

Web-basierte Anwendungen 2:

Verteilte Systeme

Nazan Aysune

Matrikelnummer: 11074900

Ideenfindung

Die zweite Phase fängt damit an einen Projektpartner zu finden. Da Ushpal und ich öfters zusammenarbeiteten, war es nicht schwer mich festzulegen. Nach der Teameinteilung konnten wir ohne Probleme mit der ersten Aufgabe anfangen. Die erste Aufgabe bestand darin eine Idee zu finden, die im Laufe des Workshops weitergeführt werden sollte.

Nach einigen Überlegungen und Besprechungen mit den Betreuern kamen wir auf die folgenden Ideen:

1. Tic Tac Toe - Spiel
2. Online Katalog für Bibliothek
3. FahrradVerleih

Um uns für eines dieser Ideen zu entscheiden, haben wir mit den Betreuern gesprochen, welches dieser Ideen für unser Projekt angemessen ist. Die Entscheidung fiel auf den Fahrradverleih. Nach einigen Recherchen konnten wir uns einen Überblick über die wesentlichen Funktionen eines solchen Systems verschaffen.

Die ersten Funktionen sollten wir in einem Exposé festhalten. Am selben Tag habe ich dieses erstellt und Online gestellt.

Exposé

Projektidee

In der zweiten Phase der Veranstaltung WBA2 haben wir uns Gedanken darüber gemacht, welches Szenario als verteiltes System dargestellt werden kann. Unsere erste Idee war es ein Zweipersonen-Strategiespiel zu erstellen (Tic Tac Toe). Dieses wurde uns von den Mentoren abgeraten, weil es nicht komplex genug ist.

Als weitere Idee hatten wir einen Online Katalog für eine Bibliothek (Bücher, Video). Diese Anwendung sollte den Verleih und die Ausleihe organisieren bzw. Benachrichtigungen an die Anwender schicken, jedoch würde es zu wenig gebrauch finden. Diese Idee wurde uns ebenfalls abgeschlagen.

Unser Mentor Benjamin K. hat uns eine weitere Idee genannt, und zwar die WG-Organisation in Bezug auf die Einkäufe. Jedoch konnten wir uns mit dieser Idee nicht anfreunden, da es keine sinnvollen Funktionen zu definieren gab.

Als Finale Idee haben wir an einem privaten Auto und Fahrradverleih gearbeitet. Aus Versicherungsgründen haben wir den Autoverleih rausgenommen und uns auf den Fahrradverleih konzentriert. Folgende Funktionen wurden als Möglichkeit angesehen:

- Online Registrierung
- Anzeige der verfügbaren Fahrzeuge und des Ortes
- Reservierung von Fahrzeugen
- Account einsehen bzw. bearbeiten
- evtl. Benachrichtigung schicken, wenn ein Fahrzeug sich in der Nähe befindet
- (Mitfahrgelegenheit anbieten (Liste/Benachrichtigung)).

Diese Idee wird von uns in den weiteren Wochen ausgearbeitet.

Welche Daten werden ausgetauscht?

Synchrone Kommunikation:

- Online Registrierung (Clientseitig) :

- Telefonnummer
- Vorname
- Nachname
- Straße
- PLZ
- Ort
- Land
- E-Mail
- Zahlungsart (Kreditkarte / Lastschrift)
- Fahrrad zur Ausleihe vorhanden / nicht vorhanden

Eingegebene Daten werden auf Validität überprüft und an den Server geschickt. Der Sender erhält eine Bestätigung (E-Mail / SMS)

- Anzeige der verfügbaren Fahrräder und des Ortes

- Fahrrad im Ort und im bestimmten Umkreis suchen und Anzeigen (Request, Clientseitig)
 - genaue Adresse eingeben (per GPS bestimmen lassen)
 - Umkreis eingeben
 - evtl. Zeitraum eingeben in dem ein Fahrrad benötigt wird (sonst Anzeigen von Fahrrädern, die ganztägig verfügbar sind)
- Fahrrad und Ort Anzeigen (nach erfolgreicher Suche (Response, serverseitig))
 - Fahrrad-ID
 - Besitzer

- Standort bzw. Abgabeort
 - Verfügbarkeitszeitraum
 - Preis
- Reservierung von Fahrzeugen (Clientseitig)
 - Zeitraumangabe

Asynchrone Kommunikation:

- Account einsehen bzw. bearbeiten (Clientseitig)
 - persönliche Daten können bei nicht aktiver Ausleihe bearbeitet werden
- Benachrichtigungen verschicken (Serverseitig)
 - Wenn sich ein verfügbares Fahrrad in der Nähe befindet wird eine Nachricht per E-Mail, SMS oder als Push-Benachrichtigung an den User gesendet
- Bewertung von den Usern (positiv / negativ)

Kommunikationsabläufe und Interaktionen

Weiterführend musste ein Informationsflussdiagramm erstellt werden um die Kommunikation der verschiedenen Komponenten darzustellen. Durch die Recherche im Internet verschafften wir uns einen Ansatz, jedoch gab es keinen konkreten Ansatz wie man Vorgehen musste. Somit erstellten wir die ersten zwei Entwürfe, die wir als angemessen empfanden.

Informationsflussdiagramm

Nach einem Gespräch mit den Betreuern in Bezug auf die Diagramme, wurde uns empfohlen beide Diagramme zu verbinden und eine Middleware abzubilden, die Client und Server verbindet. Zusätzlich wurde uns gesagt, dass ein Anwendungsfall (Szenario) unser Vorgehen erleichtern würde.

Ushpal und ich entschieden uns die Aufgaben aufzuteilen. Ich schrieb einen Anwendungsfall, der den Ablauf des Systems beschrieb (aus der Sicht der Nutzer). Anhand dieses Anwendungsfalles erstellte Ushpal den Informationsflussdiagramm. Da ich vorher noch keinen Anwendungsfall geschrieben habe, fiel es mir schwer dieses zu verfassen. Ich hatte mir vorgestellt die Abläufe des Systems darzustellen ohne die Nutzer in Betracht zu ziehen. Anschließend stellte ich meinen Entwurf meiner Teamkollegin vor. Sie machte mich darauf aufmerksam, dass die Sicht der Nutzer fehlte. Somit schrieb ich einen neuen Anwendungsfall. Szenario

Anwendungsfall Version 1

In dem Projekt geht es darum, dass Personen, die ein Fahrrad besitzen, dieses an andere verleihen können. Das System soll dies ohne großen Aufwand möglich machen. Die Personen müssen sich, um das System nutzen zu können, registrieren und angeben, ob sie ein Fahrrad zur Verfügung stellen. Nach der Registrierung kann der Nutzer seine persönlichen Daten bearbeiten. Um ein Fahrrad ausleihen zu können muss man je nach Ort und Zeit eine Suche durchführen. Nachdem dies geschehen ist erscheint eine Liste mit den passenden Fahrzeugen und der Nutzer kann sich eines auswählen. Dieses Vorgehen kann man auch als Benachrichtigung an sein mobiles Gerät senden lassen, sobald ein entsprechendes Fahrzeug in der Nähe ist. Die Nutzer haben auch die Möglichkeit sich ein Fahrrad für einen bestimmten Zeitraum zu reservieren. Die Abgabe findet zu dem vorher abgeklärten Zeitpunkt und Ort statt.

Anwendungsfall Version 2

Peter (User1) hat ein Fahrrad, das er nur gelegentlich braucht. Dieses möchte er anderen Personen, die ein Fahrrad brauchen zur Verfügung stellen und dazu trägt er es mit Herstellername, ModelNr und dem Betrag den er für das Ausleihen verlangt im Verleihsystem ein.

Hans möchte umweltbewusster von einem Ort zum anderen kommen. Deshalb möchte er sich ein Fahrrad anschaffen.

Er sucht im Verleihsystem nach einem verfügbaren Fahrrad in seiner Nähe. Der Server sucht in seinen Daten nach einem Teil mit dem Typ Fahrrad und übermittelt die unter der id des Anbieters abgelegt IP an Hans, der sich das Fahrrad von Peter aussucht. Nachdem Hans das Fahrrad von Peter abgeholt hat gibt Peter nun im Verleihsystem an, dass sein Fahrrad vergeben ist. Das System erhält dabei die id von Hans, die bei der Ortsanfrage mit übermittelt wurde, als Ausleihenden.

Projektbezogenes XML Schema

Im Meilenstein 1 sollte ein valides XML-Schema erstellt werden. Die Funktionen wurden schon in der vorherigen Aufgabe festgelegt. Somit mussten wir uns Gedanken um die Daten machen.

Das erste XML-Schema beinhaltete Attribute (Name, Vorname, usw.) die mitgegeben wurden. Jedoch sahen wir dieses Schema kritisch, deshalb haben wir uns mit den Betreuern Unterhalten und die haben uns geraten solche Informationen als Elemente darzustellen. Zudem stellten wir fest, dass einige wichtige Informationen fehlen. Im weiteren Vorgehen haben wir diese Kriterien umgesetzt.

In unserem System werden viele Daten referenziert. Da uns nicht bekannt war wie man vorgeht, haben wir in Literatur und Internet Recherchen betrieben. Wir haben zwei Varianten gefunden wie man Daten referenzieren kann. Die ID/IDREF's und KEY/KEYREF's .

ID/IDREF war einfach zu verwenden, jedoch hatte eine/zwei entscheidende Nachteile: Als UserVerweis können irrtümlich FahrradID Werte (und umgekehrt) eingesetzt werden. Die KEY/KEYREF's haben den Vorteil, dass die ID's eindeutig den jeweiligen Elementen zugeordnet werden. Diesen Vorteil bildet das ID/IDREF Verfahren nicht. Nach einer kurzen Überlegung stellte Ushpal fest, dass es sinnvoll wäre KEY/KEYREF's zu verwenden.

Da Probleme bei der Umsetzung entstanden und eine Fehlermeldung ausgegeben wurde, haben wir uns an Kommilitonen gewendet. Sie rieten uns ID/IDREF's zu verwenden, da JAXB kein KEY/KEYREF unterstützt. Zudem rieten Sie uns eine XML-Datei zu erstellen um die Daten für das XSD-Schema eindeutiger zu machen. Diesen Rat haben wir befolgt. Ich habe Ushpal bei der Umsetzung des XML-Dokumente unterstützt.

Ressourcen und die Semantik der HTTP-Operationen für das Projekt

In diesem Meilenstein haben wir uns für die Ressourcen Users, Fahrräder und Verleih entschieden und die Hauptressourcen bestimmt. Ich habe die Wahl der Ressourcen und die Methoden in der Dokumentation erläutert.

RESTful Webservice

Nachdem die Ressourcen festgelegt wurden haben wir mithilfe des JAXB Frameworks Java Klassen generiert. Anschließend haben wir uns an den verlinkten Tutorial Seiten für das Marshalling und Unmarshalling orientiert. Ushpal hat die Main Klasse entwickelt. Als die Main Klasse vollständig war ist beim Ausführen ein Fehler aufgetreten. Im Internet haben wir nach der Lösung für den Fehler gesucht, jedoch keine gefunden. Nach einem Gespräch mit den Betreuern haben sie uns darauf hingewiesen, dass keine Umlaute verwendet werden können und das dies zu Problemen führen würde. Somit haben wir das XML Schema überarbeitet und die Java Klassen neu erzeugt, aber das Problem bestand immer noch.

Die Betreuer boten uns an den Code per E-Mail an sie weiterleiten sollen. Die Antwort erhielt Ushpal und arbeitete dies aus. Ich habe anschließend die Operationen, die mit PUT, GET, POST und DELETE verarbeitet werden, festgelegt und beschrieben. Die Beschreibung und die Ausgabe ist in der Dokumentation festgehalten.

Konzeption + XMPP Server einrichten

Aus zeitlichen Gründen und wegen fehlendem Wissen konnten wir diesen Meilenstein nicht umsetzen. Jedoch habe ich mir Gedanken darüber gemacht wie man es umsetzen könnte und dieses in der Dokumentation beschrieben.

Ausarbeitung

LeafNodes sind Topics, in denen etwas veröffentlicht wird und somit auch asynchrone Benachrichtigungen. Als Topics werden in dieser Anwendung der Standort definiert. Wenn ein Nutzer benachrichtigt werden möchte, wenn sich ein verfügbares Fahrrad in der Nähe befindet, wird eine Nachricht per E-Mail, SMS oder als Push-Benachrichtigung an den User gesendet.

Hierbei wäre es möglich die Typen als Collection Nodes zu definieren und darunter einer verfeinerte Suchmöglichkeit zu bieten z.B über den Hersteller. Dies war für das Konzept nicht erforderlich, da es für den Subscriber wichtiger ist ein Fahrrad eines bestimmten Types auszuleihen. Die Suche über den Hersteller ect wäre optional, deshalb wäre es sinnvoll die Lösung mit den Typen als Nodes umzusetzen. Eine Möglichkeit wäre eine Verknüpfung zwischen

PubSubService und dem RestfulWebService und könnte eine URI innerhalbdes WebServices des entsprechendem Fahrrades an den Subscriber weitergeben, welcher sich mit dieser wiederum das Fahrrad auf dem Webservice genauer anschauen kann. Aus zeitlichen Gründen konnte dies nicht Umgesetzt werden.

Publisher und Subscriber

Publisher sind in diesem Fall die Nutzer die ein Fahrrad zur Verfügung stellen und dieses ausleihbar ist. Sobald ein Fahrrad in der Nähe ist soll eine Benachrichtigung erstellt werden, die dem Subscriber eine Nachricht sendet. Subscriber sind die Personen, die sich im System befinden und benachrichtigt werden möchten. Wenn dies der Fall ist muss der Standort freigegeben werden, damit dieser Dienst genutzt werden kann. Ein Nutzer kann sowohl als Publisher auftreten, wenn er ein Fahrrad zu verleihen hat, sowie auch als Subscriber, wenn er ein Fahrrad benötigt. Wenn ein Fahrrad vergeben wurde soll es wieder aus der Queue gelöscht werden, sodass nur verfügbare Fahrräder angezeigt werden. Sinnvollerweise würde bei diesem Verfahren nur eine URI zum entsprechenden Fahrrad übermittelt, möglich wäre hier die Verknüpfung mit dem Webservice.

Während Ushpal sich vorwiegend mit der Programmierung befasst hat, habe ich ihr mit Ratschlägen und Recherchen zur Seite gestanden. Die Umsetzung der Aufgaben wurde von mir ausgiebig dokumentiert, dass ich problemlos die Dokumentation fertigstellen konnte. Die Aufgaben haben wir in der Workshopzeit verarbeitet und das weitere Vorgehen ausdiskutiert. Da die Umsetzung nicht einfach war setzten wir uns mit den Betreuern in Kontakt, aber es lief über Ushpals E-Mail, deswegen hat sie die Lösungen umgesetzt.