Software Requirements Specification

Version 1.0
<<Annotated Version>>

May 26, 2021

Tasks Scheduling

Abd Ulaziz Al-Abd
Mahmoud Hashem
Mohsen Al-Ebrahim

Submitted in partial fulfillment
Of the requirements of
CS 310 Software Engineering

<<Any comments inside double brackets such as these are *not* part of this SRS but are comments upon this SRS example to help the reader understand the point being made.

Refer to the SRS Template for details on the purpose and rules for each section of this document.

This work is based upon the submissions of the Spring 2004 CS 310. The students who submitted these team projects were Thomas Clay, Dustin Denney, Erjon Dervishaj, Tiffanie Dew, Blake Guice, Jonathan Medders, Marla Medders, Tammie Odom, Amro Shorbatli, Joseph Smith, Jay Snellen, Chase Tinney, and Stefanie Watts. >>

Table of Contents

Introduction

1.1. Purpose

The purpose of this document is to present a detailed description of tasks scheduling system and features of this system and all requirements from this system and who are the people who benefit from the system and how do they deal with it. This document is intended for both the stakeholders and the developers of the system

1.2. Scope of Project

this system relevant to most companies of various fields as a means for communicating between manager and employees in a seamless way that includes assigning tasks to each employee and provide the ability to exchange messages about work

1.3. Glossary

Term	**Definition**
Database	Collection of all the information monitored by this system.
Software Requirements Specification	A document that completely describes all of the functions of a proposed system and the constraints under which it must operate. For example, this document.
Stakeholder	Any person with an interest in the project who is not a developer.
Developers	Experienced persons responsible for building the system and always paying attention to its improvement
Report	A message explaining the details of a specific event in the system
Notification	A notification that alerts the user to the arrival of some information or the occurrence of a new event

1.4. References

IEEE. *IEEE Std 830-1998 IEEE Recommended Practice for Software Requirements Specifications*. IEEE Computer Society, 1998

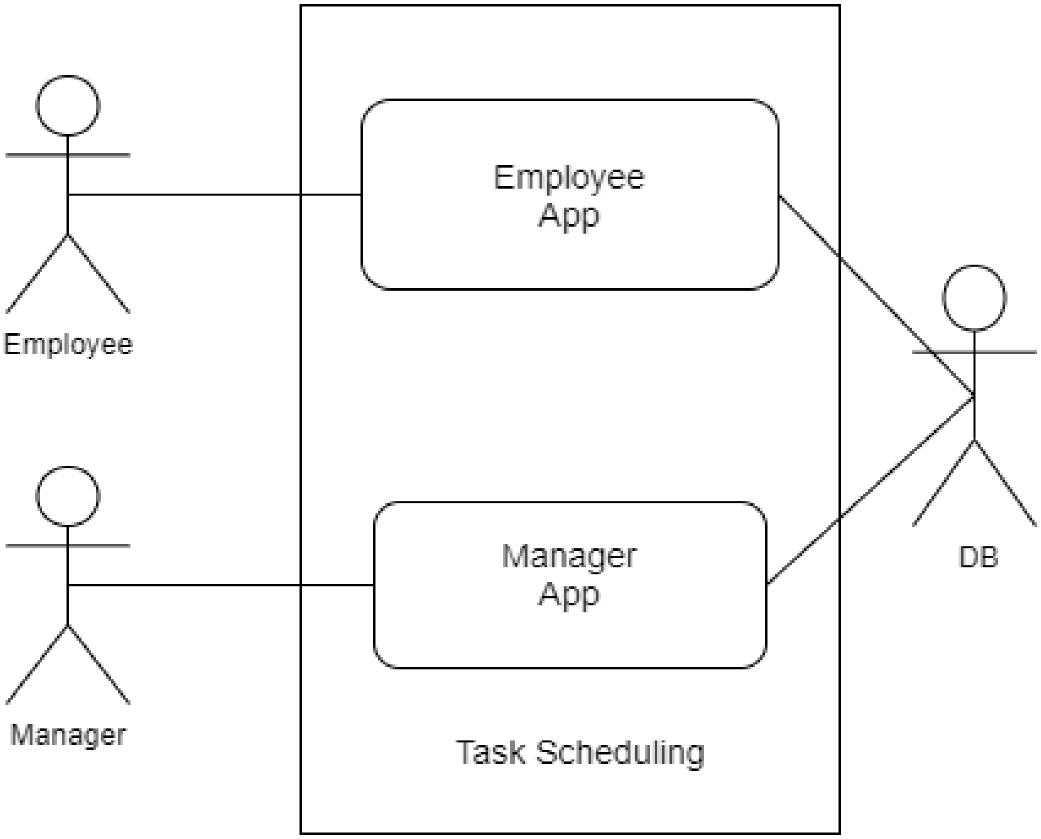
1.5. Overview of Document

The next chapter, the Overall Description section, of this document gives an overview of the functionality of the product. It describes the informal requirements and is used to establish a context for the technical requirements specification in the next chapter.

The third chapter, Requirements Specification section, of this document is written primarily for the developers and describes in technical terms the details of the functionality of the product.
Both sections of the document describe the same software product in its entirety, but are intended for different audiences and thus use different language.

Overall Description

2.1 System Environment

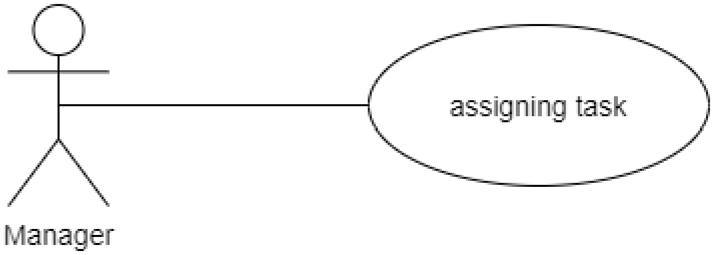


The Task Scheduling System has two active actors and one cooperating system and consist of two apps.
The Manager add the tasks to the system through his app and chose the right employee for the subtask based on his/her skills ,the Employee receives the assignment through his app and the employee can either accept it or refuse it with a note to the manager , And all that via the Database.

2.2 Functional Requirements Specification

2.2.1 Manager Use Cases

Use case: **assigning task**
Diagram:

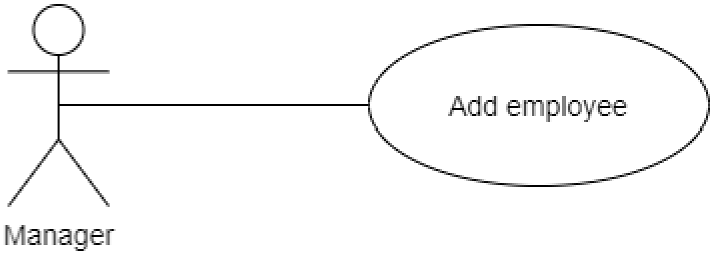


Brief Description :

The manager select new task and assign it to any employee
Initial Step-By-Step Description :

1-The manager requests a new assignment.
2-The system displays the tasks not yet assigned.
3- The manager chooses the task to be assigned.
4- The system displays the available employees who meet the conditions of the assigned task.
5- The manager chooses an employee.
6-The manager saves the assignment process.
7-The system makes sure that all the required details are entered and saves them, otherwise it alert the manager to the necessity of completing the information.

Use case: **add employee**
Diagram:



Brief Description :

The manager requests ‘add new employee’ option to enter a new employee in the system and fill in the required information for this employee

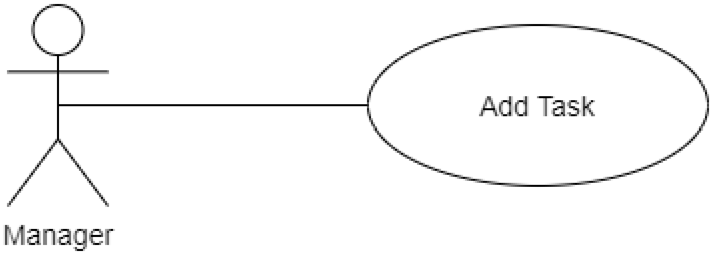
Initial Step-By-Step Description :

1-The manager asks to add a new employee.
2-The system displays a form to insert the new employee information.
3- The manager fills out the form and sends information.
4- The system makes sure that the required information is completed and saves it, otherwise it alerts the manager to correctly inserting information

---

Use case: **Add Task**

Diagram:



Brief Description :

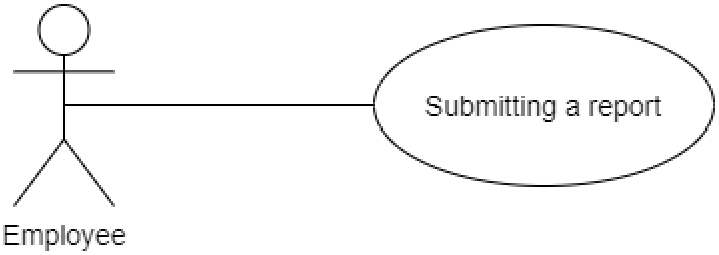
The manager requests ‘add task’ option and enter a new task to fill in the required information and send it

Initial Step-By-Step Description :

1-The manager requests adding a new task to the system.
2-The system displays a form for entering the new task information.
3-The manager fills in the form with the information.
4-The system makes sure that the entered task information is completed and saves it, otherwise it asks the manager to complete the information
related to the entered task

2.2.2 Employee Use Cases

Use case: **submitting a report**
Diagram:



Brief Description :

The employee sends a note to the manager that he has completed the task or sends a report includes the reasons for the delay in the event of delay in the completion of the task

Initial Step-By-Step Description :

1- Employee receive new task
2- If employee finished this task then he send message to manager that the task completed
3- If the employee is late in completing this task, he sends a report to the manager explaining the reasons for the delay

2.3 User Characteristics

As we mentioned earlier, the focus has been on the element of ease, so that it does not require any high features from the user other than knowing the mechanism for creating accounts and logging in, nothing more.

2.4 Non-Functional Requirements

This system will be smart phone application doing on Android and IOS so we created it by using Flutter framework for programming ui and we connected it with MySQL database by using php Rest API. This system takes into account, easy to use so that it suits all work environments regardless of the educational level of workers in the environment in which the system will operate

Requirements Specification

3.1 External Interface Requirements

The only link to an external system is the link to Database to connect the apps and save the data

The Task assignment use case sends the Task ID to the Database and a Boolean is returned denoting the presence of the task .

3.2 Functional Requirements

3.2.1 Task assignment

Use Case Name	Task assignment
XRef	
Trigger	The Manager selects to assign an task and the tasks is already in the database.
Precondition	The Manager has accessed the Task assignment main screen.
Basic Path	1-The manager requests a new assignment. 2-The system displays the tasks not yet assigned. 3- The manager chooses the task to be assigned. 4- The system displays the available employees who meet the conditions of the assigned task. 5- The manager chooses an employee. 6-The manager saves the assignment process. 7-The system makes sure that all the required details are entered and saves them
Alternative Paths	Don't find an appropriate employee
Postcondition	The Task has been assigned

3.2.2 Employee adding

Use Case Name	Employee adding
XRef	
Trigger	The Manager selects to add a new Employee to the database .
Precondition	The Manager has accessed the Add Employee main screen.
Basic Path	1-The manager asks to add a new employee. 2-The system displays a form to insert the new employee information. 3- The manager fills out the form and sends information. 4- The system makes sure that the required information is completed and saves it
Alternative Paths	If any of fields is empty , the Manager must complete the information related to the entered Employee
Postcondition	The Employee has been added to the database.

3.2.3 Task adding

Use Case Name	Task adding
XRef	
Trigger	The Manager selects to add a new Task to the database .
Precondition	The Manager has accessed the Add Task main screen.
Basic Path	1-The manager requests adding a new task to the system. 2-The system displays a form for entering the new task information. 3-The manager fills in the form with the information. 4-The system makes sure that the entered task information is completed and saves it,
Alternative Paths	if any of fields is empty , the Manager must complete the information related to the entered task
Postcondition	The Employee has been added to the database

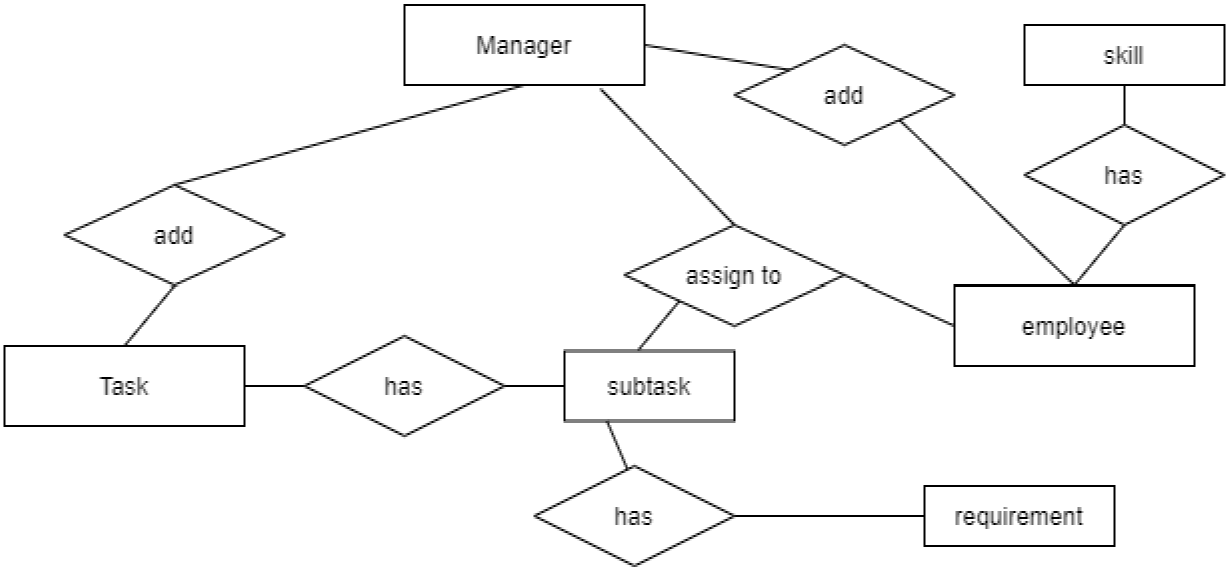
3.2.4 Submit a report about the task

Use Case Name	Submit a report about the task
XRef	
Trigger	The Employee send notifications to the Manager about the Task
Precondition	The Employee has accessed the send notifications main screen
Basic Path	The employee sends a notification to the manager that he has completed the task or sends a report
Alternative Paths	If the task not completed the Employee send the reasons for the delay in the event of delay in the completion of the task
Postcondition	The report has been sent

3.3 Detailed Non-Functional Requirements

3.3.1 Logical Structure of the Data

The logical structure of the data to be stored in the system database is given below.



The data descriptions of each of these data entities is as follows:

Manager and Employee has the same Data Entity :

Data Item	**Type**	**Description**	**Comment**
ID	Int	ID number of Employee in the system	
Name	String	Name of the Employee	
Email	String	Email of the employee to log into his/her app	
Phone	Long Int	The phone number of the employee	
Salary	Double	Salary of the employee	
Start work date	Date	The date of Start working in the company	
Birth date	Date	His/Her Birthdate	

Task Data Entity :

Data Item	Type	Description	Comment
ID	Int	ID number of the task in the system	
Requesting Entity	String	The entity or the party that asked for the task	
Starting date	Date	The date of start working on this task	
Type	String	The type of the task	

Subtask Data Entity :

Data Item	Type	Description	Comment
ID	Int	ID number of the subtask in the system	
title	String	The title of the subtask	
Starting date	Date	The date of start working on this subtask	
deadline	Date	The deadline of this subtask	
description	String	Description about this subtask	

Requirement and Skill has the same Data Entity :

Data Item	Type	Description	Comment
ID	Int	ID number of the requirement	
title	String	The title of requirement	
description	String	Description about the requirement	

3.3.2 Security

The Database will have its own security , the employee can’t log into his/her app until the manager add him/her in the system , the employee can’t delete , edit , or manipulate any information in the system . Otherwise , The manager has full permissions in the system , he/she can add, delete and edit any employee or task .