Project – „DeskNote“

TBZ – AP11a

Von Niklas Liechti & Keerthikan T.

06.07.2013

Inhaltsverzeichnis

1. Einleitung: Projektbeschreibung

Unser Ziel ist es, eine Anwendungsprogramm „DeskNote“ zu entwickeln. DeskNote is eine To-Do list Applikation, welche durch Notizen geordnet ist. Alle Notizen und To-Dos sind in Online Datenbank gespeichert.

2. Thema/ Ziele der Applikation

- Notizen mit Online Datenbank (Notizen sind dem Benutzer überall erreichbar)
- Benutzerfreundliche Graphische Oberfläche
- Sicherheit (Passwörter mit BCrypt schlüssen)

3. Datenbank

Host: sql2.freemysqlhosting.net
Database name: sql210465
Database user: sql210465
Database password: kP4*wV1*

4. Use-Cases (Funktionale Anforderung)

Pre-Condition: MySQL-Verbindung ist erfolgreich.

4.1. Users

1. **Neu registrieren**

Vorbedingung	-
Beschreibung Anwendungsfall	Benutzer gibt folgende Daten ein. - Username - Passwort(2x) - Firstname, Surname, Address, Postal Code, Place
Nachbedingung	Die Daten sind im Datenbank gespeichert. Startseite von DeskNote wird anzeigt.
Ausnahmen	Wenn die "Username" bereits existiert, zeigt eine Fehlermeldung.

2. **Anmelden**

Vorbedingung	Benutzer hat bereits ein Konto.
Beschreibung Anwendungsfall	Benutzer gibt "Username & Password" ein.
Nachbedingung	Wenn die Anmeldung erfolgreich ist, zeigt die Startseite von DeskNote.
Ausnahmen	Falls die Benutzername & Passwort nicht stimmen, zeigt eine Fehlermeldung.

4.2. Notizen

1. Notizen erstellen

Vorbedingung	Benutzer hat bereits angemeldet.
Beschreibung Anwendungsfall	1. Benutzer klickt auf dem Button "New Note". 2. Benutzer gibt folgende Daten ein. - Title (mussfeld) - Content
Nachbedingung	Note ist erfolgreich gespeichert.
Ausnahmen	Titel fehlt → Fehlermeldung

2. Notizen bearbeiten

Vorbedingung	Benutzer hat bereits ein "Note" gewählt und klickt auf dem Button "Edit Note".
Beschreibung Anwendungsfall	Benutzer bearbeitet folgende Daten ein. - Titel (mussfeld) - Content
Nachbedingung	Beim Speichern wird der Note erfolgreich gespeichert.
Ausnahmen	Titel fehlt → Fehlermeldung

3. Notizen löschen

Vorbedingung	Benutzer hat bereits ein “Note” gewählt.
Beschreibung Anwendungsfall	Benutzer klickt auf dem Button „Delete Note“.
Nachbedingung	Note ist erfolgreich gelöscht und GUI wird aktualisiert.
Ausnahmen	-

4.3. To-Dos

1. To-Do hinzufügen erstellen

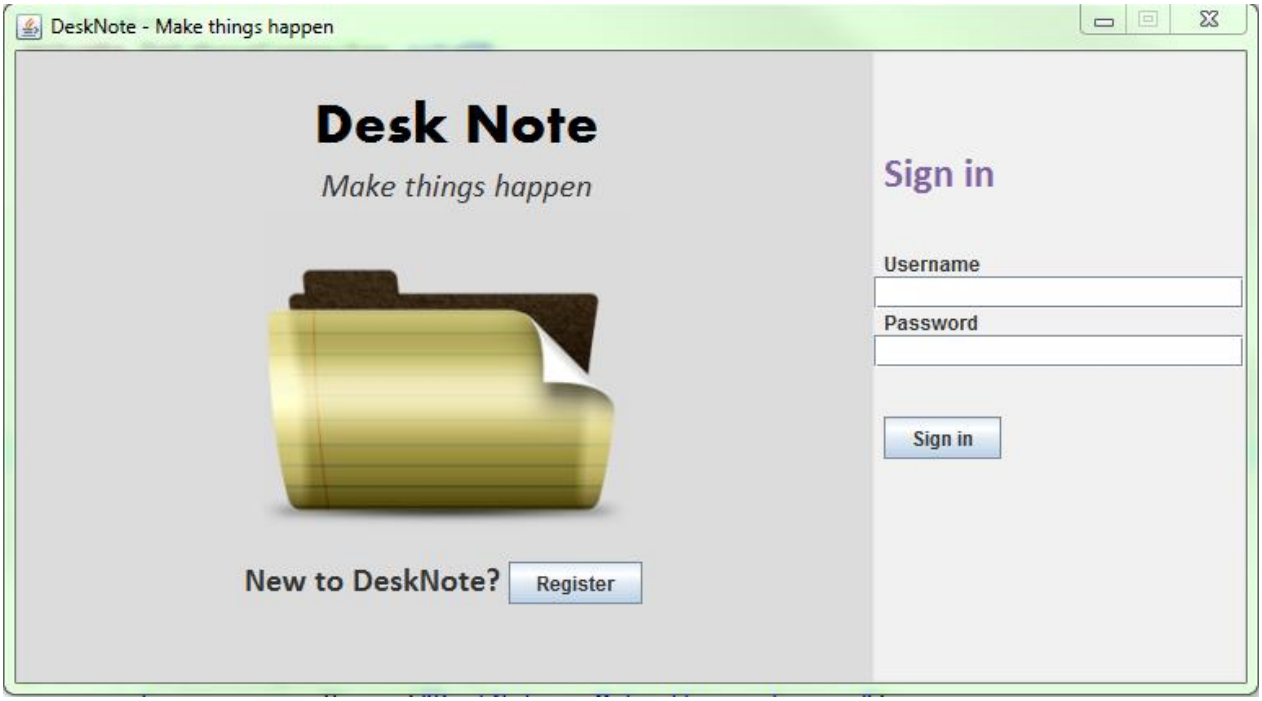
Vorbedingung	Benutzer hat bereits ein “Note” gewählt.
Beschreibung Anwendungsfall	1. Benutzer klickt auf dem Button “Add To-DO”. 2. Benutzer fügt den neuen To-DO Eintrag hinzu.
Nachbedingung	To-DO Eintrag ist gespeichert.
Ausnahmen	Beschreibung fehlt → To-DO Eintrag wird nicht gespeichert.

2. To-Do bearbeiten und löschen

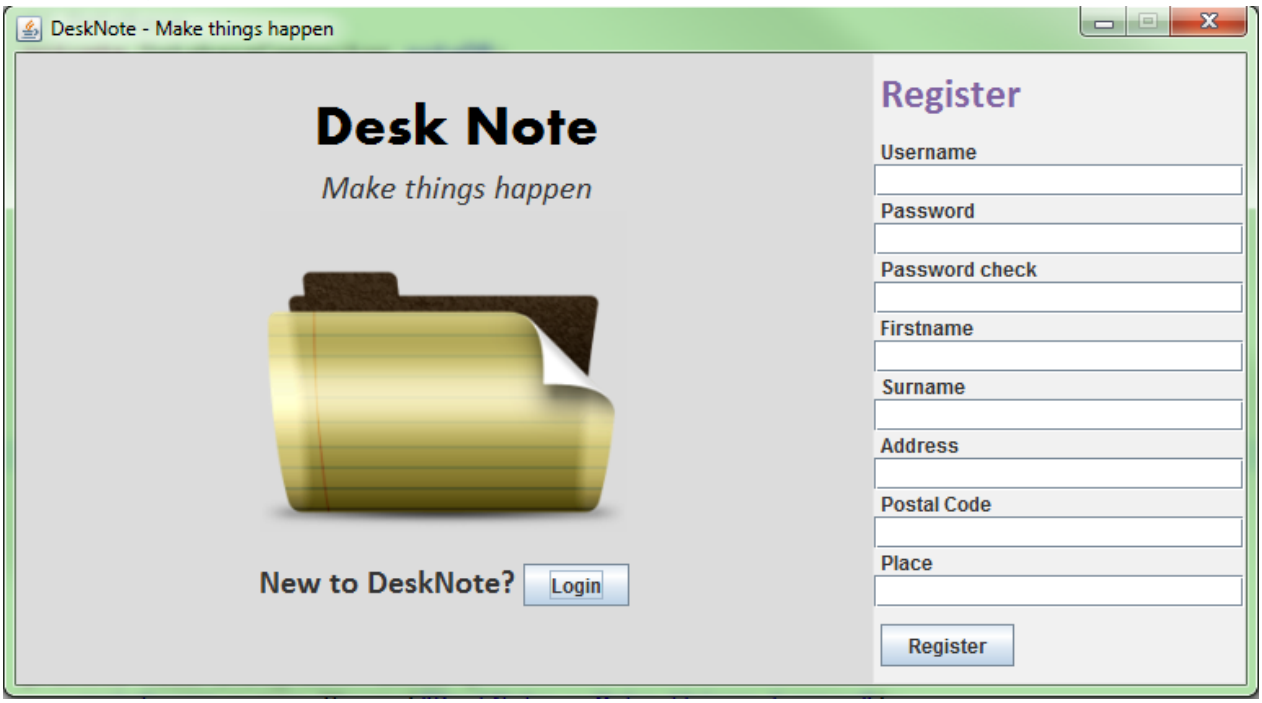
Vorbedingung	Benutzer hat bereits ein “Note” gewählt.
Beschreibung Anwendungsfall	1. Benutzer klickt auf dem Checkbox. 2. Checkbox wird umgeschaltet. Wenn er gewählt ist, wird er unterstrichen. 3. Right-Klick auf dem Checkbox. 4. Benutzer kann die Möglichkeit - Beschreibung Ändern - Eintrag löschen
Nachbedingung	Änderungen sind in Datenbank gespeichert.
Ausnahmen	-

5. Design Überblick

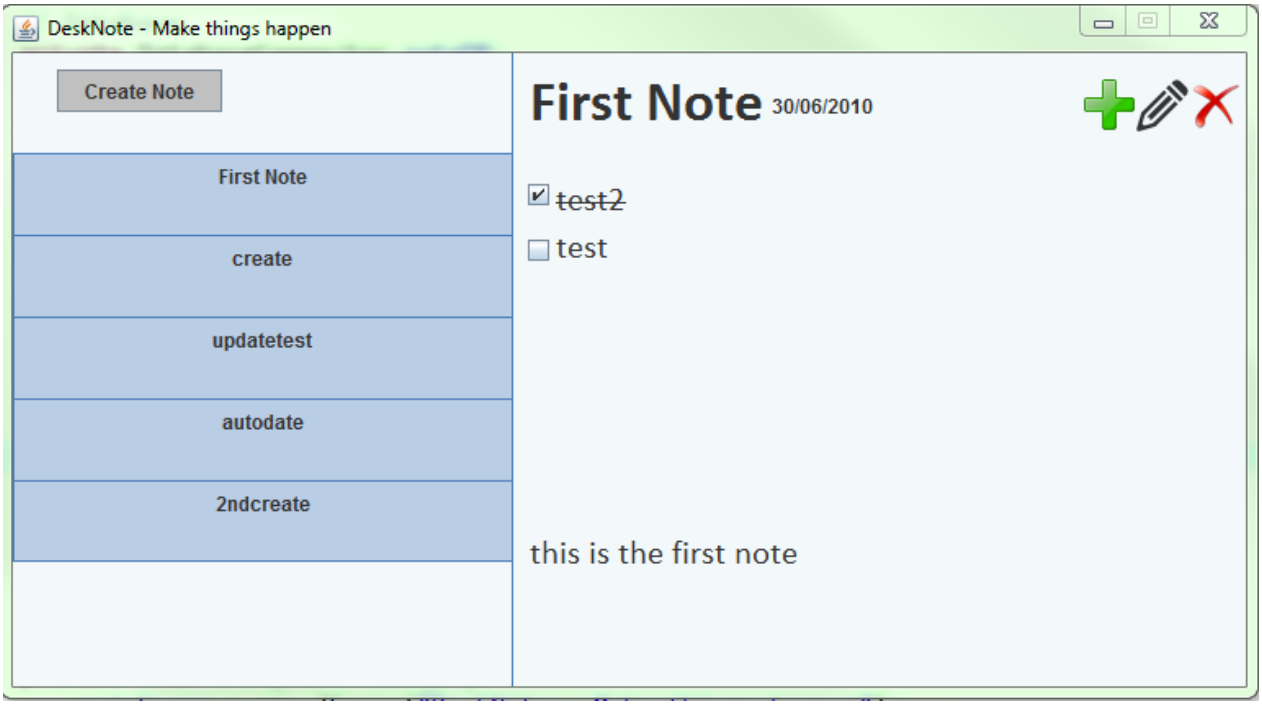
5.1. Login



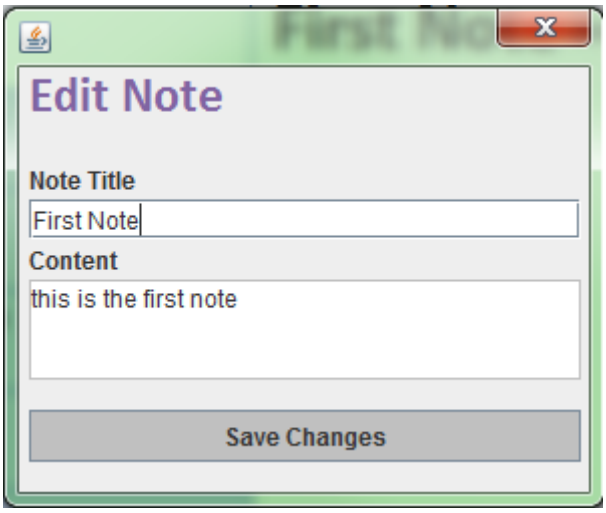
5.2. Register



5.3. DeskNote View

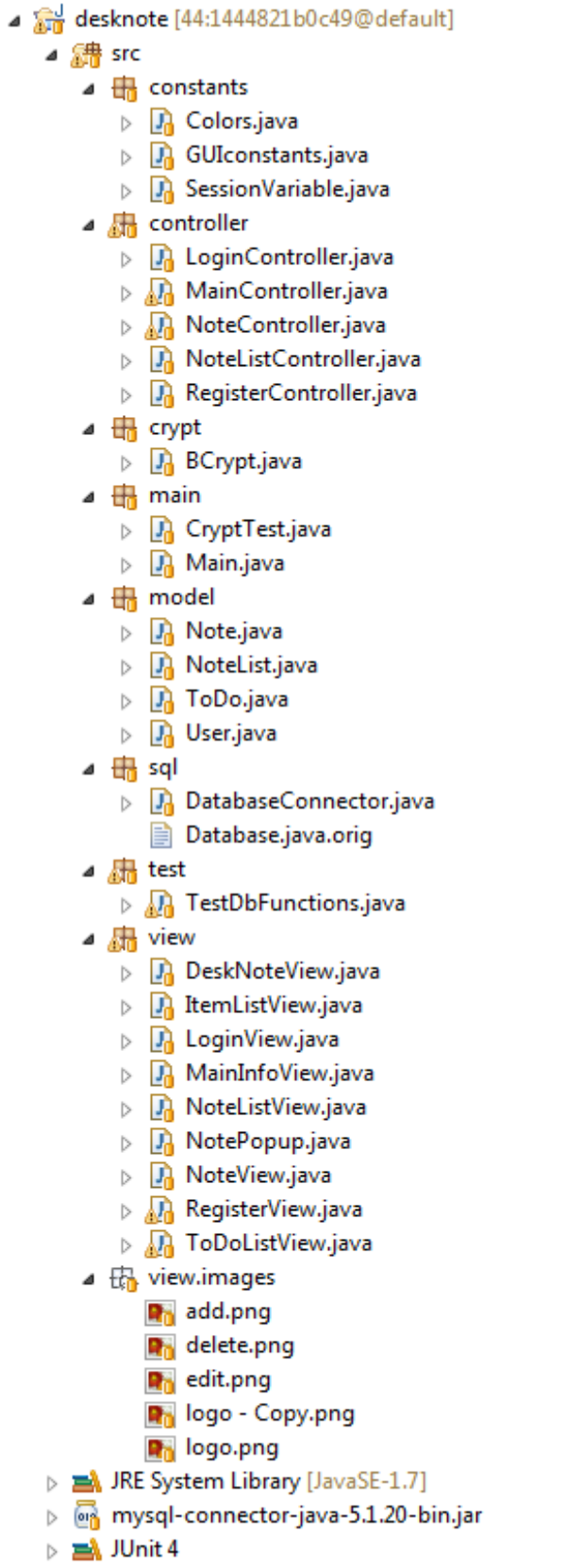


5.4. Create / Edit Note Popup



6. Klassendiagramm

6.1. Alle Klassen

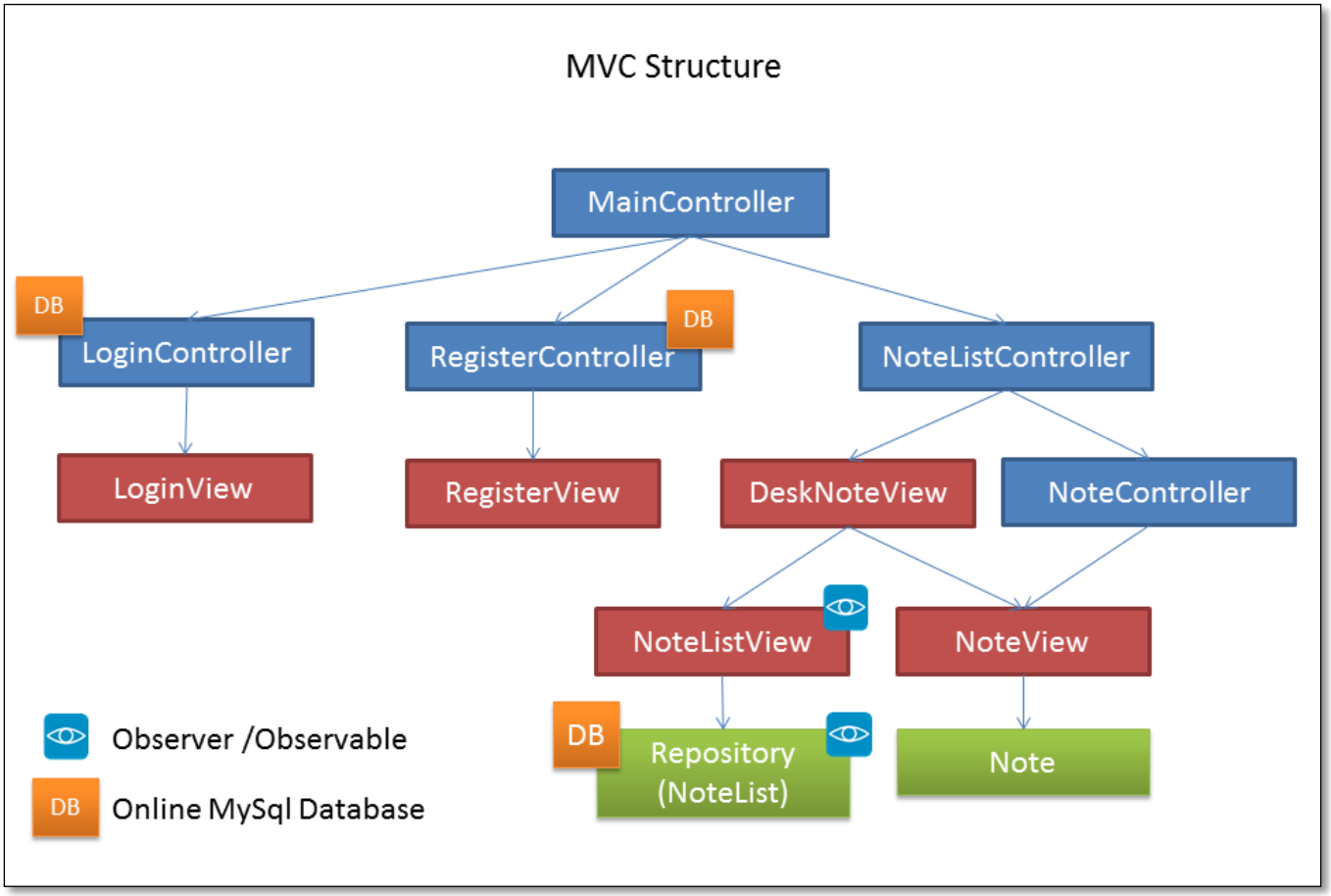


6.2. UML-Klassendiagramm

Siehe Beilage: UML- Klassendiagram.png

7. Entwurfsmuster

7.1. MVC und Observer/Observable Pattern

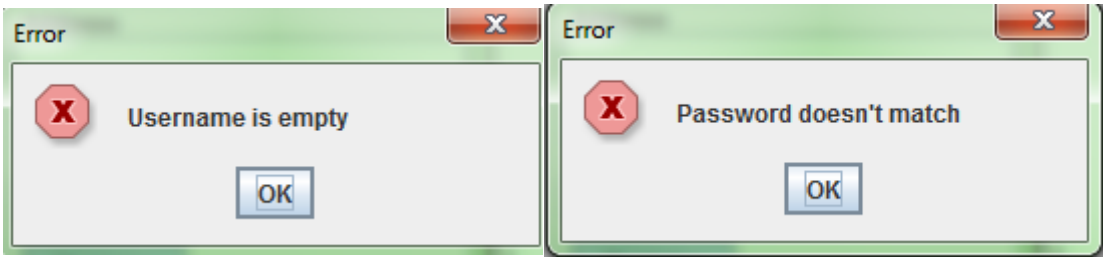


7.2. Datenbank mit Singleton Pattern

```
public class DatabaseConnector {

    private static DatabaseConnector INSTANCE = null;
    private Connection connection;

    public static DatabaseConnector getInstance() {
        if (INSTANCE == null) {
            INSTANCE = new DatabaseConnector();
        }
        return INSTANCE;
    }

    // Private constructor
    private DatabaseConnector() {
        mysql_connect();
    }
```

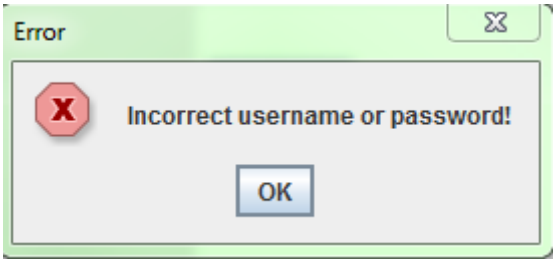
8. Testfälle

8.1. Registration



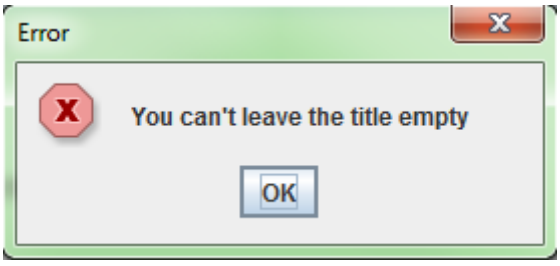
Step	Was wird vorausgesetzt? (Precondition)	Action	Was wird erwartet? (Postcondition)	Yes	No
#001	-Datenbank Festgelegt	Username nicht eingeben	Fehlermeldung	☐	☐
#002	-Datenbank festgelegt	Passwort nicht eingeben	Fehlermeldung	☐	☐
#003	-Datenbank festgelegt	Geben Sie zwei verschiedene Passwörter ein (Passwort & check)	Fehlermeldung	☐	☐
#004	-Datenbank festgelegt	Feld PLZ mit Buchstaben befüllen (z.B. abc)	Fehlermeldung	☐	☐
#005	-Datenbank festgelegt	Geben Sie als Username „admin“ ein	Fehlermeldung → Benutzername existiert bereits im DB	☐	☐
#006	-Datenbank festgelegt	Befüllen Sie die Felder mit gültigen Werte.	Information: „Erfolgreich registriert“ Danach wird „Login-Panel“ angezeigt.	☐	☐

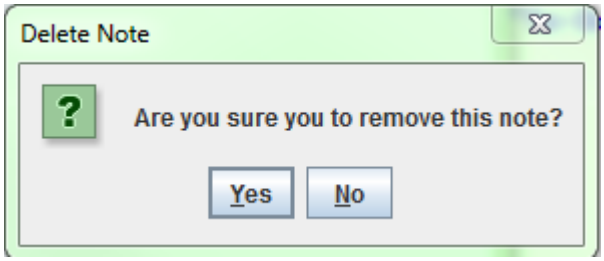
8.2. Login



Step	Was wird vorausgesetzt? (Precondition)	Action	Was wird erwartet? (Postcondition)	Yes	No
#101	-Datenbank Festgelegt	Username und Passwort falsch eingeben	Fehlermeldung	☐	☐
#102	-Benutzer bereits existiert.	Username und Passwort richtig eingeben	Startseite von DeskNote wird angezeigt.	☐	☐

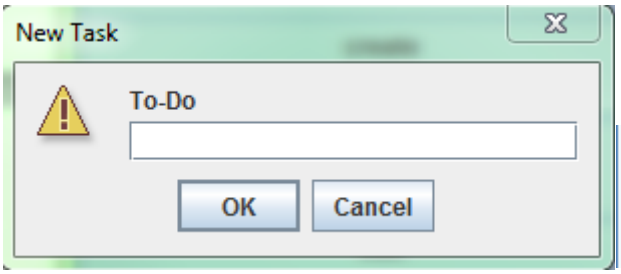
8.3. Notiz

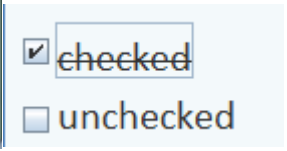


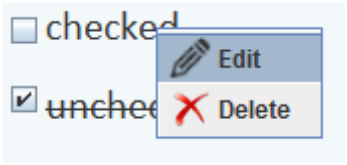


Step	Was wird vorausgesetzt? (Precondition)	Action	Was wird erwartet? (Postcondition)	Yes	No
#201	- Bereits angemeldet.	- Benutzer klickt auf „Create Note“. - Title nicht eingeben	Fehlermeldung	☐	☐
#202	- Bereits angemeldet.	- Benutzer klickt auf „Create Note“. - Title und Content(optional) eingeben	Notiz wird in DB erfolgreich gespeichert und in DeskNote-View angezeigt.	☐	☐
#203	- Notiz wurde schon erstellt	- Benutzer klickt auf „Edit Note“ - Benutzer löscht den Titel und speichert.	Fehlermeldung	☐	☐
#204	- Notiz wurde schon erstellt	- Benutzer klickt auf „Edit Note“ - Benutzer ändert den Title und speichert.	Notiz wird in DB erfolgreich gespeichert und in DeskNote-View aktualiesiert.	☐	☐
#205	- Notiz wurde schon erstellt.	- Benutzer klickt auf „Delete Note“. - Benutzer wählt „no“ beim ShowConfirmDialog.	Notiz wird nicht gelöscht.	☐	☐
#206	- Notiz wurde schon erstellt	- Benutzer klickt auf „Delete Note“. - Benutzer wählt „yes“ beim ShowConfirmDialog.	Notiz wird von Notizliste und auch vom DB gelöscht. DeskNote View wird entsprechend aktualiesiert.	☐	☐

8.4. To-Do (Checkbox)







Step	Was wird vorausgesetzt? (Precondition)	Action	Was wird erwartet? (Postcondition)	Yes	No
#301	-Notiz wurde schon erstellt	- Benutzer klickt auf „Add“ Button -Benutzer fügt den neuen To-DO Eintrag hinzu.	- To-Do wird in DB und repository hinzufügt.	☐	☐
#302	- To-Do wurde schon hinzufügt	- Benutzer wechselt den Status von Task(checked/unchecked)	- Checked To-Do wird gestrichelt anzegeit und DB aktualisiert. - Unchecked To-Do wird wieder normal angezeigt und DB aktualiesiert.	☐	☐
#303	- To-Do wurde schon hinzufügt	- Right-Click auf To-Do - Rename To-Do Content	- To-Do content wird geändert und DB aktualisiert.	☐	☐
#304	- To-Do wurde schon hinzufügt	- Right-Click auf To-Do - Delete To-Do	- To-Do wird von DB gelöscht	☐	☐

9. Projekt Zeitplan

Der Zeitplan enthält die Aufwandschätzung für

- Planung, Dokumentieren
- Realisieren (Vorbereiten, Programmieren, Funktionstest)
- Testing (JUnit & Manuel)

Anforderung		Keerthikan	Niklas	Aufwand Total in (h)
Allgemeines	Planung	4		
Dokumente	Pflichtenheft	1	1	
	Testfälle	1	1	
		0	0	
Realisieren	GUI	4	0	
	Database	0	4	
	Flowlogic	4	4	
	Model	2	2	
		0	0	
Testing	JUnit Test	0	2	
	Manuelle Test	2	2	
Total geschätzter Aufwand System …				**34**

10. Tools/Ablage

Das Projekt wird in BitBucket abgelegt und zwar unter:

https://bitbucket.org/nliechti/desknote

11. Anhang: Änderung an diesem Dokument

Version	Datum	Bearbeiter	Änderung
1.0	03.06.2013	Keerthikan T.	Use Cases, Tools/Ablage
2.0	05.07.2013	Keerthikan T.	Design Überblick, Entwurfsmuster
3.0	06.07.2013	Keerthikan T.	Use Cases, Testfälle
4.0	06.07.2013	Niklas Liechti	Einleitung, Ziele der Applikation, Zeitplan