

PrakMan: Der Praktikums-Manager

Philipp Rollwage Christian Lins Kai Ritterbusch
Andreas Depping

17. September 2007

Inhaltsverzeichnis

1 Programmkenndaten	4
1.1 Programmname	4
1.2 Versionsnummer und Freigabedatum	4
1.3 Mitarbeiter	4
1.4 Pflichtenheft	4
1.4.1 Zielbestimmung	5
1.4.2 Produkteinsatz	6
1.4.3 Schnittstellen	6
1.4.4 Produktfunktionen	7
1.4.5 Produktdaten	13
1.4.6 Produktleistungen	13
1.4.7 Benutzungsoberfläche	13
1.4.8 Qualitätszielbestimmungen	14
1.4.9 Globale Testszenarien und Testfälle	14
1.4.10 Entwicklungsumgebung	14
1.4.11 Ergänzungen	14
1.4.12 Begriffserklärung	14
1.5 Lösungsverfahren und Algorithmen	14
1.6 Hardware- und Softwareanforderungen	15
1.7 Benötigte Dateien und Directories	15
2 Programmaufbau und Dokumentation	16
2.1 Programmstruktur	16
2.1.1 Datenflussdiagramm	16
2.1.2 Gliederung der Packages	17
2.2 Klassenhierarchiediagramm	17
2.3 Klassendokumentation	18
2.4 Programmablauf	19
2.5 Anwendungsgrenzen und Fehlersituationen	25
2.6 Benutzeroberflächenhierarchie	25
2.7 Funktionsbeschreibung der Benutzeroberflächen und deren Komponenten	27
2.8 Dateiaufbau aller Anwenderdateien mit DD- Beschreibung und Funktion .	28
2.8.1 Export von Studentendaten:	28
2.8.2 Export von Dozentendaten:	28
2.8.3 Export von Veranstaltungsdaten:	28

3 Datenbank	30
3.1 Dokumentation der Datenbank	30
3.2 Entity- Relationship- Modell (ERM) der Datenbank	30
3.3 Prüfung des Relationenschemas auf Einhaltung der Normalisierung	30
3.3.1 1te Normalform	30
3.3.2 2te Normalform	30
3.3.3 3te Normalform	31
3.4 Funktionsbeschreibung der Formatprogramme	31
3.5 Funktionsbeschreibung der Reportprogramme	31
3.6 Funktionsbeschreibung der SQL- /ESQL- /JDBC- Programme	31
4 Installation und Test	35
4.1 Testverfahren und Testdokumente mit automatischen Tests	35
4.1.1 Individuelle Komponenten-Tests	35
4.1.2 Tests des Komponenten-Zusammenspiels	35
4.1.3 Performance-Test	35
4.2 Übergabedokument	36
5 Programmbetrieb	37
5.1 Bedienungsanleitung	37
5.1.1 Startbildschirm	37
5.1.2 Hauptbildschirm	37
5.1.3 Eine Veranstaltung: Teilnehmer	38
5.1.4 Eine Veranstaltung: Projekte	38
5.1.5 Eine Veranstaltung: Termine	39
5.1.6 Eine Veranstaltung: Gruppen	39
5.1.7 Ein Tutor: Betreute Veranstaltungen	39
5.1.8 Ein Student: Gewählte Veranstaltungen	40
5.1.9 Ein Student: Noten	40
5.2 Datensicherung	46

1 Programmkenndaten

1.1 Programmname

Das Programm trägt den Namen seiner Funktionsbestimmung: Praktikumsmanager, abgekürzt "PrakMan".

1.2 Versionsnummer und Freigabedatum

Das Programm liegt aktuell in der Version 1.0 vor und ist seit dem 17.09.2007 zur Nutzung freigegeben.

1.3 Mitarbeiter

Mitarbeiter, die zu gleichen Teilen zur Fertigstellung der aktuell vorliegenden Version beigetragen haben sind:

- | | |
|--|---|
| • Andreas Depping
Bramscher Straße 158
49088 Osnabrück | Andreas.Depping@fh-osnabrueck.de
Telefon: 0541/98252103
Mat-Nr. #325677 |
| • Christian Lins
Limberger Straße 102
49080 Osnabrück | Christian.Lins@fh-osnabrueck.de
Telefon: 0541/3342152
Mat-Nr. #322141 |
| • Kai Ritterbusch
Beethovenstraße 8
49191 Belm | Kai.Ritterbusch@fh-osnabrueck.de
Telefon: 05406/819606
Mat-Nr. #327879 |
| • Philipp Rollwage
Neulandstraße 6
49084 Osnabrück | Philipp.Rollwage@fh-osnabrueck.de
Telefon: 0541/3247040
Mat-Nr. #329302 |

1.4 Pflichtenheft

Pflichtenheft zur Erstellung einer Software zur Praktikaverwaltung. Das Projekt Praktikaverwaltung beinhaltet die Planung und Realisierung eines Java-Programmsystems zur Verwaltung von Lehrveranstaltungen an der FH Osnabrück.

1.4.1 Zielbestimmung

Es sollen Praktika (zum Beispiel zur Lehrveranstaltung Software Engineering) verwaltet werden. Dazu ist es notwendig, ca. 60 Studierende mit Hilfe einer GUI zu erfassen, in Praktikumsgruppen zu jeweils ca. 20 aufzuteilen und deren Anwesenheit in einer der ca. 15 Veranstaltungen im Semester zu erfassen. Außerdem sollen die Studierenden Projekte zugeordnet werden können. Dazu müssen ca. 6 Projekte zum Teil mehrfach angelegt werden können, denen jeweils ca. 1-8 Studierende zugeordnet werden. Außerdem sollen den Studierenden Hausarbeiten zugeordnet werden können. Dazu müssen ca. 4-8 Hausarbeiten zum Teil mehrfach angelegt werden können, denen jeweils ca. 1-8 Studierende zugeordnet werden. Die Anwendung soll unterschiedliche Listen in unterschiedlicher Sortierung drucken können:

- Gesamtliste aller Studierenden der Veranstaltung
- Liste der Studierenden je Praktikum
- Liste mit Anwesenheit (gesamt und je Praktikum)
- Notenlisten (“bestanden” für Projekt, Note für Hausarbeit)

Das Programm muss über eine persistente Datenhaltung verfügen, d.h. die Daten müssen in einer Datenbank oder in CSV-Dateien gehalten werden können. Die CSV-Dateien sollen mit Excel eingelesen werden können. Von jedem Studierenden werden die üblichen Daten erfasst (Name, Mat.-Nr, Semester, Studiengang, Email). Wünschenswert wäre eine Datenübernahme aus den von StudIP exportierten Teilnehmerdaten. Insgesamt soll eine benutzerfreundliche und leicht erweiterbare Anwendung zur Unterstützung von Lehrenden entstehen.

Musskriterien

- Drucken von Anwesenheitslisten, Notenlisten
- Wechsel von Studierenden zwischen Praktika
- Möglichkeit, Studierende aus Praktika zu entfernen
- Sortierungen nach unterschiedlichen Kriterien

Wunschkriterien

- Schnittstelle zu StudIP: Import der dort exportierten Teilnehmerdaten
- Verwaltung eines Fotos je Teilnehmer

Abgrenzungskriterien

- Das Programm stellt keinen Ersatz für StudIP dar, sondern eine Unterstützung.
- Es soll keine direkte Anbindung an die StudIP-Datenbank geben.

1.4.2 Produkteinsatz

Das Programm soll von Professoren oder Tutoren, welche mit der Leitung von Praktikumsgruppen betraut sind, genutzt werden.

Anwendungsbereiche

Das Programm soll es Professoren bzw. Tutoren erleichtern, ihre Praktikumsgruppen einzuteilen, Aufgaben und Noten zu verteilen und auf all diese Informationen in einer übersichtlichen Oberfläche zugreifen zu können. Zur Haltung der Daten kann dabei eine eigene, lokale Datenbank genutzt werden, oder auch eine externe, auf die mehrere Tutoren mit "PrakMan" zugreifen und gemeinsam ihre Daten verwalten.

Zielgruppen

Das Programm soll von Professoren oder wissenschaftlichen Mitarbeitern, primär der Fachhochschule Osnabrück, eingesetzt werden. Möglich wäre aber auch ein Betrieb in anderen Hochschulen. Vorausgesetzt werden grobe Kenntnisse im Bereich von Datenbanken (Zugangsdaten) oder Allgemeinwissen über Programme (Mögliche Optionen über Rechtsklick erreichbar etc.).

Betriebsbedingungen

Das Programm benötigt keine gesonderte Beaufsichtigung, sondern kann direkt auf jedem PC, der die auf Seite 15 genannten Voraussetzungen erfüllt, genutzt werden.

Software

Siehe Punkt 1.6 auf Seite 15.

Hardware

Das Programm erfordert keine besondere Hardware.

Orgware

Um den zu organisierenden Veranstaltungen Studenten hinzufügen zu können, müssen Daten über diese bekannt sein. Dazu gehören die Matrikelnummer, Vorname, Nachname und eine E-Mail-Adresse. Diese Daten sollten im Voraus entweder auf Papier oder per CSV (für den Import) organisiert werden.

1.4.3 Schnittstellen

Das Programm bietet eine JDBC-Schnittstelle, über die unterschiedliche Datenbanken angebunden werden können. "PrakMan" unterstützt momentan HSQL und PostgreSQL.

1.4.4 Produktfunktionen

/F0001/Verbindung zur Datenbank

Voraussetzung: Keine

Ablauf:

1. Der Benutzer startet das Programm und wählt einen Datenbanktypen aus: Lokal (HSQL) oder PostgreSQL.

Alternativen:

- 1a. Die Verbindung schlägt fehl. Das kann an einer nicht vorhandenen Internetverbindung oder einem nicht gestarteten Datenbank-Server liegen.

/F0002/Erstellen einer Veranstaltung

Voraussetzung:

1. Der Benutzer hat PrakMan gestartet und die Verbindung zu einer Datenbank erfolgreich hergestellt.

Ablauf:

1. Der Benutzer klickt im Baum mit der rechten Maustaste auf den Ast “Veranstaltungen” und wählt “Hinzufügen” aus.
2. Ein Professor wird ausgewählt, der die Veranstaltung betreuen soll.
3. Die neue Veranstaltung ist erstellt, nun können weitere Einstellungen getroffen werden.

Alternativen:

- 1a. Ein Datenbankfehler tritt auf. Die Verbindung zur Datenbank sollte überprüft werden.

/F0003/Hinzufügen von Studenten zu einer Veranstaltung

Voraussetzung:

1. Der Benutzer hat PrakMan gestartet und die Verbindung zu einer Datenbank erfolgreich hergestellt.
2. Es existiert mindestens eine Veranstaltung.

Ablauf:

1. Der Benutzer öffnet die Ansicht der Veranstaltung über einen Doppelklick auf Selbige.
2. Über den Plus-Button am unteren Rand des Fensters wird ein neues Panel geöffnet, über welches Studenten ausgewählt und durch einen Druck auf “Ok” der Veranstaltung hinzugefügt werden.

Alternativen:

- 2a. Es existieren keine Studenten. In diesem Fall müssen erst neue Studenten manuell erstellt oder über CSV eingelesen werden.

/F0004/Einteilung der Studenten in Gruppen**Voraussetzung:**

1. Der Benutzer hat PrakMan gestartet und die Verbindung zu einer Datenbank erfolgreich hergestellt.
2. Es existiert mindestens eine Veranstaltung, deren Gruppen-Ansicht geöffnet ist.

Ablauf:

1. Ein Klick auf den Plus-Button erstellt eine neue Gruppe. Über einen Doppelklick auf ihren Namen kann deren Beschreibung editiert werden.
2. Der Benutzer wechselt in die Teilnehmer-Ansicht und markiert dort über einfaches Ziehen mit der Maus, oder Shift/Strg-Klick einen oder mehrere Studenten, die zusammen eine Gruppe bilden sollen. Über einen rechtsklick auf die Tabelle kann diesen markierten Studenten nun eine Gruppe zugewiesen werden.

Alternativen:

- 1a. Ein Datenbankfehler tritt auf. Die Verbindung zur Datenbank sollte überprüft werden.
- 2a. Es existieren noch keine Studenten in der Veranstaltung. In diesem Fall, können sie einfach über den Plus-Button in der Teilnehmer-Ansicht hinzugefügt werden.

/F0005/Ein Projekt hinzufügen**Voraussetzung:**

1. Der Benutzer hat PrakMan gestartet und die Verbindung zu einer Datenbank erfolgreich hergestellt.
2. Es existiert mindestens eine Veranstaltung, deren Projekte-Ansicht geöffnet ist.

Ablauf:

1. Ein Klick auf den Plus-Button erstellt ein neues unbenanntes Projekt.
2. Über einen Doppelklick auf Selbiges, können diesem Projekt Teilnehmer hinzugefügt und außerdem dessen Beschreibung editiert werden. Zusätzlich kann eine "Deadline" gesetzt werden, zu welcher dieses Projekt beendet sein muss.

Alternativen:

- 1a. Ein Datenbankfehler tritt auf. Die Verbindung zur Datenbank sollte überprüft werden.
- 2a. Es existieren keine Studenten, die zugeordnet werden können. In diesem Fall, können sie einfach über den Plus-Button in der Teilnehmer-Ansicht hinzugefügt werden.

/F0006/Einen Termin hinzufügen**Voraussetzung:**

1. Der Benutzer hat PrakMan gestartet und die Verbindung zu einer Datenbank erfolgreich hergestellt.
2. Es existiert mindestens eine Veranstaltung, deren Termine-Ansicht geöffnet ist.

Ablauf:

1. Ein Klick auf den Plus-Button öffnet ein Auswahlfenster, in dem ein Termin für die Veranstaltung ausgewählt werden kann. Zuerst sollte eine Uhrzeit angegeben werden. Ein Klick auf einen Tag im Kalender fügt diesen Termin dann der Veranstaltung hinzu.

Alternativen: Keine**/F0007/Den Tutor einer Veranstaltung ändern****Voraussetzung:**

1. Der Benutzer hat PrakMan gestartet und die Verbindung zu einer Datenbank erfolgreich hergestellt.
2. Es existiert mindestens eine Veranstaltung.

Ablauf:

1. Über einen Doppelklick auf eine Veranstaltung links im Baum, öffnet sich deren Hauptansicht. Nun kann über die Dropdown-Box im oberen rechten Bereich einer der in der Datenbank registrierten Professoren ausgewählt werden.
2. Über den Button “Speichern” (oben links) wird die Information in die Datenbank übernommen.

Alternativen:

- 1a. Statt einer Veranstaltung wird im Baum ein Lehrender ausgewählt. Bei diesem kann dann über den Plus-Button eine Veranstaltung zugeordnet werden.
- 1b. Es ist kein weiterer Professor vorhanden. In dem Fall müssen manuell oder per CSV-Import neue Lehrende hinzugefügt werden.
- 2a. Ein Datenbankfehler tritt auf. Die Verbindung zur Datenbank sollte überprüft werden.

/F0008/Einen neuen Dozenten hinzufügen

Voraussetzung:

1. Der Benutzer hat PrakMan gestartet und die Verbindung zu einer Datenbank erfolgreich hergestellt.

Ablauf:

1. Der Benutzer klickt im Baum mit der rechten Maustaste auf den Ast “Lehrende” und wählt “Hinzufügen” aus.
2. Im auftauchenden Panel werden die benötigten Daten “Vorname” und “Nachname” eingetragen.
3. Ein Druck auf den Button “Speichern” überträgt die Informationen in die Datenbank, der neue Dozent ist erstellt.

Alternativen:

- 3a. Ein Datenbankfehler tritt auf. Die Verbindung zur Datenbank sollte überprüft werden.

/F0009/Einen neuen Studenten hinzufügen

Voraussetzung:

1. Der Benutzer hat PrakMan gestartet und die Verbindung zu einer Datenbank erfolgreich hergestellt.

Ablauf:

1. Der Benutzer klickt im Baum mit der rechten Maustaste auf den Ast “Studenten” und wählt “Hinzufügen” aus.
2. Im auftauchenden Eingabefeld wird die 6-stellige Matrikelnummer des Studenten angegeben. Ein Druck auf “OK” lässt ein neues Panel erscheinen.
3. Im erschienenen Panel werden die benötigten Daten “Vorname”, “Nachname” und “E-Mail” eingetragen.
4. Ein Druck auf den Button “Speichern” überträgt die Informationen in die Datenbank, der neue Student ist erstellt.

Alternativen:

- 4a. Ein Datenbankfehler tritt auf. Die Verbindung zur Datenbank sollte überprüft werden.

/F0010/Daten in CSV exportieren**Voraussetzung:**

1. Der Benutzer hat PrakMan gestartet und die Verbindung zu einer Datenbank erfolgreich hergestellt.

Ablauf:

1. Der Benutzer klickt im Baum mit der rechten Maustaste auf einen der Äste “Lehrende”, “Studenten” *oder* direkt auf eine bestimmte Veranstaltung und wählt “Export”.
Wird ein Ast ausgewählt, werden nur die Informationen der Objekte in diesem Ast in der CSV-Datei gespeichert. Wird eine einzelne Veranstaltung ausgewählt, werden deren gesamte Informationen detailliert gespeichert. Dies beinhaltet Gruppen, Aufgaben, Teilnehmer und Termine.
2. Der Benutzer wählt im auftauchenden Dialog einen Dateinamen. Ein Druck auf “Speichern” beendet die Aktion und erstellt die CSV-Datei mit den gewünschten Informationen.

Alternativen:

- 1a. Es gibt noch keine Veranstaltungen, Lehrenden oder Studenten. In dem Fall wird auch nichts exportiert.
- 2a. Der Benutzer hat kein Schreibrecht, das Speichern schlägt fehl. In diesem Fall sollten Sie Ihre Benutzerrechte auf dem PC überprüfen.

/F0011/CSV-Daten importieren

Voraussetzung:

1. Der Benutzer hat PrakMan gestartet und die Verbindung zu einer Datenbank erfolgreich hergestellt.

Ablauf:

1. Der Benutzer klickt im Baum mit der rechten Maustaste auf einen der Äste “Lehrende”, “Studenten” *oder* direkt auf eine bestimmte Veranstaltung und wählt “Import”.

Wird ein Ast ausgewählt werden nur Listeninformationen der Objekte in diesen Ast übernommen. Wird eine einzelne Veranstaltung ausgewählt, wird auch eine CSV-Datei mit detaillierten Informationen über diese Veranstaltung benötigt. Dies beinhaltet Gruppen, Aufgaben, Teilnehmer und Termine.

2. Der Benutzer wählt im auftauchenden Dialog einen Dateinamen. Ein Druck auf “Öffnen” beendet die Aktion und erstellt die Objekte im Baum respektive füllt die gewählte Veranstaltung mit detaillierten Informationen.

Alternativen:

- 1a. Es gibt noch keine Veranstaltung, die direkt ausgewählt werden kann. In diesem Fall müssen Sie zuerst eine Veranstaltung erstellen.
- 2a. Ihre CSV-Datei ist falsch formatiert und wird vom System nicht verstanden.

/F0012/Ein Projekt benoten

Voraussetzung:

1. Der Benutzer hat PrakMan gestartet und die Verbindung zu einer Datenbank erfolgreich hergestellt.
2. Es existiert mindestens eine Veranstaltung, die zumindest ein Projekt enthält und deren Projekte-Ansicht geöffnet ist.

Ablauf:

1. Ein Doppelklick auf das Projekt öffnet ein Auswahlfenster, in dem Ausgabe -und Abgabetermin festgelegt, Studenten zum Projekt hinzugefügt und dessen Beschreibung editiert werden kann.
2. Ein Doppelklick auf das Notenfeld eines Studenten öffnet eine Eingabemaske, in der eine neue Note für diesen Studenten festgelegt werden kann.

Alternativen: Keine

1.4.5 Produktdaten

/D0001/Studentendaten

Vorname, Nachname, E-Mail-Adresse und Matrikelnummer von Studenten. Des weiteren Projekte und Termine, an denen Studenten teilnehmen bzw. teilgenommen haben sowie die Benotung einzelner Projekte.

/D0002/Dozentendaten

Vorname und Nachname eines Dozenten.

/D0003/Veranstaltungsdaten

Name, Beschreibung, Teilnehmer, Projekte und Termine einer Veranstaltung.

1.4.6 Produktleistungen

/L0001/Komfort

Das Erstellen von Veranstaltungen, Füllen mit Studenten und Verteilen von Aufgaben soll möglichst einfach und schnell möglich sein. Das Programm soll die Arbeit erleichtern und nicht verkomplizieren.

1.4.7 Benutzungsoberfläche

Bildschirmlayout

Das Bildschirmlayout soll die Daten übersichtlich darstellen und einen möglichst schnellen und effektiven Zugriff ermöglichen. Es soll möglich sein, mehrere Informationen gleichzeitig zu öffnen.

Drucklayout

Das Drucklayout soll möglichst übersichtlich Informationen über die Teilnehmer einer Veranstaltung und deren Benotung liefern.

Tastaturlayout

Eine gesonderte Hotkey-Nutzung der Tastatur ist nicht vorgesehen. Das Programm soll es dafür ermöglichen, mit der Maus möglichst einfach von einer Information zur nächsten damit zusammenhängenden zu kommen.

Dialoglayout

Die Dialoge sollen möglichst viele Icons enthalten, die ihren Zweck direkt deutlich machen. Dazu sollen Tooltips existieren, die auch bei einer Unklarheit weiterhelfen.

1.4.8 Qualitätszielbestimmungen

- | | |
|--------------------|------------------|
| • Funktionalität: | Sehr wichtig |
| • Zuverlässigkeit: | Wichtig |
| • Benutzbarkeit: | Relativ wichtig |
| • Effizienz: | Unwichtig |
| • Übertragbarkeit: | Vernachlässigbar |

1.4.9 Globale Testszenarien und Testfälle

/T0001/Eingabe einer bereits vorhandenen oder negativen Matrikel-Nr.

Die Eingabe wird mit einer Fehlermeldung zurückgewiesen.

/T0002/Angeben einer ungültig formatierten CSV-Datei beim Import

Die Datei wird als falsch formatiert zurückgewiesen.

1.4.10 Entwicklungsumgebung

- Java 1.5
- Eclipse
- Subversion zur verbesserten Teamarbeit

1.4.11 Ergänzungen

Keine Ergänzungen.

1.4.12 Begriffserklärung

Vollständige Erklärung wichtiger Begriffe, siehe Abschnitt 5.2 (S.46).

1.5 Lösungsverfahren und Algorithmen

Es werden keine außergewöhnlichen Algorithmen verwendet.

1.6 Hardware- und Softwareanforderungen

Zum Betrieb des Programms erforderlich sind

- Java Version 1.5 oder höher
- Für Mehrbenutzer-Verwendung: Eine externe Datenbank (Postgres)
- Bei lokaler Nutzung: Speicherplatz für die zu haltenden Daten

1.7 Benötigte Dateien und Directories

Zum Start des Programms wird allein die Datei “prakman.jar” benötigt, welche das Programm sowie alle Treiber für die unterstützten Datenbanken enthält.

2 Programmaufbau und Dokumentation

2.1 Programmstruktur

Die folgenden Abschnitte präsentieren, oftmals in graphischer Form, den Aufbau sowie die Funktionsweise und den Fluss der Daten in “PrakMan”.

2.1.1 Datenflussdiagramm

Sämtliche Daten werden in “PrakMan” in einer Datenbank verwaltet. Alle Informationen gehen dort hinein und kommen von dort zurück. Diese verwaltet somit immer den aktuellen Stand der Informationen. Abbildung 2.1 (S.16) veranschaulicht den Datenfluss in einem Diagramm.

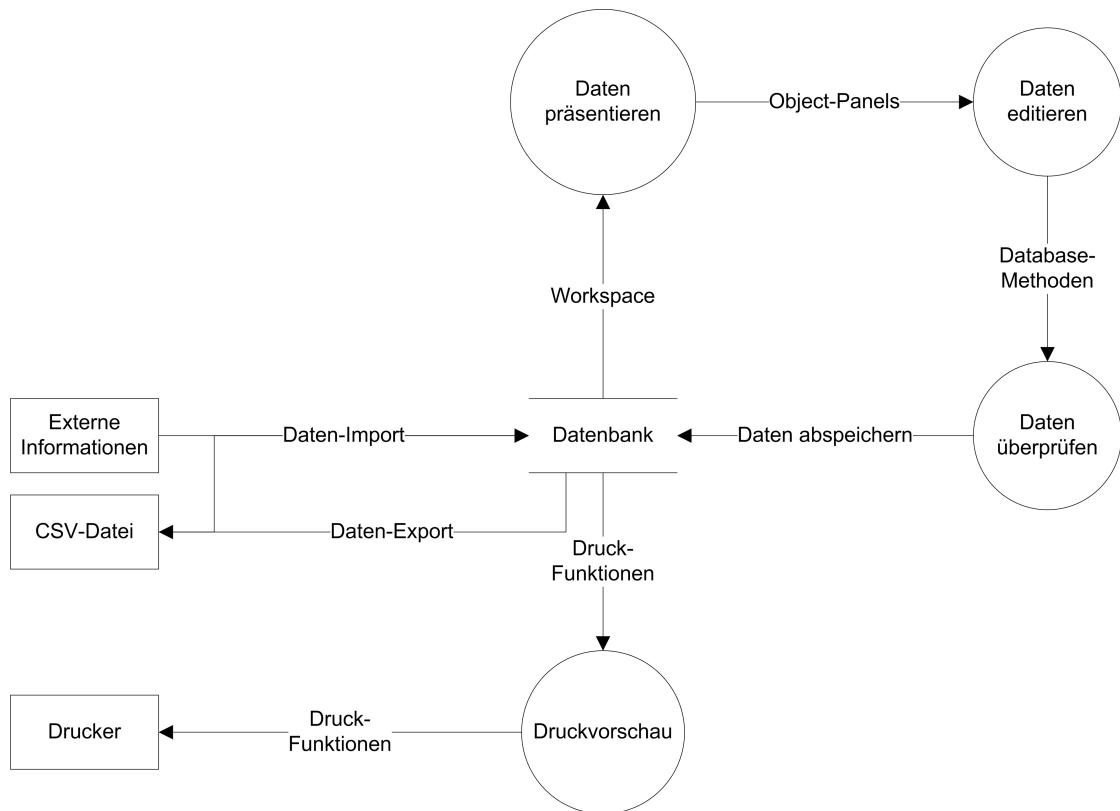


Abbildung 2.1: Datenflussdiagramm von “Prakman”

2.1.2 Gliederung der Packages

Abbildung 2.2 (S.17) stellt die Gliederung der Pakete in “PrakMan” dar. Das Grundpaket ist hierbei in hellblau dargestellt, darunter folgen unterschiedliche Farben für jeweils andere Bereiche. Die Farben sind je nach weiterer Spezialisierung des Pakets in steigender Intensität gewählt.

Hier ist - wie auch im Klassenhierarchiediagramm unter Punkt 2.3 (S.18) - deutlich die Aufteilen von “PrakMan” zu erkennen:

- Das Paket “prakman.io” sorgt für den Informationsfluss. Es speichert die Daten in der Datenbank und stellt sie wiederum bereit.
- Das Paket “prakman.model” stellt Objekte bereit, die die Informationen aus der Datenbank logisch kapseln und dann zur Weiterverarbeitung genutzt werden können.
- Das Paket “prakman.view” stellt die Daten für den Benutzer dar, sorgt aber auch für die Interaktion mit dem Benutzer.
- Das Paket “resource” ist lediglich ein Speicherort für Icons, könnte jedoch noch auf weitere Resourcen wie z.B. abgelegte Fotos etc. erweitert werden.

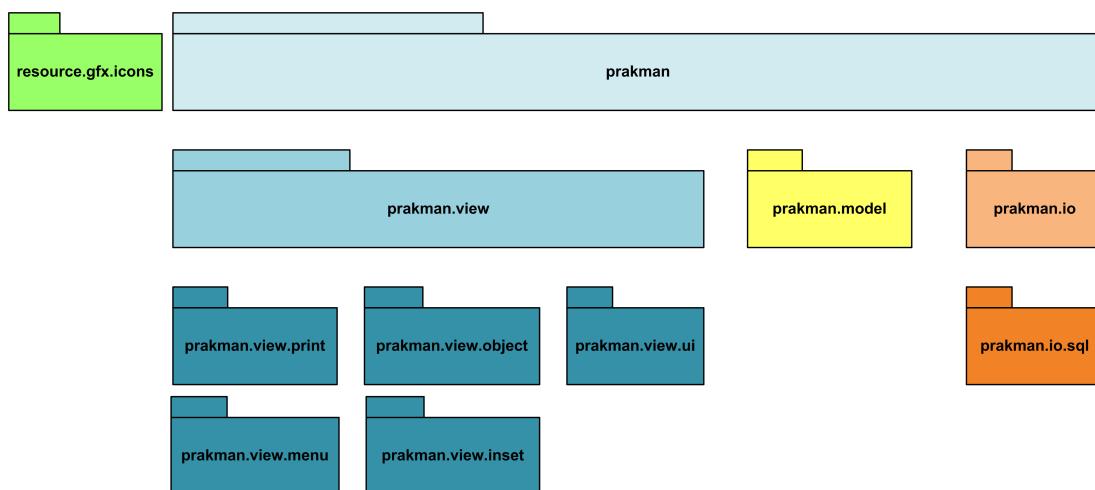


Abbildung 2.2: Paketgliederung von “Prakman”

2.2 Klassenhierarchiediagramm

Abbildung 2.3 (S.18) stellt die Schichtung des Programms und das Zusammenspiel der wichtigsten Klassen von “PrakMan” in Form eines Klassenhierarchiediagramms dar. Hierbei befinden sich in der obersten Schicht die Klassen, welche die Darstellung übernehmen. In der Schicht darunter folgen die Objekte, welche die Informationen aus

der Datenbank in Objekten gekapselt präsentieren. Die darunter liegende Schicht enthält die Klassen, über die diese Informationen bezogen werden. Schließlich ist in der untersten Schicht zusätzlich die Startklasse sowie eine Klasse, welche das Speichern von lokalen Konfigurationsdaten übernimmt, dargestellt.

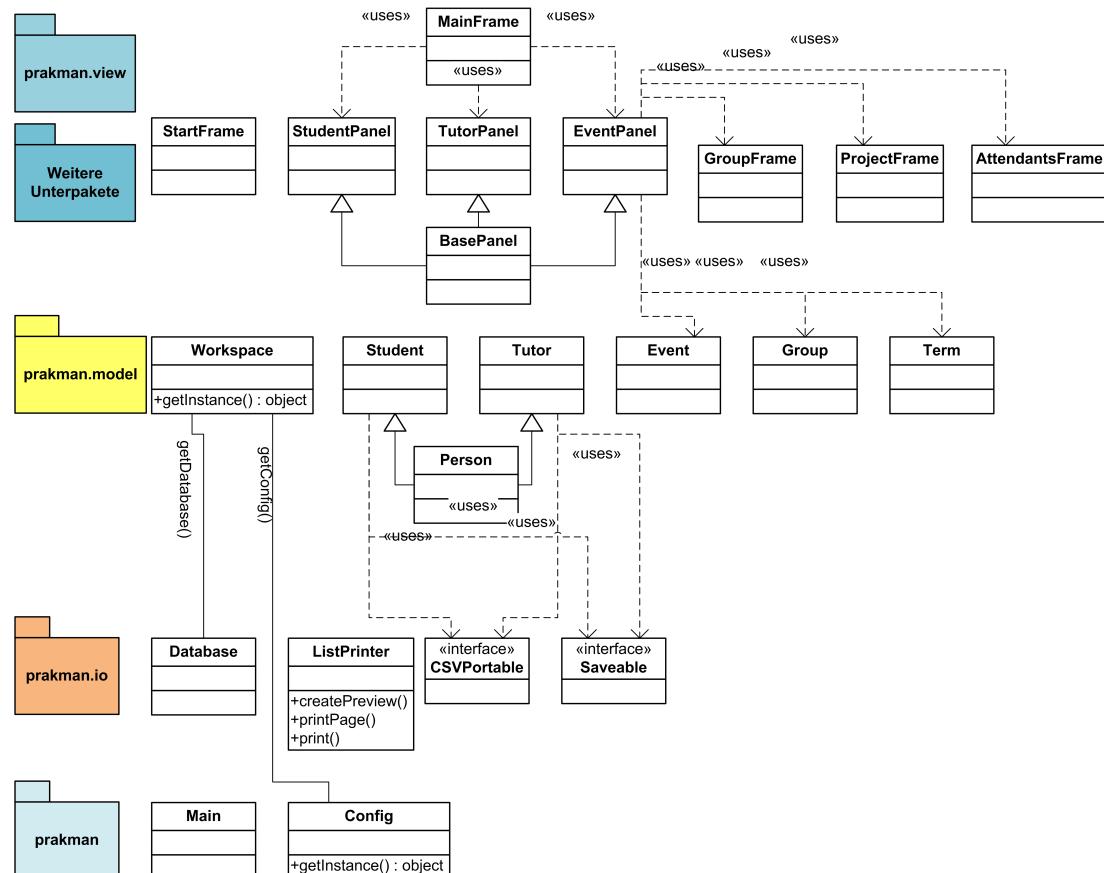


Abbildung 2.3: Klassenhierarchiediagramm von “Prakman” (wichtige Klassen)

2.3 Klassendokumentation

Siehe Verzeichnis “Javadoc” auf der beiliegenden CD.

2.4 Programmablauf

Die folgenden Programmablaufpläne verdeutlichen, wie die in Kapitel 1.4.4 (S.7) geforderten Funktionen in der Praxis in "PrakMan" umgesetzt sind.

/F0001/Verbindung zur Datenbank: Die Planung der Funktion kann dem Pflichtenheft, Kapitel 1.4.4 (S.8) entnommen werden. Die Umsetzung ist in der Abbildung 2.4 (S.19) dargestellt.

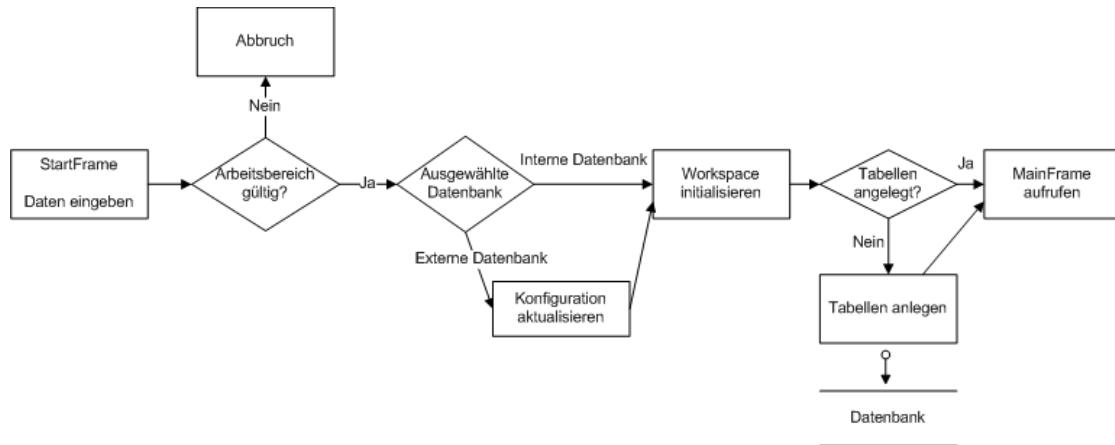


Abbildung 2.4: /F0001/ Ablaufplan: Programmstart

/F0002/Erstellen einer Veranstaltung: Die Planung der Funktion kann dem Pflichtenheft, Kapitel 1.4.4 (S.7) entnommen werden. Die Umsetzung ist in der Abbildung 2.5 (S.19) dargestellt.

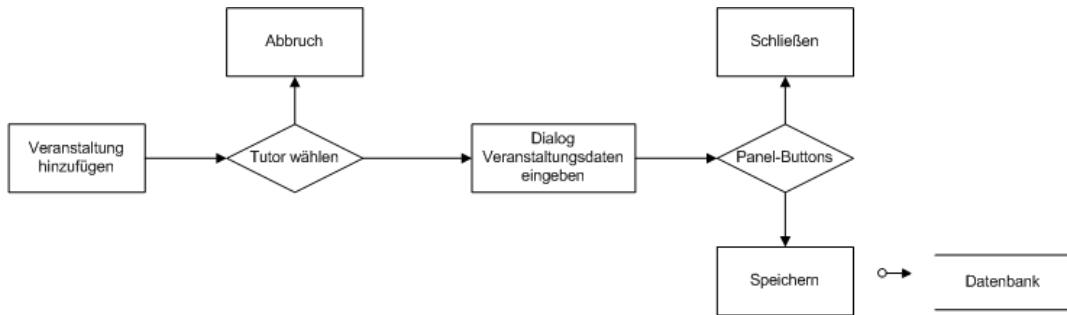


Abbildung 2.5: /F0002/ Ablaufplan: Eine Veranstaltung hinzufügen

/F0003/Hinzufügen von Studenten zu einer Veranstaltung: Die Planung der Funktion kann dem Pflichtenheft, Kapitel 1.4.4 (S.7) entnommen werden. Die Umsetzung ist in der Abbildung 2.6 (S.20) dargestellt.

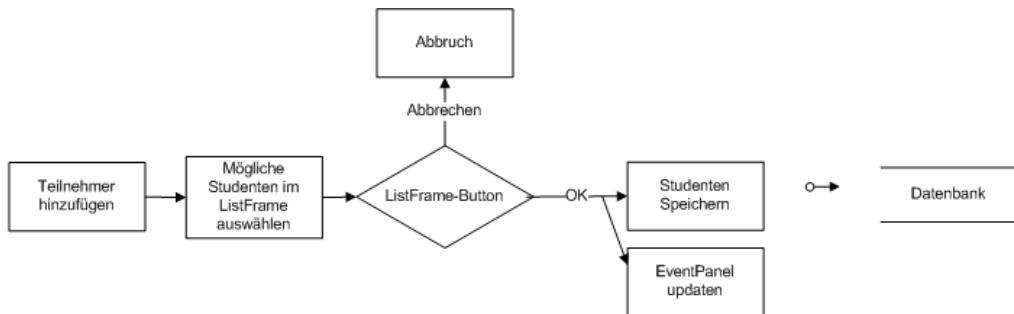


Abbildung 2.6: /F0003/ Ablaufplan: Studenten zu einer Veranstaltung hinzufügen

/F0004/Einteilung der Studenten in Gruppen: Die Planung der Funktion kann dem Pflichtenheft, Kapitel 1.4.4 (S.8) entnommen werden. Die Umsetzung ist in der Abbildung 2.7 (S.20) dargestellt.

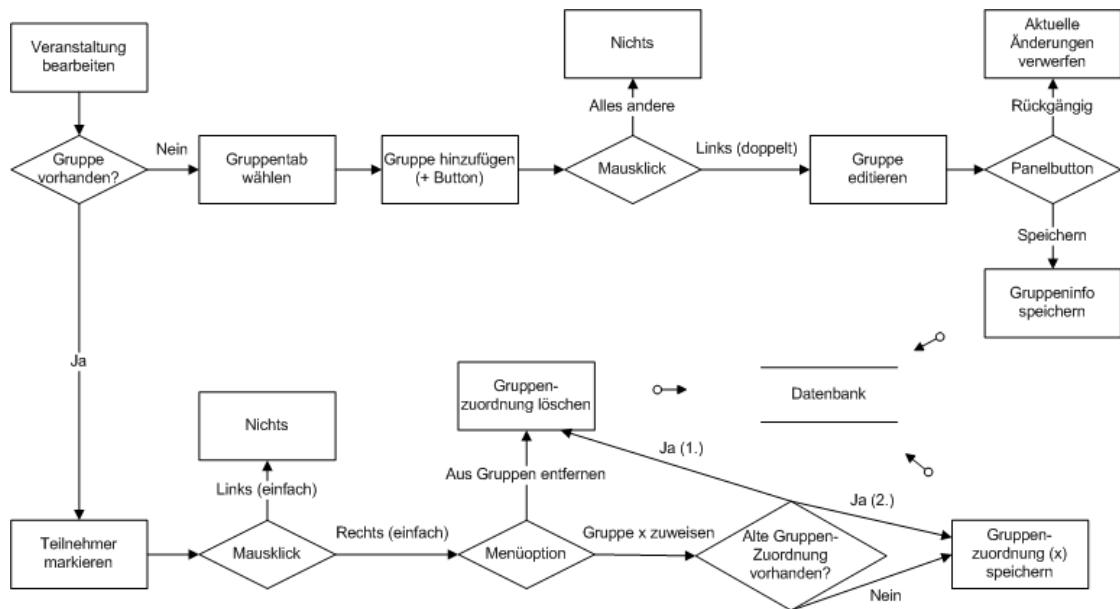


Abbildung 2.7: /F0004/ Ablaufplan: Studenten einer Veranstaltung in Gruppen einteilen

/F0005/Ein Projekt hinzufügen: Die Planung der Funktion kann dem Pflichtenheft, Kapitel 1.4.4 (S.8) entnommen werden. Die Umsetzung ist in der Abbildung 2.8 (S.21) dargestellt.

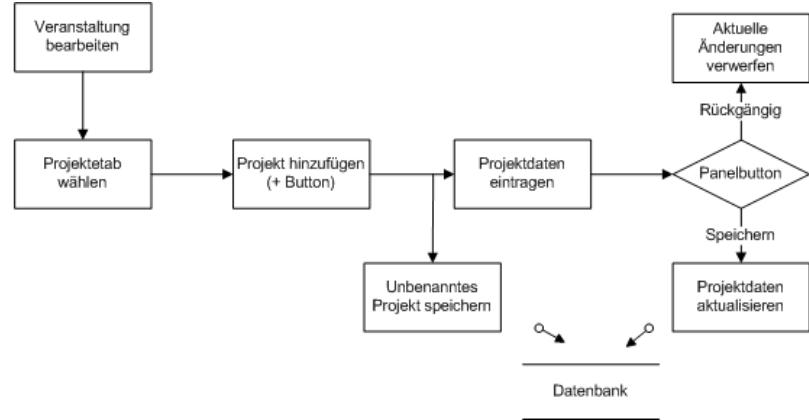


Abbildung 2.8: /F0005/ Ablaufplan: Ein neues Projekt hinzufügen

/F0006/Einen Termin hinzufügen: Die Planung der Funktion kann dem Pflichtenheft, Kapitel 1.4.4 (S.9) entnommen werden. Die Umsetzung ist in der Abbildung 2.9 (S.21) dargestellt.

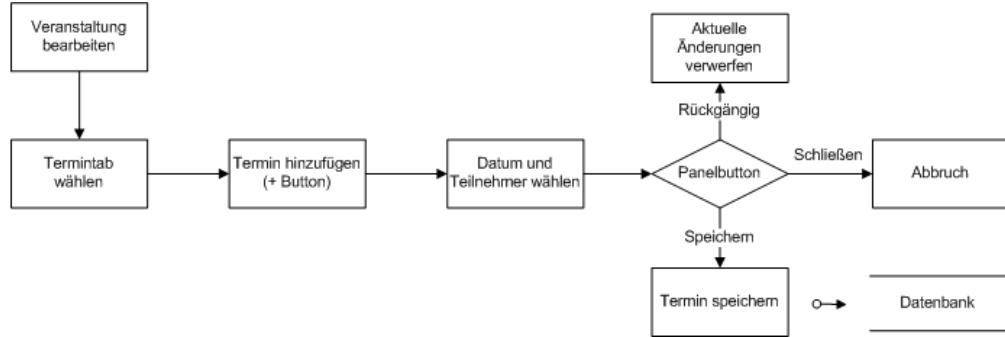


Abbildung 2.9: /F0006/ Ablaufplan: Einen neuen Termin für eine Veranstaltung hinzufügen

/F0007/Den Tutor einer Veranstaltung ändern: Die Planung der Funktion kann dem Pflichtenheft, Kapitel 1.4.4 (S.9) entnommen werden. Die Umsetzung ist in der Abbildung 2.10 (S.23) dargestellt.

/F0008/Einen neuen Dozenten hinzufügen: Die Planung der Funktion kann dem Pflichtenheft, Kapitel 1.4.4 (S.10) entnommen werden. Die Umsetzung ist in der Abbildung 2.11 (S.23) dargestellt.

/F0009/Einen neuen Studenten hinzufügen: Die Planung der Funktion kann dem Pflichtenheft, Kapitel 1.4.4 (S.10) entnommen werden. Die Umsetzung ist in der Abbildung 2.12 (S.23) dargestellt.

/F0010/Daten in CSV exportieren und /F0011/CSV-Daten importieren: Die Planung der Funktion kann dem Pflichtenheft, Kapitel 1.4.4 (S.11) entnommen werden. Die Umsetzung ist in der Abbildung 2.13 (S.24) dargestellt.

/F0012/Ein Projekt benoten Die Planung der Funktion kann dem Pflichtenheft, Kapitel 1.4.4 (S.12) entnommen werden. Die Umsetzung ist in der Abbildung 2.14 (S.24) dargestellt.

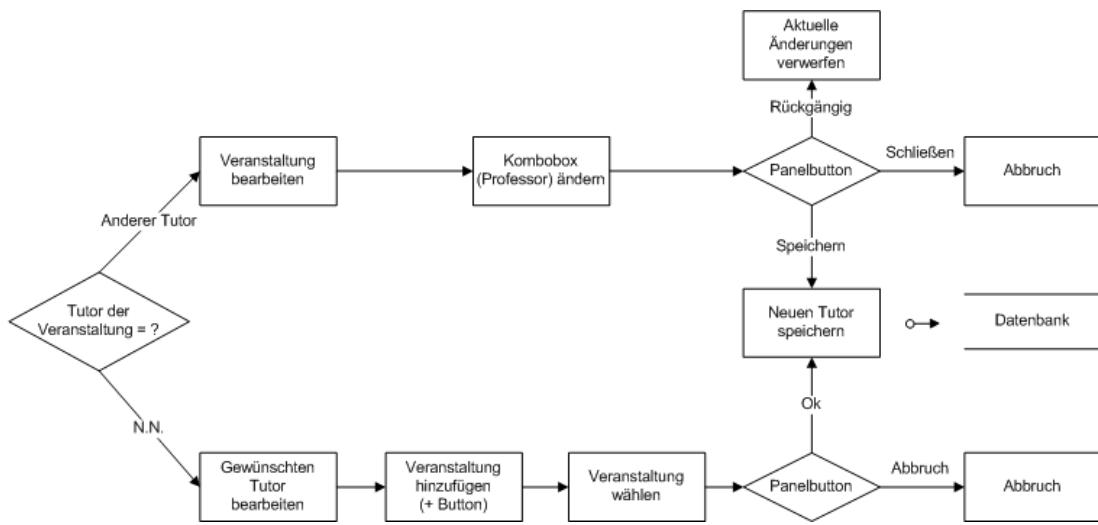


Abbildung 2.10: /F0007/ Ablaufplan: Den Tutor einer Veranstaltung ändern

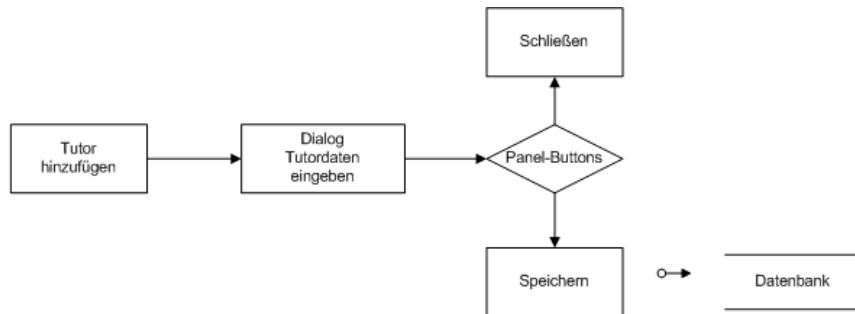


Abbildung 2.11: /F0008/ Ablaufplan: Einen neuen Dozenten hinzufügen

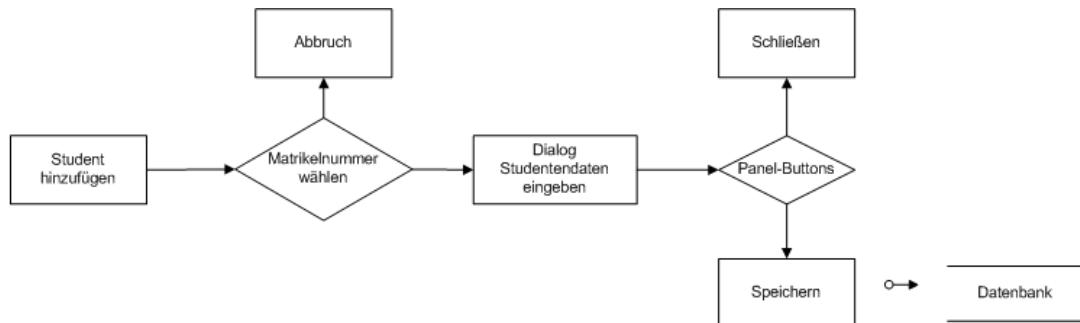


Abbildung 2.12: /F0009/ Ablaufplan: Einen neuen Studenten hinzufügen

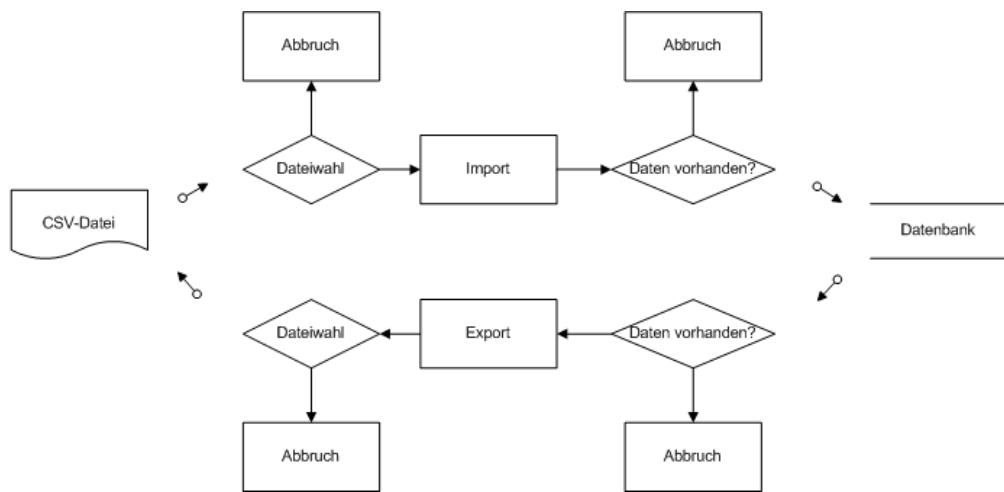


Abbildung 2.13: /F0010/ und /F0011/ Ablaufplan: CSV-Import/Export

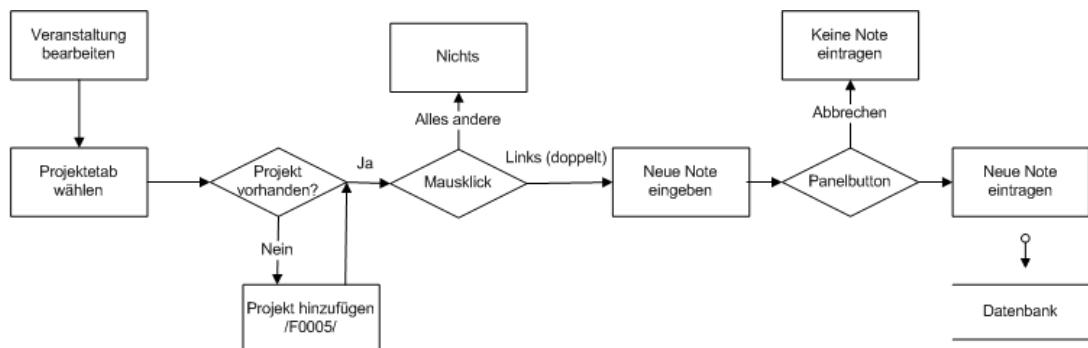


Abbildung 2.14: /F0012/ Ablaufplan: Ein Projekt benoten

2.5 Anwendungsgrenzen und Fehlersituationen

“PrakMan” erlaubt es bei Verwendung einer externen Datenbank, dass mehrere Benutzer auf der selben Datenbasis arbeiten. In diesem Fall können Probleme auftreten, falls unterschiedliche Benutzer gleichzeitig versuchen, die selben Datensätze zu verändern. In den meisten Fällen konnte diesem Problem mit dem gezielten Einsatz von Batches entgegengewirkt werden. Alle Problemsituationen können damit jedoch nicht sicher umgangen werden.

Ein Beispiel: Löschen und Editieren eine Veranstaltung: Dozent “A” und Dozent “B” greifen beide auf die selbe Datenbank “DB” zu. A öffnet nun die Veranstaltung “VC” in seinem Programm in einem Tab. Kurz danach löscht B die Veranstaltung VC aus der Datenbank DB. A möchte nun, neue Mitglieder zu VC hinzufügen und wählt diese aus. Sobald A diese Informationen in DB übertragen hat, ist VC auf einmal wieder vorhanden - jedoch ohne Projekte, Noten und Termine.

Diese Fälle stellen nicht wirklich einen Fehler dar, zeigen jedoch das Problem, das besteht, weil Datenbank und Programm nicht ständig auf dem gleichen Stand gehalten werden können.

2.6 Benutzeroberflächenhierarchie

In “PrakMan” existiert ein Hauptfenster, der “MainFrame”. In diesem befindet sich auf der linken Seite ein Baum, auf der rechten Seite ein “TabbedPane”, welches unterschiedliche Panels hält, die die Informationen der Objekte aus dem Baum für den Benutzer darstellen.

Den Zusammenhang bzw. die Hierarchie dieses Bereichs stellt die Abbildung 2.15 (S.26) dar.

Abbildung 2.16 (S.26) geht weiter auf den speziellen Aufbau der Kontextmenüs im Baum (linke Seite der Benutzeroberfläche) ein.

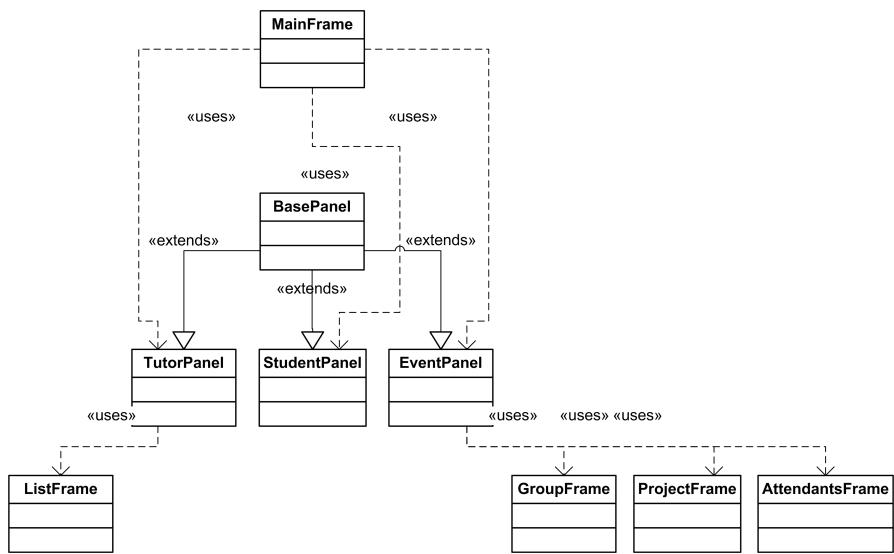


Abbildung 2.15: Ablaufplan: Hierarchie der Benutzeroberfläche, ausgehend vom MainFrame

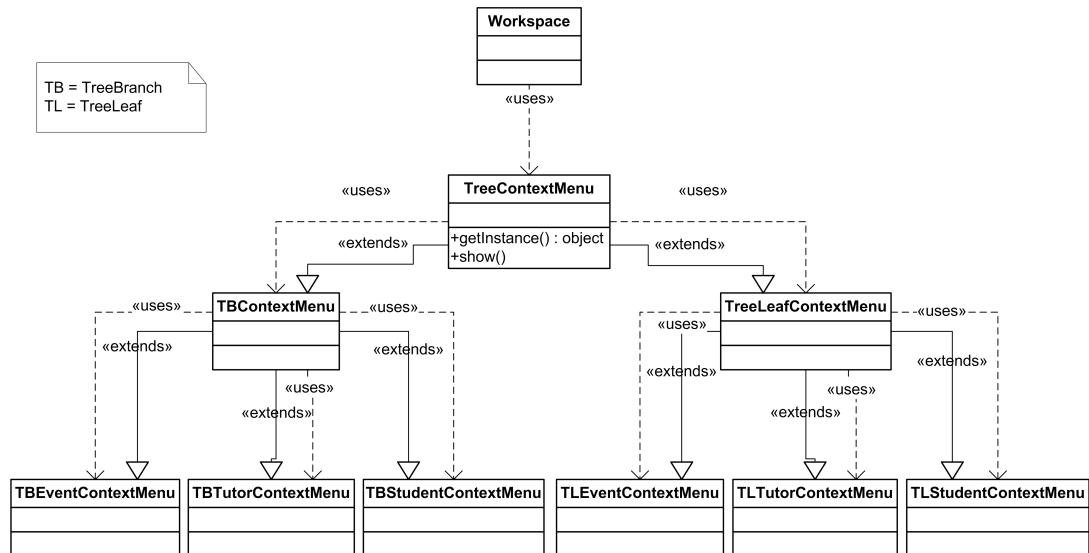


Abbildung 2.16: Ablaufplan: Hierarchie der Benutzeroberfläche, speziell des Baumes, ausgehend vom Workspace

2.7 Funktionsbeschreibung der Benutzeroberflächen und deren Komponenten

Exemplarisch die Zusammenstellung der wichtigsten GUI-Masken und eine Beschreibung ihrer Funktion(en):

MainFrame = JMenuBar + txtFilter + splitPane + Workspace (Tree) + tabbedPane
+ [EventPanel — TutorPanel — StudentPanel]

Im Mainframe laufen alle Fenster zusammen. Er stellt das äußere Gerüst dar, zu welchem der Baum gehört, der alle Objekte übersichtlich zur Verfügung stellt, sowie die Menüleiste, welche weitere Optionen anbietet. Außerdem erlaubt er durch eine “splitPane” das einfache Verändern der Größe der beiden Teilbereiche des Hauptfensters. Zuletzt ist auch das “tabbedPane”, in welchem weitere Unterpanels angezeigt werden, ein Teil dieses Hauptfensters.

BasePanel = JPanel

Das BasePanel stellt eine Basisklasse für weitere Objektpansels dar. Es legt hauptsächlich abstrakte Methoden fest, die in den abgeleiteten Klassen implementiert werden müssen. Ansonsten ist es ein einfaches JPanel, das anschließend mit Inhalten gefüllt werden kann.

EventPanel = BasePanel + SaveResetEditPanel + eventContentPanel + tabbedPane
+ [eventMemberPanel — eventResultPanel — eventTermsPanel — eventGroupsPanel]

Das EventPanel ist eine Ableitung des BasePanel und wird im “tabbedPane” des MainFrame angezeigt. Dieses Panel zeigt alle Informationen an, die über eine Veranstaltung existieren. Dazu gehören eine Beschreibung, die Teilnehmer, Arbeitsgruppen, Projekte bzw. Hausarbeiten, und Termine sowie Anwesenheitslisten für Letztere. All diese Daten können über das Panel auch editiert werden. Es erlaubt somit eine umfassende Administration einer gewählten Veranstaltung.

TutorPanel = BasePanel + SaveResetEditPanel + tutorContentPanel + tabbedPane
+ [tutorPraktPanel]

Das TutorPanel stellt ebenfalls eine Ableitung des BasePanel dar und wird an der gleichen Stelle angezeigt. Hier können Dozenten-bezogene Daten gefunden werden: Vorname, Nachname und welche Veranstaltungen vom gewählten Dozenten angeboten werden. All diese Informationen lassen sich wiederum editieren.

StudentPanel = BasePanel + SaveResetEditPanel + studentContentPanel + tabbedPane
+ [studentPraktPanel — studentResultsPanel]

Auch das StudentPanel ist eine Ableitung des BasePanel und wird im “tabbedPane” des MainFrame angezeigt. Hier können Studenten-bezogene Daten eingesehen und editiert werden: Vorname, Nachname, E-Mail-Adresse, an welchen Veranstaltungen nimmt der/die Student/in teil und welche Projekte werden bearbeitet oder wurden auch schon

benötigt. Die beiden letztgenannten Funktionen müssen allerdings über eine Veranstaltung vorgenommen werden und können hier lediglich angezeigt werden. Die Tabellen bieten dafür allerdings die Möglichkeit, über einen Doppelklick auf Veranstaltungs/Tutor/Studentendaten das entsprechende Objekt separat zu öffnen, um dort direkt weiterarbeiten zu können.

2.8 Dateiaufbau aller Anwenderdateien mit DD-Beschreibung und Funktion

2.8.1 Export von Studentendaten:

Beim Exportieren von Studentendaten werden Vorname, Nachname, Matrikelnummer und E-Mail-Adresse in einer CSV-Datei gespeichert. Der Aufbau sieht folgendermaßen aus.

studenten.csv = Vorname + Name + E-Mail + Matrikelnummer

2.8.2 Export von Dozentendaten:

Beim Exportieren von Dozentendaten werden Vorname, Nachname sowie eine intern vergebene Dozenten-ID in einer CSV-Datei gespeichert. Der Aufbau sieht folgendermaßen aus.

dozenten.csv = Vorname + Name + Tutor-ID

2.8.3 Export von Veranstaltungsdaten:

Beim Exportieren von Daten einer gewählten Veranstaltung werden alle verfügbaren Informationen in einer mehrschichtigen CSV-Datei zusammengestellt. Alle Zeitangaben werden als Unix-Timestamp exportiert. Der Aufbau dieser CSV ist etwas komplexer und sieht folgendermaßen aus.

veranstaltung.csv = Veranstaltungs-Info + Tutor-Info + Studenten-Info + Projekt-Info + Benotungs-Info + Termin-Info

Alle Informations-Abschnitte sind durch eine Leerzeile sowie eine Zeile, in der die Bezeichnung der Spalten untergebracht ist, getrennt.

Veranstaltungs-Info = Veranstaltungsname + Veranstaltungs-ID

Tutor-Info = Tutor-ID + Nachname + Vorname

Studenten-Info = Matrikelnummer + Name + Vorname + E-Mail

Projekt-Info = Projekt-ID + Beschreibung + Startzeit + Endzeit

Benotungs-Info = Projekt-ID + Matrikelnummer + Note

Termin-Info = Termin-ID + Zeitpunkt

3 Datenbank

3.1 Dokumentation der Datenbank

Als Datenbank ist für “PrakMan” die Nutzung unterschiedlicher Angebote möglich. Unterstützt werden derzeit Hypersonic-DB (HSQL, sowohl lokal als auch extern) und PostgreSQL. Dafür wurde der Aufbau der Tabellen relativ einfach gehalten, um allen Datenbanken gerecht zu werden.

In den folgenden Abschnitten wird der genaue Aufbau der für “PrakMan” benötigten Tabellen beschrieben bzw. dargestellt.

3.2 Entity- Relationship- Modell (ERM) der Datenbank

Der Aufbau der Datenbank, welche für den Praktikums-Manager unabdingbar ist, ist in den Grafiken auf den Seiten 32, 33 und 34 dargestellt. Einige Tabellen werden dabei mehrfach gezeigt, um jeweils den Zusammenhang zu den Grundobjekten (Student, Tutor, Event) darzustellen.

3.3 Prüfung des Relationenschemas auf Einhaltung der Normalisierung

3.3.1 1te Normalform

Alle Tabellen enthalten atomare Attribute. Listen werden durch zusätzliche Tabellen dargestellt.

Somit ist 1NF erfüllt.

3.3.2 2te Normalform

Alle Relationen sind in 1NF. Außerdem haben alle Objekte (Studenten, Tutoren, Events, Termine, Gruppen, Projekte) eine in der entsprechenden Relation einmalige Identifikationsnummer (ID). Jegliche Verweise laufen ausschließlich über solche IDs. Somit bestehen auch sämtliche Primärschlüssel aus einer ID oder mehreren IDs, von denen alle weiteren Attribute voll funktional abhängen. Weitere Daten werden für Referenzen nicht benötigt, weshalb auch keine redundaten Daten in diesem Zusammenhang auftreten können.

Somit ist 2NF erfüllt.

3.3.3 3te Normalform

Alle Relationen sind in 2NF. Auch hier ist die Eindeutigkeit ausschließlich über die vorhandenen IDs hergestellt. Keine andere Kombination von Schlüsselkandidaten existiert und würde eine weitere Identifikation eines Objekts in einer Tabelle ermöglichen. Demnach können auch keine Nicht-Schlüssel-Attribute von anderen Nicht-Schlüssel-Attributen funktional abhängen.

Somit ist 3NF erfüllt.

3.4 Funktionsbeschreibung der Formatprogramme

Zum Erstellen der Tabellen wird kein Formatprogramm benötigt. Bei Programmstart wird automatisch eine passende SQL-Datei gelesen und die Tabellen, falls nötig, erstellt.

3.5 Funktionsbeschreibung der Reportprogramme

Es wurde kein spezielles Reportprogramm zum Auslesen der Datenbank geschrieben, da das, in verbesserter Darstellung, eine Funktion des Programms “PrakMan” ist.

3.6 Funktionsbeschreibung der SQL- /ESQL- /JDBC- Programme

Wie bereits in Punkt 3.4 erwähnt, liest “PrakMan” zum Erstellen der Tabellen eine SQL-Datei ein und erstellt nach diesen Vorgaben, falls nötig, das Grundgerüst für eine korrekte Funktionsweise. Die genannte Datei ist “CreateTables.sql”, welche sich im Paket prakman.io.sql befindet. Auf diese Weise lassen sich bei Änderung eines Datenbankprogramms auf eine andere Version sehr einfach die Tabellen anpassen.

Für die Zukunft wäre auch eine Erweiterung denkbar, die für jede Datenbankform eine extra angepasste SQL-Datei enthält, um optimal die Möglichkeiten der unterschiedlichen Datenbanktypen nutzen zu können.

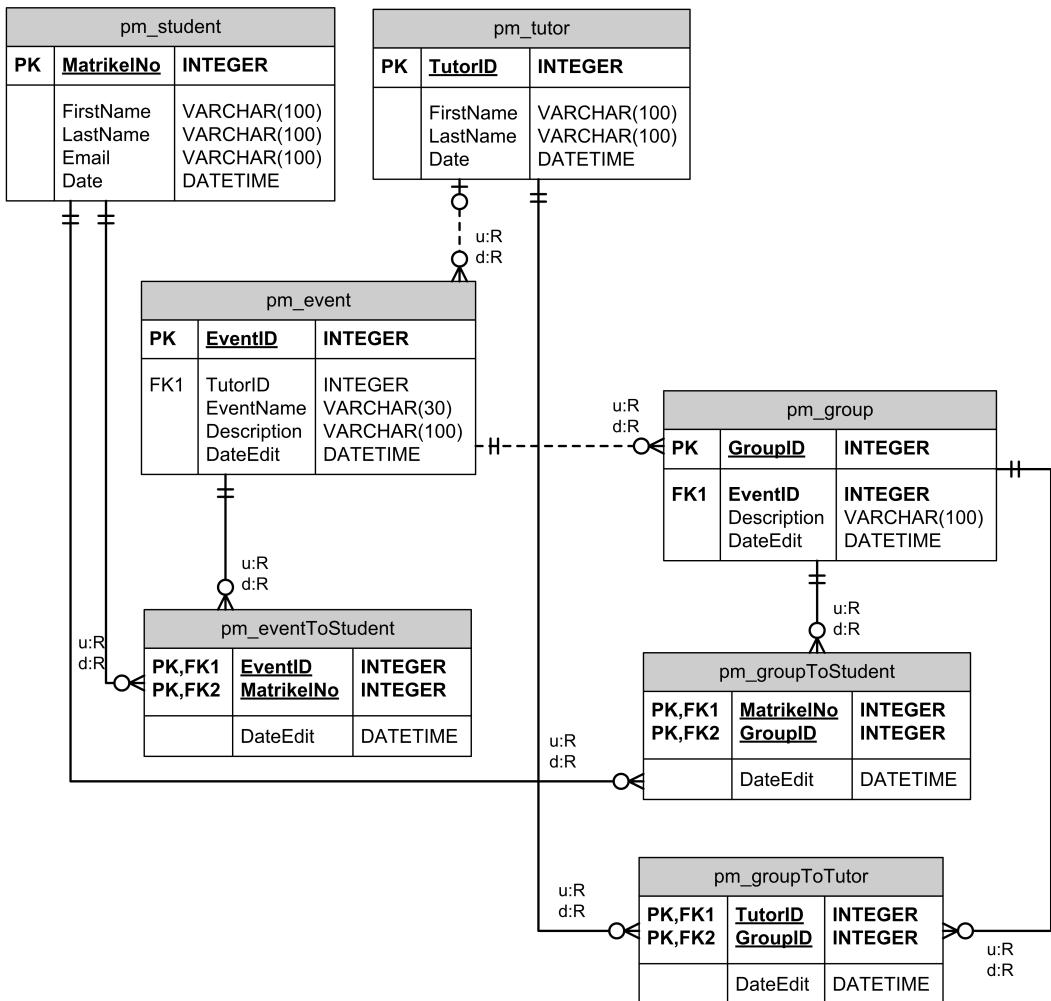


Abbildung 3.1: ER-Modell der Beziehungen zwischen Student, Tutor und Event

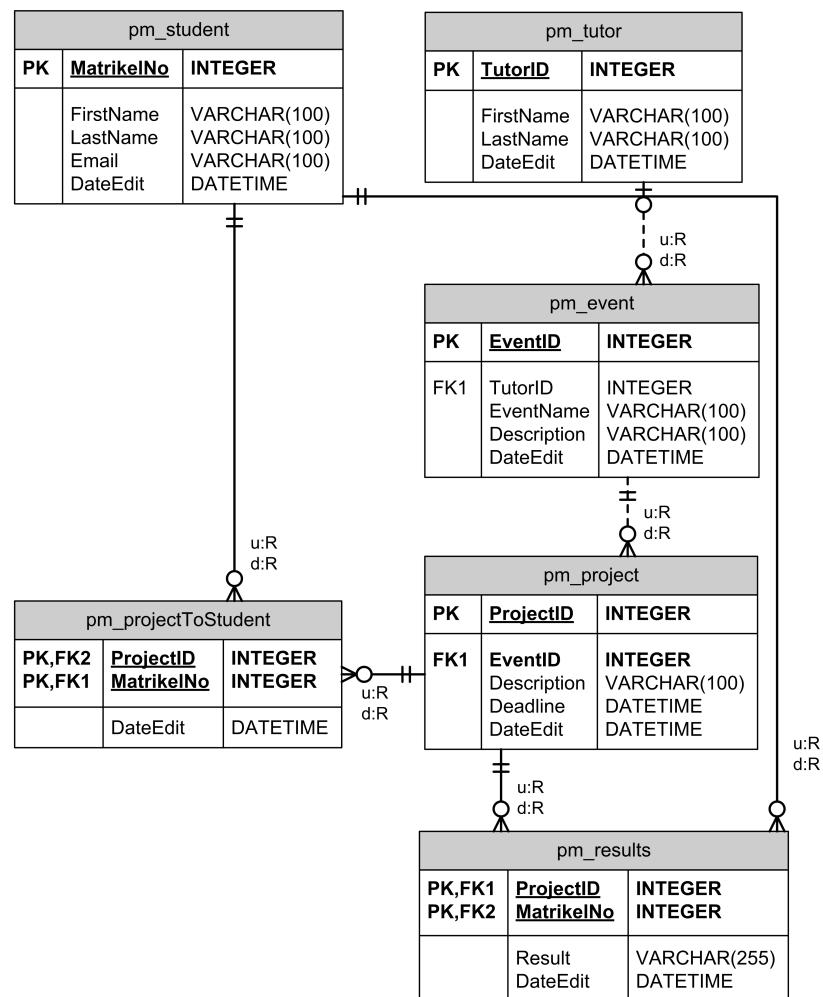


Abbildung 3.2: ER-Modell der Beziehungen zwischen Student, Tutor, Events und Projekten

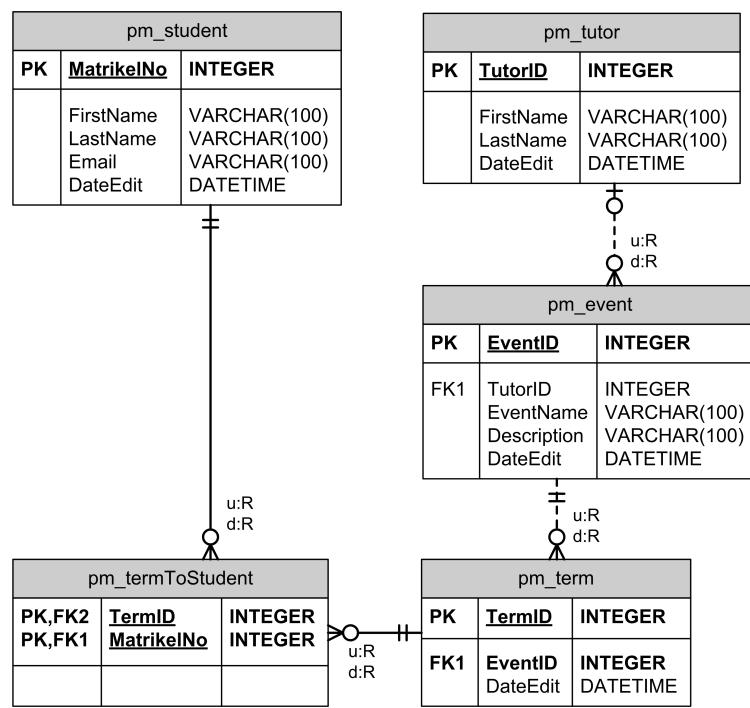


Abbildung 3.3: ER-Modell der Beziehungen zwischen Student, Tutor, Events und Terminen

4 Installation und Test

4.1 Testverfahren und Testdokumente mit automatischen Tests

Die Software wurde auf unterschiedliche Arten auf korrekte Funktionsweise überprüft. Zuerst individuell durch den Entwickler, der seine Komponente direkt nach Fertigstellung prüft. Anschließend durch die anderen Entwickler, die diese Komponente im Zusammenspiel mit ihren eigenen testen. Schließlich ein Performance-Test durch eine vorher programmierte Automatik.

4.1.1 Individuelle Komponenten-Tests

Jeder Programmierer testet seine neu erstellte oder geänderte Komponente selbstständig auf korrekte Funktionsweise, um von Anfang an interne Fehler der Komponenten ausschließen zu können, wenn sie im Zusammenspiel mit dem Rest des Programms getestet wird.

4.1.2 Tests des Komponenten-Zusammenspiels

Dieses Tests wurden durchgeführt, wenn größere Abschnitte des Programms fertiggestellt sind bzw. neue Komponenten zum bereits funktionierenden Hauptteil hinzugefügt wurden. Dabei wurde sichergestellt, dass die neuen Komponenten richtig mit den anderen zusammenarbeitet aber auch keine neuen Fehler mit sich bringt.

4.1.3 Performance-Test

Beim automatisierten Test wurden 15.000 zufällig generierte Studentendaten in das Programm eingefügt. Dabei ließ sich "PrakMan" weiterhin gut nutzen und blieb in seiner Funktionsweise stabil. Da diese Anzahl das normal erwartete Datenaufkommen von etwa 1.000 - 5.000 Studenten deutlich übersteigt, gilt eine gute Funktionsweise als gesichert. Der Test kann selbstständig durchgeführt, indem der Konfigurationsdatei "config.conf" im gewählten Arbeitsverzeichnis der Eintrag "CrashTest= $\{$ Anzahl einzufügender Testwerte $\}$ " hinzugefügt wird. Beim nächsten Start wird dann die angegebene Anzahl an Testwerten der Datenbank hinzugefügt.

4.2 Übergabedokument

Übergeben wird eine CD mit dem Programm als JAR-Datei, den .java Dateien, der Dokumentation als PDF-Dokument sowie Javadoc und allen MS Visio-Dateien.

5 Programmbetrieb

5.1 Bedienungsanleitung

5.1.1 Startbildschirm

Wird “PrakMan” gestartet, bietet sich dem Benutzer die in Abb.5.1 (S.41) dargestellte Eingabemaske. Dort müssen beim ersten Mal mehrere Informationen angegeben werden. “PrakMan” merkt sich zusätzlich die vier letztgewählten Arbeitsverzeichnisse und die dazugehörigen Verbindungsparameter. Beim nächsten Start sind so immer automatisch die letzten Einstellungen ausgewählt.

1. Ein Arbeitsverzeichnis muss gewählt werden. Hier legt “PrakMan” seine Konfigurationsdatei und ggf. die interne Datenbank ab.
2. Entschieden werden, ob eine interne oder externe Datenbank genutzt werden soll.
Eine interne Dateibank wird in einigen Dateien im gewählten Arbeitsverzeichnis abgelegt.
3. Bei Nutzung einer externen Datenbank müssen noch weitere Informationen für eine korrekte Verbindung angegeben werden. Dazu gehört der Datenbanktyp, die Adresse des Servers, der Name der Datenbank, der Benutzername und das zugehörige Passwort.

Sind all diese Angaben gemacht, kann “PrakMan” durch Drücken des OK-Buttons gestartet werden.

5.1.2 Hauptbildschirm

Wurden alle Einstellungen im Startdialog gemacht und eine Verbindung zur gewählten Datenbank erfolgreich hergestellt, wird die in Abb.5.2 (S.41) dargestellte Oberfläche aufgerufen. Dieses Fenster bleibt die ganze Zeit über sichtbar und ist in drei Bereiche aufgeteilt.

1. Die Menüleiste: Hier kann der Arbeitsbereich bzw. die Datenbank gewechselt sowie sämtliche Studentendaten ausgedruckt werden. Auch Informationen über “PrakMan” sind von dort aus erreichbar.
2. Der Baum im linken Teil. Hier sind sämtliche Informationen aus der Datenbank aufgelistet: Alle Veranstaltungen, Dozenten sowie Studenten. Über einen Doppelklick können diese Objekte dann im rechten Teil des Fensters geöffnet werden.

3. Die Arbeitsfläche im rechten Teil. Hier werden alle Objekte geöffnet, um ihre Informationen anzusehen oder zu editieren.

Der Baum bietet zudem für die Äste Veranstaltungen, Lehrende und Studenten sowie die darin enthaltenen Objekte unterschiedliche Menüs (siehe Abbildung 5.3, S.41), mit denen Daten importiert, exportiert, gedruckt, bearbeitet oder gelöscht werden können.

5.1.3 Eine Veranstaltung: Teilnehmer

Öffnet der Benutzer per Doppelklick darauf (oder Rechtsklick, anschließend Bearbeiten) eine Veranstaltung, so öffnet sich die in Abbildung 5.4 (S.42) dargestellte Oberfläche. Dort können über die Eingabefelder im oberen Bereich allgemeine Einstellungen der Veranstaltung verändert werden: Der Name, der Dozent sowie die Beschreibung der Veranstaltung. Sollen diese Informationen gespeichert oder zurückgesetzt werden, helfen die Buttons “Speichern” und “Zurücksetzen”, die direkt darüber positioniert sind.

Die Tabelle darunter zeigt im standardmäßig geöffneten “Teilnehmer-Tab” die eingetragenen Teilnehmer der Veranstaltung samt Matrikelnummer, Nachname, Vorname und gegebenenfalls der bereits zugewiesenen Arbeitsgruppe.

Sollen genauere Infos über einzelne Studenten angezeigt werden, können diese mit einem einfachen Doppelklick geöffnet werden. Wenn Sie die Studenten in Gruppen einteilen möchten, markieren Sie die gewünschten Teilnehmer in der Tabelle (Shift und Strg-Tasten sind zur Unterstützung möglich) und wählen Sie über einen Rechtsklick im darauf auftauchenden Menü die neue Gruppe aus.

Die vier Buttons im unteren Teil der Ansicht ermöglichen es, auf einen Schlag alle oder keinen Studenten zu markieren, sowie neue Teilnehmer hinzuzufügen oder alte zu entfernen.

5.1.4 Eine Veranstaltung: Projekte

Die in Abbildung 5.5 (S.42) dargestellte Ansicht öffnet sich, wenn beim Editieren eines Projektes (siehe Punkt 5.1.3) der “Projekte-Tab” der Tabelle ausgewählt wird. Hier können alle für diese Veranstaltung angelegten Projekte eingesehen werden: Ihre Beschreibung, wann sie vergeben wurden und wann sie abgegeben werden soll(t)en.

Die vier Buttons am unteren Rand ermöglichen es nun, entweder alle oder kein Projekt anzuwählen sowie neue Projekte hinzuzufügen oder alte zu löschen.

Ein Doppelklick auf ein existierendes Projekt öffnet die in Abbildung 5.6 (S.42) gezeigte Ansicht. In dieser ist es möglich, alle Daten eines Projektes zu ändern: Die Beschreibung, Start- und Endtermin, sowie die Teilnehmer und deren Benotung.

Zum Editieren der Beschreibung, wird einfach eine neue in das entsprechende Textfeld eingetippt. Der “Speichern-Button” überträgt die Informationen in die Datenbank. Zum Setzen des Start- und Endtermins müssen die Buttons am rechten Rand dieser Eingabezeilen genutzt werden. Diese öffnen das in 5.7 (S.43) gezeigte Fenster, in welchem eine Uhrzeit eingegeben sowie das gewünschte Datum über die Oberfläche ausgewählt werden kann. Die Studenten, die an diesem Projekt teilnehmen (sollen), können über die im

unteren Teil implementierten “Hinzufügen- und Entfernen-Buttons” eingestellt werden. Zur Benotung eines einzelnen Studenten, reicht ein Doppelklick auf dessen Zeile in der Tabelle, welches das ebenfalls in Abbildung 5.6 (S.42) gezeigte Eingabefeld öffnet, über welches dann eine neue Note vergeben werden kann.

5.1.5 Eine Veranstaltung: Termine

Die in Abbildung 5.8 (S.43) dargestellte Ansicht öffnet sich, wenn beim Editieren eines Projektes (siehe Punkt 5.1.3) der “Termine-Tab” der Tabelle ausgewählt wird. Hier können alle für diese Veranstaltung angelegten Termine eingesehen sowie neue hinzugefügt, alte gelöscht oder die tatsächlich anwesenden Teilnehmer eingetragen werden.

Über die “Hinzufügen- und Entfernen-Buttons” am unteren Rand können nun neue Termine hinzugefügt sowie in der Tabelle markierte Termine entfernt werden. Soll ein neuer Termin angelegt werden, öffnet sich das in Abbildung 5.8 (S.43) gezeigte Fenster. Dort muss über den Datumsbutton ein Zeitpunkt gewählt werden. Anschließend kann über Häkchen in der darunter befindlichen Tabelle gewählt werden, welche Studenten anwesend waren. Die selbe Ansicht (mit den selben Möglichkeiten) öffnet sich ebenfalls bei Doppelklick auf einen bereits vorhandenen Termin. Somit können auch alle Informationen im Nachhinein geändert werden.

5.1.6 Eine Veranstaltung: Gruppen

Die in Abbildung 5.9 (S.43) dargestellte Ansicht öffnet sich, wenn beim Editieren eines Projektes (siehe Punkt 5.1.3) der “Gruppen-Tab” der Tabelle ausgewählt wird. Hier können alle für diese Veranstaltung angelegten Gruppen eingesehen sowie neue hinzugefügt oder alte gelöscht werden.

Die Buttons unter der Tabelle ermöglichen es, von links nach rechts, eine neue Gruppe zu erstellen, eine markierte Gruppe zu löschen, alle Gruppen zu selektieren und zuletzt alle abzuwählen.

Wird eine Gruppe über den “Hinzufügen-Button” erstellt, erscheint in der Tabelle eine neue Gruppe ohne Beschreibung. Diese kann dann noch, wie alle anderen Gruppen auch, über einen Doppelklick editiert werden. Dabei wird das in Abbildung 5.9 (S.43) gezeigte Eingabefenster aufgerufen, in welchem die Beschreibung editiert und per Button gespeichert werden kann.

5.1.7 Ein Tutor: Betreute Veranstaltungen

Öffnet der Benutzer per Doppelklick darauf oder Rechtsklick ↗ Bearbeiten einen Tutor, so öffnet sich die in Abbildung 5.10 (S.44) dargestellte Oberfläche. Dort können über die Eingabefelder im oberen Bereich allgemeine Informationen über den Tutor editiert werden. Sollen diese Informationen gespeichert oder zurückgesetzt werden, helfen die Buttons “Speichern” und “Zurücksetzen”, die direkt darüber positioniert sind.

Die Tabelle im zentralen Bereich des Fensters listet alle Veranstaltungen auf, für die dieser Tutor als Betreuer eingesetzt ist. Ein Doppelklick auf eine Veranstaltung öffnet diese in einem separaten Tab.

Die vier Buttons unter der Tabelle ermöglichen es, von links nach rechts, den Tutor für eine noch unbetreute Veranstaltung einzutragen, ihn von einer Veranstaltung als Tutor zu entfernen, alle Veranstaltungen zu wählen und schließlich alle abzuwählen.

5.1.8 Ein Student: Gewählte Veranstaltungen

Öffnet der Benutzer per Doppelklick darauf oder Rechtsklick „Bearbeiten“ einen Studenten, so öffnet sich die in Abbildung 5.11 (S.44) dargestellte Oberfläche. Dort können über die Eingabefelder im oberen Bereich allgemeine Informationen über den Studenten editiert werden. Sollen diese Informationen gespeichert oder zurückgesetzt werden, helfen die Buttons „Speichern“ und „Zurücksetzen“, die direkt darüber positioniert sind.

Die Tabelle im zentralen Bereich des Fensters listet alle Veranstaltungen auf, an denen dieser Student teilnimmt als auch, falls dies der Fall ist, die Arbeitsgruppe in der er sich befindet sowie deren Beschreibung. Ein Doppelklick auf eine Veranstaltung öffnet diese in einem separaten Tab.

5.1.9 Ein Student: Noten

Die in Abbildung 5.12 (S.45) dargestellte Ansicht öffnet sich, wenn beim Editieren eines Projektes (siehe Punkt 5.11) der „Noten-Tab“ der Tabelle ausgewählt wird. Hier können alle Noten eingesehen werden, die der Student bereits in jeglichen Veranstaltungen für unterschiedliche Projekte bekommen hat.



Abbildung 5.1: Startbildschirm von “PrakMan”

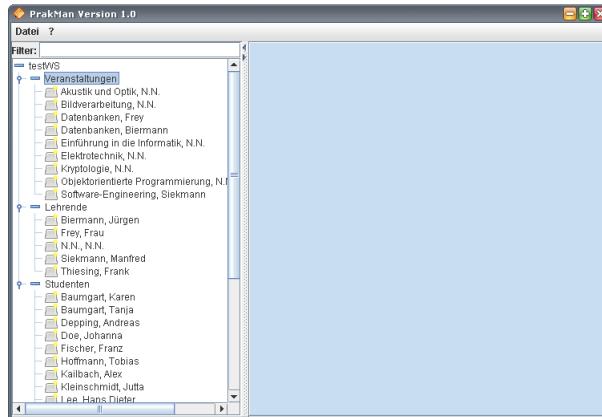


Abbildung 5.2: Hauptbildschirm von “PrakMan”

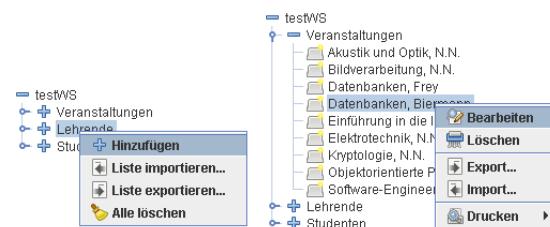


Abbildung 5.3: Menü eines Astes sowie eines Objekts im Dateibaum von “PrakMan”

Datenbanken, Biermann

Veranstaltung: Datenbanken
Verantwortlicher Professor: Biermann, Jürgen
Beschreibung: Praktikum von Prof. Biermann

Teilnehmer	Projekte	Termine	Gruppen	
#	Matri-Nr.	Nachname	Vorname	Gruppe
320001	Ritterbusch	Kai		4
320002	Degenhardt	Andreas		0
320003	Uhl	Gruppe setzen	1	1
320004		Aus Gruppe(n) entfernen	3	1
320005	Kaill		4	1
320006	Schwerdt	Daniela		4
320007	Baumgart	Tanja		1
320008	Baumgart	Karen		4
320009	Lins	Christian		4
320010	Meier	Anneliese		4
320011	Müller	Karl Heinz		4
320012	Schulz	Karl		3
320013	Fischer	Franz		3

Abbildung 5.4: Teilnehmeransicht einer Veranstaltung sowie Zuweisung einer Gruppe in “PrakMan”

Datenbanken, Biermann

Veranstaltung: Datenbanken
Verantwortlicher Professor: Biermann, Jürgen
Beschreibung: Praktikum von Prof. Biermann

Teilnehmer	Projekte	Termine	Gruppen
#	Beschreibung	Vergeben am	Abzugeben am
	Erstellen eines S...	15.09.2007	01.09.2007
	Erarbeitung von T...	15.09.2007	Unbegrenzt

Abbildung 5.5: Projektansicht einer Veranstaltung in “PrakMan”

Projekt Editor

Beschreibung: Erstellen eines Skripts in LaTeX
Start-Termin: 2007-09-15 14:43:23,218
Abgabe-Termin: 2007-09-01 00:00:00,0

#	Matri-Nr.	Nachname	Vorname	Note
	320001	Ritterbusch	Kai	Sehr gut
	320009	Lins	Christian	Sehr gut

Eingabe

Bitte eine Note eingeben

OK Abbrechen

Studenten

Abbildung 5.6: Editieren eines Projektes sowie dessen Benotung in “PrakMan”



Abbildung 5.7: Wählen eines Termins in “PrakMan”

Anwesend	Matr.-Nr.	Nachname	Vorname
	320001	Ritterbusch	Kai
	320002	Depping	Andreas
	320003	Rollwage	Philipp
	320004	Uhlenbrok	Jan
	320005	Kalibach	Alex
	320006	Schwerdt	Daniela
	320007	Baumgart	Tanja
	320008	Baumgart	Karen
	320009	Ans	Christian
	320010	Hilpert	Annetmarie
	320011	Müller	Karl Heinz
	320012	Schulz	Karl
	320013	Fischer	Franz
	320014	Schneider	Paula
	320015	Dein	Johanna
	320016	Kleinschmidt	Jutta
	320017	Hoffmann	Tobias
	320018	Remarque - Scha.	Erich Maria Hans Jr
	320019	Walter	Hellmut
	320020	ToChun	Won
	320021	Lee	Hans Dieter
	320022	Wing	Chan
	320023	Schinack	Schack

Abbildung 5.8: Terminansicht einer Veranstaltung sowie Editieren eines Termins in “PrakMan”

Gruppen-Nr.	Beschreibung
1	Gruppe 2 v. Datenbanken Biermann

Abbildung 5.9: Gruppenansicht einer Veranstaltung sowie Editieren von Gruppeninformationen in “PrakMan”

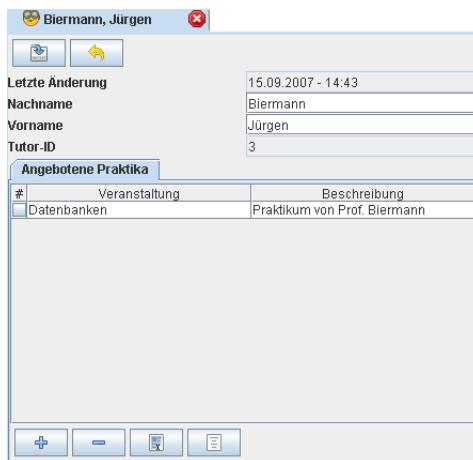


Abbildung 5.10: Ansicht aller betreuten Veranstaltungen eines Tutors in “PrakMan”

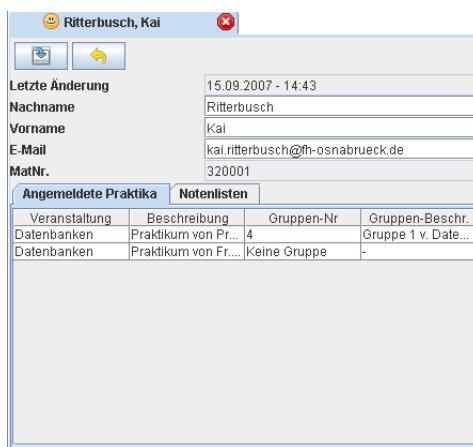


Abbildung 5.11: Ansicht aller gewählten Veranstaltungen eines Studenten in “PrakMan”

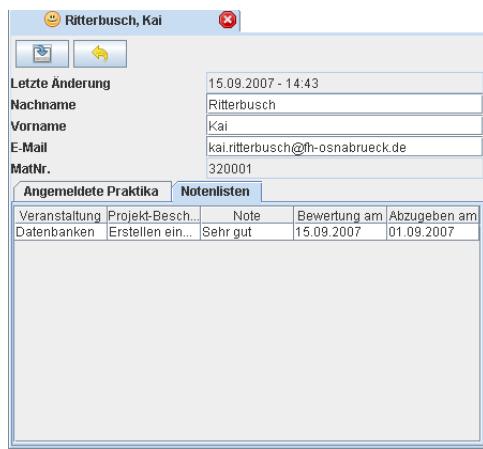


Abbildung 5.12: Ansicht aller bisher erreichten Noten eines Studenten in “PrakMan”

5.2 Datensicherung

Sämtliche Daten von “PrakMan” sind in der Datenbank gesichert und werden von dieser verwaltet. Eine eigene Datensicherung ist somit nicht notwendig, da die unterstützten Datenbanken eigene Backup-Methoden anbieten, welche genutzt werden sollten. Der CSV-Export macht es zudem möglich, manuell wichtige Daten zu extrahieren und eigenständig zu sichern.

Begriffserklärung

Event	Eine Veranstaltung oder ein Praktikum für eine gegebene Veranstaltung
PrakMan	Steht für Praktikums-Manager, das dokumentierte Programm.
Project	Eine Aufgabe, die an Studierende verteilt werden kann. Projekte sind notwendig, um Noten zu verteilen. Ein Projekt kann demnach auch das Bestehen der gesamten Veranstaltung darstellen.
SubVersion (SVN)	SubVersion ist eine Open-Source-Software zur Versionsverwaltung von Dateien und Verzeichnissen.