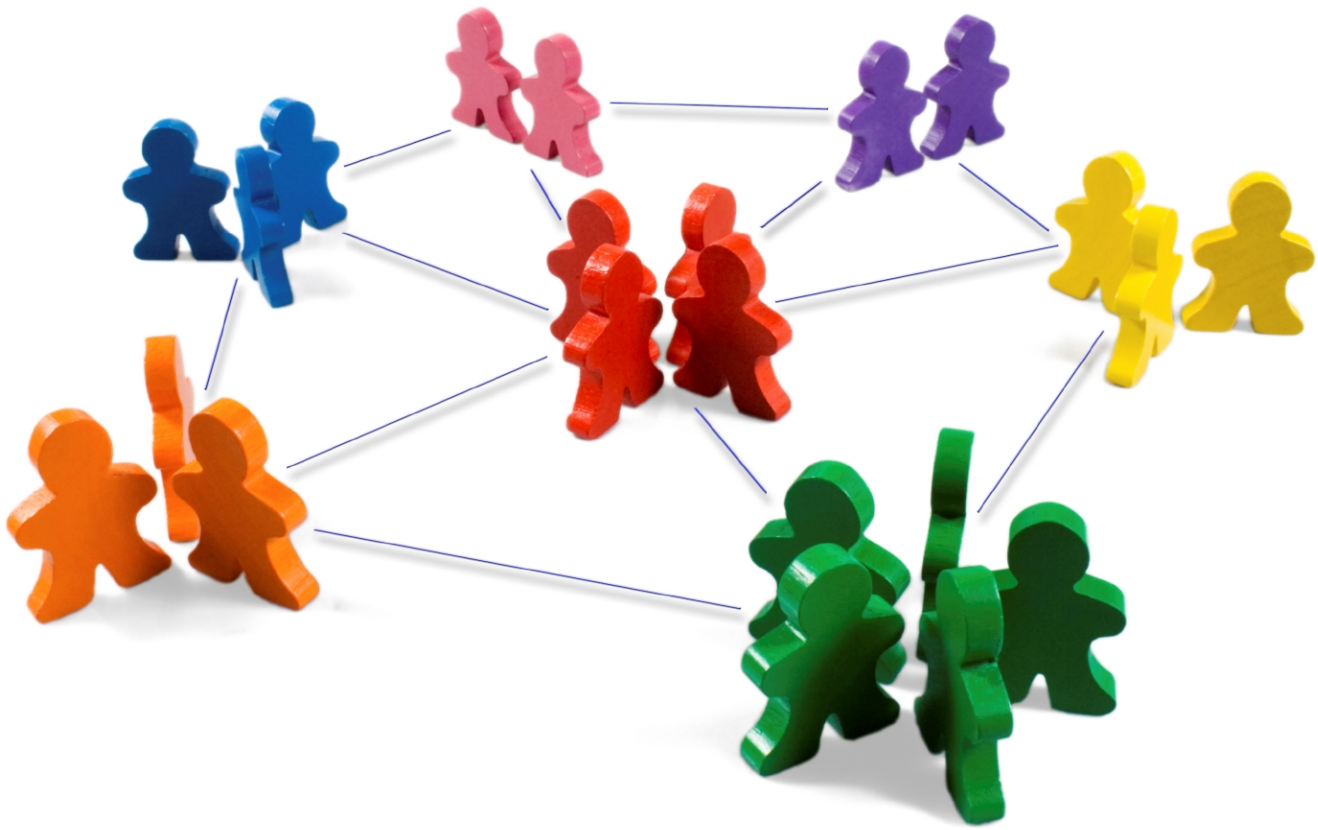
Softwareprojekt
Multichannel



ZHAW BA Informatik
2. Semester
Datum: 12.06.2013

Simon Bächler
Dan-Linh Nguyen-Kim
Milijana Tomic

Inhaltsverzeichnis

Einleitung

Multichannel ist eine Simulation eines Nachrichtensystems, welches auf einer diversivizieren Infrastruktur basiert. Nachrichten verschieder Typen können von unterschiedlichen Geräten aus über verschiedene Netzwerke transportiert werden. Nachrichten können offline erstellt werden und werden verschickt, sobald das Gerät online geht. Ein ansprechendes GUI stellt die Nachrichten und die Übersicht dar. Bei Versandfehlern, zB. einer nicht existierenden Adresse, wird der Absender benachrichtigt.

Auftrag

Konzipieren Sie eine Software die Nachrichten auf unterschiedlichen Art versenden / transportieren kann.

Eine Nachricht kann als 'SMS', 'MMS', 'Email', 'Print' versendet werden.

Jede Nachricht kann an einzelne Personen aber auch Gruppen gesendet werden.
Jede Nachricht kann zu einem bestimmten Zeitpunkt versendet werden oder auch sofort.Jede SMS, MMS und Email kann zu einem bestimmten Termin eine Nachricht vorab senden.
(Reminder) (Bzw. Termin ist um 18:00 Uhr – Erinnerung ist um 17:00)

Bedenken Sie, welche Faktoren einen Einfluss bei Nachricht haben. Berücksichtigen Sie nur logische Gründe nicht physische Ursachen/Faktoren.

Bei allen 4 Kanälen müssen Validierungen durchgeführt werden, damit der mögliche Transport gewährleistet werden kann.

Überlegen Sie sich die Use Cases zeichnen und beschreiben Sie diese.

Bilden Sie ein Klassendiagramm, welches die Beziehungen und Interaktionen aufzeigt. Erklären Sie, welche Voraussetzungen Sie warum angenommen haben.

Entwickeln Sie ein Test Programm, dass entweder auf User – Eingaben reagiert oder für alle 4Kanäle mit simulierten Testwerten durchläuft.

Die Eingaben bzw. Defaultwerte und die Resultate der einzelnen Kanäle muss auf der Konsole einsehbar sein.

Das Programm muss erklärt und funktionstüchtig sein.

(Achtung: das Versenden muss nur als Konsolenmeldung simuliert werden. Für funktionierende SMS, MMS, Emails oder Printaufträge werden keine Punkte gegeben!!)

Wichtig: Die oben gemachten Angaben lassen bewusst Spielraum für eigene Ideen.

Diese Aufgabe kann smart und effizient mit möglichem Zukunftspotential gelöst werden, das Gegenteil wäre eine einfache, minimalistische ohne jede Wiederverwendbarkeit oderAusbaufähigkeit im Sinne der Objektorientiertheit.

Es geht ums OO – Denken und Entwickeln, nicht um ein Prográmmchen das einfach läuft.

Bauen Sie die Software so, dass Sie den Kern der Aufgabe erfasst und im Sinne von OO ausgebaut werden kann.

Die Bewertung der Aufgabe zu 2/3 erfolgt im Stil einer Experteneinschätzung, 1/3 erfolgt im Stile
Korrektur von harten Kriterien (Konventionen, UML formale Fehler)

In der Praxis gilt dasselbe: Auch korrekte UML / Klassendiagramme, erfüllte Konventionen können auch in einem völlig unbrauchbaren Programm vorhanden sein, dennoch bleibt das Programmunbrauchbar.

Arbeitsweise

Kommunikationsmittel

Aufgabe	Tool	Link
Gemeinsames Dokument	Google Docs	http://docs.google.com
Kommunikation	Skype, Email, Whats App	http://www.skype.com
UML	UML Violet Editor	http://alexdp.free.fr/violetumleditor/page.php
Entwicklungsumgebung	Eclipse	http://www.eclipse.com
Versionskontrolle	Git / Github	https://github.com/
Testen	JUnit	http://www.junit.org

Planung

Aufgabenstellung

- Eine Nachricht kann als SMS, MMS, EMail oder Print versendet werden
- An Personen oder auch Gruppen
- Kann sofort oder zu einem bestimmten Zeitpunkt versendet werden
    - SMS, MMS und Email: kann zu einem bestimmten Termin vorab senden (Reminder)
- Welche Faktoren einen Einfluss bei Nachricht haben können (nur logische Gründe)
- Validierungen bei 4 Kanälen durchführen
- Use Case zeichnen und **beschreiben**
- Klassendiagramm zeichnen und **erklären**
- Testprogramm
- Die Eingaben bzw. Defaultwerte und die Resultate der einzelnen Kanäle muss auf der Konsole einsehbar sein.
- das Versenden muss nur als Konsolenmeldung simuliert werden

Zeitplan

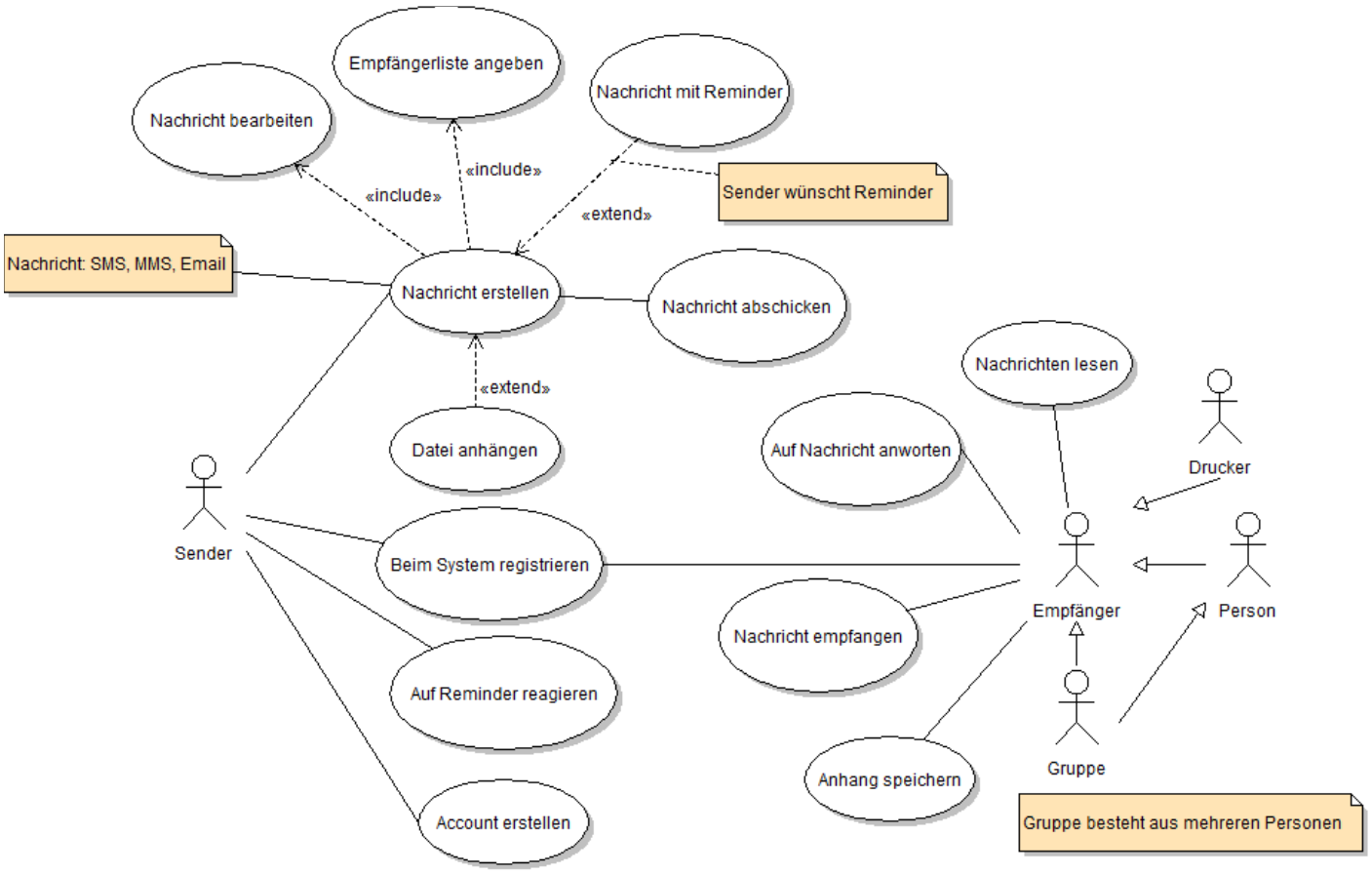
26.02.2013	Besprechung bei Dan Linh um 18:30
04.03.2013	ZHAW (Besprechung in der Mittagspause)
06.03.2013	Treffen in der ZHAW um 17:40
30.03.2013	Treffen bei Dan Linh um 11:00 Diagramme fertig, beginn mit Umsetzung
03.04.2013	Tutoriatsabend
15.5.2013	Dan Linh steigt aus dem Studium aus.
22.05.2013	Tutoriatsabend
29.05.2013	Besprechung in der ZHAW um 20:00
02.06.2013	Um 14:00 Skypen
03.06.2013	Kurze Besprechung in der Mittagspause
13.06.2012	Abgabetermin

Softwaredesign

Wir entschlossen uns, mit diese Software die Funktion diese Kanäle in der realen Welt zu modellieren. Daraus entstand das Konzept mit den Gruppen Geräte, Clients, Server und Messages. Die einzelnen Komponenten sind so genrisch gehalten, wie es geht, und wir achteten auf geringe Koppelung und hohe Kohäsion. Auch haben wir ein Auge auf die Performance gehalten, so laufen einige Aktionen, z.B. die Weiterleitung einer Nachricht an einen andern Server und alle Gui-Fenster in eigenen Threads.

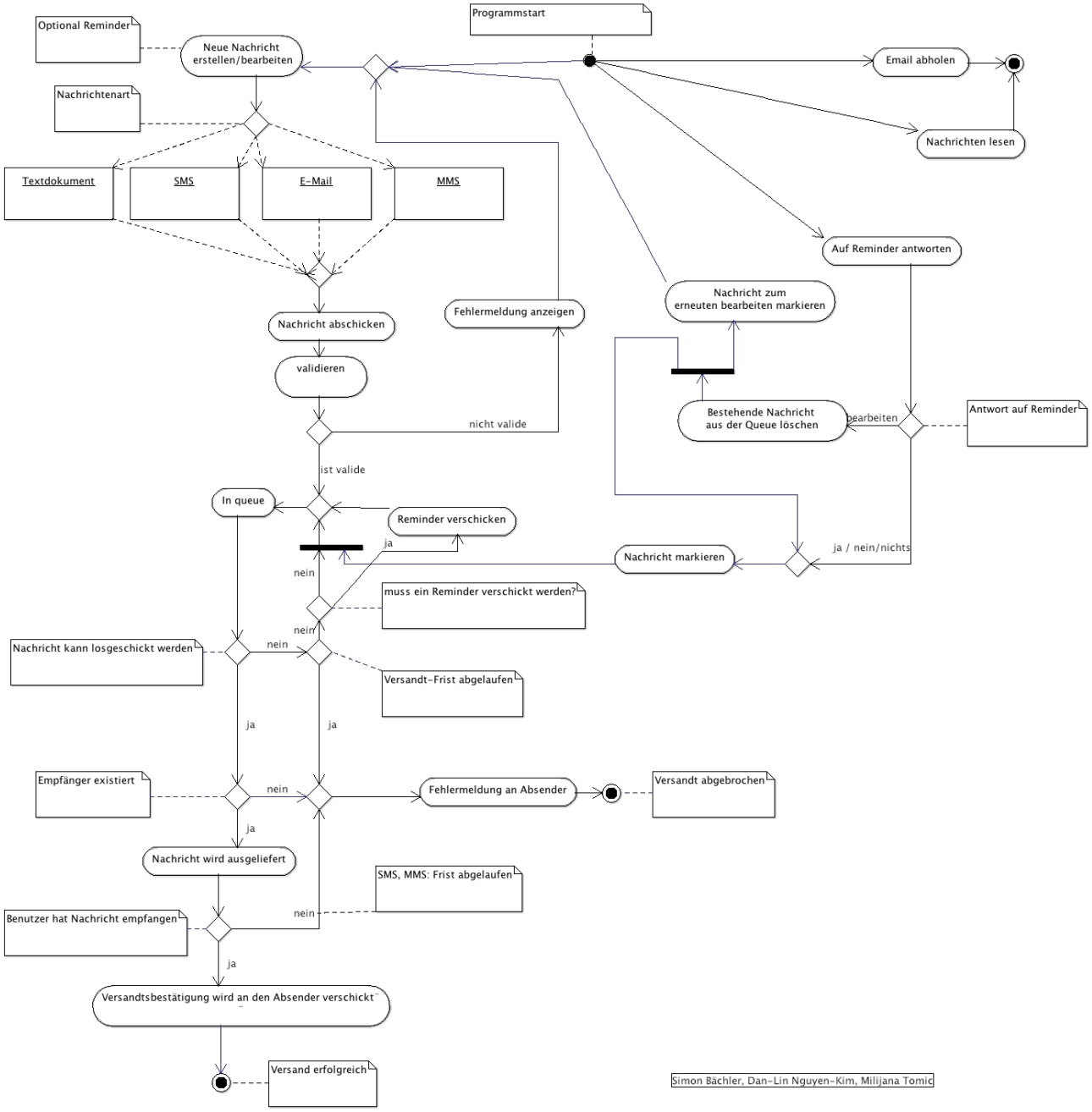
Use Case

Das Use-Case Diagramm war das erste Diagramm welches wir erstellt hatten. Mit diesem Diagramm haben wir alle möglichen Tätigkeiten des Senders und Empfängers dargestellt.



Activity Diagramm

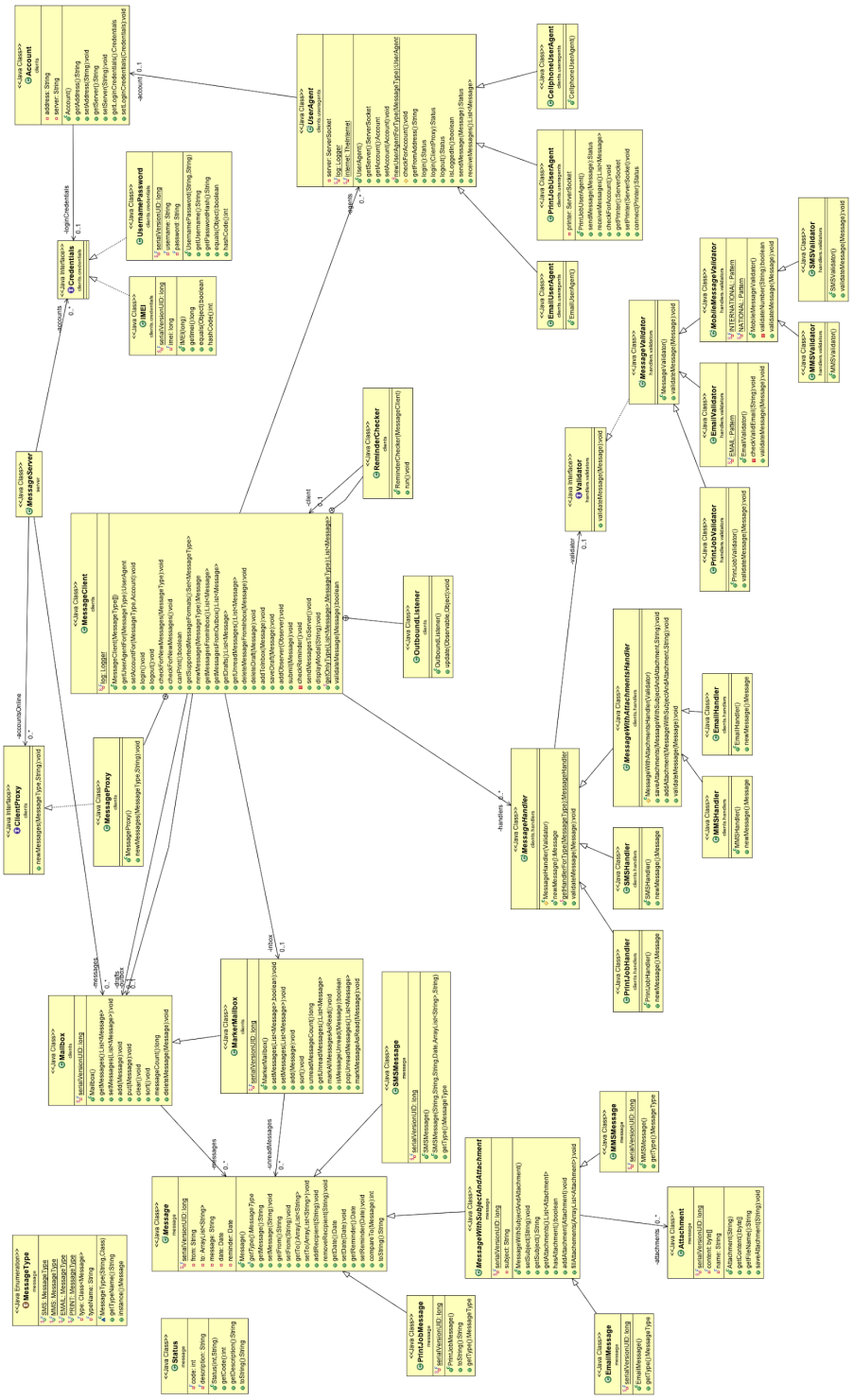
Wir erstellten dieses Diagramm noch vor dem Klassendiagramm. Es ist ein wichtiges Diagramm zum Verständnis der Abläufe und hat dem Team insoweit geholfen, als dass alle sich bei der Entwicklung von Komponenten darauf beziehen konnten.
Eine höher auflösende Version befindet sich im Projektordner.



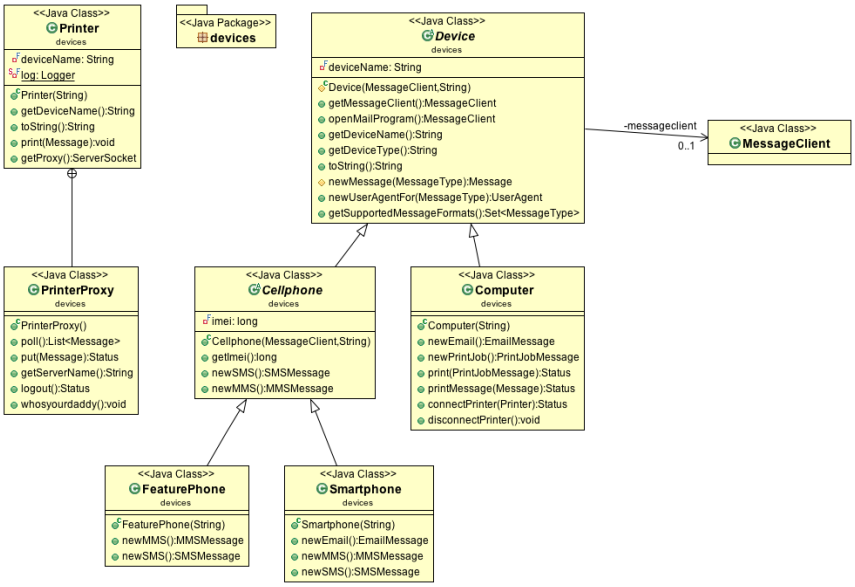
Klassendiagramme

Das Klassendiagramm hat mehrere Entwicklungsstufen hinter sich. Die einzelnen Versionen befinden sich im Doku-Ordner unter Archiv. Die aktuellen Diagramme sehen so aus:

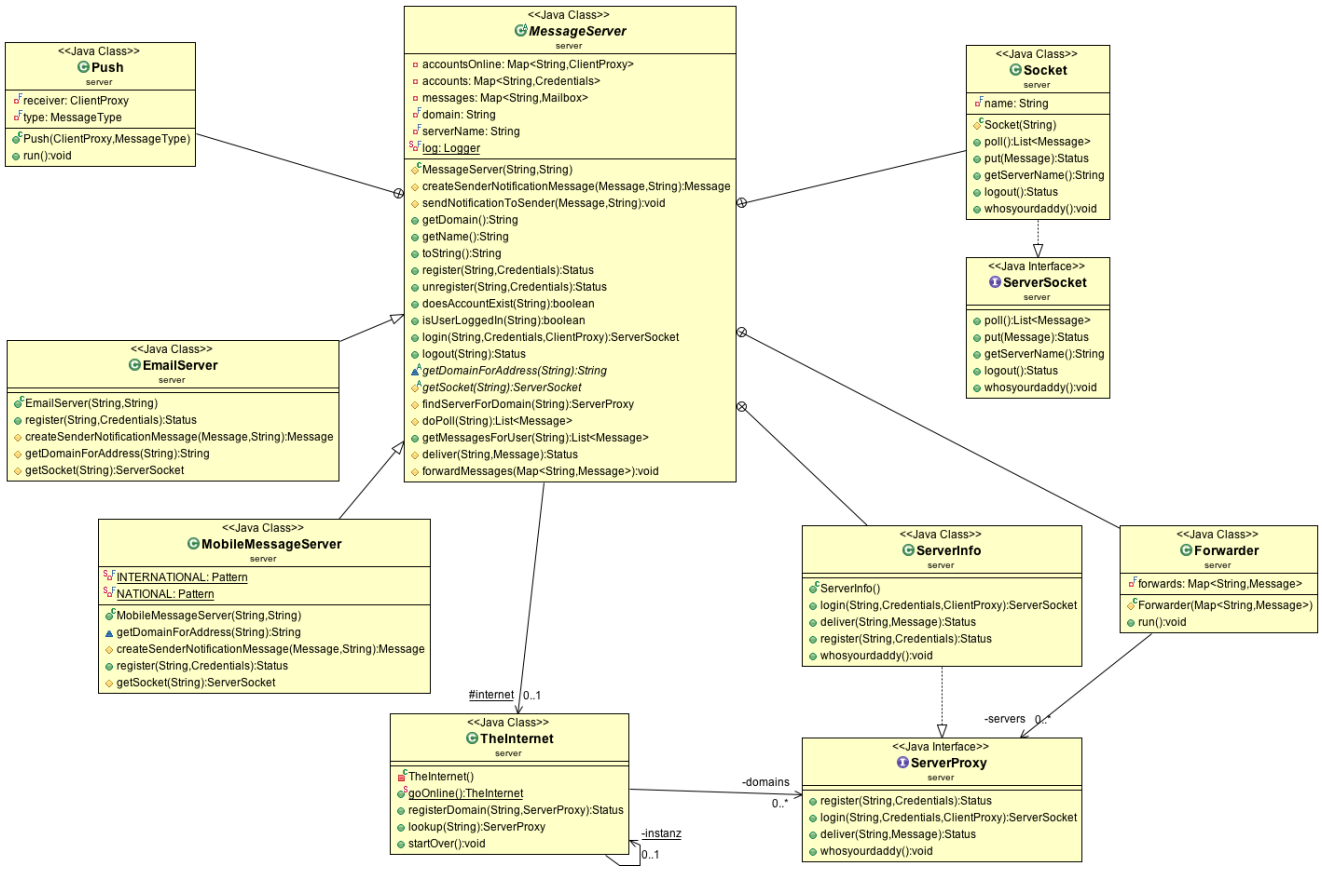
Clients



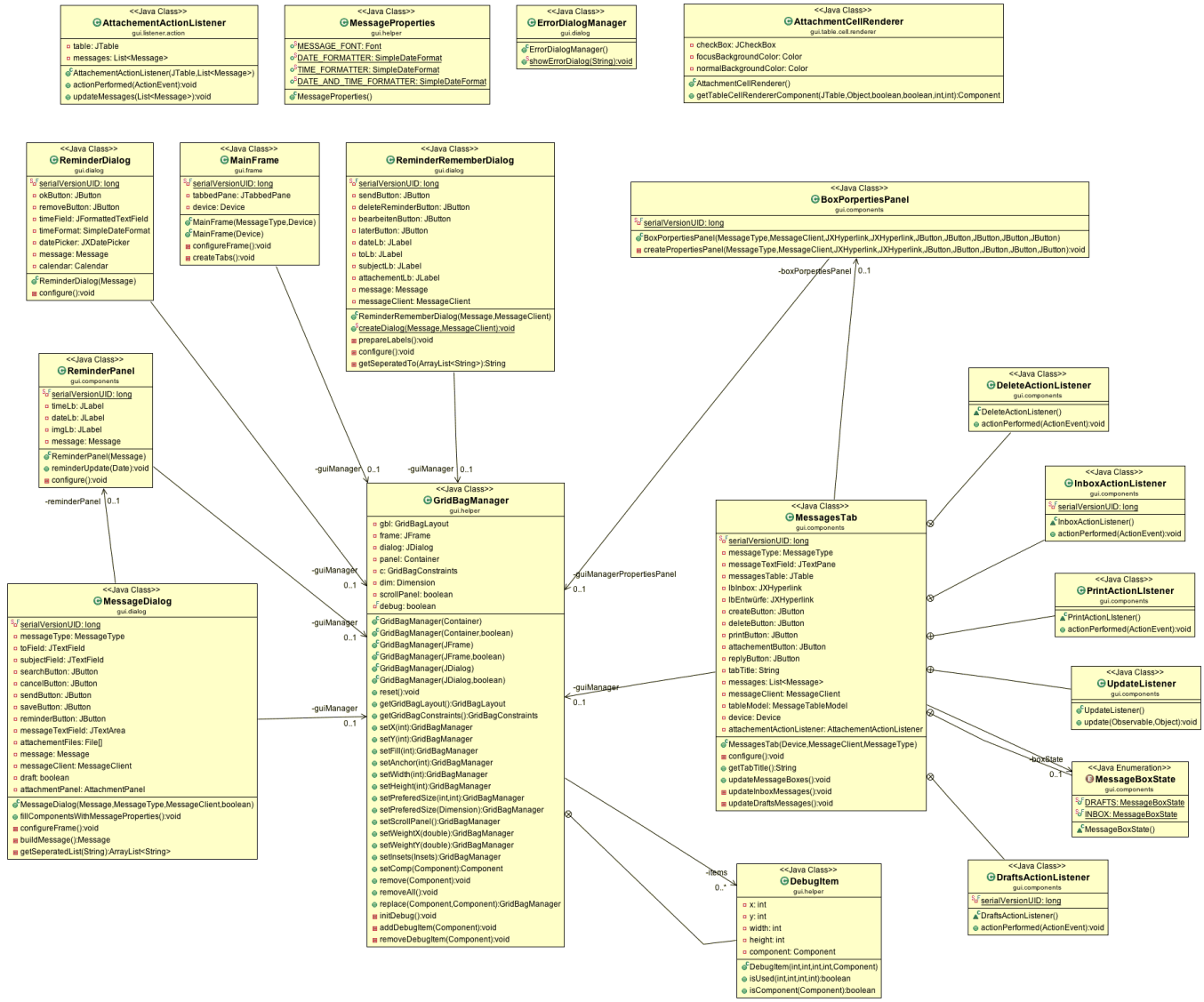
Devices



Server



Gui



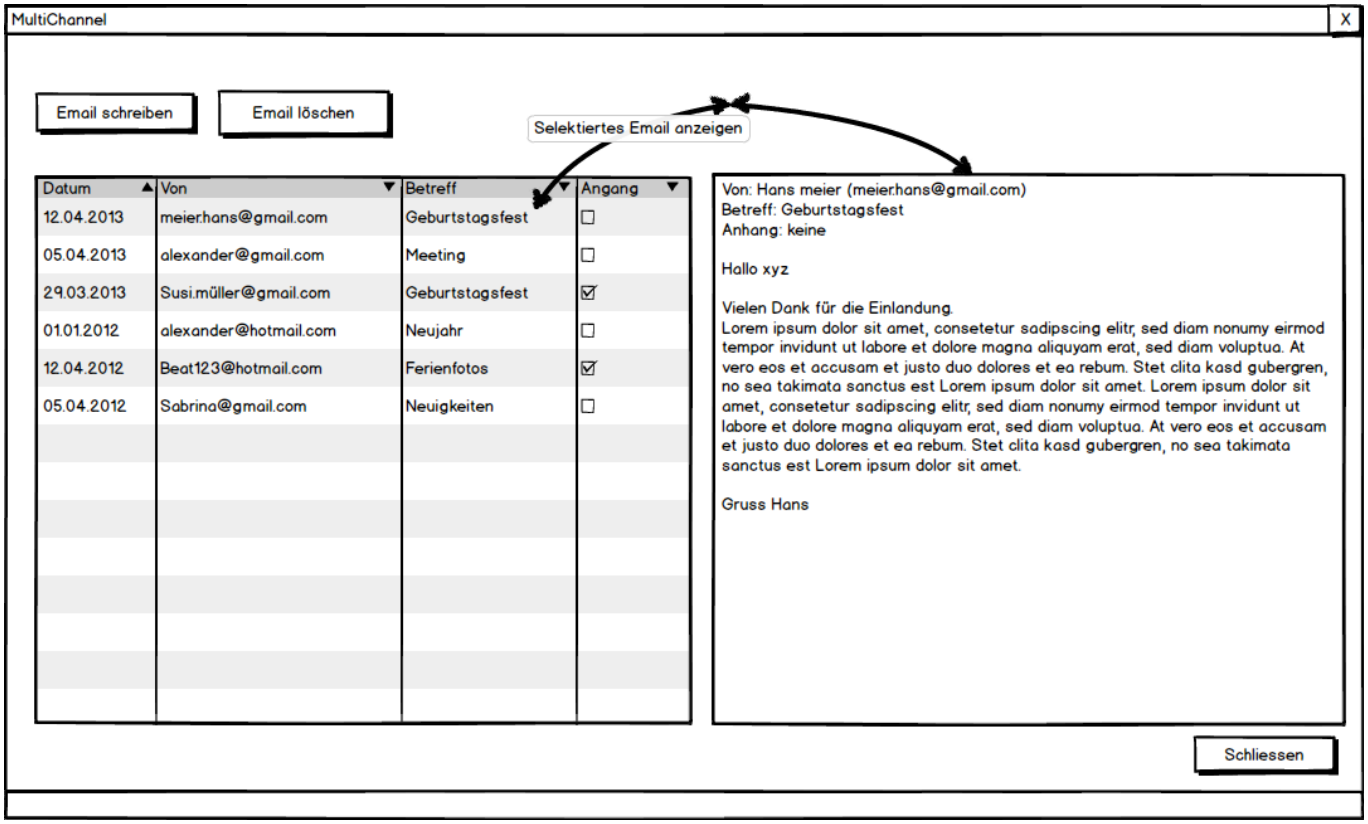
GUI Skizzen

Bevor wir angefangen haben das GUI zu programmieren hatten wir zuerst Skizzen gemacht. Erst als wir mit den Skizzen zufrieden waren, setzten wir diese um.

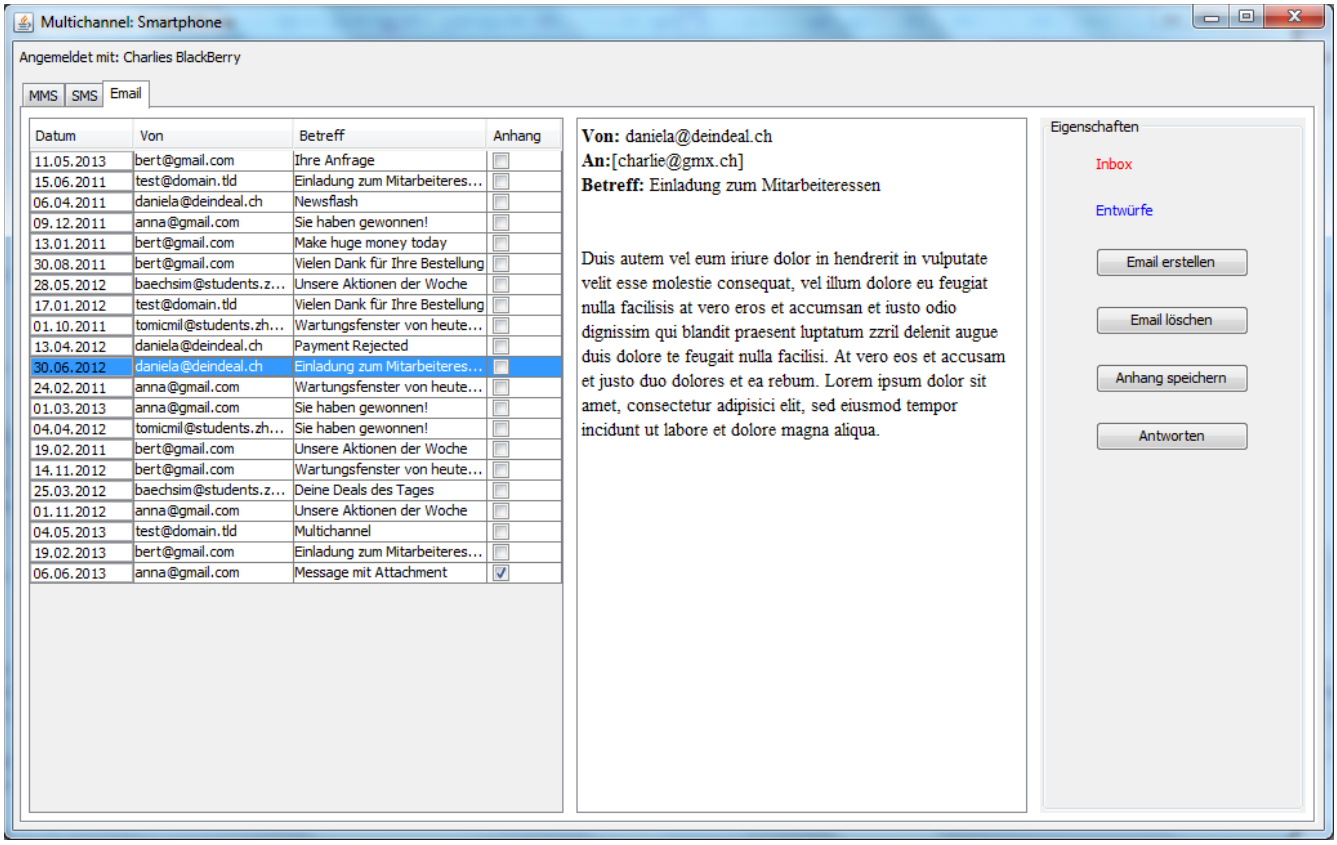
Hauptfenster

Wir wollten, dass der User alle Nachrichten im Hauptfenster sieht, und diese bearbeiten kann. Es sollte ähnlich aussehen wie Outlook, d.h. auf der rechten Seite sieht man den Inhalt der Nachrichten mit den wichtigen Daten, schön formatiert.
Auf der Skizze sieht man eine Tabelle mit allen Emails. Die Tabellen für SMS und MMS sehen ähnlich aus. Bei der Skizze nahmen wir einen konkreten Fall.
Die Skizze sieht sehr ähnlich aus wie das Programm. Das Programm hat zusätzlich noch Tabs, sowie mehrere Buttons und Links. Aber es gibt keine grossen Abweichungen.

Skizze: Hauptfenster



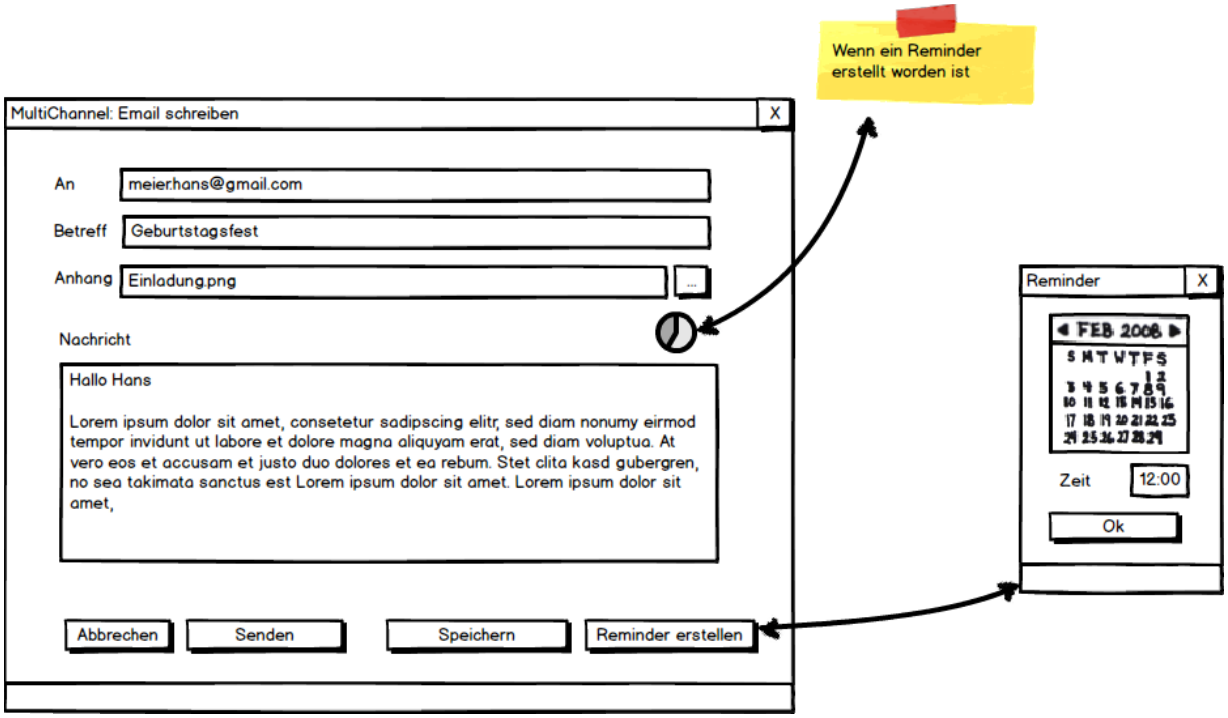
Hauptfenster im Programm



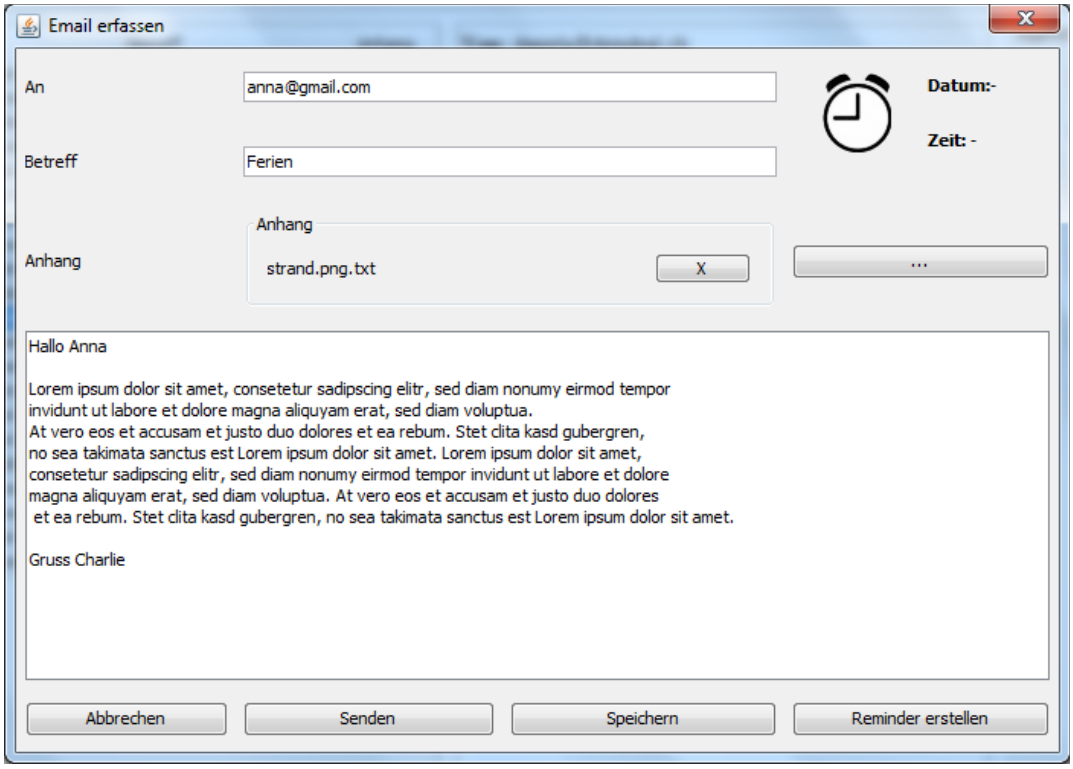
Nachricht erstellen

Wenn man eine Nachricht erstellen möchte, erscheint ein neues Fenster. Bei der Skizze beschränkten wir uns wiederum auf einen konkreten Fall, dem Mail. Da man einen Reminder erstellen kann, wird das Reminderfenster nach dem Klick auf den Reminder-Button angezeigt.
Nach dem Speichern befindet sich die Nachricht im Postfach «Entrüwfe».
Wenn man die Skizze mit dem Programm vergleicht, sieht man keine grossen Änderungen. Im Programm wird noch das ausgwählte Datum und die Zeit vom Reminder angezeigt.

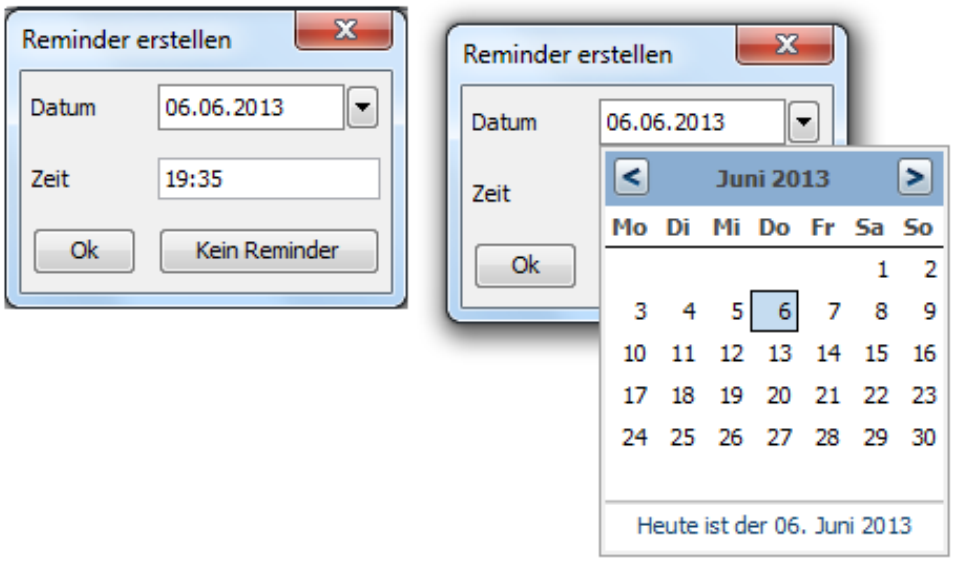
Skizze: Nachricht erstellen



Nachricht erstellen im Programm



Reminder



Die Skizzen haben sehr geholfen. Mit den Skizzen hat man nichts Wichtiges vergessen und man konnte sich nur auf das Programmieren konzentrieren.

Umsetzung

Verwendete Bibilotheken

Name	URL/Quelle	Verwendung
swingx-all-1.6.4.jar	https://swingx.java.net/	Erweiterung von Java Swing GUI toolkit
commons-codec-1.8.jar	http://commons.apache.org/	SHA1 Hash für Passwort
hamcrest-core-1.3.jar	http://www.junit.org	- JUnit
hamcrest-library-1.3.jar	http://www.junit.org	- JUnit
junit-4.11.jar	http://www.junit.org	- Test Suite
GridBagManager.java	Andreas Butti	Hilfsklasse für das GUI

Automatisierte Tests

Es gibt Unit Tests für die wichtigen Klassen und einen Integrationstest, der das Einloggen und den Versand der Nachrichten testet. Die Tests wurden parallel zur Entwicklung des Codes geschrieben und waren sehr hilfreich. Fehler konnten rasch behoben werden und die Hürden eines Refactorings wichtiger Klassen war deutlich tiefer.

JavaDoc

Die JavaDoc sind als Html im Projektordner.