

Übungsangabe

184.153 Distributed Systems Engineering

2012S, VU, 2.0h, 3.0EC

© Dipl.-Ing. Dr. techn. Johannes Weidl-Rektenwald 2012

1 Einleitung

Der österreichische Ministerrat hat beschlossen, die Wartelisten für planbare Operationen transparent zu machen. Beachten Sie dazu die folgenden Medienberichte:

PLANBARE OPERATIONEN

Nationalrat: Ja zu transparenten OP-Wartelisten
08. Juli 2011 19:11

Bevorzugungen bei planbaren Operationen sollen der Vergangenheit angehören

Wien - Bevorzugungen für Privatpatienten bei Hüft- oder Augenoperationen sollen der Vergangenheit angehören. Der Nationalrat hat Freitagabend einstimmig Gesundheitsminister Alois Stöger dazu ermächtigt, ein transparentes Wartelisten-System im Gesundheitswesen zu etablieren.

Das Wartelistenregime für planbare Eingriffe soll in anonymisierter Form vorliegen und für Operationen etwa in der Augenheilkunde, der Orthopädie, orthopädischen Chirurgie sowie Neurochirurgie gelten. Die für den Eingriff vorgemerkte Person ist auf ihr Verlangen über die gegebene Wartezeit zu informieren.

Quelle: <http://derstandard.at/1308680805456/Planbare-Operationen-Nationalrat-Ja-zu-transparenten-OP-Wartelisten>

CHIRURGISCHE EINGRIFFE

OP-Wartelisten bleiben in Länderhand
18. Mai 2011 18:23

Wie die vom Ministerrat beschlossene Datenbank für planbare Operationen in der Praxis aussehen soll, obliegt den Ländern. Auch private Krankenhäuser müssen ab Frühjahr 2012 ihre Wartelisten transparent machen

Wien - Dass es künftig Datenbanken geben soll, mittels derer Wartelisten für Operationen transparent gemacht werden, darauf hat man sich im Ministerrat bereits geeinigt. Wie diese genau aussehen - und wie wirksam sie daher sind - liegt aber im Einflussbereich der Länder. Denn der Gesetzesentwurf von Gesundheitsminister Alois Stöger ist nur eine Rahmenvorgabe, die Ausführung obliegt den Landespolitikern. Grundsätzlich sollen Spitäler ab dem Frühjahr 2012 Online-Wartelisten führen

müssen, um zu verhindern, dass Privatpatienten bei planbaren Operationen Kassenpatienten bei der Terminvergabe "überholen" können.

Quelle: <http://derstandard.at/1304552267364/Chirurgische-Eingriffe-OP-Wartelisten-bleiben-in-Laenderhand>

Im Rahmen dieser Übung werden Sie ein System für die Online-Wartelistenverwaltung planbarer Operationen umsetzen. Nachdem man aus der Einführung des Spritpreisrechners (spritpreise.at) im Jahr 2011 gelernt hat (siehe z.B. hier: <http://derstandard.at/1313024769721/Nach-Problemen-Spritpreisrechner-stottert-auch-beim-zweiten-Anlauf>), soll das System auf elastische Skalierbarkeit ausgelegt werden, um dynamisch auf Lastspitzen reagieren zu können. Für die Übung ist die Platform-as-a-Service Cloud Umgebung „CloudFoundry“ (cloudfoundry.com) zu verwenden. Cloud Foundry ist zurzeit ohne Kosten per Registrierung zugänglich.

Um die elastische Skalierbarkeit optimal zu unterstützen, wird das Service in mehrere Komponenten aufgeteilt. Dadurch ergibt sich ein verteiltes System, wodurch grundlegende Probleme verteilter Systeme im Design und in der Umsetzung gelöst werden müssen.

Die fachlichen Anforderungen stellen sich wie folgt dar – Akteure werden unterstrichen dargestellt:

- 1) Die interessierte Öffentlichkeit kann jederzeit Einsicht in die OP-Wartelisten der einzelnen Krankenhäuser nehmen
 - a. Dabei wird eine Liste von angebotenen OP Slots angezeigt und eine etwaige Reservierung durch einen Patienten
 - i. Aufgrund des Datenschutzes wird der Name des Patienten, dem dieser Slot zugeteilt wurde, nicht angezeigt
 - ii. Die Liste ist nach Datum sortiert - der zeitlich nächste OP-Slot wird ganz oben angezeigt
 - iii. Optional: Begonnene OP Termine werden dynamisch aus der Liste entfernt (klarerweise ohne Refresh der Seite)
 - iv. Die Liste kann nach Datum, Uhrzeit von, Uhrzeit bis, Krankenhaus, Arzt, OP Typ und Status eingeschränkt werden
 - b. Die Liste kann auf einzelne Krankenhäuser eingeschränkt werden
 - c. Die Liste kann auf einzelne OP Typen eingeschränkt werden
- 2) Patienten können
 - a. Ihre persönliche Reservierungsliste von OP Slots anzeigen lassen
 - i. Die Liste ist nach Datum sortiert - der zeitlich nächste OP-Slot wird ganz oben angezeigt
 - ii. Die Liste kann nach Datum, Uhrzeit von, Uhrzeit bis, Krankenhaus, Arzt, OP Typ und Status eingeschränkt werden
 - b. Ihre persönlichen Notifications anzeigen lassen
 - i. Durch die vollständige asynchrone Arbeitsweise des Systems bekommt der Patient eine Notification, wenn eine vom Arzt durchgeführte Reservierung eines OP Slots für einen bestimmten OP Typ für diesen Patienten erfolgreich war bzw. eine bestehende Reservierung erfolgreich wieder gelöscht wurde

- ii. Optional: Die Notifications werden in einem Frame in Echtzeit angezeigt
- 3) Ärzte (Operateure) können
 - a. Ihre persönliche OP-Slot Liste als Operateur anzeigen lassen. Dabei kann es Slots geben, die bereits reserviert sind und noch freie angebotene Slots. Der Unterschied ist entsprechend zu visualisieren.
 - i. Die Liste ist nach Datum sortiert - der zeitlich nächste OP-Slot wird ganz oben angezeigt
 - ii. Die Liste kann nach Datum, Uhrzeit von, Uhrzeit bis, Krankenhaus, Arzt, OP Typ und Status eingeschränkt werden
 - b. Eine Terminreservierung für einen Patienten für einen passenden OP Slot ihres Krankenhauses vornehmen
 - c. Eine so vorgenommene Terminreservierung wieder stornieren
 - d. Ihre persönlichen Notifications anzeigen lassen
 - i. Durch die vollständige asynchrone Arbeitsweise des Systems bekommt der Arzt eine Notification, wenn eine von ihm durchgeführte Reservierung eines OP Slots für einen bestimmten Patienten erfolgreich war bzw. die Löschung einer bestehenden Reservierung erfolgreich war
 - ii. Optional: Die Notifications werden in einem Frame in Echtzeit angezeigt
- 4) Krankenhäuser können
 - a. Ihre OP-Slot Liste anzeigen lassen
 - i. Die Liste ist nach Datum sortiert - der zeitlich nächste OP-Slot wird ganz oben angezeigt
 - ii. Die Liste kann nach Datum, Uhrzeit von, Uhrzeit bis, Krankenhaus, Arzt, OP Typ und Status eingeschränkt werden
 - b. OP-Slots für geplante OPs eintragen
 - c. OP-Slots für geplante OPs wieder löschen (sofern sie noch nicht reserviert sind)
 - d. Ihre persönlichen Notifications anzeigen lassen
 - i. Durch die vollständige asynchrone Arbeitsweise des Systems bekommt ein Krankenhaus eine Notification, wenn eine für dieses Krankenhaus durchgeführte Reservierung eines OP Slots für einen bestimmten Patienten erfolgreich war bzw. die Löschung einer bestehenden Reservierung erfolgreich war
 - ii. Optional: Die Notifications werden in einem Frame in Echtzeit angezeigt
- 5) Der Administrator kann
 - a. sich alle OP Slot Listen inklusive der Reservierungen und der Namen der Ärzte und Patienten anzeigen lassen.
 - i. Listen sind nach Datum sortiert - der zeitlich nächste OP-Slot wird ganz oben angezeigt
 - ii. Die Liste kann nach Datum, Uhrzeit von, Uhrzeit bis, Krankenhaus, Arzt, OP Typ und Status eingeschränkt werden

OP Slot Listen GUI Mockup:

Anzeige für Alle, Patient, Arzt, Krankenhaus

Datum	von	bis	Typ	KH	Arzt	Status
12.03.2012	12:00	13:30	Augen	AKH	Dr. Aufmesser	reserviert
13.03.2012	06:00	08:00	Ortho	AKH	Dr. Gott	frei
...						

Anzeige für Administrator:

Datum	von	bis	Typ	KH	Arzt	Patient
12.03.2012	12:00	13:30	Kardio	AKH	Dr. Aufmesser	Ben Bäcker
...						

Anforderungen zur Sicherheit:

Optional: Patienten, Ärzte und Krankenhäuser benötigen für Ihre Spezialaktionen einen Account (Username + Passwort).

Optional: Verwendung von HTTPS zur Sicherung des Übertragungskanal.

Anforderungen zu den Schnittstellen:

Alle Aktionen sind sowohl über ein Web GUI als auch via REST anzubieten.

Optional: Das REST Interface muss ebenso abgesichert sein wie das Web GUI.

Grundlegende Lösungs-Architektur:

Es sind drei Komponenten zu entwickeln:

- 1) Frontend
 - a. Diese Komponente stellt das Web GUI und das REST Interface zur Verfügung. Sie speichert Daten in der Datenbank bzw. liest Daten aus und gibt sie an den User zurück
 - i. Optional: Verwendung des CQRS Patterns
 - b. Reservierungsanfragen werden an den Matcher weitergeleitet, der diese asynchron abarbeitet
- 2) Matcher
 - a. Der Matcher nimmt Reservierungsanfragen entgegen und versucht, den zeitlich und geografisch (zum Patienten) nächsten OP Slot im eingegebenen Umkreis [in km] innerhalb des gewünschten Terminranges zu reservieren
 - i. Dazu müssen Patienten und Krankenhäuser mit Geokoordinaten ausgestattet werden, um eine Geo-Suche durchführen zu können
 - ii. Hinweis: MongoDB unterstützt Geo-Suchen nativ!

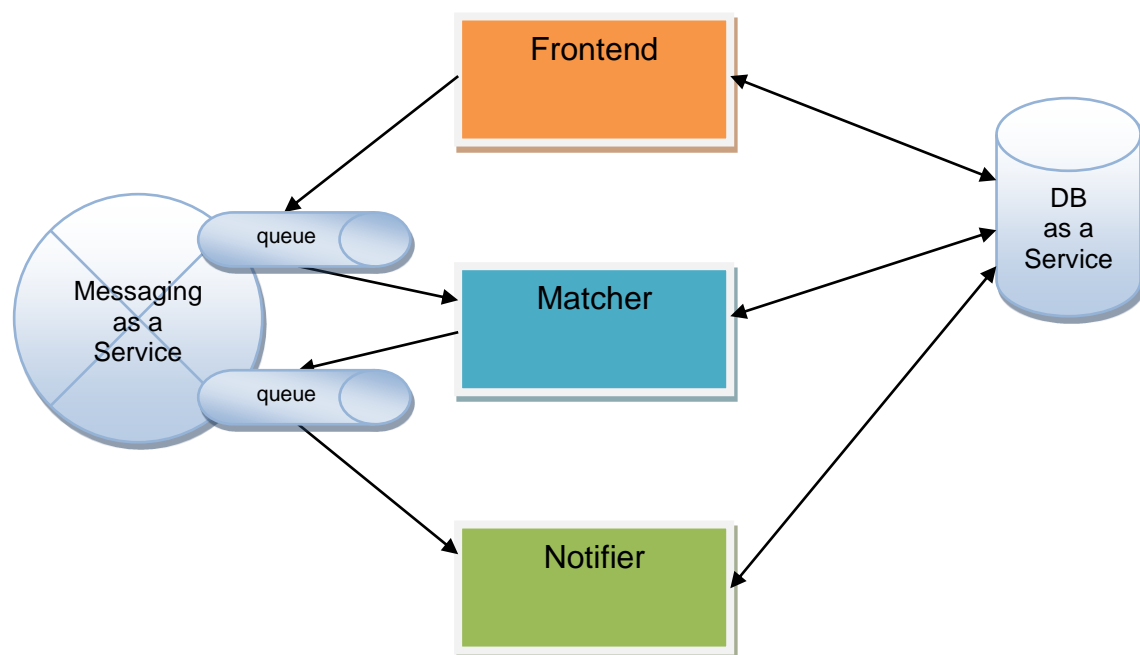
- iii. Im positiven Fall erfolgt eine Reservierung und die entsprechenden Notification-Requests werden an den Notifier weitergeleitet
- iv. Im negativen Fall wird keine Reservierung durchgeführt und die entsprechenden Notifications-Requests werden an den Notifier weitergeleitet
- v. Optional: Wurde kein OP Slot gefunden, kann inkrementell der Umkreis und die Terminrange für die Suche ausgedehnt werden, um Treffer zu erhalten.

3) Notifier

- a. Der Notifier nimmt Notification-Requests entgegen und speichert sie in der Datenbank für die weitere Verarbeitung
- b. Optional: Der Notifier ist als eigene Komponente ausgelegt, um komplexere Notifications – z.B. per Smartphone Push Notifications – asynchron abarbeiten zu können.

Eine Auslagerung des Domain Models in eine 4. Komponente kann zweckmäßig sein.

Architekturskizze:



Anforderungen zum Deployment in CloudFoundry:

- Die Komponenten laufen aus Fehlertoleranz-Gründen mindestens doppelt auf verschiedenen Rechnerknoten.
- Durch mehrfaches Deployment einzelner Komponenten kann bedarfsgetrieben skaliert werden.
- Die Komponenten arbeiten und kommunizieren vollständig asynchron über ein Messaging Service.

- Arbeitsaufgaben (vom Frontend zum Matcher und vom Matcher zum Notifier) müssen persistent abgelegt werden, d.h. einen Ausfall des Messaging Servers überleben.
- Der Ausfall einer Komponente darf die Funktionsweise der anderen Komponenten nicht beeinträchtigen.

Anforderungen zur Entwicklungsumgebung:

Verpflichtend zu verwenden sind:

- 1) CloudFoundry als Laufzeitumgebung
 - a. Jede Gruppe erstellt einen CloudFoundry Account
- 2) Apache Maven
- 3) JUnit

Die Verwendung wird angeraten:

- 1) Spring Framework (3.1)
- 2) Spring ROO (1.2.1) zum Rapid Development der Komponenten
 - a. Tutorial zu Spring ROO und MongoDB:
 - b. <http://static.springsource.org/spring-roo/reference/html/base-layers.html>
- 3) MongoDB (auch Redis und PostgreSQL sind in CloudFoundry vorhanden) mit eingebauter Geo-Suche
- 4) Spring Data MongoDB zum Zugriff auf die MongoDB
- 5) RabbitMQ (einzige Messaging Alternative in CloudFoundry) als MOM
- 6) Spring Tool Suite (STS) mit CloudFoundry Plugin (alternativ: VMC Konsole für CloudFoundry)
 - a. <http://start.cloudfoundry.com/getting-started.html>

Die Verwendung ist optional, hilft aber Lines Of Code einzusparen bzw. die Testbarkeit zu verbessern.

- 1) Lombok
- 2) Mockito

Folgende Daten sind für Tests automatisiert zu erzeugen:

Patienten:

1. Hr. Franz Meier
2. Fr. Maria Müller
3. Hr. Martin Moser
4. Fr. Beate Bauer
5. Hr. Ben Bäcker
6. Fr. Gloria Glaser

Ärzte:

1. Dr. Hochweiß

2. Dr. Gott
3. Dr. Aufmesser
4. Dr. Augenschein

Krankenhäuser:

1. AKH Wien
2. Barmherziger Brüder
3. LKH Klosterneuburg
4. LKH Tulln

OP Typen:

1. Augenheilkunde
2. Orthopädie
3. Orthopädische Chirurgie
4. Neurochirurgie
5. Kardiologie

OP Slot Längen:

1. 60 Minuten
2. 120 Minuten
3. 180 Minuten

Optional: Erstellung von umfangreicheren Testdaten

Tests:

Die Testabdeckung über Unit Tests soll ca. 50% betragen.

Schreiben Sie zumindest einen Lasttest unter Verwendung der „rest-assured“ Library (<http://code.google.com/p/rest-assured/>).

Optional: Automatisierte GUI Tests mit z.B. Selenium können helfen, die Testabdeckung zu verbessern.

2 Deliverables

2.1 *Projektplan*

Erstellen Sie ein Dokument mit folgendem Inhalt:

Deliverable 1: Dokument: Projektplan

- Enthält eine kurze Projektbeschreibung mit den Projektzielen und Nicht-Zielen (Projektzieleplan)
- Enthält die Rollenvergabe im Projektteam (Projektorganigramm)
 - Wer ist der Projektleiter etc.?
- Enthält die Arbeitspaketspezifikation oder Work Breakdown Structure (WBS)

- Welchen Aufwand planen Sie für die Tätigkeiten ein (Aufwandsschätzung)?
 - Wie hoch ist der Gesamtaufwand für jedes Projektteammitglied?
 - Wie hoch ist der Gesamtaufwand?
- Enthält die Ressourcenplanung und Milestoneplanung
 - Wer im Team macht was und bis wann?
 - Festlegung der Fertigstellungstermine für die Arbeitspakete und Deliverables
- Welche Methoden und Tools werden Sie verwenden und wozu (Technische Planung)?
- Ein Projektbalkenplan kann optional erstellt werden.

Beachten Sie die zeitgerechte Abgabe des Projektplans (siehe „Übungsablauf und Organisation“ unten)!

2.2 Architektur- und Designdokument

Erstellen Sie ein Dokument mit dem folgenden Inhalt:

Deliverable 2: Dokument: Architektur- und Designdokument

- Beschreibung der Software Architektur
 - Welche Komponenten gibt es?
 - Welche Schnittstellen gibt es?
- Stellen Sie den „Logical View“ und den „Deployment View“ wie im „4+1 Architectural Model“ von Philippe Kruchten dar (siehe dazu auch [http://en.wikipedia.org/wiki/4%2B1 Architectural View Model](http://en.wikipedia.org/wiki/4%2B1_Architectural_View_Model))

2.3 Wartelistenimplementierung für planbare OPs

Erstellen Sie das in Übungsangabe beschriebene Verteilte System.

Deliverable 3: Codeabgabe: Wartelistenimplementierung für planbare OPs

2.4 Wartungshandbuch

Erstellen Sie ein Dokument mit dem folgenden Inhalt:

Deliverable 4: Dokument: Wartungshandbuch
--

- Beschreibung der Entwicklungsumgebung
- Beschreibung der eingesetzten Frameworks
- Beschreibung des Build Prozesses
- Beschreibung der Unit, GUI- und Lasttests

2.5 Projektabschluss

Jedes Teammitglied erstellt einen Projektabschlussbericht. Der Name des Teammitglieds muss aus dem Titel des Dokuments klar erkennbar sein!

Deliverable 5: Lessons Learned and Conclusio

- Beurteilen Sie den Ablauf des Projektes
 - Was hat gut funktioniert, was weniger gut. Begründen Sie ihre Beurteilung.
 - Haben Sie den ursprünglich geschätzten Aufwand überschritten? Um wieviel und warum?
 - Was würden Sie anders bzw. besser machen, wenn Sie mit ihrer jetzigen Erfahrung das Projekt nochmals machen müssten.
- Beurteilen Sie die finale Software
 - Was ist gut geglückt, welche Teile sind aus ihrer Sicht gar nicht oder nur mangelhaft umgesetzt worden.
- Verfassen Sie ihr persönliches Fazit.

3 Übungsablauf und Organisation

- Über die Gruppenanmeldung im TISS System werden Übungsgruppen zu je drei (3) Personen gebildet (<http://tiss.tuwien.ac.at>).
- Laden Sie den Projektplan (Deliverable Dokument 1.1) **bis spätestens Freitag, 30.03.2012, 24:00** im TUWEL hoch.
 - **Alle** Dokumente müssen in einem der folgenden Formate abgegeben werden: .rtf, .doc, .pdf oder .html! Im speziellen ist für den Projektplan kein Microsoft Project® oder anderes Projektmanagementtool-spezifisches Format erlaubt! Achten Sie darauf, dass keine Links auf externe Objekte (Bilder, Diagramme) in den Dokumenten vorhanden sind. Achten Sie darauf, dass die Dokumente nicht exorbitant groß sind (z.B. durch Einbinden großer Bilder).
 - **Alle** abgegebenen Dokumente müssen als Dateinamen bzw. Dokumenttitel die Dokumentnummer und den Dokumenttitel (wie in der Aufgabenstellung spezifiziert) tragen! Der Titel muss auf der ersten Seite sowie über Fußzeilen im ganzen Dokument ersichtlich sein.
 - Der Dateiname für die Projektplanabgabe als pdf ist also „DSE12Gruppe<nn>_1_1_Projektplan.pdf“
 - <nn> entspricht ihrer Gruppennummer formatiert auf zwei Stellen
 - Der Dateiname der Projektplan Datei in der Gesamtabgabe ist also „1_1_Projektplan.pdf“
 - Falls Sie z.B. HTML als Format wählen und das Dokument mehrere HTML Dateien umfasst, packen Sie die Dateien in ein ZIP Archiv. Achten Sie darauf, dass das Dokument direkt aus dem ZIP Archiv durchgebrowst werden kann, d.h. das ZIP Archiv nicht auf die Festplatte entpackt werden muss.
 - Der Dateiname für die Projektplanabgabe als zip Archiv ist also „DSE12Gruppe<nn>_1_1_Projektplan.zip“
- Buchen Sie pro Gruppe im TUWEL einen Präsentationstermin **bis spätestens Freitag, 04.05.2012, 24:00**.

- Laden Sie die Gesamtabgabe aller Deliverables im TUWEL hoch **bis spätestens Sonntag, 15.06.2012, 24:00**.
 - Alle Files der Abgabe müssen in einer sinnvollen Verzeichnisstruktur abgelegt werden, die der Nummerierung der Deliverables entspricht
 - Legen Sie der Abgabe unbedingt auch wieder Deliverable 1.1 (Projektplan) bei.
 - Der Dateiname der Abgabe muss **DSE12AbgabeGruppe<nn>.zip** lauten
- Ablauf der Präsentation
 - Zur Präsentation ist die Anwesenheit **aller** Gruppenmitglieder erforderlich.
 - Sie präsentieren Ihre Lösung mittels
 - Kurzüberblick
 - Code Walkthrough
 - Online Demo
 - Build Durchlauf mit erfolgreicher Ausführung aller Tests
 - Jedes Gruppenmitglied muss die von ihm erarbeiteten Teile präsentieren
 - Jedes Gruppenmitglied muss das Gesamtprojekt - d.h. auch die Teile, die die anderen Gruppenmitglieder präsentieren – kennen, vollständig verstehen und auch Fragen dazu beantworten können.
 - Wir behalten uns vor, Gruppenmitgliedern weniger Punkte zu vergeben wenn der Eindruck entsteht, dass diese nicht adäquat am Gesamtumfang der Übung mitgearbeitet haben.

4 Übungsinhalte

- Projektarbeit/Projektplanung
- Architektur- und Designentwurf eines verteilten Systems im Kontext von PaaS Cloud Computing
- Implementierung
- Test
- Deployment
- Erstellung eines Wartungshandbuch
- Gruppenkommunikation- und Zusammenarbeit, Präsentationstechnik

5 Generelle Anmerkungen und Tipps

- **Studieren Sie die Angabe genau!** Es werden immer wieder Deliverables vergessen bzw. Themen verfehlt. Für eine gute Note müssen alle Aufgaben erfüllt und alle Deliverables bereitgestellt werden. Wir behalten uns vor, formal nicht entsprechende Deliverables nicht zu akzeptieren.
- Teilen Sie Rollen bzw. Zuständigkeiten innerhalb der Gruppe ein (für z.B. Projektleitung, Koordination, Technik, Tools, Dokumentation). Planen Sie Projektmeetings auch während der Laufzeit, z.B. zu bestimmten Milestones. Heben Sie nicht alles für den Schluss auf.

- Bei der Präsentation muss man über das Gesamtprojekt informiert sein – nicht nur über seinen eigenen Teil!
- Wenn Sie bei der Lösung aufgrund von unvollständigen Angaben bzw. Informationen *Annahmen* treffen, dokumentieren Sie diese unbedingt! Annahmen dürfen in keinem Fall die Lösung trivialisieren.
- Plagiate: Bitte kopieren Sie keine Lösungen anderer Gruppen oder Lösungen, die im Internet angeboten werden. Wenn wir Plagiate entdecken, vergeben wir 0 Punkte und Sie können die Übung nicht positiv absolvieren. Natürlich ist es erlaubt, Problemstellungen und mögliche Lösungen mit Ihren Kollegen zu diskutieren, aber die abgegebene Lösung muss zu 100% von Ihnen erstellt worden und einzigartig („unique“) sein. Auch wenn Gruppen gemeinsam arbeiten, dürfen die Lösungen zueinander nicht zu ähnlich sein.
- Deadline Verschiebungen: Deadline Verschiebungen werden nur in speziellen Situationen zugestanden, nachdem eine individuelle Vereinbarung mit der Kursleitung getroffen wurde. Wenn Sie Ihr Beispiel nicht rechtzeitig beenden können, geben sie Ihr Beispiel unvollendet ab – das ist immer noch besser, als gar nichts abzugeben. Achten Sie darauf, dass bei der Abgabe immer etwas schief gehen kann und planen Sie das entsprechend ein. Geben Sie ihre Lösung rechtzeitig ab und beachten Sie, dass kurz vor der Deadline die Server überlastet sein können. Wir können keine Email Abgaben oder andere Uploads akzeptieren. Wenn Sie Probleme mit der Abgabe haben, senden sie einen Hash (z.B. MD5) Ihrer Abgabe an die Kursleitung rechtzeitig vor der Abgabe-Deadline. Mittels dieses Hashes können wir überprüfen, ob die Abgabe nach der Deadline modifiziert worden ist.

6 Anhang

Dokumenteninhalt:

1	Einleitung	1
2	Deliverables	7
2.1	Projektplan	7
2.2	Architektur- und Designdokument	8
2.3	Wartelistenimplementierung für planbare OPs.....	8
2.4	Wartungshandbuch	8
2.5	Projektabschluss.....	8
3	Übungsablauf und Organisation	9
4	Übungsinhalte	10
5	Generelle Anmerkungen und Tipps	10
6	Anhang.....	11

Auflistung der Deliverables, Unterverzeichnisstruktur der Abgabe und Grading-Informationen:

Deliverable 1: Dokument: Projektplan

5 Punkte – Abzüge bei Unvollständigkeit bzw. fehlender Plausibilität

Deliverable 2: Dokument: Architektur- und Designdokument

5 Punkte – Abzüge bei Unvollständigkeit bzw. nicht nachvollziehbaren Designentscheidungen, die im Gegensatz zu dem in der Vorlesung vermittelten state-of-the-art stehen.

Deliverable 3: Codeabgabe: Wartelistenimplementierung für planbare OPs

45 Punkte – Abzüge bei Unvollständigkeit: fehlendes Javadoc, fehlende Inline Kommentare, unvollständiges Deployment, fehlende Funktionalität, fehlende asynchrone Kommunikation, fehlende Unabhängigkeit der Komponenten, nicht funktionierendes User Interface, unerlaubte Trivialisierung, fehlende Tests, etc.

Deliverable 4: Dokument: Wartungshandbuch

3 Punkte – Abzüge bei Unvollständigkeit

Deliverable 5: Dokument: Lessons Learned und Conclusio

2 Punkte – Abzüge bei Unvollständigkeit

Höchstpunktezahl: **60 Punkte**

Mindestpunktezahl: **30 Punkte**

Für die Umsetzung von optionalen Anforderungen (gekennzeichnet mit dem Wort „Optional:“) werden keine Punkte vergeben. Es können damit auch keine Punkte für Optionen gegen nicht erreichte Punkte nicht-optionaler Teile „eingetauscht“ werden.

Dokumentenlenkung:

V1.0 Version vom 01.03.2012

Copyright:

Diese Beispielangabe darf weder vollständig noch in Auszügen ohne Zustimmung des Autors verwendet werden. Bei Interesse bitte um Kontaktaufnahme unter jwr--AT--infosys.tuwien.ac.at

Kontaktadresse für die Übung:

dse--AT--infosys.tuwien.ac.at