

Projektbegründungen

Grundlagen

Projektidee

Die erste Projektidee entstand durch die Betrachtung eines Alltagsproblems der TH-Köln. Das Problem besteht in mangelnden Parkplätzen und unzureichende Organisation dieser. Häufig wird beobachtet wie die Parkplatzkapazität nahezu vollkommen ausgeschöpft ist, die Situation auf dem Parkplatz durch Falschparker verschlechtert wird, die zum Teil mehr als eine Parklücke benutzen und häufig Schäden an Fahrzeugen auftreten.

Nach ersten Überlegungen zu Ziel- und Lösungsansätzen zeigte sich, dass das Problem dem Umfang des Projekts nicht zu Genüge ist, da aufgrund von zu wenigen Stakeholdern die wirtschaftliche bzw. gesellschaftliche Relevanz zu gering wäre. Ein Projekt ohne diese Eigenschaften verspricht im Regelfall nicht profitabel zu sein.

Aus den genannten Gründen wurden weitere Stakeholder mit ähnlichen Problemen gesucht und erweitert. Das Ergebnis war eine Ausweitung der Zielsetzung auf alle Beteiligten des allgemeinen Straßenverkehrs. Die Projektidee beschreibt nun im Exposé die Parkplatzsuche in unmittelbarer Umgebung, Vermietung von privaten Parkplätzen und Navigation zu diesen. Dies hat eine enorme Erweiterung der Zielgruppen zu Folge und erhöht gesellschaftliche sowie wirtschaftliche Relevanz.

Domänenrecherche

In einer ausgeführten Domänenrecherche ergeben sich festgeschriebene Gesetze, die die Entwicklung des Projekts einschränken können. Beispielsweise darf ein Mobiltelefon gesetzlich nicht am Steuer eines Fahrzeuges benutzt werden. Dieses muss bei der Gestaltung der Funktionalitäten berücksichtigt werden. Aus der Domänenrecherche ging hervor, dass Parkplätze Unterschiede in Struktur, Organisation und Eigenschaften aufweisen. Aus diesem Grund wurden Zusatzinformationen wie Öffnungszeiten und Größe mit in die Dokumentation und Entwicklung miteinbezogen.

Marktrecherche

Die Marktrecherche bezieht sich auf die Konkurrenz zu dem im Projekt geplanten System auf dem aktuellen Markt. Das Augenmerk wurde hier auf die Eigenschaften und Funktionalitäten der größten und erfolgreichsten Konkurrenten gelegt, um bereits erfolgreiche Konzepte zu erkennen und Probleme in der Nutzbarkeit o.Ä. herauszustellen. Schlussendlich wurden Nachteile und Vorteile der Konkurrenz herausgearbeitet um unser Alleinstellungsmerkmal zu verfeinern und anschließend deutlich zu machen. Dadurch hängen Alleinstellungsmerkmal und die Marktrecherche eng zusammen und verweisen aufeinander.

Kommunikationsmodell

Das Kommunikationsmodell zeigt die Unterschiedlichen Stakeholder im Problemraum.

Im Ist-Zustand benutzen alle Stakeholder unterschiedliche Wege um das selbe Problem zu lösen. Der Soll-Zustand zeigt in einem Kommunikationsmodell die Problemlösung der Stakeholder mithilfe des Projekts. Das zu entwickelnde System allen gleichzeitig helfen. Die Stakeholder brauchen nicht mehr unterschiedlichste Unternehmungen machen, sondern nur noch ein einziges System zu nutzen.

Risikoanalyse

Mögliche Risiken des Projekts wurden in einer Risikoanalyse erarbeitet. Hier flossen Recherchen und vorangegangene Artefakte mit ein. Vorerst gab es ein Risiko auf unzureichende Datenmengen, was eine Recherche zu Quellen die OpenData anbieten geführt hat. Daraus resultierte das Risiko auf Redundanzen unterschiedlicher Daten und Datenformate. Hierzu wurde Aufgrund der Risikoanalyse eine Serialisierung bzw. Zusammenführung der Daten geplant. Weitere Einflüsse waren beispielsweise die gesetzlichen Rahmenbedingungen aus der Domänenrecherche. So kann ein Risiko definiert werden, das es unmöglich macht eine Anwendung zu gestalten, die während der Fahrt benutzbar ist.

Proof-of-Concepts

Um die Risiken besser abwägen zu können wurden Proof of Concepts erstellt; bei der Erstellung dieser wurden die hauptsächlich problematischen Teile des Dienstes ermittelt und versucht umzusetzen.

Eine der Hauptfragen war, ob es möglich ist die Google Maps API weitestgehend zu überarbeiten. Im Proof of Concepts wurde die Google Maps API überarbeitet und somit die Möglichkeit der Umsetzung bewiesen.

Es wurde sich für eine Google Maps Schnittstelle entschieden, da die eigene Umsetzung von deutschlandweitem Kartenmaterial den Arbeitsumfang bei weitem übersteigen würde. Google Maps lässt sich bis zu einem Traffic von 1000 Requests pro 24 Stunden kostenlos verwenden. Bei einer kommerziellen Umsetzung des Projektes ließe sich aber eine kostengünstige Lizenz bei Google erwerben die den Traffic weiter erhöht. Die Entscheidung ist auf Google Maps gefallen, da es keine für unsere Zwecke ausreichende Karten Alternative gibt, welche genug Unterstützung für frische Entwickler bieten.

Zusätzlich als problematisch bewertet wurde die prototypische Umsetzung des Dienstes mit Open-Data aus Köln, da die Aktualität der Daten in Frage gestellt wurde. Die Daten wurden in regelmäßigen Zeitabständen überprüft und es haben sich Änderungen gezeigt, was die regelmäßige Aktualisierung und somit die Aktualität der Daten bewies. Zusätzlich wurde die Implementierung einer Schnittstelle zur Einbindung der Open Data als Problem gewertet, und es musste überprüft werden, ob es möglich ist die Daten einzubinden, diese Frage wurde schnell beantwortet, da die Daten sich in JSON Format befinden und somit die Einbinung relativ leicht zu bewerkstelligen sein sollte.

Durch die Proof of Concepts ist eine Erweiterung der Anwendungslogik entstanden. So wurde unter anderen kleineren Überarbeitungen des Konzeptes die Entwicklung eines Parkleitsystems beschlossen.

Methodischer Rahmen MCI

Bei der Entscheidung welcher Methodische Rahmen gewählt wird, muss eine umfangreiche Abwägung der verschiedenen populären Rahmen gemacht werden. Im Anschluss haben wir Argumente gesammelt welche für und gegen die einzelnen Methodischen Rahmen sprechen. Im Endeffekt wurde sich für eine Mischung aus der DIN EN-9241 und Teilen von anderen Methoden entschieden.

Usability Engineering Lifecycle

nach Deborah Mayhew

Der Usability Engineering Lifecycle besteht aus einem unflexiblen Drei-Phasen Vorgehen was neben der Entwicklung von benutzerfreundlichen Programmen vor allem die Entwicklung von grafischen Benutzeroberfläche vereinfacht.

Ähnlich wie bei der ISO DIN EN-9241 sind auch im Usability Engineering Lifecycle

Iterationen nach den angewendeten Arbeitsschritten vorgesehen. Die Unflexibilität dieses Drei Phasen Vorgehens besteht hauptsächlich aus festen Zeitpunkten und konkreten Folgeschritten, die den Entwicklern nur wenig Handlungsspielraum neben diesen Iterationen gibt.

Ein Vorteil besteht dadurch allerdings in der leichteren Planbarkeit des Projektes, die unerfahrenen Projektteams Unterstützung sowie Sicherheit bieten kann.

Für die zu entwickelnde Parkplatz-App ist dieses Vorgehen nicht flexibel genug, da gerade in der Planungsphase die Zielgruppen eine erhöhte Aufmerksamkeit benötigen, da diese den kompliziertesten Teil der Entwicklung beherbergen. Ein Teil des Usability Lifecycle wird verwendet, damit die Oberfläche in der Planungsphase möglichst greifbar und leicht verständlich bleibt werden Mockups erstellt und diese nach Prüfung in Iterationen überarbeitet.

Usage Centered Design

nach Lockwood & Constatine

Das weltweit populäre Usage Centered Design nach Lockwood und Constantine ähnelt sehr dem MCI-Modell des ISO DIN EN-9241, wobei der Schwerpunkt der Entwicklung im Vergleich zur ISO nicht auf Benutzer, sondern auf die Benutzung des Endproduktes gelegt wird.

Für die Parkplatzapp ist dieses Vorgehensmodell neben der ISO Norm das interessanteste, da in der Entwicklung eines Schwerpunktes eine Abwägung zwischen der Benutzung und den Benutzern gemacht werden muss. Um allerdings keine App zu entwickeln, die den gegebenen Umfang übersteigt, wird in der Entwicklung des Projektes Parkplatzapp der Schwerpunkt auf die Benutzer gelegt um die Größe des Projektes im Rahmen zu halten.

Scenario Based Usability Engineering

nach Rosson & Carrol

Scenario Based Usability Engineering sieht Szenarios als universelles Werkzeug der Projektentwicklung. In Scenario Based Usability Engineering geht es darum das Menschen typische denken in Szenarios zu unterstützen und somit Schritt für Schritt die Usability des Projektes zu stärken.

Ein Problem bei der Entwicklung mit Szenarios besteht darin, dass Szenarios häufig ein sogenanntes „Festfahren“ also die verfrühte Entscheidung einer Teilentwicklung ohne genügend Evaluation, da die Entwicklung mit Szenarien durch die hohe Spezifizierung Alternative Lösungen erschwert und somit im Worst Case ausblendet.

Dies ist als das größte Risiko in der Entwicklung mit Szenarien zu sehen und gerade bei unerfahrenen Entwicklern als Problem zu werten. Da die präzisen Details der Parkapp erst in der Entwicklung nach Abschätzung des Marktes sowie der Analyse der Benutzern fest stehen, kann es passieren, dass man sich im Scenario Based Usability Engineering zu früh auf ein Szenario versteift, welches die Lösungsanforderungen aufgrund von unverhergesehenen Wendungen nicht mehr erfüllen kann.

Discount Usability

nach Jakob Nielsen

Discount Usability versucht gut bedienbare grafische Oberflächen mit dem geringsten Ressourceneinsatz zu ermöglichen. Die vorgegebenen Methoden eignen sich besonders für statische Kommunikation wie beispielsweise Webseiten oder Formulare sehr gut, versagen allerdings ohne weitere Bearbeitung der Vorgehensweisen für dynamische bzw.

interaktive Dialoge.

Da die Parkplatzapp in ihrer Oberfläche von Grund auf dynamisch sein sollte, lässt sich die Discount Usability Methode nach Jakob Nielsen ohne weitere Bearbeitung nur äußerst beschränkt anwenden und wird deshalb für die Parkplatz App vernachlässigt.

Contextual Design

nach Beyer & Holtzblatt

Contextual Design bezieht sich vor allem auf den menschenzentrierten Nutzungskontext des Systems. In einem contextual inquiry werden Daten beispielsweise über Umfragen gesammelt und diese anschließend ausgewertet, um den Nutzungsrahmen der Benutzer besser verstehen zu können.

Da der Nutzungskontext der Parkapp von Anfang an relativ klar ist, werden wir keine Methoden aus dem Contextual Design verwenden.

DIN EN ISO-9241

Die Normenreihe DIN EN ISO 9241 ist im Bezug auf das hier beschriebene System vor allem in den Teilen 110 (Anforderungen an ein interaktives System) und 210 (Prozess zur Gestaltung eines Interaktiven Systems) interessant, und deshalb wird auch nur auf diese beiden Teile eingegangen.

Diese Norm bezieht sich hauptsächlich auf die Anforderungen von Nutzern anstatt auf die Nutzung einzugehen. Die ISO-9241 unterstützt durch Empfehlungen und vorgegebenen Vorgehensweisen bei der Entwicklung für individuelle Nutzergruppen und hilft ihre Anforderungen herauszuarbeiten.

Insgesamt kann man sagen, dass die DIN EN ISO-9241 ein in den sieben Grundsätzen der Dialoggestaltung aus Teil 110 ein sehr flexibles sowie skalierbares Rahmenwerk um die Interaktion mit den Benutzern möglichst einfach zu gestalten. Hierbei helfen die in Teil 210 Erwähnung findende Artefakte bei der Entwicklung. Durch ihre hohe Flexibilität und die Entwicklung nahe am Benutzer wird die DIN EN ISO-9241 in der Entwicklung der Parkapp zur Hilfe gezogen.

Im Zuge dessen werden die Anforderungen an die Nutzer entwickelt, dazu werden Stakeholder festgestellt, definiert und daraufhin analysiert. Anschließend werden User Profiles erstellt um die Nutzer noch besser kennen zu lernen. Bis schlussendlich in den daraus zu entwickelnden Personas potentielle Beispielnutzer entstehen. Das Ergebnis dieser Artefakte sollen die Anforderungen an die Nutzer feststellen. Durch Iteration an verschiedenen Entwicklungsphasen des Projektes können diese Anforderungen weiter überarbeitet und verfeinert werden.

Entwicklung des Architekturmodells

Das Architekturmodell entstand aus den Erfahrungswerten der Teammitglieder.

Jeweils beide haben nur etwas Erfahrung mit Node.js und Redis sammeln können. Dies hat uns gezeigt, dass es das Beste ist dieses Modul zu nutzen, da es sehr einfach zu handhaben ist und viele Möglichkeiten besitzt sich mit Tools/Plugins wie zum Beispiel Express zu erweitern. Auch Redis wurde gewählt, weil die Daten der Kommunikation sehr einfach gehalten werden, was sehr zu Gunsten von Redis kommt.

Um eine Übersicht über die einzelnen Teile des Dienstes und ihre Kommunikation zu erhalten wurde ein Architektordiagramm erzeugt, welches im wesentlichen aus drei Teilen

besteht; zum ersten dem Dienst selber welcher die vollständige Präsentationslogik, sowie für den Benutzer individuelle Anwendungslogik beinhaltet. Die Anwendungslogik wurde nicht vollständig auf dem Server geplant, da Teile der Logik nur für einen Nutzer interessant sind und keinen Austausch mit anderen Nutzern benötigt werden. Ein Beispiel dafür ist der Parkzeitwecker, welcher die geparkte Zeit mitzählt und dem Nutzer den Preis für den aktuellen Parkvorgang anzeigt, oder die Ortung des eigenen Fahrzeugs. Die Logik wurde auf den Dienst gelegt um die Kommunikation zwischen Server und Dienst möglichst gering zu halten, um auch ohne die Nutzung der mobilen Daten auf dem Mobiltelefon weiterhin Teile der App zu verwenden.

Auf der Serverseite des Dienstes werden vor allem Daten gehalten, welche für alle User die gleiche Relevanz haben oder jene die zwischen ihnen ausgetauscht werden. Ein Beispiel hierfür ist die aktuelle Belegung der Parkplätze. Außerdem enthält der Server die restliche Anwendungslogik, wie beispielsweise die Kommunikation zwischen den einzelnen Parkteilnehmern.

Im Hintergrund werden die abzurufenden Daten auf einer Redis Datenbank gespeichert, da die Programmierer mit dieser Art von Datenbank bereits gearbeitet haben und somit keine zusätzliche Einarbeitungszeit für die Datenbank benötigt wird. Außerdem erfüllt die Redis Datenbank vollständig die ermittelten Voraussetzungen an die benötigte Datenbank ohne überflüssige Datenmengen zu übertragen oder abzuspeichern.

Die Kommunikation zwischen den Entitäten Dienst und Server werden vollständig in JSON stattfinden, da JSON sich seit langem als zuverlässiges Kommunikationsprotokoll bewährt hat und im Vergleich zu XML kleine Datenübertragungen nicht so „aufbläht“. Außerdem ist es nahtlos mit JavaScript kompatibel und erleichtert somit den Programmieraufwand, weil auf Serialisierung verzichtet werden kann.

Anforderungsermittlung

Personae

Als weiterer Schritt der Anforderungsermittlung und als Teil der Vorgabe des methodischen Rahmens wurden Personae erstellt, welche möglichst genau Personen beschreiben sollen, welche in Zukunft mit dem fertigen Produkt arbeiten könnten. Es wurden bewusst drei völlig verschiedene Personen erdacht, um ein möglichst vielseitiges Anforderungsbild zu erschaffen. Die Erzeugung fertiger Personae half vor allem bei der Vorstellung und Identifizierung der zukünftigen Benutzer im Cognitive Walkthrough um die genauen Bedürfnisse der Benutzer herausarbeiten zu können.

Stakeholder-Analyse

Vor der Entwicklung der Nutzergruppen, und um ein möglichst umfassendes Bild aller an der Entwicklung sowie Verwendung des Services beteiligten Menschen zu haben, wurde eine Stakeholderanalyse durchgeführt. Während dieser Analyse fiel auf, dass es einen weit größeren Nutzerkreis gab als zunächst angenommen. Und somit ein genaueres herausfinden der Nutzergruppen nötig ist.

Usermodelle

Um Nutzergruppen möglichst vollständig zu identifizieren wurden User Modelle erstellt, welche verschieden und möglichst vollständig Nutzer abbilden soll. Dies hat sehr geholfen die bereits vorgeplanten Teile des Programms zu verschiedenen Nutzern zuzuordnen und zu planen wie für die Nutzer am leichtesten zu verwendenden Oberflächen und Funktionalitäten auszusehen haben. Zusätzlich hat die Analyse der verschiedenen Fähigkeiten sowie der grundlegenden als auch psychischen Voraussetzungen der Nutzer

geholten die genauen Anforderungen der Benutzer zu finden. Außerdem hat es durch Überlegung geholfen das Projekt weiter einzugrenzen und somit den Workload zu schmälern und die Entwicklung eines schwer zu bedienenden Produktes mit viel zu vielen Funktionalitäten zu verhindern. Beispielsweise wurde die Planung eines Parksystems für LKW-Fahrer in der Planung zurückgestellt, da diese keine all zu große Nutzergruppe sind und LKW Parkplätze zumeist bereits über spezielle Navigationsgeräte und Karten organisiert sind.

Anforderungen an das System

Aus den vorhergehenden Artefakten zur Anforderungsermittlung wurden anschließend die Anforderungen der relevanten Stakeholder herausgearbeitet und in funktionalen sowie nicht funktionalen Anforderungen zusammengefasst.

Als wichtigster Anforderungsteil, im Dokument „Anforderungen“ als F10 bezeichnet, stellte sich schnell die Registrierung, sowie der Login heraus. Ein Login fungiert neben anderen relevanten Punkten als zusätzliche Sicherheit und hilft dem Parkplatzsuchenden bei der Suche nach individuellen der eigenen Rolle entsprechenden Parkplätzen. Zusätzlich benötigt man den Login als Grundvoraussetzung für die Kommunikation der einzelnen Parkplatzteilnehmern, um auch hier eine gewisse Sicherheit zu bieten. Beispielsweise ist eine Vorstellung für den späteren Projektverlauf das Nummernschild auf dem eigenen Profil mit einem Foto zu zeigen, damit User sich nicht für jemand anderen ausgeben können, wenn sie beispielsweise einen Parkplatz privat mieten möchten.

F20 die Kommunikation mit anderen Usern wurde als zweitwichtigste Anforderung erarbeitet, als größtes Alleinstellungsmerkmal ist sie nötig um sich auf dem Markt der Parkplatzapps durchzusetzen und die Parkplatzapp als erfolgreiches Projekt zu beenden. In der Anforderungsermittlung haben die potentiellen Nutzer der Parkplatzapp ein Verlangen nach erweiterter Kommunikation. In Zeiten von Social Media, wie Facebook und Instagram wird die zwischenmenschliche Kommunikation auch in Zukunft immer weiter in Richtung des Internets und über Apps geschehen, die Parkplatzapp soll dabei kein Nachzügler sein und sollte deshalb eine Kommunikationsschnittstelle beinhalten. Die Möglichkeit mit anderen Parkenden zu reden ist in vielen Situationen hilfreich. Als Nutzer kann man mit dieser Schnittstelle beispielsweise einen anderen Nutzer erinnern sein Parkticket zu verlängern, ihn darauf hinweisen, dass er eine Einfahrt zuparkt, oder einfach warnen, dass Politessen in der Gegend sind. Ein Problem dieser Schnittstelle bleibt allerdings, dass sie nur zu einhundert Prozent funktioniert, wenn die Parkplatzapp einen breiten Anklang in der Bevölkerung findet und nahezu jeder mit jedem kommunizieren kann. Außerdem muss sich mit dem Thema der Identifizierung von Parkplatzapp und unserer Kommunikationsschnittstellen Benutzern auf den entsprechenden Parkplätzen befasst werden.

Als Kern der Parkplatzapp und somit als nicht verzichtbare Funktionalität wird F30, die Parken-Navigation bewertet. Die Navigation zu freien Parkplätzen in der Nähe oder in einem spezifizierbaren Gebiet ist das Herzstück der Parkplatzapp und von den meisten Nutzern in der Anforderungsermittlung, die sich vorstellen können eine App die sich mit Parkplätzen befasst zu benutzen, gefordert. Es wird benötigt um die Zielsetzung das Parken zu erleichtern zu erreichen.

Als zwar immer noch wichtige, allerdings nicht als unverzichtbar oder überaus wichtige, sondern eher als Nice-To-Have Funktionalitäten wurden ein Parkzeitwecker sowie eine Hilfe beim Wiederfinden des Autos priorisiert. Beide dieser Funktionalitäten (F40 und F50) sind Hilfen nachdem das Auto geparkt wurde.

Das Auto wiederzufinden ist für einige Menschen mit nicht so gutem Orientierungssinn häufig schwierig, also scheint es eine unterschwellige Anforderung zu sein, eben diesen Menschen mit einem kleinen Tool, welches am besten GPS Koordinaten sowie eventuell die Etage im Parkhaus etc. speichern kann zu helfen. Darüber hinaus ist auch eine

Navigation zu Fuß zurück zum Parkplatz realisierbar.

Der Parkzeitwecker (F50) unterstützt den User das benötigte Zeitmanagement bei kostenpflichtigen Parkplätzen einzuhalten. Im Klartext sollte er auf Parkplätzen, bei denen vor der Abfahrt bezahlt werden muss möglichst die aktuelle Standzeit mit den aktuellen Kosten verbinden, und bei Parkplätzen bei denen im Voraus bezahlt werden muss die restliche Parkdauer anzeigen.

Als zusätzliche Anforderung wird die Vermietung von eigenen Parkplätzen als letztes mit in die Anforderungen aufgenommen. Die Vermietung von eigenen Parkplätzen hilft der Parkplatzapp einen neuen Markt zu erschließen, welcher momentan von Apps die sich nur auf dieses Feature beschränken dominiert wird. Eine Zusammenführung von Parkplatzsuche und Parkplatz Vermietung erleichtert die Benutzung durch die User, da die Parkplatzapp alle Features vereint und somit die Verwendung von nur einer App im Gegensatz zu einer App für jedes Feature ermöglicht, was übersichtlicher und leichter zu erlernen ist. Dieses Feature wurde genauso wie der Parkzeitwecker und das wiederfinden des Autos priorisiert, da aufgrund der eher niedrigen Downloadzahlen von Parkplatzvermietungsapps keine all zu große Nachfrage für dieses Feature erwartet wird.

Datenstrukturen

Die Entwicklung der verwendeten Datenstrukturen wurde größtenteils an den gegebenen Daten der Open-Data Quelle Opendata Köln (**siehe <http://www.stadt-koeln.de/externe-dienste/open-data/parking.php>**) orientiert, da diese vorerst die einzige Quelle für Informationen zu Parkplätzen sein wird und somit bestenfalls inhaltlich exakt übernommen werden sollte um möglichst alle relevanten Informationen übernehmen zu können.

Zu den Usern wurde zunächst nur die eindeutige Identifikation über ID sowie Nummernschild geplant und eine Rollenzuweisung um spezielle Parkplätze die nur für bestimmte Gruppen verfügbar sind nur zu entsprechenden Usern zu übertragen; Beispiele hierfür sind u.A. Parkplätze die nur an Frauen weitervermietet werden sollen oder Studentenparkplätze.

Zu den Entitäten Parkplatz und User kommen noch einige zusätzliche Schnittstellen, die Benötigt werden um alle geplanten Funktionen abbilden zu können, darunter fällt die bereits zuvor begründete Google Maps API, die Implementierung des Parkplatzweckers, sowie eine Kommunikationsschnittstelle um asynchrone Kommunikation zu anderen Parkplatztteilnehmern zu ermöglichen. Die Kommunikatons-Schnittstelle wurde an dieser Stelle zunächst mit dem Pub-Sub Tool „Faye“ von JSON geplant, da dieses bereits bekannt unter den Entwicklern war und somit weiterer Workload zur Einarbeitung gespart werden kann. Allerdings wurden an dieser Stelle die grundlegenden Vor- und Nachteile dieses Vorgehens noch nicht ermittelt und somit ist diese Lösung zunächst nur als Übergangslösung zu betrachten.

Webbasierte Anwendungen Modellierung

Im Meilenstein Webbasierte Anwendungen Modellierung, wurden die Ressourcen für die REST Spezifikation noch einmal ausformuliert und mit den entsprechend benötigten Methoden versehen. Die API des Konzeptes wird basierend auf dem REST Prinzip entwickelt, in dessen Rahmen die HTTP-Verben für die Kommunikation auf den Ressourcen stattfinden. Ansonsten kann man mit vielen Fragen auf den Meilenstein Datenstrukturen verweisen.

Evaluation der UI

Die Evaluationsergebnisse des UI Prototyps wurden mit Hilfe der aus dem Mensch Computer Interaktions Modells abgeleiteten Anforderungen sowie direkt an den Personae kontrolliert. Es wurde das Modell der Usability Evaluation gewählt. In der Usability – Evaluation haben wir uns in einzelne Persona versetzt um anhand dessen Anforderungen und Vorlieben den Prototyp zu bewerten. Der Prototyp bestand währenddessen aus Bild Dateien weshalb die Funktionalität nur theoretisch in der Vorstellung berücksichtigt werden.

Kommunikationsziele des Posters

Die Kommunikationsziele haben wir komplett auf unser Alleinstellungsmerkmal ausgelegt und dieses versucht so gut wie möglich zu verkaufen.

Das Parkzeichen soll der Person die dieses Poster sieht vermitteln um welches Thema es sich dreht. Schnell und präzise. Zusätzlich haben wir das Stilmittel als Android-Smartphone mit aufgenommen um direkt klar zu machen, dass es sich um eine mobile Anwendung handelt.

Aussichten des Projekts

Im weiteren Verlauf des Projekts würde die weitere Realisierung der Funktionalitäten angestrebt werden. So würden alle Funktionalitäten wie in der Planungsphase bedacht ineinandergreifen und das weitere Alleinstellungsmerkmal viele Hauptfunktionalitäten zu verbinden realisiert werden. Als nächstes könnte die Opendata eingebunden werden und in Verbindung des Systems Zeitnah aktuelle Daten liefern. Dies liegt der Risikoanalyse zu Grunde, da Aufgrund der mangelnden Kenntnisse der Teammitglieder in der Entwicklung von Android Applikationen Zeit fehlte um diese zum Projektende zu realisieren. Somit traf der Fall ein manche Funktionalitäten weniger zu Berücksichtigen und sich auf das Parkleitsystem zu konzentrieren. Weiter könnte das Projekt auch in Zusammenarbeit mit weiteren Institutionen wachsen und noch weitere und präzisere Daten bekommen.