

โครงการเลขที่ วศ.คพ. S006-2/2564

เรื่อง

เว็บแอปพลิเคชันสำหรับจับคู่เพื่อนร่วมห้องและจองหอพักในมหาวิทยาลัย

โดย

นายกมลพัฒน์ สุนทรพงศ์ รหัส 620610771

รายงานนี้เป็นส่วนหนึ่งของวิชาสำรวจเพื่อโครงการ
ตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
ภาควิชาวิศวกรรมคอมพิวเตอร์
คณะวิศวกรรมศาสตร์ มหาวิทยาลัยเชียงใหม่
ปีการศึกษา 2564

PROJECT No. CPE S006-2/2564

Web application for matching roommates and booking campus dormitory

Kamonpat Sunthonpong 620610771

**A Report Submitted in Partial Fulfillment of Project Survey Course
as Required by the Degree of Bachelor of Engineering**

Department of Computer Engineering

Faculty of Engineering

Chiang Mai University

2021

หัวข้อโครงการ : เว็บแอปพลิเคชันสำหรับจับคู่เพื่อนร่วมห้องและจองหอพักในมหาวิทยาลัย
: Web application for matching roommates and booking campus dormitory
โดย : นายกมลพัฒน์ สุนทรพงศ์ รหัส 620610771
ภาควิชา : วิศวกรรมคอมพิวเตอร์
อาจารย์ที่ปรึกษา : อ.ดร. ชินวัตร อิศราดิศัยกุล
ปริญญา : วิศวกรรมศาสตรบัณฑิต
สาขา : วิศวกรรมคอมพิวเตอร์
ปีการศึกษา : 2564

ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ มหาวิทยาลัยเชียงใหม่ ได้อนุมัติให้โครงการนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต (สาขาวิศวกรรมคอมพิวเตอร์)

..... หัวหน้าภาควิชาวิศวกรรมคอมพิวเตอร์
(รศ.ดร. สันติ พิทักษ์กีนูกร)

คณะกรรมการสอบโครงการ

..... ประธานกรรมการ
(อ.ดร. ชินวัตร อิศราดิศัยกุล)

..... กรรมการ
(ผศ.ดร. นวदनย์ คุณเลิศกิจ)

..... กรรมการ
(รศ.ดร. จักรพงศ์ นาทวีชัย)

สารบัญ

สารบัญ	ข
1 บทนำ	1
1.1 ที่มาของโครงการ	1
1.2 วัตถุประสงค์ของโครงการ	1
1.3 ขอบเขตของโครงการ	1
1.3.1 ขอบเขตด้านฮาร์ดแวร์	1
1.3.2 ขอบเขตของระบบ	1
1.4 ประโยชน์ที่ได้รับ	2
1.5 เทคโนโลยีและเครื่องมือที่ใช้	2
1.5.1 เทคโนโลยีด้านซอฟต์แวร์	2
1.6 แผนการดำเนินงาน	2
1.7 ผลกระทบด้านสังคม สุขภาพ ความปลอดภัย กฎหมาย และวัฒนธรรม	3
2 ทฤษฎีที่เกี่ยวข้อง	4
2.1 ความรู้เกี่ยวกับด้านอัลกอริทึม	4
2.1.1 Stable roommate matchings Problem	4
2.1.2 NP-Problem	4
2.1.3 Stable roommates problem with triple rooms	4
2.1.4 Multi objective optimization problem	5
2.1.5 Genetic algorithm	5
2.2 Containerization	5
2.2.1 Docker	5
2.3 ความรู้ตามหลักสูตรซึ่งถูกนำมาใช้หรือบูรณาการในโครงการ	5
3 โครงสร้างของโครงการ	6
3.1 Alice in Wonderland	6
3.1.1 The Black Kitten	6
3.1.2 The Reproach	6
4 การประเมินระบบ	8
บรรณานุกรม	9

บทที่ 1

บทนำ

1.1 ที่มาของโครงการ

หอพักในนักศึกษา เป็นหอพักที่นักศึกษาชั้นปีที่ 1 ส่วนมากให้ความสนใจ เนื่องจากราคาที่ถูก อีกทั้งตั้งอยู่ในเขตมหาวิทยาลัยที่มีโครงสร้างพื้นฐานมากมาย แต่เนื่องจากระบบการลงทะเบียน มีปัญหาในการรองรับผู้ใช้งานจำนวนมากไม่ได้ เมื่อมีจำนวนผู้ใช้งานเป็นจำนวนมากจะทำให้ระบบ ไม่ตอบสนองแล้วไม่มี package ส่งกลับมาให้ผู้ใช้งาน ประกอบกับขั้นตอนการลงทะเบียนที่มีหลายขั้น ตอนคือ 1 ลงชื่อเข้าใช้งาน 2 เลือกหอพักที่ต้องการพักอาศัย 3 เลือกห้องพักที่ต้องการพักอาศัย 4 ยืนยันการลงทะเบียน ซึ่งในแต่ละขั้นตอนต้องส่ง request เรียกหน้าเว็บไซต์ในทุกขั้นตอน อีกทั้งเว็บไซต์ยังประมวลผลการลงทะเบียนตลอดเวลา หากห้องที่เลือก หรือ หอที่เลือกเต็มแล้วต้อง ย้อนกลับไปหน้าก่อนหน้า ซึ่งเป็นการส่ง request ใหม่ไปยังเซิร์ฟเวอร์เช่นกัน ส่งผลให้นักศึกษาต้องรีบ ลงทะเบียนให้ได้อยู่ในหอที่ต้องการ หลังจากนั้นจึงไปแลกห้องให้ได้อยู่กับเพื่อนที่ต้องการ ซึ่งอาจจะทำให้ เพื่อนไม่ได้อยู่หอเดียวกัน หรือห้องเดียวกัน ทำให้เพื่อนร่วมห้องที่มี มาจากการสุ่ม ซึ่งอาจจะทำให้ไม่เข้ากัน นำไปสู่การทะเลาะกัน หรืออยู่ด้วยกันแบบอึดอัดใจกัน

ทางผู้จัดทำจึงได้คิดวิธีการแก้ปัญหาตัวระบบเก่าโดยการลดจำนวน request ที่ส่งมายัง server เพื่อแก้ปัญหา server รองรับ request จำนวนมากไม่ได้ ซึ่งระบบที่จะพัฒนาขึ้นใหม่นั้นจะไม่ประมวลผลระบบ ณ ขณะที่เปิดให้จอง แต่จะรอให้ระบบปิดก่อนจึงจะประมวลผล และให้ผู้ใช้งานกรอกแบบสอบถามเพื่อนำมาใช้ในการเลือกเพื่อนร่วมห้องที่เหมาะสมที่สุด

1.2 วัตถุประสงค์ของโครงการ

1. พัฒนาเว็บแอปพลิเคชันตามหลักการออกแบบ UX/UI
2. ลดกระบวนการการคำนวณของ server
3. จับคู่รวมเมท, หอพัก และ ห้องพักไม่ให้เกิด rouge couple

1.3 ขอบเขตของโครงการ

1.3.1 ขอบเขตด้านฮาร์ดแวร์

1. เครื่องคอมพิวเตอร์ที่สามารถเชื่อมต่ออินเทอร์เน็ตได้
2. เครื่องคอมพิวเตอร์ที่สามารถจัดเก็บข้อมูลการลงทะเบียนทั้งหมดได้

1.3.2 ขอบเขตของระบบ

1. ระบบจะรองรับเฉพาะการจองของหอพักในนักศึกษา มช.
2. การจับคู่จะพิจารณาจากคุณสมบัติที่ระบบมีให้เท่านั้น
3. ผู้ใช้จะไม่สามารถเลือกเฉพาะเจาะจงห้องพัก หรือ หอพักได้โดยตรง

1.4 ประโยชน์ที่ได้รับ

1. ผู้ใช้ไม่รู้สึกรีดอัดขณะใช้งาน
2. สังคมหอพักน่าอยู่ยิ่งขึ้น
3. ภาพลักษณ์ที่ดีของมหาวิทยาลัย

1.5 เทคโนโลยีและเครื่องมือที่ใช้

1.5.1 เทคโนโลยีด้านซอฟต์แวร์

1. Typescript
2. React.js
3. Next.js
4. Gin
5. Go
6. mySQL
7. GraphQL
8. Docker

1.6 แผนการดำเนินงาน

ขั้นตอนการดำเนินงาน	ธ.ค. 2564	ม.ค. 2565	ก.พ. 2565	มี.ค. 2565	เม.ย. 2565	พ.ค. 2565	มิ.ย. 2565	ก.ค. 2565	ส.ค. 2565	ก.ย. 2565	ต.ค. 2565	พ.ย. 2565	ธ.ค. 2565	ม.ค. 2566	ก.พ. 2566
ศึกษาค้นคว้าอัลกอริทึม และ งานที่คล้ายคลึงกัน															
สอบถามข้อมูลจาก สำนักรงานหอพัก															
รวบรวมข้อมูล สำหรับการทดสอบระบบ															
เลือก เครื่อง มือ ใน การ พัฒนาระบบ															
ศึกษา เรียน รู้ เกี่ยว กับ เทคโนโลยี และ เครื่องมือที่ใช้พัฒนาระบบ															

ขั้นตอนการดำเนินงาน	ธ.ค. 2564	ม.ค. 2565	ก.พ. 2565	มี.ค. 2565	เม.ย. 2565	พ.ค. 2565	มิ.ย. 2565	ก.ค. 2565	ส.ค. 2565	ก.ย. 2565	ต.ค. 2565	พ.ย. 2565	ธ.ค. 2565	ม.ค. 2566	ก.พ. 2566
ออกแบบเว็บไซต์แอปพลิเคชัน															
พัฒนาระบบฐานข้อมูล															
พัฒนาเว็บไซต์แอปพลิเคชัน															
ทดสอบระบบ															
เขียน รายงาน สรุป ผลของการพัฒนาระบบ															

1.7 ผลกระทบด้านสังคม สุขภาพ ความปลอดภัย กฎหมาย และวัฒนธรรม

แนวทางและโยชนในการประยุกต์ใช้งานโครงการกับงานในด้านอื่นๆ รวมถึงผลกระทบในด้านสังคมและสิ่งแวดล้อมจากการใช้ความรู้ทางวิศวกรรมที่ได้

บทที่ 2

ทฤษฎีที่เกี่ยวข้อง

การทำโครงการ เริ่มต้นด้วยการศึกษาค้นคว้า ทฤษฎีที่เกี่ยวข้อง หรือ งานวิจัย/โครงการ ที่เคยมีผู้เสนอไว้แล้ว ซึ่งเนื้อหาในบทนี้ก็จะเกี่ยวกับการอธิบายถึงสิ่งที่เกี่ยวข้องกับโครงการ เพื่อให้ผู้อ่านเข้าใจเนื้อหาในบทถัดๆ ไปได้ง่ายขึ้น

2.1 ความรู้เกี่ยวกับด้านอัลกอริทึม

2.1.1 Stable roommate matchings Problem

Stable roommate matching problem เป็นปัญหาหนึ่งในกลุ่ม Stable matching problem โจทย์ของปัญหามีอยู่ว่า มีผู้เข้าร่วมอยู่ทั้งหมด $2n$ คนโดยทุกๆคนจะทำการจัดอันดับความต้องการ roommate ของตนเองเป็นจำนวน $2n-1$ อันดับ ซึ่งการจับคู่ผู้เข้าร่วมจะถูกนิยามเป็นคู่อันดับ (m,n) ใดๆ โดยการจับคู่ (m,n) ใดๆ จะเรียกว่าเป็น stable matching ก็ต่อเมื่อ ไม่มีผู้เข้าร่วมคนอื่นคนใด หรือ n' ที่พร้อมจะจับคู่กับตนเอง หรือ m มากกว่า roommate n ที่ได้รับซึ่ง (m, n') จะเรียกว่าเป็นคู่อุปสรรคให้ไม่มีการเกิด stable matching เกิดขึ้น

ซึ่งจากปัญหาข้างต้นนั้นจะเห็นได้ว่ามีลักษณะคล้ายคลึงกับโครงการฉบับนี้และ ณ ขณะนี้ศึกษา โครงการอยู่นั้นก็มีอัลกอริทึมที่สามารถแก้ปัญหานี้ได้แล้วนั้นคือ Irving's Algorithm ที่สามารถหา stable matching ได้ภายในเวลา $O(n^2)$

2.1.2 NP-Problem

ต่อไปนี้จะเป็นการอธิบายนิยามสั้นๆ ของกลุ่มปัญหา NP เพื่อต่อยอดความรู้สู่ส่วนถัดไป

1. NP-Problem คือเซตของปัญหาที่สามารถหาคำตอบได้ด้วย Nondeterministic Turing machine ภายในเวลา Polynomial time ของ input
2. NP-Hard คือปัญหาที่สามารถนำวิธีแก้ปัญหานั้นไปแก้ปัญหา NP อื่นๆได้
3. NP-Complete คือปัญหาที่สามารถพิสูจน์ได้ว่าเป็นทั้ง NP-Problem และ NP-Hard

ซึ่งเมื่อใดก็ตามที่ปัญหานั้นๆ ถูกสรุปว่าเป็น NP-Complete แล้วจะไม่นิยมหาวิธีที่ดีที่สุด แต่จะเปลี่ยนเป็นหาวิธีประมาณการแทน

2.1.3 Stable roommates problem with triple rooms

[3] จากบทพิสูจน์ ข้างต้นจึงสรุปได้ว่า Stable roommate problem ที่ 1 ห้องมีมากกว่า 2 คนนั้น เป็นปัญหา NP-Complete ทางผู้จัดทำจึงเลือกที่จะไม่หาวิธีที่ดีที่สุด ในการแก้ปัญหานี้

2.1.4 Multi objective optimization problem

2.1.5 Genetic algorithm

2.2 Containerization

ส่วนต่อไปจากนี้จะเกี่ยวกับการ deploy เว็บแอปพลิเคชัน โดยเริ่มจากทำความรู้จักกับ Containerization ว่าจริงๆแล้วคืออะไร

Containerization[2] คือการนำเอาแอปพลิเคชันที่พัฒนา, library หรือสิ่งต่างๆที่ จำเป็นในการทำงานของแอปพลิเคชันทั้งหมด บรรจุลงในกล่องๆเดียวเรียกกล่องนั้นว่า container เมื่อต้องการที่จะนำแอปพลิเคชันนั้นไปทดสอบในระบบต่างๆเพียงเปิดกล่องดังกล่าวขึ้นมา ก็จะมีอุปกรณ์ทุกอย่างที่พร้อมจะทำให้แอปพลิเคชันทำงานได้ปกติ การทำ containerization จึงมีความคล่องตัวในการ deploy แอปพลิเคชันในสภาพแวดล้อมต่างๆได้ดี

2.2.1 Docker

docker[4] เป็น container runtime engine ที่ช่วยสร้าง และ รัน container

2.3 ความรู้ตามหลักสูตรซึ่งถูกนำมาใช้หรือบูรณาการในโครงการ

1. Database design: ทำให้มีความรู้พื้นฐานในการออกแบบและพัฒนาฐานข้อมูล
2. Object Oriented Program(OOP): ช่วยให้เข้าใจการออกแบบและหลักการของการเขียนโปรแกรมเชิงวัตถุ
3. Data Structure and Algorithm: นำความรู้ไปต่อยอดในการศึกษาและพัฒนาอัลกอริทึม

บทที่ 3

โครงสร้างของโครงการ

ในบทนี้จะกล่าวถึงหลักการ และการออกแบบระบบ

3.1 Alice in Wonderland

3.1.1 The Black Kitten

One thing was certain, that the WHITE kitten had had nothing to do with it:—it was the black kitten’s fault entirely [1]. For the white kitten had been having its face washed by the old cat for the last quarter of an hour (and bearing it pretty well, considering); so you see that it COULDN’T have had any hand in the mischief.

The way Dinah washed her children’s faces was this: first she held the poor thing down by its ear with one paw, and then with the other paw she rubbed its face all over, the wrong way, beginning at the nose: and just now, as I said, she was hard at work on the white kitten, which was lying quite still and trying to purr—no doubt feeling that it was all meant for its good.

But the black kitten had been finished with earlier in the afternoon, and so, while Alice was sitting curled up in a corner of the great arm-chair, half talking to herself and half asleep, the kitten had been having a grand game of romps with the ball of worsted Alice had been trying to wind up, and had been rolling it up and down till it had all come undone again; and there it was, spread over the hearth-rug, all knots and tangles, with the kitten running after its own tail in the middle.

3.1.2 The Reproach

‘Oh, you wicked little thing!’ cried Alice, catching up the kitten, and giving it a little kiss to make it understand that it was in disgrace. ‘Really, Dinah ought to have taught you better manners! You OUGHT, Dinah, you know you ought!’ she added, looking reproachfully at



รูปที่ 3.1: The Walrus and the Carpenter

the old cat, and speaking in as cross a voice as she could manage—and then she scrambled back into the arm-chair, taking the kitten and the worsted with her, and began winding up the ball again. But she didn't get on very fast, as she was talking all the time, sometimes to the kitten, and sometimes to herself. Kitty sat very demurely on her knee, pretending to watch the progress of the winding, and now and then putting out one paw and gently touching the ball, as if it would be glad to help, if it might.

'Do you know what to-morrow is, Kitty?' Alice began. 'You'd have guessed if you'd been up in the window with me—only Dinah was making you tidy, so you couldn't. I was watching the boys getting in stick for the bonfire—and it wants plenty of sticks, Kitty! Only it got so cold, and it snowed so, they had to leave off. Never mind, Kitty, we'll go and see the bonfire to-morrow.' Here Alice wound two or three turns of the worsted round the kitten's neck, just to see how it would look: this led to a scramble, in which the ball rolled down upon the floor, and yards and yards of it got unwound again.

'Do you know, I was so angry, Kitty,' Alice went on as soon as they were comfortably settled again, 'when I saw all the mischief you had been doing, I was very nearly opening the window, and putting you out into the snow! And you'd have deserved it, you little mischievous darling! What have you got to say for yourself? Now don't interrupt me!' she went on, holding up one finger. 'I'm going to tell you all your faults. Number one: you squeaked twice while Dinah was washing your face this morning. Now you can't deny it, Kitty: I heard you! What that you say?' (pretending that the kitten was speaking.) 'Her paw went into your eye? Well, that's YOUR fault, for keeping your eyes open—if you'd shut them tight up, it wouldn't have happened. Now don't make any more excuses, but listen! Number two: you pulled Snowdrop away by the tail just as I had put down the saucer of milk before her! What, you were thirsty, were you?

บทที่ 4

การประเมินระบบ

ในบทนี้จะทดสอบเกี่ยวกับการทำงานในฟังก์ชันหลักๆ

บรรณานุกรม

- [1] Lewis Carroll. *Alice's Adventures in Wonderland*. George MacDonald, 1865.
- [2] Red Hat. *What is containerization?* <https://www.redhat.com/en/topics/cloud-native-apps/what-is-containerization>, 2021.
- [3] Shuichi Miyazaki Kazuo Iwama and Kazuya Okamoto. *Stable roommates problem with triple rooms*. School of Informatics, Kyoto University, 2007.
- [4] Docker team. *Docker Overview*. <https://docs.docker.com/get-started/overview/>.