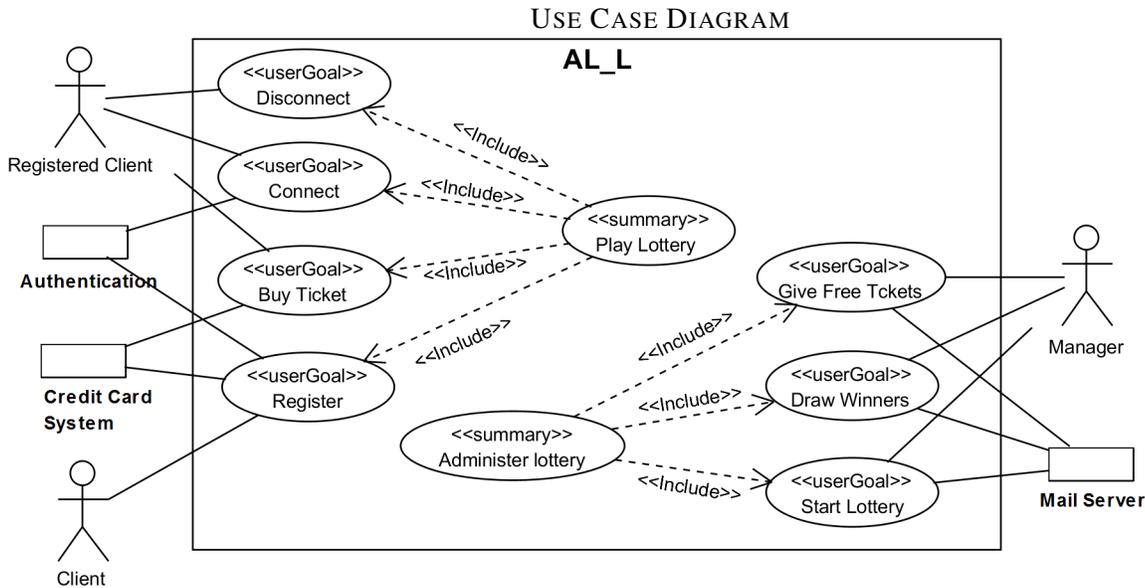


# APPENDIX:

## Disciplined Use Cases and Screen Mockups

Gianna Reggio, Maurizio Leotta, Filippo Ricca  
 DIBRIS, Università di Genova, Italy  
 gianna.reggio@unige.it, maurizio.leotta@unige.it, filippo.ricca@unige.it



### GLOSSARY

#### Data

- (\*\*1) *free ticket law*: an operation that given a natural number  $K$ , a list of client identifications, and a set of integer numbers  $IS$  returns  $K$  pairs, consisting of a client identification and of a number in  $IS$ ; it will be used to determine which tickets will be given away and to which clients
- (\*\*2) *winning order*: a total order on the integer numbers; it will be used to determine the winning tickets of a lottery (they will be those whose numbers are the first three after having ordered all the ticket numbers using this order)
- (\*\*3) *credit card data*: information characterizing a credit card (issuer, number, expiration date)
- (\*\*4) *client info*: information about a registered client (email, credit card data, numbers of the bought tickets)
- (\*\*5) *password*: more than 8 characters containing at least a special character
- (\*\*6) *email*: valid email address (RFC 5322 compliant)

#### System attributes

- (\*\*7) *Current Lottery Dimension*: a positive natural number representing the dimension of the current lottery (i.e., how many tickets it has)
- (\*\*8) *Current Free Ticket Law* and *Current Winning Order*: those relative to the current lottery
- (\*\*9) *Running*: a boolean, true if a lottery is currently running
- (\*\*10) *Available tickets*: set of integer numbers, the numbers of the tickets of the current lottery still not assigned to some client
- (\*\*11) *Registered Clients*: information about the registered clients
- (\*\*12) *Connected Clients*: the emails of the registered clients currently connected.

- A summary use case cannot be included in either a user-goal or a subfunction use case.
  - A user-goal use case cannot be included in a subfunction use case.
  - A summary use case must have the intention in context and the stakeholder parts.
  - If a use case C includes C1 in the use case diagram, then at least a line corresponding to include C1 must appear in the description of C, and vice versa (i.e. every inclusion in the use case descriptions must appear in the use case diagram).
  - If a use case C extends C1 in the use case diagram, then at least a line corresponding to an extension point for C must appear in the description of C1, and vice versa (i.e. every extension point in the use case descriptions must correspond to an extension relationship in the use case diagram).
  - The actors listed in the use case descriptions should be in accord with those appearing in the use case diagram and vice versa.
  - If a use case has no actors, then it must have a trigger\*.
  - If a step has a condition *cond* different from true, then there should be some extensions starting from the same step with conditions  $cond_1, \dots, cond_n$  s.t. the logical disjunction of  $cond, cond_1, \dots, cond_n$  is true.
  - Each complete scenario must include at least a step where the subject is the system.
  - Each listed actor of a use case must be the subject at least of one step of its scenarios.
  - All the initial steps of a set of extensions starting from the same point must have the same subject.
  - The subject of a step of a scenario of a use case different from the system, must appear among the use case actors.
  - Each data listed in the glossary must appear at least in one line of a use case.
  - Each system attribute listed in the glossary must at least:
    - appear in the effect part of a step of a use case,
    - appear either in the condition or in the interaction part of a step of a use case.
- \* (e.g., a use case describing a periodic activity made by the system each hour)

TABLE I  
WELL-FORMEDNESS CONSTRAINTS FOR REQUIREMENTS SPECIFICATION

- Let  $S_0, S_1, \dots, S_n$  ( $n \geq 0$ ) be some steps having an actor as subject s.t.  $S_1, \dots, S_n$  are the first steps of extensions starting from  $S_0$ , and let  $M$  be the initial mockup of  $S_0, S_1, \dots, S_n$ .
  - If the interaction part of  $S_i$  ( $0 \leq i \leq n$ ) refers to some communication from the actor to system, then some means to represent it must appear in  $M$  (e.g., when  $S_0$ ="Client confirms",  $S_1$ ="Client refuses", and two buttons "Confirm" and "Refuse" appear in  $M$ ).
  - If  $M$  contains some means for realizing some communication from the actor to system, then there should be  $S_i$  ( $0 \leq i \leq n$ ) referring to such interaction.

- Let  $S_1, \dots, S_k$  ( $k \geq 1$ ) be some steps having an actor as subject, and let  $M$  be the final mockup of  $S_1, \dots, S_k$  ( $S_1, \dots, S_k$  must be steps having the same interaction part appearing in different scenarios of even different use cases).
  - If the interaction part of  $S_1$  (that it is coincident with those of  $S_2, \dots, S_k$ ) refers to some communication from the actor to system, then  $M$  should show how it is going to be realized (e.g., the step  $S_1$  has the form "User confirms the deletion", and in  $M$  there is a button "Confirm Deletion"; notice that this step may represent the confirmation of different kinds of deletions).
  - If the interaction part of  $S_1$  (that it is coincident with those of  $S_2, \dots, S_k$ ) includes a reference to some specific information (flowing from the actor to system), then such information must appear in some way in  $M$  (e.g., "Actor inserts the password" and "password" appears in  $M$ ).
  - If  $M$  shows how some communication is going to be realized (from the actor to system), then the interaction part of step  $S_1$  (that it is coincident with those of  $S_2, \dots, S_k$ ) should refer to it.

- Let  $S_1, \dots, S_m$  ( $m \geq 1$ ) be some steps having the system as subject, and let  $M$  be the final mockup of  $S_1, \dots, S_m$  ( $S_1, \dots, S_m$  must be steps having the same interaction part appearing in different scenarios of even different use cases).
  - If some information appears in  $M$ , then it should be derived by the interaction parts of the previous steps or by the system attributes (e.g., in  $M$  appears "You are logged as John Doe", and the name of the current logged user "John Doe" is recoverable by the system attributes or it was provided by the user in some previous step).
  - If the interaction part of  $S_1$  (that it is coincident with those of  $S_2, \dots, S_m$ ) refers to some communication from system to actor, then some means to represent such communication must appear in  $M$  (e.g., "System confirms the required deletion" and either a pop-up or a message box containing a sentence equivalent to "deletion confirmed" appears in  $M$ ).

TABLE II  
WELL-FORMEDNESS CONSTRAINTS FOR SCREEN MOCKUPS

## USE CASE DESCRIPTIONS

**Use Case Register****Level:** User Goal**Priority:** 1**Frequency:** Periodically**Intention in Context:** A client wants to register to AL\_L to be able to play in the lotteries**Primary Actor:** Client**Secondary Actors:** Credit Card Service, Authentication**Main Success Scenario:**

1. Client asks AL\_L to be registered, giving an *email* (\*\*6) and some *credit card data* (\*\*3).
2. If no one among the *Registered Clients* (\*\*11) is using the given email, and the *credit card data* are well-formed (\*\*3), then AL\_L asks Credit Card Service to check them.
3. Credit Card Service informs AL\_L that the submitted credit card is valid.
4. AL\_L asks Authentication to register the client giving his/hers email.
5. Authentication confirms the registration and gives to AL\_L the client password (\*\*5).
6. AL\_L informs Client that (s)he has been registered and gives hers/his password. The information about Client (hers/his email, credit card data) (\*\*4) is added to the *Registered Clients* (\*\*11). The use case ends with success.

**Extensions:**

- 2a.1 If the credit card data are ill-formed (\*\*3), then AL\_L informs Client that the registration has failed. The use case ends with failure.

- 2b.1 If someone among the *Registered Clients* is using the given email, AL\_L informs Client that the registration has failed. The use case ends with failure.
- 3a.1 Credit Card Service informs that the submitted credit card is invalid.
- 3a.2 AL\_L informs Client that the registration has failed. The use case ends with failure.

**Use Case Connect****Level:** User Goal**Priority:** 1**Frequency:** Frequently**Intention in Context:** A registered client wants to connect herself/himself to AL\_L to be able to play in the lottery.**Primary Actor:** Registered Client**Secondary Actor:** Authentication**Precondition:** The email of Registered Client does not belong to *Connected Clients* (\*\*12).**Main Success Scenario:**

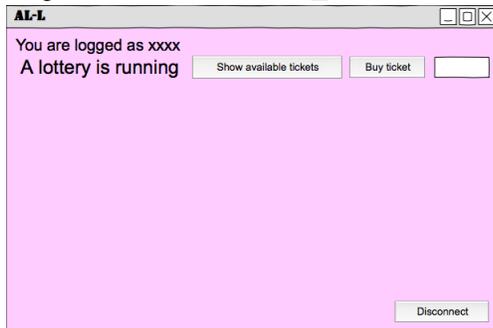
1. Registered Client asks AL\_L to be connected giving hers/his *email* (\*\*6) and *password* (\*\*5).
2. AL\_L asks Authentication to validate such *email* and *password*.
3. Authentication informs AL\_L that they are valid.
4. AL\_L informs Registered Client that s(he) has been connected, and hers/his email is added to *Connected Clients*. The use case ends with success.

**Extensions:**

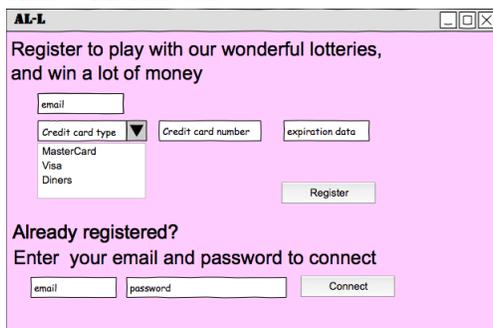
- 3a.1 Authentication informs AL\_L that they are not valid.
- 3a.2 AL\_L informs Registered Client that the connection has failed. The use case ends with failure.

**Use Case Disconnect****Level:** User Goal**Priority:** 1**Frequency:** Frequently**Intention in Context:** A connected client want to disconnect herself/himself from AL\_L.**Primary Actor:** Registered Client**Precondition:** The Registered Client email belongs to *Connected Clients* (\*\*12).**Main Success Scenario:**

1. Registered Client asks AL\_L to be disconnected.



2. AL\_L informs Registered Client that the connection has ended, and the email of Registered Client is removed by *Connected Clients*. The use case ends with success.



## Use Case Buy Ticket

**Level:** User Goal

**Priority:** 1

**Frequency:** Frequently

**Intention in Context:** A connected client wants to buy a ticket by choosing its number.

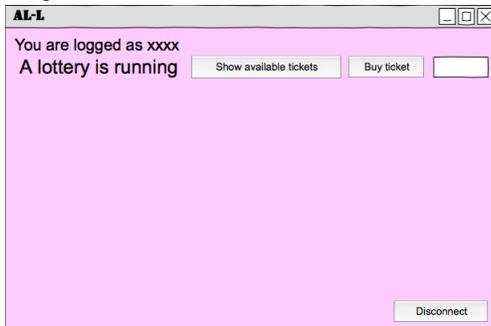
**Primary Actor:** Registered Client

**Secondary Actor:** Credit Card Service

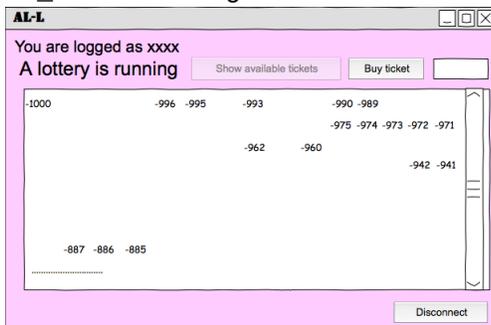
**Precondition:** The Registered Client email belongs to *Connected Clients* (\*\*12).

**Main Success Scenario:**

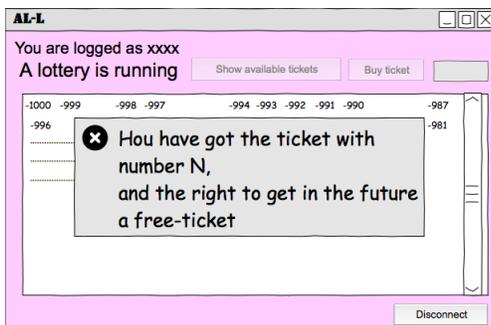
1. Registered Client asks AL\_L which are the available tickets.



2. AL\_L shows to Registered Client the *Available tickets* (\*\*10).



3. Registered Client asks to buy the ticket with number N.
4. If N belongs to *Available tickets*, then AL\_L asks Credit Card Service to charge 100 Euro to the credit card of the Registered Client recovered using *Registered Clients* (\*\*11).
5. Credit Card Service communicates to AL\_L that the amount has been charged.
6. AL\_L confirms to Registered Client that has got the ticket with number N, N is not any more among the *Available tickets* (\*\*10), and it is recorded in *Registered Clients* (\*\*11) that N is assigned to Registered Client. The use case ends with success.



**Extensions:**

- 4a.1 If N does not belong to *Available tickets*, then AL\_L communicates to the Registered Client that ticket N is not available. The use case ends with failure.
  
- 5a.1 Credit Card Service communicates to AL\_L that the amount cannot be charged.
- 5a.2 AL\_L informs Registered Client that the ticket price cannot be charged. The use case ends with failure.

## Use Case Give Free Tickets

**Level:** User Goal

**Priority:** 3

**Frequency:** Periodically

**Intention in Context:** Manager wants to give away some tickets to help finish the current lottery.

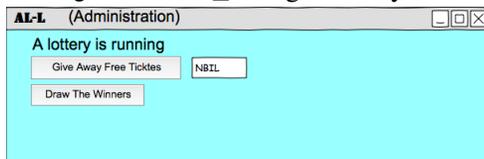
**Primary Actor:** Manager

**Secondary Actor:** Mail Server

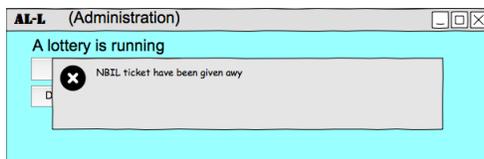
**Precondition:** The set of the *Available tickets* (\*\*10) is not empty and its size is less than the *Current Lottery Dimension* (\*\*7).

### Main Success Scenario:

1. Manager asks AL\_L to give away NBIL free tickets.

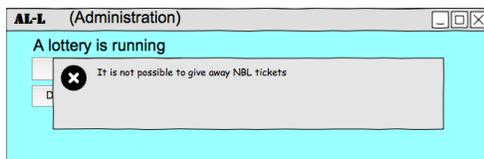


2. If NBIL is less or equal than the size of *Available tickets* (\*\*10), there are at least NBIL clients that have bought a ticket in the current lottery, and *Current Winning Order* applied to NBIL, *Available tickets* and *Registered Clients* returns (email<sub>1</sub>, nbil<sub>1</sub>), ..., (email<sub>NBIL</sub>, nbil<sub>NBIL</sub>), then AL\_L asks Mail Server to send an email to email<sub>1</sub>, ..., email<sub>NBIL</sub> informing them that they have received the free tickets nbil<sub>1</sub>, ..., nbil<sub>NBIL</sub> respectively; and such tickets are assigned to such clients by updating *Registered Clients*.
3. AL\_L informs Manager that the NBIL tickets have been given away. The use case ends with success.



### Extensions:

- 2a.1 If NBIL greater than the size of *Available tickets* (\*\*10) or there are not NBIL clients that have bought a ticket in the current lottery, then AL\_L informs Manager that it is not possible to give away NBL tickets. The use case ends with failure.



## Use Case Draw Winners

**Level:** User Goal

**Priority:** 1

**Frequency:** Periodically

**Intention in Context:** When *Available tickets* is empty the Manager draws the winners of the current lottery.

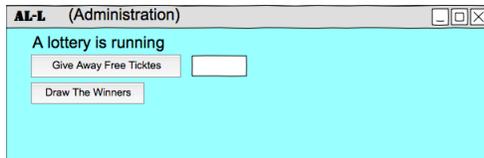
**Primary Actor:** Manager

**Secondary Actor:** Mail Server

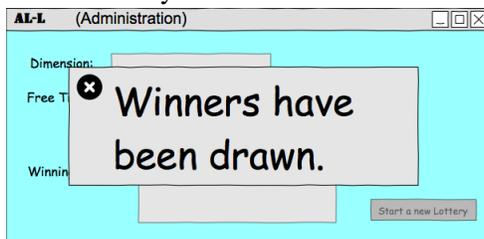
**Precondition:** *Running* is true (\*\*9) and *Available tickets* (\*\*10) is empty.

**Postcondition:** *Running* is false.

**Main Success Scenario:**



1. The Manager asks AL\_L to draw the winners.
2. If  $em_1$ ,  $em_2$  and  $em_3$  are the emails recovered by *Registered Clients* of the clients that bought the three tickets whose numbers are the three highest with respect to *Current Winning Order*, then AL\_L asks Mail Server to send an email to  $em_1$ ,  $em_2$  and  $em_3$  informing them that they won.
3. AL\_L asks Mail Server to send an email to all registered clients determined by *Registered Clients* informing them that the winners of the current lottery have been drawn.
4. AL\_L informs the Manager that the winners have been drawn and that the current lottery is terminated. The use case ends with success.



**Use Case Start Lottery****Level:** User Goal**Priority:** 1**Frequency:** Periodically**Intention in Context:** The Manager wants to start a new lottery**Primary Actor:** Manager**Secondary Actor:** Mail Server**Precondition:** *Running* is true (\*\*9)**Postcondition:** *Running* is false.**Main Success Scenario:**

1. The Manager asks AL\_L to start a new lottery, and gives its dimension, its free ticket law (\*\*1) and its winning order (\*\*2).
2. AL\_L asks Mail Server to send an email to all *Registered Clients* (\*\*11) informing them that a new lottery has started. *Running* (\*\*9) will become true, and *Current Lottery Dimension* (\*\*7), *Current Free Ticket Law* and *Current Winning Order* (\*\*8) are updated with the received data. The use case ends with success.