

Projektauftrag

autoPSI

Inhalt

Inhalt.....	2
Einleitung und Projektbeschreibung.....	3
Vertragspartner.....	3
Auftraggeber.....	3
Ansprechpersonen des Auftraggebers.....	3
Auftragnehmer.....	3
Ansprechpersonen des Auftragnehmers.....	3
Work-Breakdown-Structure.....	4
Rahmenbedingungen.....	5
Projektziele.....	5
Anforderungen an autoPSI.....	6
Allgemeine Anforderungen.....	6
Termine.....	6
Verwaltung der lokalen Daten.....	6
Funktionalität.....	6
Lokale Funktionen.....	6
SBC Funktionen.....	7
Projektabgrenzung.....	7
Ressourcen.....	8
Verfügbare Personen.....	8
Finanzielle Ressourcen.....	8
Sonstige Ressourcen.....	8
Hardware.....	8
Softwaretool.....	8
Sonstiges.....	8
Projektorganisation.....	8
Rollenverteilung.....	8
Zeitplan.....	9
Risikoabschätzung.....	9
Lieferkomponenten.....	11
Weitere Komponenten.....	11
Grobe Datenbankbeschreibung.....	11

Einleitung und Projektbeschreibung

autoPSI (PSI steht für PersonalStudentInformation) ist ein Projekt, welches im Rahmen der Übung Software Engineering 1 an der TU Wien entstehen soll. Das Hauptziel des Projektes ist die Entwicklung einer Applikation, die es Studenten aller Fakultäten ermöglichen soll, universitäre Daten und Termine zu organisieren. Dabei wird besonders Wert darauf gelegt, den Studenten eine einfache Möglichkeit zu geben, wichtige Daten möglichst direkt auszutauschen.

Vertragspartner

Auftraggeber

Auftraggeber ist der Studienassistent bzw. die Technische Universität Wien.

Name: Matthias Heindl

E-Mail: matthias.heindl@qse.ifs.tuwien.ac.at

Ansprechpersonen des Auftraggebers

Name: Jakob Frank

Funktion: SE-Tutor

E-Mail: jakob.frank@qse.ifs.tuwien.ac.at

Telefon: 0699 / 105 88 742

Auftragnehmer

Auftragnehmer ist die Gruppe 06 der Übung SE1 im SS06.

Ansprechpersonen des Auftragnehmers

Name: Rudolf Mildner

Funktion: Teamkoordinator

E-Mail: r.mildner@gmx.at

Name: Stephan Zapotocky

Funktion: Technischer Leiter

E-Mail: stephe@kabsi.at

Name: Daniel Feledi

Funktion: Dokumentationsbeauftragter

E-Mail: e0426231@student.tuwien.ac.at

Name: Alpay Firato

Funktion: Testbeauftragter, Risikobeauftragter

EMail: e0426403@stud3.tuwien.ac.at

Work-Breakdown-Structure

<i>Nr</i>	<i>Struktur</i>	<i>Aufgaben</i>	<i>Beginn</i>	<i>Ende</i>	<i>Aufw and/K osten in Stund en</i>
1	Projektstart		06.04. 2006	02.05.2006	14
1.1		Kick-Off-Meeting	06.04.2006		1
1.2		Projektauftrag erstellen	07.04.2006	02.05.2006	12
1.3		MR1	24.04.2006	24.04.2006	1
2	Analysephase		25.04.2006	Mitte Mai	71
2.1		Evaluierung			5
2.2		Anforderungen feststellen			5
2.3		Use-Case erstellen			20
2.4		Anforderungsdokument erstellen			10
2.5		grobe Programmstruktur			15
2.6		UI-Prototypen erstellen			15
2.7		IR1	11.05.2006	11.05.2006	1
3	Designphase		Mitte Mai	Ende Mai	95
3.1		Feinstruktur planen			83
3.1.1		Datenbank			25
3.1.2		User Interface			15
3.1.3		Datenbankzugriff			18
3.1.4		SpaceBasedComputing			25
3.3		Dokumentation			10
3.4		IR2	23.05.2006	23.05.2006	1
3.5		MR2	30.05.2006	30.05.2006	1
4	Implementierung		Ende Mai	Ende Juni	267
4.1		Datenbank			7

<i>Nr</i>	<i>Struktur</i>	<i>Aufgaben</i>	<i>Beginn</i>	<i>Ende</i>	<i>Aufw and/K osten in Stund en</i>
4.1.1		DB-Initialisierungsscripte			7
4.2		Datenbankzugriff			90
4.2.1		DataAccessObjects			50
4.2.2		Datensatzobjekte			40
4.3		User Interface			110
4.3.1		Hauptfenster/Terminsicht			40
4.3.2		SBC-Interface			30
4.3.3		Eingabe/Bearbeiten			30
4.3.4		DB-Tabellenverwaltung			10
4.4		SpaceBasedComputing			60
4.4.1		Verwaltungskomponenten			60
4.5		Finale Dokumentation			20
4.6		IR3	13.06.2006	13.06.2006	1
5	Testphase		Mitte Juni	Ende Juni	113
5.1		Handbuch			20
5.2		interner Test			70
5.3		externer/offener (Beta-)Test			20
5.4		Release/Präsentation/MR3	23.06.2006	23.06.2006	3

Rahmenbedingungen

- autoPSI soll in Java programmiert werden
- Als Datenbank soll hsqlDB zum Einsatz kommen
- zum Austausch von Daten soll auf jeden Fall SpaceBasedComputing zum Einsatz kommen

Projektziele

1. Entwicklung einer lauffähigen Java-Anwendung
2. Keine Überziehung des Zeitrahmens
3. Erfüllung der unbedingt zu erfüllenden Anforderungen und möglichst auch Erfüllung der optionalen Anforderungen

4. Erstellung einer Anwenderdokumentation
5. Erstellung einer Programmiererdokumentation inklusive eventuell im Rahmen des Projekts entwickelter APIs

Anforderungen an autoPSI

Allgemeine Anforderungen

Da autoPSI von Studenten aller Fakultäten benutzbar sein soll, muss das Interface intuitiv bedienbar sein. Das heißt, flüssiges Arbeiten, Bedienung und Wartbarkeit stehen im Vordergrund der GUI-Entwicklung.

autoPSI soll aufgrund der Verschiedenheit der Betriebssysteme die von Studenten verwendet werden, möglichst plattformübergreifend funktionieren. Da die Entwicklung mit Java stattfindet, kann dieses Ziel mit relativ kleinem Aufwand erreicht werden.

Termine

autoPSI dient dazu, Termine universitärer und privater Natur zu organisieren. Der Termin ist eines der zentralen Objekte von autoPSI.

Ein Termin ist ein einzelnes bzw. ein sich wiederholendes Ereignis (es soll also zB. möglich sein, einen Termin zu erstellen, der von 11.2.2006 bis 30.6.2006 jeden Montag und Mittwoch zur selben Zeit stattfindet)

An einen Termin kann man beliebige (im Sinne von beliebige Datenbankobjekte) Objekte anhängen (wie zB. Lehrveranstaltung (LVA), Mitschrift, URL, Notiz, Kontakt, Buchinformation, Datei, etc.) Jeder einzelne Zeitpunkt eines Termins kann auch editiert werden (zB. am Mi., 23.7.2006 findet die Veranstaltung eine Stunde später statt) und an jeden einzelnen Zeitpunkt eines Termins kann man ebenfalls spezifischere, speziell für diesen Zeitpunkt relevante (wiederum beliebige, im Sinne von beliebige Datenbankobjekte) Objekte anhängen (zB. eine Notiz für Mo., den 6.2.2006: „Heute muss ich mein Notebook mitnehmen“).

Man kann eine Alarm/Erinnerungsfunktion für Termine und Zeitpunkte aktivieren, die hilft, Termine nicht zu übersehen.

Verwaltung der lokalen Daten

Die Daten werden lokal in einer Datenbank gespeichert, welche mit hsqldb implementiert wird.

Aus den in dieser Datenbank gespeicherten Daten werden bei jedem Zugriff Objekte erstellt, die die Handhabung der Daten komfortabler gestalten.

Bei Änderung der Objekte werden diese wieder in die Datenbank zurückgeschrieben.

Funktionalität

Lokale Funktionen

Die gespeicherten Daten können über das Programm angezeigt, geändert und gelöscht werden und es können neue Tupel hinzugefügt werden. Dabei soll eine komfortable Eingabemöglichkeit geschaffen werden, die es dem Benutzer ermöglicht, Daten schnell und korrekt einzugeben. Die eingegebenen Daten sollen auf ihre Gültigkeit was den Datentyp und den Wertebereich angeht,

autoPSI::Projektauftrag

überprüft werden.

Die Daten umfassen auf jeden Fall Kontaktdaten, LVAs, Lehrmittel (zB. Bücher, Dateien wie Vorlesungsfolien), Notizen, Prüfungsdaten, URLs, Dateien.

Kontaktdaten umfassen auf jeden Fall Namen, Adresse, Geburtsdatum (optional), EMail-Adresse (beliebig viele Eingaben möglich), TelefonNr (beliebig viele Eingaben möglich).

Eine LVA hat auf jeden Fall eine LVA-Nr., einen Titel und Vortragende/r, URL der LVA (optional).

Lehrmittel haben auf jeden Fall einen Namen und einen Typ.

Notizen umfassen einen Titel und eine Beschreibung.

Prüfungsdaten besitzen auf jeden Fall eine zugehörige LVA und eine Note (das Datum wird dadurch bestimmt, dass das Prüfungsobjekt an einen Termin angehängt wird).

URLs haben einen Namen und eine URL.

Dateien haben einen Namen/Titel und eine Referenz. Dateien, die in die Datenbank eingefügt werden, werden in einen eigenen Ordner kopiert und diese Kopie der Datei wird referenziert.

SBC Funktionen

autoPSI besitzt eine auf JavaSpaces basierende Sharingfunktion. Der Benutzer kann aus seinen lokalen Daten diejenigen auswählen, die er mit anderen Benutzern sharen will.

Geht ein autoPSI-Client online, so sucht er auf einem Server nach dem passenden JavaSpace und stellt wenn möglich die Verbindung her.

autoPSI überprüft, welche Teile der Datenbank geshared werden, erstellt daraus sharebare Objekte und stellt diese für eine bestimmte, endliche Lease-Zeit in den JavaSpace.

Wenn an einem gesharten Objekt Änderungen vorgenommen werden, so wird das veraltete Objekt aus dem Space genommen und das neue Objekt in den Space gestellt.

Wenn autoPSI schließlich mit dem Space verbunden ist, ist es möglich, mittels Template-Search gewünschte Datenobjekte zu finden, herunterzuladen und in die lokale Datenbasis zu integrieren. Dadurch lassen sich Daten, die viele Studenten benötigen, schnell, unkompliziert und sicher verteilen.

Projektabgrenzung

- autoPSI soll kein allgemeiner PersonalInformationManager (PIM) sein, sondern speziell auf Studenten abgestimmt
- autoPSI bietet keine oder nur die wichtigsten Verbindungen zu anderen Programmen, wie z.B. Email-Client oder Browser
- autoPSI ist nur für die Benutzung von Studenten konzipiert; es ist möglich, dass autoPSI zB. auch für Lehrende verwendbar ist, es ist aber nicht auf diesen Aufgabenbereich optimiert.
- Eine weitere Betreuung des Projektes nach Fertigstellung der ersten Version ist möglich, aber nicht sicher; das Projekt wird der OpenSource-Gemeinde übergeben
- Es sollen möglichst keine finanziellen Mittel zum Einsatz kommen (exklusive Kleinigkeiten). Die Entwicklung soll mit vorhandener Hardware und OpenSource-Softwaretools erfolgen
- autoPSI kann keine Daten von anderen PIMs übernehmen
- Inhalt bzw. Korrektheit von Daten wie zB. korrekte LVA-Nummern kann das Programm nicht überprüfen
- auch wenn autoPSI die Möglichkeit bietet, Dateien (und zwar als Lehrmittel) auszutauschen,

autoPSI::Projektauftrag

so ist das Programm dennoch kein FileSharing-Tool

Ressourcen

Verfügbare Personen

Folgende Personen sind verfügbar:

- Daniel Feledi
- Alpay Firato
- Rudolf Mildner
- Stephan Zapotocky
- Eventuell weitere Tester aus dem Bekanntenkreis

Finanzielle Ressourcen

Da wir arme Studenten sind, stehen uns nur äußerst begrenzte finanzielle Mittel zur Verfügung. Da im Prinzip alles notwendige vorhanden ist, sollte das kein größeres Problem darstellen.

Sonstige Ressourcen

Hardware

- 3 Notebooks
- 4 Computer

Softwaretool

- Eclipse IDE
- Ants
- OpenOffice
- Microsoft Office (verschiedene Versionen?!)
- WinXp, evtl. auch Windows Vista und andere Versionen
- Linux (Debian, ?)

Sonstiges

- (Eventuell) Sourceforge-Projekt inklusive SVN

Projektorganisation

Rollenverteilung

Folgende Rollen stehen derzeit fest:

- Daniel Feledi: Dokumentationsverantwortlicher
- Alpay Firato: Testkoordinator, Risikobeauftragter
- Rudolf Mildner: Projektleiter
- Stephan Zapotocky: Technischer Leiter

Zeitplan

Wichtige Termine:

- Kick-Off-Sitzung am 6. April 2006 um 10:00
- MR1 (Projektauftrag) am 24. April 2006 um 9:00
- IR1 (Erstellung von Anforderungen) am 11. Mai 2006, 9:00-10:00
- IR2 (Entwurf) am 23. Mai 2006, 9:00-10:00
- MR2 (Zwischenbericht mittels Präsentation) am 30. Mai 2006, 9:00-10:00
- IR 3 (Implementierung) am 13. Juni 2006, 9:00-10:00
- MR3 (Präsentation mittels Produktpräsentation) am 27. Juni 2006 9:00-10:00

Risikoabschätzung

Aus Platzgründen ist die Übersicht auf zwei Tabellen aufgeteilt.

Nr.	Name	Kurzbeschreibung	Art	Typisierung	Priorisierung
1	Personalausfall	Ausfall eines Teammitglieds	1	Gesamtes Projekt	1
2	Datenverlust	---	1	Gesamtes Projekt	1
3	Verzögerung	Geplanter Fertigstellungstermin nicht eingehalten	2	Planung, Management	1
3a	Ziel nicht erreicht	Vorgegeb. Ziele bis zum Fertigstellungstermin nicht vollständig erreicht	2	Planung, Management	1
4a	Schlechtes UI-Design	Benutzerschnittstelle schlecht designt oder implementiert	2	Entwicklung	2
4b	Schlechtes Softwaredesign	Software fehlerhaft oder funktioniert nicht wie	2	Planung, Entwicklung	2

		gewünscht			
5	Konkurrenz	Konkurrenz bringt ähnliches bzw. besseres Produkt früher auf den Markt	1	Auftraggeber	3
6	Projektabbruch	Projekt wird für Auftraggeber uninteressant	2	Auftraggeber	3
7	Interne Konflikte	Konflikte innerhalb des Teams	1	Gesamtes Team	2
8	Mangelndes Know-How	---	2	Gesamtes Team	1
9	Motivationseinbrüche	Team verliert Interesse an Projekt	2	Gesamtes Team	2

Nr.	Eintrittswahrsch.	Folgewirkung	Gegenmaßn.	Verantwortlich	Auftreten	Eingetretene Folgewirkungen
1	2	Zeitraumen verlängert sich	keine	Gesamtes Personal		
2	1	Qualitätssenkung, Zeitraumen verlängert sich	Backup	Technischer und Dokumentations-Leiter	7.4.2006-2.5.2006	Projektauftrag neu geschrieben
3	1	Schlechte Beurteilung	Rechtzeitige Planung	Projektleiter		
3a	1	Schlechte Beurteilung	Planung und Kontrolle	Projektleiter		
4a	2	Software für User uninteressant	Planung und Design	Technischer		
4b	2	Software für Auftraggeber uninteressant	Planung und Codedesign	Technischer Leiter		
5	3	Auftraggeber unzufrieden	---	Auftraggeber		
6	3	---	---	Auftraggeber		
7	3	Qualitätssenkung	Arbeitsatmosphäre angenehm gestalten	Gesamtes Team		
8	1	Projektabbruch, spez. Ziel kann nicht erreicht	Regelmäßige Weiterbildung des	Gesamtes Team	24.4.2006 -	Weiterbildung in SBC

		werden	Teams			
9	2	Qualitätssenkung , möglicherweise Projektabbruch	Projekt interessant gestalten, Team motivieren	Projektleiter		

Legende:

Risikoarten:	Priorisierung:	Eintrittswahrscheinlichkeit:
1... allgemeines Risiko	1... hoch	1... hoch
2... projektspezifisches Risiko	2... mittel	2... mittel
	3...niedrig	3...niedrig

Komponenten

Lieferkomponenten

- Software
- elektronische Handbuch

Weitere Komponenten

- Dokumentation
- Organisationsdokumente
- technische Dokumente (Diagramme, Testergebnisse)
- Tools: Eclipse, hsqldb, JavaSpaces, evtl. Java-Tools zur GUI Erstellung, diverse Grafikprogramme

Grobe Datenbankbeschreibung

<i>Tabellenname</i>	<i>sicher vorkomme nde Attribute</i>	<i>möglicherweis e vorkommende Attribute</i>	<i>Primärschlüssel</i>	<i>Fremdschlüssel</i>	<i>Beschreibung</i>
LVA	ID, LVA-Nr, UniversitätID, Bezeichnung, URL		ID	UniversitätID (-> Universität.ID) URLID (->URL .ID)	
Datei	ID, Name/Titel, Referenz				

Tabellenname	sicher vorkommende Attribute	möglicherweise vorkommende Attribute	Primärschlüssel	Fremdschlüssel	Beschreibung
Notiz	ID, Titel,Text		ID		
Universität	ID, Name		ID		
URL	ID, Name, URL		ID		
Prüfung	ID, LVA, Note		ID	LVA(- >LVAs.ID)	
Kontakt	ID, Name, Adresse	Geburtsdatum	ID		
Kontakt_EMail	ID, KontaktID, email		ID	KontaktID(- >Kontakt.ID)	
Kontakt_Tel	ID, KontaktID, tel		ID	KontaktID(- >Kontakt.ID)	
Lehrmittel	ID, Name, LMTypID		ID	LMTypID (-> LMTyp.ID)	
LMTyp	ID, Bezeichnung		ID		
Sonstige Tabellen					Tabellen evtl. für Stundenplan, mehrere Benutzer, Prüfer, EMail, Adresse, Telefon, Tabellen für Relationen Tabelle für die Datenbankobjekte, die vom Benutzer freigegeben wurden