



รายงาน

ระบบจองโรงแรม (Hotel reservation)

จัดทำโดย

นายกฤษฎา ฉายแก้ว	64070507201
นายธนภัทร ไทยกิ่ง	64070507207
นางสาวประภัสสร แสงรอด	64070507210
นายศุภณัฐ ขาวงาม	64070507216
นายอัครินทร์ รุ่งสว่าง	64070507222

นำเสนอ

Dr.Khajonpong Akkarajitsakul

Dr.Phond Phunchongharn

รายงานนี้เป็นส่วนหนึ่งของวิชา

CPE 334 Software Engineering

มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี

คณะวิศวกรรมคอมพิวเตอร์

Introduction

เอกสารฉบับนี้เป็น Software design document ที่จัดทำขึ้นเพื่อเป็นเอกสารการออกแบบเว็บไซต์สำหรับจองโรงแรม(Hotel Reservation) ซึ่งภายในเอกสารประกอบไปด้วย Requirement, Scope, Design และ การใช้ Iterative Waterfall ในการดำเนินงาน

โดยเอกสารฉบับนี้จะช่วยสื่อสารให้เกิดความเข้าใจที่ตรงกันระหว่างผู้พัฒนาซอฟต์แวร์และผู้ที่มีส่วนเกี่ยวข้องในด้านต่างๆ ข้อมูลภายในเอกสารฉบับนี้คาดว่าจะมีประโยชน์ในการวางแผน ปรับปรุง และแก้ไขซอฟต์แวร์ต่อไปในอนาคต

สารบัญ

เนื้อหา	หน้า
Introduction.....	A
Requirements.....	1
Scope.....	2
Design	
- Use case diagram.....	3
- Use case.....	4
- Navigation map.....	16
- Data design.....	17
- Architectural design.....	18
- User interface.....	20
Process.....	24

Requirements

1.การจองห้องพัก:

- ผู้ใช้สามารถดูรายละเอียดของห้องพักที่ว่างในวันที่ต้องการเข้าพัก
- ผู้ใช้สามารถเลือกประเภทของห้องพัก (เช่น ห้องเตียงเดี่ยว, ห้องเตียงคู่) และจำนวนผู้เข้าพัก
- ระบบต้องสามารถแสดงราคาของห้องพักแต่ละคืนให้ผู้ใช้เห็น

2.การจัดการการจอง:

- ผู้ใช้สามารถเลือกวันที่เข้าพักและวันที่ออกจากโรงแรม
- ระบบต้องแสดงสถานะการจองให้ผู้ใช้เห็นผ่านหน้าเว็บได้

3.การจัดการบัญชีผู้ใช้:

- ผู้ใช้สามารถสร้างบัญชีผู้ใช้และเข้าสู่ระบบด้วยชื่อผู้ใช้และรหัสผ่านได้
- ผู้ใช้ต้องสามารถออกจากระบบได้

4.การสนับสนุนลูกค้า:

- ผู้ใช้สามารถติดต่อฝ่ายบริการลูกค้าผ่านช่องทางออนไลน์ เช่น เฟสบุ๊ก, ไลน์ หรือ อินสตาแกรม เป็นต้น

5.การจัดการการจองของพนักงาน:

- พนักงานต้องสามารถยืนยันและยกเลิกการจองของลูกค้าได้
- พนักงานสามารถดูได้ว่าห้องไหนมีคนจอง หรือไม่มีคนจองได้

Scope

การจัดการการเงิน:

- ระบบจะไม่เกี่ยวข้องกับข้อมูลทางการเงินของลูกค้า

การจัดการความปลอดภัยของเครือข่าย:

- ระบบจะไม่รับผิดชอบในการประกันความปลอดภัยของเครือข่ายที่ใช้ในการเชื่อมต่อ

การดูแลรักษาอุปกรณ์และโปรแกรมคอมพิวเตอร์:

- ระบบจะไม่รับผิดชอบในการดูแลรักษาหรือซ่อมแซมอุปกรณ์คอมพิวเตอร์หรือโปรแกรมที่ลูกค้าใช้

การบริหารจัดการโรงแรมแบบรายละเอียด:

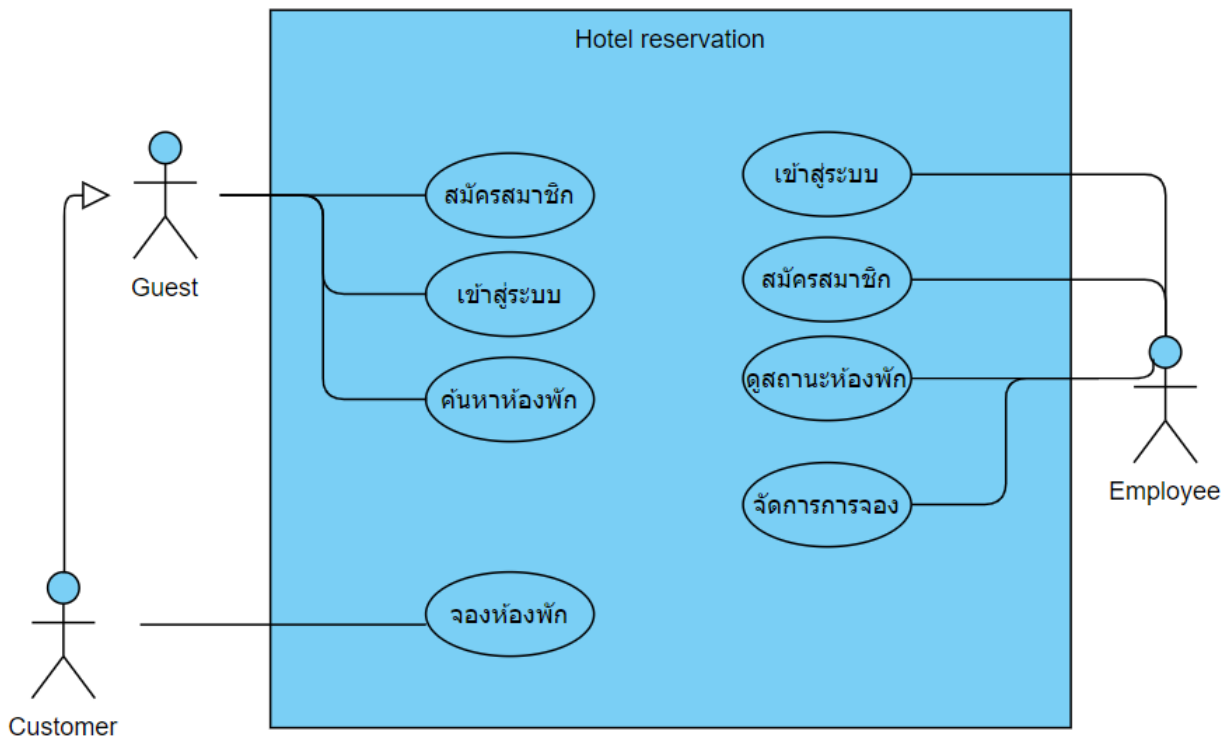
- ระบบจะไม่รวมถึงการบริหารจัดการโรงแรมเชิงละเอียดเช่น การจัดการสต็อกห้องพักหรือการจัดการบุคลากร

การให้คำปรึกษาด้านการท่องเที่ยว:

- ระบบจะไม่ให้คำปรึกษาหรือแนะนำเกี่ยวกับสถานที่ท่องเที่ยวหรือกิจกรรมท่องเที่ยว

Design

- Use case diagram



Use case: ลงทะเบียน

Actor: ผู้ใช้ที่ยังไม่ได้ลงทะเบียน

Goal: ผู้ใช้ลงทะเบียนสำเร็จ

Precondition: ผู้ใช้เข้าสู่หน้าแรกของระบบจองโรงแรม

Main success scenario

1. ผู้ใช้เลือกไปที่ปุ่มลงทะเบียน
2. ระบบแสดงฟอร์มการลงทะเบียน
3. ผู้ใช้กรอกข้อมูลต่างๆจนครบแล้วกดยืนยัน
4. ระบบแจ้งเตือนว่าลงทะเบียนสำเร็จ
5. ผู้ใช้กด “OK”
6. ระบบพาผู้ใช้ไปยังหน้าค้นหาห้องพัก

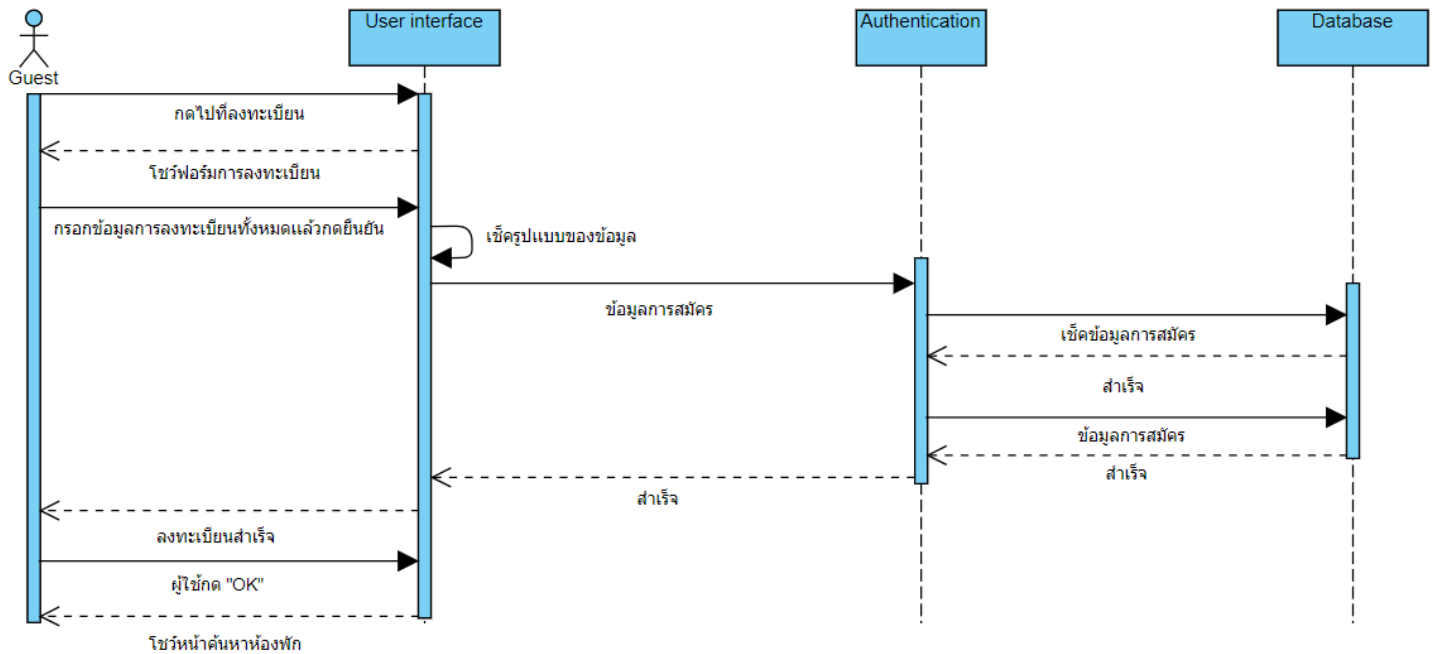
Extension (a):

- 3a ผู้ใช้กรอกข้อมูลรหัสบัตรประชาชน อีเมล หรือ ชื่อผู้ใช้ที่ซ้ำกับผู้ใช้คนอื่นแล้วกดยืนยัน
- 4a ระบบแจ้งเตือนว่าผู้ใช้กรอกข้อมูลซ้ำ กลับไปทำขั้นตอนที่ 3

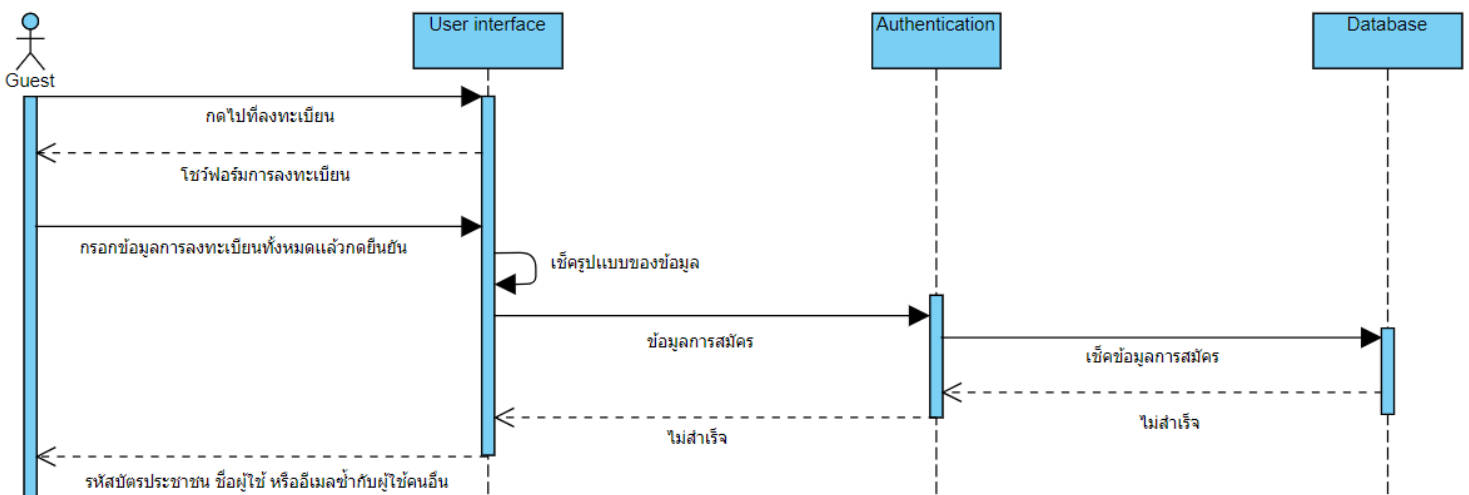
Extension (b):

- 3b ผู้ใช้กรอกข้อมูลที่ไม่ถูกต้องตามรูปแบบที่กำหนดไว้
- 4b ระบบแจ้งเตือนว่าข้อมูลที่กรอกไม่ตรงตามรูปแบบที่กำหนด กลับไปทำขั้นตอนที่ 3

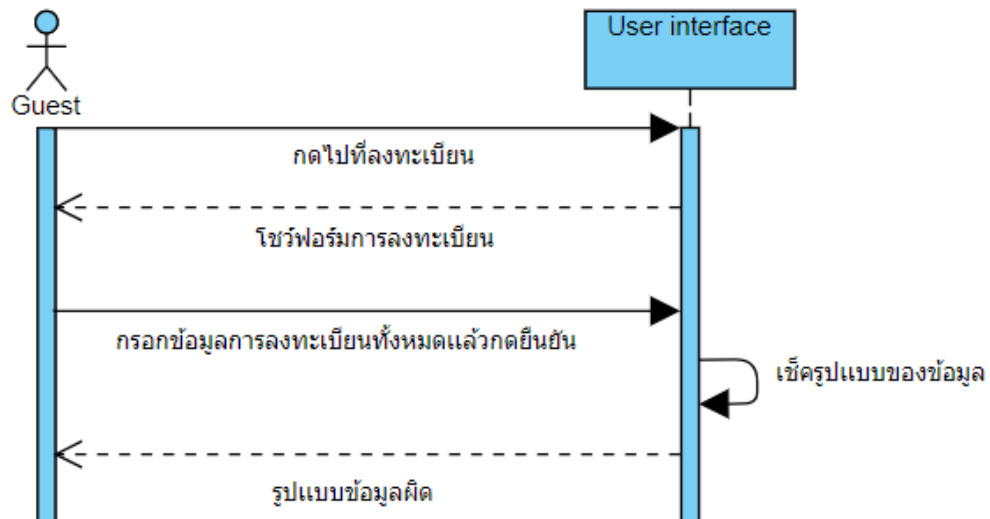
Main success scenario



Extension (a)



Extension (b)



Use case: เข้าสู่ระบบ

Actor: ผู้ใช้ที่ลงทะเบียนแล้ว

Goal: ผู้ใช้เข้าสู่ระบบจองโรงแรม

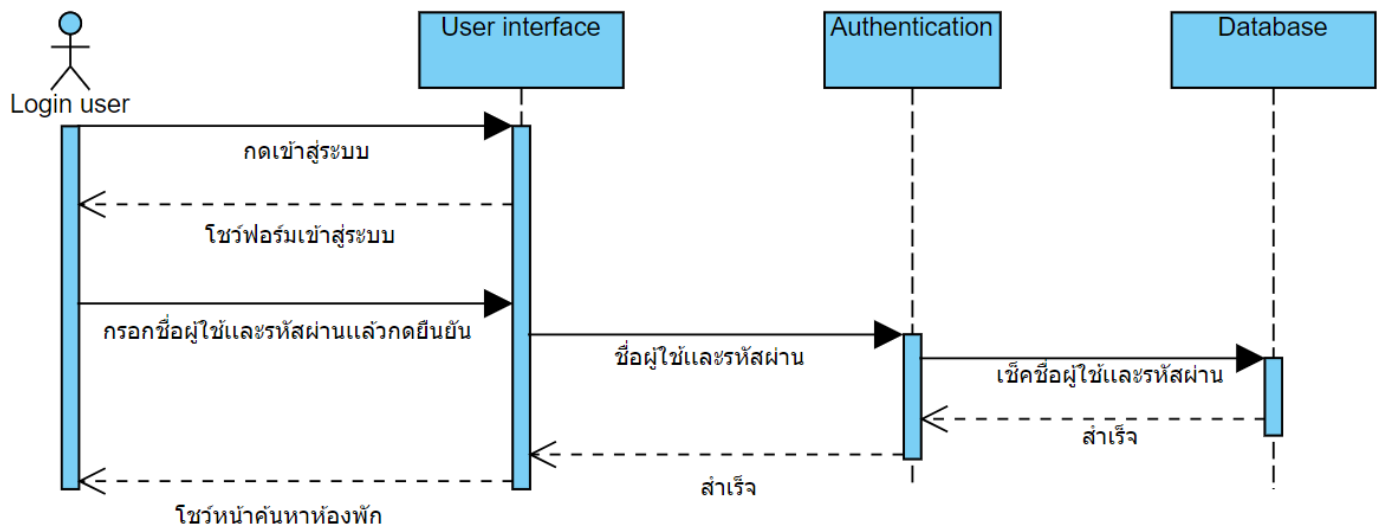
Main success scenario

1. ผู้ใช้เลือกเข้าสู่ระบบ
2. ระบบแสดงฟอร์มการเข้าสู่ระบบ
3. ผู้ใช้เข้าสู่ระบบด้วยชื่อผู้ใช้และรหัสผ่านแล้วกดยืนยัน
4. ระบบพาผู้ใช้ไปยังหน้าค้นหาห้องพักโดยอัตโนมัติ

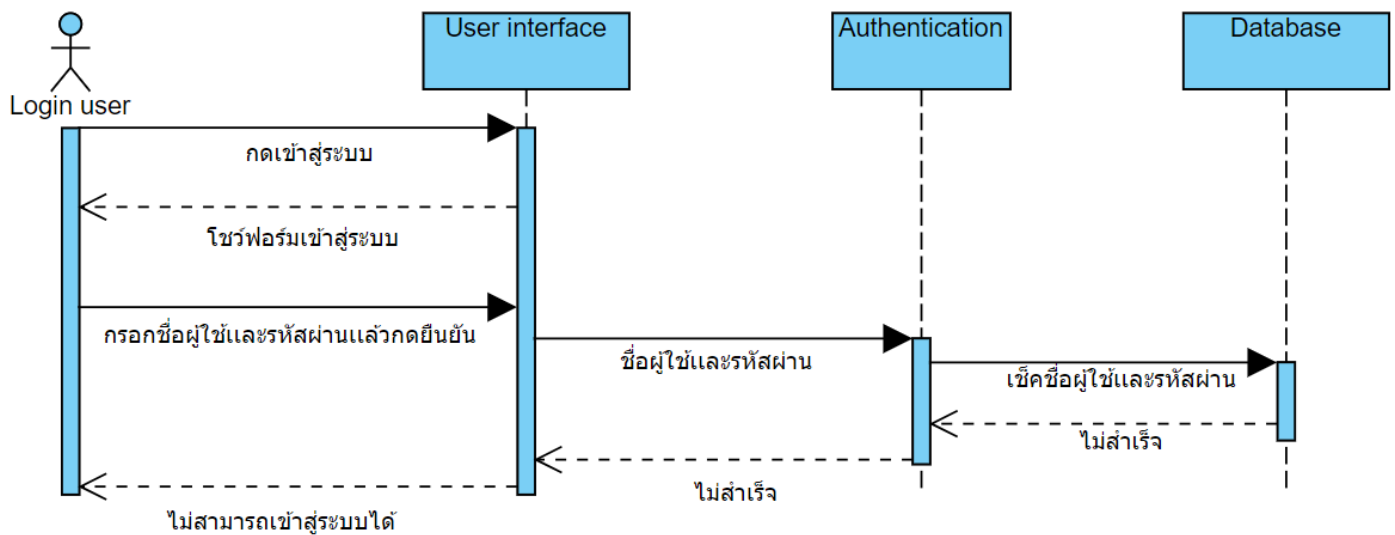
Extension (a):

- 3a ผู้ใช้กรอกชื่อผู้ใช้และรหัสผ่านผิดแล้วกดยืนยัน
- 4a ระบบแจ้งเตือนว่าชื่อผู้ใช้และรหัสผ่านผิด กลับไปทำขั้นตอนที่ 3

Main success scenario



Extension (a)



Use case: ค้นหาห้องพัก

Actor: ลูกค้า, ลูกค้าที่ยังไม่ลงทะเบียน

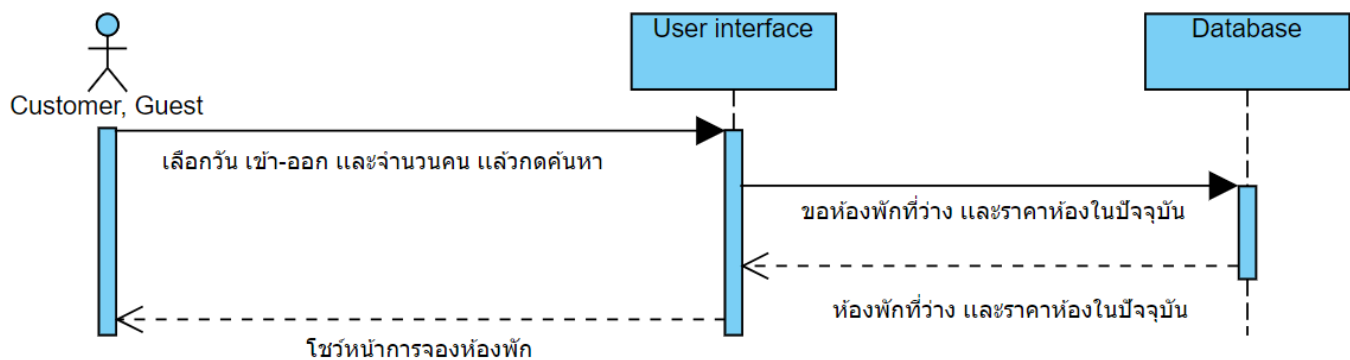
Goal: ลูกค้าอยู่หน้าการจอง

Precondition: ลูกค้าเข้าสู่ระบบ

Main success scenario

- 1.ลูกค้าเลือกวันที่จะ เข้า-ออก และเลือกจำนวนคนที่จะเข้าพัก แล้วกดค้นหา
- 2.ระบบพาไปหน้าการจอง

Main success scenario



Use case: ดูสถานะห้องพัก

Actor: พนักงาน

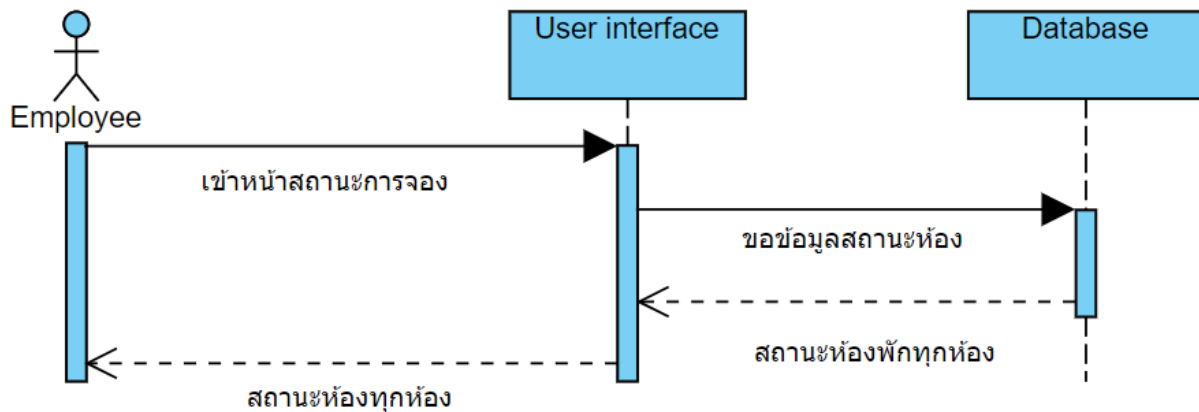
Goal: พนักงานอยู่หน้าสถานะห้องพัก

Precondition: พนักงานเข้าสู่ระบบ

Main success scenario

- 1.พนักงานกดปุ่มเข้าไปที่หน้าสถานะห้อง
- 2.ระบบโชว์สถานะห้องของแต่ละห้อง

Main success scenario



Use case: การจองห้องพัก

Actor: ลูกค้า

Goal: ลูกค้าจองห้องพักสำเร็จ

Precondition: ลูกค้าอยู่หน้าการจอง

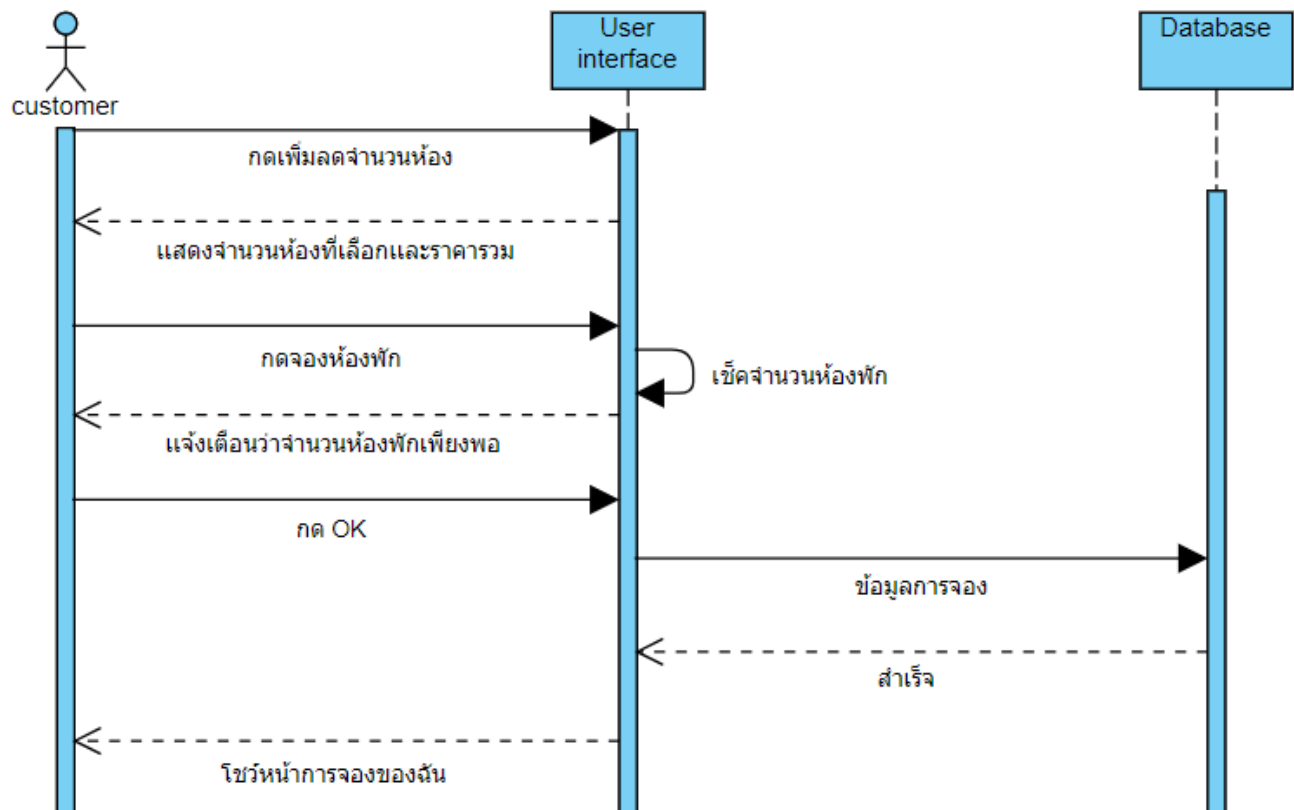
Main success scenario

- 1.ลูกค้ากดปุ่มเพิ่มลดห้อง
- 2.ลูกค้ากดปุ่ม “จองห้อง” เพื่อทำการจอง
- 3.ระบบแจ้งเตือนว่าจองสำเร็จ
- 4.ผู้ใช้กด “OK”
- 5.ระบบพาผู้ใช้ไปยังหน้าการจองของฉัน

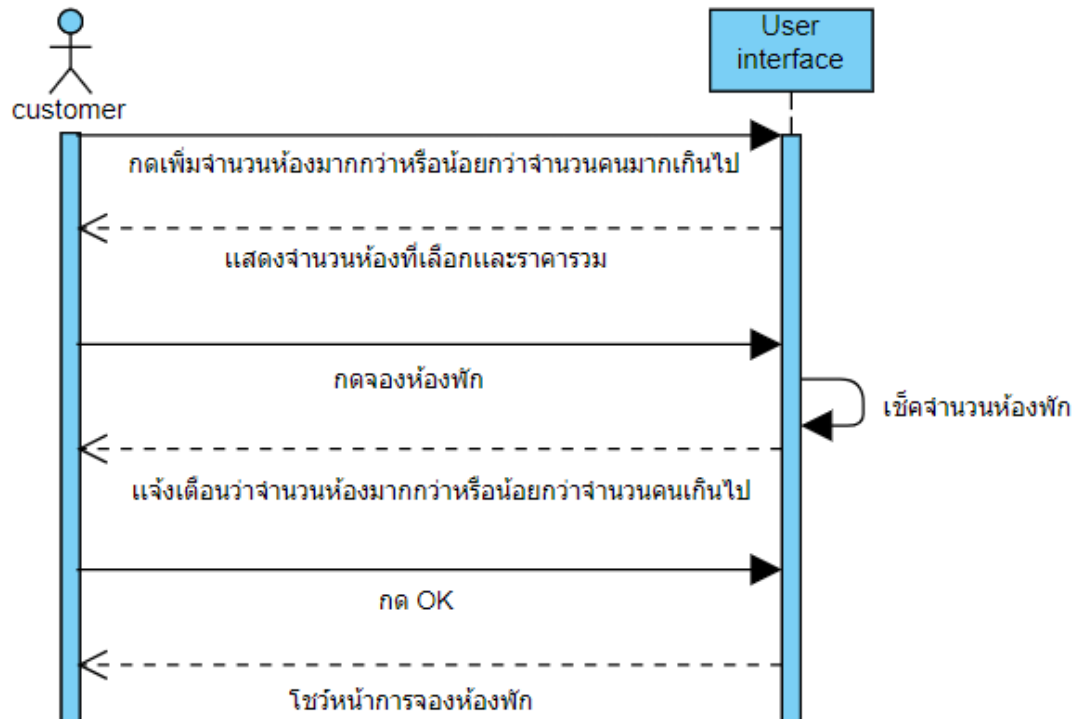
Extension (a):

- 1a ลูกค้ากดเพิ่มห้องมากกว่าหรือน้อยกว่าจำนวนคนที่เข้าพักมากเกินไป
- 2a ลูกค้ากดปุ่มจองห้องพัก
- 3a ระบบแจ้งเตือนว่าไม่สามารถจองได้ กลับไปทำขั้นตอนที่ 1

Main success scenario



Extension (a)



Use case: จัดการการจองห้องพัก

Actor: พนักงาน

Goal: พนักงานยืนยันการจองของลูกค้าสำเร็จ

Precondition: พนักงานเข้าสู่ระบบ

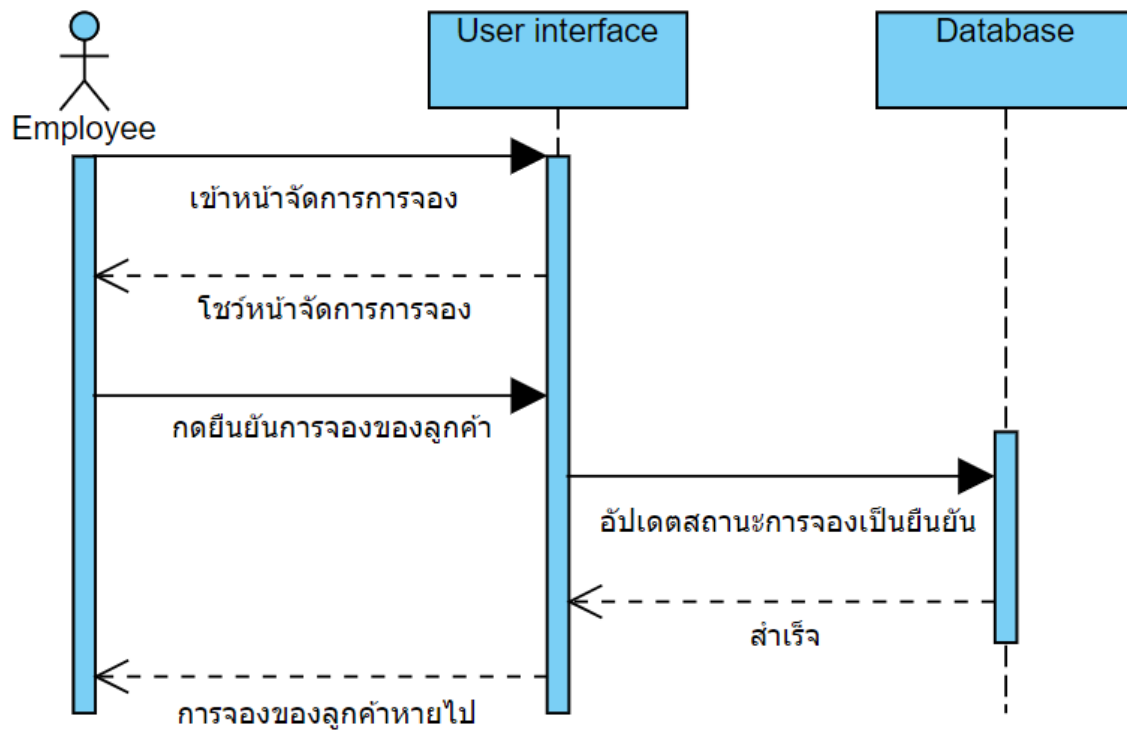
Main success scenario

- 1.พนักงานกดปุ่มเข้าไปที่หน้าการจอง
- 2.ระบบแสดงหน้าการจอง
- 3.พนักงานกดปุ่ม "ยืนยัน" เพื่อยืนยันการจอง
- 4.การจองของลูกค้าที่ถูกยืนยันจะหายไปจากหน้าจอ

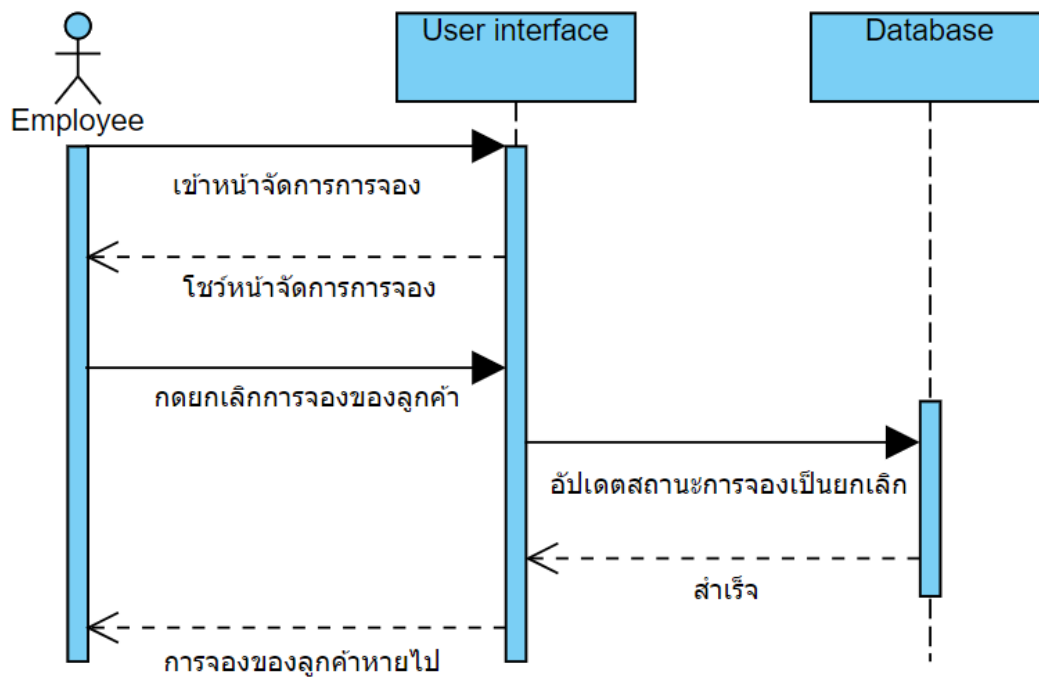
Extension (a):

- 3a พนักงานกด “ยกเลิก” การจองของลูกค้า
- 4a การจองของลูกค้าที่ถูกยกเลิกจะหายไปจากหน้าจอ

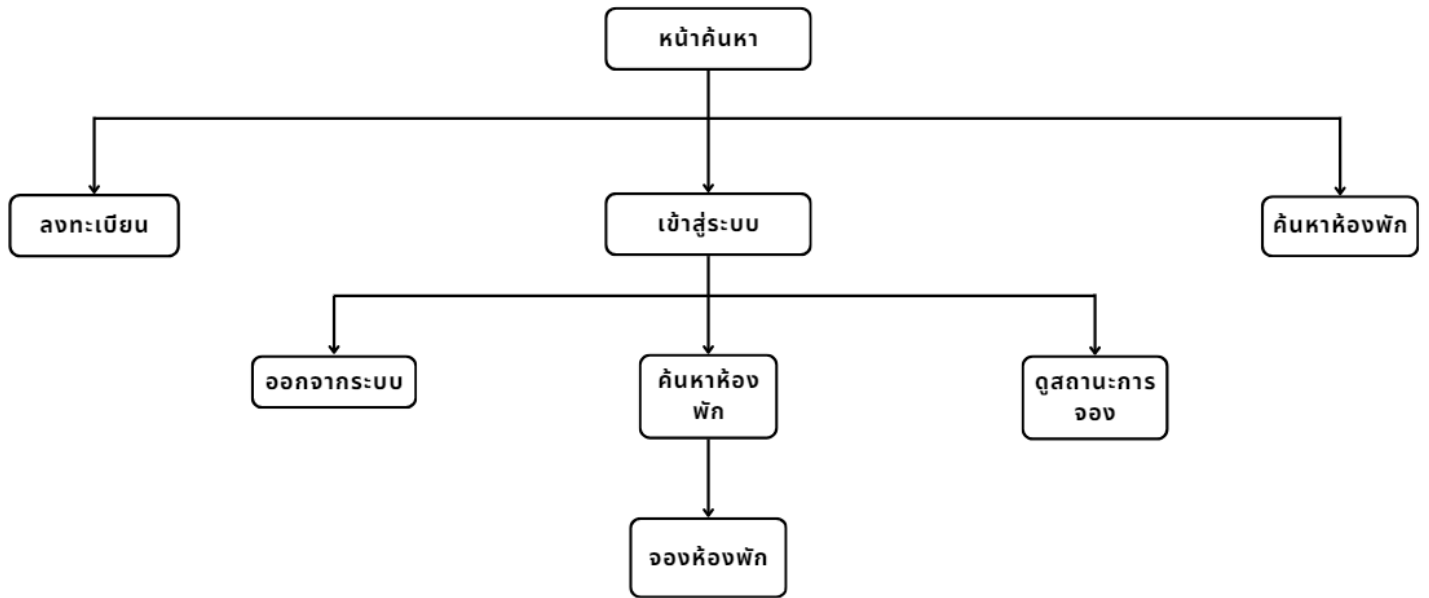
Main success scenario



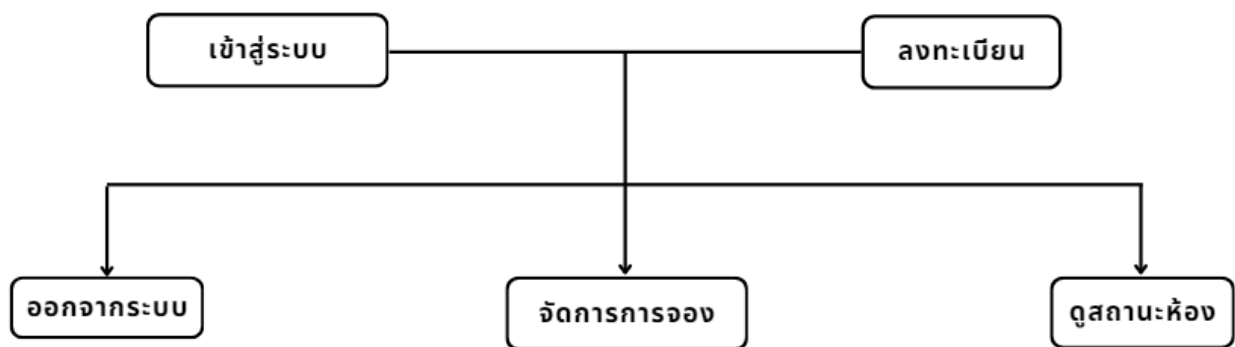
Extension (a)



- Navigation map



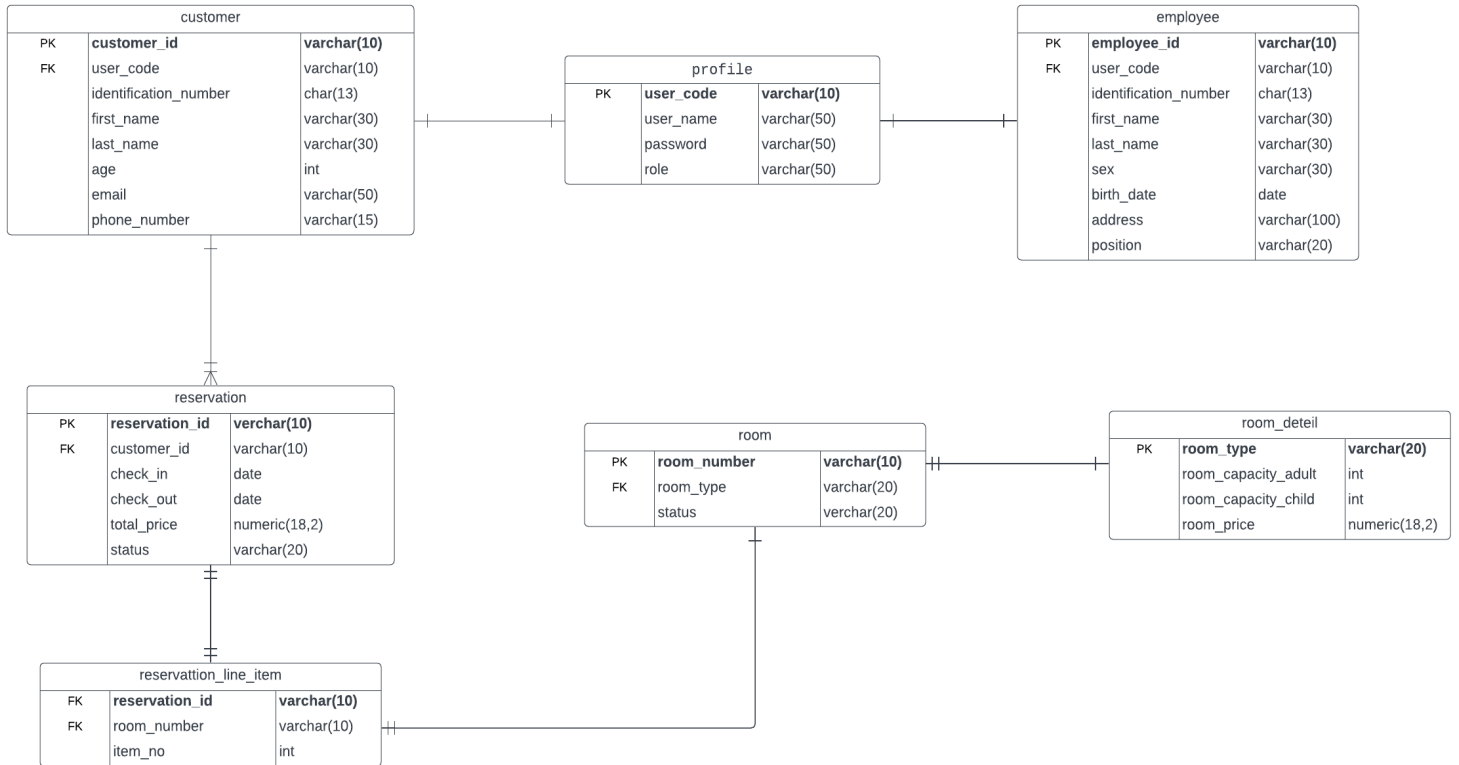
รูปที่ 1 Navigation map ของ customer



รูปที่ 2 Navigation map ของ employee

Data design

- Logical design



customer : เก็บข้อมูลลูกค้า

employee : เก็บข้อมูลพนักงาน

profile : เก็บบัญชีผู้ใช้ที่สมัครเข้ามาทั้งลูกค้าและพนักงาน

reservation : เก็บข้อมูลการจอง

reservation_line_item : เก็บรายละเอียดในการจอง

room : เก็บเลขห้องและสถานะของห้องพัก

room_detail : บอกข้อมูลของประเภทห้อง

Architectural design

Framework

1. Django
2. jquery

Language

1. Javascript

- ภาษา Javascript ในการทำ front end และ back end

2. Python

- ภาษา Python ในส่วนที่เกี่ยวข้องกับการทำงานกับข้อมูลและการแสดงผลข้อมูล

2. HTML

- ภาษา HTML ในการสร้างองค์ประกอบต่างที่มีในหน้าเว็บ

3. CSS

- ภาษา CSS ในการตกแต่งหน้าเว็บ

Tools

1. **Visual Studio Code** ใช้เป็น Code editor หลัก สำหรับการพัฒนาเว็บไซต์ของเราขึ้นมา โดยใช้ทั้งในการ run server และ ใช้ไลบรารี livereload เพื่อให้หน้าเว็บมีการอัปเดตแบบเรียลไทม์เมื่อมีการแก้ไขโค้ด.
2. **GitHub** ใช้เป็น Software Configuration Management (SCM) ของทีม เพื่อใช้ในการบริหารจัดการและควบคุมการเปลี่ยนที่ของ Source Code
3. **pgAdmin4** ใช้เป็นตัวจัดการฐานข้อมูล

Designing

1. **Figma** ใช้ในการออกแบบ User Interface ของแอปพลิเคชัน
2. **Visual Paradigm** ใช้ในการออกแบบ Use Case Diagram, Sequence

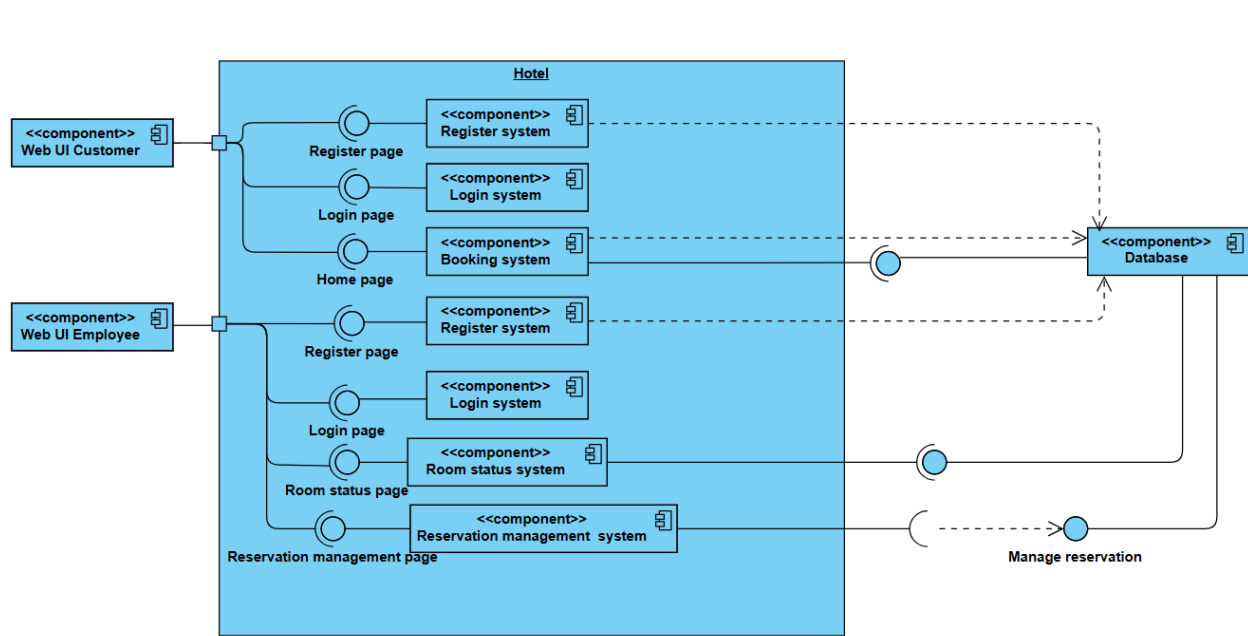
Diagram และ Component Diagram

3. **Canva** ใช้ในการออกแบบ Navigation map
4. **lucid. App** ใช้ในการออกแบบฐานข้อมูล

Documentation

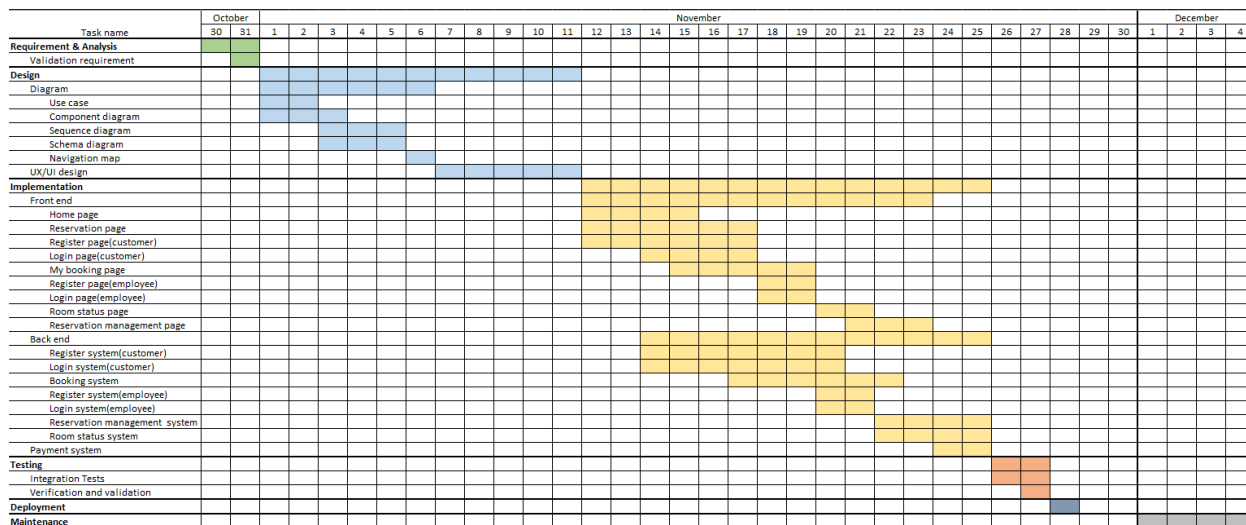
1. **Microsoft Word** ใช้เพื่อการเขียนรายงาน
2. **Canva** ใช้สำหรับการนำเสนอโปรเจค

Component diagram



Process

ทางกลุ่มของเราได้มีการใช้ Iterative Waterfall model ในการทำโปรเจกในครั้งนี้ และมีแผนการดำเนินการดังนี้



สำหรับรายละเอียดเพิ่มเติม: [Process chart](#)

Requirement & Analysis

เริ่มจากการ Analysis จาก story ของเว็บจองโรงแรมที่เราจะทำ เพื่อให้ได้ Requirement และในตอนสุดท้ายกลุ่มของเราได้ทำการ Validation Requirement ที่ Analysis มาเพื่อตรวจสอบความถูกต้องของ Requirement

Design

หลังจากทำการ Validation requirement เรียบร้อยแล้ว กลุ่มของเราจึงเริ่มทำการสร้าง Use case Diagram เพื่อออกแบบการโต้ตอบของผู้ใช้กับระบบ และยังสร้าง Component diagram เพื่ออธิบายถึงส่วนประกอบต่าง ๆ ที่ใช้ในการทำหน้าที่ของระบบ หลังจากเสร็จ Use case Diagram กลุ่มของเราได้เริ่มสร้าง Sequence diagram ในการแสดงลำดับการโต้ตอบและลำดับของกระบวนการในแต่ละขั้นตอน นอกจากนี้ยังมี Database schema เนื่องจากการออกแบบฐานข้อมูลใน Database และในตอนสุดท้ายของการ Design เราได้ใช้ Navigation map ในการบอกถึงสิ่งที่ผู้ใช้จะเห็นเมื่อเข้ามาใน Website

ทำการออกแบบ UX/UI design ตามหลัก golden rule โดยใช้ Figma ในการออกแบบ Website เพื่อช่วยให้เห็นภาพรวมของการออกแบบได้มากขึ้น

Implementation

เริ่มด้วยทำการสร้าง Git ใช้ในการแชร์งานระหว่างสมาชิกภายในกลุ่ม และใช้ทำ version control เพราะในบางครั้งจำเป็นต้องมีการ Roll back version

ต่อมาทำ Front end ให้เหมือน UX/UI design และทำการเขียนโค้ดบน VS code โดยใช้ภาษา HTML ,CSS และ JavaScript กลุ่มของเราจะเริ่มทำจากหน้าที่ทำได้ง่ายที่สุดก่อน

Back end จะเริ่มทำระบบหลังจากที่ Front end ของหน้าเว็บเสร็จแล้วเพื่อให้สามารถทำ Unit testing ในส่วนนั้น ๆ ได้ทันทีหลังจากที่ทำเสร็จ เพื่อตรวจสอบความถูกต้องในการทำงาน รวมถึงสามารถเชื่อมต่อ API ได้ง่าย ทำการเขียนโค้ดบน VS code โดยใช้ Framework Django ในการสร้างแต่ละหน้าให้มีระบบตามที่ได้ Design รวมทั้งการดึง แก้ไขและปรับแต่งข้อมูลใน Database

ในส่วนของ Payment system เนื่องจากกลุ่มของเรายังไม่มีระบบจ่ายเงิน จึงประชุมกันว่าจะให้ลูกค้าจ่ายเงินเองภายนอกและใช้ Google from ในการส่งสลิปการโอนเงิน

Testing

หลังจากขั้นตอน Implement เสร็จเรียบร้อย กลุ่มของเราทำ Integration Tests โดยการใช้ Robot Framework และ Manual Test ซึ่งหลังจากเจอปัญหาทำการส่งไปให้ Front end และ Back end แก้ไข.

รวมทั้งยังมี Verification and Validation ดูความถูกต้องในการใช้งานและตรงตาม Requirement ก่อนที่จะทำการ Deploy.

Deployment

Deploy ในวันที่ Present final project Database วันนั้นมีการแสดงวิธีใช้งานและให้กลุ่มอื่นใช้ Website

Maintenance

ทำการ Refactor code เพื่อให้สะดวก Update version ในอนาคต และมีการแก้ไขบัคที่พบเจอ

Self-Evaluation

Success Goals

สามารถสร้างระบบหลักของเว็บไซต์ให้ออกมาสำเร็จได้เช่น การลงทะเบียนผู้ใช้ใหม่ การเข้าสู่ระบบ การจองโรงแรม การจ่ายเงิน การจัดการการจอง สถานะของห้องต่าง ๆ และเมื่อนำไปทดสอบสามารถใช้งานได้จริง และบรรลุเป้าหมายตาม Requirements ได้อย่างครบถ้วน

Problem

1. การเปลี่ยนโปรเจกต์กระทันหัน เพราะว่าต้องทำโปรเจกต์ของรายวิชาดาต้าเบสด้วย ทำให้ตัดสินใจเปลี่ยนโปรเจกต์ให้เหมือนกับดาต้าเบสส่งผลให้ระยะเวลาในการทำโปรเจกต์ลดลง
2. การออกแบบ Design ยังไม่ชำนาญทำให้ในบาง Design ขาดรายละเอียดที่สำคัญ พอถึงขั้น Implement ต้องเติมเข้าไป แล้วกลับมาแก้ Design อีกรอบ
3. การใช้ Git ในช่วงต้น ๆ มีปัญหาในการ Setup และในเรื่องของไฟล์ cache เพราะไม่รู้จักการใช้ไฟล์ .gitignore ในช่วงแรกทำให้ใช้เวลาค่อนข้างเยอะในการหาทางใช้ Git
4. มีการเขียน CSS และ JavaScript ของแต่ละระบบรวมในไฟล์เดียวกันทำให้เกิด God module ในตอนหลังทำการแยกออกมาให้แต่ละไฟล์มีหน้าที่ของตัวเองอย่างชัดเจน
5. วางแผนระยะเวลาในการทำงานผิดพลาด จากที่จะเริ่มทำ Back end หลังจาก Front end ทั้งหมดเสร็จ แต่ในการทำงานจริง Front end ใช้เวลาในการทำงานมากกว่าที่วางแผนไว้ ทำให้ Back end ต้องเริ่มทำงานหลังจากที่ Front end ของระบบนั้น ๆ เสร็จแล้ว

Future Development

1. การจ่ายเงินให้สามารถ สแกนจ่ายเงินบนเว็บไซต์ได้ทันทีโดยไม่ต้องไปกรอกใน Google form เพื่อให้ Customer สามารถใช้งานได้สะดวกมากขึ้น
2. สามารถยกเลิกการจองห้องพักในกรณีที่ Customer เปลี่ยนใจหรือไม่สามารถเข้ามาใช้ห้องพักได้
3. Customer กับ Employee สามารถเปลี่ยนรหัสผ่านของตัวเองในกรณีที่ไม่สามารถจำรหัสผ่านได้แต่จะมีเงื่อนไขในการกดเปลี่ยนรหัสผ่าน