



Software Engineering Übung 2

Anforderungsspezifikation und GWT Tutorien

1 Informationen

1.1 Daten

- Ausgabe Di 01.10.2013
- Abgabe Mo 14.10.2013 bis 23:59 Uhr
- Besprechung am Di 22.10.2013 um 12:15 Uhr

1.2 Formales

Die Lösungen sollen als PDF Datei mit dem Namen Ex[n] [NameA Matrikelnummer].pdf abgegeben werden, wobei [n] die Nummer der Übung ist und [NameA Matrikelnummer] der Name und Matrikelnummer des Studenten sind. Die PDF Datei sollte außerdem ebenfalls Ihren Namen und Matrikelnummer beinhalten.

Bitte senden Sie Ihre Lösungen via OLAT. Der Betreff der Lösung sollte mit [SE EX HS13] beginnen. Falls Sie zusätzliche Abgabematerialien (z.B. Source Code) haben, senden Sie diese als Archiv (.zip-File), welches alle Dateien, einschliesslich dem PDF, enthält. Benennen Sie das Archiv anhand der oben erwähnten Konventionen.

Der Student muss über alle Teile der Lösungen Auskunft geben können. Verspätete Abgaben werden korrigiert, aber nicht bewertet.

2 Überblick und Ziele

Mit dieser Übung beginnt das Übungsprojekt, bei dem Sie in Gruppen eine kleine Applikation entwickeln. Übung 2 besteht aus zwei Teilen:

Im ersten Teil (16 Punkte) beschäftigen Sie sich mit der Anforderungsspezifikation für die Applikation. Diesen Teil müssen Sie gemeinsam in Ihrer *Übungsgruppe* bearbeiten.

Im zweiten Teil (4 Punkte) lernen Sie den Umgang mit dem Google Web Toolkit, welches Sie als Basis für die Implementierung Ihrer Applikation verwenden werden. Diesen Übungsteil müssen Sie *individuell* bearbeiten.

3 Projektbeschreibung

Ihr Auftraggeber (gespielt durch den Übungsleiter) will eine Applikation zur Visualisierung von öffentlich zugänglichen statistischen Daten der Schweiz entwickeln lassen. Als Pilot soll ein Datensatz über die Ergebnisse von eidgenössischen Abstimmungen (Volksentscheide auf Bundesebene) visualisiert werden. Die Daten finden Sie auf der Webseite über Abstimmungen des Bundesamts für Statistik: <http://www.bfs.admin.ch/bfs/portal/de/index/themen/17/03.html>

Kernanforderungen an die Applikation sind:

- Die Applikation soll die Daten des ausgewählten Datensatzes interaktiv visualisieren.
- Die Applikation soll die Daten wahlweise tabellarisch oder geographisch visualisieren.
- Der Benutzer soll aus einer Liste möglicher Auswertungen auswählen können.
- In geographischen Visualisierungen soll der Benutzer aus Übersichtsansichten in Detailansichten hineinzoomen können.
- Der Benutzer soll zu einer ausgewählten Visualisierung Kommentare und Bilder hinzufügen können.
- Der Benutzer soll Visualisierungen speichern und mit anderen Benutzern teilen können.

Ihr Auftraggeber hat Ihnen in einem Vorgespräch folgende Informationen gegeben:

„Unsere Vision ist, eine Web Applikation bereit zu stellen, welche kostenlos benutzt werden kann und für verschiedene Zielgruppen interessant ist. Für Stimmbürger wollen wir durch coole Visualisierungen und die Möglichkeit, Visualisierungen mit anderen zu teilen attraktiv sein. Journalisten ermöglichen wir eine schnelle und einfache Erstellung von Visualisierungen. Politikern und Parteien soll die Applikation als Analyseinstrument und in Zukunft auch als Werbeplattform dienen. Mit der Möglichkeit, die Applikation für Wahl- und Abstimmungswerbung zu nutzen, wollen wir mittelfristig mit der Applikation Geld verdienen.“

„Welche Auswertungen wir genau anbieten wollen, wissen wir noch nicht genau, hier sind wir für Vorschläge seitens der Entwickler offen. In einem ersten Schritt könnten wir uns vorstellen, eine Auswertung nach Siedlungsstruktur (Zentrumsstädte, Agglomerationsgemeinden, übrige Städte, ländliche Gemeinden) anzubieten. Das scheint einem verbreiteten Bedürfnis zu entsprechen.“

Ihr Auftraggeber hat ferner während der letzten Parlamentssession im Bundeshaus verschiedene Leute befragt, was sie von der Idee dieser App halten. Hier sind die erhaltenen Antworten:

„Das ist eine coole Idee. Ich fände beispielsweise eine Visualisierung der Parteiparolen zu einer Abstimmung zusammen mit dem Abstimmungsergebnis hilfreich, vielleicht sogar nicht nur für eine Abstimmung, sondern für alle Abstimmungen eines Jahres.“ (Peter N., Nationalrat)

„Wir würden uns vor allem eine Visualisierung der Stadt-Agгло-Land Stimmenverhältnisse bei einer Abstimmung wünschen, gegebenenfalls kombiniert mit Ergebnissen nach Sprachregionen. Wir würden eine solche Applikation sowohl für uns selbst nutzen, als auch die Leser unserer Zeitungen auf diese Möglichkeit hinweisen. Mit Werbung auf der App könnten wir leben, sofern sie sauber von den Inhalten getrennt dargestellt wird.“ (Nadine Q., Heiri K. und Domenico A., Bundeshaus-Journalisten)

„Das mit der Stadt-Agгло-Land-Visualisierung sehe ich genauso wie Nadine Q. Solche Visualisierungen muss ich heute selbst aus den BfS-Daten extrahieren, während einfache Visualisierungen wie Ja/Nein-Stimmen nach Kantonen überall leicht erhältlich sind. Damit auch einfache Stimmbürgerinnen und Stimmbürger die App benutzen, müsste sie allerdings leicht und intuitiv zu benutzen sein.“ (Max M., Journalist und Politologe)

„Hmmm, eine solche App müsste schon sehr attraktiv aufgemacht und sehr leicht zu benutzen sein. Sonst sehe ich keinen Mehrwert gegenüber einer Nutzung der Originaldaten über die Webseiten des Bundesamts für Statistik. Die App-Entwickler müssen sich ferner darüber klar sein, dass das BfS die Präsentation der statistischen Daten auf seinen Webseiten jederzeit ändern kann.“ (Sandra S., Mitarbeiterin Bundesamt für Statistik)

„Werbung? Auf einer solchen Plattform? Wäre für mich ein Grund, diese App zu meiden.“ (Gian R., TV-Journalist)

„Bei einer werbefinanzierten App sehe ich die Gefahr, dass die Darstellung nicht mehr neutral ist.“ (Fabio G., Parteimitglied der G-Partei)

„Die Möglichkeit, auf einer solchen App Wahl- und Abstimmungswerbung zu schalten, würden wir höchstwahrscheinlich nutzen.“ (Markus E., Parteisekretär V-Partei und Nationalrat)

„Tönt spannend. Habe eben gehört, was die Journalistengruppe über eine Auswertung nach Siedlungsstrukturen gesagt hat. Ich mache solche Auswertungen seit einer Weile von Hand, und ich sage Ihnen, das ist sehr aufschlussreich. Sollte man auf jeden Fall in einer solchen App drin haben. Ich würde mir allerdings darüber hinaus einen Visualisierungsmodus wünschen, wo man die Ergebnisse von mehreren Abstimmungen nach bestimmten Kriterien, eben zum Beispiel der Siedlungsstruktur, visualisieren kann“ (Sergio C., politikinteressierter Bürger)

4 Aufgabenstellung Teil 1: Anforderungsspezifikation

Teilaufgabe 2.1: Analyse der Interesseneigner (2 Punkte)

Führen Sie eine Analyse der Interesseneigner (Stakeholder) für die Applikation durch. Erstellen Sie eine Liste der Interesseneigner-Rollen. Begründen Sie Ihre Entscheidungen kurz und geben Sie für jede Rolle an, für wie wichtig Sie diese Rolle halten (Skala: kritisch-wichtig-unwichtig).

Teilaufgabe 2.2: Kontext (2 Punkte)

Um die Systemgrenze besser verstehen und mit dem Auftraggeber diskutieren zu können, erstellen Sie auf der Basis der vorliegenden Informationen ein Kontextdiagramm.

Teilaufgabe 2.3: Anforderungsermittlung (2 Punkte)

Am Dienstag 8.10. steht Ihnen der Auftraggeber (in der Übungsstunde nach der Vorlesung) für ein Interview zur Verfügung. Schreiben Sie Ihre Interviewfragen auf. Geben Sie zu jeder Frage in Stichworten eine Begründung, warum Sie diese Frage stellen. Damit sich der Auftraggeber auf das Interview vorbereiten kann, geben Sie Ihre Fragen bis Freitag 5.10. auf dem dafür eingerichteten Forum in OLAT ein.

Teilaufgabe 2.4: Anwendungsfälle (4 Punkte)

Spezifizieren Sie die Anwendungsfälle „Auswertung auswählen“, „Tabellarisch visualisieren“ und „Graphisch visualisieren“. Die letzten beiden Anwendungsfälle verwenden («include») die Sub-Anwendungsfälle „Kommentar oder Bild hinzufügen“, „Visualisierung speichern“, „Visualisierung mit Freunden teilen“ und „Visualisierung exportieren“. Spezifizieren Sie diese vier Sub-Anwendungsfälle. Zeichnen Sie ein Anwendungsfall-Diagramm.

Für die Spezifikation eines Anwendungsfall ist folgendes Schema zu verwenden:

Name: <Name des Anwendungsfalls>

Akteur(e): <Hauptakteur, ggf. weitere beteiligte Akteure>

Ausgelöst durch: <Typisch ein vom Hauptakteur ausgelöstes Ereignis>

Voraussetzung: <Was erfüllt sein muss, damit der Anwendungsfall ausgeführt werden darf>

Resultat: <Hauptresultat, ggf. Alternativresultate>

Normalablauf: <In Schritte gegliederter, nummerierter Ablauf>

Alternative Abläufe: <mögliche Alternativen im Ablauf*>

*Die Zuordnung zum Normalablauf erfolgt über die Nummerierung: beispielsweise tragen zwei Alternativen zum Normalschritt 3 die Nummern 3a und 3b.

Teilaufgabe 2.5: Anforderungsspezifikation erstellen (6 Punkte)

Erstellen Sie eine Software-Anforderungsspezifikation für die Applikation gemäss der nachstehenden Gliederungsvorgabe. Zu jeder Einzelanforderung sind folgende Metadaten anzugeben: eindeutiger ID, Quelle (woher stammt die Anforderung), Autor, Erstellungsdatum.

Teil I: Überblick

1 Anlass und Ziele

[Identifizieren Sie in den Ihnen zur Verfügung stehenden Informationen den Projektanlass und die Ziele für die Applikation. Stellen Sie beides in diesem Kapitel dar.

2 Interesseneigner

- [Hier fügen Sie die in Teilaufgabe 4.1 erstellte Liste der Interesseneigner ein]
- 3 **Umfang und Kontext**
[Hier fügen Sie das in Teilaufgabe 4.2 erstellte Kontextdiagramm ein]
 - 4 **Kernanforderungen**
[Hier geben Sie eine kurze Übersicht über die wichtigsten Anforderungen. Da es sich um eine kleine Applikation handelt, erfolgt keine Untergliederung in Subsysteme]
 - 5 **Annahmen**
[Dokumentieren Sie hier die Annahmen, welche Sie in den zur Verfügung stehenden Informationen gefunden haben.]

Teil II: Einzelanforderungen

- 6 **Nutzungsszenarien**
 - 6.1 **Anwendungsfälle**
[Hier fügen Sie die von Ihnen erstellten Anwendungsfälle und das Anwendungsfalldiagramm ein.]
 - 6.2 **Benutzergeschichten**
[Dieses Unterkapitel bleibt vorerst leer]
- 7 **Klassenmodell**
[Modellieren Sie die Informationen, welche die Applikation kennen muss um die Anwendungsfälle abzuwickeln, in einem Klassendiagramm]
- 8 **Schnittstellen**
 - 8.1 **Benutzerschnittstelle**
[Dieses Unterkapitel bleibt vorerst leer]
 - 8.2 **Externe Schnittstellen**
[Beschreiben Sie hier Import- und Export-Schnittstellen der Applikation]
- 9 **Qualitätsanforderungen**
[Hier beschreiben Sie die ermittelten Qualitätsanforderungen]
- 10 **Randbedingungen**
[Hier beschreiben Sie die ermittelten Randbedingungen]

Anhang

Glossar

[Geben Sie mindestens sechs Fachbegriffe und/oder Abkürzungen an, welche im Rahmen der Erstellung dieser Applikation verwendet werden, und definieren Sie diese.]

Verzeichnis referenzierter Dokumente

[Falls es mitgeltende Unterlagen gibt, listen Sie diese hier auf.]

Hinweis: Einige der von den befragten Leuten genannten Wünsche und Ideen lassen sich nicht als Anwendungsfälle fassen. Wir werden diese Wünsche später in Form von Benutzergeschichten spezifizieren.

Nach Abgabe der Übung müssen alle Gruppenmitglieder in der Lage sein, den Tutoren die Anforderungsspezifikation Ihrer Gruppe zu erläutern und zu erklären, wo ein bestimmter Benutzerwunsch spezifiziert ist oder warum er noch nicht spezifiziert ist.

5 Aufgabenstellung Teil 2: Google Web Toolkit

Den zweiten Übungsteil bearbeiten Sie nicht als Gruppe, sondern individuell. Er besteht aus der Bearbeitung von zwei Tutorien. Im ersten Tutorial lernen Sie, wie Sie mit Hilfe von GWT (Google Web Toolkit) Web Applikationen bauen können. Am Ende des Tutorials müssen Sie die *StockWatcher* Applikation lauffähig haben und den *Debugger* benutzen um eventuelle *Bugs* zu entfernen. Die Tutoren werden in der Besprechung überprüfen, ob Sie das Tutorial erfolgreich durchgearbeitet haben und werden einfache Fragen zum Code stellen. Das zweite Tutorial behandelt die Google App Engine. Sie lernen dabei die Benutzung der Google Online Infrastruktur um eine Web Applikation online zu hosten. Ziel des Tutorials ist, dass Sie in der Lage sind, die *StockWatcher* Applikation in der PaaS (Platform as a Service) App Engine von Google hosten zu können. Am Ende müssen Sie einen Link zu Ihrer Applikation abgeben. Damit alle abgegebenen Links eindeutig voneinander unterscheidbar sind,

müssen Sie Ihren Namen und Ihre Matrikelnummer in Ihren Link einbauen. Die Tutoren werden überprüfen, ob die *StockWatcher* Web Applikation mit dem von Ihnen abgegebenen Link erreichbar ist und ob die Applikation konform zu den Anforderungen funktioniert.

5.1 Installation und Konfiguration

5.1.1 Installation und Konfiguration von Java unter Windows

Wenn Sie keine aktuelle Version des Java SDK haben dann können Sie es hier herunterladen:

<http://www.oracle.com/technetwork/java/javase/downloads/index.html>

Damit Java auf Ihrem Rechner korrekt läuft, ist es wichtig dass die Systemvariablen richtig gesetzt sind. Bitte navigieren Sie in Windows zu: 1. *My Computer* 2. *Properties* 3. *Advanced System Settings* 4. *Environment Variables*. Unter *System Variables* sollte folgendes Verzeichnis existieren: C:\Programm Files(x86)\Java\jdk1.6.0_21\bin. Wenn das Verzeichnis nicht existiert dann, können Sie es manuell hinzufügen. Anschliessend müssen Sie ihren Rechner neu starten. Nach den Neustart überprüfen Sie, ob Windows die Variable übernommen hat, indem Sie in ein Terminal-Fenster (cmd.exe) den Befehl *java* eingeben. Wenn Sie das in Bild 1 dargestellte Ergebnis sehen, ist Java auf ihren Rechner installiert und lauffähig.

```
C:\>java
Usage: java [-options] class [args...]
           (to execute a class)
   or  java [-options] -jar jarfile [args...]
           (to execute a jar file)

where options include:
-server          to select the "server" VM
-hotspot         is a synonym for the "server" VM [deprecated]
                 The default VM is server.

-cp <class search path of directories and zip/jar files>
-classpath <class search path of directories and zip/jar files>
             A ; separated list of directories, JAR archives,
             and ZIP archives to search for class files.
-D<name>=<value>
             set a system property
-verbose[:class[:gc[:jni]]
             enable verbose output
-version         print product version and exit
-version:<value>
             require the specified version to run
-showversion    print product version and continue
-jre-restrict-search | -jre-no-restrict-search
               include/exclude user private JREs in the version search
```

Bild 1. Kommandozeilenoutput in Windows nach Ausführung des Befehls *java*

5.1.2 Installation und Konfiguration von Java unter Mac OS

Wenn Sie keine aktuelle Version des Java SDK haben dann können Sie es hier herunterladen:

<http://www.oracle.com/technetwork/java/javase/downloads/java-archive-downloads-javase7-521261.html>

Zur Installation von Java unter Mac OS gehen Sie folgendermassen vor:

1. Laden Sie die Datei *jre-7u6-macosx-x64.dmg* herunter. Lesen Sie die Lizenzvereinbarung, und stimmen Sie den Bedingungen zu, bevor Sie die Datei herunterladen.
2. Doppelklicken Sie auf die DMG-Datei, um sie zu starten. Doppelklicken Sie auf das Packagesymbol, um den Installationsassistenten zu starten.
4. Im Installationsassistent wird der Installationsbildschirm „Willkommen bei Java 7“ angezeigt. Klicken Sie auf Weiter.
5. Klicken Sie in „Standardinstallation auf Macintosh HD“ auf die Schaltfläche Installieren.
6. Nach Abschluss der Installation wird ein Bestätigungsbildschirm angezeigt. Klicken Sie auf Schließen, um den Installationsvorgang zu beenden.

Testinstallation: Wenn Sie prüfen möchten, ob Java auf Ihrem Computer installiert ist und problemlos ausgeführt wird, führen Sie das **Test-Applet** aus, welches Sie unter <https://www.java.com/de/download/help/testvm.xml> finden oder geben Sie den Befehl *java* in ein Terminal-Fenster und prüfen, ob Sie ein Ergebnis wie in Bild 2 unten bekommen.

```
localhost:~ muntean$ java
Usage: java [-options] class [args...]
           (to execute a class)
   or java [-options] -jar jarfile [args...]
           (to execute a jar file)

where options include:
-d32          use a 32-bit data model if available
-d64          use a 64-bit data model if available (implies -server, only for x86_64)
-client      to select the "client" VM
-server      to select the "server" VM
-jvm         is a synonym for the "client" VM [deprecated]
-hotspot     is a synonym for the "client" VM [deprecated]
             The default VM is client.

-cp <class search path of directories and zip/jar files>
-classpath <class search path of directories and zip/jar files>
           A : separated list of directories, JAR archives,
           and ZIP archives to search for class files.
-D<name>=<value>
           set a system property
-verbose[:class|gc|jni]
           enable verbose output
-version     print product version and exit
-version:<value>
           require the specified version to run
```

Bild 2. Kommandozeilenoutput in Mac OS X nach Ausführung des Befehls *java*.
(Aus Platzgründen ist hier nicht das ganze Ergebnis aufgelistet)

5.1.3: Installation des Google Web Toolkit

Für das GWT Tutorial müssen Sie folgende Packages installieren:

1. Installieren Sie zunächst die Eclipse IDE.
<http://www.eclipse.org/downloads/>
2. Installieren Sie das Google Plugin für Eclipse. Das GWT wird benutzt um extra Funktionalität für Eclipse Benutzer zu bieten und um GWT Projekte entwickeln zu können. Das Google Plugin für Eclipse können Sie hier herunterladen:
<https://developers.google.com/eclipse/docs/install-eclipse-4.3>

Nach erfolgreicher Installation wählen Sie in der Menuleiste von Eclipse
1. Help 2. Install New Software und fügen Sie den nachfolgenden link ein:
<https://dl.google.com/eclipse/plugin/4.3>

Tipp: Wenn Sie irgendwelche Unklarheiten haben, konsultieren Sie:
<http://code.google.com/eclipse/docs/faq.html>

5.2 Aufgaben

Teilaufgabe 2.6: Das Google Web Toolkit Tutorial (3Punkte)

Arbeiten Sie das Google Web Toolkit Tutorial durch. Klicken Sie den Link <http://www.gwtproject.org/doc/latest/tutorial/gettingstarted.html> und folgen Sie den dort gegebenen Anweisungen.

Wichtig: Wir empfehlen Ihnen dringend, das Tutorial Schritt für Schritt gemäss den gegebenen Anweisungen durcharbeiten. Es bringt Ihnen nichts, wenn Sie den Code mit *copy* und *paste* in das Eclipse Projekt einsetzen. Wichtig ist, dass Sie genau verstehen, was der Code macht. Nur so können Sie später das GWT erfolgreich für Ihr eigenes Projekt nutzen.

Hinweise: Bei der Bearbeitung des Tutorials können die folgenden Probleme auftreten:

1. Wenn Sie das *StockWatcher* Tutorial machen und die entsprechende Webseite öffnen, dann kann es sein, dass Sie neu kompilieren müssen, ohne dass Sie dazu aufgefordert werden. Bitte auf das Projekt klicken und 1. *Google 2. GWT Compile* ausführen.
2. In Schritt 4 (Manage Events on the Client) von „Build a sample GWT App“ unter Punkt 2, geht es um das Handling von Keyboard Events. Sie können *event.getCharCode()* benutzen um den Buchstabe zu bekommen der auf der Tastatur gedrückt worden ist. Um den ASCII-Code (int) des eingetippten Zeichens zu bekommen, benutzen Sie die folgende Methode: *event.getNativeEvent().getKeyCode()*.
3. In Schritt 5 unter Punkt 3 kann es sein, dass Eclipse Sie wegen deprecated/veraltete Funktionsaufrufe warnt für *getMediumDateFormat* wenn Sie die folgende Zeile hinzufügen:

```
//Display timestamp last refresh.  
lastUpdateLabel.setText(„Last update:“  
+DateFormat.getMediumDateFormat().format(new Date()));
```

Eine *deprecated Warning* weist darauf hin, dass diese Funktion in zukünftigen Versionen der SDK nicht mehr verfügbar sein wird. Man könnte hier versuchen, dieselbe Funktionalität zu erreichen, indem man eine nicht *deprecated* Methode benutzt. Wir werden hier diese Warnung ignorieren.

4. In Schritt 6 steht nicht explizit dass man „*Debug as*“ anstelle von „*Run as*“ benutzen soll, um den Code zu debuggen.
5. GWT ist nicht für alle Web Browser optimiert. Es kann sein, dass es für einige Web Browser besser läuft als für andere. Um das Problem zu lösen gibt es externe Plug-Ins, die man in seinem Web Browser installieren kann.

Teilaufgabe 2.7: Das App Engine Tutorial (1 Punkt)

Im Rahmen dieses Tutorials müssen Sie die in Teilaufgabe 2.6 entwickelte *StockWatcher* Applikation in der Google App Engine hosten. Lesen Sie in <https://developers.google.com/appengine/> was die Google App Engine ist.

Bitte befolgen Sie die Schritte, die hier aufgelistet sind, um die *StockWatcher* Applikation online zu hosten:

<http://www.gwtproject.org/doc/latest/tutorial/appengine.html>

Hinweis: Das Tutorial beinhaltet Schritte, die mit oder ohne Eclipse gemacht werden können. Bitte befolgen Sie nur die Schritte, die Sie mit Eclipse machen müssen.

Nach Abgabe der Übung müssen Sie in der Lage sein, Ihre lauffähige StockWatcher Applikation den Tutoren zu demonstrieren und Fragen zur Implementierung der Applikation zu beantworten.