
CONSULTANT TRACKER ARCHITECTURAL DESIGN

EDITED BY

SIBEKEZELO MAMBA 16095414

JOHAN DE WAAL 16155140

STEPHEN MUNRO 16024479

HULISANI MUDIMELI 16073364

NGONIDZASHE MUJURU 16285256

TATENDA MAFUNGA 16094965

University of Pretoria

2018

Contents

1	Introduction	1
1.1	Purpose	1
2	Overall Description	2
2.1	Product Perspective	2
2.1.1	System Interface	2
2.1.2	User Interfaces	2
2.1.3	Hardware Interface	3
2.1.4	Communication Interfaces	3
3	Specific Requirements	4
3.1	External interface requirements	4
3.2	Functional Requirements	4
3.2.1	Login	4
3.2.2	Create Project	4
3.2.3	View Project	4
3.2.4	View and rate team members	4
3.2.5	Feedback	4
3.2.6	Notification subsystem	5
3.3	Non-functional Requirements	5
3.3.1	Performance	5
3.3.2	Security	5
3.3.3	Software Quality Assurance	5
3.4	Assumptions and Dependencies	6
4	Use cases	7
4.1	Create project	7
4.2	Create consultant	8
4.3	Notification subsystem	9

1 Introduction

The Consultant Tracker application is a Human-Resource system. The application will facilitate easy tracking of resource utilisation and also comparisons between allocated and actual resources spent on each project.

1.1 Purpose

The purpose of this document is to provide a detailed overview of the software product and its goals. It describes the user interface, hardware and software requirements.

2 Overall Description

2.1 Product Perspective

The system will run as both as mobile and desktop web application. Both versions will have the same functionalities, the mobile web application is useful for quick on the go use cases such as viewing progress and feedback.

2.1.1 System Interface

The system will make use of a web-based Front-end that sends and receives data using Java Persistence API(JPA) to and from a hosted server. The sever will have access to external database from which it can retrieve or store data.

2.1.2 User Interfaces

The interface should be easy to use and intuitive. Shown below is our prototype interface.

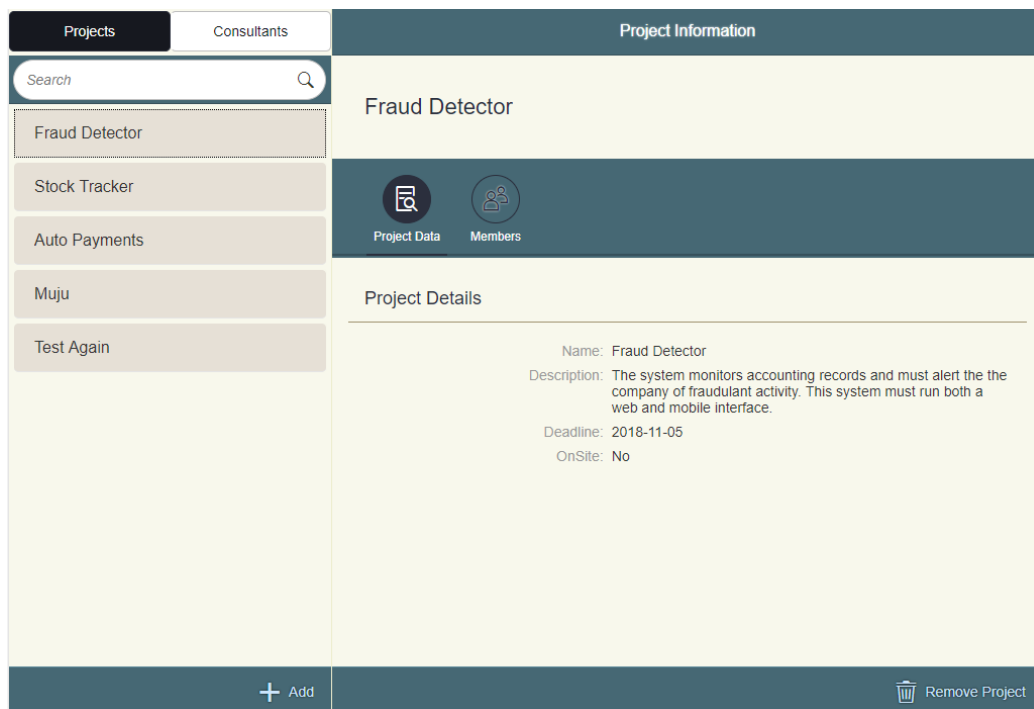


Figure 1: View Project

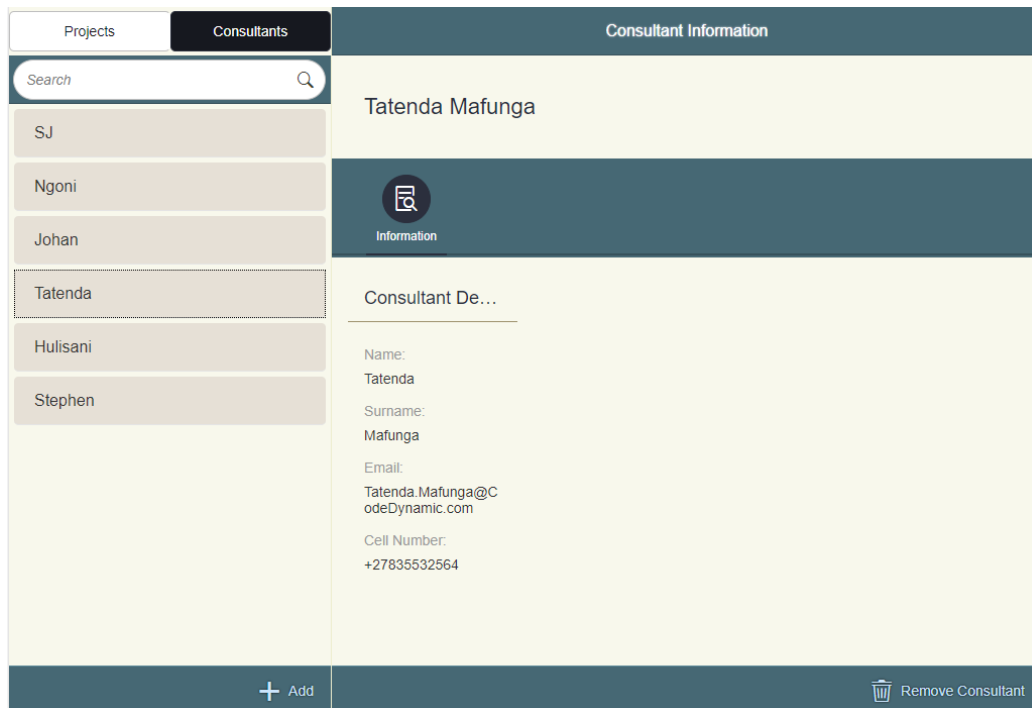


Figure 2: View Consultant

2.1.3 Hardware Interface

- As the Front-end is web-based, it can be run on any device that has an internet connection and can browse the web.
- The server will be hosted on a computer which will be maintained to ensure that the server is always running or a company that provide hosting services can be used.
- The database that will be used to store the data for the application can make use of a computer that is maintained to host it or use a company that provide hosting services.

2.1.4 Communication Interfaces

The system requires HTTP to communicate with server. The system can be configured to be accessed via any available port. The system is accessible through all popular web browsers that interact with HTML pages.

3 Specific Requirements

3.1 External interface requirements

The system will use MYSQL database and JDBC connector to communicate with the database.

3.2 Functional Requirements

The actions/functions depend on the users' permissions. The users of the consultant tracker system are divided into two groups, namely, an administrator and a consultant. The administrator group can perform all actions in the consultant group but not vice versa. Functions include:

3.2.1 Login

This allows the user to enter into the application. The user is required to provide username and password. After authentication, a user will have access to the main menu. The availability of specific functions will depend on the users' permissions.

3.2.2 Create Project

Allows an administrator user to log new projects and information related to that specific project.

3.2.3 View Project

Allows an administrator user to view all projects and the respective details currently in progress. A consultant user will only be able to view projects they have been assigned to.

3.2.4 View and rate team members

Allows a consultant to view team members that he or she will be working with. At the end of a project, a consultant must be able to rate the performance of the team members. Ratings will remain anonymous, but only visible to an administrator user.

3.2.5 Feedback

Allows a consultant user to log any issues encountered and also allows the administrator user to view and respond accordingly.

3.2.6 Notification subsystem

The consultant must be notified by the system when they are assigned to a new project, as well as be notified when the allocated time is nearly reached.

3.3 Non-functional Requirements

The non-functional requirements will specify the limitations of the system.

3.3.1 Performance

The system is expected to be responsive even under a high load (multiple users making requests).

3.3.2 Security

The system must be able to differentiate between the types of users. From this, the system will be able to grant specific functions to users, thus protecting the data from possible corruption caused by unauthorized actions by users.

3.3.3 Software Quality Assurance

- **Reliability**

The system shall perform all its stated functions consistently. Computations of utilisation and any date related functions will be accurate.

- **Robustness**

The system will be designed to work on any device which has a browser. The system must be data efficient so that it can work even though there are poor internet connection speeds. The presentation layer will be created knowing that the system will run on different devices, with different processing capabilities, that it should be kept as power efficient as possible to support all the devices the system can run on.

- **Availability**

The system should be always available during the hours it is most popular. Any maintenance where the system needs to be offline should be done outside these identified times. The geographical location of the server must also not affect the availability of the system.

- **Efficiency**

The system will be data efficient so that it does not waste any users data.

- **Portability**

A responsive JavaScript framework will be used for making the application so that it can run on different platforms using one codebase.

- **Maintainability**

The code should be built modularly, such that independent parts can function independently of one another. Common coding styles should be utilised.

3.4 Assumptions and Dependencies

It is assumed that the application may be used on mobile phones. Although the product can be used on desktops, the product will be designed in a way that best suits mobile users needs.

4 Use cases

4.1 Create project

This is a main function of the administrator. The administrator must

- enter project details which also include client details.
- assign consultants to a project.

The use case diagram is given below.

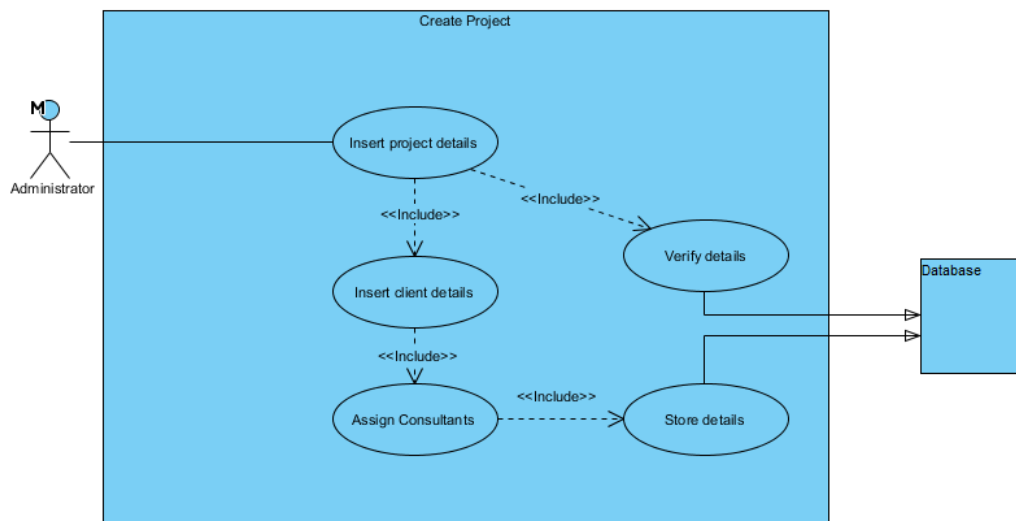


Figure 3: Create project use case diagram

The following sequence diagram shows the interaction between an administrator and the system interface.

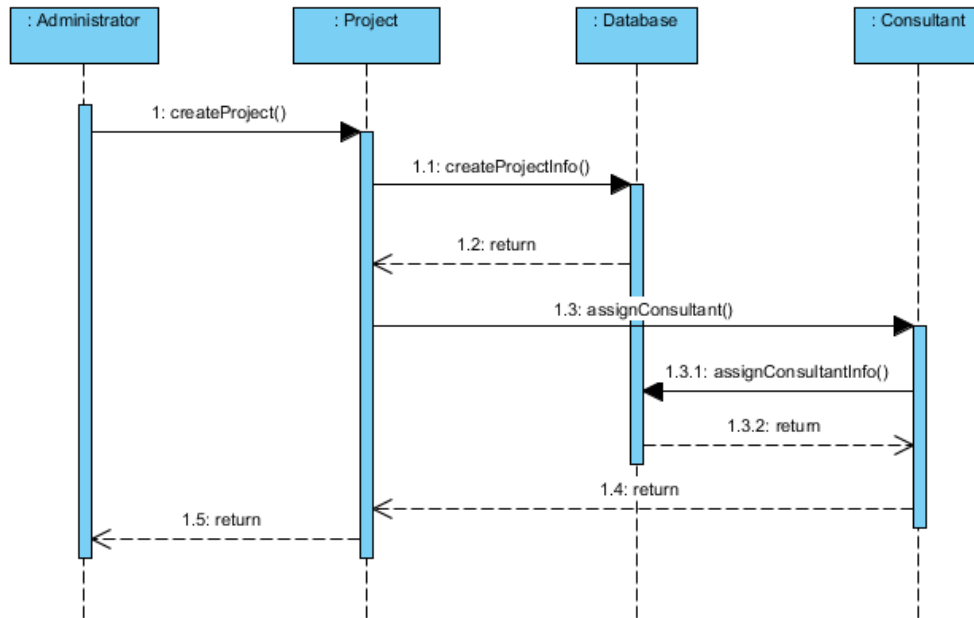


Figure 4: Create project sequence diagram

4.2 Create consultant

The administrator must be able to create a consultant account. The following image shows the use case diagram.

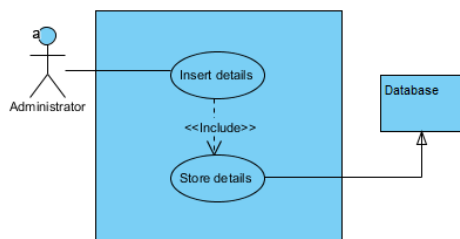


Figure 5: Create consultant use case diagram

4.3 Notification subsystem

The system must notify a consultant once they are assigned to a project. The following image shows the use case diagram.

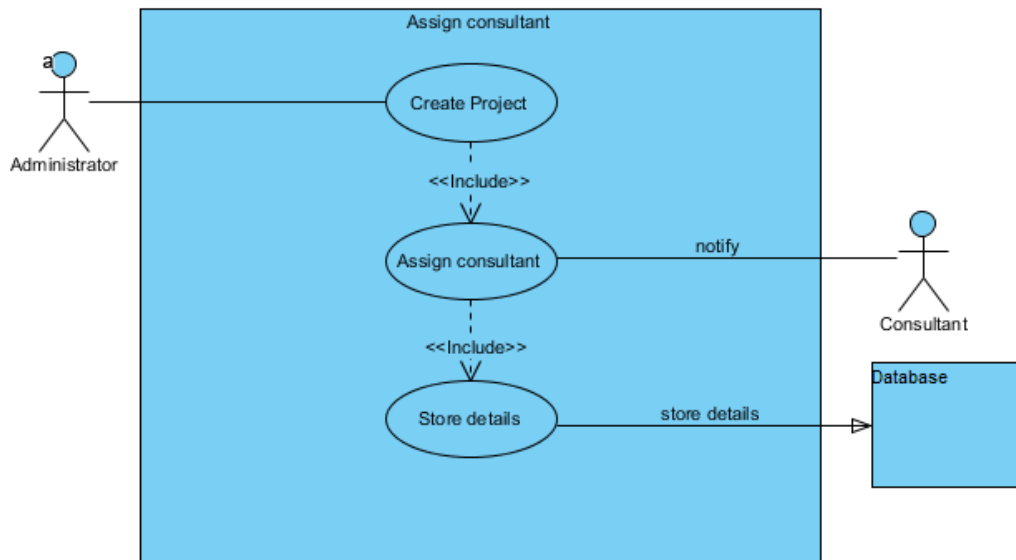


Figure 6: Notification use case diagram