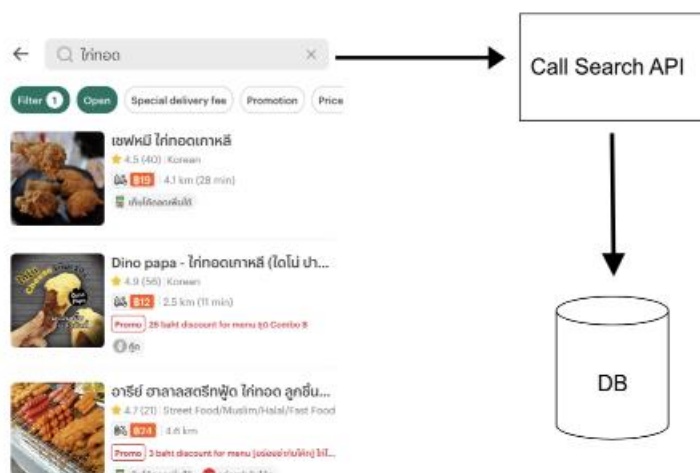


Question 2

You have been assigned to be a QA of LINE MAN Search service and your squad is implementing a new feature called search suggestion which has the following business requirements:

Requirements	Example input & output
When user search by menu name, the service will suggest restaurants which contains that menu	Input - ไก่ทอด Output - KFC - Texas Chicken
When user enter misspelling word, the result should still be shown correctly (allow misspelling only 2 character)	Input = ราดหนัส Output - ราดหน้าสมโภชเอ็มไพร์ - ราดหน้าเคียงเอ็มไพร์

Given architecture of LINE MAN Womngai applications as below.



As a QA, please answer the questions below.

2.1 Design your test strategy. How do you test this system? What kind of test will you do? Be specific about techniques/tools as well as pros and cons.

Example Answer: (Please write as much as possible)

Test Strategy	Pros	Cons
เราจะใช้ Performance Test ในการตรวจสอบควบคู่กับ Usability Testing โดย Performance Test จะใช้เพื่อตรวจสอบความสามารถในการรองรับจำนวนผู้ใช้และเพื่อดูความเหมาะสมของ Search API	- Performance Test สามารถหาจุดคอขวดหรือขีดจำกัดของจำนวน user ที่สามารถใช้ได้ หรือ Scalability ได้ดี ทำให้เห็นว่าจะสามารถรองรับ user ตามที่เราคาดการณ์ไว้ได้ และยัง	- ข้อเสียคือหากต้องมีการ ขื้อทดสอบ ในจำนวนมหาศาลจะทำให้ต้องสูญเสียทรัพยากรจำนวนมากได้ ทั้งขนาดของ computing power หรือ infrastructure ดังนั้นต้องมี

<p>รวมถึง Server ด้วย จะช่วยในการยืนยันว่าระบบสามารถทำงานได้อย่างมีประสิทธิภาพ โดย tools นั้นจากที่ได้ศึกษามา APACHE jMeter จะเหมาะสมที่สุดเนื่องจากมีความในการทำงาน dynamic application รวมถึงสามารถใช้ในการ test ภายใต้ varying load condition และ มีการไหลลงใน server หนักได้ด้วย</p> <p>บวกกับการทำ รวมถึงสามารถใช้ Usability testing เพื่อเก็บรวบรวมข้อมูลทั้งการใช้งานของผู้ใช้หรือความพึงพอใจของผู้ใช้ ซึ่งจะช่วยให้สามารถแก้ไขได้ตรงจุดและทำให้การ Testing ด้วยวิธีอื่นมีประสิทธิภาพดีขึ้นด้วย โดย Tools ใช้เป็นทั้งการเก็บข้อมูลพฤติกรรมของผู้ใช้ หรือสามารถเก็บ feedback จากแหล่งที่มาภายนอกได้เช่น feedback จาก application store ต่างๆ ได้ และเราจะทำการตรวจสอบ ด้วย Unit Testing โดยทำการทดสอบ Source Code โดยการเขียน Code เพื่อตรวจสอบอีกว่า Code ที่พัฒนาออกมานั้นทำงานถูกต้องหรือไม่ ซึ่ง Unit test จะเป็นส่วนสำคัญใน Searching suggestion เพราะเราจะทำการตรวจสอบอย่างลึกใน API และยังรวมถึงตรวจสอบความถูกต้องของ Core Business Logic ด้วย โดยจะใช้ Tools คือ Pytest ซึ่งข้อดีคือ มี plugin ให้ใช้หลากหลายและมีการเขียนที่เข้าใจง่าย และจากการศึกษาเพิ่มเติมพบว่าเราสามารถนำ CI/CD เพื่อนำไปใช้เป็น Testing</p>	<p>ส่งเสริมให้ ช่วยเพิ่มประสิทธิภาพ Response time ของ search suggestion feature ประกอบกับ Tools ที่เลือกมาคือ Apache JMeter ทำให้สามารถทดสอบระบบได้หลากหลายสถานการณ์และตรงกับประโยชน์ที่กล่าวมาข้างต้นอีกด้วย สามารถทดสอบขีดจำกัด API Rate ได้ ซึ่งจะช่วยกันภัยคุกคามทางไซเบอร์ได้นั้นคือการป้องกันการโจมตีแบบ DOS attacks หรือการใช้งานที่มหาศาลได้ และรวมถึง Concurrency testing โดยจะวัดการจัดการคำขอที่มาพร้อมๆกัน โดยเฉพาะเมื่อติดต่อกับฐานข้อมูลที่มีจำนวนมากๆ จะช่วยให้แอปพลิเคชันทำงานได้ดีเมื่อมีการค้นหาที่เกิดขึ้นพร้อมกัน ในส่วนของฐานข้อมูล (Database) สามารถตรวจสอบประสิทธิภาพการทำงานได้ โดยทดสอบในสถานการณ์ที่มีการดึงข้อมูลจำนวนมากๆ และยังสามารถตรวจสอบเวลาเฉลี่ยของการ Response time ได้ทำให้สามารถพัฒนาการทำงานของฐานข้อมูล ได้ตามสถานการณ์</p> <p>- Unit test ช่วยในการตรวจสอบความถูกต้องของการค้นหาเมนู โดยการทดสอบ</p>	<p>ความระมัดระวังในการทดสอบ เพื่อไม่ให้กระทบในส่วนอื่นๆ ของระบบ และ การทำ Testing ในรูปแบบนี้จะมุ่งเน้นไปที่การทดสอบในเชิงลึกและมีจำนวนมหาศาลทำให้ในบางครั้งการ ทายผลลัพธ์ของปัญหาอาจจะไม่ตรงตามจุดประสงค์ ทำให้เมื่อมีการใช้งานจริงอาจพบข้อผิดพลาดของระบบได้ รวมไปถึงการวิเคราะห์ข้อมูลที่ได้มาจากการ testing ซึ่งเมื่อมีจำนวนมาก จะทำให้การตรวจสอบจุดบกพร่องได้ยาก แต่สามารถแก้ไขได้โดยมีผู้เชี่ยวชาญคอยช่วยเหลือในด้านการวิเคราะห์ข้อมูล</p> <p>- Unit test อาจไม่ครอบคลุมการทดสอบได้ทุกกรณี บางครั้งการทดสอบที่ใช้งานจริงอาจมีปัญหาก็ไม่เกินไปตามที่ทดสอบได้ เราสามารถแก้ไขได้โดยเพิ่ม การทดสอบในรูปแบบอื่นๆ เช่น usability test เพื่อช่วยในการหาข้อผิดพลาดของระบบได้อย่างตรงจุด แต่การทำ CI/CD Automated test บางครั้งอาจสร้าง case ที่เป็น false positive ได้ซึ่งจะมากหรือน้อยขึ้นอยู่กับความซับซ้อนของทั้ง Application และ Algorithm ด้วย ซึ่งสามารถแก้ไขได้โดยการปรับเปลี่ยน</p>
--	---	---

<p>Automation ได้ที่จะช่วยในการตรวจสอบให้อัตโนมัติทุกครั้งที่มีการเปลี่ยนแปลง และลอง Deploy ใน Real-world Environments แบบจำลอง แล้วดูผลลัพธ์ที่ได้ หากพบปัญหาจะทำการแก้ไขและอัปเดตเพื่อให้สามารถรองรับผู้ใช้งานได้จริง</p>	<p>ระบบสามารถให้คำแนะนำร้านอาหารที่มีเม็ดเงินได้อย่างถูกต้อง ตาม Requirements ที่กำหนดมาให้ ซึ่งมีประโยชน์เป็นอย่างมากในการทำหาข้อผิดพลาดของระบบ เพราะ Unit test นั้นจะครอบคลุม Code ที่สำคัญของเรารวมถึงครอบคลุม business logic ที่มีความซับซ้อนด้วย และยังสามารถรับการเปลี่ยนแปลงของ Code ได้ดี รวมถึงการทดสอบเพื่อยืนยันการ Response time ระหว่าง Search API, ฐานข้อมูล (Database) ซึ่งรวมถึงการตรวจสอบ ความถูกต้องของข้อมูลด้วย (Integrity), การตรวจสอบว่า API ทำการ Query ได้ถูกต้อง และการตรวจสอบคำตอบที่ได้กับข้อมูลที่คาดหวัง ยังเป็นส่วนสำคัญของระบบ</p> <p>Automation Testing CI/CD ซึ่งทำให้การปรับปรุงและการทดสอบสามารถทำได้ทุกครั้งที่มีการเปลี่ยนแปลง รวมถึงสนับสนุนการทำงาน Process ที่รวดเร็วและมีประสิทธิภาพ</p> <ul style="list-style-type: none"> - สามารถแก้ไขปัญหาได้อย่างตรงจุดทั้งด้านของ Web application หรือ Application โดยการรับ feedback การใช้งานจากท 	<p>วิธีการ monitoring ตามสถานะของ Application ใน ณ ขณะนั้น</p>
---	---	--

	ทั้งสองด้านซึ่งได้มาจาก Usability testing	
--	--	--

2.2 Please write test cases to cover cases as much as possible.

Scenario 1: ดึงข้อมูลชื่อร้านผ่านการเขียนชื่อเมนู	
Input	Output
“กะเพรา” Database: “กะเพราเสี้อป่า, อาหารตามสั่งยายจันทร์, กะเพรากัน”	กะเพราเสี้อป่า,กะเพรากัน
Scenario 2: ดึงข้อมูลชื่อร้านผ่านการเขียนชื่อเมนู (ผิดที่อยู่ใน 2 Characters)	
Input	Output
“ราดนา” Database: “ราดหน้าสมโภชเอ็มไพร์, ราดหน้าเคียงเอ็มไพร์, แปะราดหน้า มหาชน, ราดหน้าสยาม”	ราดหน้าสมโภชเอ็มไพร์, แปะราดหน้า มหาชน
Scenario 3: ดึงข้อมูลชื่อร้านผ่านการเขียนชื่อเมนู (ผิดมากกว่า 2 Characters)	
Input	Output
“โกทาด” Database:” KFC, Texas Chicken, Five Star Chicken, ไก่ทอดหาดใหญ่, McDonald's, Burger King”	KFC, Texas Chicken
Scenario 4: ไม่ใส่อะไรเลย	
Input	Output
“” Database:” KFC, Texas Chicken, Five Star Chicken, ไก่ทอดหาดใหญ่, McDonald's, Burger King, กะเพราเสี้อป่า, อาหารตามสั่งยายจันทร์, กะเพรากัน”	แนะนำร้านที่น่าสนใจ เช่น KFC ,อาหารตามสั่งยายจันทร์ เป็นต้น
Scenario 5: Special Character	
Input	Output
“@#\$!#%” Database:” KFC, Texas Chicken, Five Star Chicken, ไก่ทอดหาดใหญ่, McDonald's, Burger King, กะเพราเสี้อป่า, อาหารตามสั่งยายจันทร์, กะเพรากัน”	ขออภัยไม่พบสิ่งที่คุณสนใจ

Scenario 6: Common stop word or Boolean Search (and, or, not)	
Input	Output
“ไก่ทอดกับเบอเกอร์” Database:” KFC, Texas Chicken, Five Star Chicken, ไก่ทอดขนาดใหญ่, McDonald's, Burger King, กระเพาะเลื้อป่า, อาหารตามสั่งยายจันทร์, กระเพรา ก้น”	KFC, McDonald's, Five Star Chicken, Burger King
Scenario 7: Mixed language	
Input	Output
“Fried ไก่” Database:” KFC, Texas Chicken, Five Star Chicken, ไก่ทอดขนาดใหญ่, McDonald's, Burger King, กระเพาะเลื้อป่า, อาหารตามสั่งยายจันทร์, กระเพรา ก้น”	KFC, Texas Chicken, Five Star Chicken
Scenario 8: Geospatial Search หาเมนูผ่านการเขียนชื่อสถานที่เช่น จังหวัด เมือง	
Input	Output
“นนทบุรี” Database:” KFC, Texas Chicken, Five Star Chicken, ไก่ทอดขนาดใหญ่, McDonald's, Burger King, กระเพาะเลื้อป่า, อาหารตามสั่งยายจันทร์, กระเพรา ก้น”	Five Star Chicken, กระเพาะเลื้อป่า, อาหารตามสั่งยายจันทร์, McDonald's
Scenario 9: numeric values	
Input	Output
“12314” Database:” KFC, Texas Chicken, Five Star Chicken, ไก่ทอดขนาดใหญ่, McDonald's, Burger King, กระเพาะเลื้อป่า, อาหารตามสั่งยายจันทร์, กระเพรา ก้น”	ขอภัยไม่พบสิ่งที่คุณสนใจ
Scenario 10: Performance Testing	
Input	Expected Output
ทำการสุ่มเรียกเมนูขึ้นมาแบบหลาย user เข้ามาพร้อมกัน	มี Response time และความเสถียรของ system ที่ดี ภายใต้ Condition Varying loads ได้
Scenario 11: Database with Duplicates	
Input	Output
“ไก่ทอด” Database:” KFC, Texas Chicken, Five Star Chicken, ไก่ทอดขนาดใหญ่, KFC, Texas Chicken, “	KFC, Texas Chicken, Five Star Chicken, ไก่ทอดขนาดใหญ่

2.3 [Optional] If the system needs to run against millions of users, What are points of concerns about system performance? How to improve performance from the above design?

Ans ผมคิดว่าสิ่งที่ต้องเป็นห่วงมากที่สุดคือ **Response time** ของแอปในขณะนั้นรวมถึงขีดจำกัดในการโหลดของ **Server** ผมคิดว่าหากเป็นรอบแรกๆ อาจจะไม่สามารถรับมือกับปัญหาได้ทัน ทำให้ **System** อาจจะมีอาการช้าหรือแย่ที่สุดคือระบบล่มไปเลย โดยวิธีที่ผมคิดได้ก็คือ เราอาจจะต้องทำระบบ **Queue** เพื่อให้คนเข้าไปใช้ระบบให้พอดีกับที่จะรับได้ แล้วหลังจากนั้น เราก็จะหาวิธีแก้ไขรวมถึงรับมือกับสถานการณ์ในอนาคตได้ คือการเก็บสถิติหรือข้อมูลว่าในช่วงไหนของเวลาหรือของวัน เดือน ปี ที่จะมีคนเข้ามาใช้งานมากที่สุด แล้วลอง **Load Testing** ในจำนวนที่ใกล้เคียงดูว่าสามารถรับได้มากแค่ไหน แล้วเมื่อใกล้ถึงช่วงเวลานั้นเราก็ทำการเพิ่มขนาดของ **Server** ให้รองรับได้มากขึ้นในขณะนั้นตามค่าของ **Load Testing** ที่เราได้ทำไป มาใช้ในการพัฒนา อาจมีการทำ **connection pooling** เพื่อลดอาการ **overhead** ลง แต่จาก **test strategy** ที่สร้างมาแล้วผมคิดว่าน่าจะสามารถรองรับคนได้เนื่องจากมีการใช้ **Performance test** ในการหาจุด **bottleneck** ของ **Server** ไว้แล้วครับ

2.4 What would you do as a feature/improvement to build a good search experience for users?

- เพิ่ม **Search Filters** เพื่อเพิ่มทางเลือกในการหาเมนูรวมถึงร้านที่สนใจ เช่น พื้นที่ใกล้เคียง ร้านที่มีรีวิวที่ดีและมีจำนวนการรีวิวที่เยอะ โดยผมคิดว่าหากมีการใช้ร่วมกับ **Personalized Recommendation** เพื่อแนะนำร้านอาหารหรือเมนูเฉพาะให้กับทางผู้ใช้งานด้วย ซึ่งจะมาจากพฤติกรรมในการเลือกร้าน, **search history** และอาหารที่ผู้ใช้งานชอบสั่ง จากที่ศึกษาอาจจะใช้ **A/B Testing** เข้ามาช่วยในการเปรียบเทียบข้อมูลของผู้ใช้ในหลายๆแบบ เพื่อเพิ่ม ประสิทธิภาพให้ดียิ่งขึ้นเพิ่มเติม เพิ่มการ **Query** ร้านค้าที่มีโปรโมชันขึ้นมาเป็นอันดับต้นๆ ผมคิดว่าจะช่วยให้ผู้ได้เลือกทานอาหารที่ราคาถูกลงครับ
- เพิ่ม **voice search** เพื่อเพิ่มทางเลือกหากผู้ใช้หรือลูกค้าไม่ต้องการพิมพ์ และควมมีการ **Manual testing** เพื่อให้แน่ใจว่าสามารถ **Response** ได้ตามที่เรากำหนด
- เพิ่มความเร็วของ **Response** ของทั้ง **searching** และ **application** เองโดยการ **testing system performance** ภายใต้การใช้งานของผู้ใช้ที่มีจำนวนแตกต่างกันหลายๆรอบเพื่อให้มั่นใจว่าสามารถใช้ได้จริง