

INTERVIEWER

Testdokumentation

Interviewer

Qualitätsprüfungen durch JUnit und manuelle
Oberflächentests

Interviewer

16.01.2014

Software Projekt im WS 13/14 an der Leibniz Universität Hannover

Dieses Dokument enthält Erläuterungen zu den durchgeführten Tests im Softwareprojekt Interviewer. Die Abnahmetests wurden erfolgreich durchgeführt auf Revision 729. Die JUnit-Tests wurden durchgeführt an der Revision 764. Die spezielle Testmethodik für die Datenbank wird erläutert. Außerdem dienen die JUnit-Testfälle selbst der Dokumentation

Inhalt

| | |
|--|----|
| Einleitung..... | 3 |
| Protokoll der Abnahmetests | 4 |
| Qualitätssicherung in der Serveranwendung..... | 8 |
| JUnit-Tests | 9 |
| Qualitätssicherung in der Clientanwendung..... | 10 |
| Praktische Erprobung | 10 |

Der Quality Agent bestätigt, dass alle erfolgreich Testfälle und wie hier angegeben abgelaufen sind.

Thiemo Fischer, QA Interviewer

Einleitung

Dieses Dokument enthält Erläuterungen zu den durchgeführten Tests im Softwareprojekt Interviewer. Die Abnahmetests wurden erfolgreich durchgeführt auf Revision 729. Die JUnit-Tests wurden erfolgreich durchgeführt auf Revision 764. Die spezielle Testmethodik für die Datenbank wird erläutert. Außerdem dienen die JUnit-Testfälle selbst der Dokumentation. Die Abnahmetestfälle sind, wie in der Anforderungsspezifikation dokumentiert, hier übernommen. Kleine Abweichungen sind dokumentiert.

Protokoll der Abnahmetests

Die Abnahmetests wurden durchgeführt am 15.01.14 auf Revision 729.

Um die Voraussetzungen zu erfüllen wurde ein Benutzer Admin eingerichtet, der die beiden Benutzer erstellt.

Für die Abnahmetests wurden Firefox und Chrome verwendet.

| Abnahmetest: Anmelden und Projekte verwalten (erfolgreich) | |
|--|---|
| Testfall-ID | 01 |
| Voraussetzungen | <ul style="list-style-type: none">• Benutzer Franz und Julia wurden erstellt.• Benutzer Franz hat noch keine Projekte erstellt. |
| Beschreibung | <ol style="list-style-type: none">1. Benutzer Franz meldet sich mit seiner Email-Adresse „franz@email.de“ und dem Passwort „super-franz“ an.2. System zeigt eine Projektübersicht mit keinen Projekten.3. Benutzer Franz erstellt ein Projekt mit dem Titel „JUnit-Umfrage“.4. Benutzer Franz speichert das Projekt.5. System zeigt die Projektübersicht mit einem Projekt „JUnit-Umfrage“.6. Franz bearbeitet Projekt und verändert den Namen zu „Eclipse-Umfrage“.7. Franz fügt zum Projekt den Benutzer Julia hinzu.8. Franz fügt eine Intervieweigenschaft „Programmiersprachen“ hinzu.9. Der Benutzer Franz speichert das Projekt.10. System zeigt in der Projektübersicht ein Projekt mit dem Namen „Eclipse-Umfrage“.11. Der Benutzer Julia meldet sich an.12. System zeigt Projekt mit dem Namen „Eclipse-Umfrage“.13. Benutzer Julia entfernt das Projekt „Eclipse-Umfrage“.14. System zeigt Projektübersicht mit keinem Projekt. |

| Abnahmetest: Interviews verwalten und Datei hochladen (erfolgreich) | |
|---|---|
| Testfall-ID | 02 |
| Voraussetzungen | <ul style="list-style-type: none">• Benutzer Franz wurde mit einem Projekt „JUnit-Umfrage“ mit der Intervieweigenschaft „Programmiersprachen“ erstellt.• Benutzer Franz ist angemeldet.• Ergänzung: Eine Datei abnahmetest.mp3 steht zur Verfügung. |
| Beschreibung | <ol style="list-style-type: none">1. System zeigt Projektübersicht.2. Benutzer Franz wählt Projekt „JUnit-Umfrage“ aus.3. System wechselt zur Interviewansicht und zeigt eine Übersicht mit keinen Interviews.4. Benutzer Franz erstellt ein neues Interview mit ID „Interview01“. |

| | |
|--|--|
| | <ol style="list-style-type: none"> 5. Benutzer Franz speichert das Interview. 6. System zeigt die Interviewübersicht mit dem Interview „Interview01“. 7. Benutzer Franz klickt auf den Bearbeiten-Button von Interview „Interview01“. 8. System zeigt Interview-Bearbeiten-Dialog an. 9. Franz ändert die Intervieweigenschaft Programmiersprachen zu „Java“. 10. Benutzer Franz lädt die Audiodatei abnahmetest.mp3 hoch. 11. System zeigt im Dialog die Datei abnahmetest.mp3 an. 12. Franz speichert das Interview. 13. Franz lädt die Seite neu. 14. System zeigt im Bearbeiten-Dialog von Interview01 weiterhin die Intervieweigenschaft „Java“ und die Audiodatei „abnahmetest.mp3“ an. (Änderung: Upload prinzipiell nicht mehr möglich) 15. Benutzer klickt auf Upload. (Änderung: Upload prinzipiell nicht mehr möglich) 16. System gibt eine Fehlermeldung aus. (Änderung: Upload prinzipiell nicht mehr möglich) 17. Franz löscht das Interview „Interview01“. 18. System zeigt nicht mehr das Interview „Interview01“ an. |
|--|--|

| Abnahmetest: Aussagen verwalten und Audio-Datei wiedergeben (erfolgreich) | |
|---|---|
| Testfall-ID | 03 |
| Voraussetzungen | <ul style="list-style-type: none"> • Benutzer Franz wurde erstellt. • Projekt „JUnit-Umfrage“ wurde erstellt. • Im Projekt „JUnit-Umfrage“ wurde ein Interview „Interview01“ erstellt. • Zu dem Interview „Interview01“ wurde die Audiodatei „abnahmetest.mp3“ (3 min lang) hochgeladen. • Benutzer Franz ist angemeldet und befindet sich in der Interviewübersicht zu Projekt „JUnit-Umfrage“. |
| Beschreibung | <ol style="list-style-type: none"> 1. Benutzer Franz wählt das Interview „Interview01“ aus. 2. System wechselt zum Aussageneditor und zeigt keine Aussagen an. 3. Benutzer Franz startet den Player. 4. System fängt an die Audiodatei abzuspielen. 5. Benutzer klickt nach 15 Sekunden auf „Neue Aussage erstellen“. 6. System pausiert den Player. 7. System zeigt eine leere Aussagenzeile an mit der Zeitstelle „00:11“. 8. Benutzer Franz verändert die Zeitstelle um 5 Sekunden nach oben. 9. System zeigt in der Zeitstelle „00:16“ an. 10. Benutzer Franz beschreibt die Aussage mit „Erfolgreicher Abnahmetest“. 11. Franz schreibt ins Codes-Feld „Testfall“. 12. Benutzer Franz speichert die Aussage ab. 13. System zeigt die Aussage „Erfolgreicher Abnahmetest“ mit dem Code „Testfall“ und der Zeitstelle „00:16“ an. 14. Benutzer Franz klickt auf die Aussage „Erfolgreicher Abnahmetest“. 15. System lässt den Player auf die Zeitstelle „00:16“ springen, ohne die Audiodatei abzuspielen. 16. Benutzer Franz startet den Player. 17. System spielt die Audiodatei „abnahmetest.mp3“ bei der Stelle „00:16“ |

| | |
|--|---|
| | ab. 18. Benutzer verändert die Aussage „Erfolgreicher Abnahmetest“ zu „Zufriedener Kunde“ und speichert die Änderung. 19. System zeigt nun die Aussage mit der Veränderung „Zufriedener Kunde“ an. 20. Franz löscht die Aussage „Zufriedener Kunde“. 21. System zeigt nun keine Aussagen mehr an. |
|--|---|

| Abnahmetest: Detaillierte Bedienung des Players (erfolgreich) | |
|--|---|
| Testfall-ID | 04 |
| Voraussetzungen | <ul style="list-style-type: none"> • Benutzer Franz wurde erstellt. • Projekt „JUnit-Umfrage“ wurde erstellt. • Im Projekt „JUnit-Umfrage“ wurde ein Interview „Interview01“ erstellt. • Zu dem Interview „Interview01“ wurde die Audiodatei „abnahmetest.mp3“ (3 min lang) hochgeladen. • Benutzer Franz ist angemeldet und befindet sich in dem Aussageneditor zum Interview „Interview01“. |
| Beschreibung | <ol style="list-style-type: none"> 1. Benutzer Franz klickt auf Play. 2. System spielt die Audiodatei abnahmetest.mp3 ab der Stelle 00:00 ab. 3. Benutzer Franz pausiert die Wiedergabe nach 12 Sekunden. 4. System pausiert die Wiedergabe bei der Stelle 00:12. 5. Benutzer Franz wählt auf der Zeitleiste die Zeitstelle 0:56 aus. 6. System springt zur Stelle 0:56 7. Benutzer wählt Abspielen 8. System spielt die Audiodatei abnahmetest.mp3 bei der Stelle 0:56 ab. 9. Benutzer Franz zoomt einmal in die Audiodatei hinein. 10. System zeigt die Zeitleiste von ca. 0:50 bis 1:10 an. (Änderung: Zoomlevel ändert sich mit Faktor 2: Angezeigter Abschnitt halbiert oder verdoppelt sich) 11. Benutzer Franz zoomt einmal aus der Audiodatei heraus. 12. System zeigt die Zeitleiste von 0:00 bis 3:00 an. (Anmerkung: Zeigt die gesamte Audiodatei) |

| Abnahmetest: Exportieren (erfolgreich) | |
|---|--|
| Testfall-ID | 05 |
| Voraussetzungen | <ul style="list-style-type: none"> • Benutzer Franz wurde erstellt. • Projekt „JUnit-Umfrage“ wurde erstellt. • Im Projekt „JUnit-Umfrage“ wurde ein Interview „Interview01“ erstellt. • Zu dem Interview „Interview01“ wurde die Audiodatei „abnahmetest.mp3“ (3 min lang) hochgeladen. • Zu dem Interview „Interview01“ wurde eine Aussage „Erfolgreicher |

| | |
|--------------|---|
| | Abnahmetest“ mit Zeitpunkt 0:16 mit dem Code „Programmiersprachen“ erstellt. <ul style="list-style-type: none"> • Benutzer Franz ist angemeldet und befindet sich in der Projektübersicht. |
| Beschreibung | <ol style="list-style-type: none"> 1. Benutzer Franz wählt das Projekt „JUnit-Umfrage“ zum Exportieren aus. 2. System lädt die Datei JUnit-Umfrage.xml bzw. JUnit-Umfrage.json herunter und es sind alle Informationen zum Projekt „JUnit-Umfrage“ enthalten. |

| Abnahmetest: Codes-Übersicht verwenden | |
|---|--|
| Testfall-ID | 06 (erfolgreich) |
| Voraussetzungen | <ul style="list-style-type: none"> • Benutzer Franz wurde erstellt. • Projekt „JUnit-Umfrage“ wurde erstellt. • Im Projekt „JUnit-Umfrage“ wurde ein Interview „Interview01“ erstellt. • Zu dem Interview „Interview01“ wurde eine Aussage „Erfolgreicher Abnahmetest“ mit Zeitpunkt 0:16 mit dem Code „Programmiersprachen“ erstellt. • Benutzer Franz ist angemeldet und befindet sich in der Projektübersicht. |
| Beschreibung | <ol style="list-style-type: none"> 1. Benutzer Franz wählt die Codeübersicht zu Projekt „JUnit-Umfrage“ aus. 2. System zeigt die Codeübersicht mit der Codegruppe „Programmiersprachen“ an. 3. Benutzer Franz klappt die Codegruppe „Programmiersprachen“ auf. 4. System zeigt die Aussage „Erfolgreicher Abnahmetest“ an. 5. Benutzer Franz minimiert die Codegruppe „Programmiersprachen“. 6. System blendet die Aussage „Erfolgreicher Abnahmetest“ wieder aus und zeigt nun die Codeübersicht mit der Codegruppe „Programmiersprachen“ an. |

Qualitätssicherung in der Serveranwendung

Getestet wurden alle Klassen des models und utils Packages mit Hilfe des BlackBox-Verfahrens oder durch TestFirst. An manchen Stellen wurde auf den Test von Gettern und Settern verzichtet. Ansonsten decken die Tests alle Funktionen dieser Klassen ab. Zusätzlich wurde der von jedem Entwickler geschriebene Code von mindestens einem anderen Teammitglied in einem Walkthrough überprüft.

Das Testen der EBean-Datenbankanbindung hat sich oft als schwierig erwiesen, sodass die Funktionalität der Datenbank in manchen Fällen in einem einzelnen Test durchgeführt wurde. In einem neuen Ansatz enthalten die Testfälle vor den assert-Anweisungen eine refresh()-Anweisung um sicherzustellen, dass immer die Daten geprüft werden, die in der Datenbank vorliegen. An anderen Stellen fehlen die refresh()-Anweisungen, da geprüft wird ob die Geschäftslogik/EBeans den Refresh selber durchführt (EBean-Managed-Relationships). Mit anderen Worten der Refresh wird nicht durchgeführt, wenn eine Änderung innerhalb einer anderen Klasse automatisch zu dem getesteten Objekt propagieren soll. Die alten Testfälle werden zu diesem Zeitpunkt noch auf die neue Vorgehensweise angepasst. Allerdings ohne den Anspruch auf Vollständigkeit. Der Test der Geschäftslogik wurde davon unbeeinflusst nach den jeweils unten angegebenen Verfahren durchgeführt.

Um den Initialisierungsaufwand zu verringern steht eine TestData-Klasse zur Verfügung, die die Datenbank mit sinnvollen Werten füllt und das Testen des Codes auf Package-Ebene ermöglicht. Eine Testsuite fasst die Tests des jeweiligen Packages zusammen. Die Superklasse BaseModelTest stellt die Testumgebung bereit.

Bei den Lowlevel-Klassen PropertyType und DynamicProperty wurde auf den gerade beschriebenen Datenbank-Test verzichtet, da ihre Datenbanktauglichkeit vollständig in ProjectTest und InterviewTest nachgewiesen wird.

Um die Tests in der bei der Entwicklung verwendeten Eclipse IDE durchzuführen, muss als Parameter in der Run Configuration die Datei

```
-javaagent:"<path_to>/ebean-2.8.1-agent.jar"
```

angegeben werden.

JUnit-Tests

- models.AudioFile (Blackbox, Datenbank-Test)
- models.CodeTest (Blackbox, Datenbank-Test)
- models.DynamicPropertyTest (Blackbox, TestFirst)
- models.InterviewTest (Blackbox, Datenbank-Test)
- models.ProjectTest (Blackbox, Whitebox, z.T. TestFirst, Datenbank-Test)
- models.PropertyTypeTest (Blackbox)
- models.StatementTest(Blackbox, Datenbank-Test)
- models.TimeTest(Blackbox, z.T. Datenbank-Test)
- models.UserTest(Blackbox, Datenbank-Test)
- utils.ExportTest(Blackbox)
- utils.SqlUtilsTest(z.T. Blackbox)
- testenvironment.TestData(Whitebox, Datenbank-Test)

Qualitätssicherung in der Clientanwendung

Das Team hat sich dagegen entschieden, automatische Oberflächentests zu verwenden, da die korrekte Funktion der einzelnen GUI-Elemente während der Entwicklung oft erprobt wurde. Die Abnahmetestfälle weisen die korrekte Funktion der meistgenutzten GUI-Elemente nach. Falls es dennoch zu einer Fehlfunktion der GUI kommen sollte, ist dieser Fehler, dank der umfangreichen Dokumentation, leicht zu beheben. Die Controller führen eine Plausibilitätsprüfung der übermittelten Daten durch, sodass ungewollte Datenmanipulationen unwahrscheinlich sind. Der erforderliche Aufwand rechtfertigt also nicht das systematische Testen der GUI.

Alle Skripte wurden nicht nur vom Entwickler, sondern auch von mindestens einem anderen Teammitglied kontrolliert.

Praktische Erprobung

Die Software wurde während der gesamten Entwicklungsphase durch die Entwickler erprobt. Hierzu stehen dieselben Testdaten zur Verfügung, die auch in den Testfällen verwendet werden. Das Vorladen der Datenbank wird bei Auslieferung der Software abgeschaltet.

Alle Funktionen der Software wurden erfolgreich über LAN und WAN verwendet.