

Software-Projekt 2 2014

VAK 03-BA-901.02

Testplan



Patrick Hollatz	phollatz@tzi.de	2596537
Tobias Dellert	tode@tzi.de	2936941
Tim Ellhoff	tellhoff@tzi.de	2520913
Daniel Pupat	dpupat@tzi.de	2703053
Olga Miloevich	halfelv@tzi.de	2586817
Tim Wiechers	tim3@tzi.de	2925222

Inhaltsverzeichnis

1	Einführung	3
1.1	Zweck	3
1.2	Umfang	3
1.3	Beziehungen zu anderen Dokumenten	3
1.4	Aufbau der Testbezeichner	4
1.5	Dokumentation der Testergebnisse	4
1.6	Definitionen und Akronyme	4
1.7	Referenzen	4
2	Systemüberblick	4
2.1	Module der Anwendungsschicht und deren Funktionen	5
3	Merkmale (Daniel)	6
3.0.1	Funktionale Anforderungen	7
3.1	Nicht zu testende Merkmale	7
4	Abnahme- und Testendekriterien	7
5	Vorgehensweise	8
5.1	Komponenten- und Integrationstest	8
5.2	Funktionstest	8
6	Aufhebung und Wiederaufnahme	8
7	Hardware- und Softwareanforderungen	9
7.1	Hardware	9
7.2	Software	9
8	Testfälle	9
8.1	Komponententest	9
8.2	Integrationstest	11
8.3	Funktionstest	11
8.4	Leistungstest	15
8.4.1	Härtetest	15
8.4.2	Volumentest	15
8.4.3	Sicherheitstest	15
8.4.4	Erholungstest	15
9	Testzeitplan	16

Version und Änderungsgeschichte

Version	Datum	Änderungen
1.0	04.07.2014	Dokumentvorlage als initiale Fassung kopiert
1.1	05.07.2014	
1.2	06.07.2014	Testplan abgabefertig

Wichtiger Hinweis: Dieser Testplan wurde zu einem Teil aus verschiedenen Dokumententeilen von dem Testplan unserer Gruppenmitglieder aus dem Wintersemester 2013/14 erstellt (Gruppe *IT_REVOLUTION*). Einige Teile wurden komplett übernommen, andere überarbeitet bzw. angepasst.

Diese Vereinbarung haben wir in der Kick-Off-Veranstaltung für RE SWP 2014 mit dem Veranstalter Dr. Karsten Hölscher getroffen.

1 Einführung

1.1 Zweck

Der Testplan bietet einen Überblick über die geplanten Tests und dient u.a. als Anleitung für die Tester. Die Software soll dabei ausführlich auf Funktionalität getestet werden.

Im Testplan wird festgelegt, wie man welche Komponenten testet. Dazu wird außerdem definiert, welchen Umfang die Tests haben sollen und wann ein Test erfolgreich ist und wann nicht.

Während der Implementierungsphase werden wir uns nach dem Testplan richten und ihn gegebenenfalls weiterführen und vervollständigen.

1.2 Umfang

Der Testplan entspricht der vereinfachten Form des *IEEE Standard for Software Test Documentation 829-1998*.

1.3 Beziehungen zu anderen Dokumenten

Dieser Testplan bezieht sich zum einen auf die Anforderungsspezifikation, da dort die Systemeigenschaften und Systemattribute spezifiziert wurden. Die Testfälle werden auf Grundlage der dortigen Anwendungsfälle entwickelt.

Außerdem gibt es Referenzen zur Architekturbeschreibung, da in dieser die Module und Komponenten definiert wurden, die in diesem Dokument getestet werden sollen.

1.4 Aufbau der Testbezeichner

Der Aufbau der Testbezeichner richtet sich nach folgendem Schema:

- Die ersten beiden Buchstaben geben die Art des Tests vor. Dabei unterscheiden wir zwischen vier verschiedenen Testarten:
 - Komponententests = KT
 - Integrationstests = IT
 - Funktionstests = FT
 - Leistungstests = LT
- Die Nummer steht für die jeweilige Testfallnummer
- Optional: in alphabetischer Reihenfolge werden hier Variationen oder untergeordnete Testfälle definiert

Nach diesem Schema sieht ein Testbezeichner nun folgendermaßen aus:

IT-3-A: Integrationstest, Nr. 3, Variante 1

1.5 Dokumentation der Testergebnisse

Zu jedem Testfall wird ein kurzes Testprotokoll angefertigt. Dieses beinhaltet den Ablauf des Testfalls und die möglichen Komplikationen, die bei der Durchführung entstehen können. Dann werden die Resultate des Testfalls bestimmt und eventuell gefundene Fehler beschrieben.

1.6 Definitionen und Akronyme

1.7 Referenzen

IEEE Standard for Software Test Documentation 829-1998

<http://standards.ieee.org/findstds/standard/829-1998.html>

2 Systemüberblick

Das System besteht aus der Server- und der Clientkomponente. Die konzeptionelle Sicht der Architekturbeschreibung (vgl. Abschnitt 3 der Architekturbeschreibung) dient als Grundlage für den Testplan, da dort die verschiedenen Komponenten beschrieben werden.

Auf der Serverseite gibt es die Komponenten **Communication**, **BusinessLogic** und **Persistence** (vgl. Abbildung 3: Konzeptionelle Sicht Server; Architekturbeschreibung).

Die Clientseite besteht aus den Komponenten **Communication**, **Model** und **User Interface** (vgl. Abbildung 4: Konzeptionelle Sicht Client; Architekturbeschreibung).

Da starke Abhängigkeiten zwischen all diesen Komponenten bestehen, ist es wichtig, dass diese Komponenten fehlerfrei funktionieren.

2.1 Module der Anwendungsschicht und deren Funktionen

In der nachfolgenden Tabelle werden die Module verfeinert die in Punkt 5 visualisiert sind.

GUI	webapp(xhtml)
AndroidApp	AsyncBookTask.java BookAdapter.java MainActivity.java Show-BookActivity.java
Model	it_r3v.bibjsf.presentation
Communication	Network.java
UserInterface	GUI AndroidApp
BusinessLogic	AdministrationHandler.java MediumHandler.java BusinessHandler.java BusinessObjectHandler.java ReaderHandler.java LibrarianHandler.java BorrowHandler.java
BibCommon	Medium.java BusinessObject.java Reader.java Admin.java Librarian.java
Persistence	Data.java Persistence.java

3 Merkmale (Daniel)

Zu testende Merkmale sind in erster Linie Funktionen, welche alle Anwendungsfälle abdeckt, welche die Kundin sich gewünscht hat. Dabei muss sowohl die App als auch die Website getestet werden.

App

1. Online

- 1.1 Benutzer registrieren
- 1.2 Benutzer anmelden
- 1.3 Benutzer abmelden
- 1.4 Gegner herausfordern
- 1.5 Angefangenes Spiel weiterspielen
- 1.6 Frage beantworten
- 1.7 Einstellungen ändern

2. Offline

- 2.1 Neues Spiel starten
- 2.2 Frage beantworten

3. Website

- 3.1 Admin anmelden
- 3.2 Admin abmelden
- 3.3 Frage hinzufügen
- 3.4 Frage bearbeiten
- 3.5 Frage löschen
- 3.6 Frageliste importieren
- 3.7 Frageliste exportieren
- 3.8 User löschen
- 3.9 Passwort ändern

3.0.1 Funktionale Anforderungen

Besonders wichtig ist der Spielvorgang bei der App, da die Nutzer diesen hauptsächlich nutzen werden. Dabei muss darauf geachtet werden, dass der Nutzer sowohl online als auch offline spielen kann. Auch das spielen gegen eine andere Person muss getestet werden.

3.1 Nicht zu testende Merkmale

Da wir ein komplett neues Projekt erstellen, ist es wichtig das alle Merkmale getestet werden. Sollte es triviale Funktionen geben, müssen diese nicht getestet werden.

4 Abnahme- und Testendekriterien

Fehler werden in eine Kategorie eingeordnet und erhalten entsprechende Fehlerwerte. Aus diesen Fehlerwerten ergeben sich Prioritäten, welche die Reihenfolge der Fehlerbehandlung angibt. Das Testen wird beendet, wenn der berechnete Fehlerwert aller Fehler pro 1000 Zeilen Code unter dem Wert 10 liegt und die Software nicht beeinträchtigt wird, d.h. es keinen Fehler der Fehlerklasse **Mittel** oder höher gibt.

Testabdeckung Die Testabdeckung soll so hoch wie möglich sein. Für ein stabiles System spricht, dass die Testabdeckung in systemkritischen Bereichen soweit vollständig ist. Jeder Fehler in diesem Bereich kann das System zum Absturz bringen und muss somit verhindert werden. In anderen Bereichen, die das laufende System bei einem Fehler weniger beeinträchtigen, wird die Testabdeckung nicht so vollständig sein, wie in kritischen Bereichen.

Fehlerbewertung:

Die nachfolgende Tabelle spezifiziert die Auswirkung eines Fehlers, durch die man diese nach Priorität einordnen kann.

Fehlerkl. ¹	Beschreibung	Wert
Leicht	Unwesentliche Fehler, die den Programmablauf nicht beeinträchtigen, aber trotzdem behandelt werden sollten.	1
Mittel	Fehler in dieser Art haben Auswirkungen auf den Programmablauf. Dieser beeinträchtigt aber nicht die grundlegenden Funktionen.	10
Schwer	Fehler der Klasse „Schwer“ beeinträchtigen die Funktionsfähigkeit des Systems sehr stark und müssen sofort behandelt werden.	20
Fatal	Diese Fehler machen den Programmablauf unmöglich und können zum Absturz des Systems führen.	100

5 Vorgehensweise

5.1 Komponenten- und Integrationstest

Hier findet sich das konkrete Vorgehen bei der Integration: Welche Klassen werden zunächst zusammen getestet, welche kommen dann hinzu? Das kann man z.B. geeignet in Form eines Baumes aufzeigen.

5.2 Funktionstest

Die Funktionstest sind durch jene Anwendungsfälle aus der Anforderungsspezifikation vorgegeben. Jede dieser Funktionen muss durch Tests gedeckt sein.

6 Aufhebung und Wiederaufnahme

Wir werden Tests unterbrechen, wenn ein gewisser Wert überschritten wird, welcher über die Tabelle in Abschnitt 4 berechnet wird. In diesem Fall werden wir sofort wieder mit der Implementierung anfangen. Da wir mit der Bottom-up Strategie testen, werden wir bei Fehlern in der unteren Schicht einen niedrigeren Wert nehmen.

Bei Fehlern der Data setzen wir einen Wert von 10, bei Fehlern in der Logik einen Wert von 20 und bei Fehlern, welche die GUI betreffen, einen von 40 und bei den restlichen Faktoren einen von 100.

Sollten die Fehler behoben sein, testen wir noch einmal alle Komponenten, die mit den veränderten interagieren.

¹=Fehlerklasse

7 Hardware- und Softwareanforderungen

Hier soll das zu prüfende Material spezifiziert werden. Auf welchen Rechnern werden die Tests durchgeführt? Welche anderen Hilfsmittel oder Ressourcen sind dafür nötig?

7.1 Hardware

Als Hardware stehen uns unsere Notebooks und Smartphones, sowie die Unirechner zur Verfügung. Dabei haben wir alle geforderten Betriebssysteme mindestens einmal auf unseren Notebooks installiert, sodass wir auf jeden Gerät testen können. Da Android Unterstützung gefordert ist, werden wir die App über unseren vorhandenen Smartphones, die Android haben, testen. Andere Systeme (iOS) werden wir testen, falls noch genügend Zeit vorhanden ist.

7.2 Software

Als Software benutzen wir in der Eclipse Umgebung JUnit-Tests. Diese werden in Form von BlackBox- und WhiteBox-Tests implementiert. Die App werden wir mithilfe eines Android Emulators und unseren Smartphones testen.

8 Testfälle

*Dies ist der wichtigste (und vermutlich umfangreichste) Teil des Testplans. Hier wird genau aufgelistet, **was wie und von wem** getestet wird. Das spätere Testen besteht dann einfach aus einer Durchführung dieser Tests.*

*Zu jedem Testfall muss es eine **Testfallspezifikation** geben (außer Komponententests, siehe unten). Schwerpunkt sollen hier Integrations- und Leistungstests sein. Welche Arten von Integrationstests und Leistungstests seht Ihr vor und wie wollt Ihr diese genau ausgestalten?*

8.1 Komponententest

Auf die genaue Spezifikation der Komponententests könnt Ihr verzichten – diese werden durch die JUnit-Testfälle gegeben. Uns genügt hierzu eine Beschreibung, welche Komponenten/Klassen wie, wann und durch wen getestet werden sollen. Die zugehörigen JUnit-Tests werden separat abgegeben (Black-Box jetzt, White-Box später bei der Abgabe der Implementierung) und müssen im Testplan nicht als Code aufgeführt werden.

Klasse	Implementierer	Tester	Testart
ReferenceManagement	Klaus	Hans	Blackbox
UserManagement	Anna	Klaus	Blackbox
...
ReferenceTable	Heinz	Bert	Whitebox

Tabelle 1: Komponententests

Achtet darauf, dass der Tester ein anderer ist als der Implementierer. Jeder soll mindestens eine Klasse als Black-Box und eine Klasse als White-Box testen!

Hier könnten später noch die Kontrollflussdiagramme für die Whiteboxtests eingefügt werden.

8.2 Integrationstest

Die Integrationstests werden hier genauer beschrieben: Welche Klassen sind beteiligt? Wie ist der Zustand des Systems vor Beginn? Welche Eingaben werden getätigt? Welche Ergebnisse/welches Verhalten wird erwartet?

Aufbau einer Testfallspezifikation:

- 1. eindeutiger Testfallbezeichner (entspricht Namenskonvention aus Abschnitt 1.4)*
- 2. Testobjekte: welche Komponenten werden getestet?*
- 3. Eingabespezifikationen (Eingaben des Testfalls)*
- 4. Ausgabespezifikationen (erwartete Ausgaben)*
- 5. Umgebungserfordernisse (notwendige Software- und Hardwareplattform sowie Testtreiber und -rumpfe)*
- 6. besondere prozedurale Anforderungen (Einschränkungen wie Zeitvorgaben, Belastung oder Eingreifen durch den Operator)*
- 7. Abhängigkeiten zwischen Testfällen*

8.3 Funktionstest

Bezeichner:	FT-1
Anwendungsfall	App starten
Eingabe	Start-Button wird gedrückt
Ausgabe	App ist gestartet Bei Fehlerfall: Rückmeldung Probleme beim starten der App
Umsetzung	Start-Button wird gedrückt, Benutzer hat die App bereits heruntergeladen

Bezeichner:	FT-2
Anwendungsfall	Offline-Modus starten
Eingabe	Offline Modus starten wird gedrückt
Ausgabe	Der Modus wurde gestartet, spielen nun möglich Bei Fehlerfall: Rückmeldung starten nicht erfolgreich
Umsetzung	Auf Offline Modus starten drücken

Bezeichner:	FT-3
Anwendungsfall	Benutzer registrieren
Eingabe	Informationen des Nutzers(Nickname, Passwort)
Ausgabe	Rückmeldung über erfolgreiches Anmelden. Nutzer ist nun in Datenbank gespeichert Bei Fehlerfall: Fehlerrückmeldung
Umsetzung	Manuelles eintragen der Daten, dann auf registrieren klicken

Bezeichner:	FT-4
Anwendungsfall	Benutzer anmelden
Eingabe	Anmeldedaten(Nutzername, Passwort)
Ausgabe	Rückmeldung über erfolgreiche Anmeldung Bei Fehlerfall: Rückmeldung Anmeldung nicht erfolgreich
Umsetzung	Manuelle Eingabe der Nutzerdaten, Benutzer ist registriert

Bezeichner:	FT-5
Anwendungsfall	Benutzer abmelden
Eingabe	Logout-Button wird gedrückt
Ausgabe	Rückmeldung über erfolgreiche Abmeldung Bei Fehlerfall: Rückmeldung Abmeldung nicht erfolgreich
Umsetzung	Manuelles ausloggen, Benutzer ist angemeldet

Bezeichner:	FT-6
Anwendungsfall	Spielen(Offline)
Eingabe	Spielen-Button drücken
Ausgabe	Spiel wird gestartet Bei Fehlerfall: Rückmeldung Spielen nicht möglich
Umsetzung	Manuell, Benutzer spielt Offline

Bezeichner:	FT-7
Anwendungsfall	Spielen(Online)
Eingabe	Spielen-Button wird gedrückt
Ausgabe	Auswahl zwischen Neues Spiel starten und offenes Spiel wird angezeigt Bei Fehlerfall: Rückmeldung Spielen nicht möglich
Umsetzung	Manuell, Benutzer ist angemeldet

Bezeichner:	FT-8
Anwendungsfall	Neues Spiel
Eingabe	Neues Spiel wird gedrückt
Ausgabe	Liste aller Spieler die Online sind erscheint Bei Fehlerfall: Rückmeldung Neues Spiel nicht erfolgreich
Umsetzung	Manuell, Benutzer war auf Spielen

Bezeichner:	FT-9
Anwendungsfall	Gegner herausfordern
Eingabe	Gegner der herausgefordert werden soll
Ausgabe	Spiel wird gestartet Bei Fehlerfall: Rückmeldung herausfordern nicht erfolgreich
Umsetzung	Manuell, Gegner ist Online

Bezeichner:	FT-10
Anwendungsfall	Offenes Spiel spielen
Eingabe	Antwort auf die Frage
Ausgabe	Nachricht ob richtige oder falsche Antwort Bei Fehlerfall: Rückmeldung Fehler
Umsetzung	Manuell, Spiel wurde bereits angefangen

Bezeichner:	FT-11
Anwendungsfall	Einstellungen ändern
Eingabe	Benutzername, Passwort
Ausgabe	Rückmeldung über erfolgreiches Ändern Bei Fehlerfall: Rückmeldung Ändern war nicht erfolgreich
Umsetzung	Manuell, Benutzer ist angemeldet

Bezeichner:	FT-12
Anwendungsfall	Website wird aufgerufen
Eingabe	URL
Ausgabe	Website wird angezeigt Bei Fehlerfall: Rückmeldung anzeigen war nicht erfolgreich
Umsetzung	Manuell, Internet Verbindung vorhanden

Bezeichner:	FT-13
Anwendungsfall	Admin anmelden
Eingabe	Nutzername, Passwort
Ausgabe	Rückmeldung, anmelden erfolgreich Bei Fehlerfall: Rückmeldung anmelden war nicht erfolgreich
Umsetzung	Manuell

Bezeichner:	FT-14
Anwendungsfall	Admin abmelden
Eingabe	Logout-Button drücken
Ausgabe	Rückmeldung abmelden erfolgreich Bei Fehlerfall: Rückmeldung abmelden nicht erfolgreich
Umsetzung	Manuell, Admin ist angemeldet

Bezeichner:	FT-15
Anwendungsfall	Frage hinzufügen
Eingabe	Frageinformationen(Frage, Antworten)
Ausgabe	Rückmeldung über erfolgreiches Hinzufügen Bei Fehlerfall: Rückmeldung Hinzufügen war nicht erfolgreich
Umsetzung	Manuell, Benutzer ist Admin

Bezeichner:	FT-16
Anwendungsfall	Frage bearbeiten
Eingabe	Frageinformationen(Frage, Antworten)
Ausgabe	Rückmeldung über erfolgreiches Ändern Bei Fehlerfall: Rückmeldung Ändern nicht erfolgreich
Umsetzung	Manuell, Benutzer ist Bibliothekar
Bezeichner:	FT-17
Anwendungsfall	Frage löschen
Eingabe	zu löschende Frage
Ausgabe	Rückmeldung über erfolgreiches Löschen Bei Fehlerfall: Rückmeldung Löschen war nicht erfolgreich
Umsetzung	Manuell, Benutzer ist Admin
Bezeichner:	FT-18
Anwendungsfall	Frageliste importieren
Eingabe	Frageliste
Ausgabe	Rückmeldung über erfolgreiches importieren Bei Fehlerfall: Rückmeldung Importieren war nicht erfolgreich
Umsetzung	Manuell, Benutzer ist Admin
Bezeichner:	FT-19
Anwendungsfall	Frageliste exportieren
Eingabe	Frageliste
Ausgabe	Rückmeldung über erfolgreiches exportieren Bei Fehlerfall: Rückmeldung Exportieren nicht erfolgreich
Umsetzung	Manuell, Benutzer ist Admin
Bezeichner:	FT-20
Anwendungsfall	User löschen
Eingabe	User
Ausgabe	Rückmeldung über erfolgreiches Löschen Bei Fehlerfall: Rückmeldung Löschen war nicht erfolgreich
Umsetzung	Manuell, User existiert
Bezeichner:	FT-21
Anwendungsfall	Passwort ändern
Eingabe	Passwort
Ausgabe	Rückmeldung über erfolgreiches Ändern Bei Fehlerfall: Rückmeldung Ändern war nicht erfolgreich
Umsetzung	Manuell, Admin ist angemeldet

8.4 Leistungstest

8.4.1 Härtetest

Bezeichner:	LT-1
Beschreibung	100 Nutzer starten gleichzeitig ein Spiel
Ziel:	Robustheit der Datenbank mit vielen Anfragen umzugehen wird getestet
Bei Erfolg:	Datenbank kann Anfragen bearbeiten; es gibt keine langen Wartezeiten
Fehler:	TimeOut, Absturz

8.4.2 Volumentest

Bezeichner:	LT-2
Beschreibung	Eine sehr große CSV-Datei wird importiert
Ziel:	Robustheit der Datenbank mit großen Datenmengen umzugehen wird getestet
Bei Erfolg:	Datenbank kann mit der Verarbeitung umgehen; es gibt keine Fehler
Fehler:	TimeOut, Absturz

8.4.3 Sicherheitstest

Bezeichner:	LT-3
Beschreibung	Ein Nutzer gibt ein falsches Passwort ein(App und Website)
Ziel:	Korrekte Authentifizierung wird getestet
Bei Erfolg:	Nutzer kann sich nicht einloggen; System gibt Fehlermeldung
Fehler:	Benutzer kann sich anmelden

8.4.4 Erholungstest

Bezeichner:	LT-4
Beschreibung	Ein Nutzer gibt mehrmals ein falsches Passwort ein
Ziel:	Erholt sich das System; Kann man sich danach problemlos mit dem richtigen Passwort einloggen
Bei Erfolg:	Bei richtiger Eingabe der Logindaten, ist man eingeloggt
Fehler:	Man kann sich nicht mehr einloggen, Absturz

Bezeichner:	LT-5
Beschreibung	Der Server geht offline und startet die Verbindung neu
Ziel:	Ist das System nach neuer Verbindung wieder funktionstüchtig
Bei Erfolg:	System produziert keine Fehler
Fehler:	Funktionen werden nicht mehr unterstützt, Datenverlust, TimeOut, Absturz

9 Testzeitplan