

Homework #2 solutions

- 6.4 Do the following for the Study Abroad Management System (SAMS) presented in Chapter 4:**
- Identify the type of system and briefly justify your answer.**
 - Identify an architectural style and produce an architectural design for the system.**
 - Specify the functionality and interface for each of the subsystems and components in the architectural design.**
 - Discuss which software design principles are applied in the design, how they are applied, and the benefits of applying each of the principles. Also point out the potential problems associated with the application of the design principles, if any.**

Author Solution: The answer to this exercise is very similar to the calendar system except that the contents are different. Figure 6.4(b) of the textbook gives the general architectural design, which can be adapted for SAMS.

Professor comment: By similarity...

- This type of system is an Interactive System because it meets the criteria for that type of system as listed on page 144. A case could be made for at least part of the system (the database subsystem) to be classed as an Object-Persistence System, but this would not be entirely complete.*
- The architectural style is client-server, because the system provides services to a number of clients and meets the architectural description from page 147. Figure 6.5 on page 150 is typical of this type of system – merely replace the words "Flight Information System" in the «server» box with the acronym "SAMS" and replace the words "Passenger Check-in" in the «client» boxes with the word "applicant".*
- Subsystems and their functionality and interfaces:*
 - + *client subsystem:* web browser
 - + *server subsystem:* server computer which provides connections to multiple clients, and access to a database of overseas exchange study programs; the database must be searchable, which implies the clients' requests must be translated into a form that the database engine can understand and from which queries can be derived*Components and their functionality and interfaces:*
 - + *client subsystem:* desktop or laptop computer or tablet computer or web-enabled mobile phone
 - + *server subsystem:* server computer, web server software, database management system, several other software technologies are possible to aid in web implementation; further, requirements state that the application process be handled on line, so a method of translating applications to quantifiable and qualitative data for evaluation must be implemented as part of the system.
- Software design principles discussion:*

Although it is a good idea to consider all six of the software design principles (as listed on pp 161 – 166), several of them can be singled out for this application. First, low coupling is important to ensure that any system component does only the job for which it is best suited. This concept also plays into the "separation of concerns" principle. The web-based model is excellent for this reason. Second, the

high cohesion principle applies when using a database-driven architecture, as does the information hiding concept. Both of these ideas are necessary to help ensure the integrity of the database and to help maintain the speed and accuracy of queries so that the user gets a "snappy" response when searching for overseas programs.

7.4 Do the following for the Car Rental system (CRS) given in Appendix D.1:

- a. **Derive use cases from the requirements you produced in exercise 4.8 [Instructor note: you are not required to actually do exercise 4.8; only turn in the use cases.]**
- b. **Construct a requirements-use case traceability matrix.**
- c. **Specify the high-level use cases.**
- d. **Draw use case diagrams.**

Author Solution: A partial solution is given here. Example use cases that can be derived for the car rental system are (actor and system are omitted):

- UC1. Create Account (a solution may or may not include this use case)
- UC2. Search for a Car
- UC3. Reserve a Car
- UC4. Update a Reservation
- UC5. Delete a Reservation
- UC6. Checkout a Reserved Car
- UC7. Make Block Reservation (this could be merged with UC3)
- UC8. Process Reservation
- UC9. Return a Car
- UC10. Pay Invoice
- UC11. Create Rental Contract
- UC12. View Reservations
- UC13. View Invoices

B. The requirements-use case traceability matrix depends on the requirements. That is, if a use case is derived from a requirement, then the entry for the requirement and use case is checked. The solution to this part is not difficult and is omitted.

C. The following are some of the high level use cases:

UC2. Search for a Car

TUCBW the customer clicks the Search Car link on any of the CRS Web pages.

TUCEW the customer sees a list of cars that satisfy her/his search criteria.

UC3. Reserve a Car

TUCBW the customer

(1) clicks the Reserve a Car link on any of the CRS Web pages, or

(2) checks the cars s/he wants to reserve on the list of cars displayed by the Search for a Car use case and clicks the Reserve link on that page.

TUCEW the customer sees a message stating that the reservation is completed successfully.

UC4. Update a Reservation

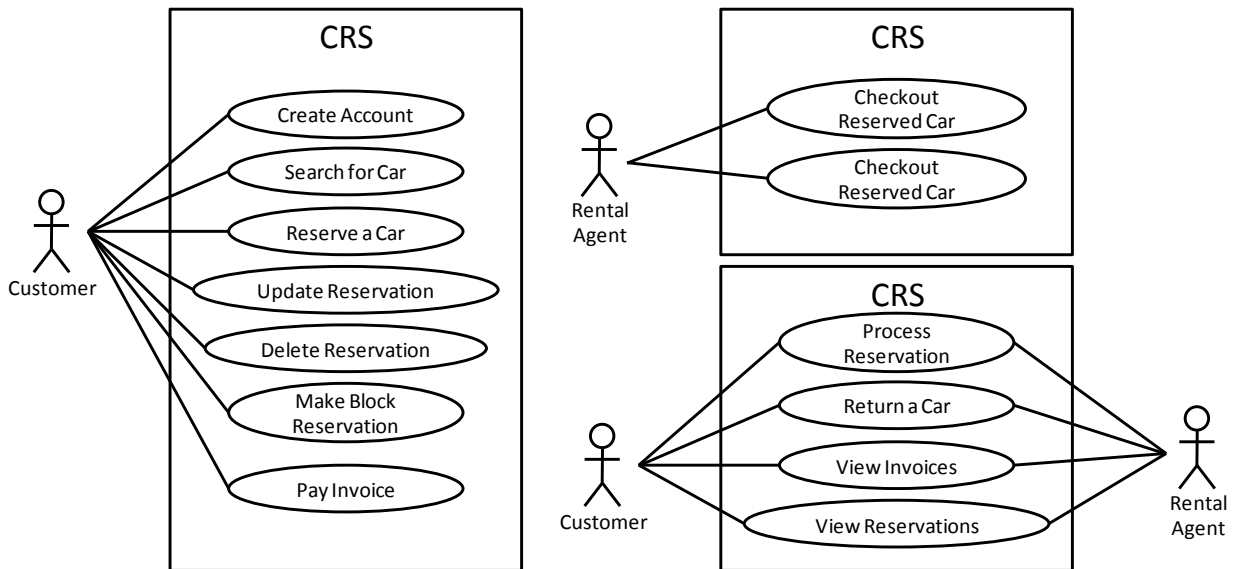
TUCBW the customer clicks the Update Reservation link on any of the CRS Web pages.
 TUCEW the customer sees a message stating that the reservation is updated successfully.

UC5. Delete a Reservation

TUCBW the customer clicks the Delete Reservation link on any CRS page.

TUCEW the customer sees a message stating that the reservation is successfully deleted.

Professor comment: Pretty complete answer, except for the use case diagram(s). Here's my take on those, for the Use Cases given:



Exercise 7.4 Use Case Diagrams

23.2 Apply the COCOMO II Application Composite Model described in Section 23.2.2 to estimate the effort required to develop the Add Program use case for a Study Abroad web application for the Office of International Education (OIE) of a university. Assume that the object point productivity is nominal and no reuse of existing components is expected. The Add Program use case allows an OIE staff to add an overseas exchange program, such as French Language and Culture. This use case needs to display three web pages and a message as follows:

- a. Initially, it displays a Welcome page, which consists of three frames. The main frame shows a description of the Study Abroad program. The top frame shows the OIE logo along with a row of buttons, such as OIE Home, About Study Abroad program, Contact Us, etc. The left frame shows a list of three menu items:**

Program Management, User Management, and System Settings. The Welcome page is displayed after an OIE staff member successfully logs into the system

- b. To add a program, the OIE staff clicks the Program Management item, which expands the item and shows a list of four menu items under the expanded item — Add Program, Update Program, Delete Program, and Import Programs.**
- c. The OIE staff clicks the Add Program item. The system displays and Add Program Form in the main frame of the page. The Add Program Form lets the OIE staff enter program information into the various fields. The OIE staff clicks the Submit button to save the program.**
- d. When the Submit button is clicked, the system saves the program into a database and displays a "program is successfully saved" message.**

Author Solution: The Application Composition Model consists of seven steps. This exercise simply applies these steps to the Study Abroad Management System application. There are four web pages, which are counted as screens. Three of these are simple and the one that requires the OIE staff to enter program information is considered difficult. There is no report or 4GL components, and the productivity level is nominal. The spreadsheet shown in Figure 23.2 illustrates how the 0.46 person-month effort is computed.

Step	#screens	#reports	#3GL Comp.	
1. Count the number of screens, reports, and 3GL components that will comprise the application.	4	0	0	
2. Determine the complexity levels of each of the screens and reports using Figure 23.2(a).	3 of the screens are simple, 1 is difficult.			
3. Look up the complexity weights for each of the screens, reports, and 3GL components from Figure 23.2(b).	Complexity weight=1 for a simple screen, it is 3 for a difficult screen.			
				Object point
4. Add the weighted counts of screens, reports and 3GL components to produce one number, called the Object Point (OP) count.	3*1 + 1*3			6
				New object point
5. Estimate the percentage of reuse to be achieved, and compute the New Object Points (NOPs) to be developed in the project: NOP = OP × (100 - Reuse)/100	reuse=0			6
				Productivity
6. Determine the object point productivity from Figure 23.2(c), that is, the average of two capabilities shown in Figure 23.2(c).	Nominal=13			13
7. Compute the person-month effort: Effort PM = NOP/PROD				0.46

Figure 23.2

Professor comment: I have nothing to add to this solution.

23.8 Suppose that a company is at level 1 of the CMMI. Propose a process improvement plan for the company to improve its process to level 5. Make necessary assumptions about the level 1 company.

Author Solution: Different solutions could be produced by the students, due to the assumptions made and the improvement plan proposed. Grading should be relative – that is, classify the solutions into three to five ranks according to various factors including validity of the assumptions, merits of the improvement plan, organization of the solution, readability, grammar, and spelling.

Professor comment: Another global author solution without providing any exemplars. Here we go on this one:

First, there is a CMMI level 0 which should be mentioned. It is labeled "incomplete", and is characterized by a processes and requirements management that are either not performed at all or don't achieve all goals and objectives which are needed to achieve CMMI level 1. It is basically the "no process" process. CMMI Level 1, labeled "Initial", has processes, but they are unpredictable, poorly controlled, and mostly reactive to situations rather than being proactive and planned in their approach.

- a. **Moving from level 1 to level 2:** Level 2 is called "Managed" and is characterized by processes that are actively managed, but not necessarily subjected to periodic reviews and updates. In addition, processes are still frequently reactive. Moving from this level to Level 2 requires that the organization spend the resources and time to plan, monitor, and control projects, perform configuration management and quality assurance tasks, and manage requirements and suppliers.
- b. **Moving from level 2 to level 3:** Level 3 is known as "Defined"; organizations which are at this level of maturity exhibit processes that are proactive and can be characterized or tailored for specific projects. Organizational process standards exist and are followed, including processes for tailoring. An organization which moves from Level 2 to Level 3 will implement a system of managing the processes to disseminate the information to all employees, and will review projects to ensure that the processes are being followed properly as part of a quality approach. Training will be done to ensure that all affected personnel know the drill.
- c. **Moving from level 3 to level 4:** CMMI Level 4 is called "Quantitatively Managed" which means that the organization has metrics. There will be frequent audits on all processes and on employees to ensure they are being followed. Other metrics will be kept, such as numbers of "escapes", "travelled work", and different categories of defaults in both the work products and the processes. Review and oversight are done on a frequent periodic basis, and resources are allocated to fix problems.
- d. **Moving from level 4 to level 5:** the Level 5, or "Optimizing" level, is the pinnacle of CMMI. At this level, the organization will fulfill ALL of the requirements for the other levels, and will also implement a "root-cause-corrective-action" analysis and process approach to review problems, assess the cause, determine the corrective action needed, and track the action to completion. In addition, risk assessment becomes part of this process, and is also incorporated into project planning and on-going project and process reviews. Continuous Measurable Improvement programs are instituted to facilitate identification of processes and other areas where

improvement is needed or might save cost or schedule for projects and the organization.

10.8 Describe three realistic, not dummy or hypothetical, situations to apply the controller, expert, and creator patterns.

Author Solution: There are many applications. Some of them are the following:

Controller: Banking systems, online retailing systems, email systems that support different types of user interfaces such as web-based, cellphone, and smart devices. The controller pattern uses one use case controller to fulfill requests sent from different types of user interfaces.

Expert: Login, reset password that needs to authenticate the user are excellent applications of the expert pattern. Without using the expert pattern, not only coupling is increased but security is also compromised.

Creator: The diagram editor, and modeling tools such as the OIM tool described in one of the exercises of Chapter 9 are good applications of the creator pattern. In these applications, the aggregate object such as the diagram object may create its components.

Professor comment: I have nothing to add to this solution.

11.1 Derive a DCD from the design sequence diagram in Figure 9.21.

Author Solution: The DCD is shown in Figure 11.1 below.

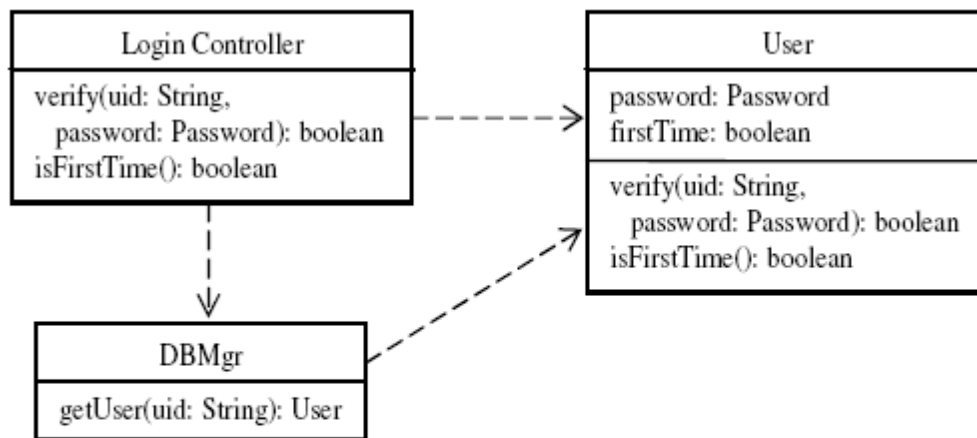


Figure 11.1: DCD derived from Figure 9.21

Professor comment: I have nothing to add to this solution.

12.1 Produce an expanded use case description, then perform the user interface design steps to produce a user interface design for the Reserve a Car use case for an online car rental application. A description of such an application is presented in Appendix D.1.

Author Solution: The expanded use case for Reserve a Car use case is shown in Figure 9.10 of this solution manual. From the expanded use case, one can identify three major system displays:

- 1) display of a list of cars as search result, that is, cars that satisfy the search criteria
- 2) a car reservation form
- 3) a reservation confirmation message

The results for the first three steps of the UI design is shown in Figure 12.1.

Expanded Use Case Step	System Display	Information Displayed	User Input	User Actions
0)	Search result page.	List of cars resulting from search car use case.		
2)	Car reservation form.	Attributes of Reservation class in the car rental system domain model (see Figure 5.20 in the textbook)		Clicking the Reserve button next to the car to be reserved.
4)	Confirmation message.	Reservation successful, or car is not available.	Reservation information.	Fill reservation form and clicks the Submit button.

You search finds the following cars:

Car	Summary	Description
Car 1	XXXXXXXXXXXXXXXXXX	View Reserve
Car 2	XXXXXXXXXXXXXXXXXX	View Reserve
Car 3	XXXXXXXXXXXXXXXXXX	View Reserve
Car 4	XXXXXXXXXXXXXXXXXX	View Reserve

(0) Search result page

Car Reservation Form

Customer name:	<input type="text"/>
Phone number:	<input type="text"/>
Address:	<input type="text"/>
From:	<input type="text"/>
To:	<input type="text"/>

(2) Reservation form

Reservation Confirmation Message

Thank you for your business. Your reservation is successful. Below is the reservation detail:

XXXXXXXXXX
XXXXXXXXXX
XXXXXXXXXX

(4) Confirmation message

Specification of Car 1

Make:	XXXXXX
Model:	XXXXXX
Number of seats:	x
Price:	XXX.XX
Mileage:	XX

View detail of a car

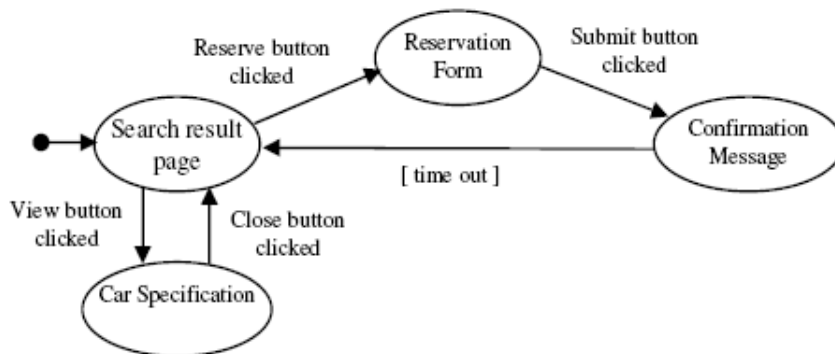


Figure 12.1: User interface design for reserve car use case

Professor comment: I have nothing to add to this solution.