

## Gruppenbeiträge

2447899

- Erstellung der kompletten Authentifizierung und Terminfindung.
- Flags Erstellung
- Server handler Definition und vereinfachung (Error handling und Authentifizierungsboilerplatecodereduzierung)

8689159

- vollständige Implementierung von Dateisystem
- Entwicklung des Cachings
- Realisierung der Export-Funktionalitäten

3000685

- Kalenderansicht
- Terminliste
- Templates für Bearbeitung/Erstellung von Terminen
- Suchfunktion der Termine
- Handler für die Ansichten sowie der Suchfunktion

In dem Projekt habe mich um die Benutzeroberfläche der Termindarstellungen gekümmert. Dies hat folgende Bereiche umfasst:

**Templates:** Um die Termindarstellung in den verschiedenen Ansichten zu ermöglichen, habe ich mehrere Templates erstellt. Diese werden je nach Request von den entsprechenden Handlern als Response ausgeführt. Des Weiteren habe ich auch Templates erstellt, die auf mehreren Seiten eingebunden werden, die also mehrmals Verwendung finden. Bei diesen handelt es sich um das Template für die Erstellung der Termine, das Template für die Bearbeitung/Löschung der Termine sowie eines für die Darstellung von Terminen in einer Liste. Letzteres Template wird in der Suchansicht und in der Listenansicht verwendet, da diese ähnlich aufgebaut sind. Beide repräsentieren beispielsweise ihre Ergebnisse in einer untereinander folgenden Auflistung von Terminen und die Möglichkeit die Anzahl der Einträge pro Seite auszuwählen. Die mehrmals verwendeten Templates lassen sich in dem Ordner `assets/templates` finden.

**Handler:** Um die Anfragen betreffend der Termindarstellungen zu bearbeiten, habe ich mich um entsprechende Handler gekümmert. Hier gibt es für jede Ansicht einen spezifischen Handler und einen diesen übergeordneten Handler, der nur bei gültiger Berechtigung des Users die Anfrage an die entsprechenden Handler weiterleitet. Hier werden auch die Abfragen zur Termin-Erstellung und Termin-Bearbeitung abgefangen und entsprechende Funktionen aufgerufen, da diese unabhängig von den Ansichten sind. Die vier Handler und die dazu benötigten Tests lassen sich in `templateRouter.go` sowie in `templateRouterViewHandler_test.go` finden.

**Packages:** Des Weiteren habe ich mich um das Package Kalenderansicht und die dort benötigten Tests gekümmert. Dieses Package beinhaltet Klassen für Objekte, die für die Termindarstellung benötigt werden. Die Objekte werden hierfür den Templates bei ihrer Ausführung mitgegeben. Das Template für die Listenansicht erhält bei seiner Ausführung beispielsweise das Objekt `ListView`. Die Objekte besitzen Attribute und Funktionen, die in den Templates aufgerufen werden.

## Anwenderdokumentation

Wenn man einen Shared Termin erstellt wird man automatisch auf die Shared termin Id weitergeleitet. In Dieser kann man Nutzer hinzufügen, wobei diese dann über einen link verfügen, den man kopieren kann. Des Weiteren gibt es eine Möglichkeit sich alle eingeladenen Personen und deren Links anzuschauen. Man kann auch Vorschläge hinzufügen und diese dann hinzufügen. Dabei ist es egal ob der Termin der Vorgeschlagen ist wiederholend ist oder nicht und es geht dann in diesem Fall um den Start Termin. Anschließend kann man als Anwender des Kalendersystems über die verschiedenen Tage abstimmen. Dies geht unabhängig davon ob man eine Person eingeladen hat oder nicht, da wir es wichtig finden auch den Termin von vornherein festzulegen.

Die Person die den link bekommt kann dann auf der ihr zugeteilten Webseite sehr einfach und unkompliziert abstimmen wofür diese in jedem Termin ist.

Wenn Probleme auftauchen werden diese genannt: Bei Authentifizierungerror kann es sein, dass ihr Cookie abgelaufen ist und sie müssten sich einfach noch mal neu anmelden.

Bei internem Error ist dies bei der Webseite ein Problem und sie sollten sich an einen Administrator oder die Ersteller der Webseite melden.

Die Benutzung des Knopfes: „Herunterladen“, verlangt eine erneute Authentifizierung des Users. Diese ist über Basic Auth implementiert und losgelöst vom Sessionmanagement. Nachdem sich der User erneut angemeldet hat, wird aus den für ihn hinterlegten Terminen eine Ical generiert. Diese Ical wird anschließend heruntergeladen und kann in bestehende Kalender Programme integriert werden.

### Kalenderansicht und Listenansicht

**Navigation in der Kalenderansicht:** Nach der erfolgreichen Anmeldung, landet der Nutzer direkt auf der Kalenderansicht seiner Termine. Hier sieht er in dem entsprechenden Tabellenfeld die Anzahl seiner Termine an diesem Tag. Diese kann er sich durch klicken des Links „anzeigen“ anzeigen lassen und die Termininfos erscheinen im linken Teil des Bildschirms. Geteilte Terminvorschläge sind hier durch eine gelbe Umrandung und eine entsprechende Mitteilung gekennzeichnet. Unter den einzelnen Terminen kann durch klicken des „Bearbeiten“-Buttons der entsprechende Termin bearbeiten werden. Unter geteilten Terminen existiert kein „Bearbeiten“-Button. Hier gibt es einen „Vorschlag Löschen“-Button und ein „Details anzeigen“-Link. Um sich einen anderen Monat anzeigen zu lassen kann der Nutzer entweder einen spezifischen Monat im Drop-Down-Menü über der Kalenderansicht wählen oder er kann im Kopf des Kalenders mit den Pfeilen einen Monat vor- oder zurücknavigieren. Ebenso kann er mit Hilfe von Pfeilen Jahre vor- oder zurück springen. Mit dem „Heute“-Button kann jederzeit zum heutigen Datum gesprungen werden.

**Wechseln zwischen den Ansichten:** Möchte der Nutzer zur Listenansicht wechseln, kann er dies im Menü im oberen Teil der Webseite. Hier gibt es einen Dropdown-Button „Ansicht“, in dem zwischen Listen- und Kalenderansicht gewechselt werden kann.

**Navigation in der Listenansicht:** In der Listenansicht kann der Nutzer im oberen Bereich der Seite durch klicken auf ein Formularfeld ein spezifisches Datum auswählen. Anschließend werden ihm alle Termin und deren Infos in Boxen angezeigt, die ab diesem Datum auftreten. Die Termine sind so sortiert, dass der der als nächstes auftretende Termin ganz oben in der Liste erscheint. Geteilte Terminvorschläge sind durch eine gelbe Umrandung und eine entsprechende Mitteilung gekennzeichnet. Durch klicken des „Bearbeiten“-Links unter den jeweiligen Termineinträgen, kann der Termin bearbeitet werden. Im rechten Bildschirm kann in einem Drop-Down-menü gewählt werden,

ob 5, 10 oder 15 Einträge pro Seite angezeigt werden sollen. Ganz unten in der Seite kann mit Hilfe von „Zurück“- und „Vor“-Buttons zwischen den Seiten gesprungen werden.

**Beide Ansichten:** Kalender- sowie Listenansicht sind in drei Spalten unterteilt. In der linken befindet sich in beiden Seiten ein Formular, welches dem Erstellen eines Termins dient.

### Termin erstellen oder bearbeiten/löschen

**Termin erstellen:** Möchte man einen Termin erstellen, kann man dies in der Kalender- oder in der Listenansicht. Hier kann man entweder einen normalen Termin erstellen oder einen Terminvorschlag, den man anschließend teilen kann, wenn man in den Ansichten auf „Details anzeigen“ klickt. Es ist nötig bei der Termin-Erstellung alle Felder auszufüllen, ansonsten kommt eine entsprechende Warnung. Wird ein Termin mit der Wiederholung „niemals“ ausgewählt wird, unabhängig von der Eingabe des Nutzers, das Enddatum auf das Startdatum gesetzt. Dies geschieht auch, wenn das Enddatum zeitlich vor dem Startdatum liegt. Dennoch muss immer auch ein Enddatum angegeben werden.

**Termin bearbeiten/löschen:** Um einen Termin zu bearbeiten, kann man entweder den entsprechenden „Bearbeiten“-Link in der Kalenderansicht oder der Listenansicht drücken. Anschließend erscheint ein Termin-bearbeiten-Formular, indem der alte Titel und die alte Beschreibung als Default-Werte enthalten sind. Es werden auch die ehemaligen Start- und Enddaten angezeigt. In einem Drop-Down Menü kann gewählt werden, ob der Termin bearbeitet oder gelöscht werden soll. Nach drücken des „Termin bearbeiten“-Buttons landet man wieder auf der vorherigen Termin-Ansicht. Nur normale Termine besitzen einen „Bearbeiten“-Link. Terminvorschläge können nur gelöscht werden, da es sinnvoller ist einen neuen Terminvorschlag zu machen, statt einen existierenden zu bearbeiten.

### Suchfunktion

Um zur Suchfunktion zu gelangen, muss im Menü im oberen Teil der Webseite auf „Termin filtern“ geklickt werden. Anschließend wird man auf die entsprechende Seite weitergeleitet. Hier kann nach Titel und/oder Beschreibung gefiltert werden. Hierzu gibt man den gewünschten Titel und/oder die Beschreibung (oder nur Bestandteile dieser) in die dafür vorgesehenen Formularfelder ein und drückt den „Termin-Filtern“-Button. Es gibt verschiedene Varianten, die bei der Suche auftreten können:

- Suche nach Titel, Beschreibungsfeld ist leer:
  - ➔ Rückgabe: alle Termine, die den gesuchten Bestandteil im Titel haben
- Suche nach Beschreibungsfeld, Titel ist leer:
  - ➔ Rückgabe: alle Termine, die den gesuchten Bestandteil in der Beschreibung haben
- Suche nach Beschreibungsfeld & Titel:
  - ➔ Rückgabe: alle Termine, die dem gesuchten Titel **und** der gesuchten Beschreibung entsprechen
- Beide Felder sind leer:
  - ➔ Rückgabe: alle Termine

### Dokumentation des Betriebs

Nach dem Starten der Anwendung kann man per https sich auf die Webseite angehen → dabei ist der hostname entscheidend, dieser wird automatisch festgelegt und kann localhost sein, wenn es im lokalen Umfeld ist oder sonstige Adresse. Dabei kommt man anschließend auf die Startwebseite. Da kann man sich mit *admin* im Nutzernamen/ Passwortfeld einloggen und kommt dann auf den eigenen

Kalender. Über die Navigationsbar Konto kann man sein Nutzer Name einsehen und das Passwort ändern oder einen neuen Nutzer hinzufügen.

Beim Betrieb des Dateisystems ist darauf zu achten, dass die gewählte Plattform den schnellen zugriff auf vergleichsweise kleine Dateien unterstützt.

Wie der Zugriff auf die Kalenderansicht abläuft, kann anhand von Abbildung 1: Sequenzdiagramm Kalenderansicht nachvollzogen werden. Sendet der Nutzer der Webseite eine Request um entweder die Kalender- Listen- oder Suchansicht zu Besuchen, wird die Anfrage zunächst an den View-Handler weitergereicht. Dieser ruft zunächst eine Funktion auf, um den User zu überprüfen. Wurde dieser überprüft, wird die Request des Nutzers an den entsprechenden Handler weitergereicht. Bei diesem kann es sich um den Handler der Kalenderansicht (Table-View Handler, *Sequenz 1* im Sequenzdiagramm), der Listenansicht (List-View Handler, *Sequenz 2* im Sequenzdiagramm) oder der Suchansicht (Filter-View Handler, *Sequenz 3* im Sequenzdiagramm) handeln. Der entsprechende Handler führt die Request aus indem, je nach übergebenen Werten in der Request-URL, die entsprechende Funktion aufgerufen wird. Anschließend wird das entsprechende Template (liste.html, tbl.html, filterTermins.html) zurückgegeben.

Kann der User nicht authentifiziert werden (*Sequenz 4* im Sequenzdiagramm), wird die Anfrage nicht an weitere Handler weitergeleitet, sondern der View Handler teilt dem Benutzer mit, dass dieser nicht authentifiziert ist.

Handelt es sich bei der Anfrage um eine Termin-bearbeiten/-löschen/-erstellen Anfrage, wird diese auch vom View Handler bearbeitet und die Request an den Handler der Ansicht, auf der diese Aufgerufen worden ist , weitergeleitet. Dieser gibt dem User das entsprechende Template zurück (*Sequenz 5* im Sequenzdiagramm, der Ablauf ist hier gezeigt anhand einer Termin-Erstell-Anfrage über die Kalenderansicht) .

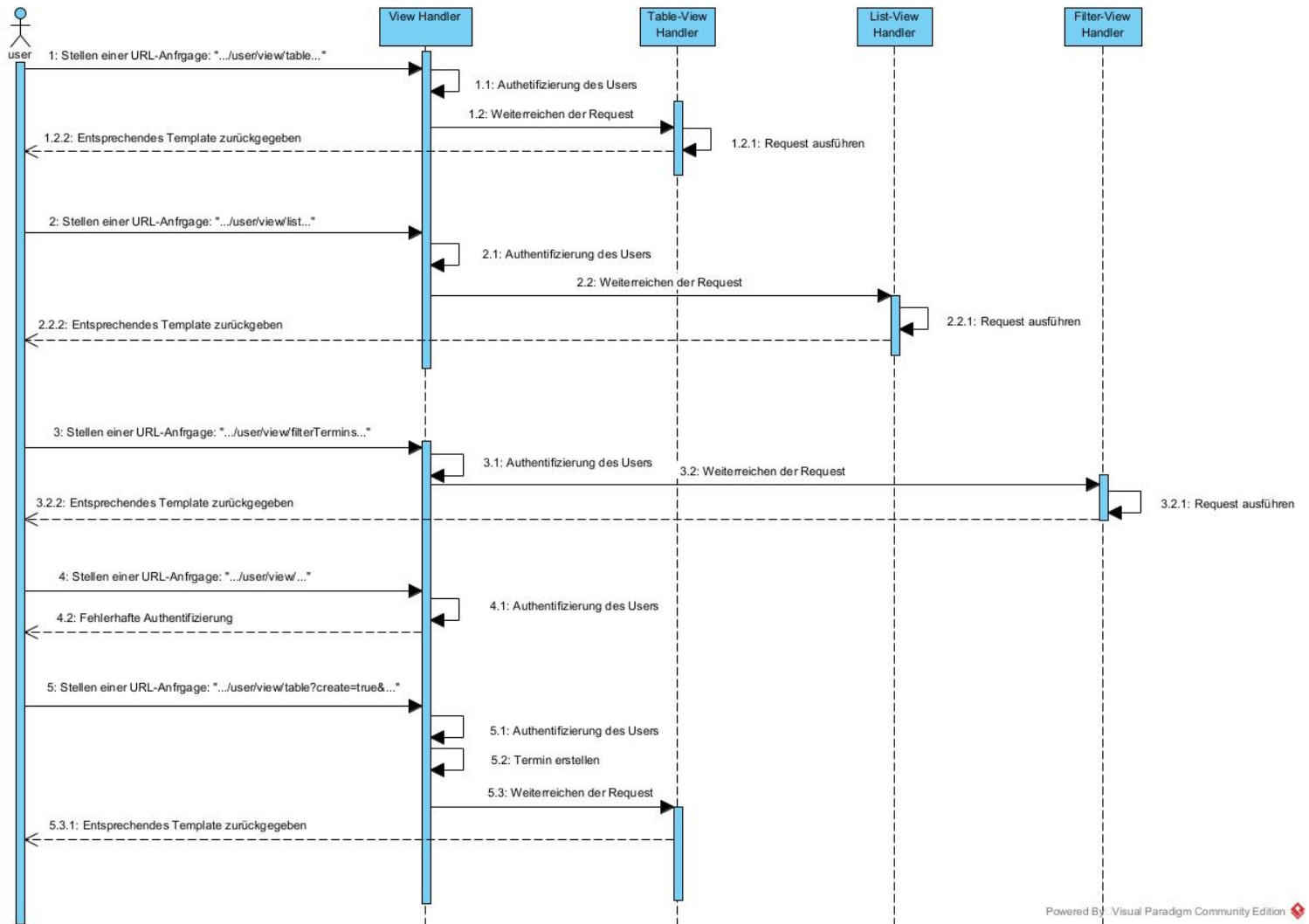


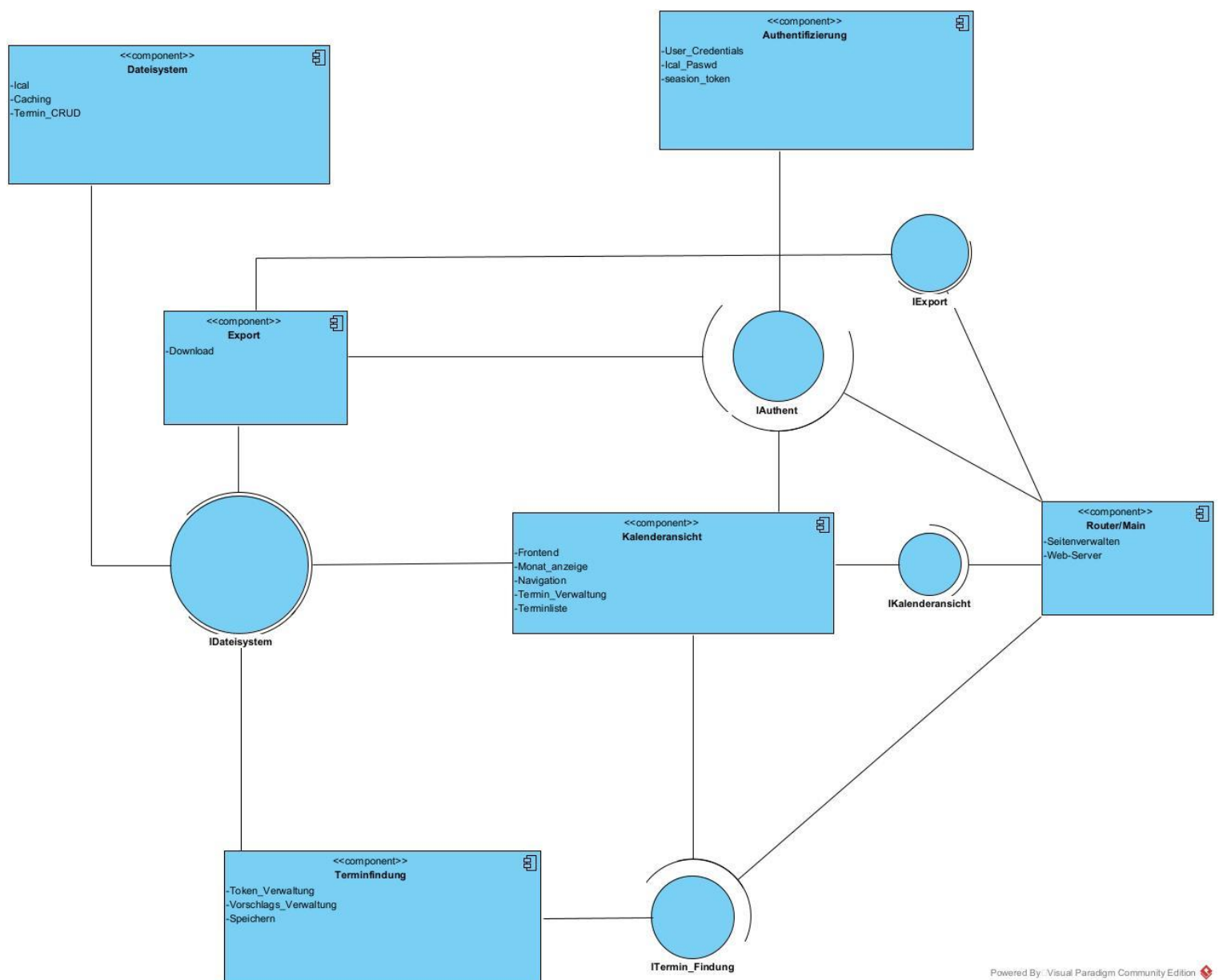
Abbildung 1: Sequenzdiagramm Kalenderansicht

## Architekturdokumentation

Abbildung 2: Komponentendiagramm zeigt die Komponente der Anwendung und deren Zusammenspiel. Dieses Komponentendiagramm entstand zu Beginn der Entwicklung der Anwendung und setzt sich aus fünf großen Komponenten zusammen. Diese werden in den folgenden Abschnitten weiter erläutert. Bei den fünf Komponenten handelt es sich um: das Dateisystem, die Authentifizierung, die Terminfindung, die Kalenderansicht, den Export sowie um den Router bzw. die Main-Funktion.

In den Kästen der einzelnen Komponenten kann abgelesen werden, welche Bereiche diese umfassen bzw. was genau die einzelne Komponenten beinhalten. Die Komponente Dateisystem muss zum Beispiel ein Caching der Termine des Nutzers ermöglichen, so dass diese nicht jedes Mal neu geladen werden müssen, wenn der User sich auf der Webseite bewegt.

Die Runden Kreise stellen die Schnittstellen der einzelnen Komponente da. Es müssen zum Beispiel Export, Kalenderansicht sowie die Terminfindung auf das Dateisystem, welches die Termin verwaltet, zugreifen können. Daher sind diese Komponente mit der Schnittstelle IDateisystem verbunden.



Powered By: Visual Paradigm Community Edition

Abbildung 2: Komponentendiagramm

## Terminfindung

Ein Teil der Komponenten sind die Terminfindungswebseiten und Komponenten. Diese wurden separat zu dem normalen Dateisystem entwickelt, da diese auch Vorschlags Termine enthalten, die natürlich nicht mit im normalen Kalender enthalten sein sollten. Deswegen werden diese separat gehalten.

Die Terminfindung bekommt vom Kalendersystem einmal einen Termin übergeben mit einer ID, der dem Termin entspricht, der erstellt wird um einen Anker im normalen System zu ermöglichen. Ab da arbeitet die Terminfindung unabhängig von allen anderen Komponenten. Dabei werden innerhalb dessen die Personen angelegt und Vorschlags Termine erstellt und die verschiedenen Votes verwaltet.

Bei Auswahl eines Tages wird auch einzigartig im Dateisystem Komponente ein Termin erstellt und der alte raus gelöscht. Der neue bekommt die neue einzigartige ID und somit ist der shared Termin final und kann nicht mehr bearbeitet werden.

## Authentifizierung

Die Authentifizierung ist eine Komponente die größtenteils nur Funktionen für das User Management anbietet. Dabei werden hier die Passwörter und Benutzernamen in einer map gespeichert um schnellen Zugriff auf diese zu gewähren. Des Weiteren werden Cookies in einer separaten globalen aber für das Paket ausschließlich bestimmten Variabel gespeichert. Diese enthalten neben einem random String auch den Nutzer auf den Sie verweisen und eine Zeit. Wenn diese Zeit von 15 Minuten überschritten wird ist das Cookie nicht mehr gültig. Wenn man innerhalb dessen auf dem Cookie zugreift wird das Cookie aktualisiert und die Laufzeit verlängert.

## Dateisystem

Die Komponente Dateisystem bündelt alle zur Terminverwaltung relevanten Funktionen und stellt das Caching bereit.

Ein einzelner Termin wird als JSON gespeichert. Für jeden User wird ein eigenes Verzeichnis angelegt, welches die einzelnen Termine eines Users bündelt. Der Kalender eines Users wird aus dem Einlesen der einzelnen Termine erzeugt und liegt anschließend als Slice vor. Änderungen an einzelnen Terminen erfolgen nur über den zugehörigen Kalender. Die Operationen löschen, update und erzeugen eines Termins wirken direkt auf die zugrunde liegende JSON. Ein erneutes Einlesen aller, oder einzelner Dateien ist nicht nötig. Durch diese Architektur können einzelne Änderungen schnell und effizient gespeichert und die Robustheit der Anwendung erhöht werden. Gleichzeitig wird nur eine einzige Lese Operation benötigt. Zusätzlich stellt das Dateisystem Such und Filterfunktionen auch Cache ebene zur Verfügung.

## Export

Die Komponente Export erlaubt es den Kalender eines Users als Ical zum Download bereitzustellen.

Entsprechend der Anforderungen wurde die Export Funktionalität mittels Basic Auth abgesichert. Der Code hierfür wurde aus den Vorlesungsunterlagen entnommen und in die bestehende Anwendung integriert. Ergänzend wurde ein Parser integriert, welcher aus einem existierenden Kalender eine Ical generiert. Abschließend wurde eine Funktion geschaffen um die neu generierte Datei als pausier baren Download zurückzugeben.

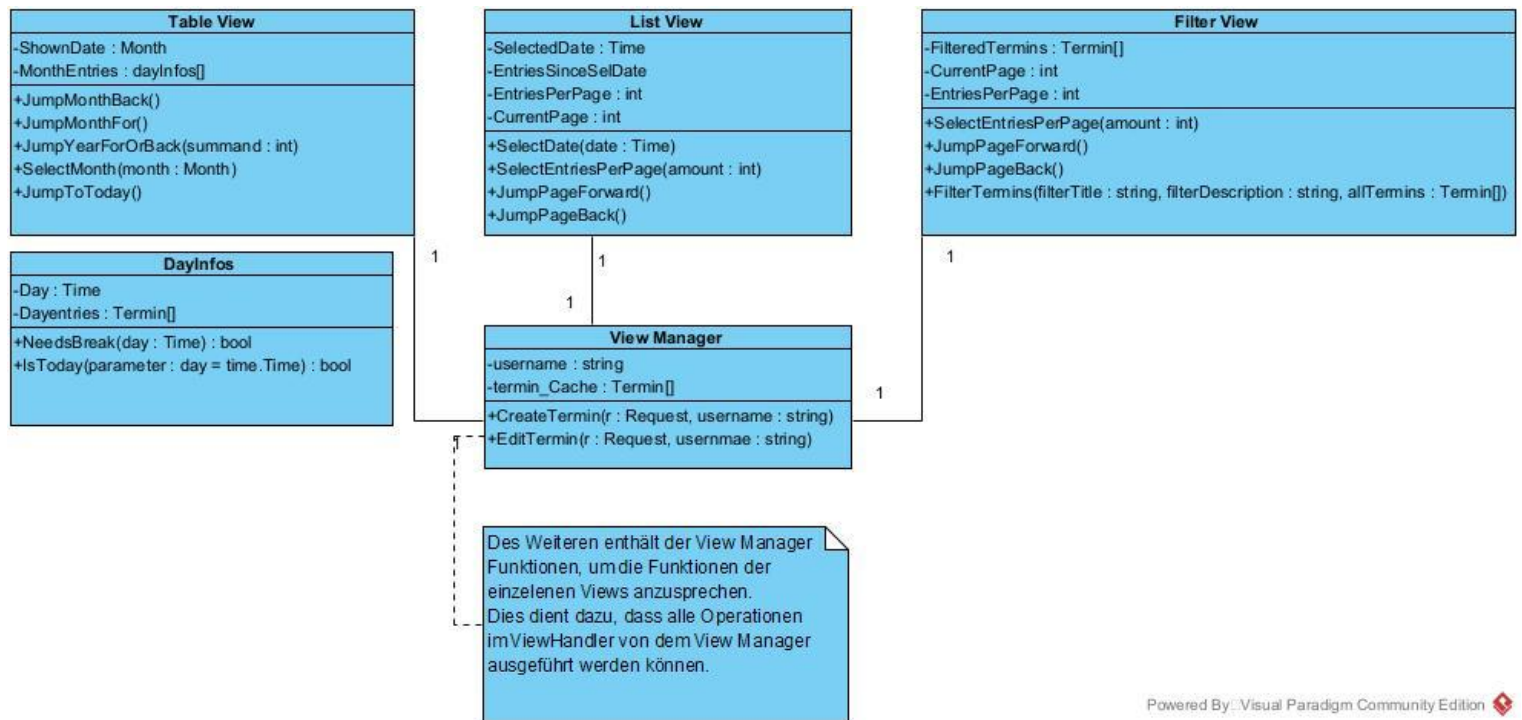
## Kalenderansicht und Suchansicht

Kalenderansicht und Suchansicht sind zu einer Komponente zusammengefasst, da diese ähnlich aufgebaut sind. Beide dienen dazu, dem Nutzer seine Termine zu präsentieren.

Die Komponente ist objektorientiert aufgebaut und setzt sich aus vier Objekten zusammen: Der Table-View, der List-View, der Filter-View sowie dem View-Manager, der diese steuert.

Wie diese Objekte in Relation zusammenstehen zeigt Abbildung 3: Kalenderansicht-Klassendiagramm.





Powered By: Visual Paradigm Community Edition

Abbildung 3: Kalenderansicht-Klassendiagramm

## Objekt View Manager

Der View Manager besitzt hier jeweils ein Objekt von der Table-View, der List-View sowie der Filter-View. Des Weiteren besitzt der View Manager Funktionen, um alle Objekt-Funktionen, die in Abbildung 3: Kalenderansicht-Klassendiagramm zu sehen sind, aufzurufen. Damit können in den Handlern (siehe Abbildung 1: Sequenzdiagramm Kalenderansicht) alle Funktionen der verschiedenen Ansichten, wie beispielsweise „Springe eine Seite vor in der Listenansicht“, mit Hilfe des View Manager-Objektes umgesetzt werden. In den Handlern wird als nur auf das Objekt View Manager zugegriffen.

Des Weiteren besitzt der View-Manager noch den Usernamen des momentanen Nutzers sowie einen Cache mit allen Terminen des Nutzers. Den Cache reicht er den verschiedenen Ansichts-Objekten weiter, damit diese die zu repräsentierenden Einträge entsprechend filtern können.

Termin-Erstellen sowie Termin-Bearbeiten/Löschen wird auch vom View-Manager übernommen, da dies unabhängig von den Ansichten ist. Die Funktionen um Termine zu bearbeiten oder zu löschen greifen hierzu auf die Komponenten Dateisystem und ggf. Terminfindung zu und rufen dort entsprechende Funktionen auf. Hierzu wird teilweise der Username benötigt.

## Objekt Table-View, List-View und Filter-View

Die Objekte Table-View, der List-View sowie der Filter-View besitzen die in Abbildung 3: Kalenderansicht-Klassendiagramm gezeigten Funktionen und Attribute, um innerhalb der Ansichten dem User eine Navigation zu ermöglichen, wie beispielsweise auszuwählen, wie viele Einträge angezeigt werden sollen.

Die Objekte besitzen noch weitere Funktionen, die aber nicht relevant sind um die Beziehung zueinander zu verdeutlichen. Diese dienen beispielsweise dazu, die anzuzeigenden Einträge zu filtern, zeitlich zu sortieren und weiteres.

Table-View besitzt des Weiteren noch ein Attribut MonthEntries. Bei diesem handelt es sich um ein Slice welches die Länge der Tagesanzahl des angezeigten Monats besitzt. Das Slice enthält Objekte von DayInfos. Diese Objekte enthalten die Informationen der Termine der einzelnen Tagen und sind entsprechend der Monatstage ins Slice eingeordnet. Das Objekt DayInfos, das die Termine des dritten Monatstages enthält, befindet sich also an der dritten Position des Slices. Das Objekt DayInfos besitzt auch noch zwei Funktionen, die im Template benötigt werden. NeedsBreak gibt Auskunft darüber, ob es sich um einen Sonntag handelt und dementsprechend ein Zeilenumbruch in der Tabellenansicht benötigt wird. IsToday gibt Auskunft darüber, ob es sich bei dem Tag um den heutigen handelt, um diesen in der Ansicht entsprechend zu markieren.