Software-Qualitätssicherung

Übung 2

Inhaltsverzeichnis

0 Disclaimer

0.1 Angabe

Die Übung ist als aufbauende Übung ausgelegt. Verwenden Sie daher für alle Beispiele den Projektauftrag aus Übung 1 sowie alle bisherigen Übungsergebnisse.

0.2 Allgemeine Informationen

Bei der Übung handelt es sich um eine Einzelarbeit. Alle Abgaben werden von uns auf Plagiate überprüft. Generell werden Plagiate mit 0 Punkten bewertet. Weiters behalten wir es uns vor, Studierende im Verdachtsfall zu einem verpflichtenden Kontrollgespräch einzuladen.

Bedenken Sie, dass wenn sehr viele Studierende gleichzeitig auf das TUWEL zugreifen, der Server überlastet werden kann und es zu längeren Upload-Zeiten bei Ihren Abgaben kommen könnte. Lassen Sie sich deshalb in Ihrem eigenen Interesse nicht bis zur letzten Minute mit Ihrer Abgabe Zeit.
Verspätete Abgaben werden ausnahmslos nicht akzeptiert!

0.3 Benennung der Abgaben & Kompilierbarkeit

Bitte verwenden Sie, sofern mitgeliefert, die Vorlage und beachten Sie, dass ausschließlich richtig benannte Dateien im angegebenen Format bewertet werden.

Achten Sie bei der Benennung der Dateien auch darauf das danach noch all Ihre abgegebenen Quelldateien kompilierbar sind! Sollte Ihre Abgabe nicht kompilieren wird Sie nicht bewertet.

0.4 Verwendung von Git

Einige Beispiele enthalten ein git-Repository. Halten Sie jeden Ihrer Schritte im git-Repository fest. Sie benötigen dazu lediglich folgende Befehle:

```
# Add all untracked files
git add -A
# Commit all changes with the given message
git commit -am "<message>"
# Edit last commit message
git commit --amend -m "<new message>"
# Show commit log
git log
# Checkout a specific revision
git checkout <revisionHash>
```

Denken Sie immer daran aussagekräftige Commit Messages zu wählen!

Achten Sie darauf, dass Sie am Ende ihrer Arbeit immer alles auf dem master-Branch liegen haben und squashen Sie keine Commits.

Nähere Informationen zur verwendung von Git sowie den Download erhalten Sie unter:
https://git-scm.com/

0.5 Entwicklungsumgebung

Das Beispiel ist auf die Verwendung von IntelliJ IDEA (https://www.jetbrains.com/idea/) ausgelegt. Die Entwicklungsumgebung steht Studierenden, unter https://www.jetbrains.com/student/, kostenfrei zur Verfügung.
Die Verwendung von IntelliJ IDEA als Entwicklungsumgebung ist nicht zwingend erforderlich. Sie können alternativ auch Eclipse (https://eclipse.org/) verwenden oder mit einem Codeeditor wie zum Beispiel Atom (https://atom.io/) arbeiten.

1 Testplanerstellung

In der ersten Übung haben Sie bereits einen Plan für Reviews des Projektes 'Movierental' entworfen. Nun geht es darum, ein umfassendes Testkonzept für 'Movierental' zu entwerfen. Dazu ist es notwendig, sich zuerst zu überlegen, welche Anforderungen wie getestet werden können und bei welchen das Testen nicht möglich ist. Die zweite Aufgabe besteht darin, die Reihenfolge bzw. Zusammenhänge der Tests darzustellen und auch festzulegen, wann und in welchen Abhängigkeiten diese zueinander ausgeführt werden.

1.1 Anforderungseinteilung

Teilen Sie zuerst die Anforderungen in funktionale und nicht-funktionale Anforderungen auf. Wählen Sie nun 5 nicht-funktionale Anforderungen und beantworten Sie folgende Fragen zu jeder Anforderung:

- Ist die Anforderung testbar? Begründen Sie Ihre Antwort! Nur 'Ja' oder 'Nein' hinzuschreiben reicht nicht aus!
- Wie kann die Anforderung getestet werden? (Keine genaue Beschreibung, sondern z.B. automatisiert (wie?), nur manuell mit speziellen Geräten (welche?), gar nicht, etc.) Begründen Sie auch hier Ihre Antwort

1.2 Testkonzept und Testablauf

Überlegen Sie sich, auf welche Art und Weise das Projekt getestet werden soll, (Unit Tests, Integration- und Systemtests, Akzeptanztests, Regressionstests, etc.), bzw. wie Tests und deren Ergebnisse verifiziert werden sollen (Testorakel, Spezifikation, Durchlaufzeiten, Kunde, etc.) und wann, wie oft, warum, mit welchen Vorbedingungen diese im Projekt durchgeführt werden sollten sowie was genau getestet werden soll (Funktionalität, Usability, Anwortzeiten).

2 Äquivalenzklassen

2.1 Analyse und Erstellung

Bestimmen Sie für folgende Anforderungen die Äquivalenzklassen. Achten Sie dabei genau darauf, dass die Unterscheidung zwischen Äquivalenzklasse und Klassenkombination aus hervorgeht. Markieren Sie auch die entsprechenden Grenzwerte.
Sie können die Klassen grafisch aufbereiten. Achten Sie aber darauf, dass in den meisten Fällen, eine textuelle Beschreibung zur Unterstützung hilfreich/notwendig ist.

10. Anforderung: Rabatte
11. Anforderung: Videopoints
12. Anforderung: Altersfreigabe

2.2 Testfallableitung

Leiten Sie auf Basis der in **Analyse und Erstellung** erstellten Äquivalenzklassen eine ausreichend Anzahl an Testfälle, für Normalfall sowie Fehlerfall ab. Achten Sie beim Erstellen der Testfalldefinition auch auf die passende Vorbedingung und Nachbedingung sowie das passende Fehlerverhalten.

3 State Charts

Der Kunde denkt über die Möglichkeit eines Selbstbedienungsterminals für seine Kunden nach. Mittels des Selbstbedienungsterminals soll es möglich sein sich als Neukunde zu registrieren, Videos auszuleihen und diese Zurückzugeben.

3.1 Erstellung

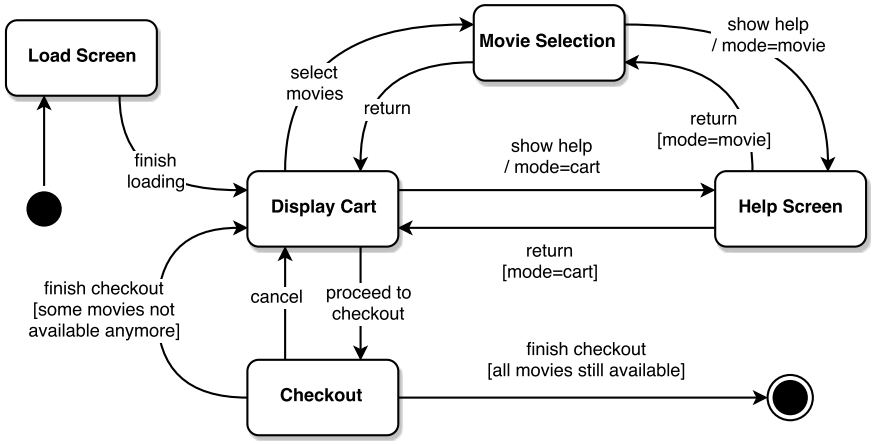
Das Selbstbedienungsterminal soll folgendes User-Interface zur Verfügung stellen:

Nach dem Einschalten des Selbstbedienungsterminals wird ein Selbsttest durchgeführt. Wenn dabei ein Fehler festgestellt wird, verharrt er in einem Fehlerzustand. Ansonsten zeigt er Bilder von aktuellen Blockbustern bis eine Userinteraktion erfolgt. Sobald eine Interaktion erfolgt, wird der Auswahlschirm für die Registrierung angezeigt. Der Kunde kann nun wählen ob er sich als Neukunde registrieren möchte oder ob er bereits ein Kundenkonto hat. Sollte der Benutzer 30 Sekunden lang keine Wahl treffen, geht das Terminal wieder in das Anzeigen von Bildern über. Je nach Auswahl des Kunden wird entweder der Login- oder der Registrierungsbildschirm dargestellt. (Wichtig: Die detaillierten Abläufe des Login- und Registrierungs- bildschirms/prozesses sollen hier nicht modelliert werden!) Wenn der jeweils gewählte Vorgang abgeschlossen ist gibt das Terminal eine Erfolgsmeldung aus und geht dann nach 15 Sekunden wieder zum Anzeigen von Bildern über. Sollte während des gewählten Vorgangs länger als 120 Sekunden keine Interaktion erfolgen geht das Terminal aus Sicherheitsgründen wieder zum Anzeigen von Bildern über und bricht dabei den gewählten Vorgang ab.

Entwerfen Sie ein State Chart für das Interface des Selbstbedienungsterminals.

3.2 Testfallableitung

Nach einer Evaluierung der verschiedenen Möglichkeiten steht fest, dass für das Selbstbedienungsterminal eine fertige Lösung verwendet werden. Das Selbstbedienungsterminal hat folgendes, folgendes, durch ein State Chart beschriebenes, Top Level Interface des Ausleihprozesses. Leiten Sie aus dem State Chart Testfälle ab, sodass alle Transitionen mindestens einmal ausgeführt werden.



4 Abgabe

Folgende Dateien sind abzugeben:

- QSVU_UEbung2_<Matrikelnummer>_<Nachname>_<Vorname>.pdf

Bitte verwenden Sie, sofern mitgeliefert, die Vorlage und beachten Sie, dass ausschließlich richtig benannte Dateien im angegebenen Format bewertet werden.
Bitte laden Sie ihre Abgabe rechtzeitig hoch. Verspätete Abgaben werden ausnahmslos nicht akzeptiert! Beachten Sie unbedingt den **Disclaimer**.