CS324 – SOFTWARE DESIGN & ANALYSIS

FALL 2020 PROJECT (Course Instructor: Ubaid Aftab Chawla)

Students will be working in groups of IDEALLY 4.

Objectives:

1. Understand the modeling concepts in developing of distributed information systems.

2. Use an object-oriented analysis and design methodology to develop simple web based Client-server solution.

3. Use the UML to model design solutions.

Requirements:

Project topic should be discussed with the course instructor. You can combine this project with a project that you are carrying out for another subject. For example, you can analyze and design a web based distributed system that you have to implement for another subject.

Project should involve stakeholders other than yourself. Project scope should be so that you can deliver completed model and generate some Java classes from your model.

Project due date:

3rd week of December 2020.

Example project:

Train Ticket Reservation System

This assignment models a distributed system for train ticket reservation. A person planning a trip will need to book trains for the destinations he/she plans to visit in Pakistan.

Details of the example project:

There are three main functions the system must support:

1) Planning the route & availability (Explanation in detail will be required).

2) Reservation (Explanation in detail will be required).

3) Payments & cancellation (Explanation in detail will be required).

Deliverables and some suggestions:

1. Requirements gathering: Detailed specification of all requirements. Clearly explain each process, its alternative paths, success paths, exceptions, activities needed to complete the system. Try to investigate and set some domain requirements.

a. Use MS Word, clearly name each requirements, and provide detailed description of each requirement and the features the system will provide.

b. Identify and name all the stakeholders and users of the system. Give detail description of how they will use the system and what features are critical for them. How would the system benefit each of them?

c. Each requirements should have the following subsections:

i. Requirement description, functionality

ii. Business rules and regulations

iii. Usability (who and how the requirements is used), any restrictions etc

iv. Technical and performance impact -as additional section

d. Identify all functional, non-functional and domain requirements.

2. Use case Analysis: Construct a full use case diagram for the entire system. Explain appropriate and relevant Use Cases to describe the proposed reservation system: you are advised to use a proper template to describe different use cases.

a. Each Use Case will be properly documented:

i. Name and use case description

ii. Processing details: input, output, DB access, UI, basic flow, alternate flows (if needed)

[Use case template that has been given in the class can be used]

3. Analysis and detail class diagram(s)

a. Start with the analysis class diagram and then add detail as the functionality and design become clearer.

b. You need to deliver only one detail class diagram for entire system. Since you will be generating java code in the second part of the prototype, you have to be precise in specification of types, attributes and method signatures. Use of proper inheritance and object oriented skills learned in the course will give you more score.

c. Entity class diagram description. Describe different entities that you have incorporated as part of your system.

4. ECB Robustness: Perform ECB Robustness analysis and construct a detailed ECB analysis diagram based on the rule that only controller can communicate with entities and boundary classes.

5. Behavior & Dynamic models The scenarios selection must provide full coverage of the scenarios.

a. Construct at least two activity diagrams for two scenarios. Apply business logics.

b. Construct at least two sequence diagrams for two selected scenarios.

c. Construct at least two collaboration diagrams for two scenarios. These two scenarios should be different from the one you have constructed in sequence diagram.

These diagrams should reflect

1. Alternate Paths (select one scenario only, java code will not be required):

2. Exceptions to implement: e.g. Reservation, payment or cancellation failed e.g. connection failure. Show post-events after the failure.

The dynamic models that you will be implementing should reflect the actual realization of use case diagram. The objects used in these dynamic models should be consistent with the ECB analysis.

5. State Transition Diagram: Demonstrate the use of proper states for at least two class objects. Construct two state transition diagrams for at least two classes.

6. Deployment Diagrams Construct deployment diagrams showing clients, servers, and system architecture (access via web clients, persistent storage etc).

7. .Java code generation for all classes described in the class diagram. In the code generation, select the same way as you have done in the lab.

NOTE: FOR ALL THE DIAGRAMS YOU WILL BE CONSTRUCTING, YOUR SCORES WILL BE BASED ON HOW WELL YOU USE EACH AND EVERY COMPONENT OF ALL THE DIAGARAMS TAUGHT IN THE CLASS. EACH DIAGRAM SHOULD MATCH THE OTHER IN DETAILS AND ANALYSIS AND PROPER SYNCHRONIZATION SHOULD BE DONE.