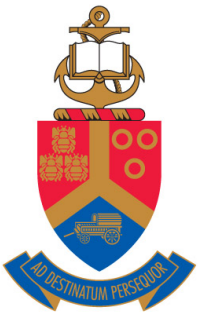

UNIVERSITEIT VAN PRETORIA
UNIVERSITY OF PRETORIA
YUNIBESITHI YA PRETORIA

Denkleiers • Leading Minds • Dikgopolo tša Dihlalefi

Capstone 2017
Project Management System

Seonin David

Joshua Moodley

Jacques Smulders

Jordan Daubinet

Nicaedin Suklul

KPMG

Contents

Document Title	Software Requirements Specification
Document Identification	Document 0.0.2
Author	Seonin David, Joshua Moodley, Jaques Smulders, Jordan Daubinet, Nicaedin Suklul
Version	0.0.2
Document Status	Second Version - contains an updated non-functional requirements section, as well as a new format for the Architectural Requirements

Version	Date	Summary	Authors
0.0.1	25 May 2017	First draft contains functional requirements, use case diagrams and a deployment diagram	Seonin David, Joshua Moodley, Jaques Smulders, Jordan Daubinet, Nicaedin Suklul
0.0.2	21 June 2017	Second draft contains an updated structure for Architectural requirements and non-functional requirements	Seonin David, Joshua Moodley, Jaques Smulders, Jordan Daubinet, Nicaedin Suklul

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.

1.3 Assumptions and Dependencies

- # : number
- FR : Functional Requirement
- DB : Database
- ms : Milliseconds

2 Architectural Design

2.1 Introduction

The system design to be implemented will be referred to as a Hybrid-tier System. This hybrid model is based off the n-tier (multi-layered) architecture, where the system is divided into layers. This multi-layered system is also referred to as a Client-Sever architecture. For this architecture, there are four tiers/layers. Those being:

1. Presentation Tier
2. Server Tier
3. Database Tier
4. AI (Artificial Intelligence) Algorithm Tier

Each layer will be modifiable without needing 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. Since the layers do not act like a traditional n-tier where one would delegate tasks to other layers; once a request goes to the server tier, the server will decide if it will go to the database tier or the AI tier which then goes to the database tier. Thus showing an optimised parallel system to deal with requests. Additionally, the use of this architecture system will allow different members of our team to modify separate layers without interference. It is possible to deploy each layer (specifically the database and server layers) over multiple locations for better reliability and performance. What differentiates the hybrid architecture from a traditional n-tier or client-server, is in how the layers are implemented. Figure 1 shows the overall architecture of this hybrid system.

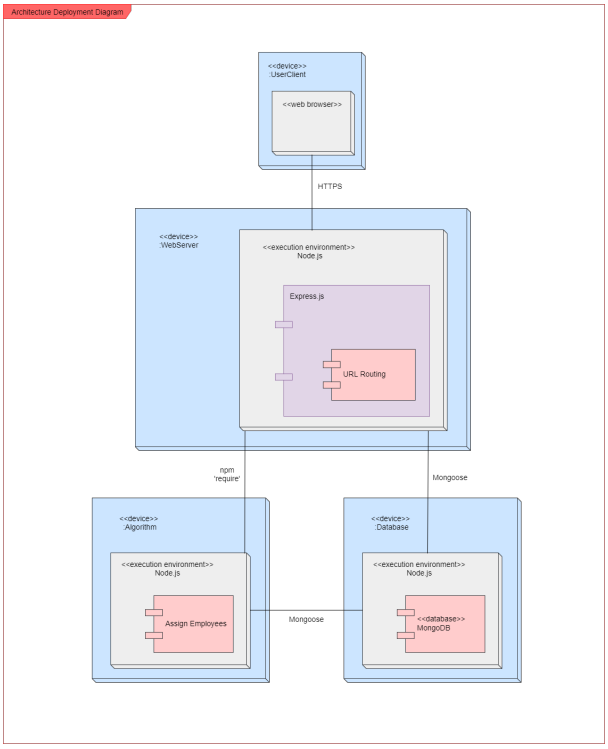

Figure 1: Deployment Diagram: Hybrid-Tier Architecture

2.2 Presentation Tier

This is the layer the user will be interacting with. It encompasses the actual applications that can be developed (Android, iOS and web), and any interactions that the users may have with them. In this system, the user will primarily work with a web application as the front-end.

The presentation tier includes:

- Displaying the user interface (UI), which the user will interact with.
- Adding notification, forms, buttons and pages.
- Providing an interface to trigger requests to the system.
- Ultimately serving as a communication medium between pages on the front-end, and to send and receive data between the back-end.

2.3 Server Tier

The server layer implements the business logic, and process all requests made by the front-end - it makes logical decisions based on the interactions from the presentation layer and then decides what or where data needs to go; either to the database layer or the AI layer.

The server tier is responsible for:

- Allow admin users to be able to add/remove/edit client projects, and assign employees to specific projects.
- Automatically update calendars of employees.
- Crosscheck projects before assigning them to an employee.
- Server communication over the internet.
- Scheduling Assistant calculations.
- Limit an employee from being able to edit projects they have been assigned to.
- Allow an employee to add other events to his/her calendar if these do not clash with the pre-populated events.
- Session management.

2.4 Database Tier

The data layer is where the information used by the presentation layer for sending and retrieving, is stored. This layer consists of:

- Recommended time slots where all employees are available.
- Server storage with MongoDB
- View individual employee calendar and assigned projects.

2.5 AI (Artificial Intelligence) Algorithm Tier

The AI layer houses the resource allocation algorithm, and is used by the server layer to allocate employees to a project. This layer runs off its own independent resources, and is dependent only on the database tier from which it retrieves resources necessary for the allocation algorithm.

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

3.2 Non Functional Requirements

3.2.1 Performance requirements

Reasons for performance as a non-functional requirement

- Performance is one of the core components of what is required from the project management system, it needs to be effective and efficient.

- Performance can play a large role in usability, usability describes the users experience well using the product. The project management system must deliver a positive user experience.

Strategies to achieve this non-functional requirement

- The system needs to be able encompass over 200 concurrent connections.
- The DB needs to handle over 1000 requests and responses each second.
- The latency associated with verifying a users log in credentials must be fall under 200ms.
- The latency associated with requesting and receiving a response for an accessible web page must fall under 50ms.

Patterns to achieve these strategies

- Expert Pattern
- Layer pattern

The Expert pattern can be used to direct requests to different locations. The pattern ensures your program utilizes efficient resource management by directing requests and responses to the respectful parts of the system that can Handel them correctly. This relieves bottle necking a single part of the system with all responsibilities.
Utilizing the layer pattern allows us to divide our system into subsystems which can be run in separate locations.We can divide our initial system into A front end, a back end and a database manager. By splitting our system up, we can alter how many resources we allocate to each system, thus we utilize our recourse's efficiently and provide loose coupling between the different parts of the system.
By combining the expert and layer pattern, we can handle different requests efficiently and correctly with their associated subsystem.

3.2.2 Quality requirements

Reasons for quality as a non-functional requirement

- By stating Quality requirements, you define the desired attributes which your program must have. Defining your programs attributes moulds key factors of its development and sets quality milestones and goals.
- When the project is ready for deployment, the quality requirements which were set at the begging of development define how reliable the end product is or should be.

Strategies to achieve this non-functional requirement

- When Queried to allocate a team, the resource allocation algorithm must produce one within a maximum of 10 seconds.
- The system shall be available 99% of the time, only going down in the duration of maintenance or while implementing updates.
- When queried for a project team and members are available, the resource allocation algorithm should never respond with no allocated team members.

- When queried for a project team by a project manager, the resource allocation algorithm should provide a compatible and satisfactory project team to the project manager 85% of queries made.

Patterns to achieve these strategies

- This will depend on the pattern used by the artificial intelligence resource allocation algorithm.

- layer

Using the layer pattern we can divide the responsibilities of quality to the respective three layers. Each layer handles its own quality management, this promotes loose coupling regarding quality dependencies between subsystems. Improvements can be made to separate layers independently to satisfy the systems quality requirements.

3.2.3 Security requirements

Reasons for Security as a non-functional requirement

- Kpmg offers services in both data analysis and auditing. Both areas deal with highly confidential client information and regard security as top priority. The project management system may not have anything to do with these services, however it is a representation of kpmg's ethics and it must value security regarding client information with the same importance.

- If the project management system is to be hosted from the same physical servers as other company systems, it must have no area for exploitation. Security requirements ensures this.

- confidential information regarding projects should have no opportunities to be exploited to the public via malicious attacks on the system, security requirements protect the system from letting this happen.

Strategies to achieve this non-functional requirement

- There is a hierarchy of users for the application. The range includes administrators, directors, project managers and further employees below these roles. Each user has different responsibilities and authority over the project which gives them project management options to govern the project that are not available to other users. Access to options available to users with a certain role should only be available to these users, session and verification management of the system will ensure this.

Patterns to achieve these strategies

- Chain of responsibility

Using the chain of responsibility pattern, we can create a chain of handlers for our requests and responses. The client can send a request to the server which either handles the request or sends it to the database manager whereby the response is then sent back up the chain. This allows us to have greater control over the security of the system as a whole by carefully checking and handling each request at different stages before it can access any information from the database.

3.2.4 Interface requirements

Reasons for Interface as a non-functional requirement

- Interface requirements cover a broad spectrum including the usability of the system as well as its hardware and software interfaces and their respective capabilities.
- The usability of the system defines the user experience well using the system. To elicit a positive experience, the system must take into account and deploy many usability guidelines. KPMG wish to update their current project management system with a new design due to the lack of usability effectiveness of the current system.

Strategies to achieve this non-functional requirement

- The new system's front should be developed with usability as its key target. its interface should have a modern and intuitive design which brings about a positive experience to the user using it.
- All subsystems should communicate with one another through http Get and Post requests.

Patterns to achieve these strategies

- layered pattern

4 Use Case Diagrams

4.1 Assigning Projects

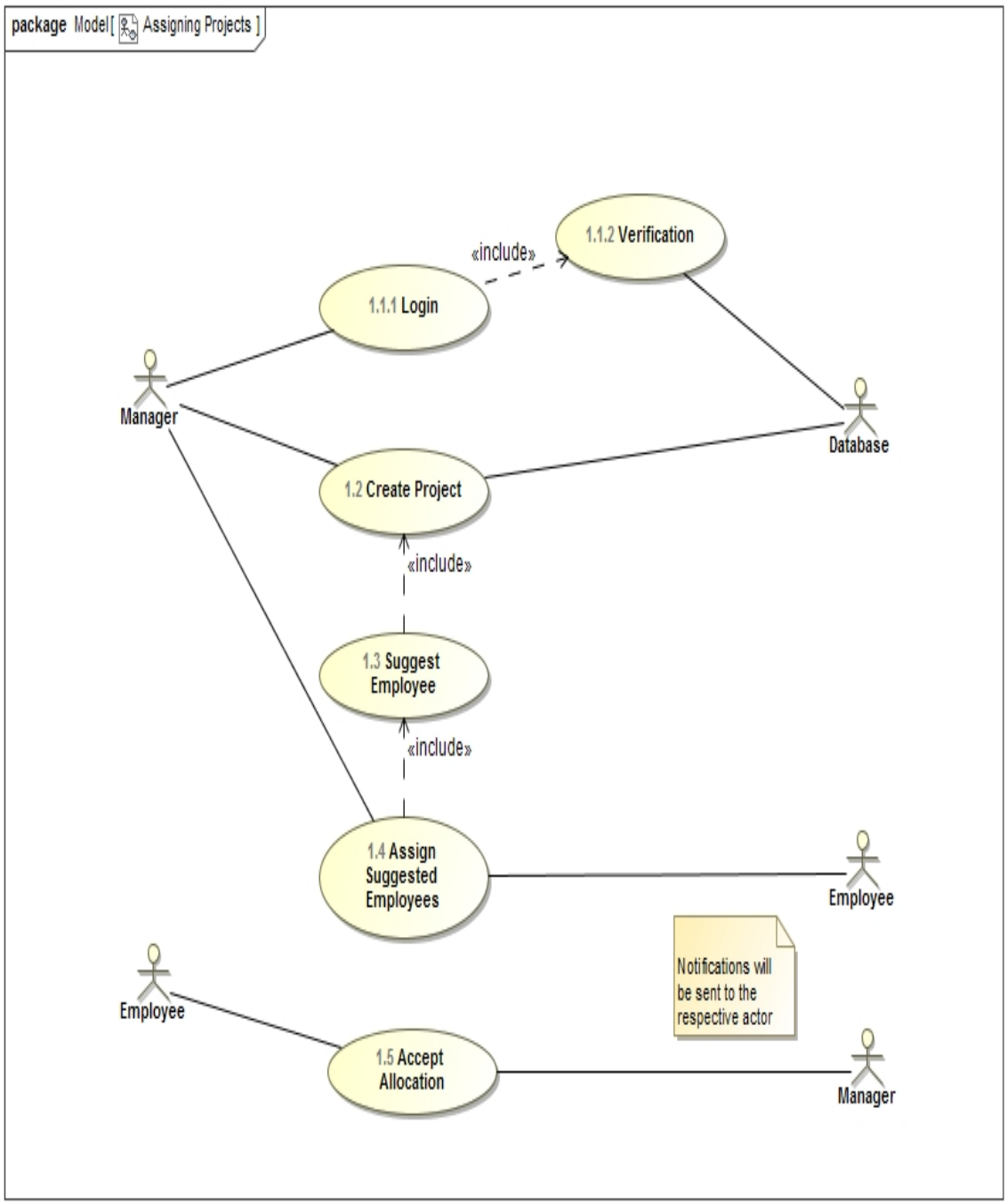

Figure 2: Use Case: Assigning Projects

4.2 Managing Projects

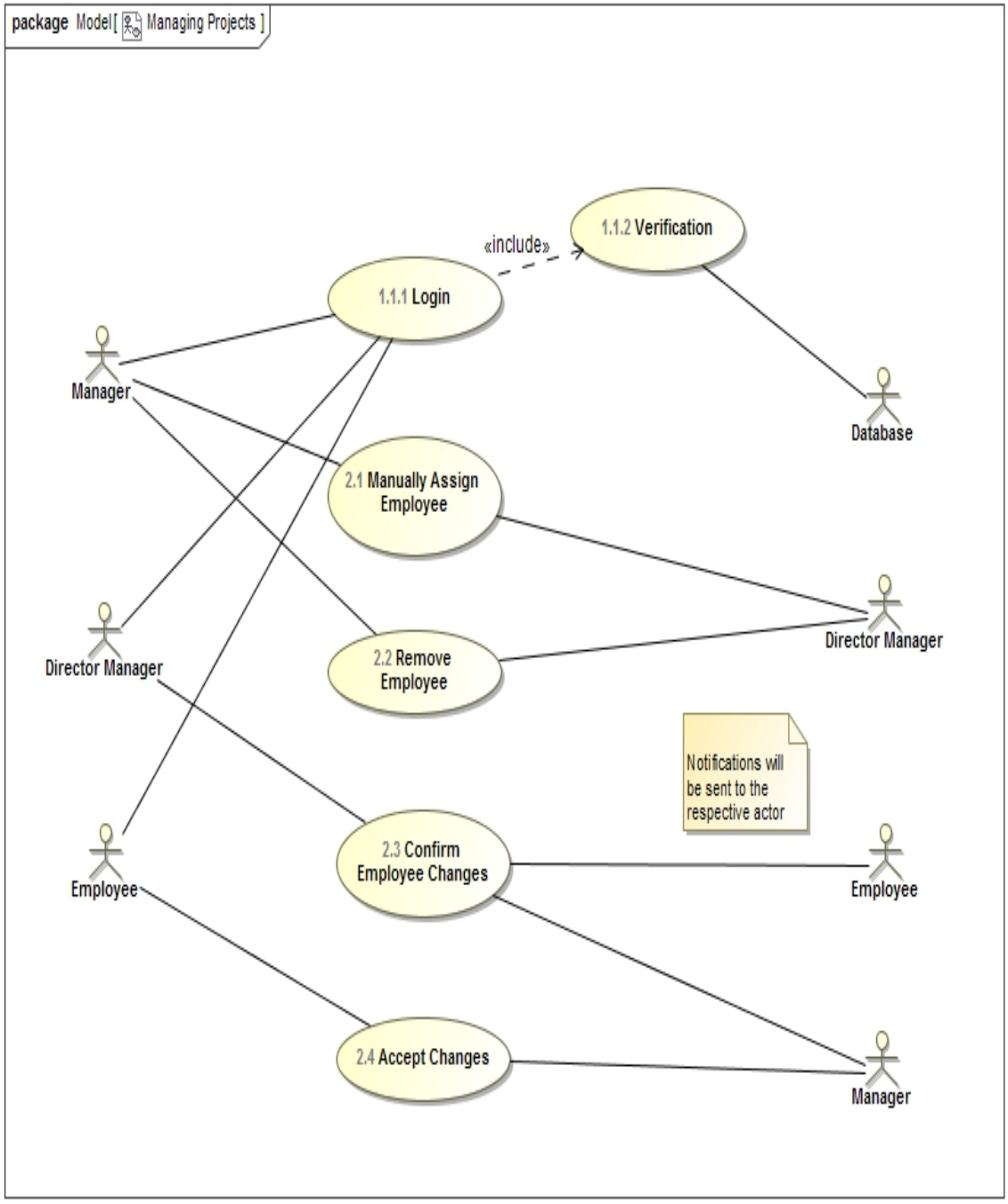

Figure 3: Use Case: Managing Projects

4.3 View Projects

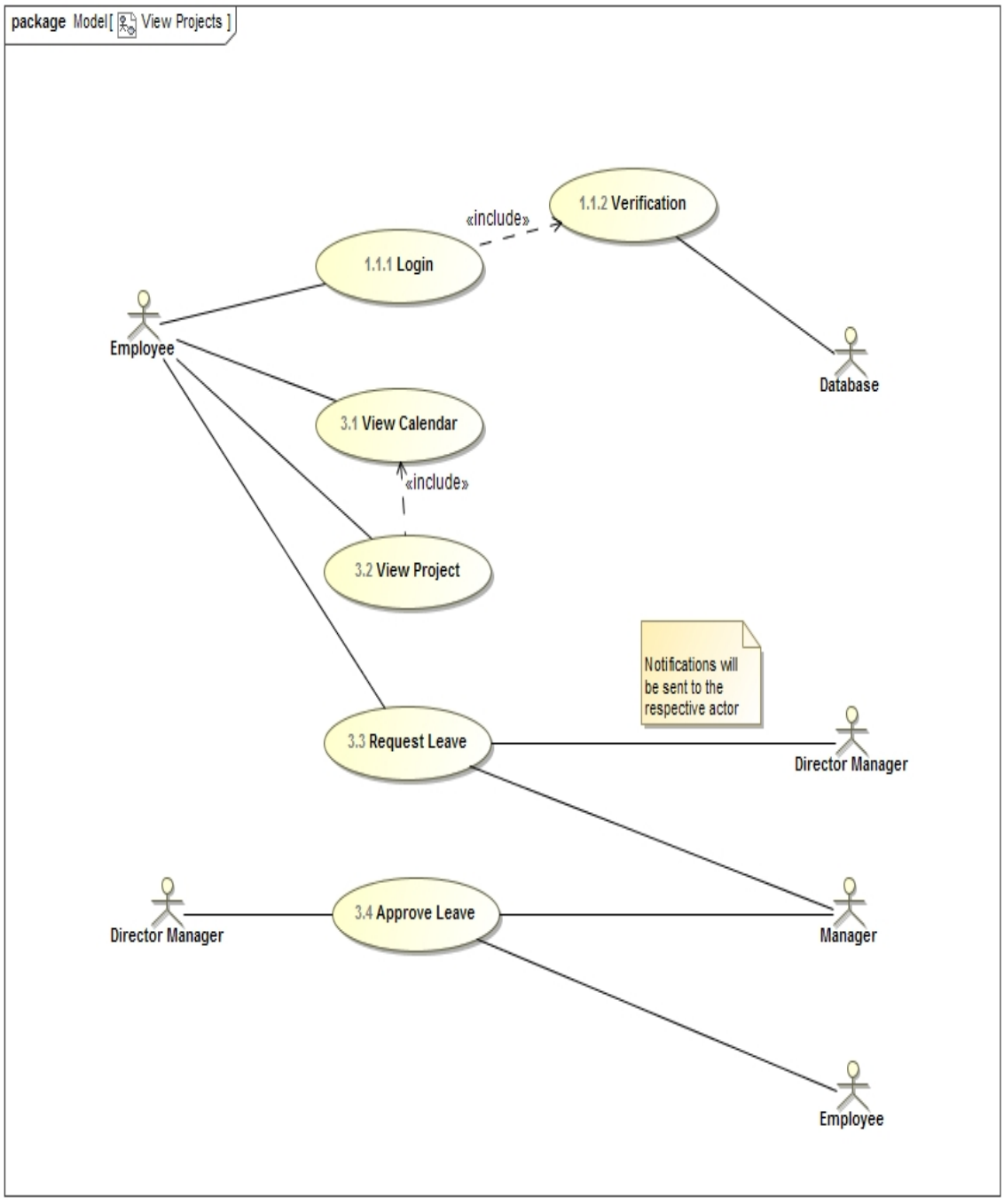

Figure 4: Use Case: View Projects

4.4 Training

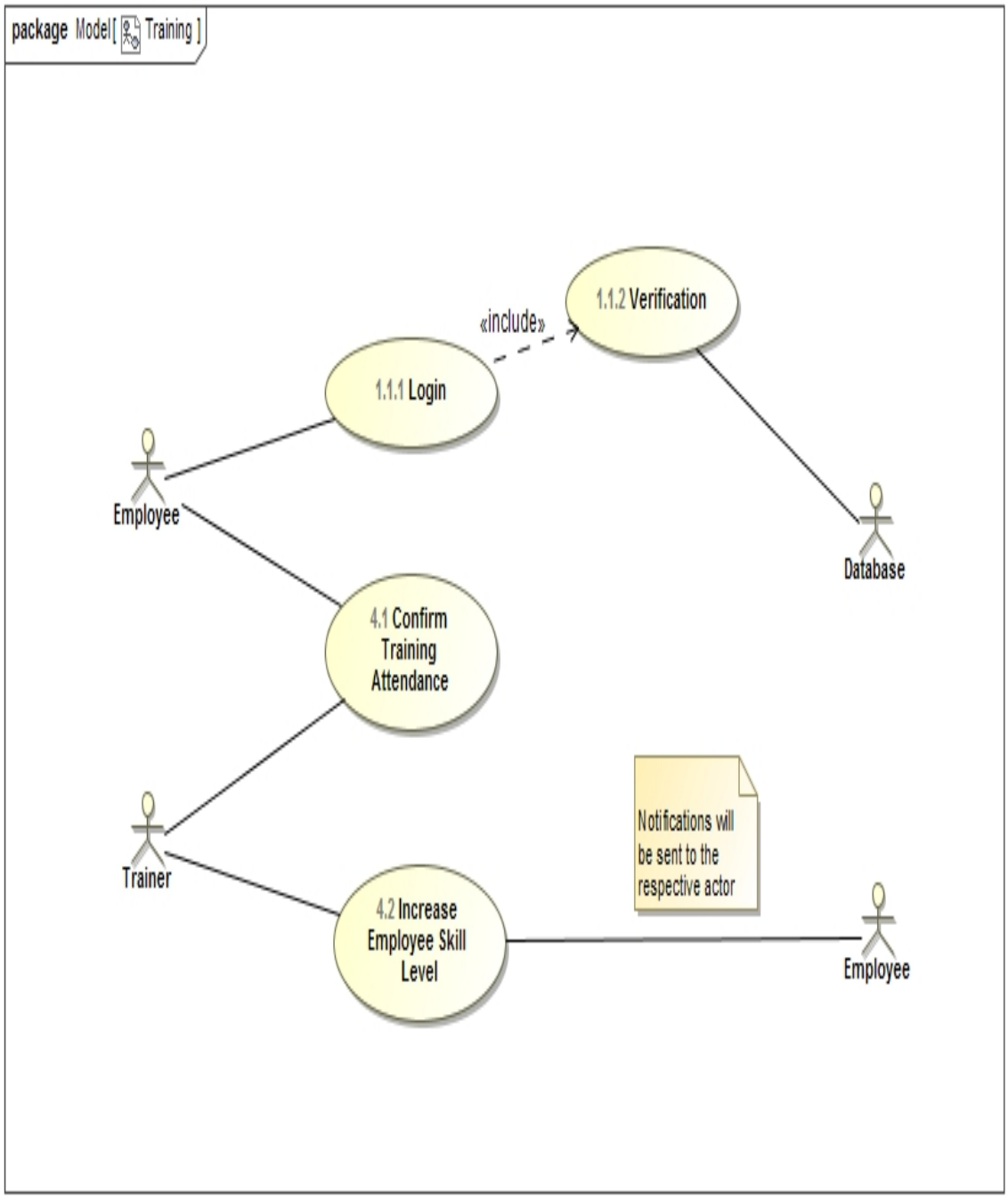

Figure 5: Use Case: Training

5 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	