

# EBU6304 Software Engineering

## Exercises Set 2

### Analysis and Design

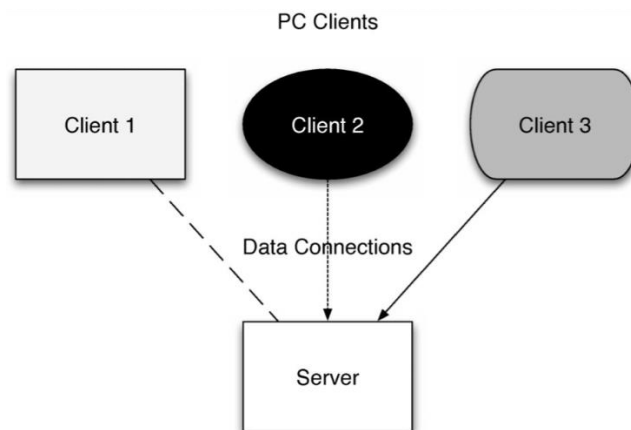
1. Describe the relationship between the Bank Account, Current Account, Junior Account and Saving Account with a UML class diagram.
2. Read the following description in a user story:

*A digital library system is being developed for QMUL library. It is to be used by QMUL staff, QMUL students and public members. Some users can read books inside the library, but cannot borrow them, whereas other users can read and borrow books. Library admin staff can check in/check out books. Librarians can perform all of the above duties plus order new books.*

**This description is not well written.** You need to do **analysis** to better understand the requirements. Identify **Entity** classes from the description and draw a UML class diagram to show their relationships.

### Software Architecture

1. In the lecture on Software Architecture we said that “box and arrow” diagrams sketched on a whiteboard are often sufficient to start reasoning about the architecture of a system. However, some issues can arise if one is too relaxed about the diagrams’ notation. Consider for example the diagram below: can you think of any issue or potential source of confusion in it?



2. Many APIs provided by companies are labelled as “RESTful”. For example, the Twitter API is often called “Twitter REST API” (<https://www.w3resource.com/API/twitter-rest-api/>), however it’s hard to understand if this API is really satisfying the 6 (5 compulsory + 1 optional) architectural constraints of the REST architecture. Search the Internet for some evidences of why the Twitter API may or may not be a truly RESTful API.

*(Hint: there are some interesting discussions on this topic on stackoverflow)*

## Testing

1. Why does software have bugs despite developers' carefully following software engineering methodologies?

2. Test case design.

The table below shows the discount rates for theatre tickets. The current price of an Adult ticket is £100. Children under age 6 are not allowed to get admission into the theatre.

Age	Discount
Adult (age 26-64)	No discount
Young person (age 6-25)	20% discount
Concession (age 65+)	10% discount

You need to test the software component which is used to calculate the price. This component takes the age as input, and outputs the price. Design Test Cases using **Partition Testing**.

## Test Driven Development (TDD)

1. We use TDD approach to test the above price calculation component. Fill in the missing code of this test program.

```
import static org.junit.jupiter.api.Assertions.*;
import org.junit.jupiter.api.Test;

public class TicketTest {

    @Test
    public void testCalPrice(){
        Ticket t1 = new Ticket(100);

        assertEquals(_____, t1.getPrice(26));
        assertEquals(_____, t1.getPrice(64));
        assertEquals(_____, t1.getPrice(45));
        assertEquals(80.0, _____);
        assertEquals(80.0, _____);
        assertEquals(80.0, _____);
        assertEquals(_____, _____);
        assertEquals(_____, _____);
    }
}
```

2. Write the product program to pass the above test.

```
public class Ticket {  
    private double fullPrice;  
    public Ticket(double price){  
        fullPrice = price;  
    }  
    public double getPrice(int age){  
        //write your code here to pass the test  
    }  
}
```