Systems Development: Object Oriented Analysis and Design

(H172 35)

UML Diagram Formative

Use case

Class diagram

Case Study: Passenger check-in and boarding

**Requirements Gathering meeting with client**

The client described an airline check-in and boarding system as follows:

We differentiate between 3 options for check-in:

* Normal check-in with luggage at a normal check-in counter
* Express check-in without luggage at a special check-in counter
* Automated check-in without luggage at a machine

The process involves a passenger who wishes to travel, and a check-in representative who has the task of performing the passenger’s check-in process.

* The check-in procedure includes submitting the ticket, baggage check-in, seat reservation and issuing the boarding pass
* Passengers who only have hand luggage can use express check-in, no baggage check-in is performed
* Automated check-in is conducted without the help of a check-in representative, and is performed by the passenger via a check-in screen. Baggage cannot be checked-in.
* The passenger can choose between a normal check-in, express check-in and automated check-in
* The check-in representative can perform a regular check-in, but is not able to perform express check-in or automated check-in
* The express check-in has a separately appointed check-in representative
* During boarding, the boarding pass of the passenger is verified at the gate. The passenger walks to the gate and presents his/her boarding pass
* If we take a closer look, we notice that a passenger often travels with luggage, which he/she checks-in. Baggage transportation is responsible for loading luggage onto the airplane and it carried out by an independent organisation. It is essentially treated by the airport as an external service provider
* 10 minutes before a flight leaves, baggage transportation requests a passenger list from passenger services, which includes every passenger who checked-in, but did not board the plane. On the basis of this list all affected luggage will be unloaded from the airplane. If the flight is an international flight, the customs authorities of the country in which the destination airport is located also request a passenger list

**User Stories**

Describes a single goal from the user’s perspective:

As a (type of user)

I want (goal)

*so that (reason) optional*

**Outcome: write user stories on index cards**

**Use Case**

Use Case is a record of conversation that has happened and details particular steps for goal:

* **Title**: **what is the goal?**
  + Short phrase with active verb e.g. register new member, transfer funds
* **Actor**: **who desires it?**
  + Who is having this interaction? e.g. customer, member, ACME system (non-human)
* **Scenario**: how is it accomplished?
  + As a paragraph, details of accomplishing a goal step-by-step. Could write a numbered list of steps BUT remember this is NOT pseudocode!
  + *extensions* – for alternative flows i.e. when situation does not follow “normal” e.g. item out of stock
  + *precondition* – add a precondition to the scenario e.g. must have minimum of 1 item in shopping cart
  + *postconditions, triggers, stakeholders*

**Outcome: identify all use cases and complete the “use case template”. Draw the use case diagram including all associations (include, extend, generalisation)**

**Conceptual Object Model**

Move away from the user focus to identify the most important objects (not s/w objects) in the application. Generic objects – what are the things in the application we need to be aware of?

Identify objects, start to refine them, draw them in a simple diagram, which will allow us to show associations and interactions.

Go through user stories and use cases and pick out nouns to help identify objects, then verbs to help identify behaviours/responsibilities. Assign responsibilities to objects.

**Outcome: create a conceptual model or alternatively follow the CRC process below:**

**Class Responsibility Collaboration**

* Use index cards to record information, with each card representing **1 class**
* each CRC card has 3 sections:
  + **C** – class name – use nouns from use cases
  + **R** – responsibilities – use verbs from use cases
  + **C** – collaborations – other classes it interacts with

**Outcome: use individual index cards to document CRC for each class**

**Conceptual Class Diagram**

Create the conceptual class diagram as a result of the information gathered up until this point, giving consideration to encapsulation, abstraction, and showing the relationships between classes (association (multiplicity), aggregation, composition, generalisation, dependency)

Iteratively model your class diagram moving towards a design class diagram, where you should be additionally considering coupling, cohesion, abstract classes and interfaces.

**Outcome: create an initial conceptual class diagram, and iteratively model towards a final design class diagram**

**Using Visio**

Electronically model the use case diagram and the class diagram using Visio 2013