

Anwendungspraktikum aus JAVA
Priv.Doiz. Dr. Michael Hahsler
Institut für Informationswirtschaft
WS 2006/07

Dokumentation Projekt Virtuelles Tagebuch

Esra Wedam
(Matr. Nr. 9806106)

1 Problembeschreibung

Das Ziel dieses Projektes ist es, ein Tagebuch in elektronischer Form zu entwickeln. Den Benutzern soll dabei ermöglicht werden, in einem – mit einer GUI gestalteten – Textfeld Einträge anzulegen, die automatisch mit der aktuellen Uhrzeit und dem dazugehörigen Datum versehen werden und auch einen eigenen Titel tragen können. Die Daten sollen anschließend in einer vorher definierten Datei gespeichert werden können. Nach dem ursprünglichen Festhalten in einfacher Textform hat sich im Lauf der Entwicklung die Speicherung als XML-Dokument für die weitere Nutzung der Einträge als sinnvoll herausgestellt. Das XML-Format hat den Vorteil, dass es die Daten sehr portabel macht bzw. die Weiterverarbeitung stark vereinfacht. So können z.B. im Fall des virtuellen Tagebuchs die Einträge Hilfe eines Stylesheet im Browser betrachtet werden.

2 Analyse und Modellierung

Use-Case Diagramm

Wie Abbildung 1 zeigt ist der Aufbau des Use-Case Diagramms nicht allzu komplex. Benutzer können Einträge anlegen, dazu müssen sie zuerst einen Text schreiben und ihn dann speichern. Das Tagebuch, das aus den einzelnen Einträgen besteht, kann schließlich unter Einbeziehung zusätzlicher Hilfsmittel (Stylesheet, Browser) betrachtet werden.

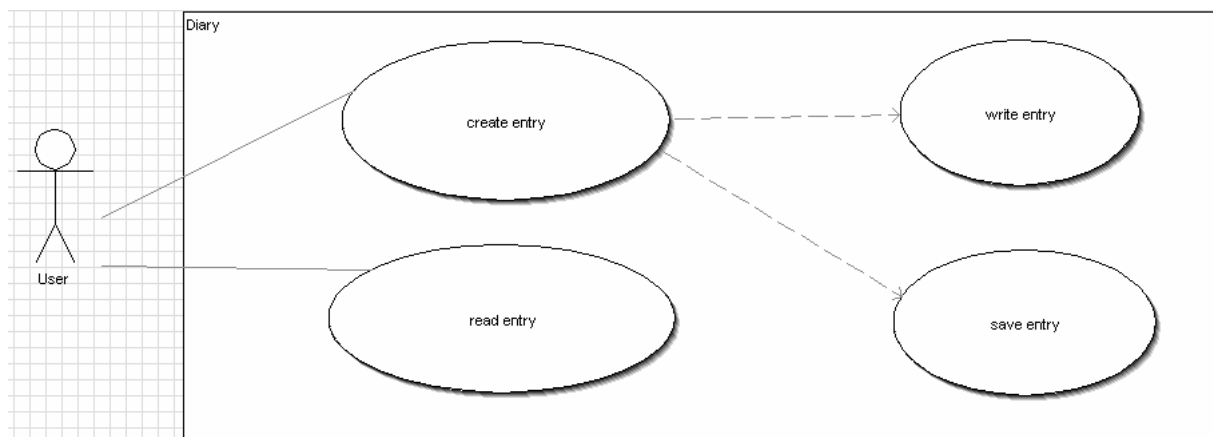


Abbildung 1 – Use-Case Diagramm

Beschreibung des Use-Case:

- Use-Case Name: Diary
- Akteure: User (Benutzer des Tagebuchs)
- Vorbedingungen: Computer, Java-Entwicklungsumgebung (JDK, Eclipse)
- Nachbedingungen: Ein Eintrag wurde angelegt.
- Auslöser: Wunsch einen Eintrag anzulegen
- Ablaufbeschreibung:
 - Der Benutzer startet das Programm.
 - Die GUI erscheint.
 - Optional: der Benutzer trägt einen Titel ein.
 - Der Benutzer schreibt einen Text.
 - Der Benutzer speichert den Eintrag.
 - Der Benutzer schließt das Fenster.
 - Der Benutzer liest im Browser alte Einträge.
- Fehlsituationen:
 - Der Benutzer möchte gleichzeitig zwei Einträge anlegen.
 - Der Benutzer möchte gleichzeitig Einträge speichern und lesen.
 - Der Benutzer benutzt Zeichen, die von XML nicht richtig interpretiert werden.
- Variationen: der Benutzer legt mehrere Einträge hintereinander an.
- Instanzen: Esra ist auf Urlaub und möchte mehrmals täglich ihre Erlebnisse festhalten. Sie hat ihren Computer dabei und startet deshalb das virtuelle Tagebuch. Das Fenster erscheint, Esra trägt einige Erlebnisse des Vormittags ein und speichert sie. Am Abend schreibt sie einen weiteren Eintrag. Da es sich um eine längere und besonders interessante Episode handelt, vergibt Esra einen Titel für diesen Eintrag. Wieder zurück zuhause sieht sich Esra ihre Urlaubseinträge im Browser an.
- Ergebnisse: Fertigstellung des Eintrags, Abspeicherung als XML.
- Nicht-funktionale Anforderungen: Freude über festgehaltene Erinnerungen
- Autor: Esra Wedam
- Ursprung: Dokumentation des Programms virtuelles Tagebuch

Klassendiagramm

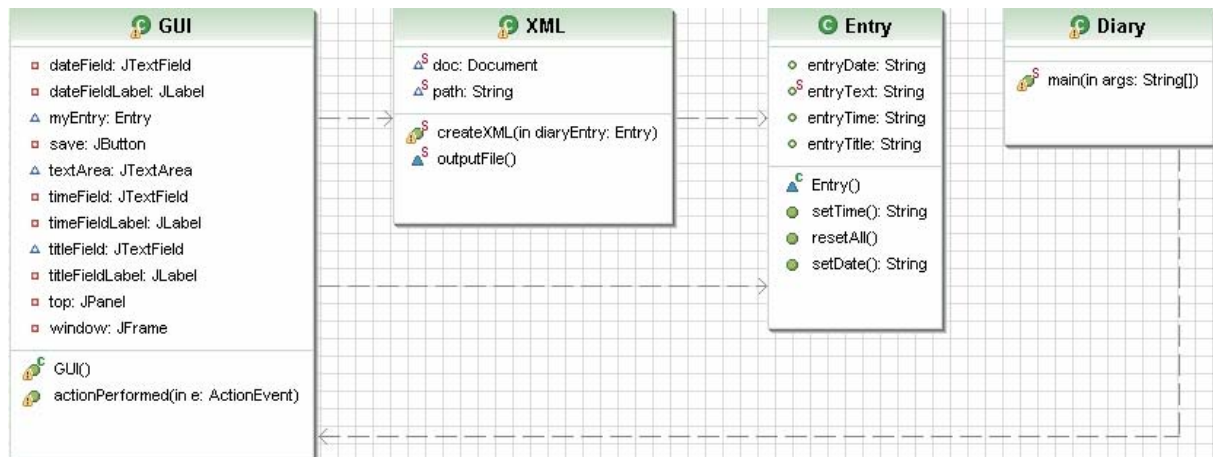


Abbildung 2 - Klassendiagramm

Klasse Entry: diese Klasse repräsentiert die kleinste Einheit innerhalb des Programms – einen einzelnen Eintrag. Wie anhand der Variablen *entryDate* und *entryTime* zu sehen ist, kann jedem Eintrag ein Datum und eine Uhrzeit zugeordnet werden. Außerdem besitzen Einträge selbstverständlich einen Textkörper (*entryText*) und können mit einem Titel (*entryTitle*) versehen werden. Die Methoden *setTime* und *setDate* benutzen die Klasse *GregorianCalendar* aus der Java API und weisen den Variablen *entryDate* und *entryTime* das aktuelle Datum bzw. die Uhrzeit zu. Die Methode *resetAll* aktualisiert Datum und Uhrzeit und setzt Text und Titel zurück.

Klasse XML: hier werden alle Vorkehrungen getroffen, damit ein gültiges XML-Dokument angelegt werden kann, das anschließend unter einem angegebenen Pfad (*path*) gespeichert wird. Dazu greift diese Klasse auf die Klasse *Entry* zu und verwandelt die dortigen Variablen in XML-Elemente. Die Methode *createXML* überprüft zuerst, ob die vorgegebene Datei bereits besteht. Ist dies der Fall, dann werden die einzelnen Elemente eines Eintrags zum bestehenden Dokument hinzugefügt. Die Grundeinheit Eintrag sieht in XML folgendermaßen aus:

```

<entry>
  <entryDate/>
  <entryTime/>
  <entryTitle/>
  <entryText/>
</entry>
  
```

Wurde die Datei hingegen noch nicht angelegt, so sorgt die Methode dafür, dass ein gültiges Dokument erstellt wird. Es wird ein Grundgerüst gebildet, in das die Einträge nach und nach eingefügt werden können. Eine vollständige, gültige XML-Datei könnte beispielsweise wie folgt aussehen:

```
<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet href="Tagebuch.css" type="text/css"?>
<!DOCTYPE diary SYSTEM "Tagebuch.dtd">
<diary>
  <entry>
    <entryDate>11.02.2007</entryDate>
    <entryTime>14:48</entryTime>
    <entryTitle>Neuer Titel</entryTitle>
    <entryText>Neuer Text</entryText>
  </entry>
</diary>
```

Die Methode *outputFile* speichert schließlich das zuvor erstellte Dokument ab.

Klasse GUI: dieser Bereich stellt die Schnittstelle mit dem Benutzer dar. Es wird ein Fenster (*window*) erzeugt, das im oberen Bereich (*top*) Felder für das Datum, die Uhrzeit und den Titel des Eintrags hat (*dateField*, *timeField*, *titleField*), wobei Datum und Uhrzeit automatisch eingetragen werden. Titel und Text können händisch in die vorgesehenen Bereiche eingetragen werden. Bei anschließender Betätigung des Speicherknopfes (*save*) werden die einzelnen Elemente des Eintrags in XML-Elemente umgewandelt und im Zuge der Speicherung in eine neue bzw. bestehende Datei geschrieben. Gleichzeitig werden die einzelnen Felder der *GUI* zurückgesetzt, sodass der Benutzer sieht, dass er einen neuen Eintrag beginnen kann. Beim Schließen des Fensters wird das Programm beendet. Die Klasse *GUI* greift sowohl auf die Klasse *Entry*, als auch auf die Klasse *XML* zu.

Klasse Diary: diese Klasse beinhaltet die *main* Methode, welche die *GUI* aufruft.

Sequenzdiagramm

Das Sequenzdiagramm zeigt besonders gut die Interaktion der einzelnen Klassen. Der Benutzer interagiert eigentlich nur mit der *GUI*, indem er den Eintrag schreibt und speichert. Die *GUI* wiederum legt mit Hilfe der Klassen *Entry* und *XML* den Eintrag an. *Entry* liefert die Definition von Datum und Uhrzeit, *XML* überprüft, ob es sich um einen Eintrag in ein neues oder bestehendes Dokument handelt und speichert die Daten als XML-Datei.

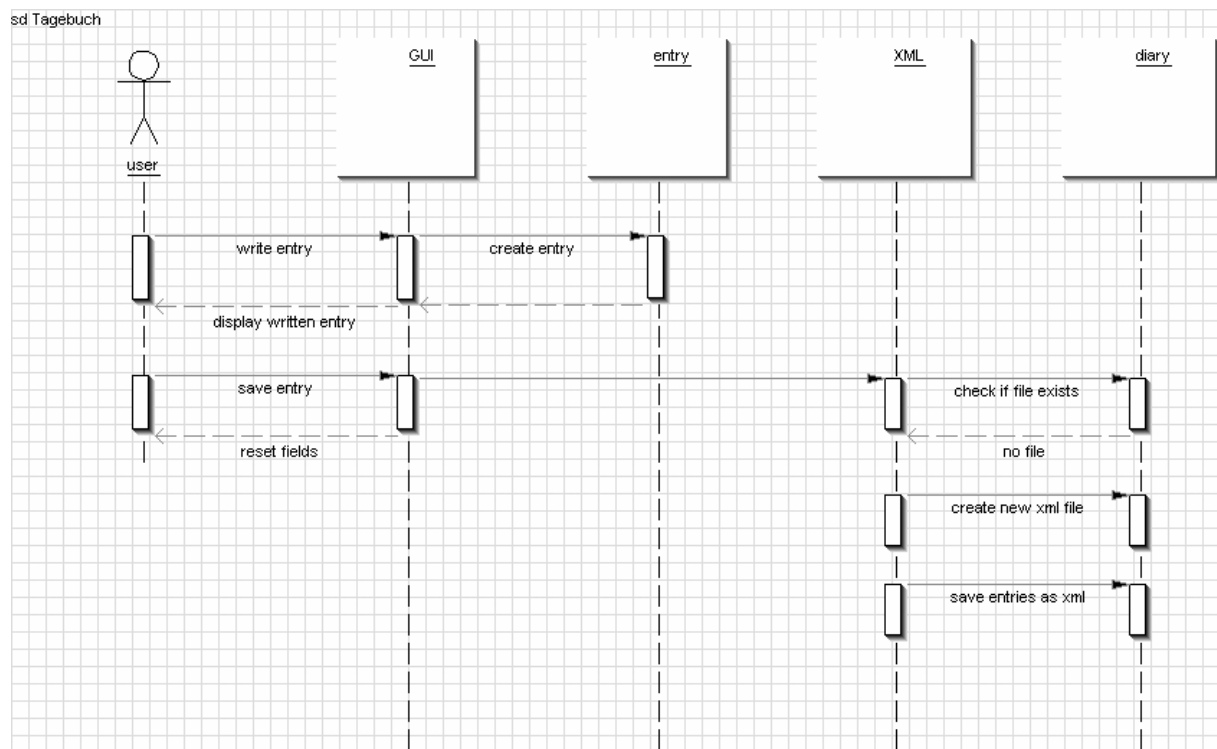


Abbildung 3 – Sequenzdiagramm

3 Ein Blick auf das Programm

Die Benutzung des virtuellen Tagebuchs ist äußerst einfach. Datum und Uhrzeit werden vom Programm automatisch ermittelt und eingetragen. Das Feld für den Titel kann, muss aber nicht beschriftet werden, dasselbe trifft auch auf das Textfeld zu. Sobald der Benutzer den Eintrag abschließen möchte, muss er nur den Speicherknopf drücken.

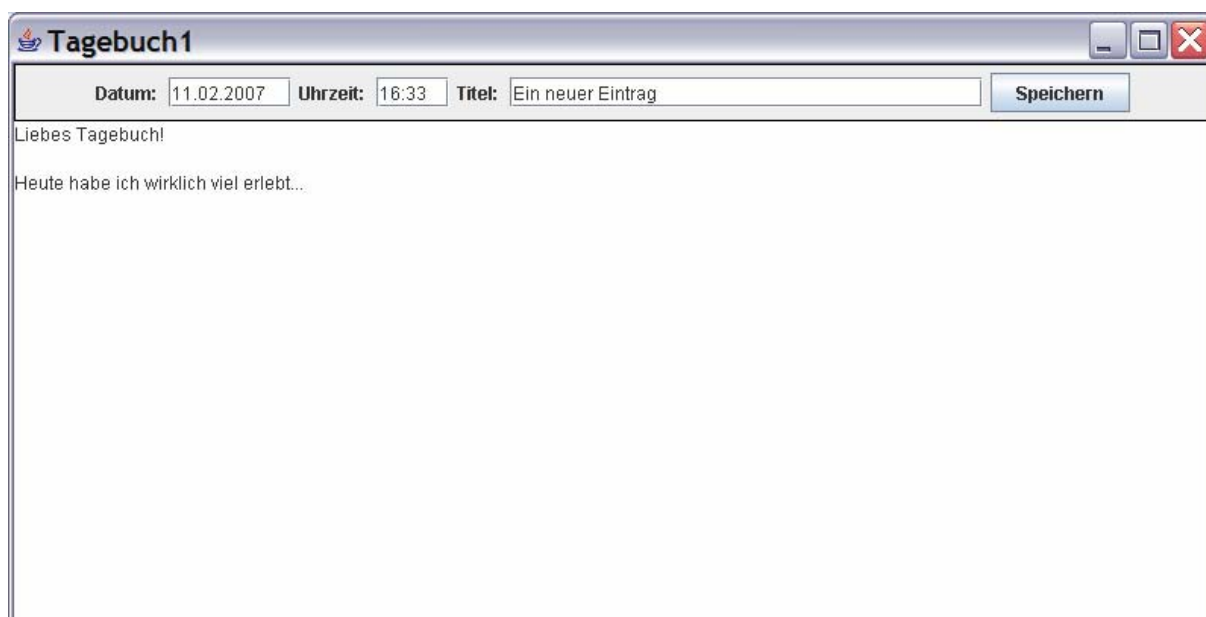


Abbildung 4 - Ein neuer Eintrag

Die Felder in der GUI werden dabei zurückgesetzt bzw. geleert, sodass für den Benutzer zu erkennen ist, dass ein neuer Eintrag angelegt werden kann.

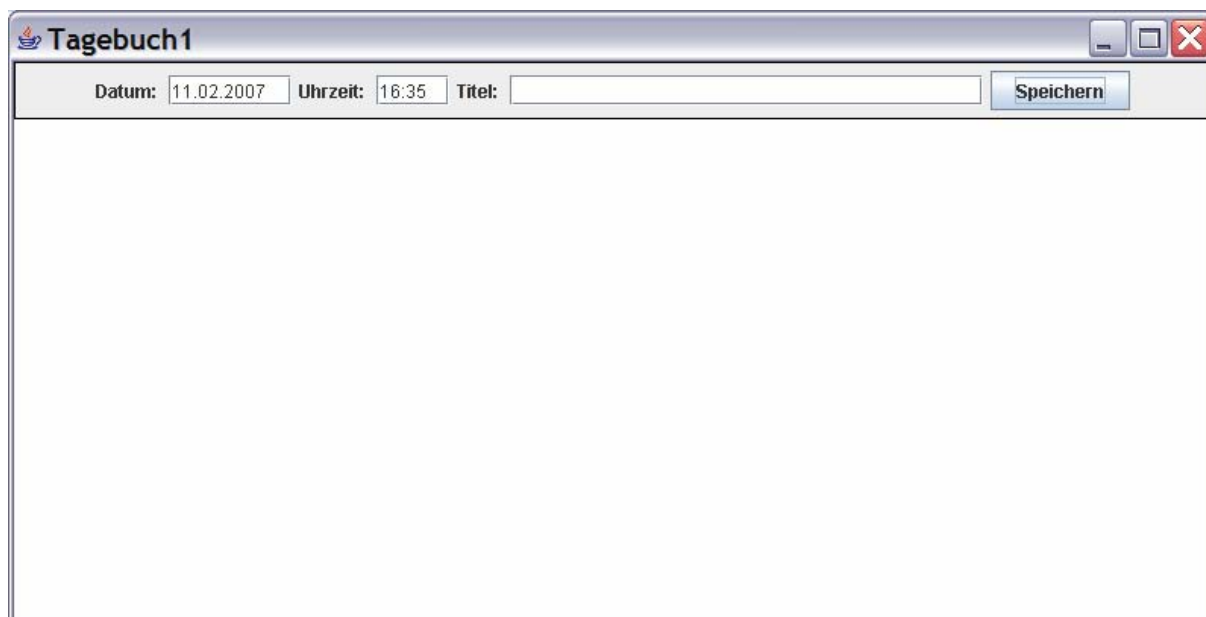


Abbildung 5 - Zurückgesetzte Felder

4 Installation und Wartung

Das Programm benötigt zur Ausführung eine Java-Entwicklungsumgebung, wie *JDK 5.0* oder *Eclipse* (im Internet verfügbar:

<http://java.sun.com/j2se/1.5.0/download.jsp> bzw. unter: <http://www.eclipse.org>).

Des Weiteren wird *JDOM* (API zur Bearbeitung von XML) benötigt – ein Werkzeug, das ebenfalls im Internet kostenlos zur Verfügung steht (<http://www.jdom.org>). Nach dem Download muss die Datei *jdom.jar* dem Klassenpfad hinzugefügt werden, damit das Programm korrekt kompiliert werden kann.

Die Tagebuch-Dateien werden beim Speichern unter *C:* abgelegt. Es ist empfehlenswert, die dazugehörige DTD-Datei und das CSS-Stylesheet (beide können gemeinsam mit dem Programmcode heruntergeladen werden) am selben Ort zu speichern, damit das XML-Dokument formatiert im Browser betrachtet werden kann.

Für das virtuelle Tagebuch ist keine Wartung vorgesehen.