

OOAD Übungsblatt 3

Dominik Tödling, Lukas Neuhold, Christoph Huber,
Stefan Mitterrutzner, Emanuel Moser

25. Januar 2016

Abgabe 1

Relationales Schema

foerdergeber (foerdID, name, adresse)
foerdergeber_telno (foerdID, telno)
projekt (projektID, name, startdatum, enddatum)
industrieforschungsprojekt (projektID, deckungsbeitrag)
Grunlagenforschungsprojekt (projektID, ausschreibungsakzeptanzrate)
foerdert (foerdID, projektID)
partnerfirma (partnerID, name, adresse)
ist_zugeordnet (industrieprojektID, partnerID)
mitarbeiter (manr, name)
arbeitet_in_projekt (id, projektID, manr, von, bis)
abteilung (abtnr, name, manr)
arbeitet_in_abteilung (id, abtnr, manr, von, bis)

Abgabe 2

Business Case

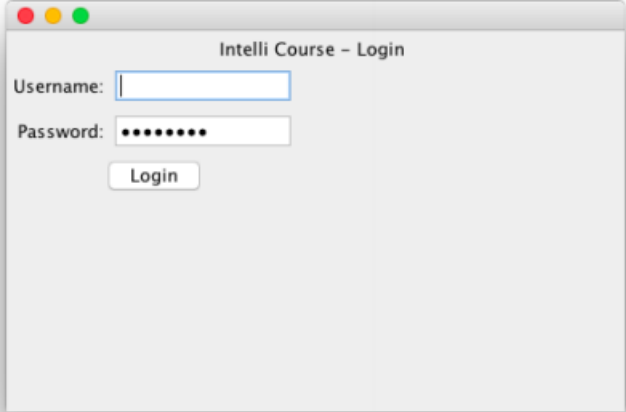
IntelliCourse ist eine Software zur Zeitplanung von Kursen. Durch die automatische Zeitplanerstellung kann sich jeder ganz einfach seinen eigenen Zeitplan einsehen. Ein jeder User kann einen individualisierten Kalender mit allen Terminen ganz einfach einsehen. Beim erstellen eines Events wird automatisch auf Zeit- und Raumüberschneidungen überprüft, und gegebenenfalls ein Warning ausgegeben, und es wird automatisch versucht alles so anzupassen, dass es zu keiner Überschneidung kommt. Durch diese automatisierte Einteilung von Terminen auf mögliche Räume und Timeslots wird sich viel an administrativem Aufwand erspart, was ermöglicht die Kosten in der Administration zu senken. Zusätzlich ist es möglich das Lehrende Zeitpräferenzen einzugeben, die in die Erstellung des Zeitplans miteinbezogen werden. Im Falle dass ein Termin zu einem bestimmten Zeitpunkt stattfinden MUSS, kann ein Administrator den Zeitplan immer manuell bearbeiten.


Use Cases

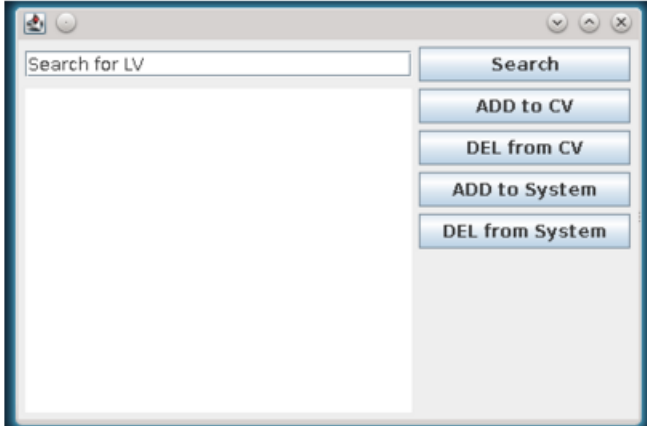
Priorisierung und Szenarien

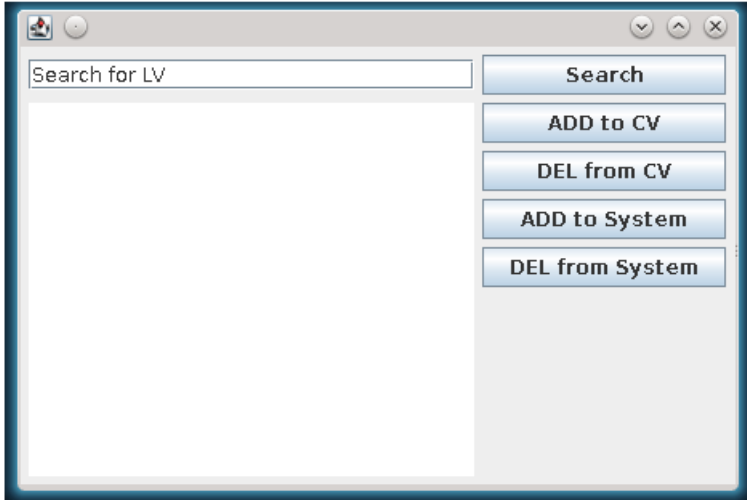
Student	Score	Lehrender	Score	Admin	Score
Anmelden	10	Anmelden	10	Anmelden	10
Für Veranstaltung registrieren	10	Für Veranstaltung registrieren	10	Account erstellen	10
Account erstellen	10	Account erstellen	10	Accounts autorisieren	10
Kalender einsehen	9	Prüfung erstellen	10	Raum erstellen	10
Event erstellen	9	Kurs erstellen	10	Für Veranstaltung registrieren	10
Event löschen	9	Kalender einsehen	9	Kurs erstellen	10
		Prüfung löschen	9	Prüfung erstellen	10
		Kurs löschen	9	Event erstellen	9
		Event löschen	9	Event löschen	9
				Kurs löschen	9
				Prüfung löschen	9
				Raum löschen	9
				Kalender einsehen	9


Use Case Spezifizierung

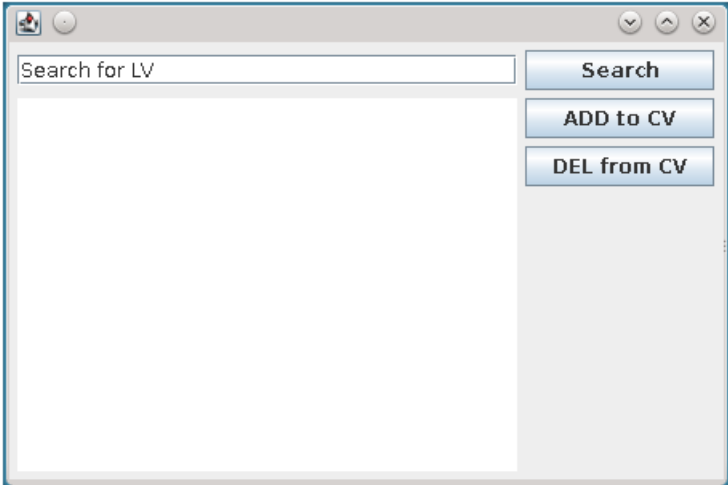
Use Case: Anmelden	Priorität: 10
<i>Beschreibung</i>	
Wie ein User sich am System anmeldet	
<i>Scenario 1</i>	
User startet das Programm	
System fragt nach Benutzernamen und Passwort	
User gibt richtige Daten ein und klickt 'Anmelden'	
System zeigt nächstes Fenster an	
<i>Scenario 2</i>	
User startet das Programm	
System fragt nach Benutzernamen und Passwort	
User gibt inkorrekte Daten ein und klickt 'Anmelden'	
System gibt ' Falscher Benutzernahme oder Passwort' zurück	
	

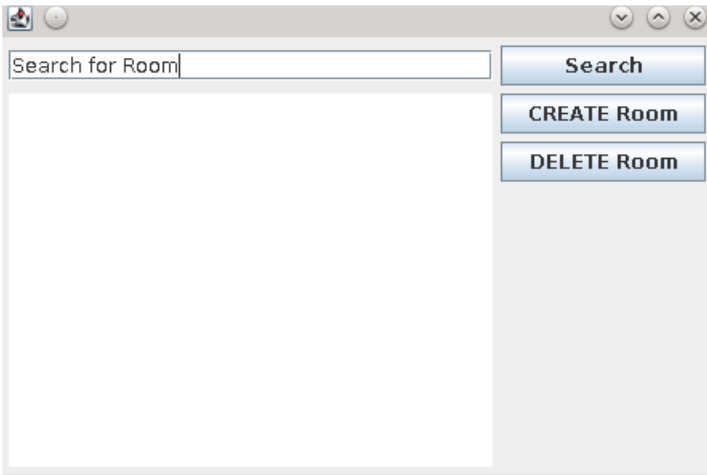
Use Case: Account erstellen	Priorität: 10
<i>Beschreibung</i> Wie ein User sich am System registriert	
<i>Scenario 1</i> User klickt auf 'Registrieren' System fragt nach Namen, Benutzernamen, Passwort, Geburtsdatum, email und Handynummer User gibt Daten ein und klickt auf 'Registrieren' Daten werden überprüft, Account angelegt und der User weitergeleitet	
<i>Scenario 2</i> User klickt auf 'Registrieren' System fragt nach Namen, Benutzernamen, Passwort, Geburtsdatum, email und Handynummer User gibt Daten ein und klickt auf 'Registrieren' Benutzername ist bereits vergeben, System gibt 'Benutzername bereits vergeben!' zurück	
	
Use Case: Account autorisieren	Priorität: 9
<i>Beschreibung</i> Wie ein Admin accounts autorisiert	
<i>Scenario 1</i> User klickt auf "Manage Accounts" User wählt Account und klickt "Make Teacher" Das System registriert den User als Lehrenden	
<i>Scenario 2</i> User klickt auf "Manage Accounts" User wählt Account und klickt "Make Administrator" Das System registriert den User als Administrator	
UI TBD	

Use Case: Veranstaltung erstellen	Priorität: 10
<i>Beschreibung</i> Wie ein User eine neue Veranstaltung erstellt	
<i>Scenario 1</i> User klickt auf 'Veranstaltungen' User klickt auf 'Neue Veranstaltung' System fragt nach Namen, Lehrenden und Präferenzen bzgl. Zeit und Raum User klickt auf 'Erstellen' und die Veranstaltung wird erstellt	
<i>Scenario 2</i> User klickt auf 'Veranstaltungen' User klickt auf 'Neue Veranstaltung' System fragt nach Namen, Lehrenden und Präferenzen bzgl. Zeit und Raum User klickt auf 'Erstellen' System überprüft Daten und findet Kollision mit bereits bestehenden Veranstaltungen System gibt Warnung aus dass von den Präferenzen abgewichen werden muss User klickt auf 'Abbrechen'	
<i>Scenario 3</i> User klickt auf 'Veranstaltungen' User klickt auf 'Neue Veranstaltung' System fragt nach Namen, Lehrenden und Präferenzen bzgl. Zeit und Raum User klickt auf 'Erstellen' System überprüft Daten und findet Kollision mit bereits bestehenden Veranstaltungen System gibt Warnung aus dass von den Präferenzen abgewichen werden muss User klickt auf 'OK' und die Veranstaltung wird mit veränderten Parametern erstellt Das System gibt den neuen Raum/die neue Zeit zurück	
	

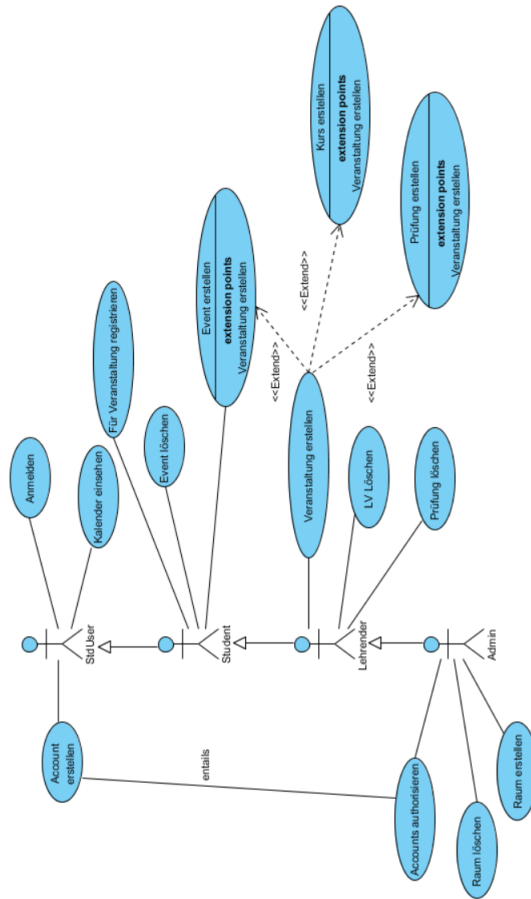
Use Case: Veranstaltung stornieren	Priorität: 9
<i>Beschreibung</i>	
Wie ein Administrator/Ersteller der Veranstaltung eine Veranstaltung storniert	
<i>Scenario 1</i> User klickt auf 'Veranstaltungen' User wählt eine Veranstaltung und klickt auf 'Verwalten' User klickt auf 'Stornieren' System löscht die Veranstaltung und gibt Erfolgsmeldung zurück	
<i>Scenario 2</i> User klickt auf 'Veranstaltungen' User wählt eine Veranstaltung und klickt auf 'Verwalten' User klickt auf 'Stornieren' System gibt Warnung aus dass Personen zu dieser Veranstaltung angemeldet sind User klickt auf 'OK' Registrierte Personen werden abgemeldet und die Veranstaltung gelöscht	
	

Use Case: Für Veranstaltung registrieren	Priorität: 10
<i>Beschreibung</i> Wie ein User sich für Veranstaltungen registriert	
<i>Scenario 1</i> User klickt auf 'Veranstaltungen' User wählt Veranstaltung aus und klickt auf 'Anmelden' User wird für Veranstaltung eingetragen	
<i>Scenario 2</i> User klickt auf 'Veranstaltungen' User wählt Veranstaltung aus und klickt auf 'Anmelden' User wird für Veranstaltung eingetragen System findet überschneidung mit Kalender des Users und gibt Warnung aus	
	

Use Case: Veranstaltung suchen	Priorität: 10
<i>Beschreibung</i> Wie ein User Veranstaltungen nach Namen sucht	
<i>Scenario 1</i> User klickt auf 'Veranstaltungen' User gibt Suchbegriffe in das Suchfeld ein und klickt auf 'Suchen' Das System zeigt eine gefilterte Liste von Veranstaltungen an	
<i>Scenario 2</i> User klickt auf 'Veranstaltungen' User gibt Suchbegriffe in das Suchfeld ein und klickt auf 'Suchen' Das System findet keine Veranstaltungen und gibt 'Keine Veranstaltungen gefunden' zurück	
	

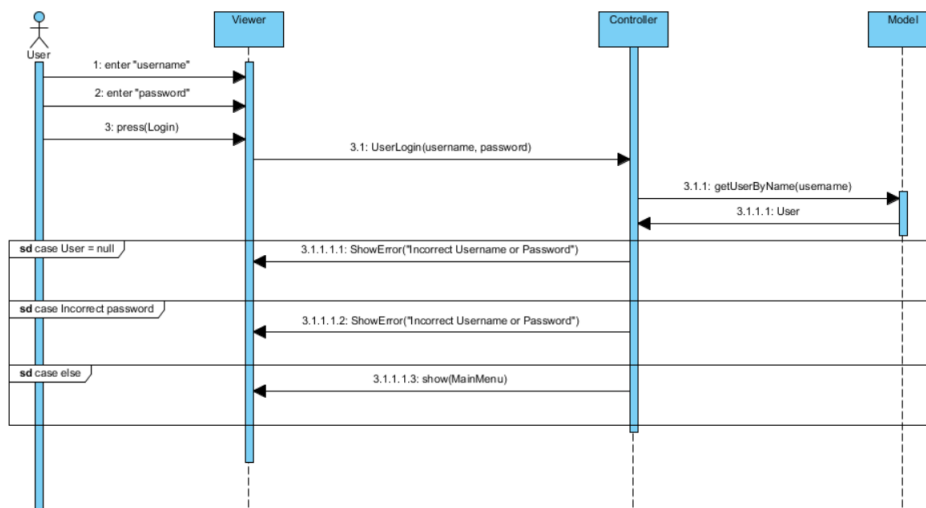
Use Case: Raum hinzufügen	Priorität: 10
<i>Beschreibung</i> Wie ein Administrator einen Raum hinzufügt	
<i>Scenario 1</i> User klickt auf 'Räume verwalten' User klickt auf 'Neuer Raum' System fragt nach Namen und Kapazität User gibt Daten ein und klickt auf 'Erstellen' Der Raum wird erstellt	
<i>Scenario 2</i> User klickt auf 'Räume verwalten' User klickt auf 'Neuer Raum' System fragt nach Namen und Kapazität User gibt Daten ein und klickt auf 'Erstellen' Das System überprüft die Daten und findet einen Raum mit gleichem Namen Das System gibt eine Fehlermeldung zurück	
	
Use Case: Kalender einsehen	Priorität: 8
<i>Beschreibung</i> Wie ein User sich den Kalender ansieht	
<i>Scenario 1</i> User klickt auf 'Kalender' Aus Terminen wird graphischer Kalender erzeugt und in einem neuen Fenster angezeigt	
UI TBD	

Use Case Diagramm

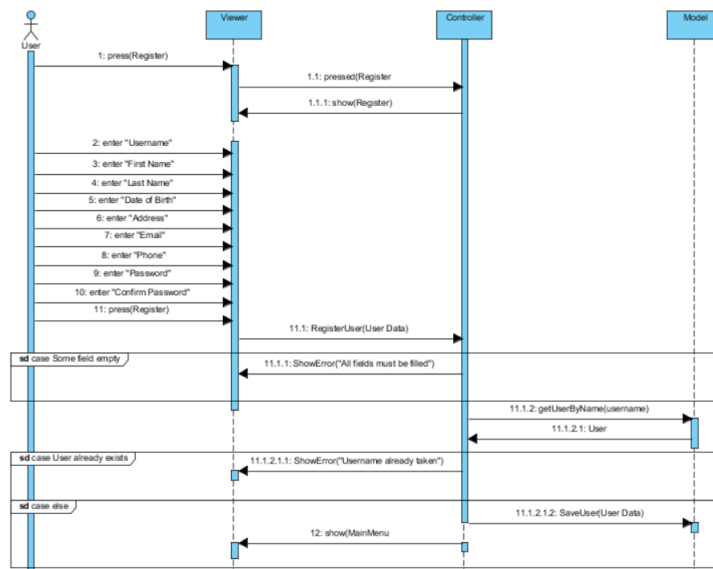


Sequenzdiagramme

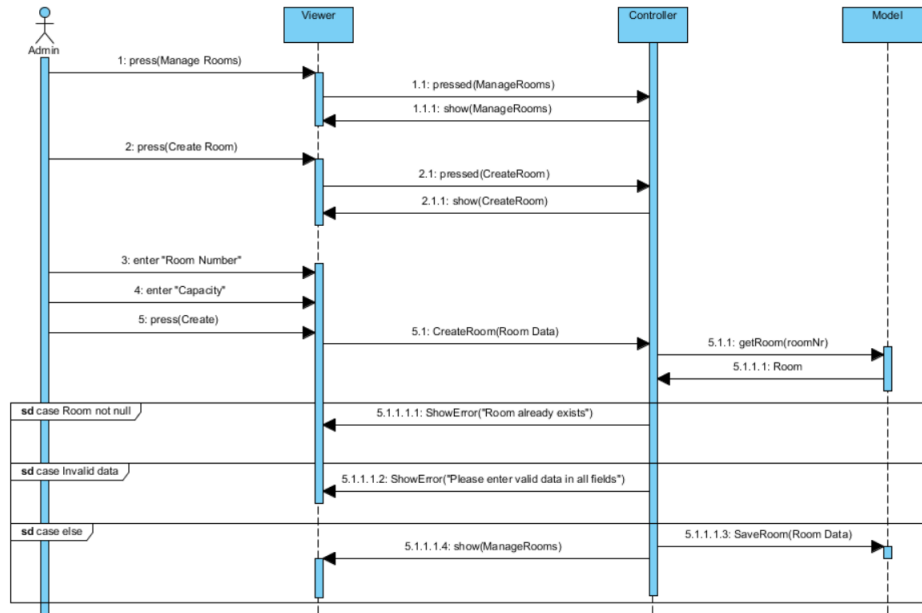
Login



Registrieren

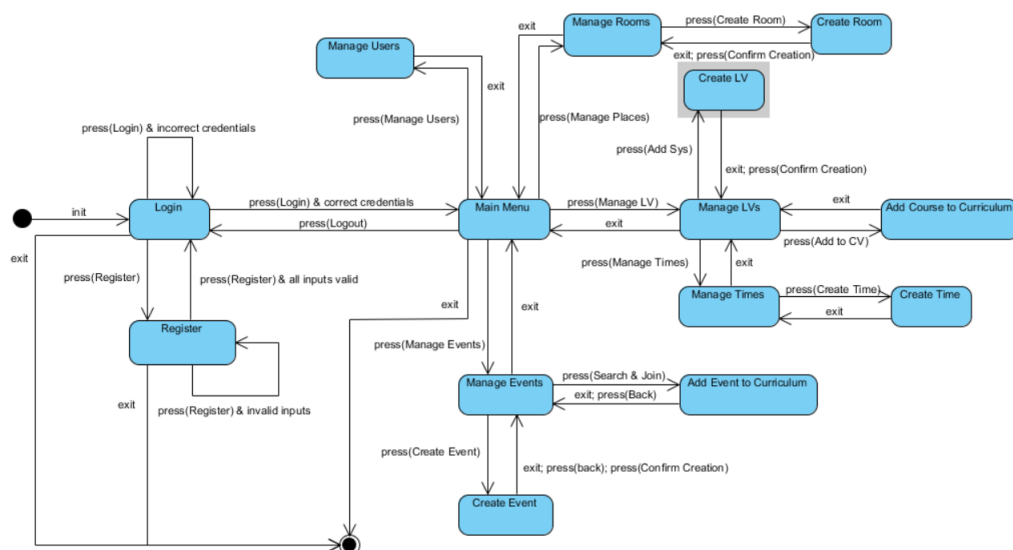


Raum erstellen

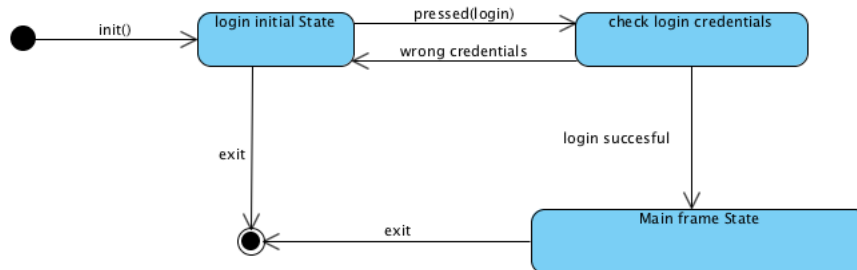


State Machine Diagramme

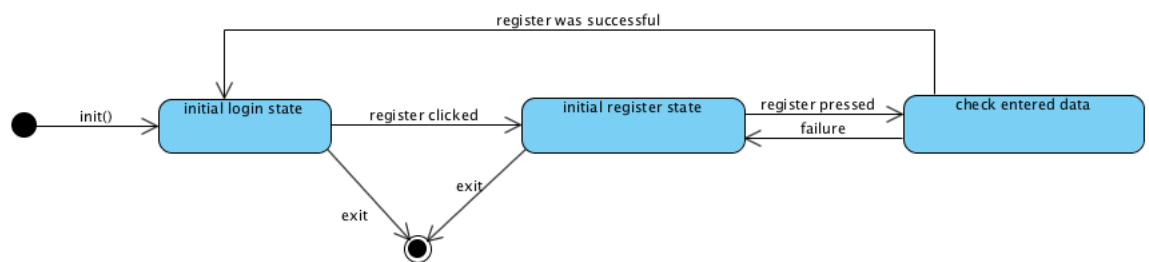
Gesamtes Programm



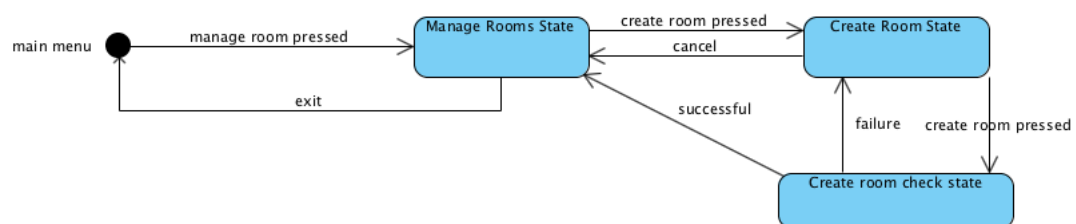
Login



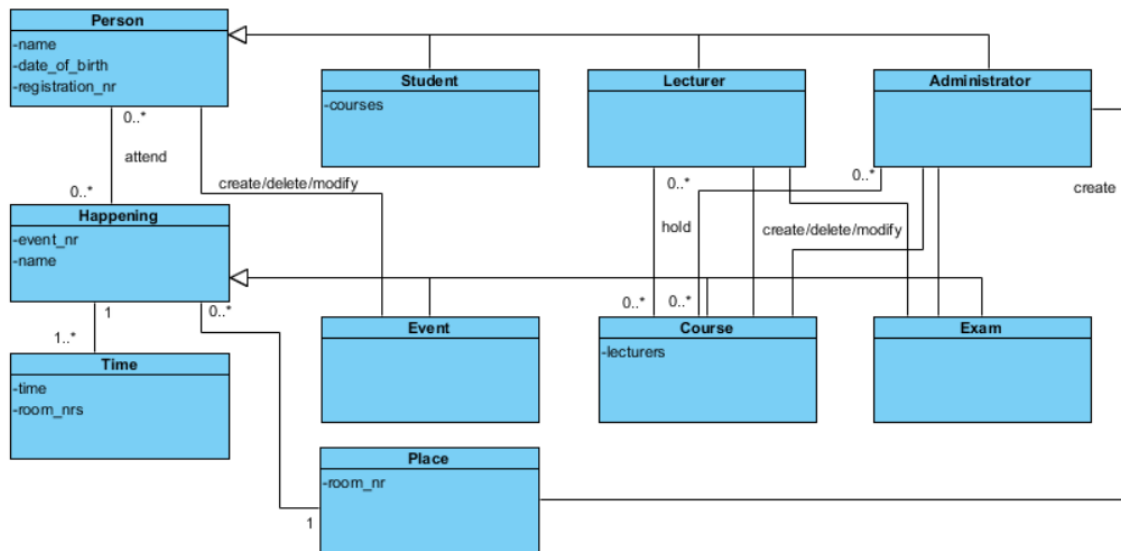
Registrieren



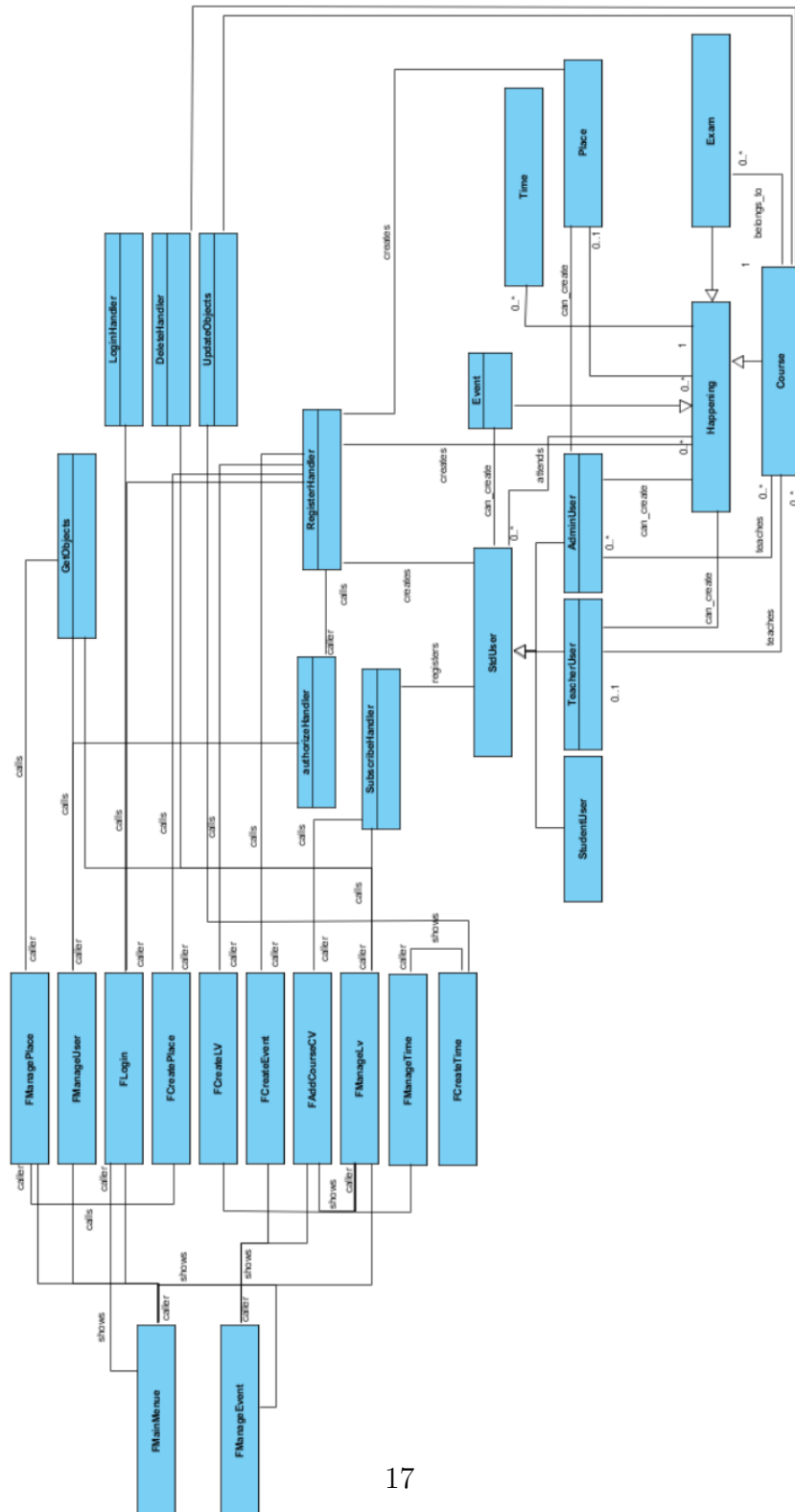
Raum erstellen



UML Analyse-Klassendiagramm

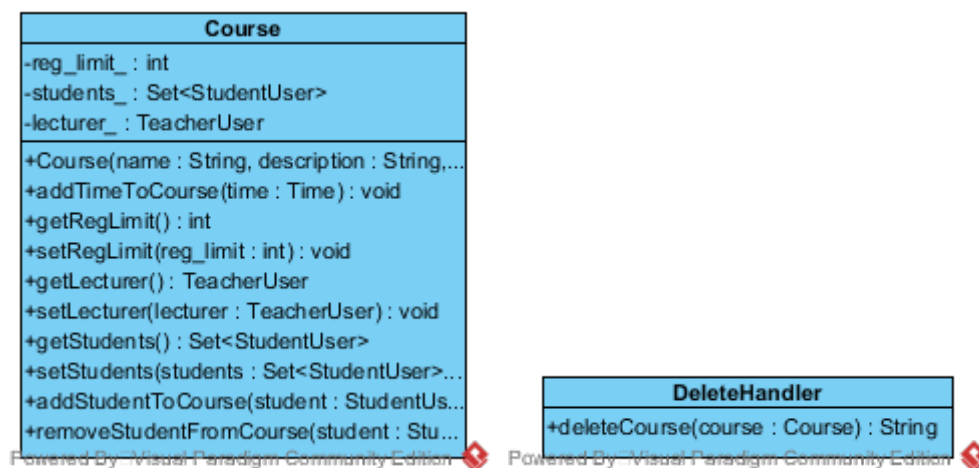
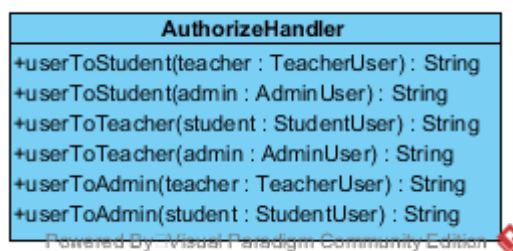
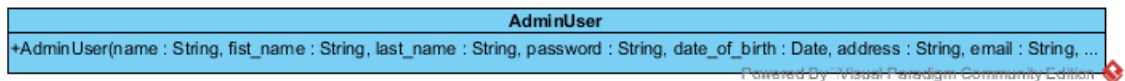


UML Design-Klassendiagramm



Einzelne Klassen

Aus Gründen der Übersichtlichkeit wurden im obigen Diagramm die genauen Klassendefinitionen weggelassen und sind im folgenden zu sehen:



Exam
-course_ : String
+getCourse() : String
+setCourse(course : String) : void

Powered By Visual Paradigm Community Edition

FAddCourseCV
-content_panel_ : JPanel
-jt_search_lv : JTable
-sp_search_lv : JScrollPane
-btn_add_lv_cv : JButton
-btn_back_ : JButton
-courses : List<Course>
-student_ : StudentUser
-teacher_ : TeacherUser
-admin_ : AdminUser
+FAddCourseCV()
+FAddCourseCV(student : StudentUser)
+FAddCourseCV(teacher : teacherUser)
+FAddCourseCV(admin : adminUser)
-init() : void
-createAddLvInterface() : void

Powered By Visual Paradigm Community Edition

FCreateEvent
-content_panel_ : JPane
-lb_event_name_ : JLabel
-lb_event_max_students_ : JLabel
-lb_event_description_ : JLabel
-text_event_name_ : JTextField
-text_event_max_students_ : JFormatt...
-text_event_description_ : JTextField
-btn_confirm_creation_ : JButton
-btn_back_ : JButton
+FCreateEvent()
-createPanel() : void

Powered By Visual Paradigm Community Edition

FCreateLV
-content_panel_ : JPanel
-lb_l_name_ : JLabel
-lb_l_teacher_ : JLabel
-lb_l_max_students_ : JLabel
-lb_l_description_ : JLabel
-text_l_name_ : JTextField
-text_l_max_students_ : JFormattedTe...
-cb_teacher_ : JComboBox<TeacherU...
-teachers : List<TeacherUser>
-text_l_description_ : JTextField
-btn_confirm_creation_ : JButton
+FCreateLv()
-createPanel() : void

Powered By Visual Paradigm Community Edition

FCreatePlace
-content_panel_ : JPanel
-lb_l_room_nr_ : JLabel
-lb_l_capacity_ : JLabel
-text_l_room_nr_ : JFormattedTextField
-text_l_capacity_ : JFormattedTextField
-btn_confirm_creation_ : JButton
+FCreatePlace()
-createPanel()

Powered By Visual Paradigm Community Edition

FCreateTime
-content_panel_ : JPanel
-lb_l_date_ : JLabel
-lb_l_start_time_ : JLabel
-lb_l_end_time_ : JLabel
-date_picker_ : JDatePicker
-text_l_start_time_ : JTextField
-text_l_end_time_ : JTextField
-btn_confirm_creation_ : JButton
+FCreateTime(happening : Happening)
-createPanel() : void

Powered By Visual Paradigm Community Edition

FLogin
-content_panel_ : JPanel
-register_panel_ : JPanel
-lb_name_ : JLabel
-lb_passwd_ : JLabel
-text_f_name_ : JTextField
-pass_f_passwd_ : JPasswordField
-btn_confirm_login_ : JButton
-btn_register_ : JButton
-lb_name_ : JLabel
-lb_rfirst_name_ : JLabel
-lb_rlast_name_ : JLabel
-lb_date_of_birth_ : JLabel
-lb_adress_ : JLabel
-lb_email_ : JLabel
-lb_phone_ : JLabel
-text_f_name_ : JTextField
-text_f_rfirst_name_ : JTextField
-text_f_rlast_name_ : JTextField
-text_f_date_of_birth_ : JTextField
-text_f_adress_ : JTextField
-text_f_email_ : JTextField
-text_f_phone_ : JTextField
-lb_rpasswd_ : JLabel
-lb_rpasswd_confirmed_ : JLabel
-pass_f_rpasswd_ : JPasswordField
-pass_f_rpasswd_confirmed_ : JPasswo...
-btn_confrim_register_ : JButton
+FLogin()
-createLoginPanel() : void
-createRegisterPanel() : void

FMainMenu
-content_panel_ : JPanel
-lb_header_info_ : JLabel
-lb_header_actions_ : JLabel
-lb_t_name_ : JLabel
-lb_t_user_type_ : JLabel
-lb_name_ : JLabel
-lb_user_type_ : JLabel
-btn_logout_ : JButton
-btn_update_user_ : JButton
-btn_manage_lv_ : JButton
-btn_manage_events_ : JButton
-btn_calendar_ : JButton
-btn_manage_users_ : JButton
-btn_manage_places_ : JButton
-lb_upcoming_events_ : JLabel
-jt_upcoming_events_ : JTable
-sp_upcoming_events_ : JScrollPane
-student_ : StudentUser
-teacher_ : TeacherUser
-admin_ : AdminUser
+FMainMenu(student : StudentUser)
+FMainMenu(teacher : TeacherUser)
+FMainMenu(admin : AdminUser)
-init() : void
-createContent() : void

Powered By Visual Paradigm Community Edition



Powered By Visual Paradigm Community Edition



FManageEvent
-content_panel_ : JPanel -text_f_sevent_ : JTextField -btn_s_event : JButton -jt_current_events : JTable -sp_current_events : JScrollPane -btn_more_info_ : JButton -btn_create_event_ : JButton -btn_del_event_ : JButton -student_ : StudentUser -teacher_ : TeacherUser -admin_ : AdminUser
+FManageEvent(student : StudentUser) +FManageEvent(teacher : TeacherUser) +FManageEvent(admin : AdminUser) -init() : void -createEventInterface() : void

Powered By Visual Paradigm Community Edition

FManageLv
-content_panel_ : JPanel -text_f_slv_ : JTextField -btn_s_lv_ : JButton -jt_current_lvs_ : JTable -jtm_current_lvs_ : JDefaultTableModel -jt_timer_ : JTimer -sp_current_lvs_ : JScrollPane -btn_add_curriculum : JButton -btn_del_curriculum : JButton -btn_add_system : JButton -btn_del_system : JButton -student_ : StudentUser -teacher_ : TeacherUser -admin_ : AdminUser -courses_student_subscribed_to : List<...
+FManageLv(student : StudentUser) +FManageLv(teacher : TeacherUser) +FManageLv(admin : AdminUser) -init() : void -createLvInterface() : void -fillTable() : void

Powered By Visual Paradigm Community Edition

FManagePlace
-content_panel_ : JPanel -jt_current_places_ : JTable -jtm_current_places_ : DefaultTableMo... -jt_timer_ : Timer -btn_del_place_ : JButton -btn_add_place_ : JButton -admin_ : AdminUser
+FManagePlace(admin : AdminUser) -init() : void -createManagePlaceInterface() : void -fillTable() : void

Powered By Visual Paradigm Community Edition

FManageTime
-content_panel_ : JPanel -jt_current_lv_time_ : JTable -jtm_current_lv_time_ : DefaultTableM... -jt_timer_ : Timer -sp_current_lv_times_ : JScrollPane -btn_add_time_ : JButton -btn_delete_time_ : JButton -course_ : Course
+FManageTime(course : Course) -init() : void -createTimeInterface() : void -fillTable() : void

Powered By Visual Paradigm Community Edition

FManageUser
-content_panel_ : JPanel -text_f_suser_ : JTextField -btn_s_user_ : JButton -jt_search_output_users_ : JTable -jtm_search_output_users_ : JDefaultT... -sp_search_output_users_ : JScrollPane -btn_make_student_ : JButton -btn_make_teacher_ : JButton -btn_make_admin_ : JButton -student_ : StudentUser -teacher_ : TeacherUser -admin_ : AdminUser -students_ : List<StudentUser> -teachers_ : List<TeacherUser> -admins_ : List<AdminUser>
+FManageUser(admin : AdminUser) -init() : void -createManageUserInterface() : void -fillTable() : void

GetObjects
+getStudentByName(name : String) : ... +getAllStudents() : List<StudentUser> +getTeacherByName(name : String) : ... +getAllTeachers() : List<TeacherUser> +getAdminByName(name : String) : A... +getAllAdmins() : List<AdminUser> +getPlaceByNumber(number : int) : P... +getAllPlaces() : List<Place> +getCourseByName(name : String) : ... +getAllCourses() : List<Course>

Powered By Visual Paradigm Community Edition

Happening
-id_ : int -name_ : String -description : String
+getID() : int +getName() : String +setName(name : String) : void +getDescription() : String +setDescription(description : String) : void

Powered By Visual Paradigm Community Edition

LoginHandler
+studentLogin(username : String, password : String) : StudentUser +teacherLogin(username : String, password : String) : TeacherUser +adminLogin(username : String, password : String) : AdminUser

Powered By Visual Paradigm Community Edition

Place
-id_ : int -room_nr : int -capacity : int -name_ : String
+Place(room_nr : int, capacity : int) +getRoomNr() : int +getCapacity() : int +setCapacity(capacity : int) : void +getName() : String +setName(name : String) : void

Powered By Visual Paradigm Community Edition

RegisterHandler
+registerUser(name : String, fist_name : String, last_name : String, password : String, date_of_birth : Date, address : String, email : String, phone : String) : String +registerPlace(name : String, room_nr : int, capacity : int) : String +registerCourse(name : String, description : String, reg_limit : int, lecturer : TeacherUser) : String

Powered By Visual Paradigm Community Edition

StdUser
-id_ : int -first_name_ : String -last_name_ : String -name_ : String -password_ : String -date_of_birth_ : Date -registration_nr_ : int -address_ : String -email_ : String -phone_ : String
+toString() : String +operation() +getFirstName() : String +setFirstName(first_name : String) : void +getLastName() : String +setLastName(last_name : String) : void +getName() : String +setName(name : String) : void +getPassword() : String +setPassword(password : String) : void +getDateOfBirth() : Date +setDateOfBirth(date_of_birth : Date) : void +getRegistrationNr() : int +setRegistrationNr(registration_nr : int) : void +getAddress() : String +setAddress(address : String) : void +getEmail() : String +setEmail(email : String) : void +getPhone() : String +setPhone(phone : String) : void +getID() : int

Powered By Visual Paradigm Community Edition

StudentUser
-field_of_study_ : String
+StudentUser(name : String, fist_name : String, last_name : String, password : String, date_of_birth : Date, address : Stri... +getFieldOfStudy() : String +setFieldOfStudy(field_of_study : String) : void +getCourses() : Set<Course> +setCourses(courses : Set<Course>) : void +addCourseToStudent(course : Course) : void +removeCourseFromStudent(course : Course) : void

Powered By Visual Paradigm Community Edition

SubscribeHandler
+registerForCourse(user : StdUser, course : Course) : String
+unregisterFromCourse(user : StdUser, course : Course) : String

Powered By Visual Paradigm Community Edition

TeacherUser
+TeacherUser(name : String, first_name : String, last_name : String, password : String, date_of_birth : Date, address : String, email : String, phone : String)
+getTeachingCourses() : List<Course>
+setTeachingCourses(courses : List<Course>) : void

Powered By Visual Paradigm Community Edition

UpdateObjects
+updateCourse(course : Course, name : String, desc : String, lecturer : TeacherUser, start_time : Date, end_time : Date) : String

Powered By Visual Paradigm Community Edition

Projektplan

Accounts erstellen	fertig
Räume erstellen	fertig
Veranstaltungen erstellen	fertig
Veranstaltung suchen	fertig
Für Veranstaltungen registrieren/abmelden	fertig
Berechtigungslevel für User	fertig
Authorisierung von Usern	fertig
Veranstaltungen suchen	fertig
Manuelle verschiebung von Veranstaltungen	fertig
Zeitplan einsehen	fertig

Projektbericht

Unser Ziel mit diesem Projekt war es ein System zu entwickeln mit dem es möglich ist Kurse und Prüfungen, beziehungsweise Anmeldungen and diesen, einfach zu verwalten. Dabei mussten wir im Laufe der Entwicklung aufgrund von zeitlichen und technischen Problemen ein paar der höheren Ziele (wie etwa Zeitpräferenzen für Vortragende oder ein Semestersystem) weglassen und haben uns stattdessen auf die Grundfunktionalitäten fokussiert.

Die größten Probleme bei der Entwicklung wurden von der Datenbankanbindung ausgelöst und große Teile des Programmes mussten neu überarbeitet werden um bestimmte Fehler zu lösen. Erwähnenswert wäre hier zum Beispiel die Art und Weise wie Berechtigungslevel implementiert sind. Der ursprüngliche Ansatz nutzte eine Vererbungskette, da ein Lehrender ja alles darf was ein Student darf usw. Dies verursachte jedoch beim Löschen und Suchen nach Personen einige Probleme und wurde im Endeffekt durch eine Einfache Vererbung ersetzt bei der jede der drei Klassen von einer gemeinsamen Klasse abgeleitet war.