|  |
| --- |
| LONDON 2012 OLYMPICS MANAGEMENT SYSTEM |
| SWE 316 – Software Design and Architecture |
| Phase 2 - Report |
|  |
| **Team 1 – Abdulaziz, Shaeq, Yazeed and Ibrahim** |
| **5/10/2010** |

|  |
| --- |
| Phase 2 of this report contains alternative package structures for the use case model, sequence diagrams for objects participating in each use case and state charts, depicting different states of the state dependent objects. |

Contents

[ALTERNATIVE PACKAGES 3](#_Toc261293955)

[Alternative Package 1 4](#_Toc261293956)

[Alternative Package 2 5](#_Toc261293957)

[SEQUENCE DIAGRAMS 7](#_Toc261293958)

[1. Schedule Game 8](#_Toc261293959)

[2. Reschedule Game 9](#_Toc261293960)

[3. Check Schedule 10](#_Toc261293961)

[4. Register Athlete 11](#_Toc261293962)

[5. Buy Ticket 12](#_Toc261293963)

[6. Issue Ticket 17](#_Toc261293964)

[7. Request Ticket (Athlete) 18](#_Toc261293965)

[8. Request Security 21](#_Toc261293966)

[9. Assign Security 22](#_Toc261293967)

[10. Request Autograph Session 23](#_Toc261293968)

[11. View Schedule 24](#_Toc261293969)

[12. Schedule Ceremony 25](#_Toc261293970)

[13. Reschedule Ceremony 26](#_Toc261293971)

[STATE CHARTS 27](#_Toc261293972)

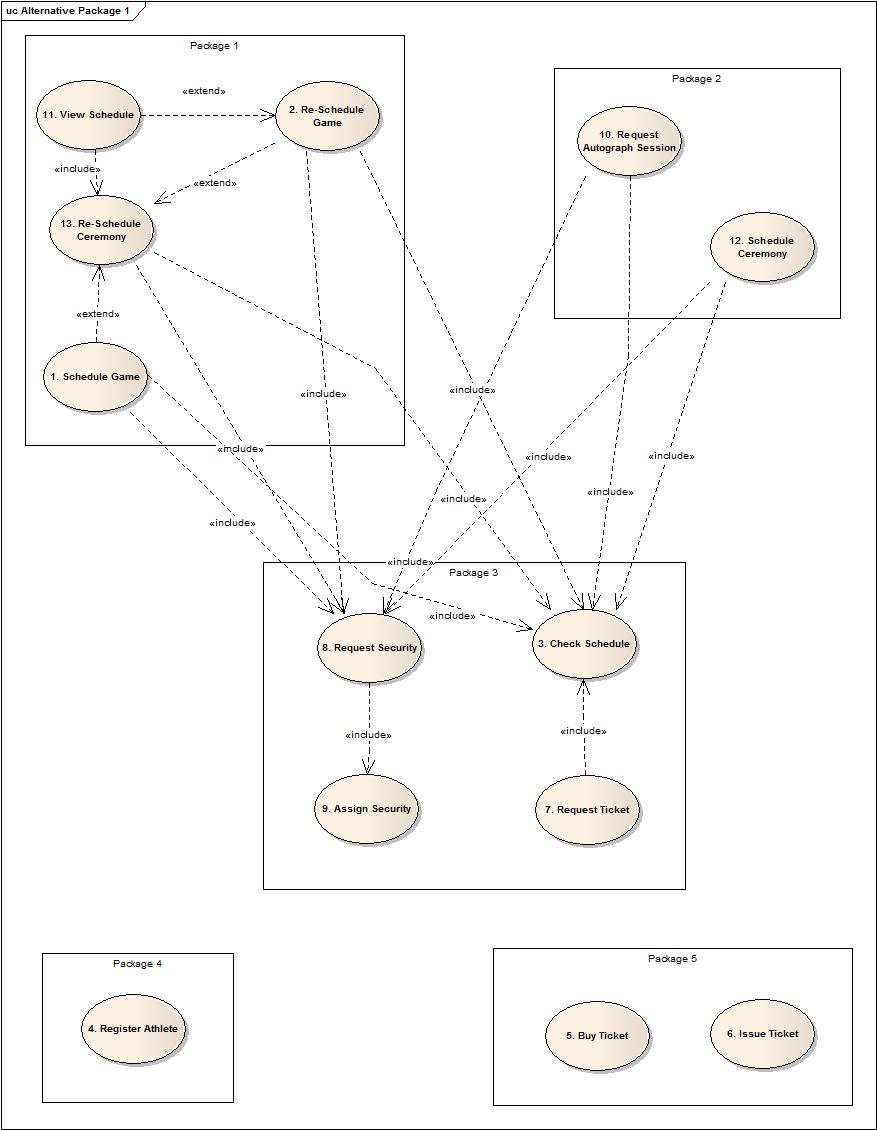
[Add Event to the system 28](#_Toc261293973)

[Buying a ticket 29](#_Toc261293974)

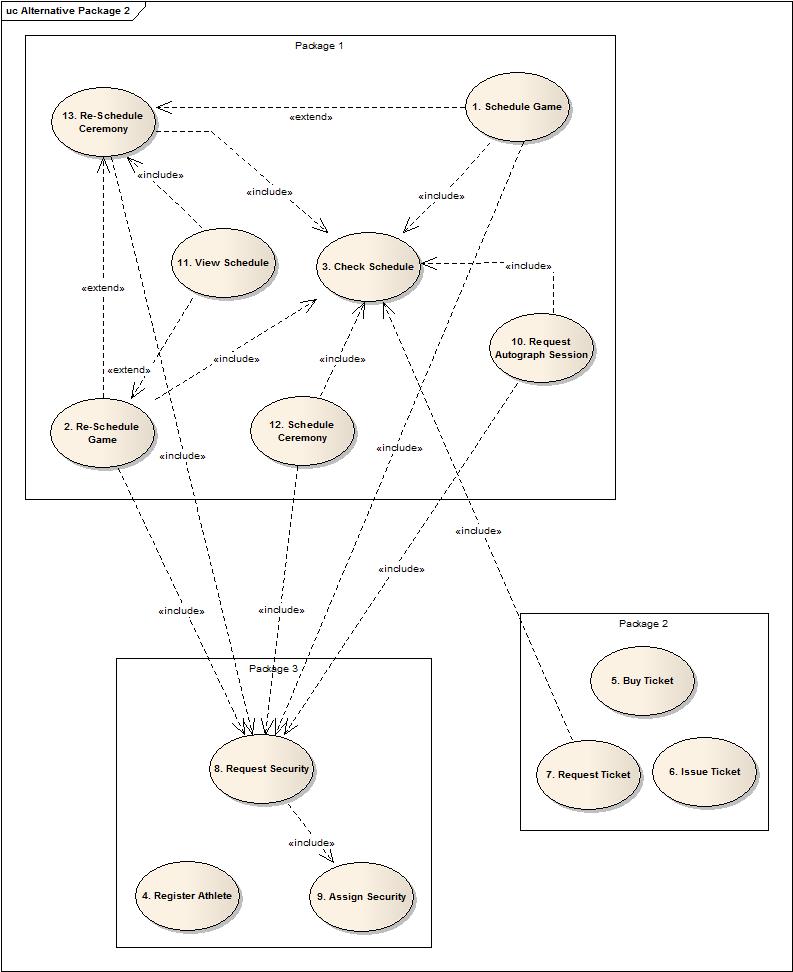
[Requesting for security 30](#_Toc261293975)

# ALTERNATIVE PACKAGES

## Alternative Package 1



## Alternative Package 2



We are anticipating that the packages used in this system will be used by future Olympic organizers for their system. We base the grouping of our classes into packages on the “Common Reuse Principle” where the classes that aren’t reused together are not grouped together.

In the design for the first package, the number of packages that change in a new release are lesser are package 1, 2 and 3 from the design above.

In case of the second package, the new release will make us update all the 3 packages in the system. This means we will have to rebuild almost the entire system, test it and then deploy the release.

So we choose the first package as a better design.

Also since we are applying the CRP principle for this system, we want to have small packages in the system which is satisfied by the first package.

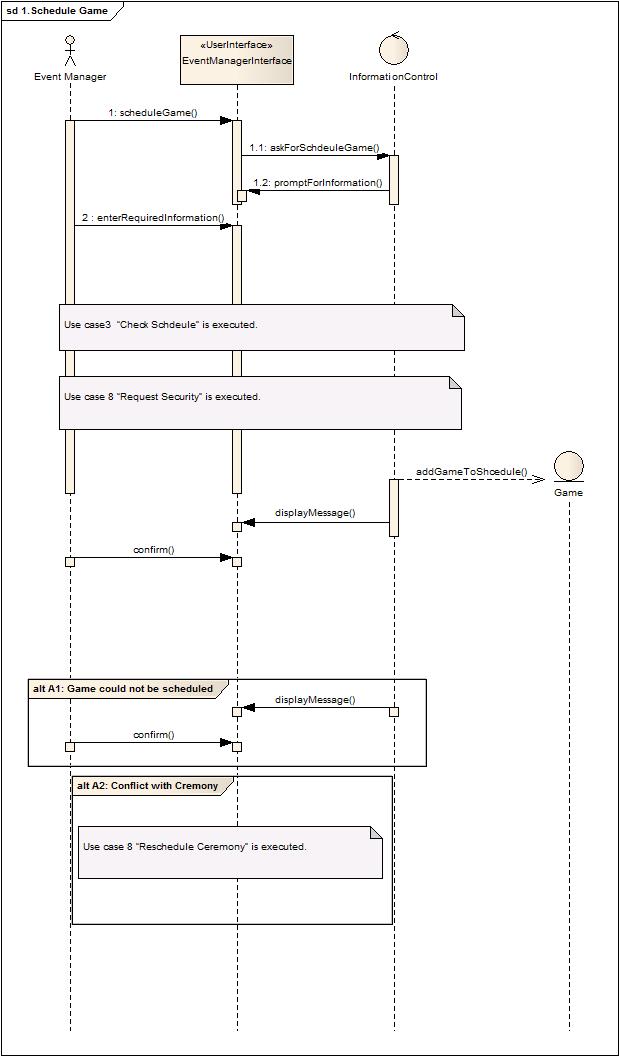
None of the dependencies in both package suggestions form a cycle, so this problem is ignored for this case.

For the first package, the stable package in the design is “Package 3” since most of the use cases in Package one and two use the include relation for the use cases in package 3. This gives us a clear reason to declare classes involved with package 3 to be abstract classes.

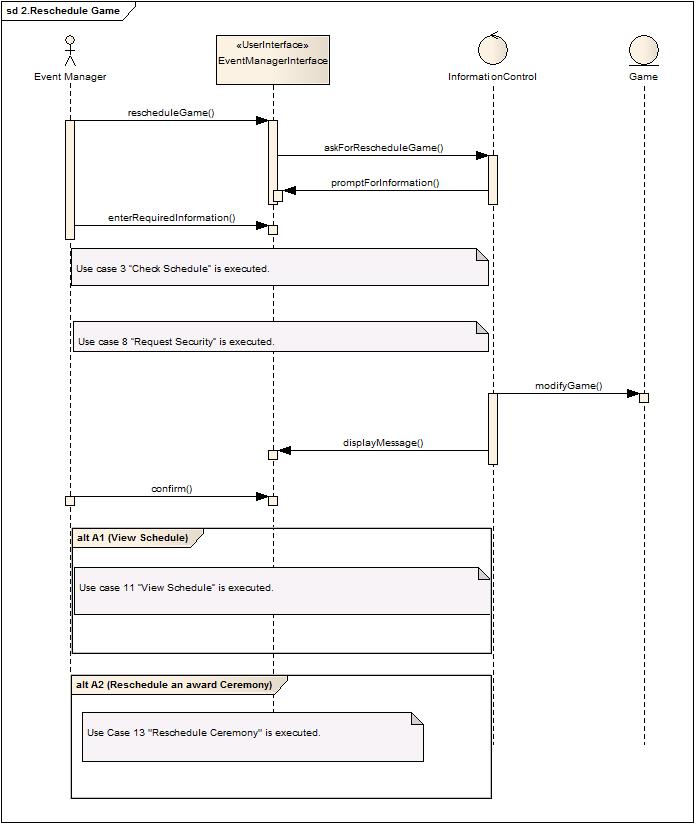
So overall, we feel that Package 1 is a better implementation than Package 2.

# SEQUENCE DIAGRAMS

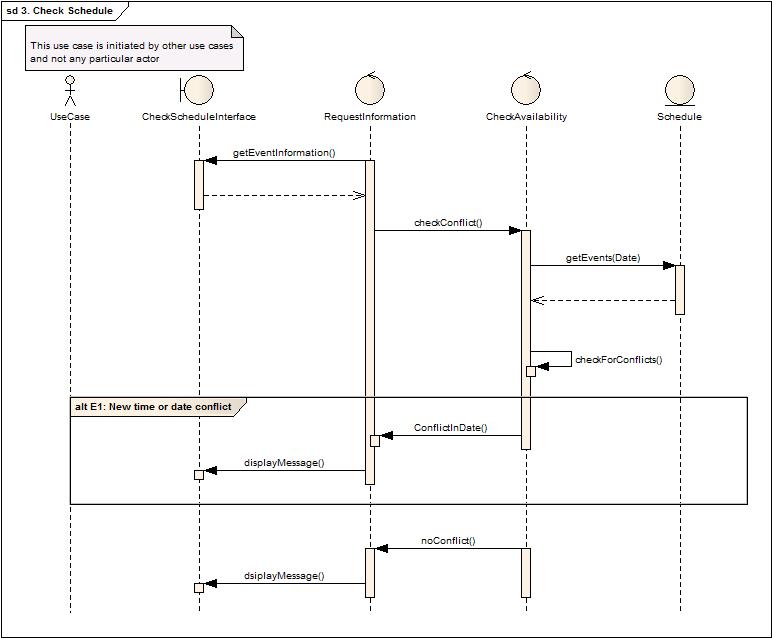
1. Schedule Game



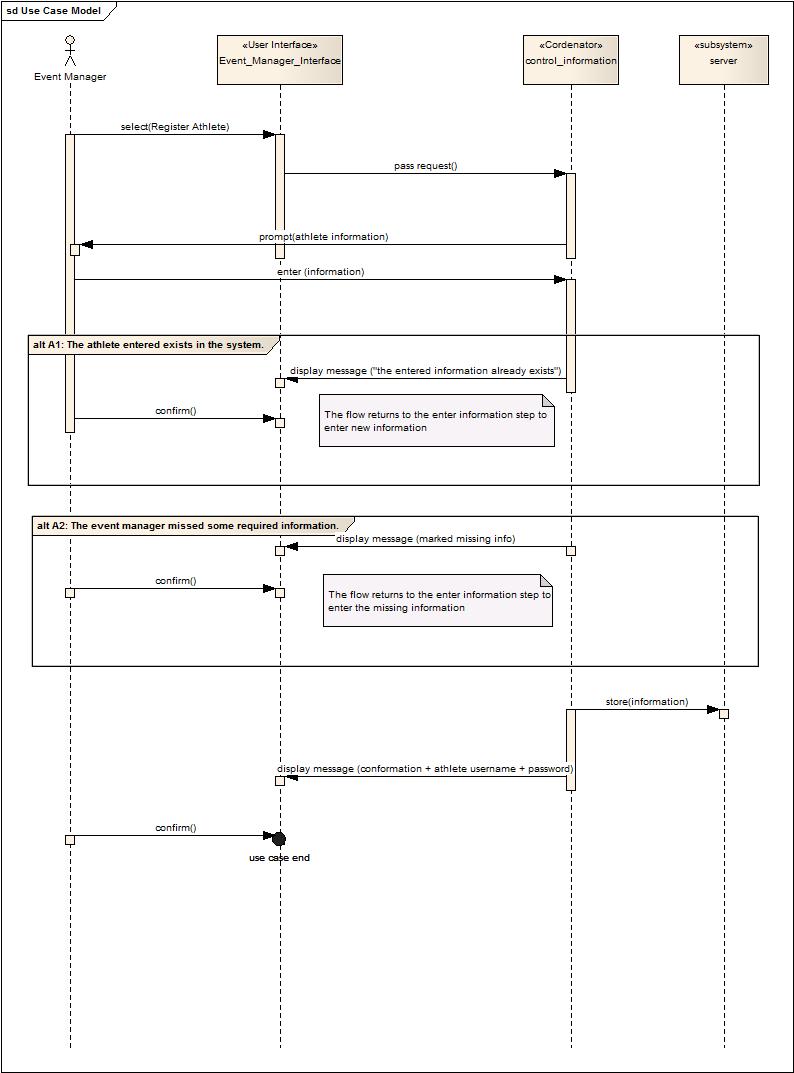
1. Reschedule Game



1. Check Schedule

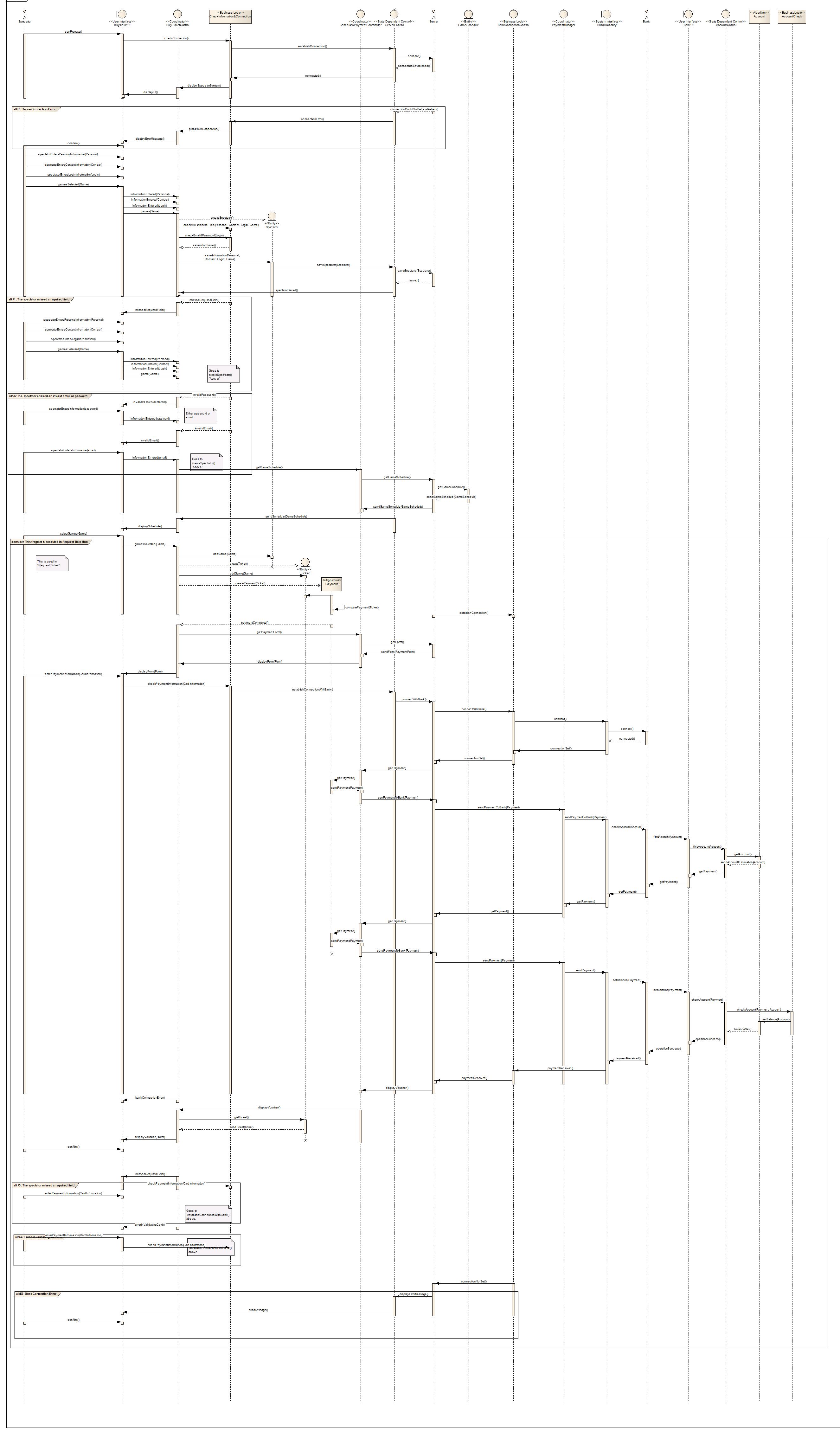


1. Register Athlete

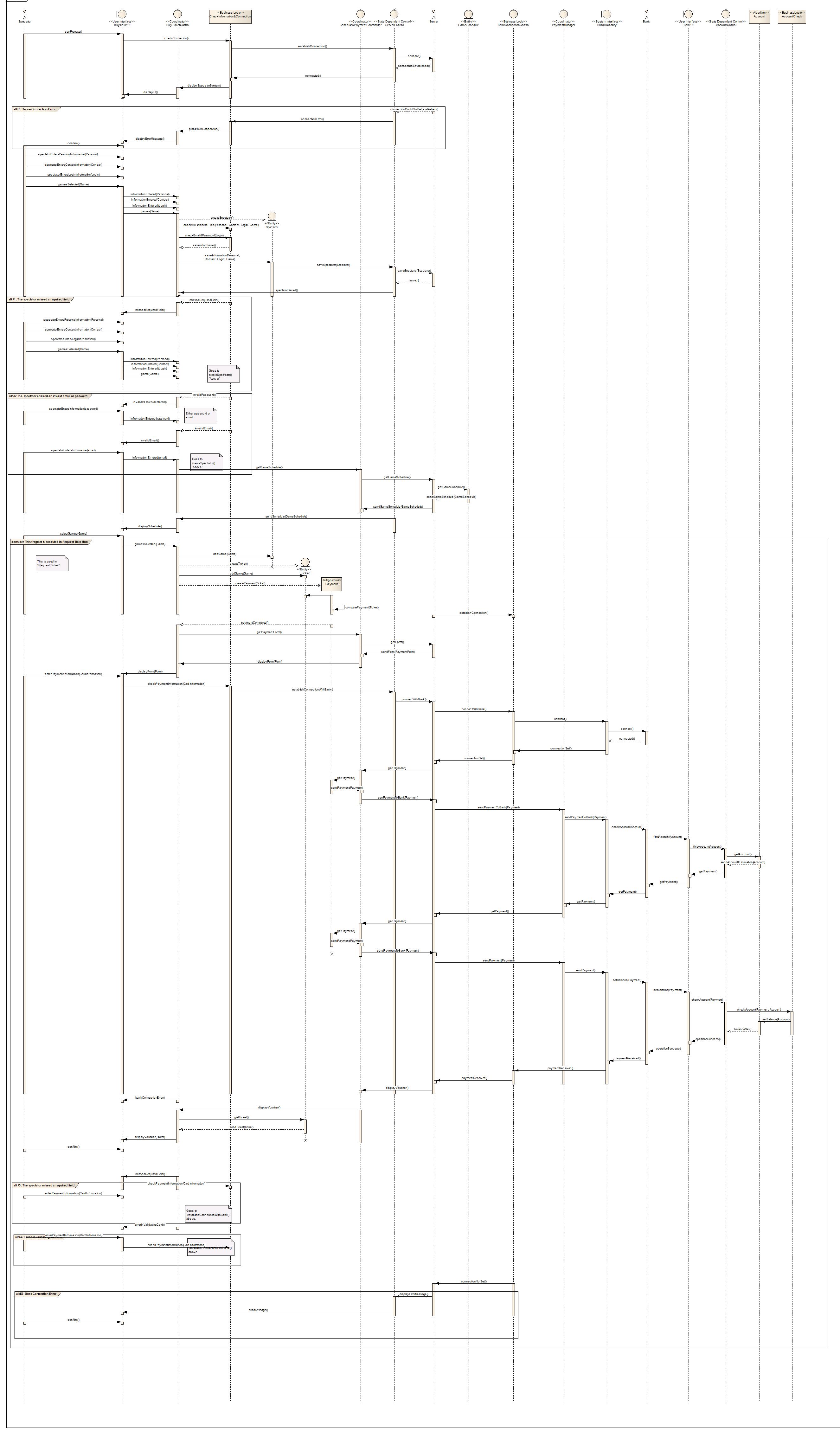


1. Buy Ticket

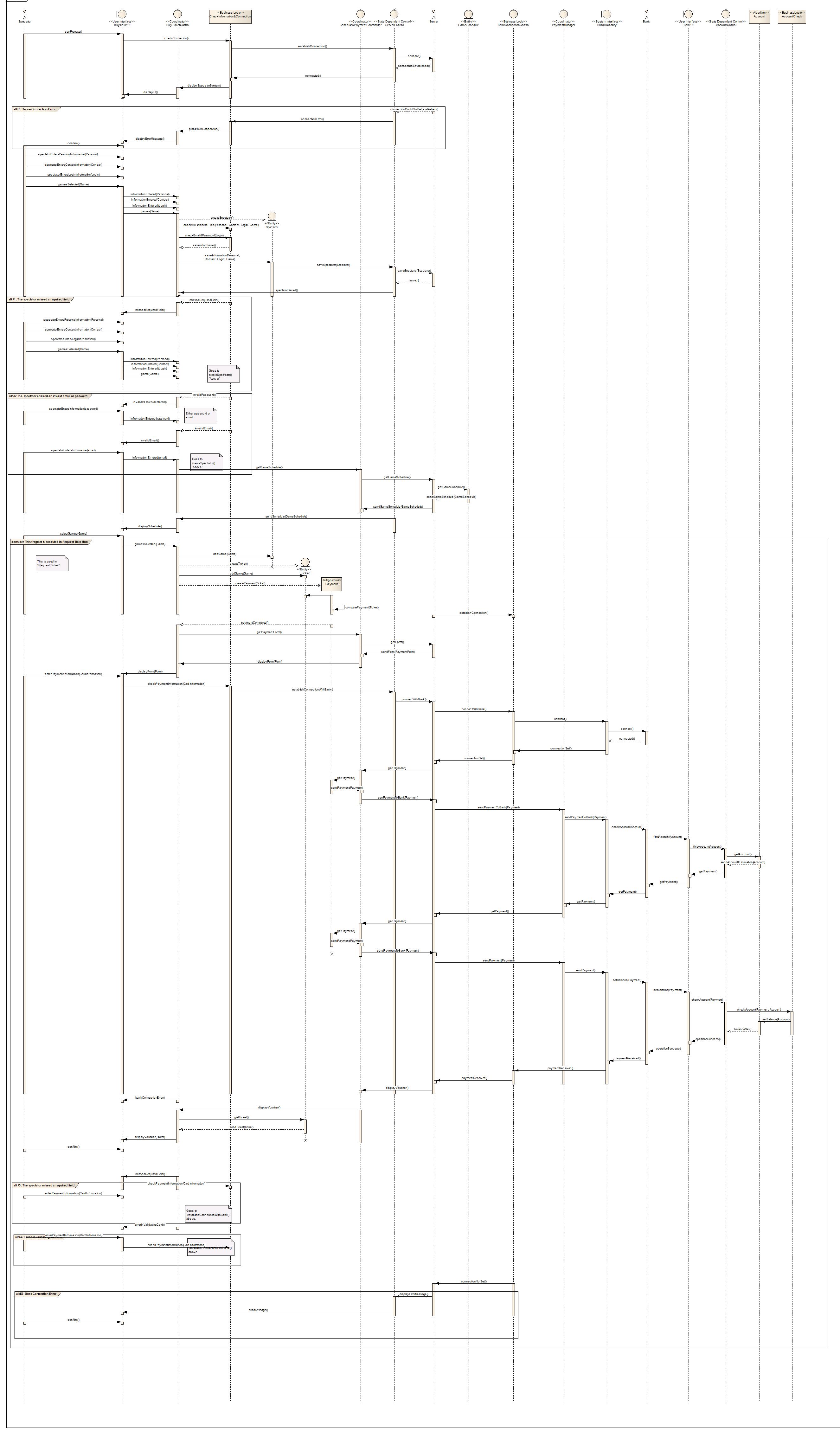
Part 1



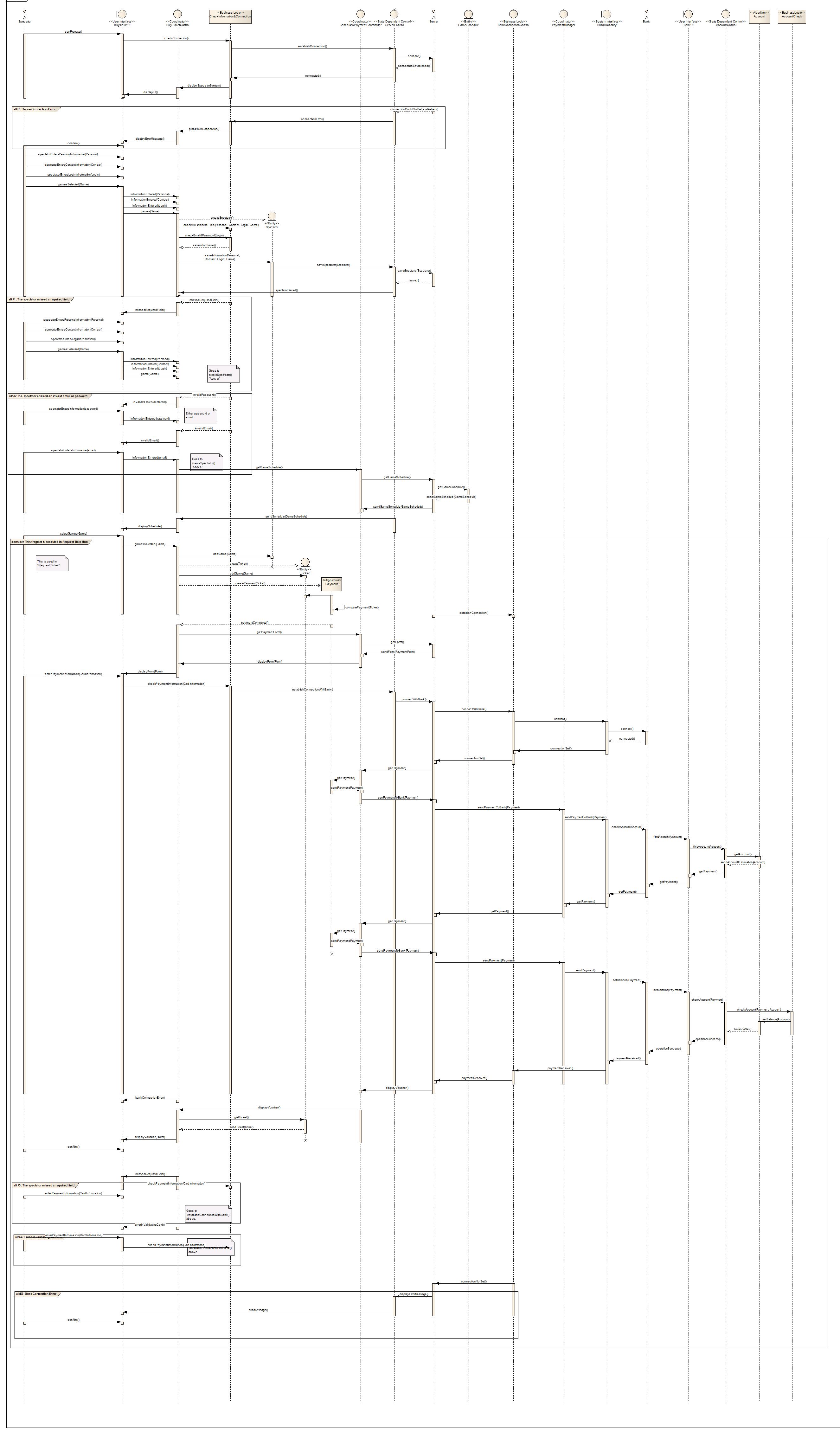
Part 2



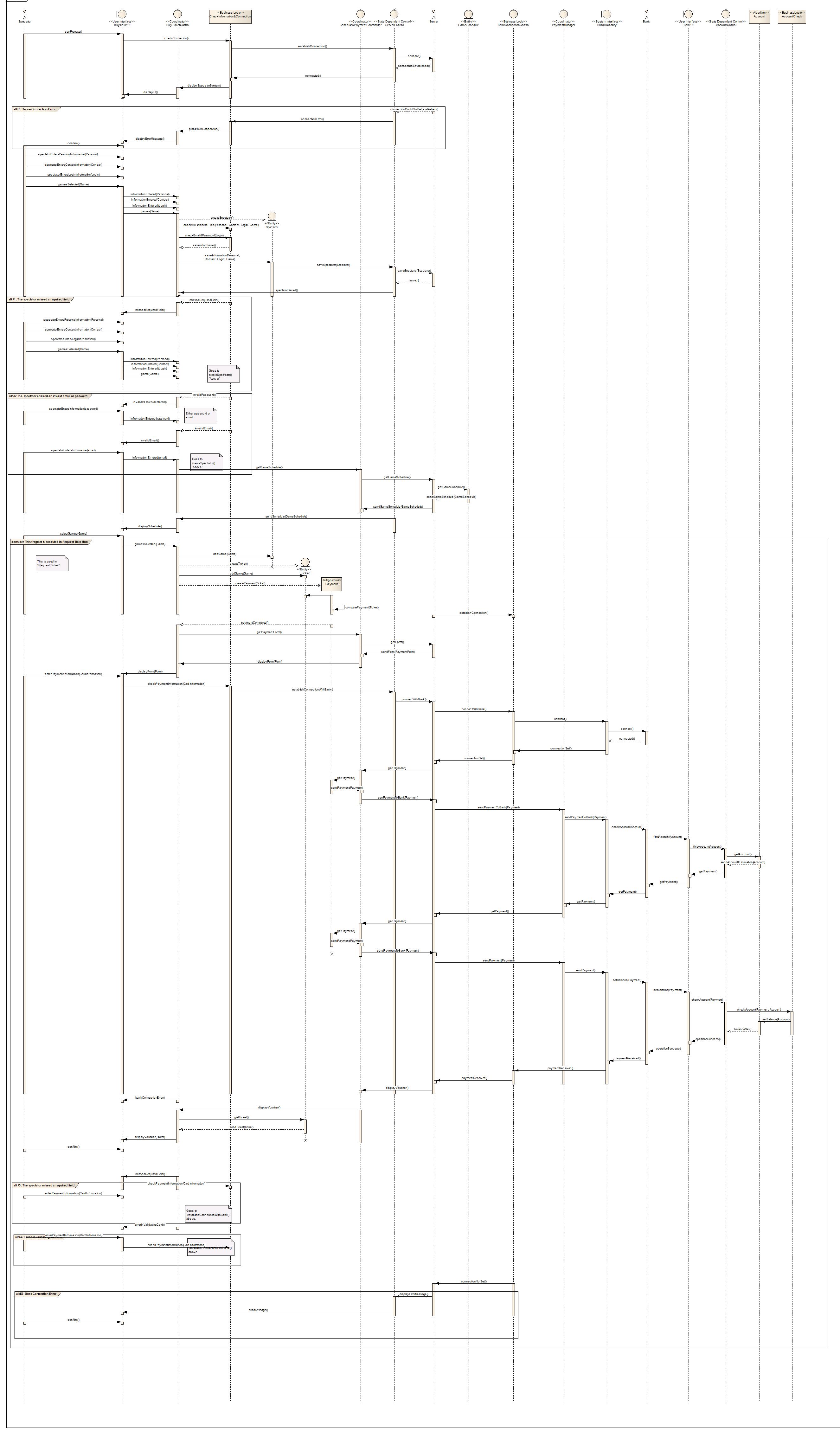
Part 3



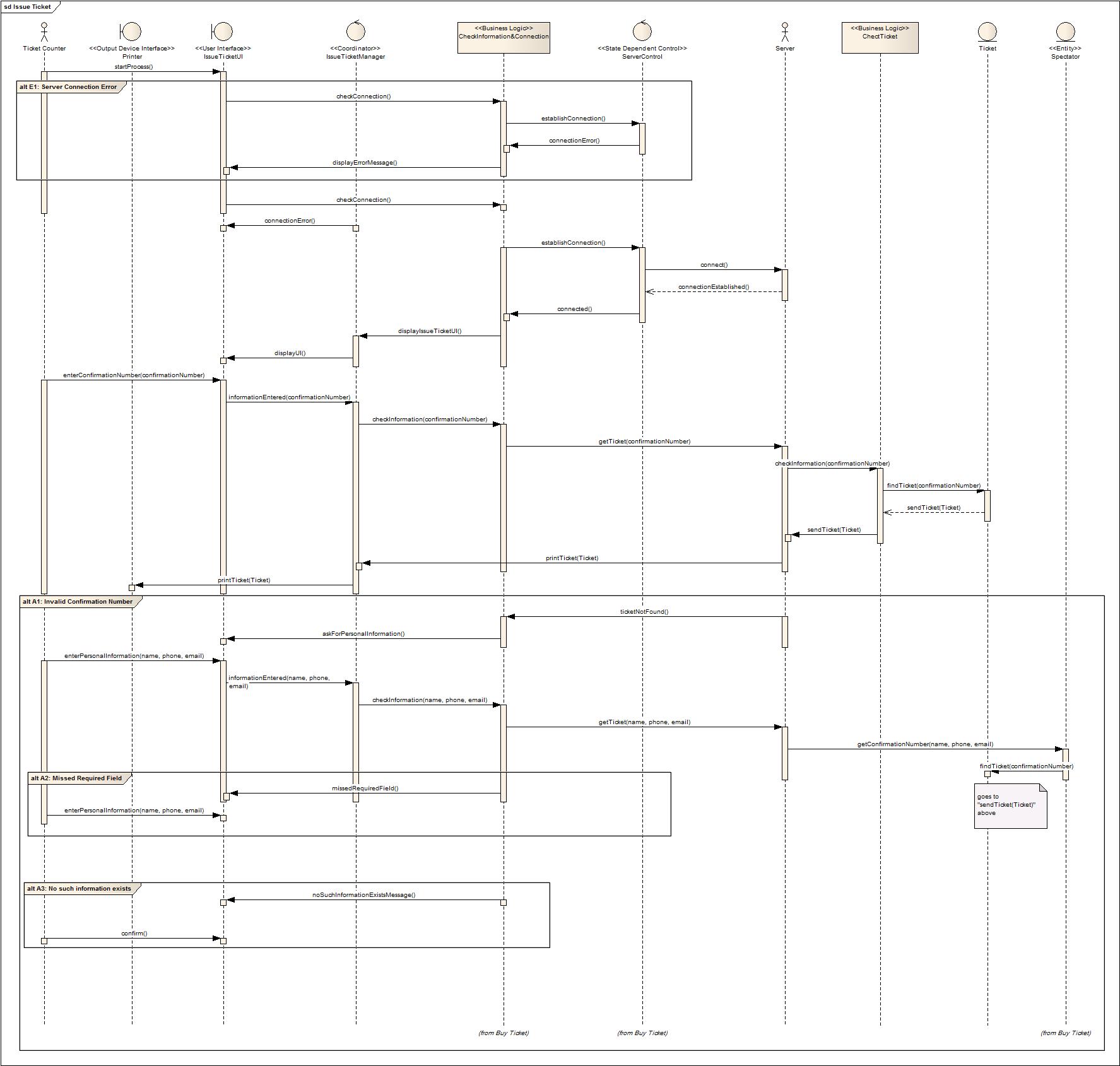
Part 4



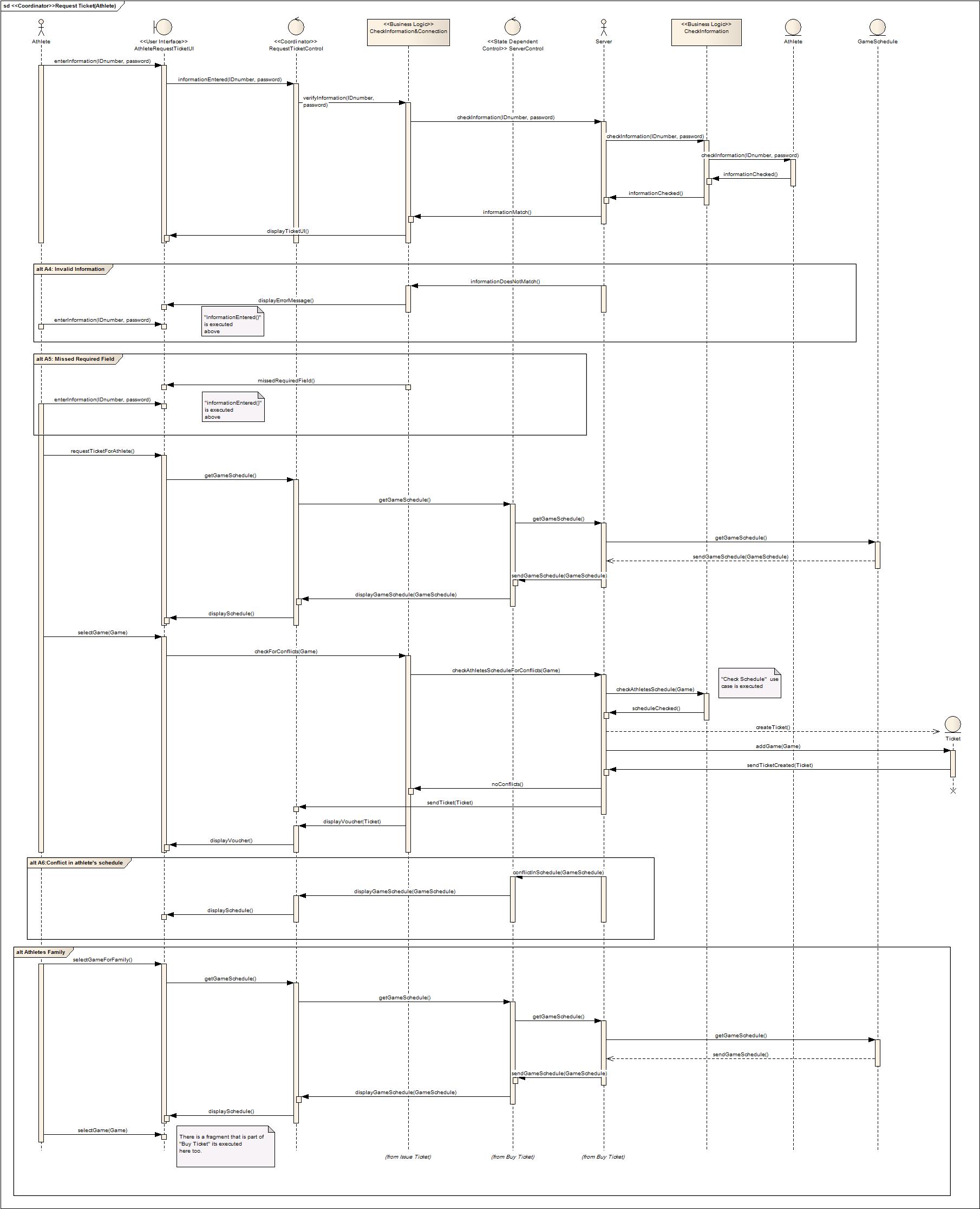
Part 5



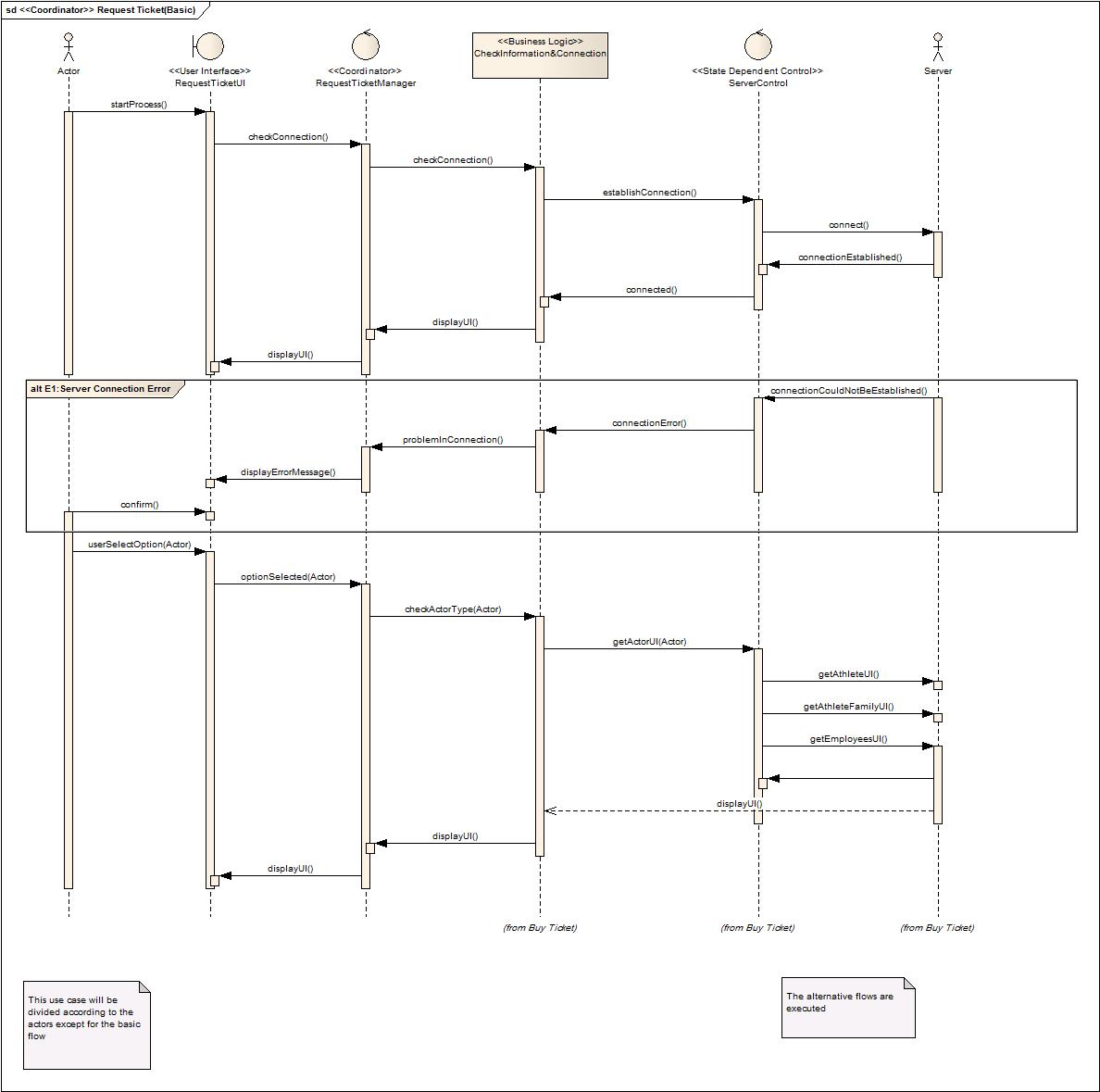
1. Issue Ticket



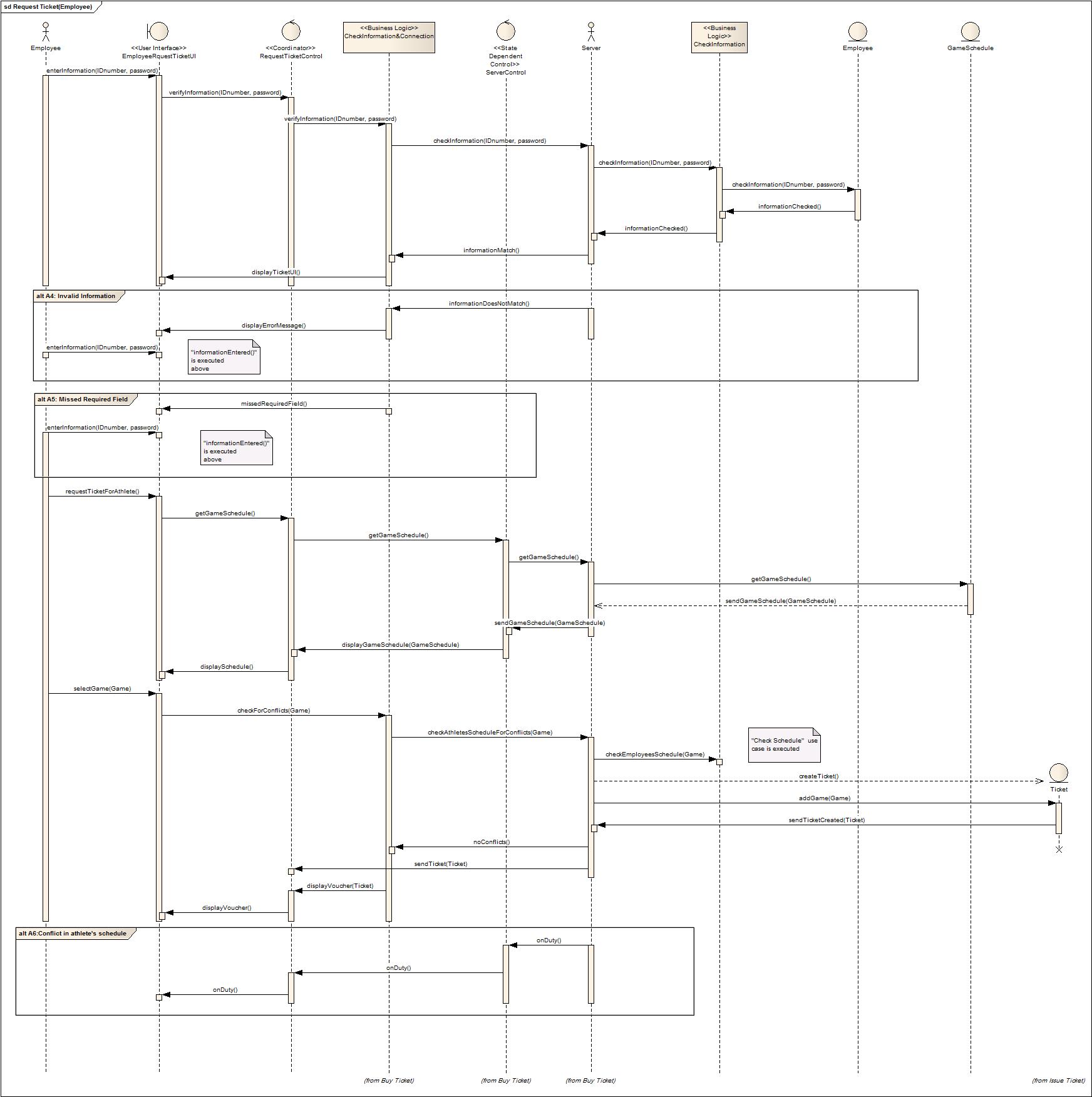
1. Request Ticket (Athlete)



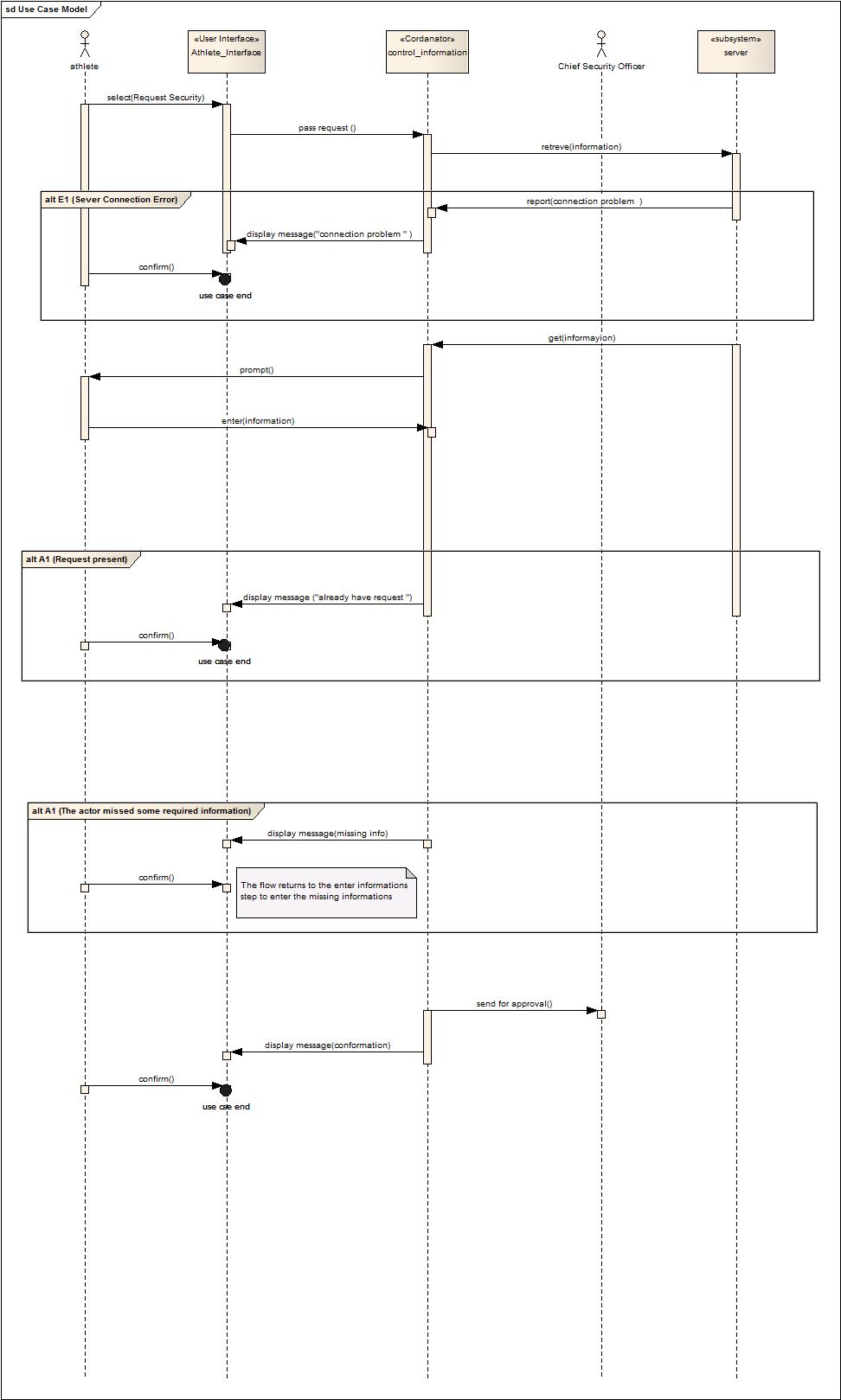
Request Ticket (Basic)



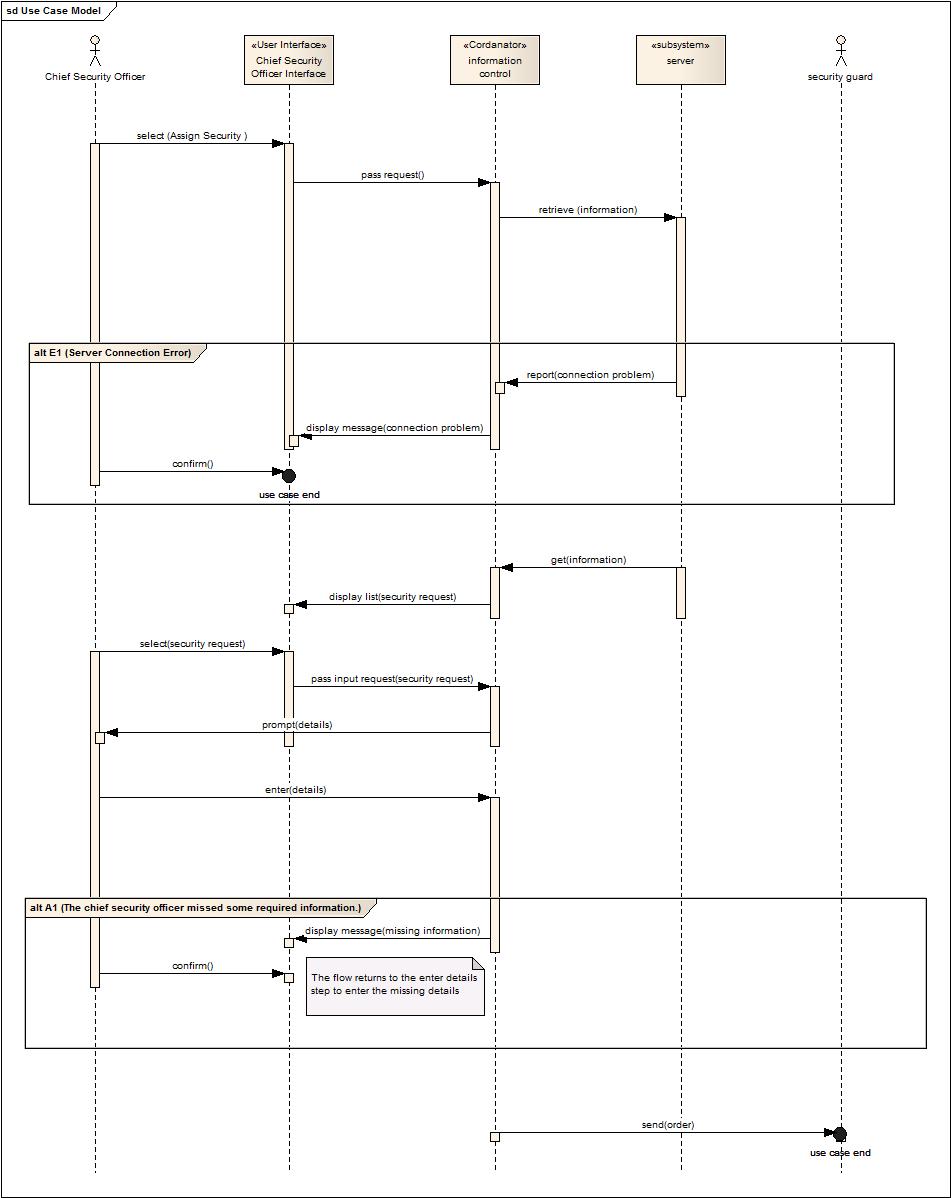
Request Ticket (Employee)



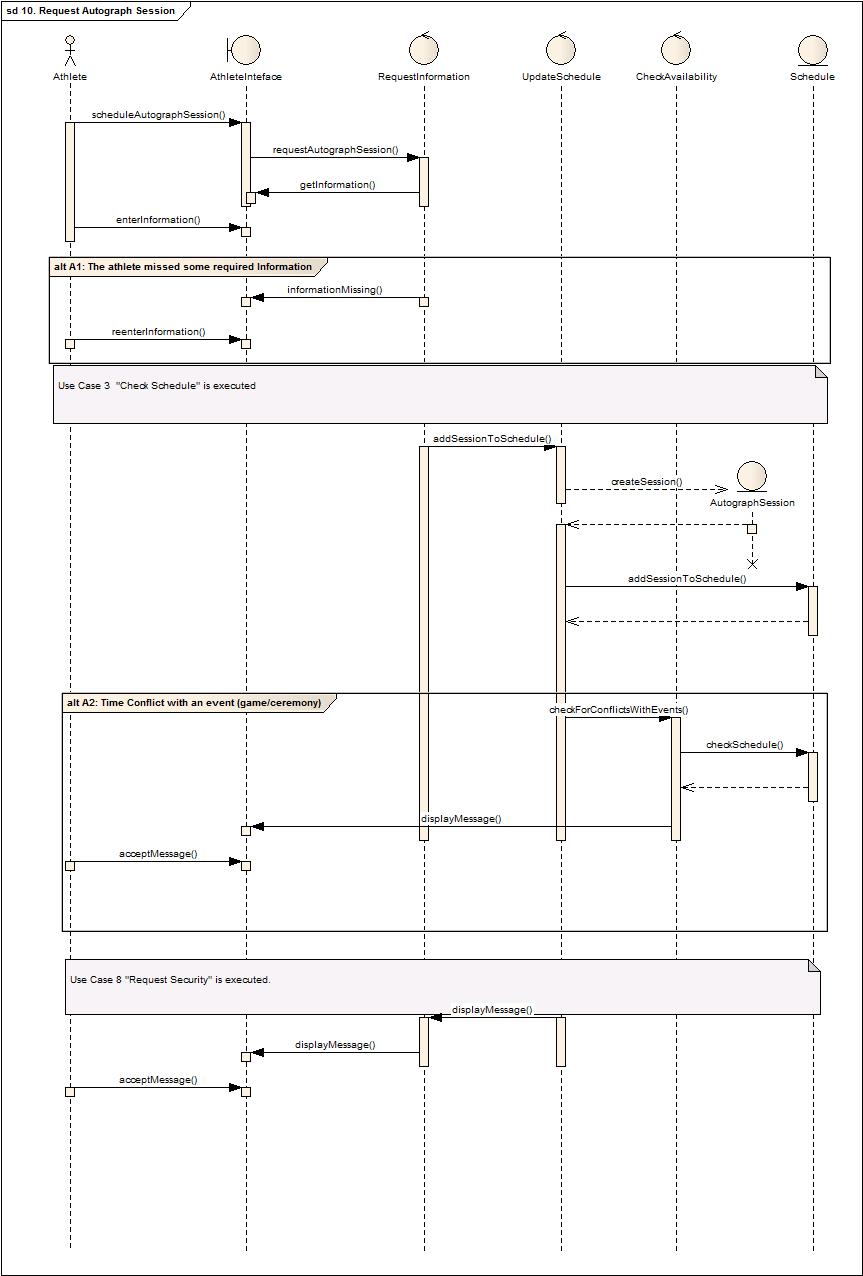
1. Request Security



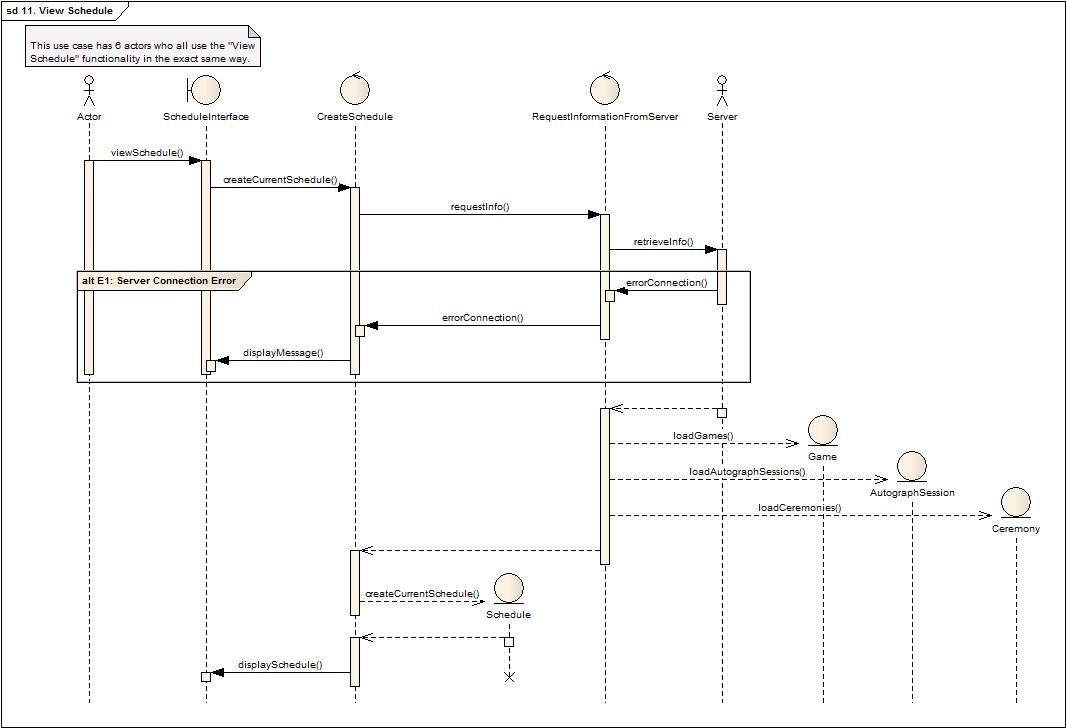
1. Assign Security



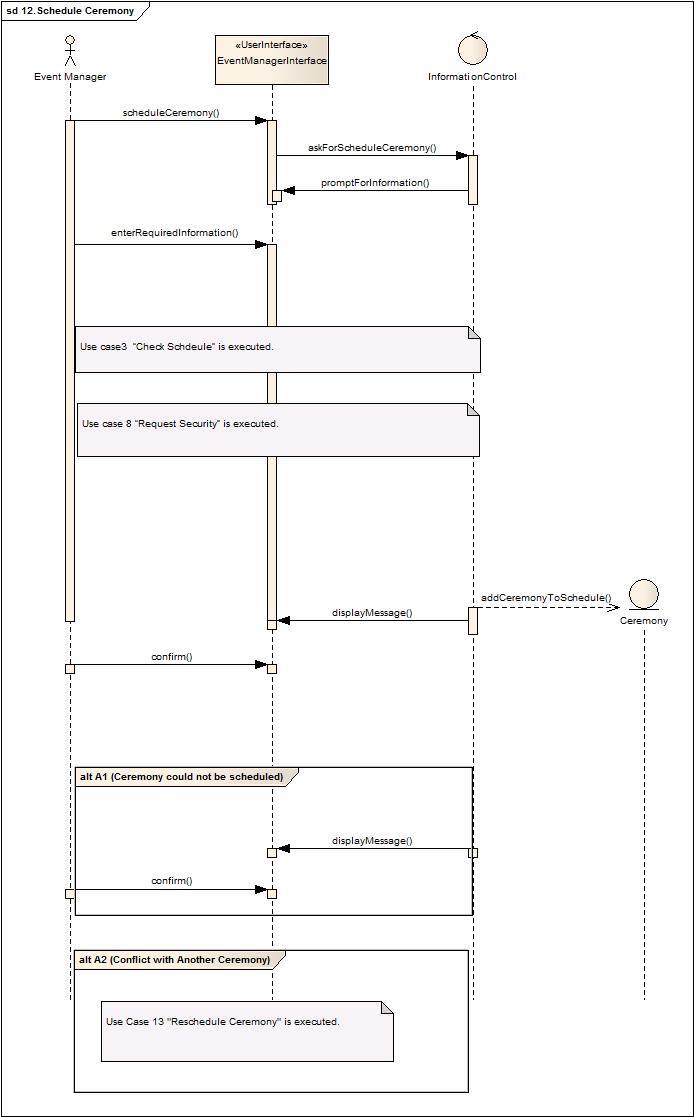
1. Request Autograph Session



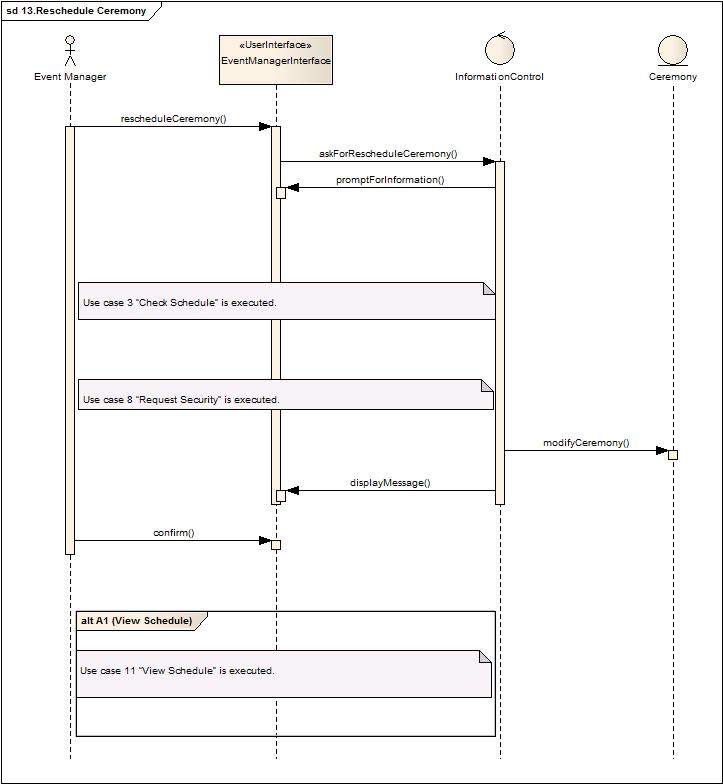
1. View Schedule



1. Schedule Ceremony

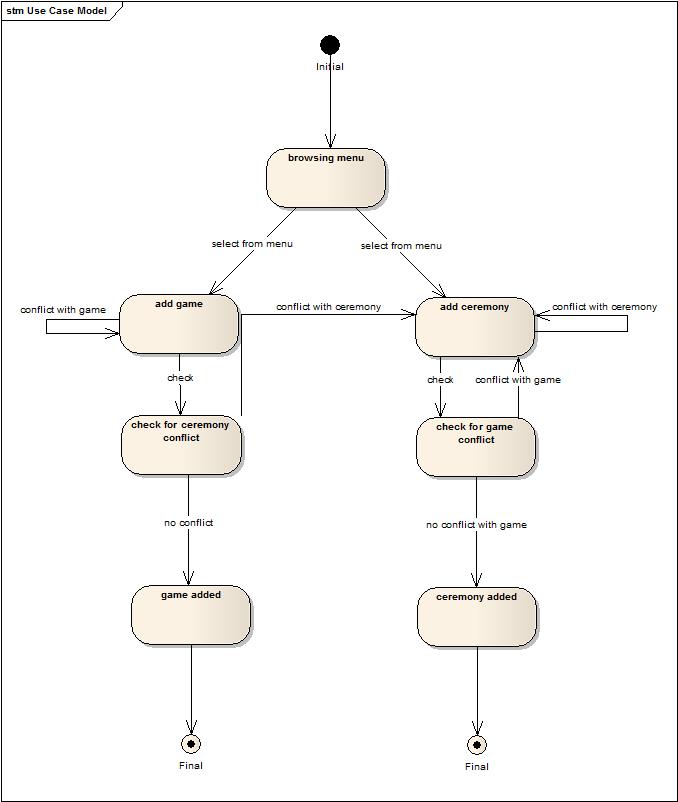


1. Reschedule Ceremony

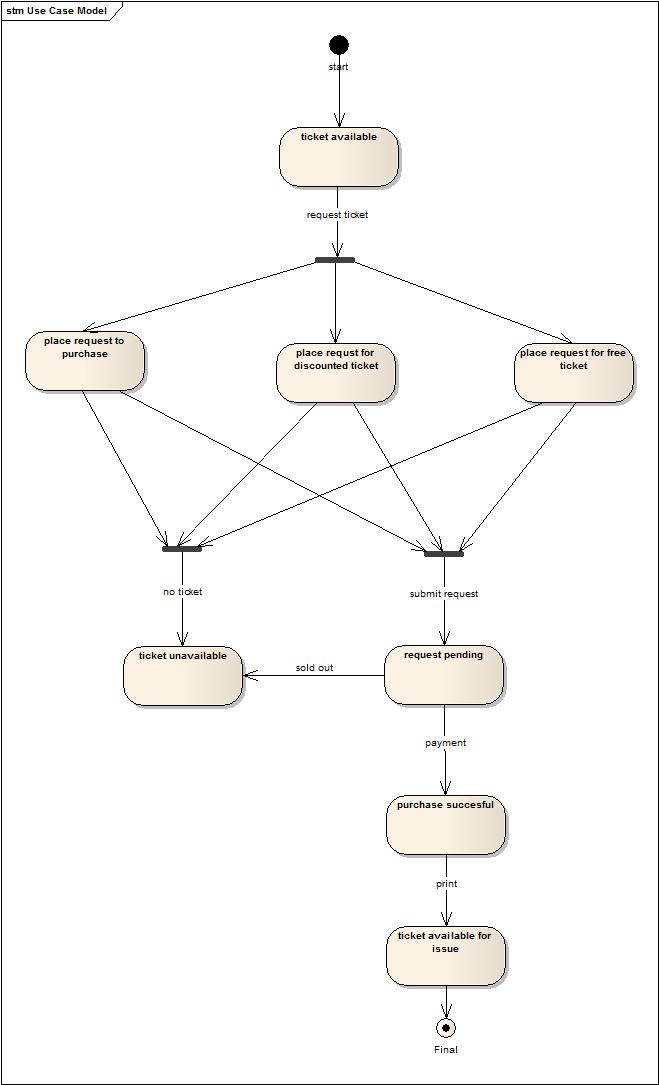


# STATE CHARTS

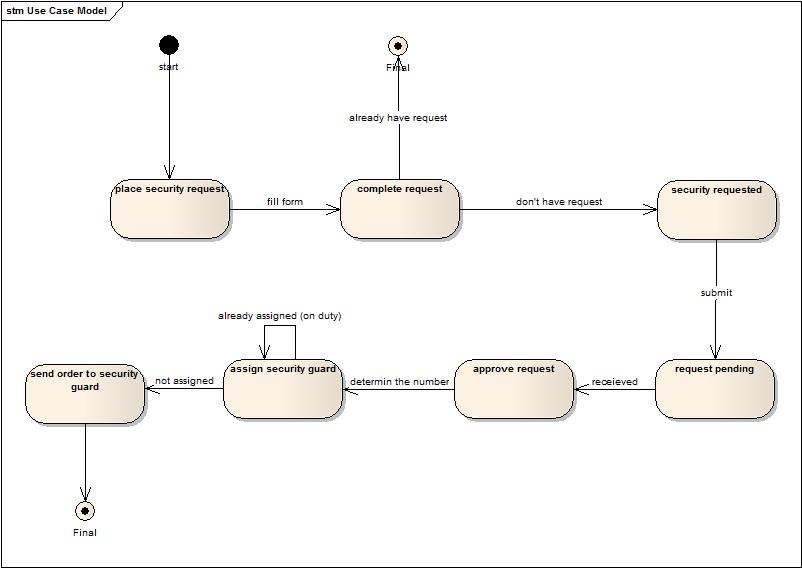
## Add Event to the system



## Buying a ticket



## Requesting for security



**END OF DOCUMENT**