

Pflichtenheft für Projekt Dashboard

Michél Fischer, Eike Schaubeck, Lydia Kaiser, Michael Bendel

28. Oktober 2019

Inhaltsverzeichnis

1	Einleitung	2
2	Kriterien	2
2.1	Muss-kriterien	2
2.2	Kann-kriterien	3
2.3	Abgrenzungskriterien	3
3	Technologien	4
4	Diagramme	5
5	Zeitplan	6
6	Einsatz	6

1 Einleitung

Folgende Projektarbeit haben Eike Schaubeck, Michael Bendel, Michel Fischer und Lydia Kaiser zusammen erarbeitet, wobei André Bauer als Betreuer fungierte.

Die Idee von diesem Projekt ist, ein Dashboard in einer Web-Anwendung zu programmieren, um die Planung mittels Kalender einfacher, strukturierter und übersichtlicher zu gestalten. Es soll ein Überblick über das Wetter, die Bus/Bahnverbindungen und weitere wichtige Termine geschaffen werden, sodass man nicht mehr zwischen mehreren Apps wechseln muss. Die Oberfläche des Programms soll ansprechend und intuitiv gestaltet werden. Um unnötigen Datenverbrauch zu vermeiden, soll das Dashboard asynchron geupdatet werden.

2 Kriterien

Folgende Kriterien wurden zur Eingrenzung der Funktionalitäten definiert:

Hierbei wird jeweils auf die Bereiche Datenbank, GUI und Controller eingegangen.

2.1 Muss-kriterien

Datenbank

Der Nutzer kann Termine aus seinem Kalender verwalten. Dazu gehört das Laden, das Hinzufügen, das Löschen und das Ändern von Terminen.

GUI

Die graphische Oberfläche soll das (aktuelle) Wetter, Bus/Bahnverbindungen und Termine in Textform anzeigen.

Controller

- **Bus/Bahnverbindung**

Passend zu den Terminen sollen Bus/Bahnverbindungen angezeigt werden. Die Verbindungen sollen mit Start-/Zielbahnhof und der jeweiligen Zeit angegeben werden. Wenn Umstiege vorhanden sind, dann sollen auch die Zwischenhaltestellen (ebenfalls mit Zeit) angezeigt werden.

- **Wetter**

Es soll das aktuelle Wetter und eine Prognose von kleiner gleich vier Tagen zu sehen sein. Dabei soll der Standort, die Regenwahrscheinlichkeit, die aktuelle Temperatur, die Niederschlagsart sowie die Tiefst- und Höchsttemperatur dargestellt werden. Bei der Auswahl einer Bahnverbindung soll die Wetterlage am Start-/Zielort angezeigt werden.

- **Kalendereinträge**

Kalendereinträge werden unterteilt in Termine und Notizen. Dabei kann man bei Terminen den Namen, Start- und Endzeit und den Ort angeben.

Die Notizen bestehen aus einem begrenzt langen String.

- **Notizen**

Die Notizen werden als Text angezeigt.

- **Allgemein**

Die Wetterdaten und die Verbindungsdaten sollen standortbezogen sein. Die Aktualisierung der einzelnen Elemente soll dabei asynchron durchgeführt werden.

Die Bedienung des Dashboards soll auch über die Kommandozeile möglich sein.

Beim Auftreten von Fehlern bei dem Verbindungsaufbau, nicht verwertbaren Angaben zu Zeit und Orten sollen passende Fehlermeldungen angezeigt werden.

2.2 Kann-kriterien

GUI

Die räumliche Anordnung der Elemente soll für den Nutzer frei wählbar sein, wobei auch Elemente ein- und ausgeblendet werden können.

Controller

- **Weitere Funktionen und Termine**

Der Nutzer hat die Möglichkeit eine direkte Notiz zu speichern. Außerdem können die Geburtstage im Kalender verwaltet werden und mit Namen und Alter angezeigt werden.

- **Allgemein**

Die Synchronisationsrate kann vom Benutzer frei gewählt werden.

2.3 Abgrenzungskriterien

Datenbank

Es können keine Datenanhänge im Kalender angelegt werden.

Controller

- **Wetter**

Weitere Daten zum Wetter könne nicht abgerufen werden und die Temperatur wird nur in Grad Celsius ausgegeben. Eine Prognose für mehr als vier Tage ist nicht möglich.

- **Bus/Bahnverbindung**

Eine manuelle Abfrage von Bus/Bahnverbindungen ist nicht möglich. Nur auf den Termin bezogene Verbindungen stehen zur Verfügung.

- **Termine**

Termine (als Datentyp String) können nicht beliebig lang sein.

3 Technologien

APIs für Dienste

- Für Wetter(prognose): DarkSky
<https://darksky.net/dev>
- Für Bahnverbindungen: Google Directions API
<https://developers.google.com/api-client-library/java>

Konzepte

- AJAX; *Asynchronous JavaScript and XML*.

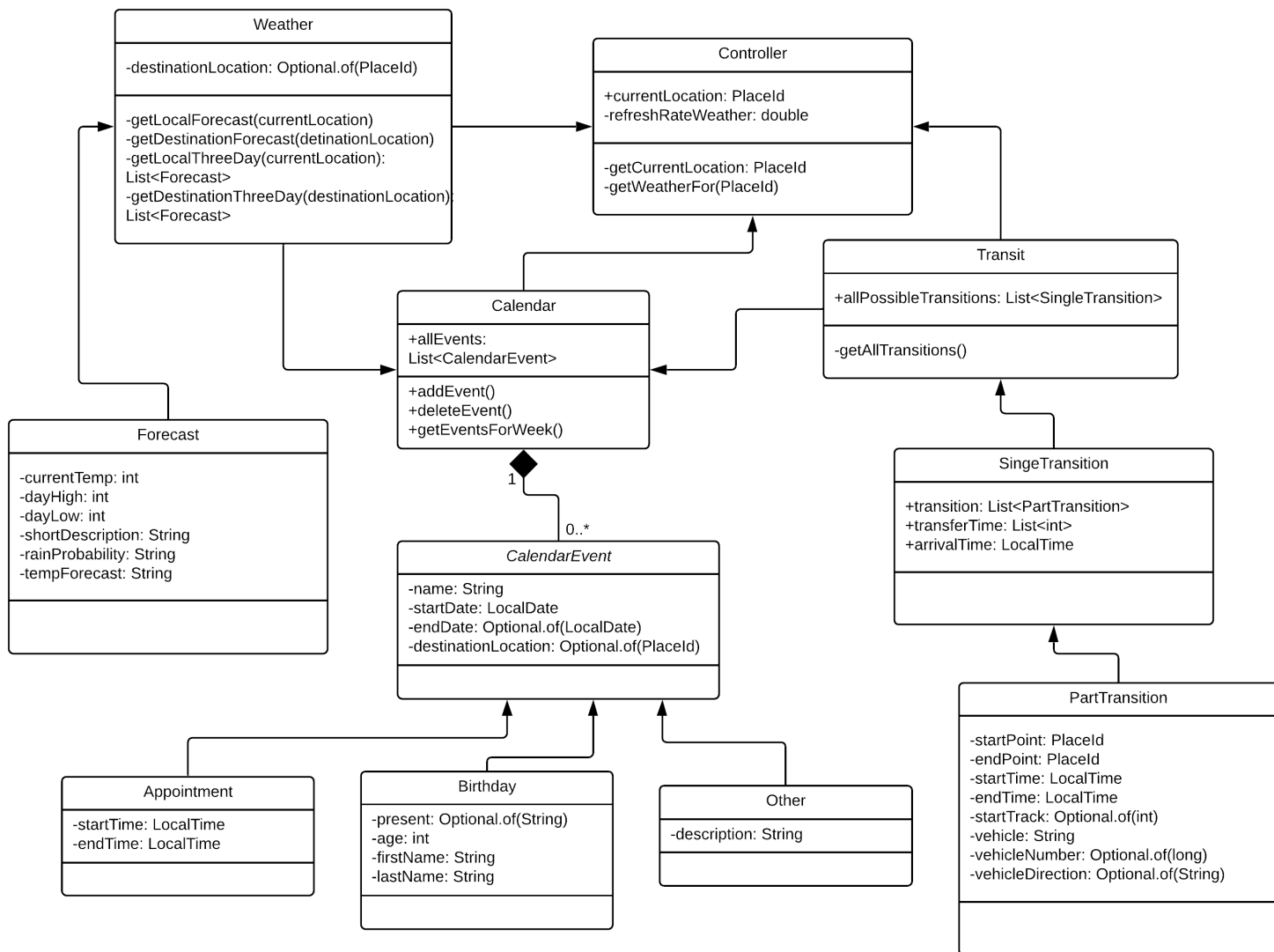
Wie schon in den Muss-Kriterien genannt, sollen sich die einzelnen Module der Anwendung asynchron aktualisieren, wozu sich die Verwendung von AJAX anbietet.

Programmiersprachen

- HTML5 mit Bootstrap
- JavaScript
- Java 13 unter Benutzung von JSP's

4 Diagramme

UML-Klassendiagramm



5 Zeitplan

Beim Planen des Projekts ergab sich der Zeitplan zur Realisierung wie folgt:

Kalenderwoche	Fortschritt
44	Fehlerbehebung und Fertigstellung des Wettermoduls
45	Einarbeiten in Bootstrap, Ajax und JSP's
46-47	Erste Version von Kalender- und Bahnmodul
48	Zusammenfügen aller Module und Bugfixing
49-50	Erster GUI-Entwurf
51-52	GUI-Fertigstellung und Bugfixing, ggf. sonstige Optimierungen
1-3 (2020)	Sämtliche Handbücher, Präsentation, etc.
4 (2020)	Testfälle und Endversion

6 Einsatz

Anwendungsbereich

Das Dashboard soll einen Überblick über den kommenden Tag, oder auch die kommende Woche schaffen, wodurch die Planung und das Zeitmanagement des Alltags erleichtert werden soll. Der Nutzer kann auf einen Blick sehen, welche wichtigen Termine anstehen und somit seinen Tagesablauf anpassen.

Zielgruppe

Die Anwendung richtet sich an Benutzer, die ihren Kalender als Terminplaner verwenden und somit die Möglichkeit haben, auf einfache Weise einen strukturierten Einblick über ihre Termine zu bekommen.