

MME2 ÜBUNGSDOKUMENTATION

WS 12/13

Team:

1. Name: Schultz	Vorname: Martin	Matr.-Nummer: 780331
2. Name: Feldmann	Vorname: Sascha	Matr.-Nummer: 778455

Vortragsthema: Karten API

Projekt: JumpUp

Wie wird das Endergebnis ungefähr aussehen (incl. Benutzerschnittstelle)?

Das Portal trennt sich in einen Konfigurationsbereich für den Fahrer und einen innovativen Suchbereich für Mitfahrer.

Der Benutzer/Mitfahrer interagiert über eine Map. In dieser Map werden alle verfügbaren Fahrten mit den von den Fahrern konfigurierten Routen angezeigt. Mit einem Filter kann der Mitfahrer nach bestimmten Routen (evtl. mit einem konfigurierten Ziel) in einem bestimmten Umkreis suchen. Änderungen des Filters wirken sich live auf die Map aus. Ziel ist es, dass der Fahrer lediglich Start- und Endpunkt angibt und sich Mitfahrer für beliebige Etappen finden. Etappen sollen vom Fahrer konfigurierbar sein. Das System berechnet einen Preisvorschlag für die jeweilige Etappe auf Basis des vom Fahrer eingegebenen Gesamtpreis der Strecke.

Welche Technologien, Geräte, Programmiersprachen, DBMS, Entwicklungsumgebungen werden verwendet? ***BITTE aktuell halten***

- Entwicklungsumgebung: Eclipse mit PHP Development Tools (Plugins von Zend zur Entwicklung inkl. Debugging mit dem Zend – Framework)
- Programmiersprachen: PHP 5.3, Javascript
- DBMS: MySQL 5.1
- Architektur-Frameworks: Zend 2.1.4 (PHP MVC – Applikationen → Backend)
- Bibliotheken: JQuery (Javascript), RequireJS
- API: Google Maps
- ORM: Doctrine
- Weitere Compiler: SCSS / Sass zur CSS-Kompilierung

Was ist daran multimedial?

Die zentrale Map, die viele innovative Konzepte vereint.

Welche Daten werden zwischen Client und Server übertragen?

- Formular-Daten: Profilinformationen (Fahrzeug, Inserate mit Preis,...)
- JSON: Ortspunkte (Koordinaten), Routen (mehrere geordnete Ortspunkte)

- Client – Google Maps Api: Konfigurationseinstellungen, Ortspunkte, Routen, Distanzen,...

Mit welchen Technologien werden die Daten übertragen?

- Client-Server: Formular-Daten bei Reloads, JSON – Daten bei der asynchronen Übertragung

Was wird in der Datenbank serverseitig verwaltet?

Die Routen mit Attributen wie Fahrzeug (Marke, Benzinverbrauch, Sitzplätze, Beinfreiheit,...), Etappen (Startpunkt, Endpunkt, individuelle Route, evtl. Raststätten, Mitfahrer, Preis, ...), Benutzerverwaltung (Name,...).

Funktionalitäten:

Tabellarische Übersicht von allen wichtigen und selbstverständlichen Produktfunktionen (mit Angabe der zugehörigen Ausbaustufe und Zuordnung im Team (wer macht was wann?).

Achtung, auch wenn die Dokumentation minimal ausfallen soll, ist es wichtig, die Produktfunktionen vollständig und genau anzugeben (so, wie Use Cases in UML). Nur dann lässt sich der Aufwand schätzen, nur so kann ein Kunde erkennen, was er erhalten soll.

ID	UseCase	Ausbaustufe	Zuständigkeit	Fortschritt (%)
1	Fahrer-Profil anlegen mit Personen- und Fahrzeuginformationen (Marke, Benzinverbrauch, Beinfreiheit, Stauraum, Ausstattung, durchschnittliche Reisegeschwindigkeit)	1 (Backend) 3 (Frontend)	Sascha Martin	100 (Backend)
2	Fahrer: Route mit Hilfe von GoogleMaps erstellen; Eingabe eines Start- und Zielpunkts und Abfahrts-/Ankunftszeit	1 (Backend) 1 (Frontend mit Maps)	beide	100
3	Mitfahrer-Profil mit Personen-Informationen anlegen (entspricht im wesentlichen #1)	1 (Backend) 3 (Frontend)	Sascha Martin	50 (Personen-Felder müssen noch ergänzt werden)
4	Mitfahrer: anhand von Standort und Wunschziel werden passende Routen von Fahrern mithilfe der GoogleMaps-API angezeigt	1-*(Backend) 1-* (Frontend)	beide	80 (Strategy zur Auswahl naher Routen fehlt noch)
5	automatische Berechnung des Fahrpreis (Anpassung möglich) [abh. von #4]	2 (Backend) 2 (Frontend)	beide	100
6	Mitfahrer: Einschränkung des Wunschradius möglich [abh. von #4]	2 (Backend) 2 (Frontend)	beide	0
7	Buchung: System schlägt Treffpunkte vor	2-* (Backend) 2-* (Frontend)	beide	0
8	Buchung: Mitfahrer kann Fahrt mit	3 (Backend)	beide	0

	vorgeschlagenen Treffpunkt buchen; Buchung enthält Fahrdetails (Treffpunkte, Zeiten,...)	3 (Fronend)		
9	Buchung: Fahrer bestätigt Buchung mit den angegebenen Details, „Bordkarte“ wird ausgestellt	3 (Backend) 3 (Fronend)	beide	0
10	Profil (Account) löschen	1 (Backend) 3 (Fronend)	beide	0
11	Bewertungssystem für Mitfahrer und Fahrer nach abgelaufener Fahrt	1 (Backend) 3 (Frontend)	beide	0
12	Kommentarfunktion für Inserate (z.B. „komme 5 min. später“) inkl. Automatischer eMail-Benachrichtung und Konfigurationsmöglichkeit	1 (Backend) 3 (Frontend)	beide	0
13	Benutzer muss sich registrieren.	1	Sascha	100
14	Der Benutzer muss die Registrierung in einer eMail bestätigen.	1	Sascha	100
15	Benutzer muss sich einloggen.	1	Sascha	100
16	Benutzer kann die Sprachen Englisch oder Deutsch einstellen-.	1	Sascha	100
17	Benutzereingaben müssen gefiltert und validiert werden.	1	beide	100
18	Routen werden asynchron (AJAX) vom Backend gepolled, um Reloads zu vermeiden.	1	Martin	100

Projektmanagement

An welchen Schwierigkeiten – außer mangelnden Wissen - könnte das Projekt scheitern?

- Komplexe Algorithmen (Trennung der Etappen, Abweichung von der eigentlichen Route, Finden des Treffpunktes,...)

Welche Alternativen könnten bei nicht überwindbaren Problemen das Projekt retten?

- Fahrer definiert die Zwischenstopps (Etappen) manuell (*Nachteil*: er hat mehr Planungsaufwand).

Projektfortschrittsdoku 1/3-Abgabe

ID	UseCase	Ausbaustufe	Zuständigkeit	Fortschritt (%)
1	Fahrer-Profil anlegen mit Personen- und Fahrzeuginformationen (Marke, Benzinverbrauch, Beinfreiheit, Stauraum, Ausstattung, durchschnittliche Reisegeschwindigkeit)	1 (Backend) 3 (Frontend)	Sascha Martin	100 (Backend)
2	Fahrer: Route mit Hilfe von GoogleMaps erstellen; Eingabe eines Start- und Zielpunkts und Abfahrts-/Ankunftszeit	1 (Backend) 1 (Frontend mit Maps)	beide	100
3	Mitfahrer-Profil mit Personen-Informationen anlegen (entspricht im wesentlichen #1)	1 (Backend) 3 (Frontend)	Sascha Martin	50 (Personen-Felder müssen noch ergänzt werden)
4	Mitfahrer: anhand von Standort und Wunschziel werden passende Routen von Fahrern mithilfe der GoogleMaps-API angezeigt	1-*(Backend) 1-* (Frontend)	beide	80 (Strategy zur Auswahl naher Routen fehlt noch)
5	automatische Berechnung des Fahrpreis (Anpassung möglich) [abh. von #4]	2 (Backend) 2 (Frontend)	beide	100
6	Mitfahrer: Einschränkung des Wunschradius möglich [abh. von #4]	2 (Backend) 2 (Frontend)	beide	0
7	Buchung: System schlägt Treffpunkte vor	2-* (Backend) 2-* (Frontend)	beide	0
8	Buchung: Mitfahrer kann Fahrt mit vorgeschlagenen Treffpunkt buchen; Buchung enthält Fahrdetails (Treffpunkte, Zeiten,...)	3 (Backend) 3 (Frontend)	beide	0
9	Buchung: Fahrer bestätigt Buchung mit den angegebenen Details, „Bordkarte“ wird ausgestellt	3 (Backend) 3 (Frontend)	beide	0
10	Profil (Account) löschen	1 (Backend) 3 (Frontend)	beide	0
11	Bewertungssystem für Mitfahrer und Fahrer nach abgelaufener Fahrt	1 (Backend) 3 (Frontend)	beide	0
12	Kommentarfunktion für Inserate (z.B. „komme 5 min. später“) inkl. Automatischer eMail-Benachrichtung und Konfigurationsmöglichkeit	1 (Backend) 3 (Frontend)	beide	0
13	Benutzer muss sich registrieren.	1	Sascha	100
14	Der Benutzer muss die Registrierung in einer eMail bestätigen.	1	Sascha	100
15	Benutzer muss sich einloggen.	1	Sascha	100
16	Benutzer kann die Sprachen Englisch oder Deutsch einstellen.	1	Sascha	100
17	Benutzereingaben müssen gefiltert und validiert werden.	1	beide	100

18	Routen werden asynchron (AJAX) vom Backend gepollt, um Reloads zu vermeiden.	1	Martin	100
----	--	---	--------	-----

Backend:

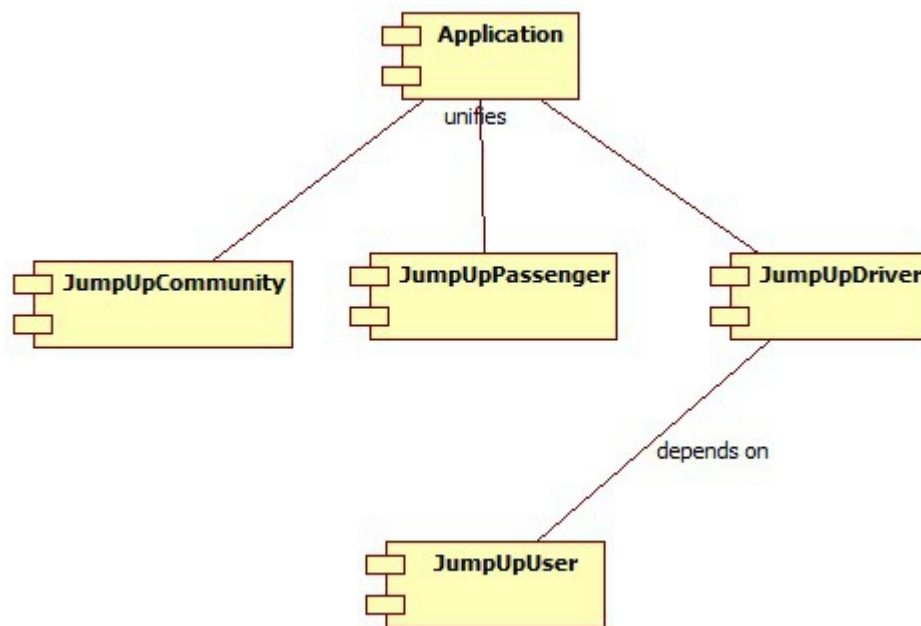
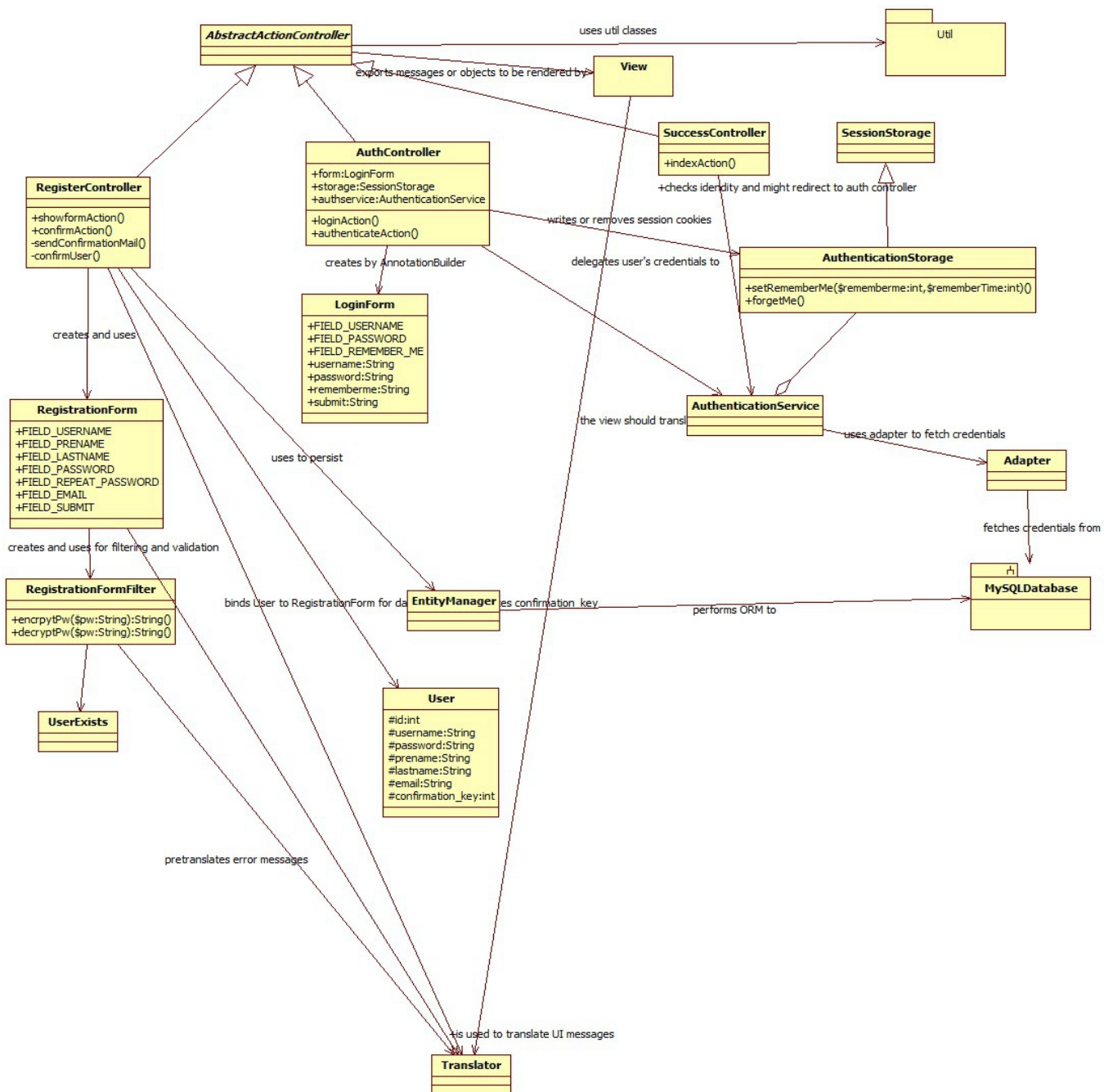
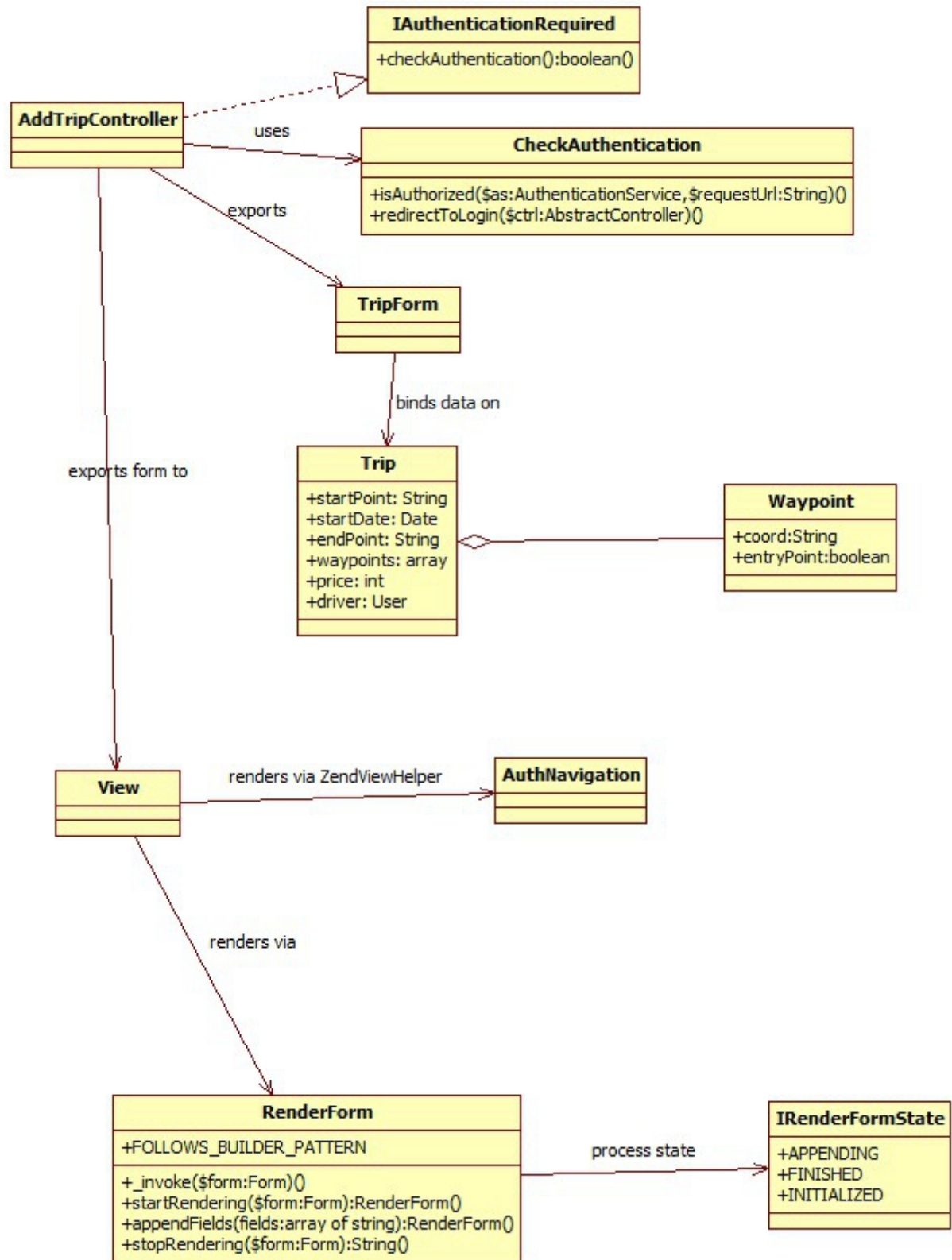


Abbildung 1: Backend - Modularisierung

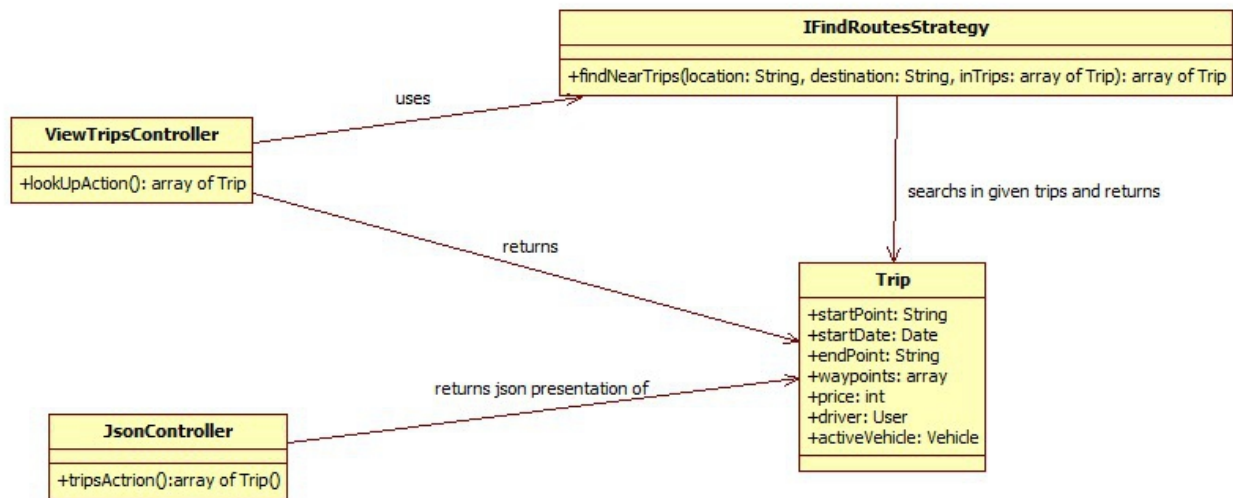
Backend – Modul JumpUpUser:



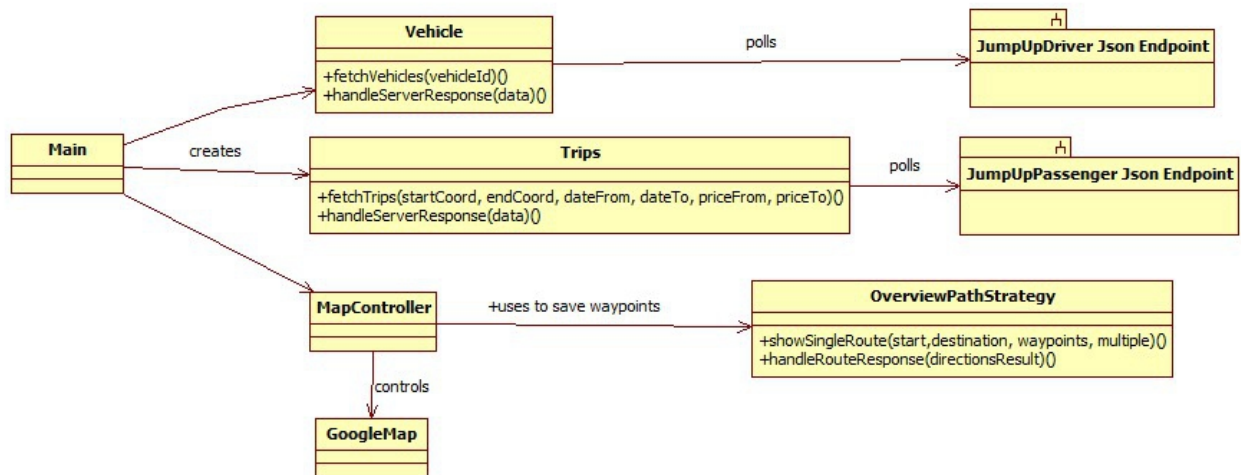
Backend – Modul JumpUpDriver:



Backend – Modul JumpUpPassenger:



Frontend:



Projektfortschrittsdoku 2/3-Abgabe

Tabellarische Übersicht von allen wichtigen und selbstverständlichen Produktfunktionen (mit Angabe der zugehörigen Ausbaustufe und Zuordnung im Team (wer macht was wann?) sowie Angabe, zu wie viel % diese realisiert und wie sicher sie getestet sind.

Gute Visualisierungen werden gerne gesehen.

Projektfortschrittsdoku 3/3-Abgabe

Tabellarische Übersicht von allen wichtigen und selbstverständlichen Produktfunktionen (mit Angabe der zugehörigen Ausbaustufe und Zuordnung im Team (wer macht was wann?) sowie Angabe, zu wie viel % diese realisiert und wie sicher sie getestet sind.

Gute Visualisierungen werden gerne gesehen.

Abschlussdokumentation:

(Bitte möglichst konkret / bei Bedarf bitte Anlagen beifügen)

1. Welche Architektur- und Designmuster wurden verwendet?
2. Welche wieder verwendbaren Komponenten wurden erstellt?
3. Welche externen Snippets, Klassen, Frameworks, Komponenten wurden verwendet?
4. Welche Features des Projekts sind innovativ?

Empfohlene Projektnote:

Hiermit gestatten wir Prof. Dr. Targo Pavlista, dem Fachbereich VI und der BHT Berlin die Video-Dokumentation unseres Projekts zu veröffentlichen und zum Download bereit zu stellen.

Berlin, den

(Unterschriften aller Team-Mitglieder)

Folgende kurze Textbeschreibung sollte neben dem Icon unseres Projektes sichtbar sein (was gemacht wurde, welche Technologien):

Wir haben ein Icon (Format max. 400 * 400 Pixel) selbst erstellt und mit auf die DVD/CD gepackt (ja, nein):