|  |
| --- |
|  |

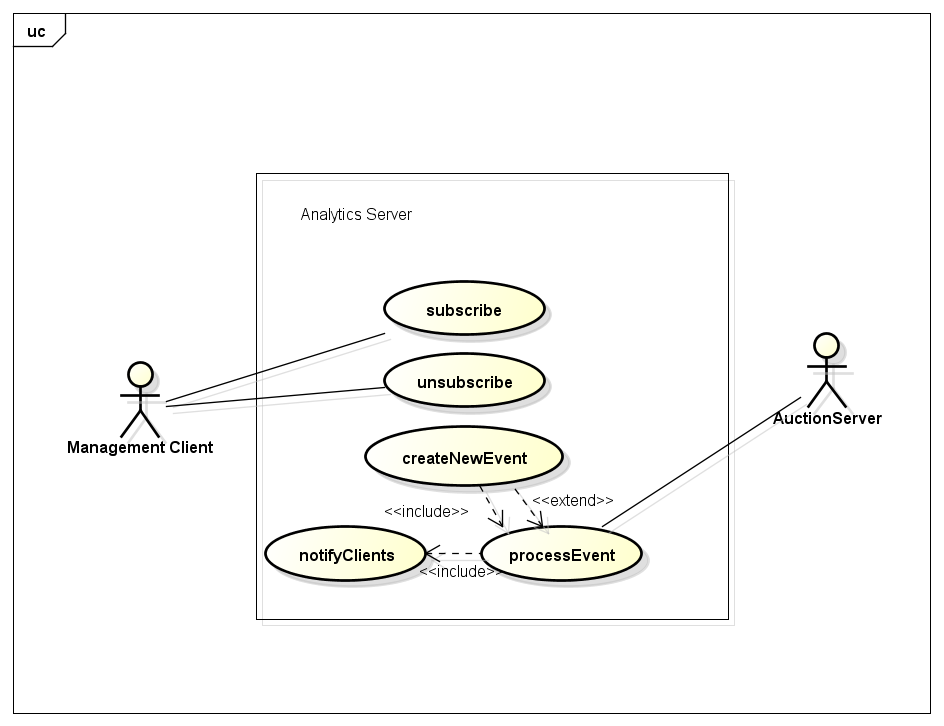
|  |
| --- |
| **Task 08** |
| RMI Auctionsystem |
|  |
|  |
|  |
| **Krepela, Lipovits, Reichmann, Tattyrek, Traxler** |
| **29.01.2014** |
|  |

Insert Specification Here

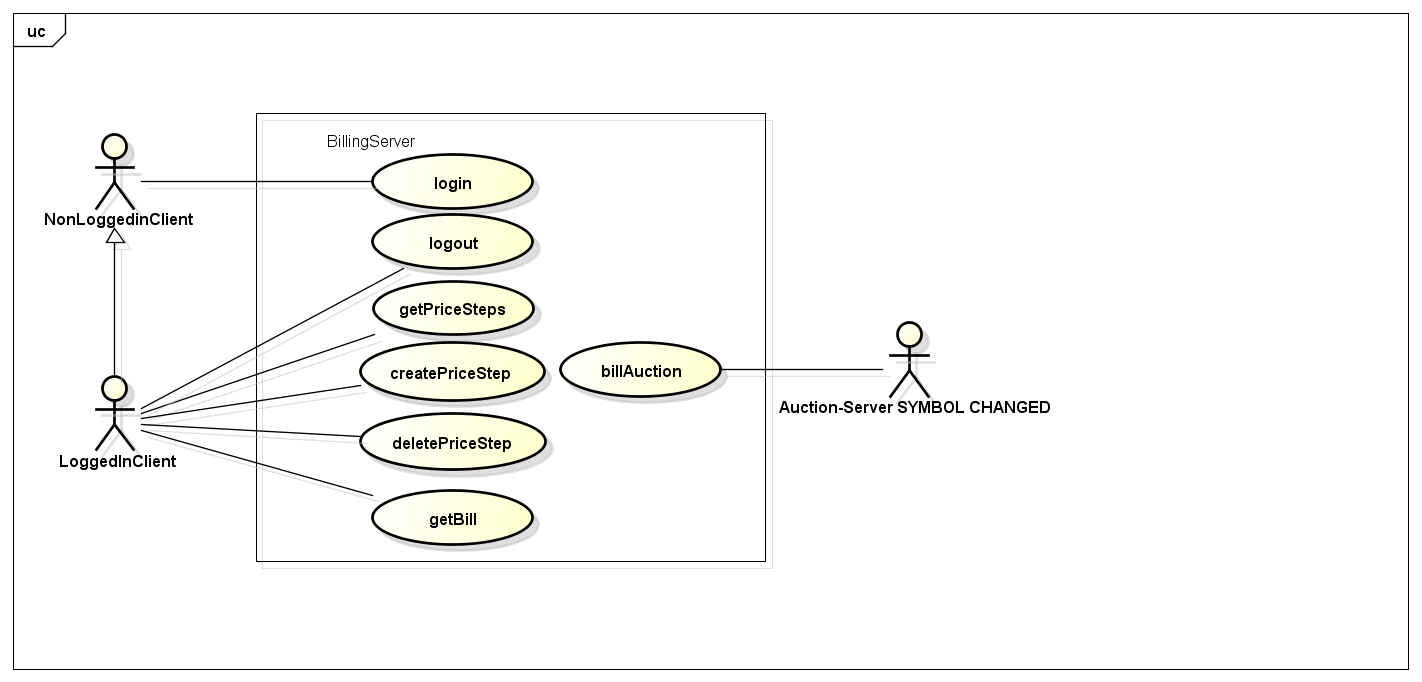
Designüberlegung

## Use Case

Analytic Server:



BillingServer:



## AktivitätsdiagrammAuktionen anzeigen.jpgAuktionsende.jpgBieten.jpgClient Login.jpgClient Logout.jpgEinschreiben.jpgErstellen .jpgEvent verarbeiten.jpgManagement Login.jpgManagement Logout.jpgStep bearbeiten.jpgSteps anzeigen.jpg

## Klassendiagramme

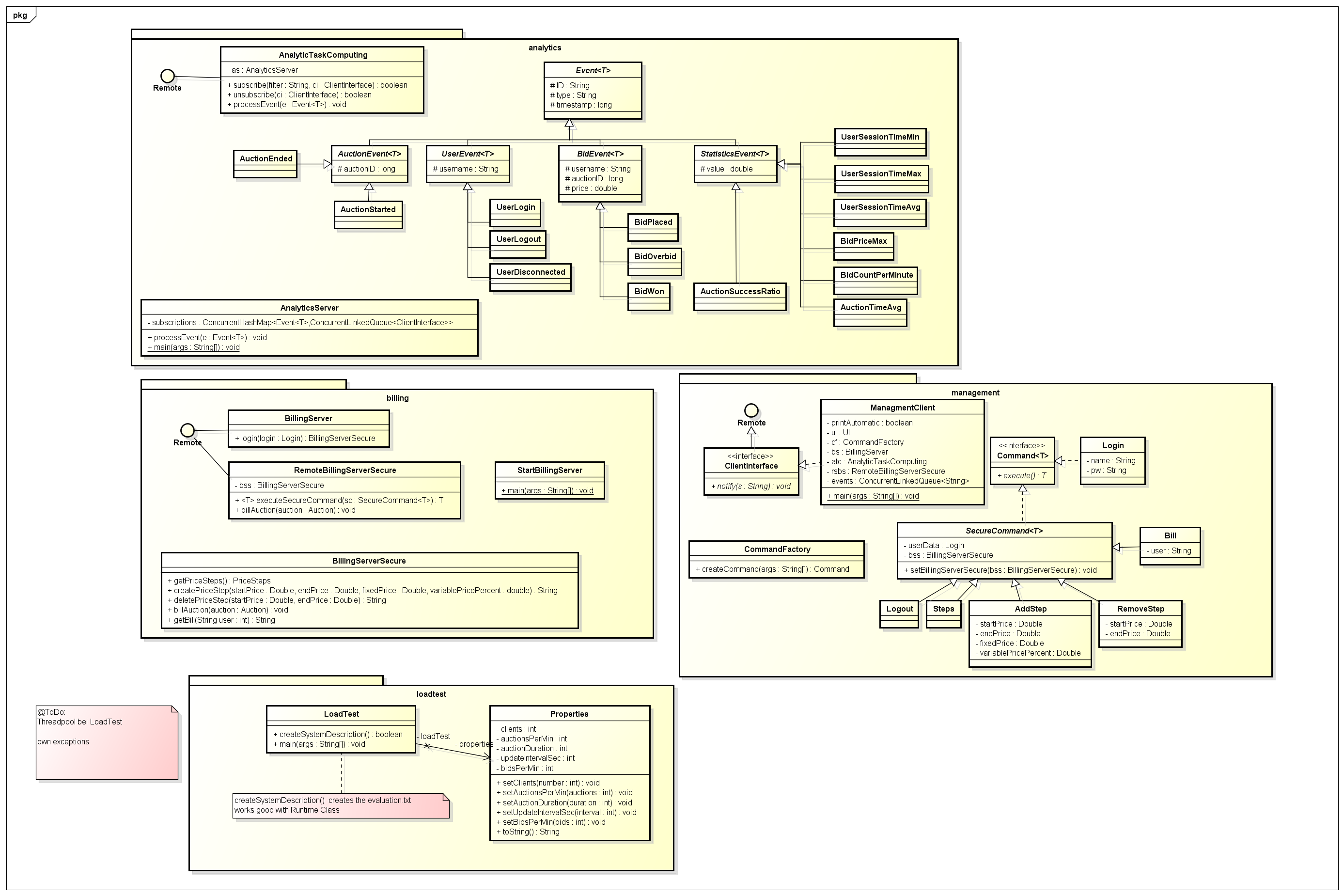
### Analytic + Events

### C:\Users\Daniel\Desktop\Class Diagram2.pngBilling

### C:\Users\Daniel\Desktop\Class Diagram2.png

### ManagemenclientC:\Users\Daniel\Desktop\Class Diagram2.png

### Testing Component



### Client

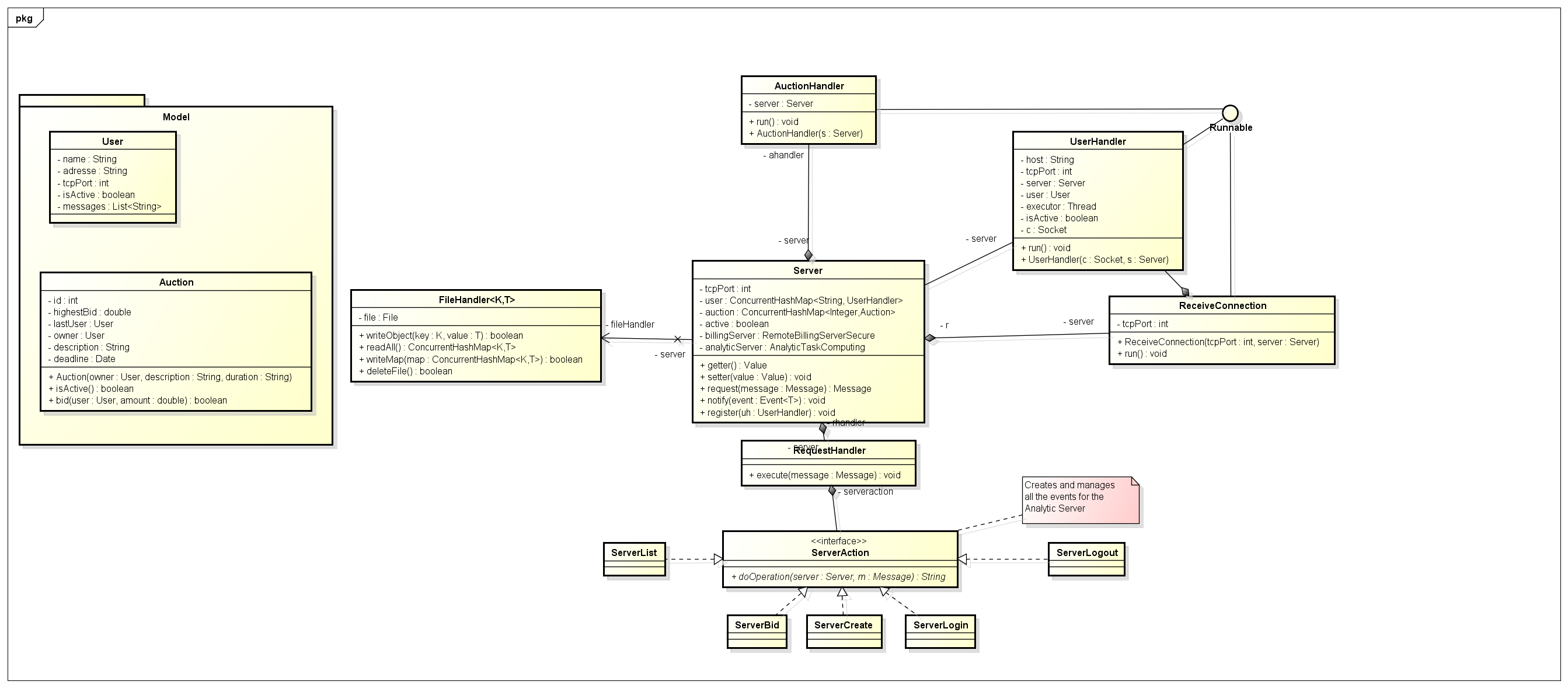


### Message-Model

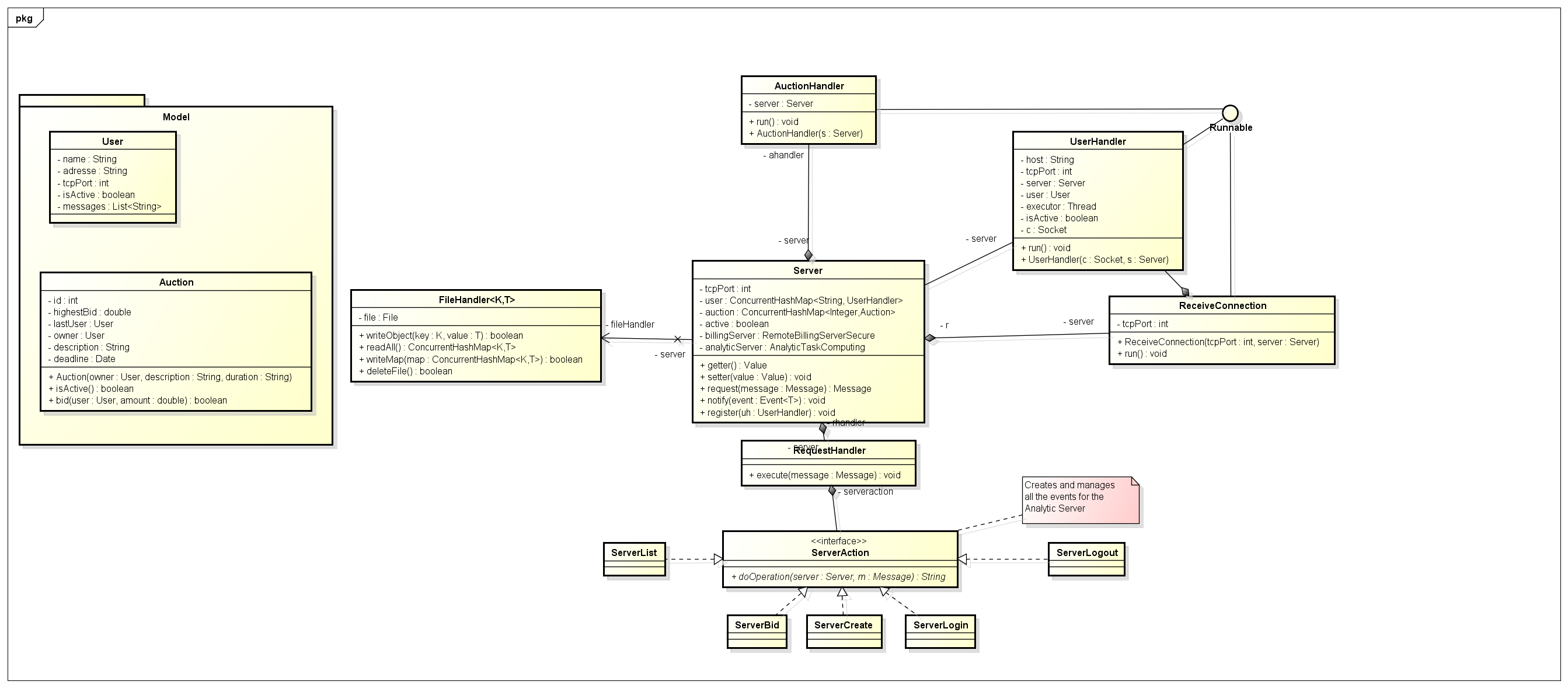


### Server

+ FileHandler



### Model Server



# Zeitschätzungen und Arbeitsaufteilung



# Absprachen

-> Analytic Server

-> Billing Server (Frage GRAFIK?) schritte setzen, abrechnung erstellen -> keine persistenz

-> Testing Load Client (Viele Clients machen bids etc.)

-> Management Client (Befehle für Billing Server, Benachrichtungungen etc)

Altes Programm → List-> ConcurrentHashMap, Eigene Exceptions → UDP Notification brauchen wir nicht mehr

UML-Klassendiagramm → Krepela, Traxler

Aktivitätdiagrann → Lipovits

Use Case Diagramme → Reichmann

Checker → Tattyrek

Tasks:

* RMI-Verbindungen → Traxler
* Analytics Server → Reichmann, Tattyrek
* Billing Server → Krepela
* Management Client → Lipovits
* Testing Component → Lipovits
* Model →Tattyrek (Model JUnitTests)
* Ausbesssern alten Code → Traxler, Reichmann
* Ant,Protokoll → Reichmann

JEDER TESTET SEINEN TEIL + TECHNOLOGIEBESCHREIBUNG FÜRS PROTOKOLL!!!