01076254 Object-Oriented Analysis and Design

Homework #2 Sequence Diagrams

System Analysis

We can capture system requirement in a form of use case diagrams. A use case diagram explains functionalities of a system, but do not describe a set of interacting objects that support those functionalities. The purpose of Object-Oriented analysis is to construct a model demonstrating how interacting objects could deliver the behavior specified in the use cases.

The further explanation of how objects in the system co-operate with others to perform a behavior specified by the use cases is done by means of **realization**. In a realization process, a high-level interaction of instances of suitable classes is developed for each use case. With this information, a class diagram of the domain model can be further refined by incorporating the additional information of object responsibilities required to support the functionalities specified in the use cases.

Use case realization is conducted by using UML's **interaction diagrams**. There are two types of interaction diagrams, that is, collaboration diagrams and sequence diagrams.

Software Architecture

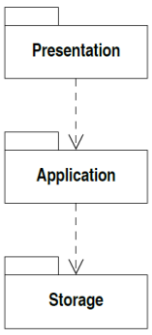
Software architecture refers to a set of high-level decisions on how the system will be divided into subsystems. One of the popular strategies is to apply the "**Model-View-Controller**" or **MVC** architecture to the system being analyzed. The Model class is responsible for maintaining the data and a View class is responsible for displaying it. The Controller interprets the inputs from the user, informing the model and/or the view to change as appropriate.

In this architecture, the two separate layers in the system are identified. Classes that maintain the system state are in an **application layer**, whereas classes concerned with the user interface are within a **presentation layer**. Each layer is represented by a **package** in UML.

1. Creating a component diagram

From create a new diagram, select "**Diagram**" from the application menu. Choose "**Component Diagram**". After creating a Component diagram, you can right click on it and select "**Rename**" to change the diagram name to "**Component Diagram Restaurant**". You can select a component, from the diagram tools in the drawing toolbar, to add into the activity diagram. You can select the component and add the detail on the left bottom corner if you want to customize the component to add to it any further detail description.

- **Draw the following package diagram:**



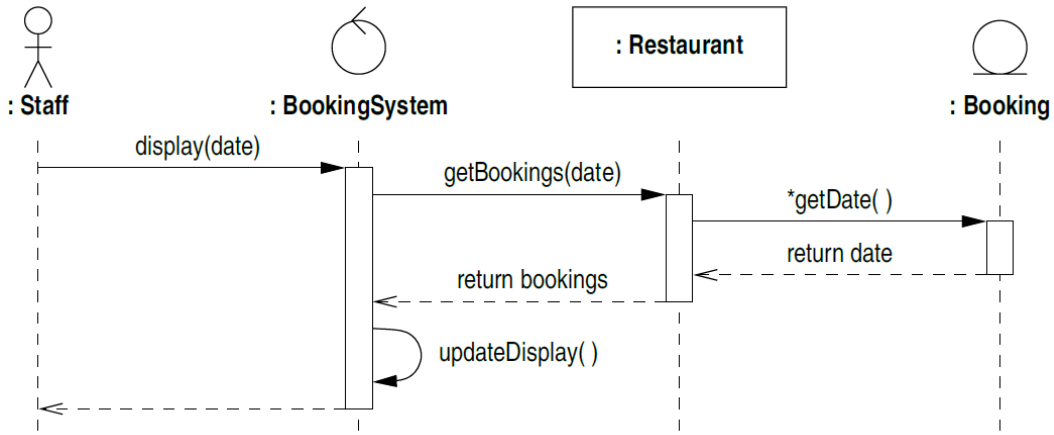
Use Case Realization: Restaurant System Analysis

The simplest case in booking system is probably the "**Display bookings**" case. From the interaction specified in the basic course of events, we can make the realization of the "Display bookings" use case and produce a sequence diagram to show the messages in the use case. When there are alternative and exceptional courses of events, it is usually preferable to show the realization of each course of events on a separate sequence diagram. Alternatively, you can also use the collaboration diagram to conduct use case realization.

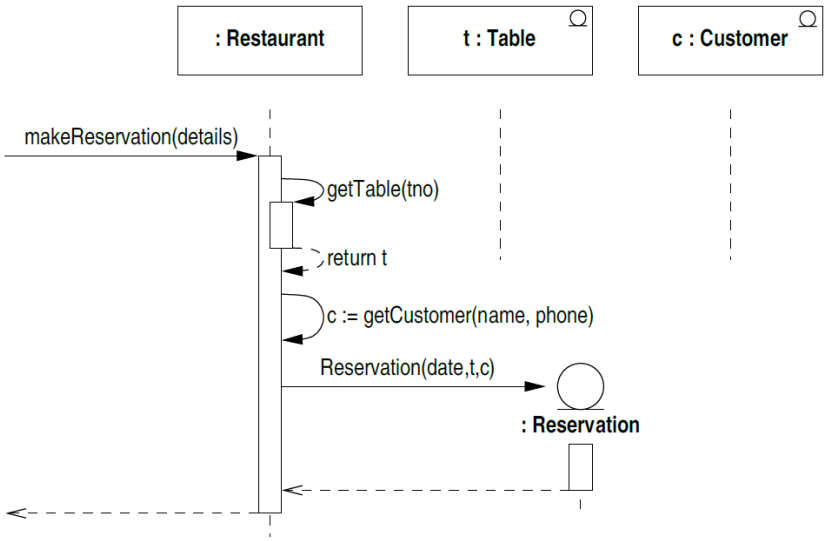
2. Creating a sequence diagram

From create a new diagram, select "**Diagram**" from the application menu. Choose "**Sequence Diagram**". After creating a Sequence diagram, you can right click on it and select "**Rename**" to change the diagram name to "**Sequence Diagram Restaurant Display bookings**". You can select a component, from the diagram tools in the drawing toolbar, to add into the activity diagram. You can select the component and add the detail on the left bottom corner if you want to customize the component to add to it any further detail description.

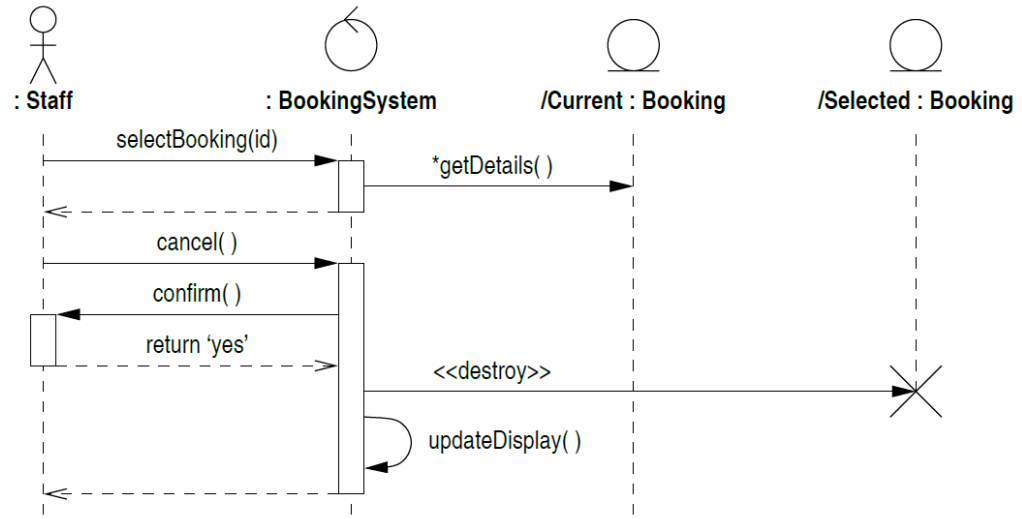
- **Draw the following sequence diagram for the realization of the "Display bookings" use case:**



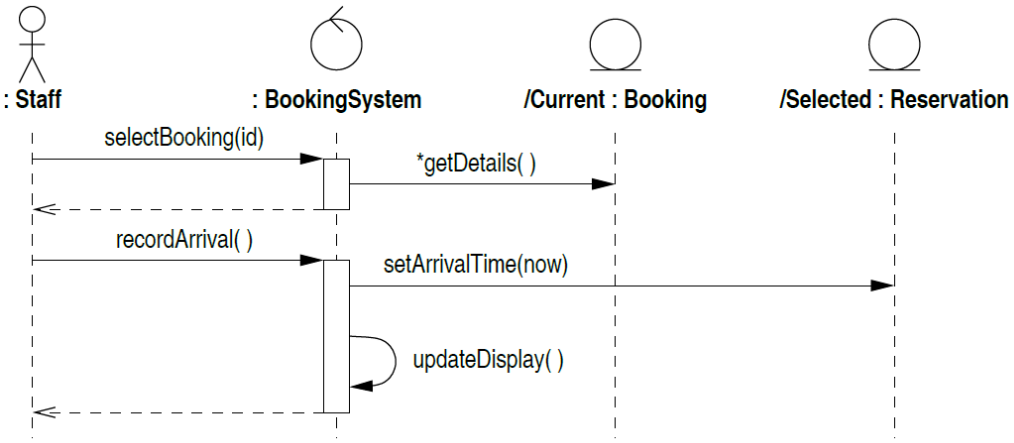
- **Draw the following sequence diagram for the realization of the "Record bookings" use case:**



- **Draw the following sequence diagram for the realization of the "Cancel bookings" use case:**



- **Draw the following sequence diagram for the realization of the "Record arrival" use case:**



Exercises

1. Read the description above and create a project in Enterprise Architect for the restaurant system. After that, create sequence diagrams for a realization of some use cases of the restaurant system (in the text book) and the refined domain model of the system.
2. Discuss your Java project with your team members. Then 2.1 Create sequence diagrams for the realization of **two** use cases of your project (You must create sequence diagrams that describe basic courses of events, alternative courses of events, and exceptional courses of events), 2.2 Refine the domain model as being guided by those sequence diagrams you have created in 2.1.