

**Diploma in Computing (with strands in Software Development & Computer Networks)**

**Level 7**

**(Software Development Strand)**

**DC304 Object-Oriented Analysis and Design**

|  |  |
| --- | --- |
| Liana Pak | yib000013o |
| Omid Ardakani | Yib00001kq |
| Sviatlana Belabarodava | yib00001yl |

**Table of Content**

[Introduction and overview of the project 3](#_Toc509991838)

[Construction Site Attendance Application 3](#_Toc509991839)

[1.1. Use case diagram 4](#_Toc509991840)

[1.2. Use case specifications 5](#_Toc509991841)

[1.3. Activity diagrams 7](#_Toc509991842)

[2.1. Class diagram with attributes and operations 9](#_Toc509991843)

[2.2. Object diagram 10](#_Toc509991844)

[3.1. Sequence diagrams 11](#_Toc509991845)

[3.2. State machine diagrams 17](#_Toc509991846)

[3.3. Communication diagrams 18](#_Toc509991847)

# Introduction and overview of the project

## Construction Site Attendance Application

Company W is a New Zealand construction organization that is based in Auckland. It constantly has several objects under construction called projects and each of these projects has one or several sites. Employees are daily assigned to the sites depending on their specialization and qualification. Site managers have to track and approve workers’ working hours on a daily basis. After that Project Managers can reject or approve the Timesheet.

The company W was facing a problem of processing all the information described above using paper work. Because of it, multi-steps approval process included:

* Attendance form printing out and filling in
* Approval by Site Manager with sending papers by post to the office
* Final Approval by Project Manager

All these issues lead to postpones in salary payment.

The company W wanted to have an automated system with the following requirements (functional):

* A tool that could handle a multi-level management of attendance
* Kept information about site and project
* Kept information about workers’ personal information and working hours.

During our communication with the customer representatives, our company had found out that W also needs an application that can (non-functional requirements):

* Handle at least 6 projects with 6 sites and 50 workers
* Be available 24/7
* Have a user friendly and simple UI.

For handling that task was chosen a team consisting of 3 people and 9 weeks to finish the project. As the team is not big and the timeframe is short, there was chosen an Incremental Model which is used when requirements are clearly defined and understood, and there’s a need for early withdrawal of the product to the market.

Here is the list of the diagrams that the team was using during the design period:

* Use case diagrams
* Use case specifications
* Activity diagrams
* Class diagram
* Object diagram
* Sequence diagrams
* State machine diagrams
* Communication diagrams

# 1.1. Use case diagram

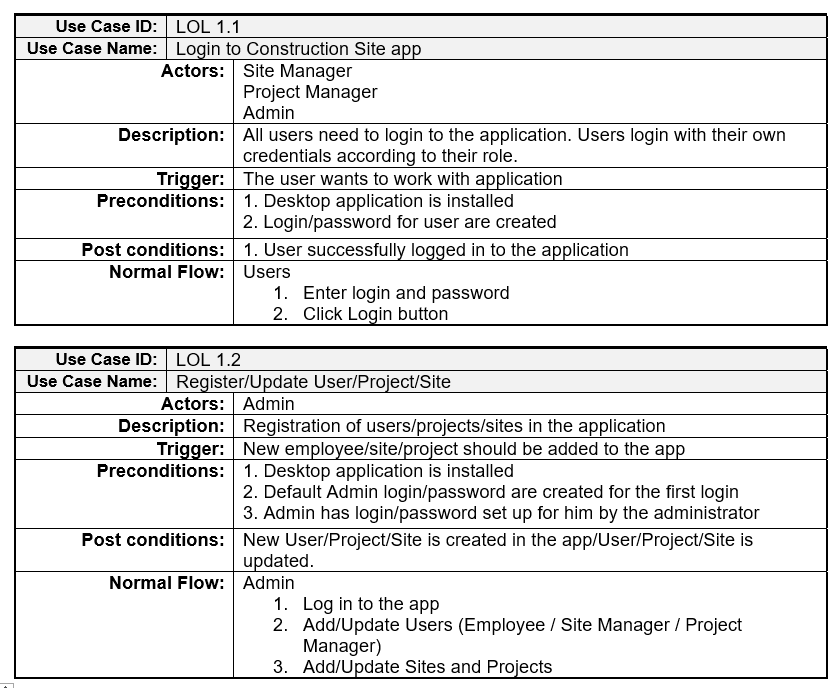
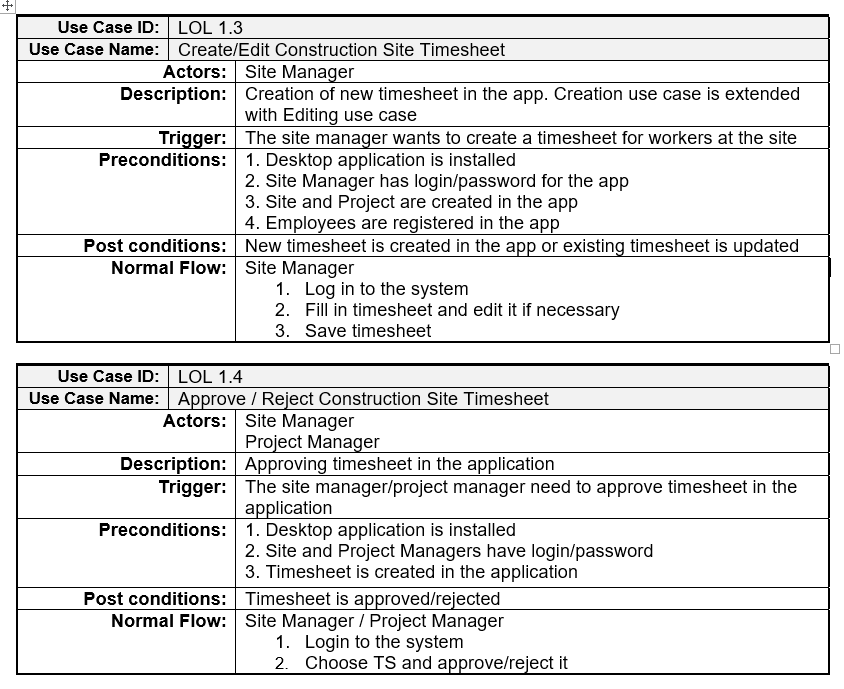
The diagram displays the interactions of users and system

A close up of a map

Description generated with high confidence

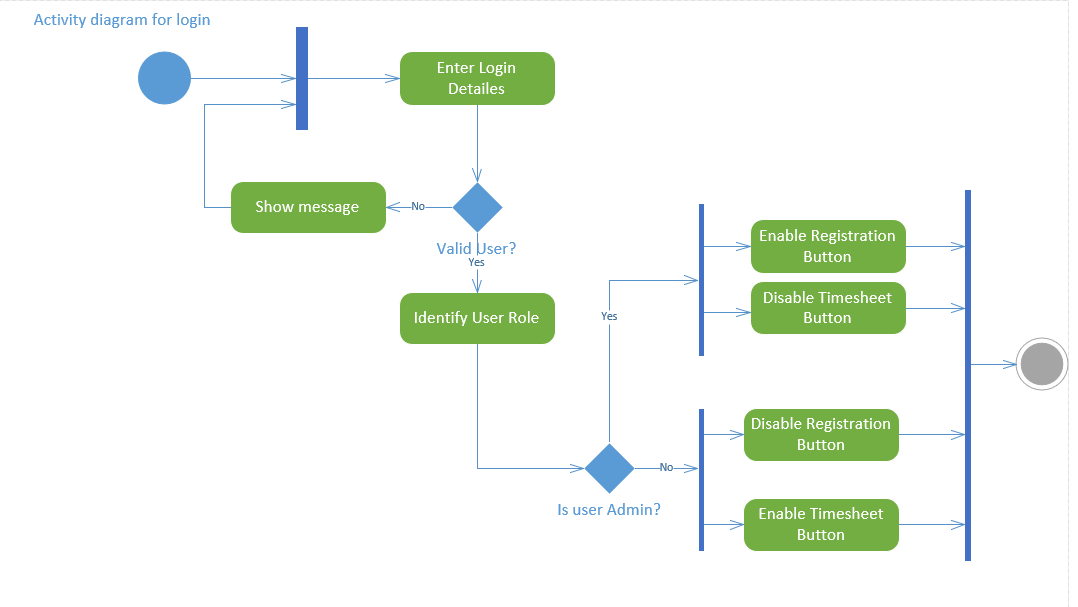
# 1.2. Use case specifications

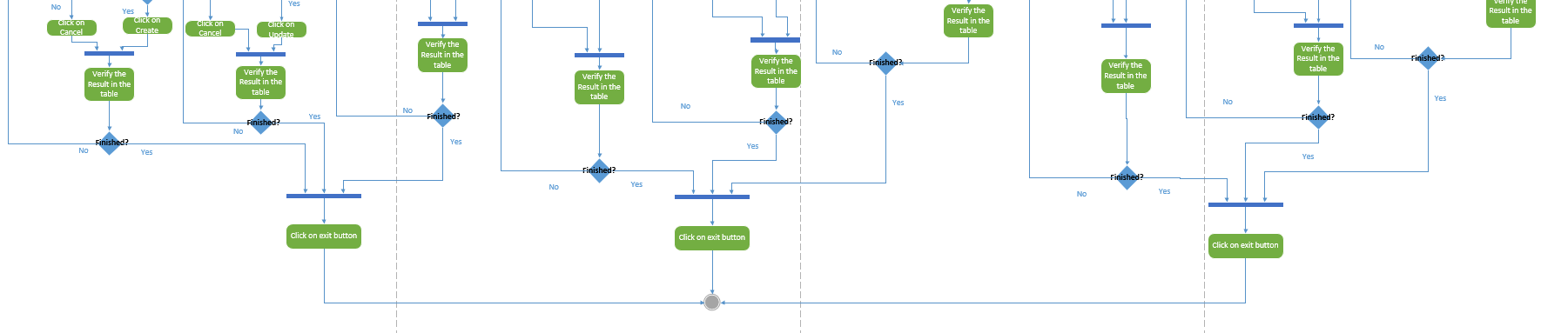
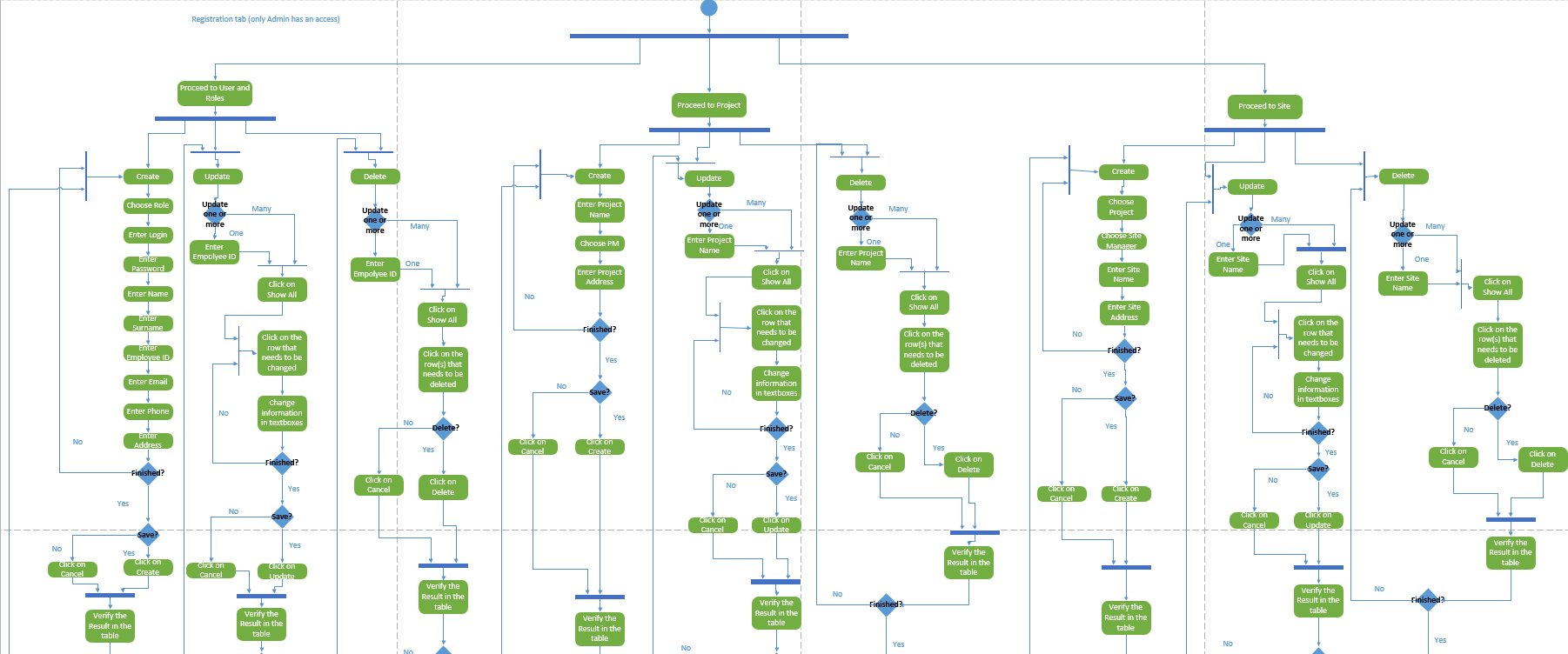
These specifications are formed to capture the users’ fundamental needs within the system. They are formed based on Use case diagram for capturing functional requirements of the system.

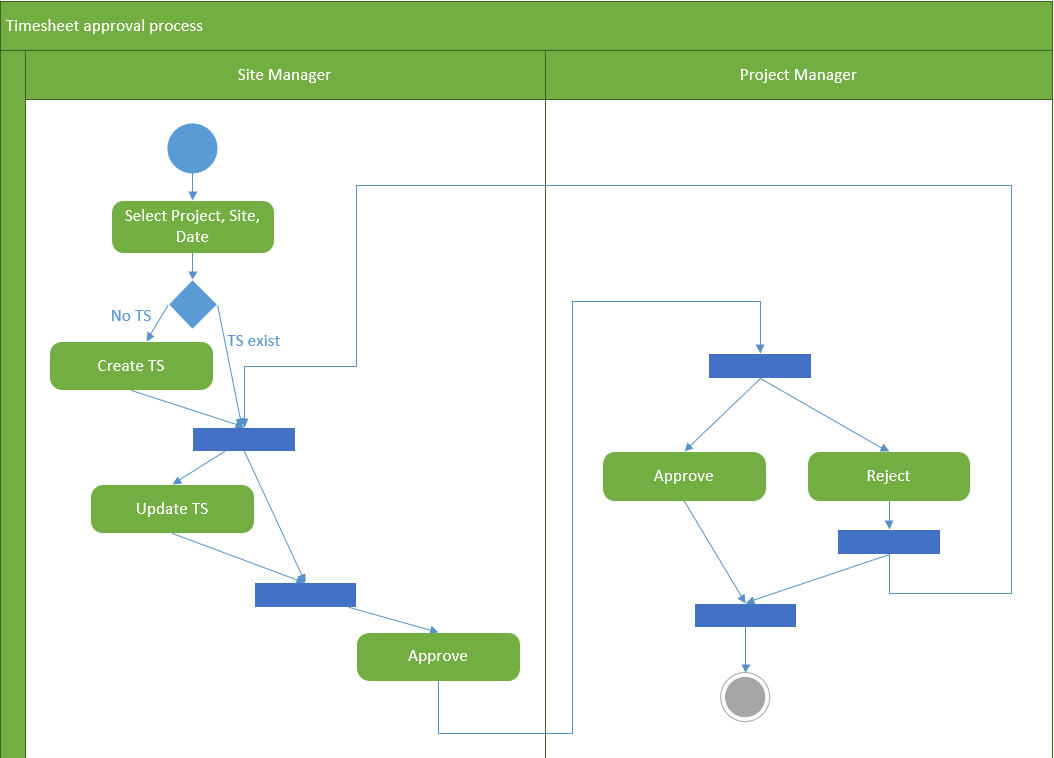


# 1.3. Activity diagrams

These diagrams are showing the dynamic nature of the system by passing messages and showing the user different options to act.

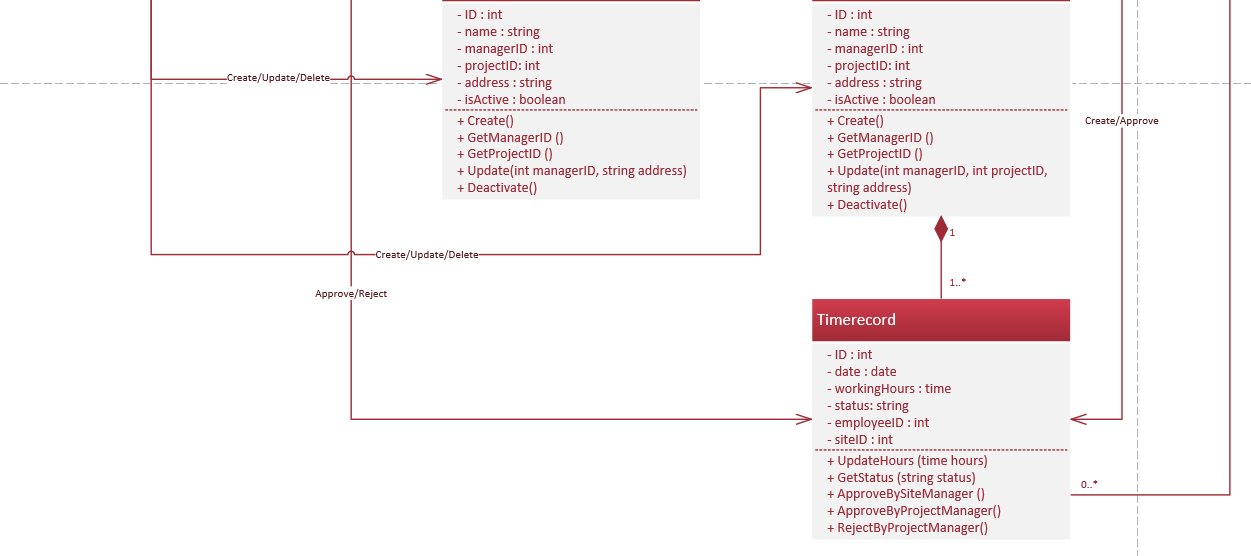
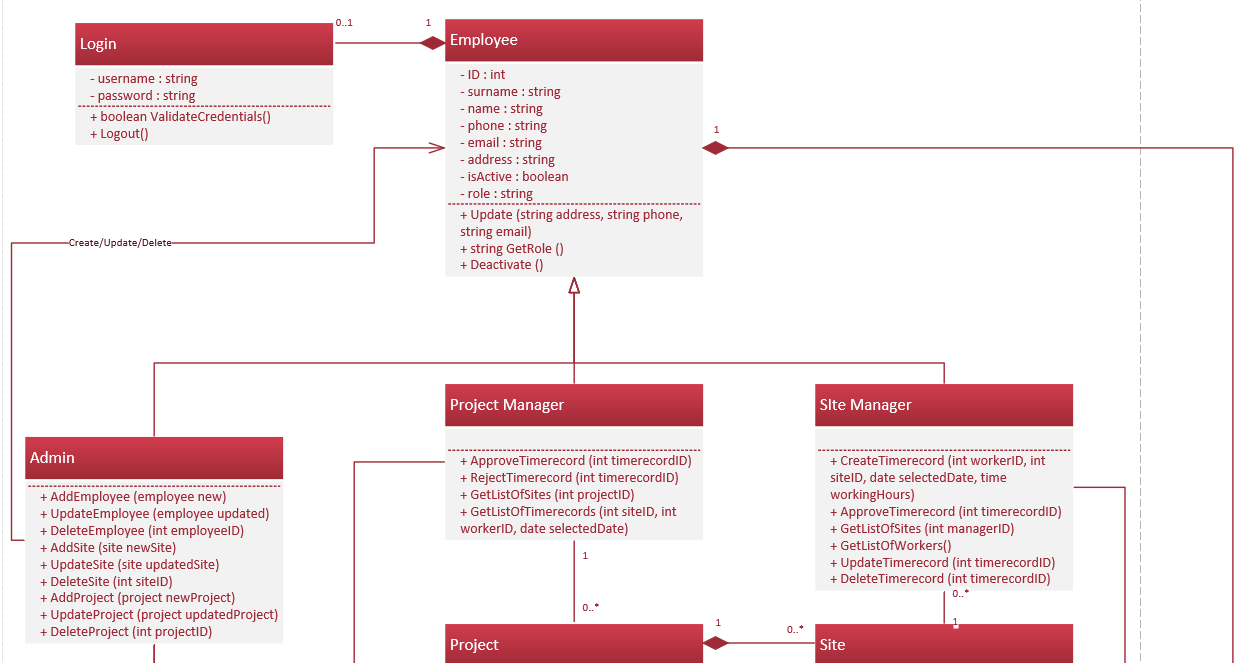






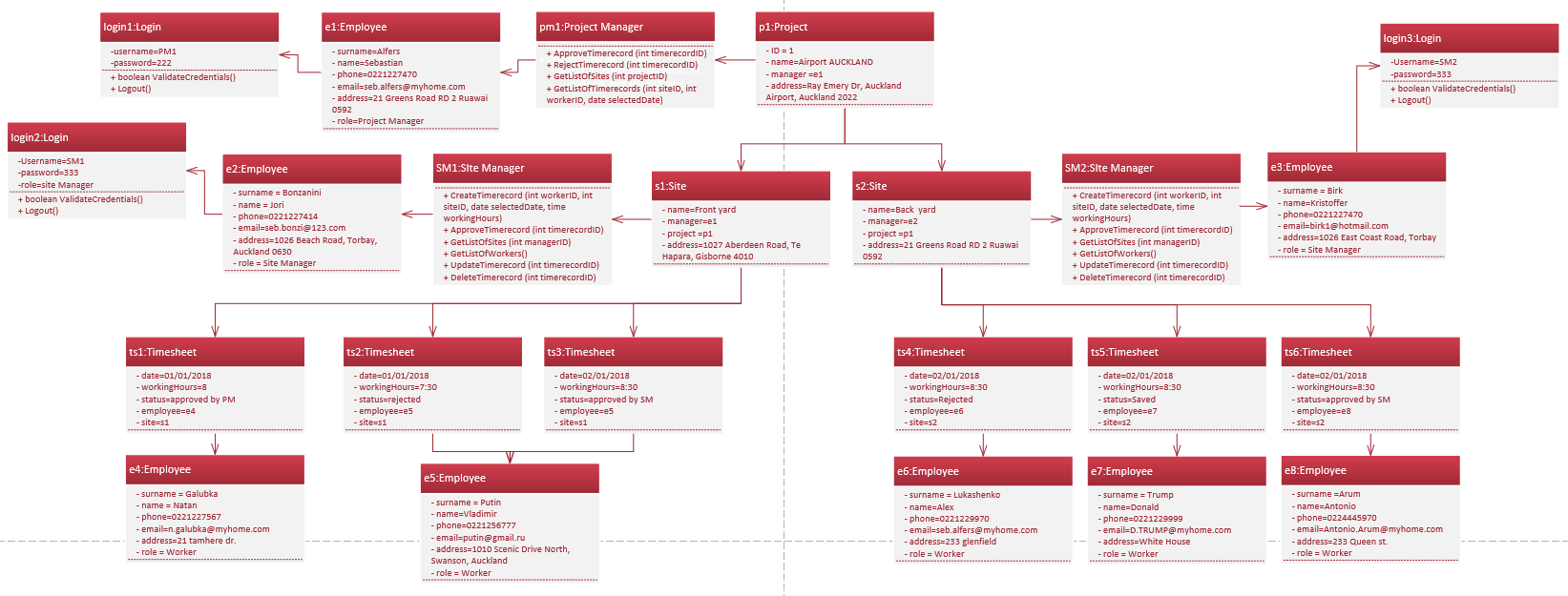
# 2.1. Class diagram with attributes and operations

Class diagram is used for modelling the system for further coding, understanding the classes, attributes, methods and their interconnection.



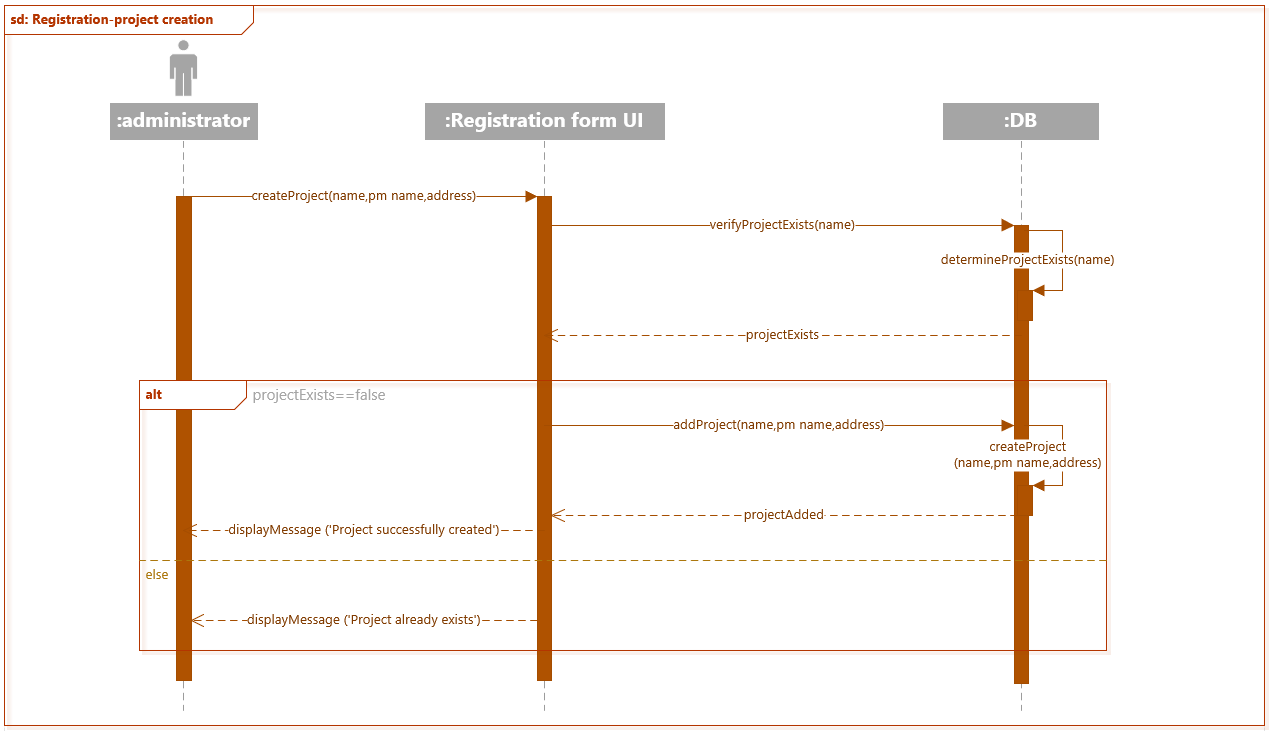
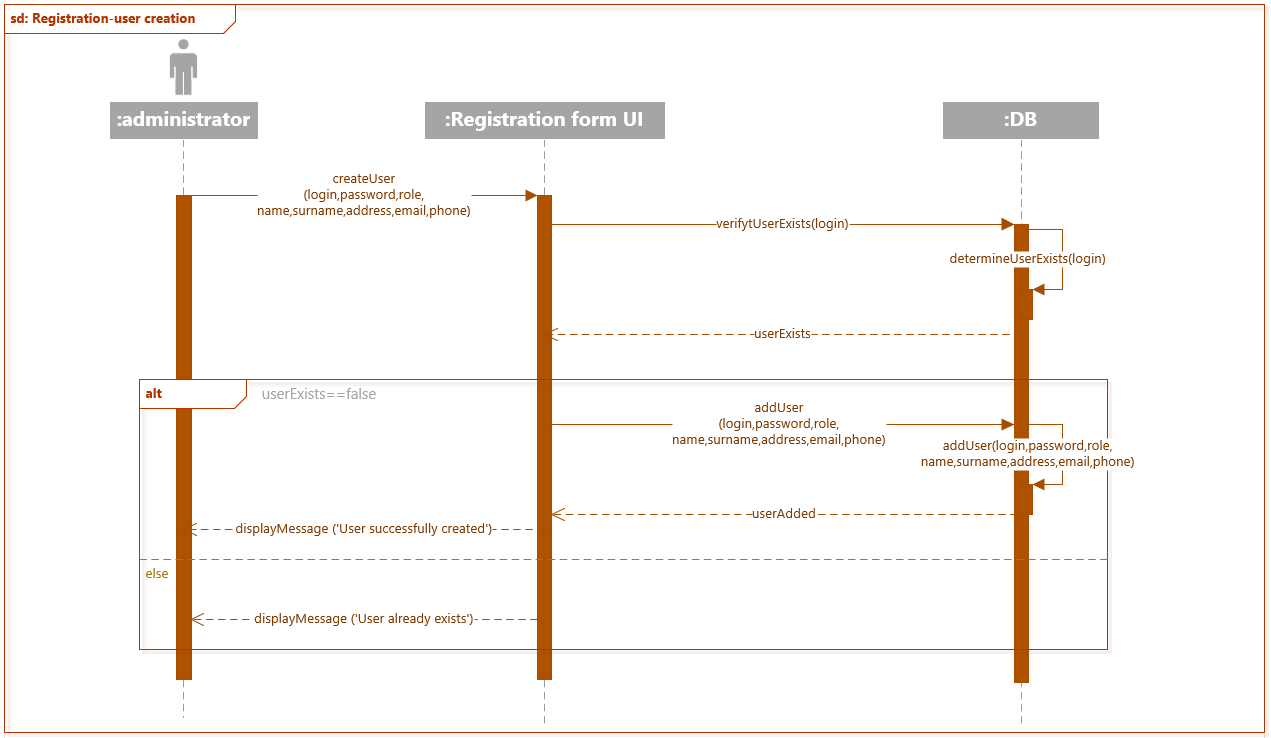
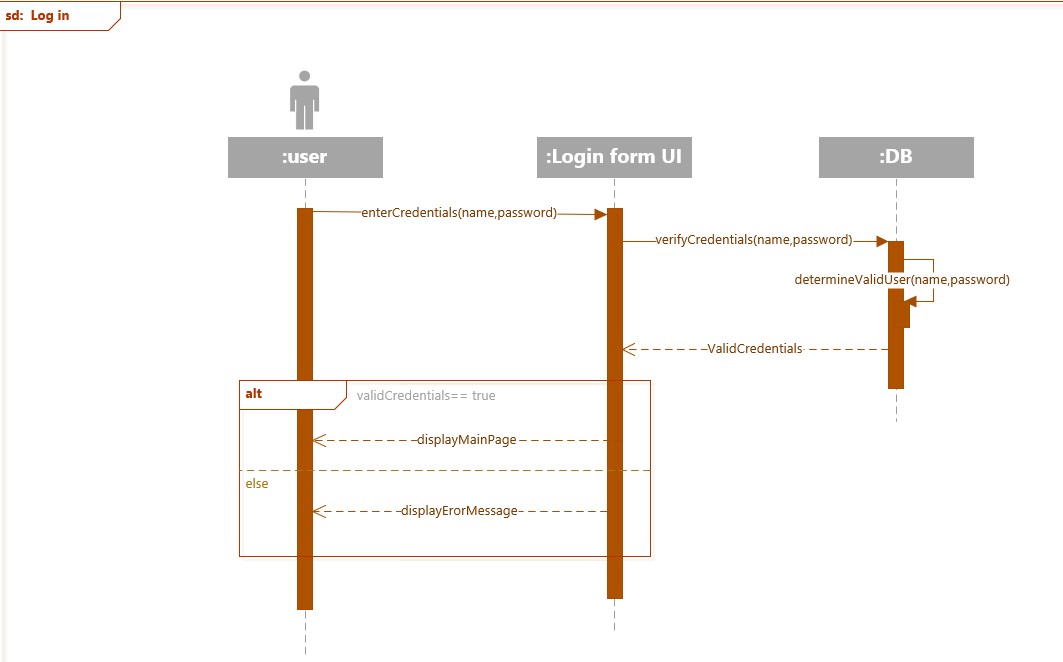
# 2.2. Object diagram

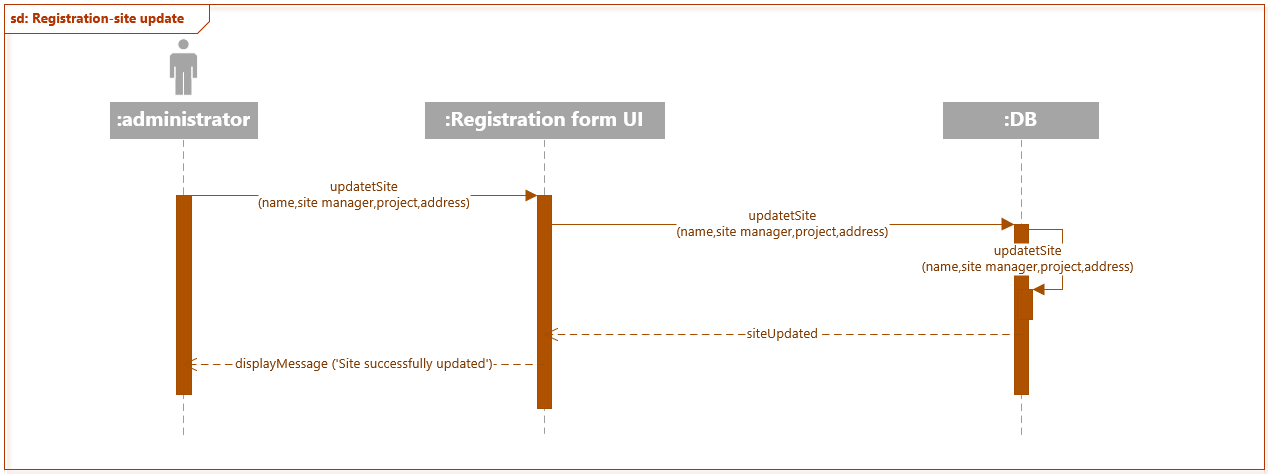
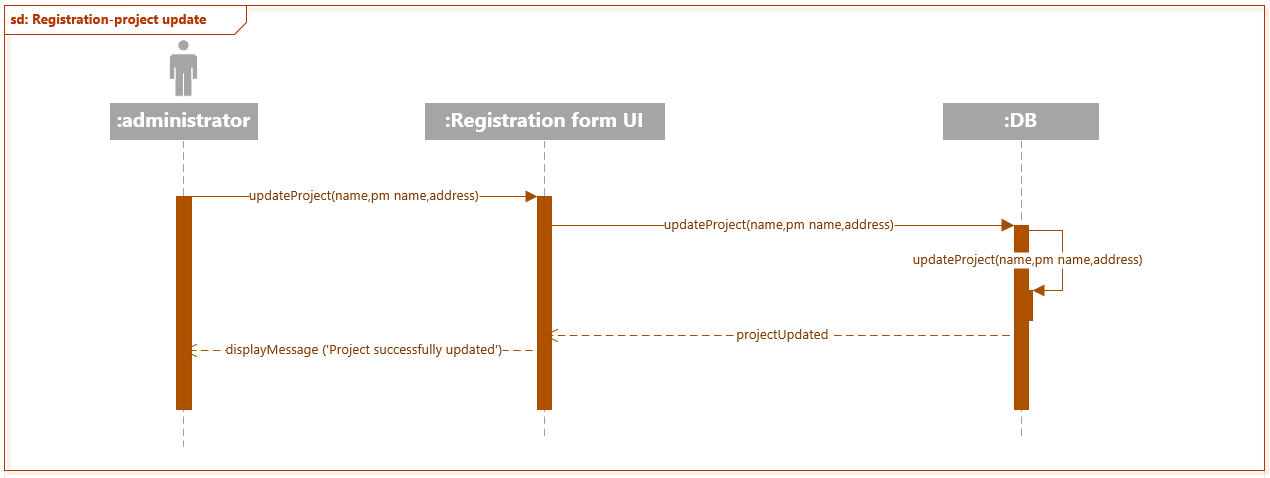
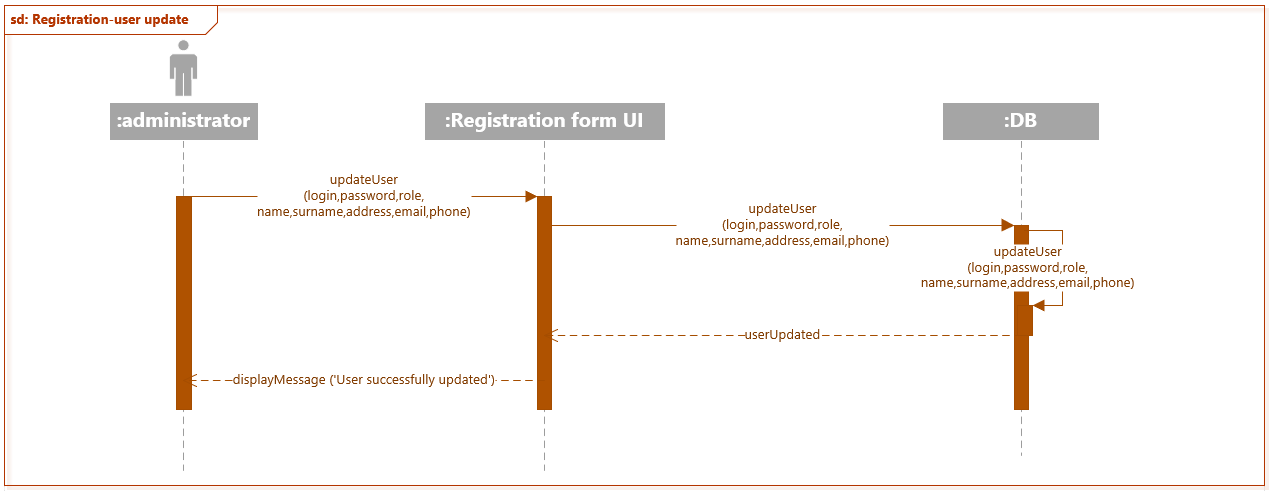
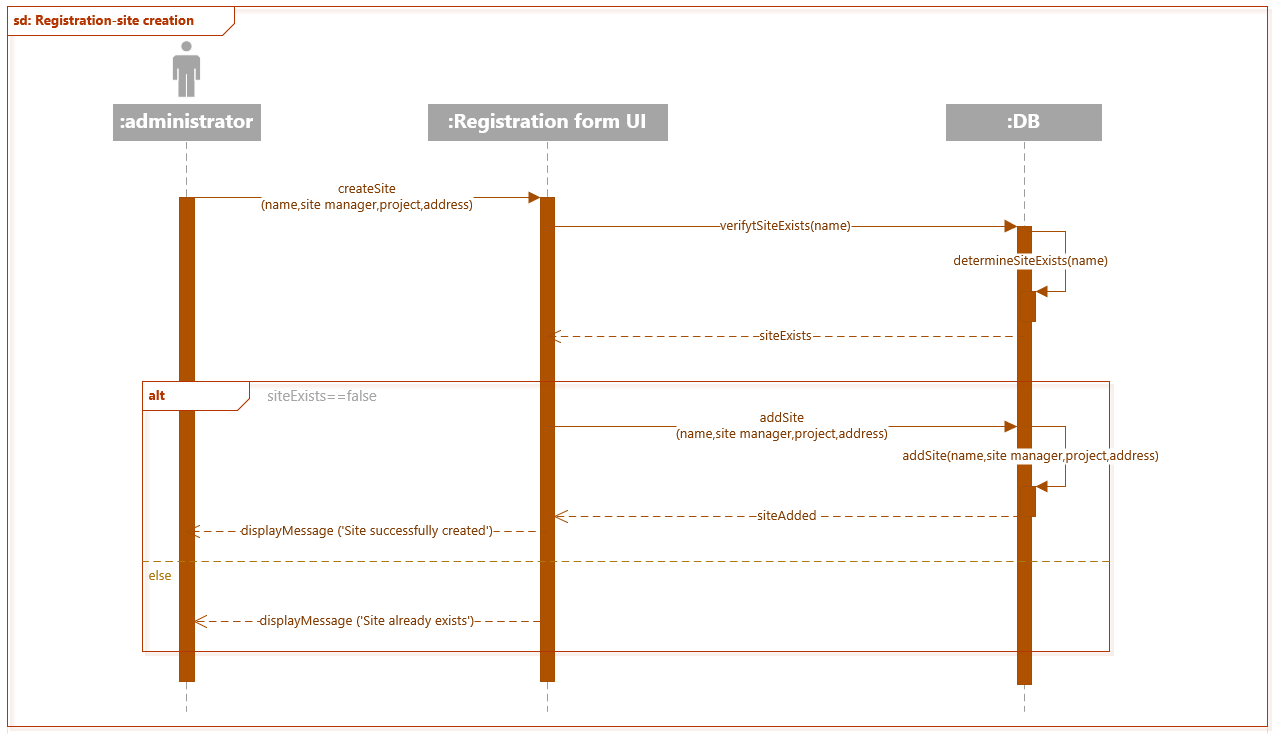
The diagram is used for modelling the system for further coding, understanding the objects and their interconnection.

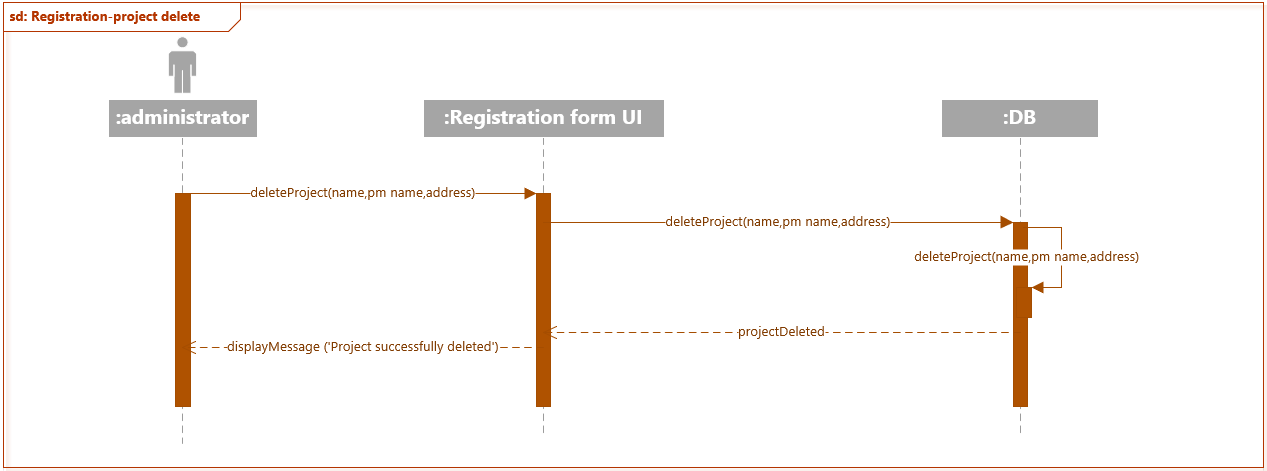
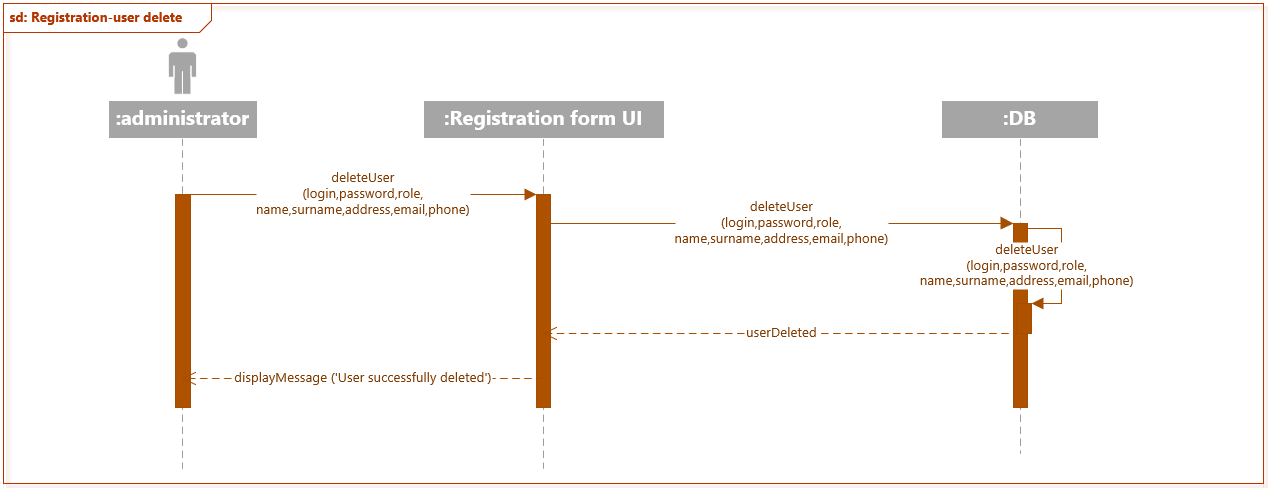


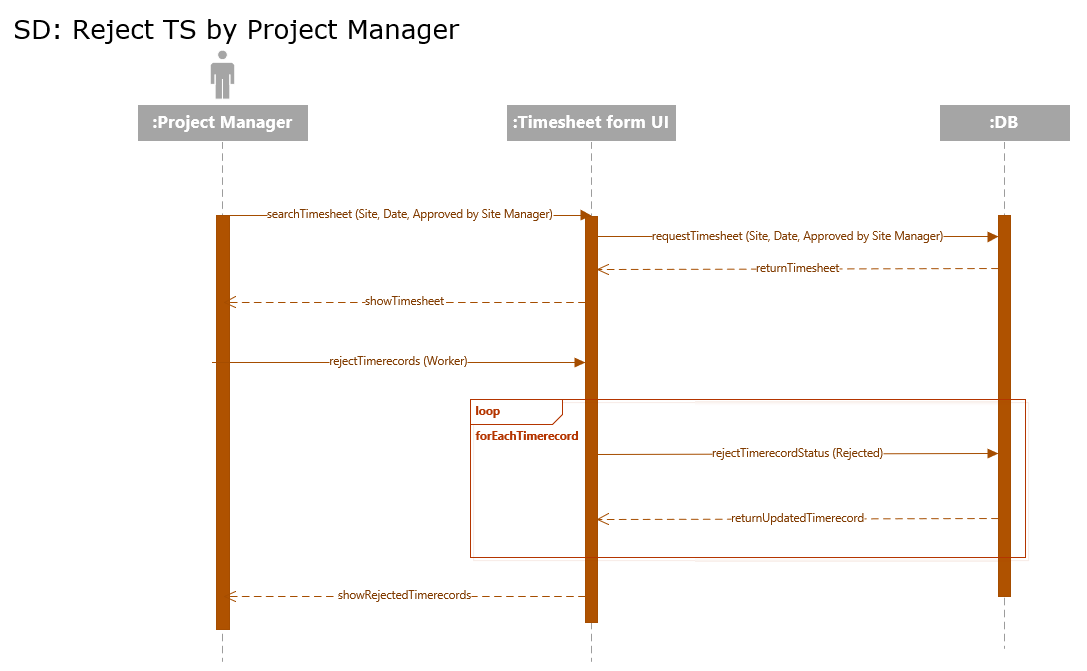
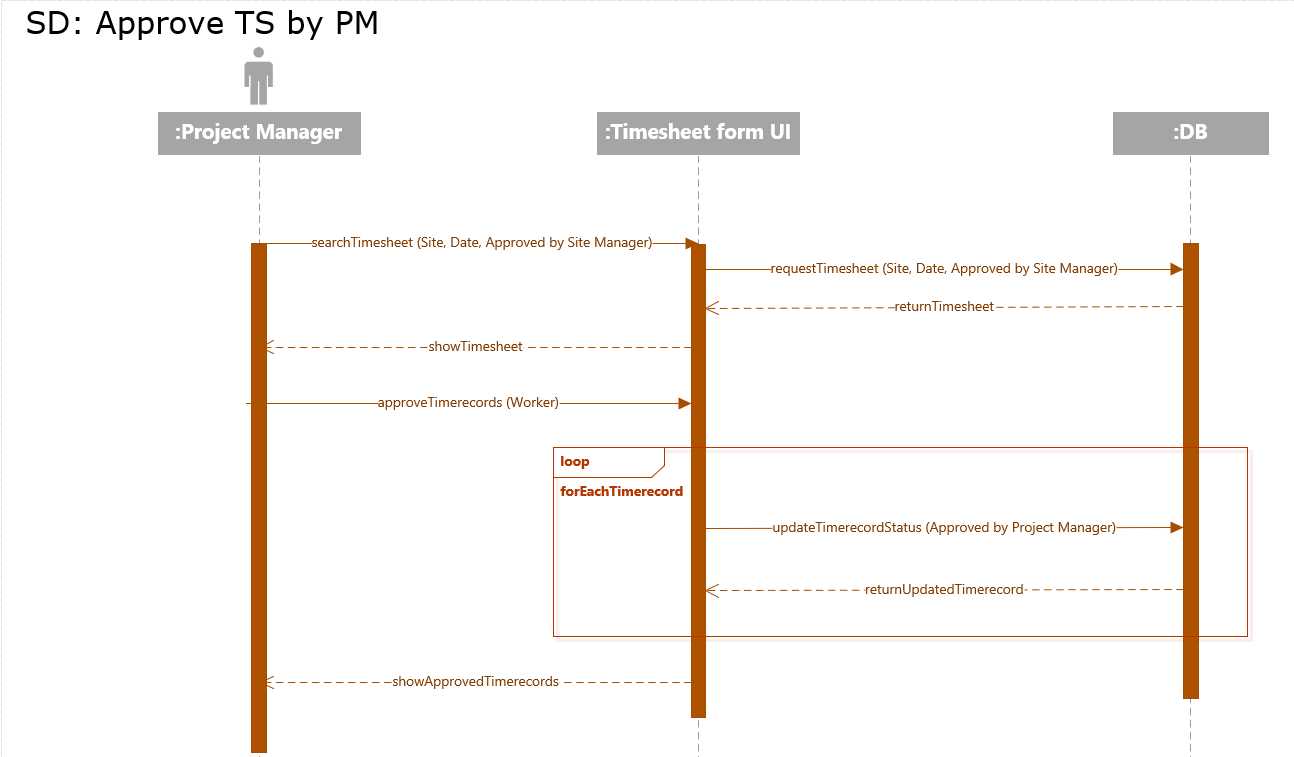
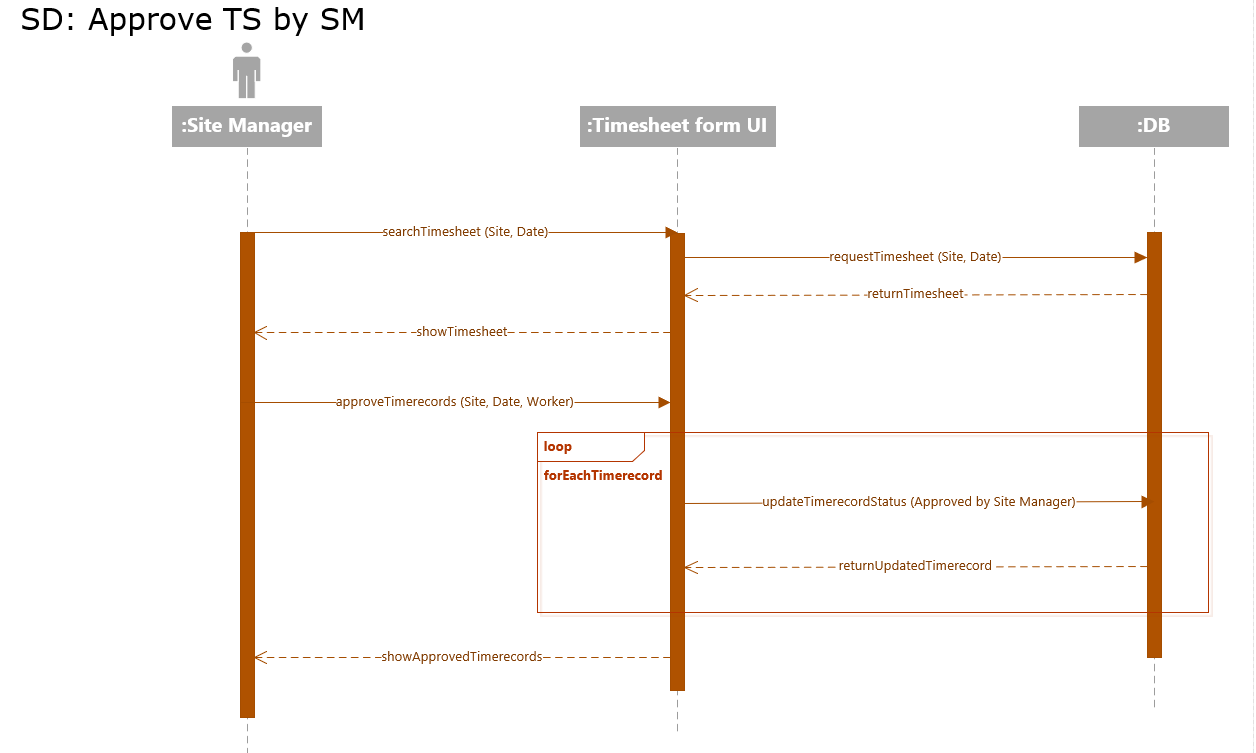
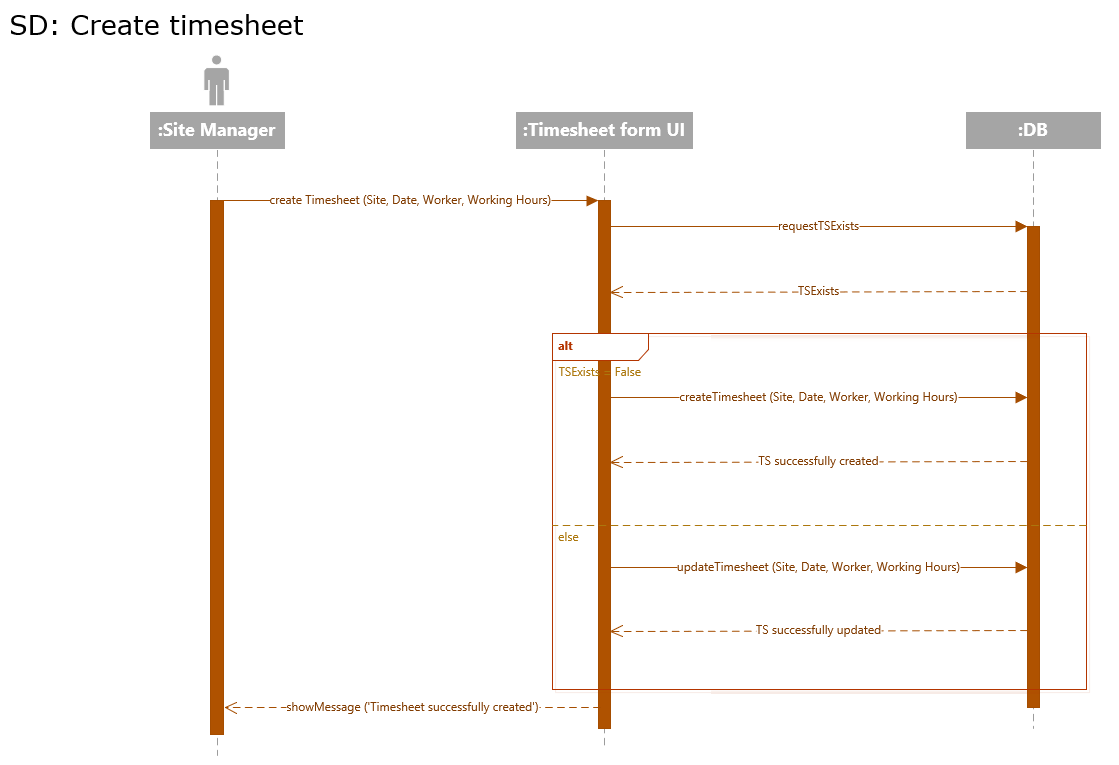
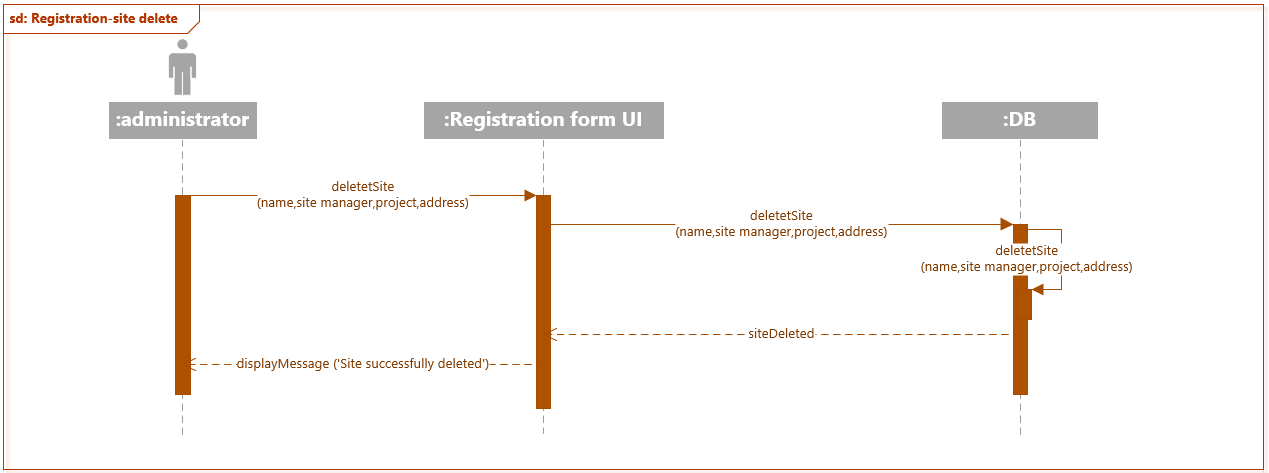
# 3.1. Sequence diagrams

This type of diagram is showing how the objects interact in time sequence.



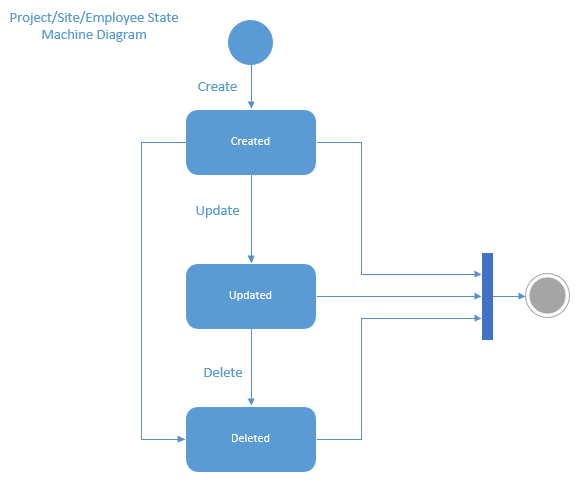
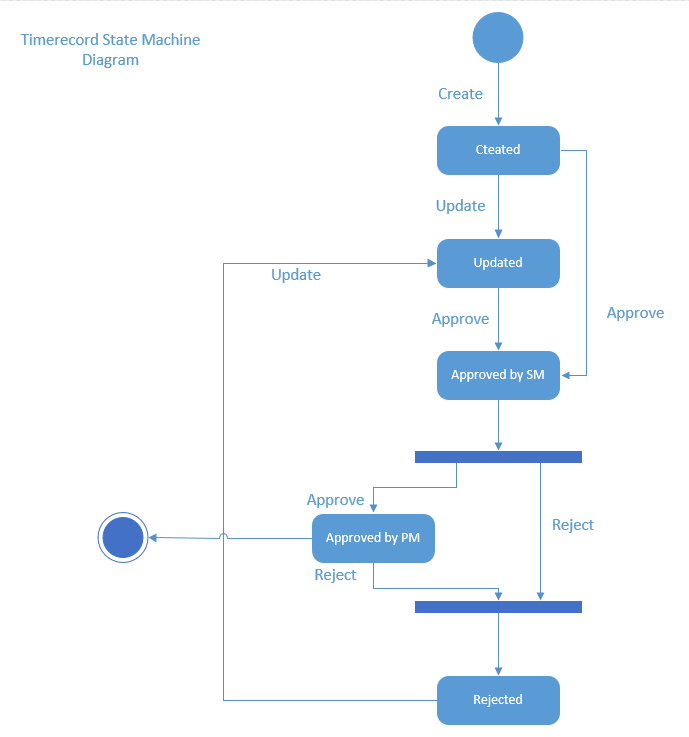






# 3.2. State machine diagrams

The diagrams are showing different variations of events that an object goes through.



# 3.3. Communication diagrams

The diagrams show interactions between objects. It helps to create alternative scenarios of interactions if needed.

