

Software Engineering Übung

Gruppe: 11

Übungsleiter: Michael Jakl

Designmodell v.1.0

Projekttitel: MyEvents

Projekthomepage:

https://cewebs.cs.univie.ac.at/SWE/ws16/index.php?m=D&t=info&c=show&CEWebS_c=g050052-11t7

Gruppenmitglieder:

MatNr:	Nachname:	Vorname:	e-mail:
1307652	Heitger	Katharina	a1307652@unet.univie.ac.at
1200478	Romstorfer	Markus Josef	a1200478@unet.univie.ac.at
1468239	Zinatulin	Ayrat	a1468239@unet.univie.ac.at
1502050	Zvonek	Jakub	a1502050@unet.univie.ac.at

Erstellen sie ein Designmodell gemäß *Unified Process* das zumindest folgende Aspekte umfasst:

- Klassendesign
- Use-Case-Realization-Design
- Übersichtsklassendiagramm
- Architekturbeschreibungen

1 Klassendesign

Beschreiben sie die wichtigsten Klassen und spezifizieren sie die öffentlichen Methoden und Attribute. Begründen sie die grundlegenden Designentscheidungen.

User: jeder User hat als private Variablen email, Telefonnummer, seinen echten Namen, Passwort, userid, und den login-Namen.

User ist eine abstrakte Klasse und vererbt seine Attribute an Privatanutzer, Veranstalter, Admin und Analytiker.

Privatanutzer: jeder Privatanutzer hat zusätzlich zu den geerbten Attribute eine ArrayList, in welche eigene Bewertungen gespeichert werden und ein Attribut für den privaten Kalender. Es wird keine Privatkalendar Variable in der Klasse gespeichert sondern nur ein Integer Wert der als "Pointer" dient um den Privatkalendar vom Benutzer zu identifizieren. Die öffentlichen Methoden beschränken sich auf Getter und Setter und auf Methoden mit denen man die Klassenvariablen verwalten kann. Jegliche Funktionalität des Benutzers wird in eine separate Klasse ausgelagert. Die Klasse heißt PrivatanutzerManagement und enthält die Funktionalität mit der es dem Benutzer möglich ist, Tätigkeiten durchzuführen.

Veranstalter: jeder Veranstalter hat ein ArrayList von Veranstaltungen, in welcher die IDs der eigenen Veranstaltungen gespeichert werden. Zusätzlich zu diesen Attributen gibt es eine öffentliche Methode namens addIDtoMyEvents(Integer id) die es ermöglicht, Werte in die Liste einzufügen. Analog dazu gibt es noch die Methode removeIDfromMyEvents() die eine ID aus der Datenstruktur entfernt.

Admin: ein Admin hat momentan eine Variable namens Management. Die ganze Funktionalität wird auch hier in eine separate Klasse namens AdminManagement ausgelagert.

Analytiker: der Analytiker hat keine zusätzlichen privaten Variablen, auch hier wurde die Funktionalität in eine getrennte Klasse ausgelagert.

PrivaterKalender: Der Privatkalender hat in erster Linie eine Integer Variable ID die als Schlüssel dient. Es wird auch die Benutzerid vom Benutzer mitgespeichert. Es gibt eine ArrayListe namens public_events. In dieser werden nicht die Veranstaltungsobjekte direkt gespeichert sondern nur die IDs der Veranstaltungen. So muss man die Objekte nicht mehrmals speichern und man vermeidet Redundanzen.

Zuletzt gibt es noch eine ArrayList in der private Termine gespeichert werden. Im Vergleich zur vorherigen Vorgehensweise werden hier nicht die IDs der Objekte gespeichert sondern die Terminobjekte.

Als öffentliche Methoden gibt es addVeranstaltung(Veranstaltung) um Veranstaltungen zum privaten Kalender hinzuzufügen.

OeffentlicherKalender: der öffentliche Kalender hat eine private Liste aller Veranstaltungen sowie eine private Liste aus Strings mit allen Kategorien. Als öffentliche Methode gibt es zumindest getEventsforDate(Calendar) um sich die Veranstaltung für einen bestimmten Tag anzeigen zu lassen.

Veranstaltung: Jede Veranstaltung hat als private Attribute einen Namen, die Start- und Endzeiten, einen Ort, eine Platzanzahl, eine Kategorie, einen Identifier und eine Liste der Bewertungen.

Als öffentliche Methoden gibt es calcAvgBewertungen() mit welcher die durchschnittliche Bewertung einer Veranstaltung berechnet wird und addBewertung().

AnalytikerManagement: Momentan haben wir im AnalytikerManagement drei Methoden. Die erste calcAvgBewertungen() retourniert einen Doublewert mit dem Durchschnitt aller Bewertungen. Die zweite Funktion namens calcNumberOfAccounts() retourniert die Anzahl der Benutzerkonten. Die dritte und momentan letzte Methode namens sortVeranstalterByPop() retourniert eine ArrayListe mit allen Veranstaltern, diese sind in aufsteigender Reihenfolge nach Popularität sortiert.

PrivatnutzerManagement:

Im PrivatnutzerManagement kann der Benutzer eine Veranstaltungsid in seinen Privatkalender hinzufügen und dadurch wird er automatisch für die Veranstaltung angemeldet.

Er hat auch die Möglichkeit sich von Veranstaltungen abzumelden, indem die ID aus seiner ArrayListe gelöscht wird.

Die letzten zwei Methoden sind addPrivateEvent(Termin termin) und wurdeBewertet(Integer id). Bei der ersten fügt der Benutzer ein Terminobjekt in seine ArrayListe ein, dadurch wird es in seinen Privatkalender aufgenommen.

VeranstalterManagement:

Im Veranstaltermanagement ist es dem Veranstalter möglich neue Veranstaltungen zu erstellen und dadurch implizit in den öffentlichen Kalender hinzuzufügen. Analog dazu hat er die Möglichkeit eigene Veranstaltungen wieder zu löschen. Wir möchten beim Löschen einer Veranstaltung sicherstellen, dass alle Einträge in den Privatkalendern gelöscht werden. Nach Löschung einer Veranstaltung sollten keine Einträge im öffentlichen sowie im privaten Kalender vorzufinden sein.

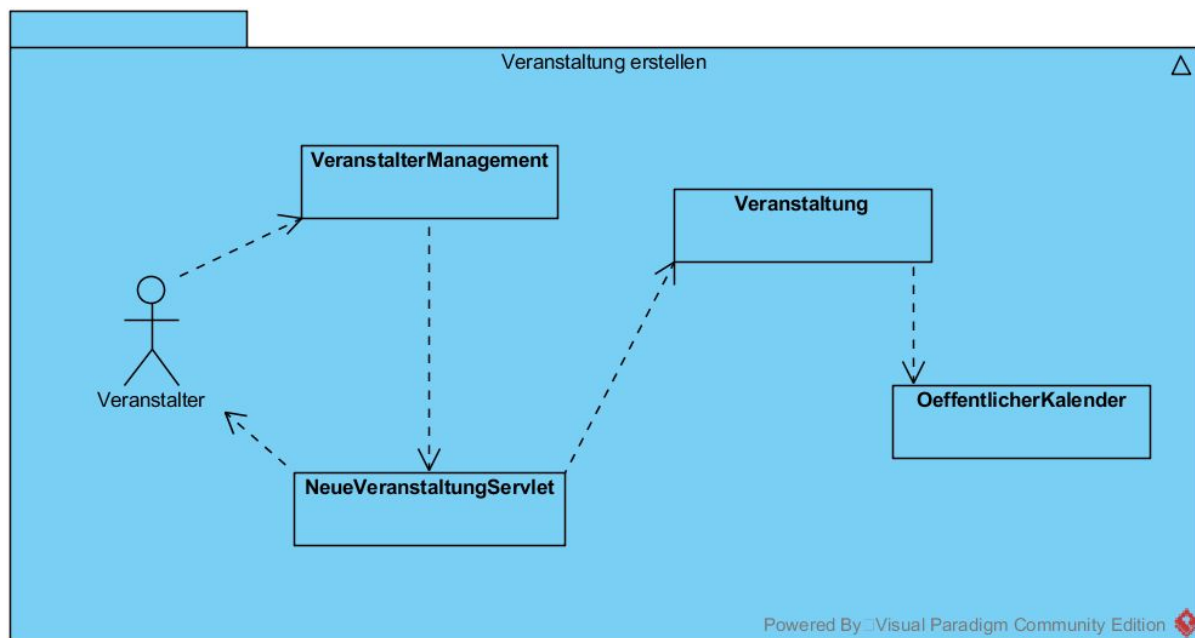
AdminManagement:

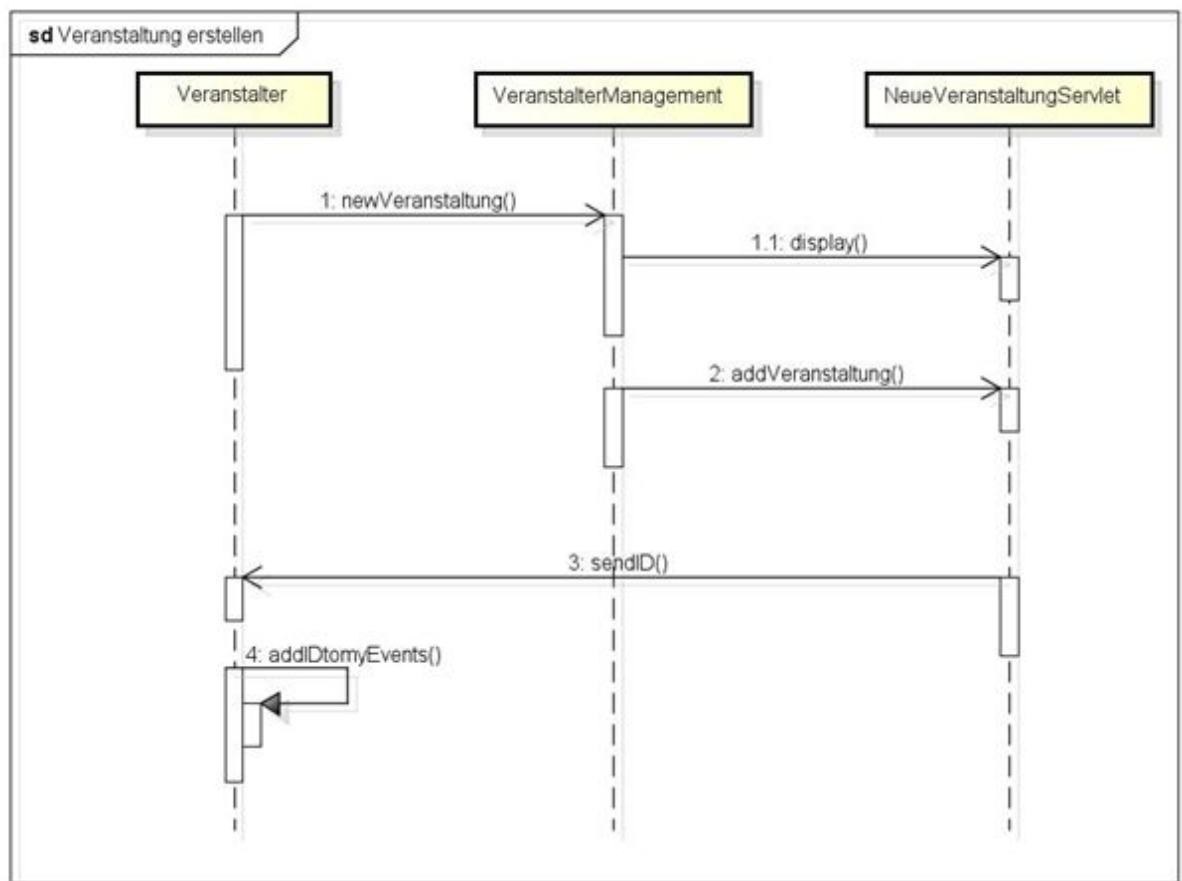
Im AdminManagement wird es dem Admin ermöglicht, neue Benutzer zu erstellen und zu löschen. Das betrifft alle Benutzertypen auch Admins direkt. Zuletzt gibt es noch die Möglichkeit einen jeweiligen Benutzer zu verwalten, indem man kontospezifische Details ändert.

2 Use Case Realization Design

Modellieren sie, wie die zentralen Use Cases ihres Systems mittels Designklassen realisiert werden. Modellieren sie dabei die wesentlichen statischen und dynamischen Aspekte jedes Use Cases mit geeigneten Klassendiagrammen und Sequenzdiagrammen.

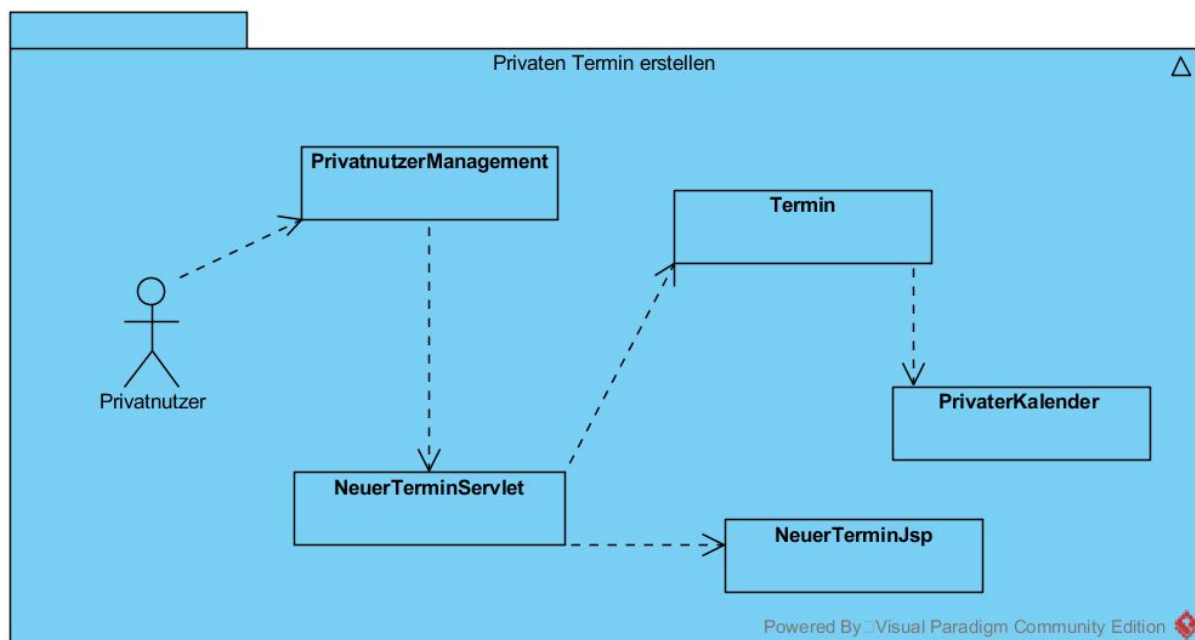
Veranstaltung erstellen:



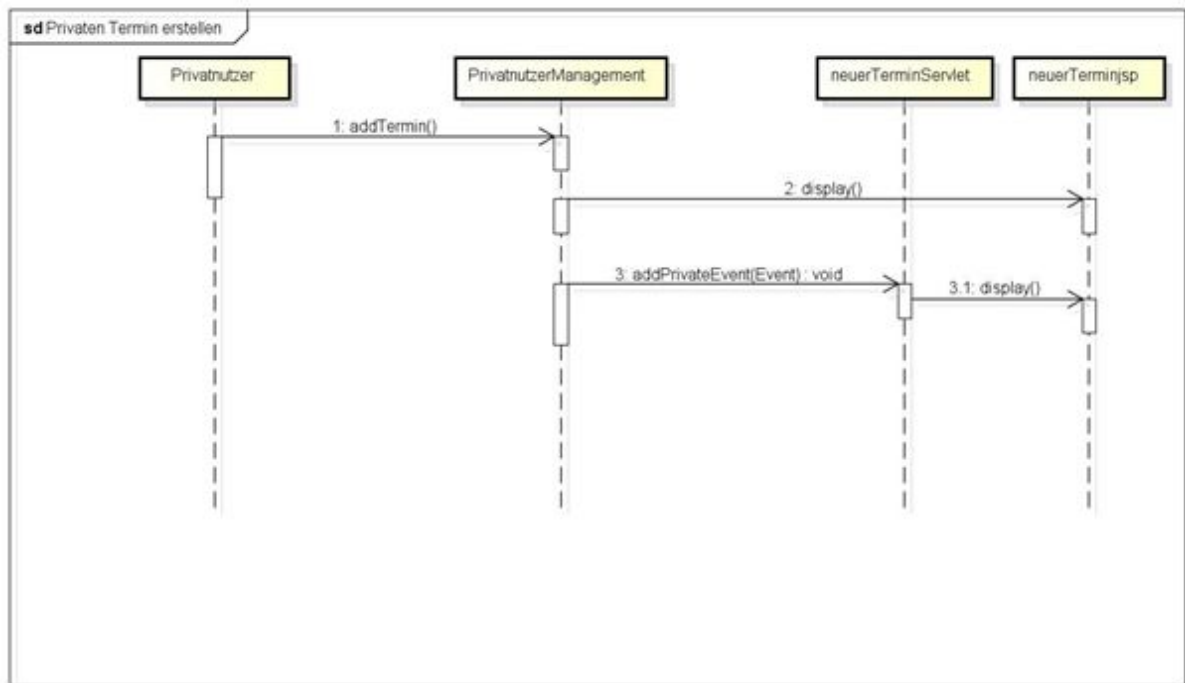


powered by Astah

Privaten Termin erstellen

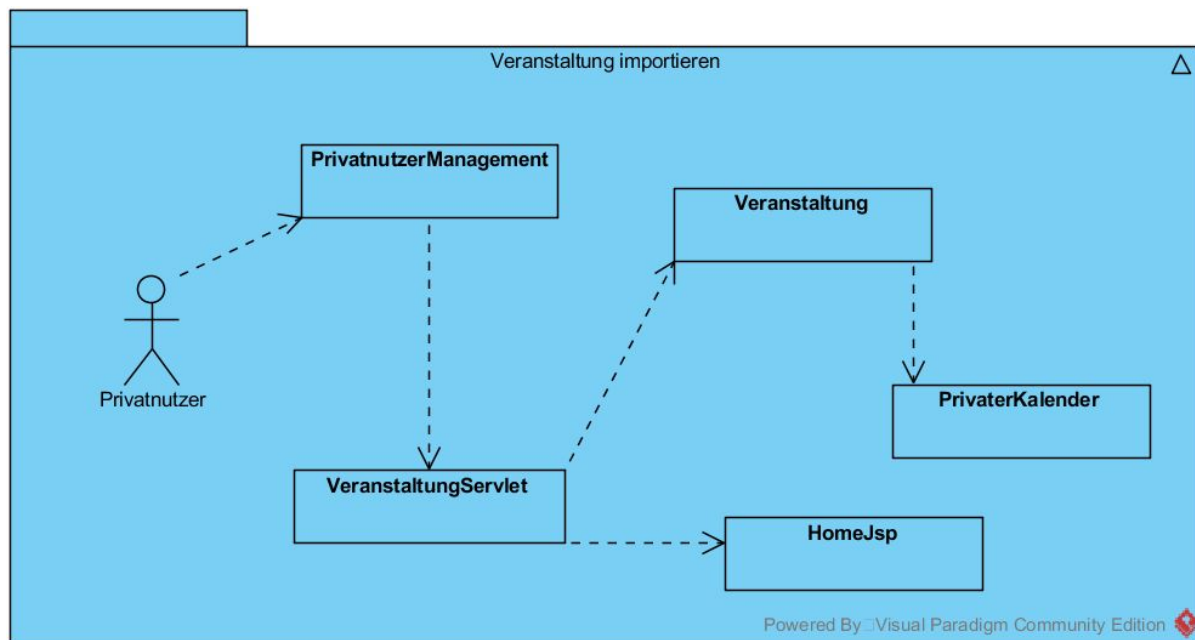


Powered By Visual Paradigm Community Edition

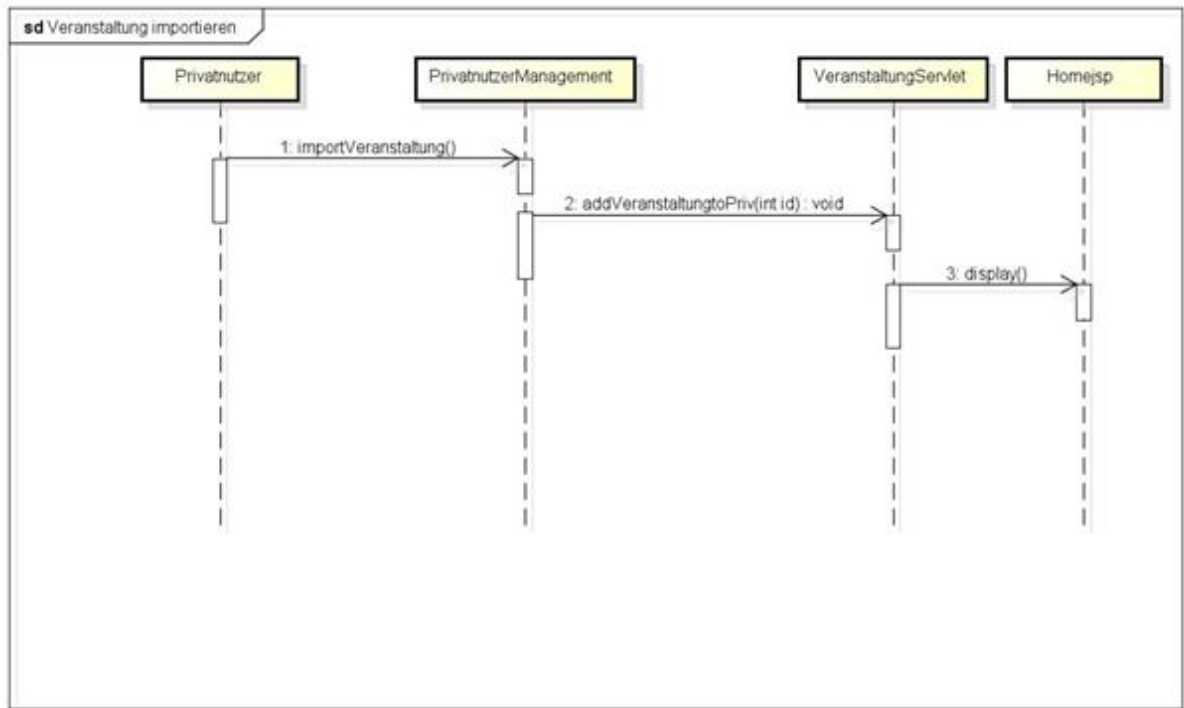


powered by Astah

Veranstaltung importieren

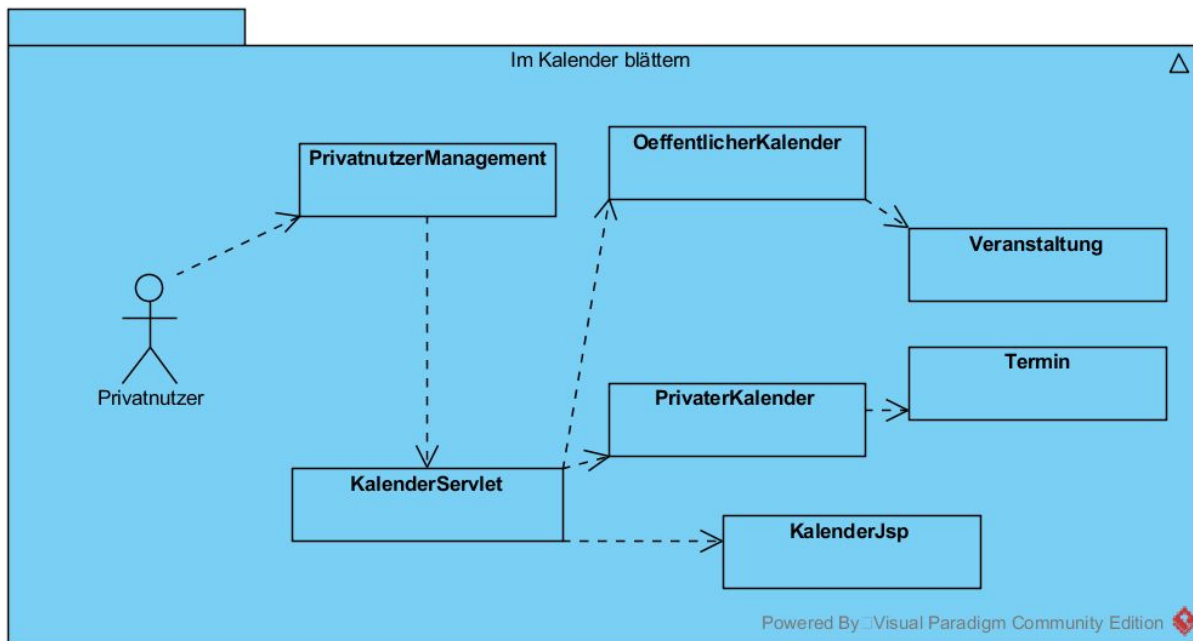


Powered By Visual Paradigm Community Edition

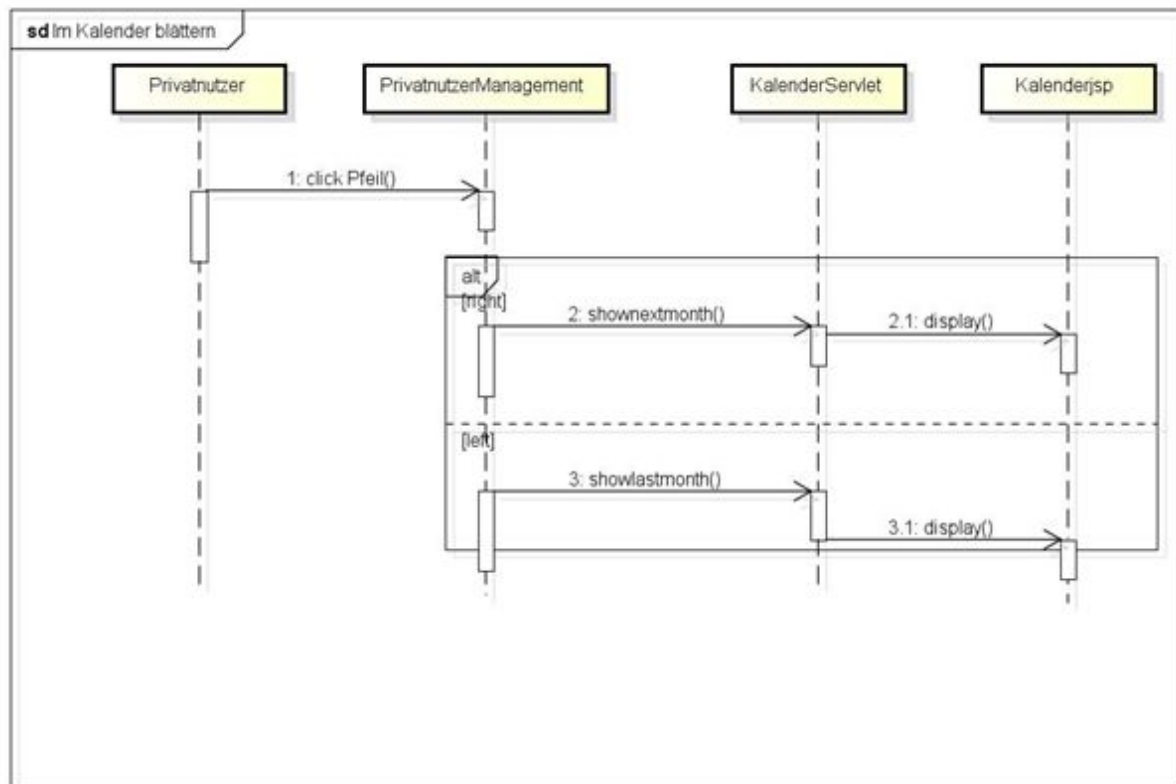


powered by Astah

Im Kalender blättern

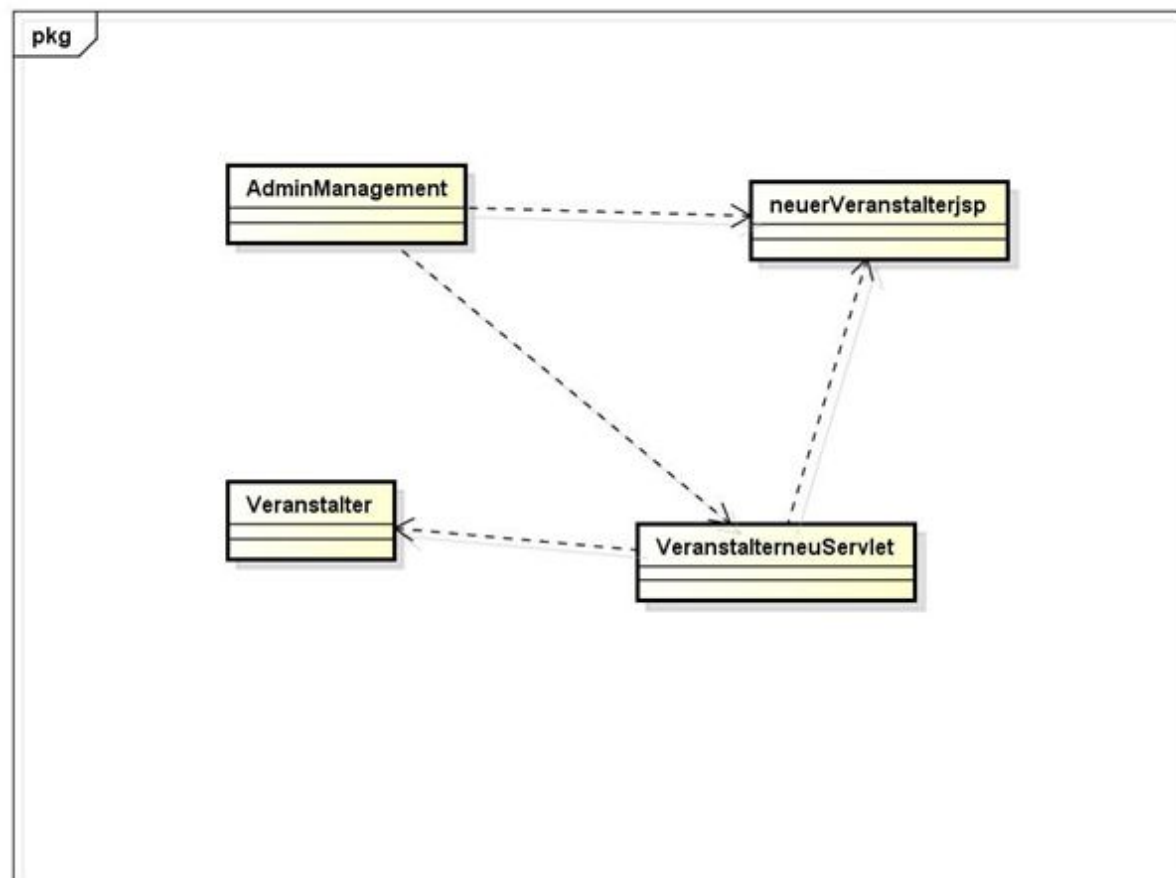


Powered By Visual Paradigm Community Edition

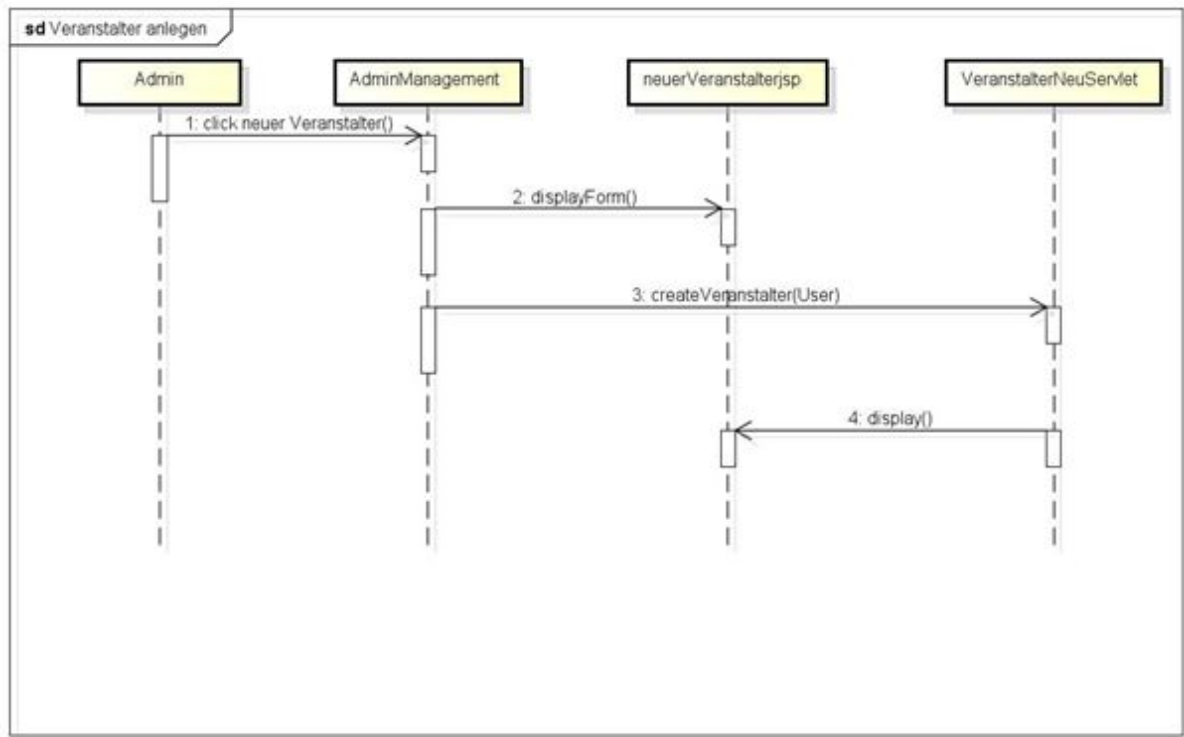


powered by Astah

Veranstalter anlegen

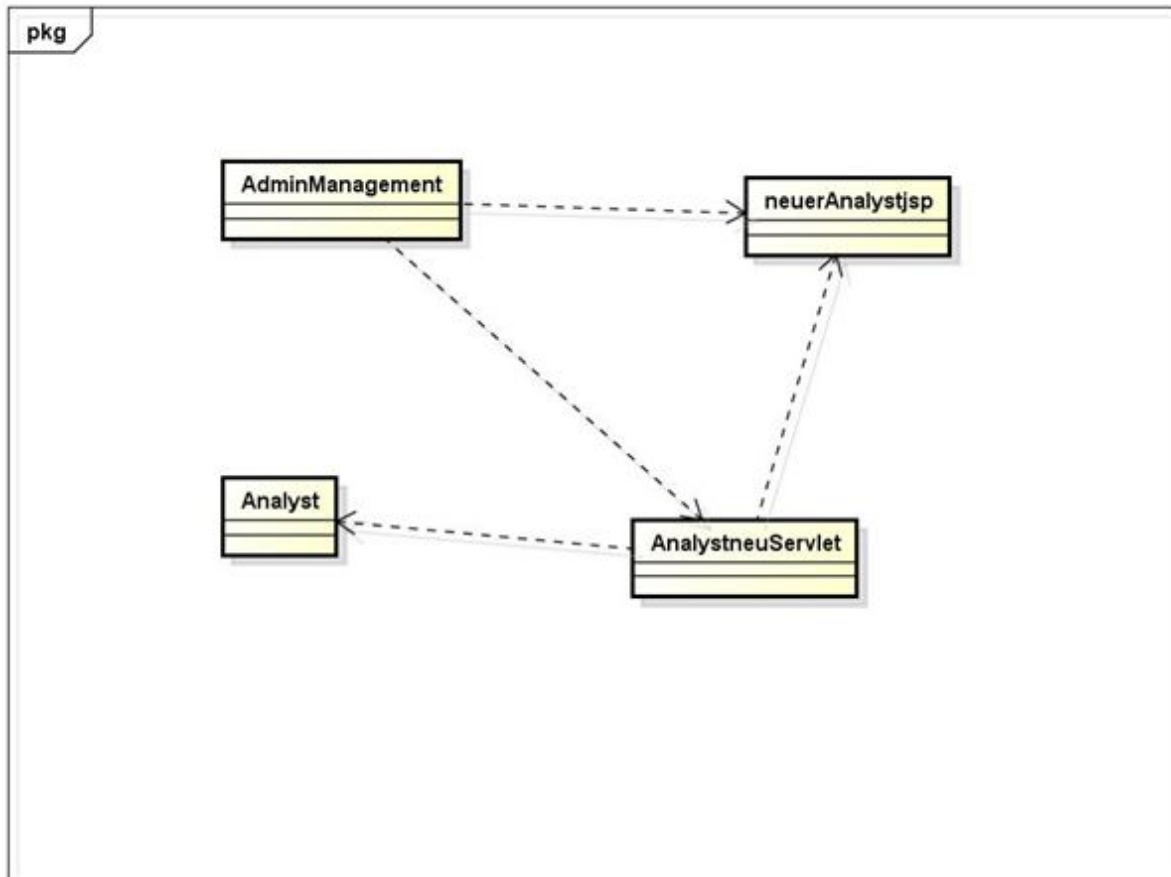


powered by Astah

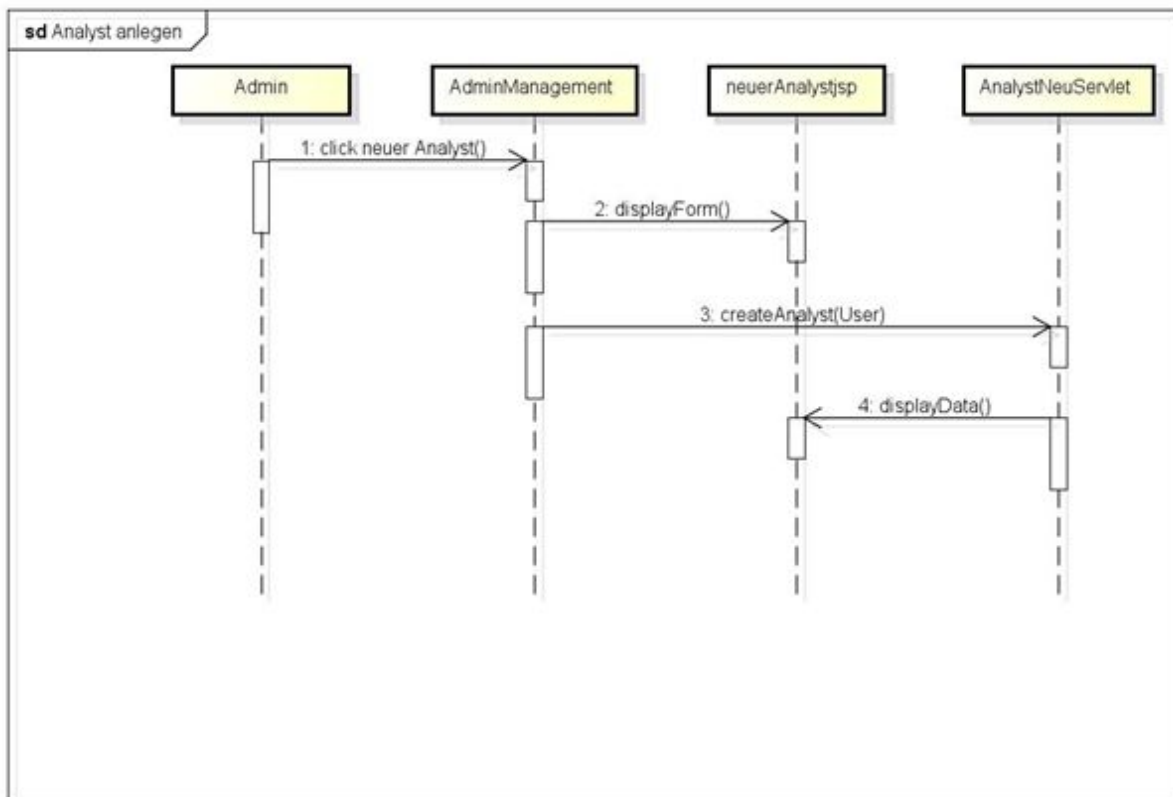


powered by Astah

Analyst anlegen

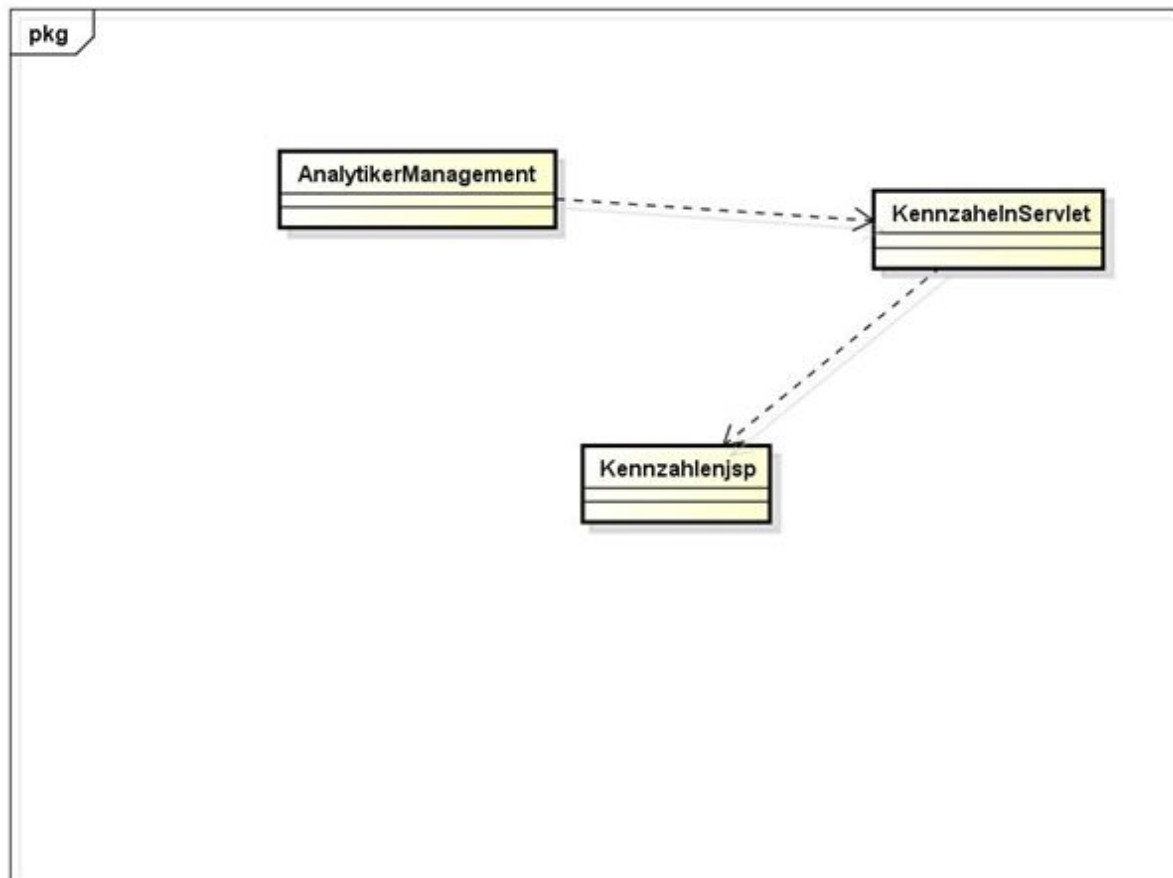


powered by Astah

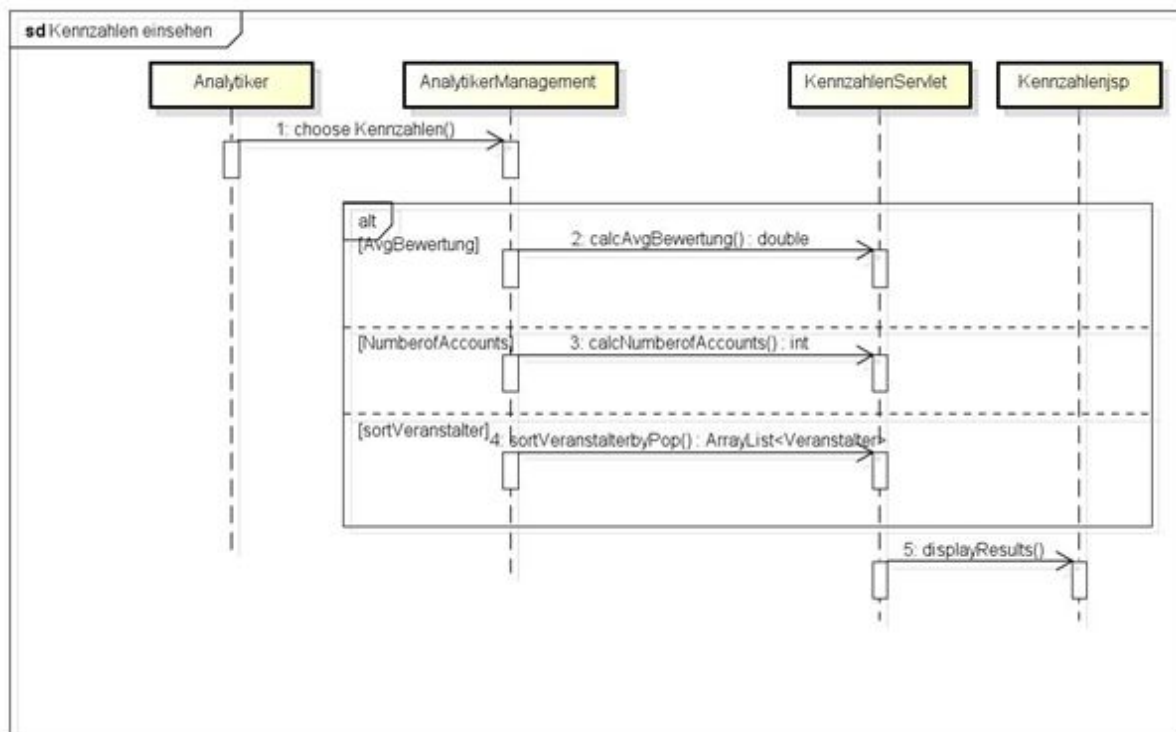


powered by Astah

Kennzahlen einsehen

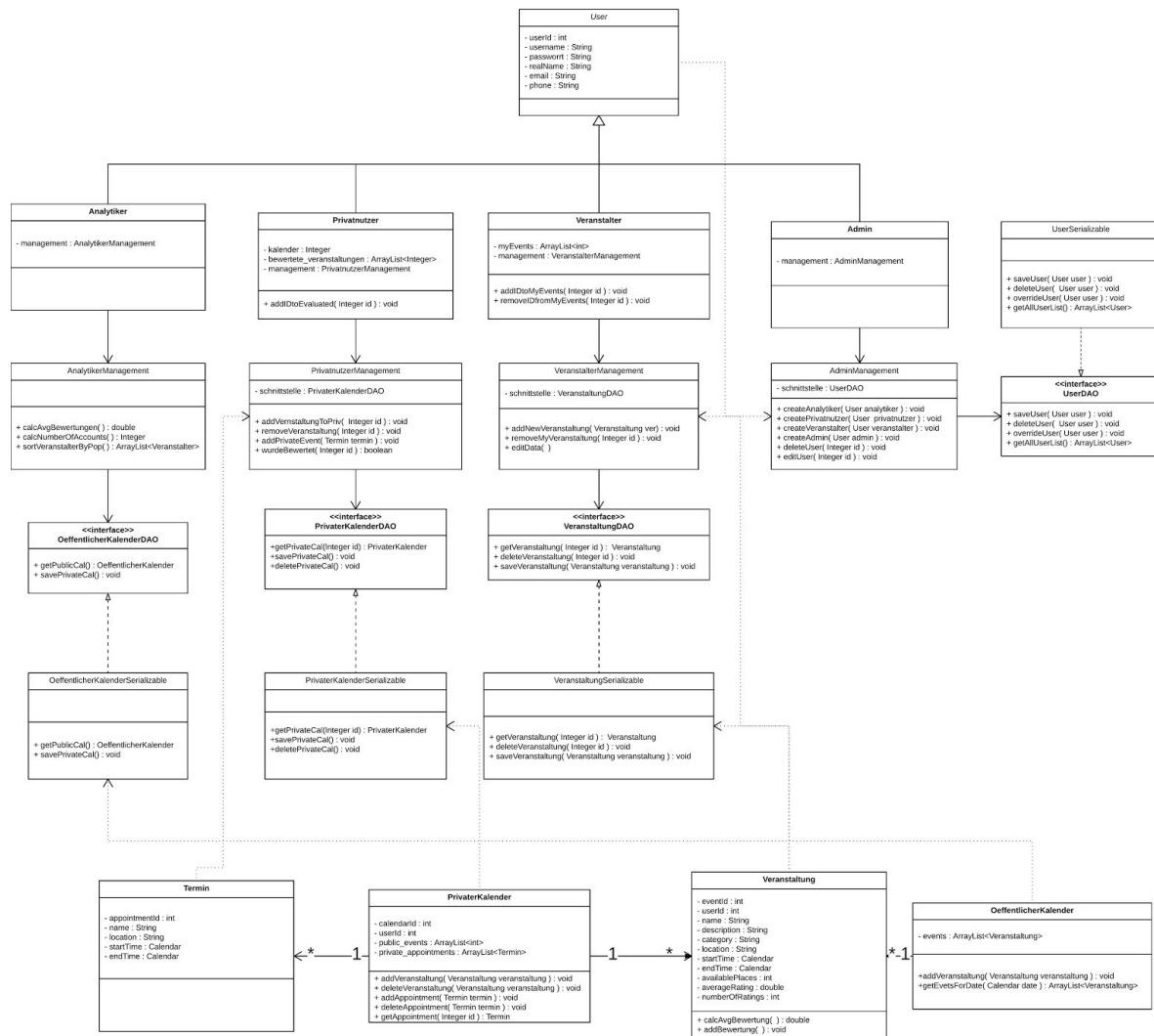


powered by Astah



powered by Astah

3 Übersichtsklassendiagramm



4 Architekturbeschreibung

