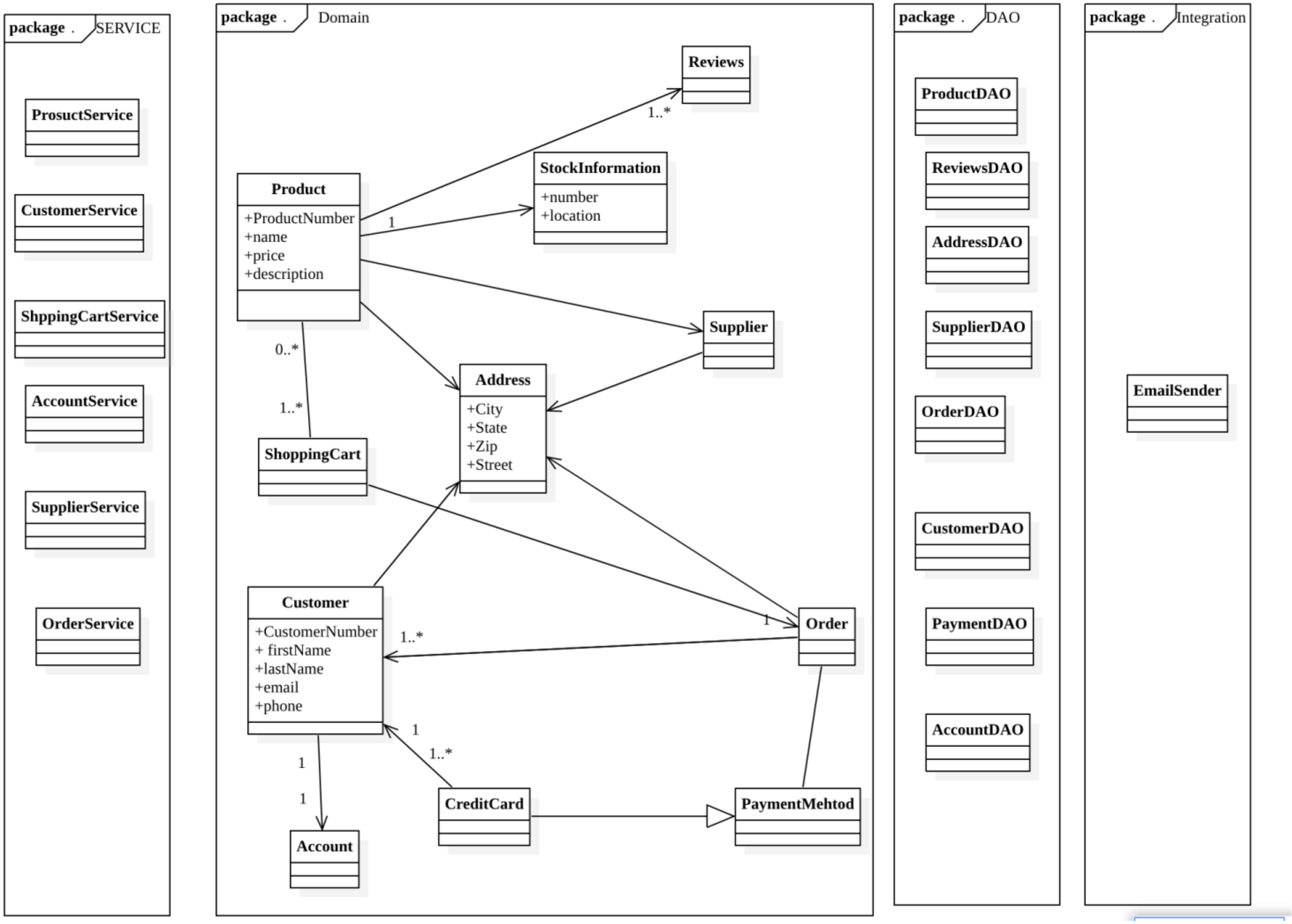
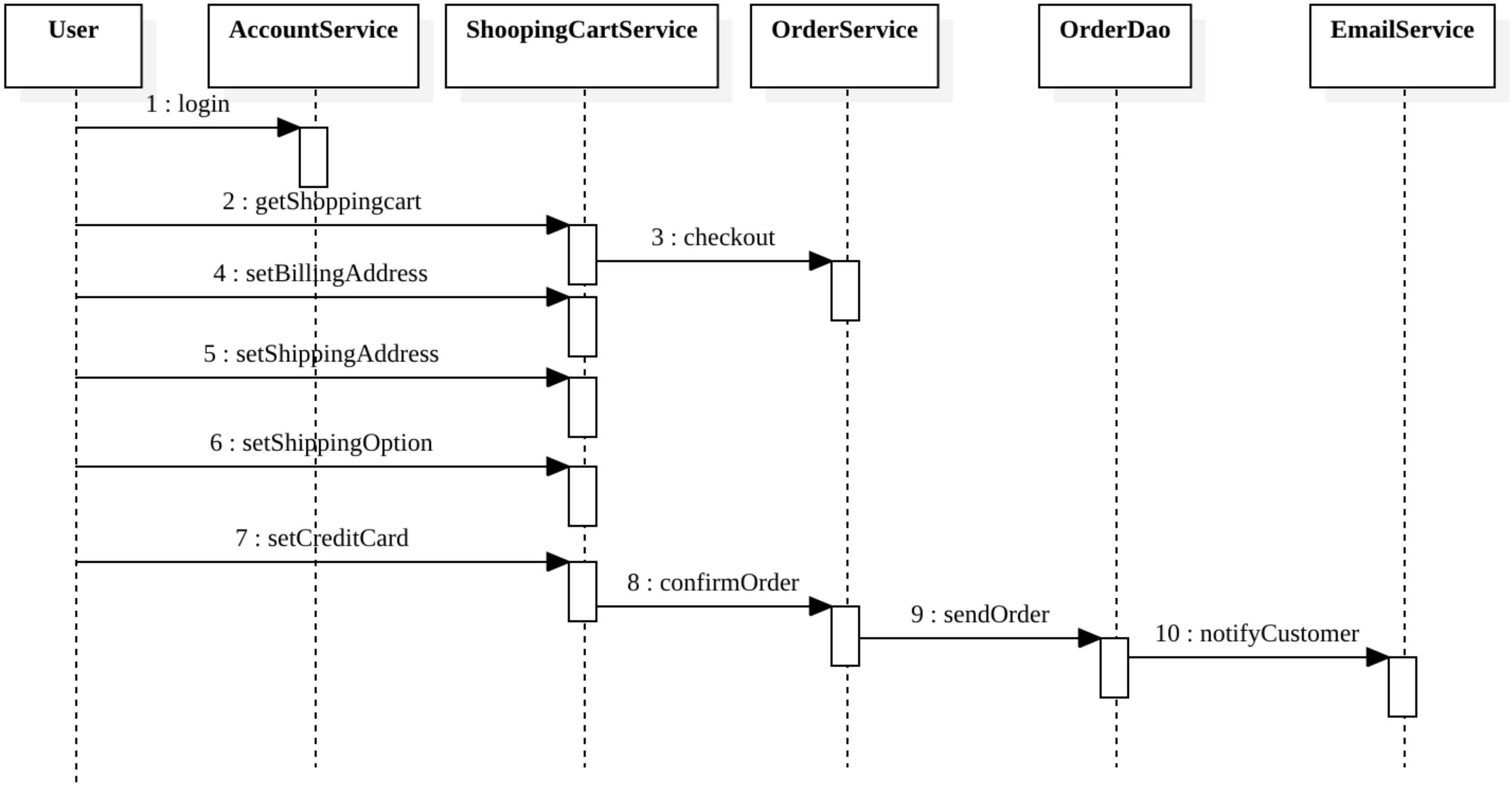
1 Application architecture

1.1 class diagram



1.2 Sequence diagram for placing an Order



2 How to store state

2.1 Client/ Browser

2.1.1 Advantages

- Applications that store state in the client such as React and Angular are highly responsive
- It's easy to scale applications that store state in the client

2.1.2 Disadvantages

- It's easy to lose information in case the browser/ client crashes
- It cannot be used as a reliable source to store data for long periods of time

2.2 Server

2.2.1 Advantages

- Information is secure and it's not easy to lose the information.

2.2.2 Disadvantages

- Its less responsive since state has to be updated in the server all the time
- Its less responsive compared to applications that's store state in the client/ browser

2.3 Database

2.3.1 Advantages

- Can be used as a reliable source to store data required after a long period of time.
- It's easy to do scaling compared to when we store state in the server
- There is reliability on the information when state is stored in the server

2.3.2 Disadvantages

- High latency and thus low performance
- Its is less responsive.

3 Integration possibilities

3.1 RMI - Remote Method Calls

RMI allows an object residing in one java virtual machine to access/invoke an object running on another java virtual machine.

3.1.1 Advantages

- Server-side implementation can be changed without knowledge of client side

3.1.2 Disadvantages

- RMI requires that serialization and deserialization of objects which increases overhead costs
- RMI works for only Java to java object calls

3.1.3 Usage

- Not good to be used

3.2 Messaging (JMS – Java messaging services)

The communication is in form of a message between different services on the network.

3.2.1 Advantages

- Asynchronous communication: JMS uses asynchronous communication by using a queue for the messages and thus this can lead to high throughput
- High reliability: Messages sent will always be delivered to the consumer without loss of data
- Loose coupling between systems instead of the systems sharing a common database
- JMS API is very easy
- JMS does efficient load balancing in case of very many messages

3.2.2 Disadvantages

- There is a need for a middle ware for the messaging service
- The middle ware can be a single point of failure

3.2.3 Usage

- Can be applied in cases where we need the communication to be asynchronous

3.3 SOAP- Simple Object Access Protocol

3.3.1 Advantages

- Has a standard HTTP protocol that makes it easier to communicate across firewalls and proxies.
- SOAP is highly secure and highly standardized

3.3.2 Disadvantages

- Has a poor performance compared to REST.

- It is more complex.
- SOAP is less flexible

3.3.3 Usage

- Where we need more security

3.4 REST - Representational State Transfer

3.4.1 Advantages

- Supports greater variety of data formats such as XML, JSON, HTTP among others
- Gives better support for browser clients
- Faster and uses less bandwidth

3.4.2 Disadvantages

- Less secure

3.4.3 Usage

- Everywhere

3.5 Serialized objects over HTTP

3.5.1 Advantages

- Its easy
- Supports secure java computing

3.5.2 Disadvantages

- Works for only Java to java
- It has large overheads and cannot be used for very large objects

3.5.3 Usage

- Used for java to Java applications

3.6 Database integration

3.6.1 Advantages

- Its easy

3.6.2 Disadvantages

- High coupling

3.6.3 Usage

- Not advisable

3.7 File based integration

3.7.1 Advantages

- Its easy
- Its reliable

3.7.2 Disadvantages

- Scalability is hard
- High coupling

3.7.3 Usage

- Transferring large files

4 Distributed systems

4.1 Advantages of distributed systems

- Improved performance since each system can be run with its own independent resources thus improving the performance of the overall system
- Easy scaling: Since the processes are distributed, it's easy to scale each process depending on the need and requirements of that particular process
- Transparency: Middleware improves transparency because processes communicating to each other need not to be aware of each other
- It is easy to use shared resources while using distributed systems.

4.2 Disadvantages

- Its more complex to plan and implement distributed systems
- Distributed services are more expensive due to the fact that each process needs its own resources in terms of hardware
- Distributed systems present security threats.
- Fault isolation in distributed systems is also harder.
- There are a lot of network calls in distributed systems and this can also impact the performance of distributed systems.