CS3342 ASM1 Solution

Q1A:

Agile process model is a software development approach, which break tasks into smaller tasks. It also can help removing the unnecessary tasks in order to reduce the time waste on those operations. The steps of Agile process model are gathering requirement, Designing requirement, Construction, Testing, Deployment and Feedback. Hence, the aim of the Agile process model is to perform a quick execution of the project (GfG, 2024).

Regarding to the benefits of the Agile process model, it can provide a superior quality product. Since the steps involve in the Agile process model include testing and it is an iterative process, the product will keep improving again and again the process (9 Key Benefits of Using the Agile Methodology, 2023). Also, Agile process model allows for flexibility and adaptability throughout the development process. Changes in requirements can be accommodated easily, enabling quick responses to feedback and market demands. It also preforms customer collaboration. Agile promotes active customer involvement throughout the development process. Regular feedback and collaboration enable better alignment with customer needs and expectations. Agile's incremental and iterative nature enables faster delivery of functional software components, leading to quicker time-to-market and early ROI. It provides faster Time-to-Market. Agile also emphasizes self-organizing, cross-functional teams that collaborate closely. This approach fosters better communication, knowledge sharing, and collective ownership of the project.

Regarding to the challenge of the Agile process model, Agile projects often face evolving requirements, making it crucial to manage scope and expectations. Resource allocation can be challenging, especially when juggling multiple projects or dispersed teams. Agile's focus on working software over comprehensive documentation may pose difficulties for organizations with regulatory requirements. Active stakeholder involvement is vital for Agile success but can be hindered by competing priorities or limited availability. Transitioning to Agile requires a learning curve for team members, and managing distributed teams requires effective coordination and communication. Overcoming organizational resistance to change is essential for successful Agile implementation. By proactively addressing these challenges through proper planning, training, and continuous improvement, Agile teams can enhance their chances of success.

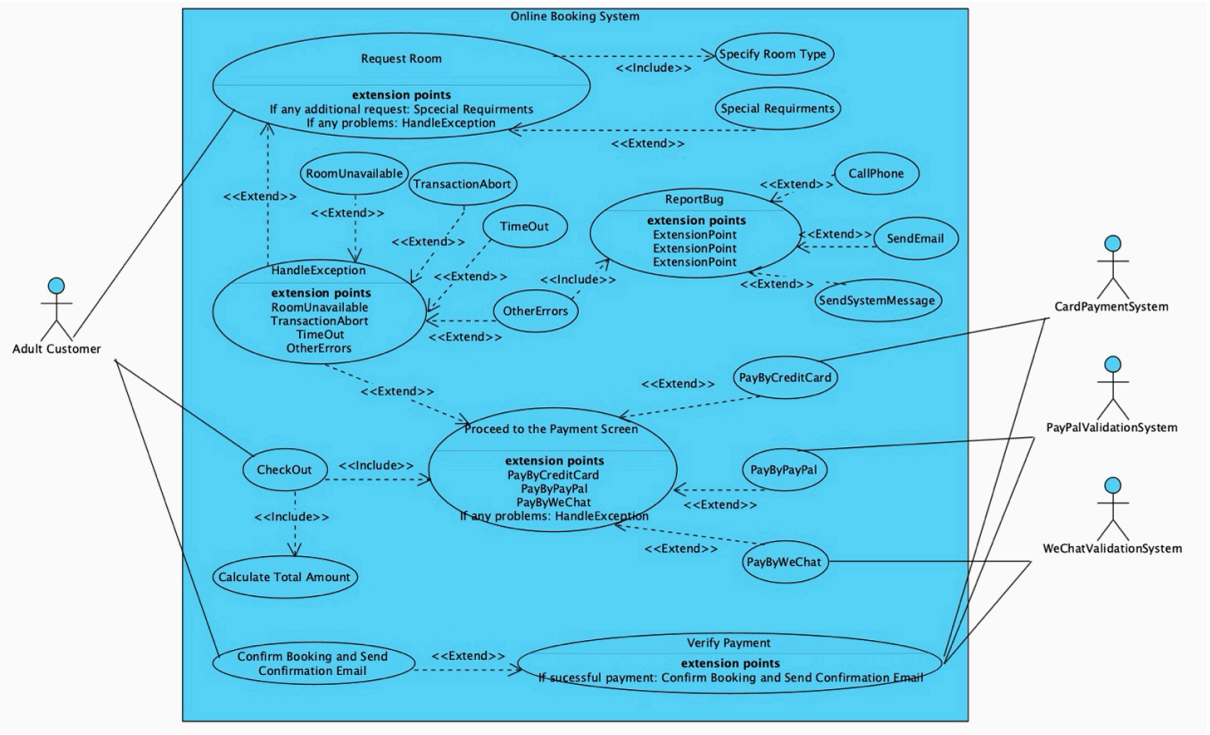
Comparing the Agile software process model with the traditional model l such as the waterfall model, the Agile model includes flexibility, adaptability and customer collaboration. On the other hand, the waterfall model follows a linear approach, which the model moves on to the next phase only when it finish its current phase. This makes projects become more stable and well-defined comparing with Agile software process model. In terms of managing requirements, Agile good at handling uncertainty and improving based on customer needs. It allows for continuous feedback and improvement, which ensure the final product meet customer expectations. On the other hand, the Waterfall model requires detailed planning to define requirements before the project begins. While this can provide clarity and structure, it may struggle to accommodate changes once the project is running. Agile ensure regular value delivery and quick time-to-market by iterative property. Furthermore, Agile has a collaborative function, giving effective communication of the project. In contrast, the waterfall model may take longer time to produce a complete product, as each phase must be completed before progress can be made. It has less customer involvement in development, which may limit feedback, and lead to a mismatch with customer expectations.

Reference: GfG. (2024, January 10). Agile development models Software engineering. GeeksforGeeks. https://www.geeksforgeeks.org/software-engineering-agile-development-models/ 9 Key benefits of using the agile methodology. (2023, December 21). https://kissflow.com/project/agile/benefits-of-agile/

Q1B:

Variable	Role	(10 Marks)
student_sc	Constant	(1 Mark)
total	Gatherer	(1 Mark)
average	Transformation	(1 Mark)
i	Stepper	(1 Mark)
student_num	Constant	(1 Mark)
max_sc	Most-recent holder	(1 Mark)
id	Temporary	(1 Mark)
neg	One-way flag	(1 Mark)
fail	Gatherer	(1 Mark)
sc	Temporary	(1 Mark)

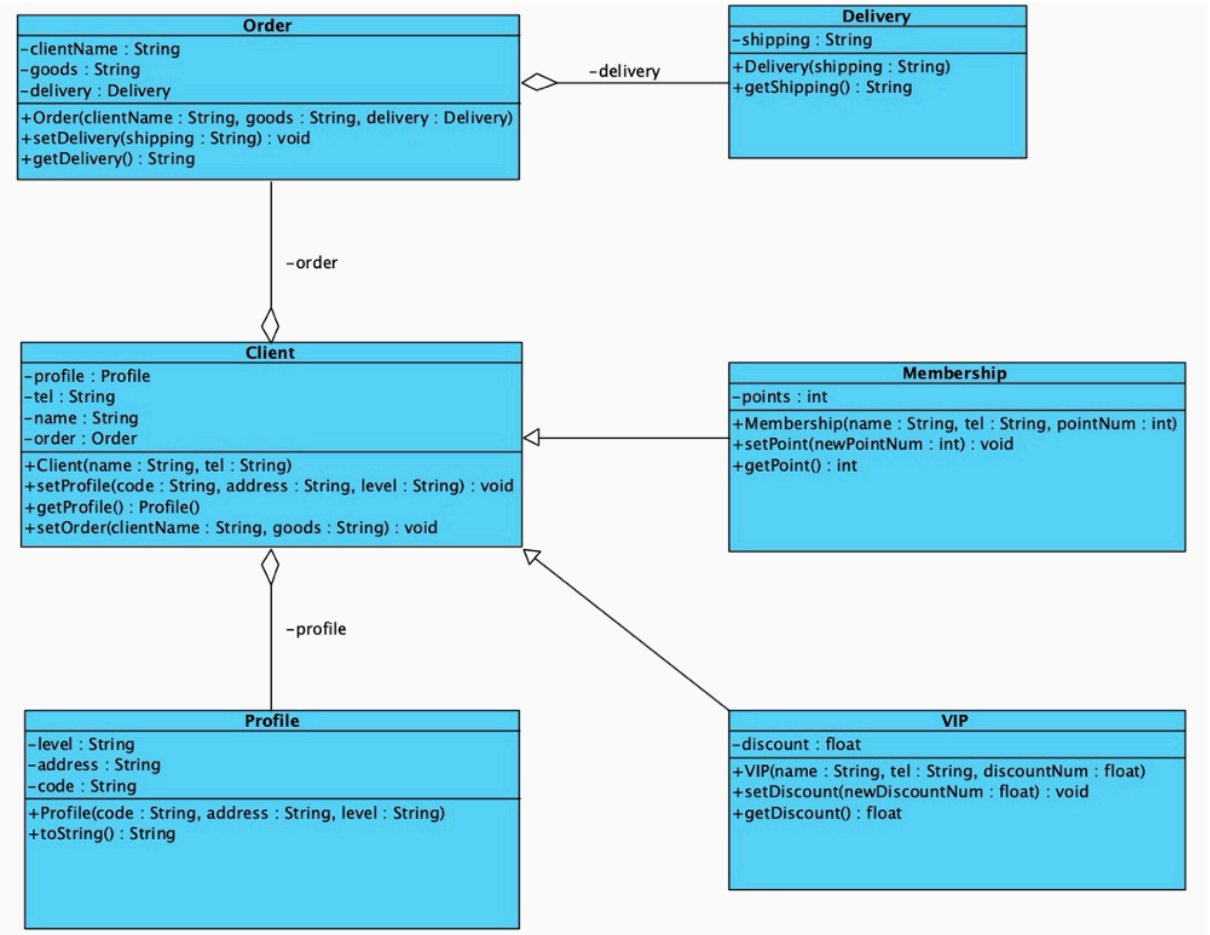
Q2A:


Online Booking System
Request Room
extension points
If any additional request: Spcecial Requirments
If any problems: HandleException
Specify Room Type
<<Include>>
Special Requirments
<<Extend>>
RoomUnavailable
TransactionAbort
TimeOut
HandleException
extension points
RoomUnavailable
TransactionAbort
TimeOut
OtherErrors
OtherErrors
ReportBug
extension points
ExtensionPoint
ExtensionPoint
ExtensionPoint
CallPhone
SendEmail
SendSystemMessage
Adult Customer
CardPaymentSystem
PayPalValidationSystem
WeChatValidationSystem
PayByCreditCard
Proceed to the Payment Screen
extension points
PayByCreditCard
PayByPayPal
PayByWeChat
If any problems: HandleException
CheckOut
PayByPayPal
PayByWeChat
Calculate Total Amount
Confirm Booking and Send Confirmation Email
Verify Payment
extension points
If sucessful payment: Confirm Booking and Send Confirmation Email

Q2B:

Use Case Name:	**Checkout**	
Actor(s):	**Customer, CardPaymentSystem, PayPalValidationSystem, WeChatValidationSystem**	
Description:	This use case describes the process of a customer completing the checkout for the room selected. On completion, the system will confirm the booking and send a confirmation email to the customer.	
Reference ID:	HK-ROOM-BOOKING-1.0	
Typical course of events:	**Actor Action**	**System Response**
	Step 1: This use case is initiated when a customer completes the checkout to be processed.	
		Step 2: The system executes the use case of "proceed to the payment screen".
		Step 3: If the customer chooses a payment method, customer proceed to the payment screen of the chosen validation system.
	Step 4: This use case is initiated when a validation system verifies the costumer's payment to be processed.	
		Step 5: The system will check whether the payment is successful or not.
		Step 6: If the payment is successful, the booking will be confirmed, and the confirmation email will be sent to the customer.
Alternative course of events:	Step5a: Customer enter to the PayByCreditCard user case. Step5b: Customer enter to the PayByPayPal user case. Step5c: Customer enter to the PayByWeChat user case.	
Precondition:	Only customer can check out.	

Q3A:


Order
-clientName : String
-goods : String
-delivery : Delivery
+Order(clientName : String, goods : String, delivery : Delivery)
+setDelivery(shipping : String) : void
+getDelivery() : String
-delivery
Delivery
-shipping : String
+Delivery(shipping : String)
+getShipping() : String
-order
Client
-profile : Profile
-tel : String
-name : String
-order : Order
+Client(name : String, tel : String)
+setProfile(code : String, address : String, level : String) : void
+getProfile() : Profile()
+setOrder(clientName : String, goods : String) : void
Membership
-points : int
+Membership(name : String, tel : String, pointNum : int)
+setPoint(newPointNum : int) : void
+getPoint() : int
-profile
Profile
-level : String
-address : String
-code : String
+Profile(code : String, address : String, level : String)
+toString() : String
VIP
-discount : float
+VIP(name : String, tel : String, discountNum : float)
+setDiscount(newDiscountNum : float) : void
+getDiscount() : float

Q3B:

Your solution/justification:

Interface should be used in Delivery. Both Standard and Premium can share the same function but with different operation.

Final Class Diagram:

< Screen Capture: Place your diagram here>