โครงงานวิทยาการคอมพิวเตอร์

ระบบการจัดการการลาป่วย-ลากิจ
Leave Management System

นางสาว ปาณิสรา	วงษ์บุบผา	รหัสประจำตัวนิสิต 6310451260
นาย พงศ์ศิริ	เจริญกฤตยาวุฒิ	รหัสประจำตัวนิสิต 6310451286

เสนอ

ผศ. ดร.อุษา สัมมาพันธ์

ภาควิชาวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์

มหาวิทยาลัยเกษตรศาสตร์

ภาคปลาย ปีการศึกษา 2566

โครงงานวิทยาการคอมพิวเตอร์

ชื่อเรื่อง ระบบการจัดการการลาป่วย-ลากิจ

Leave Management System

ผู้จัดทำ นางสาว ปาณิสรา วงษ์บุบผา รหัสประจำตัวนิสิต 6310451260

นาย พงศ์ศิริ เจริญกฤตยาวุฒิ รหัสประจำตัวนิสิต 6310451286

อาจารย์ที่ปรึกษาหลัก _____

(ผศ. ดร.อุษา สัมมาพันธ์)

7 / เม.ย. / 67

อาจารย์ที่ปรึกษาร่วม _____

(ผศ.ศิริกร จันทร์นวล)

8 / 4 / 2567

อาจารย์ที่ปรึกษาร่วม _____

(ผศ.พบสิทธิ์ กมลเวชช)

8 / 4 / 2567

หัวหน้าภาควิชาวิทยาการคอมพิวเตอร์ _____

(อ. สมโชค เรืองอิทธินันท์)

9 / 4 / 67

นางสาว ปาณิสรา วงษ์บุบผา และ นาย พงศ์ศิริ เจริญกฤตยาวุฒิ. 2566. **ระบบการจัดการการลาป่วย-ลากิจ.** โครงงานวิทยาการคอมพิวเตอร์ ปริญญาวิทยาศาสตร์บัณฑิต สาขาวิทยาการคอมพิวเตอร์ ภาควิชาวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์ มหาวิทยาลัยเกษตรศาสตร์.

อาจารย์ที่ปรึกษา: ผศ. ดร.อุษา สัมมาพันธ์

บทคัดย่อ

การออกแบบ และพัฒนาระบบการจัดการการลาป่วย-ลากิจ มีวัตถุประสงค์เพื่ออำนวยความสะดวกสบายแก่อาจารย์ และนิสิต และเพื่อหลีกเลี่ยงการเกิดข้อผิดพลาดในการเช็คเวลาเข้าเรียนของนิสิต โดยระบบนี้นิสิตสามารถแจ้งลาแก่อาจารย์ และอาจารย์สามารถอนุมัติหรือปฏิเสธคำขอลาได้ อีกทั้งยังสามารถค้นหา หรือแสดงรายละเอียดการลาของนิสิตได้ โดยในการพัฒนาเว็บแอปพลิเคชันทางผู้จัดทำได้แบ่งระบบออกเป็นสองส่วนหลัก ได้แก่ การพัฒนาระบบ Backend และส่วนของ Frontend

การพัฒนาระบบ Backend ภาษาโปรแกรมที่ใช้คือ Python และ Django Framework ใช้สร้างและจัดการกับฐานข้อมูล, การจัดการเส้นทาง (routing), การจัดการผู้ใช้, และฟอร์มต่าง ๆ ที่เกี่ยวข้องกับฝั่ง Backend ของแอปพลิเคชัน ส่วนของระบบ Frontend ภาษาโปรแกรมที่ใช้คือ HTML CSS JAVASCRIPT และ VUE.JS Framework ช่วยในการจัดการและแสดงผลข้อมูลที่ได้มาจากระบบ Backend รวมถึงการจัดการส่วนต่าง ๆ ของหน้าเว็บ

การพัฒนาเว็บแอปพลิเคชัน ระบบจัดการการลาป่วย-ลากิจจะเป็นประโยชน์แก่อาจารย์และนิสิตเนื่องจากมีฟังก์ชันการใช้งานที่สะดวกและรวดเร็ว ลดการใช้เอกสาร และ การตกหล่นของใบลาในอีเมลของอาจารย์ ลดจำนวนอีเมลที่อาจารย์จะได้รับทำให้อีเมลของอาจารย์เป็นระเบียบมากขึ้น และนิสิตสามารถติดตามสถานะการอนุมัติได้ในทันที

คำสำคัญ: Django Framework, HTML, CSS, JAVASCRIPT, VUE.JS Framework

Ms.Panisra Wongbubpa and Mr.Pongsiri Charoenkittayawut. 2023. **Leave Management System**. Computer Science Project. Bachelor of Science in Computer Science, Department of Computer Science, Faculty of Science, Kasetsart University.

Project Advisor: Asst.prof. Usa Sammapun, Ph.D.

ABSTRACT

The objective of designing and developing a sick leave and business leave management system is to provide convenience for both teachers and students, and to avoid errors in checking students' attendance. In this system, students can notify teachers of their leave requests, and teachers can approve or reject these requests. Additionally, the system allows for searching and displaying detailed leave information for students. The web application development is divided into two main parts: Backend Development and Frontend Development.

For Backend Development, Python with the Django Framework is used to create and manage databases, handle routing, manage users, and create various forms related to the backend of the application. For Frontend Development, HTML, CSS, JavaScript, and Vue.js Framework are used to manage and display data from the backend, as well as handle different parts of the web pages.

Developing the sick leave and business leave management web application benefits both teachers and students by providing convenient and efficient functionalities, reducing paperwork and email clutter for teachers, and allowing students to track approval status instantly.

.

Keywords: Django Framework, HTML, CSS, JAVASCRIPT, VUE.JS Framework

กิตติกรรมประกาศ

การจัดทำโครงงานวิทยาการคอมพิวเตอร์หัวข้อระบบการจัดการการลาป่วย-ลากิจในครั้งนี้ คณะผู้จัดทำได้รับความอนุเคราะห์จากหลายๆ ฝ่ายทำให้โครงงานสำเร็จลุล่วงไปด้วยดีโดยได้รับคำแนะนำการช่วยเหลือและการสนับสนุนจากบุคคล ดังนี้

ขอขอบพระคุณ ผศ.ดร.อุษา สัมมาพันธ์ อาจารย์ที่ปรึกษาโครงงาน ที่กรุณาให้คำแนะนำต่าง ๆ ตั้งแต่การเลือกหัวข้อโครงงานไปจนถึงแนวทางในการดำเนินงานทำให้โครงงานเสร็จสิ้นสมบูรณ์ด้วยดีรวมไปถึงการให้คำปรึกษาเกี่ยวกับปัญหาที่พบในการจัดทำโครงงานทำให้ตลอดเวลาระยะการจัดทำโครงงานทำให้โครงงานมีความราบรื่น ขอขอบพระคุณ ผศ.ศิริกร จันทร์นวล และ ผศ.พบสิทธิ์ กมลเวชช อาจารย์ที่ปรึกษาร่วมที่ให้คำแนะนำรวมไปถึงข้อผิดพลาดที่พบในโครงงานเพื่อให้คณะผู้จัดทำนำมาปรับปรุงแก้ไขให้โครงงานมีความสมบูรณ์มากยิ่งขึ้นสุดท้ายนี้ขอขอบพระคุณผู้ที่มีส่วนเกี่ยวข้องที่คอยให้คำแนะนำให้คำปรึกษาเมื่อมีข้อสงสัยให้ความช่วยเหลือในการจัดทำโครงงาน ชี้แนะแนวทางในการทำงาน ให้ความรู้ในการทำงาน และคอยให้กำลังใจเมื่อได้รับแรงกดดันจากการทำโครงงานเพื่อให้คณะผู้จัดทำสามารถดำเนินงานได้สำเร็จลุล่วงตามเป้าหมายในการจัดทำโครงงานวิทยาการคอมพิวเตอร์ในครั้งนี้

ผู้จัดทำ
นางสาว ปาณิสรา วงษ์บุบผา
นาย พงศ์ศิริ เจริญกฤตยาวุฒิ

สารบัญ

เรื่อง	หน้า
บทที่1	
บทนำ	**1**
1.1 ที่มาและความสำคัญ	1
1.2 วัตถุประสงค์ของโครงงาน	1
1.3 ประโยชน์ที่คาดว่าจะได้รับ	1
1.4 ขอบเขตและข้อจำกัดของระบบ	2
1.5 ผู้ใช้ประโยชน์จากระบบ	2
1.6 เครื่องมือที่ใช้ในการพัฒนาระบบ	2
1.7 แผนการดำเนินงาน	3
บทที่2	
ระบบงานและทฤษฎีที่เกี่ยวข้อง	**4**
2.1 หลักการและทฤษฎีที่ใช้ในการพัฒนาโครงงาน	4
2.1.1 ทฤษฎีของ Django	4
2.1.2 ทฤษฎีของ Vue.js	5
2.1.3 ทฤษฎีของ HTTP (HyperText Transfer Protocol)	5
2.1.4 JSON Web Token	6
บทที่3	
การวิเคราะห์และออกแบบระบบ	**7**
3.1 การวิเคราะห์และออกแบบระบบ	7
3.2 Input/Output Specification	8
3.3 Use case diagram	11

เรื่อง **หน้า**

3.4 การวิเคราะห์ขั้นตอนการทำงานของฟังก์ชันโดยใช้ยูสเคส 12

3.5 ซีเควนไดอะแกรม 21

3.6 การออกแบบฐานข้อมูล 30

3.7 การออกแบบส่วนต่อประสานกับผู้ใช้ 39

บทที่4

การพัฒนาระบบ **44**

4.1 เข้าสู่ระบบ 44

4.2 แจ้งลา-ยกเลิกการแจ้งลา 51

4.3 แสดงคำขอลาทั้งหมด 70

4.4 แสดงรายละเอียดการลา 81

4.5 อนุมัติการลาและปฏิเสธการลา 96

4.6 ค้นหาการลาตามเงื่อนไข 99

4.7 รายงานสรุปการลา 106

บทที่5

การทดสอบระบบ **113**

5.1 การทดสอบระบบรวม 113

5.2 การทดสอบระบบย่อย 114

บทที่6

บทสรุป **143**

6.1 บทสรุป 143

เรื่อง **หน้า**

6.2 ปัญหาและข้อเสนอแนะ 143

6.3 แนวทางการพัฒนาต่อยอด 143

บรรณานุกรม **145**

ภาคผนวก ก

การติดตั้งระบบ **146**

1. ติดตั้ง Node.js 146

2. ติดตั้ง python 146

3. ติดตั้ง Visual Studio Code 146

4. ติดตั้ง MySQL Database 146

5. ติดตั้งโปรเจค Backend 147

6. ติดตั้งโปรเจค Frontend 147

ภาคผนวก ข

วิธีการใช้งานระบบ **149**

1. หน้าแรก - นิสิต 149

2. หน้าแจ้งลา - นิสิต 150

3. หน้าคำขอลาทั้งหมด - นิสิต 150

4. หน้าแรก - อาจารย์ 151

5. คำขอลาทั้งหมด - อาจารย์ 151

6. ค้นหาการลาตามเงื่อนไข - อาจารย์ 152

7. รายงานสรุปการลา - อาจารย์ 152

8. แสดงรายละเอียดการลา 153

สารบัญรูป

รูปที่ **หน้า**

รูปที่ 2.1 Architecture of Django 4
รูปที่ 2.2 Communication process according to HTTP 5
รูปที่ 3.1 แสดงแผนภาพการทำงาน หรือ work flow หรือ Activity diagram ของระบบจัดการการลาป่วยลากิจ 7
รูปที่ 3.2 แสดงแผนภาพยูสเคส 11
รูปที่ 3.3 แสดงแผนภาพซีเควนไดอะแกรม 1 21
รูปที่ 3.4 แสดงแผนภาพซีเควนไดอะแกรม 2 22
รูปที่ 3.5 แสดงแผนภาพซีเควนไดอะแกรม 3 23
รูปที่ 3.6 แสดงแผนภาพซีเควนไดอะแกรม 4 24
รูปที่ 3.7 แสดงแผนภาพซีเควนไดอะแกรม 5 25
รูปที่ 3.8 แสดงแผนภาพซีเควนไดอะแกรม 5 25
รูปที่ 3.9 แสดงแผนภาพซีเควนไดอะแกรม 6 26
รูปที่ 3.10 แสดงแผนภาพซีเควนไดอะแกรม 7 27
รูปที่ 3.11 แสดงแผนภาพซีเควนไดอะแกรม 8 28
รูปที่ 3.12 แสดงแผนภาพซีเควนไดอะแกรม 9 29
รูปที่ 3.13 แสดงแผนภาพการออกแบบฐานข้อมูลระดับแนวคิด โดย ER-diagram 30
รูปที่ 3.14 แสดงแผนภาพการออกแบบฐานข้อมูลระดับกายภาพ 31
รูปที่ 3.15 แสดงการออกแบบส่วนต่อประสานกับผู้ใช้ในส่วนของหน้าเข้าสู่ระบบ 39
รูปที่ 3.16 แสดงการออกแบบส่วนต่อประสานกับผู้ใช้ในส่วนของหน้าแรกของนิสิต 39
รูปที่ 3.17 แสดงการออกแบบส่วนต่อประสานกับผู้ใช้ในส่วนของหน้าแรกของอาจารย์ 40
รูปที่ 3.18 แสดงการออกแบบส่วนต่อประสานกับผู้ใช้ในส่วนของหน้าแจ้งลา 40
รูปที่ 3.19 หน้าคำขอลาทั้งหมด 41
รูปที่ 3.20 หน้าคำขอลาทั้งหมดของอาจารย์ 41
รูปที่ 3.21 หน้ารายละเอียดคำขอลา 42
รูปที่ 3.22 หน้าค้นหาการลา 42
รูปที่ 3.23 หน้ารายงานสรุปการลา 43
รูปที่ 3.24 หน้ารายงานสรุปผลการลา 43
รูปที่ 4.1 โปรแกรมส่วนเข้าสู่ระบบ 44
รูปที่ 4.2 ตัวอย่างโค้ดส่วนเทมเพลตของ App.vue 45
รูปที่ 4.3 ตัวอย่างโค้ดส่วนสคริปต์ของ App.vue 45
รูปที่ 4.4 ตัวอย่างโค้ดส่วนสไตล์ของ App.vue 46
รูปที่ 4.5 ตัวอย่างโค้ดส่วนเทมเพลตของ LogInView.vue 47
รูปที่ 4.6 ตัวอย่างโค้ดส่วนสคริปต์ของ LogInView.vue 48
รูปที่ 4.7 ตัวอย่างโค้ดส่วนสไตล์ของ LogInView.vue 49

รูปที่ **หน้า**

รูปที่ 4.8 ตัวอย่างโค้ดส่วนโมเดล Class CustomUserManager 50
รูปที่ 4.9 ตัวอย่างโค้ดส่วนโมเดล User 50
รูปที่ 4.10 ตัวอย่างโค้ดส่วน account/urls.py 51
รูปที่ 4.11 โปรแกรมส่วนแจ้งลา 51
รูปที่ 4.12 ตัวอย่างโค้ดส่วนสคริปต์ของ LeavePaperView.vue 58
รูปที่ 4.14 ตัวอย่างโค้ดส่วนสไตล์ของ LeavePaperView.vue 61
รูปที่ 4.15 ตัวอย่างโค้ดส่วน leave/models.py 62
รูปที่ 4.16 ตัวอย่างโค้ดส่วน LeaveRequestSerializer 63
รูปที่ 4.17 ตัวอย่างโค้ดส่วน LeaveRequestDetailSerializer 63
รูปที่ 4.18 ตัวอย่างโค้ดส่วน FileSerializer 64
รูปที่ 4.19 ตัวอย่างโค้ดส่วน LeaveRequestViewSet 65
รูปที่ 4.20 ตัวอย่างโค้ดส่วน LeaveRequestDetailViewSet 65
รูปที่ 4.21 ตัวอย่างโค้ดส่วน FilesViewSet 65
รูปที่ 4.22 ตัวอย่างโค้ดส่วน leave/urls.py 66
รูปที่ 4.23 โปรแกรมส่วนแสดงรายละเอียดการลาของนิสิต 66
รูปที่ 4.24 ตัวอย่างโค้ดส่วนเทมเพลตของ DetailPage.view 67
รูปที่ 4.25 ตัวอย่างโค้ดส่วนสคริปต์ของ DetailPage.view 68
รูปที่ 4.26 ตัวอย่างโค้ดส่วน LeaveDetailViewSet 69
รูปที่ 4.27 โปรแกรมส่วนแสดงคำขอลาทั้งหมดของนิสิต 70
รูปที่ 4.28 ตัวอย่างโค้ดส่วนเทมเพลตของ RequestView.vue 72
รูปที่ 4.29 ตัวอย่างโค้ดส่วนสคริปต์ของ RequestView.vue 74
รูปที่ 4.30 ตัวอย่างโค้ดส่วน LeaveRequestDetailViewSet 75
รูปที่ 4.31 โปรแกรมส่วนแสดงคำขอลาทั้งหมดของนิสิตที่ส่งมาให้อาจารย์ 75
รูปที่ 4.32 ตัวอย่างโค้ดส่วนเทมเพลตของ RequestTeacherView.vue 77
รูปที่ 4.33 ตัวอย่างโค้ดส่วนสคริปต์ของ RequestTeacherView.vue 80
รูปที่ 4.34 ตัวอย่างโค้ดส่วน LeaveRequestDetailViewSet 81
รูปที่ 4.35 โปรแกรมส่วนแสดงรายละเอียดการลาของนิสิต 81
รูปที่ 4.36 ตัวอย่างโค้ดส่วนเทมเพลตของ DetailPage.vue 84
รูปที่ 4.37 ตัวอย่างโค้ดส่วนสคริปต์ของ DetailPage.vue 86
รูปที่ 4.38 ตัวอย่างโค้ดส่วน LeaveRequestViewSet 87
รูปที่ 4.39 ตัวอย่างโค้ดส่วน LeaveRequestDetailViewSet 88
รูปที่ 4.40 ตัวอย่างโค้ดส่วน FilesViewSet 88
รูปที่ 4.41 โปรแกรมส่วนแสดงรายละเอียดการลาของนิสิตที่ส่งให้อาจารย์ 89
รูปที่ 4.42 ตัวอย่างโค้ดส่วนเทมเพลตของ DetailTeacherPage.vue 91
รูปที่ 4.43 ตัวอย่างโค้ดส่วนสคริปต์ของ DetailTeacherPage.vue 93
รูปที่ 4.44 ตัวอย่างโค้ดส่วน LeaveRequestViewSet 94

รูปที่ **หน้า**

รูปที่ 4.45 ตัวอย่างโค้ดส่วน LeaveRequestDetailViewSet 95
รูปที่ 4.46 ตัวอย่างโค้ดส่วน FilesViewSet 95
รูปที่ 4.47 ตัวอย่างโค้ดส่วนเทมเพลตของ RequestTeacherView.vue 97
รูปที่ 4.48 ตัวอย่างโค้ดส่วนสคริปต์ของ 98
รูปที่ 4.49 ตัวอย่างโค้ดส่วน LeaveRequestDetailViewSet 99
รูปที่ 4.50 โปรแกรมส่วนค้นหาการลาตามเงื่อนไข 99
รูปที่ 4.51 ตัวอย่างโค้ดส่วนเทมเพลตของ SearchLeaveView.vue 101
รูปที่ 4.52 ตัวอย่างโค้ดส่วนสคริปต์ของ SearchLeaveView.vue 105
รูปที่ 4.53 ตัวอย่างโค้ดส่วน LeaveRequestDetailViewSet 106
รูปที่ 4.54 โปรแกรมส่วนแสดงรายงานสรุปผลการลาของอาจารย์ตอนที่ยังไม่กดขยายแถบวิชา 106
รูปที่ 4.55 โปรแกรมส่วนแสดงรายงานสรุปผลการลาของอาจารย์ตอนกดขยายแถบวิชา 107
รูปที่ 4.56 ตัวอย่างโค้ดส่วนเทมเพลตของ SummaryReportView.vue 108
รูปที่ 4.57 ตัวอย่างโค้ดส่วนสคริปต์ของ SummaryReport.View.vue 111
รูปที่ 4.58 ตัวอย่างโค้ดส่วน LeaveRequestDetailFilter 111
รูปที่ 4.59 ตัวอย่างโค้ดส่วน LeaveRequestDetailView 111
รูปที่ 4.60 ตัวอย่างโค้ดส่วน leave/urls.py 112
รูปที่ 5.1 แสดงแผนภาพการทดสอบระบบรวม 113
รูปที่ 5.2 แสดงแผนภาพการทดสอบโปรแกรมเข้าสู่ระบบ 114
รูปที่ 5.3 แสดงภาพประกอบกรณีทดสอบ TC 1.1 115
รูปที่ 5.4 แสดงภาพประกอบกรณีทดสอบ TC 1.2 115
รูปที่ 5.5 แสดงภาพประกอบกรณีทดสอบ TC 1.3 116
รูปที่ 5.6 แสดงแผนภาพการทดสอบโปรแกรมการแจ้งลา 116
รูปที่ 5.7 แสดงภาพประกอบกรณีทดสอบ TC 2.1 118
รูปที่ 5.8 แสดงภาพประกอบกรณีทดสอบ TC 2.2 118
รูปที่ 5.9 แสดงภาพประกอบกรณีทดสอบ TC 2.3 119
รูปที่ 5.10 แสดงภาพประกอบกรณีทดสอบ TC 2.4 119
รูปที่ 5.11 แสดงภาพประกอบกรณีทดสอบ TC 2.5 119
รูปที่ 5.12 แสดงภาพประกอบกรณีทดสอบ TC 2.6 120
รูปที่ 5.13 แสดงแผนภาพการทดสอบโปรแกรมคำขอลาทั้งหมดของนิสิตที่ส่งให้อาจารย์ 120
รูปที่ 5.14 แสดงภาพประกอบกรณีทดสอบ TC 3.1 121
รูปที่ 5.15 แสดงภาพประกอบกรณีทดสอบ TC 3.2 122
รูปที่ 5.16 แสดงภาพประกอบกรณีทดสอบ TC 3.3 122
รูปที่ 5.17 แสดงภาพประกอบกรณีทดสอบ TC 3.4 123
รูปที่ 5.18 แสดงภาพประกอบกรณีทดสอบ TC 3.5 123
รูปที่ 5.19 แสดงภาพประกอบกรณีทดสอบ TC 3.6 123
รูปที่ 5.20 แสดงภาพประกอบกรณีทดสอบ TC 3.7 124

รูปที่ **หน้า**

รูปที่ 5.21 แสดงภาพประกอบกรณีทดสอบ TC 3.8 124
รูปที่ 5.22 แสดงภาพประกอบกรณีทดสอบ TC 3.8 125
รูปที่ 5.23 แสดงภาพประกอบกรณีทดสอบ TC 3.9 125
รูปที่ 5.24 แสดงภาพประกอบกรณีทดสอบ TC 3.9 125
รูปที่ 5.25 แสดงภาพประกอบกรณีทดสอบ TC 3.10 126
รูปที่ 5.26 แสดงแผนภาพการทดสอบโปรแกรมส่วนรายละเอียดการลาของนิสิตที่ส่งให้อาจารย์ 126
รูปที่ 5.27 แสดงภาพประกอบกรณีทดสอบ TC 4.1 127
รูปที่ 5.28 แสดงภาพประกอบกรณีทดสอบ TC 4.2 128
รูปที่ 5.29 แสดงภาพประกอบกรณีทดสอบ TC 4.3 128
รูปที่ 5.30 แสดงภาพประกอบกรณีทดสอบ TC 4.3 129
รูปที่ 5.31 แสดงภาพประกอบกรณีทดสอบ TC 4.4 129
รูปที่ 5.32 แสดงภาพประกอบกรณีทดสอบ TC 4.4 129
รูปที่ 5.33 แสดงแผนภาพการทดสอบโปรแกรมค้นหาการลาตามเงื่อนไข 130
รูปที่ 5.34 แสดงภาพประกอบกรณีทดสอบ TC 5.1 131
รูปที่ 5.35 แสดงภาพประกอบกรณีทดสอบ TC 5.2 132
รูปที่ 5.36 แสดงภาพประกอบกรณีทดสอบ TC 5.3 132
รูปที่ 5.37 แสดงภาพประกอบกรณีทดสอบ TC 5.4 132
รูปที่ 5.38 แสดงภาพประกอบกรณีทดสอบ TC 5.5 133
รูปที่ 5.39 แสดงภาพประกอบกรณีทดสอบ TC 5.6 133
รูปที่ 5.40 แสดงแผนภาพการทดสอบโปรแกรมรายงานสรุปการลา 134
รูปที่ 5.41 แสดงภาพประกอบกรณีทดสอบ TC 6.1 135
รูปที่ 5.42 แสดงภาพประกอบกรณีทดสอบ TC 6.2 135
รูปที่ 5.43 แสดงภาพประกอบกรณีทดสอบ TC 6.3 135
รูปที่ 5.44 แสดงแผนภาพการทดสอบโปรแกรมส่วนแสดงคำขอลาทั้งหมดของนิสิต 136
รูปที่ 5.45 แสดงภาพประกอบกรณีทดสอบ TC 7.1 137
รูปที่ 5.46 แสดงภาพประกอบกรณีทดสอบ TC 7.2 138
รูปที่ 5.47 แสดงภาพประกอบกรณีทดสอบ TC 7.3 138
รูปที่ 5.48 แสดงภาพประกอบกรณีทดสอบ TC 7.4 139
รูปที่ 5.49 แสดงแผนภาพการทดสอบโปรแกรมหน้ารายละเอียดการลาของนิสิต 139
รูปที่ 5.50 แสดงภาพประกอบกรณีทดสอบ TC 8.1 140
รูปที่ 5.51 แสดงภาพประกอบกรณีทดสอบ TC 8.2 141
รูปที่ 5.52 แสดงภาพประกอบกรณีทดสอบ TC 8.3 141
รูปที่ 5.53 แสดงภาพประกอบกรณีทดสอบ TC 8.4 142
รูปที่ ข.1 หน้าแรก - นิสิต 149
รูปที่ ข.2 วิธีการใช้ระบบ-หน้าแจ้งลา (นิสิต) 150
รูปที่ ข.3 วิธีการใช้ระบบ-คำขอลาทั้งหมด (นิสิต) 150

รูปที่ **หน้า**

รูปที่ ข.4 วิธีการใช้ระบบ-หน้าแรก (อาจารย์) 151
รูปที่ ข.5 วิธีการใช้ระบบ-คำขอลาทั้งหมด (อาจารย์) 151
รูปที่ ข.6 วิธีการใช้ระบบ-ค้นหาการลาตามเงื่อนไข 152
รูปที่ ข.7 วิธีการใช้ระบบ-รายการสรุปการลา 152
รูปที่ ข.8 วิธีการใช้ระบบ-แสดงรายละเอียดการลา 153

สารบัญตาราง

ตารางที่ **หน้า**

บทที่1 บทนำ

ตารางที่ 1.1 ตารางแผนการดำเนินงาน 3

บทที่2 ระบบงานและทฤษฎีที่เกี่ยวข้อง

บทที่3 การวิเคราะห์และออกแบบระบบ

ตารางที่ 3.1 ข้อกำหนดและผลการใช้งานข้อมูลขาเข้า/ขาออก 8

ตารางที่ 3.2 Use case description เข้าสู่ระบบ 12

ตารางที่ 3.3 Use case description แจ้งลา 13

ตารางที่ 3.4 Use case description ยกเลิกการแจ้งลา 15

ตารางที่ 3.5 Use case description แสดงคำขอลาทั้งหมดที่ส่งไปทุกวิชา 16

ตารางที่ 3.6 Use case description แสดงคำขอลาทั้งหมด 17

ตารางที่ 3.7 Use case description แสดงรายละเอียดการลาของนิสิตแต่ละคน 17

ตารางที่ 3.8 Use case description อนุมัติการลา 18

ตารางที่ 3.9 Use case description ปฏิเสธคำขอลา 19

ตารางที่ 3.10 Use case description ค้นหาการลาตามเงื่อนไข 20

ตารางที่ 3.11 Use case description รายงานสรุปผลการลา 20

ตารางที่ 3.12 account_user 32

ตารางที่ 3.13 student_student 33

ตารางที่ 3.14 student_studentregistercourser 33

ตารางที่ 3.15 teacher_teacher 34

ตารางที่ 3.16 teacher_instructorcourse 34

ตารางที่ 3.17 education_course 35

ตารางที่ 3.18 education_faculty 35

ตารางที่ **หน้า**

ตารางที่ 3.19 education_department 36

ตารางที่ 3.20 leave_leaverequest 36

ตารางที่ 3.21 leave_leaverequestdetail 37

ตารางที่ 3.22 leave_files 38

บทที่4 การพัฒนาระบบ

บทที่5 การทดสอบระบบ

ตารางที่ 5.1 แสดงกรณีทดสอบที่ 1 114

ตารางที่ 5.2 แสดงกรณีทดสอบที่ 2 117

ตารางที่ 5.3 แสดงกรณีทดสอบที่ 3 121

ตารางที่ 5.4 แสดงกรณีทดสอบที่ 4 127

ตารางที่ 5.5 แสดงกรณีทดสอบที่ 5 131

ตารางที่ 5.6 แสดงกรณีทดสอบที่ 6 134

ตารางที่ 5.7 แสดงกรณีทดสอบที่ 7 137

ตารางที่ 5.8 แสดงกรณีทดสอบที่ 8 140

บทที่6 บทสรุป

ภาคผนวก ก การติดตั้งระบบ

ภาคผนวก ข วิธีการใช้งานระบบ

บทที่ 1

บทนำ

1.1 ที่มาและความสำคัญ

เนื่องจากการเรียนในระดับมหาวิทยาลัยปฏิเสธไม่ได้เลยว่ามักจะมีนิสิตจำนวนไม่น้อยที่ต้องการลาเรียนเนื่องจากเหตุจำเป็น เหตุฉุกเฉินหรือแม้กระทั่งเกิดอุบัติเหตุกับนิสิต และระบบการลาในปัจจุบันเป็นการส่งอีเมลไปยังอาจารย์ประจำวิชา ไม่มีการเก็บข้อมูลใด ๆ และไม่เป็นระเบียบ ทำให้ยุ่งยากต่อการอนุมัติหรือตรวจสอบ คณะผู้จัดทำจึงมีความคิดริเริ่มในการพัฒนาระบบการจัดการการลาป่วย-ลากิจโดยนิสิตสามารถส่งคำขอลาและหลักฐานในการลาได้และอาจารย์ประจำวิชาสามารถอนุมัติหรือปฏิเสธคำขอลาตรวจสอบรายละเอียดและเหตุผลในการลาอีกทั้งยังสามารถแสดงคำขอลาทั้งหมดและค้นหาคำขอลาที่ต้องการได้

ในการพัฒนาระบบการจัดการการลาป่วย-ลากิจโดยใช้ ภาษา Python และ Django Framework ในการพัฒนาระบบ Backend และในส่วนของระบบ Frontend คณะผู้จัดทำได้ใช้ภาษาโปรแกรม HTML CSS JAVASCRIPT และ VUE.JS Framework ช่วยในการจัดการและแสดงผลข้อมูลที่ได้มาจากระบบ Backend รวมถึงการจัดการส่วนต่าง ๆ ของหน้าเว็บเป็นเครื่องมือและเทคโนโลยีที่ทันสมัย มีความสะดวกในการใช้งาน เพื่อทำให้การพัฒนาเว็บแอปพลิเคชันมีประสิทธิภาพสูงสุด

1.2 วัตถุประสงค์ของโครงงาน

เพื่อพัฒนาเว็บแอปพลิเคชันระบบลา โดยสามารถรองรับการใช้งานที่เกี่ยวกับการลาประเภทต่างๆอย่างครบถ้วน ทั้งในด้านของการยื่นขอลาและการอนุมัติลา

1.3 ประโยชน์ที่คาดว่าจะได้รับ

1.3.1 ความสะดวกสบายในการจัดการลา

ผู้ใช้สามารถยื่นขอลาและติดตามสถานะได้ผ่านทางแพลตฟอร์มออนไลน์ทำให้กระบวนการจัดการลาเป็นไปอย่างรวดเร็วและสะดวกสบาย

1.3.2 ลดภาระงานของอาจารย์

ระบบที่อัตโนมัติและประสิทธิภาพช่วยลดภาระงานที่เกี่ยวข้องกับการจัดการลาทำให้เวลาและทรัพยากรมนุษย์สามารถนำมาใช้ในงานอื่นได้มากขึ้น

1.3.3 ความครบถ้วนในการจัดการลา

ระบบสามารถรองรับทุกประเภทของการลา, เช่น ลาป่วย, ลากิจ, และลาพิเศษอื่น ๆ ทำให้สามารถจัดการได้ทุกรูปแบบของการลา

1.3.4 การอนุมัติลาที่รวดเร็ว

ระบบทำให้กระบวนการอนุมัติลาเป็นไปอย่างรวดเร็วและมีประสิทธิภาพ ซึ่งช่วยลดเวลาที่ต้องใช้ในการรอคำตอบและเพิ่มประสิทธิภาพในการบริหารจัดการ.

1.3.5 การติดตามประวัติการลา

ระบบบันทึกประวัติการลาของแต่ละบุคคลทำให้ง่ายต่อการติดตามและทราบข้อมูลเกี่ยวกับการลาที่ผ่านมา

1.4 ขอบเขตและข้อจำกัดของระบบ

1.4.1 นิสิต

- สามารถแจ้งลา
- สามารถยกเลิกการแจ้งลา
- สามารถแสดงคำขอลาทั้งหมดที่ส่งไปทุกวิชา

1.4.2 อาจารย์

- สามารถแสดงคำขอลาทั้งหมด
- แสดงรายละเอียดการลาของนิสิตแต่ละคน
- สามารถอนุมัติการลา
- สามารถปฏิเสธคำขอลา
- สามารถค้นหาการลาตามเงื่อนไข
- สามารถดูรายงานสรุปการลา

1.5 ผู้ใช้ประโยชน์จากระบบ

- อาจารย์
- นิสิต

1.6 เครื่องมือที่ใช้ในการพัฒนาระบบ

1.6.1 เครื่องมือสำหรับ Frontend

1. JavaScript – Vue.js Framework
2. Vuetify 3 - Vuetify Framework
3. CSS - Cascading Style Sheets
4. HTML - HyperText Markup Language

1.6.2 เครื่องมือสำหรับ Backend

1. Python – Django Framework

1.6.3 เครื่องมือสำหรับ Database

1. MySQL
2. XAMPP

1.6.4 เครื่องมือสำหรับการทดสอบ API

1. Postman

1.7 แผนการดำเนินงาน

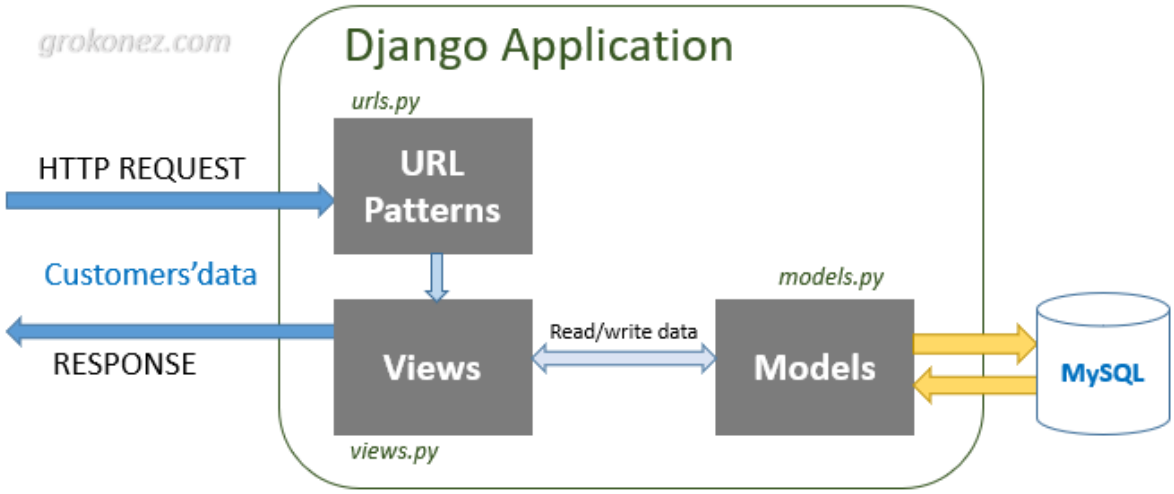
ตารางที่ 1.1 ตารางแผนการดำเนินงาน

กิจกรรม	ระยะเวลา พ.ศ. 2566-2567			
	ธันวาคม	มกราคม	กุมภาพันธ์	มีนาคม
1. ศึกษาข้อมูล	/			
2. ออกแบบและวิเคราะห์ระบบ	/	/		
3. พัฒนาระบบ		/	/	/
4. ทดสอบและแก้ไขระบบ			/	/
5. จัดทำเอกสาร				/

บทที่ 2
ระบบงานและทฤษฎีที่เกี่ยวข้อง

2.1 หลักการและทฤษฎีที่ใช้ในการพัฒนาโครงงาน

2.1.1 ทฤษฎีของ Django



รูปที่ 1.1 Architecture of Django

ที่มา : bezkoder. (2021). Django + Vue.js: CRUD App with Django Rest Framework from https://www.bezkoder.com/django-vue-js-rest-framework/

Django [1] เป็นเฟรมเวิร์กพัฒนาเว็บแอปพลิเคชัน (web framework) ภาษา Python ที่ออกแบบมาในรูปแบบของ Model-View-Controller (MVC) และโปรแกรมประยุกต์ Model-View-Template (MVT) ซึ่งเป็นพื้นฐานในการสร้างแอปพลิเคชันเว็บที่มีประสิทธิภาพ

หลักการทำงานแบ่งออกเป็น

1. Model

   รับผิดชอบในการจัดการกับฐานข้อมูล ซึ่งรวมถึงการสร้าง, อ่าน, ปรับปรุง, และลบ ข้อมูล (CRUD operations) ซึ่งช่วยในการจัดเก็บข้อมูลของแอปพลิเคชัน.

2. View

   จัดการกับการประมวลผลข้อมูลและการติดต่อกับ Model โดยจัดการกับการรับ ข้อมูลจากผู้ใช้และส่งผลลัพธ์กลับไปยังผู้ใช้หรือ Template.

3. Template

   รับผิดชอบในการแสดงผลและส่วน User Interface ของแอปพลิเคชัน โดยนำ ข้อมูลที่ได้มาจาก View และแสดงให้ผู้ใช้เห็น

2.1.2 ทฤษฎีของ Vue.js

Vue.js [2] เป็นเฟรมเวิร์ก JavaScript สำหรับสร้างอินเตอร์เฟซผู้ใช้ (user interface) โดยมีความสามารถในการเพิ่มความสามารถให้กับ HTML ในรูปแบบของ Model-View-ViewModel (MVVM).

หลักการทำงานแบ่งออกเป็น

1. Model

   เป็นส่วนที่เก็บข้อมูลที่จะถูกแสดงผลหรือนำมาใช้ในแอปพลิเคชัน โดยข้อมูลนี้สามารถเป็นตัวแปร, อ็อบเจ็กต์, หรือข้อมูลที่ได้จากการโหลดจากแหล่งข้อมูลอื่น ๆ เช่น API หรือฐานข้อมูล.

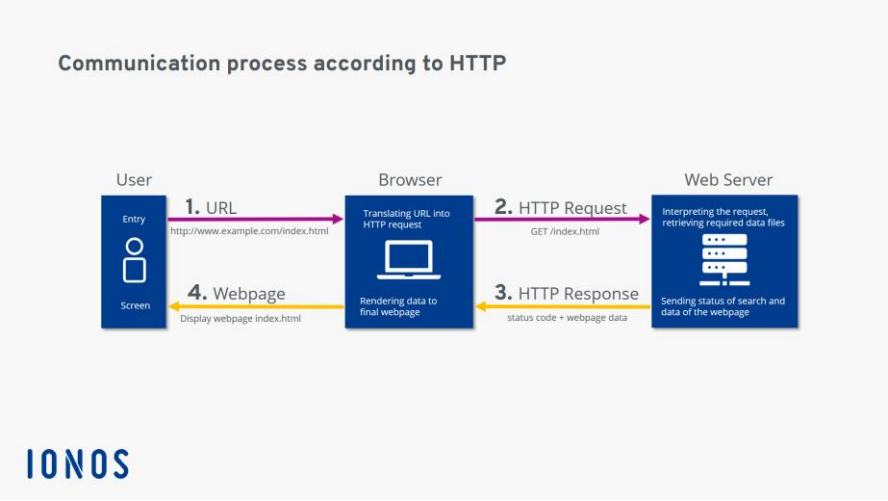
2. View

   เป็นส่วนที่รับผิดชอบในการแสดงผลข้อมูลที่ได้จาก Model ส่วนนี้ประกอบไปด้วย HTML, CSS, และคำสั่ง Vue.js ที่ใช้เพื่อให้ View สามารถเชื่อมโยงกับ Model ได้

3. View Model

   เป็นตัวกลางที่เชื่อมต่อ Model และ View ในกระบวนการทำงาน โดยที่ View Model รับข้อมูลจาก Model และส่งต่อให้ View แสดงผล

2.1.3 ทฤษฎีของ HTTP (HyperText Transfer Protocol)



รูปที่ 2 Communication process according to HTTP

ที่มา : Digital Guide. (2017). What is HTTP?, from https://www.ionos.co.uk/digitalguide/hosting/technical-matters/what-is-http/

HTTP [3] เป็นโปรโตคอลที่ใช้ในการสื่อสารระหว่างเว็บเบราว์เซอร์ (Client) ละเซิร์ฟเวอร์ (Server) ในโลกอินเทอร์เน็ต โดยมีลักษณะการทำงานดังนี้

1. HTTP เป็นโปรโตคอลแบบ Stateless ซึ่งหมายความว่าแต่ละคำขอ (Request) ที่ส่งไปยังเซิร์ฟเวอร์เป็นอิสระต่อกันและเซิร์ฟเวอร์ไม่จำจองข้อมูลสถานะ (State) ของผู้ใช้ระหว่างคำขอต่าง ๆ
2. HTTP แบ่งการสื่อสารออกเป็นคำขอ (Request) จากเว็บเบราว์เซอร์ไปยังเซิร์ฟเวอร์ คือ GET, POST, PUT, DELETE และคำตอบ (Response) จากเซิร์ฟเวอร์กลับไปยังเว็บเบราว์เซอร์
3. HTTP แบ่งหัวข้อคำขอ(Request) เป็น Headers, ข้อมูลเพิ่มเติม (Body)
4. HTTP แบ่งหัวข้อคำตอบ(Response) เป็นสถานะของคำขอ (Status Line), Headers, และข้อมูลเพิ่มเติม (Body)
5. HTTP ไม่เก็บข้อมูลสถานะ (State) ของผู้ใช้ ระหว่างคำขอต่าง ๆ ซึ่งทำให้ง่ายต่อการทำงานแบบ Scale-out และโครงสร้างเว็บแอปพลิเคชันแบบแยกชั้น (Layered Architecture)

2.1.4 JSON Web Token

เป็นรูปแบบของ token ที่ใช้สำหรับรับรองตัวตนและการเข้าถึงแอปพลิเคชัน ซึ่งประกอบด้วยสามส่วนหลักคือ Header, Payload และ Signature

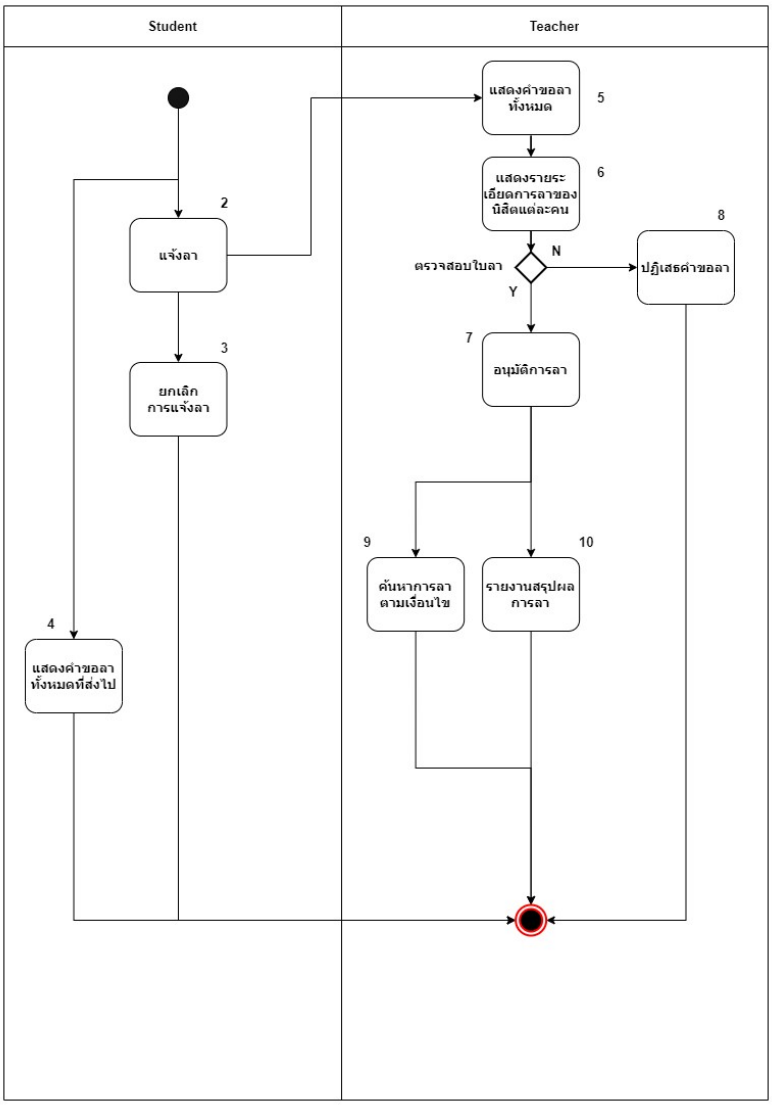
1. **Header** ของ JWT [4] เป็นอ็อบเจกต์ JSON ที่ประกอบด้วยสองสมาชิกคือ "alg" ซึ่งระบุวิธีการเข้ารหัสและ "typ" ซึ่งระบุประเภทของ token ซึ่งจะมีค่าคงที่เป็น "JWT"
2. **Payload** ของ JWT เป็นอ็อบเจกต์ JSON ที่ประกอบด้วยข้อมูลที่เราต้องการเก็บใน token
3. **Signature** ของ JWT เป็นส่วนที่ใช้สำหรับตรวจสอบว่า token ไม่ถูกแก้ไข

บทที่ 3
การวิเคราะห์และออกแบบระบบ

3.1 การวิเคราะห์และออกแบบระบบ

การจัดการแสดงผลกิจกรรมผู้ใช้ เป็นการนำเสนอข้อมูลเกี่ยวกับกิจกรรมและพฤติกรรมของผู้ใช้ในระบบลา โดยจะมีการแบ่งการกระทำของผู้ใช้ที่เกี่ยวข้องแต่ละระบบอย่างชัดเจน โดยผู้ที่เกี่ยวข้องกับระบบลามีดังนี้

1. นิสิต
2. อาจารย์



รูปที่ 3.1 แสดงแผนภาพการทำงาน หรือ work flow หรือ Activity diagram ของระบบจัดการการลาป่วยลากิจ

3.2 Input/Output Specification

ตารางที่ 3.1 ข้อกำหนดและผลการใช้งานข้อมูลขาเข้า/ขาออก

Function	Input	Process	Output
login	– username – password	ตรวจสอบข้อมูลการเข้าสู่ระบบ	– เข้าสู่หน้า Home
requestLeave	– subjects – start_date – end_date – leave_type – leave_document – description – creation_date – student_id	บันทึกรายละเอียดการลาลงในฐานข้อมูล (Record leave details in the database)	– ข้อความบันทึกสำเร็จ
cancelLeaveRequest	– leave_request_id	แก้ไขสถานะใบลาเป็น “ยกเลิก” ลงในฐานข้อมูล (update the status of the leave request to "ยกเลิก" in the database)	– ข้อความยืนยันการยกเลิกการลา – ข้อความแจ้งเกิดข้อผิดพลาด
showAllLeaveRequestsForAllSubjects		ดึงข้อมูลคำขอลาทั้งหมดจากฐานข้อมูล (retrieve all leave requests from the database)	– รายการคำขอลาทั้งหมดที่ส่งไปทุกวิชา
showAllLeaveRequests		ดึงข้อมูลคำขอลาทั้งหมดที่ส่งหาอาจารย์จากฐานข้อมูล (retrieve all requests sent to a specific teacher from the database)	– รายการคำขอลาทั้งหมด

Function	Input	Process	Output
showLeaveDetailsForStudent	– student_id	ดึงข้อมูลการลาของนิสิตจากฐานข้อมูล (retrieve leave details for student from the database)	– รายละเอียดการลาของนิสิตคนนั้นๆ
approveLeave	leavePaper_id	แก้ไขสถานะใบลาเป็น “อนุมัติ” ลงในฐานข้อมูล (update the status of the leave request to "อนุมัติ" in the database)	– ข้อความยืนยันการอนุมัติการลา – ข้อความแจ้งเกิดข้อผิดพลาด
rejectLeave	– leavePaper_id	แก้ไขสถานะใบลาเป็น “ไม่อนุมัติ” ลงในฐานข้อมูล (update the status of the leave request to "ไม่อนุมัติ" in the database)	– ข้อความยืนยันการอนุมัติการลา – ข้อความแจ้งเกิดข้อผิดพลาด
searchLeaveByCriteria	– name – subject – leaveType – status	ค้นหาคำขอลาที่ตรงกับเงื่อนไขจากฐานข้อมูล (search for leave requests that match the criteria from the database)	– รายการคำขอลาที่ตรงกับเงื่อนไข
leaveSummaryReport		สรุปข้อมูลการลาตามระยะเวลาหรือวิชาเพื่อนำมาสร้างรายงาน	สรุปข้อมูลการลาตามระยะเวลาหรือวิชาเพื่อนำมาสร้างรายงาน

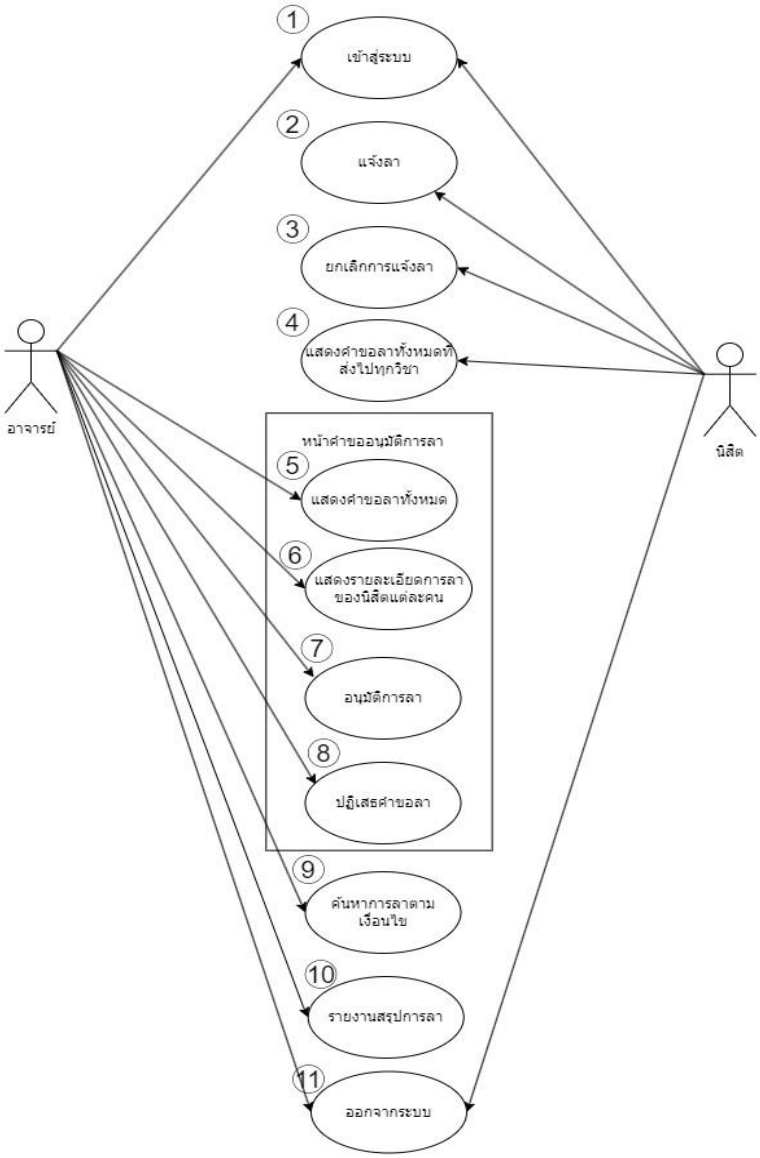
Function	Input	Process	Output
logout	–	ลบข้อมูลเซสชันของผู้ใช้และออกจากระบบ	– ไปที่หน้า Login

3.3 Use case diagram

Use Case Diagram ของระบบลาในมหาวิทยาลัยเป็นแผนภาพที่ช่วยแสดงฟังก์ชันหลักที่เกี่ยวข้องกับกระบวนการลาของนิสิต แผนภาพนี้จะช่วยให้เข้าใจฟังก์ชันหลักที่นิสิตและอาจารย์สามารถใช้งานได้ โดยมีผู้เกี่ยวข้องกับระบบดังนี้.

ผู้เกี่ยวข้อง (Actors):

1. นิสิต (Student): นิสิตที่ต้องการใช้ระบบลา
2. อาจารย์ (Teacher): อาจารย์



รูปที่ 3.2 แสดงแผนภาพยูสเคส

3.4 การวิเคราะห์ขั้นตอนการทำงานของฟังก์ชันโดยใช้ยูสเคส

การวิเคราะห์การทำงานของโปรแกรมช่วยให้เราสามารถมองเห็นภาพรวมของโปรแกรมได้ดีขึ้น นอกขากนี้ยังสามารถช่วยเรื่องข้อผิดพลาดที่เกิดขึ้นได้ เพราะมีขั้นตอนในการดำเนินของโปรแกรมระบุไว้อย่างชัดเจนจึงช่วยให้สามารถพัฒนาโปรแกรมให้มีประสิทธิภาพได้

ตารางที่ 3.2 Use case description เข้าสู่ระบบ

Use case ID	1	
Use case Name	เข้าสู่ระบบ	
Actor	นิสิต/อาจารย์	
Description	นิสิต/อาจารย์ ทำการเข้าสู่ระบบ	
Precondition	มีบัญชีผู้ใช้ในระบบ	
Process	**นิสิต/อาจารย์**	**ระบบ**
	2. นิสิต/อาจารย์ กรอก username และ password และกดปุ่ม “เข้าสู่ระบบ”	1. แสดงหน้าเข้าสู่ระบบ 3. Frontend ทำการตรวจสอบข้อมูลว่าค่า username และ password เป็นช่องว่างหรือไม่ หลังจากตรวจสอบเสร็จส่งข้อมูลให้ Backend 4. Backend ทำการตรวจสอบข้อมูลที่ได้รับและส่งคำสั่งค้นหาว่ามีบัญชีผู้ใช้ที่เข้าสู่ระบบจริงหรือไม่ ไปยัง Database 5. Database ทำการค้นหาและพบบัญชีผู้ใช้ที่ตรงกับข้อมูลที่ส่งมา แล้วส่งค่ากลับมาให้ Backend 6. Backend ส่ง Response กลับไปที่ Frontend เพื่อแจ้งผลการทำงาน 7. Frontend Redirect ไปที่หน้า Home

ตารางที่ 3.3 Use case description แจ้งลา

Use case ID	2	
Use case Name	แจ้งลา	
Actor	นิสิต	
Description	นิสิตส่งคำขอลาพร้อมแนบหลักฐานการลาส่งให้อาจารย์ประจำวิชา	
Precondition	เข้าสู่ระบบและเลือกเมนูการแจ้งลา	
Process	**นิสิต**	**ระบบ**
		1. ดึงข้อมูลนิสิตและแสดงข้อมูลของนิสิต 2.ดึงข้อมูลรายวิชาที่นิสิตลงทะเบียนเรียน
	3. นิสิตเลือก “รายชื่อวิชา” ที่ต้องการลา 4. นิสิตเลือก “หมู่เรียน” ที่ต้องการลา 5. นิสิตเลือก “วันที่” ที่ต้องการลา 6. นิสิตเลือก “เวลา” ที่ต้องการลา 7. นิสิตเลือก “ประเภทการลา” ที่ต้องการลา 8. นิสิตกดปุ่ม “img” เพื่อแนบหลักฐานการลา 9. นิสิตกรอกรายละเอียดเหตุผลการลาในช่อง “คำอธิบาย” 10. นิสิตกดปุ่ม “ส่ง”	11. ตรวจสอบข้อมูลใน Form ผ่าน Frontend 11.1 วันที่และเวลา ต้องเป็นตัวเลข 11.2 ต้องกรอกข้อมูลครบทุกช่อง 12. Verify ข้อมูล 12.1 ข้อมูลทั้งหมดต้อง not null 13. Frontend ส่งคำขอบันทึกข้อมูลไปยัง Backend ผ่าน REST API. 14. Backend ทำการตรวจสอบข้อมูลที่ได้รับและทำการบันทึกข้อมูลลงในฐานข้อมูล. 15.เมื่อการบันทึกข้อมูลเสร็จสมบูรณ์ Backend จะส่ง response กลับมายัง Frontend เพื่อแจ้งผลการทำงาน success 16 ระบบแจ้งว่า “บันทึกสำเร็จ”
	17. นิสิตกดปิด Popup	

Alternative [1]		1. ดึงข้อมูลนิสิตและแสดงข้อมูลของนิสิต 2.ดึงข้อมูลรายวิชาที่นิสิตลงทะเบียนเรียน
	3-9. นิสิตไม่ เลือก/กรอก “รายวิชา”, “หมู่เรียน”, “วันที่”, “เวลา”, “ประเภทการลา”, “IMG” หรือ คำอธิบาย	
	10. นิสิตกดปุ่ม “ส่ง”	11. ตรวจสอบข้อมูลใน Form ผ่าน Frontend 11.1 วันที่และเวลา ต้องเป็นตัวเลข 11.2 ต้องกรอกข้อมูลครบทุกช่อง 12. Verify ข้อมูล 12.1 ข้อมูลทั้งหมดต้อง not null 13. Frontend ส่งคำขอบันทึกข้อมูลไปยัง Backend ผ่าน REST API. 14. Backend ทำการตรวจสอบข้อมูลที่ได้รับและทำการบันทึกข้อมูลลงในฐานข้อมูล. 15.เมื่อการบันทึกข้อมูลเสร็จสมบูรณ์ Backend จะส่ง response กลับมายัง Frontend เพื่อแจ้งผลการทำงาน error 16 ระบบแจ้งว่า “เกิดข้อผิดพลาด”
	17. นิสิตกดปิด Popup	

ตารางที่ 3.4 Use case description ยกเลิกการแจ้งลา

Use case ID	3	
Use case Name	ยกเลิกการแจ้งลา	
Actor	นิสิต	
Description	นิสิตยกเลิกคำขอลาที่ทำการส่งให้อาจารย์ประจำวิชา	
Precondition	เข้าสู่ระบบและเลือกเมนูคำขอลาทั้งหมด	
Process	**นิสิต**	**ระบบ**
	3. นิสิตกดปุ่ม “ยกเลิก” ใบลาที่ต้องการยกเลิก 5. นิสิตกดปุ่ม “ใช่, ลบ”	1. ดึงข้อมูลนิสิตและแสดงข้อมูลของนิสิต 2.ดึงข้อมูลรายวิชาที่นิสิตส่งใบลาทุกใบ 4. แสดง pop up “ยืนการลบ” ใบลาที่นิสิตกด 6. Frontend ส่งคำขอบันทึกข้อมูลไปยัง Backend ผ่าน REST API. 7. Backend ทำการตรวจสอบข้อมูลที่ได้รับและทำการลบข้อมูลในฐานข้อมูล. 8.เมื่อการลบข้อมูลเสร็จสมบูรณ์ Backend จะส่ง response กลับมายัง Frontend เพื่อแจ้งผลการทำงาน 9 ระบบแจ้งว่า “ลบสำเร็จ”
Alternative [1]	3. นิสิตกดปุ่ม “ยกเลิก” ใบลาที่ต้องการยกเลิก 6. นิสิตกดปุ่ม “ยกเลิก”	1. ดึงข้อมูลนิสิตและแสดงข้อมูลของนิสิต 2. ดึงข้อมูลรายวิชาที่นิสิตส่งใบลาทุกใบ 4. ดึงข้อมูลใบลาที่นิสิตกดยกเลิก 5. แสดง pop up “ยืนการลบ” ใบลาที่นิสิตกด 7. ระบบปิด pop up

ตารางที่ 3.5 Use case description แสดงคำขอลาทั้งหมดที่ส่งไปทุกวิชา

Use case ID	4	
Use case Name	แสดงคำขอลาทั้งหมดที่ส่งไปทุกวิชา	
Actor	นิสิต	
Description	แสดงคำขอลาทั้งหมดที่นิสิตส่งไปทุกวิชาในภาคเรียนปัจจุบัน	
Precondition	เข้าสู่ระบบและเลือกเมนูคำขอลาทั้งหมด	
Process	**นิสิต**	**ระบบ**
	3.นิสิตค้นหาโดย “ชื่อวิชา”	1. ดึงข้อมูลนิสิตและแสดงข้อมูลของนิสิต 2.ดึงข้อมูลรายวิชาที่นิสิตส่งใบลาทุกใบ 4. Frontend ทำการกรองข้อมูลโดยชื่อวิชา 5. หน้าบ้าน อัปเดตตารางข้อมูลโดยแสดงข้อมูลเป็นชื่อวิชาที่นิสิตค้นหา
Alternative[1]	3.นิสิตค้นหาโดย “ประเภทการลา”	1. ดึงข้อมูลนิสิตและแสดงข้อมูลของนิสิต 2. ดึงข้อมูลรายวิชาที่นิสิตส่งใบลาทุกใบ 4. Frontend ทำการกรองข้อมูลโดยใช้ประเภทการลา 5. Frontend อัปเดตตารางข้อมูลโดยแสดงข้อมูลเป็นชื่อวิชาที่นิสิตค้นหาทันที

ตารางที่ 3.6 Use case description แสดงคำขอลาทั้งหมด

Use case ID	5	
Use case Name	แสดงคำขอลาทั้งหมด	
Actor	อาจารย์	
Description	แสดงคำขอลาทั้งหมดที่นิสิตส่งมาให้อาจารย์	
Precondition	เข้าสู่ระบบ และ เลือกเมนู “คำขอลาทั้งหมด”	
Process	**อาจารย์**	**ระบบ**
		1.ส่งคำขอดึงข้อมูลใบลาทุกใบที่นิสิตส่งมาจาก Backend 2. Backend ส่งคำขอดึงข้อมูลไปที่ฐานข้อมูล 3.เมื่อดึงข้อมูลสำเร็จ Backend จะส่งข้อมูลตอบกลับไปให้ Frontend 4. Frontend แสดงผลข้อมูลที่ได้รับ

ตารางที่ 3.7 Use case description แสดงรายละเอียดการลาของนิสิตแต่ละคน

Use case ID	6	
Use case Name	แสดงรายละเอียดการลาของนิสิตแต่ละคน	
Actor	อาจารย์	
Description	แสดงรายละเอียดการลาของนิสิตแต่ละคน	
Precondition	เข้าสู่ระบบ และ เลือกเมนู “คำขอลาทั้งหมด”	
Process	**อาจารย์**	**ระบบ**
	2. กดเลือก “ชื่อ” ที่ต้องการแสดงรายละเอียด	1.แสดงข้อมูลใบลาทุกใบที่นิสิตส่งมา 3. Redirect ไปที่หน้า “Show detail leave request.” 4.แสดงข้อมูลใบลาตาม “ชื่อ” ที่เลือกมา

ตารางที่ 3.8 Use case description อนุมัติการลา

Use case ID	7	
Use case Name	อนุมัติการลา	
Actor	อาจารย์	
Description	อนุมัติคำขอลาที่นิสิตส่งมาในหน้า “คำขอลาทั้งหมด”	
Precondition	เข้าสู่ระบบ และ เลือกเมนู “คำขอลาทั้งหมด”	
Process	**อาจารย์**	**ระบบ**
	2.ดูรายละเอียดการลา 2.1 รหัสประจำตัว 2.2 รายชื่อนิสิต 2.3 รายชื่อวิชา 2.4 หมู่เรียน 2.5 วันที่ 2.6 เวลา 2.7 ประเภทการลา 2.8 สถานะการอนุมัติ 3. กดเลือก “อนุมัติ”	1.แสดงข้อมูลใบลาทุกใบที่นิสิตส่งมา 4. แก้ไขสถานะใบลาเป็น “อนุมัติ” 5. อัปเดตการแสดงผลของตารางใบลา
Alternative[1]	2. กดเลือก “ชื่อ” ที่ต้องการแสดงรายละเอียด 5.ดูรายละเอียดการลา 5.1 ชื่อ 5.2 รหัสประจำตัว 5.3 รายวิชา 5.4 วันที่ 5.5 หลักฐานการลา 5.6 ประเภทการลา 5.7 คำอธิบาย 6. กดเลือก “อนุมัติ”	1.แสดงข้อมูลใบลาทุกใบที่นิสิตส่งมา 3. Redirect ไปที่หน้า “Show detail leave request.” 4.แสดงข้อมูลใบลาตาม “ชื่อ” ที่เลือกมาก 7. แก้ไขสถานะใบลาเป็น “อนุมัติ” 8. Redirect ไปที่หน้า “Show all leave request”

ตารางที่ 3.9 Use case description ปฏิเสธคำขอลา

Use case ID	8	
Use case Name	แสดงรายละเอียดการลาของนิสิตแต่ละคน	
Actor	อาจารย์	
Description	ปฏิเสธคำขอลาที่นิสิตส่งมาในหน้า “คำขอลาทั้งหมด”	
Precondition	เลือกเมนู “คำขอลาทั้งหมด”	
Process	**อาจารย์**	**ระบบ**
	2.ดูรายละเอียดการลา 2.1 รหัสประจำตัว 2.2 รายชื่อนิสิต 2.3 รายชื่อวิชา 2.4 หมู่เรียน 2.5 วันที่ 2.6 เวลา 2.7 ประเภทการลา 2.8 สถานะการอนุมัติ 3. กดเลือก “ไม่อนุมัติ”	1.แสดงข้อมูลใบลาทุกใบที่นิสิตส่งมา 4. แก้ไขสถานะใบลาเป็น “ไม่อนุมัติ” 5. อัปเดตการแสดงผลของตารางใบลา
Alternative[1]	2. กดเลือก “ชื่อ” ที่ต้องการแสดงรายละเอียด 5.ดูรายละเอียดการลา 5.1 ชื่อ 5.2 รหัสประจำตัว 5.3 รายวิชา 5.4 วันที่ 5.5 หลักฐานการลา 5.6 ประเภทการลา 5.7 คำอธิบาย 6. กดเลือก “ไม่อนุมัติ”	1.แสดงข้อมูลใบลาทุกใบที่นิสิตส่งมา 3. Redirect ไปที่หน้า “Show detail leave request” 4.แสดงข้อมูลใบลาตาม “ชื่อ” ที่เลือกมา 7. แก้ไขสถานะใบลาเป็น “ไม่อนุมัติ” 8. Redirect ไปที่หน้า “Show all leave request”

ตารางที่ 3.10 Use case description ค้นหาการลาตามเงื่อนไข

Use case ID	9	
Use case Name	ค้นหาการลาตามเงื่อนไข	
Actor	อาจารย์	
Description	ค้นหาคำขอลาตามเงื่อนไขที่ต้องการ	
Precondition	เลือกเมนู “ค้นหาการลาตามเงื่อนไข”	
Process	**อาจารย์**	**ระบบ**
	2. กดค้นหาโดยใช้ ชื่อนิสิต	1.แสดงข้อมูลใบลาทั้งหมด 3. แสดงข้อมูลใบลาทั้งหมดตาม “ชื่อ” ที่ค้นหา
Alternative [1]	2. กดค้นหาโดยใช้ วิชา	1.แสดงข้อมูลใบลาทั้งหมด 3. แสดงข้อมูลใบลาทั้งหมดตาม “วิชา” ที่ค้นหา

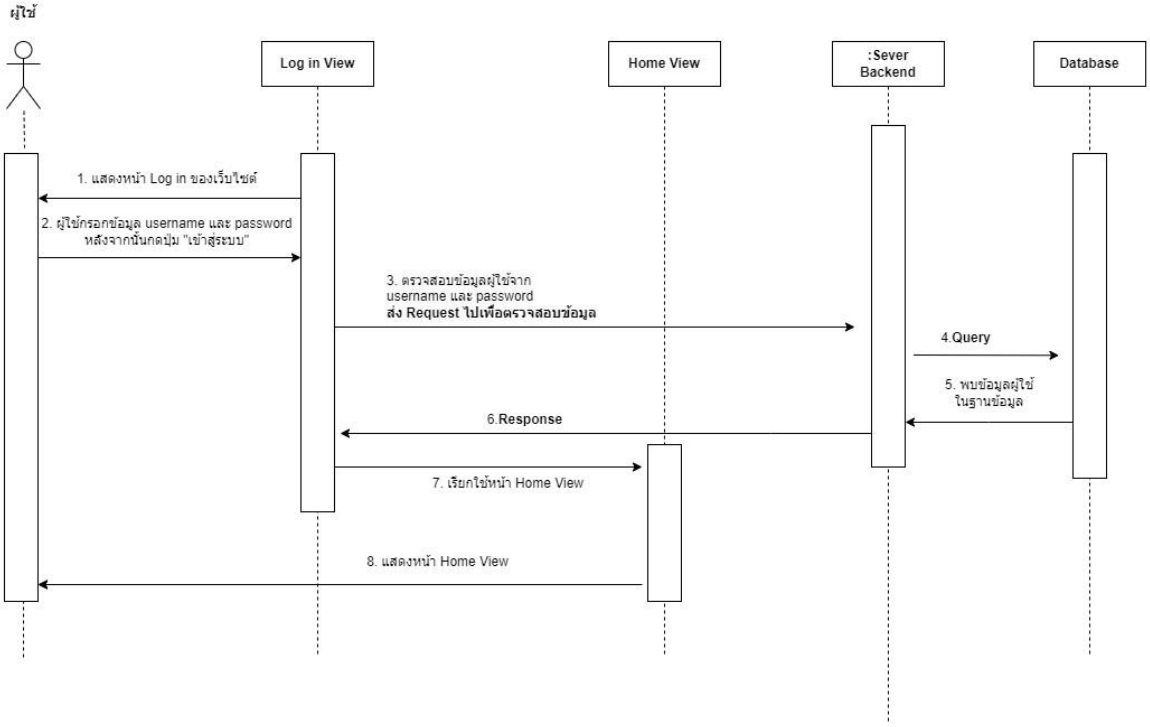
ตารางที่ 3.11 Use case description รายงานสรุปผลการลา

Use case ID	10	
Use case Name	รายงานสรุปผลการลา	
Actor	อาจารย์	
Description	สรุปผลการลาทุกวิชาที่อาจารย์สอน	
Precondition	เลือกเมนู “รายงานสรุปผลการลา”	
Process	**อาจารย์**	**ระบบ**
	3.อาจารย์กดเลือกค้นหา “วิชา” ที่ตนสอน	1.แสดงข้อมูลใบลาทุกวิชาที่อาจารย์สอน 2.ระบบสรุปผลการลาแต่ละประเภทการลาตามวิชาที่อาจารย์สอน 4.ระบบทำการสรุปผลตามวิชาที่อาจารย์เลือก
Alternative [1]	3. อาจารย์กดเลือกค้นหา “หมู่เรียน” ที่ตนสอน	1.แสดงข้อมูลใบลาทุกวิชาที่อาจารย์สอน 2.ระบบสรุปผลการลาแต่ละประเภทการลาตามวิชาที่อาจารย์สอน 4.ระบบทำการสรุปผลตามหมู่เรียนที่อาจารย์เลือก

3.5 ซีเควนไดอะแกรม

3.5.1 ซีเควนไดอะแกรม เข้าสู่ระบบ

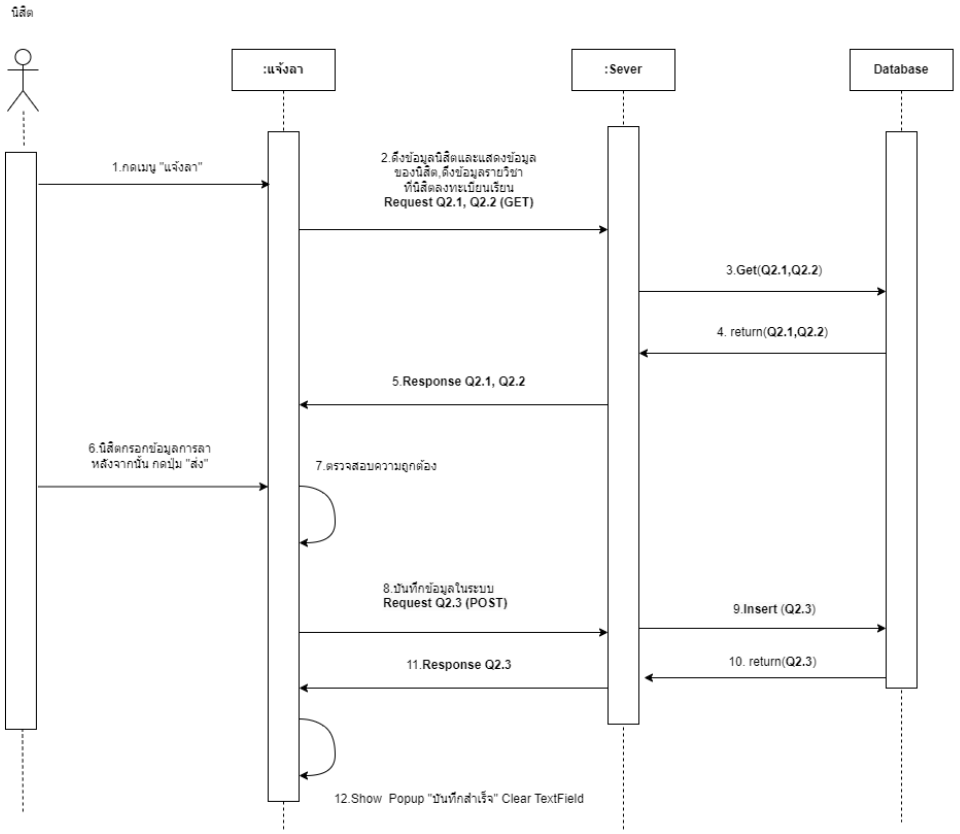
ซีเควนไดอะแกรมเข้าสู่ระบบเป็นแผนภาพที่แสดงลำดับขั้นตอนการเข้าสู่ระบบของผู้ใช้ในระบบซอฟต์แวร์หรือแอปพลิเคชัน โดยเน้นการตรวจสอบข้อมูลการรับรองตัวตนก่อนการเข้าสู่ระบบ เพื่อให้การเข้าถึงและใช้งานระบบเป็นไปตามหลักการ



รูปที่ 3.3 แสดงแผนภาพซีเควนไดอะแกรม 1

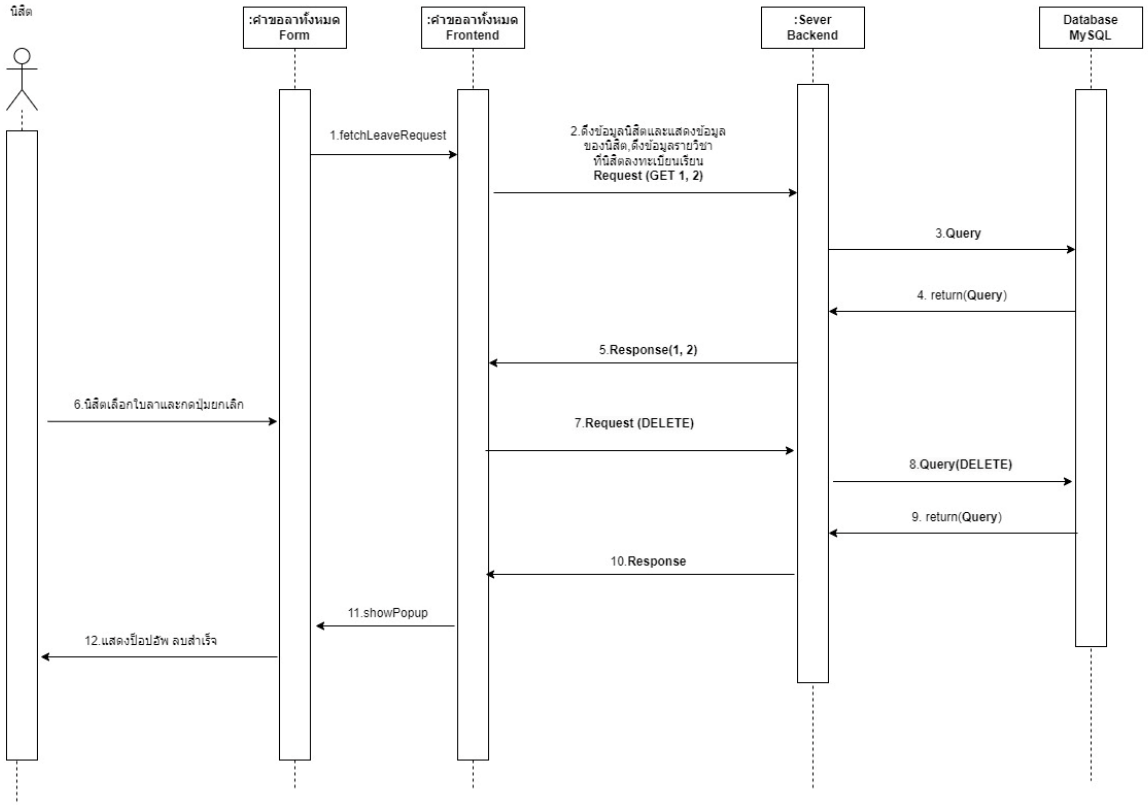
3.5.2 ซีเควนไดอะแกรม แจ้งลา

การแจ้งลาในระบบซอฟต์แวร์หรือแอปพลิเคชัน โดยมุ่งเน้นให้เห็นการโต้ตอบระหว่างผู้ใช้ที่ต้องการแจ้งลากับระบบ และขั้นตอนที่ต้องผ่านเพื่อให้การแจ้งลาเป็นไปตามข้อกำหนดและกระบวนการที่กำหนดไว้ในระบบนั้น ๆ เช่น การเลือกประเภทของลา, การระบุวันที่และเวลาที่ต้องการลา, การกรอกรายละเอียดเพิ่มเติม, และขั้นตอนการอนุมัติหรือปฏิเสธการลา ซึ่งช่วยให้การจัดการการลาเป็นไปอย่างมีประสิทธิภาพและเป็นระบบได้อย่างชัดเจนและเรียบง่าย



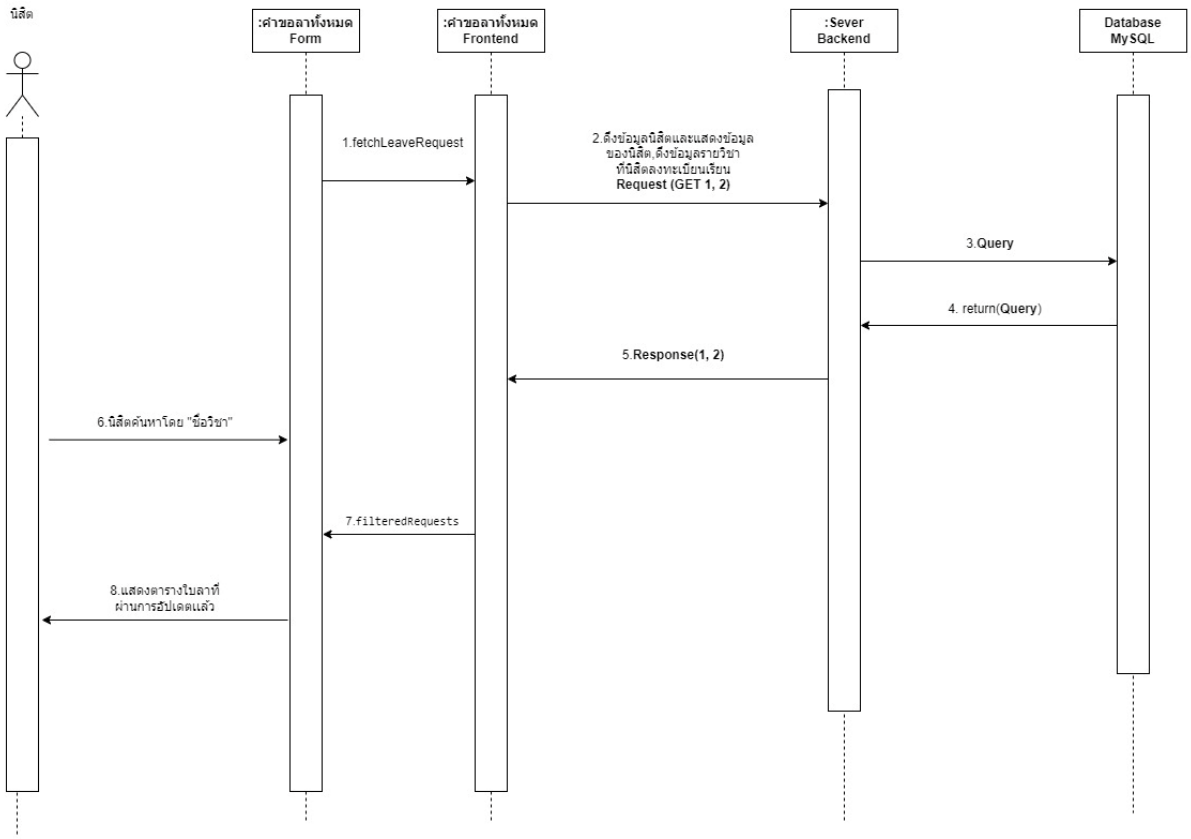
รูปที่ 3.4 แสดงแผนภาพซีเควนไดอะแกรม 2

3.5.3 ซีเควนไดอะแกรม ยกเลิกการแจ้งลา



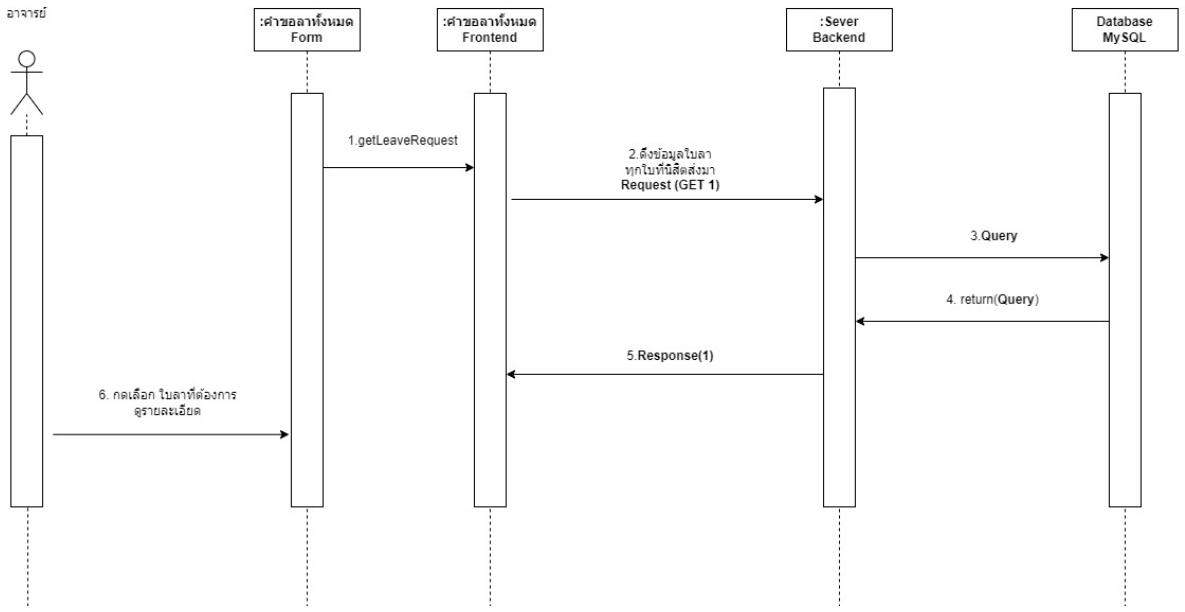
รูปที่ 3.5 แสดงแผนภาพซีเควนไดอะแกรม 3

3.5.4 ซีเควนไดอะแกรม แสดงคำขอลาทั้งหมดที่ส่งไปทุกวิชา

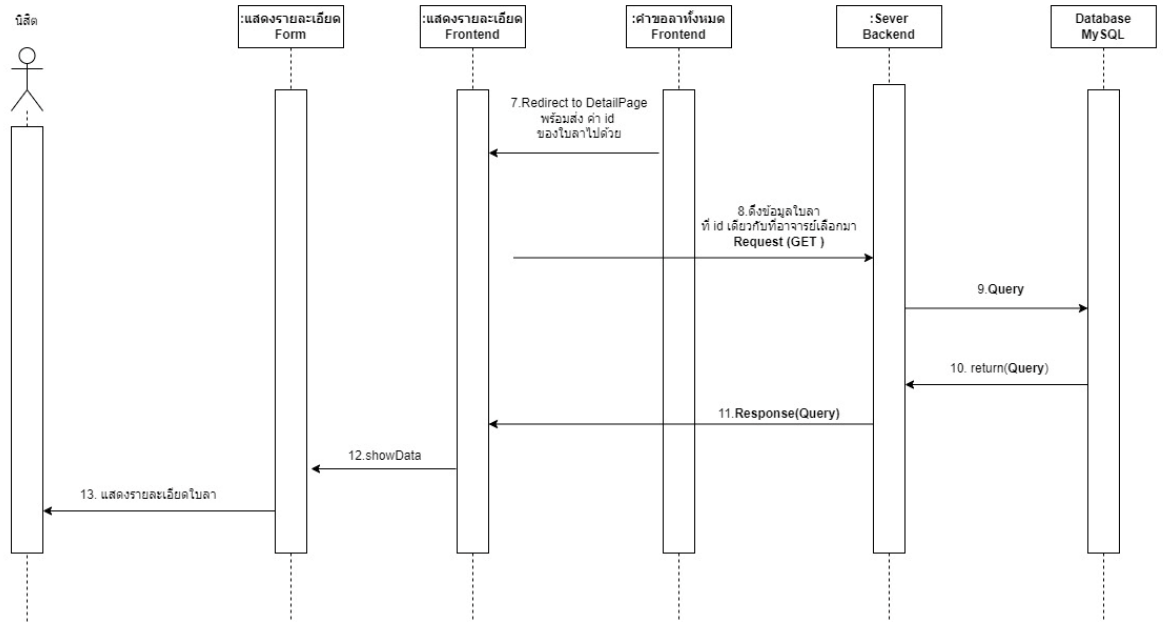


รูปที่ 3.6 แสดงแผนภาพซีเควนไดอะแกรม 4

3.5.5 ซีเควนไดอะแกรม แสดงรายละเอียดการลาของนิสิตแต่ละคน

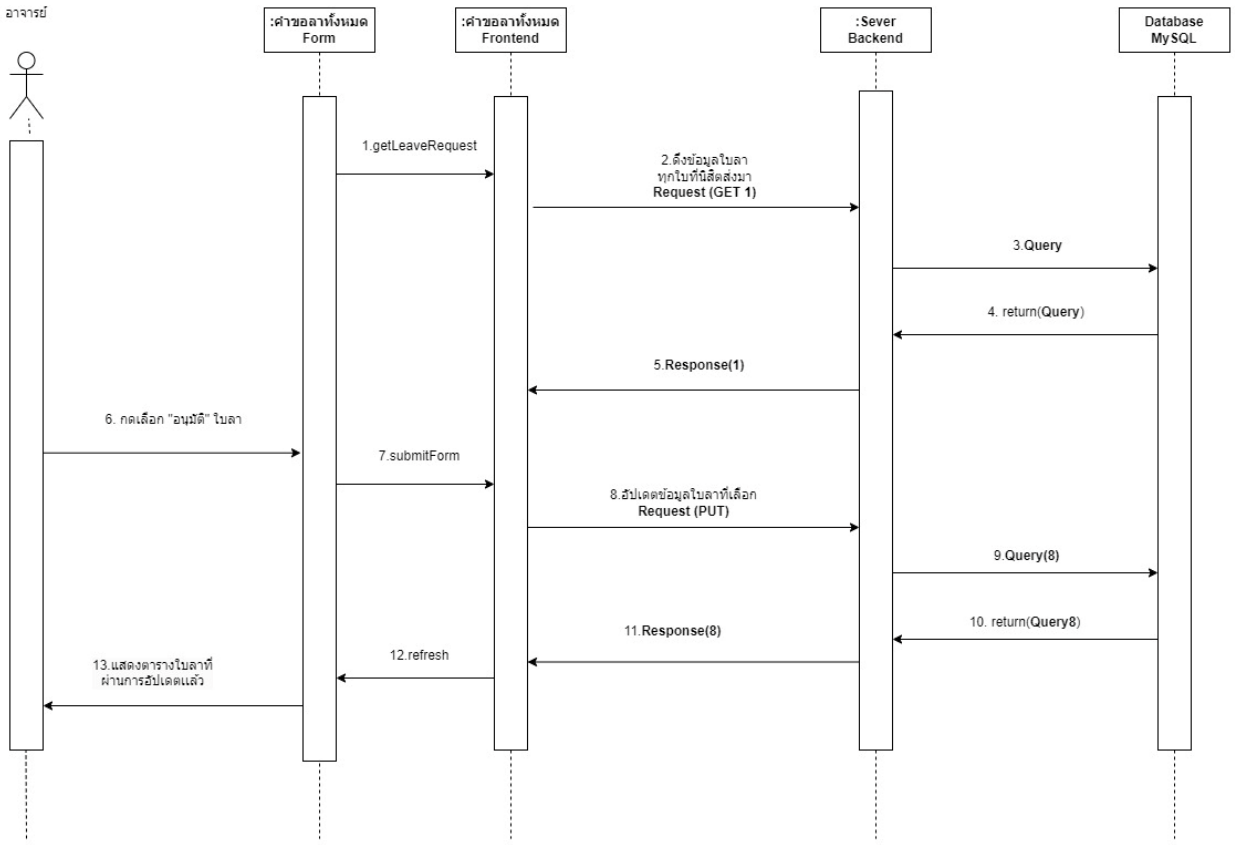


รูปที่ 3.7 แสดงแผนภาพซีเควนไดอะแกรม 5



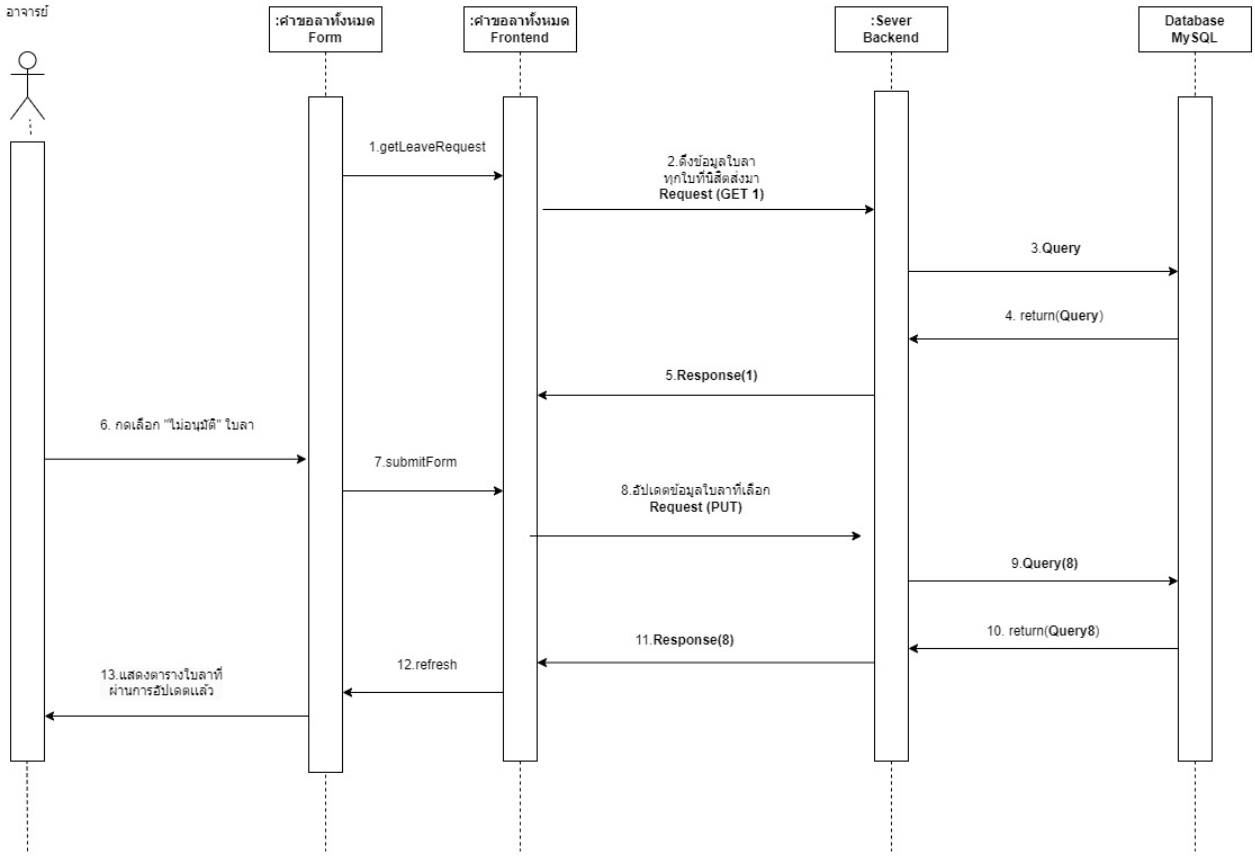
รูปที่ 3.8 แสดงแผนภาพซีเควนไดอะแกรม 5

3.5.6 ซีเควนไดอะแกรม อนุมัติการลา



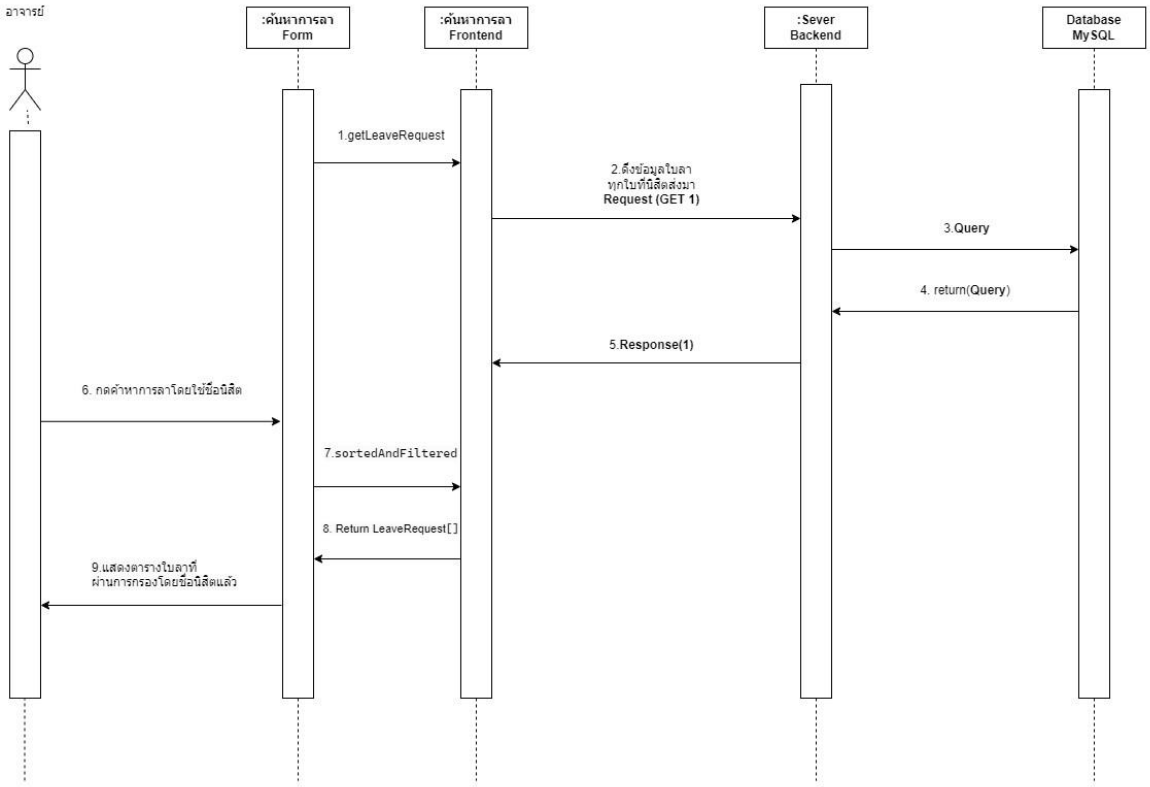
รูปที่ 3.9 แสดงแผนภาพซีเควนไดอะแกรม 6

3.5.7 ซีเควนไดอะแกรม ไม่อนุมัติการลา



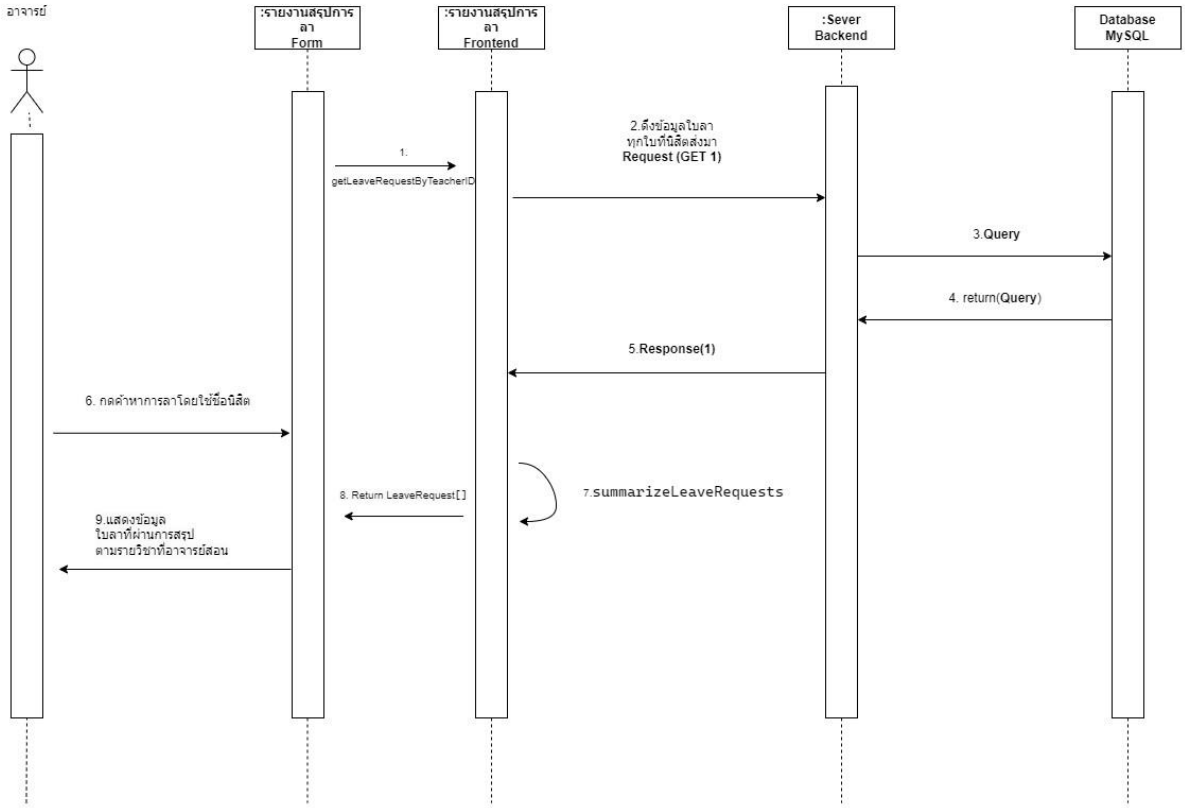
รูปที่ 3.10 แสดงแผนภาพซีเควนไดอะแกรม 7

3.5.8 ซีเควนไดอะแกรม ค้นหาการลา



รูปที่ 3.11 แสดงแผนภาพซีเควนไดอะแกรม 8

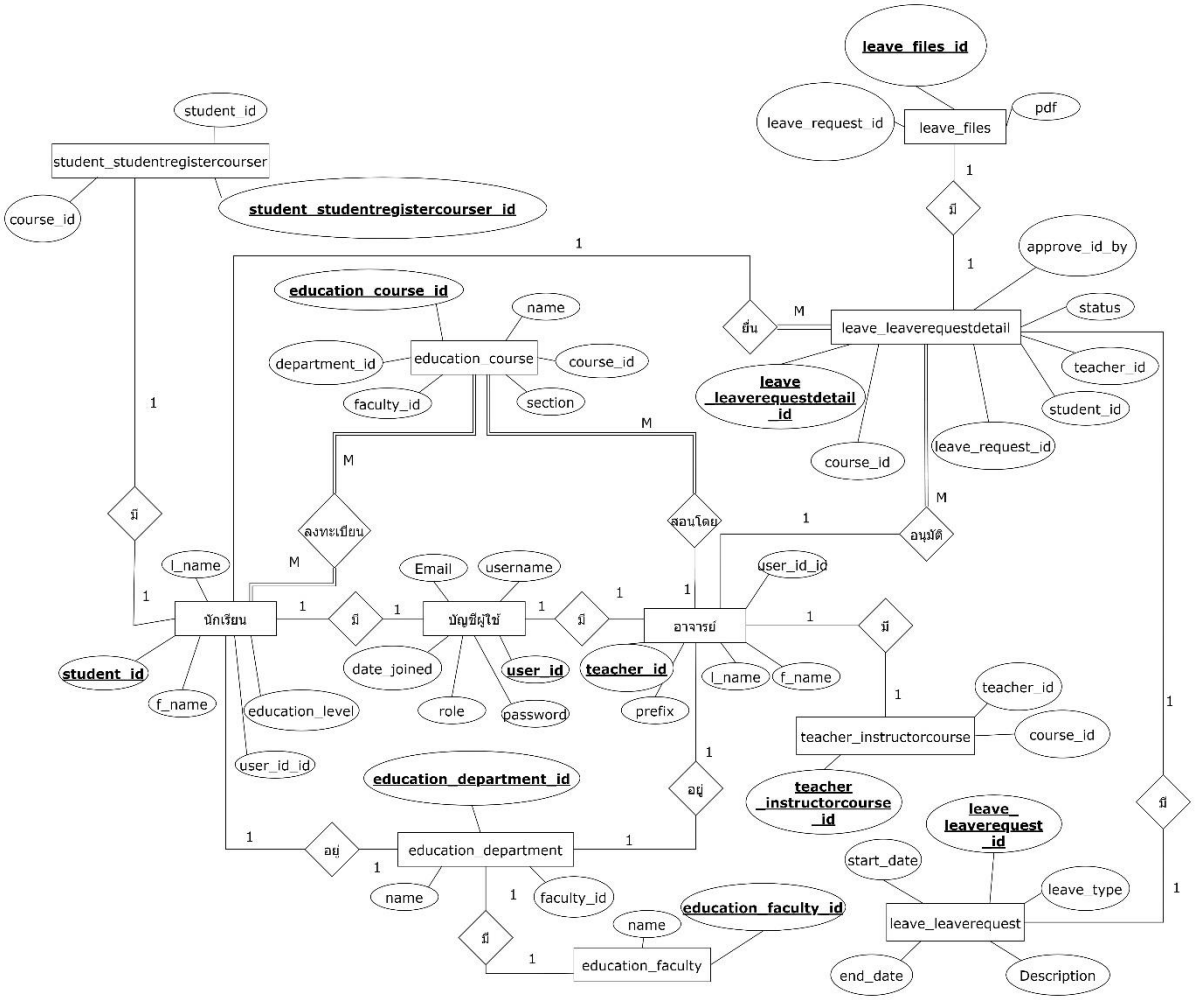
3.5.9 ซีเควนไดอะแกรม รายงานสรุปผลการลา



รูปที่ 3.12 แสดงแผนภาพซีเควนไดอะแกรม 9

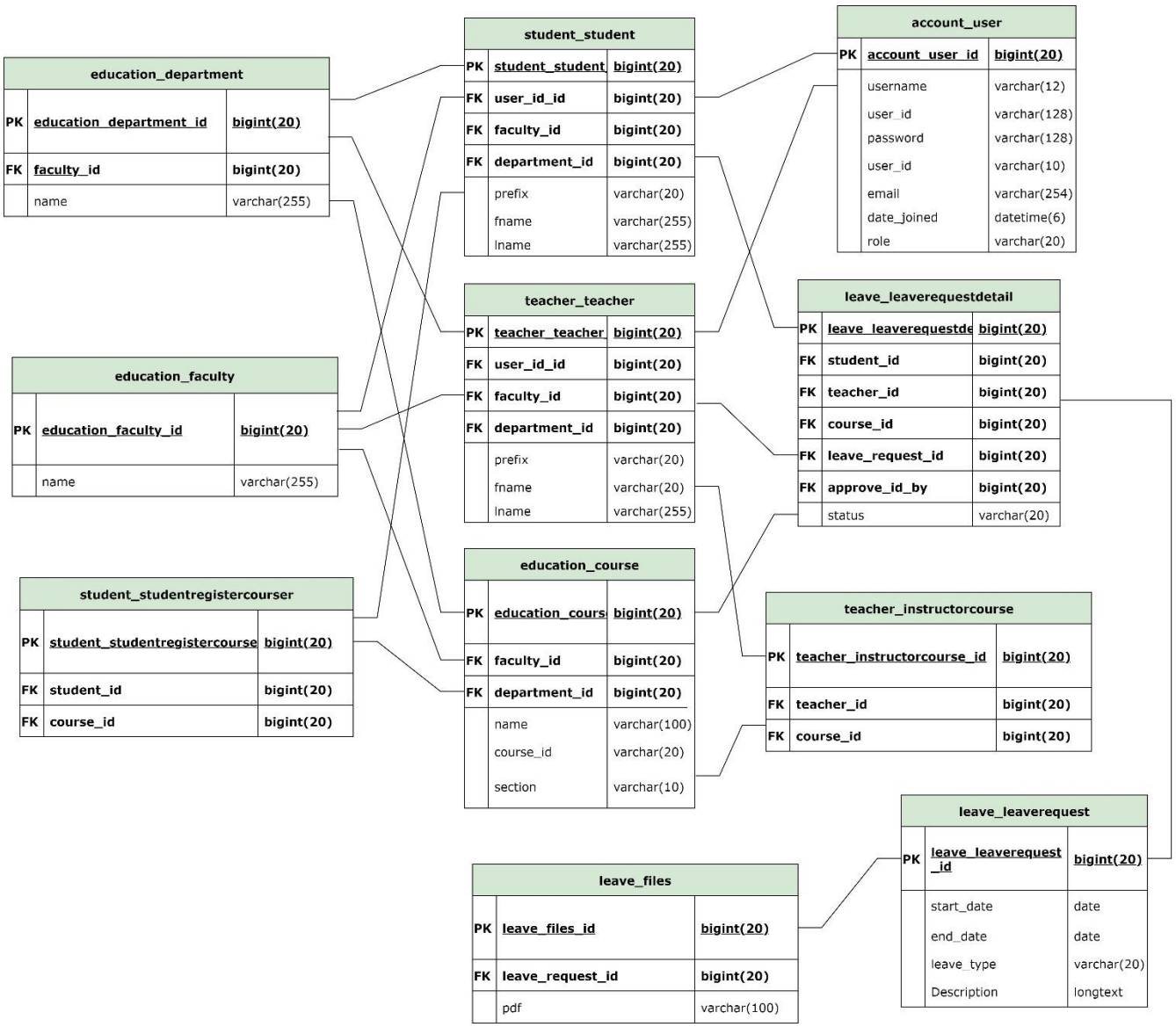
3.6 การออกแบบฐานข้อมูล

3.6.1 แผนภาพการออกแบบฐานข้อมูลระดับแนวคิด



รูปที่ 3.13 แสดงแผนภาพการออกแบบฐานข้อมูลระดับแนวคิด โดย ER-diagram

3.6.2 แผนภาพการออกแบบฐานข้อมูลระดับกายภาพ



รูปที่ 3.14 แสดงแผนภาพการออกแบบฐานข้อมูลระดับกายภาพ

3.6.3 พจนานุกรมข้อมูล

ตารางที่ 3.12 account_user

คำอธิบาย: ตารางเก็บข้อมูลล็อคอินของนิสิตและอาจารย์

Attribute	Data Type	Key	Reference Table	Description	Constraints	Example
id	bigint(20)	Primary Key	-	Object Identifier	Not Null	1
username	varchar(12)	-	-	ชื่อผู้ใช้	Not Null	b6310451260
password	varchar(128)	-	-	รหัสผ่าน	Not Null	“passwodeExample1234”
user_id	varchar(10)	-	-	รหัสประจำตัว	Not Null	6310451260
email	varchar(254)	-	-	อีเมล	Not Null	email@ku.th
date_joined	datetime(6)	-	-	วันที่สมัครบัญชี	Not Null	2024-02-02 16:53:05.965860
role	varchar(20)	-	-	ตำแหน่ง	Not Null	student

ตารางที่ 3.13 student_student

คำอธิบาย: ตารางเก็บข้อมูลของนิสิต

Attribute	Data Type	Key	Reference Table	Description	Constraints	Example
id	bigint(20)	Primary Key	-	Object Identifier	Not Null	1
prefix	varchar(20)	-	-	คำนำหน้า	Not Null	นาย
fname	varchar(255)	-	-	ชื่อนิสิต	Not Null	ชัชชัย
lname	varchar(255)	-	-	นามสกุลนิสิต	Not Null	ขจรศักดิ์โกศล
faculty_id	bigint(20)	Foreign Key	education_faculty	คณะ	Not Null	วิทยาศาสตร์
department_id	bigint(20)	Foreign Key	education_department	ภาควิชา	Not Null	วิทยาการคอมพิวเตอร์
user_id_id	bigint(20)	Foreign Key	account_user	รหัสบัญชีผู้ใช้งาน	Not Null	b6310451260

ตารางที่ 3.14 student_studentregistercourser

คำอธิบาย: ตารางเก็บข้อมูลของนิสิตลงทะเบียนรายวิชา

Attribute	Data Type	Key	Reference Table	Description	Constraints	Example
id	bigint(20)	Primary Key	-	Object Identifier	Not Null	1
student_id	bigint(20)	Foreign Key	student_student	รหัสนักเรียน	Not Null	1
course_id	bigint(20)	Foreign Key	education_course	รหัสวิชา	Not Null	1

ตารางที่ 3.15 teacher_teacher

คำอธิบาย: ตารางเก็บข้อมูลของนิสิต

Attribute	Data Type	Key	Reference Table	Description	Constraints	Example
id	bigint(20)	Primary Key	-	Object Identifier	Not Null	1
prefix	varchar(20)	-	-	คำนำหน้า	Not Null	นาย
fname	varchar(255)	-	-	ชื่ออาจารย์	Not Null	อภิลดา
lname	varchar(255)	-	-	นามสกุลอาจารย์	Not Null	ต้นทอง
faculty_id	bigint(20)	Foreign Key	education_faculty	คณะ	Not Null	วิทยาศาสตร์
department_id	bigint(20)	Foreign Key	education_department	ภาควิชา	Not Null	วิทยาการคอมพิวเตอร์
user_id_id	bigint(20)	Foreign Key	account_user	รหัสบัญชีผู้ใช้งาน	Not Null	D1234

ตารางที่ 3.16 teacher_instructorcourse

คำอธิบาย: ตารางเก็บข้อมูลหลักสูตรที่เป็นอาจารย์ผู้สอน

Attribute	Data Type	Key	Reference Table	Description	Constraints	Example
id	bigint(20)	Primary Key	-	Object Identifier	Not Null	1
teacher_id	bigint(20)	Foreign Key	teacher_teacher	รหัสอาจารย์	Not Null	1
course_id	bigint(20)	Foreign Key	education_course	รหัสวิชา	Not Null	1

ตารางที่ 3.17 education_course

คำอธิบาย: ตารางเก็บข้อมูลรายวิชา

Attribute	Data Type	Key	Reference Table	Description	Constraints	Example
id	bigint(20)	Primary Key	-	Object Identifier	Not Null	1
name	varchar(100)		-	ชื่อวิชา	Not Null	Computer Science Project
course_id	varchar(20)	-	-	รหัสวิชา	Not Null	01418499
section	varchar(10)	-	-	หมู่เรียน	Not Null	200
department_id	bigint(20)	-	-	ภาควิชา	Not Null	1
faculty_id	bigint(20)	-	-	คณะ	Not Null	1

ตารางที่ 3.18 education_faculty

คำอธิบาย: ตารางเก็บข้อมูลใบลา

Attribute	Data Type	Key	Reference Table	Description	Constraints	Example
id	bigint(20)	Primary Key	-	Object Identifier	Not Null	1
name	varchar(255)	-	-	ชื่อคณะ	Not Null	วิทยาศาสตร์

ตารางที่ 3.19 education_department
คำอธิบาย: ตารางเก็บข้อมูลภาควิชา

Attribute	Data Type	Key	Reference Table	Description	Constraints	Example
id	bigint(20)	Primary Key	-	Object Identifier	Not Null	1
name	varchar(255)	-	-	ชื่อภาควิชา	Not Null	วิทยาการคอมพิวเตอร์
faculty_id	bigint(20)	Foreign Key	education_faculty	คณะ	Not Null	1

ตารางที่ 3.20 leave_leaverequest
คำอธิบาย: ตารางเก็บข้อมูลใบลา

Attribute	Data Type	Key	Reference Table	Description	Constraints	Example
id	bigint(20)	Primary Key	-	Object Identifier	Not Null	1
start_date	date	-	-	วันที่เริ่มลา	Not Null	2024-03-21
end_date	date	-	-	วันที่สิ้นสุดการลา	Not Null	2024-03-21
leave_type	varchar(20)	-	-	ประเภทการลา	Not Null	ลาป่วย
Description	longtext	-	-	คำอธิบายการลา	-	Covid-19

ตารางที่ 3.21 leave_leaverequestdetail

คำอธิบาย: ตารางเก็บข้อมูลรายละเอียดใบลา

Attribute	Data Type	Key	Reference Table	Description	Constraints	Example
id	bigint(20)	Primary Key	-	Object Identifier	Not Null	1
course_id	bigint(20)	Foreign Key	education_course	วิชาที่ลา	Not Null	1
leave_request_id	bigint(20)	Foreign Key	leave_leaverequest	ใบลา	Not Null	1
student_id	bigint(20)	Foreign Key	student_student	นิสิที่ลา	Not Null	1
teacher_id	bigint(20)	Foreign Key	teacher_teacher	อาจารย์ประจำวิชาที่ลา	Not Null	1
status	varchar(20)	-	-	สถานะของใบลา	-	รอดำเนินการ/อนุมัติ/ไม่อนุมัติ
approve_id_by	bigint(20)	Foreign Key	teacher_teacher	อาจารย์ที่อนุมัติ/ไม่อนุมัติใบลา	-	1

ตารางที่ 3.22 leave_files

คำอธิบาย: ตารางเก็บไฟล์แนบหลักฐานการลา

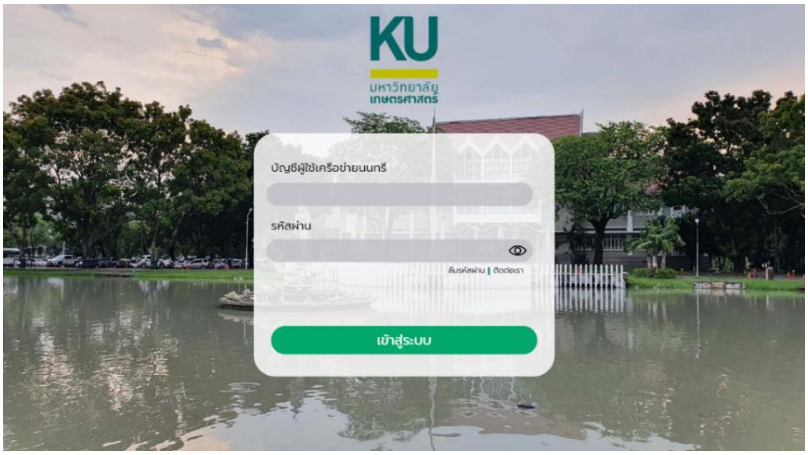
Attribute	Data Type	Key	Reference Table	Description	Constraints	Example
id	bigint(20)	Primary Key	-	Object Identifier	Not Null	1
pdf	varchar(100)	-	-	ชื่อไฟล์ของหลักฐานแนบการลา	-	store/pdfs/6310451260.pdf
leave_request_id	bigint(20)	Foreign Key	leave_leaverequest	ใบลา	Not Null	1

3.7 การออกแบบส่วนต่อประสานกับผู้ใช้

การออกแบบส่วนต่อประสานผู้ใช้งานเป็นการวางแผนเพื่อพัฒนาระบบเพื่อให้ผู้ใช้สามารถใช้งานได้อย่างมีประสิทธิภาพและสะดวกสบายโดยมีส่วนต่างๆ ดังต่อไปนี้

3.7.1 หน้าเข้าสู่ระบบ

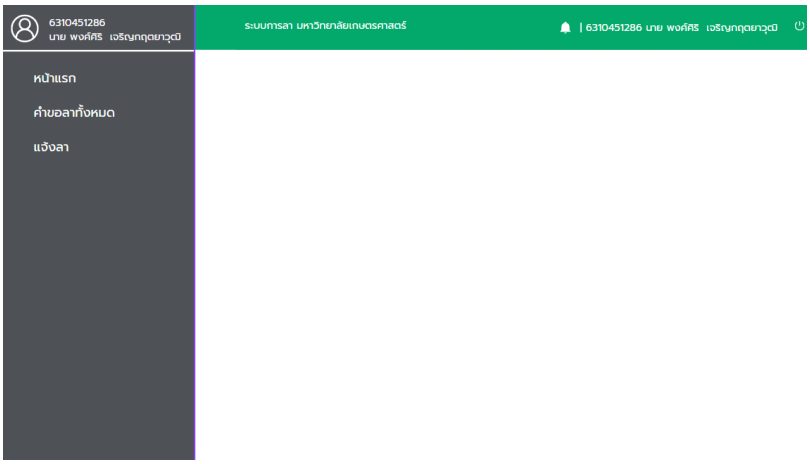
หน้าเข้าสู่ระบบเป็นหน้าให้นิสิตและอาจารย์เข้าสู่ระบบเพื่อใช้งานเว็บไซต์ในส่วนต่อไป โดยจะต้องกรอกบัญชีผู้ใช้เครือข่ายนนทรี และ รหัสผ่านที่ถูกต้อง จึงจะเข้าสู่ระบบได้



รูปที่ 3.15 แสดงการออกแบบส่วนต่อประสานกับผู้ใช้ในส่วนของหน้าเข้าสู่ระบบ

3.7.2 หน้าแรกของนิสิต

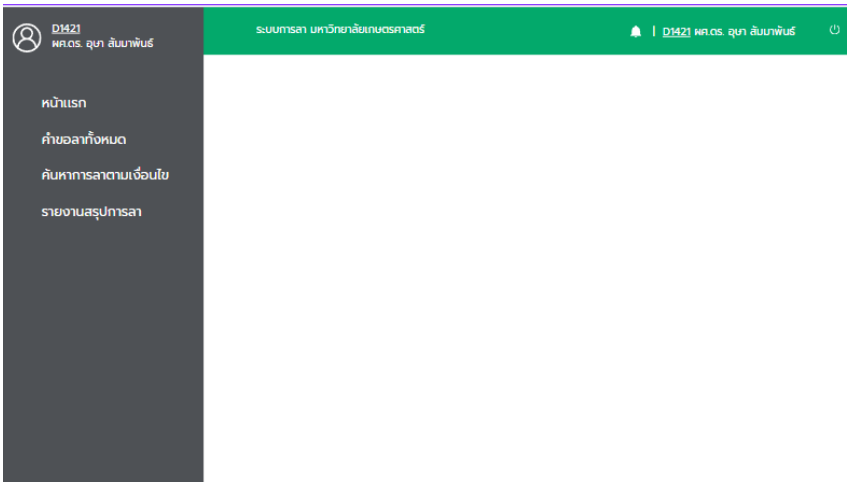
หน้าแรกของนิสิตหลังจากนิสิตเข้าสู่ระบบจะพบหน้านี้ โดยสามารถกดที่เมนูด้านข้างเพื่อใช้งานในส่วนต่างๆของนิสิตได้



รูปที่ 3.16 แสดงการออกแบบส่วนต่อประสานกับผู้ใช้ในส่วนของหน้าแรกของนิสิต

3.7.3 หน้าแรกของอาจารย์

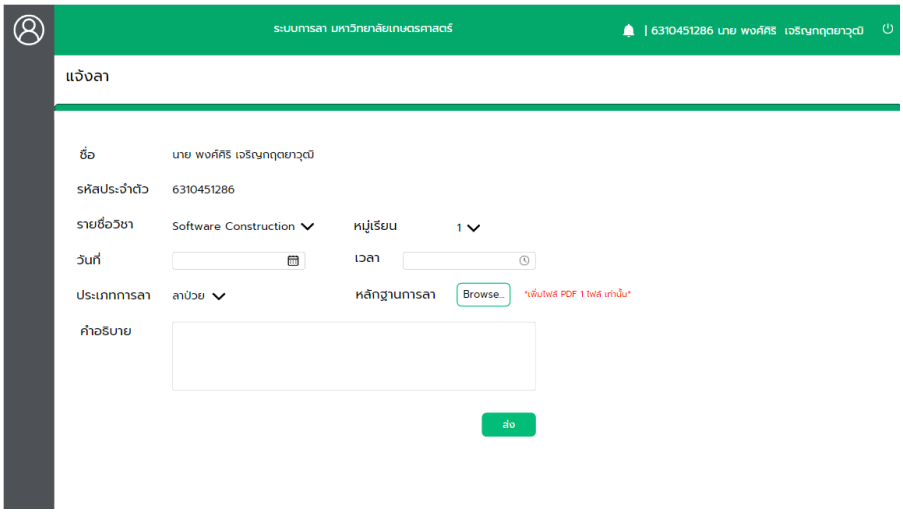
หน้าแรกของอาจารย์ หลังจากอาจารย์ที่เข้าสู่ระบบจะพบหน้านี้ โดยสามารถกดที่เมนูด้านข้างเพื่อใช้งานในส่วนต่างๆของอาจารย์



รูปที่ 3.17 แสดงการออกแบบส่วนต่อประสานกับผู้ใช้ในส่วนของหน้าแรกของอาจารย์

3.7.4 หน้าแจ้งลา

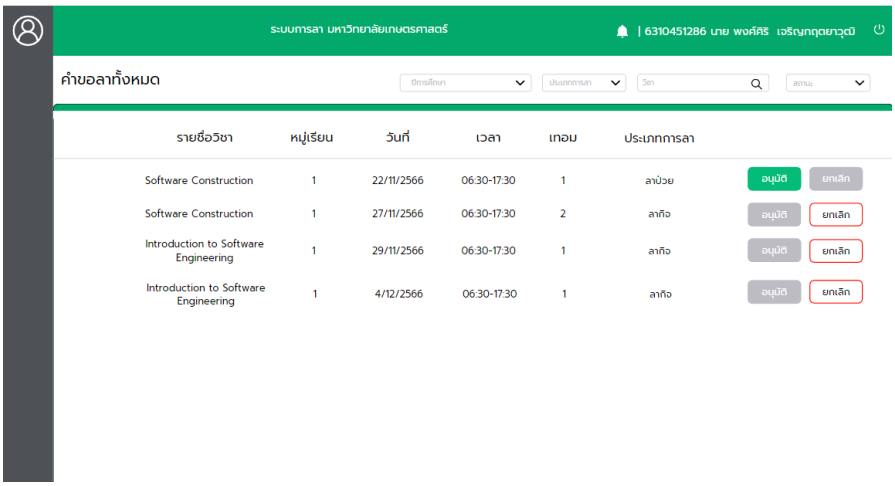
หน้าแจ้งลาเมื่อนิสิตกดเมนู “แจ้งลา” นิสิตจะสามารถแจ้งลาโดยการกรอกรายละเอียดและแนบหลักฐานการลา และส่งไปให้อาจารย์ประจำวิชาได้



รูปที่ 3.18 แสดงการออกแบบส่วนต่อประสานกับผู้ใช้ในส่วนของหน้าแจ้งลา

3.7.5 หน้าคำขอลาทั้งหมดของนิสิต

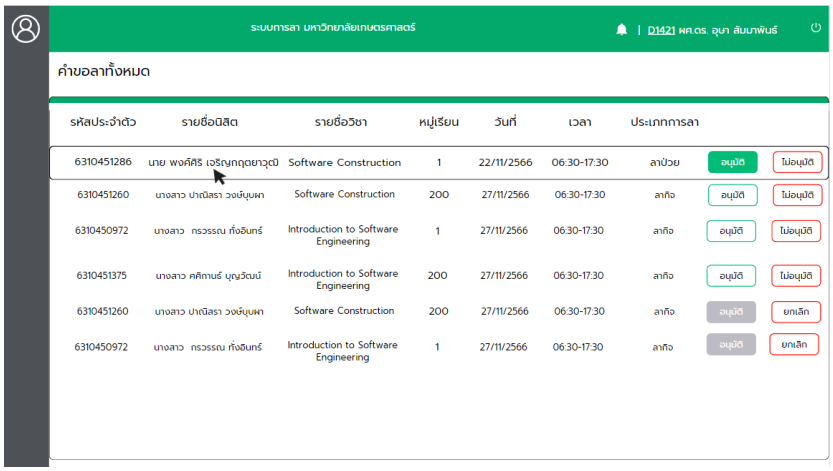
หน้าคำขอลาทั้งหมด เมื่อนิสิตกดเมนู “คำขอลาทั้งหมด” นิสิตจะสามารถดูคำขอลาทั้งหมดที่ส่งให้อาจารย์ประจำวิชาแต่ละวิชา และสามารถยกเลิกคำขอลาได้ ในหน้านี้



รูปที่ 3.19 หน้าคำขอลาทั้งหมด

3.7.6 หน้าคำขอลาทั้งหมดของอาจารย์

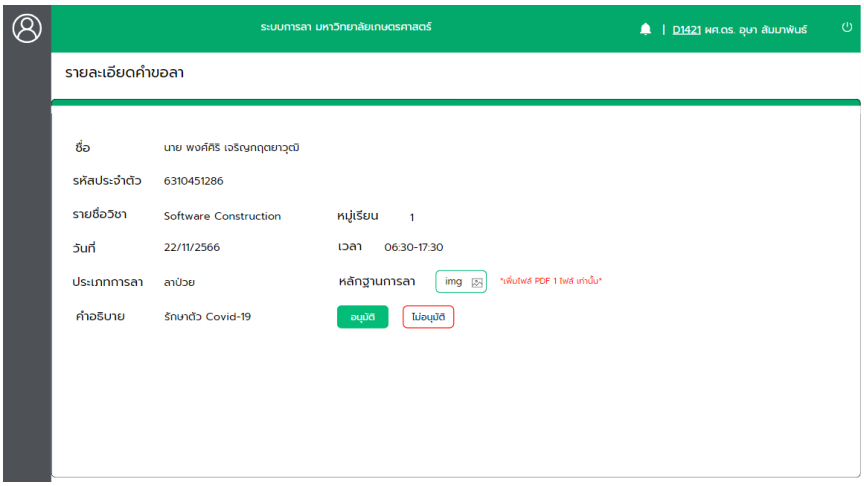
หน้าคำขอลาทั้งหมด เมื่ออาจารย์กดเมนู “คำขอลาทั้งหมด” จะแสดงคำขอลาทั้งหมดที่นิสิตส่งมาให้อาจารย์ประจำวิชา อาจารย์สามารถอนุมัติและไม่อนุมัติได้ นอกจากนี้ยังสามารถกดที่รายชื่อเพื่อดูรายละเอียดคำขอลาที่นิสิตส่งมาได้



รูปที่ 3.20 หน้าคำขอลาทั้งหมดของอาจารย์

3.7.7 หน้ารายละเอียดคำขอลา

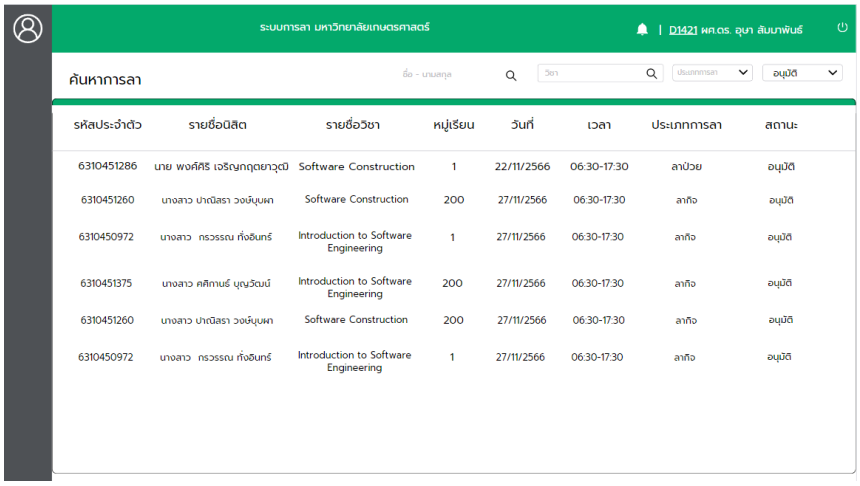
อาจารย์กดเมนู “คำขอลาทั้งหมด” จะแสดงคำขอลาทั้งหมดที่นิสิตส่งมาให้อาจารย์ หลังจากนั้นอาจารย์สามารถกดเพื่อดูรายละเอียดคำขอลาของนิสิต และสามารถอนุมัติหรือไม่อนุมัติได้จากหน้านี้ เช่นเดียวกันกับ หน้าคำขอลาทั้งหมดของอาจารย์ ใน 3.7.6



รูปที่ 3.21 หน้ารายละเอียดคำขอลา

3.7.8 หน้าค้นหาการลา

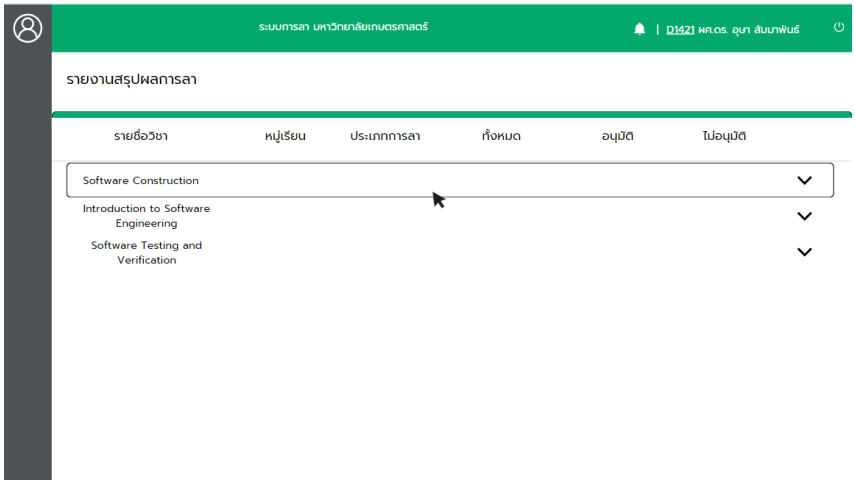
อาจารย์กดเมนู “ค้นหาการลาตามเงื่อนไข” อาจารย์จะสามารถค้นหาคำขอลาจากเงื่อนไขต่างๆได้ เช่น ชื่อ-นามสกุล วิชา ประเภทการลา และคำขอลาที่อนุมัติและไม่อนุมัติแล้ว



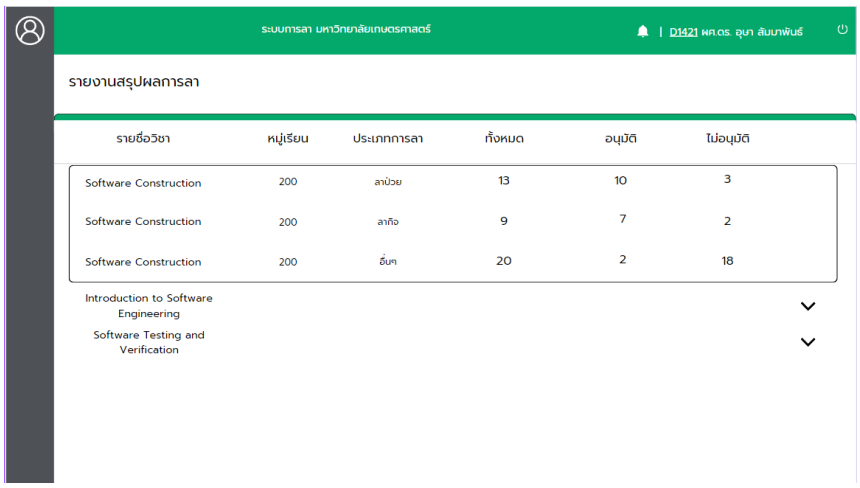
รูปที่ 3.22 หน้าค้นหาการลา

3.7.9 หน้ารายงานสรุปผลการลา

อาจารย์กดเมนู “รายงานสรุปการลา” อาจารย์จะสามารถดูรายงานผลสรุปการลาเทอมปัจจุบัน โดยการกดไปที่รายวิชาที่ต้องการดู หลังจากนั้นระบบจะแสดงรายละเอียดการลาของวิชานั้นๆ



รูปที่ 3.23 หน้ารายงานสรุปการลา



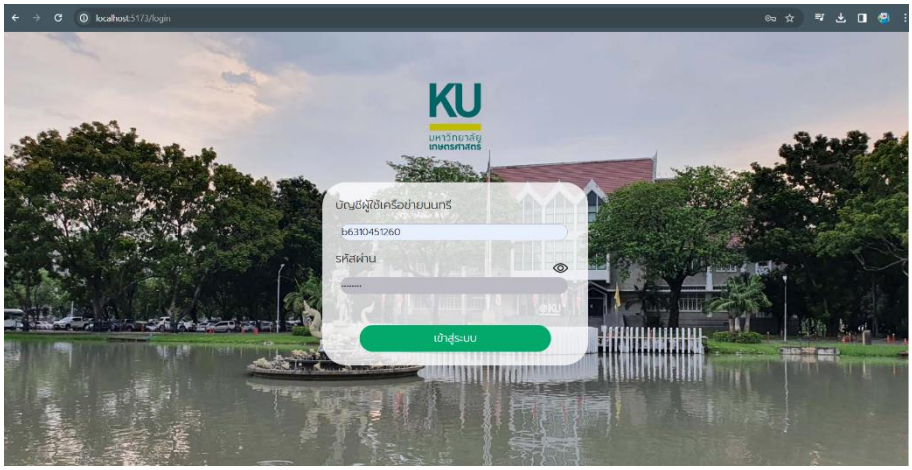
รูปที่ 3.24 หน้ารายงานสรุปผลการลา

บทที่ 4
การพัฒนาระบบ

ในการพัฒนาเว็บแอปพลิเคชัน ระบบการลาในมหาวิทยาลัย ทางคณะผู้จัดทำได้ใช้ภาษา Python โดยใช้ Django เฟรมเวิร์ค ในการพัฒนาระบบ Backend และใช้ภาษา HTML CSS JAVASCRIPT และ VUE.JS Framework ร่วมกับ Vuetify framework สร้าง User Interface (UI) ของระบบ โดยมี Component ต่าง ๆ เช่น ฟอร์มสำหรับการลา, ปุ่มสำหรับการอนุมัติหรือปฏิเสธ, และตารางแสดงข้อมูลการลา ในส่วนของการเชื่อมต่อกับฐานข้อมูลผู้จัดทำได้ใช้ Django ซึ่งเป็นเฟรมเวิร์กของภาษา Python จัดการฐานข้อมูลโดยใช้เทคนิค ORM (Object-Relational Mapping) ซึ่งเป็นเทคนิคการเขียนโปรแกรมที่ช่วยให้โปรแกรมสามารถทำงานกับฐานข้อมูลได้โดยไม่ต้องใช้ภาษา SQL โดยตรงสามารถใช้โมเดล Python แทนตารางของฐานข้อมูลและเปลี่ยนแปลงข้อมูลในฐานข้อมูลผ่านเมทอดของโมเดล นอกจากนี้ยังใช้ Axios ใน Vue.js เพื่อส่งคำขอ HTTP ระหว่าง Frontend และ Backend ของระบบเพื่อการสื่อสารข้อมูลของแอปพลิเคชัน โดยใช้ Postman เป็นเครื่องมือในการทดสอบ API endpoints และการส่งคำขอไปยัง Backend เพื่อปรับปรุงและทดสอบความถูกต้องของระบบก่อนนำไปใช้งาน

4.1 เข้าสู่ระบบ

การเข้าสู่ระบบใช้ในการตรวจสอบและยืนยันตัวตนของผู้ใช้ก่อนที่จะเข้าถึงและใช้งานฟังก์ชันต่าง ๆ ของระบบ



รูปที่ 4.1 โปรแกรมส่วนเข้าสู่ระบบ

4.1.1 login

หากเข้าสู่ระบบด้วย username และ password ของนิสิต ระบบจะเข้าไปที่หน้าแรกของนิสิต ในส่วนของอาจารย์ หากเข้าสู่ระบบด้วย username และ password ของอาจารย์ ระบบจะเข้าไปที่หน้าแรกของอาจารย์ และสามารถใช้งานฟังก์ชันอื่นๆได้

การพัฒนาระบบ Frontend

เมื่อรับข้อมูลการเข้าสู่ระบบจากผู้ใช้งานจากฟอร์มในหน้าเว็บ Axios จะส่งคำขอการลงชื่อเข้าใช้งานไปยังเซิร์ฟเวอร์ Backend เมื่อ Backend ตอบกลับมาเป็น JSON Web Token (JWT) ส่งกลับไปยัง Frontend เพื่อใช้ในการตรวจสอบสิทธิ์ในการเข้าถึงและใช้งานระบบต่อไป การใช้ JWT ในการยืนยันตัวตนช่วยเพิ่มความปลอดภัยและความสะดวกในการจัดการสิทธิ์ในระบบ

```
<template>
  <v-app>
    <router-view />
    <Toast />
  </v-app>
</template>
```

รูปที่ 4.2 ตัวอย่างโค้ดส่วนเทมเพลตของ App.vue

```
<script>
import { useUserStore } from '@/stores/user'
import axios from 'axios'
import TopNavBar from './components/TopNavBar.vue'
import Navbar from './components/navbar.vue'
export default {
  setup() {
    const userStore = useUserStore()
    return {
        userStore
    }
  },
  name: 'App',
  components: {
    Navbar,
    TopNavBar
  },
  beforeCreate() {
      this.userStore.initStore()

      const token = this.userStore.user.access

      if (token) {
          axios.defaults.headers.common["Authorization"] = "Bearer " + token;
      } else {
          axios.defaults.headers.common["Authorization"] = "";
      }
  },
  data() {
    return {}
  }
}
</script>
```

รูปที่ 4.3 ตัวอย่างโค้ดส่วนสคริปต์ของ App.vue

```
<style>
body {
  margin: 0;
  padding: 0;
  font-family: 'Prompt', sans-serif;
  height: 100vh;
}
.layout-container {
  display: flex;
}
.left-section {
  flex: 0 0 auto;
}
.right-section {
  flex: 1;
  display: flex;
  flex-direction: column;
}
.content {
  padding: 20px;
  flex: 1;
}
</style>
```

รูปที่ 4.4 ตัวอย่างโค้ดส่วนสไตล์ของ App.vue

```
<template>
  <v-container class="background1">
    <div class="img1">
      <img src="../assets/logo-1.png" width="200" height="250" />
    </div>
    <div class="rec">
      <form class="block1_login" v-on:submit.prevent="submitform">
        <br />
        <label >บัญชีผู้ใช้เครือข่ายนนทรี</label>
        <input
          type="text"
          v-model="formData.username"
          id="fname"
          name="firstname"
          placeholder=""
        />

        <label >รหัสผ่าน</label>
        <div class="flex items-center">
          <input
            :type="passwordVisible ? 'text' : 'password'"
            v-model="formData.password"
            id="passwordInput"
            name="lastname"
            placeholder=""
          />
          <button
            class="eye-icon"
            v-on:click.prevent="togglePasswordVisibility"
          >
            <img src="../assets/view.png" width="25" height="25" />
          </button>
        </div>
```

```
        <button class="logInButton">เข้าสู่ระบบ</button>
      </form>
      <!-- <div class="box1"></div>
      <div class="box2"></div> -->
    </div>
    <br />
    <template v-if="errors.length > 0">
      <div class="bg-red-300 text-white rounded-lg p-4 grid place-items-center">
        <p v-for="error in errors" v-bind:key="error">{{ error }}</p>
      </div>
    </template>
  </v-container>
</template>
```

รูปที่ 4.5 ตัวอย่างโค้ดส่วนเทมเพลตของ LogInView.vue

```
<script>
import { useToastStore } from '@/stores/toast'
import { useUserStore } from '@/stores/user'
import axios from 'axios'
export default {
  setup() {
    const userStore = useUserStore()
    const toastStore = useToastStore()
    return {
      userStore,
      toastStore
    }
  },
  data: () => ({
    formData: {
      username: '',
      password: ''
    },
    passwordVisible: false,
    errors: []
  }),
  methods: {
    async submitform() {
      this.errors = []

      if (this.formData.username === '') {
        this.errors.push('Your username is missing')
      }

      if (this.formData.password === '') {
        this.errors.push('Your password is missing')
      }

      if (this.errors.length === 0) {
        await axios
          .post('/api/login/', this.formData)
          .then((response) => {
            this.userStore.setToken(response.data)
            axios.defaults.headers.common['Authorization'] =
              'Bearer ' + response.data.access
          })
          .catch((error) => {
            this.errors.push(
              'The formData or password is incorrect! Or the user is not
activated!'
```

```
            )
          })
      }
      if (this.errors.length === 0) {
        await axios
          .get('/api/me/')
          .then((response) => {
            this.userStore.setUserInfo(response.data)
            this.$router.push('/home')
          })
          .catch((error) => {
            console.log('error', error)
          })
      }
    },
    togglePasswordVisibility() {
      this.passwordVisible = !this.passwordVisible
    }
  }
}
</script>
```

รูปที่ 4.6 ตัวอย่างโค้ดส่วนสคริปต์ของ LogInView.vue

```
<style>
.rec {
  width: 30%;
  height: 44% !important;
  background-color: rgba(255, 255, 255, 0.8);
  display: flex;
  flex-direction: column;
  margin: auto;
  margin-top: 0%;
  border-radius: 40px;
  z-index: 1;
}
.img1 {
  margin: 3% 44.6%;
  margin-bottom: 2%;
}
form.block1_login {
  margin: auto 5% !important;
}
input[type='text'],
input[type='password'] {
  width: 95% !important;
  padding: 1px !important;
  margin: 16px 2% !important;
  display: inline-block;
  border: 1px solid #bdbcc4;
  border-radius: 50px;
  box-sizing: border-box;
  background-color: #bdbcc4;
}
label {
  /* padding: 0 0 30% 0; top right bottom left */
  font-size: 16px !important;
  font-family: 'Prompt', sans-serif;
}
.logInButton {
  color: white;
  font-size: 18px !important;
```

```
  border: 1px solid #03a96b;
  background-color: #03a96b;
  width: 80%;
  padding: 2% 0%;
  border-radius: 25px;
  margin: 10%;
  margin-top: 9% !important;
}
button.eye-icon {
  position: absolute;
  margin: 1.3% -2% 0;
}
.background1 {
  background: url('../assets/SignUp-1.jpg');
  background-size: cover;
  height: 100vh;
  position: absolute;
}
.v-container {
  max-width: 1900px !important;
}
</style>
```

รูปที่ 4.7 ตัวอย่างโค้ดส่วนสไตล์ของ LogInView.vue

การพัฒนาระบบ Backend

กำหนด Model User

- AbstractBaseUser สร้าง custom user model โดยกำหนด field ต่างๆ เช่น username, email, role, และสิทธิ์ของผู้ใช้งาน
- PermissionsMixin เพื่อจัดการสิทธิ์และกลุ่มผู้ใช้งานในระบบ
- CustomUserManager สำหรับการจัดการผู้ใช้งาน
- กำหนด USERNAME_FIELD และ EMAIL_FIELD สำหรับการลงชื่อเข้าใช้งาน

โดยส่วนดังกล่าวเป็นส่วนของการสร้างโมเดล User และการจัดการสิทธิ์ของผู้ใช้งานในระบบ โดยเฉพาะส่วนที่เกี่ยวข้องกับการทำ authentication และ authorization ใน Django framework ซึ่งสามารถนำไปใช้ในการอธิบายเกี่ยวกับการ login และการจัดการสิทธิ์ในระบบได้

```
ROLES = [
    ("teacher", "Teacher"),
    ("student", "Student"),
    ("admin", "Admin"),
]

class CustomUserManager(UserManager):
    def _create_user(self, username, email, password, **extra_fields):
        if not username:
            raise ValueError("You have not provided a valid username")

        username = self.model.normalize_username(username)
        email = self.normalize_email(email)
        user = self.model(username=username, email=email, **extra_fields)
        user.set_password(password)
```

```
        user.save(using=self._db)

        return user

    def create_user(self, username=None, email=None, password=None,
**extra_fields):
        extra_fields.setdefault("is_staff", False)
        extra_fields.setdefault("is_superuser", False)
        return self._create_user(username, email, password, **extra_fields)



    def create_superuser(self, username=None, email=None, password=None,
**extra_fields):
        extra_fields.setdefault("is_staff", True)
        extra_fields.setdefault("is_superuser", True)
        return self._create_user(username, email, password, **extra_fields)
```

รูปที่ 4.8 ตัวอย่างโค้ดส่วนโมเดล Class CustomUserManager

```
class User(AbstractBaseUser, PermissionsMixin):
    username = models.CharField(max_length=12, unique=True,
default="default_user")
    user_id = models.CharField(max_length=10, blank=True, null=True)
    email = models.EmailField(unique=True)
    role = models.CharField(max_length=20, choices=ROLES, default="")
    is_activate = models.BooleanField(default=True)
    is_superuser = models.BooleanField(default=False)
    is_staff = models.BooleanField(default=False)

    date_joined = models.DateTimeField(
        default=timezone.now() + timezone.timedelta(hours=7)
    )
    last_login = models.DateTimeField(blank=True, null=True)

    objects = CustomUserManager()

    USERNAME_FIELD = "username"
    EMAIL_FIELDS = "email"
    REQUIRED_FIELDS = []
```

รูปที่ 4.9 ตัวอย่างโค้ดส่วนโมเดล User

ในการเข้าสู่ระบบ ผู้จัดทำได้ใช้ Django REST Framework ร่วมกับ Django JWT library เพื่อตรวจสอบข้อมูลที่ผู้ใช้ส่งมา หากข้อมูลถูกต้อง ระบบจะสร้าง JSON Web Token (JWT) และส่งกลับไปยัง Frontend เพื่อใช้ในการตรวจสอบสิทธิ์ในการเข้าถึงและใช้งานระบบต่อไป

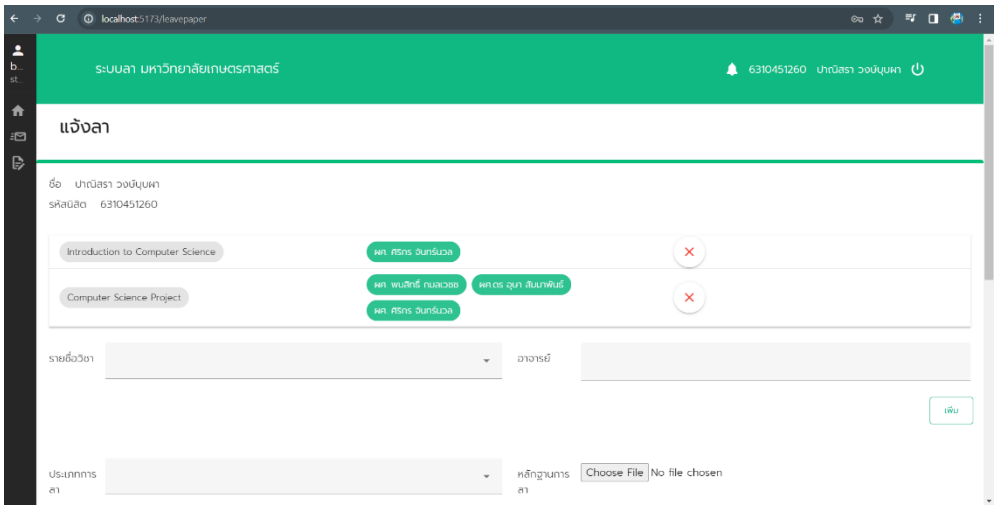
```
from django.urls import path, include
from rest_framework_simplejwt.views import TokenObtainPairView, TokenRefreshView

urlpatterns = [
path("login/", TokenObtainPairView.as_view(), name="token_obtain"),
path("refresh/", TokenRefreshView.as_view(), name="token_refresh"),
]
```

รูปที่ 4.10 ตัวอย่างโค้ดส่วน account/urls.py

4.2 แจ้งลา-ยกเลิกการแจ้งลา

4.2.1 แจ้งลา

นิสิตสามารถใช้ระบบย่อยนี้เพื่อส่งคำขอการลาในกรณีที่ต้องการขอลาหลายสาเหตุ เช่น ลาป่วย, ลากิจ, หรือลาอื่นๆ นิสิตต้องแนบเอกสารหรือหลักฐานเพิ่มเติมเพื่อรองรับคำขอลา



รูปที่ 4.11 โปรแกรมส่วนแจ้งลา

การพัฒนาระบบ Frontend

```
<template>
  <div class="layout-container">
    <Navbar />
    <div class="right-section">
      <TopNavBar />
      <div class="content-Page">
        <slot></slot>
        <div class="head">แจ้งลา</div>
      </div>
```

```
      <div class="content">
        <slot></slot>
        <div class="leaveblock1">
          <div class="content-head">ชื่อ</div>
          <div class="content-sub">{{ nameStudentLogin }}</div>
        </div>
        <div class="leaveblock2">
          <div class="content-head">รหัสนิสิต</div>
          <div class="content-sub">{{ userIdStudentLogin }}</div>
        </div>
        <!-- Leave Table -->
        <v-row v-if="serverItems && serverItems.length > 0">
          <v-col cols="12">
            <v-data-table
              :headers="headers"
              :items="serverItems"
              :items-per-page="itemsPerPage"
              class="elevation-1"
            >
              <template v-slot:item.coursesItem="{ item }">
                <v-chip-group v-if="item.coursesItem">
                  <v-chip small>
                    {{ item.coursesItem.name }}
                  </v-chip>
                </v-chip-group>
                <div v-else>No subjects to display</div>
              </template>

             <template v-slot:item.teachersItem="{ item }">
                <v-chip-group>
                  <v-chip
                    v-for="teacher in item.teachersItem"
                    :key="teacher.id"
                    small
                    :style="{
                      backgroundColor: item.coursesItem ? '#10B981' : '',
                      color: '#ffff'
                    }"
                  >
                    {{ teacher.prefix }} {{ teacher.fname }} {{ teacher.lname }}
                  </v-chip>
                </v-chip-group>
              </template>
              <template v-slot:item.actions="{ item }">
                <v-btn
                  icon
                  small
                  class="my-delete-btn"
                  @click="deleteItem(item)"
                  :style="{ fontSize: '10px', padding: '1px 2px' }"
                >
                  <v-icon small color="red">mdi-close</v-icon>
                </v-btn>
              </template>
            </v-data-table>
          </v-col>
        </v-row>
        <v-row>
          <v-col cols="12" md="6">
            <!-- รายชื่อวิชา -->
            <div class="leaveblock0">
              <div class="leaveblock1">
                <div class="Leave-content-head mt-3">รายชื่อวิชา</div>
                <v-select
                  v-model="selectedSubjects"
                  :items="subjects"
                  :item-props="subjectsItemProps"
                  style="width: 100%"
                ></v-select>
```

```
            </div>
          </div>
        </v-col>
        <v-col cols="12" md="6">
          <!-- อาจารย์ -->
          <div class="leaveblock0">
            <div class="leaveblock1">
              <div class="Leave-content-head mt-3">อาจารย์</div>
              <v-combobox
                v-model="selectedTeachers"
                :items="teachers"
                :item-props="teachersItemProps"
                multiple
                chips
                outlined
                @keydown.prevent
                @click.right.prevent
                @click.middle.prevent
                class="no-placeholder"
              ></v-combobox>
            </div>
          </div>
        </v-col>
      </v-row>
      <v-row justify="end">
        <v-col cols="12" md="8" lg="9" class="pa-md-2">
        </v-col>

        <v-col
          cols="12"
          md="4"
          lg="3"
          class="text-right pa-md-2"
          style="margin-top: -20px"
        >
          <button class="leave-addBtn mt--2" @click="addItem">เพิ่ม</button>
        </v-col>
      </v-row>

      <v-row style="margin-top: 45px">
        <!-- ประเภทการลา -->
        <v-col cols="12" md="6">
          <div class="leaveblock0">
            <div class="leaveblock1">
              <div class="Leave-content-head mt-3">ประเภทการลา</div>
              <v-select
                v-model="selectedLeaveType"
                :items="leaveTypes"
                style="width: 100%"
              ></v-select>
            </div>
          </div>
        </v-col>
        <!-- หลักฐานการลา -->
        <v-col cols="12" md="6">
          <div class="layout-side">
            <div class="Leave-content-head mt-3">หลักฐานการลา</div>
            <div class="input-file">
              <input
                id="input-file-button"
                type="file"
                @change="uploadedFile"
                ref="file"
              />
            </div>
          </div>
        </v-col>
      </v-row>
```

```
        <v-row>
          <v-col cols="12" md="6">
            <div class="content-head mr-3 mb-3.5">คำอธิบาย</div>
            <v-textarea
              class="custom-textarea"
              rows="4"
              placeholder="กรุณาป้อนคำอธิบายที่นี่"
              style="width: 100%"
              v-model="formDataLeaveRequest.description"
            ></v-textarea>
          </v-col>
          <v-col cols="12" md="6">
            <v-date-picker
              v-model="selectedDates"
              range
              multiple
              show-current
              locale="th"
              no-title
              style="margin: 0% 0% 0% 11%"
            ></v-date-picker>
          </v-col>
        </v-row>
        <v-row>
          <v-col cols="12">
            <template v-if="errors.length > 0">
              <div class="bg-red-300 text-white rounded-lg p-6" cols="12">

                <p v-for="error in errors" v-bind:key="error">{{ error }}</p>
              </div>
            </template>
          </v-col>
        </v-row>
        <v-row>
          <v-col cols="12" md="6">
            <button class="leave-addBtn" @click="submitForm">ส่ง</button>
          </v-col>
        </v-row>
      </div>
    </div>
  </div>
</template>
```

รูปที่ 4.12 ตัวอย่างโค้ดส่วนเทมเพลตของ LeavePaperView.vue

```
<script>
export default {
  setup() {
    const userStore = useUserStore()
    return {
      userStore
    }
  },
  data() {
    return {
      // Option
      leaveTypes: ['ลากิจ', 'ลาป่วย', 'อื่น ๆ'],

      // getStudentLogin
      testId: '',
```

```
      testStudentId: '',
      student: [],
      user: [],
      nameStudentLogin: '',
      userIdStudentLogin: '',
      // getSubject
      subjects: [],
      // getTechers
      teachers: [],

      // Input
      selectedTeachers: {},
      selectedSubjects: {},
      selectedDates: [],
      selectedLeaveType: '',
      files: [],

      // playload
      errors: [],
      formDataLeaveRequest: {
        approve_id_by: '',
        start_date: '',
        end_date: '',
        leave_type: '',
        description: '',
        status: 'pending'
      },
      leaveRequestId: '',
      formDataLeaveDetails: {
        leave_request_id: '',
        student_id: '',
        course_id: '',
        teacher_id: ''
      },
      serverItems: [],
      headers: [
        { text: 'Subject', value: 'coursesItem' },
        { text: 'Teacher', value: 'teachersItem' },
        { text: 'Actions', value: 'actions', sortable: false }
      ],
      // file
      files: [],
      upload_status: '',
      filename: ''
    }
  },
  components: {
    Navbar,
    TopNavBar

  },
  beforeCreate() {},
  async mounted() {
    await this.getStudentLogin()
    await this.getSubjects()
  },
  watch: {
    selectedSubjects: {
      handler() {
        this.selectedTeachers = []
        this.getTeachers()

      },
      immediate: true
    }
  },
  methods: {
    deleteItem(item) {
      const index = this.serverItems.indexOf(item)
```

```
      if (index > -1) {
        this.serverItems.splice(index, 1)
      }
    },
    addItem() {

      if (!this.selectedSubjects || !this.selectedSubjects.id) {
        console.log('No subject selected, item not added.')

        return
      }

      if (!this.selectedTeachers || this.selectedTeachers.length === 0) {
        console.log('No teacher selected, item not added.')

        return
      }

      const isSubjectExists = this.serverItems.some(
        (item) =>
          item.coursesItem && item.coursesItem.id === this.selectedSubjects.id
      )

      if (isSubjectExists) {
        console.log('This subject is already in the table.')

        return
      }
      const newItem = {
        coursesItem: this.selectedSubjects,
        teachersItem: this.selectedTeachers
      }
      this.serverItems.push(newItem)
      this.selectedSubjects = {}
      this.selectedTeachers = []
    },
    teachersItemProps(item) {
      const name = item.fname + ' ' + item.lname
      return {
        title: name,
        subtitle: item.id
      }
    },
    subjectsItemProps(item) {
      return {
        title: item.name,
        subtitle: item.course_id
      }
    },
    async getStudentLogin() {
      this.userStore.initStore()
      this.user = this.userStore.user
      this.testId = this.userStore.user.id
      if (this.testId) {
        const url = `/student/?user_id=${this.testId}`
        await axios
          .get(url)
          .then((response) => {
            this.student = response.data[0]
            this.testStudentId = this.student.id
            this.nameStudentLogin =
              this.student.fname + ' ' + this.student.lname
            this.userIdStudentLogin = this.userStore.user.user_id
          })
          .catch((error) => {
            console.log('error', error)
          })
      } else {
        this.student = []
```

```
      }
    },
    async getSubjects() {
      if (!this.testStudentId) {
        console.error('Missing student ID')
      }
      const url = `/studentRegister/?student_id=${this.student.id}`

      await axios
        .get(url)
        .then((response) => {
          this.subjects = response.data.map((item) => item.course_data)
        })
        .catch((error) => {
          console.error('Error fetching subjects data:', error)
        })
    },
    async getTeachers() {
      if (this.selectedSubjects && this.selectedSubjects.id) {
        const url = `/instructorCourse/?course_id=${this.selectedSubjects.id}`
        await axios
          .get(url)
          .then((response) => {
            this.teachers = response.data.map((item) => item.teacher_data)
          })
          .catch((error) => {
            console.log('error', error)
          })
      } else {
        this.teachers = []
    },
    async submitForm() {
      this.errors = []
      this.selectedDates = sortSelectedDates(this.selectedDates)
      if (!validateConsecutiveDates(this.selectedDates)) {
        this.errors.push('Please select consecutive dates only.')
        return
      }
      if (!validateSelectedDates(this.selectedDates)) {
        this.errors.push('Please select dates from today onwards.')
        return
      }
      const firstDate = formatDate(new Date(this.selectedDates[0]))
      const lastDate = formatDate(
        new Date(this.selectedDates[this.selectedDates.length - 1])
      )
      if (firstDate !== 'NaN-NaN-NaN') {
        this.formDataLeaveRequest.start_date = firstDate
        this.formDataLeaveRequest.end_date = lastDate
        this.formDataLeaveRequest.leave_type = this.selectedLeaveType
        const { value: confirmed } = await this.$swal.fire({
                title: 'บันทึกสำเร็จ',
                text: 'ข้อมูลของคุณถูกบันทึกเรียบร้อยแล้ว',
                icon: 'success',
                confirmButtonText: 'ตกลง',
                confirmButtonColor: '#02bc77',
              });
        await axios
          .post('/leaveRequest/', this.formDataLeaveRequest)
          .then((response) => {
            this.leaveRequestId = response.data.leaveRequest_id
            if (this.leaveRequestId) {
              console.log('leaveRequestId: ' + this.leaveRequestId)
              this.formDataLeaveDetails.leave_request_id = this.leaveRequestId
              this.formDataLeaveDetails.student_id = this.testStudentId
              this.addLeaveRequestDetail()
              this.saveFile()
              this.selectedLeaveType = ''
              this.files = ''
```

```
              this.formDataLeaveRequest.description = ''
              this.selectedDates = []
              this.serverItems = []
              console.log('Requests cancelled successfully:', response.data);
            }
          })
          .catch((error) => {
            console.log('error', error)
            this.$swal.fire('เกิดข้อผิดพลาด', 'เกิดข้อผิดพลาดในการบันทึกข้อมูล', 'error');
          })
      } else {
        this.errors.push('Please select dates')
      }
    },
    async addLeaveRequestDetail() {
      for (const item of this.serverItems) {
        if (item && item.coursesItem && item.teachersItem) {
          let courseId = item.coursesItem.course_id
          let couseOfId = item.coursesItem.id
          let teachersId = item.teachersItem.map((teacher) => teacher.id)

          for (const id of teachersId) {
            this.formDataLeaveDetails.teacher_id = id
            this.formDataLeaveDetails.course_id = couseOfId
            await axios
              .post('/leaveDetail/', this.formDataLeaveDetails)
              .then((response) => {
              })
              .catch((error) => {
                console.log('error', error)
              })
          }
        } else {
          console.log(
            'this.item or this.item.coursesItem or this.item.teachersItem is
undefined'
          )
        }
      }
    },    async saveFile() {
      let formData = new FormData()
      formData.append('pdf', this.filename)
      formData.append('leave_request_id', this.leaveRequestId)

      let axiosConfig = {
        headers: {
          'Content-Type': 'multipart/form-data'
        }
      }
      await axios
        .post('/files/', formData, axiosConfig)
        .then((response) => {
          this.upload_status = 'File Upload Success'
        })
        .catch((error) => {
          console.log(error)
        })
    },
    uploadedFile() {
      this.filename = ''
      this.filename = this.$refs.file.files[0]
    }
  }
}
</script>
```

รูปที่ 4.13 ตัวอย่างโค้ดส่วนสคริปต์ของ LeavePaperView.vue

```
style>
.v-combobox input {
  background-color: transparent !important;
  border: none !important;
}
.v-combobox:not(.v-input--is-disabled) .v-input__control {
  background-color: transparent !important;
  border: none !important;
}
#input-14,
#input-63 {
  display: none;
}
.custom-input {
  background-color: rgba(117, 117, 117, 0.3);
  border: none;
  height: 100%;
  width: 100%;
  border-bottom: 1px solid #908e8e;
  color: black;
}

.custom-input:hover {
  border-bottom: 1px solid #000;
}

.custom-input:focus {
  outline: none;
  border-bottom: 2px solid #000;
}
.custom-input::placeholder {
  color: #000;
  opacity: 1;
}
.content {
  border-top: solid 6px #02bc77;
  border-radius: 5px;
}
.head {
  font-size: 28px;
  font-weight: 600;
  margin: 0% 2.5%;
  padding: 1% 0% 2% 0%;
}
.leaveblock1 {
  display: flex;
  padding: 0% 0% 0.5% 0%;
}
.leaveblock2 {
  display: flex;
  padding: 0% 0% 2.5% 0%;
}
.content-sub {
  margin: 0% 0% 0% 1.5%;
}
.Leave-content-head.mt-3 {
  margin: 0% 2% 0% 0%;
  width: 12%;
}
#input-8,
#input-21 {
  margin: 0% 0% !important;
}
#input-63 {
  display: none;
}
.leave-addBtn {
```

```
  --color: #00a97f;
  padding: 0.8em 1.7em;
  background-color: transparent;
  border-radius: 0.3em;
  position: relative;
  overflow: hidden;
  cursor: pointer;
  transition: 0.5s;
  font-weight: 400;
  font-size: 13px;
  border: 1px solid;
  font-family: inherit;
  text-transform: uppercase;
  color: var(--color);
  z-index: 1;
}

.leave-addBtn::before,
.leave-addBtn::after {
  content: '';
  display: block;
  width: 50px;
  height: 50px;
  transform: translate(-50%, -50%);
  position: absolute;
  border-radius: 50%;
  z-index: -1;
  background-color: var(--color);
  transition: 1s ease;
  background-color: transparent;
}

.leave-addBtn::before {
  top: -1em;
  left: -1em;
}

.leave-addBtn::after {
  left: calc(100% + 1em);
  top: calc(100% + 1em);
}

.leave-addBtn:hover::before,
.leave-addBtn:hover::after {
  height: 410px;
  width: 410px;
  background-color: var(--color);
}

.leave-addBtn:hover {
  color: rgb(10, 25, 30);
}

.leave-addBtn:active {
  filter: brightness(0.8);
}
.my-delete-btn .v-btn__content {
  font-size: 16px;
}
.v-data-table-footer {
  display: none !important;
}
th.v-data-table__td.v-data-table-column--align-start.v-data-table__th {
  display: none !important;
}
button.v-btn.v-btn--elevated.v-btn--icon.v-theme--light.v-btn--density-default.v-
btn--size-default.v-btn--variant-elevated.my-delete-btn {
  display: flex;
  justify-content: center;
```

```
  align-items: center;
}
.v-input__details {
  padding-top: 0px;
}
#input-8-messages {
  display: none;
}
.v-col-md-8 v-col-lg-9 v-col-12 pa-md-2 {
  padding: 0%;
}
.layout-side {
  display: flex;
}
.input-file {
  margin-top: 1%;
}
#input-file-button {
  color: #000;
  cursor: pointer;
  position: relative;
  border: none;
  background: none;
  transition-timing-function: cubic-bezier(0.25, 0.8, 0.25, 1);
  transition-duration: 400ms;
  transition-property: color;
}

#input-file-button:focus,
#input-file-button:hover {
  color: #10b981;
}

#input-file-button:focus:after,
#input-file-button:hover:after {
  width: 100%;
  left: 0%;
}

#input-file-button:after {
  content: '';
  pointer-events: none;
  bottom: -2px;
  left: 50%;
  position: absolute;
  width: 0%;
  height: 2px;
  background-color: #fff;
  transition-timing-function: cubic-bezier(0.25, 0.8, 0.25, 1);
  transition-duration: 400ms;
  transition-property: width, left;
}
style>
```

รูปที่ 4.14 ตัวอย่างโค้ดส่วนสไตล์ของ LeavePaperView.vue

การพัฒนาระบบ Backend

```
STATUS = [
    ("approve", "Approved"),
    ("reject", "Rejected"),
    ("pending", "Pending"),
]

TYPE = [
    ("ลากิจ", "Personal Leave"),
    ("ลาป่วย", "Sick Leave"),
    ("อื่น ๆ", "Other Leave"),
]
class LeaveRequest(models.Model):
    start_date = models.DateField(null=True, blank=True)
    end_date = models.DateField(null=True, blank=True)
    leave_type = models.CharField(max_length=20, choices=TYPE,
default="undefined")
    description = models.TextField()
    created_at = models.DateField(auto_now_add=True)

class LeaveRequestDetail(models.Model):
    leave_request_id = models.ForeignKey(LeaveRequest, null=True,
on_delete=models.CASCADE)
    student_id = models.ForeignKey(Student,on_delete=models.CASCADE)
    course_id = models.ForeignKey(Course, null=True, blank=True,
on_delete=models.CASCADE)
    teacher_id = models.ForeignKey(
        Teacher, null=True,
        blank=True,
        on_delete=models.CASCADE,
        related_name="teacher_id",)
    status = models.CharField(max_length=20, choices=STATUS, default="pending")
    approve_id_by = models.ForeignKey(
        Teacher, null=True, blank=True,
        related_name="approve_id_by",
        on_delete=models.CASCADE
    )
class Files(models.Model):
    leave_request_id = models.ForeignKey(LeaveRequest, null=True, blank=True,
on_delete=models.CASCADE)
    pdf = models.FileField(upload_to="store/pdfs/")

    def __str__(self):
        return self.pdf
```

รูปที่ 4.15 ตัวอย่างโค้ดส่วน leave/models.py

```
class LeaveRequestSerializer(serializers.ModelSerializer):
    file_id = serializers.FileField(required=False)

    class Meta:
        model = LeaveRequest
        fields = "__all__"
        dept = 2

    def validate_file_id(self, attrs):
        if not attrs.content_type == "application/pdf":
```

```
            raise serializers.ValidationError("Only PDF files are allowed.")
        return attrs

    def save(self):
        data = self._validated_data
        leaveRequest = self.Meta.model(**data)
        leaveRequest.save()
        print("Saved LeaveRequest")
        print(leaveRequest.id)

        return leaveRequest
```

รูปที่ 4.16 ตัวอย่างโค้ดส่วน LeaveRequestSerializer

```
class LeaveRequestDetailSerializer(serializers.ModelSerializer):
    leave_request_id = serializers.PrimaryKeyRelatedField(
        queryset=LeaveRequest.objects.all(), required=False
    )

    student_id = serializers.PrimaryKeyRelatedField(
        queryset=Student.objects.all(), required=False
    )
    course_id = serializers.PrimaryKeyRelatedField(
        queryset=Course.objects.all(), required=False
    )
    teacher_id = serializers.PrimaryKeyRelatedField(
        queryset=Teacher.objects.all(), required=False
    )

    leave_request_data = LeaveRequestSerializer(
        source="leave_request_id", read_only=False, required=False
    )
    approve_id_by = serializers.PrimaryKeyRelatedField(
        queryset=Teacher.objects.all(),
        required=False,
        allow_null=True
    )
    teacher_data = TeacherSerializer(source="teacher_id", read_only=True)
    student_data = StudentSerializer(source="student_id", read_only=True)
    course_data = CourseSerializer(source="course_id", read_only=True)
    approve_id_by_data = TeacherSerializer(source="approve_id_by",read_only=True)
    class Meta:
        model = LeaveRequestDetail
        fields = "__all__"
        dept = 2
```

รูปที่ 4.17 ตัวอย่างโค้ดส่วน LeaveRequestDetailSerializer

```
class FileSerializer(serializers.ModelSerializer):
    leave_request_id = serializers.PrimaryKeyRelatedField(
        queryset=LeaveRequest.objects.all(), required=False
    )
    class Meta:
        model = Files
        fields = "__all__"
        dept = 2
```

รูปที่ 4.18 ตัวอย่างโค้ดส่วน FileSerializer

```
class LeaveRequestViewSet(viewsets.ModelViewSet):
    queryset = LeaveRequest.objects.all()
    serializer_class = LeaveRequestSerializer

    def get_queryset(self):
        queryset = super().get_queryset()
        id = self.request.query_params.get('id')

        start_date = self.request.query_params.get('start_date')
        end_date = self.request.query_params.get('end_date')
        leave_type = self.request.query_params.get('leave_type')
        status = self.request.query_params.get('status')

        if start_date:
            queryset = queryset.filter(start_date__icontains=start_date)

        if end_date:
            queryset = queryset.filter(end_date__icontains=end_date)

        if leave_type:
            queryset = queryset.filter(leave_type__icontains=leave_type)

        if status:
            queryset = queryset.filter(status__icontains=status)

        if id:
            queryset = queryset.filter(id=id)

        return queryset

    def create(self, request, *args, **kwargs):
        serializer = self.get_serializer(data=request.data)
        if serializer.is_valid():
            leaveRequest = serializer.save()
            lid = leaveRequest.id

            leaveRequest_url = f'http://localhost:8000/leaveRequest/{lid}/'
            leaveRequest_id = lid

            response_data = serializer.data
            response_data['id-url'] = leaveRequest_url
            response_data['leaveRequest_id'] = leaveRequest_id
```

```
            return Response(response_data, status=status.HTTP_201_CREATED)
        else:
            return Response(serializer.errors,
status=status.HTTP_400_BAD_REQUEST)
```

รูปที่ 4.19 ตัวอย่างโค้ดส่วน LeaveRequestViewSet

```
class LeaveRequestDetailViewSet(viewsets.ModelViewSet):
    queryset = LeaveRequestDetail.objects.all()

    serializer_class = LeaveRequestDetailSerializer


    def get_queryset(self):
        queryset = super().get_queryset()
        leave_request_id = self.request.query_params.get('leave_request_id')
        student_id = self.request.query_params.get('student_id')
        teacher_id = self.request.query_params.get('teacher_id')
        course_id = self.request.query_params.get('course_id')
        status = self.request.query_params.get('status')


        if leave_request_id:
            queryset = queryset.filter(leave_request_id_by=leave_request_id)

        if teacher_id:
            queryset = queryset.filter(teacher_id=teacher_id)
        if course_id:
            queryset = queryset.filter(course_id=course_id)
        if student_id:
            queryset = queryset.filter(student_id=student_id)
        if status:
            queryset = queryset.filter(status__icontains=status)

        return queryset
```

รูปที่ 4.20 ตัวอย่างโค้ดส่วน LeaveRequestDetailViewSet

```
class FilesViewSet(viewsets.ModelViewSet):
    queryset = Files.objects.all()
    serializer_class = FileSerializer
    def get_queryset(self):
        queryset = super().get_queryset()
        leave_request_id = self.request.query_params.get('leave_request_id')

        if leave_request_id:
            queryset = queryset.filter(leave_request_id=leave_request_id)


        return queryset
```

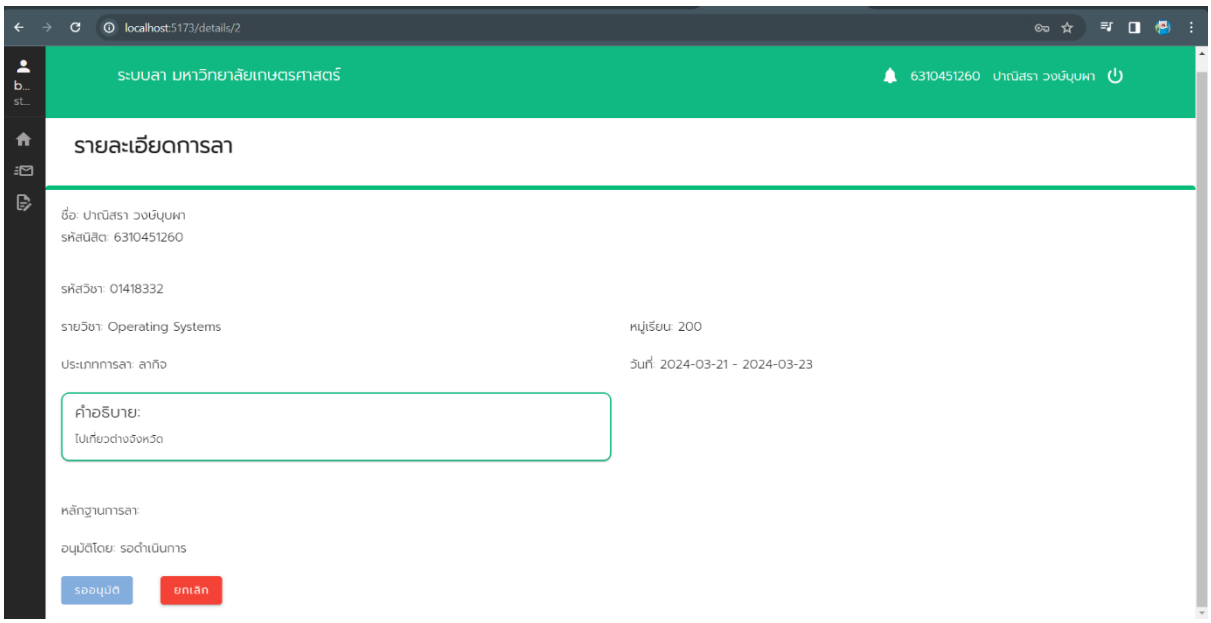
รูปที่ 4.21 ตัวอย่างโค้ดส่วน FilesViewSet

```
leave/urls.py
router = DefaultRouter()
router.register("files", FilesViewSet, basename="files")
router.register("leaveRequest", LeaveRequestViewSet)

router.register("leaveDetail", LeaveRequestDetailViewSet)

urlpatterns = [
    path("", include(router.urls)),
    path('leaveDetail/delete_multiple/',
LeaveRequestDetailViewSet.as_view({'delete': 'delete_multiple'})),
    path('leaveDetail/update_multiple/',
LeaveRequestDetailViewSet.as_view({'put': 'update_multiple'})),
    path('leave_request_details/', LeaveRequestDetailView.as_view(), name='leave-
request-detail'),
]
```

รูปที่ 4.22 ตัวอย่างโค้ดส่วน leave/urls.py

4.2.2 ยกเลิกการแจ้งลา

นิสิตสามารถใช้ฟังก์ชันนี้เพื่อยกเลิกหรือเปลี่ยนแปลงคำขอการลาที่ส่งไปแล้ว จะสามารถยกเลิกได้ในกรณีที่อาจารย์ประจำรายวิชานั้นยังไม่ได้กด อนุมัติ หรือ ไม่อนุมัติ เท่านั้น



รูปที่ 4.23 โปรแกรมส่วนแสดงรายละเอียดการลาของนิสิต

การพัฒนาระบบ Frontend

```
<template>
  <div class="layout-container">
    <Navbar />

    <div class="right-section">
      <TopNavBar />
      <div class="content-Page">
        <slot></slot>

        <div class="head">รายละเอียดการลา</div>
      </div>
      <div class="content">
       // other code
        <v-row>
          <v-col cols="12" md="6">
            <v-btn
              small
              :color="
                leaveRequest.status === 'approve'
                  ? 'green'
                  : leaveRequest.status === 'reject'
                    ? 'grey'
                    : 'primary'
              "
              @click.stop="
                leaveRequest.status === 'pending'
                  ? onPending(leaveRequest)
                  : leaveRequest.status === 'approve'
                    ? onApproved(leaveRequest)
                    : approveRequest(leaveRequest)
              "
              :disabled="
                leaveRequest.status === 'approve' ||
                leaveRequest.status === 'reject' ||
                leaveRequest.status === 'pending'
              "
            >
              {{
                leaveRequest.status === 'approve'
                  ? 'Approved'
                  : leaveRequest.status === 'reject'
                    ? 'Rejected'
                    : leaveRequest.status === 'pending'
                      ? 'รออนุมัติ'
                      : 'Approve'
              }}
            </v-btn>
            <v-btn
              small
              color="red"
              @click.stop="cancelRequest(leaveRequest)"
              style="margin-left: 5%"
              v-if="leaveRequest.status === 'pending'"
            >
              ยกเลิก
            </v-btn>
          </v-col>
        </v-row>
    </div>
  </div>
</template>
```

รูปที่ 4.24 ตัวอย่างโค้ดส่วนเทมเพลตของ DetailPage.view

```
<script>
   // other code

  methods: {
    // other code

    async cancelRequest(request) {
      const { value: confirmed } = await this.$swal.fire({
        title: 'ยืนยันการลบ',
        text: 'คุณต้องการลบรายการนี้หรือไม่?',
        icon: 'warning',
        showCancelButton: true,
        confirmButtonColor: '#02bc77',
        cancelButtonColor: '#d33',
        confirmButtonText: 'ใช่, ลบ!',
        cancelButtonText: 'ยกเลิก'
      })

      if (confirmed) {
        const leave_request_id = request.leave_request_data.id
        const course_id = request.course_data.id
        const url = `/leaveDetail/delete_multiple/`
        const params = {
          course_id: course_id,
          leave_request_id: leave_request_id
        }

        try {
          const response = await axios.delete(url, { params: params })
          console.log('Requests cancelled successfully:', response.data)
          this.$swal.fire({
            title: 'ลบสำเร็จ',
            text: 'ลบรายการเรียบร้อยแล้ว',
            icon: 'success',
            customClass: {
              confirmButton: 'btn btn-success'
            },
            confirmButtonColor: '#02bc77'
          })
          this.$router.push('/deleted-page')
        } catch (error) {
          console.error(
            'Error cancelling requests:',
            error.response || error.message
          )
          this.$swal.fire(
            'เกิดข้อผิดพลาด',
            'เกิดข้อผิดพลาดในการลบรายการ',
            'error'
          )
        }
      }
  }
</script>
```

รูปที่ 4.25 ตัวอย่างโค้ดส่วนสคริปต์ของ DetailPage.view

การพัฒนาระบบ Backend

```
def destroy(self, request, *args, **kwargs):
      try:
          course_id = request.query_params.get('course_id')
          leave_request_id = request.query_params.get('leave_request_id')

          if course_id and leave_request_id:

              leave_request_details = LeaveRequestDetail.objects.filter(
                  course_id=course_id,
                  leave_request_id=leave_request_id
              )

              if not leave_request_details.exists():
                  return Response(status=status.HTTP_404_NOT_FOUND)

              leave_request_details.delete()
          else:

              leave_request_detail = self.get_object()
              leave_request_detail.delete()

          return Response(status=status.HTTP_204_NO_CONTENT)

      except LeaveRequestDetail.DoesNotExist:
          return Response(status=status.HTTP_404_NOT_FOUND)
      except Exception as e:

          return Response(str(e), status=status.HTTP_500_INTERNAL_SERVER_ERROR)



  @action(detail=False, methods=['delete'])
  def delete_multiple(self, request):
      course_id = request.query_params.get('course_id')
      leave_request_id = request.query_params.get('leave_request_id')

      if course_id and leave_request_id:
          LeaveRequestDetail.objects.filter(
              course_id=course_id,
              leave_request_id=leave_request_id
          ).delete()

          return Response(status=status.HTTP_204_NO_CONTENT)
      return Response(status=status.HTTP_400_BAD_REQUEST)
```

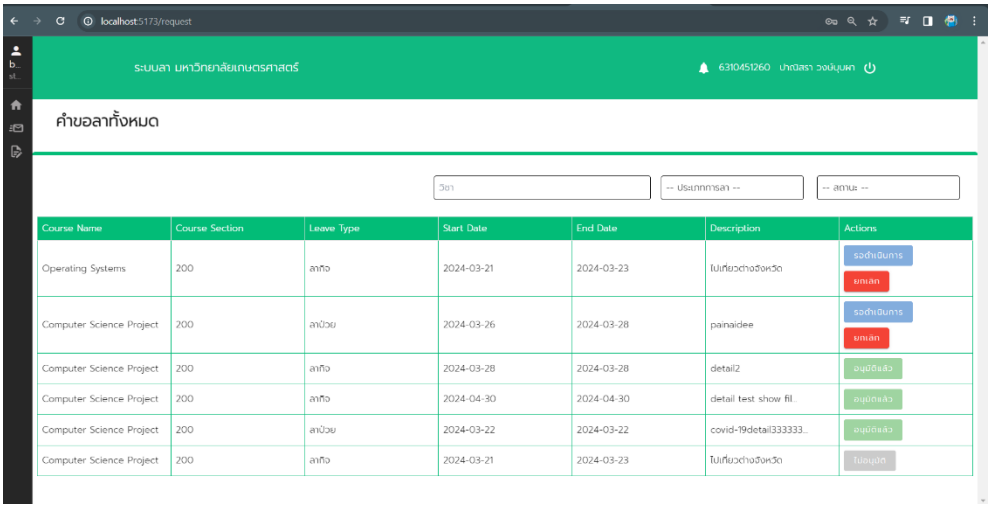
รูปที่ 4.26 ตัวอย่างโค้ดส่วน LeaveDetailViewSet

4.3 แสดงคำขอลาทั้งหมด

การทำงานของระบบย่อยนี้แสดงคำขอลาทั้งหมดเพื่อจัดการคำขอลาของอาจารย์มีความสะดวกและมีประสิทธิภาพ

4.3.1 แสดงคำขอลาทั้งหมดในหน้าของนิสิต

ระบบย่อยนี้สามารถดูคำขอลาและรายละเอียดทั้งหมดที่นิสิตส่งไปให้อาจารย์และยังสามารถยกเลิกการแจ้งลาภายในหน้านี้ได้



รูปที่ 4.27 โปรแกรมส่วนแสดงคำขอลาทั้งหมดของนิสิต

การพัฒนาระบบ Frontend

```
<template>
  <div class="layout-container">
    <Navbar />

    <div class="right-section">
      <TopNavBar />
      <div class="content-Page">
        <slot></slot>

        <div class="head">คำขอลาทั้งหมด</div>
      </div>
      <div class="content">
        <!-- Search fields in a single row -->
        <v-row>
          <v-col cols="12" md="2"> </v-col>
          <v-col cols="12" md="3"></v-col>
          <v-col cols="12" md="3" class="field-container">
            <div class="label-input-pair">
              <input type="text" placeholder="วิชา" v-model="search"/>
            </div>
          </v-col>

          <v-col cols="12" md="2" class="field-container">
            <div class="label-input-pair">

              <select v-model="selectedOption">
                <option value="" disabled selected>
```

```
                -- ประเภทการลา --
              </option>
              <option value="">None</option>
              <option value="ลากิจ">ลากิจ</option>
              <option value="ลาป่วย">ลาป่วย</option>
              <option value="อื่นๆ">อื่นๆ</option>

            </select>
          </div>
        </v-col>
        <v-col cols="12" md="2" class="field-container">
          <div class="label-input-pair">

              <select v-model="statusSearch">
                <option value="" disabled selected>
                  -- สถานะ --
                </option>
                <option value="">-</option>
                <option value="pending">รอดำเนินการ</option>
                <option value="approve">อนุมัติแล้ว</option>
                <option value="reject">ไม่อนุมัติ</option>

              </select>
          </div>
        </v-col>
      </v-row>
      <!-- table -->
      <v-row>
        <table class="leave-requests-table">
          <thead>
            <tr>
              <!-- <th>ID</th> -->
              <th>Course Name</th>
              <th>Course Section</th>
              <th>Leave Type</th>
              <th>Start Date</th>
              <th>End Date</th>
              <th>Description</th>
              <th>Actions</th>
            </tr>
          </thead>
          <tbody>
            <tr
              v-for="request in filteredRequests"
              :key="request.id"
              @click="navigateToDetail(request.id)"
              class="request-row"
            >
              <!-- <td>{{ request.id }}</td> -->
              <td>{{ request.course_data.name }}</td>
              <td>{{ request.course_data.section }}</td>
              <td>{{ request.leave_request_data.leave_type }}</td>
              <td>{{ request.leave_request_data.start_date }}</td>
              <td>{{ request.leave_request_data.end_date }}</td>
              <td>{{
getRequestDescription(request.leave_request_data.description) }}</td>
              <td class="actions-cell">
                <v-btn
                  small
                  :color="
                    request.status === 'approve'
                      ? 'green'
                      : request.status === 'reject'
                        ? 'grey'
                        : 'primary'
                  "
                  @click.stop="
                    request.status === 'pending'
```

```
                        ? onPending(request)
                        : request.status === 'approve'
                          ? onApproved(request)
                          : approveRequest(request)
                    "
                    :disabled="
                      request.status === 'approve' ||
                      request.status === 'reject' ||
                      request.status === 'pending'
                    "
                  >
                    {{
                      request.status === 'approve'
                        ? 'อนุมัติแล้ว'
                        : request.status === 'reject'
                          ? 'ไม่อนุมัติ'
                          : request.status === 'pending'
                            ? 'รอดำเนินการ'
                            : 'Approve'
                    }}
                  </v-btn>
                  <div>
                    <v-btn
                      small
                      color="red"
                      @click.stop="cancelRequest(request)"
                      style="margin-top: 4%"
                      v-if="request.status === 'pending'"
                    >
                      ยกเลิก
                    </v-btn>
                  </div>
                </td>
              </tr>
            </tbody>
          </table>
        </v-row>
      </div>
    </div>
  </div>
</template>
```

รูปที่ 4.28 ตัวอย่างโค้ดส่วนเทมเพลตของ RequestView.vue

```
<script>
export default {
  setup() {
    const router = useRouter()
    const toastStore = useToastStore()
    const userStore = useUserStore()
    function navigateToDetail(id) {
      router.push({ name: 'DetailPage', params: { id } })
    }
    return {
      navigateToDetail,
      toastStore,
      userStore
    }
  },
  components: {
```

```
    Navbar,
    TopNavBar
  },
  mounted() {
    this.fetchLeaveRequests()
  },
  computed: {
  filteredRequests() {
      let filtered = this.leaveRequests.filter(request => {
        const matchesCourse = this.search
        ?
request.course_data.name.toLowerCase().includes(this.search.toLowerCase())
        : true;
        const matchesLeaveType = this.selectedOption ?
request.leave_request_data.leave_type === this.selectedOption : true;
        const matchesStatus = this.statusSearch ?
request.status.toLowerCase().includes(this.statusSearch.toLowerCase()) : true;
        return matchesCourse && matchesLeaveType && matchesStatus;
      });
      const statusOrder = {
        'pending': 1,
        'approve': 2,
        'reject': 3
      };
      filtered = filtered.sort((a, b) => {
        const orderA = statusOrder[a.status] || 999;
        const orderB = statusOrder[b.status] || 999;

        if (orderA === orderB) {
          return a.id - b.id;
        }
        return orderA - orderB;
      });
      return filtered;
    }
  },
  data() {
    return {
      leaveRequests: [],
      loading: false,
      headers: [
        { text: 'ID', align: 'start', value: 'id', sortable: true },
        { text: 'Course Name', value: 'course_data.name' },
        { text: 'Course Section', value: 'course_data.section' },
        { text: 'Leave Type', value: 'leave_request_data.leave_type' },
        { text: 'Start Date', value: 'leave_request_data.start_date' },
        { text: 'End Date', value: 'leave_request_data.end_date' },
        { text: 'Actions', value: 'actions' }
      ],
      // getStudentLogin
      testId: '',
      testStudentId: '',
      student: [],
      user: [],
      // for Search
      search: '',
      selectedOption: '',
      statusSearch: '',
    }
  },
  methods: {
    getRequestDescription(description) {
      if (description) {
        if (description.length > 20) {
          return description.slice(0, 20) + '...';
        } else {
          return description;
        }
      }
```

```
      return '';
    },
    async getStudentLogin() {
      this.userStore.initStore()
      this.user = this.userStore.user
      this.testId = this.userStore.user.id
      if (this.testId) {
        const url = `/student/?user_id=${this.testId}`
        await axios
          .get(url)
          .then((response) => {
            this.student = response.data[0]
            this.testStudentId = this.student.id
          })
          .catch((error) => {
            console.log('error', error)
          })
      } else {
        this.student = []
      }
    },
    getUniqueSubjects(items) {
      const unique = {}
      items.forEach((item) => {
        const uniqueKey = `${item.course_data.name}-${item.leave_request_id}`
        if (!unique[uniqueKey]) {
          unique[uniqueKey] = item
        }
      })
      return Object.values(unique)
    },
    async fetchLeaveRequests() {
      this.loading = true
      await this.getStudentLogin()
      console.log('test Student ID: ' + this.testStudentId)
      try {
        if (this.testStudentId) {
          const response = await axios.get(
            `/leaveDetail/?student_id=${this.testStudentId}`
          )
          this.leaveRequests = response.data
          this.leaveRequests = this.getUniqueSubjects(response.data)
        } else {
          console.log("Doesn't have Student ID:  " + this.testStudentId)
        }
      } catch (error) {
        console.error('There was an error fetching the leave requests:', error)
      } finally {
        this.loading = false
      }
    },
  }
}
</script>
```

รูปที่ 4.29 ตัวอย่างโค้ดส่วนสคริปต์ของ RequestView.vue

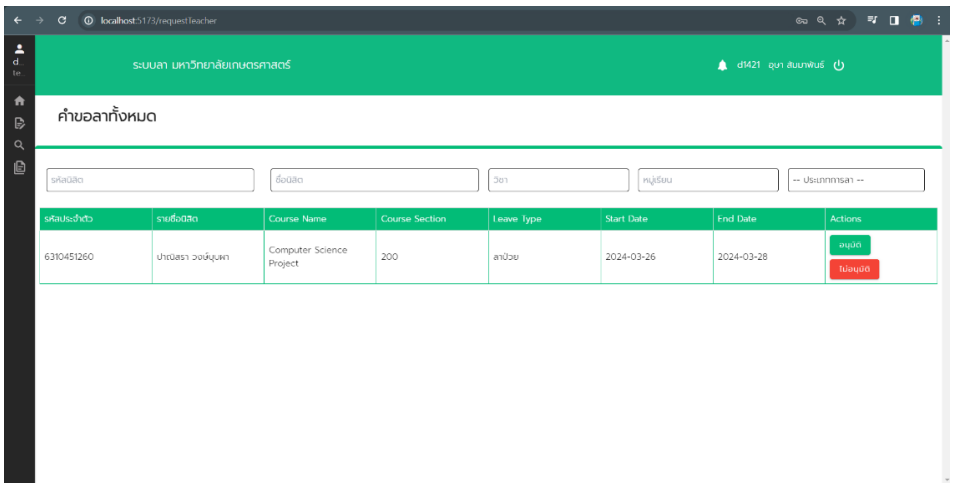
การพัฒนาระบบ Backend

```
class LeaveRequestDetailViewSet(viewsets.ModelViewSet):
    queryset = LeaveRequestDetail.objects.all()
    serializer_class = LeaveRequestDetailSerializer


    def get_queryset(self):
        queryset = super().get_queryset()
        leave_request_id = self.request.query_params.get('leave_request_id')
        student_id = self.request.query_params.get('student_id')
        teacher_id = self.request.query_params.get('teacher_id')
        course_id = self.request.query_params.get('course_id')
        status = self.request.query_params.get('status')


        if leave_request_id:
            queryset = queryset.filter(leave_request_id_by=leave_request_id)

        if teacher_id:
            queryset = queryset.filter(teacher_id=teacher_id)
        if course_id:
            queryset = queryset.filter(course_id=course_id)
        if student_id:
            queryset = queryset.filter(student_id=student_id)
        if status:
            queryset = queryset.filter(status__icontains=status)

        return queryset
```

รูปที่ 4.30 ตัวอย่างโค้ดส่วน LeaveRequestDetailViewSet

4.3.2 แสดงคำขอลาทั้งหมดในหน้าของอาจารย์

ระบบย่อยนี้สามารถดูคำขอลาและรายละเอียดทั้งหมดที่นิสิตส่งให้อาจารย์และยังสามารถอนุมัติหรือไม่อนุมัติภายในหน้านี้ได้



รูปที่ 4.31 โปรแกรมส่วนแสดงคำขอลาทั้งหมดของนิสิตที่ส่งมาให้อาจารย์

การพัฒนาระบบ Frontend

```
<template>
  <div class="layout-container">
    <Navbar />

    <div class="right-section">
      <TopNavBar />
      <div class="content-Page">
        <slot></slot>

        <div class="head">คำขอลาทั้งหมด</div>
      </div>
      <div class="content">
        <v-row>
          <v-col cols="12" md="3">
            <div class="field-container">
              <div class="label-input-pair">
                <input type="text" placeholder="รหัสนิสิต" v-model="idStudentSearch"
/>
              </div>
            </div>
          </v-col>
          <v-col cols="12" md="3">
            <div class="field-container">
              <div class="label-input-pair">
                <input type="text" placeholder="ชื่อนิสิต" v-model="nameSearch" />
              </div>
            </div>
          </v-col>
          <v-col cols="12" md="2">
            <div class="field-container">
              <div class="label-input-pair">
                <input type="text" placeholder="วิชา" v-model="search" />
              </div>
            </div>
          </v-col>
          <v-col cols="12" md="2">
            <div class="field-container">
              <div class="label-input-pair">
                <input type="text" placeholder="หมู่เรียน" v-model="sectionSearch" />
              </div>
            </div>
          </v-col>
          <v-col cols="12" md="2">
            <div class="field-container">
              <div class="label-input-pair">
                <select v-model="selectedOption">
                  <option value="" disabled selected>-- ประเภทการลา --</option>
                  <option value="">None</option>
                  <option value="ลากิจ">ลากิจ</option>
                  <option value="ลาป่วย">ลาป่วย</option>
                  <option value="อื่นๆ">อื่นๆ</option>
                </select>
              </div>
            </div>
          </v-col>
        </v-row>
        <v-row>
          <table class="leave-requests-table">
            <thead>
              <tr>
                <th>รหัสประจำตัว</th>
                <th>รายชื่อนิสิต</th>
                <th>Course Name</th>
                <th>Course Section</th>
```

```
                <th>Leave Type</th>
                <th>Start Date</th>
                <th>End Date</th>
                <th>Actions</th>
              </tr>
            </thead>
            <tbody>
              <tr
                v-for="request in filteredRequests"
                :key="request.id"
                @click="navigateToDetail(request.id)"
                class="request-row"
              >
                <!-- <td>{{ request.id }}</td> -->
                <td>{{ request.student_data.user_data.user_id }}</td>
                <td>{{ request.student_data.fname }} {{
request.student_data.lname }}</td>
                <td>{{ request.course_data.name }}</td>
                <td>{{ request.course_data.section }}</td>
                <td>{{ request.leave_request_data.leave_type }}</td>
                <td>{{ request.leave_request_data.start_date }}</td>
                <td>{{ request.leave_request_data.end_date }}</td>
                <td class="actions-cell">
                  <template v-if="request.status === 'pending'">
                    <v-btn small color="#02BC77" @click.stop="onApprove(request)"
>
                      อนุมัติ
                    </v-btn>
                    <div>
                      <v-btn small color="red" @click.stop="onReject(request)"
style="margin-top: 4%" >
                        ไม่อนุมัติ
                      </v-btn>
                    </div>

                  </template>
                </td>
              </tr>
            </tbody>
          </table>
        </v-row>
      </div>
    </div>
  </div>
</template>
```

รูปที่ 4.32 ตัวอย่างโค้ดส่วนเทมเพลตของ RequestTeacherView.vue

```
<script>
export default {
  setup() {
    const router = useRouter()
    const toastStore = useToastStore()
    const userStore = useUserStore()
    function navigateToDetail(id) {
      router.push({ name: 'DetailTeacherPage', params: { id } })
    }
    return {
      navigateToDetail,
      toastStore,
      userStore
    }
  },
  components: {
```

```
    Navbar,
    TopNavBar
  },
  mounted() {
    this.fetchLeaveRequests()
  },
  computed: {
    filteredRequests() {
      return this.leaveRequests.filter((request) => {
        const searchTermLower = this.search.toLowerCase();
        const matchesCourse =
request.course_data.name.toLowerCase().includes(searchTermLower);
        const matchesLeaveType = this.selectedOption
          ? request.leave_request_data.leave_type === this.selectedOption
          : true;
        const matchesIdStudentSearch = this.idStudentSearch
          ? request.student_data.user_data.user_id.includes(this.idStudentSearch)
          : true;
        const matchesSection = this.sectionSearch
          ? request.course_data.section.toString() === this.sectionSearch
          : true;
        const fullName = `${request.student_data.fname}
${request.student_data.lname}`.toLowerCase();
        const searchTerms = this.nameSearch.toLowerCase().split('
').filter(Boolean);
        const matchesName = searchTerms.length
          ? searchTerms.every(term => fullName.includes(term))
          : true;

        return matchesCourse && matchesLeaveType && matchesIdStudentSearch &&
matchesName && matchesSection;
      });
    }
  },
  data() {
    return {
      leaveRequests: [],
      loading: false,
      headers: [
        { text: 'ID', align: 'start', value: 'id', sortable: true },
        { text: 'Course Name', value: 'course_data.name' },
        { text: 'Course Section', value: 'course_data.section' },
        { text: 'Leave Type', value: 'leave_request_data.leave_type' },
        { text: 'Start Date', value: 'leave_request_data.start_date' },
        { text: 'End Date', value: 'leave_request_data.end_date' },
        { text: 'Actions', value: 'actions' }
      ],
      // getTeacherLogin
      testId: '',
      testeacherId: '',
      teacher: [],
      user: [],
      // for Search
      search: '',
      selectedOption: '',
      statusSearch: '',
      nameSearch: '',
      idStudentSearch: '',
      sectionSearch: '',

    }
  },
  methods: {
    async getTeacherLogin() {
      this.userStore.initStore()
      this.user = this.userStore.user
      this.testId = this.userStore.user.id
      if (this.testId) {
        const url = `/teacher/?user_id=${this.testId}`
```

```
        await axios
          .get(url)
          .then((response) => {
            this.teacher = response.data[0]
            this.testeacherId = this.teacher.id

          })
          .catch((error) => {
            console.log('error', error)
          })
      } else {
        this.student = []
      }
    },
    getUniqueSubjects(items) {
      const unique = {}
      items.forEach((item) => {
        const uniqueKey = `${item.course_data.name}-${item.leave_request_id}`
        if (!unique[uniqueKey]) {
          unique[uniqueKey] = item
        }
      })
      return Object.values(unique)
    },
    async fetchLeaveRequests() {
      this.loading = true
      await this.getTeacherLogin()
      console.log('testeacher ID: ' + this.testeacherId)
      try {
        if (this.testeacherId) {
          const response = await axios.get(
            `/leaveDetail/?teacher_id=${this.testeacherId}&status=pending`
          )
          this.leaveRequests = response.data
          this.leaveRequests = this.getUniqueSubjects(response.data)
        } else {
          console.log("Doesn't have Teacher ID:  " + this.testeacherId)
        }
      } catch (error) {
        console.error('There was an error fetching the leave requests:', error)
      } finally {
        this.loading = false
      }
    },
    async onApprove(request) {
      const updatedData = {
        status: 'approve'
      };
      try {
        const response = await
axios.put(`/leaveDetail/update_multiple/?course_id=${request.course_id}&leave_reque
st_id=${request.leave_request_id}`, updatedData);
        console.log('Data approved successfully:', response.data);
        await this.approveBy(request);
        await this.fetchLeaveRequests();
      } catch (error) {
        console.error('Error approving data:', error);
      }
    },
    async approveBy(request){
      const newApproveIdBy = this.testeacherId;
      const updatedData = {
        approve_id_by: newApproveIdBy
      };
      try {
        const response = await
axios.put(`/leaveDetail/update_multiple/?course_id=${request.course_id}&leave_reque
st_id=${request.leave_request_id}`, updatedData);
        console.log('Data approved successfully:', response.data);
```

```
      } catch (error) {
        console.error('Error approving data:', error);
      }
    },
    async onReject(request) {
      const updatedData = {
        status: 'reject'
      };
      try {
        const response = await axios
        .put(`/leaveDetail/update_multiple/?course_id=${request.course_id}&leave_re
quest_id=${request.leave_request_id}`, updatedData);
        console.log('Data approved successfully:', response.data);
        await this.approveBy(request);
        await this.fetchLeaveRequests();
      } catch (error) {
        console.error('Error approving data:', error);
      }
    },
  }
}
</script>
```

รูปที่ 4.33 ตัวอย่างโค้ดส่วนสคริปต์ของ RequestTeacherView.vue

การพัฒนาระบบ Backend

```
class LeaveRequestDetailViewSet(viewsets.ModelViewSet):
    queryset = LeaveRequestDetail.objects.all()
    serializer_class = LeaveRequestDetailSerializer


    def get_queryset(self):
        queryset = super().get_queryset()
        leave_request_id = self.request.query_params.get('leave_request_id')
        student_id = self.request.query_params.get('student_id')
        teacher_id = self.request.query_params.get('teacher_id')
        course_id = self.request.query_params.get('course_id')
        status = self.request.query_params.get('status')


        if leave_request_id:
            queryset = queryset.filter(leave_request_id_by=leave_request_id)

        if teacher_id:
            queryset = queryset.filter(teacher_id=teacher_id)
        if course_id:
            queryset = queryset.filter(course_id=course_id)
        if student_id:
            queryset = queryset.filter(student_id=student_id)
        if status:
            queryset = queryset.filter(status__icontains=status)

        return queryset


 @action(detail=False, methods=['put'])
    def update_multiple(self, request):
```

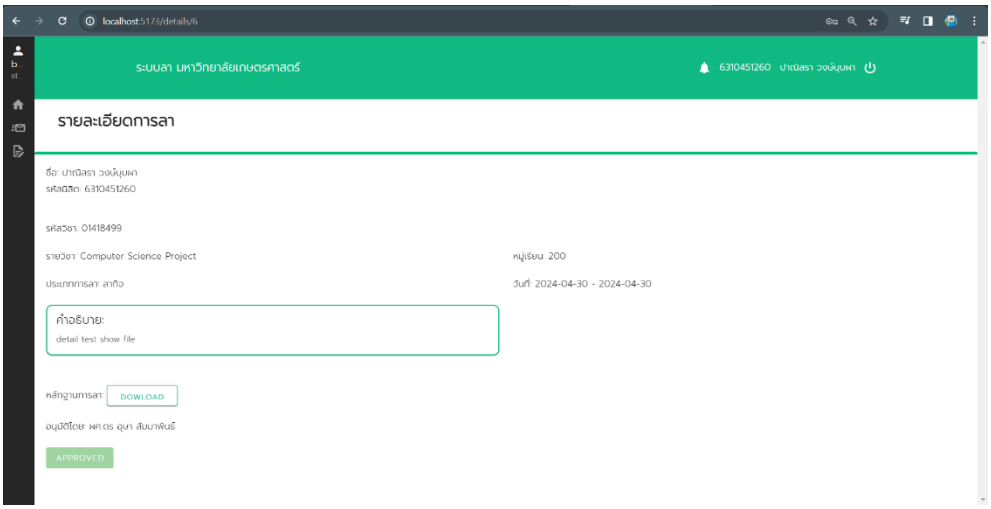
```
        course_id = request.query_params.get('course_id')
        leave_request_id = request.query_params.get('leave_request_id')
        status = request.data.get('status')

        approve_id_by = request.data.get('approve_id_by')

        if course_id and leave_request_id and status:
            updated_count = LeaveRequestDetail.objects.filter(
                course_id=course_id,
                leave_request_id=leave_request_id
            ).update(status=status)  # Direct update without serialization

            return Response({'updated_records': updated_count})

        if course_id and leave_request_id and approve_id_by:
            updated_count = LeaveRequestDetail.objects.filter(
                course_id=course_id,
                leave_request_id=leave_request_id
            ).update(approve_id_by=approve_id_by)  # Update approve_id_by field
directly

            return Response({'updated_records': updated_count})
        else:
            return HttpResponseBadRequest('Invalid parameters provided')
```

รูปที่ 4.34 ตัวอย่างโค้ดส่วน LeaveRequestDetailViewSet

4.4 แสดงรายละเอียดการลา

4.4.1 แสดงรายละเอียดการลาของนิสิต

นิสิตสามารถดูคำขอลาและรายละเอียดทั้งหมดที่ส่งไปให้อาจารย์และยังสามารถยกเลิกการแจ้งลาภายในหน้านี้ได้



รูปที่ 4.35 โปรแกรมส่วนแสดงรายละเอียดการลาของนิสิต

การพัฒนาระบบ Frontend

```
<template>
  <div class="layout-container">
    <Navbar />

    <div class="right-section">
      <TopNavBar />
      <div class="content-Page">
        <slot></slot>

        <div class="head">รายละเอียดการลา</div>
      </div>
      <div class="content">
        <v-row>
          <v-col cols="12" md="6">
            <div class="leaveblock1">
              <div v-if="leaveRequest.student_data">
                ชื่อ: {{ leaveRequest.student_data.fname }}
                {{ leaveRequest.student_data.lname }}
              </div>
            </div>
            <div class="leaveblock2">
              <div v-if="leaveRequest.student_data">
                รหัสนิสิต: {{ leaveRequest.student_data.user_data.user_id }}
              </div>
            </div>
          </v-col>
        </v-row>

        <v-row>
          <v-col cols="12" md="6">
            <div v-if="leaveRequest.course_data">
              รหัสวิชา: {{ leaveRequest.course_data.course_id }}
            </div>
          </v-col>
          <v-col cols="12" md="6">
          </v-col>
        </v-row>
        <v-row>
          <v-col cols="12" md="6">
            <div v-if="leaveRequest.course_data">
              รายวิชา: {{ leaveRequest.course_data.name }}
            </div>
          </v-col>
          <v-col cols="12" md="6">
            <div>หมู่เรียน: {{ leaveRequest.course_data.section }}</div>
          </v-col>
        </v-row>
        <v-row>
          <v-col cols="12" md="6">
            <div>
              ประเภทการลา: {{ leaveRequest.leave_request_data.leave_type }}
            </div>
          </v-col>
          <v-col cols="12" md="6">
            <div>
              วันที่: {{ leaveRequest.leave_request_data.start_date }} -
              {{ leaveRequest.leave_request_data.end_date }}
            </div>
          </v-col>
        </v-row>
        <v-row>
          <v-col cols="12" md="6">
            <v-card class="card columns">
```

```
              <v-card-title>คำอธิบาย:</v-card-title>
              <v-card-text>
                <template
                  v-for="index in Math.ceil(
                    leaveRequest.leave_request_data.description.length / 100
                  )"
                >
                  <div>
                    {{
                      leaveRequest.leave_request_data.description.slice(
                        (index - 1) * 100,
                        index * 100
                      )
                    }}
                  </div>
                </template>
              </v-card-text>
            </v-card>
          </v-col>
        </v-row>
        <v-row>
          <v-col cols="12">
            <div>
              หลักฐานการลา:
              <span v-for="file in files">
                <v-btn class="leave-addBtn" type="submit"
@click="dowloadWithAxios(file.pdf, file.id)">Dowload</v-btn>
              </span>
            </div>
          </v-col>
        </v-row>
        <v-row>
          <v-col cols="12" md="6">
            <div v-if="leaveRequest.status === 'approve'">
              อนุมัติโดย: {{ leaveRequest.approve_id_by_data.prefix }} {{
leaveRequest.approve_id_by_data.fname }} {{ leaveRequest.approve_id_by_data.lname
}}
            </div>
            <div v-if="leaveRequest.status === 'reject'">
              ไม่อนุมัติโดย: {{ leaveRequest.approve_id_by_data.prefix }} {{
leaveRequest.approve_id_by_data.fname }} {{ leaveRequest.approve_id_by_data.lname
}}
            </div>
            <div v-if="leaveRequest.status === 'pending'">
              อนุมัติโดย: รอดำเนินการ
            </div>
          </v-col>
        </v-row>
        <v-row>
          <v-col cols="12" md="6">
            <v-btn
              small
              :color="
                leaveRequest.status === 'approve'
                  ? 'green'
                  : leaveRequest.status === 'reject'
                    ? 'grey'
                    : 'primary'
              "
              @click.stop="
                leaveRequest.status === 'pending'
                  ? onPending(leaveRequest)
                  : leaveRequest.status === 'approve'
                    ? onApproved(leaveRequest)
                    : approveRequest(leaveRequest)
              "
              :disabled="
                leaveRequest.status === 'approve' ||
                leaveRequest.status === 'reject' ||
```

```
                leaveRequest.status === 'pending'
              "
            >
              {{
                leaveRequest.status === 'approve'
                  ? 'Approved'
                  : leaveRequest.status === 'reject'
                    ? 'Rejected'
                    : leaveRequest.status === 'pending'
                      ? 'รออนุมัติ'
                      : 'Approve'
              }}
            </v-btn>
            <v-btn
              small
              color="red"
              @click.stop="cancelRequest(leaveRequest)"
              style="margin-left: 5%"
              v-if="leaveRequest.status === 'pending'"
            >
              ยกเลิก
            </v-btn>
          </v-col>
        </v-row>
      </div>
    </div>
  </div>
</template>
```

รูปที่ 4.36 ตัวอย่างโค้ดส่วนเทมเพลตของ DetailPage.vue

```
<script>
import axios from 'axios'
export default {
  components: {
    Navbar,
    TopNavBar
  },
  data() {
    return {
      leaveRequest: {
        id: '',
        student_data: { fname: '', lname: '', user_data: {user_id: '',}, },
        course_data: { name: '', id: '', course_id: '' },
        leave_request_data: {
          id: '',
          start_date: '',
          end_date: '',
          leave_type: '',
          description: ''
        }
      },
      requestId: null,
      files: [],
      upload_status: '',
    }
  },
  async created() {
    this.requestId = this.$route.params.id
    await this.fetchLeaveRequest()
    await this.getFile()
  },
```

```
  methods: {
    async fetchLeaveRequest() {
      // Use the `requestId` to fetch the correct leave request details
      const url = `/leaveDetail/${this.requestId}`

      await axios
        .get(url)
        .then((response) => {
          this.leaveRequest = response.data
          console.log('Leave request')
          console.log(this.leaveRequest)
        })
        .catch((error) => {
          console.error(
            'There was an error fetching the leave request details:',
            error
          )
        })
    },
    async cancelRequest(request) {
      const { value: confirmed } = await this.$swal.fire({
        title: 'ยืนยันการลบ',
        text: 'คุณต้องการลบรายการนี้หรือไม่?',
        icon: 'warning',
        showCancelButton: true,
        confirmButtonColor: '#02bc77',
        cancelButtonColor: '#d33',
        confirmButtonText: 'ใช่, ลบ!',
        cancelButtonText: 'ยกเลิก'
      })

      if (confirmed) {
        const leave_request_id = request.leave_request_data.id
        const course_id = request.course_data.id
        const url = `/leaveDetail/delete_multiple/`
        const params = {
          course_id: course_id,
          leave_request_id: leave_request_id
        }

        try {
          const response = await axios.delete(url, { params: params })
          console.log('Requests cancelled successfully:', response.data)
          this.$swal.fire({
            title: 'ลบสำเร็จ',
            text: 'ลบรายการเรียบร้อยแล้ว',
            icon: 'success',
            customClass: {
              confirmButton: 'btn btn-success'
            },
            confirmButtonColor: '#02bc77'
          })
          this.$router.push('/deleted-page')
        } catch (error) {
          console.error(
            'Error cancelling requests:',
            error.response || error.message
          )
          this.$swal.fire(
            'เกิดข้อผิดพลาด',
            'เกิดข้อผิดพลาดในการลบรายการ',
            'error'
          )
        }
      }
    },
    async getFile() {
      await
```

```
axios.get(`/files/?leave_request_id=${this.leaveRequest.leave_request_data.id}`).th
en(
        response => {
          console.log(response.data)
          this.files = response.data

        }
      ).catch(error => {
        this.upload_status= 'File NOt Upload Success'
      })
    },
    dowloadWithAxios(url, title){
      axios({
        method: 'get',
        url,
        responseType: 'arraybuffer',

      }).then(response =>{
        this.forceDownload(response, title)
      }).catch((error) => console.log(error))
    },
    forceDownload(response, title){
      const url = window.URL.createObjectURL(new Blob([response.data], {type:
'application/pdf'}))
      const link = document.createElement('a')
      link.href = url
      // link.setAttribute('download', title+'.pdf') // Corrected the typo here
      link.setAttribute('download', `${title}.pdf`);
      document.body.appendChild(link)
      link.click()
    },
  }
}
</script>
```

รูปที่ 4.37 ตัวอย่างโค้ดส่วนสคริปต์ของ DetailPage.vue

การพัฒนาระบบ Backend

```
class LeaveRequestViewSet(viewsets.ModelViewSet):
    queryset = LeaveRequest.objects.all()
    serializer_class = LeaveRequestSerializer

    def get_queryset(self):
        queryset = super().get_queryset()
        id = self.request.query_params.get('id')
        start_date = self.request.query_params.get('start_date')
        end_date = self.request.query_params.get('end_date')
        leave_type = self.request.query_params.get('leave_type')
        status = self.request.query_params.get('status')

        if start_date:
            queryset = queryset.filter(start_date__icontains=start_date)

        if end_date:
            queryset = queryset.filter(end_date__icontains=end_date)

        if leave_type:
            queryset = queryset.filter(leave_type__icontains=leave_type)

        if status:
```

```
            queryset = queryset.filter(status__icontains=status)

        if id:
            queryset = queryset.filter(id=id)

        return queryset

    def create(self, request, *args, **kwargs):
        serializer = self.get_serializer(data=request.data)
        if serializer.is_valid():
            leaveRequest = serializer.save()
            lid = leaveRequest.id

            leaveRequest_url = f'http://localhost:8000/leaveRequest/{lid}/'
            leaveRequest_id = lid

            response_data = serializer.data
            response_data['id-url'] = leaveRequest_url
            response_data['leaveRequest_id'] = leaveRequest_id

            return Response(response_data, status=status.HTTP_201_CREATED)
        else:
            return Response(serializer.errors,
status=status.HTTP_400_BAD_REQUEST)

    def destroy(self, request, *args, **kwargs):
        try:
            leave_request = LeaveRequest.objects.get(pk=kwargs.get('pk'))
            leave_request.delete()
            return Response(status=status.HTTP_204_NO_CONTENT)
        except LeaveRequest.DoesNotExist:
            return Response(status=status.HTTP_404_NOT_FOUND)
        except Exception as e:
            return Response(str(e), status=status.HTTP_500_INTERNAL_SERVER_ERROR)
```

รูปที่ 4.38 ตัวอย่างโค้ดส่วน LeaveRequestViewSet

```
class LeaveRequestDetailViewSet(viewsets.ModelViewSet):
    queryset = LeaveRequestDetail.objects.all()
    serializer_class = LeaveRequestDetailSerializer


    def get_queryset(self):
        queryset = super().get_queryset()
        leave_request_id = self.request.query_params.get('leave_request_id')
        student_id = self.request.query_params.get('student_id')
        teacher_id = self.request.query_params.get('teacher_id')
        course_id = self.request.query_params.get('course_id')
        status = self.request.query_params.get('status')

        if leave_request_id:
            queryset = queryset.filter(leave_request_id_by=leave_request_id)

        if teacher_id:
            queryset = queryset.filter(teacher_id=teacher_id)
        if course_id:
            queryset = queryset.filter(course_id=course_id)
        if student_id:
            queryset = queryset.filter(student_id=student_id)
        if status:
            queryset = queryset.filter(status__icontains=status)

        return queryset
```

```
    def destroy(self, request, *args, **kwargs):
        try:
            course_id = request.query_params.get('course_id')
            leave_request_id = request.query_params.get('leave_request_id')

            if course_id and leave_request_id:

                leave_request_details = LeaveRequestDetail.objects.filter(
                    course_id=course_id,
                    leave_request_id=leave_request_id
                )

                if not leave_request_details.exists():
                    return Response(status=status.HTTP_404_NOT_FOUND)

                leave_request_details.delete()
            else:

                leave_request_detail = self.get_object()
                leave_request_detail.delete()

            return Response(status=status.HTTP_204_NO_CONTENT)

        except LeaveRequestDetail.DoesNotExist:
            return Response(status=status.HTTP_404_NOT_FOUND)
        except Exception as e:

            return Response(str(e), status=status.HTTP_500_INTERNAL_SERVER_ERROR)

    @action(detail=False, methods=['delete'])
    def delete_multiple(self, request):
        course_id = request.query_params.get('course_id')
        leave_request_id = request.query_params.get('leave_request_id')

        if course_id and leave_request_id:
            LeaveRequestDetail.objects.filter(
                course_id=course_id,
                leave_request_id=leave_request_id
            ).delete()
            return Response(status=status.HTTP_204_NO_CONTENT)
        return Response(status=status.HTTP_400_BAD_REQUEST)
```

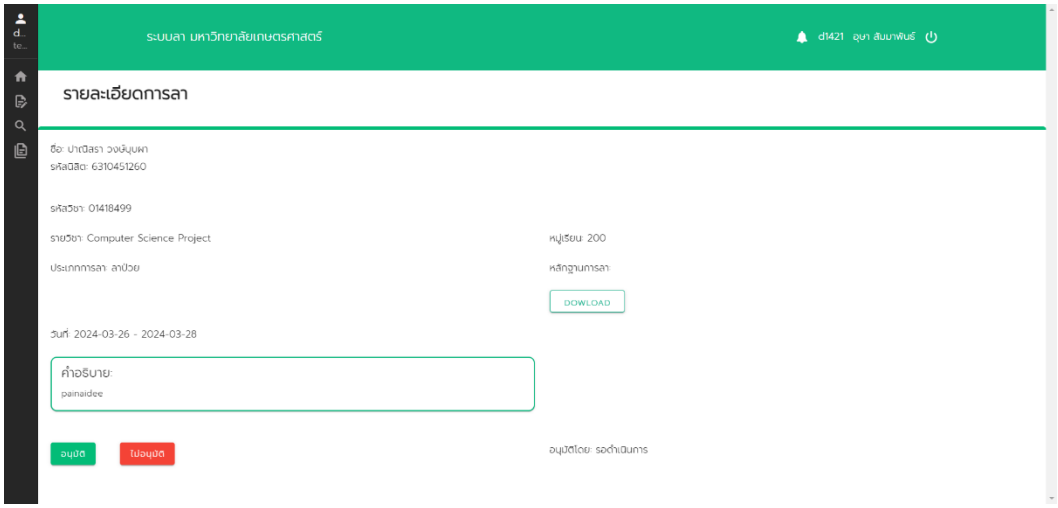
รูปที่ 4.39 ตัวอย่างโค้ดส่วน `LeaveRequestDetailViewSet`

```
class FilesViewSet(viewsets.ModelViewSet):
    queryset = Files.objects.all()
    serializer_class = FileSerializer
    def get_queryset(self):
        queryset = super().get_queryset()
        leave_request_id = self.request.query_params.get('leave_request_id')


        if leave_request_id:
            queryset = queryset.filter(leave_request_id=leave_request_id)


        return queryset
```

รูปที่ 4.40 ตัวอย่างโค้ดส่วน `FilesViewSet`

4.4.2 แสดงรายละเอียดการลาของนิสิตที่ส่งให้อาจารย์

ระบบย่อยนี้อาจารย์สามารถดูคำขอลาและรายละเอียดของนิสิตที่อาจารย์เลือกและยังสามารถอนุมัติหรือไม่อนุมัติภายในหน้านี้ได้



รูปที่ 4.41 โปรแกรมส่วนแสดงรายละเอียดการลาของนิสิตที่ส่งให้อาจารย์

การพัฒนาระบบ Frontend

```
<template>
  <div class="layout-container">
    <Navbar />

    <div class="right-section">
      <TopNavBar />
      <div class="content-Page">
        <slot></slot>

        <div class="head">รายละเอียดการลา</div>
      </div>
      <div class="content">
        <v-row>
          <v-col cols="12" md="6">
            <div class="leaveblock1">
              <div v-if="leaveRequest.student_data">
                ชื่อ: {{ leaveRequest.student_data.fname }}
                {{ leaveRequest.student_data.lname }}
              </div>
            </div>
            <div class="leaveblock2">
              <div v-if="leaveRequest.student_data">
                รหัสนิสิต: {{ leaveRequest.student_data.user_data.user_id }}
              </div>
            </div>
          </v-col>
        </v-row>
        <v-row>
          <v-col cols="12" md="6">
```

```
            <div v-if="leaveRequest.course_data">
              รหัสวิชา: {{ leaveRequest.course_data.course_id }}
            </div>
          </v-col>
          <v-col cols="12" md="6">
          </v-col>
        </v-row>
        <v-row>
          <v-col cols="12" md="6">
            <div v-if="leaveRequest.course_data">
              รายวิชา: {{ leaveRequest.course_data.name }}
            </div>
          </v-col>
          <v-col cols="12" md="6">
            <div>
              หมู่เรียน: {{ leaveRequest.course_data.section }}
            </div>
          </v-col>
        </v-row>
        <v-row>
          <v-col cols="12" md="6">
            <div>
              ประเภทการลา: {{ leaveRequest.leave_request_data.leave_type }}
            </div>
          </v-col>
          <v-col cols="12" md="6">
            <div>
              หลักฐานการลา:
              <span v-for="file in files">
                <!-- {{ file.pdf}} -->
                <br><br>
                <v-btn class="leave-addBtn" type="submit"
@click="dowloadWithAxios(file.pdf, file.id)">Dowload</v-btn>
              </span>
            </div>
          </v-col>
        </v-row>
        <v-row>

        </v-row>
        <v-row>
          <v-col cols="12" md="6">
            <div>
              วันที่: {{ leaveRequest.leave_request_data.start_date }} -
              {{ leaveRequest.leave_request_data.end_date }}
            </div>
          </v-col>
          <v-col cols="12" md="6">
          </v-col>
        </v-row>
        <v-row>
          <v-col cols="12" md="6">
            <v-card class="card columns">
                <v-card-title>คำอธิบาย:</v-card-title>
                <v-card-text>
                  <template v-for="index in
Math.ceil(leaveRequest.leave_request_data.description.length / 100)">
                    <div>{{
leaveRequest.leave_request_data.description.slice((index - 1) * 100, index * 100)
}}</div>
                  </template>
                </v-card-text>
              </v-card>
          </v-col>
          <v-col cols="12" md="6"></v-col>
        </v-row>
        <v-row>
          <v-col cols="12" md="6">
```

```
            <!-- content -->
            <template v-if="leaveRequest.status === 'pending'">
                <v-btn small color="#02BC77"
@click.stop="onApprove(leaveRequest)" >
                  อนุมัติ
                </v-btn>
                <v-btn small color="red" @click.stop="onReject(leaveRequest)"
style="margin-left: 5%" >
                  ไม่อนุมัติ
                </v-btn>
            </template>
            <template v-if="leaveRequest.status === 'approve'">
                <v-btn small color="green" disabled=True>
                  อนุมัติแล้ว
                </v-btn>
            </template>
            <template v-if="leaveRequest.status === 'reject'">
                <v-btn small color="grey"   disabled=True>
                  ไม่อนุมัติ
                </v-btn>
            </template>
          </v-col>
          <v-col cols="12" md="6">
            <div v-if="leaveRequest.status === 'approve'">
              อนุมัติโดย: {{ leaveRequest.approve_id_by_data.prefix }} {{
leaveRequest.approve_id_by_data.fname }} {{ leaveRequest.approve_id_by_data.lname
}}
            </div>
            <div v-if="leaveRequest.status === 'reject'">
              ไม่อนุมัติโดย: {{ leaveRequest.approve_id_by_data.prefix }} {{
leaveRequest.approve_id_by_data.fname }} {{ leaveRequest.approve_id_by_data.lname
}}
            </div>
            <div v-if="leaveRequest.status === 'pending'">
              อนุมัติโดย: รอดำเนินการ
            </div>
          </v-col>
        </v-row>
        <v-row>
        </v-row>
      </div>
    </div>
  </div>
</template>
```

รูปที่ 4.42 ตัวอย่างโค้ดส่วนเทมเพลตของ DetailTeacherPage.vue

```
<script>
export default {
  data() {
    return {
      leaveRequest: {
        id: '',
        student_data: { fname: '', lname: '', user_data: {user_id: '',}, },
        course_data: { name: '', id: '', section: '', course_id: '' },
        leave_request_data: {
          id: '',
          start_date: '',
          end_date: '',
          leave_type: '',
          description: '',
        },
        leave_request_id: '',
```

```
      },
      requestId: null,
      files: [],
      upload_status: '',
    }
  },
  async created() {
    this.requestId = this.$route.params.id
    await this.fetchLeaveRequest()
    await this.getFile()
  },
  methods: {
    async getFile() {
      await
axios.get(`/files/?leave_request_id=${this.leaveRequest.leave_request_data.id}`).th
en(
        response => {
          console.log(response.data)
          this.files = response.data

        }
      ).catch(error => {
        this.upload_status= 'File NOt Upload Success'
      })
    },
    dowloadWithAxios(url, title){
      axios({
        method: 'get',
        url,
        responseType: 'arraybuffer',

      }).then(response =>{
        this.forceDownload(response, title)
      }).catch((error) => console.log(error))
    },
    forceDownload(response, title){
      const url = window.URL.createObjectURL(new Blob([response.data], {type:
'application/pdf'}))
      const link = document.createElement('a')
      link.href = url
      link.setAttribute('download', `${title}.pdf`);
      document.body.appendChild(link)
      link.click()
    },
    async fetchLeaveRequest() {
      const url = `/leaveDetail/${this.requestId}`

      await axios
        .get(url)
        .then((response) => {
          this.leaveRequest = response.data
          console.log('Leave request')
          console.log(this.leaveRequest)
        })
        .catch((error) => {
          console.error(
            'There was an error fetching the leave request details:',
            error
          )
        })
    },
    async onReject(request) {
      const { value: confirmedReject } = await this.$swal.fire({
        title: 'ยืนยันการปฏิเสธคำขอลา',
        text: 'คุณต้องการปฏิเสธคำขอลารายการนี้หรือไม่?',
        icon: 'warning',
        showCancelButton: true,
        confirmButtonColor: '#02bc77',
```

```
        cancelButtonColor: '#d33',
        confirmButtonText: 'ยืนยัน',
        cancelButtonText: 'ยกเลิก'
      })
      if (confirmedReject) {
        const updatedData = {
          status: 'reject'
        };
        try {
          const response = await
axios.put(`/leaveDetail/update_multiple/?course_id=${request.course_id}&leave_reque
st_id=${request.leave_request_id}`, updatedData);
          console.log('Data approved successfully:', response.data);
          await this.approveBy(request);
          await this.fetchLeaveRequest();
        } catch (error) {
          console.error('Error approving data:', error);
        }
      }
    },
    async onApprove(request) {
      const { value: confirmedApprove } = await this.$swal.fire({
        title: 'ยืนยันการอนุมัติ',
        text: 'คุณต้องการอนุมัติรายการนี้หรือไม่?',
        icon: 'warning',
        showCancelButton: true,
        confirmButtonColor: '#02bc77',
        cancelButtonColor: '#d33',
        confirmButtonText: 'ยืนยัน',
        cancelButtonText: 'ยกเลิก'
      })
      if (confirmedApprove) {
        const updatedData = {
          status: 'approve'
        };
        try {
          const response = await
axios.put(`/leaveDetail/update_multiple/?course_id=${request.course_id}&leave_reque
st_id=${request.leave_request_id}`, updatedData);
          await this.approveBy(request);
          await this.fetchLeaveRequest();
        } catch (error) {
          console.error('Error approving data:', error);
        }
      }
    },
    async approveBy(request){
      const newApproveIdBy = request.teacher_id;
      const updatedData = {
        approve_id_by: newApproveIdBy
      };
      try {
        const response = await
axios.put(`/leaveDetail/update_multiple/?course_id=${request.course_id}&leave_reque
st_id=${request.leave_request_id}`, updatedData);
        console.log('Data approved successfully:', response.data);

      } catch (error) {
        console.error('Error approving data:', error);
      }
    }
  }
}
</script>
```

รูปที่ 4.43 ตัวอย่างโค้ดส่วนสคริปต์ของ DetailTeacherPage.vue

การพัฒนาระบบ Backend

```
class LeaveRequestViewSet(viewsets.ModelViewSet):
    queryset = LeaveRequest.objects.all()
    serializer_class = LeaveRequestSerializer

    def get_queryset(self):
        queryset = super().get_queryset()
        id = self.request.query_params.get('id')
        start_date = self.request.query_params.get('start_date')
        end_date = self.request.query_params.get('end_date')
        leave_type = self.request.query_params.get('leave_type')
        status = self.request.query_params.get('status')

        if start_date:
            queryset = queryset.filter(start_date__icontains=start_date)

        if end_date:
            queryset = queryset.filter(end_date__icontains=end_date)

        if leave_type:
            queryset = queryset.filter(leave_type__icontains=leave_type)

        if status:
            queryset = queryset.filter(status__icontains=status)

        if id:
            queryset = queryset.filter(id=id)

        return queryset

    def create(self, request, *args, **kwargs):
        serializer = self.get_serializer(data=request.data)
        if serializer.is_valid():
            leaveRequest = serializer.save()
            lid = leaveRequest.id

            leaveRequest_url = f'http://localhost:8000/leaveRequest/{lid}/'
            leaveRequest_id = lid

            response_data = serializer.data
            response_data['id-url'] = leaveRequest_url
            response_data['leaveRequest_id'] = leaveRequest_id

            return Response(response_data, status=status.HTTP_201_CREATED)
        else:
            return Response(serializer.errors,
status=status.HTTP_400_BAD_REQUEST)
```

รูปที่ 4.44 ตัวอย่างโค้ดส่วน LeaveRequestViewSet

```
class LeaveRequestDetailViewSet(viewsets.ModelViewSet):
    queryset = LeaveRequestDetail.objects.all()
    serializer_class = LeaveRequestDetailSerializer


    def get_queryset(self):
        queryset = super().get_queryset()
        leave_request_id = self.request.query_params.get('leave_request_id')
        student_id = self.request.query_params.get('student_id')
        teacher_id = self.request.query_params.get('teacher_id')
```

```
        course_id = self.request.query_params.get('course_id')
        status = self.request.query_params.get('status')


        if leave_request_id:
            queryset = queryset.filter(leave_request_id_by=leave_request_id)

        if teacher_id:
            queryset = queryset.filter(teacher_id=teacher_id)
        if course_id:
            queryset = queryset.filter(course_id=course_id)
        if student_id:
            queryset = queryset.filter(student_id=student_id)
        if status:
            queryset = queryset.filter(status__icontains=status)

        return queryset


    @action(detail=False, methods=['put'])
    def update_multiple(self, request):
        course_id = request.query_params.get('course_id')
        leave_request_id = request.query_params.get('leave_request_id')
        status = request.data.get('status')

        approve_id_by = request.data.get('approve_id_by')

        if course_id and leave_request_id and status:
            updated_count = LeaveRequestDetail.objects.filter(
                course_id=course_id,
                leave_request_id=leave_request_id
            ).update(status=status)  # Direct update without serialization

            return Response({'updated_records': updated_count})

        if course_id and leave_request_id and approve_id_by:
            updated_count = LeaveRequestDetail.objects.filter(
                course_id=course_id,
                leave_request_id=leave_request_id
            ).update(approve_id_by=approve_id_by)  # Update approve_id_by field
directly

            return Response({'updated_records': updated_count})
        else:
            return HttpResponseBadRequest('Invalid parameters provided')
```

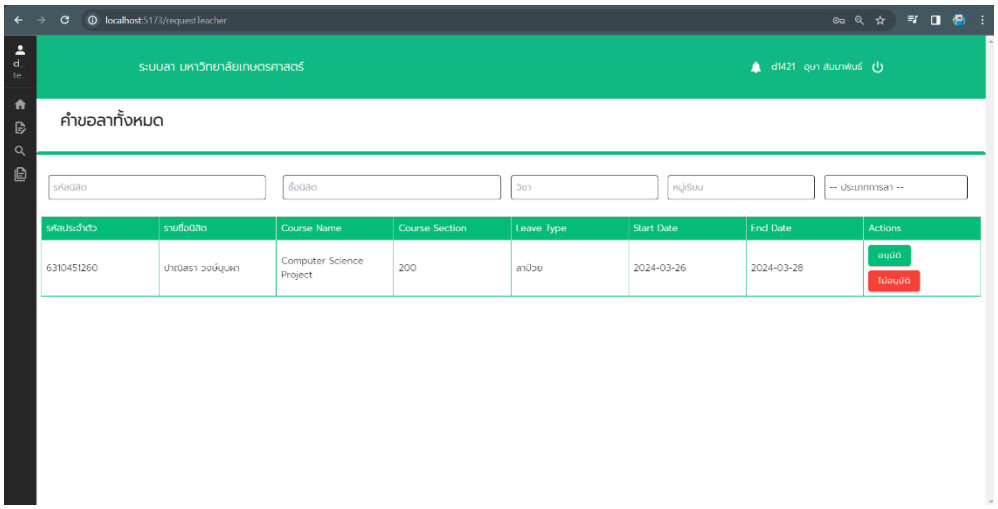
รูปที่ 4.45 ตัวอย่างโค้ดส่วน `LeaveRequestDetailViewSet`

```
class FilesViewSet(viewsets.ModelViewSet):
    queryset = Files.objects.all()
    serializer_class = FileSerializer
    def get_queryset(self):
        queryset = super().get_queryset()
        leave_request_id = self.request.query_params.get('leave_request_id')


        if leave_request_id:
            queryset = queryset.filter(leave_request_id=leave_request_id)
        return queryset
```

รูปที่ 4.46 ตัวอย่างโค้ดส่วน `FilesViewSet`

4.5 อนุมัติการลาและปฏิเสธการลา

ระบบย่อยนี้สามารถอนุมัติการลาและปฏิเสธการลาโดยให้อาจารย์ประจำวิชาพิจารณาคำขอลาของนิสิตเพื่ออนุมัติหรือไม่อนุมัติได้ อย่างรวดเร็วเนื่องจากมีการสรุปผลโดยการใช้ตารางเข้ามาช่วย



โปรแกรมส่วนแสดงคำขอลาทั้งหมดที่สามารถกดอนุมัติและไม่อนุมัติได้

4.5.1 อนุมัติการลา

เมื่อนิสิตส่งคำขอการลาผ่านระบบ, อาจารย์ประจำวิชาสามารถดูรายละเอียดและตรวจสอบคำขอลานั้นๆ และทำการอนุมัติตามความเหมาะสม

4.5.2 ปฏิเสธการลา

เมื่อนิสิตส่งคำขอการลาผ่านระบบ, อาจารย์ประจำวิชาสามารถดูรายละเอียดและตรวจสอบคำขอลานั้นๆ และทำการปฏิเสธตามความเหมาะสม

การพัฒนาระบบ Frontend

```
<template>
      <div class="content-Page">
        <slot></slot>

        <div class="head">คำขอลาทั้งหมด</div>
      </div>
      <div class="content">
        <v-row>
          <table class="leave-requests-table">
            <thead>
              <tr>
                <th>รหัสประจำตัว</th>
                <th>รายชื่อนิสิต</th>
                <th>Course Name</th>
                <th>Course Section</th>
                <th>Leave Type</th>
                <th>Start Date</th>
```

```
                <th>End Date</th>
                <th>Actions</th>
              </tr>
            </thead>
            <tbody>
              <tr
                v-for="request in filteredRequests"
                :key="request.id"
                @click="navigateToDetail(request.id)"
                class="request-row"
              >
                <!-- <td>{{ request.id }}</td> -->
                <td>{{ request.student_data.user_data.user_id }}</td>
                <td>{{ request.student_data.fname }} {{
request.student_data.lname }}</td>
                <td>{{ request.course_data.name }}</td>
                <td>{{ request.course_data.section }}</td>
                <td>{{ request.leave_request_data.leave_type }}</td>
                <td>{{ request.leave_request_data.start_date }}</td>
                <td>{{ request.leave_request_data.end_date }}</td>
                <td class="actions-cell">
                  <template v-if="request.status === 'pending'">
                    <v-btn small color="#02BC77" @click.stop="onApprove(request)"
>
                      อนุมัติ
                    </v-btn>
                    <div>
                      <v-btn small color="red" @click.stop="onReject(request)"
style="margin-top: 4%" >
                        ไม่อนุมัติ
                      </v-btn>
                    </div>

                  </template>
                </td>
              </tr>
            </tbody>
          </table>
        </v-row>
      </div>
    </div>
  </div>
</template>
```

รูปที่ 4.47 ตัวอย่างโค้ดส่วนเทมเพลตของ RequestTeacherView.vue

```
<script>
import axios from 'axios'
export default {
 //other code
  methods: {
    async onApprove(request) {
      const updatedData = {
        status: 'approve'
      };
      try {
        const response = await
axios.put(`/leaveDetail/update_multiple/?course_id=${request.course_id}&leave_reque
st_id=${request.leave_request_id}`, updatedData);
        console.log('Data approved successfully:', response.data);
        await this.approveBy(request);
        await this.fetchLeaveRequests();
      } catch (error) {
        console.error('Error approving data:', error);
      }
```

```
    },
    async approveBy(request){
      const newApproveIdBy = this.testeacherId;
      const updatedData = {
        approve_id_by: newApproveIdBy
      };
      try {
        const response = await
axios.put(`/leaveDetail/update_multiple/?course_id=${request.course_id}&leave_reque
st_id=${request.leave_request_id}`, updatedData);
        console.log('Data approved successfully:', response.data);

      } catch (error) {
        console.error('Error approving data:', error);
      }
    },
    async onReject(request) {
      const updatedData = {
        status: 'reject'
      };
      try {
        const response = await axios
        .put(`/leaveDetail/update_multiple/?course_id=${request.course_id}&leave_re
quest_id=${request.leave_request_id}`, updatedData);
        console.log('Data approved successfully:', response.data);
        await this.approveBy(request);
        await this.fetchLeaveRequests();
      } catch (error) {
        console.error('Error approving data:', error);
      }
    },
  }
}
</script>
```

รูปที่ 4.48 ตัวอย่างโค้ดส่วนสคริปต์ของ

การพัฒนาระบบ Backend

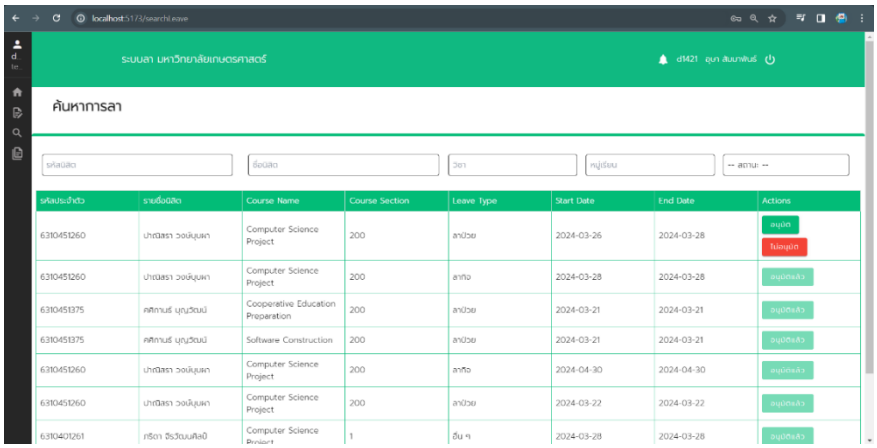
```
class LeaveRequestDetailViewSet(viewsets.ModelViewSet):
    queryset = LeaveRequestDetail.objects.all()
    serializer_class = LeaveRequestDetailSerializer


    def get_queryset(self):
        queryset = super().get_queryset()
        leave_request_id = self.request.query_params.get('leave_request_id')
        student_id = self.request.query_params.get('student_id')
        teacher_id = self.request.query_params.get('teacher_id')
        course_id = self.request.query_params.get('course_id')
        status = self.request.query_params.get('status')


        if leave_request_id:
            queryset = queryset.filter(leave_request_id_by=leave_request_id)

        if teacher_id:
            queryset = queryset.filter(teacher_id=teacher_id)
        if course_id:
            queryset = queryset.filter(course_id=course_id)
        if student_id:
            queryset = queryset.filter(student_id=student_id)
```

```
        if status:
            queryset = queryset.filter(status__icontains=status)

        return queryset


    @action(detail=False, methods=['put'])
    def update_multiple(self, request):
        course_id = request.query_params.get('course_id')
        leave_request_id = request.query_params.get('leave_request_id')
        status = request.data.get('status')

        approve_id_by = request.data.get('approve_id_by')

        if course_id and leave_request_id and status:
            updated_count = LeaveRequestDetail.objects.filter(
                course_id=course_id,
                leave_request_id=leave_request_id
            ).update(status=status)  # Direct update without serialization

            return Response({'updated_records': updated_count})

        if course_id and leave_request_id and approve_id_by:
            updated_count = LeaveRequestDetail.objects.filter(
                course_id=course_id,
                leave_request_id=leave_request_id
            ).update(approve_id_by=approve_id_by)  # Update approve_id_by field
directly

            return Response({'updated_records': updated_count})
        else:
            return HttpResponseBadRequest('Invalid parameters provided')
```

รูปที่ 4.49 ตัวอย่างโค้ดส่วน LeaveRequestDetailViewSet

4.6 ค้นหาการลาตามเงื่อนไข

ระบบย่อยนี้สามารถค้นหาการลาตามเงื่อนไขช่วยให้อาจารย์สามารถค้นหาข้อมูลเกี่ยวกับการลาของนิสิตตามเงื่อนไขที่กำหนดได้อย่างมีประสิทธิภาพ



รูปที่ 4.50 โปรแกรมส่วนค้นหาการลาตามเงื่อนไข

4.6.1 ค้นหาการลาตามเงื่อนไข

อาจารย์สามารถกำหนดเงื่อนไขการค้นหาต่างๆ เพื่อค้นหาการลาของนิสิต เช่น รหัสนิสิต, ชื่อ-นามสกุลของนิสิต, วิชา, หมู่เรียน,ประเภทของการลา, หรือสถานะของใบลา

4.6.2 การกรองข้อมูล

ช่วยให้อาจารย์สามารถกรองข้อมูลการลาตามเงื่อนไขที่กำหนด เพื่อลดขอบเขตของข้อมูลที่จะแสดงให้มีเฉพาะข้อมูลที่เป็นประโยชน์แก่อาจารย์

การพัฒนาระบบ Frontend

```
<template>
        <div class="content-Page">
          <slot></slot>

          <div class="head">ค้นหาการลา</div>
        </div>
        <div class="content">
          <!-- Search fields in a single row -->
          <v-row>
            <v-col cols="12" md="3">
              <div class="field-container">
                <div class="label-input-pair">
                  <input type="text" placeholder="รหัสนิสิต" v-
model="idStudentSearch" />
                </div>
              </div>
            </v-col>
            <v-col cols="12" md="3">
              <div class="field-container">
                <div class="label-input-pair">
                  <input type="text" placeholder="ชื่อนิสิต" v-model="nameSearch" />
                </div>
              </div>
            </v-col>
            <v-col cols="12" md="2">
              <div class="field-container">
                <div class="label-input-pair">
                  <input type="text" placeholder="วิชา" v-model="search" />
                </div>
              </div>
            </v-col>
            <v-col cols="12" md="2">
              <div class="field-container">
                <div class="label-input-pair">
                  <input type="text" placeholder="หมู่เรียน" v-model="sectionSearch"
/>
                </div>
              </div>
            </v-col>
            <v-col cols="12" md="2">
              <div class="field-container">
                <div class="label-input-pair">
                  <select v-model="statusSearch">
                    <option value="" disabled selected>
                      -- สถานะ --
                    </option>
                    <option value="">-</option>
                    <option value="pending">รอดำเนินการ</option>
                    <option value="approve">อนุมัติแล้ว</option>
                    <option value="reject">ไม่อนุมัติ</option>
```

```
                  </select>
            </v-col>
          </v-row>

          <!-- table -->
          <v-row>
            <table class="leave-requests-table">
              <thead>
                <tr>
                  <!-- <th>ID</th> -->
                  <th>รหัสประจำตัว</th>
                  <th>รายชื่อนิสิต</th>
                  <th>Course Name</th>
                  <th>Course Section</th>
                  <th>Leave Type</th>
                  <th>Start Date</th>
                  <th>End Date</th>
                  <th>Actions</th>
                </tr>
              </thead>
              <tbody>
                <tr
                  v-for="request in sortedAndFilteredLeaveRequests"
                  :key="request.id"
                  @click="navigateToDetail(request.id)"
                  class="request-row"
                >
                  <!-- <td>{{ request.id }}</td> -->
                  <td>{{ request.student_data.user_data.user_id }}</td>
                  <td>{{ request.student_data.fname }} {{
request.student_data.lname }}</td>
                  <td>{{ request.course_data.name }}</td>
                  <td>{{ request.course_data.section }}</td>
                  <td>{{ request.leave_request_data.leave_type }}</td>
                  <td>{{ request.leave_request_data.start_date }}</td>
                  <td>{{ request.leave_request_data.end_date }}</td>
                  <td class="actions-cell">
                    <template v-if="request.status === 'pending'">
                      <v-btn small color="#02BC77"
@click.stop="onApprove(request)" >
                        อนุมัติ
                      </v-btn>
                      <div>
                        <v-btn small color="red" @click.stop="onReject(request)"
style="margin-top: 4%">
                          ไม่อนุมัติ
                        </v-btn>
                      </div>
                    </template>
                    <template v-if="request.status === 'approve'">
                      <v-btn small color="#02BC77"  disabled=True>
                        อนุมัติแล้ว
                      </v-btn>
                    </template>
                    <template v-if="request.status === 'reject'">
                      <v-btn small color="grey"  disabled=True>
                        ไม่อนุมัติ
                      </v-btn>
                    </template>
    </div>
  </template>
```

รูปที่ 4.51 ตัวอย่างโค้ดส่วนเทมเพลตของ SearchLeaveView.vue

```
  <script>
  import { useToastStore } from '@/stores/toast'
  import { useUserStore } from '@/stores/user'
  import axios from 'axios'
  import { useRouter } from 'vue-router'
  import TopNavBar from '../../components/TopNavBar.vue'
  import Navbar from '../../components/navbar.vue'
  export default {
    setup() {
      const router = useRouter()
      const toastStore = useToastStore()
      const userStore = useUserStore()
      function navigateToDetail(id) {
        router.push({ name: 'DetailTeacherPage', params: { id } })
      }
      return {
        navigateToDetail,
        toastStore,
        userStore
      }
    },
    components: {
      Navbar,
      TopNavBar
    },
    mounted() {
      this.fetchLeaveRequests()
    },
    computed: {
      sortedAndFilteredLeaveRequests() {
        let filtered = this.leaveRequests.filter((request) => {
          const matchesCourse =
request.course_data.name.toLowerCase().includes(this.search.toLowerCase());
          const matchesLeaveType = this.selectedOption ?
request.leave_request_data.leave_type === this.selectedOption : true;
          const matchesIdStudentSearch = this.idStudentSearch ?
request.student_data.user_data.user_id.includes(this.idStudentSearch) : true;
          const matchesSection = this.sectionSearch
          ? request.course_data.section.toString() === this.sectionSearch
          : true;
          const matchesStatus = this.statusSearch ?
request.status.toLowerCase().includes(this.statusSearch.toLowerCase()) : true;
          const fullName = `${request.student_data.fname}
${request.student_data.lname}`.toLowerCase();
          const searchTerms = this.nameSearch.toLowerCase().split('
').filter(Boolean);
          const matchesName = searchTerms.every(term => fullName.includes(term));
          return matchesCourse && matchesLeaveType && matchesIdStudentSearch &&
matchesName && matchesSection && matchesStatus;
        });

        const statusOrder = {
          'pending': 1,
          'approve': 2,
          'reject': 3
        };

        filtered.sort((a, b) => {
          const orderA = statusOrder[a.status] || 999;
          const orderB = statusOrder[b.status] || 999;

          if (orderA === orderB) {
            return a.id - b.id;
          }

          return orderA - orderB;
        });

        return filtered;
```

```
      },
      filteredRequests() {
        return this.leaveRequests.filter((request) => {
          const matchesCourse = request.course_data.name
            .toLowerCase()
            .includes(this.search.toLowerCase())
          const matchesLeaveType = this.selectedOption
            ? request.leave_request_data.leave_type === this.selectedOption
            : true
          const matchesIdStudentSearch =
request.student_data.user_data.user_id.includes(this.idStudentSearch);
          const fullName = `${request.student_data.fname}
${request.student_data.lname}`.toLowerCase();
          const searchTerms = this.nameSearch.toLowerCase().split('
').filter(Boolean);
          const matchesName = searchTerms.every(term => fullName.includes(term));
          return matchesCourse && matchesLeaveType && matchesIdStudentSearch &&
matchesName
        })
      },
      sortedLeaveRequestsByStatus() {
        const statusOrder = {
          'pending': 1,
          'approve': 2,
          'reject': 3
        };

        return this.leaveRequests.slice().sort((a, b) => {
          const orderA = statusOrder[a.status] || 999;
          const orderB = statusOrder[b.status] || 999;
          if (orderA === orderB) {
            return a.id - b.id;
          }
          return orderA - orderB;
        });
      }
    },
    data() {
      return {
        leaveRequests: [],
        loading: false,
        headers: [
          { text: 'ID', align: 'start', value: 'id', sortable: true },
          { text: 'Course Name', value: 'course_data.name' },
          { text: 'Course Section', value: 'course_data.section' },
          { text: 'Leave Type', value: 'leave_request_data.leave_type' },
          { text: 'Start Date', value: 'leave_request_data.start_date' },
          { text: 'End Date', value: 'leave_request_data.end_date' },
          // { text: 'Status', value: 'leave_request_data.status' },
          { text: 'Actions', value: 'actions' }
        ],
        // getStudentLogin
        testId: '',
        testeacherId: '',
        teacher: [],
        user: [],
        // for Search
        search: '',
        selectedOption: '',
        statusSearch: '',
        nameSearch: '',
        idStudentSearch: '',
        sectionSearch: '',
      }
    },
    methods: {
      async getTeacherLogin() {
        this.userStore.initStore()
        this.user = this.userStore.user
```

```
        this.testId = this.userStore.user.id
        if (this.testId) {
          const url = `/teacher/?user_id=${this.testId}`
          await axios
            .get(url)
            .then((response) => {
              this.teacher = response.data[0]
              this.testeacherId = this.teacher.id
            })
            .catch((error) => {
              console.log('error', error)
            })
        } else {
          this.student = []
        }
      },
      getUniqueSubjects(items) {
        const unique = {}
        items.forEach((item) => {
          const uniqueKey = `${item.course_data.name}-${item.leave_request_id}`
          if (!unique[uniqueKey]) {
            unique[uniqueKey] = item
          }
        })
        return Object.values(unique)
      },
      async fetchLeaveRequests() {
        this.loading = true
        await this.getTeacherLogin()
        console.log('testeacher ID: ' + this.testeacherId)
        try {
          if (this.testeacherId) {
            const response = await axios.get(
              `/leaveDetail/?teacher_id=${this.testeacherId}`
            )
            this.leaveRequests = response.data
            this.leaveRequests = this.sortLeaveRequestsByStatus(data);
            this.leaveRequests = this.getUniqueSubjects(response.data)

          } else {
            console.log("Doesn't have Teacher ID:  " + this.testeacherId)
          }
        } catch (error) {
          console.error('There was an error fetching the leave requests:', error)
        } finally {
          this.loading = false
        }
      },
      async onApprove(request) {
        const updatedData = {
          status: 'approve'
        };
        try {
          const response = await
axios.put(`/leaveDetail/update_multiple/?course_id=${request.course_id}&leave_reque
st_id=${request.leave_request_id}`, updatedData);
          console.log('Data approved successfully:', response.data);
          await this.approveBy(request);
          await this.fetchLeaveRequests();
        } catch (error) {
          console.error('Error approving data:', error);
        }
      },
      async approveBy(request){
        const newApproveIdBy = this.testeacherId;
        const updatedData = {
          approve_id_by: newApproveIdBy
        };
        try {
```

```
          const response = await
axios.put(`/leaveDetail/update_multiple/?course_id=${request.course_id}&leave_reque
st_id=${request.leave_request_id}`, updatedData);
          console.log('Data approved successfully:', response.data);
        } catch (error) {
          console.error('Error approving data:', error);
        }
      },
      async onReject(request) {
        const updatedData = {
          status: 'reject'
        };
        try {
          const response = await
axios.put(`/leaveDetail/update_multiple/?course_id=${request.course_id}&leave_reque
st_id=${request.leave_request_id}`, updatedData);
          console.log('Data approved successfully:', response.data);
          await this.approveBy(request);
          await this.fetchLeaveRequests();
        } catch (error) {
          console.error('Error approving data:', error);
        }
      },
    }
  }
  </script>
```

รูปที่ 4.52 ตัวอย่างโค้ดส่วนสคริปต์ของ SearchLeaveView.vue

การพัฒนาระบบ Backend

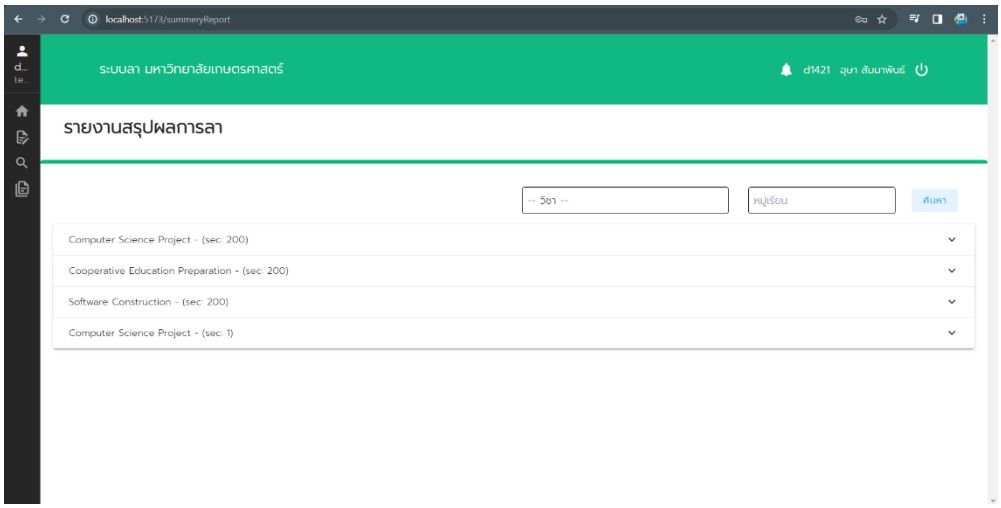
```
class LeaveRequestDetailViewSet(viewsets.ModelViewSet):
    queryset = LeaveRequestDetail.objects.all()
    serializer_class = LeaveRequestDetailSerializer


    def get_queryset(self):
        queryset = super().get_queryset()
        leave_request_id = self.request.query_params.get('leave_request_id')
        student_id = self.request.query_params.get('student_id')
        teacher_id = self.request.query_params.get('teacher_id')
        course_id = self.request.query_params.get('course_id')
        status = self.request.query_params.get('status')


        if leave_request_id:
            queryset = queryset.filter(leave_request_id_by=leave_request_id)

        if teacher_id:
            queryset = queryset.filter(teacher_id=teacher_id)
        if course_id:
            queryset = queryset.filter(course_id=course_id)
        if student_id:
            queryset = queryset.filter(student_id=student_id)
        if status:
            queryset = queryset.filter(status__icontains=status)

        return queryset
```

```
 @action(detail=False, methods=['put'])
    def update_multiple(self, request):
        course_id = request.query_params.get('course_id')
        leave_request_id = request.query_params.get('leave_request_id')
        status = request.data.get('status')

        approve_id_by = request.data.get('approve_id_by')

        if course_id and leave_request_id and status:
            updated_count = LeaveRequestDetail.objects.filter(
                course_id=course_id,
                leave_request_id=leave_request_id
            ).update(status=status)  # Direct update without serialization

            return Response({'updated_records': updated_count})

        if course_id and leave_request_id and approve_id_by:
            updated_count = LeaveRequestDetail.objects.filter(
                course_id=course_id,
                leave_request_id=leave_request_id
            ).update(approve_id_by=approve_id_by)  # Update approve_id_by field
directly

            return Response({'updated_records': updated_count})
        else:
            return HttpResponseBadRequest('Invalid parameters provided')
```

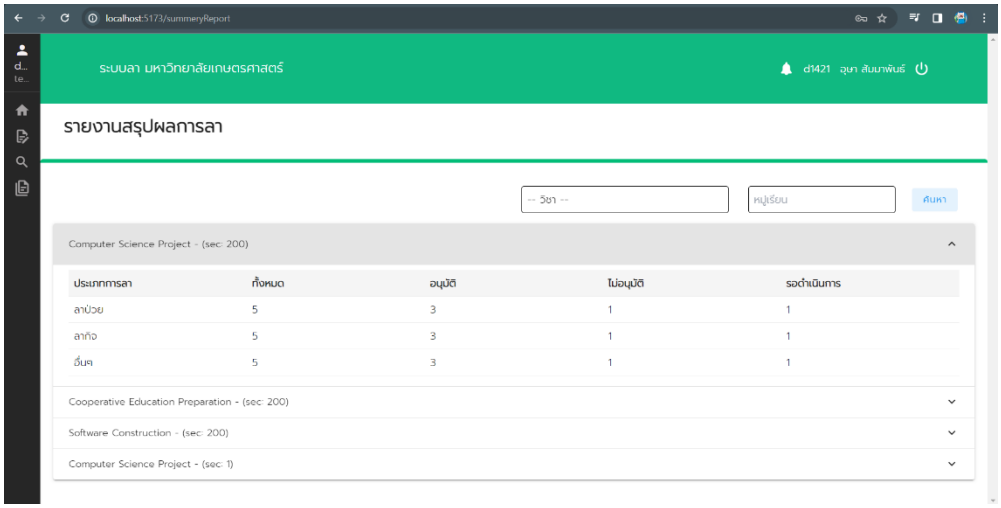
รูปที่ 4.53 ตัวอย่างโค้ดส่วน LeaveRequestDetailViewSet

4.7 รายงานสรุปการลา

ระบบย่อยรายงานสรุปการลาคือระบบสรุปข้อมูลการลาตามระยะเวลาหรือวิชา นำมาสร้างรายงานเพื่อให้อาจารย์สามารถดูคำขอลาทั้งหมดได้



รูปที่ 4.54 โปรแกรมส่วนแสดงรายงานสรุปผลการลาของอาจารย์ตอนที่ยังไม่กดขยายแถบวิชา



รูปที่ 4.55 โปรแกรมส่วนแสดงรายงานสรุปผลการลาของอาจารย์ตอนกดขยายแถบวิชา

4.7.1 สรุปการลา

สามารถสรุปการลาแต่ละรายวิชาของอาจารย์ที่เข้าสู่ระบบมา โดยเรียกตามรายวิชาและมีการจำแนกตามประเภทการลา ซึ่งมีการสรุปดังนี้ ใบลาทั้งหมดที่ส่งเข้ามา ใบลาที่อนุมัติในรายวิชานี้ ใบลาที่ไม่อนุมัติในรายวิชานี้ และ ใบลาที่รอดำเนินการอยู่ โดยสามารถค้นหาสรุปการลาโดยใช้ชื่อวิชาและหมู่เรียนในการค้นหาได้

การพัฒนาระบบ Frontend

```
<template>
      <div class="content-Page">
        <slot></slot>
        <div class="head">รายงานสรุปผลการลา</div>
      </div>
      <div class="content">
        <v-row>
          <v-col cols="12" md="3"></v-col>
          <v-col cols="12" md="3"></v-col>
          <v-col cols="12" md="3">
            <div class="field-container">
              <div class="label-input-pair">
                <select v-model="selectedSubject">
                  <option value="" disabled selected>-- วิชา --</option>
                  <option value="" >-</option>
                  <option v-for="item in subjects" :value="item.course_data.name"
:key="item.id">{{ item.course_data.name }}</option>
                </select>
              </div>
            </div>
          </v-col>
          <v-col cols="12" md="2">
            <div class="field-container">
              <div class="label-input-pair">
                <input type="text" placeholder="หมู่เรียน" v-model="sectionSearch" />
              </div>
            </div>
          </v-col>
          <v-col cols="12" md="1">
```

```
            <v-btn

              color="blue"
              variant="tonal"
              @click="findBysection(this.sectionSearch)"
            >
              ค้นหา
            </v-btn>
          </v-col>
        </v-row>
        <v-expansion-panels variant="accordion">
          <v-expansion-panel v-for="(summary, index) in leaveRequests"
:key="index" :title="`${summary.course_name} - (sec: ${summary.section})`">
            <v-expansion-panel-content>
              <v-expansion-panel-text>
                <table>
                  <thead>
                    <tr>
                      <th>ประเภทการลา</th>
                      <th>ทั้งหมด</th>
                      <th>อนุมัติ</th>
                      <th>ไม่อนุมัติ</th>
                      <th>รอดำเนินการ</th>
                    </tr>
                  </thead>
                  <tbody>
                    <!-- ลาป่วย -->
                    <tr>
                      <td>ลาป่วย</td>
                      <td>{{ summary.total }}</td>
                      <td>{{ summary.approve }}</td>
                      <td>{{ summary.reject }}</td>
                      <td>{{ summary.pending }}</td>
                    </tr>
                    <!-- ลากิจ -->
                    <tr>
                      <td>ลากิจ</td>
                      <td>{{ summary.total }}</td>
                      <td>{{ summary.approve }}</td>
                      <td>{{ summary.reject }}</td>
                      <td>{{ summary.pending }}</td>
                    </tr>
                    <!-- อื่นๆ -->
                    <tr>
                      <td>อื่นๆ</td>
                      <td>{{ summary.total }}</td>
                      <td>{{ summary.approve }}</td>
                      <td>{{ summary.reject }}</td>
                      <td>{{ summary.pending }}</td>
                    </tr>
                  </tbody>
                </table>
            </v-expansion-panel-text>
            </v-expansion-panel-content>
          </v-expansion-panel>
      </v-expansion-panels>
      </div>
    </div>
  </div>
</template>
```

รูปที่ 4.56 ตัวอย่างโค้ดส่วนเทมเพลตของ SummaryReportView.vue

```
<script>
import Navbar from '../../components/navbar.vue'
import TopNavBar from '../../components/TopNavBar.vue'
import { useToastStore } from '@/stores/toast'
import { useUserStore } from '@/stores/user'
import axios from 'axios'

export default {
  setup() {
    const toastStore = useToastStore()
    const userStore = useUserStore()
    return {
      toastStore,
      userStore
    }
  },
  data() {
    return {
      // getTeacherLogin
      testId: '',
      testeacherId: '',
      teacher: [],
      user: [],
      sectionSearch: '',
      selectedSubject: '',
      // leave Detail
      leaveRequests: [],
      // fetchInstructorCourse
      subjects: [],
    }
  },
  components: {
    Navbar,
    TopNavBar
  },
  beforeCreate() {},
  async mounted() {
    await this.fetchLeaveRequests()
    await this.fetchInstructorCourse()
  },
  created() {

  },
  watch: {
     sectionSearch: {
      async handler() {
        if (this.sectionSearch=== "") {
          await this.fetchLeaveRequests()
        }
      },
      immediate: true
    }
  },
  methods: {
    async getTeacherLogin() {
      this.userStore.initStore()
      this.user = this.userStore.user
      this.testId = this.userStore.user.id
      if (this.testId) {
        const url = `/teacher/?user_id=${this.testId}`
        await axios
          .get(url)
          .then((response) => {
            this.teacher = response.data[0]
            this.testeacherId = this.teacher.id
          })
          .catch((error) => {
            console.log('error', error)
          })
```

```
      } else {
        this.student = []
      }
    },
    async fetchInstructorCourse() {
      this.loading = true
      await this.getTeacherLogin()
      try {
        if (this.testeacherId) {
          const response = await axios.get(
            `/instructorCourse/?teacher_id=${this.testeacherId}`
          )
          let filteredSubjects = response.data.filter((item, index, self) =>
            index === self.findIndex((t) => (
              t.course_data.name === item.course_data.name
            ))
          );
          this.subjects = filteredSubjects;
          console.log(this.subjects)
        } else {
          console.log("Doesn't have Teacher ID:  " + this.testeacherId)
        }
      } catch (error) {
        console.error('There was an error fetching the leave requests:', error)
      } finally {
        this.loading = false
      }
    },
    async fetchLeaveRequests() {
    this.loading = true;
    await this.getTeacherLogin();
    console.log('Teacher ID: ' + this.testeacherId);
    try {
      if (this.testeacherId) {
        const response = await
axios.get(`/leave_request_details/?teacher_id=${this.testeacherId}&section=${this.s
ectionSearch}&course_name=${this.selectedSubject}`);
        const rawData = response.data;
        const summarizedData = this.summarizeLeaveRequests(rawData);
        this.leaveRequests = summarizedData;
        console.log(this.leaveRequests);
      } else {
        console.log("No Teacher ID: " + this.testeacherId);
      }
    } catch (error) {
      console.error('Error fetching the leave requests:', error);
    } finally {
      this.loading = false;
    }
  },

  summarizeLeaveRequests(data) {
    const summary = {};

    data.forEach(request => {
      const { course_data } = request;
      const key = course_data.id + '-' + course_data.name;

      if (!summary[key]) {
        summary[key] = {
          course_id: course_data.id,
          course_name: course_data.name,
          leaveType: request.type,
          total: 0,
          approve: 0,
          reject: 0,
          pending: 0,
          section: course_data.section
        };
```

```
      }

      summary[key].total++;
      if (request.status === 'approve') {
        summary[key].approve++;
      } else if (request.status === 'reject') {
        summary[key].reject++;
      } else if (request.status === 'pending') {
        summary[key].pending++;
      }
    });

    return Object.values(summary);
  },
  async findBysection (sectionSearch) {
    await this.fetchLeaveRequests()

  }
  }
}
</script>
```

รูปที่ 4.57 ตัวอย่างโค้ดส่วนสคริปต์ของ SummaryReport.View.vue

การพัฒนาระบบ Backend

```
import django_filters
from .models import LeaveRequestDetail

class LeaveRequestDetailFilter(django_filters.FilterSet):
    section = django_filters.CharFilter(field_name='course_id__section',
lookup_expr='exact')
    teacher_id = django_filters.NumberFilter(field_name='teacher_id',
lookup_expr='exact')
    course_name = django_filters.CharFilter(field_name='course_id__name',
lookup_expr='icontains')

    class Meta:
        model = LeaveRequestDetail
        fields = ['section', 'teacher_id', 'course_name']
```

รูปที่ 4.58 ตัวอย่างโค้ดส่วน LeaveRequestDetailFilter

```
class LeaveRequestDetailView(generics.ListAPIView):
    queryset = LeaveRequestDetail.objects.all()
    serializer_class = LeaveRequestDetailSerializer
    filter_backends = [DjangoFilterBackend]
    filterset_class = LeaveRequestDetailFilter
```

รูปที่ 4.59 ตัวอย่างโค้ดส่วน LeaveRequestDetailView

```
from django.urls import path, include
from rest_framework.routers import DefaultRouter
from .views import FilesViewSet, LeaveRequestViewSet, LeaveRequestDetailViewSet,
LeaveRequestDetailView

urlpatterns = [
    path('leave_request_details/', LeaveRequestDetailView.as_view(), name='leave-
request-detail'),
]
```

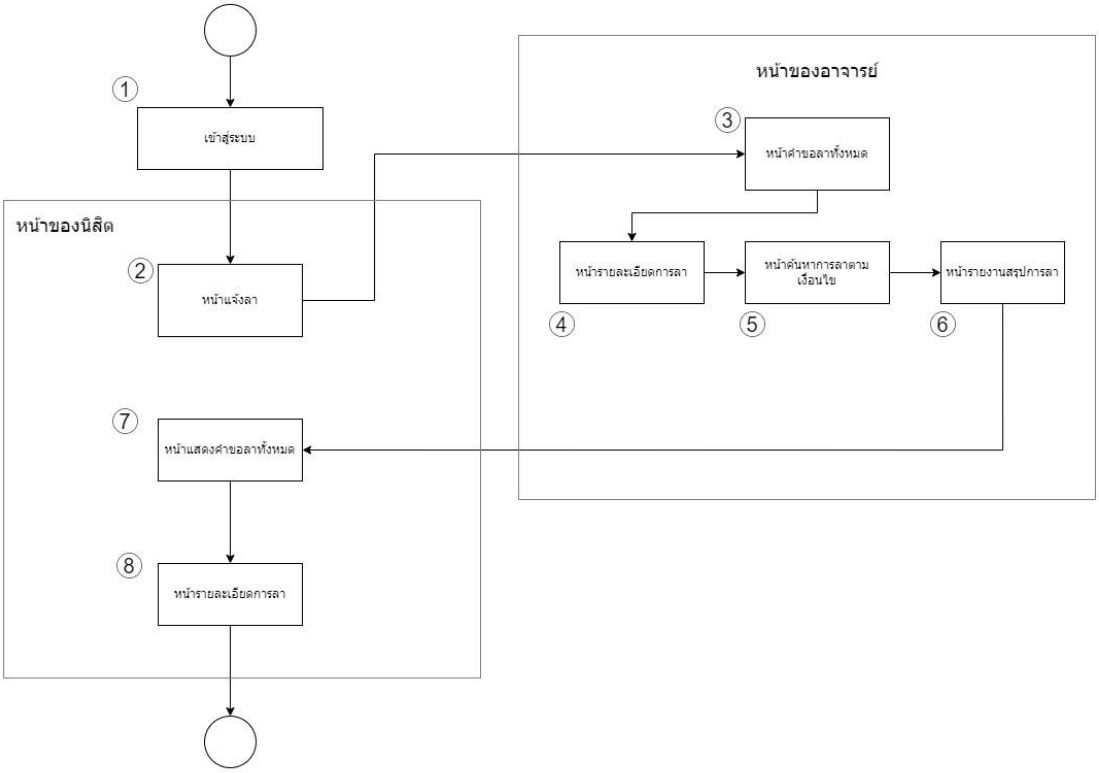
รูปที่ 4.60 ตัวอย่างโค้ดส่วน leave/urls.py

บทที่ 5
การทดสอบระบบ

5.1 การทดสอบระบบรวม

แผนผังการทดสอบระบบโดยรวมช่วยให้วางแผนและดำเนินการทดสอบโปรแกรมอย่างมีระเบียบ

Strategies to test : Top-Down



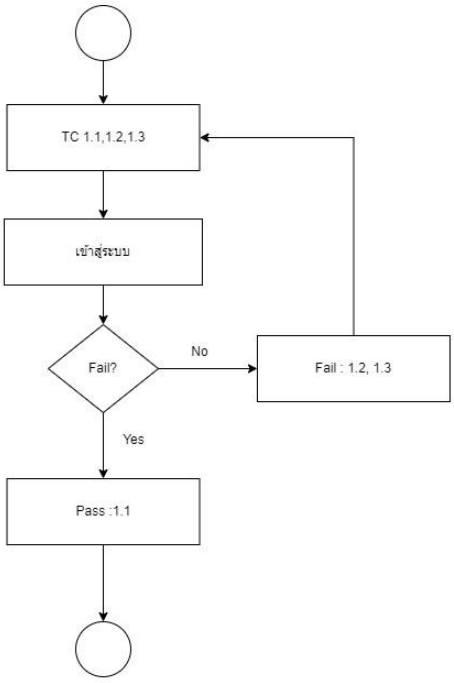
รูปที่ 5.1 แสดงแผนภาพการทดสอบระบบรวม

5.2 การทดสอบระบบย่อย

การทดสอบระบบย่อยแต่ละระบบช่วยให้โปรแกรมมีประสิทธิภาพมากขึ้น ทำให้ผู้ใช้มั่นใจว่าโปรแกรมจะตอบสนองตามผลลัพธ์ที่คาดหวังได้

5.2.1 ทดสอบโปรแกรมส่วนการเข้าสู่ระบบ

การทดสอบการเข้าสู่ระบบ ทดสอบเพื่อตรวจสอบความถูกต้องและปลอดภัยของระบบ ไม่ให้ผู้ที่ไม่ได้มีบัญชีผู้ใช้ระบบการจัดการการลาเข้าสู่ระบบได้



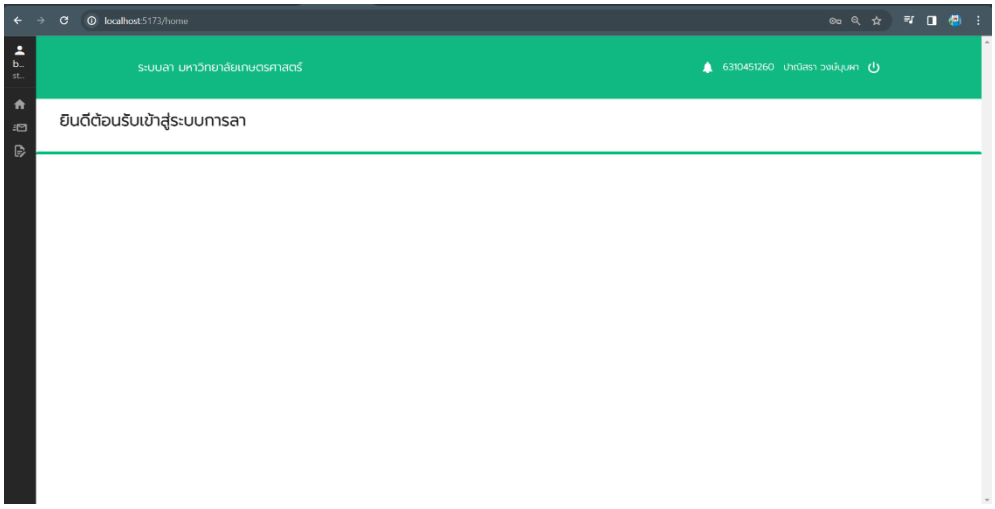
รูปที่ 5.2 แสดงแผนภาพการทดสอบโปรแกรมเข้าสู่ระบบ

ตารางที่ 5.1 แสดงกรณีทดสอบที่ 1

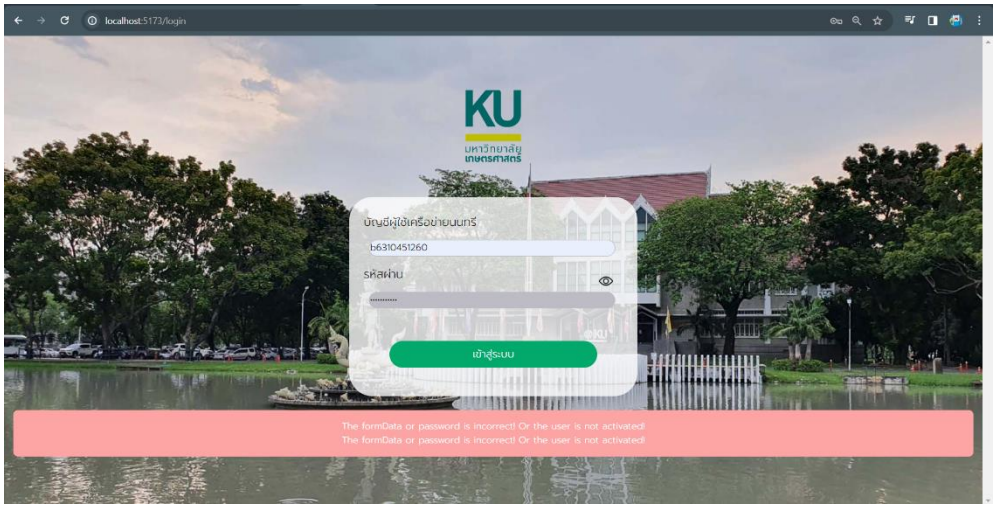
TC 1

กรณีทดสอบ	เงื่อนไขก่อนการทดสอบ	ผลการทดสอบ	หมายเหตุ
TC 1.1 การเข้าสู่ระบบด้วยชื่อผู้ใช้และรหัสผ่านที่ถูกต้อง	มีบัญชีผู้ใช้งานที่ถูกต้องในระบบ	ผ่าน	-
TC 1.2 การเข้าสู่ระบบด้วยชื่อผู้ใช้ที่ถูกต้องแต่รหัสผ่านไม่ถูกต้องจะเข้าสู่ระบบไม่สำเร็จ	มีบัญชีผู้ใช้งานที่ถูกต้องในระบบ	ผ่าน	-

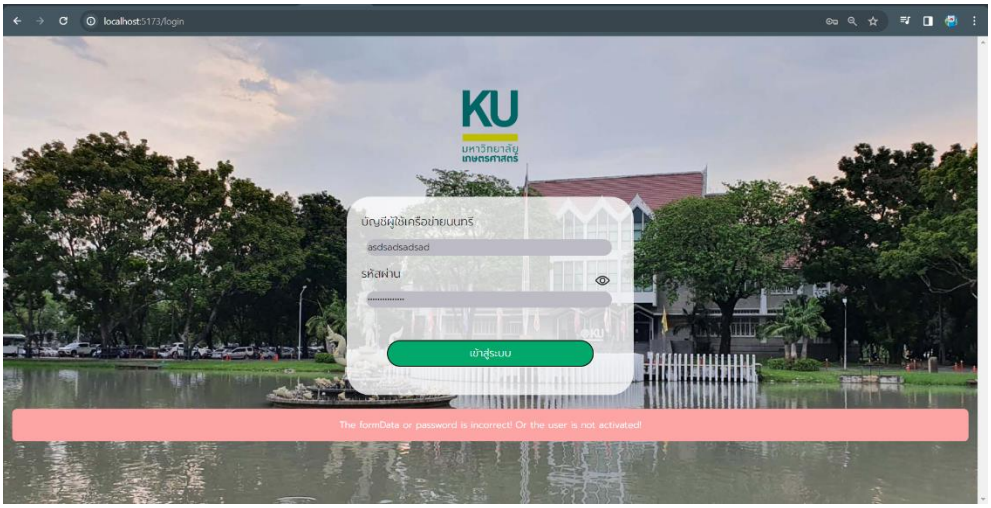
TC 1			
กรณีทดสอบ	เงื่อนไขก่อนการทดสอบ	ผลการทดสอบ	หมายเหตุ
TC 1.3 การเข้าสู่ระบบด้วยชื่อผู้ใช้ที่ไม่ถูกต้องจะเข้าสู่ระบบไม่สำเร็จ	ไม่มีบัญชีผู้ใช้งานในระบบ	ผ่าน	-



รูปที่ 5.3 แสดงภาพประกอบกรณีทดสอบ TC 1.1

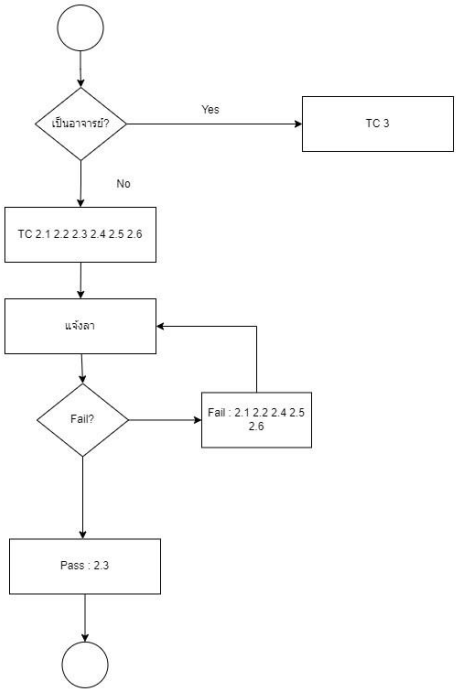


รูปที่ 5.4 แสดงภาพประกอบกรณีทดสอบ TC 1.2



รูปที่ 5.5 แสดงภาพประกอบกรณีทดสอบ TC 1.3

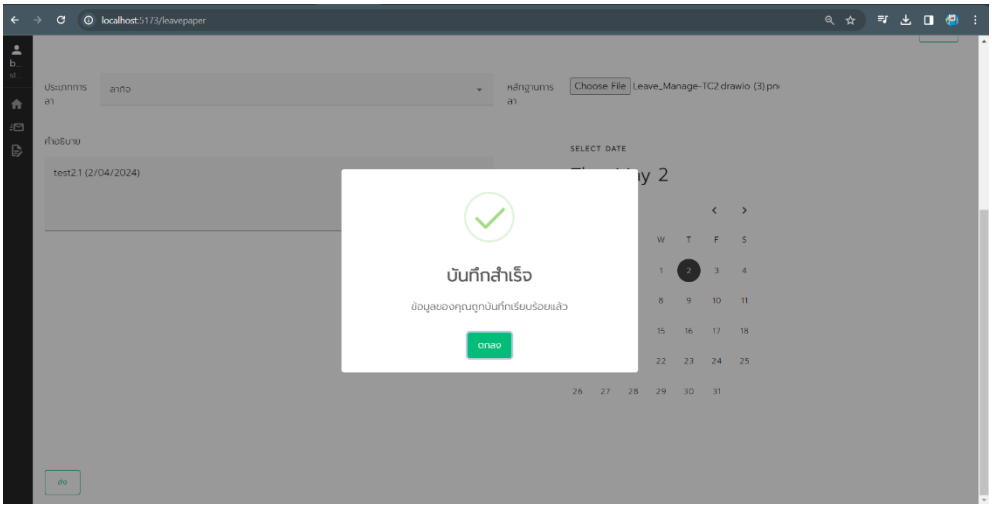
5.2.2 ทดสอบโปรแกรมส่วนการแจ้งลา



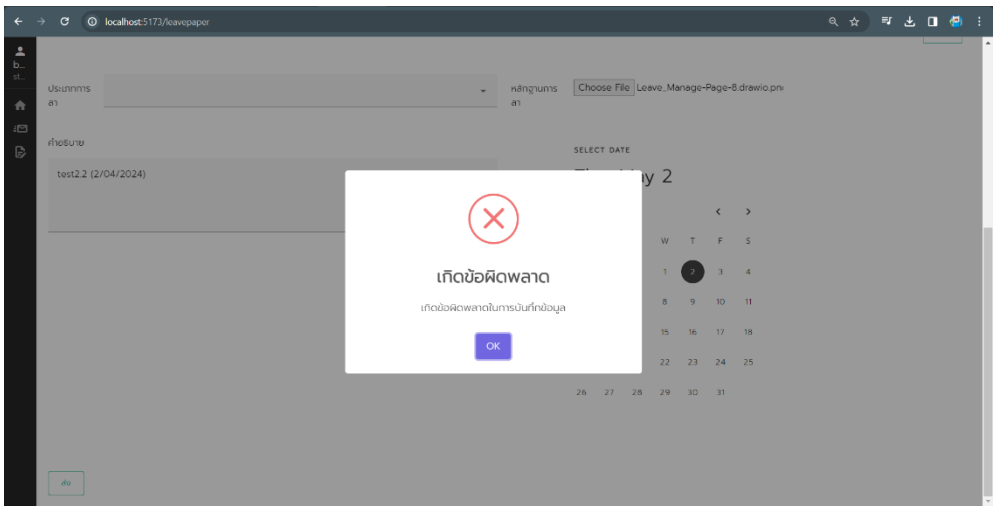
รูปที่ 5.6 แสดงแผนภาพการทดสอบโปรแกรมการแจ้งลา

ตารางที่ 5.2 **แสดงกรณีทดสอบที่** 2

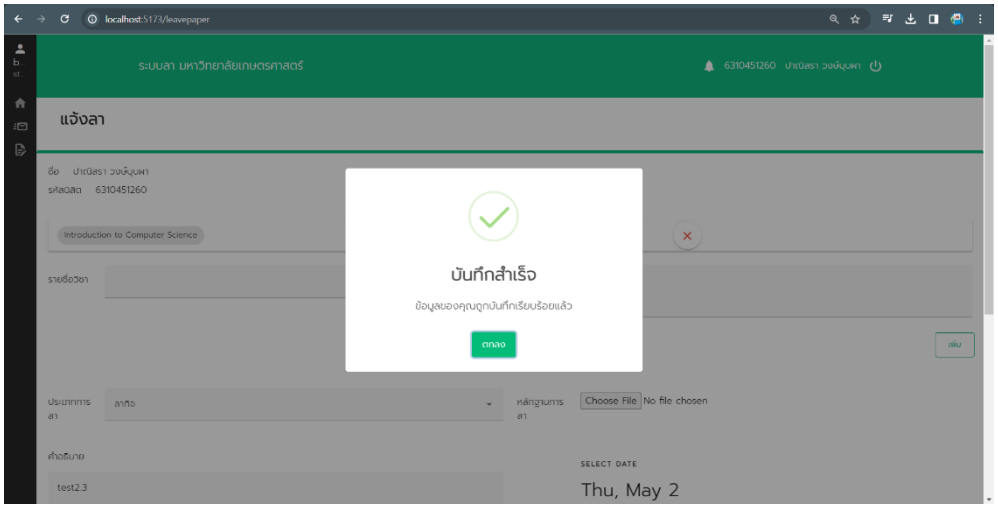
TC 2			
กรณีทดสอบ	เงื่อนไขก่อนการทดสอบ	ผลการทดสอบ	หมายเหตุ
TC 2.1 ไม่เลือกวิชา ไม่เลือกอาจารย์ แล้วกดส่งจะไม่สามารถส่งใบลาได้	-	ผ่าน	- ระบบคืนค่าข้อความผิดพลาด - ไม่สามารถเพิ่มใบลาได้
TC 2.2 ไม่เลือกประเภทการลาแล้วกดส่งจะไม่สามารถส่งใบลาได้	-	ไม่ผ่าน	ก่อนระบบคืนค่าข้อความผิดพลาด ระบบคืนค่าว่าส่งข้อมูลสำเร็จ
TC 2.3 ไม่เพิ่มไฟล์หลักฐานการลาแล้วกดส่งสามารถส่งใบลาได้	-	ผ่าน	- ผลลัพธ์เป็นไปตามที่คาดหวัง - สามารถเพิ่มใบลาได้
TC2.4 ไม่ใส่วันที่แล้วกดส่งจะไม่สามารถส่งใบลาได้	-	ผ่าน	ระบบคืนค่าข้อความ อื่นแทนค่า“กรุณาระบุวันที่”
TC 2.5 เลือกวิชาแล้วกดปุ่มเพิ่มจะไม่สามารถกดเพิ่มได้	-	ผ่าน	ไม่สามารถเพิ่มวิชาได้
TC 2.6 ส่งวิชาซ้ำแล้วกดปุ่มเพิ่มจะไม่สามารถกดเพิ่มได้	-	ผ่าน	จะไม่สามารถกดเพิ่มได้



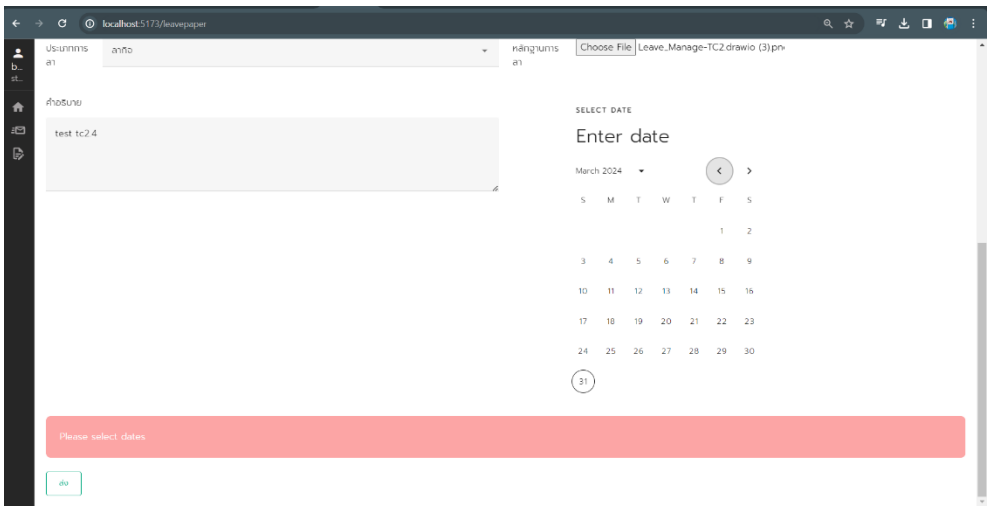
รูปที่ 5.7 แสดงภาพประกอบกรณีทดสอบ TC 2.1



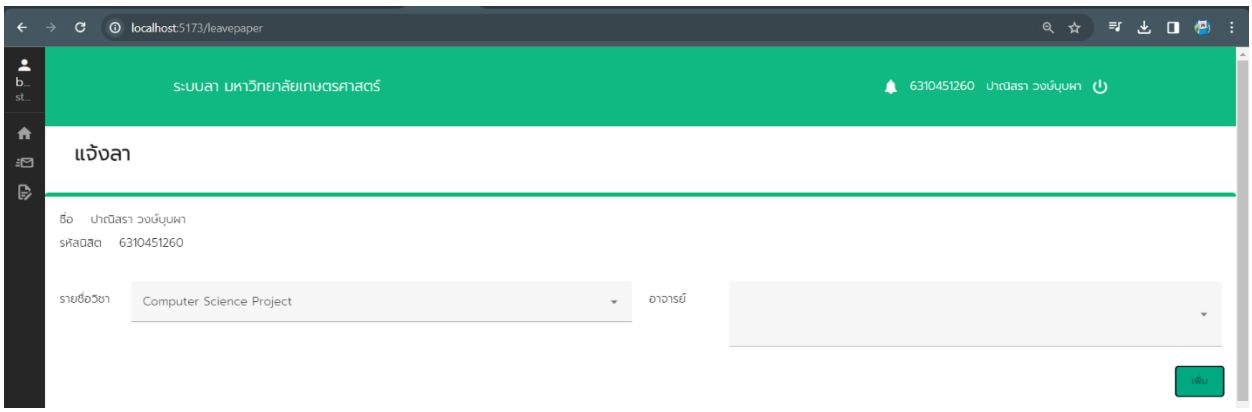
รูปที่ 5.8 แสดงภาพประกอบกรณีทดสอบ TC 2.2



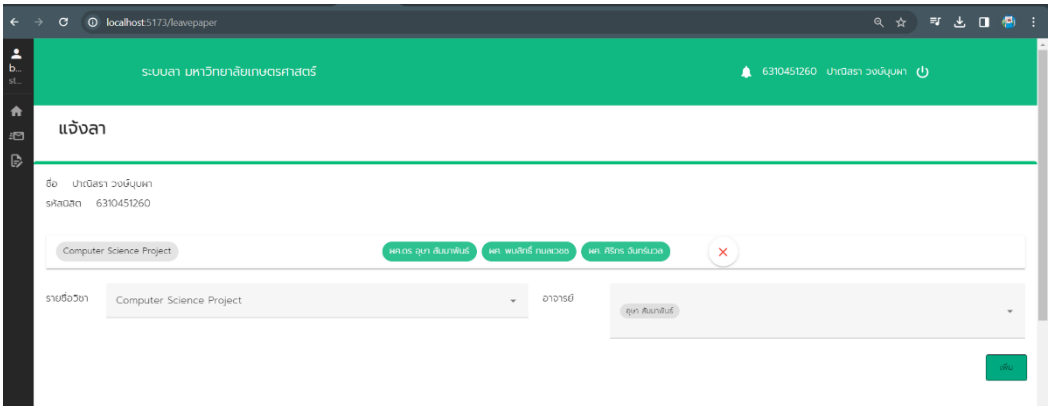
รูปที่ 5.9 แสดงภาพประกอบกรณีทดสอบ TC 2.3



รูปที่ 5.10 แสดงภาพประกอบกรณีทดสอบ TC 2.4

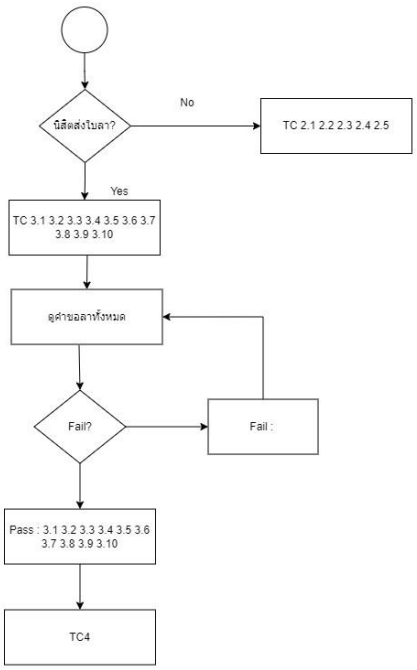


รูปที่ 5.11 แสดงภาพประกอบกรณีทดสอบ TC 2.5



รูปที่ 5.12 แสดงภาพประกอบกรณีทดสอบ TC 2.6

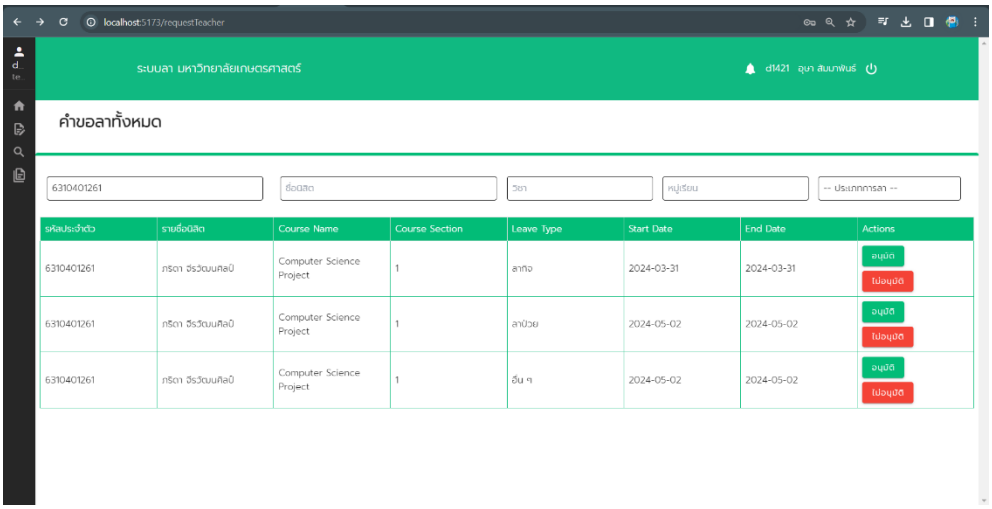
5.2.3 ทดสอบโปรแกรมส่วนคำขอลาทั้งหมดของนิสิตที่ส่งให้อาจารย์



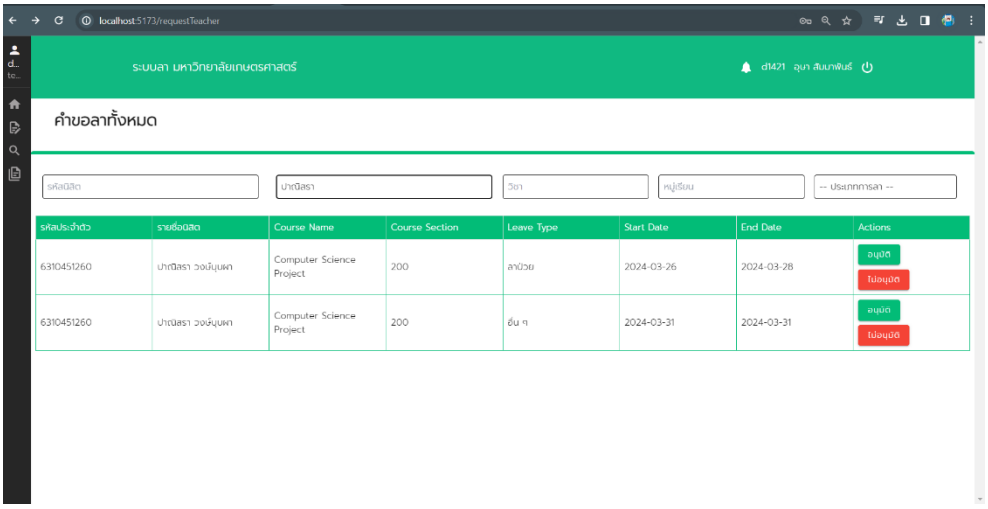
รูปที่ 5.13 แสดงแผนภาพการทดสอบโปรแกรมคำขอลาทั้งหมดของนิสิตที่ส่งให้อาจารย์

ตารางที่ 5.3 แสดงกรณีทดสอบที่ 3

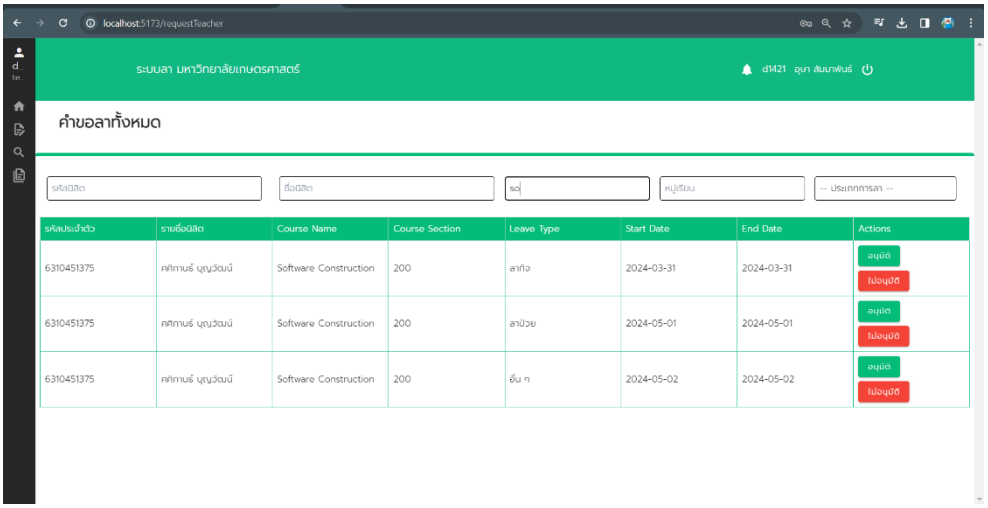
TC 3			
กรณีทดสอบ	เงื่อนไขก่อนการทดสอบ	ผลการทดสอบ	หมายเหตุ
TC 3.1 ค้นหาด้วยรหัสนิสิต	นิสิตส่งใบลา	ผ่าน	-
TC 3.2 ค้นหาด้วยชื่อนิสิต	นิสิตส่งใบลา	ผ่าน	-
TC 3.3 ค้นหาด้วยชื่อวิชา	นิสิตส่งใบลา	ผ่าน	-
TC 3.4 ค้นหาด้วยหมู่เรียน	นิสิตส่งใบลา	ผ่าน	-
TC 3.5 ค้นหาด้วยประเภทการลา “ลากิจ”	นิสิตส่งใบลาที่มีประเภทการลาเป็น ลากิจ	ผ่าน	-
TC 3.6 ค้นหาด้วยประเภทการลา “ลาป่วย”	นิสิตส่งใบลาที่มีประเภทการลาเป็น ลาป่วย	ผ่าน	-
TC 3.7 ค้นหาด้วยประเภทการลา “อื่น ๆ”	นิสิตส่งใบลาที่มีประเภทการลาเป็น อื่นๆ	ไม่ผ่าน	ใบลาที่มีประเภทการลาเป็นอื่นๆ ไม่สามารถค้นหาได้
TC 3.8 กดอนุมัติได้	นิสิตส่งใบลา	ผ่าน	-
TC 3.9 กดไม่อนุมัติได้	นิสิตส่งใบลา	ผ่าน	-
TC 3.10 แสดงใบรอทุกใบของนิสิตที่รออาจารย์อนุมัติ/ไม่อนุมัติ	นิสิตส่งใบลา	ผ่าน	-



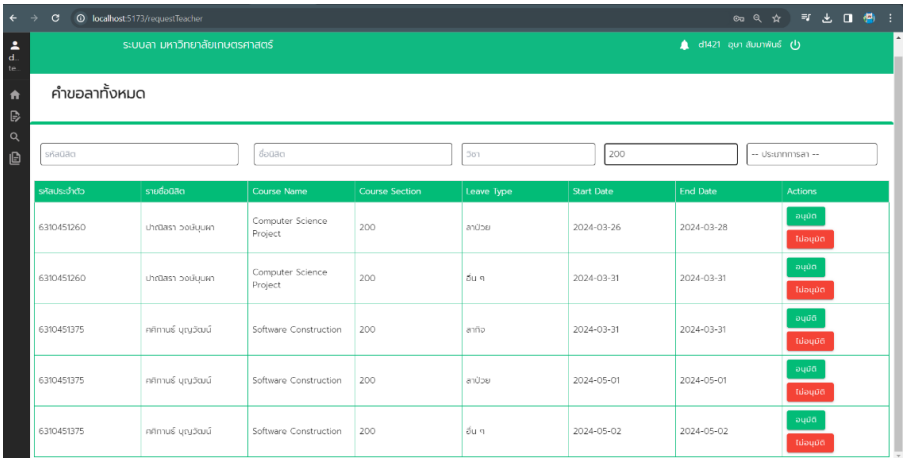
รูปที่ 5.14 แสดงภาพประกอบกรณีทดสอบ TC 3.1



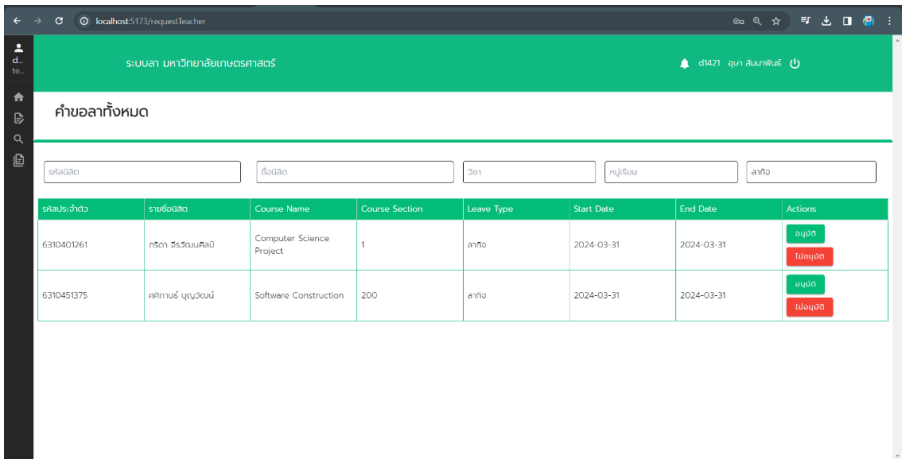
รูปที่ 5.15 แสดงภาพประกอบกรณีทดสอบ TC 3.2



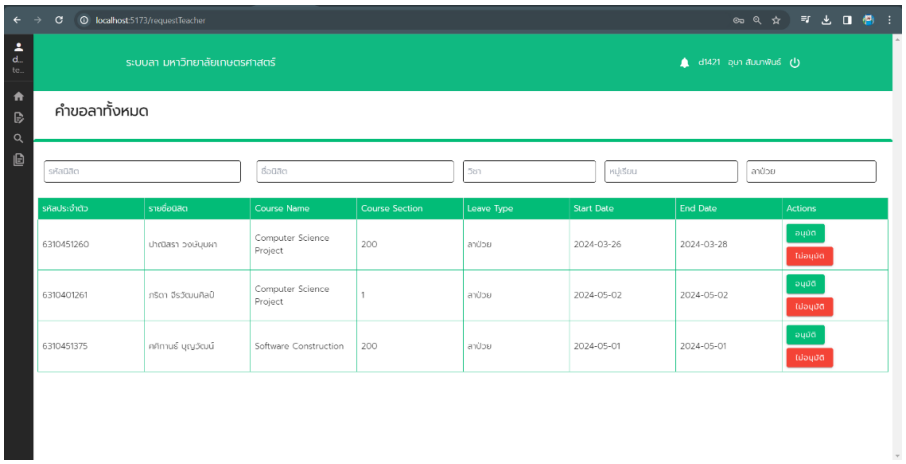
รูปที่ 5.16 แสดงภาพประกอบกรณีทดสอบ TC 3.3



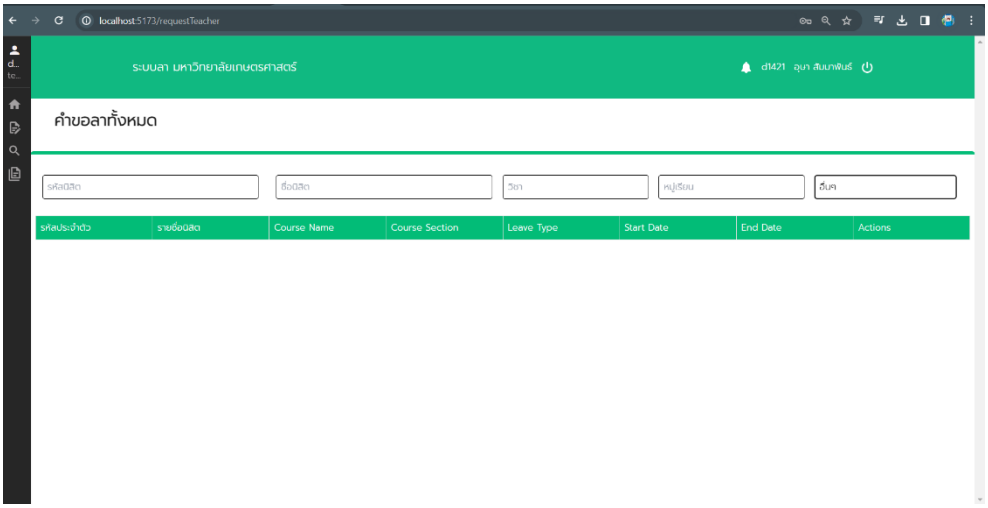
รูปที่ 5.17 แสดงภาพประกอบกรณีทดสอบ TC 3.4



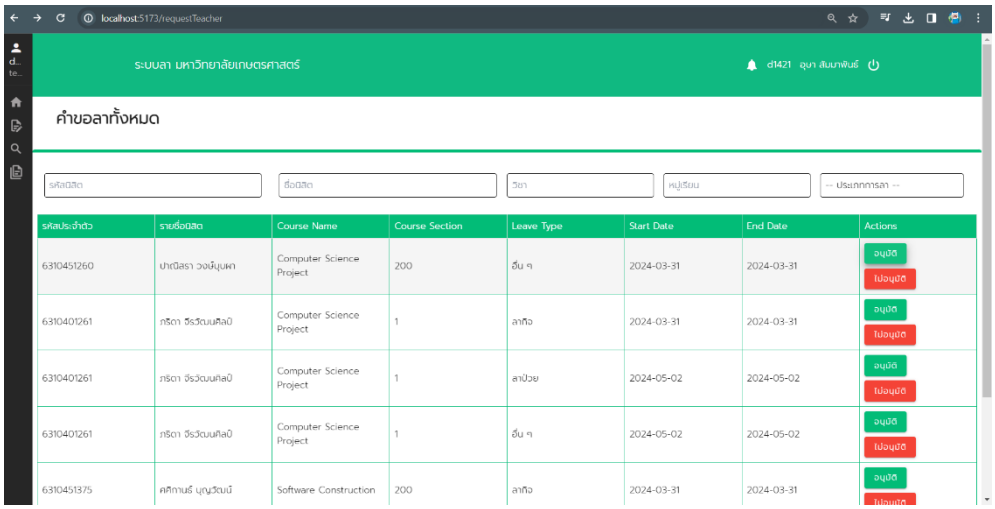
รูปที่ 5.18 แสดงภาพประกอบกรณีทดสอบ TC 3.5



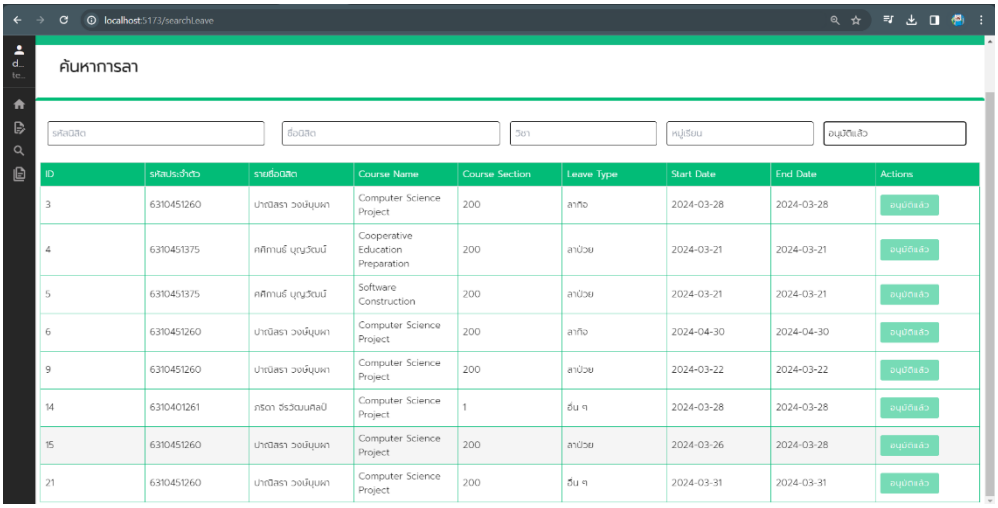
รูปที่ 5.19 แสดงภาพประกอบกรณีทดสอบ TC 3.6



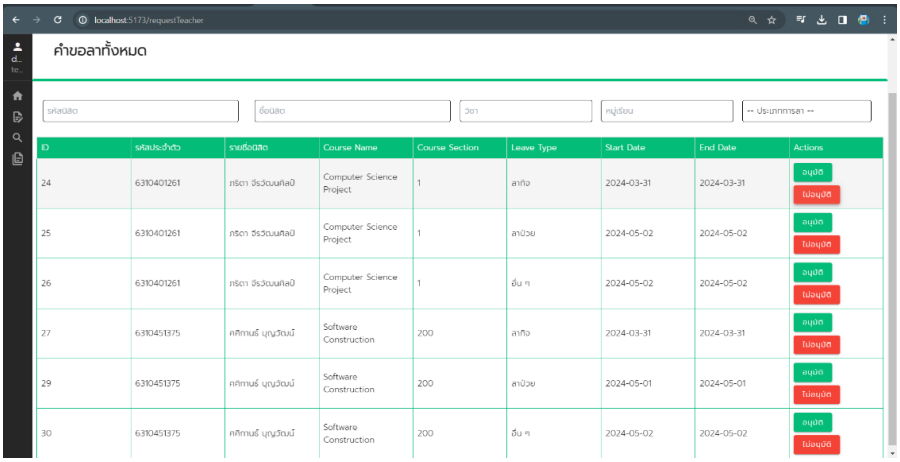
รูปที่ 5.20 แสดงภาพประกอบกรณีทดสอบ TC 3.7



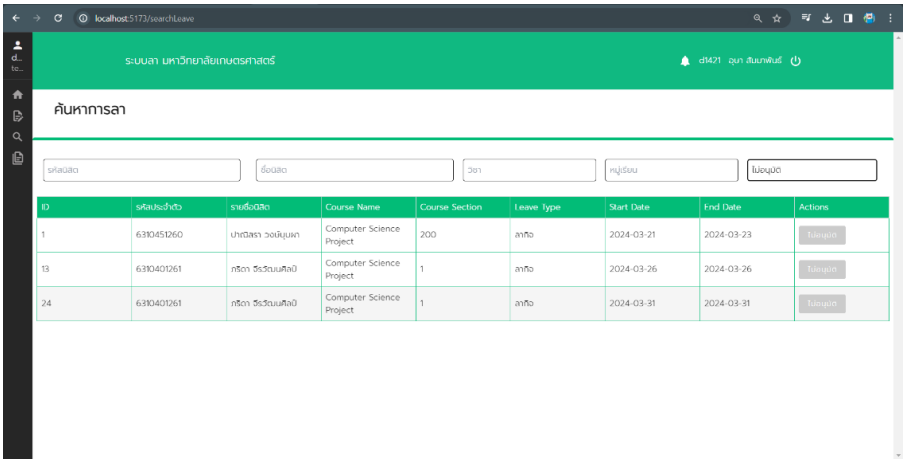
รูปที่ 5.21 แสดงภาพประกอบกรณีทดสอบ TC 3.8



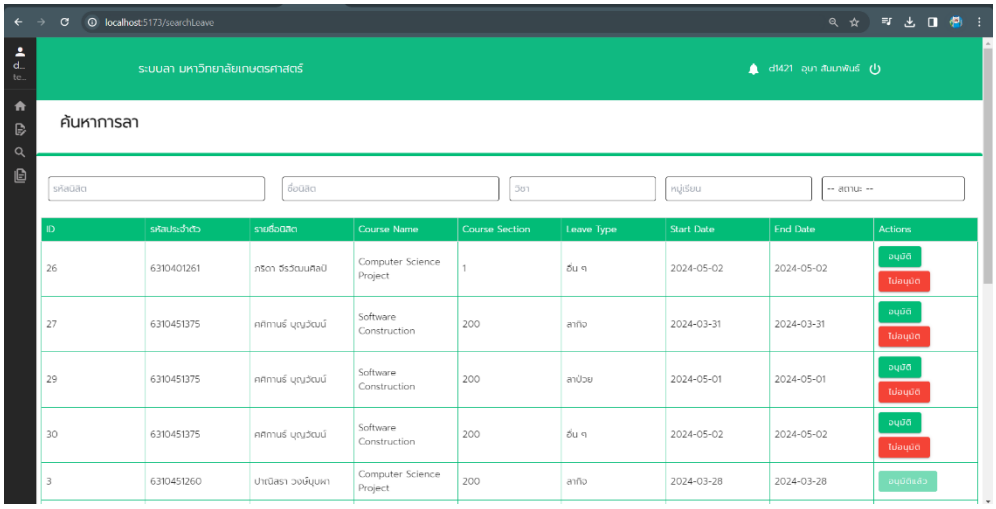
รูปที่ 5.22 แสดงภาพประกอบกรณีทดสอบ TC 3.8



รูปที่ 5.23 แสดงภาพประกอบกรณีทดสอบ TC 3.9

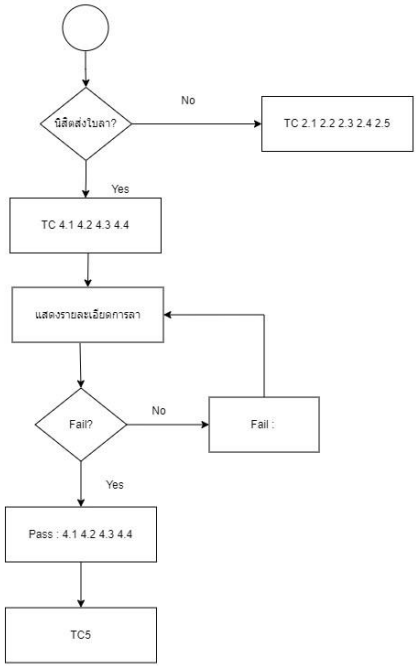


รูปที่ 5.24 แสดงภาพประกอบกรณีทดสอบ TC 3.9



รูปที่ 5.25 แสดงภาพประกอบกรณีทดสอบ TC 3.10

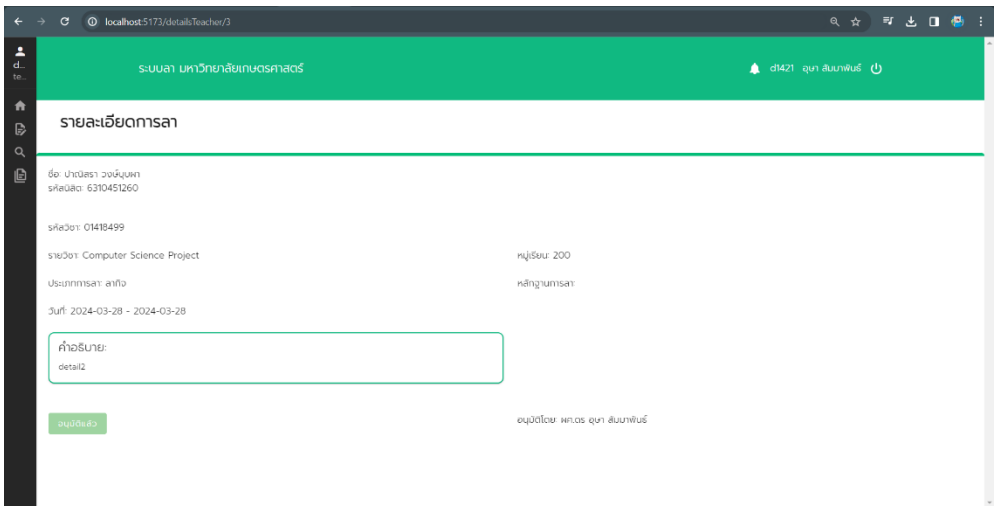
5.2.4 ทดสอบโปรแกรมส่วนรายละเอียดการลาของนิสิตที่ส่งให้อาจารย์



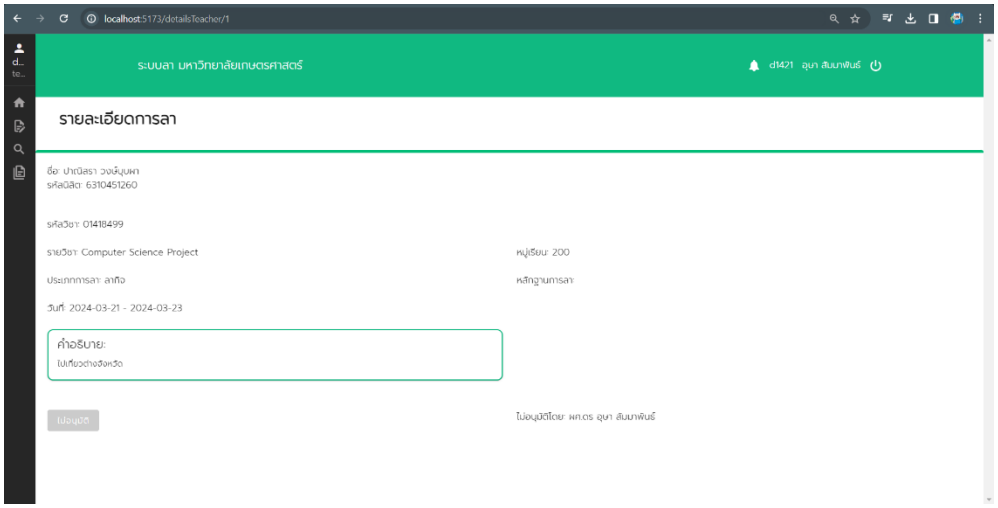
รูปที่ 5.26 แสดงแผนภาพการทดสอบโปรแกรมส่วนรายละเอียดการลาของนิสิตที่ส่งให้อาจารย์

ตารางที่ 5.4 แสดงกรณีทดสอบที่ 4

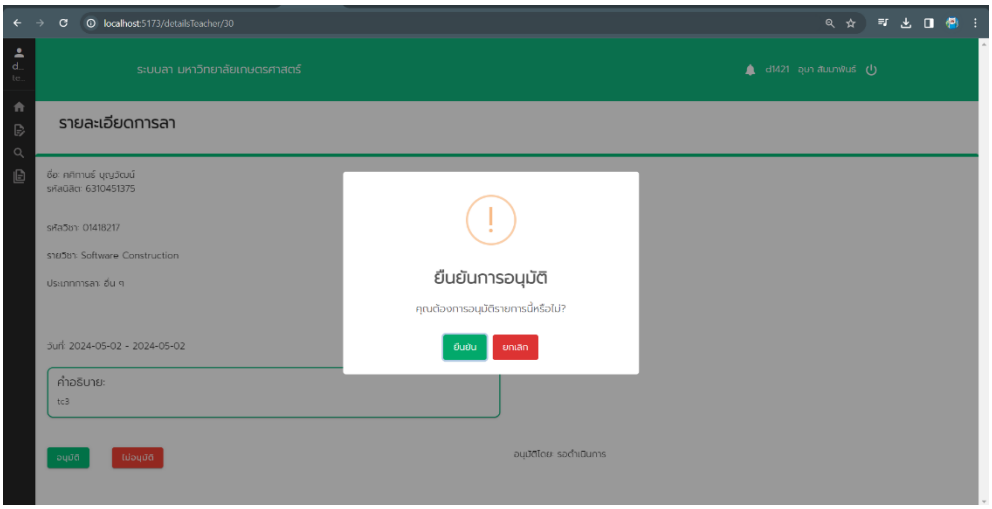
TC 4			
กรณีทดสอบ	เงื่อนไขก่อนการทดสอบ	ผลการทดสอบ	หมายเหตุ
TC 4.1 แสดงรายละเอียดการลาของนิสิตครบถ้วน พร้อมแสดงสถานะ “อนุมัติ”	อาจารย์กดอนุมัติใบลาที่จะแสดงรายละเอียด	ผ่าน	-
TC 4.2 แสดงรายละเอียดการลาของนิสิตครบถ้วน พร้อมแสดงสถานะ “ไม่อนุมัติ”	อาจารย์กดไม่อนุมัติใบลาที่จะแสดงรายละเอียด	ผ่าน	-
TC 4.3 แสดงรายละเอียดการลาของนิสิตครบถ้วน พร้อมแสดงสถานะ “รอดำเนินการ” แล้วสามารถกด “อนุมัติ” ได้	นิสิตส่งใบลา	ผ่าน	-
TC 4.4 แสดงรายละเอียดการลาของนิสิตครบถ้วน พร้อมแสดงสถานะ “รอดำเนินการ” แล้วสามารถกด “ไม่อนุมัติ” ได้	นิสิตส่งใบลา	ผ่าน	-



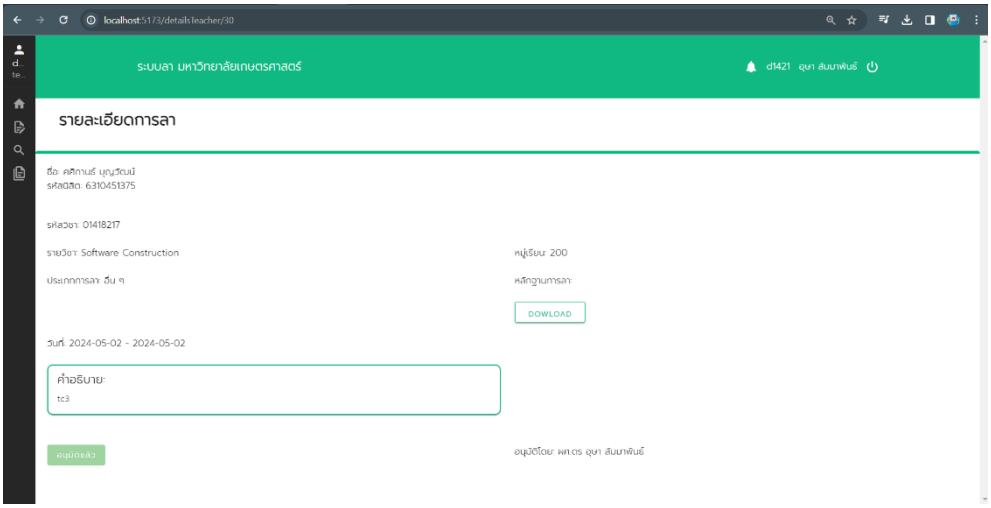
รูปที่ 5.27 แสดงภาพประกอบกรณีทดสอบ TC 4.1



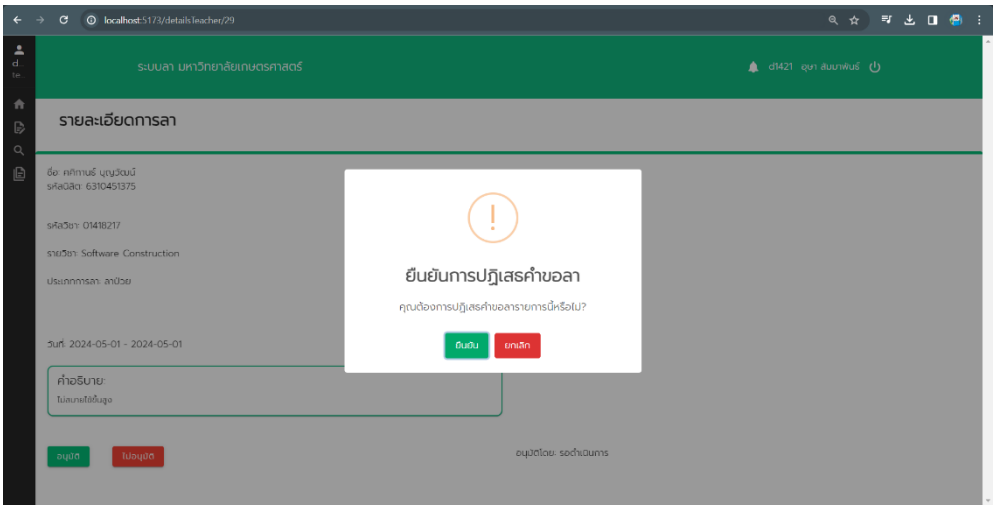
รูปที่ 5.28 แสดงภาพประกอบกรณีทดสอบ TC 4.2



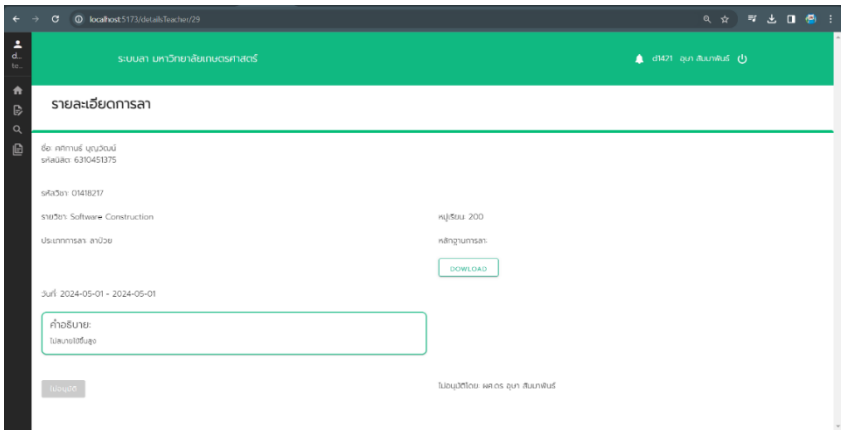
รูปที่ 5.29 แสดงภาพประกอบกรณีทดสอบ TC 4.3



รูปที่ 5.30 แสดงภาพประกอบกรณีทดสอบ TC 4.3

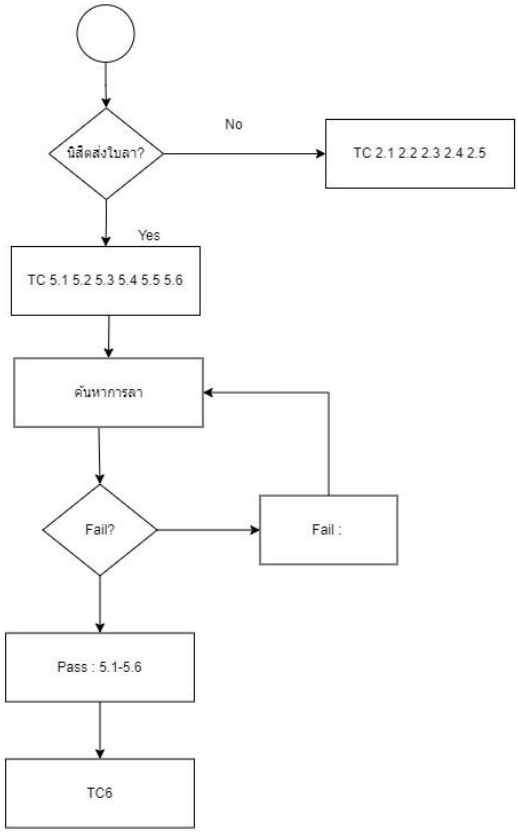


รูปที่ 5.31 แสดงภาพประกอบกรณีทดสอบ TC 4.4



รูปที่ 5.32 แสดงภาพประกอบกรณีทดสอบ TC 4.4

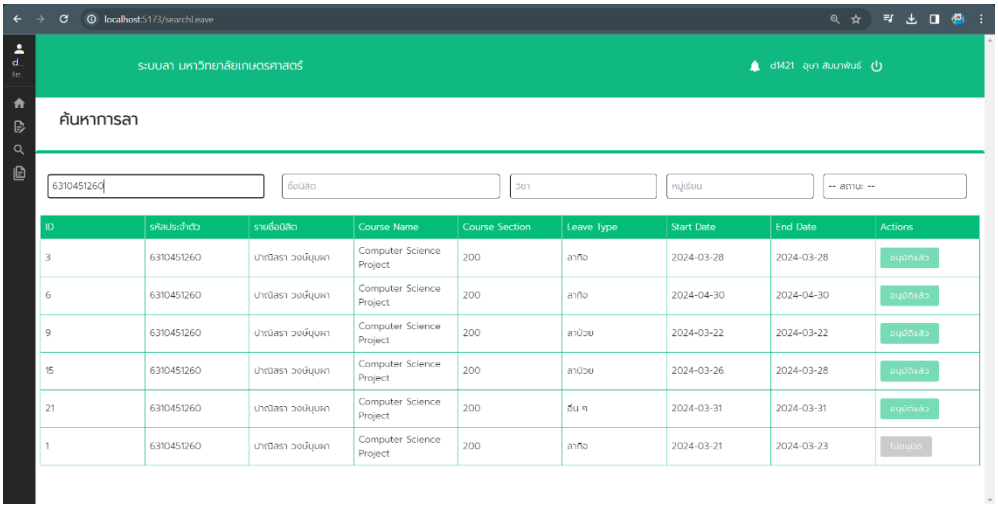
5.2.5 ทดสอบโปรแกรมส่วนการค้นหาการลาตามเงื่อนไข



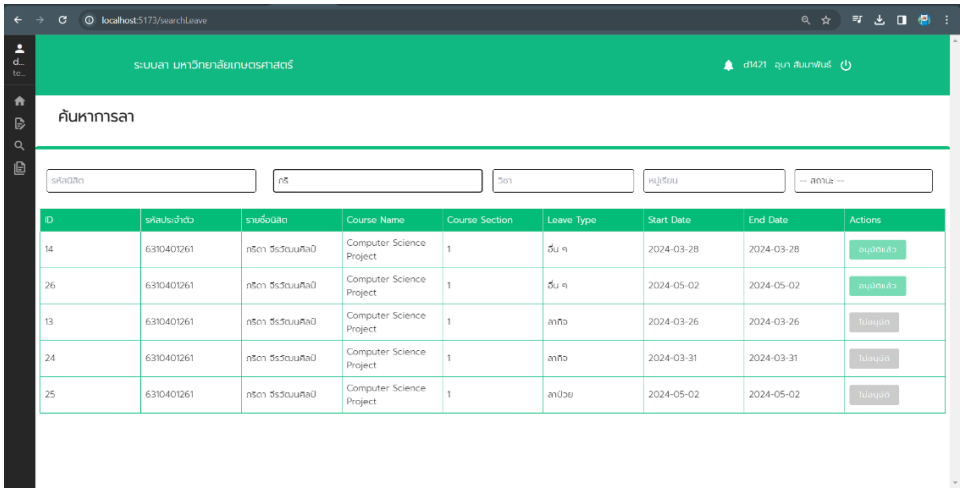
รูปที่ 5.33 แสดงแผนภาพการทดสอบโปรแกรมค้นหาการลาตามเงื่อนไข

ตารางที่ 5.5 แสดงกรณีทดสอบที่ 5

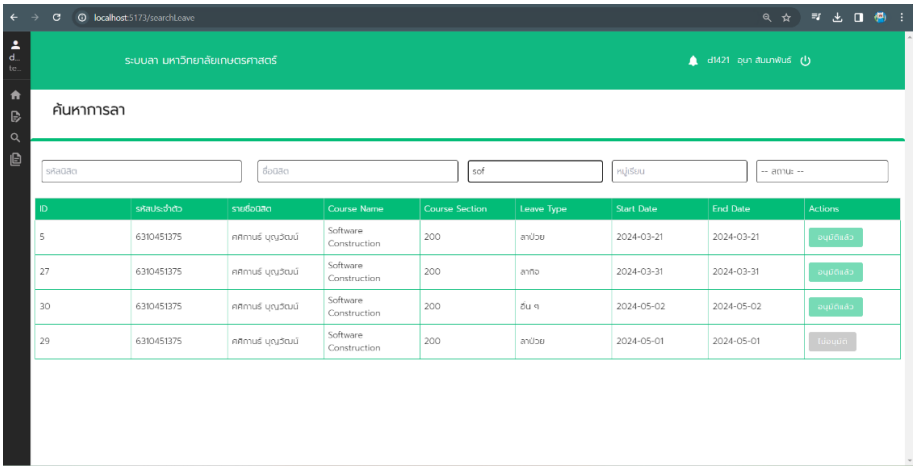
TC 5			
กรณีทดสอบ	เงื่อนไขก่อนการทดสอบ	ผลการทดสอบ	หมายเหตุ
TC 5.1 ค้นหาด้วยรหัสนิสิต	นิสิตยื่นใบลา	ผ่าน	-
TC 5.2 ค้นหาด้วยชื่อนิสิต	นิสิตยื่นใบลา	ผ่าน	-
TC 5.3 ค้นหาด้วยวิชา	นิสิตยื่นใบลา	ผ่าน	-
TC 5.4 ค้นหาด้วยหมู่เรียน	นิสิตยื่นใบลา	ผ่าน	-
TC 5.5 ค้นหาด้วยสถานะใบลา	นิสิตยื่นใบลา	ผ่าน	-
TC 5.6 ค้นหาด้วยประเภทการลา	นิสิตยื่นใบลา	ไม่ผ่าน	ไม่สามารถค้นหาจากประเภทการลาได้ เนื่องจากไม่มีช่องค้นหาโดยใช้ประเภทการลา



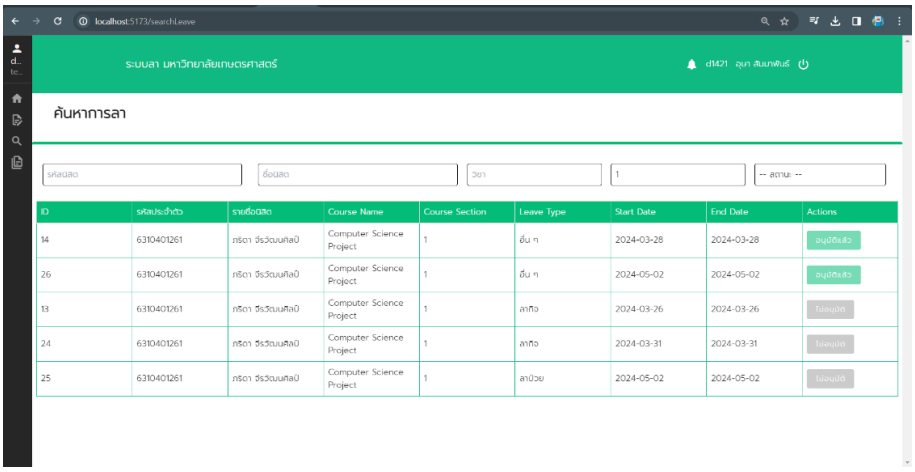
รูปที่ 5.34 แสดงภาพประกอบกรณีทดสอบ TC 5.1



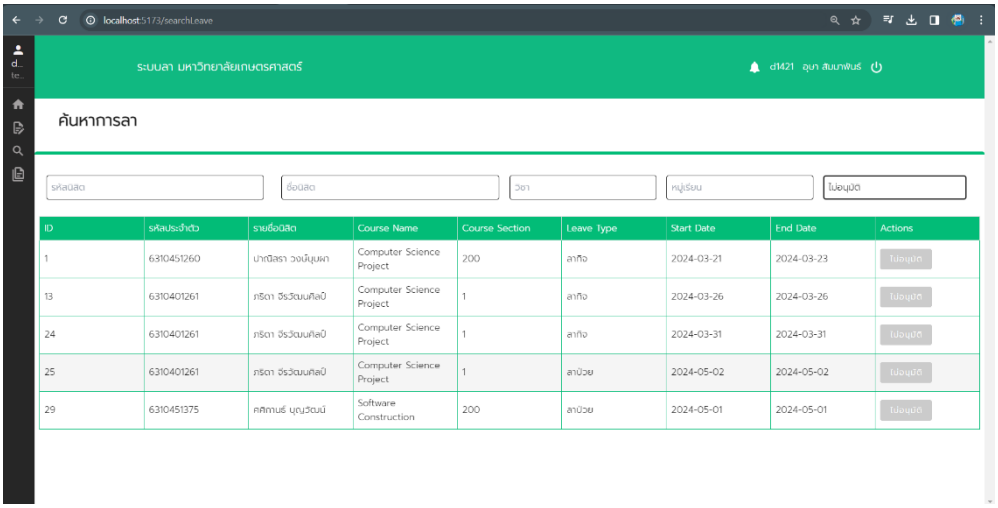
รูปที่ 5.35 แสดงภาพประกอบกรณีทดสอบ TC 5.2



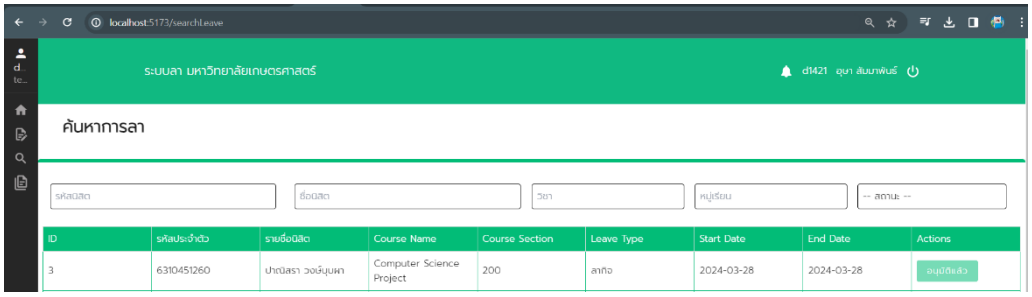
รูปที่ 5.36 แสดงภาพประกอบกรณีทดสอบ TC 5.3



รูปที่ 5.37 แสดงภาพประกอบกรณีทดสอบ TC 5.4

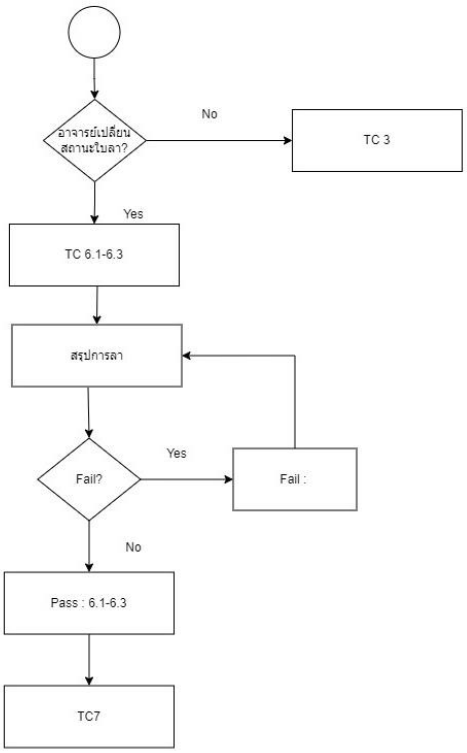


รูปที่ 5.38 แสดงภาพประกอบกรณีทดสอบ TC 5.5



รูปที่ 5.39 แสดงภาพประกอบกรณีทดสอบ TC 5.6

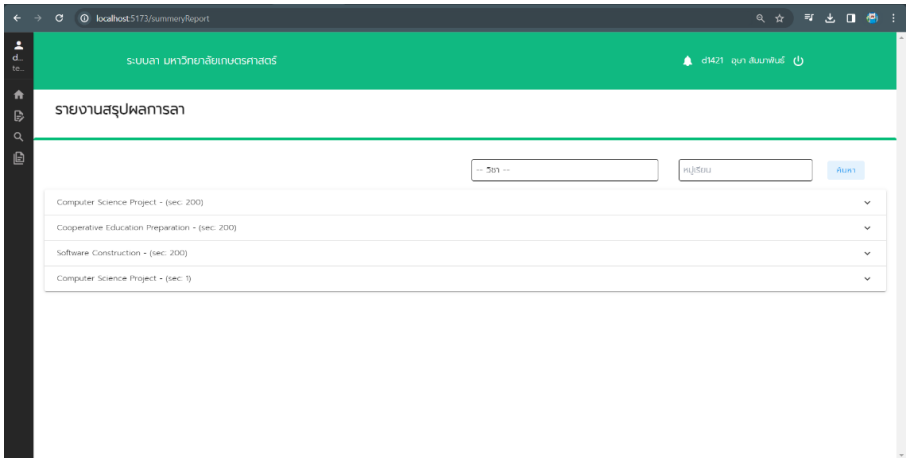
5.2.6 ทดสอบโปรแกรมส่วนรายงานสรุปการลา



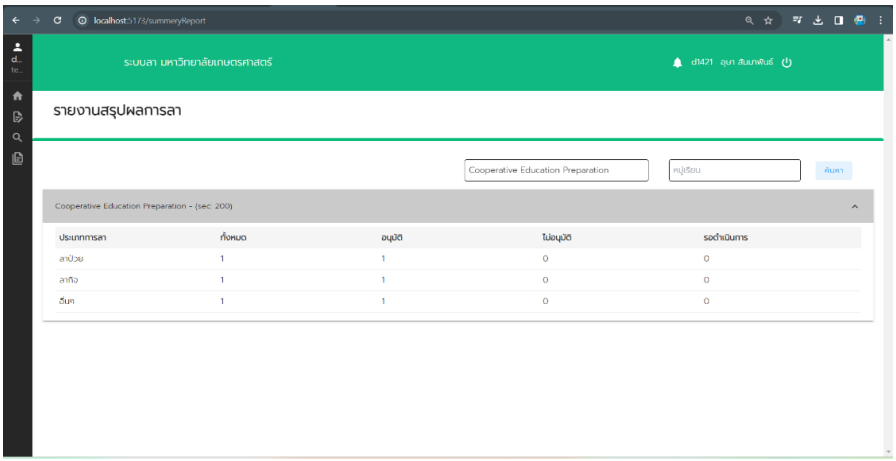
รูปที่ 5.40 แสดงแผนภาพการทดสอบโปรแกรมรายงานสรุปการลา

ตารางที่ 5.6 แสดงกรณีทดสอบที่ 6

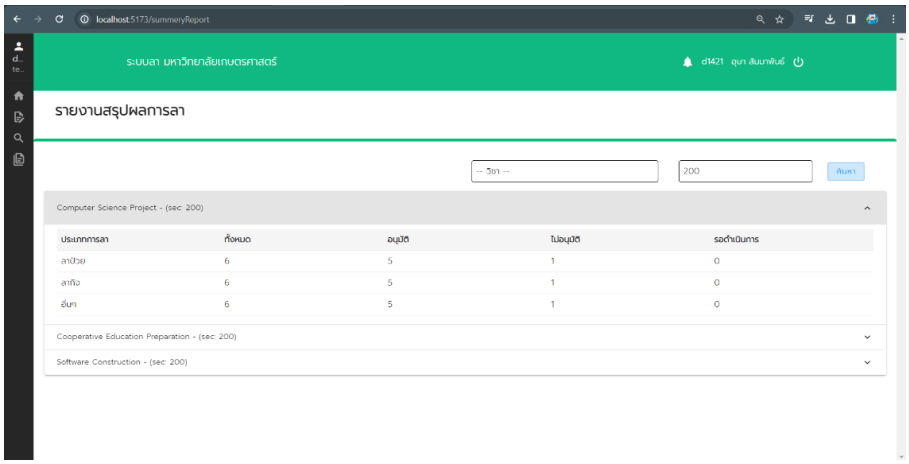
TC 6			
กรณีทดสอบ	เงื่อนไขก่อนการทดสอบ	ผลการทดสอบ	หมายเหตุ
TC 6.1 แสดงสรุปการลาทุกวิชาที่อาจารย์สอน	อาจารย์เปลี่ยนสถานะใบลาให้มีทั้งอนุมัติ/ไม่อนุมัติ/รอดำเนินการ	ผ่าน	-
TC 6.2 สามารถค้นหาสรุปการลาที่อาจารย์ต้องการตามชื่อวิชาได้	อาจารย์เปลี่ยนสถานะใบลาให้มีทั้งอนุมัติ/ไม่อนุมัติ/รอดำเนินการ	ผ่าน	-
TC 6.3 สามารถค้นหาสรุปการลาตามหมู่เรียนได้	อาจารย์เปลี่ยนสถานะใบลาให้มีทั้งอนุมัติ/ไม่อนุมัติ/รอดำเนินการ	ผ่าน	-



รูปที่ 5.41 แสดงภาพประกอบกรณีทดสอบ TC 6.1

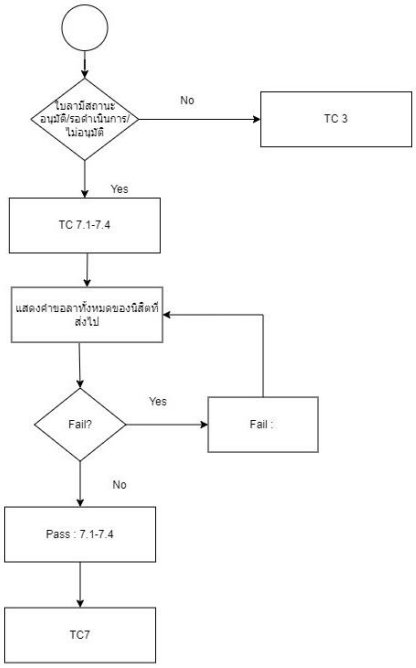


รูปที่ 5.42 แสดงภาพประกอบกรณีทดสอบ TC 6.2



รูปที่ 5.43 แสดงภาพประกอบกรณีทดสอบ TC 6.3

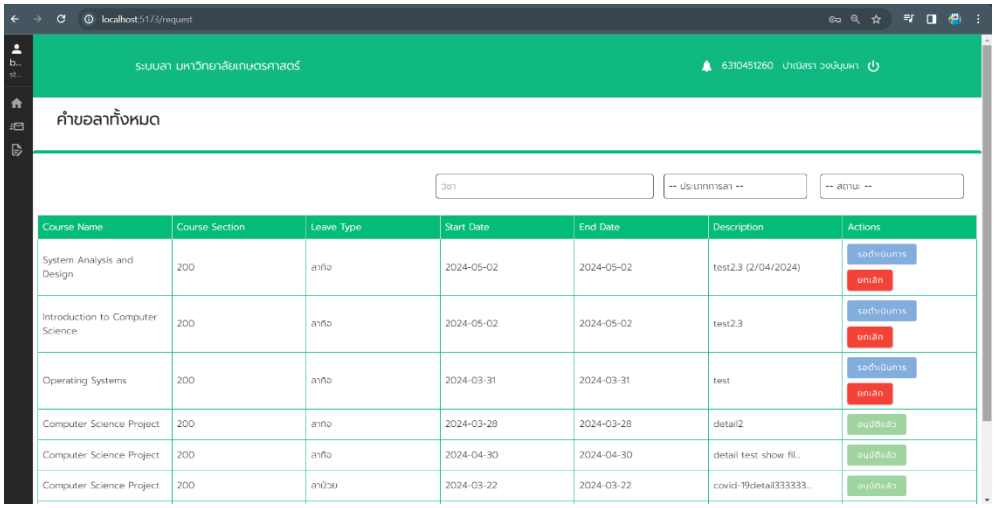
5.2.7 ทดสอบโปรแกรมส่วนแสดงคำขอลาทั้งหมดของนิสิต



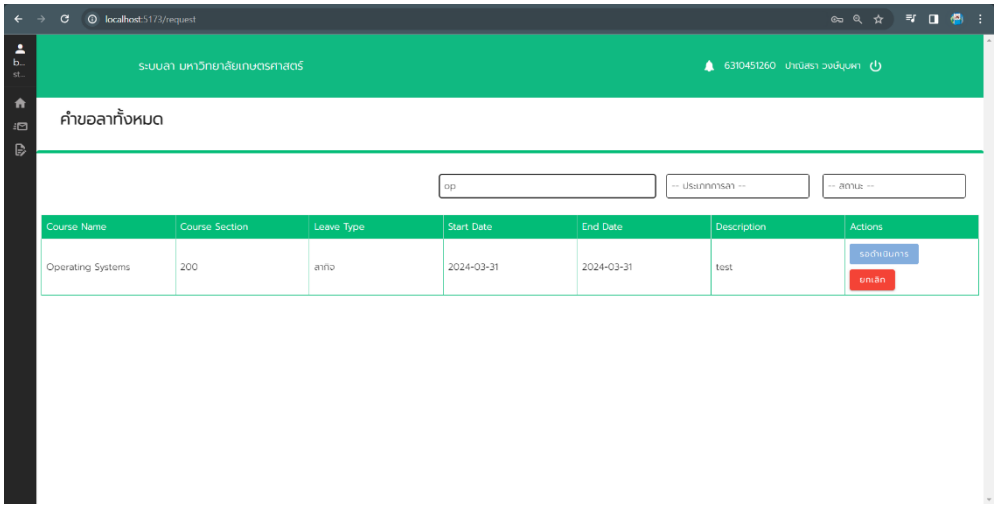
รูปที่ 5.44 แสดงแผนภาพการทดสอบโปรแกรมส่วนแสดงคำขอลาทั้งหมดของนิสิต

ตารางที่ 5.7 แสดงกรณีทดสอบที่ 7

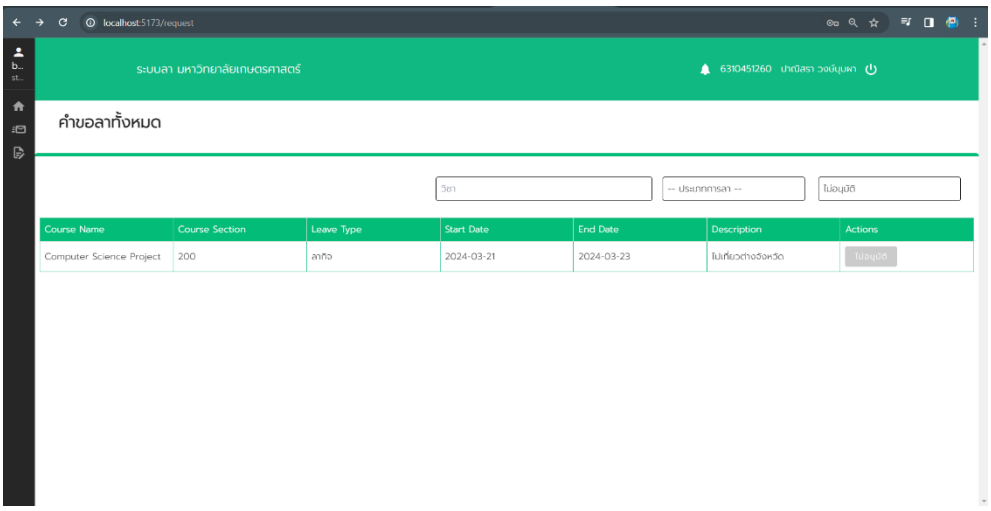
TC 7			
กรณีทดสอบ	เงื่อนไขก่อนการทดสอบ	ผลการทดสอบ	หมายเหตุ
TC 7.1 แสดงใบลาของวิชาที่นิสิตลงทะเบียนเรียนทั้งหมดที่ส่งไป	อาจารย์เปลี่ยนสถานะใบลาให้มีทั้งอนุมัติ/ไม่อนุมัติ/รอดำเนินการ	ผ่าน	-
TC 7.2 นิสิตสามารถค้นหาใบลาของตนตามชื่อวิชาที่ส่งไปได้	อาจารย์เปลี่ยนสถานะใบลาให้มีทั้งอนุมัติ/ไม่อนุมัติ/รอดำเนินการ	ผ่าน	-
TC 7.3 นิสิตสามารถค้นหาใบลาของตนตามสถานะใบลาได้	อาจารย์เปลี่ยนสถานะใบลาให้มีทั้งอนุมัติ/ไม่อนุมัติ/รอดำเนินการ	ผ่าน	-
TC 7.4 นิสิตสามารถยกเลิกใบลาที่กำลังรอดำเนินการอยู่ของตนเองได้	อาจารย์เปลี่ยนสถานะใบลาให้มีทั้งอนุมัติ/ไม่อนุมัติ/รอดำเนินการ	ผ่าน	-



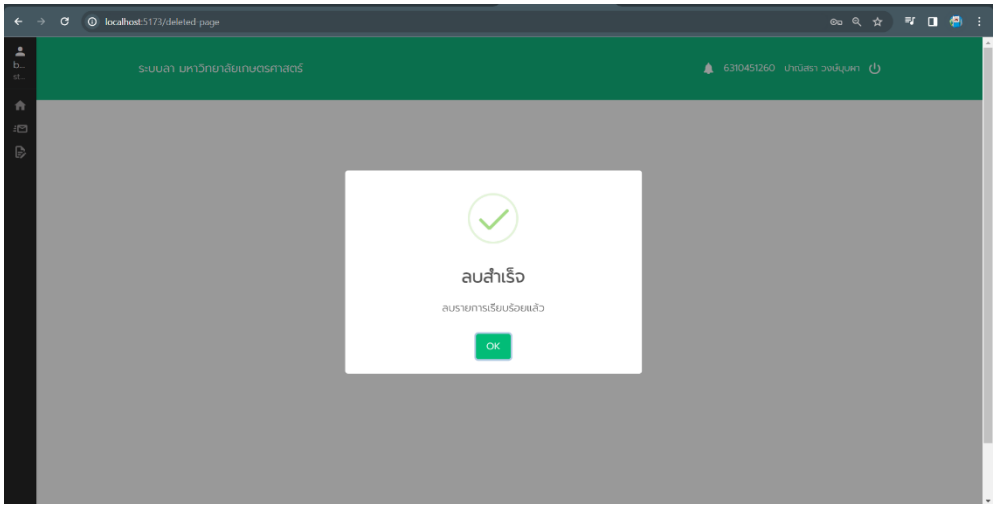
รูปที่ 5.45 แสดงภาพประกอบกรณีทดสอบ TC 7.1



รูปที่ 5.46 แสดงภาพประกอบกรณีทดสอบ TC 7.2

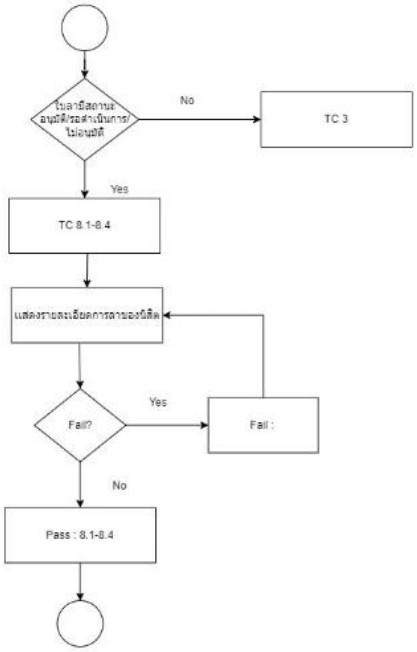


รูปที่ 5.47 แสดงภาพประกอบกรณีทดสอบ TC 7.3



รูปที่ 5.48 แสดงภาพประกอบกรณีทดสอบ TC 7.4

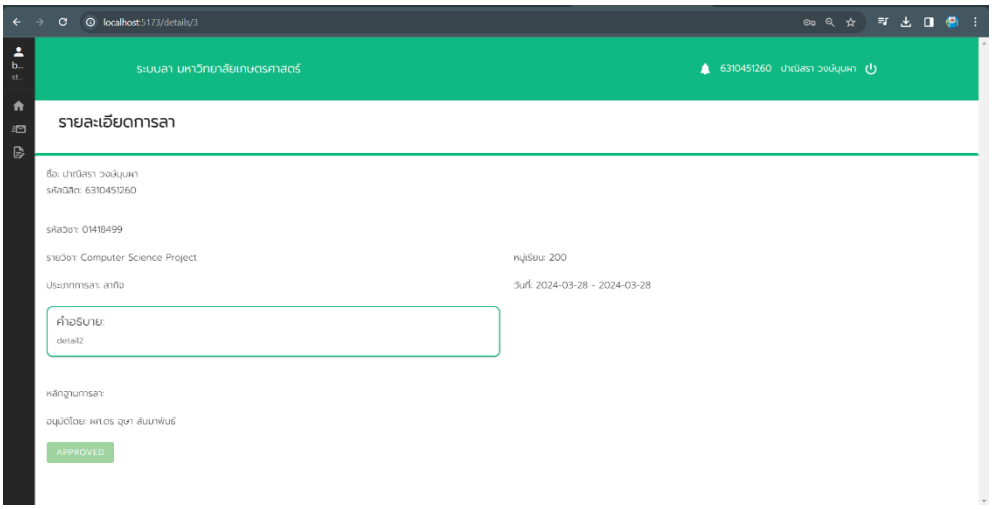
5.2.8 ทดสอบโปรแกรมส่วนหน้ารายละเอียดการลาของนิสิต



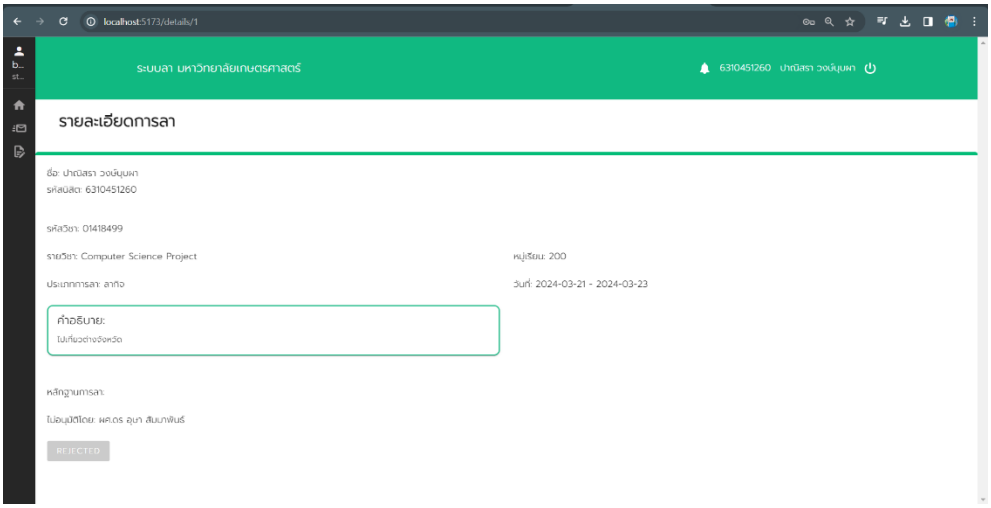
รูปที่ 5.49 แสดงแผนภาพการทดสอบโปรแกรมหน้ารายละเอียดการลาของนิสิต

ตารางที่ 5.8 แสดงกรณีทดสอบที่ 8

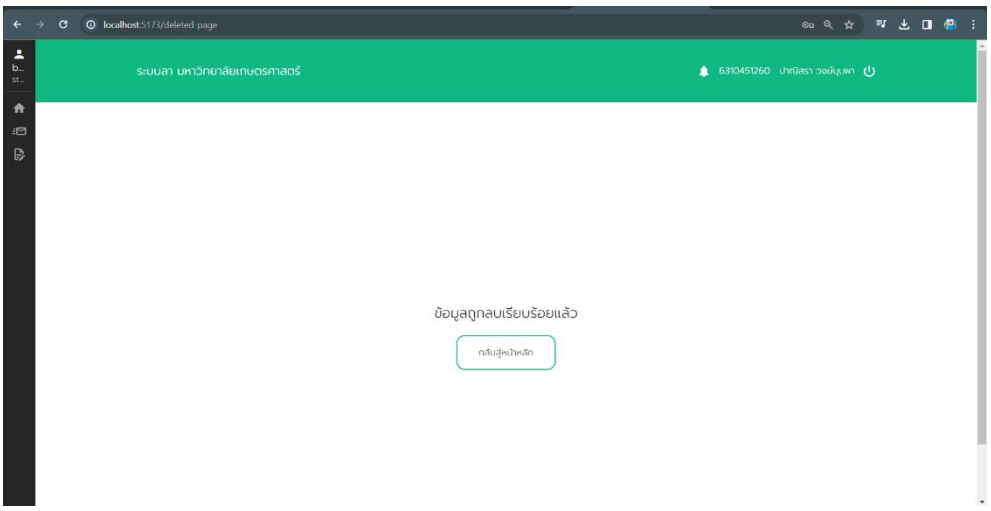
TC 8			
กรณีทดสอบ	เงื่อนไขก่อนการทดสอบ	ผลการทดสอบ	หมายเหตุ
TC 8.1 แสดงรายละเอียดการลาของนิสิตครบถ้วน พร้อมแสดงสถานะ “อนุมัติ”	อาจารย์กดอนุมัติใบลาที่จะแสดงรายละเอียด	ผ่าน	-
TC 8.2 แสดงรายละเอียดการลาของนิสิตครบถ้วน พร้อมแสดงสถานะ “ไม่อนุมัติ”	อาจารย์กดไม่อนุมัติใบลาที่จะแสดงรายละเอียด	ไม่ผ่าน	ปุ่มแสดงสถานะแสดงเป็น Rejected
TC 8.3 แสดงรายละเอียดการลาของนิสิตครบถ้วน พร้อมแสดงสถานะ “รออนุมัติ” แล้วสามารถ กด “ยกเลิก” ได้	-	ผ่าน	-
TC 8.4 สามารถ Download ไฟล์หลักฐานการลาได้ในใบลาที่แนบไฟล์มา	นิสิตส่งใบลาที่มีไฟล์หลักฐานการลาแนบ	ผ่าน	-



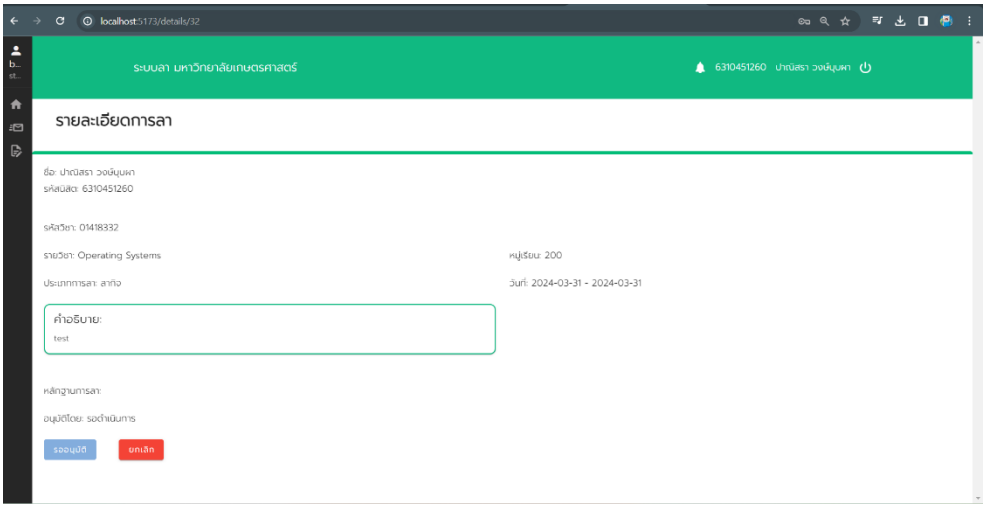
รูปที่ 5.50 แสดงภาพประกอบกรณีทดสอบ TC 8.1



รูปที่ 5.51 แสดงภาพประกอบกรณีทดสอบ TC 8.2



รูปที่ 5.52 แสดงภาพประกอบกรณีทดสอบ TC 8.3



รูปที่ 5.53 แสดงภาพประกอบกรณีทดสอบ TC 8.4

บทที่ 6
บทสรุป

6.1 บทสรุป

คณะผู้จัดทำได้บรรลุตามวัตถุประสงค์ในการพัฒนาเว็บแอปพลิเคชันการจัดการการลาของนิสิตในมหาวิทยาลัยเกษตรศาสตร์ที่มีการทำงานครบตามขอบเขตของงานที่ได้กำหนดไว้ คือ แจ้งลา-ยกเลิกการแจ้งลา แสดงคำขอลาทั้งหมด แสดงรายละเอียดการลา อนุมัติ-ปฏิเสธการลา ค้นหาใบลา และรายงานสรุปการลา

ซึ่งจากการพัฒนาเว็บแอปพลิเคชันการจัดการการลาของนิสิตในมหาวิทยาลัยเกษตรศาสตร์นี้ทำให้คณะผู้จัดทำมีความรู้ความเชี่ยวชาญมากขึ้นเกี่ยวกับการพัฒนาเว็บแอปพลิเคชันโดยใช้โปรแกรม Visual Studio Code เป็นเครื่องมือในการเขียน ภาษา Python, เฟรมเวิร์ค Django ในการพัฒนาระบบ Backend ได้ทำการต่อ Database MySQL ในการจัดการฐานข้อมูล และในส่วนของ Frontend คณะผู้จัดทำได้ใช้ภาษาโปรแกรม HTML CSS JAVASCRIPT และ VUE.JS Framework ทำให้คณะผู้จัดทำมีความรับผิดชอบต่องานที่ทำมากขึ้นตามหน้าที่ที่ได้รับมอบหมายจากการแบ่งการทำงานภายในคณะผู้จัดทำ เพิ่มความละเอียด รอบคอบแม่นยำให้คณะผู้จัดทำมากขึ้น เนื่องจากต้องมีการออกแบบโครงสร้างของระบบ ออกแบบฐานข้อมูลให้ออกมาเหมาะสมมากที่สุด ถ้าเริ่มต้นออกแบบโครงสร้างของระบบ หรือออกแบบฐานข้อมูลไม่เหมาะสม จะทำให้เป็นเรื่องยากลำบากในการแก้ไข มีความสามัคคีภายในคณะผู้จัดทำ เนื่องจากในระหว่างการทำงาน ย่อมเจอปัญหา หรือข้อผิดพลาดในงานบ่อยครั้ง ซึ่งเป็นการเพิ่มทักษะในการร่วมกันช่วยแก้ไขปัญหาเพื่อให้งานดำเนินต่อไปอย่างราบรื่น และทำให้งานออกมาสมบูรณ์

6.2 ปัญหาและข้อเสนอแนะ

- ในช่วงแรกเนื่องจากคณะผู้จัดทำยังไม่มีความเชี่ยวชาญในการใช้เครื่องมือเกี่ยวกับ Library Vuetify ทำให้คณะผู้จัดทำต้องใช้เวลาในการศึกษา และทำความเข้าใจ
- ในระหว่างกระบวนการพัฒนาเว็บแอปพลิเคชันมักพบปัญหาในการเขียนโปรแกรมเนื่องจากไม่เข้าใจไวยกรณ์ของภาษา และการทำงานของเครื่องมือที่มากพอ ทำให้พบข้อผิดพลาดในการทำงานอยู่บ่อยครั้ง
- ควรเพิ่มเติมให้มีการแสดงรูป เช่น รูปประจำตัวของนิสิตหรืออาจารย์เมื่อ login เข้าสู่ระบบ เพื่อใช้แสดงในแถบเมนูของผู้ใช้งาน และถือเป็นการยืนยันตัวตนของผู้ใช้บริการได้ในระดับหนึ่ง

6.3 แนวทางการพัฒนาต่อยอด

จากการวิเคราะห์ระบบเว็บแอปพลิเคชันที่คณะผู้จัดทำได้พัฒนาขึ้น ทำให้พบข้อจำกัดในการทำงานหลายๆอย่างในแอปพลิเคชัน คณะผู้จัดทำจึงมีความคิดในการพัฒนาต่อยอดเว็บแอปพลิเคชัน เพื่อให้รองรับความต้องการของผู้ใช้ได้มากขึ้น และเพื่อให้เว็บแอปพลิเคชันทำงานได้สมบูรณ์มากยิ่งขึ้น ดังนี้

1. การสร้างระบบการแจ้งเตือนเพื่อแจ้งเตือนนิสิตและอาจารย์เมื่อคำขอลาได้รับการอนุมัติหรือปฏิเสธ และแจ้งเตือนเกี่ยวกับวันเริ่มต้นและสิ้นสุดของการลาล่วงหน้า และแจ้งเตือนอาจารย์เมื่อมีคำขอลาส่งมาให้

2. การรองรับหลายภาษา เนื่องจากในมหาวิทยาลัยต่าง ๆ มักจะมีนิสิตหรืออาจารย์ต่างประเทศ การพัฒนาระบบที่สามารถให้บริการในหลายภาษาเพื่อรองรับแก่นิสิตและอาจารย์ที่มีส่วนเกี่ยวข้องในมหาวิทยาลัยให้มีความสะดวกสบาย

3. การใช้เทคโนโลยีสมัยใหม่ การใช้เทคโนโลยีเชิงลึก (Deep Learning) หรือปัญญาประดิษฐ์ (Artificial Intelligence) เพื่อช่วยในการวิเคราะห์และประมวลผลข้อมูล เช่น การระบุเนื้อหาจากคำขอลาเพื่อให้การตอบรับเป็นไปอย่างรวดเร็วและแม่นยำมากขึ้น

4. รองรับการใช้งานบนอุปกรณ์เคลื่อนที่ผ่านสมาร์ทโฟนและแท็บเล็ต เพื่อความสะดวกสบายในการใช้งานแก่นิสิตและอาจารย์

บรรณานุกรม

[1] bezkoder. (2021). Django + Vue.js: CRUD App with Django Rest Framework.
สืบค้นจาก https://www.bezkoder.com/django-vue-js-rest-framework/ วันที่สืบค้น: (1 ธันวาคม 2566)
[2] borntoDev . Vue.Js 101 ฉบับมาเร็ว ไปเร็ววว
สืบค้นจาก https://www.borntodev.com/2020/07/14/vue-js-101/ วันที่สืบค้น: (1 ธันวาคม 2566)
[3] Digital Guide. (2017). What is HTTP?.สืบค้นจาก
https://www.ionos.co.uk/digitalguide/hosting/technical-matters/what-is-http/
วันที่สืบค้น: (1 ธันวาคม 2566)
[4] duckfollow. (2017). JWT ด้วย Node.js.สืบค้นจาก https://duckfollow.co/view/jwt
วันที่สืบค้น: (1 ธันวาคม 2566)

ภาคผนวก ก
การติดตั้งระบบ

1. ติดตั้ง Node.js

- เข้าไปที่เว็บไซต์ Node.js และดาวน์โหลดตัวติดตั้งที่เป็นไฟล์ .msi หรือ .pkg ตามระบบปฏิบัติการ (Windows, macOS, Linux).
- เปิดไฟล์ตัวติดตั้งที่ดาวน์โหลดมา และทำตามขั้นตอนการติดตั้งบนหน้าจอ สามารถเลือกติดตั้งด้วยค่าเริ่มต้นได้ แต่ให้เลือกเพิ่มการติดตั้ง npm (Node Package Manager) ด้วย
- ตรวจสอบการติดตั้งโดยการเปิด Command Prompt (Windows) หรือ Terminal (macOs, Linux) พิมคำสั่ง “node -v” หรือ “npm -v”

2. ติดตั้ง python

- ไปที่เว็บไซต์หลักของ Python https://www.python.org/
- เมื่อติดตั้งเสร็จสิ้น สามารถทดสอบการติดตั้ง Python ได้โดยเปิด Command Prompt (Windows) python --version

3. ติดตั้ง Visual Studio Code

3.1 ดาวน์โหลดและติดตั้ง Visual Studio Code จากเว็บไซต์ https://code.visualstudio.com/

3.2 Extensions สำหรับ Django:

- Python: สำหรับการพัฒนาโปรเจค Django ควรติดตั้ง extension "Python" เพื่อให้ VS Code รองรับภาษา Python และ Django.
- Django: เพื่อรองรับการพัฒนาโปรเจค Django ใน VS Code อย่างมีประสิทธิภาพ.

3.3 Extensions สำหรับ Vue.js:

- Vetur เพื่อรองรับการพัฒนา Vue.js ใน VS Code โดยมีความสามารถในการเขียนโค้ด Vue.js และรันและทดสอบโค้ด.

3.4 อื่นๆ:

- Django Snippets เป็นคำสั่ง Snippets ที่ช่วยในการเขียนโค้ด Django
- Django Template

4. ติดตั้ง MySQL Database

4.1 ติดตั้ง XAMPP จากเว็บไซต์อย่างเป็นทางการของ XAMPP
https://www.apachefriends.org/download.html

4.2 เริ่มต้นเซิร์ฟเวอร์ XAMPP

- กดปุ่ม Start ที่ Apache และ MySQL
- กดปุ่ม Admin ที่ MySQL

4.3 สร้างฐานข้อมูล MySQL

- เข้าไปที่ http://localhost/phpmyadmin/ บนเว็บเบราว์เซอร์

- คลิกที่แท็บ Databases แล้วกรอกชื่อฐานข้อมูล “leave_management” ในช่อง Create database แล้วกดปุ่ม Create.

5. ติดตั้งโปรเจค Backend

5.1 สร้าง Folder ชื่อ Backend

5.2 Clone Git Hub Project https://github.com/dd3ho/Leave_Management_System_Banckend.git แล้ววางใน Folder Backend

5.3 เปิด Command Prompt เข้าไปยัง Directory ของ Backend

5.4 ใช้คำสั่ง “python -m venv env” สร้าง Virtual environment ใน Python

5.5 เมื่อสร้าง Virtual environment สำเร็จ (Folder env) ให้ Activate Virtual Environment โดย cd ไปที่ Directory .\env\Scripts\ และรันคำสั่ง activate

5.6 หลังจาก Activate Virtual environment สำเร็จ ให้ cd path/to/your/project เพื่อเข้าไปยัง Directory ของโปรเจกต์ที่ได้ Clone มาจาก GitHub โดยตำแหน่งที่เข้าไปใน Directory จะต้องมีไฟล์ manage.py อยู่ด้วย

5.7 ทำการติดตั้ง Dependencies ดังนี้

- pip install Django
- pip install djangorestframework
- pip install djangorestframework-simplejwt
- pip install pillow
- pip install django-cors-headers
- pip install mysqlclient (MySQL adapter)
- pip install

5.8 python -m pip install --upgrade pip เพื่ออัพเกรด pip เป็นเวอร์ชั่นล่าสุด

5.9 python manage.py migrate เพื่อสร้างฐานข้อมูลบน mysql (ต้องทำข้อ 4 ให้เสร็จก่อน)

5.10 python manage.py runserver เพื่อรันเซิร์ฟเวอร์ของโปรเจกต์ Backend

6. ติดตั้งโปรเจค Frontend

5.1 Clone GitHub Project https://github.com/PsYcHo6908/Leave_Management_Forntend.git

5.2 เปิด Command Prompt หรือ PowerShell ใช้คำสั่ง cd path/to/your/project เพื่อเข้าไปยัง Directory ของโปรเจกต์ที่ได้ Clone มาจาก GitHub

5.3 ใช้คำสั่ง npm install เพื่อติดตั้ง dependencies

5.4 npm run dev ใช้คำสั่งนี้เพื่อทำการรันเซิร์ฟเวอร์ของโปรเจกต์ Frontend

5.5 คัดลอก URL http://localhost:5173/signup ทำการเพิ่มผู้ใช้งานในระบบโดยมีจะสามารถสมัครบัญชีผู้ใช้สำหรับนิสิต อาจารย์ และ ผู้ดูแลระบบ

5.6 คัดลอก URL http://localhost:5173/login เพื่อนำ username และ password ที่สมัครบัญชีผู้ใช้ในข้อ 5.5 มาทำการเข้าสู่ระบบเพื่อเข้าใช้งานส่วนต่างๆ ตามประเภทผู้ใช้งานต่อไป

หมายเหตุ: สำหรับการจัดการระบบลาของ admin จะรองรับการทำงานดังนี้ หลังจาการเข้าสู่ระบบ

1. การลงทะเบียนบัญชีผู้ใช้งาน
2. การเพิ่มคณะ และ ภาควิชา
3. การเพิ่มรายวิชา

ในส่วนของการเพิ่มอาจารย์ประจำวิชา และนิสิตที่ลงเรียนในวิชานั้นทางผู้จัดทำได้ใช้โปรแกรม Postman เข้ามาช่วยในการสร้างข้อมูลเริ่มต้นดังนี้

5 ทำการสมัครบัญชีผู้ใช้งานในข้อ 5.5

6 เข้าสู่ระบบเพื่อนำ JWT Token(access Token) มาใช้เพื่อส่ง Request

URL: http://127.0.0.1:8000/loogin /

Method: Post

Headers: Content-Type: application/x-www-form-urlencoded

Body: username=ชื่อผู้ใช้&password=รหัสผ่าน ในรูปแบบ key/value pairs

7 การเพิ่มอาจารย์ประจำวิชา

URL: http://127.0.0.1:8000/instructorCourse/

Method: POST

Headers:

- Content-Type: application/x-www-form-urlencoded
- Authorization: Bearer [JWT Token] (ใส่ JWT Token ที่ได้รับจากขั้นตอนที่ 1 ในส่วน Value ของ Authorization Header)

Body:

teacher_id= Primary Key Value &course_id= Primary Key Value ในรูปแบบ key/value pairs

8 นิสิตที่ลงเรียนในวิชานั้น

http://127.0.0.1:8000/studentRegister/

Method: POST

Headers:

- Content-Type: application/x-www-form-urlencoded
- Authorization: Bearer [JWT Token] (ใส่ JWT Token ที่ได้รับจากขั้นตอนที่ 1 ในส่วน Value ของ Authorization Header)

Body:

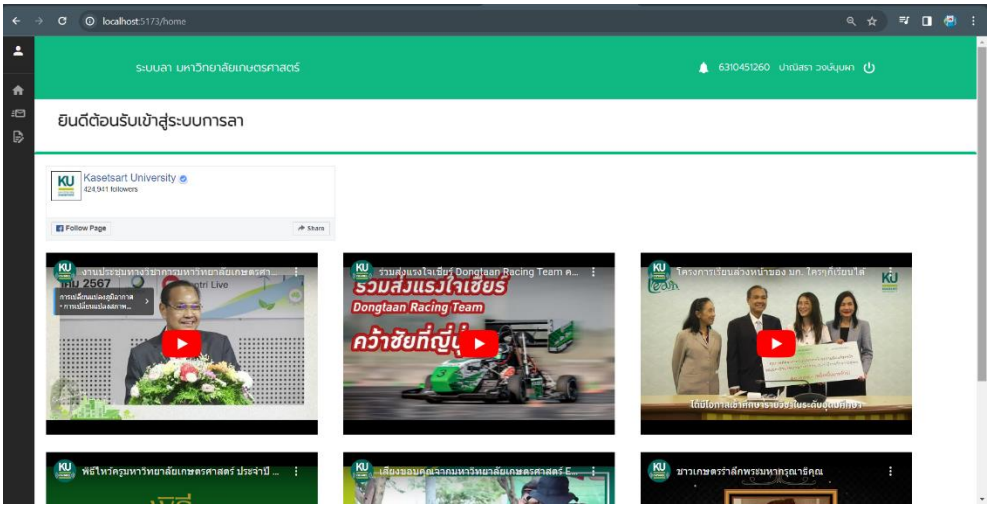
student_id= Primary Key Value &course_id= Primary Key Value ในรูปแบบ key/value pairs

เมื่อทำการสร้างข้อมูลเริ่มต้นระบบสำเร็จระบบลาก็จะพร้อมใช้งาน

ภาคผนวก ข
วิธีการใช้งานระบบ

1. หน้าแรก - นิสิต

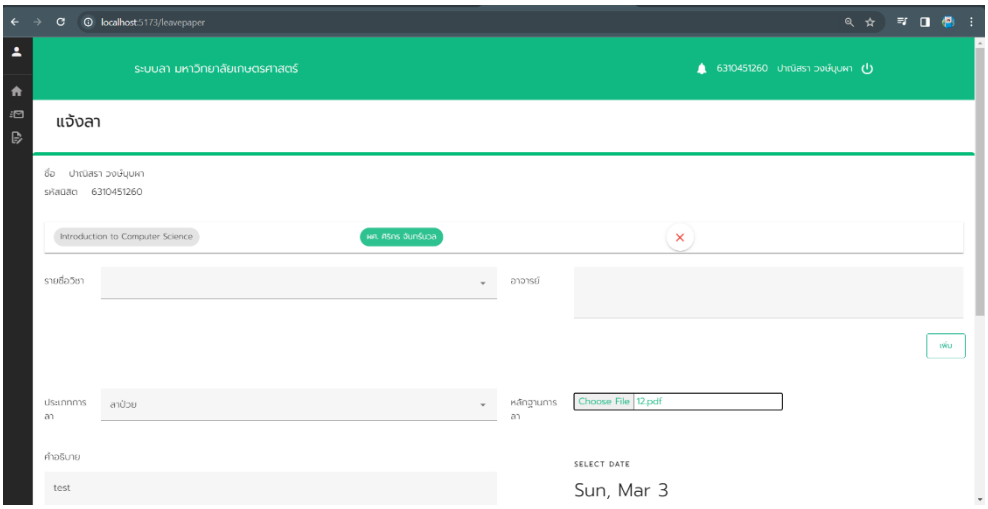
นิสิตเข้าสู่ระบบแล้ว จะพบหน้าแรกของเว็บ โดยนิสิตสามารถเลือกรายการที่ต้องการได้จากแถบเมนูด้านซ้าย โดยจะมีรายการ ดังนี้ 1. หน้าแรก เป็นรายการเพื่อให้นิสิตกลับมายังหน้าแรกของเว็บ 2. แจ้งลา เป็นรายการเพื่อให้นิสิตไปยังหน้าแจ้งลา 3. คำขอลาทั้งหมด เป็นหน้าเพื่อให้นิสิตไปยังหน้าคำขอลาทั้งหมด นิสิตสามารถออกจากระบบได้โดยการกดปุ่มทางด้านขวาบนของโปรแกรม



รูปที่ ข.1 หน้าแรก - นิสิต

2. หน้าแจ้งลา - นิสิต

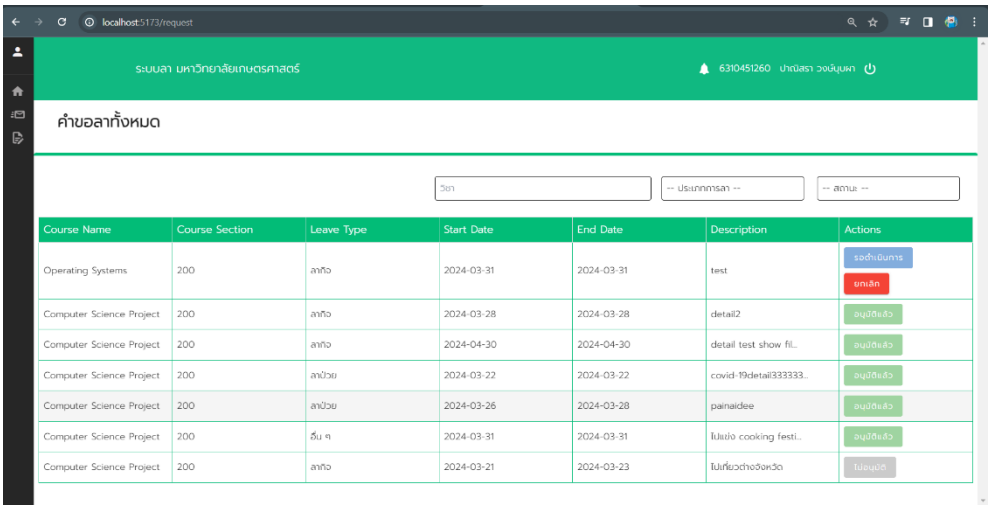
นิสิตกรอกข้อมูลรายชื่อวิชา และอาจารย์ประจำวิชาที่ต้องการส่งใบลา หลังจากนั้นกดปุ่ม เพิ่ม และกรอกรายละเอียดอื่น ๆ ให้ครบ หลังจากนั้นกดปุ่ม ส่ง



รูปที่ ข.2 วิธีการใช้ระบบ-หน้าแจ้งลา (นิสิต)

3. หน้าคำขอลาทั้งหมด - นิสิต

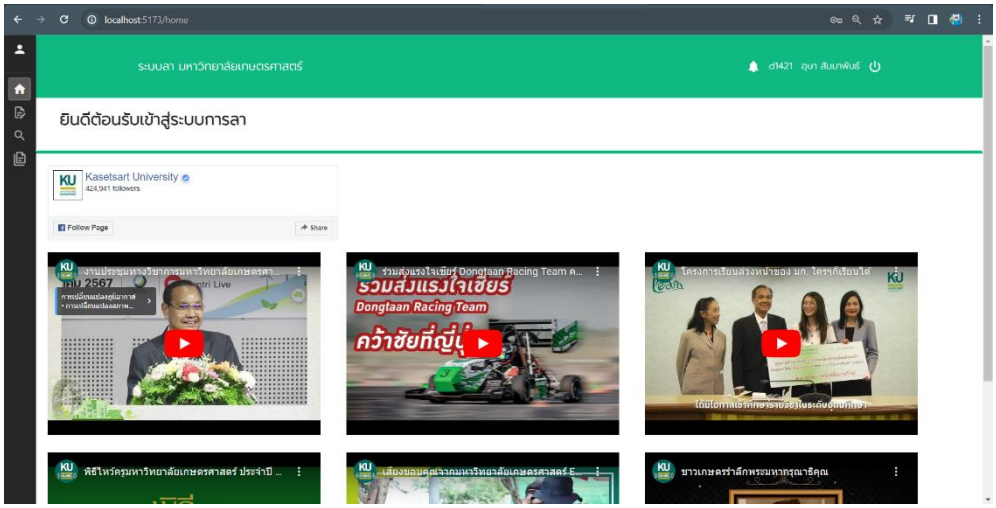
นิสิตกรอกข้อมูลรายชื่อวิชา ประเภทการลา หรือ สถานะใบลา เพื่อดูคำขอลาที่ต้องการดูได้ นิสิตกดคำขอลาเพื่อดูรายละเอียดคำขอลาและกดปุ่ม ยกเลิก ของใบลาที่ต้องการยกเลิกใบลานั้นได้



รูปที่ ข.3 วิธีการใช้ระบบ-คำขอลาทั้งหมด (นิสิต)

4. หน้าแรก - อาจารย์

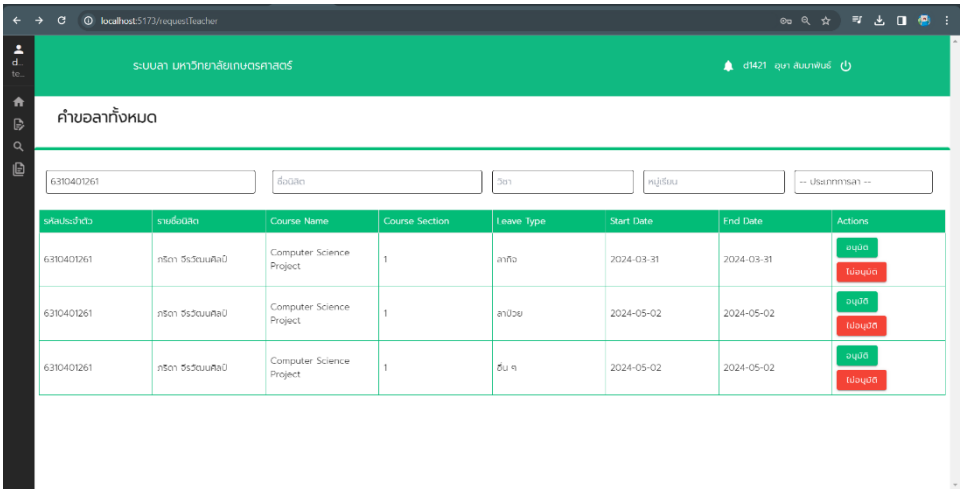
อาจารย์ทำการเข้าสู่ระบบแล้ว จะพบหน้าแรกของเว็บ โดยอาจารย์สามารถเลือกรายการที่ต้องการได้จากแถบเมนูด้านซ้าย โดยจะมีรายการ ดังนี้ 1. หน้าแรก เป็นรายการเพื่อให้อาจารย์กลับมายังหน้าแรกของเว็บ 2. คำขอลาทั้งหมด เป็นหน้าเพื่อให้อาจารย์สามารถดูคำขอลาทั้งหมดที่นิสิตส่งมา 3. ค้นหาการลาตามเงื่อนไข เป็นหน้าเพื่อให้อาจารย์ค้นหาการลาที่ต้องการได้ 4. รายงานสรุปการลา เป็นหน้าเพื่อให้อาจารย์สามารถดูสรุปผลรายงานการลาของวิชานั้น ๆ ได้



รูปที่ ข.4 วิธีการใช้ระบบ-หน้าแรก (อาจารย์)

5. คำขอลาทั้งหมด - อาจารย์

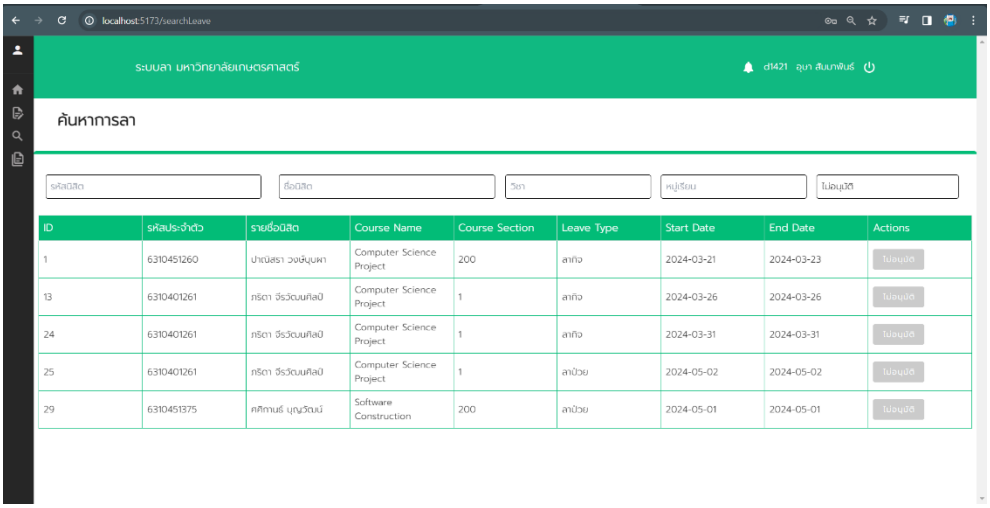
อาจารย์กรอกข้อมูล รหัสนิสิต ชื่อ-นามสกุล วิชา หมู่เรียน หรือประเภทการลาเพื่อหาคำขอลาที่นิสิตส่งมาได้ อาจารย์สามารถกดที่คำขอลาเพื่อดูรายละเอียดใบลานั้น ๆ ได้ และกดปุ่ม อนุมัติ เพื่ออนุมัติใบลาหรือกดปุ่ม ไม่อนุมัติ เพื่อไม่อนุมัติใบลา



รูปที่ ข.5 วิธีการใช้ระบบ-คำขอลาทั้งหมด (อาจารย์)

6. ค้นหาการลาตามเงื่อนไข - อาจารย์

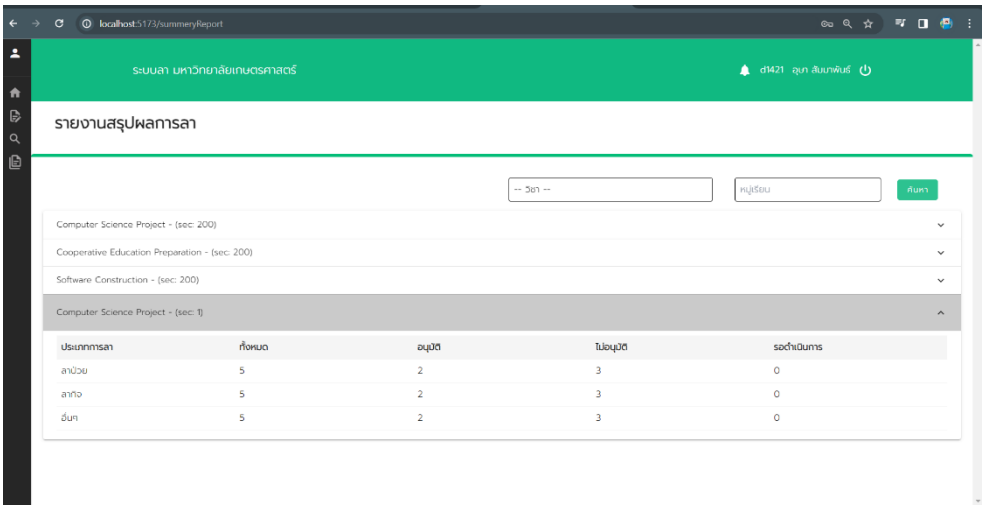
อาจารย์กรอกข้อมูล รหัสนิสิต ชื่อ-นามสกุล วิชา หมู่เรียน หรือประเภทการลาเพื่อหาคำขอลาที่นิสิตส่งมาได้ อาจารย์สามารถกดที่คำขอลาเพื่อดูรายละเอียดใบลานั้น ๆ ได้ และกดปุ่ม อนุมัติ เพื่ออนุมัติใบลาหรือกดปุ่ม ไม่อนุมัติ เพื่อไม่อนุมัติใบลา



รูปที่ ข.6 วิธีการใช้ระบบ-ค้นหาการลาตามเงื่อนไข

7. รายงานสรุปการลา - อาจารย์

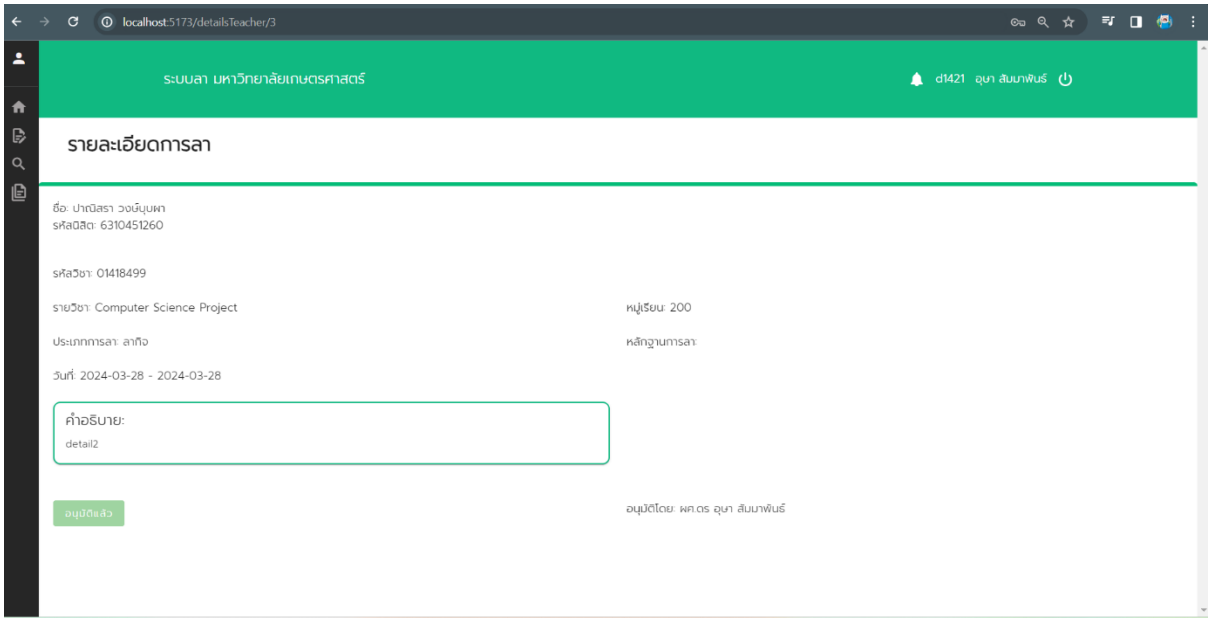
อาจารย์กรอกข้อมูล รายชื่อวิชา หรือหมู่เรียน และกดปุ่ม ค้นหา เพื่อค้นหารายงานสรุปผลของวิชานั้น และกดแถบรายการเลือกแบบดึงลง เพื่อดูผลการรายงานของวิชานั้น ๆ



รูปที่ ข.7 วิธีการใช้ระบบ-รายการสรุปการลา

8. แสดงรายละเอียดการลา

อาจารย์กดเข้าไปยังใบลาต่าง ๆ ในหน้าคำขอลาทั้งหมด นิสิตหรืออาจารย์สามารถดูข้อมูลรายละเอียดของใบลานั้นได้ และอาจารย์สามารถกดอนุมัติหรือไม่อนุมัติตามพิจารณาได้



รูปที่ ข.8 วิธีการใช้ระบบ-แสดงรายละเอียดการลา