20122161 832002117 Hanlin CAI. No. 01.04 Date P1.

Q1.

Generic process framework :

① Communication → ② Planning → ③ Modeling

→ ④ Construction → ⑤ Deployment.

Understanding :

① Communication : includes project initiation and requirement gathering. Good communication can make the whole project advanced smoothly.

② Planning : includes : includes project estimating, scheduling and tracking. A suitable development plan is the key to success.

③ Modeling : aims to ~~ardy~~ analyze and design the software. Generally speaking, there are four main method for modeling, such as Scenario-based, class-based modeling.

④ Construction : mainly includes coding and testing, it is the main process for software development.

⑤ Deployment : includes software delivery, software support and the feedback from the client. The deployment process is the final step of generic process framework.

Q1.-2. sol

Requirement modeling method:

① Scenario-based Modeling method

Scenario-based method is mainly based on the software used scenario and the corresponding use cases. This method is intutive for the developer to code and test. However, this method is highly depending on the communication between the developer and clients(user), hence, the company should improve their connection with their client once they adapt this method.

② Class-based method:

Class-based method is highly ~~recommand~~ welcome by developer because of its intutive and efficient expressions. This method can illustrate the relationship between different classes ~~an~~ (or subclasses), which can make developments easier and more reliable.

Q1·3 sol.

① Cohesion is the tightness of each element in a module, which describe the "sigle-mindedness" of a component.

② Coupling is the degree of correlation between module, which is a qualitative measure of the closeness between different module.

③ The good componet-level design is resulted as high cohesion and low coupling, which can greatly enhance the reusability and portability of the module.

Q1.4 Task network.

(i) Paths = 1 → 2 → 3 → 4 → 6
1 → 2 → 3 → 5 → 6
Hence, there are 2 paths on the network.

(ii) Paths ①: 1 → 2 → 3 → 4 → 6
Length of paths ① is: A+B+C+E = 10

Paths ②: 1 → 2 → 3 → 5 → 6
Length of path ② is: A+B+D+F = 16

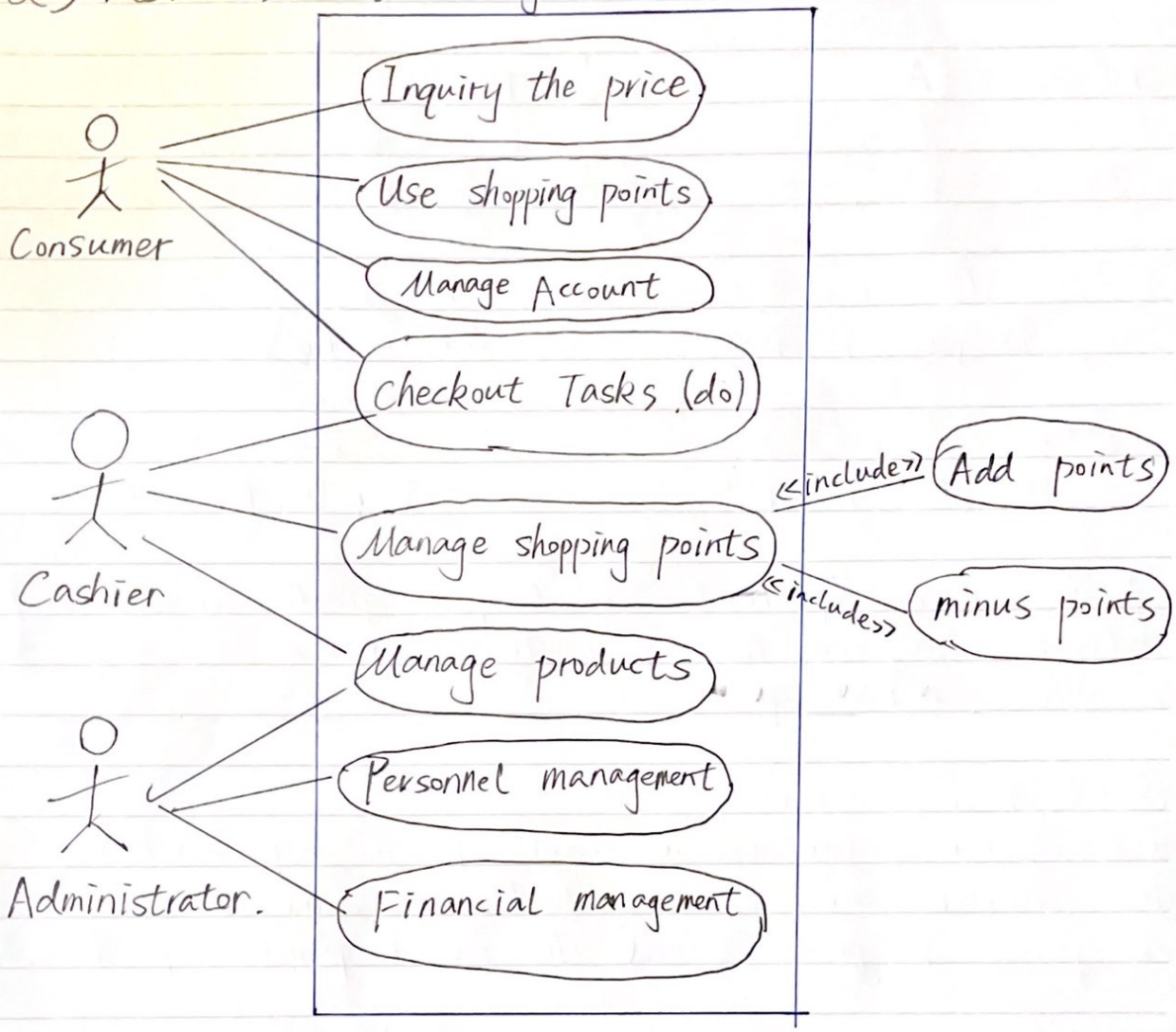
(iii) The critical path is the longest path in the network diagram, hence, path ②: 1→2→3→5→6 is the critical path.

Q2.5(a) There actors:
① Consumer: who is the client of the supermarket, the consumer can use the checkout system to inquiry the price of products and the rewards points of them.

② Cashier: who is the main actor to ~~uti~~ utilize the checkout system. The cashier should use the system to finish the cheakout task. ~~or~~

③ System Administrator: who is the manager or &tech-worker of the supermarket. Administrator can modify the information of the product.

Q5. (b) use case diagram:



Q2 (c)

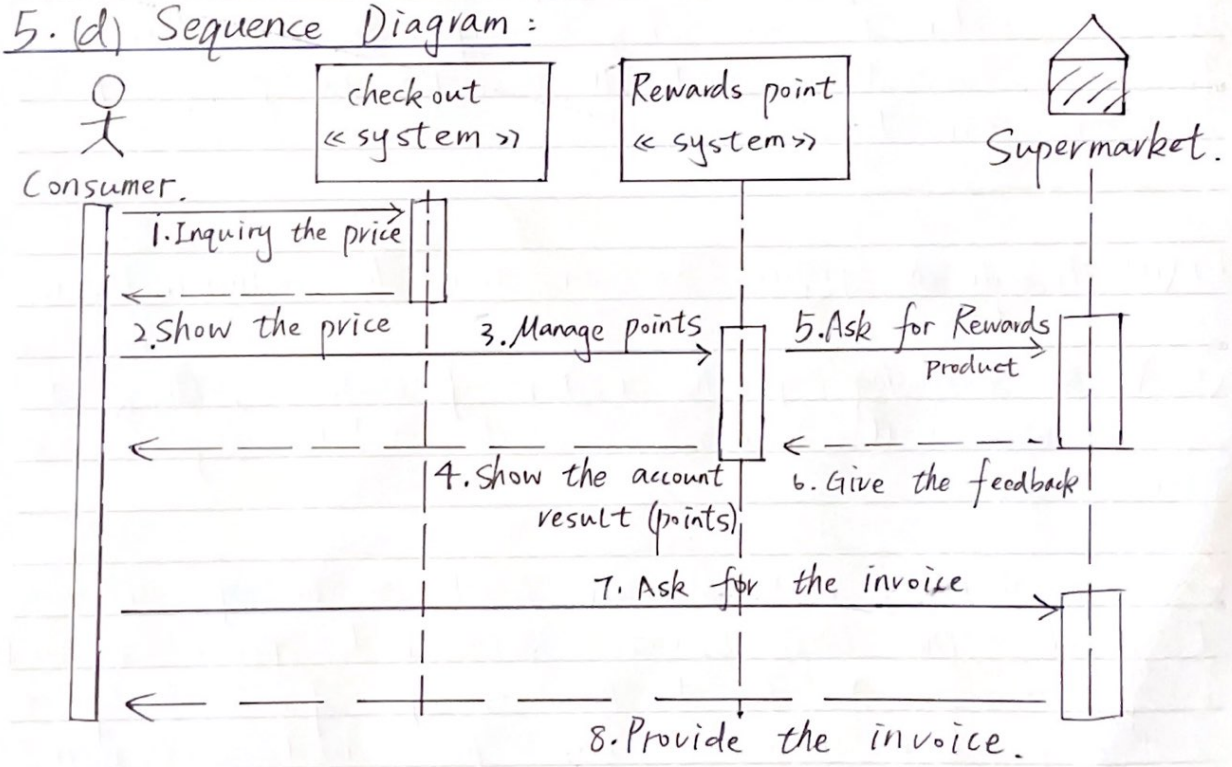
Q2. 5. (c) Normal Scenario:

① Consumer : will use the checkout system to inquiry the price of product, and utilize the shopping rewards points system to manage their rewards points.

② Cashier : will use the checkout system to finish the check-out tasks, they can also use the system to manage the rewards points of client (Add or Minus points).

③ System Adminstrator : can use checkout system to manage the personnel information or product information.

5. (d) Sequence Diagram :



No.

Date

Q3. Sol.

(a) Formula (1) $C(p1) > C(p2)$ then $E(1) > E(2)$

This formula means if the complexity of task p1 is higher than p2, then the workload required to p1 is also higher than p2.

Formula (2) means the complexity of the whole class is higher than its subclasses, then the workload can be reduce if we decomposed the task into smaller blocks.

(b) (i) If we divide the software without limitation, the cost and burden of integration and communication will increase. Hence, the cost of the whole project will increase, it may be even harder to solve these multiple subproblems. Therefore, we need to find the best balance point of modularization degree.

(b) (ii) Through the figure, we can draw the conclusion below:

① As the increasing of the number of modules, the development cost of software will decrease, but the module integration cost will increase.

② The number of modules can reach the optimal number to realize the best balance between development cost and integration cost. At the end, the cost of software development cost may not be minumum, but the whole cost will be minumum.

Q4. 1 (a) Test.

(i) Unit : refers to the inspection and verification of the smallest testable unit.

(ii) Integration : refers to assembling the tested unit module into a system or a subsystem and then test. It is the testing process of small increment.

(iii) White-Box testing : examine the process detail, and then test the logical path and the collaboration between component.

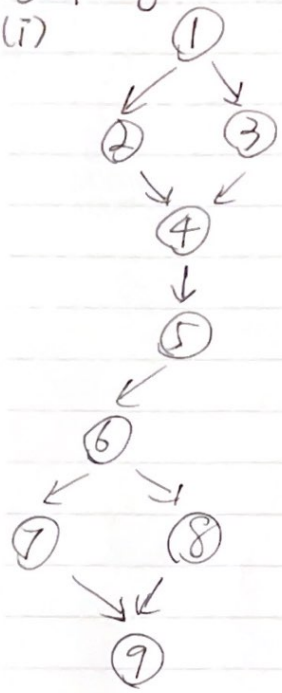
(iv) Black-Box testing : the test is performed at the interface to check the functional asept of the software without knowing the internal structure.

(v) Validation test : is a set of tests determining conformity to software requirement. It includes test plan, test cases, specific cases and documentation.

new.

Q4. 8.

(i)



(ii) Basic Path:

① 1 → 3 → 4 → 5 → 6 → 8 → 9

② 1 → 3 → 4 → 5 → 6 → 7 → 9

③ 1 → 3 → 4 → 5 → 7 → 9

④ 1 → 2 → 4 → 5 → 7 → 9

(iii)

[(0, -5)	(0, 5, 1)]
[(0, -15)	(0, 15, 0)]
[(0, 1)	(0, -1, 0)]
[(0, 0)	(0, -1, 0)]