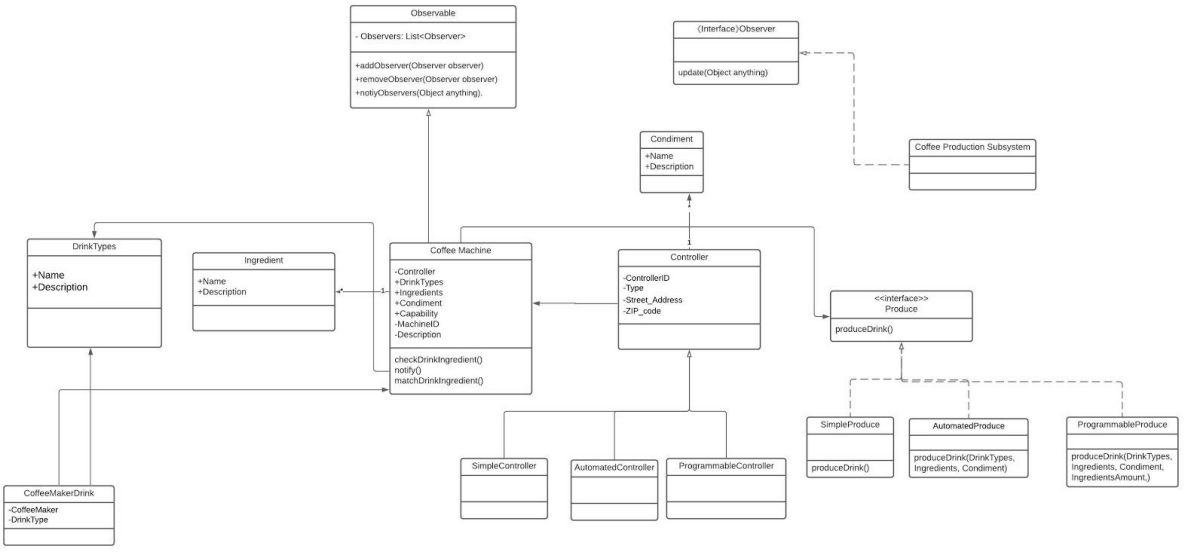
Team name/number: T5
List of team members: Doris Chen, Sybil Chen, Jiaqi Shen, Jiafan Lin
Assignment title: *T5-M2P2-ProjectArchitecture*
Date submitted: 1/28/2020

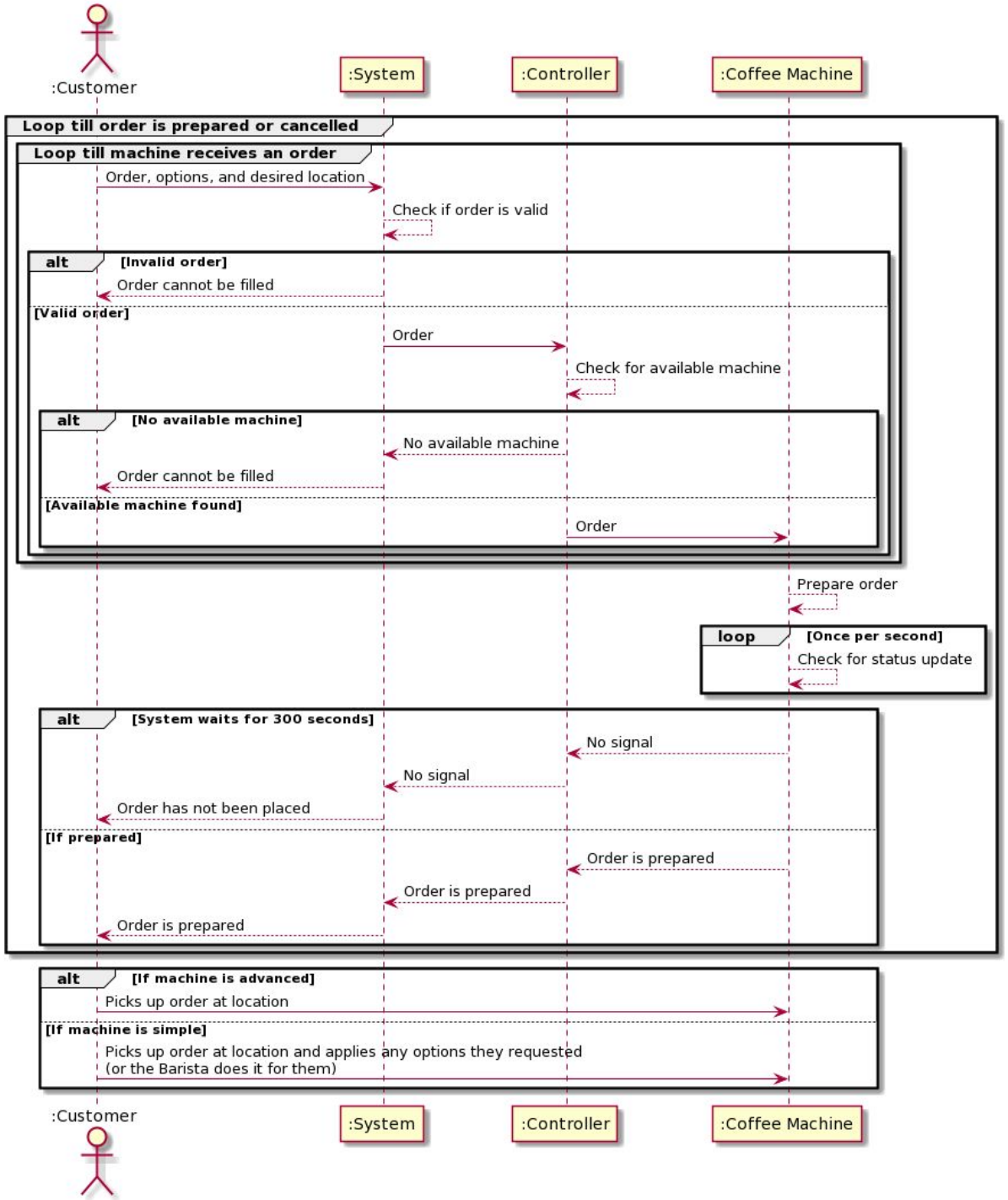
DCD



Strategy Design Patterns:
Define each algorithm/policy/behavior in a separate class with a common interface.
We make Controller an abstract class, and there is an interface called “Produce” for producing drinks for Controller. Three different controller classes with simple, automated, and programmable capabilities can choose their required methods corresponding to capability from inheritances of the “Produce” interface.

Observer Design Patterns:
Subject: Coffee Machine
Observer: Coffee Production System
The observer pattern is an elegant, reusable solution
The use of interfaces ensures that (changeable) concrete classes only refer to (stable) interfaces.
Gets right “knows about” relationships.
Localization of change (low coupling).
Define an observer interface that the subject objects can implement. Observers register with the subject-object. The Subject object notifies all of its observers.
The coffee machine is observable, and its change in storage can be observed. The observer is the Coffee production subsystem which can record and extract coffee data from the database, and the observer does not apply to our design at this step.

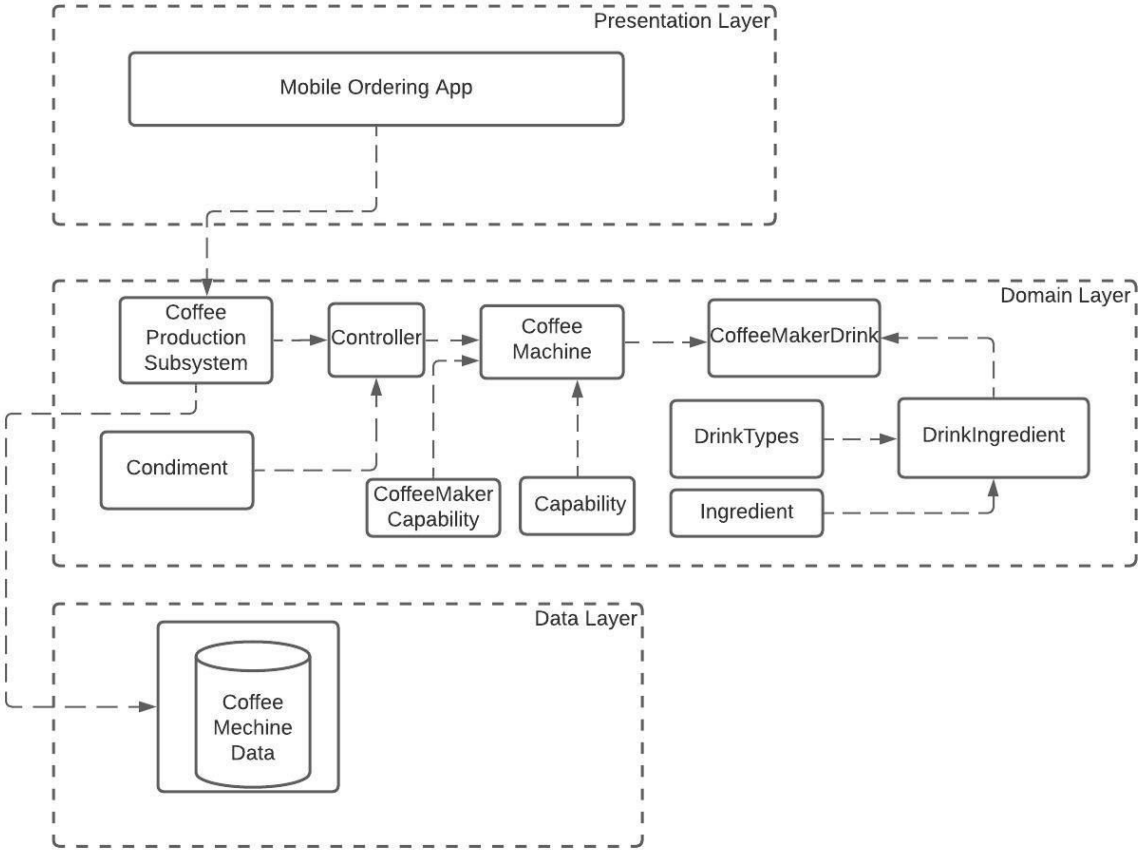
SSD



Considering the large similarity between UC1 and UC2, we decided to combine the two flows into one sequence diagram. The difference appears in the last alt loop.

Note: The frequency of checking for an update and the waiting time for an update response are tentative and subjective to any change in requirements.

Assemble all the major classes into the three Principle Layer



Sample Input (Input data in the console)

UC1

```
System: Start order
Simple Machine Press 1
Advanced Machine Press 2
1
Please enter your street address
Fifth Avenue
Please enter your zip address
47803
Please enter your drink type
Americano
Simple
Size in import order 0
orderID:2 drink: Americano street: Fifth Avenue ZIP: 47803
System: Processing
Your coffee has been prepared with your desired options.
-----------------------------------------
System: Start order
Simple Machine Press 1
Advanced Machine Press 2
```

UC2

```
-----------------------------------------
System: Start order
Simple Machine Press 1
Advanced Machine Press 2
2
Please enter your street address
Baker Street 211
Please enter your zip address
83051
Please enter your drink type
Cappuccino
Automated
Please enter your condiment
sugar
Please enter quantity
2
condiment: sugar
quantity: 2
Do you still want to add condiment? (Y=Yes/N=No)
N
Size in ask for order 1
Order condimentsugarSize in import order 1
sugarorderID:1 drink: Cappuccino street: Baker Street 211 ZIP: 83051
System: Processing
Your coffee has been prepared with your desired options.
-----------------------------------------
System: Start order
Simple Machine Press 1
Advanced Machine Press 2
```

Sample Output for UC 1&2

```
Simlulated Test for ordering coffee......

drinkResponse:
orderID: 1
status: 0
orderID: 2
status: 0

OrderInput:
condiment: [{"qty":2,"name":"sugar"}]
address: {"ZIP":"1","street":"Baker Street 211"}
orderId: 1

condiment: []
address: {"ZIP":"2","street":"Fifth Avenue"}
orderId: 2


Command:
command: {"Options":[{"Name":"sugar"},{"qty":2}],"controller_id":1,"orderID":1,"coffee_machine_id":1,"DrinkName":"Cappuccino","Requesttype":"Automated"}
command: {"Options":null,"controller_id":2,"orderID":2,"coffee_machine_id":10,"DrinkName":"Americano","Requesttype":"Simple"}

AppResponse:
status-message: Your coffee has been prepared with your desired options.
orderID: 1
coffee_machine_id: 1
status: 0

status-message: Your coffee has been prepared with your desired options.
orderID: 2
coffee_machine_id: 10
status: 0
```