QB7Solution

[A, 7 points] Fill-in:

[A1] To show the dynamics of a use case, use an \_\_\_\_\_\_\_\_\_\_\_\_\_ diagram.

(activity)

[A2] To describe the main flow of a use case, use the \_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_ diagram.

(system sequence)

[A3] In the flight booking system, the domain expert clarified the issue by drawing an

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ diagram.

(object)

[A4] Derived attributes are shown in a class diagram using a \_\_\_\_\_\_\_\_\_\_\_\_\_\_.

(/)

[A5] Constraints in a class diagram are enclosed in \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_.

( { } )

[A6] In the class diagram for the Flight Booking System, the Flight class was

split into two classes using the \_\_\_\_\_\_\_\_\_\_ pattern.

(meta)

[A7] The class diagram for the Flight Booking System was divided into two modules so as to

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ the coupling between the modules.

(minimize)

[B , 7 points] True/ False

[B1] Use “include” and “extend” in use case diagrams in moderation. (T)

[B2] Limit the number of use cases to 20. (T)

[B3] The notion of state must not appear as an attribute on class diagrams. (T)

[B4] Only use the generalization relationship when the subclass is 100 % in accordance with

the specification of its superclass. (T)

[B5] Derived attributes allow the analyst not to make an overly premature decision

with regard to design. (T)

[B6] Make sure your classes do not have too many responsibilities. (T)

[B7] For a package to be a reusable component, it must not depend on other packages. (T)