

Aufgabenblatt 6 – Entwurfsbeschreibung – Iteration 1

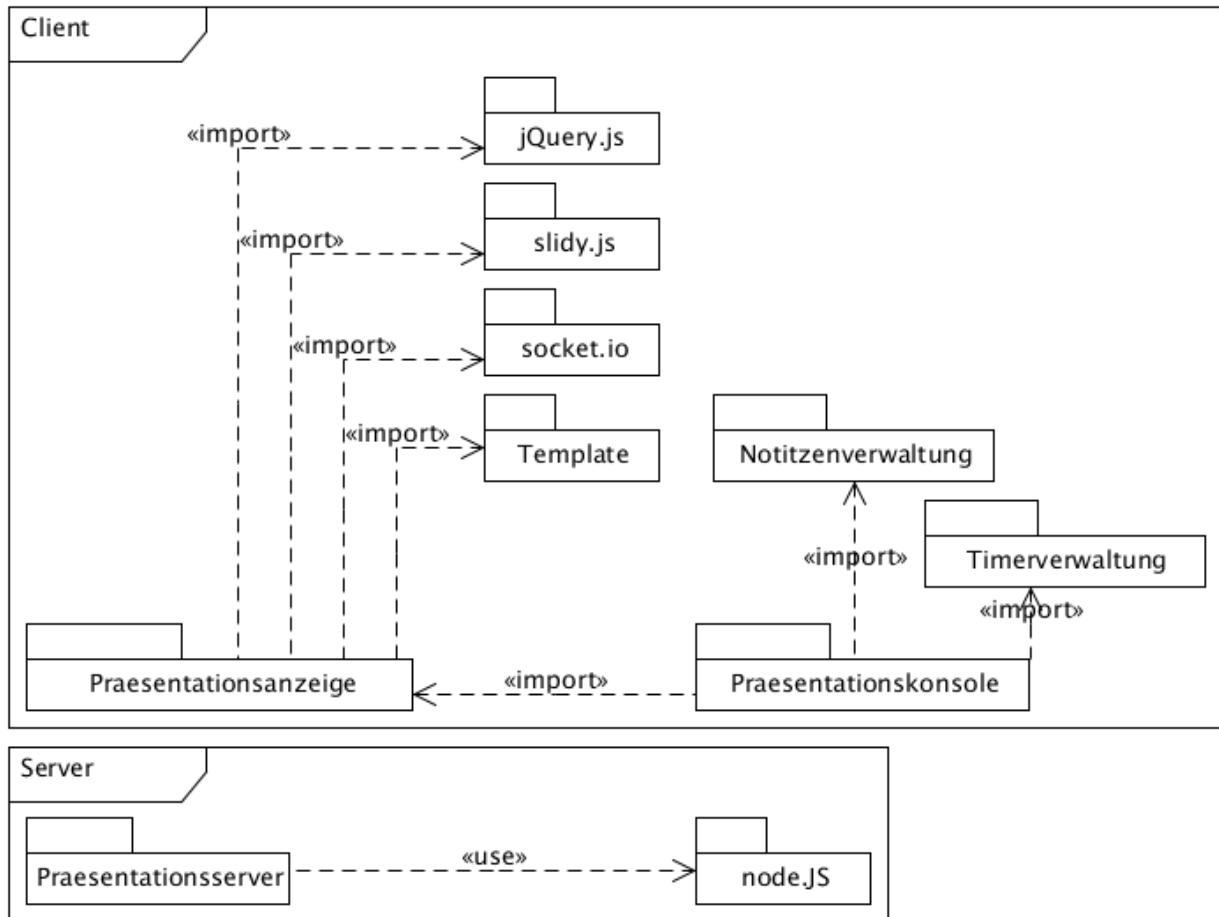
Projekt: Keynode 11
Gruppe: swp11-5
Projektleiter: Florian Golemo
Leiter Modellierung: Andy Wermke
Leiter Implementierung: Enrico Reich

Inhaltsverzeichnis

1 Statisches Modell – Paket- und Klassendiagramm.....	3
2 Dynamisches Modell – Aktivitäts- und Sequenzdiagramme.....	3
2.1 Verbindungsaufbau.....	3
2.2 Folien weiterschalten.....	3
2.3 Notiz- und Timerverwaltung.....	3

1 Statisches Modell – Paket- und Klassendiagramm

Das Projekt ist in Pakete aufgegliedert, wie folgt:



Wie man sieht, unterscheiden wir zwischen der Server- und der Client-Seite, wobei die Server-Seite wirklich nur einen Host für die Tastatureingaben der Clients bereitstellen muss und dementsprechend schmal gehalten werden kann. Der Code kommt in einer node.js-Umgebung zum Einsatz.

Der Client ist komplexer und setzt sich aus Präsentationsanzeige und Präsentationskonsole zusammen. Hierbei wird die letztere nur geladen, wenn der Betrachter über eine entsprechende Autorisierung verfügt. (Dazu im dynamischen Modell, Punkt 2.1, mehr)

Die Präsentationsanzeige benötigt, wie gehabt das Slidy-Framework zur Darstellung des Template, die Socket-IO-Bibliothek vom Node-Server, um Sockets bereit zu stellen und in Zukunft jQuery für häufig verwendete Funktionen und evtl. für Transitionen.

Die Präsentationskonsole greift auf mehrere Instanzen der Präsentationsanzeige zurück (für aktuelle Folie und Vorschau), und verfügt außerdem über eine Timersteuerung und eine Klasse zur Ausgabe und später Verwaltung von Notizen.

Zur weiteren Detaillierung folgt nun das vorläufige Klassendiagramm, wobei während der ersten Iteration untersucht wird, inwiefern diese Klassenstruktur in JavaScript umgesetzt werden kann. Sollten einige wenige Dinge nicht objektorientiert implementierbar sein (wie Event-Handler und Server-Anweisungen), werden diese in der ersten Iteration strukturiert umgesetzt.

-Klassendiagramm-folgt

2 Dynamisches Modell – Aktivitäts- und Sequenzdiagramme

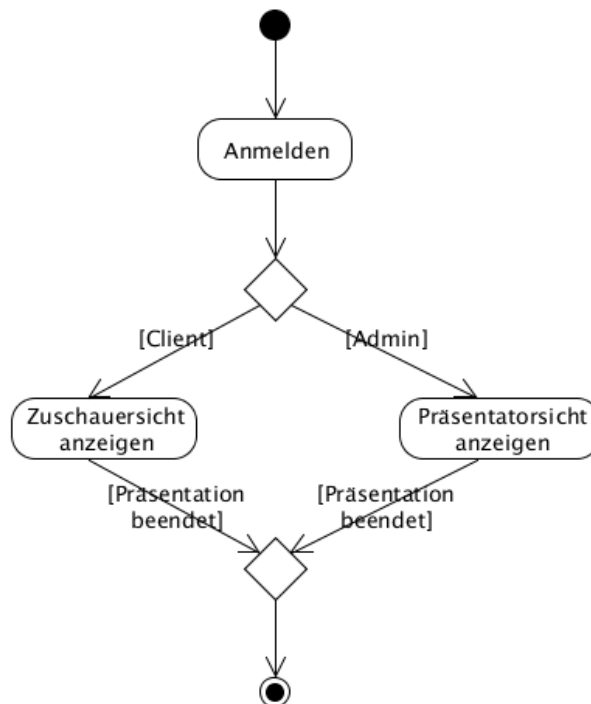
Im Folgenden werden die wichtigsten Prozesse identifiziert und je nach Bedarf entweder durch ein Aktivitäts-, Sequenzdiagramm oder durch beide beschrieben.

2.1 Gesamtsystem

-Aktivitätsdiagramm folgt-

2.2 Client-Anmeldung

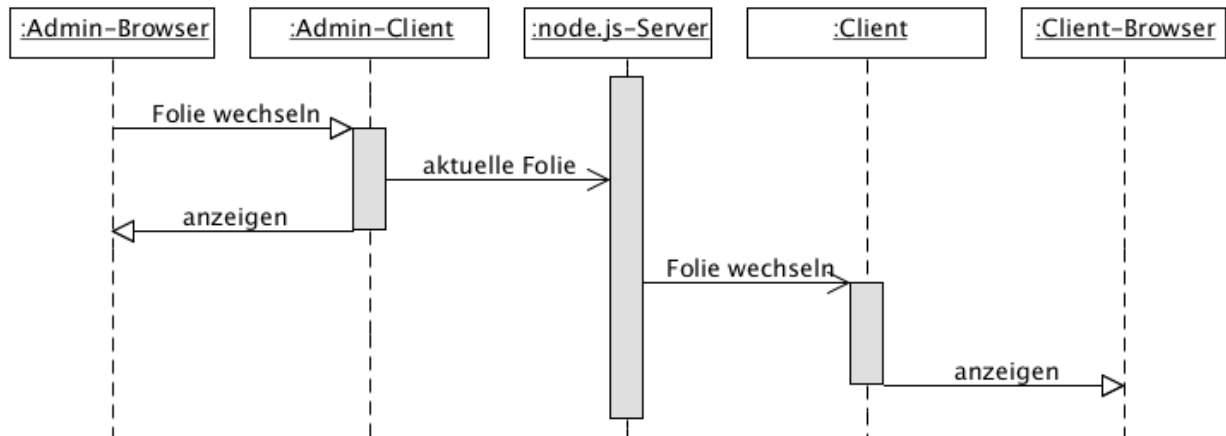
Wenn sich ein Client „anmeldet“, also die Adresse der Präsentation im Browser öffnet, passiert folgendes:



Der Browser lädt zuerst die Std.-Client-Klasse und die Präsentation. Wenn der Server feststellt, dass es sich um den Admin handelt, wird außerdem die Admin-Client-Klasse nachgeladen, die dann die Präsentationskonsole aufbaut und ein zweites Fenster mit der Präsentationsanzeige startet. Beim Zuschauer wird nur die Präsentationsanzeige dargestellt. Bei beiden endet die Präsentation indem der Admin die Präsentation beendet.

2.3 Folien weiter schalten

Ein wichtiger Teil wurde von uns bereits im Rahmen der Vorbereitung abgearbeitet: die Studie zur Umsetzbarkeit von gegenseitigem Weiterschalten der Folien, „SlidyTest“. Da allerdings hier nur der Admin die Folien kontrollieren darf, müssen kleine Änderungen umgesetzt werden:

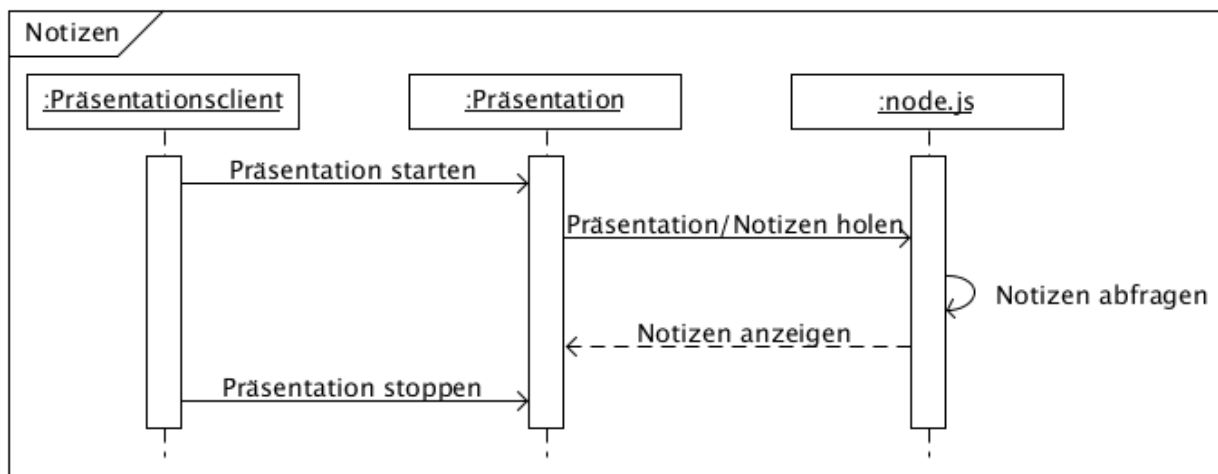


Der Browser des Admins erfasst das Tastaturevent und lässt dies vom Admin-Client verarbeiten. Dieser sendet die neue Foliennummer an den Node-Server und schaltet bei sich die Folie weiter. Der Node-Server broadcastet die neue Foliennummer an alle Clients, die gerade verbunden sind und speichert sie lokal für neu-verbundene Clients. Der Client-Code erwirkt dann die Weiterschaltung im Browser des Clients.

2.4 Notiz- und Timerverwaltung

In der Präsentationskonsole sollen weitere Funktionen zur Verfügung stehen als nur die aktuelle Folie und die Vorschau: die Darstellung von Notizen und die Steuerung des Timers, der die restliche verbleibende Zeit für die Präsentation anzeigt.

(Das Notiz-System wurde in Punkt 2.1 schon kurz beschrieben)



Nachdem beim Client die Präsentation gestartet wurde, fragt der Client die Notizen vom Server ab und zeigt diese jeweils zu einer Folie an.

Der Timer funktioniert ähnlich einfach: er zeigt immer die aktuell verbleibende Zeit an und man kann ihn pausieren und wiederaufnehmen. Der gesamte Timer-Code liegt beim Admin-Client.

