

fit4IPA

Aufgabenstellung für
Leo Rudin



Inhalt

1	Ausgangslage	2
2	Auftragsformulierung	2
2.1	Funktionale Anforderungen.....	3
2.1.1	Präsentations-Schicht.....	3
2.1.2	Business-Schicht	3
2.1.3	Persistenz-Schicht	4
2.2	Nicht funktionale Anforderungen	7
3	Mittel und Methoden	8
3.1	Tools und Technologien	8
3.2	Methodik	8
4	Bewertungskriterien	9
4.1	Individuelle Kriterien Teil B.....	9

1 Ausgangslage

Die *BbcBank* modernisiert ihre Bank-Applikation aus den 90er-Jahren. Technologien, wie CGI (Common Gateway Interface) und CORBA werden abgelöst durch Java Enterprise. Objektorientierte Anwendungen ersetzen prozedurale Programm-Logik.

2 Auftragsformulierung

Im Rahmen der fit4IPA sind folgende Use Case als JEE 7 Anwendung zu implementieren:

- *Konto-Auszug anzeigen (e-banking)*
- *Konto Einzahlung/Auszahlung am Bankschalter*

Zum Lieferumfang gehören:

- Präsentations-, Business- und Persistenz-Schicht
- JUnit Testfälle für die Business-Services

2.1 Funktionale Anforderungen

2.1.1 Präsentations-Schicht

Das User Interface für den Use Case *Konto-Auszug anzeigen* besteht aus zwei HTML-Seiten:

- Kunde authentifizieren (nicht Teil von fit4IPA)
- Konto-Auszug anzeigen

Die Seite für die Anzeige des Konto-Auszugs umfasst folgende Angaben:

- Adresse des im ersten Arbeitsschritt authentifizierten Kunden
- Tabellarische Darstellung des Konto-Auszugs mit einer Zeile pro Transaction-Record:
 - Kolonne 1: Datum (TT.MM.JJJJ)
Konto-Auszug aufsteigend nach Datum sortiert
 - Kolonne 2: Informationen
 - Kolonne 3: Belastungen (Transaction-Typ D)
 - Kolonne 4: Gutschriften (Transaction-Typ C)
 - Kolonne 5: Kontostand

Das User Interface für den Use Case *Konto Einzahlung/Auszahlung* besteht aus zwei HTML-Seiten:

- Kunde suchen (nicht Teil von fit4IPA)
- Einzahlung/Auszahlung durchführen

Die Seite für die Durchführung der Einzahlung/Auszahlung umfasst folgende Angaben:

- Adresse des im ersten Arbeitsschritt selektierten Kunden
- Liste zum Selektieren des Kunden-Kontos
- Checkbox zum Selektieren des Transaktions-Typs (Einzahlung/Auszahlung)
- Eingabefeld für den Transaktions-Betrag (immer CHF)
- Eingabefeld für den Transaktions-Text
- Button zum Auslösen der Transaktion
- Status-Feld für die Resultatanzeige:
 - Bestätigung erfolgreiche Transaktions-Durchführung mit Anzeige des neuen Saldos
 - ungültige Kontonummer
 - bei Auszahlung: Transaktion wegen Überschreiten des Kredit-Limits nicht durchgeführt mit Anzeige des aktuellen Saldos

Die Präsentations-Komponenten werden mit JSF entwickelt.

2.1.2 Business-Schicht

Für den Use Case *Konto-Auszug anzeigen* sind folgende Business-Services sind zu entwickeln:

- Kundendaten holen
 - Request: – Kundennummer [999-999999]
 - Response: – Kundendaten (ohne Kontoangaben)
 - Exception im Fehlerfall

- Kontonummern des Kunden holen
 - Request: - Kundennummer [999-999999]
 - Response: - Liste mit den Kontonummern des Kunden (IBAN)
 - Exception im Fehlerfall

- Kontodaten mit Transaction-Records holen
 - Request: - Kontonummer (IBAN)
 - Response: - Kontodaten
 - Transaction-Records aufsteigend sortiert nach Transaction-Zeitpunkt
 - Exception im Fehlerfall

Für den optionalen Use Case *Konto Einzahlung/Auszahlung* sind folgende Business-Services zu entwickeln:

- Einzahlung durchführen

Schreibt den Transaction-Record mit Typ C (Gutschrift) , erhöht den Konto-Saldo

- Request: - Kontonummer (IBAN)
- Betrag (im CHF)
- Text
- Response: - neuer Saldo
- Exception im Fehlerfall

- Auszahlung durchführen

Stellt sicher, dass das Konto-Limit nicht überschritten wird. Schreibt den Transaction-Record mit Typ D (Belastung), reduziert den Konto-Saldo

- Request: - Kontonummer (IBAN)
- Betrag (im CHF)
- Text
- Response: - neuer Saldo
- Exception im Fehlerfall

Die Business-Komponenten werden mit EJB Web Service (SOAP) entwickelt.

2.1.3 Persistenz-Schicht

Die Persistenz-Schicht wird im Bottom-Up Verfahren entwickelt. Die Datenbank (*bbcbank*) ist bereits erstellt und mit Testdaten initialisiert.

Die DB umfasst folgende Tabellen:

▪ CUSTOMER

Kolonne	Typ	Länge	Index/Key PK=Primär SK=Sekundär FK=Fremd	NULL erlaubt	Beschreibung
ID	INT		PK	nein	Kunden-ID fortlaufende automatisch vom DB Server generierte Nummer
CUSTOMER_NUMBER	VARCHAR	10	SK	nein	Kundennummer [999-999999]
TITLE	VARCHAR	10		nein	Anrede
FIRST_NAME	VARCHAR	30		nein	Vorname
LAST_NAME	VARCHAR	30		nein	Nachname
STREET	VARCHAR	30		nein	Strasse
CITY	VARCHAR	30		nein	Ort
ZIPCODE	SMALLINT			nein	Postleitzahl

▪ ACCOUNT

Kolonne	Typ	Länge	Index/Key PK=Primär SK=Sekundär FK=Fremd	NULL erlaubt	Beschreibung
ID	INT		PK	nein	Konto-ID fortlaufende automatisch vom DB Server generierte Nummer
IBAN	VARCHAR	30	SK	nein	Kontonummer
BALANCE	DECIMAL	9,2		nein	Saldo
OVERDRAFT	DECIMAL	7,2		nein	Kredit-Limit
CUSTOMER_ID	INT		FK	nein	Kunden-ID

▪ TRANSACTION_RECORDS

Kolonne	Typ	Länge	Index/Key PK=Primär SK=Sekundär FK=Fremd	NULL erlaubt	Beschreibung
ID	INT		PK	nein	Transaction-ID fortlaufende automatisch vom DB Server generierte Nummer
TYPE	CHAR	1		nein	Typ C = Gutschrift D = Belastung
TEXT	VARCHAR	30		nein	Text
AMOUNT	DECIMAL	9,2		nein	Betrag
TRANSACTION_TIME	TIMESTAMP			nein	Transaction- Zeitpunkt
ACCOUNT_ID	INT		FK	nein	Konto-ID

Die Persistenz-Komponenten werden mit JPA entwickelt.

2.2 Nicht funktionale Anforderungen

Die wichtigen durchgeführten System-Aktivitäten sind mittels Java Logging API zu protokollieren.

3 Mittel und Methoden

3.1 Tools und Technologien

Programmierung	<ul style="list-style-type: none">- Java SE 8u101- Java EE 7 SDK Update 2 <p>JDK Utility <i>wsimport</i> zum Generieren der Clientseitigen Web Service Artefakte</p> <p>JSF (Präsentations-Schicht), EJB Web Service (Business-Schicht), JDBC (Persistenz-Schicht), SOAP-Http(Kommunikation)</p>
IDE	<ul style="list-style-type: none">- Eclipse Neon for Java EE Developers (4.6) mit GlassFish 4.1 Plug-In (Deployment)- ArgoUML, Release 0.34
Team-Environment	<ul style="list-style-type: none">- SVN
Testing	<ul style="list-style-type: none">- JUnit 4- SoapUI 5.2.1
Applikations-Server	<ul style="list-style-type: none">- GlassFish 4.1
Datenbank-Server	<ul style="list-style-type: none">- MySQL 5.7.14- MySQL Workbench 6.3.7- Connector/J 5.1.39 (JDBC-Treiber)

3.2 Methodik

Wasserfall-Methode oder agile Softwareentwicklung.

4 Bewertungskriterien

Die Arbeit wird nach den Kriterien der IPA beurteilt. Der Kriterienkatalog findet sich hier:

https://www.kklit.ch/kklit/wp-content/uploads/Kriterienkatalog_2014.pdf

4.1 Individuelle Kriterien Teil B

Leitfrage	Inhalt
162	Entwurf – SW-Architektur Ist der Entwurf den Regeln moderner SW-Architektur entsprechend modularisiert/strukturiert?
235	Entwurf mit UML Sind Datenmodell/ERM, Use Cases und Activity Diagramm korrekt nach UML 2.0 dargestellt?
165	Implementierung von Lösungen (Programmieren) Ist der Kandidat in der Lage die vorgeschlagenen Lösungen zu implementieren?
194	Plausibilisierung der Benutzer-Eingaben Werden die Eingaben des Benutzers überprüft?
248	OR Mapping selbst implementiert Wurde der OR Mapper korrekt implementiert? Wurde eine gute, erweiterbare, wiederverwendbare Architektur implementiert? Wurde ein API zum Laden und Speichern der Businessobjekte implementiert?
224	Logging (automatisch oder von Hand) Werden relevante Ereignisse in eine Log-Datei sinnvoll protokolliert?
124	Testfälle (Programmierung) Wurde das Programm mit ausreichenden Testfällen getestet, wurden angemessene Testverfahren und Testmethoden angewendet?
225	Versionsverwaltung mit Verwaltungs-SW Wie ist sichergestellt, dass während der Facharbeit keine wichtigen Versionen von Daten, Programmen und Dokumenten verloren gehen? [Achtung: dieses Kriterium fordert deutlich mehr als '6 Organisation des eigenen Arbeitsplatzes']