Tasks Manger

Summary design

**姓名 伊万**

**学号 6318000299**

**姓名 叶寒**

**学号 6318000426**

**姓名 胡萨**

**学号 6318000428**

**姓名 叶得力**

**学号 6317000079**

Contents

[1. Introduction 1](#_Toc21277780)

[1.1 Purpose of writing 1](#_Toc21277781)

1.[2 Project background 2](#_Toc21277782)

1[.3 References 2](#_Toc21277783)

[2.](#_Toc21277784) **[Overall design](#_Toc21277784)** [2](#_Toc21277784)

2.[1 Requirements 2](#_Toc21277785)

2.[2 Operating requirements 3](#_Toc21277786)

2.[3 Interface design 3](#_Toc21277787)

2.[4 Architecture design 4](#_Toc21277788)

2.[5 Basic design concepts and process flow 4](#_Toc21277789)

2.[6 Structure 5](#_Toc21277794)

2.**[7](#_Toc21277801)** [Relationship between functional requirements and modules 6](#_Toc21277801)

2.[8 Human tasks 7](#_Toc21277802)

[3.](#_Toc21277803) **[Logical description of the data](#_Toc21277803)** [7](#_Toc21277803)

3.[1. Static data 7](#_Toc21277804)

3.[2. Dynamic data 8](#_Toc21277808)

3.[3 Internally generated data 9](#_Toc21277810)

3.[4 Data dependencies and constraints 9](#_Toc21277812)

# Introduction

## 1.1 Purpose of writing

The purpose of this plan is to detail the division and schedule of personnel involved in the system development process, determine the solution to the project problem, the system environment, documentation and criteria for product delivery and acceptance, the distribution of project development costs, organization of use, understanding the content of the work, and determine the direction of development.

To learn the theoretical knowledge of website development and master the practical process of website development, this project will develop from the point of view of the specification development process for Tasks Manger, choose a process-oriented approach to development, based on structured programming, analyze problems, confirm the viability of requirements, and then Refine the requirements, clarify the needs of users, perform the design of requirements documentation.

The document is expected to be read by all project participants, including project managers, project planners, system designers, developers, testers, and soon.

## 1.2 Project background

A Task Manger is a Website whish one of the main resource is the team. Therefore the administrator and manager must carefully manage their teams, which makes a profit for the business.

Previously, the administration of team in the compony/workplace was given in directly order. all information was written on paper, and may have been lost or misspelled. but now it is possible to automate the process of administration of the team and the office.

## 1.3 **Reference**

1. "Software Requirements Specification"
2. "Software Design Documents"
3. "Software Testing Documents"
4. "Software Deployment and Use Documents"
5. "Manuals for Division of Labor

# Overall design

## 2.1 Requirements

In accordance with the specifications set out in the requirements analysis documentation, the Task Manger system has been implemented. The system should be as simple as possible for users, as easy to install, operate, maintain, stable operation, information security and reliability.

## 2.2 Operating requirements

Support：

All browsers

Tools software： Visual Studio Code，MySQL Workbench

Language for developing a desktop application: Java, HTML, MySQL, JavaScript, CSS .

Database: MySql 8.0

## 2.3 Interface design

### User interface

Using a Website menu design (example of screenshots will be later)

### Internal interface

Object-Oriented Design class, encapsulated within the class, called in the public class

|  |  |  |
| --- | --- | --- |
| interface | | Passing data |
| Sign up |  | A new user sign up using a username and password |
| Log in |  | Already a user use the username and password to log in |
| Team leader | Add, modify, delete task | Tasks information (task name, deadline, team member） |
| Add, modify, delete team members | Team leader can choose team members and start giving them tasks |
| Team member | Can receive tasks and send reports to the team leader | After receiving the tasks and doing them can send done to the leader |
| Team member as a Team leader | Add, modify, delete team members and tasks | Team member as a Team leader can choose also team and divide the work among them |

## 

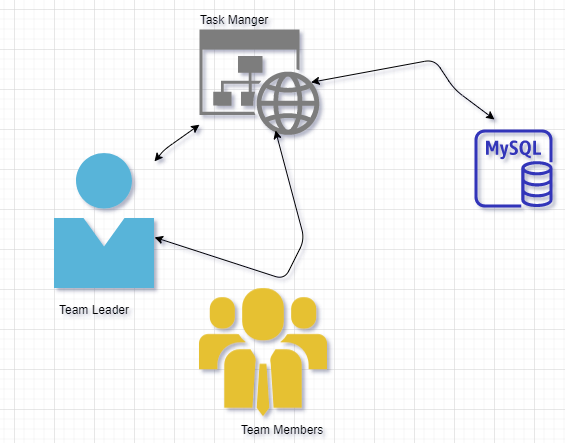
## 2.4 Architecture design

The website has access to the database and interacts with it.



2.5 Basic design concepts and process flow

This is a system where the data goes strictly through the Manager (maintenance staff).



Detailed description:

The website can be opened in any internet browser.

People can register for a new membership.

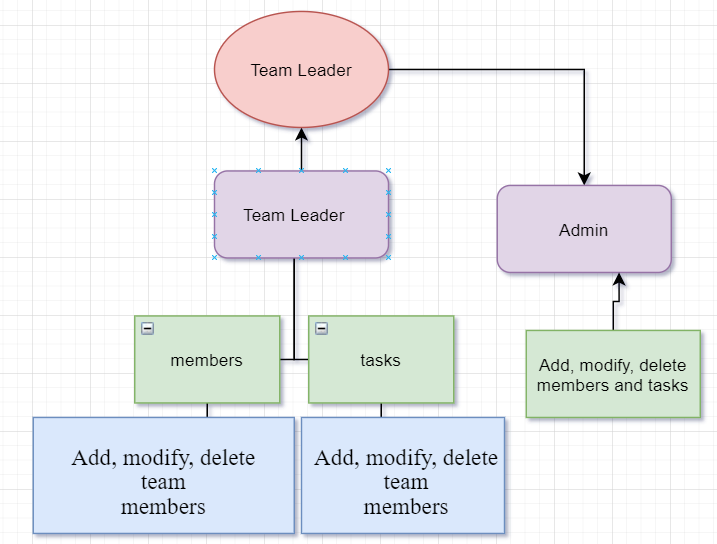
Databases implemented using MySql.

## 2.6 Structure

System has the following main functions：

Manager: authorization and management of team, Tasks.

Admin: authorization and management of managers



## 2.7 Relationship between functional requirements and modules

2.7.1 Administrator Authorization Management Module

Users are divided by functions and permissions. The administrator can change managers and has access to members and tasks. The Manager does not have access to other managers, but has the same rights regarding members and tasks.

### 2.7.2 User login/sign-out module

* Getting a username and password for managers is carried out through the administrator. The Manager can register in the system by himself.
* Administrator registration is performed when the website. To restore your password and administrator username you will need to contact technical support.

2.7.3 Personal data of the guest module

* Before registering, the user agrees to the processing of personal data.
* Manager will be able to get full access to the mebmers data

2.7.4 Tasks’ data module

* Manager can get all the information about the members, as well as change it or delete it

## 2.8 Human tasks

Manager is responsible for their account. Log in to your account when you start working, and log out after work.

# Logical description of the data

## Static data

Admin

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Parameters | Data type | Not null | Key | Properties |
| Admin\_login | Varchar | 1 | 1 | login |
| Admin\_pswrd | Varchar | 1 | 0 | password |
| Admin\_name | Int | 1 | 0 | name |

Manager

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Parameters | Data type | Not null | Key | Properties |
| Manager\_id | Int | 1 | 1 | auto-generated number |
| Manager\_login | Double | 1 | 0 | login |
| Manager\_pswrd | Varchar | 1 | 0 | password |
| Manager\_name | Varchar | 1 | 0 | username |
| Manager\_numb | Varchar | 1 | 0 | Work number |

Number

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Parameters | Data type | Not null | Key | Properties |
| Member\_id | Int | 1 | 1 | Auto-generated number |
| Member\_name | Varchar | 1 | 0 | username |
| Member\_numb | Double | 1 | 否 | work numb |
| Mermer\_sex | Varchar | 1 | 否 | gender |
| Member\_age | Varchar | 1 | 否 | age |
| Member\_phone | Varchar | 1 | 否 | phone numb |

Taskls

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Parameters | Data type | Not null | Key | Properties |
| Task\_id | Int | 1 | 1 | Auto-generated number |
| Task\_type | Varchar | 1 | 0 | Type a Task(Top priority, standard and so etc.) |
| Task\_numb | Varchar | 1 | 0 | Numb. of Task |
| Task\_Deadline | Int | 1 | 0 | Deadline |
| Task\_capacity | Int | 1 | 0 | Capacity, persons in member |

## Dynamic data

Reservation

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Parameters | Data type | Not null | Key | Properties |
| R\_id | Int | 1 | 1 | Auto-generated number |
| R\_Task | Int | 1 | 1(foreign key) | Id of Task |
| R\_Member | Int | 1 | 1(foreign key) | Id of member |
| R\_from | Date | 1 | 0 | From what date is the member reserved |
| R\_to | Date | 1 | 0 | Until what date is the member reserved |
| R\_prepayment | Int | 1 | 0 | Prepayment for member |

Room state

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Parameters | Data type | Not null | Key | Properties |
| Rs\_id | Int | 1 | 1 | Auto-generated number |
| Rs\_task | Int | 1 | 1(foreign key) | Id of task |
| Rs\_member | Int | 1 | 1(foreign key) | Id of member |
| Rs\_from | Date | 1 | 0 | From what date is the task reserved |
| Rs\_to | Date | 1 | 0 | Until what date is the task reserved |
| Rs\_reason | Varchar | 1 | 0 | Reason the task is occupied |

## Internally generated data

App will generate the following data:

* Number of available and occupied tasks
* Capacity of the compony/workplace
* Percentage of occupied tasks
* Number of reserved tasks
* Generates an report
* Additionally, the option to create schedules and work statistics

## Data dependencies and constraints

### Data type constraints

Data in the data type is not arbitrary, but requires a certain type and scope constraints: such as academic number is a string type, but the requirements of the string must be composed of numbers.

### Data dependency constraints

The data in one repository may refer to the data in another repository.

|  |  |  |  |
| --- | --- | --- | --- |
| Source repository | DB name | Reference information base | Reference data name |
| Hotel | Admin | Admin | Admin\_id |
| Manager | Manager | Manager\_id |
| Member | task | taks\_id |
| Task | member | member\_id |
|  | Reservation | Reservation | R\_id |
|  | Task\_state | Task state | Rs\_is |