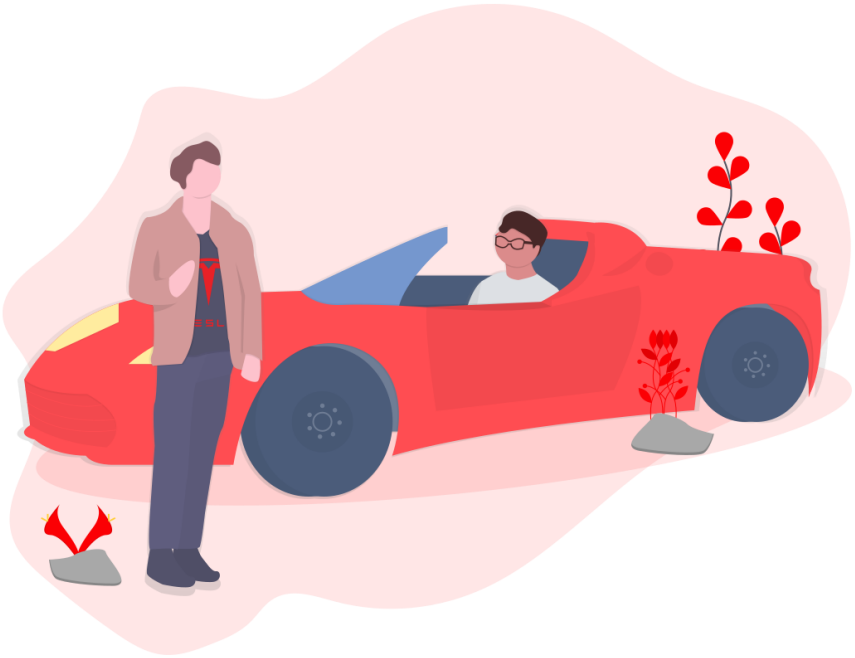


mobility

car sharing

Severin Machaz, Alex Smolders

Smolders & Machaz
Modul Auftrag
09.06.2022

Modulauftrag Mobility

Inhalt

Modulauftrag Mobility....1
Inhalt....1
Besprechungsnotizen....1
Lektion 1 – 2....1
Lektion 2 - 4....2
Lektion 4 – 6....2
Lektion 6 – 8....2
Lektion 8 – 10....3
Anforderungen....3
Klassendiagramm....3
Use-Case Diagramm....5
Wireframes....6
Logisch relational / MySQL....7
Github....8
Reflexion....8
Alex....8
Severin....8

Besprechungsnotizen

Lektion 1 – 2

Protokoll: 09.06.2022

3 Arten von Abos: Ohne Ermässigung. Mit Ermässigung, und eine wo jemand in deinem Haushalt einen Abo besitzt.

Wenn die Person im Haushalt auch ÖV abo hat, 50 -> 100

Maximale Mietdauer -> gibt keine, kostet einfach mehr desto mehr Zeit vergeht

Zahlungsmethode nur Rechnung -> erfolgt Ende Monat

Ein Abo gebühr Stundentarif und Kilometer

Gepäck stücke reinpassen Kofferraum

Protokoll: 09.06.2022

3 Arten von Abos: Ohne Ermässigung. Mit Ermässigung, und eine wo jemand in deinem Haushalt einen Abo besitzt.

Smolders & Machaz
Modul Auftrag
09.06.2022

Wenn die Person im Haushalt auch ÖV abo hat, 50 -> 100

Maximale mietdauer -> gibt keine, kostet einfach mehr desto mehr Zeit vergaht

Zahlungsmethode nur Rechnung -> erfolgt Ende Monat

Ein Abo gebühr Stundentarif und Kilometer

Gepäck stücke reinpassen Kofferraum

Lektion 2 - 4

Können sich Tarife ändern?

Ja

Muss verwaltet werden, wann ich das Auto gefahren wurde und wie lange?

Es ist ja eine Buchung als Kunde, man will ja von xx.xx.y bis xx.xx.y reservieren und dann geht man das Auto holen. Wen man 4h reserviert aber nur 3 Stunden gefahren dann werden 4h verrechnet.

Beim Darüber gehen der Zeit?

Strafgebühr, aber muss nicht dargestellt werden.

Lektion 4 – 6

Gibt es jeden Monat eine Rechnung, wenn man kein Auto gemietet hat?

Man zahlt einmal im Voraus das Abo und Rechnung nur Ende Monat für die Kosten

Braucht man eine Durchführung?

Für eine Durchführung braucht es auch Angebot, Auto wäre das Angebot also, wenn genau ein Auto irgendwo steht und eine Peron macht eine Miete kann man das als Angebot sehen.

Bei den Rabatten kann man GA und Familienrabatt haben oder nur einen?

Nur einen

Wie sieht es aus mit geblitzt zu werden?

Optional muss nicht. Gibt noch eine Strabuse

Macht alles komplexer, muss nicht modelliert und eingebaut werden. Weil sonst braucht es sonst auch Versicherung Selbstbehalt usw.

Lektion 6 – 8

Darf Person abhängig von einem Abo sein, heisst er existiert nicht ohne Das Abonnement? Auf der Datenbank heisst dies der Fremdschlüssel darf NULL sein

Smolders & Machaz
Modul Auftrag
09.06.2022

Praxis immer zuesrst Person, nacherher andere Prioritäten:

Immer zuerst die Person, immer non-identifying relationship.

Person kann auch kein Abo haben: Dies ist so, wird dann in der Datenbank erfasst.

Lektion 8 – 10

Heute waren keine Fragen dran.

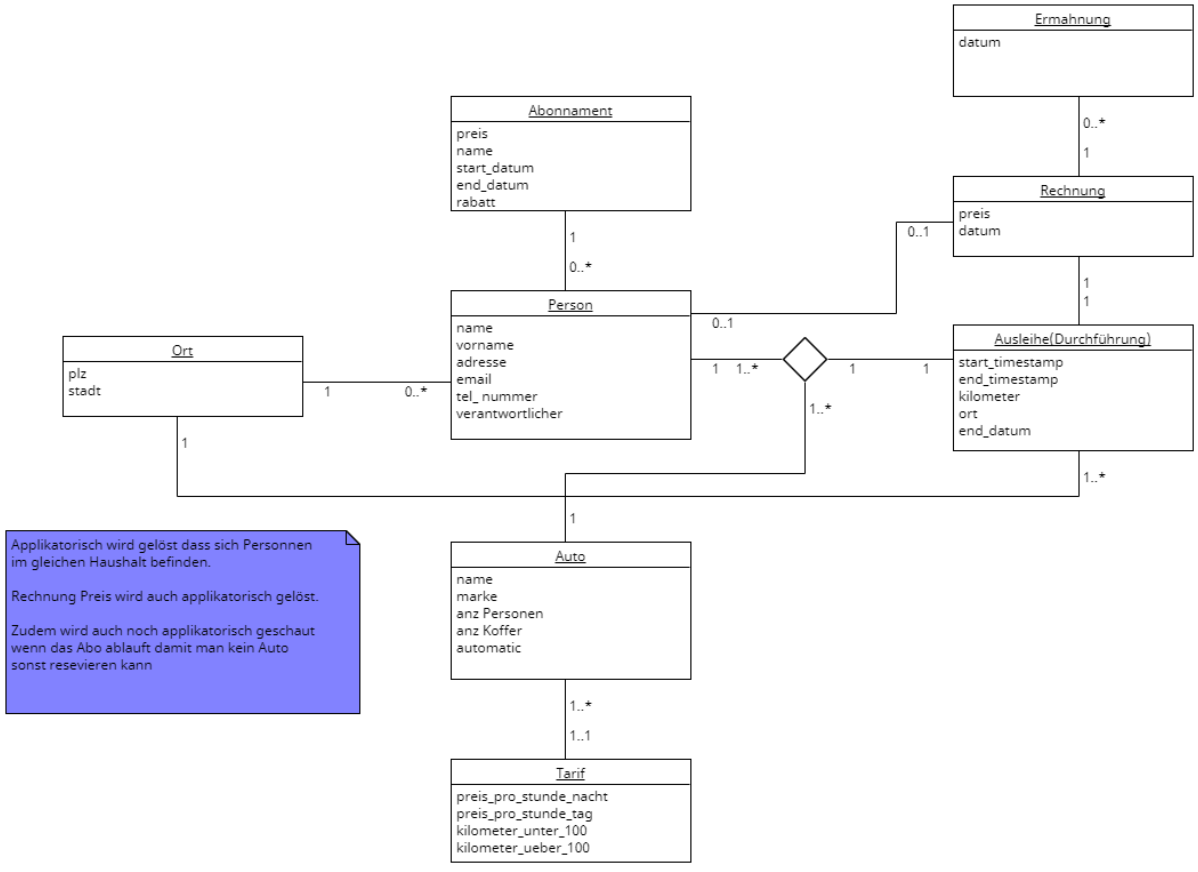
Anforderungen

Die Anforderungen für das Mobility Projekt sind sechs Lieferobjekt die der Ausgangslage entsprechend angepasst sind. (Klassendiagram, Use-Case Diagramm, Wireframes, Prozessabbildung, Logisch relational , MySQl)

Das Datenmodel hat folgende Anforderungen:

- Abbildung der Abonnement Optionen, deren Kosten und die Ermässigungen.
- Abbildung der Variablen Kosten bezogen auf Fahrzeug-Kategorie, Stundentarif (auch Zeitvariabel) und Kilometertarif
- Abbildung der Autos und deren Attributen, wie Name, Anzahl Personen oder Koffer
- Abbildung der Mitglieder samt deren Informationen wie Adresse, Name usw. ,für die Umsetzung von Ermässigungen für Haushälter
- Abildung der Verwaltung der Rechnungen.

Klassendiagramm



Smolders & Machaz
Modul Auftrag
09.06.2022

Ort: Die Klasse Ort speichert den Namen des Stades ab und zudem die dazugehörige PLZ. Es hat eine Beziehung zu Person, da man zu der Person die Adresse speichern muss. Heisst hier kann eine Person mehrmals die gleiche Ortschaft besitzen, und jede Adresse gehört zu einem Ort. Zudem wird der Ort der Ausleihe auch gebraucht, hier ist es wieder das gleiche es braucht ein Ort, welches mehrmals vorkommen kann.

Auto: Die Klasse speichert die wichtigen Attribute eines Autos wie im Auftrag beschrieben. Model, Name, Anz Personen, Koffer Manual oder nicht. Zudem wird es für die Durchführung Tabelle gebraucht, damit man Weiss welches geliehen wurde. Zudem kann dort mehrmals das gleiche Auto verwaltet werden, zudem hat es eine Beziehung mit dem Tarif, wo jedes Auto einen Tarif hat und dessen Tarif zum Teil mehrmals vorkommen kann, da ein paar den Gleichen haben.

Person: Diese Klasse speichert viele wichtige Infos zu einer Person, zudem noch ob er verantwortlich ist für den Familien Abo, wo dann die Rechnungen adressiert werden. Er hat eine Beziehung mit Ort, wurde schon erläutert, Mit Abonnement, damit zu jeder Person ein Abonnement. In der Klasse Person kann ein Abo mehrere Personen gehören, wiederum hat jede Person ein Abo. Zu diesem Abo hat jede Person individuell noch eine Rechnung zum Abonnement. Die Klasse hat auch noch eine Beziehung mit der Mehrfachassoziativer Klasse, wo dann alles zusammengefasst wird. Die Ausführung kann mehrere Personen beinhalten, mehrmals die gleiche, und jede Person eine Ausführung.

Tarif: Der Tarif speichert die wichtigen Informationen zu der Berechnung der Miete. Die Beziehung mit dem Auto ist wichtig, um dies anzuordnen. Ein Tarif gehört zu einem oder mehreren Autos, und ein Auto hat ein Tarif.

Rechnung: Eine Rechnung beinhalten den Preis, wie auch das fällige Datum. Die Rechnung gehört individuell zu einer Person und die Person hat 1 Rechnung, welches begleichen werden muss.

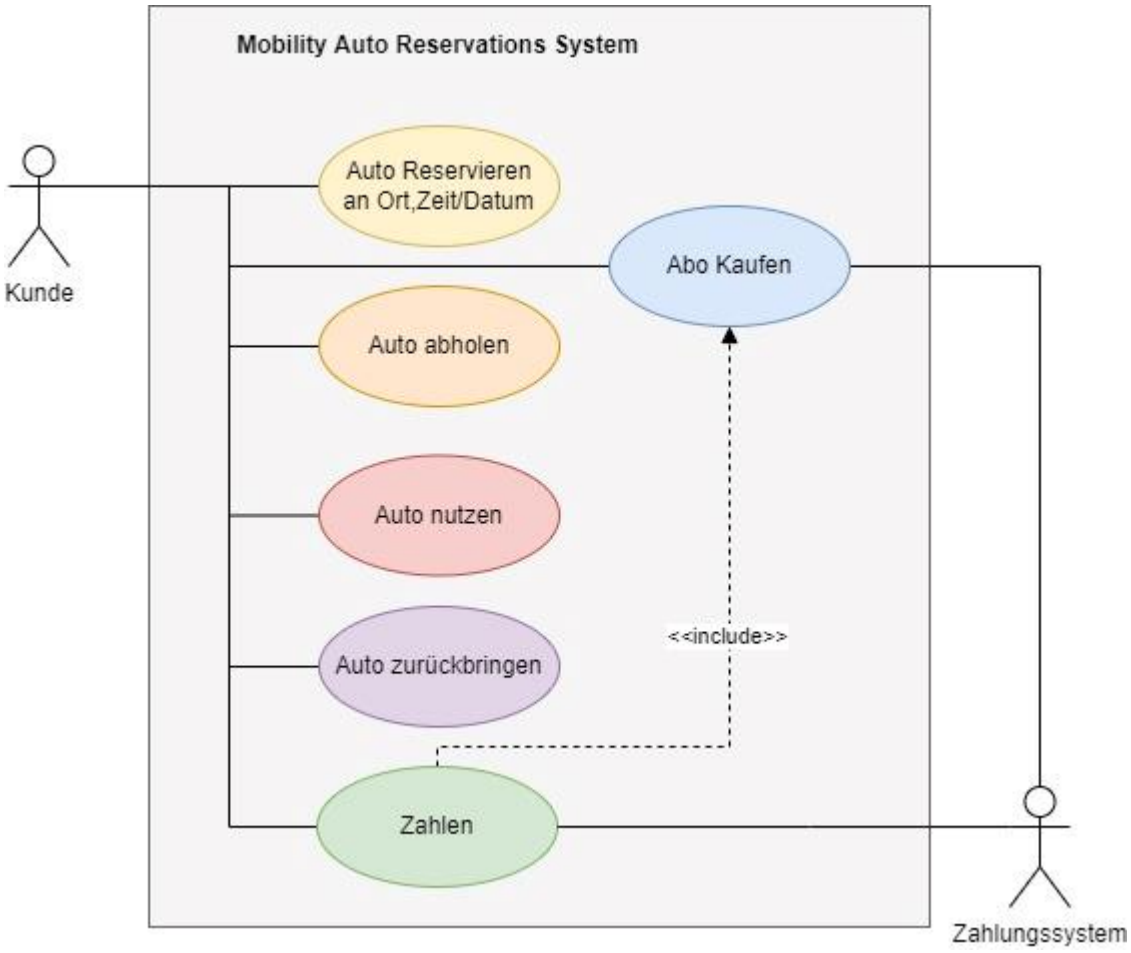
Ermahnungen: Dies ist eine Ermahnung das nächstfällige Datum. Eine Rechnung kann keine oder mehrere haben, und eine Ermahnung gehört zu einer Rechnung

Ausleihe: Der Zweck der Attribute ist die Ausleihe eines Autos zu registrieren mit den genauen Kilometer, Start end Timestamp für eine genau Zeitaufnahme. Di Ausleihe hat zudem noch eine Rechnung und die Rechnung gehört zu einer Ausleihe. Auch noch hat die Ausleihe einen Ort, wo dies Stattfindet.

Abonnement: Hier werden die jeglichen Abos erfasst, welches in der Ausgangslage fix vorgegeben sind. Zudem auch noch die Preise und allfälligen Rabatt. Eine Person hat einen Abo, ein Abo gehört zu mehreren Personen.

Smolders & Machaz
Modul Auftrag
09.06.2022

Use-Case Diagramm



Smolders & Machaz
Modul Auftrag
09.06.2022

Ein Kunde ist jeder registrierter Mobility Benutzer welcher Anspruch nimmt von dem Mobility Auto Reservation System.

Ein *Kunde* muss in der Lage sein ein Mobility Auto Reservieren um einen bestimmten Ort um an einem bestimmten Datum und Zeit.

Ein Kunde muss in der Lage sein, das von Ihm Reservierte Auto, abholen zu können am von Ihm festgelegten Standort, Datum und Zeit.

Ein Kunde muss in der Lage sein, das von Ihm reservierte Auto, entsperren zu können mit seinem Swiss Pass und dieses dann nutzen zu können

Ein Kunde muss in der Lage sein das Auto zurückzubringen und wieder mit seinem Abo dieses zu schliessen und seine Fahrt zu beenden.

Ein Kunde muss in der Lage sein sich ein Abonnement zu Kaufen. Ebenfalls ist hier der Actor Zahlungssystem wichtig, da dieses System für die Preis Berechnung zuständig ist.

Ein Kunde muss in der Lage sein, zu bezahlen für seine Fahrten. Dies beinhaltet den Abo Kauf, da dieser die Preis Berechnung beeinflusst. Ebenfalls ist hier der Actor Zahlungssystem wichtig, da dieses System für die Preis Berechnung zuständig ist.

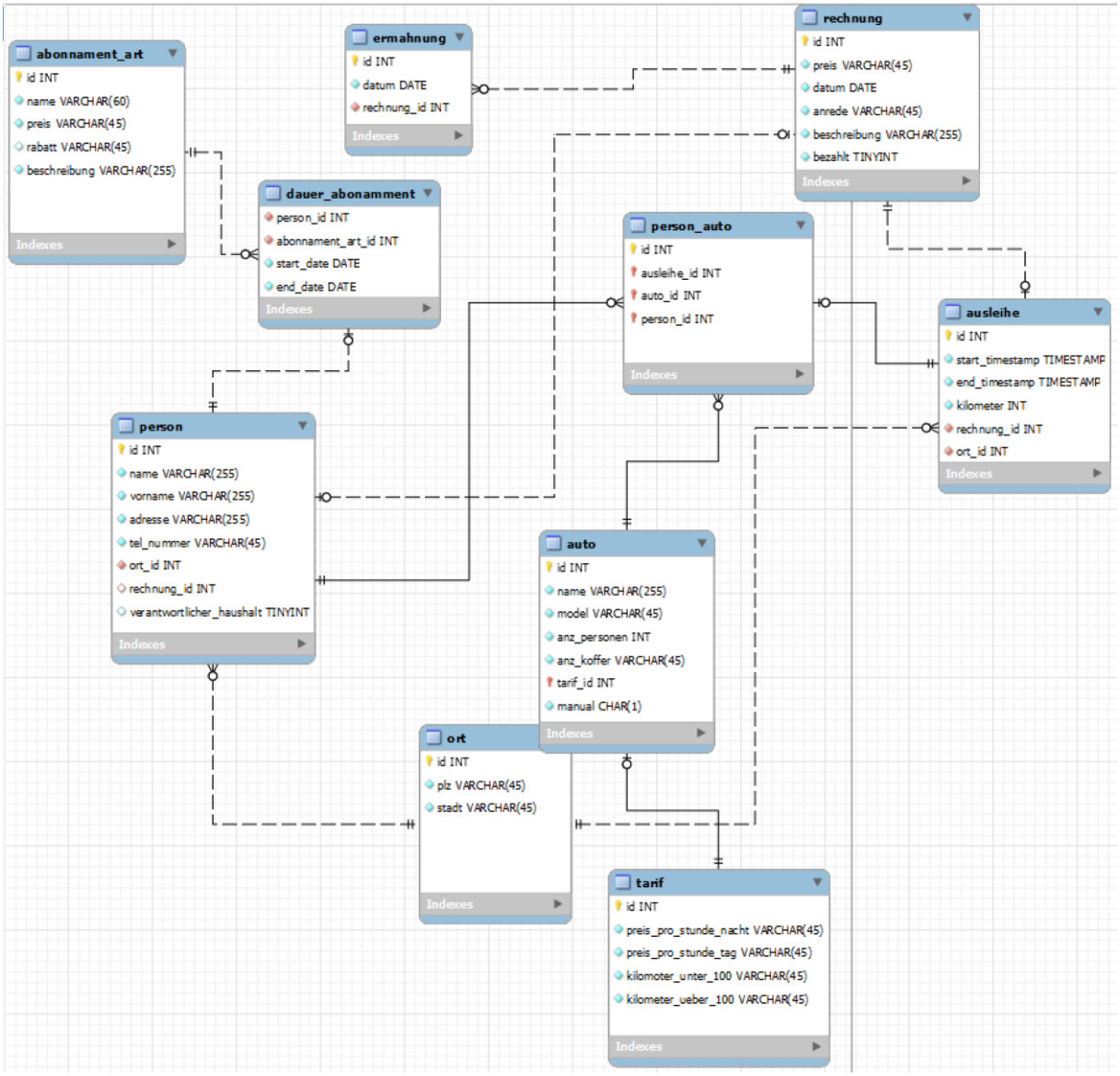
Das Zahlungssystem ist zuständig um die den Preis zu berechnen, diese dem Kunden mit zu teilen und dann die Bezahlung entgegen zu nehmen.

Wireframes

https://www.figma.com/file/597kPeipLJ72kMwiS7lpTt/Untitled

Smolders & Machaz
Modul Auftrag
09.06.2022

Logisch relational / MySQL



Hier haben sie im Klassendiagramm alle Beziehungen wo 1:1 waren zu 1:c, oder 1:1...* zu 1:mc geändert. Zudem wurden Schlüsseln hinzugefügt. Zudem die Beziehung von Abonnement zu Person ist auch komplett anders geworden, wegen der Eigenschaften des Fremdschlüssels. Falls die Eigenschaften nicht im Diagramm durch die Annotationen ersichtlich sind, im GitHub nachschauen. Es braucht zudem Identifying relationships bei person_auto, da eine Ausleihe nicht ohne die Anderen Tabellen Werte nicht existieren kann. Hier haben wir geschaut das eine Person, wie in der Praxis, immer erstellt werden kann. Bei den anderen, wie z.B ausleihe haben wir geschaut, dass die wichtigsten Sachen dabei sind, wie Ort und Rechnung. Das Attribut verantwortlicher ist auch für das Abo sehr wichtig, welches im Klassendiagramm nachgetragen wurde. Was hier geändert wurde vom Klassendiagram, ist das es eine neue Tablle gibt: dauer_abonamment. Diese Tabelle verhindert den Redundanzen welches man sonst hätte mit den Namen des Abos, Preis und Beschreibung. Diese hat hat nicht nur die Fremdschlüssel von Person und Abo_art, sondern auch end und start Datum welches applikatorisch gelöst wird.

Smolders & Machaz
Modul Auftrag
09.06.2022

Bei den Inserts wurde hauptsächlich geschaut das die Reihenfolge Sinn machen, damit man nicht im Anschluss alles updaten muss wenn man jetzt z.B eine Person vorher einträgt, ist aber möglich dies so einzutragen. Zudem wurde jeder Art von Beziehung berücksichtigt wie auch use cases. Boolean wurde geregelt mit einem TinyInt, welches 0 bedeutet false, und 1 true. Bei Manual ist m = manual, a = automatic, b = beide sim Sortiment.

Github

Repository mit sql code

Reflexion

Alex

Mir hat diese Project sehr gefallen, da ich das gelernte in den Stunden des Moduls hier in diesem Project anwenden konnte. Dies habe ich sehr wertvoll gefunden und habe auch all mein Wissen so gut wie möglich anzuwenden, dabei auch alles zu berücksichtigen. Ich habe mich hier in diesem Auftrag hauptsächlich um das logisch relationale Diagramm gekümmert und Severin um Use cases und wireframes. Ich denke die Aufteilung war gut, dennoch hätte ich ein wenig mehr provitiert wenn Severin seine Beteiligung in dessen Aufträgen grösser gewesen wäre. Wir haben immer unsere Arbeit kontrolliert und haben alles kritisch hinterfragt. Ich denke, dass ich meinen Beitrag hier gut geleistet habe und auch Severin, aber die Zusammenarbeit hätte besser laufen können bei den wichtigen Sachen, wie Klassendiagram und Log- rel-Diagramm. Aber was gut lief war die Implementierung vom Klassendiagramm zu log-rel. Hier habe ich immer wieder Fehler gefunden, welches lehrreich für mich waren, wie man am besten so eine Datenbank dann implementiert. Ich musste dann am Schluss alle Veränderungen dann auch noch im Klassendiagramm modellieren. Insgesamt würde ich sagen das Projekt hat sich definitiv gelohnt, da man einen Einblick bekommt wie man in der Praxis an einem grösseren Projekt die Lösung der persistenten Daten lösen würde.

Severin

Reflexion Mobility

Da dieses Modul mir ein wenig schwerer viel als andere, habe ich mich mehr auf Alex verlassen beim Arbeiten. Bei den Aufträgen, welche ich gemacht habe, habe ich ihn immer nochmal drüber schauen lassen, ich habe aber bei Ihm immer drauf geschaut und geholfen. Wir haben unsere Arbeiten über das Github repository geteilt und aktuell gehalten.

Er hatte zwar ein wenig mehr Arbeit als ich am Ende aber unsere Arbeitsaufteilung war trotzdem gut denke ich. Ich auch bin sehr zufrieden und zuversichtlich mit unserem Endresultat, weil wir die Zeit investiert haben und uns Mühe gegeben haben.

Der Auftrag kam zwar zu sehr einer stressigen Zeit, wo wir noch viele andere Dinge zu tuen hatten, aber an sich ist es eigentlich ein guter Auftrag. Ich würde sagen er ist nicht zu aufwändig oder stressig, aber man könnte Ihn ein wenig besser im Semester platzieren. Der Auftrag hatte definitiv ein positiven Lernwert auf uns und hat uns das Material noch mal gut nähergebracht. Ich fand es auch positiv das wir nicht unnötig viel machen mussten, und dass wir mit der Lehrperson sprechen konnten und Dinge streichen konnten. Auch die Lehrperson als Kunde für die Fragenstellung war ein Interessanter Ansatz.