2017 COS 301 Project
Project Management System

TEAM CODE 9

Seonin David

Joshua Moodley

Jaques Smulders

Jordan Daubinet

Nicaedin Suklul

KPMG

Contents

1 Introduction

1.1 Purpose

The purpose of this document is to provide a detailed description of the requirements for the Project Management System. It will explain the purpose and features of the system, as well as system constraints. This document is intended to be proposed to University of Pretoria lecturers and the KPMG the project owner, as well as a reference for developing the first version of the project management system.

1.2 Scope

KPMG currently have a web based project management system named EMS which handles allocation and management of assigned projects for all employees and project managers.The current system does not cover the spectrum of requirements that they demand. They require further functionality from the system and they wish to control more business needs and communication through a single portal. The client needs require us to create a new design of their current system and an update of their current interface.

KPMG's project management system will be a web based application that allows for the creation and management of projects. It will handle a notification system for notification requests and notification acknowledgements and and per-user calendar overview to display all current project information currently assigned to that user. For all projects currently administered, the system is also required to display client Location and direction information to the employees assigned to the project.

KPMG's project management system will further require Microsoft Outlook integration in the form of a toolbar to be created for Microsoft Outlook which manages the chain of responsibility for notification requests and notification acknowledgements.

2 Overall Description

2.1 Product Perspective

2.1.1 System interfaces

TBA

2.1.2 User interfaces

A web application will be available for the employee to interact with, as well as a quick response feature that will be provided in Outlook.

2.1.3 Hardware interfaces

N/A

2.1.4 Software interfaces

Our system will run on a Microsoft environment.

2.1.5 Communication Interfaces

Outlook will be the medium through which the user will interact and receive notifications. The protocols used will be HTTP/S with addition to SMTP/POP3/IMAP. Futhermore, communication will be encapsulated by Outlook's API.

2.1.6 Memory

For our system we will be using a MongoDB cluster, that will be hosted on a NodeJS server.

2.2 Constraints

- The project must be completed in the given time frame with progress on track with demo expectations.

- The project must not utilize technologies with require cost resources.

- The project must utilize Microsoft Outlook as a communication interface for sending and confirming requests

- The project must be run as a local host and not deployed on a company server

2.3 Assumption and Dependencies

3 Specific Requirements

3.1 Functional Requirements

3.1.1 FR1: Employee Functions

- **FR1.1**: An employee will be able to log in to their account on the system

- **FR1.2**: An employee will be able to create their own profile with their respective skills and previous work experience

- **FR1.3**: An employees will receive a notification when they are assigned to a newly created project

- **FR1.4**: An Employee must be able to to view their currently assigned project and view their past completed projects that where assigned to them on their profile.

- **FR1.5**: An employee will be able to add personal or other events to the calendar their calender, that do not clash with any existing projects they are currently assigned to. If a newly created event does clash, an error will be displayed.

- **FR1.6**: When an employee clicks on a project in the calendar, it will show all the project details and a map of the clients location should be displayed as well as directions to the client.

- **FR1.7**: An employees will not be able to remove a project that is assigned to them from their calender.

- **FR1.8**: Employees should confirm wheather they have attended training or not, this will be conformed with trainer.

3.1.2 FR2: Manager functions

- **FR2.1**: A manager will be able to log in and create a project.

- **FR2.2**: A manager will be able to assign system recommended employees to a newly created projects

- **FR2.3**: A manager will be able change employees if the manager is unhappy with the recommendation by the system, pending director approval.

- **FR2.4**: A manager will be allowed to remove a project from the system, pending director approval

- **FR2.5**: A project manager will be able to view all projects that they are managing

3.1.3 FR3: System Functions

- **FR3.1**: The system will assign employees to a project based on their skill level and past project experience.

- **FR3.2**: The system will find all the employees that have the required skills for the job and give it to the manager as a recommendation.

- **FR3.3**: The system will only select employees if they are not already assigned to a project.

- **FR3.4**: Employee data with skills and past work experience will be entered on the system and updated after each project.

- **FR3.5**: System will reassign another employee, pending approval, if an employee takes leave during the project.

- **FR3.6**: Every employee that is assigned to a project will get the full project duration and details on their calendar.

- **FR3.7**: The system should be able to provide the location of and direction to the client to all employees assigned to the respective project

- **FR3.8**:The system will preload all employees training dates at the beginning of the year.

- **FR3.9**: The system will preload all employees training dates at the beginning of the year.

3.2   Non Functional Requirements

- Color Scheme - There will have to be a pleasing colour scheme for the system, one that represents the colours of KPMG

- Security - The system will have to cater for employee and client security, There will have to be measures put into place to prevent the database and server falling prey to malicious attacks.

- Performance - The system will have to manage many users at the same time, mostly based on the number of employees stationed at the Pretoria branch

- Concurrency - The system will have to support multiple users at the same time

- Scalability - The system will have to be able to adapt to a larger environment and expand beyond its normal scope of function

3.3   Architectural Requirements

T   he system will be designed using the n-tier architecture (Specifically 3-tier), where the system is divided into three layers

- 1. Presentation Layer

- 2. Application Layer

- 3. Data Layer

Each layer will be modifiable without having to change the entire application. As a result of the aforementioned, maintenance and adding extra functionality is easier. Overall complexity of code over all layers will be reduced thus making the layers reusable in other applications.By using this architecture system will allow the different members in our teams to modify different layers without interference. It is possible to deploy each layer (Specifically the data and application layers) over multiple different locations for better reliability and performance.

3.3.1 Presentation Layer

T his is the layer the user will be interacting with. It encompasses the actual apps (Android, iOS and web), and any interactions that the users have with them. **The presentation layer include:**

- What to display and when to display it, it's the logic behind the web-pages and the control between access from one to another.

3.3.2 Application Layer

T he application layer contains all the business logic of the system - it makes logical decisions based on the interactions from the presentation layer and the data from the data layer.
The decisions made by this layer include:

- Allow admin users to be able to add/ remove client projects and assign these to employees who will be working on these projects.

- Automatically update calendars of the employees.

- Crosscheck projects before assigning projects to an employee.

- Server communication over the internet.

- Scheduling Assistant calculations.

- Limit an employee from being able to edit the projects that have been prepopulated.

- Allow an employee to add other events to his/ her calendar if these do not clash with the prepopulated events.
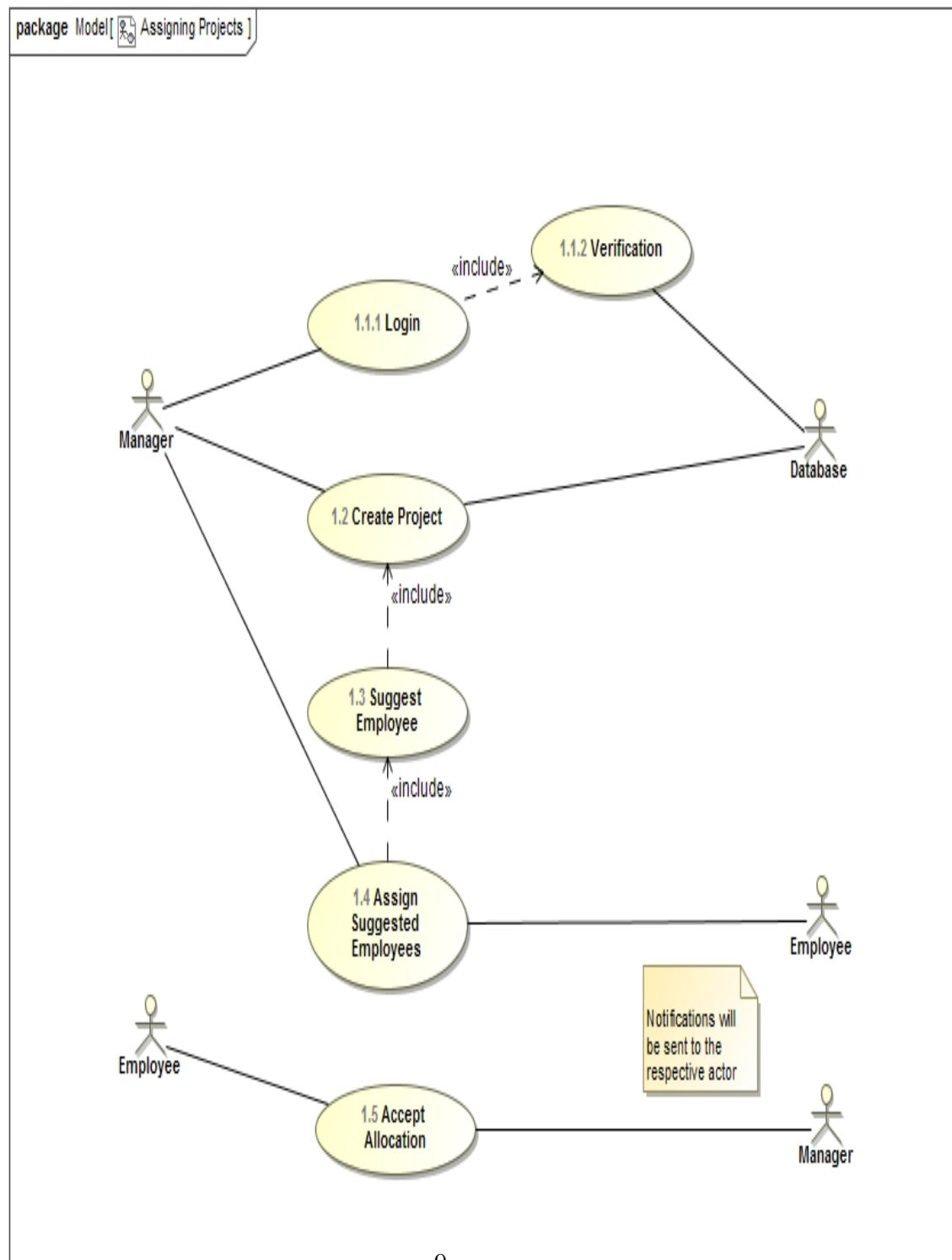
- Session management.

3.3.3 Data Layer

The data layer is where the data and information used by the application layer is stored. This information includes:
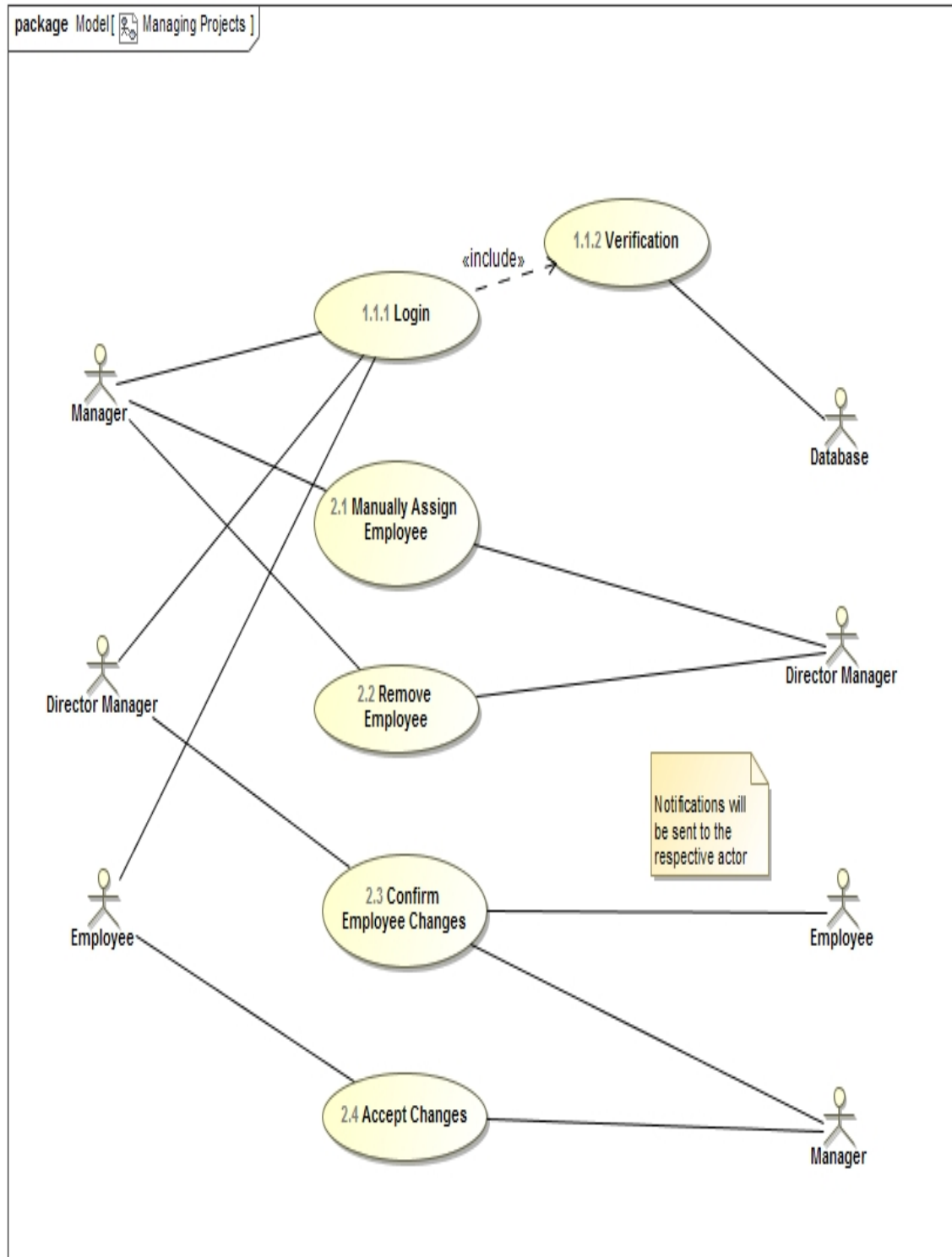
- Recommended time slots where all employees are available.

- Server storage with mongodb

- View individual employee calendar and assigned projects.
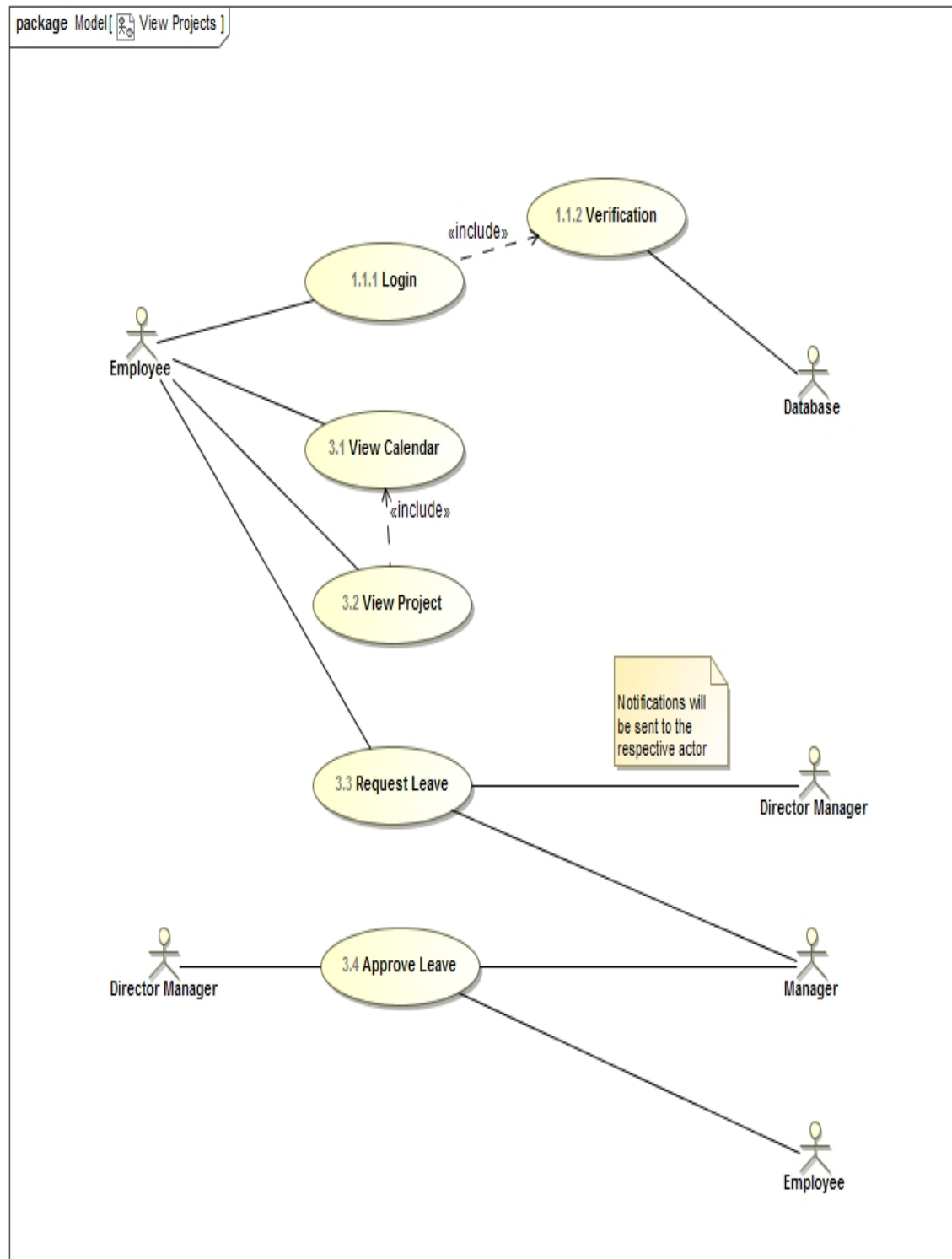
4 Use Case Diagrams
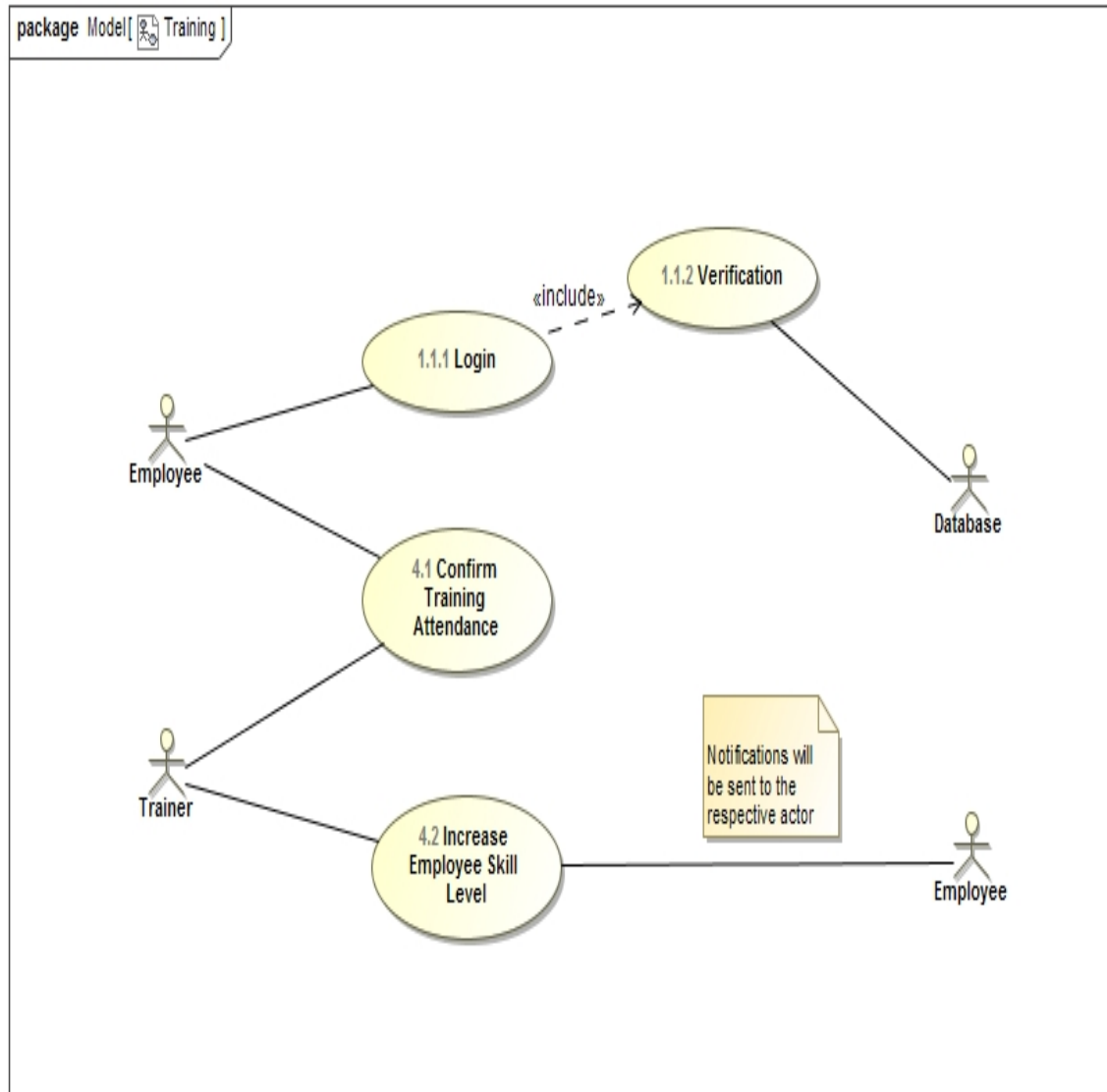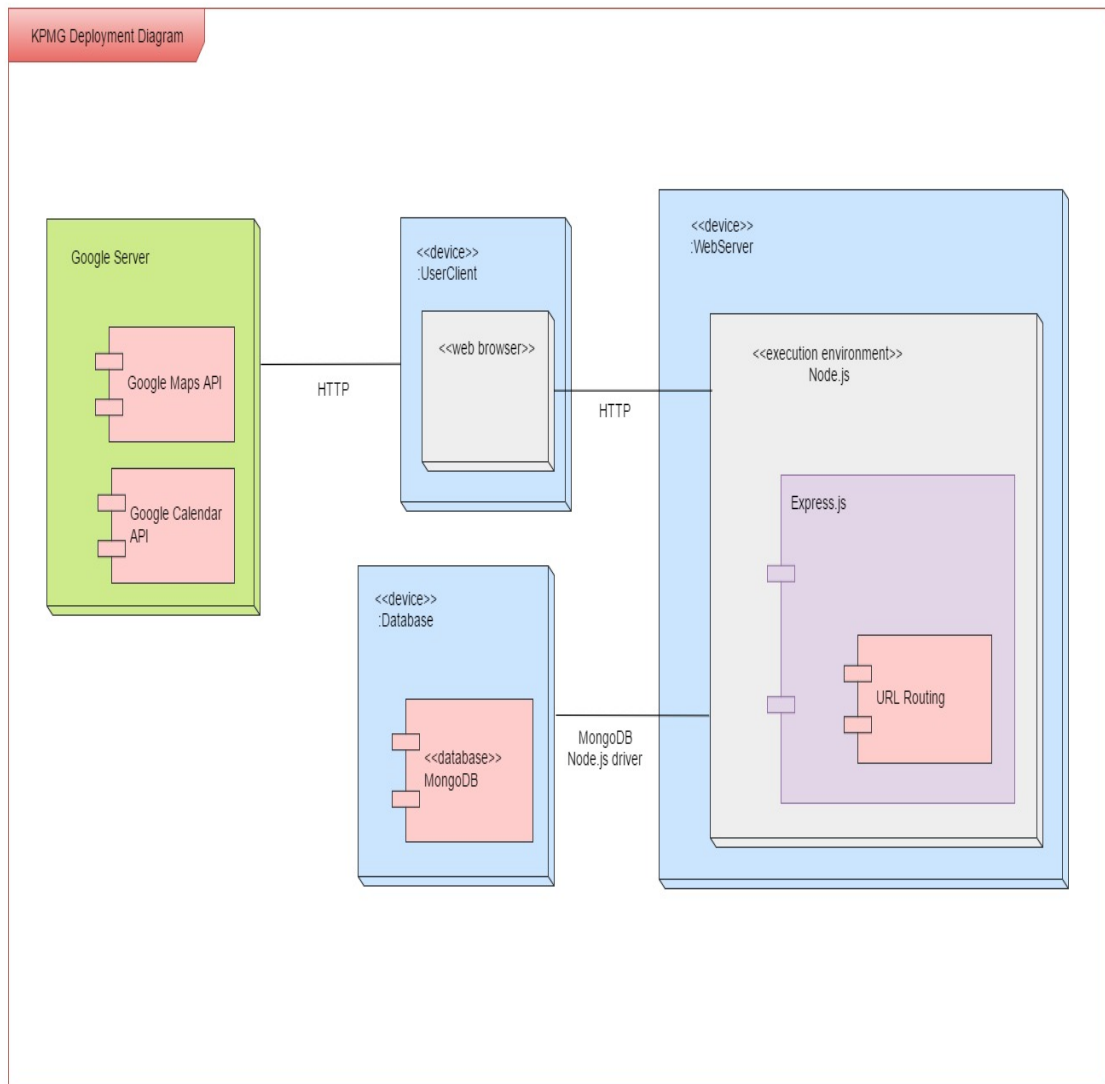
4.1 Assigning Projects

4.2 Managing Projects

4.3   Veiw Projects

4.4 Training

5 Deployment Diagram

6 Tracibility Matrix

Table 1: Tracibility Matrix

	Priority	UC 1.1	UC 1.2	UC 1.3	UC 1.4	UC 1.5	UC 2.1	UC 2.2	UC 2.3	UC 2.4	UC 3.1	UC 3.2	UC 3.3	UC 3.4	UC 4.1	UC 4.2
FR 1																
FR 1.1	2	X														
FR 1.2	1										X					
FR 1.4	3											X				
FR 2																
FR 2.1	2	X														
FR 2.2	3		X													
FR 2.3	1							X								
FR 3																
FR 3.1	4				X											
FR 3.2	1			X									X			
FR 3.3	5			X									X			
FR 3.4	6			X									X			
FR 3.5	2						X									
FR 3.6	3											X				
FR 3.8	7														X	

14