

# Terminkalender M223

JAVA WEB APPLICATION | JSF

ALEJANDRO WENZLER | SHAY MOSER

## Inhalt

Arbeit .....	3
Planung .....	3
Arbeitsprozess.....	3
Hilfsmittel.....	3
Klassendiagramm .....	4
Diagramm .....	4
Erklärung.....	5
UML-Zustandsdiagramm .....	6
Diagramm .....	6
Erklärung.....	6
Zusammenfassung .....	7

# Arbeit

## Planung

Wir haben mit der Erstellung eines Klassendiagramms begonnen, hatten jedoch Verständnisprobleme. Daher haben wir vorerst nur ein Klassendiagramm erstellt, das die Klassen enthielt, aber ohne alle Attribute und Funktionen. Dadurch konnten wir zumindest die Hierarchie der Klassen darstellen und uns einen Überblick verschaffen, wie wir bei der Programmierung vorgehen möchten.

Bei der Planung mussten wir auch berücksichtigen, dass wir als Gruppe an einem anderen Projekt beteiligt waren. Daher mussten wir sorgfältig abwägen, wann wir welche Aufgaben erledigen konnten.

Für die Zusammenarbeit und den Austausch der Projektdateien haben wir uns für die Verwendung von GitHub entschieden. GitHub ist eine effiziente Plattform, die uns ermöglicht, die Projektdateien effektiv zu teilen und zu synchronisieren.

## Arbeitsprozess

Bei dem ganzen Arbeitsprozess hatten wir oft Mahls grosse Änderungen gemacht, wie was für Klassen brauchen wir wirklich oder wegen dem Key Generator. Zu Beginn hatten wir einen Key Generator, der eine 1024 bit Key generieren konnte, doch man konnte nicht die bit Anzahl Unterstufen da es von einer Java Library kommt und diese es nicht erlaubt unter 1024 bit zu gehen da dies unsicher ist.

## Hilfsmittel

Wir haben für die Grafische Darstellung eine Library verwendet die Textfelder etc. besser darstellt.

Diese Library heisst PrimeFaces

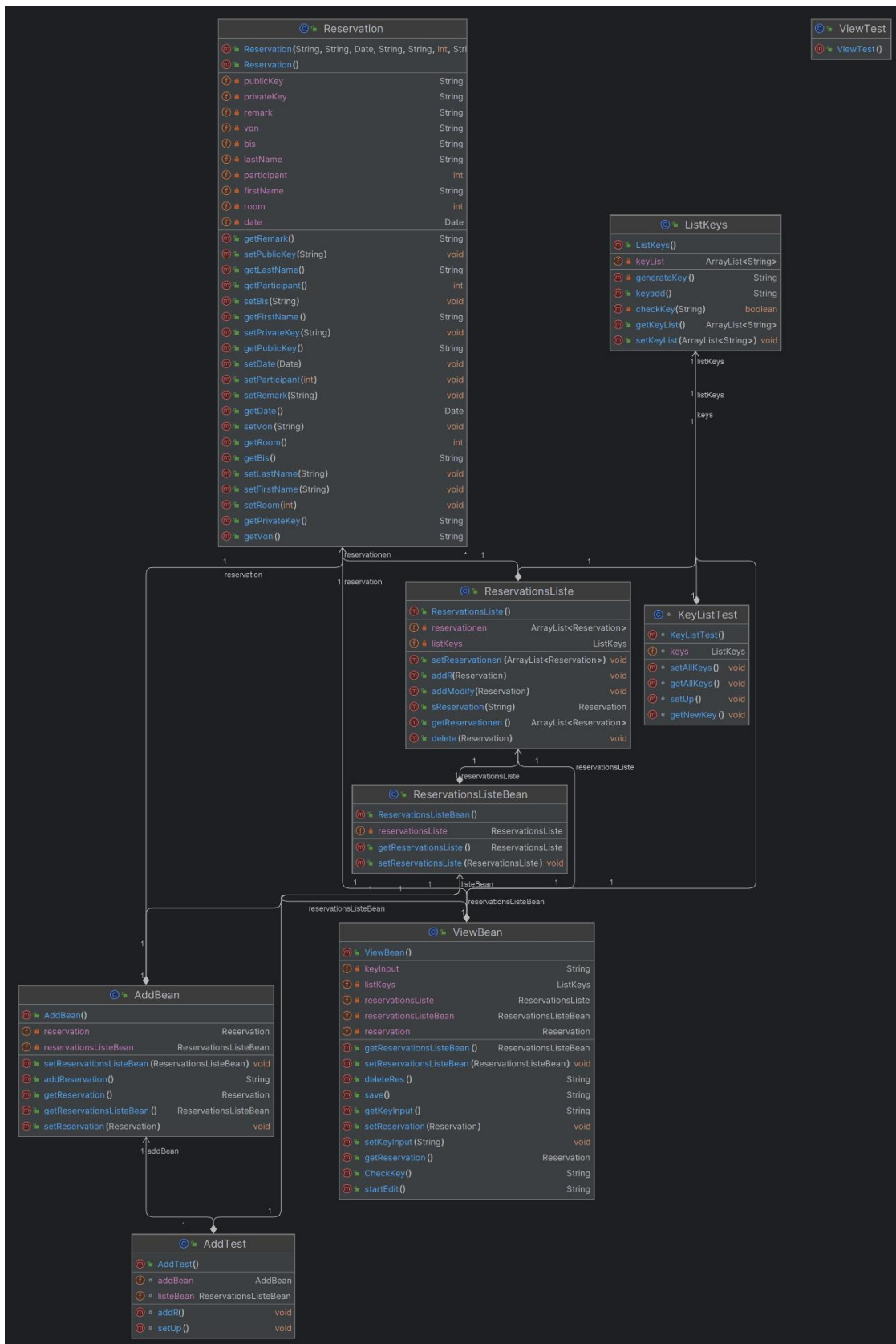
Wir haben für das Klassendiagramm ein Plugin von JettBrains verwendet, dieses Plugin stellt alle Klassen dar und Ihre Verbindungen so wie Ihre Attribute etc.

Zudem haben wir noch GitHub verwendet, um uns die Projekt Dateien besser auszutauschen.

GitHub Repository: [https://github.com/A1eeee/Terminkalender\\_M223\\_Alejandro\\_Shay.git](https://github.com/A1eeee/Terminkalender_M223_Alejandro_Shay.git)

# Klassendiagramm

## Diagramm



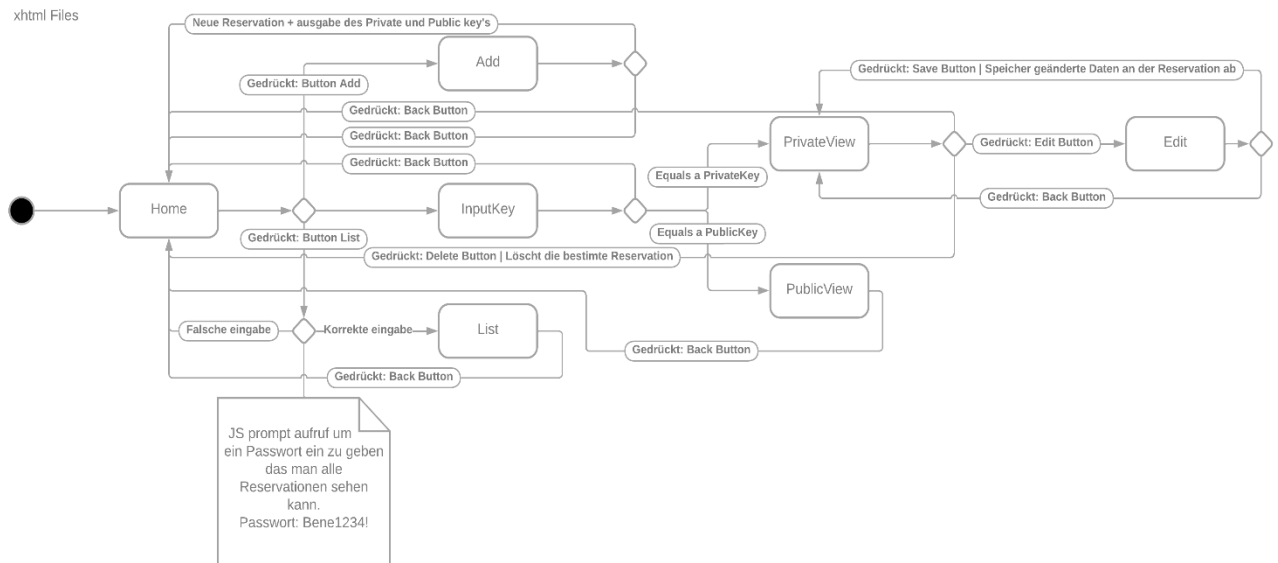
## Erklärung

Wir haben uns für diese Möglichkeit entschieden, da am Ende des Projekts ein endgültiges Klassendiagramm erstellt werden konnte, dass die tatsächliche Struktur widerspiegelt. Zudem werden die richtigen Verbindungen verwendet, die wir noch nicht vollständig beherrschen. Das bedeutet, dass erfahrene Entwickler wahrscheinlich die Struktur besser verstehen würden, wenn sie die korrekten Verbindungen verwenden würden, die uns noch nicht so vertraut sind.

Zu Beginn des Projekts haben wir ein vorläufiges Klassendiagramm erstellt, das die Hierarchie der Klassen im Projekt darstellte. Dadurch konnten wir vorankommen. Allerdings haben wir das Diagramm nicht vollständig mit Attributen usw. ergänzt, da wir anfangs Schwierigkeiten hatten, das gesamte Projekt zu verstehen. Da wir uns in den Ferien befanden, war es schwierig, unsere Lehrperson mit Fragen zu erreichen. Außerdem haben wir oft vergessen, Fragen zu stellen, da wir uns in einer Ferienstimmung befanden und hauptsächlich abends gearbeitet haben.

# UML-Zustandsdiagramm

## Diagramm



## Erklärung

Wir haben auf der Website Lucidcharts ein UML-Zustandsdiagramm erstellt und durch die Informierung im Internet konnten wir durch das Verständnis, das wir Online gesammelt haben, ein Zustandsdiagramm erstellen was unsere Verbindungen von unseren xhtml Files darstellt und Ihre Verzweigungen.

Dieses Diagramm enthält viele Verzweigungen da man in unserem Projekt auch oftmals die Variante besitzt zurück zur Startseite zu gehen.

## Zusammenfassung

Das gesamte Projekt war eine spannende Erfahrung um die Erstellung von Webapplikationen in Java, basiert auf JSF. Durch die gesammelte Erfahrung durch das Projekt stieg unsere Erfahrung zwischen der Kommunikation von Java Klassen und Bean Klassen wie die genau funktionieren und wie die Bean Klassen Daten speichern von xhtml File und wie sie, sie weitergeben könne.

Durch dieses Projekt konnten wir noch genau lernen, wie man ein Java Project aus IntelliJ mit GitHub verknüpft und somit bringt uns das auch sehr viel Vorteile.