Dipartimento di Elettronica, Informazione e Bioingegneria
Politecnico di Milano

Prof. Elisabetta Di Nitto, Raffaela Mirandola

20133 Milano (Italia)
Piazza Leonardo da Vinci, 32
Tel. (39) 02-2399.3400
Fax (39) 02-2399.3411

Software Engineering II

January 15th 2015

Last Name

First Name

Id number (Matricola)

Note

1. The exam is not valid if you don't fill in the above data.
2. Write your answers on these pages. Extra sheets will be ignored. You may use a pencil.
3. You can use textbooks, slides and your notes
4. The use of any electronic apparatus (computer, cell phone, camera, etc.) is strictly forbidden.
5. You cannot keep a copy of the exam when you leave the room.
6. The exam should finish at 10.05

General note about the solutions
The solutions given here are not the only ones possible. You can think of many others that fulfill the request of the exercises. In the case of the function points calculation we have been considering the minimum set of functions/data structures based on the given text.

Question 1 Alloy (6 points)

Consider the following fragment of an Alloy specification:

```
sig Address {}
sig Value {}
sig Computer {
 memory: Address -> Value
}
```

1. Formalize in Alloy the following invariant: *in each computer, exactly one value is associated to each memory cell*.
2. Modify the above invariant to represent also computers in which memory cells can be uninitialized.
3. Define in Alloy the writeMemory operation that, given a computer cell address, associates to it a new value, replacing the pre-existing one (if any).

Solution

1. It is enough to change the Computer signature as follows:
```
sig Computer {
  memory: Address -> one Value
}
```
2. It is enough to change the Computer signature as follows:
```
sig Computer {
 memory: Address -> lone Value
}
```
3.
```
pred writeMemory[c: Computer, a: Address, v: Value, c': Computer] {
  some val: Value | val = a.(c.memory) implies
    c'.memory = c.memory - {a -> val} + {a -> v}
no val: Value | val = a.(c.memory) implies
    c'.memory = c.memory + {a -> v}
}
```

Questions 2 Planning (5 points) and design (5 point)

We need to develop an online betting application. The application allows the users to bet for the result of some sport event. Each user has a "virtual account" used to bet and to accumulate winnings.
The operations associated to deposit and withdraw of money on the virtual account are performed outside the system.
The sport events can be of various kinds, e.g., soccer match, Formula 1 race, …
The application should allow:

- System administrators:
  - To create sport events inserting the corresponding name (e.g., "soccer championship, match Roma-Napoli"), the bet subjects (e.g, "victory of Roma", "victory of Napoli", "draw result") and the multiplication factor associated to each subject. If a user bets on a winning subject, he/she receives on the virtual account an amount of money equal to the bet multiplied by the multiplication factor.
- Registered users:
  - To visualize the sport events, the bet subjects and the associated multiplication factors.
  - To bet selecting a specific event, subject and an amount of money. This amount is detracted from the virtual account.
  - To visualize the list of bets and the associated results as soon as they are available.

The outcome of sport events is received by the system through a web service offered by a sport official authority.
Based on the received outcomes, the application automatically calculates the winnings and assigns them to the virtual accounts of users.
Given the above description, do the following:

A) **Calculate the function points** for the system and provide the rationale for each of them. Refer to the following table to associate weights to the function types:

Function types	**Weights**		
	Simple	Medium	Complex
N. Inputs	3	4	6
N. Outputs	4	5	7
N. Inquiry	3	4	6
N. Internal Files	7	10	15
N External Files	5	7	10

B) **Define the system architecture**. 1. Use any diagram to highlight the main architectural components and corresponding connectors. 2. Describe the function of each component. 3. Draw the sequence diagram associated to the calculation of winnings and their association to virtual accounts.

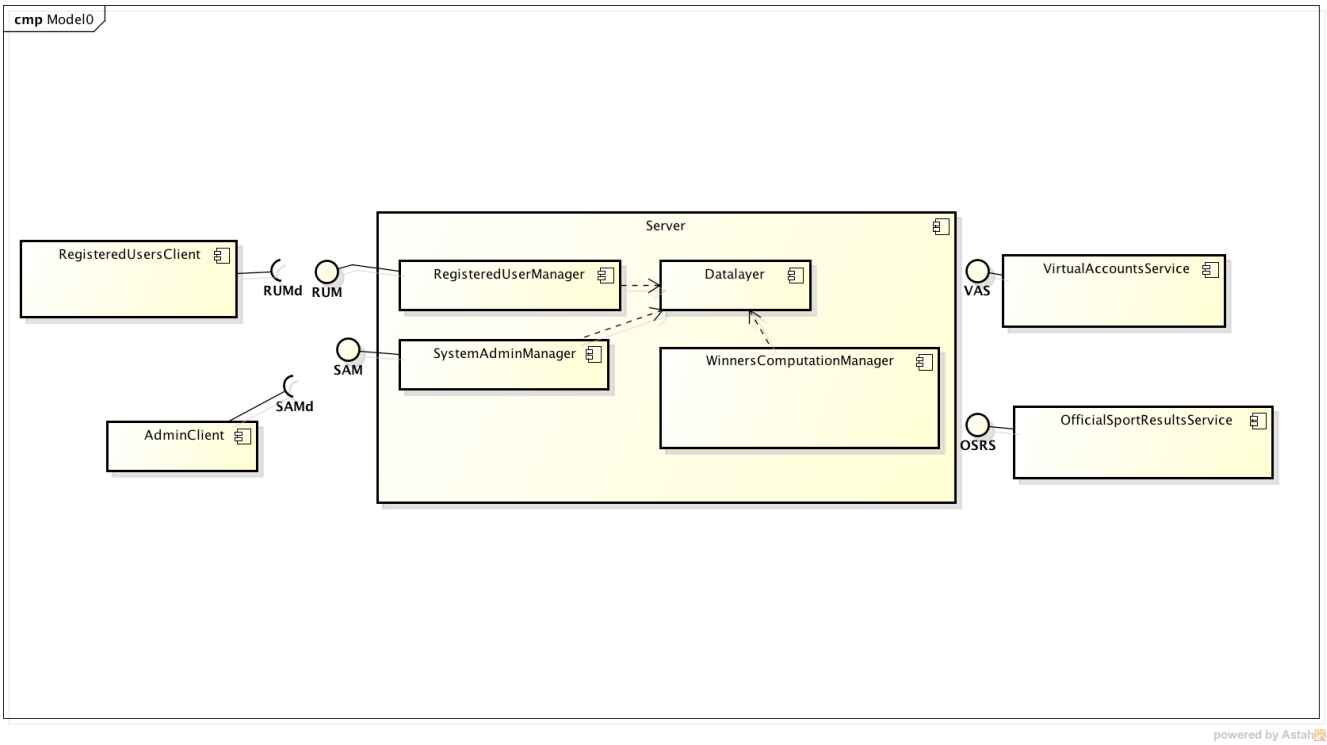
Add explicitly any hypothesis or functionality you think is important to consider in this case.

Solution

Internal Logic Files: The data structures to be defined are the following: Sport Events, Bet Subjects (including multiplication factors), Users (including the corresponding virtual account), Bets. All of them are simple structures, N. IF: 4*7 = 28
External Logic Files: a) The outcomes concerning the sport results. We can consider this of medium complexity, b) the virtual account of users, we can consider this of medium complexity N. EF: 2*7 = 14.

N. Inputs: The operations concerning acquisition of external inputs are login/logout (simple), sport event creation (this can be considered of medium complexity as it requires the insertion of various data) and bet (this can be considered simple). So, N. Inputs: 1*4+3*3=13
N. Outputs: The operation that produces data for the external environment is the one that computes the winners and transfer the money to their virtual accounts. We can consider this as a complex operation, so, N. Outputs: 7
N Inquiry: These are the operations to visualize the sport events and the list of bets. These can be considered simple. So, N. Inquiry: 2*3=6
The total number of FP is then: FP = 28+14+13+7+6=68

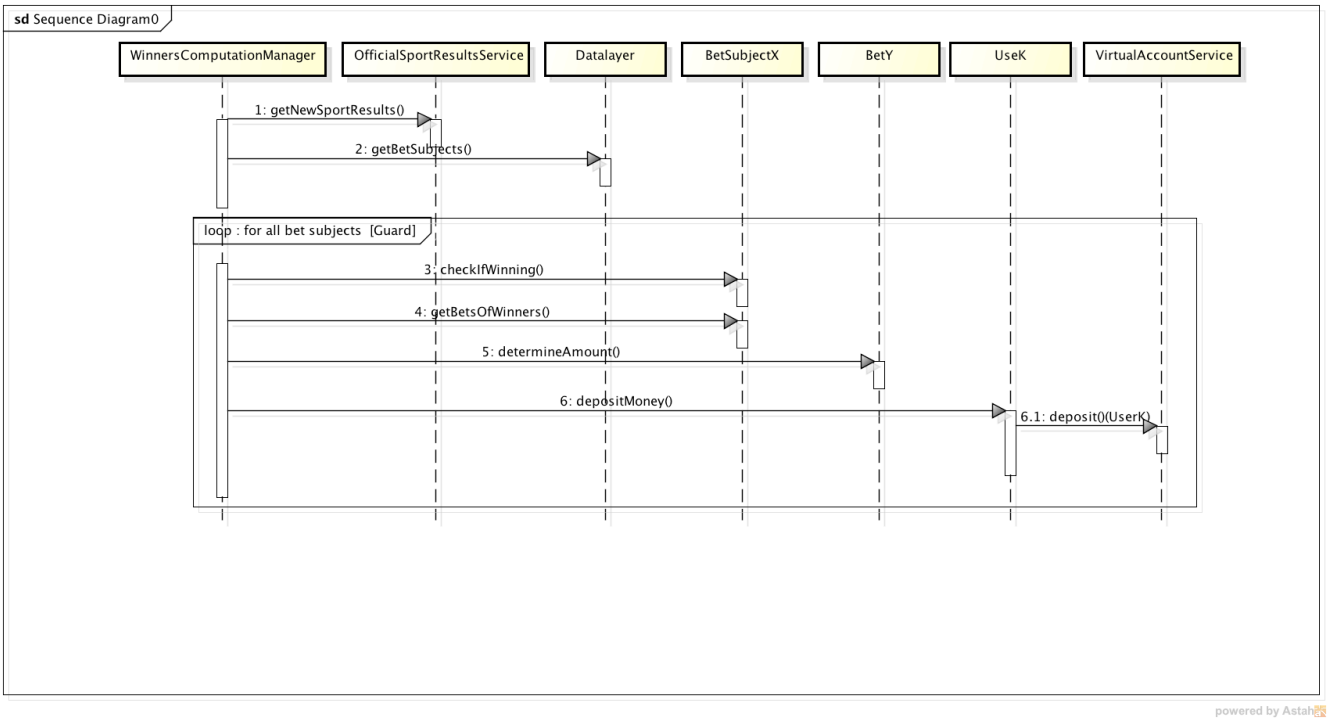
Architecture



The system follows *client-server* style for what concerns the management of the interaction with the users (including system administrators) and interacts, through a service interface (e.g., a REST interface), with the *OfficialSportResultsService* and with the *VirtualAccountsService*.
The server contains the main logic and it is in charge of interacting with two kinds of web clients, the *RegisteredUsersClient* and the *AdminClient*. Moreover, it periodically (with a period that can be configured by the system administrator) contacts the *OfficialSportResultsService* to know about all sport results that are newer than a certain time stamp. This time stamp is saved by the system server based on the previous interaction with the *OfficialSportResultsService*. Finally, the server interacts with the *VirtualAccountService* to acquire information about the status of the user virtual account, to withdraw money when a user poses a bet and to deposit them when a user wins.
The server part is itself composed of the following subcomponents:

- *RegisteredUsersManager*: this component receives from the RegisteredUsersClient all requests concerning the operations that are already listed in the Function Points analysis (login/logout, visualize sport events and the corresponding bet subjects, visualize the status of bets posed by the user, pose a bet) and interacts with the Datalayer component to accomplish all of them.
- *Datalayer*: this component encapsulates the entities relevant to the system (Sport Event, Bet Subject, User and Bet) and their storage into a DBMS. The User entity is not a simple one as it contains the logic to contact the VirtualAccountService to know the balance of the user virtual account, withdraw and deposit. Of course, withdraw is performed only if it leads to a positive or zero balance for the user.

- *SystemAdminManager*: This component is in charge of supporting AdminClients. In particular, it allows them to login/logout, create sport events, including the definition of sport subjects and corresponding multiplication factors, visualize the status of bets of all users. It interacts with the Datalayer component to accomplish all of them.
- *WinnersComputationManager*: this component cross-references sport results and pending bets (interacting with the Datalayer) to identify winners and deposit money into their account.



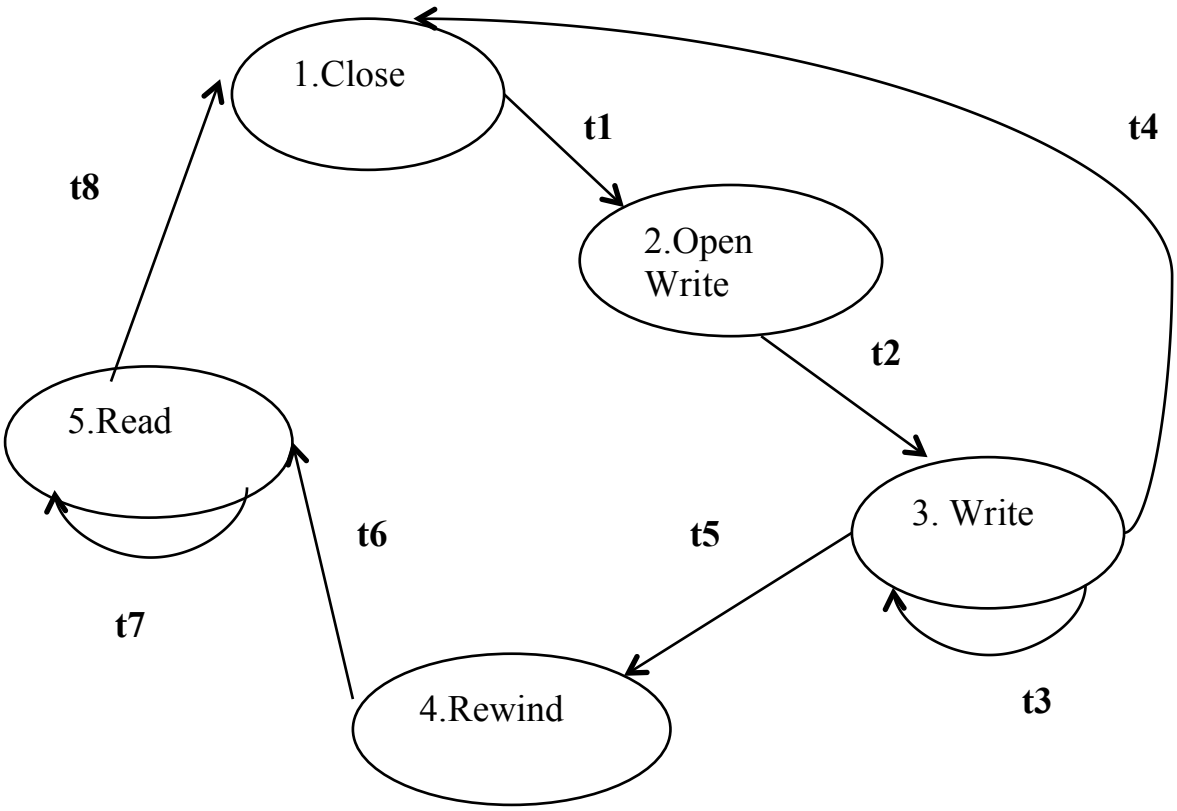
Question 3: Testing (6 points)
Consider the following specification of possible operations on sequential files.

To perform write (WRITE) operations on a given file, it is first necessary to open it (OPEN). After writing, to perform read (READ) operations, it is first necessary to carry out a rewind (REWIND) operation that moves the cursor to the beginning of the file.

Describe a possible approach to use this specification to derive test cases.
Add explicitly any hypothesis you think is important to complete the specification.

Solution

Given the description above, the approach to be used is black box testing.
Having the specification, it is possible deriving a state diagram describing the system behavior and then identifying the test cases.
The state diagram is illustrated below:



Test cases depend on the selected coverage criterion: coverage of states, or coverage of transitions.

In the first case, the test cases are (numbers refer to states):
TC1: 1 2 3 1
TC2: 1 2 3 4 5 1

In the second case:
TCa: t1 t2 t3 t3 t4
TCb: t1 t2 t4
TCc: t1 t2 t5 t6 t7 t8
TCd: t1 t2 t5 t6 t8

Given the above states, we could also test the system by trying to enforce transitions that should not be allowed. For instance, the sequences 1, 2, 4 and 1, 5, 4.