
Modul APPE | TA.APPE.F1101

APPE Filialen-Bestellsystem

T e s t p l a n

Projekt	APPE Filialen-Bestellsystem	
Dokument	Testplan	
Schule	Hochschule Luzern, Technik & Architektur	
Modul	TA.APPE.F1101	
Projektteam	<p>Galliker Thomas Studiengang Informatik (BB) Panorama 6123 Geiss Tel. +41 79 504 80 70 thomas.galliker@stud.hslu.ch</p> <p>Stocker Elias Studiengang Informatik (BB) Schybenacherweg 12 6285 Hitzkirch Tel. +41 79 603 10 89 elias.stocker@stud.hslu.ch</p>	<p>Gasser Martin Studiengang Informatik (BB) Mettenwilhöhe 4 6275 Ballwil</p> <p>Tel. +41 79 755 04 32 martin.gasser@stud.hslu.ch</p>
Dozenten	Prof. Jörg Hofstetter	
Letzte Änderung	3. Juni 2011, 11:12:00 Uhr	

Änderungsprotokoll

Version	Datum	Autor	Beschreibung
0.1	10.03.2011	gat	Initialversion von Vorlage erstellt
0.2	20.04.2011	ste	Testfälle für Systemtests erstellt
0.2	24.04.2011	gat	Testfälle für Unit- und Integrationstests erstellt
0.3	26.05.2011	ste	Erweiterung Testfälle

Inhalt

1	Einleitung.....	4
1.1	Ziel & Zweck dieses Dokuments.....	4
1.2	Begriffe & Abkürzungen	4
2	Testphilosophie	5
2.1	Unit-Test.....	5
2.2	Integrationstest	5
2.3	Systemtest.....	5
3	Testplan.....	6
4	Testspezifikation.....	6
4.1	Testobjekte	6
4.2	Wesentliche Testaspekte	6
4.3	Testfälle.....	7

Abbildungsverzeichnis

Es konnten keine Einträge für ein Abbildungsverzeichnis gefunden werden.

Tabellenverzeichnis

Tabelle 1:	Begriffserklärungen.....	4
Tabelle 2:	Abkürzungserklärungen	4
Tabelle 3:	Unit- und Integrationstests	7

1 Einleitung

1.1 Ziel & Zweck dieses Dokuments

In diesem Dokument wird das Testing im APPE-Projekt „Filialen-Bestellsystem“ geplant. Anwendungsfälle werden in Testfällen abgebildet. Jeder Testfall enthält genaue Spezifikationen bezüglich Testbedingungen, Testvorgehen und erwarteten Resultaten. Alle Informationen, welche mit der Planung von Tests im Zusammenhang stehen, werden in diesem Dokument festgehalten. Die Testprotokolle, d.h. die Protokollierung der Testabläufe, werden in separaten Dokumenten (sog. Testprotokolle) abgelegt.

1.2 Begriffe & Abkürzungen

Begriff	Erklärung
Peer Review	Verfahren zur Qualitätssicherung durch unabhängige Gutachter „Peers“ (engl. für Ebenbürtige, Gleichrangige)
Unit Test	Dient zur Verifikation der Korrektheit von Modulen einer SW
Systemtest	Testphase wo das gesamte System auf die Anforderungen (funktionale und nicht funktionale Anforderungen) getestet wird
Integrationstest	Testet die voneinander abhängigen Komponenten
fbsguiclient	Grafische Benutzerschnittstelle (Presentation Layer) für Zugriff auf fbsservice.
fbsservice	Business Logic Layer, enthält die Implementation der Geschäftslogik
fbldata	Enthält die Entity Objects, welche von JPA zum Datenaustausch mit der Datenbank genutzt werden
fbcommon	Library mit allgemein genutzten Datentypen und Konfigurationen

Tabelle 1: Begriffserklärungen

Abkürzung	Erklärung
HSLU	Hochschule Luzern
APPE	"Applikationsentwicklung"; Modulbezeichnung HSLU
gat	Namenskürzel für Galliker Thomas
ste	Namenskürzel für Stocker Elias
gam	Namenskürzel für Gasser Martin

Tabelle 2: Abkürzungserklärungen

2 Testphilosophie

Unsere Testphilosophie besteht aus drei wesentlichen Teilen:

- Testen jeder Komponente in sich (Unit-Tests)
- Testen des Zusammenspiels der Komponenten (Integrations- und Systemtests)

2.1 Unit-Test

In den Unit-Tests werden einzelne, in sich abgeschlossene Klassen getestet. Die Unit-Tests sollen – wo immer möglich – parallel zur Entwicklung erstellt und ausgeführt werden. Die Unit-Tests sollen nach dem Test-First Konzept mit JUnit durchgeführt werden.

Der Test-First Ansatz geht davon aus, dass der Entwurf von Tests und deren Ausführung auch das Design des Programms vorantreibt. Bevor ein Stück eines Programms geschrieben wird, wird zuerst ein Test entworfen. Anschliessend wird nur so viel Programmcode geschrieben, wie der Test verlangt. Dementsprechend findet die Programmentwicklung in einer raschen Folge von Test- und Implementierungsschritten statt.

Funktionalität ist bei Unit-Test gleichbedeutend mit Ein-/Ausgabeverhalten der zu prüfenden Einheit. Dabei wird jede Einheit isoliert getestet um sicher zu stellen, dass andere Einheiten keinen Einfluss auf das Verhalten der zu prüfenden Einheit haben und somit ein Fehler klar zu zuordnen ist.

2.2 Integrationstest

Der Integrationstest hingegen prüft das Zusammenspiel von Komponenten und deren Wechselwirkung. Sowohl Unit-Tests als auch Integrationstests werden an den Schnittstellen der zu prüfenden Einheit durchgeführt.

Wichtig ist, dass die Komponenten auf Basis der Systemspezifikation untereinander korrekt agieren und dass eine gute Leistung gewährleistet ist.

2.3 Systemtest

Unit- und Integrationstest können mit gängigen Tools automatisiert werden. Komplette Systemtests werden in diesem Projekt manuell durch Testpersonen durchgeführt. Getestet werden hauptsächlich die in der Kundenanforderung erfassten Anwendungsfälle (Use Cases) sowie die funktionalen und nicht-funktionalen Anforderungen.

3 Testplan

Grundsätzlich sollen alle entwickelten Komponenten und deren Wechselwirkung kontinuierlich getestet werden. Unit- und Integrationstests werden auf einem Jenkins Buildserver automatisch ausgeführt.

Unit-Tests sollen grundsätzlich nach jedem Erstellen einer Klasse erstellt und ausgeführt werden. Die System- und Integrationstests sollen erst nach dem vollständigen Erstellen der Komponenten durchgeführt werden.

4 Testspezifikation

4.1 Testobjekte

- fbsguiclient
- fbsservice
- fbsdata
- fbscommon

4.2 Wesentliche Testaspekte

Auf folgende wesentliche Testaspekte muss grossen Wert gelegt werden:

- Anforderungserfüllung: Erfüllen der geforderten Anforderungen.
- Code Coverage: Abdeckung der wichtigsten Code-Stellen muss garantiert sein.
- Datenkonsistenz: Vor und nach dem Ausführen von Tests muss die Testdatenbank in einem Zustand hinterlassen werden, welcher wiederholtes (automatisiertes) Testen zulässt.
- Interoperabilität: Verschiedene Laufzeitumgebungen müssen getestet werden.

4.3 Testfälle

Die nachfolgenden Kapitel liefern Informationen zu den Testfällen, welche im Laufe des Projekts entstanden sind. Während Unit- und Integrationstests mittels automatisierter Softwaretests (JUnit)

4.3.1 Unit- und Integrationstests

Folgende Unit- und Integrationstests wurden während der Entwicklung erstellt und ausgeführt:

Test Name	Beschreibung
fbcommon.ConfigTest	Lesen- und Schreibtests für Konfigurationsdateien.
fbdata.AccountTest	Erstellen von Entitäten, welche mit Account in Bezug stehen: Person, Account, Accountgroup und Country.
fbsguiclient.LoginTest	Starten eines StoreServers, End-zu-End Login-Versuch.
fbsservice.CountryTest	Länderliste laden und ausgeben.
fbsservice.OrdersTest	Bestellliste laden und ausgeben.
fbsservice.ProductsTest	Test Produkt erstellen, Produktliste laden, Produkte eines bestimmten Stores laden, Produktdetails aktualisieren, Produkt löschen.
fbsservice.SecurityTest	Personen- und Account-Entitäten erstellen, erfolgreiches Login testen, fehlerhaftes Login prüfen, Rollenzugehörigkeit eines Benutzers testen, Vergleich von aktueller Login-Session mit Customer Identity.
fbsservice.ShoppingTest	Warenkorb erstellen, Produkte in Warenkorb legen, Produkte aus Warenkorb löschen, falsche Cart Reservation Deadlines testen, Warenkorb löschen, Bestellung aus Warenkorb erstellen, Inhalt einer Bestellung ausgeben, Bestellung löschen.
fbsservice.StoreTest	Storedetails aktualisieren.

Tabelle 3: Unit- und Integrationstests

4.3.2 Systemtests

ID	1
Beschreibung	Erfolgreiche Anmeldung an der Applikation (Login)
Vorgehen	<ol style="list-style-type: none"> 1. FBS GUI Client starten 2. Login Fenster erscheint 3. Store wählen (localhost, wenn Server lokal, sonst IP-Adressen Eingabe möglich) 4. Korrekte Kombination von Username und Passwort eingeben 5. Login Button klicken
Voraussetzung	<ul style="list-style-type: none"> • Server läuft
Erwartetes Resultat	<ul style="list-style-type: none"> • Hauptfenster des FBS GUI Client wird angezeigt.

ID	2
Beschreibung	Fehlerhafte Anmeldung an der Applikation (Login)
Vorgehen	<ol style="list-style-type: none"> 1. FBS GUI Client starten 2. Login Fenster erscheint 3. Store wählen (localhost, wenn Server lokal, sonst IP-Adressen Eingabe

	möglich) 4. Fehlerhafte Kombination von Username und Passwort eingeben 5. Login Button klicken
Voraussetzung	<ul style="list-style-type: none"> • Server läuft
Erwartetes Resultat	<ul style="list-style-type: none"> • Fehlermeldung: „Login failed. Please check credentials.“

ID	3
Beschreibung	Erfolgreich ein Produkt aus der Produktliste in den Warenkorb legen
Vorgehen	<ol style="list-style-type: none"> 1. Customer wählen, für den der aktuell eingeloggte Account einkaufen darf 2. Meldung „Would you like to create a cart for customer [customer name]“ mit “Yes, create new cart” bestätigen 3. Drücke „F2“ (alternativ: Menüpunkt „Products“, dann „View Productlist“) 4. Klicke beim gewünschtem Produkt auf den Button „Add to cart“
Voraussetzung	<ul style="list-style-type: none"> • Server läuft • FBS GUI Client läuft
Erwartetes Resultat	<ul style="list-style-type: none"> • Stock des Produkts (siehe 6. Spalte) wird um die Anzahl 1 reduziert. • Produkt ist im Cart des gewählten Customers vorhanden. (F6 öffnet den Cart des Customers)

ID	4
Beschreibung	Ein Produkt aus der Produktliste in den Warenkorb eines nicht berechtigten Kunden legen
Vorgehen	<ol style="list-style-type: none"> 1. Customer wählen, für den der aktuell eingeloggte Account keine Berechtigung zum Einkaufen hat
Voraussetzung	<ul style="list-style-type: none"> • Server läuft • FBS GUI Client läuft
Erwartetes Resultat	<ul style="list-style-type: none"> • Fehlermeldung: “You’re not authorized to perform this operation.”

ID	5
Beschreibung	Inhalt des Warenkorbs bestellen
Vorgehen	<ol style="list-style-type: none"> 1. Drücke „F6“ (alternativ: Menüpunkt „Customers“, dann „View Cart #xy“) 2. Klicke auf „Proceed to Checkout“
Voraussetzung	<ul style="list-style-type: none"> • Server läuft • FBS GUI Client läuft • Produkte in Warenkorb vorhanden (siehe Testfall Nr. 3)
Erwartetes Resultat	<ul style="list-style-type: none"> • Meldung: Status of your order is “PROCESSING” • Bestellung wird aufgelistet beim Benutzer (F5 öffnet die Bestellungen des Customers)

ID	6
Beschreibung	Produktliste anzeigen/filtern
Vorgehen	<ol style="list-style-type: none"> 1. Drücke „F2“ (alternativ: Menüpunkt „Products“, dann „View Productlist“) 2. Suchwort (z.B. „Apple“) oben in Suche eingeben

Voraussetzung	<ul style="list-style-type: none"> • Server läuft • FBS GUI Client läuft
Erwartetes Resultat	<ul style="list-style-type: none"> • Produktliste wird angezeigt • Produktliste wird gefiltert
ID	7
Beschreibung	Produkteliste sortieren
Vorgehen	<ol style="list-style-type: none"> 1. Drücke „F2“ (alternativ: Menüpunkt „Products“, dann „View Productlist“) 2. Jede Header-Spalte zweimal anklicken und Sortierung überprüfen.
Voraussetzung	<ul style="list-style-type: none"> • Server läuft • FBS GUI Client läuft
Erwartetes Resultat	<ul style="list-style-type: none"> • Produktliste wird angezeigt • Produktliste wird sortiert
ID	8
Beschreibung	Warenkorbinhalt bestellen während noch eine unbezahlte Bestellung mit Status „1.Mahnung“ vorhanden ist.
Vorgehen	<ol style="list-style-type: none"> 1. Drücke „F6“ (alternativ: Menüpunkt „Customers“, dann „View Cart #xy“) 2. Klicke auf „Proceed to Checkout“
Voraussetzung	<ul style="list-style-type: none"> • Server läuft • FBS GUI Client läuft • Produkte in Warenkorb vorhanden (siehe Testfall Nr. 3) • Keine offene Bestellung mit Mahnung vorhanden.
Erwartetes Resultat	<ul style="list-style-type: none"> • Warnung bezüglich offener Mahnung. • Bestellung wird nicht angenommen. Warenkorb bleibt bestehen.