Week 4: test

Exercise 1

Hereafter is a program that compares two strings and prints if they are equals or not.
Q1. Propose a test set based on an equivalence partition strategy

```
equal : boolean
string1, string2: string
Read(string1, string2)
equal = (string1.length = string2.length)
if equal then
  for (i=1 ; i<= string1.length; i++) {
      equal = (string1.character[i] = string2.character[i])
  }
if equal then
   print(« same strings »)
else
   print(« different strings »)
```

Q2. Draw the control-flow graph of the program.
Q3. Check if the test data produced in question Q1. covers all statements and all branches of the control-flow graph.
Q4. What is the fault in the program?

Exercise 2

Let "example(x,y,z)" be a program with 3 inputs.
Let {x1,x2,x3}, {y1,y2,y3} and {z1,z2} be the selected data for testing.
Q1. How many tests do we obtain with an all-combination approach?
Q2. Propose a test set satisfying a pairwise coverage

Week 5: requirements

Exercise 3

Explain the limits of the following requirements. Propose reformulation.

- From the application code, interface with the package must be as simple as possible.
- The "Async" module is not supposed to interact with the operator.
- Installation of the software must be easy and straightforward.

Exercise 4

What are the risks when using an interview method as elicitation technique?

Week 6: UML

Exercise 5

Propose a UML Class diagram for the following examples.

1. A country has a capital and a surface area. It usually shares frontiers with other countries (exception exists for some islands).
2. A supermarket sells different types of articles: food products, objects and flowers. Each article has a bar code and a price. A food product has an expiry date. Flowers have a sell-limit date.

Exercise 6

Chess is a two-player board game. The player with the white pieces always moves first. After the first move, players alternately move one piece per turn (except for castling, when two pieces are moved). The objective is to ***checkmate*** the opponent's king by placing it under an inescapable threat of capture. To this end, a player's pieces are used to attack and capture the opponent's pieces, while supporting their own. In addition to checkmate, the game can be won by voluntary resignation by the opponent,

which typically occurs when too much material is lost, or if checkmate appears unavoidable. A game may also result in a draw in several ways.

Q1. Propose a state-diagram to model the game, with two principal states that are: whiteArePlaying and blackArePlaying.

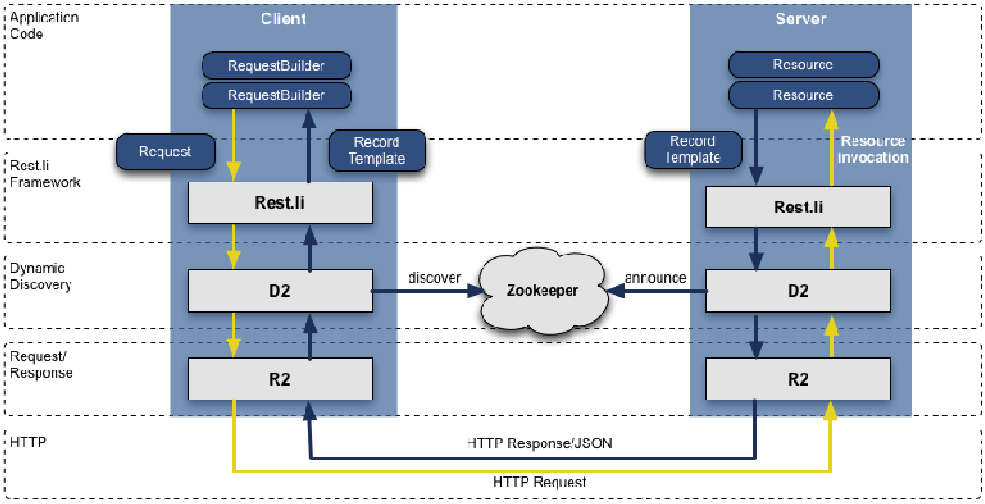
Week 7: life-cycle

Exercise 7

Q1. What are the limits of a waterfall development model?
Q2. What is the main difference between an iterative and an incremental life-cycle?

Week 8 and 9: architecture

Exercise 8



Q1. Reformulate the previous schema into an architectural description that follows the principles of (1) separation of concerns and (2) abstraction.
Q2. Which architectural styles (patterns) does this application follow?