

Software-Qualitätssicherung

Übung 2 – Abgabe

Bitte lesen Sie die Angabe sorgfältig durch und achten Sie dabei auch auf den Disclaimer.

Generell gilt, dass die Anzahl der zur Verfügung gestellten Zeilen nur aus Formatierungsgründen besteht. Es können auch Zeilen hinzugefügt oder leer gelassen werden.

Ihre Daten – Bitte füllen Sie alle Felder aus:

Nachname, Vorname:	Zahirovic, Belmin
Matrikelnummer:	01428177
Studienkennzahl:	033 534
E-Mail Adresse:	e1428177@student.tuwien.ac.at

Inhalt

1	Testplanerstellung.....	2
1.1	Anforderungseinteilung.....	2
1.2	Testkonzept und Testablauf	3
2	Äquivalenzklassen	4
2.1	Analyse und Erstellung	4
2.1.1	Anforderung: Rabatte	4
2.1.2	Anforderung: Videopoints.....	5
2.1.3	Anforderung: Altersfreigabe	6
2.2	Testfallableitung	7
3	State Charts	8
3.1	Erstellung	8
3.2	Testfallableitung	9

1 Testplanerstellung

1.1 Anforderungseinteilung

Nr.	FA / NFA	Nr.	FA / NFA	Nr.	FA / NFA	Nr.	FA / NFA	Nr.	FA / NFA	Nr.	FA / NFA
1	FA	5	FA	9	FA	13	NFA	17	NFA	21	NFA
2	FA	6	NFA	10	FA	14	FA	18	NFA	22	NFA
3	FA	7	FA	11	FA	15	FA	19	NFA		
4	FA	8	FA	12	FA	16	NFA	20	FA		

Anforderung	Testbar (mit Begründung)	Wie kann getestet werden
13	Hier sind keine Tests nötig, wenn das Design schon gut geformt ist. Die Anforderung ist daher nicht genau testbar.	Keine direkte Tests möglich. "Gutes" und funktionales UI gewährleistet, dass die Anforderung korrekt und erfüllt ist.
18	Da diese Anforderung sehr wichtig für das gesamte System ist, soll es natürlich testbar sein..Dies ist eines der wichtigsten Merkmale des Systems.	Es soll manuell überprüft werden (Systemadministratoren können das machen), ob alle Dateien (jeden Tag) um 23:55 gespeichert sind und ob man alle gespeicherte Dateien wiederherstellen kann.
19	Nein, es ist kaum Möglich, diese Anforderung(explizit) zu testen. Es besteht jedoch die Möglichkeit, die Anforderung durch Verwendung/Nutzung des Systems zu testen.	Es besteht die Möglichkeit, das System durch ständige Nutzung zu testen. (=zu überprüfen, ob Fehler auftreten oder nicht)
21	Nicht testbar. (Soll gut aktualisiert sein)	Damit es testbar wäre, müsste zuerst eine Erweiterung vorhanden (implementiert) sein.
22	Ja, testbar. Es sollte schon während der Implementation getestet werden. (Laufzeit)	Es könnte automatisch (mit Zufallsdaten und mit zB. einer Datenbank, und führen den Code auf DB. aus) getestet werden. Es wird dabei auch natürlich die Laufzeit getestet werden.

1.2 Testkonzept und Testablauf

Was wird getestet	Wann	Wie (Testart)	Verifikation	Wiederholung	Vorbedingung	Begründung
<i>Funktionalität der einzelnen Komponenten der Module</i>	<i>Laufend während der Entwicklung der einzelnen Komponenten und wenn sie zu Module zusammengefasst werden</i>	<i>Unit Tests im Test Driven Development</i>	<i>Anhand der Spezifikationen aus dem Projektantrag</i>	<i>Falls Tests negativ sind, Vor/Nach jedem Refactoring, Zur Fertigstellung von Modulen</i>	<i>Die Anforderungen und der Projektauftrag müssen erstellt & korrekt formuliert sein</i>	<i>Unit Tests eignen sich hervorragend für TDD, da der Erstellungsaufwand sehr gering ist und sie direkt von den Entwicklern erstellt werden können.</i>
Functionality	Sobald das System zusammengestellt ist./ Sobald ein Modul fertig ist.	Unit Test	Bedingungen, die einen Teil des Codes erfüllen muss.	Wenn ein Modul fertig ist.	/	Unit Test sind dafür geeignet, um individuelle Module einer Applikation zu testen.
F/NF Requirements	Der Allerletzte Test vor UA.	System Test	Verifikation, Testieren, Validieren der Anwendungsstruktur und der Geschäftsanforderungen.	/	Integrationstests und Unittests.	Die Anwendung wird getestet, bevor sie veröffentlicht wird (an einem vollständigen integrierten Projekt)

Usability	Sobald die GUI fertig ist	User Acceptance Test	Verifikation durch Benutzer/Clients	Wenn das Projekt schon fertig ist. (Falls einige Verbesserungen benötigt werden)	Systemtests, Integrationtests, Unittests.	UAT ist sehr wichtig da es zeigt, ob dabei die Anwendung nach den Wünschen des Benutzers funktioniert.
Functionality	Sobald es eine Änderung gibt, wie z.B. Updates,Patches usw..	Regression Test	Feststellen, dass die Änderung(en) zu keinem Zeitpunkt im Programm (Code) zu neuen Problemen geführt hat.	Nach jede Änderung.	Vorbedingung ist positiv abgeschlossene Unittest(s).	Regression Test ist dafür geeignet, um festzustellen, ob die Änderungen keine negative Auswirkungen auf der Anwendung (Functionality) haben.
Functionality, Supportability, Reliability	Nachdem Module in eine bestimmte Gruppe zusammengesetzt sind.	Integration Test	Functional, Performance, und Reliability-Anforderungen der größeren Teilen	Wenn Module fertig sind. (Und dabei Unittests positiv abgeschlossen sind)	Vorbedingung ist positiv abgeschlossene Unittest(s).	Integration Test zeigt uns, ob die verschiedene Implementierungen von versch. Users korrekt bzw. „integriert“ (ob alles funktioniert wie es funktionieren soll) sind.

2 Äquivalenzklassen

2.1 Analyse und Erstellung

Anforderung: Rabatte

<u>Bedingung</u>	<u>Gültig</u>	<u>Ungültig</u>	<u>Nr</u>
Stammkunden bekommen 1,5% Rabatt.	1,5 % Rabatt	NICHT 1,5% Rabatt	1
10 Video Points gewährleisten 2% Rabatt.	Auch mehr als 10VP gewährleisten 2% Rabatt.	Mehr als 10VP gewährleisten NICHT 2% Rabatt.	2
20 Video Points gewährleisten 6% Rabatt.	Auch mehr als 20VP gewährleisten 6% Rabatt.	Mehr als 20VP gewährleisten NICHT 6% Rabatt.	3
Kumulative Rabatte	Falls Stammkunde und 10VP dann 3,5% Rabatt. Falls Stammkunde und 20VP dann 7,5% Rabatt.	Falls Stammkunde und 10VP dann folgt NICHT 3,5% Rabatt. Falls Stammkunde und 20VP dann folgt NICHT 7,5% Rabatt.	4

Anforderung: Videopoints

<u>Bedingung</u>	<u>Gültig</u>	<u>Ungültig</u>	<u>Nr</u>
Für jede Rechnungssumme im Wert von 2€, bekommt man 1VP.	$\text{Rechnungssumme}/2 = \text{VP}$	$\text{Rechnungssumme}/2 \neq \text{VP}$	5
Stammkunden bekommen zusätzlich auch noch 1VP	$\text{Rechnungssumme}/2 = \text{VP} + 1$	$\text{Rechnungssumme}/2 \neq \text{VP}$	6

Anforderung: Altersfreigabe

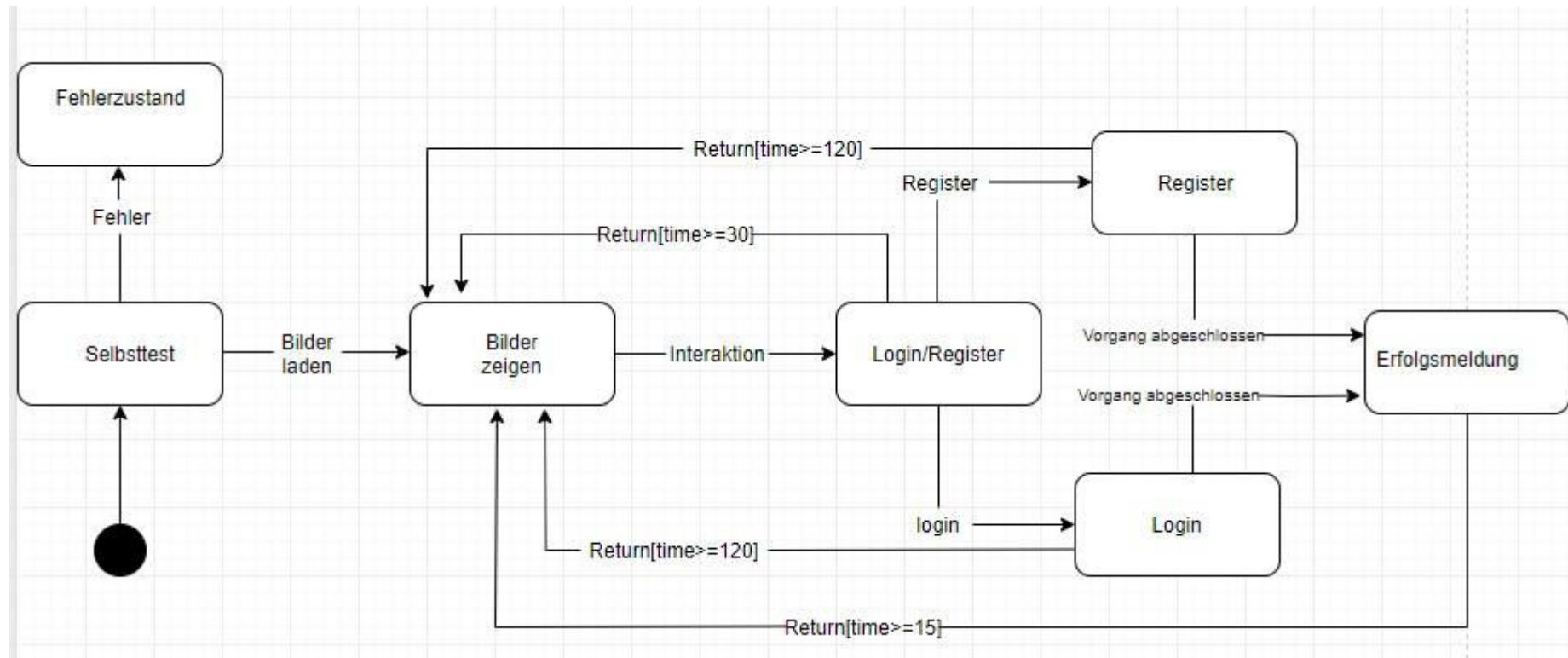
<u>Bedingung</u>	<u>Gültig</u>	<u>Un gültig</u>	<u>Nr</u>
FSK 6	Alter = 6	Alter <6	7
FSK 12	Alter = 12	Alter < 12	8
FSK 16	Alter = 16	Alter < 16	9
FSK 18	Alter = 18	Alter < 18	10
RATED 21	Alter = 21	Alter < 21	11
FSK 0	Age = 1		12

2.2 Testfallableitung

Nr.	NF / FF	Vorbedingung	Eingabe	Beschreibung	Erwartetes Ergebnis
2	FF	Auch mehr als 10VP gewährleisten 2% Rabatt. (≥ 10 VP = 2%)	8	Es gibt keinen Rabatt, wenn man 8VP hat.	False
1	NF	1,5 % Rabatt	1,5 %	Stammkunden bekommen immer pro jede Rechnung 1,5% Rabatt.	True
6	NF	Rechnungssumme/2 = VP + 1	10	Stammkunden bekommen zusätzlich auch noch 1VP.	True
8	FF	Alter = 12	7	Return True für alle Personen die 12 Jahre alt sind.	False
9	FF	Alter = 16	14	Return True für alle Personen die 16 Jahre alt sind.	False

3 State Charts

3.1 Erstellung



3.2 Testfallableitung

Nr.	NF / FF	Vorbedingung	Eingabe	Beschreibung	Erwartetes Ergebnis
1	NF	Mode=movie	Show help. /Es wird "Help Schreen" geöffnet.	Show help kann man aus Zustand Movie Selection und Display Cart aufrufen.	Zustandsänderung von Movie Selection auf Help Screen.
2	NF	Zustand=Movie selection	Return	Diagramm geht von Movie Selection Zustand auf Display Cart Zustand.	Zustandsänderung von Movie Selection auf Display Cart.
3	NF	Mode=cart	Select movies	Zustand ist 'Display Cart', die Aktion die ausgeführt wird ist 'select movies', und danach wird neue Zustand 'Movie selection' sein.	Zustandsänderung von Display Cart auf Movie Selection.
4	FF	Zustand=Checkout	finish checkout	Falls ein Film nicht vorhanden ist, dann wird Zustandsänderung von Checkout auf Display Cart. (=Fehler)	Zustandsänderung von Checkout auf Display cart.
5	NF	Zustand=Checkout	finish checkout	Alle Filme sind Verfügbar und das Prozess kann beendet werden. (=Geht in Endzustand)	Zustandsänderung von Checkout auf Endzustand.