**Document Design I**

**“Where is my car?” Application**

**Members**

Pitchaporn Likitpanjamanon 59070503436

Mullika Phanhong 59070503442

Onnicha Seemawong 59070503455

Chayutra Tatviset 59070503470

**1. Introduction**

**1.1 Purpose of the design document**

The purpose of the Software design document is to provide a description of the designing car searching application called Where is my car. In the document, it consists of architecture design and system design for better understanding software components.

This document is created to make stakeholders who associate with the software and software developer be on the same page. Moreover, it can be more useful for testing and developing software in the future.

**1.2 Document Convention**

- staff : person who works for an organization

- array : a collection of variables of the same type

- json : is an open-standard file format that uses human-readable text to

transmit data objects consisting of attribute–value pairs and array data types

(or any other serializable value)

- real-time database : a database system which is updated all the time

- application : is computer software designed to perform a group of

]coordinated functions, tasks, or activities for the benefit of the user.

**1.3 Project Background and Scope**

“Where is my car?” application is created for the driver and people who want to find a parking lot and cannot memorize their parking lot. With this application, it will make parking be more convenient and reduce the time to find parking lots. For the department store side, this application will help the department store have better management quality of car parking system, reduce the money spend on staff, and increase the satisfaction level of customer toward department store effectively.

In this application, user can

- View Parking lot map

- Use voice command (ask for available lot and etc.)

- View available normal/priority parking lot in each department store

- Suggest the best parking position

- View current parking information (position, approximately time, total

parking fee)

- Notify the parking position to user automatically

- Create your own account

- Add/Edit/Delete your privilege

- Log In and Log Out

“Where is my car?” is mobile application with real-time database. It can run on both iOS and Android platform. We develop the application by using Swift language for iOS and Java language for Android. For real-time database, we use it to connect with sensor and application and we develop it by using PHP language.

**1.4 Document Overview**

This document is composed of 6 parts which will explain how the application is created

1. Introduction

- Description of the purposes of this document

- Background and boundary of the application and also including its objectives

2. Data Design

- Description of all the data structures and their components

3. Architectural Design

- Description of software structure, components and functions

- Explanation of relationships between each module, its input and output

4. Process Manual Specification

- Explanation of plan and schedule of the project in detail

- Project monitoring and tracking of work process

- Overall cost of project

**2. Data Design**

**2.1. Internal software data structure**

When the application starts, each components will receive and send data to each others via JSON file. By using JSON file, other components that want to use same data will not need to fetch that data from database again. So it make the application be more efficient.

**2.2. Global data structure**

The application fetches data from database and then keep it in JSON file. Therefore, every components can access these data directly without fetching data from database again.

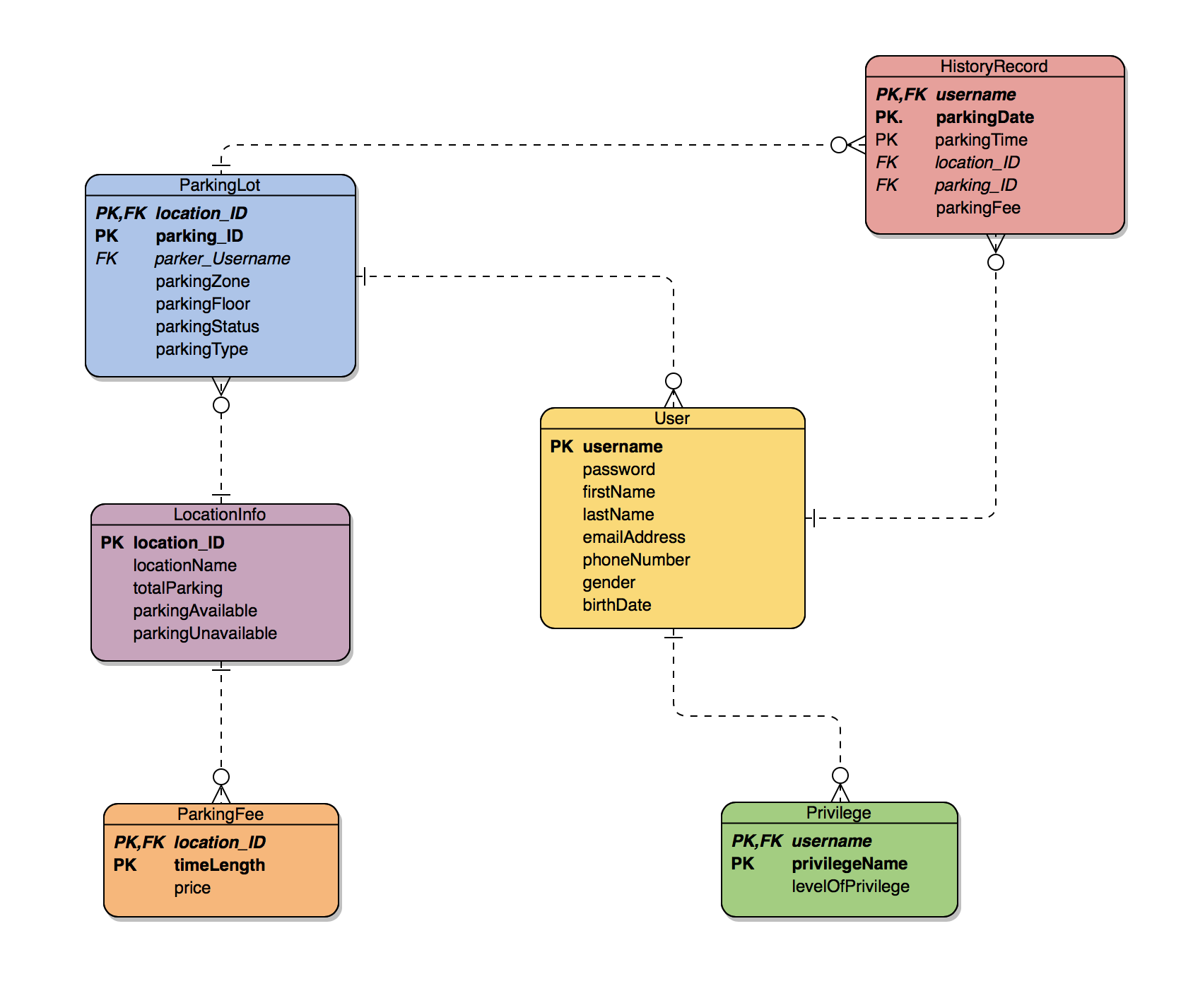
**2.3. Temporary data structure**

This will be an array structure because it just uses to keep the temporary data only so that these data does not need to insert or update into database. If the array does not be in use anymore, array will be destroyed.

**2.4. Database Description**

Database System which is designed to be able to store/manipulate all data using in the application appropriately and efficiently. The **Bold** attribute represents ‘**Primary Key**’ while *Italic* attribute represents *‘Foreign Key*’.

**ER Diagram**



**Table Name : User**

Stores all users’ personal information in detail

|  |  |  |
| --- | --- | --- |
| **Attribute** | **Type** | **Description** |
| **username** | String | The identification name used to identify a user |
| password | String | A string which gives permission to access to the system |
| firstName | String | User’s firstname |
| lastName | String | User’s lastname |
| emailAddress | String | User’s Email Address |
| phoneNumber | int | User’s phone number |
| gender | char | User’s gender |
| birthDate | Timestamp | User’s birth date |

**Table Name : Privilege**

Stores privilege of each user including the level.

|  |  |  |
| --- | --- | --- |
| **Attribute** | **Type** | **Description** |
| ***username*** | String | The identification name used to identify a user |
| **privilegeName** | String | Name of privilege |
| levelOfPrivilege | String | Level of privilege e.g. Gold, Platinum, Premium |

**Table Name : LocationInfo**

Stores information about location and its parking lots

|  |  |  |
| --- | --- | --- |
| **Attribute** | **Type** | **Description** |
| **location\_ID** | int | The identification number of location |
| locationName | String | Name of location |
| totalParking | int | The number of total parking lots in each location |
| parkingAvailable | int | The number of available parking lots |
| parkingUnavailable | int | The number of occupied parking lots |

**Table Name : ParkingFee**

Stores the fee details of each location

|  |  |  |
| --- | --- | --- |
| **Attribute** | **Type** | **Description** |
| ***location\_ID*** | int | The identification number of location |
| **timeLength** | int | Time interval of the parking time |
| price | int | Parkign fee according to the amount of time parked |

**Table Name : ParkingLot**

Stores information of each parking position in detail

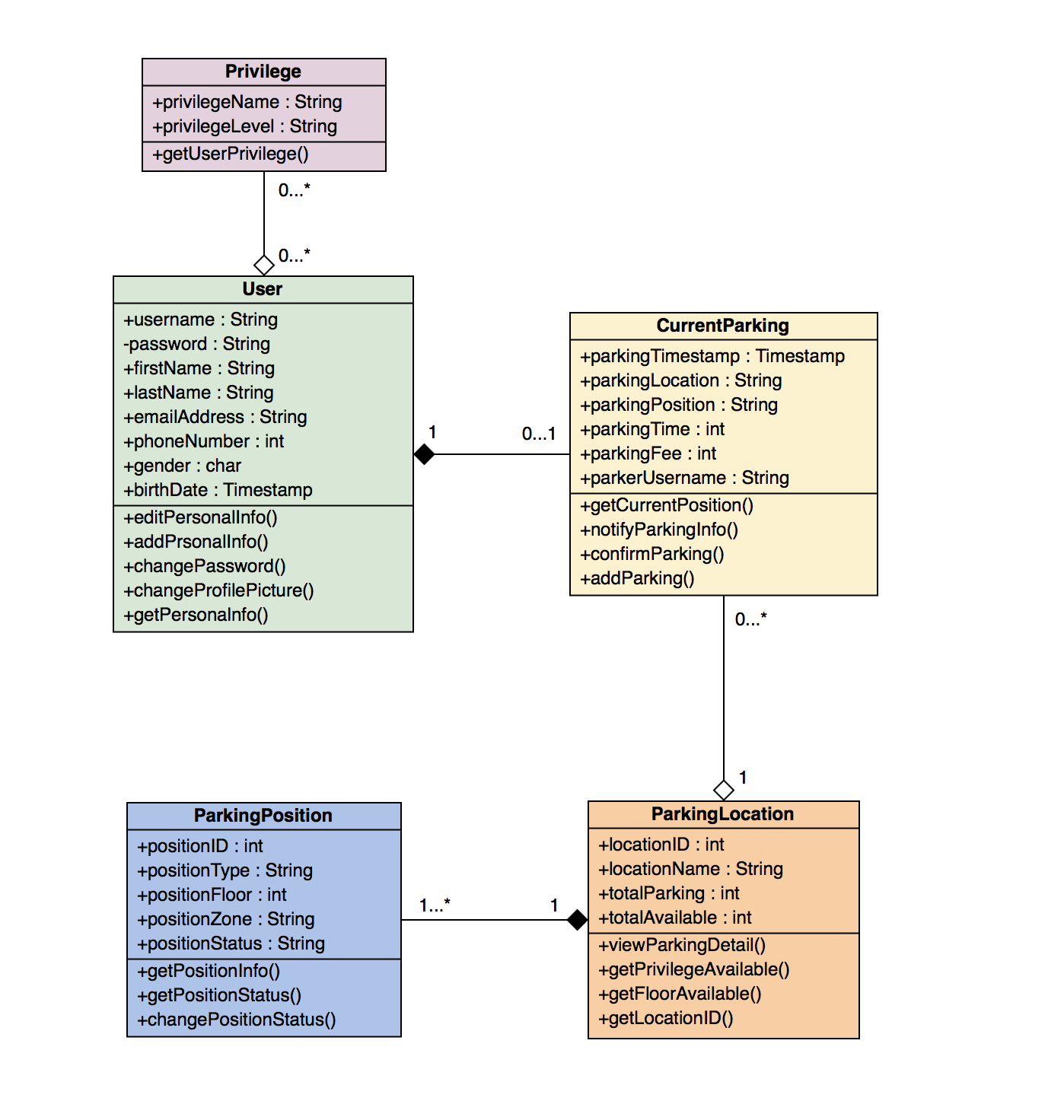
|  |  |  |
| --- | --- | --- |
| **Attribute** | **Type** | **Description** |
| ***location\_ID*** | int | ID of location where that position is in |
| **position\_ID** | int | The identification number of each parking position |
| parker\_Username | String | Username of user who parked on that position |
| parkingZone | String | The position’s zone of parking e.g. West Park, East Park |
| parkingFloor | int | Floor number of that parking position e.g. 1st Floor |
| parkingStatus | String | Status of that position whethe it is availale or busy |
| parkingType | String | Parking type of that position e.g. Lady, Disabled, Normal |

**Table Name : HistoryRecord**

Stores history of all parking activities

|  |  |  |
| --- | --- | --- |
| **Attribute** | **Type** | **Description** |
| ***username*** | String | Username of the parker |
| **parkingDate** | Timestamp | Date of parking |
| parkingTime | int | The amount of time of parking |
| *location\_ID* | int | ID of location of that parking |
| *parking\_ID* | int | Position of parking |
| parkingFee | int | Total amount of money paid for parking |

**Class Diagram**



**Class Name : User**

Class which holds data about our application’s users

|  |  |  |
| --- | --- | --- |
| **Object** | **Type** | **Description** |
| + username | String | The identification name used to identify a user |
| - password | String | A string which gives permission to access to the system |
| + firstName | String | First name of user |
| + lastName | String | Last name of user |
| + emailAddress | String | Email Address of user |
| + phoneNumber | String | Phone number of user |
| + gender | String | Gender of user |
| + birthDate | String | Birth date of user |
| **Operation/Method** | **Type (I/O)** | **Description** |
| + editPersonalInfo() | - | Used when user would like to edit user’s information |
| + addPersonalInfo() | - | Used when user signs up for new account |
| + changePassword() | - | Used when user want to change his/her password |
| + changeProfilePicture() | - | Used when user want to change his/her profile picture |
| + getPersonalInfo() | - | Used to show user’s information when it comes to profile page |

**Class Name: Privillege**

Class which holds the data of all privilleges

|  |  |  |
| --- | --- | --- |
| **Object** | **Type** | **Description** |
| + privillegeName | String | Name of each privillege |
| + privillegeLevel | String | Level of this privillege |
| **Operation/Method** | **Type (I/O)** | **Description** |
| + getUserPrivilege() | - | Used when we want to get the privillege of each user |

**Class name: CurrentParking**

Class which holds the data that needs to store when user already parks a car

|  |  |  |
| --- | --- | --- |
| **Object** | **Type** | **Description** |
| + ParkingTimestamp | Timestamp | Date and time that the particular user parked |
| + ParkingLocation | String | Location of the parking |
| + ParkingPosition | String | Specific position of parking |
| + ParkingTime | int | The amount of time that user has been parking |
| + ParkingFee | int | Current total fee the user has to pay for parking |
| + ParkerUsername | String | Username of the parker |
| **Operation/Method** | **Type (I/O)** | **Description** |
| + getCurrentPosition() | - | Used when we want to get current parking position |
| + notifyParkingInfo() | - | Used when notify the parking detail for user |
| + confirmParking() | - | Used when user has confirmed his/her position |
| + addParking() | - | Used when user leaves the parking position |

**Class name: ParkingPosition**

Class which holds the necessary data of each parking position

|  |  |  |
| --- | --- | --- |
| **Object** | **Type** | **Description** |
| + positionID | int | ID number of each parking position |
| + positionType | String | Type of each parking position (e.g. Lady parking) |
| + positionFloor | int | Floor that each parking position is at |
| + positionZone | String | Zone of each parking position |
| + positionStatus | String | Status of each parking position (e.g. Available) |
| **Operation/Method** | **Type (I/O)** | **Description** |
| + getPositionInfo() | - | Usedto show information of each parking position |
| + getPositionStatus() | - | Used to get each parking position’s status |
| + changePositionStatus() | - | Used when the program needs to update the position status (e.g. Available to Not Available) |

**Class Name: ParkingLocation**

Class which holds the data of parking location (e.g. department store)

|  |  |  |
| --- | --- | --- |
| **Object** | **Type** | **Description** |
| + locationID | int | The identification number of each location |
| + locationName | String | Name of each parking location |
| + totalParking | int | The total number of parking lots in that location |
| + totalAvailable | int | The total number of available parking lots at a time |
| **Operation/Method** | **Type (I/O)** | **Description** |
| + viewParkingDetail() | - | Used when we want to see the parking detail |
| + getPrivilegeAvailable() | - | Used when shows the number of available privilege lots |
| + getFloorAvailabble() | - | Used when shows the number of available lots each floor |
| + getLocationID | - | Used when we want to get the location ID |

**Relationship between each class**

1. **User and Privillege**

Each user can have privileges or have no privileges so that we use “aggregation” relationship. In this relationship, each user can have no privileges or have many privileges and each privilege can belong to no user or many users.

**2. User and CurrentParking**

Each current parking must belong to one user so that we use “composition” relationship. In this relationship, each user can have no current parking position or only one current parking position and each current parking position must belong to only one user.

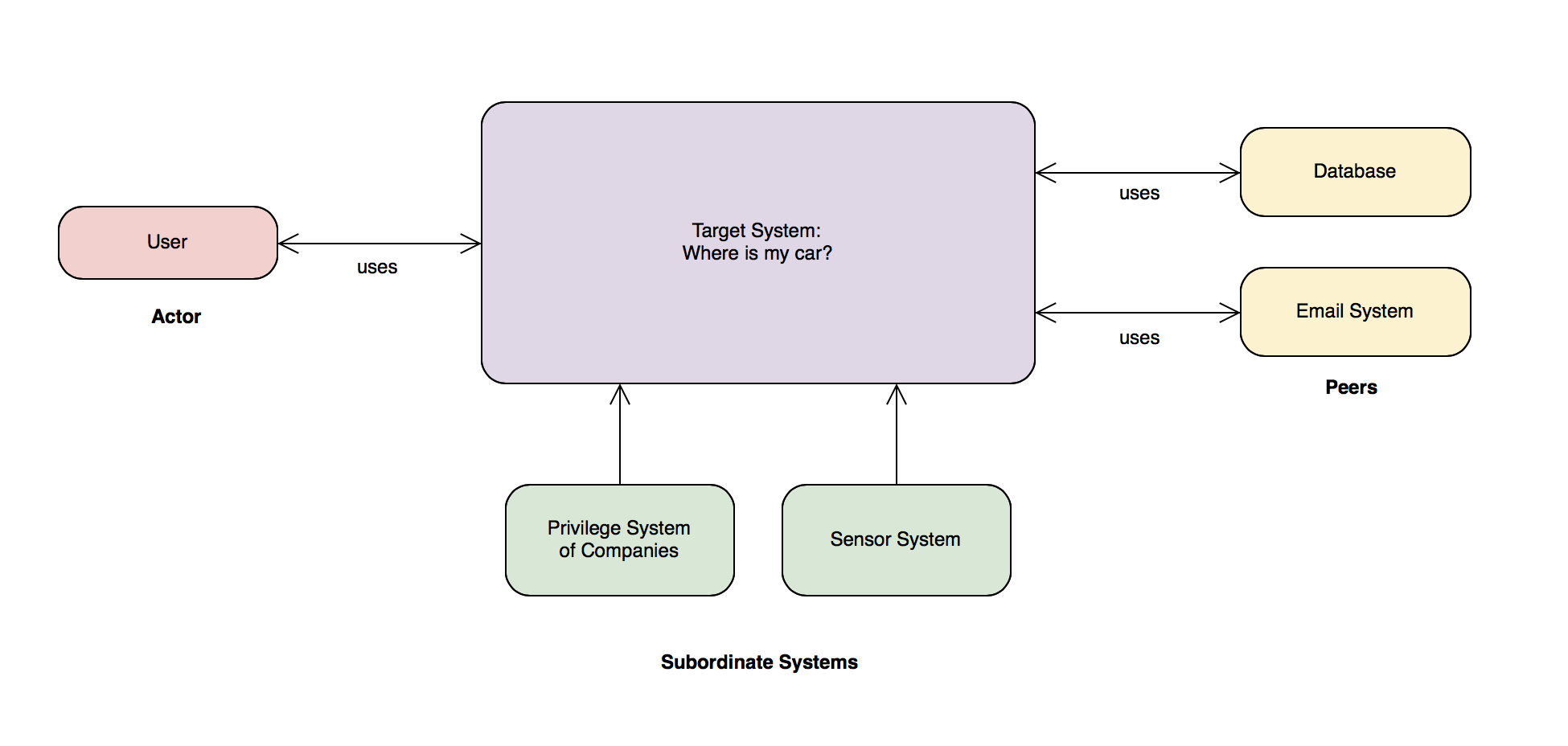
**3. CurrentParking and ParkingLocation**

Each parking location can have current parking position or have no current parking position so that we use “aggregation” relationship. In this relationship, each parking location can have no current parking position or many current parking positions and each current parking position is in only one parking location.

**4. ParkingLocation and ParkingPosition**

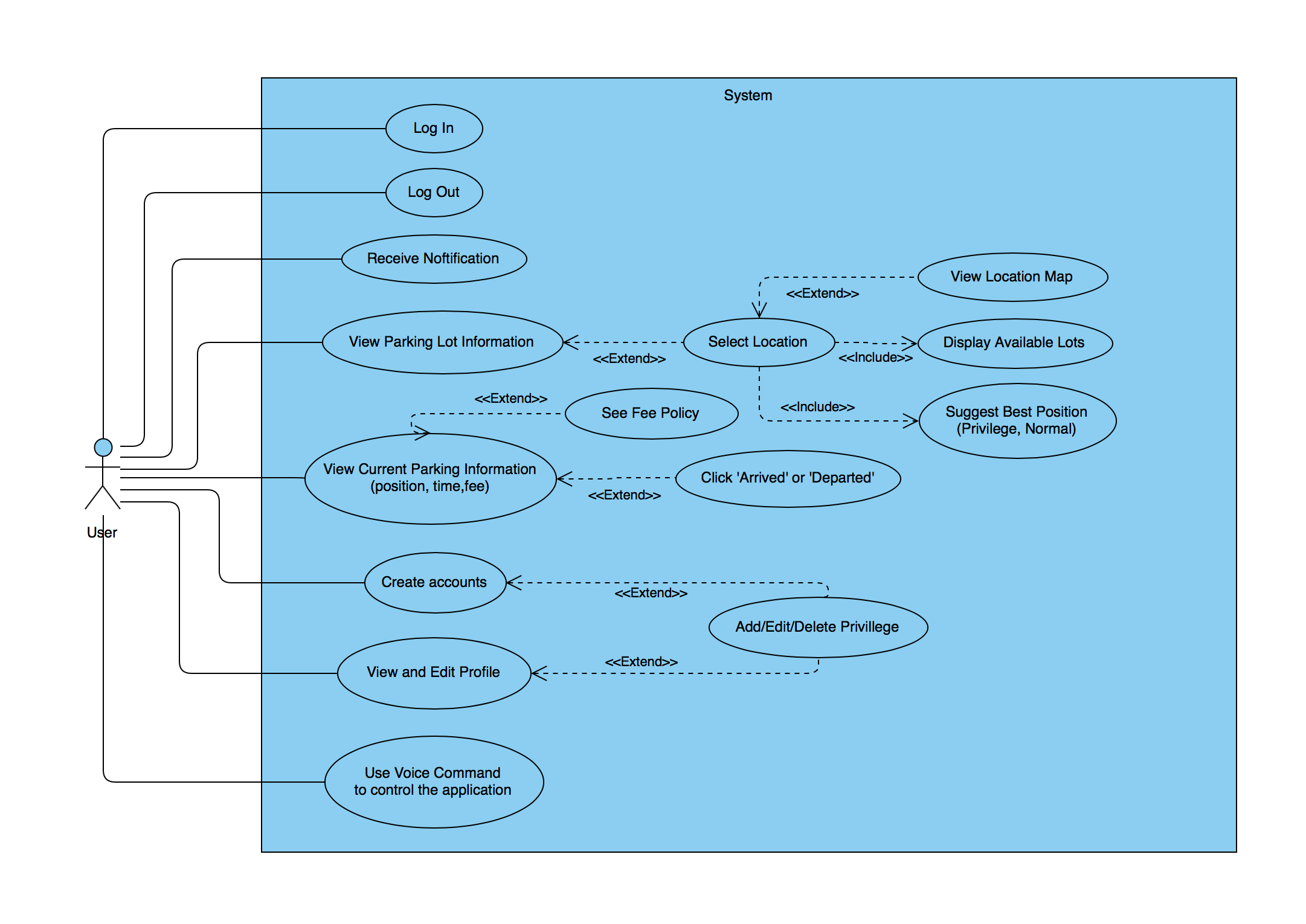
Each parking position must be in one parking location so that we use “composition” relationship. In this relationship, each parking location can have one or many parking positions and each parking position is in only one parking location.

**3. Architectural Design**

Here are the description of how the target system relates with other outside systems:

* **Target System** : System we are focusing on: Where is my car? system
* **Actor** : Entities or systems which intereact with the target system
  + User : People who would like to park his/her car
* **Peers** : Entities or systems which interact on peer-to-peer basis (exchange data)
  + Database : Systems used to store all the data needed in the target system
  + Email System : System which handles email verification/notification
* **Subordinate Systems** : Entities or systems which are used by the target system
  + Privilege System of Companies : System which is used by the target system to provide verification and details of each users’ privilege
  + Sensor System : System which manipulates sensors then provide the target system information about parking lots

**Use case Diagram**



**4. Process Manual Specifications**

**4.1. Project plan and monitoring method**

In this project, each person will be responsible on different tasks in order to finish the project in time and be easier to track each work process from the undertaker. We can evaluate the work process from checking with project plan gantt chart and performance of each member from the operation report of each members.

Here is the responsibility of each members

|  |  |  |
| --- | --- | --- |
| **Role** | **Undertaker** | **Email** |
| Project Manager | Chayutra Tatviset | cytmill@gmail.com |
| Project Sponsor | Pitchaporn Likitpanjamanon | likitpanjamanon.p@gmail.com |
| Business Analyst | Mullika Phanhong | ssycole@hotmail.com |
| Database Design | Pitchaporn Likitpanjamanon | likitpanjamanon.p@gmail.com |
| Software Design | Mullika Phanhong | ssycole@hotmail.com |
| UI Design | Onnicha Seemawong, Chayutra Tatviset | nook5\_215@hotmail.com,  cytmill@gmail.com |
| Design Document | Mullika Phanhong, Pitchaporn Likitpanjamanon | ssycole@hotmail.com, likitpanjamanon.p@gmail.com |
| Registration and Account System | Pitchaporn Likitpanjamanon | likitpanjamanon.p@gmail.com |
| Parking Lot Tracking System | Chayutra Tatviset | cytmill@gmail.com |
| Car parking information System | Onnicha Seemawong | nook5\_215@hotmail.com |
| Voice Command System | Chayutra Tatviset | cytmill@gmail.com |
| Notification System | Onnicha Seemawong | nook5\_215@hotmail.com |
| Smart Suggest System | Mullika Phanhong | ssycole@hotmail.com |
| Security System | Pitchaporn Likitpanjamanon | likitpanjamanon.p@gmail.com |

**Gantt Chart**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Task lists | Duration: 4 months | | | | | | | | | | | | | | | |
| September | | | | October | | | | November | | | | December | | | |
| 1 | 2  **Start Project (3 Sep.)** | 3 | 4 | 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 |
| **Planning Phase** | | | | | | | | | | | | | | | | |
| - Kickoff Meeting |  | **Requirement analysis (4 Sep.)** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **Requirement** | | | | | | | | | | | | | | | | |
| - Requirement  gathering |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| - Analyze customer  requirements |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| - Draft preliminary requirements |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| - Requirement document |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| - Requirement document review |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| - Incorporate feedback |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| - Develop requirement |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| - Requirement approval |  |  |  |  | **Contract Sign (27 Sep.)** |  |  |  |  |  |  |  |  |  |  |  |
| **Design Phase** | **Starting Design (28 Sep.)** | | | | | | | | | | | | | | | |
| Design Prototype |  | | | | | | | | | | | | | | | |
| - Develop database design |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| - Develop software design |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| - Develop UI design |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| - Prototype Approval |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Task lists | Duration: 4 months | | | | | | | | | | | | | | | |
| September | | | | October | | | | November | | | | December | | | |
| 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 |
| Design Document |  | | | | | | | | | | | | | | | |
| - Design document |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| - Design document presentation |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| - Design Approval |  |  |  |  |  |  |  | **Start Implementing (16 Oct.)** |  |  |  |  |  |  |  |  |
| **Implementation Phase** |  | | | | | | | | | | | | | | | |
| - Review specification |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| - Plan the integration |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Implement components |  | | | | | | | | | | | | | | | |
| - Registration and account system |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| - Parking lot tracking system |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| - Car parking information System |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| - Voice Command System |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| - Notification System |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| - Smart suggest System |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| - System Integration |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Task lists | Duration: 4 months | | | | | | | | | | | | | | | |
| September | | | | October | | | | November | | | | December | | | |
| 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 |
| **Testing and evaluating Phase** |  | | | | | | | | | | | | | | | |
| - Define evaluation mission |  |  |  |  |  |  |  |  |  |  | **Testing with users (9 Nov.)** |  |  |  |  |  |
| - Verify test approach |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| - Test and gathering feedback |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| - Testing evaluation |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| - Software improvement and bugs fix |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **Product Launch Phase** |  |  |  |  |  |  |  |  |  |  |  |  |  |  | **Product Launch (26 Nov.)** |  |
| **Monitor and control Phase** |  | | | | | | | | | | | | | | | |
| - Scope |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| - Measure team performance (KPI) |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **Closing Phase** |  | | | | | | | | | | | | | | | |
| - Lessons learned |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| - Budget report |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| - Project deliverable transfer |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| - Contracts are formally closed |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

**Close Project (25 Dec.)**

**4.2. Employee work / Task assignment process**

example of a “ operation record ”

**Operation Record**

Start Date ...…/……/.…..

Project Name: ……………………………………………..

Project Manager: …………………………………………..

Delivery Date: …./.…/…. Project Period: …………….

Assignor: ……………………….. Assign Date: …./…../…..

Task: ……………………………………………………….

Developer: ……………………… Tester: …………………………

Developer part

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No. | Work detail | Date | total time (hr.) | delayed time (hr.) |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
| Total | | |  |  |

Tester part

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No. | Work detail | Date | working time (hr.) | delayed time (hr.) |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
| Total | | |  |  |

The operation record is used for recording the work process of each developer in order to know that each task is delayed or not.

1. General Information includes

* Start Date
* Project Manager
* Delivery Date
* Project Period
* Assignor
* Assign Date
* Task Name
* Developer
* Tester

2. Work process information

It will separate into two parts which are developer part and tester part. This section shows the detail of each work process including work detail, date, working time, and delayed time. Moreover, it shows the total of working time and delayed time. This section includes

* The order (no.)
* Work detail
* Date
* Working time (hr.)
* Delayed time (hr.)
* Total of working time (hr.)
* Total of delayed time (hr.)

Example of “ filled operation record ”

**Operation Record**

Start Date 10/05/2018.

Project Name: “Where is my car?” application

Project Manager: Chayutra Tatviset

Delivery Date: 18/05/2018 Project Period: 1 weeks

Assignor: Chayutra Tatviset Assign Date: 11/05/2018

Task: Notification System

Developer: Onnicha Seemawong Tester: Pitchaporn Likitpanjamanon

Developer part

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No. | Work detail | Date | total time (hr.) | delayed time (hr.) |
| 1 | Design the notification system | 11/05/2018 | 5 | 2 |
| 2 | Start writing notification system | 13/05/2018 | 8 | - |
| 3 | Continue writing notification system | 14/05/2018 | 8 | - |
| 4 | Develop notification system | 16/05/2018 | 9 | 1 |
| Total | | | 30 | 3 |

Tester part

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No. | Work detail | Date | working time (hr.) | delayed time (hr.) |
| 1 | Check the design of notification system | 12/05/2018 | 6 | - |
| 2 | Test the notification system | 15/05/2018 | 5 | 1 |
| Total | | | 11 | 1 |

Example of “ operation report ”

**Operation Report**

Date …./…./…..

Project Name: …………………………………………..

Project Manager: ………………………………………..

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| No. | Work detail | Start Date | Finish Date | Time Expected (hr.) | Time Use (hr.) | Undertaker |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
| Total | | | |  |  |  |

The operation report shows the report of each task. It includes two parts.

1. General Information

* Date (start writing report date)
* Project Name
* Project Manager

2. Task Information

* The order (no.)
* Work detail
* Start Date
* Finish Date
* Time Expected (hr.)
* Time Use (hr.)
* Undertaker (a person who is responsible for specific task)
* Total time expected (hr.)
* Total time used (hr.)

Example of “ filled operation report ”

**Operation Report**

Date 10/05/2018

Project Name: “Where is my car?” application

Project Manager: Chayutra Tatviset

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| No. | Work detail | Start Date | Finish Date | Time Expected (hr.) | Time Use (hr.) | Undertaker |
| 1. | design overview of software | 11/05/2018 | 14/05/2018 | 24 | 24 | Mullika |
| 2. | design database | 11/05/2018 | 14/05/2018 | 20 | 20 | Pitchaporn |
| 3. | UI Design | 15/05/2018 | 17/05/2018 | 20 | 24 | Onnicha, Chayutra |
| 4. | Smart Suggest System | 16/05/2018 | 23/05/2018 | 56 | 50 | Mullika |
| Total | | | | 120 | 118 | Chayutra |

**4.3. Final project cost method with example**

Example of “ working hours record ”

**Working Hours Record**

Week …….. Month …….. Year ……..

Project Name: ……………………………………..

Name: ……………………………………….. Employee Number: …………..……

Department: ……………………………….... Position: ……………………………

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| No. | Date | Clock in | Clock out | Working hour | Work Detail | Progress (%) | Note |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |

Total working hours: …….. hrs.

Working hours record is used to record the attendance of employee or team members. It can use to determine the project cost and evaluate the working plan to make the project finish on time. It is composed of two parts.

1. General Information

* Week
* Month
* Year
* Project Name
* Name of employee
* Employee Number
* Department
* Position

2. Work Information

* The order (no.)
* Date
* Clock in (time that the employee starts work)
* Clock out (time that the employee gets off work)
* Working hour
* Work detail
* Progress (%) (progress of work)
* Note

Example of “ filled working hours record ”

**Working Hours Record**

Week: 1 Month: January Year: 2018

Project Name: “Where is my car?”application

Name: Onnicha Seemawong Employee Number: A1009

Department: Software Delopment Position: Business Developer

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| No. | Date | Clock in | Clock out | Working hour | Work Detail | Progress (%) | Note |
| 1. | 05/01/2018 | 10.00 | 16.00 | 6 | Develop car parking information system | 10 |  |
| 2. | 06/01/2018 | 10.00 | 16.00 | 6 | Develop car parking information system | 20 |  |
| 3. | 07/01/2018 | 09.00 | 16.00 | 7 | Develop car parking information system | 40 |  |

Total working hours: 19 hrs.

Example of “ project cost sheet (per month) ”

**project cost sheet (per month)**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No. | List | Cost per hour (Baht) | Number of hours | Cost per month (Baht) |
| 1. |  |  |  |  |
| 2. |  |  |  |  |
| 3. |  |  |  |  |
| 4. |  |  |  |  |
| 5. |  |  |  |  |
| Total | | |  |  |

The project expense sheet shows the expense for each month by looking at the cost that a company need to pay for one hour and number of working hours. Therefore, this sheet can show total cost that the company need to pay monthly.

Example of “ filled project cost sheet (per month) ”

**project cost sheet (per month)**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No. | List | Cost per hour (Baht) | Number of hours | Cost per month (Baht) |
| 1. | Project Manager | 800 | 50 | 40,000 |
| 2. | Project Sponsor | 600 | 35 | 21,000 |
| 3. | Business Analyst | 700 | 40 | 28,000 |
| 4. | Developers | 450 | 60 | 27,000 |
| 5. | Other expense   * Renting building * Firebase | - | - | -  20,000  875 |
| Total | | | 180 | 252,875 |

Example of “ total project cost sheet ”

**total project cost sheet**

|  |  |  |
| --- | --- | --- |
| No. | List | Cost (Baht) |
| 1. |  |  |
| 2. |  |  |
| 3. |  |  |
| 4. |  |  |
| Total | |  |

The total project cost sheet is composed of the cost that a company has to pay for this project such as cost per month, cost for software license, cost for renting the building ,and cost for equipments.

Example of “ filled total project cost sheet ”

**total project cost sheet**

|  |  |  |
| --- | --- | --- |
| No. | List | Cost (Baht) |
| 1. | Software license   * Apple developer account | 3,465 |
| 2. | Equipments   * Server | 50,000 |
| 3. | Cost per month   * January * Febuary * March | 252,875  252,875  252,875 |
| Total | | 812,090 |

From total project cost sheet, a company will know the total cost that it need to pay for whole project. Therefore, the company can make a quotation to stakeholder and estimate the net profit from this sheet.