***Software***

***Document***

***Design***

***LukMod***

Taveesub Siriphatcharakul 61070507227

Keerati Damrongsak 61070507226

Pakwan Pichai 61070507214

Rachata Chukhlailo 61070507217

Table of Contents

1. Introduction……………………………………………………………………………….. 3

1.1 Purpose Of The Design Document………………………………………….. 3

1.2 Document Conventions....……………………………………………………. 3

1.3 Project Background And Scope……………………………………………... 3

1.4 Document Overview…………………………………………………………... 4

2. Data/Class Design………………………………………………………………………. 4

2.1 Internal Software Data Structure…………………………………………….. 4

2.2 Global Data Structure…………………………………………………………. 4

2.3 Temporary Data Structure…………………………………………………….. 4

2.4 Database Description…………………………………………………………. 4

3. Architectural Design……………………………………………………………………... 8

4. Process Manual Specification………………………………………………………….. 12 4.1 Project Plan And Monitoring Method………………………………………… 12

4.2 Employee Work/Task Assignment Process…………………………………. 15

4.3 Final Project Cost Method With Example………………………………….... 19

Software Design Document

1.Introduction

1.1 Purpose of the design document

            Software Design Document คือเอกสารที่ใช้ในการจัดทำเอกสารเพื่อช่วยในการพัฒนาซอฟต์แวร์โดยมีข้อมูลของการสร้างซอฟต์แวร์ ภายในเอกสารนี้ประกอบไปด้วย ข้อมูลเชิงลึก การวางแผนการทำงาน การออกแบบ Architecture และ System design เช่น Context diagram, Class diagram และรวมไปถึงข้อมูลความต้องการอื่นๆที่รองรับ Project

           จุดประสงค์ของการสร้างเอกสารนี้ขึ้นมาเพื่อให้ทุกคนที่เกี่ยวข้องในด้าน Software Design ไม่ว่าจะเป็น ลูกค้า ผู้พัฒนาซอฟต์แวร์ หรือผู้ที่เกี่ยวข้องอื่นๆ ได้รับรู้และเข้าใจถึงข้อมูลใน Project นี้

1.2 Document Conventions

- Javascript หมายถึง ภาษาคอมพิวเตอร์สำหรับการเขียนโปรแกรม เป็นภาษาสคริปต์ เชิงวัตถุ (ที่เรียกกันว่า “สคริปต์” (script) ซึ่งในการสร้างและพัฒนาเว็บไซต์ (ใช่ร่วมกับ HTML) เพื่อให้เว็บไซต์ของเราดูมีการเคลื่อนไหว สามารถตอบสนองผู้ใช้งานได้มากขึ้น

- HTML (Hypertext Markup Language) หมายถึง ภาษาคอมพิวเตอร์รูปแบบหนึ่ง ที่มีโครงสร้างการเขียนโดยอาศัยตัวกำกับ (Tag) ควบคุมการแสดงผลข้อความ รูปภาพ หรือวัตถุ อื่น ๆ ผ่านโปรแกรมเบราเซอร์ แต่ละ Tag อาจจะมีส่วนขยาย เรียกว่า Attribute สำหรับระบุ หรือควบคุมการแสดงผล

- CSS (Cascading Style Sheets) หมายถึง ภาษาที่ใช้สำหรับตกแต่งเอกสาร HTML/XHTML ให้มีหน้าตา สีสัน ระยะห่าง พื้นหลัง เส้นขอบและอื่นๆ เป็นภาษาหนึ่งในการตกแต่งเว็บไซต์ที่ได้รับความนิยมอย่างแพร่หลาย

- Postgresql หมายถึง เป็นระบบจัดการฐานข้อมูลเชิงวัตถุ (Object Relational Database Management หรือ ORDBMS) เป็นฐานข้อมูลที่สามารถเชื่อมโยงข้อมูลหนึ่งชิ้นไปยังอีกส่วนหนึ่งได้ และสามารถทำงานได้หลายระบบปฏิบัติการรวมถึง Linux, Unix และ Windows

- LukMod หมายถึง ชื่อ web application ที่สร้างมาเพื่อจองโต๊ะและสั่งอาหารภายในร้าน Cafe

1.3 Project Background and Scope

Project นี้เป็นการสร้าง software web application เพื่อใช้ในการจองโต๊ะและสั่งอาหารในร้าน Cafe จากที่ได้เห็นปัญหาที่ลูกค้าที่ต้องการจะเข้าร้าน Cafe เเต่เมื่อเดินทางมาถึงหน้าร้านกลับพบว่าในร้านนั้นไม่มีที่นั่งเหลือพอสำหรับตนเอง ทำให้การมาในครั้งนั้นเสียโอกาสเเละเสียทรัพยากรต่างๆในการเดินทาง ตัว wep application นี้จะช่วยในเรื่องการจองโต๊ะที่นั่งภายในร้านเเละช่วยเช็คยอดโต๊ะที่นั่งที่ยังว่างอยู่ภายในร้าน ทำให้เราผู้ที่ต้องการที่จะเข้าร้าน Cafe ไม่ต้องเดินทางมาอย่างสูญเปล่า เเละเพิ่มความสะดวกในการสั่งอาหารผ่าน application นี้ให้แก่ลูกค้า ทำให้ลูกค้ารู้สึกพึงพอใจเเละมีความสุขที่การมาที่ร้าน Café

1.4 Document Overview

Software Design Document ประกอบไปด้วย 4 ส่วน ดังนี้

1 Introduction

2 Data/Class Design

3 Architectural Design

4 Process Manual Specification

2. Data/Class Design

2.1. Internal Software Data Structure

โครงสร้างข้อมูลที่อยู่ภายในระบบ เมื่อ component เริ่มมีการทำงานจะมีการเข้าถึงฐาน

ข้อมูลและมีการส่งข้อมูลไปยัง component อื่นๆ เป็นการส่งข้อมูลกันภายในซอฟแวร์โดยไม่จำเป็นต้องทำการดึงข้อมูลจากฐานข้อมูลใหม่ทุกครั้ง

2.2. Global Data Structure

โครงสร้างแบบ Global เป็นค่าเดียวกันในทุกๆ component โดยแต่ละ component

สามารถใช้งานร่วมกันได้ เมื่อมีการอัพเดทค่าตัวแปร ค่าใหม่จะถูกบันทึกลงฐานข้อมูลเสมอ

2.3. Temporary Data Structure

โครงสร้างข้อมูลแบบชั่วคราว เมื่อมีการเปลี่ยนแปลงค่าชั่วคราวเพื่อส่งไปยัง

component อื่น หรือกรณีที่มีการคำนวณนั้นจำเป็นต้องมีโครงสร้างข้อมูลแบบชั่วคราวมาเก็บผลลัพธ์ที่เกิดขึ้น

2.4. Database Description

ระบบฐานข้อมูลที่ออกแบบมาให้ซอฟต์แวร์จองโต๊ะ ใช้เก็บข้อมูลและจัดการข้อมูล

ภายในระบบ เพื่อให้สามารถจองโต๊ะผ่านระบบออนไลน์รวมไปถึงการจัดการและจัดเก็บข้อมูลของผู้ใช้ซอฟแวร์ โดยเครื่องหมาย (PK) หมายถึง primary key และ (FK) หมายถึง foreign key

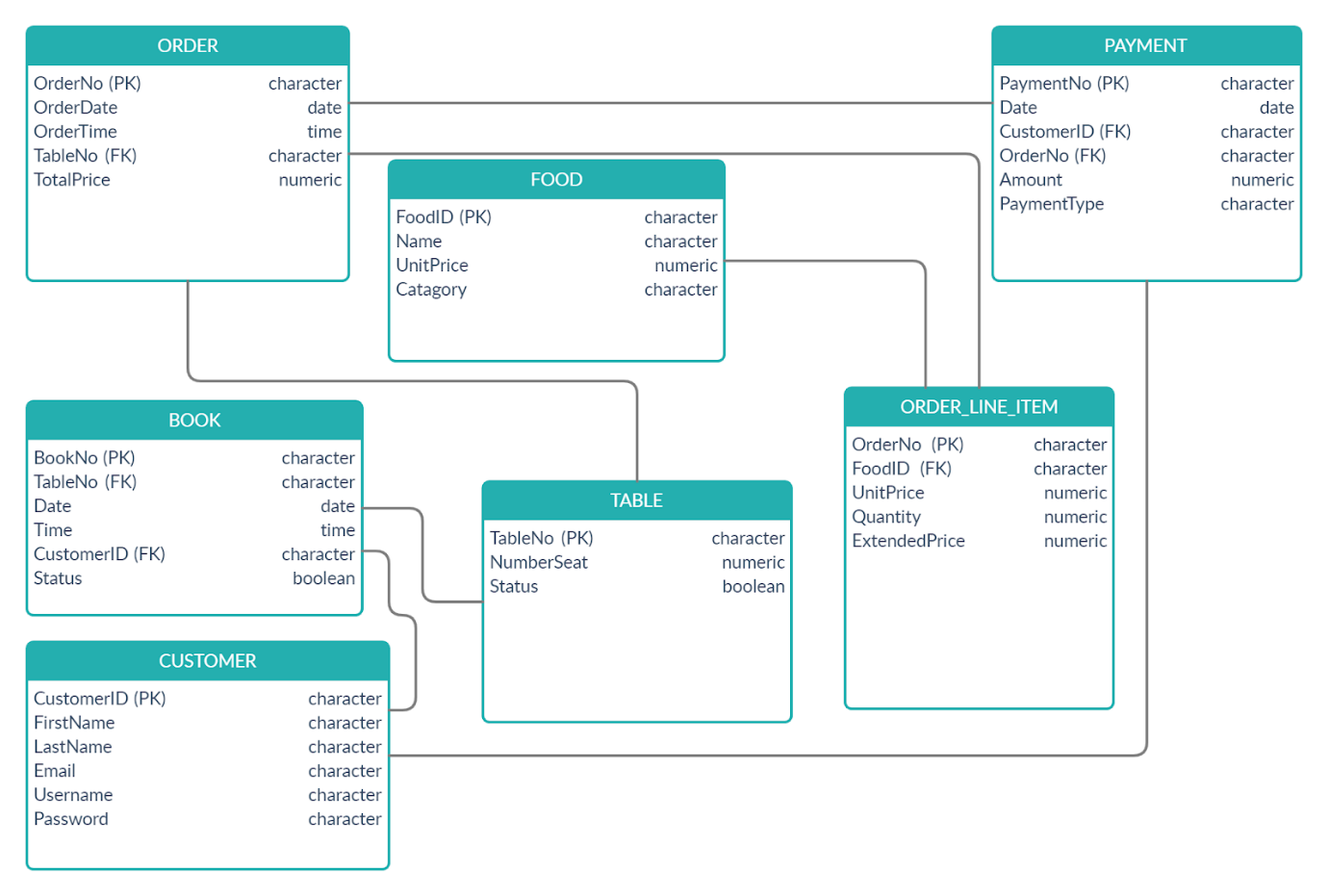


Table Name : CUSTOMER

ใช้ในการเก็บข้อมูลของลูกค้าในการเข้าใช้งาน

|  |  |  |
| --- | --- | --- |
| Attribute | Type | Description |
| CustomerID | character | รหัสระบุตัวตนของลูกค้า |
| FirstName | character | ชื่อลูกค้า |
| LastName | character | นามสกุลลูกค้า |
| Email | character | Email ลูกค้า |
| Username | character | Username สำหรับเข้าใช้งาน |
| Password | character | Password สำหรับเข้าใช้งาน |

Table Name : BOOK

ใช้ในการเก็บข้อมูลการจองโต๊ะของลูกต้า

|  |  |  |
| --- | --- | --- |
| Attribute | Type | Description |
| BookNo | character | เลขที่การจองโต๊ะ |
| TableNo | character | เลขที่โต๊ะที่จอง |
| Date | date | วันที่จอง |
| Time | time | เวลาที่จอง |
| CustomerID | character | รหัสระบุตัวตนลูกค้า |
| Status | boolean | สถานะยืนยันการจองโต๊ะ |

Table Name : TABLE

ใช้ในการเก็บข้อมูลของโต๊ะ

|  |  |  |
| --- | --- | --- |
| Attribute | Type | Description |
| TableNo | character | เลขที่โต๊ะ |
| NumberSeat | numeric | จำนวนที่นั่ง |
| Status | boolean | สถานะการว่างของโต๊ะ |

Table Name : ORDER

ใช้ในการเก็บข้อมูลการสั่งอาหารของลูกค้าเเต่ละโต๊ะ

|  |  |  |
| --- | --- | --- |
| Attribute | Type | Description |
| OrderNo | character | เลขที่การสั่ง |
| OrderDate | character | วันที่สั่ง |
| OrderTime | date | เวลาที่สั่ง |
| TableNo | time | เลขที่โต๊ะ |
| TotalPrice | numeric | ราคารวม |

Table Name : ORDER\_LINE\_ITEM

ใช้ในการเก็บข้อมูลการอาหารที่สั่งเเละคำนวนราคา

|  |  |  |
| --- | --- | --- |
| Attribute | Type | Description |
| OrderNo | character | เลขที่การสั่ง |
| FoodID | character | รหัสอาหาร |
| UnitPrice | time | ราคาต่อหนึ่งหน่วย |
| Quantity | numeric | จำนวนที่สั่ง |
| ExtendedPrice | numeric | ราคารวม |

Table Name : FOOD

ใช้ในการเก็บข้อมูลอาหาร

|  |  |  |
| --- | --- | --- |
| Attribute | Type | Description |
| FoodID | character | รหัสอาหาร |
| Name | character | ชื่ออาหาร |
| UnitPrice | numeric | ราคาต่อหนึ่งหน่วย |
| Catagory | character | ประเภทอาหาร |

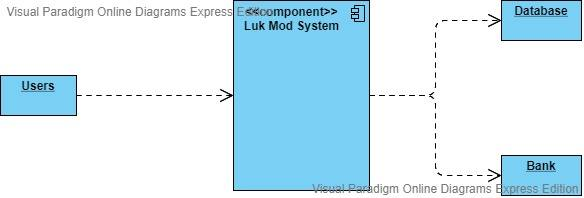
Table Name : PAYMENT

ใช้ในการเก็บข้อมูลการชำระเงิน

|  |  |  |
| --- | --- | --- |
| Attribute | Type | Description |
| PaymentNo | character | เลขที่ชำระเงิน |
| Date | date | วันที่ชำระ |
| CustomerID | charecter | รหัสประจำตัวลูกค้า |
| OrderNo | character | เลขที่การสั่ง |
| PaymentType | character | ประเภทการชำระเงิน |
| Amount | numeric | จำนวนเงินที่ชำระ |

3. Architectural Design

* Context Diagram



ระบบ Luk Mod System มีส่วนติดต่อกับระบบอื่นแยกตามประเภทได้แก่

* Actor
  + Users ผู้ใช้งานซอฟแวร์สำหรับจองโต๊ะทานอาหาร

      - Peers

    - Database ฐานข้อมูลเก็บข้อมูลลูกค้าและร้านอาหาร

    - Bank ระบบธนาคารสำหรับการจ่ายเงินผ่านธนาคาร

* Use Case Diagram



**Name :** Book table

**Actors :** ผู้ใช้งาน

**Goal :** ผู้ใช้งานจองโต๊ะ

**Precondition :** ผู้ใช้งานจำเป็นจะต้องล็อกอินเข้สสู่ระบบก่อน

**Main success scenario**

1. ผู้ใช้งานกรอกรายละเอียด วัน/เดือน/ปี เเละ เวลา
2. ระบบเเสดงโต๊ะที่ว่างที่สามารถจองได้
3. ผู้ใช้งานเลือกโต๊ะที่ว่าง

**Extensions (a)**

**2.1a.**  ระบบหาโต๊ะที่ว่างไม่เจอ

**2.2a.**  ระบบจะกลับไปที่หน้าเลือกกรอกรายละเอียด วันที่เเละเวลาใหม่

**Extension (b)**

**2.1b.** ผู้ใช้งานไม่ต้องการโต๊ะที่เลือก

**2.2b.** ผู้ใช้งานกลับไปที่หน้าเลือกกรอกรายละเอียด วันที่เเละเวลาใหม่

**Postconditions** : None

**Name :** Select food Menu

**Actors :** ผู้ใช้งาน

**Goal :** ผู้ใช้งานสั่งอาหาร

**Precondition :** ผู้ใช้งานจำเป็นจะต้องล็อกอินเข้สสู่ระบบก่อน

**Main success scenario**

1. ผู้ใช้งานต้องการเเสดงเมนูอาหาร
2. ระบบเเสดงรายการเมนูอาหาร
3. ผู้ใช้งานเลือกเมนูอาหาร

**Extension (a)**

**3a.** ผู้ใช้งานไม่ต้องการเมนูที่เลือกไปเเล้ว

**3a.** ผู้ใช้งานกลับไปที่หน้าเลือกเมนูอาหารใหม่

**Postconditions** : None

**Name :** Payment

**Actor :** ผู้ใช้งาน

**Goal :** ผู้ใช้งานชำระเงิน

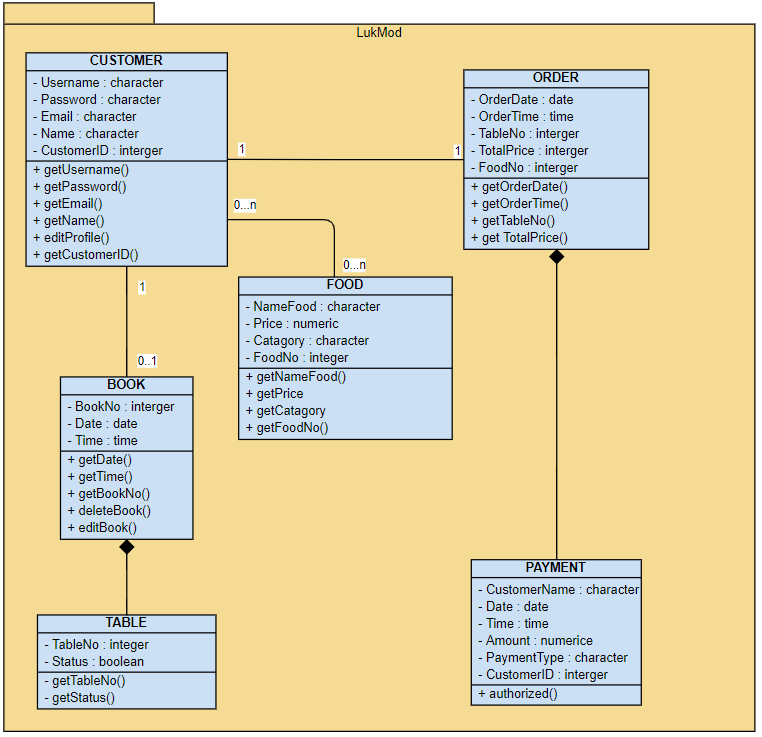
**Precondition :** None

**Main Success Scenario**

1. ผู้ใช้งานต้องการที่จะชำระเงิน
2. ระบบเเสดงจำนวนเงินที่จะต้องชำระ
3. ผู้ใช้งานชำระเงิน โดยเลือกวิธีการชำระ

**Postconditions :** None

* Class diagram



4. Process Manual Specification

4.1. Project Plan and Monitoring Method

การวางแผนการทำงาน โดยการแบ่งหน้าที่ของสมาชิกแต่ละคนอย่างชัดเจนเพื่อสร้างเข้าใจในการทำงานร่วมกันของทีม รวมไปถึงการติดตามความคืบหน้างานในส่วนย่อยๆที่ได้มอบหมายไปแก่สมาชิกแต่ละคน ซึ่งหากเกิดข้อผิดพลาดหรืองานล่าช้าก็สามารถระบุผู้รับผิดชอบและเหตุผลของปัญหานั้นได้และสามารถนำมาวางแผนในการจัดการได้ต่อไป

Role and Responsibility

|  |  |  |
| --- | --- | --- |
| Position | Name | Email |
| Project Manager | Mr.Taveesub Siriphatcharakul | Taveesub.ssub@mail.kmutt.ac.th |
| Designer | Ms.Pakwan Pichai | pakwan.p@mail.kmutt.ac.th |
| Tester | Mr. Keerati Damrongsak | Keerati.krt@mail.kmutt.ac.th |
| Developer | Mr. Rachata Chukhlailo | rachata.c@mail.kmutt.ac.th |

**GranttChart**

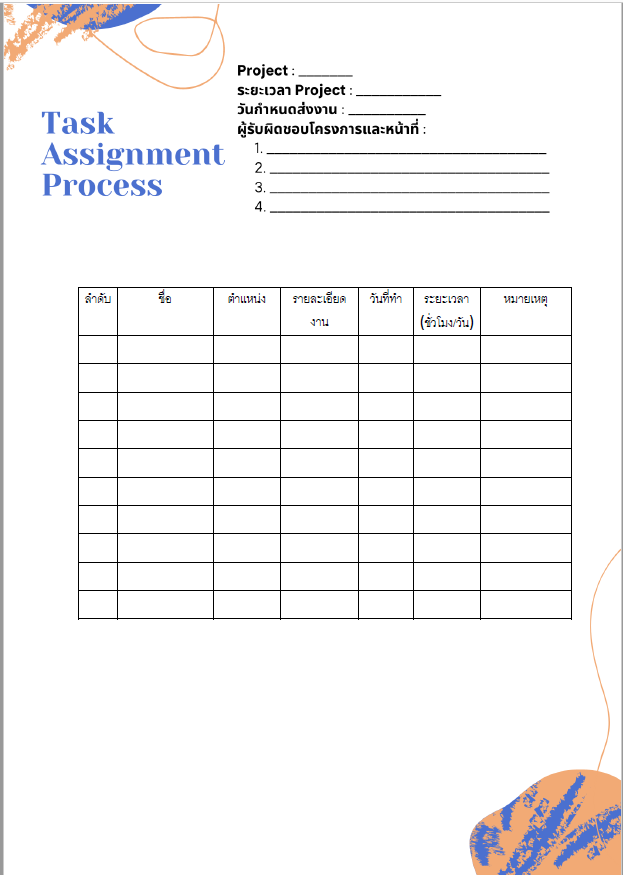
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ActivityMonth | August | | | | September | | | | October | | | | November | | | | สิ่งที่ส่งมอบ | ผู้รับผิดชอบ |
| Week | **1** | **2** | **3** | **4** | **1** | **2** | **3** | **4** | **1** | **2** | **3** | **4** | **1** | **2** | **3** | **4** |
| **Requirement analysis** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Kickoff meeting |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ข้อมูลในการติดต่อ | ทุกคน |
| Requirement document |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ทุกคน |
| Requirement document review and approval |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | เอกสารความต้องการของลูกค้า |  |
| **Software design** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Design prototype and approval |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ออกเเบบตามความต้องการของลูกค้า | ทุกคน |
| **Implementation** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Implement |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | web application ตัวต้นเเบบ | ทุกคน |
| **Integration testing** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Debug program |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ทุกคน |
| **Data migration** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Data migration |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | การติดตั้ง application เเละโอนข้อมูล | ทุกคน |
| **Training and document** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Design document |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | คู่มือการใช้งาน | ทุกคน |
| **User acceptance testing** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Design approve |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ซอฟเเวร์ที่ปรับปรุงเเล้ว | ทุกคน |
| **Production run** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Production run |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ซอฟเเวร์ตัวจริง | ทุกคน |

ตารางสรุปแผนงาน

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| กิจกรรม | วันเริ่ม | วันสิ้นสุด | ระยะเวลา(วัน) | ผู้รับผิดชอบ |
| **Requirement analysis** |  |  |  |  |
| Kickoff meeting | 19/8/2563 | 19/8/2563 | 1 | ทุกคน |
| Requirement document | 20/8/2563 | 24/8/2563 | 5 | ทุกคน |
| Requirement document review and approval | 26/8/2563 | 7/9/2563 | 13 | ทุกคน |
| **Software design** |  |  |  |  |
| Design prototype and approval | 7/9/2563 | 22/9/2563 | 16 | ทุกคน |
| Implementation |  |  |  |  |
| Implement | 5/10/2563 | 23/10/2563 | 19 | ทุกคน |
| **Integration testing** |  |  |  |  |
| Debug program | 22/10/2563 | 29/10/2563 | 8 | ทุกคน |
| **Data migration** |  |  |  |  |
| Data migration | 2/11/2563 | 5/11/2563 | 4 | ทุกคน |
| **Training and document** |  |  |  |  |
| Design document | 6/11/2563 | 13/11/2563 | 8 | ทุกคน |
| **User acceptance testing** |  |  |  |  |
| Design approve | 15/11/2563 | 18/11/2563 | 4 | ทุกคน |
| **Production run** |  |  |  |  |
| Production run | 21/11/2563 | 23/11/2563 | 3 | ทุกคน |

4.2. Employee Work/Task Assignment Process

**แบบฟอร์มแบบบันทึกการปฎิบัติงาน**



แบบฟอร์มแบบบันทึกการปฎิบัติงาน ประกอบไปด้วย 2 ส่วน

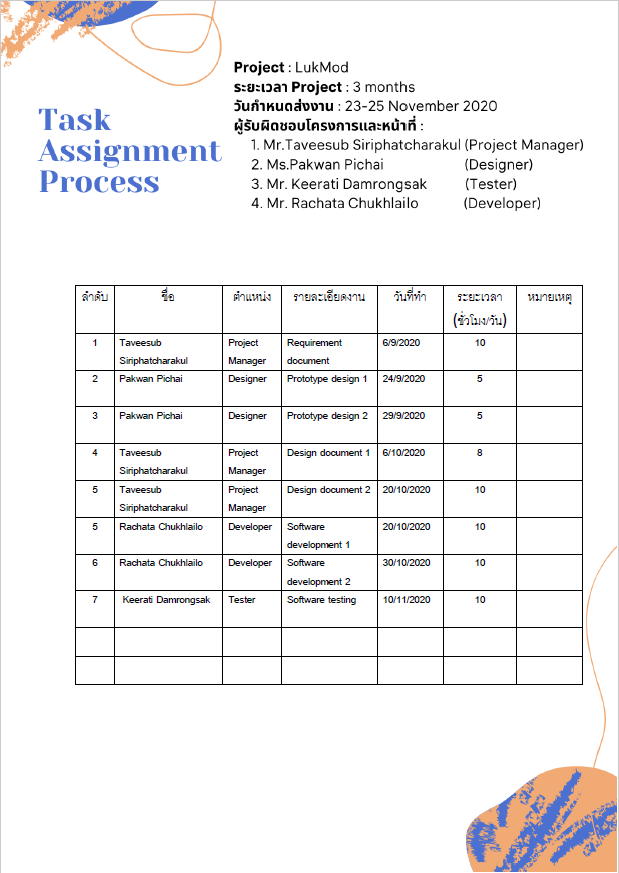
ส่วนที่ 1 ข้อมูล Project

* ชื่อ Project
* ระยะเวลา Project
* วันกำหนดส่งงาน
* ผู้รับผิดชอบโครงการและหน้าที่

ส่วนที่ 2 ข้อมูลปฎิบัติงาน

* ลำดับ
* ชื่อ
* ตำแหน่ง
* รายละเอียดงาน
* วันที่ทำ
* ระยะเวลา (ชั่วโมง/วัน)
* หมายเหตุ

ตัวอย่างแบบฟอร์มแบบบันทึกการปฎิบัติงาน



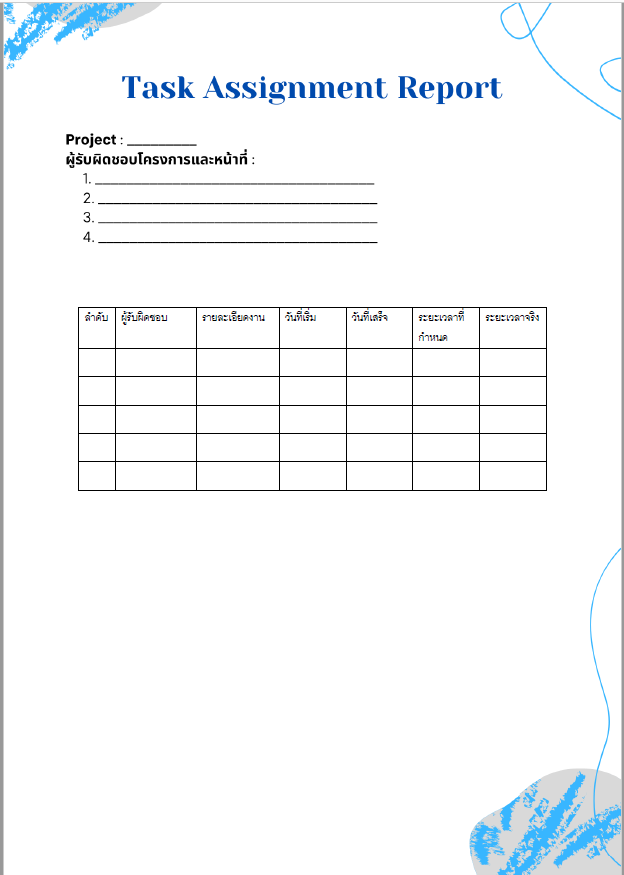
**แบบฟอร์มแบบรายงานการปฎิบัติงาน**

แบบฟอร์มแบบรายงานการปฎิบัติงานประกอบไปด้วย 2 ส่วน ดังนี้

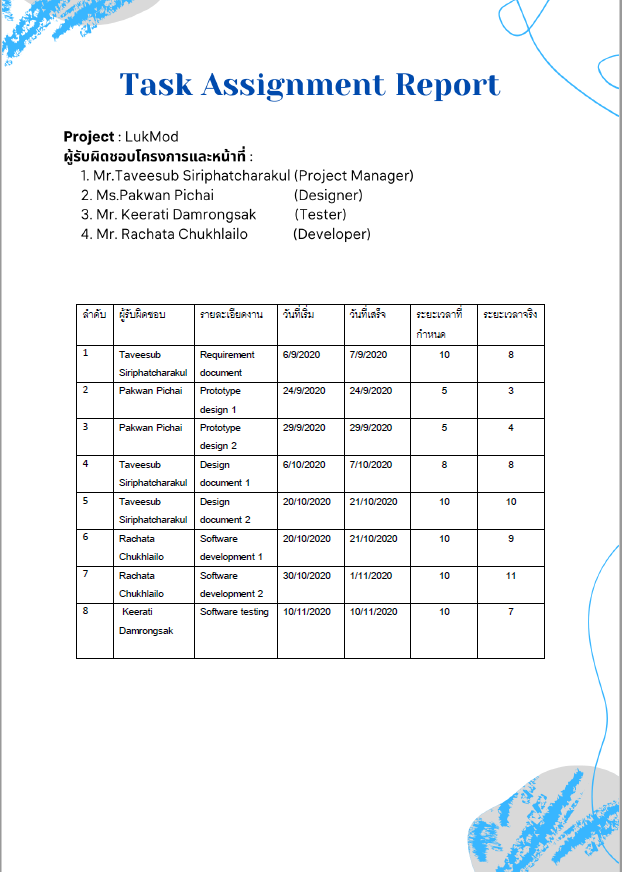
ส่วนที่ 1

* ชื่อ Project
* ผู้รับผิดชอบโครงการและหน้าที่

ส่วนที่ 2

* ลำดับ
* ชื่อผู้รับผิดชอบ
* รายละเอียดงาน
* วันที่เริ่ม
* วันที่เสร็จ
* ระยะเวลาที่กำหนด
* ระยะเวลาจริง

ตัวอย่างแบบฟอร์มแบบรายงานการปฎิบัติงาน



4.3. Final Project Cost Method with Example

แบบบันทึกทึกชั่วโมงการปฏิบัติงาน ประกอบไปด้วย 2 ส่วน

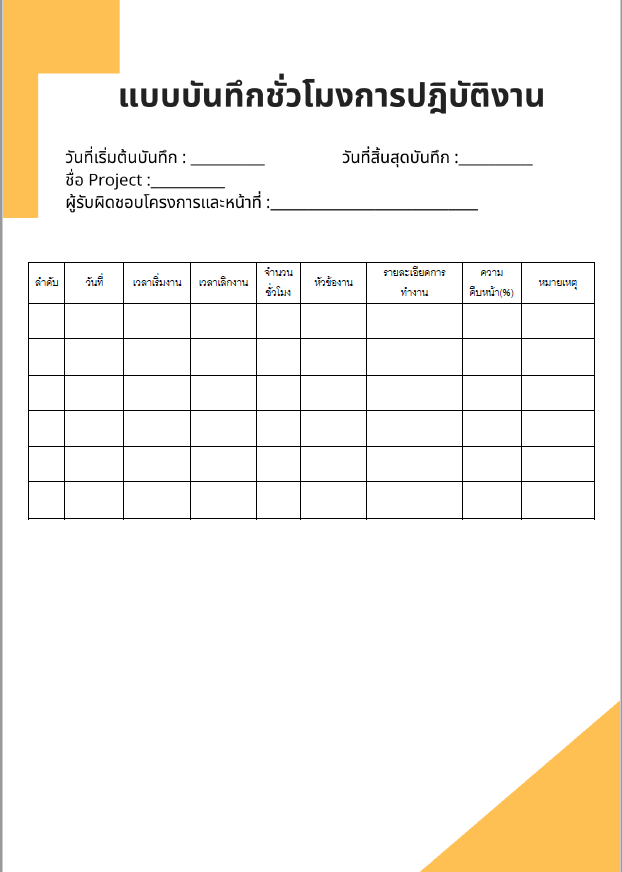
ส่วนที่ 1 ข้อมูลทั่วไป

* วันที่เริ่มต้นบันทึก
* วันสิ้นสุดบันทึก
* ชื่อโครงการ
* ชื่อพนักงาน
* หน้าที่รับผิดชอบ

ส่วนที่ 2 ตารางบันทึกการทำงาน

* ลำดับ
* วันที่
* เวลาเริ่มงาน
* เวลาเลิกงาน
* จำนวนชั่วโมงการทำงาน
* หัวข้องานที่ทำ
* รายละเอียดการทำงาน
* ความคืบหน้าของงาน(%)
* หมายเหตุ

**แบบฟอร์ม แบบบันทึกชั่วโมงการปฏิบัติการ**

****

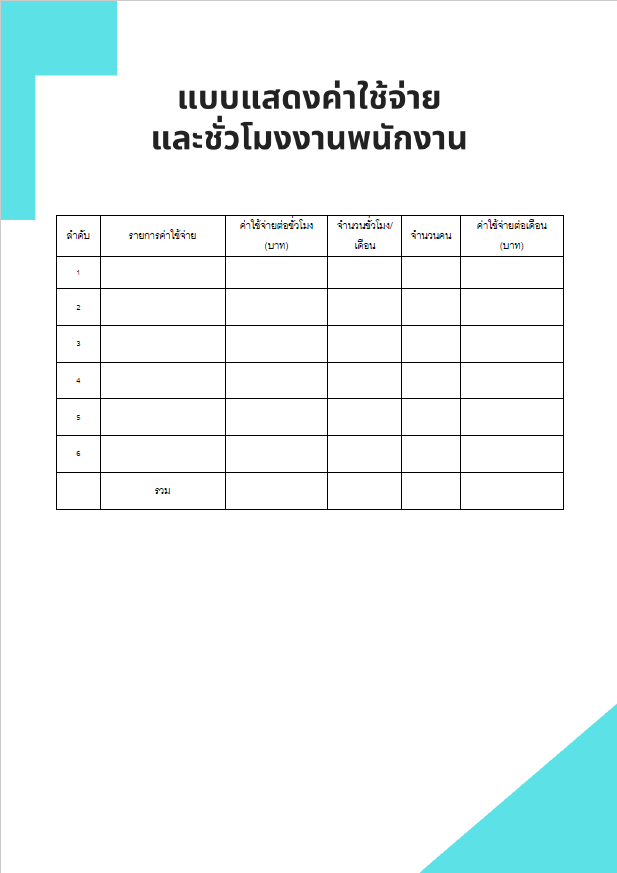
**ตัวอย่างแบบฟอร์มแบบบันทึกชั่วโมงการปฏิบัติการ**



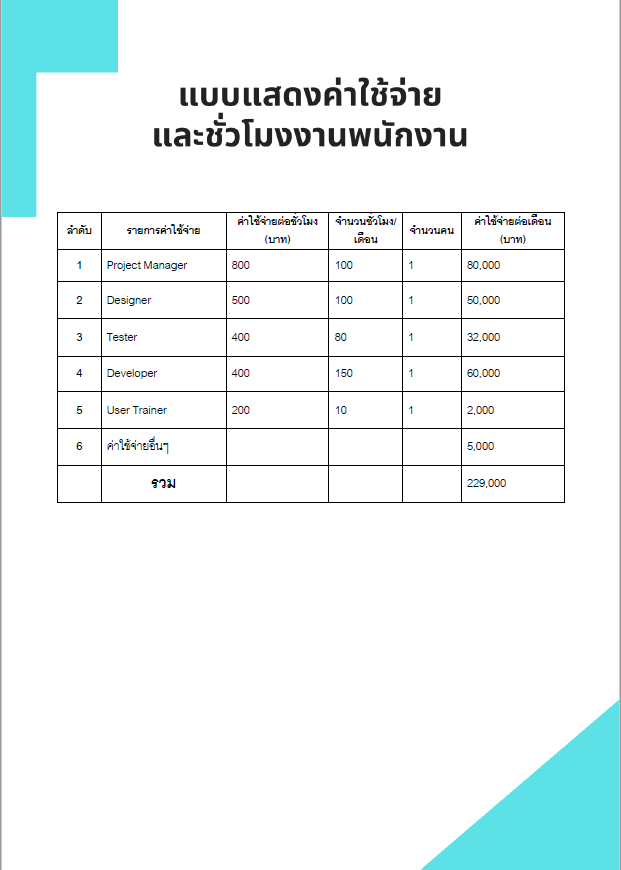
แบบแสดงค่าใช้จ่ายและชั่วโมงงานของพนักงาน มีดังนี้

* ลำดับ
* รายการค่าใช้จ่าย
* ค่าใช้จ่ายต่อชั่วโมง
* จำนวนชั่วโมงต่อเดือน
* จำนวนคน
* รวมค่าใช้จ่ายต่อเดือน

**แบบฟอร์ม แสดงค่าใช้จ่ายของพนักงาน (ต่อเดือน)**

****

**ตัวอย่างแบบฟอร์ม แสดงค่าใช้จ่ายของพนักงาน (ต่อเดือน)**



ค่าใช้จ่ายสำหรับลิขสิทธ์ซอฟท์แวร์

|  |  |
| --- | --- |
| รายการ | ราคา(บาท) |
|  |  |

ค่าใช้จ่ายสำหรับอุปกรณ์

|  |  |
| --- | --- |
| รายการ | ราคา(บาท) |
|  |  |

ค่าใช้จ่ายต่อเดือน

|  |  |
| --- | --- |
| รายการ | ราคา(บาท) |
|  |  |