SpaceDeliver

# Konzeption Des webservice

SpaceDeliver möchten Menschen die Körperlich nicht fit sind sowie Menschen unter Zeitdruck ermöglichen für sich einkaufen zu lassen, dies kann einmalig aber auch wiederkehrend sein.

# Anwendungslogik | spacedeliver

###### **Dienstgeber**

Wenn ein Nutzer erfolgreich ein Post auf eine Ressource wie z.B Bestellen gemacht hat wird eine neue Instanz erzeugt die dabei eine ID generiert. Somit kann man jede Bestellung einem Kunden zuordnen.

Bei einem POST oder PUT überprüft das System die übergebene Instanz einer Ressource und schaut ob die Attribute in der jeweiligen Jason Datei vorhanden sind.   
Im Fall dass eine fehlerhafte Datei übergeben wird zeigt der Server eine Fehlermeldung

###### **Dienstnutzer**

Der Nutzer hat die Möglichkeit Essen aus einer Liste auszuwählen und das dann zu bestellen.  
Dazu gibt er die Produkte die er benötig an oder wählt bestimmte Kategorien um die Suche zu verfeinern.  
Nachdem er Bestellt hat sucht das System einen Lieferanten der in der Nähe ist und die nötige Mobilität besitzt.  
Sobald ein geeigneter Lieferant die Lieferung bestätigt hat bekommt der Nutzer eine Berichtigung dass sein Auftrag entgegengenommen wurde.

# Externe web service (api)

Um Passende APIs zu finden wurde zuerst das Domainmodell überarbeitet um dort dann heraus lesen zu können was für Anwendungen man benötigt.  
Unser Hauptaugenmerkmal lag zuerst bei der Routenführung sowie bei unserem Zahlungsverkehr.

Da der Lieferant bei uns an der Kasse nur via Wireless Pay bezahlen kann, wird eine API eines Kreditkarteninstitut wie Master Card oder Visa benötigt, AMEX wurde hierbei bewusst außen vor gelassen da diese zurzeit nur mit Apple kooperieren.  
Nach längerem Informieren hat sich Master Card als profitabler und Nützlicher als Visa herausgestellt,  
da sie eine Reihe sehr nützliche APIs anbieten die Visa derzeit noch nicht bietet.  
Des weiteren ist die Developer Seite von Master Card benutzerfreundlicher und organisierter, hier hatte man konnte man schnell Informationen zum Thema Datenschutz finden, ganz im Gegenteil zu Visa bei denen man die Term of use und die Developer Agb lesen musste.

|  |  |
| --- | --- |
| Mastercard | Visa |
| + Übersichtliche Webseite | -Teilweise unübersichtlich |
| + Sandbox | +Sandbox |
| + Dev AGB 6.1 no fees | + Kostenloser Dev. Acc |
| +- TRUSTe Siegel | -Datenschutz , Term of use 6.1 & 6.2 ! |
| +StartUp Path Programm | -Verpflichtung sein produkt durch Visa Testen zu lassen (PenTesting) |

1. Master card Send  
   https://developer.mastercard.com/product/mastercard-send
2. Spend Controll  
   <https://developer.mastercard.com/product/spend-controls>
3. Automatic Billing Updater  
   <https://developer.mastercard.com/product/automatic-billing-updater>
4. Mastercard Identity Check Mobile  
   <https://developer.mastercard.com/product/identity-check-mobile>
5. Open Banking Connect  
   <https://developer.mastercard.com/product/open-banking-connect>

Bei der Routenführung sind 3 APIs besonders hervorgestochen, zum einen HERE die nicht nur eine gut strukturierte Webseite hat sondern auch sehr viele interessante Features. außerdem gibt es noch Google und Firefox die eine API zur routen Führung anbieten, die aber nicht so vielfältig wie Here ist.

Wir haben uns für HERE entschieden da diese zum einen -for free- sind, aber bei der Routen Führung Park Möglichkeiten und Fußgänger Wege berücksichtigen. Was HERE aber auch noch auszeichnet ist die Größe der Community.

1. HERE Tracking  
   <https://developer.here.com/products/tracking>
2. Routing  
   <https://developer.here.com/products/routing>
3. Maps  
   <https://developer.here.com/products/maps>

Da wir unserem Kunden einen gewissen Komfort bieten möchten haben wir gedacht dass es ganz nützlich wäre bei der Bezahlung Paypal mit einzubinden.  
Zu unserem Glück bietet Paypal eine art Sandbox an bei der man ein bezahlung verlauf kostenlos simulieren kann

1.Paypal REST API  
<https://developer.paypal.com/docs/api/overview/>

Es wurde auch noch überlegt Security APIs wie OAuth2.0. oder PhishTank zu benutzen, jedoch wurde das aus Komplexitätsgründen verworfen.

# Proof-of-concept

HERE

Beschreibung

Standortabfrage eines Einkaufhelfers um ihm den bestmöglichen einkaufsladen zu zeigen, des weiteren wird die Standortabfrage benutz um dem Kunden eine ungefähre Ankunftszeit mitzuteilen.

Exit-Kriterien

Erfolgreicher GPS empfang

Fail-Kriterien

* Kein Gps Signal
* Funkloch
* Ändern der Allow to access eigenschaft (IOS only)

Fallbacks

Benutze den Letzten Standort um herauszufinden wo sich die Person ungefähr aufhält.