REQUIREMENT ANALYSIS DOCUMENT : UNIPA PROJECT 3 MyCRM

- ## Purpose

This requirement analysis document aims to give information about MyCRM's functional and non-functional requirements, and explains how the system behaves. Stated ideas will be supported using system models such as use-case diagrams, E/R diagrams, class diagrams etc. Ultimate purpose for this document is describing what the software will do and how it will be expected to perform.

- ## Audience

The audience for this Requirement Analysis Document includes my responsibles from both my university, Izmir University of Economics Software Engineering Department and my place of internship, UNIPA. However, anyone else who wants to learn about the program is welcomed.

- ## Introduction

1. **Purpose of the system:** MyCRM is designed and will be implemented as a CRM application for a specific company, with specific requirements. It is not a general use CRM application, however it is possible to adapt MyCRM other companies depending on the requirements.

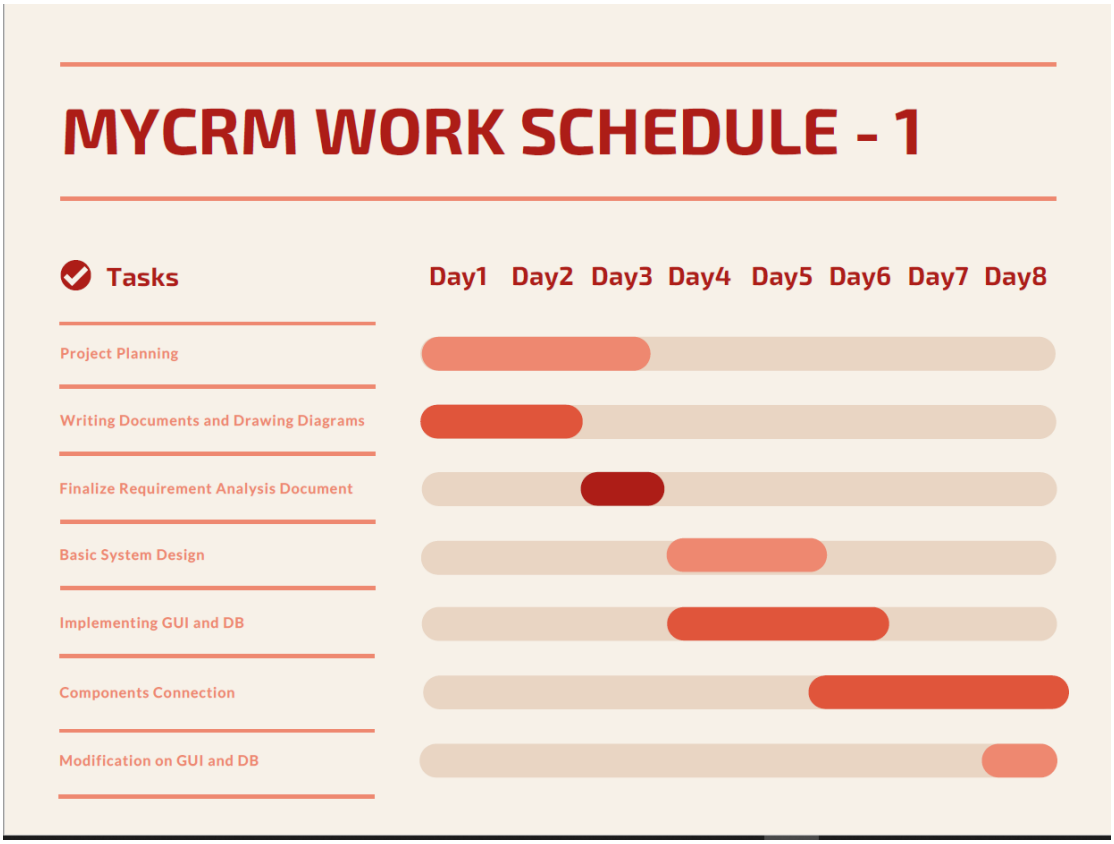
2. **Scope of the system:** This project's scope is limited to implementing the system as it is stated in the requirements list and delivering the necessary documentations. At the end of the project, all requirements will be satisfied and program can be tested and maintained optionally.

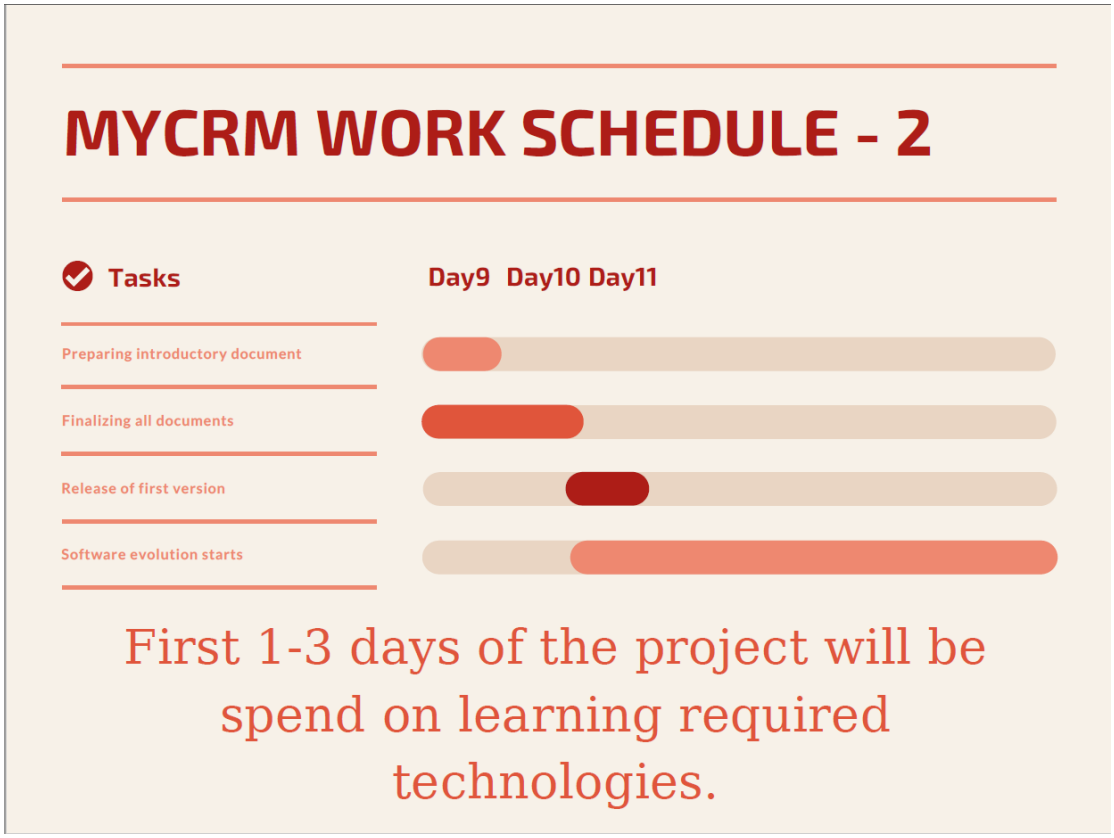
3. **Objectives and success criteria of the project:** For this particular project, delivering corresponding documentations and meeting the system requirements are the keys of the success criteria. Except the functional requirements, my ultimate objective will be delivering a decent and functional CRM application which can be used and adapted for other companies, within the circumstances.

- ## Workflow

For such a limited time to design and implement a CRM application, I picked incremental prototyping as my development method. In incremental prototyping, the final product is decimated into different small prototypes and developed individually. Eventually, the different prototypes are merged into a single product. This method is helpful to reduce the feedback time between the user and the application development team. (https://www.guru99.com/software-engineering-prototyping-model.html)

Below Gantt Chart indicates my work in eleven work days. Probably it will take less than it is shown, but in order to guarantee deliver the project on time, I decided on below dates. Also learning a new technology can take more than it's planned





- **Requirements**

  As it is shown in the below table, functional requirements rely on the needs of students and library keeper. Non-functional requirements of MyCRM System concerns usability, maintainability and performance.

UNIPA PROJECT 3 -MyCRM
REQUIREMENTS LIST

REQ. #	FUNCTIONAL REQUIREMENTS
1	Admin shall add/delete Departments and DepartmentManagers
2	Clients shall be able to add tasks and monitor them (TaskStatus and who's doing)
3	Admins shall be able to change Employee's departments
4	Employees shall add description before a task closure
5	Each Department shall have only one DepartmentManager
6	Clients shall be able to list tasks historically

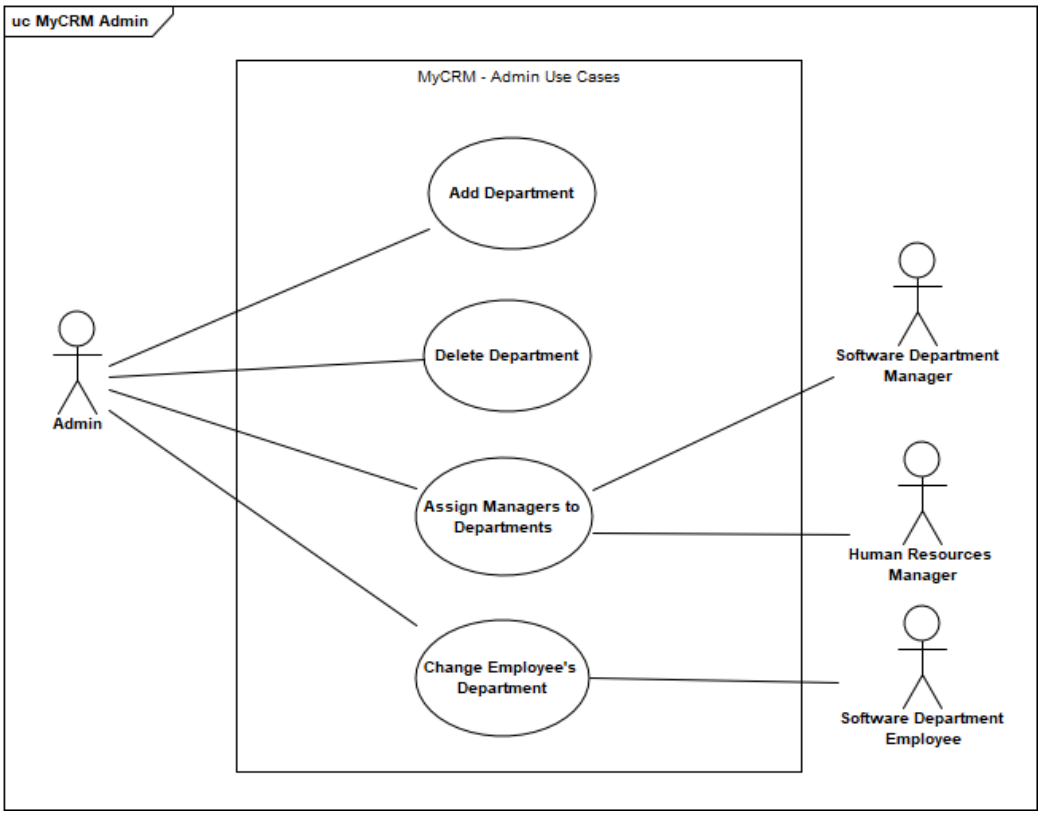
REQ. #	NON-FUNCTIONAL REQUIREMENTS
1	Program will be implemented using C# Programming Language
2	ASP.Net Core MVC will be used as the Development Framework
3	MSSQL will be used as relational database management system
4	Program will be uploaded into GitHub for version control
5	Visual Studio will be used as the Integrated Development Kit
6	(OPTIONAL) Nunit or MsTest will be used as Unit Test Framework

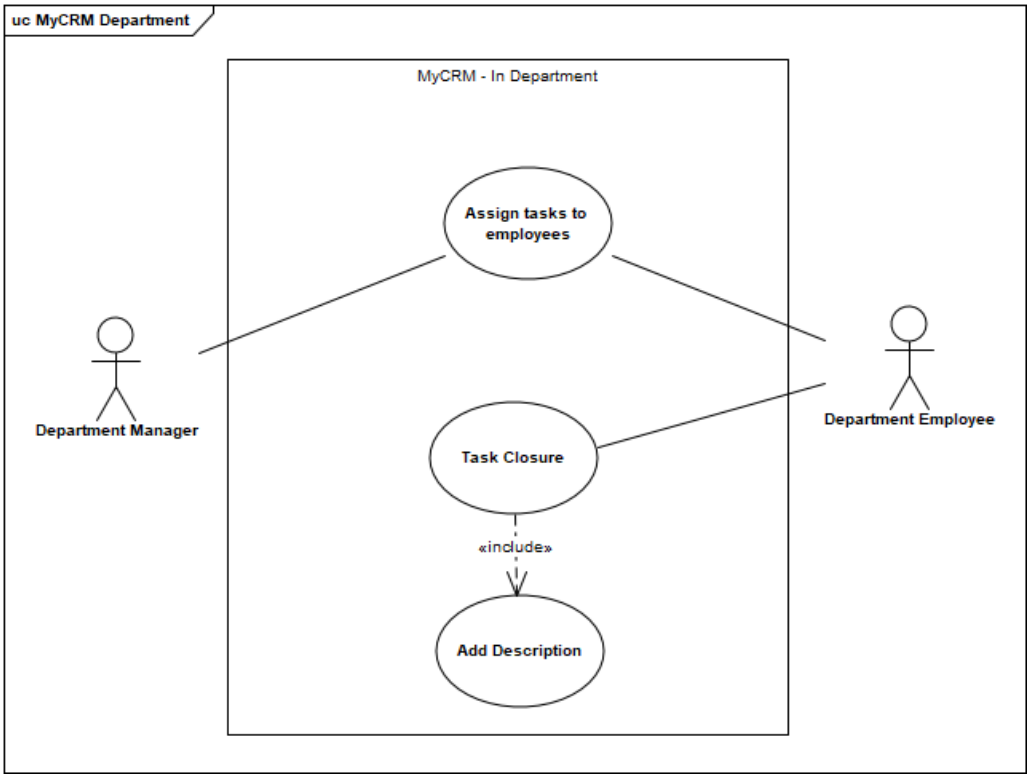
- ## System Models

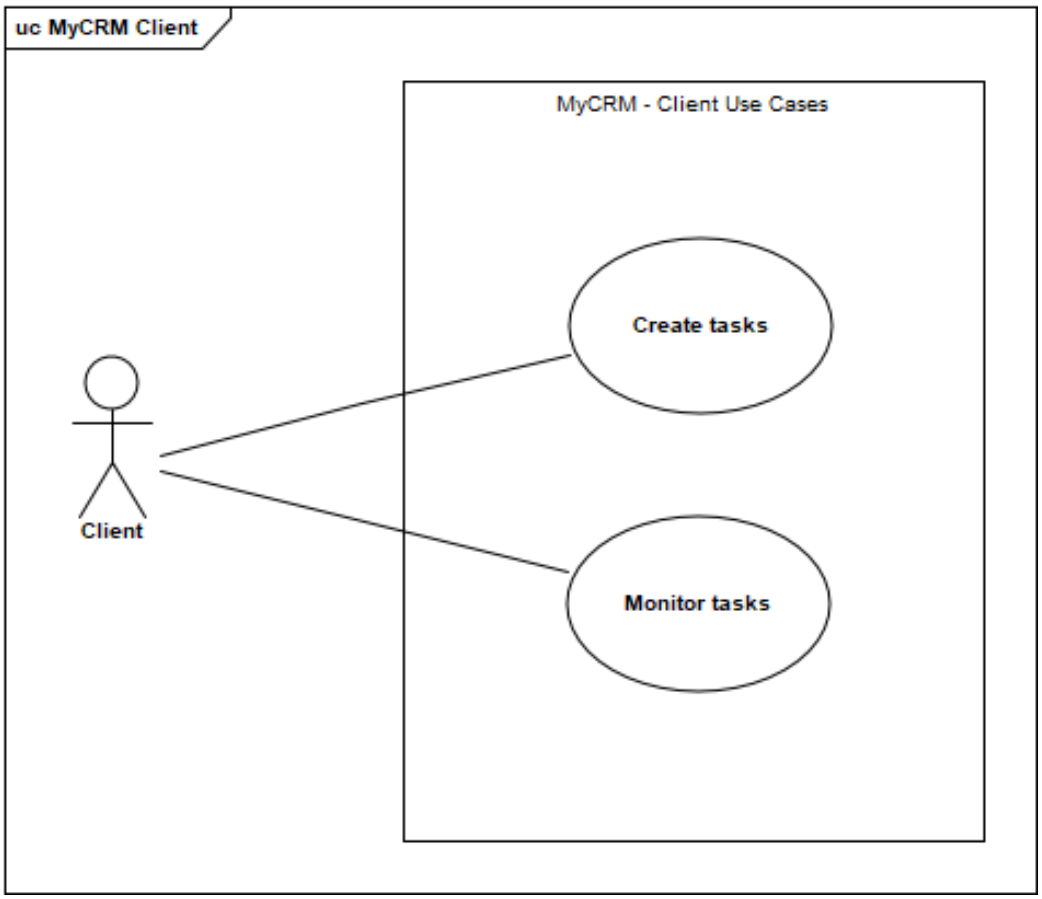
System models are implemented on Enterprise Architect. This section of the document contains Use Case Diagrams, Class Diagrams, E-R diagrams and GUI Design

Use Case Diagrams

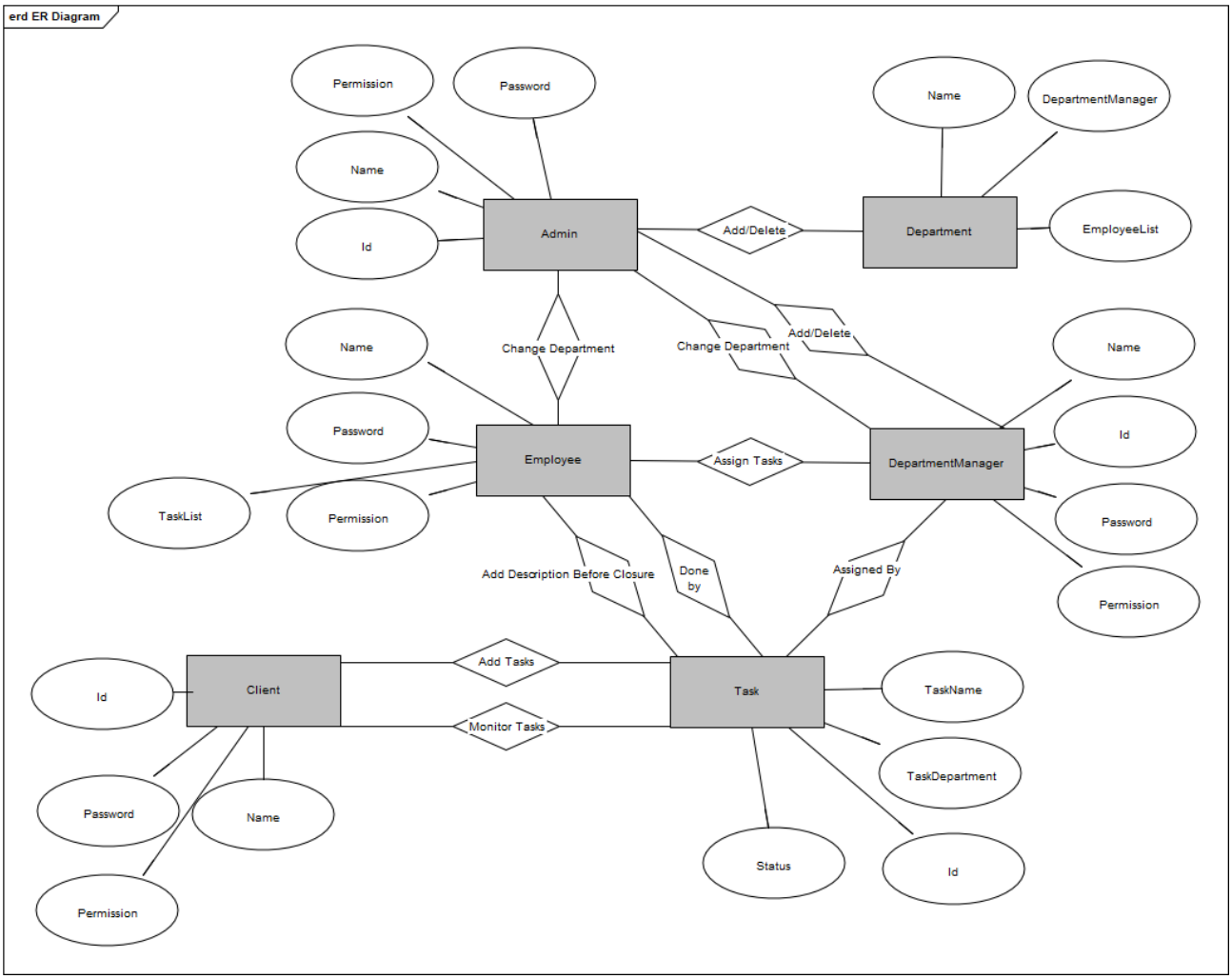
There are three different use case diagrams, one of them shows Admin's capabilities, second one shows use-cases of Client and last one shows the use-cases inside a Department. Also, use-case diagrams may contain other actors.

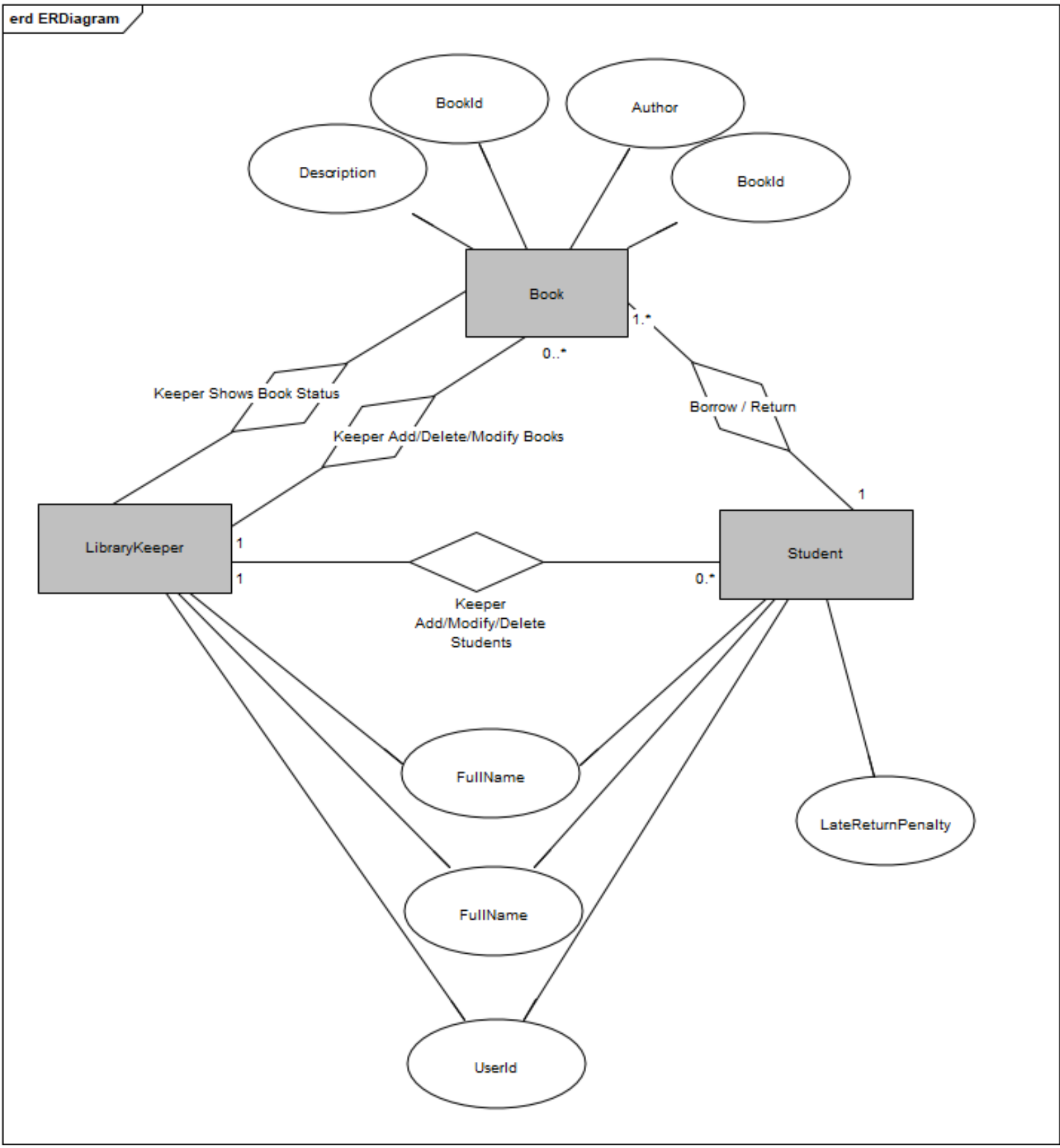

uc MyCRM Admin
MyCRM - Admin Use Cases
Add Department
Delete Department
Assign Managers to Departments
Change Employee's Department
Admin
Software Department Manager
Human Resources Manager
Software Department Employee


uc MyCRM Department
MyCRM - In Department
Assign tasks to employees
Task Closure
«include»
Add Description
Department Manager
Department Employee



E-R Diagram: Here is the E-R diagram which shows the relationship between the entities of the system. On the other hand, it is a good reference for MyCRM's application workflow, and gives a general understanding of whole system.


erd ER Diagram
Permission
Password
Name
Id
Admin
Add/Delete
Name
DepartmentManager
Department
EmployeeList
Change Department
Change Department
Add/Delete
Name
Password
Permission
Employee
Assign Tasks
DepartmentManager
Name
Id
Password
Permission
TaskList
Add Description Before Closure
Done by
Assigned By
Id
Client
Add Tasks
Monitor Tasks
Task
TaskName
TaskDepartment
Id
Password
Name
Permission
Status


erd ERDiagram
BookId
Author
Description
BookId
Book
1..*
0..*
Keeper Shows Book Status
Keeper Add/Delete/Modify Books
Borrow / Return
1
LibraryKeeper
1
1
Student
0..*
Keeper
Add/Modify/Delete
Students
FullName
LateReturnPenalty
FullName
UserId

Class Diagram: This section illustrates Class Diagram of MyCRM. Having a class diagram is important for both understanding the program and programming Object-Oriented code on implementation stages.