Die Test Driven Developement Testcases wurden so geschrieben, dass alle wesentlichen Gesichtspunkte, die aus der Aufgabenstellung und aus den bisher geschriebenen Codeteilen hervorgehen, sinnvoll auf ihre jetzige und zukünftige Funktionalität getestet werden. Derzeit existieren Testcases zum AnalyticServer, BillingServe, Model und ManagementClient. Weiters existieren noch Testfälle aus der vorherigen Aufgabe Task07 welche gegebenenfalls noch erweitert werden müssen wenn dies erforderlich ist. In der folgenden Liste sind alle Tests einzeln, pro Package aufgelistet. Angegeben ist pro Test jeweils am Anfang der getestete Methodenname und eventuelle Zusatzschritte die unternommen wurden in Spezialfällen. Am Ende befindet sich eine Liste mit Problemen die während dem Testen festgestellt wurden und ob sie noch ausständig sind oder bereits behoben.

**Wichtig: Vor dem Starten der UnitTests ist es zwingend erforderlich die beiden Server manuell zu starten, da sonst ein Teil der Testfälle fehlschlägt und Coverage verloren geht.**

Beim BillingServer im Package BillingServer werden folgende Methoden getestet:

* Login mit richtigen Usereingaben und Direktaufruf
* Login mit falschen Usereingaben
* createMD5

Beim BillingServerSecure im Package BillingServer werden folgende Methoden getestet:

* getPriceSteps
* createPriceStep mit akzeptierten Eingaben
* createPriceStep zweimal hintereinander und akzeptierten Eingaben
* createPriceStep zweimal hintereinander und überlappenden Eingaben
* createPriceStep zweimal hintereinander und gleichen Eingaben
* createPriceStep mit Startpreis größer Endpreis
* createPriceSep mit startPrice negativ
* createPriceSep mit EndPrice negativ
* createPriceSep mit FixedPrice negativ
* createPriceSep mit VariablePricePercent negativ
* deletePriceStep mit existierenden Eingaben
* deletePriceStep mit falschem Startpreis
* deletePriceStep mit nicht existierenden Eingaben
* billAuction mit akzeptierten Eingaben
* billAuction mit mehreren und gleichen Eingaben
* getBill

**Overall Coverage von Package BillingServer 59.0% (Testzeitpunkt: 25.02.2014 18:06)**

Beim Management Client im Package ManagementClient werden folgende Methoden getestet:

* auto
* hide
* processEvent mit autoprint true
* processEvent mit autoprint false
* Getter- und Setter -Methoden

Zusätzlich werden noch folgende Eingaben am ManagementClient getestet

* Keine Eingabe
* Falsche Eingabe
* !login mit richtigen Benutzereingaben
* !login zweimal hintereinander mit richtigen Benutzereingaben
* !login mit falscher Verwendung
* !login mit falschen Benutzeingaben
* !logout mit Login
* !logout ohne Login
* !steps mit Login
* !addstep mit Login und richtiger Verwendung
* !removestep mit Login und richtiger Verwendung
* !addstep ohne Login
* !addstep mit falschem Format
* !addstep mit falscher Anzahl von Argumenten
* !addstep mit falschen Parametern
* !bill ohne Login
* !bill ohne Usernamen
* !subscribe mit richtiger Verwendung
* !unsubscribe mit richtiger Verwendung
* !unsubscribe ohne ID
* !auto
* !hide
* !print
* !bill mit richtiger Verwendung
* Eingabe mit Leerzeichen am Anfang
* !steps ohne Login
* !subscribe ohne Login
* !subscribe mit falscher Anzahl von Argumenten
* !removestep mit falscherAnzahl an Argumenten
* !removestep ohne Login
* !removestep mit falschem Parameterformat
* !removestep mit nicht existierendem PriceStep

**Overall Coverage von Package ManagementClient: 78.8% (Testzeitpunkt: 25.02.2014 18:12)**

Beim AnalyticsServer werden folgende Methoden getestet:

* processEvent
* notify
* Getter und Setter Methoden

Zusätzlich wurde noch die korrekte Objekterstellung und ein Vergleich zweier Objekte überprüft. Zu beachten ist hier, dass die Methoden des AnalyticServer durch die ManagementClient Testfälle mitgetestet werden.

Beim EventHandler im Package AnalyticsServer werden folgende Methoden getestet:

* Execute mit Event AUCTION\_STARTED
* Execute mit Event AUCTION\_ENDED
* Execute mit Event BID\_PLACED
* Execute mit Event BID\_PLACED zweimal
* Execute mit Event USER\_LOGIN
* Execute mit Event USER\_LOGIN zweimal
* Execute mit Event USER\_LOGOUT
* Execute mit Event USER\_LOGOUT zweimal
* Execute mit Event USER\_DISCONNECTED
* Execute mit Event BID\_OVERBID
* Execute mit Event BID\_OVERBID zweimal
* Execute mit Event BID\_WON
* Execute mit Event BID\_WON zweimal

**Overall Coverage von Package AnalyticsServer 77.7% (Testzeitpunkt: 25.02.2014 18:15)**

Beim Package Model werden alle Methoden erfolgreich getestet. Da dies nur getter- und Setter Methoden sind, wird diesen Testfällen nicht weiter Beachtung geschenkt. Eine Ausnahme im Package Model stellt die Klasse PriceSteps dar.

Bei PriceSteps im Package Model werden folgende Methoden getestet:

* Getter und Setter-Methoden
* ToString mit leerer Liste
* toString mit gefüllter Liste

Es existieren noch weitere Testfälle in dieser Testklasse, allerdings sind diese mit den Testfällen aus dem BillingServerSecure ident.

**Overall Coverage von Package Model 95.7% (Testzeitpunkt: 25.02.2014 18:15)**

**Overall Coverage zum gesamten Projekt 73.6% (Testzeitpunkt: 25.02.2014 18:15)**

**Probleme die anhand der Testfälle festgestellt wurden**

1. Zwei PriceSteps hintereinander abspeichern nicht möglich (behoben von Lins)
2. Inkorrekte Verarbeitung überlappender PriceSteps (behoben von Lins)
3. Ausführung der Methoden billAuctions und getBill, ohne Fehler nicht möglich, Exceptions werden geworfen aufgrund Punkt 2 (behoben von Lins)
4. User.properties fehlt, deshalb keine ausreichende Testung des BillingServers (behoben von Lins)
5. Gleichzeitiger Start von Billing- und AnalyticServer auf einem Gerät nicht möglich, da beide auf den selben Port eine Registry erstellen wollen (behoben von Lins)
6. Erstellen von PriceSteps mit endPrice=0 für unendlichen EndPrice nicht möglich, da die Bedingung dafür fehlt (behoben von Klune)
7. Entsprechnde Stelle für den BillingServer (wegen Punkt 5) auskommentiert, deshalb keine ausreichende Testung des ManagementClient möglich (behoben von Auradnik)
8. Methode für Unsubscribe funktioniert nicht ordnungsgemäß, liegt (laut Auradnik) am AnalyticServer (behoben von Schuschnig)
9. Zum gleichen User kann nur eine einzige Bill gleichzeitig gespeichert werden, sollen aber mehrere möglich sein (behoben von Lins)

**Fazit**

Insgesamt konnte eine Coverage von 73.6% mit 226 Testfällen erreicht werden. Dabei wurde jedes Package getestet.

Die wichtigsten Pakete in Überblick:

Package Model wurde fast vollständig getestet, da fast alle anderen Packages darauf aufbauen. Fehler in der Implementierung des Models können zu Konflikten und zusätzlicher Arbeitszeit führen. Die meisten Tests des Packages Models wurden vor den ersten Implementierungen durchgeführt.

Grundfunktionen und besondere Szenarien von Billing- und Analyticsserver wurden ebenfalls ausführlich vor dem Zusammenführen getestet. Es konnten mehrere Bugs behoben und Zeit in der weiteren Implementierung durch Fehlersuche gespart werden. Durch Designfehler konnten einzelne Bereiche (z.B. große Teile der Main-Methode der Packages Billing- und Analyticsserver nicht komplett getestet werden. Hierbei ist zu beachten, dass dies keine Relevanz auf Fehleranfälligkeit des vorhandenen Codes hat, da alle Funktionen der Server Packages ausführlich getestet werden konnten.

Tests des Package ManagementClient wurden nach einer Teilimplementierung bearbeitet und hinzugefügt. Es kam hier zu kleinen Änderungen in anderen Packages, was dazu führte, dass sich Teile des Management Clients geändert haben. Trotz einer Coverage von 78% konnten alle Grundfunktionen und besondere Szenarien getestet werden.

|  |  |
| --- | --- |
| **Package** | **Gesamte Coverage in Prozent** |
| Model | 95.7% |
| BillingServer | 59.0% |
| ManagementClient | 78.8% |
| AnalyticsServer | 77.7% |
| … |  |
|  | 73.6% |

Weitere getestete Packages sind:

* Client
* Server
* Testing Component

Tests des Package Server und Client, welche bereits vorhanden waren, wurden überarbeitet und teilweise erweitert.

Package Testing Component wurde auf Gültigkeit der Testing Szenarien getestet. Ein höherer Coverage Anteil war schlussendlich nicht nötig. Alle Tests konnten erfolgreich implementiert werden.