Kurdistan Regional Government
Ministry of Higher Education & Scientific Research
University of Salahddin – Hawler
College of Engineering
Software Engineering Dept.

Final Examinations (2011/2012)
Subject: Systems Analysis & Design
Second Year Students
Time allowed: 3 hours
Lecturer: Amanj Sherwany

The highest obtainable mark is 100, the minimum passing mark is 50

Q1/ A: (10 points)

Explain why a software system that is used in real-world environment must change or it will become progressively less useful.

Answer

Because system requirement is changing continuously, therefore for a software to stay relevant it has to be updated to meet the changed requirements.

B: (15 points)

Giving reasons for your answer based on the type of system being developed, suggest the most appropriate generic software process model that might be used as a basis for managing the development of the following systems:

- A social network web application (a facebook-like application)
- A system to control the traffic-lights in Erbil city.
- A music player with a built-in browser for purchasing music.

Answer

- An incremental delivery would be the best, since the application can be lunched with a limited functionality with adding more functionalities in the next releases.
- Waterfall model, since this is a critical system (humans life depend on it)
- Maybe the best would be Component-Based Software model, as one can re-use an already developed web browser.

* * *

Q2: (25 points)

The following is an excerpt taken from a *Software Requirement Specification* (SRS) document:

> "The application should have three main windows: one for playing music, one for playing video and the other one for browsing the Internet. A lot of audio formats (codecs) should be supported. The program shall have a modular architecture. The video play back shall be very smooth. The Internet browser shall be very beautiful."

In the above paragraph:

- Find two ambiguous requirements. **(10 points)**
- Extract the possible functional and non-functional requirements. **(15 points)**

Answer

Ambiguous requirements

- A lot of audio formats (codecs) should be supported.
- The video playback shall be very smooth.

Functional and Non-functional requirements:

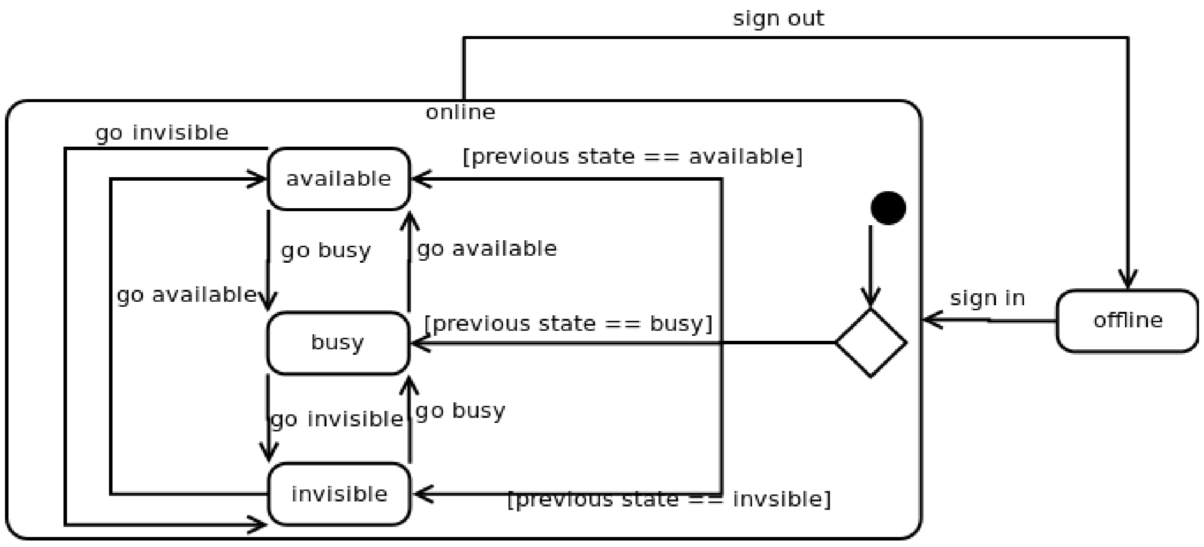
- Functional Requirements
    - The application should have three main windows: one for playing music, one for playing video and the other one for browsing the Internet.
    - A lot of audio formats (codecs) should be supported.
- Non-functional Requirements
    - The program shall have a modular architecture.
    - The video playback shall be very smooth.
    - The Internet browser shall be very beautiful.

* * *

Q3: (20 points)

On Gtalk-like messengers a user can be either *online* or *offline,* while *online* she can be *invisible, busy* or *available,* draw a state-machine diagram to express that.

Answer



Q4/ A: (15 points)

You are writing a DocumentViewer class, which should be able to open (PDF, DOC and HTML) documents, what is the best design pattern to use here? show it in code.

Answer

Factory Method Design pattern is the pattern to use here:

```
interface Document{
     public void open();
}
public class HTMLDocument implements Document{}
public class PDFDocument implements Document{}
public class DOCDocument implements Document{}
public class DocumentViewer{
     public static int PDF = 1;
     public static int DOC = 2;
     public static int HTML = 3;
     public Document createDocument(int type){
          if(type == PDF){
               return new PDFDocument();
          }else if(type == DOC){
               return new DOCDocument();
          }else if(type == HTML){
               return new HTMLDocument();
          }
     }

     public void newDocument(int type){
          Document doc = createDocument(type);
          doc.open();
     }
}
```

B: (15)

B: (15 points)

The following interface violates one of the SOLID principles, identify it and re-write the code to obey the principle.

```
interface Radio{
     public void changeStation(int newFrequency);
     public void volumeDown();
     public void volumeUp();
}
```

The above interface violates the Single Responsibility Principle (SRP), as it has two responsibilities:

1. Changing radio station
2. Adjusting the volume

```
To fix it we have to decouple the two responsibilities:

interface RadioReceiver{
     public void changeStation(int newFrequency);
}

interface VolumeAdjuster{
     public void volumeDown();
     public void volumeUp();
}

interface Radio extends VolumeAdjuster, RadioReceiver{}
```

Good Luck