDANCE enable exporting reports of “Employees Data”, “Users Data”, and “Vacation Requests”?
- ☐ Does ITTENDANCE enable exporting reports of “Settings”?

- Does ITTENDANCE enable creating, retrieving, updating, and deleting “Users”?
- Does ITTENDANCE enable creating, retrieving, updating, and deleting “Vacation Requests”?
- Does ITTENDANCE enable creating, retrieving, updating, and deleting “Employees Data”?
- Does ITTENDANCE enable creating, retrieving, updating, and deleting “Departments”?
- Does ITTENDANCE enable creating, retrieving, updating, and deleting “Insurance Levels”?
- Does ITTENDANCE enable creating, retrieving, updating, and deleting “Job Types”?
- Does ITTENDANCE enable creating, retrieving, updating, and deleting “Job Levels”?
- Does ITTENDANCE enable creating, retrieving, updating, and deleting “Vacation Types”?
- Does ITTENDANCE enable creating, retrieving, updating, and deleting “Regions”?
- Does ITTENDANCE ban viewing content without signing in?
- Are ITTENDANCE users allowed to change their passwords?
- Are ITTENDANCE admins allowed to reset passwords of other users?
- Is having a password typical to the username or less than eight characters in length banned?
- Does ITTENDANCE hash passwords?
- Does ITTENDANCE have a vacation limit for each job level?
- Does ITTENDANCE have a check-in time, checkout time, delay time limit, and minimum limit for overtime per department?

Planning

- ☐ Are the objectives of ITTENDANCE fully stated?
- ☐ Is the scope of ITTENDANCE clearly defined?
- ☐ Does the requirements list contain all the requirements of ITTENDANCE?
- ☐ Is the schedule of ITTENDANCE applicable and estimated correctly?
- ☐ Are the required resources available?
- ☐ Are all tasks broken into sub-tasks?
- ☐ Are the required diagrams of ITTENDANCE available?
- ☐ Are the deadlines realistic?

Maintenance

- ☐ Is the code of ITTENDANCE written in a reusable and easy maintained manner?
- ☐ Is there a documentation for ITTENDANCE?
- ☐ Are testing activities of ITTENDANCE completed?
- ☐ Are the side effects related to code changes considered?
- ☐ Are the changes of ITTENDANCE being reported and documented?

Quality Grid

The following are McCall and Matsumoto definitions of software quality factors:

Operation

Integrity: How well the system handles physical disturbances and prevents malicious access attempts, etc. (Today *security* would be a better term.)

Correctness: How many errors there are in the system and the extent to which a program satisfies its specifications and fulfills the user's mission objectives.

Reliability: How frequently the system malfunctions (MTBF, Mean Time Between Failures) and the percentage of time it is available (availability, e.g. 99%).

Usability: How easy it is to learn the system, how efficient it is for carrying out day-to-day tasks, etc.

Efficiency: How fast the system responds, how many resources it uses, how accurately it computes values, etc. (*Performance* is another term for this.)

Revision

Maintainability: How easy it is to locate and repair errors.

Testability: How easy it is to test the system after a change.

Flexibility: How easy it is to expand the system with new features.

Transition

Portability: How easy it is to move the system to a new software or hardware platform.

Interoperability: How easy it is for the system to cooperate with other systems, e.g. file transfer to spreadsheets, attachment of new hardware units.

Reusability: How easy it is to reuse parts of the software in other systems.

Installability: How easy it is to install the system.

Concerns

1. The system must allow the registration of a large number of employees.
2. The system must cloud-based with separate backend from the frontend.
3. The system must be in English language.
4. Employees must sign up first as users by their “Employee Code”, “National ID”, ”Region”, and ”Job Level” to use the system.
5. Users must log into the system to see content and use it.
6. The system must allow the registration of employees with foreign nationalities.
7. The system must allow adding information of employees’ dependents.
8. The system should allow adding insurance details of employees and their dependents.
9. The system must have a QR code scanner.
10. The system may have a barcode scanner.
11. The system may support signature by face recognition.
12. The system must enable submission of vacation requests.
13. The system must enable requesting permits.
14. The system should calculate available vacations balance for HR decision makers.
15. The system must enable approving permits.
16. The system must enable approving vacation requests.
17. The system may have a dashboard to show the number of attending employees, absent employees, delays, permits, and early exists.
18. The system must allow employees to access their “Dashboard” data and “Vacation Requests” only.
19. The system must allow supervisors to access their “Dashboard” data, “Vacation Requests” and “Attendance Events” only. Accessing “Attendance Events” include attendance of employees of their department at other departments.
20. The system must allow HR employees to access “Dashboard”, “Vacation Requests”, “Attendance Events”, “Settings”, and “Employees Data” of all departments.
21. The system must allow admins to access everything on the system.
22. The system must hash passwords for security purposes.
23. The system may enable exporting reports in PDF format.
24. The system should enable exporting reports in Excel format.
25. The system should enable exporting reports of “Employees Data”, “Users Data”, and “Vacation Requests”.
26. The system might enable exporting reports of “Settings”.

27. The system language may support Arabic language.
28. The system must include CRUD operations for “Users”.
29. The system must include CRUD operations for “Employees Data”.
30. The system must include CRUD operations for “Vacation Requests”.
31. The system must include CRUD operations for “Departments”.
32. The system must include CRUD operations for “Insurance Levels”.
33. The system must include CRUD operations for “Job Types”.
34. The system must include CRUD operations for “Job Levels” of employees.
35. The system must include CRUD operations for “Vacation Types” of employees.
36. The system must include CRUD operations for “Regions” of employees.
37. The system might include CRUD operations for “Marital Status” of employees.
38. Users must be able to change their passwords.
39. The system must not allow having a password that is typical to the username or less than eight characters in length.
40. Administrators must be able to reset passwords of other employees.
41. For each job level, there must be a vacation limit.
42. For all employees, there should be “Regular”, “Sick leave”, “Religious observance”, “Parental leave”, “Paid leave”, and more can be added or edited. This is to be used for future integration with payroll.
43. For each department, there should be a check-in time, checkout time, delay time limit, and minimum limit for overtime.
44. The system does not manage time shifts.
45. The system must replace the majority of paper work.

Quality Factors	Critical	Important	As usual	Unimportant	Ignore
Operation					
Integrity/security	5,22,38, 39,40	4			
Correctness	1,9,20,21, 28, 29,31	2,14,24,25, 32,33,36,41	8,18,19,23,30	17,34,43	35,37
Reliability	2				
Usability	3,14,28, 29,31,44	6,7,9,11,25,27, 32,33,36,42	12,13,15,16,30	10,34	35,37
Efficiency	9,11,14,24	17	12,13,15,16	10	26
Revision					
Maintainability	28,44	3			
Testability					
Flexibility	44		26		
Transition					
Portability	3	2	27		
Interoperability		24	8,23,24		
Reusability		24	23		
Installability	45	3	27		