UNIVERSITÄT PASSAU

Fakultät für Informatik und Mathematik

SEP Informatik – Wintersemester 2017/2018

Netzwerkfähiger Multi-User Texteditor: Lastenheft

Thomas Bock, Armin Größlinger, Kolja Thormann

1 Eine kurze Bemerkung vorab

Dies ist **Euer** Praktikum. Dieses Dokument ist kein Katalog von Aufgaben, der Punkt für Punkt abgearbeitet werden muss, um das SEP zu bestehen, sondern lediglich eine Reihe von Hinweisen, was wir erwarten. Es ist aus diesem Grund knapp gehalten. Wie **Euer** Programm letztendlich aussehen soll, müsst **Ihr** selbst entscheiden.

2 Motivation

Ein netzwerkfähiger Editor mit grafischer Benutzeroberfläche vereint mehrere für Informatiker interessante Herausforderungen: Zum Einen muss das Model-View-Controller-Prinzip zusammen mit einer Schichtenarchitektur umgesetzt werden, um die eigentliche Logik des Editors, die Netzwerkkommunikation und die grafische Oberfläche voneinander zu entkoppeln. Dies ermöglicht Flexibilität und Erweiterbarkeit.

Eine zweite Herausforderung stellen die asynchrone Kommunikation zwischen verschiedenen Instanzen des Editors (beim gemeinsamen Bearbeiten eines Dokuments) und die sich daraus ergebenden nebenläufigen Vorgänge (Multi-Threading) im Editor dar. Mehrere Benutzer können quasi gleichzeitig im selben Dokument Änderungen vornehmen. Dadurch wird eine automatisch Konfliktlösung, wenn Nutzer miteinander unvereinbare Änderungen vornehmen, nötig.

Daher ermöglicht die Erstellung des Editors das Kennenlernen von unterschiedlichen Technologien (Multi-Threading, Netzwerkkommunikation, GUI-Programmierung, Design Patterns, u. a.).

3 Aufgabenstellung

Ziel des SEP im Wintersemester 2017/2018 ist es, einen Texteditor als Standalone-Applikation (kein Applet, keine Web-Applikation) zu entwickeln, bei dem mehrere User über das Netzwerk zusammen an einem (oder mehreren) Texten arbeiten können.

Der Texteditor soll den heute üblichen Funktionsumfang für das lokale Editieren von Texten liefern und eine benutzerfreundliche grafische Oberfläche bieten.

Die Applikation umfasst neben der eigentlichen Editor-Komponente auch eine Komponente („Dokument-Master“), die die Freigabe von Dokumenten an andere Nutzer verwaltet. Andere Benutzer können sich mit dem Dokument-Master einer anderen Editor-Instanz (eines anderen Benutzers) verbinden, um an einem Text mitzueditieren (falls die nötigen Zugriffsrechte vorliegen).

Alle Benutzer, die denselben Text editieren, sehen, an welchen Stellen im Text andere Benutzer arbeiten (grafische Markierung der aktuellen Eingabeposition jedes Benutzers). Benutzer können Anmerkungen/Kommentare (die nicht Teil des eigentlichen Textes sind) an bestimmte Textstellen anhängen. Andere Benutzer sehen diese Kommentare und können sie bearbeiten. Die Benutzer, die am selben Text arbeiten, können über ein Chatsystem miteinander interagieren.

Benutzer können die Historie (Geschichte der Änderungen am Text) einsehen und Änderungen rückgängig machen (Undo). Das Undo kann benutzer-spezifisch, d. h. nur die Änderungen eines bestimmten Benutzers (sofern dies möglich ist), oder global (wie klassisches Undo) erfolgen.

Treten beim Bearbeiten eines Textes Konflikte auf, da unvereinbare Änderungen zum Dokument-Master gesendet werden, löst der Dokument-Master die Konflikte automatisch, ohne Benutzerinteraktion, auf.

Die Anwendung muss mit Java 7 oder Java 8 und Swing entwickelt werden. Es ist besonders darauf zu achten, das Model-View-Controller-Pattern konsequent umzusetzen. Ebenfalls sollen andere wichtige Design Patterns, z. B. das Observer Pattern, bei der Implementierung berücksichtigt werden.

4 Produkteinsatz

Zielgruppe der Anwendung sind Personen, die gemeinsam Texte editieren möchten. Der Einsatz soll dabei in Unternehmen, Arbeitsgruppen, Schulen, etc. in lokalen Netzwerken möglich sein.

5 Produktfunktionen

5.1 Minimale Leistungsmerkmale

5.1.1 Editor

- benutzerfreundliche GUI
- Bedienung mit Maus und Tastatur
- Anlegen/Öffnen/Speichern/Schließen von Dokumenten
- „übliche“ Editier-Operationen, z. B. Suchen und Ersetzen
- Ansicht der Historie des Texts
- Undo-Funktionalität: Rückgängig machen von Editieroperationen, benutzer-spezifisch und global
- Platzieren von Annotationen/Kommentaren an Textstellen
- Bearbeiten mehrerer Dokumente gleichzeitig
- Verbindung zu mehreren Dokument-Mastern gleichzeitig
- Chat-System zur Kommunikation mit anderen Usern, die am selben Dokument arbeiten
- Verschlüsselte Netzwerkkommunikation
- Toleranz gegenüber kurzzeitigen Netzwerkausfällen

5.1.2 Dokument-Master

- Zusammenfügen von Änderungen verschiedener User
- Auflösen von Editierkonflikten auf Zeilenebene
- Weiterleiten aller Editierungen eines Dokuments an alle verbundenen Editor-Instanzen
- Herstellen einer konsistenten Historie der Dokumenteditierungen
- Verwaltung von Zugriffsrechten/Freigaben, d. h. welcher User erhält Zugriff auf welche Dokumente, z. B. über Passwörter

5.2 Optionale Leistungsmerkmale

Folgende Ideen können als Anstoß für eigene Erweiterungen des Programms dienen:

- Syntax-Highlighting
- feinere Auflösung von Editierkonflikten, z. B. auf Wortebene
- Issue-Tracking/Verwaltung einer TODO-Liste mit Zuständigkeiten, etc. in einem Dokument
- Asynchrone Übertragung von Änderungen im Dokument zum Dokument-Master zur Steigerung der Interaktivität in Netzwerken mit hoher Latenz
- Alternative Authentifizierungsmechanismen für Benutzer (z. B. OpenID, Whitelists)

6 Organisatorisches

Für jede Phase des Praktikums muss ein Phasendokument abgegeben werden. Dieses Dokument ist Grundlage für das Kolloquium am Ende jeder Phase, in dem der Phasenverantwortliche die Ergebnisse der Phase vorträgt. Die genauen Termine entnehmt bitte der Homepage[1].

7 Bewertung

Die Benotung des SEP richtet sich nach folgenden Kriterien:

- Qualität der abgegebenen Dokumente
- Qualität der Kolloquien und der Individualleistung in wöchentlichen Meetings
- Qualität der Abschlusspräsentation
- Qualität des Source-Codes
- Erfüllung der minimalen Leistungsmerkmale (s.o.)
- Sinnvolle Erweiterungen über diese Merkmale hinaus
- Robustheit des erstellten Programms

Diese Liste hat keine Reihenfolge, die einer Gewichtung entspricht. Es gibt weitere Punkte, die wie bei Aufträgen zur Softwareentwicklung in der Wirtschaft als selbstverständlich gelten und sich bei Nichterfüllen negativ auswirken, z. B. sollte die Bedienung des Programms möglichst einfach von der Hand gehen, etc.. Umgekehrt wirkt sich eine besonders gute Erfüllung einer Anforderung oder ein besonderes Feature des Endprodukts natürlich positiv aus.

Viel Erfolg und vor allem viel Spaß!

[1] `http://www.infosun.fim.uni-passau.de/cb/Kurse/sep_ws1718/inf/`