Name: Leonardo Medina

Student Id: 619571

CS590 Final exam

1	
2	
3	
4	
5	
6	
7	
Total	

Question 1 [10 points] {5 minutes}

Give 6 different and valid issues/challenges we face when we implement a microservice architecture.

Answer:

- Service communication and Integration
  - Rest, gRPC, messaging

Challenge: Managing in-er-service communication serialization formats timeouts and network failure

Example: An slow or failed service call can cascade and effect multiple service

- Data managing and Consistency
- each microservice typically owns its own database
- challenge: Maintain data consistency across service without a shared schema
- solution: often involve eventual consistency or sagas ? which is complex to desing and debug

Deployment and DevOps Complexity

Microservice require automatization for building, testing, deploy and monitoring many independent microservice

Monitoring and Observability

with dozen or hundred of service, it becomes hard to trace request across the system.

Security management

Each microservice exposes endpoints that can be potential attacks point

team collaboration and Governance

If your answer does not fit, continue at the back of this page

Question 2 [15 points] {15 minutes}

Suppose you need to decide between a **component based monolith** (also called a modular monolith) or a **microservice architecture** for a certain application. For each of the requirements given below indicate which architectural style would you use to fulfill this requirement.

Your answer can be one of the following 3 possibilities:
1. component based monolith fits better
2. microservice architecture fits better
3. Both architectures can be used to fulfill this requirement.

a. It should be easy to add new functionality. 3

b. The architecture should be simple. 1

c. It should be easy to try new technology. 3

d. The performance should be very fast. 1

e. It should be easy to reuse existing functionality. 2

f. Deployment should be very fast. 2

g. It should be easy to scale different functionality differently. 2

h. It should be easy to use different types of databases. 2

i. The scope of the functionality is large. 2

j. We need multiple agile/scrum teams to implement and maintain the functionality. 2

If your answer does not fit, continue at the back of this page

Question 3 [15 points] {15 minutes}

Circle all statements that are correct:

1. A custom router in Spring Integration can return a list of Strings
2. A domain service object is a Spring bean annotated with @Service
3. A repository in a Spring Boot application is implemented as an interface.
4. In MongoDB you do not want data redundancy because this makes database actions slow.
5. In MongoDB you do not want joins between collections because they are very slow.
6. In a Service Oriented Architecture the registry functionality is implemented in the ESB
7. In a modular monolith the components communicate with each other using REST or messaging
8. A value object in domain driven design is immutable
9. DTO classes reside in the domain layer
10. Partitioning and replication can be used at the same time
11. In a relational database you first find the queries you need for your application and then design the tables based on these queries.
12. In a MongoDB database you first design the database collections and then write the queries in the application to get the required data.
13. In a Cassandra database you first find the queries you need for your application and then design the tables based on these queries.
14. The registry in a microservice architecture contains a client side load balancer.
15. The API gateway in a microservice architecture contains a client side load balancer.

If your answer does not fit, continue at the back of this page

Question 4 [10 points] {15 minutes}

Suppose you need to design an architecture for a webshop with the following architectural requirements:

1 • You need 1 agile devops team to implement the customers functionality → mongo
2 • You need 1 agile devops team to implement the orders functionality
3 • You need 1 agile devops team to implement the products functionality
4 • You need 1 agile devops team to implement the shoppingcart functionality
5 • You need 1 agile devops team to implement the payments functionality
6 • You need 1 agile devops team to implement the reviews functionality Mongo
7 • You need 1 agile devops team to implement the shipping functionality Redis.
8 • You need to be able to scale customers functionality differently than other functionality
- You need to be able to scale products functionality (and everything that is related to products) differently than other functionality
- You need strict consistency between the order data, payment data and shipping data

DBRMS

Explain clearly how you would design your architecture. Explain ALL architectural decisions that you need to make to fulfill the given requirements.For every architectural decision you make indicate how this decision solves one or more of the given requirements.

Answer:

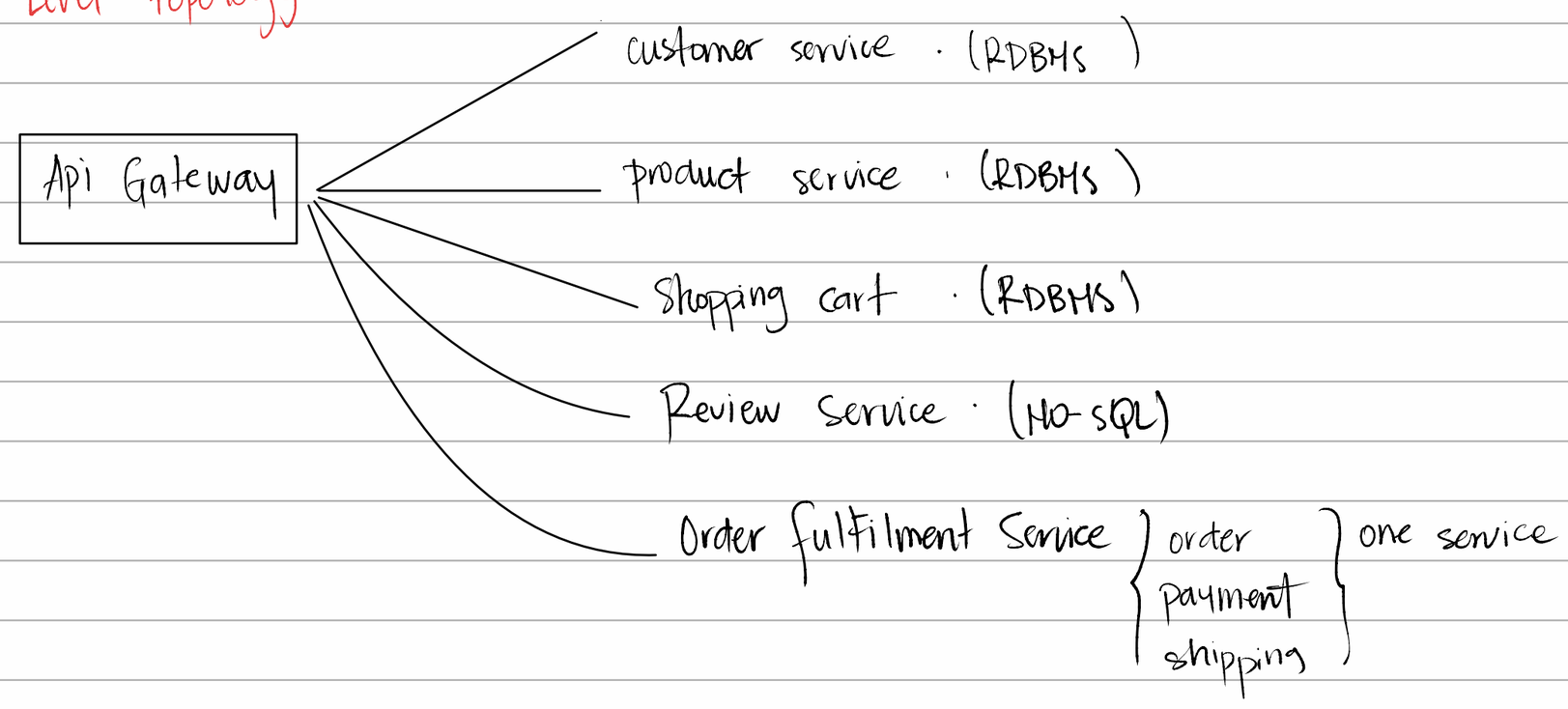
Order, payment require strict consistency in microservice

If your answer does not fit, continue at the back of this page

Bounded Context and teams

- Customer Service (team customer)
- Product / Catalog service team Product
- Shopping cart service team Shopping
- Review Service team Review
- Order Fulfillment Service (modular) – teams Orders, Payments, shipping collaborate inside one service codebase and one database to guarantee strict consistency

High Level topology



1:- create a deployable service for each service (5 service)
2:- strict consistency via single acid database for order + Payment + shipping
use a single relational data base eg. PostgreSQL
3:- order placed an event, payment capture event, shipping create an event to notify other service or future service for notification via email or sms using (Kafka, rabbitMQ)

4). Deploy each service in containers, with autoscaling policies.

- customer service scale on sing up / login
- Products / catalog scale independently and can also scale in read side (search products, CDN)
- Order fulfilment scale for checkout peaks

5) Product / catalog read model + search

search Index (Elastic Search)
CDN for assets
optionally cache (Redis) hot products

6) Shopping cart using Redis

7) A single Api Gateway (rate limiting, auth, routing)

8) Security: Payment handling inside order fulfillment service, behind the gateway with tokenized card and provider SDK

9) Observabifity – ELK

Question 5 [25 points] {50 minutes}

Suppose you need to design the class diagram of an airline application with the following requirements

Customers can browse flights. A flight has an origin and a destination. The origin and destination is recorded in the system with the attributes airport abbreviation, airport name and airport country. For a flight we also keep track of the flight number and the total duration of the flight.

For a flight we also need to know the plane name, number of seats on the plane and the maximum distance this plane can fly.

Flight are offered one or more times a day. For every flight offering we need to know the date and time of the flight and also the gate number and status of the gate.

Customers can buy tickets, and for a ticket they can choose the flight, the seat and also their meal preference. For every ticket we also need to keep track of the type of the ticket which can be economy class, first class or business class.

For a flight we also need to keep track of the seat availability of the flight.

Customers can create an account so they can login with their username and password.

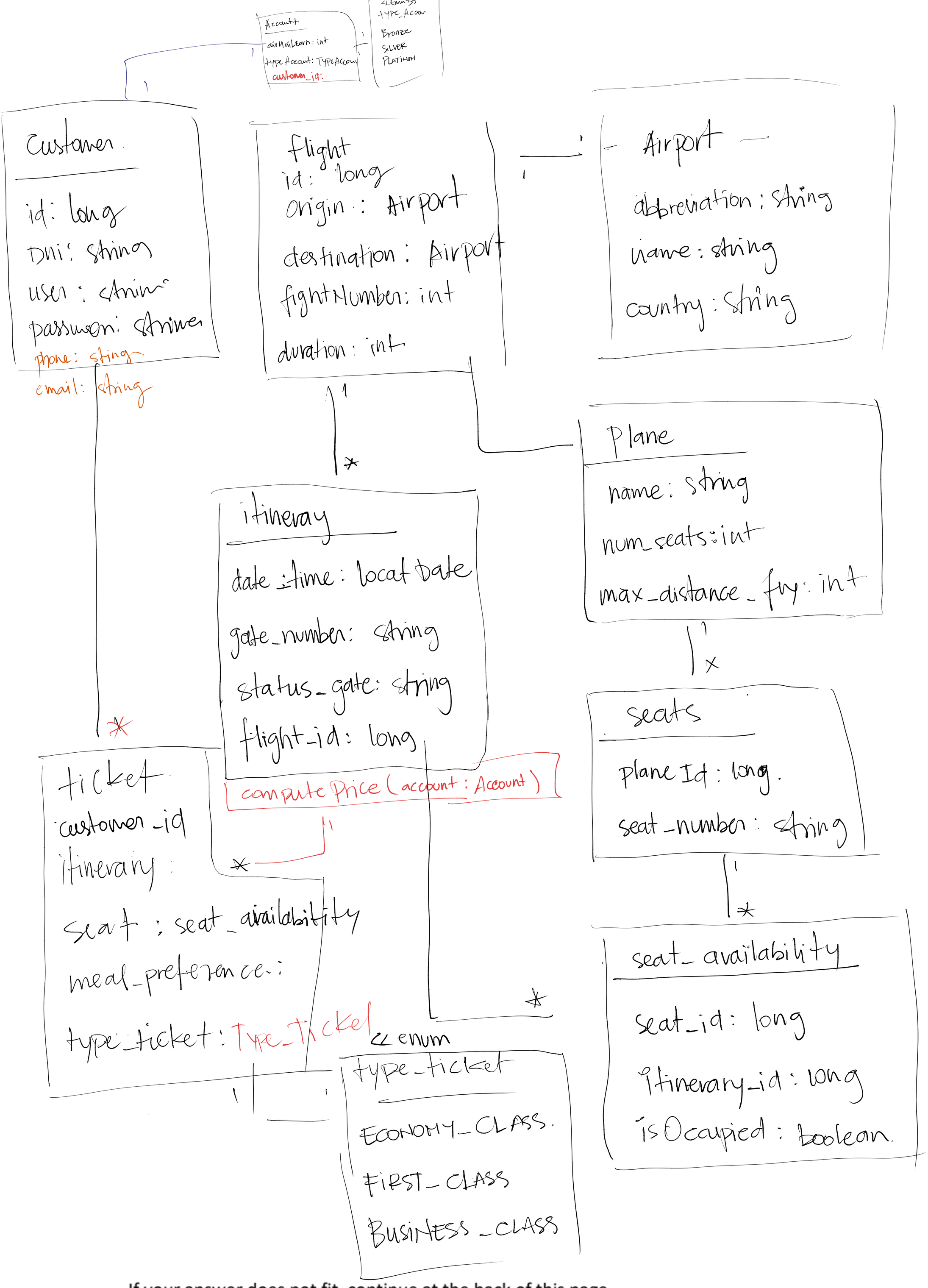
The account also keeps track of the airmiles this customer has earned. If the customer creates a new account, then this account is by default a bronze account. When the customer has enough airmiles, it is upgraded top a silver account. There is also a platinum account. For every customer we need to keep track of the name, phone number and email address.

When a customer searches for flights, the system dynamically computes the price for this flight. The prize for a flight depends on : Customer account type, date of the flight, time of the flight and seat availability of the flight and ticket type.

a. [10 points] Suppose we implement this application as one large monolith (without components). Draw the class diagram of the **domain classes** of this application. **For every domain class, indicate its type (entity, value object, etc). Important: only draw the domain classes. Make sure you show the classes, relationships, attributes and methods (if applicable).**

Draw your answer on the next page

If your answer does not fit, continue at the back of this page



If your answer does not fit, continue at the back of this page

b. **[25 points]** Suppose we implement this application as a monolith **with components**, also called a modular monolith. Draw the class diagrams of **all** components. **Important: draw ALL classes for every individual component. For every domain class, indicate its type (entity, value object, etc). For every domain class show the classes, relationships, attributes and methods (if applicable). For all NON domain classes only show the classes (not the relationships , methods and attributes)**

Answer

If your answer does not fit, continue at the back of this page

Question 6 [10 points] {10 minutes}

Suppose we have the following relational database tables:

courseNumber	courseName
CS590	Software Architecture
CS525	Advanced Software Development

studentid	firstName	lastName
5566	Frank	Brown
5532	John	Doe

studentid	courseNumber	grade
5566	CS590	A+
5566	CS525	B
5532	CS590	A-

Because the data we have does not fit on 1 node anymore we decided to replace our relational database with a Cassandra database. It is an important requirement that the application that uses this database should give the same fast performance that we had when we used the relational database.

The application has 2 main pages. The first page allows us to see all grades, including the course information, for a certain student. The second page allows us to see all grades, including the student information, from a certain course. **Show the structure of the table (or tables) that we need in Cassandra including the data in the table(s). For every table, specify what column is the partition key.**

Answer:

If your answer does not fit, continue at the back of this page

Question 7 [5 points] {10 minutes}

Describe how an **Enterprise Service Bus** (**ESB)** relates to one or more of the SCI principles you know. Your answer should be about 2 paragraphs. The number of points you get for this question depends on how well you explain the relationship between a **ESB** and the principles of SCI.

Answer:

If your answer does not fit, continue at the back of this page