

FACHHOCHSCHULE KÖLN FAKULTÄT FÜR INFORMATIK UND
INGENIEURWISSENSCHAFTEN

ENTWICKLUNGSPROJEKT INTERAKTIVE SYSTEME

Meilenstein 3

Campus Gummersbach
im Studiengang
Medieninformatik

Betreut von:

Prof. Dr. Kristian Fischer
Prof. Dr. Gerhard Hartmann
Betreuer1
Betreuer2

ausgearbeitet von:

DERYA ERGUEL
SINEM KAYA

8. Mai 2015

Inhaltsverzeichnis

1	Dokumentation der Proof-Of-Concepts	2
1.1	GPS Lokalisierung	2
1.2	Wetterdaten mit der GET-Methode abrufen	4
1.3	Kommunikation über Google Cloud Messaging	6
2	Mensch-Computer-Interaktion	7
2.1	Benutzermodelle	7
2.1.1	Identifizierung der Stakeholder	7
2.1.2	User Profiles	8
2.1.3	Real Person	11
2.1.4	Personae	12
2.1.5	Szenarien	17
2.1.6	Identifizierung der Nutzungskontexte	18
2.2	Benutzungsmodelle	20
2.2.1	Essential Use Cases	20
2.2.2	Concrete Use Cases	22
3	Anforderungen	25
3.1	Analyse der Erfordernis	25
3.1.1	Smartphone mit Android Betriebssystem	25
3.1.2	Funktionale Anforderungen	25
3.1.3	Qualitative Anforderungen	26
3.1.4	Organisatorische Anforderunge	26
4	Literaturverzeichnis	28
5	Projektplan	29

1 Dokumentation der Proof-Of-Concepts

1.1 GPS Lokalisierung

Die Lokalisierung des aktuellen Standorts in der Google Maps Karte ist von grosser Bedeutung für den Erfolg der Applikation. Durch den Erfolg dieses Proof-Of-Concepts soll die Implementierung der Google Maps Schnittstelle nicht mehr als Risiko zählen.

Ziel des Proof-Of-Concepts ist die Erstellung der Google Maps Karte, welches den aktuellen Standort anzeigt.

Die Programmierung erfordert die Registrierung eines API-Schlüssels auf der Google Developers Console (2). Hierfür ist die Anmeldung mit dem Google Konto pflicht. Danach kann das Projekt erstellt werden und der Benutzer erhält den automatisch generierten API-Schlüssel. Dieser Schlüssel muss in der 'google-maps-api.xml' an die passende Stelle hinzugefügt werden.

```
<resources>
  <!--
    TODO: Before you run your application, you need a Google Maps API key.

    To get one, follow this link, follow the directions and press "Create" at the end:
    https://console.developers.google.com/flows/enableapi?apiid=maps_android_backend&keyType=CLIENT_SIDE

    You can also add your credentials to an existing key, using this line:
    DF:78:E1:16:49:32:BD:4B:CB:8E:9F:56:C8:CF:87:CE:65:FE:7F:40;com.example.sinemkaya.myapplication

    Once you have your key (it starts with "AIza"), replace the "google_maps_key"
    string in this file.
  -->
  <string name="google_maps_key" translatable="false" templateMergeStrategy="preserve">
    xxxxxxxxxxxxxxxAPI_SCHLÜSSEL_HIER
  </string>
</resources>
```

Abbildung .1: Datei 'google-maps-api.xml' mit dem generierten API-Schlüssel

Als nächstes werden vier Textviews in der 'activity maps.xml' Datei erstellt, mit jeweils einem Label und einem Textfeld für Breiten- und Längengrad für die GPS-Koordinaten.

```

<TextView
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:id="@+id/lblLatitude"
    android:text="Latitude:"/>

<TextView
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:id="@+id/tvLatitude"/>

<TextView
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:id="@+id/lblLongitude"
    android:text="Longitude:"/>

<TextView
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:id="@+id/tvLongitude"/>

```

Abbildung .2: Ausschnitt aus der Datei 'activity maps.xml'

Um die Positionsdaten beziehen zu können, musste eine Verbindung mit einem Location-Provider hergestellt werden. Dies geschieht in der 'MapsActivity.java'. Erst werden die TextViews anhand der ID's abgerufen, die in der XML-Datei erstellt worden sind. Danach wird der Handle für den LocationManager abgerufen, der über den Code `getSystemService(Context.LOCATION_SERVICE)` eine Verbindung mit dem LocationManager herstellt. Mit `getLastKnownLocation("provider")` wird dem LocationManager eine Anfrage nach dem aktuellen Standort gestellt, dessen Ergebnisse der Breiten- und Längengrad, die TextViews füllen.

```

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_maps);

    //TextViews abrufen
    TextView tvLatitude = (TextView)findViewById(R.id.tvLatitude);
    TextView tvLongitude = (TextView)findViewById(R.id.tvLongitude);

    //Handle für LocationManager abrufen
    LocationManager lm = (LocationManager)
        getSystemService(Context.LOCATION_SERVICE);

    //Mit GPS-Provider verbinden
    Location loc = lm.getLastKnownLocation("gps");

    //TextViews füllen
    tvLatitude.setText(Double.toString(loc.getLatitude()));
    tvLongitude.setText(Double.toString(loc.getLongitude()));
}

```

Abbildung .3: OnCreate-Methode der Datei 'MapsActivity.java'

Da der Emulator keine GPS-Daten enthält, wurde die Ausführung lediglich auf einem mobilen Device durchgeführt.

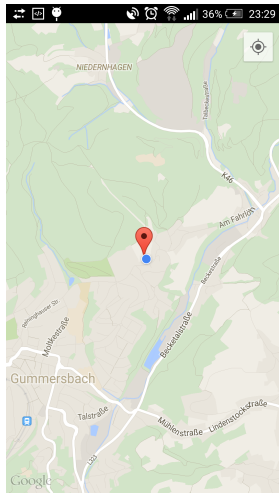


Abbildung .4: Anwendung in Ausführung - GPS-Lokalisierung

Testergebnisse

Die Anwendung sollte an 10 verschiedenen Orten getestet werden, um sicher zu gehen, dass die Präzision der Standorte wahrhaftig sind. Dabei wurden die Standorte bis auf den Straßennamen genau getestet. Die Ergebnisse waren erfolgreich, sodass das Exit-Kriterium - Lokalisierung des aktuellen Standorts - erfüllt worden ist.

Die Implementierung der Google Maps Schnittstelle sollte kein Risiko mehr für die Fortführung des Projekts darstellen.

1.2 Wetterdaten mit der GET-Methode abrufen

Der Abruf der Wetterdaten soll über die openweather Schnittstelle geschehen. Hierfür gibt es drei Such-Alternativen, wie man die aktuellen Daten abrufen kann. Zum Einen kann die Suche über eine Eingabe des erwünschten Ortes oder der Postleitzahl passieren und zum Anderen können die geografischen Koordinaten zur Initialisierung des aktuellen Standorts benutzt werden. Da sich die Benutzer nicht unbedingt immer am selben Standort aufhalten, ist die Alternative der manuellen Eingabe des Ortes sicherlich vorteilhaft.

Es sind zwei Ausgabeformate vorhanden, die von openwether angeboten werden, XML und JSON. Im Architektordiagramm ist die Entscheidung letztendlich auf XML gefallen.

Zunächst wurden fünf TextViews und fünf EditTexts in der 'activity main.xml' erstellt, um die Werte zu Standort, Land, Temperatur Luftfeuchtigkeit und Luftdruck in der XML-Datei ablegen zu können.

Es wird eine zusätzliche Datei 'HandleXML' erstellt, worin ein XML-Parser die Verarbeitung der bereitgestellten Wetterdaten übernimmt. Es ist möglich ein eigenes Programm für das Parsen der XML-Datei zu schreiben, jedoch ist dies recht komplex und ist mit einem großen Zeitaufwand einzuschätzen. Aus dem Grund wurde das XMLPullParser verwendet, da es einfach in der Nutzung ist. Die Verarbeitung der Daten und somit das Parsen geschieht in der Methode 'parseXMLAndStoreIt'. Die 'fetchXML' ist für die Verbindung mit dem Server zuständig, welches die HTTP-Methode GET nutzt, um die Ressource anzufragen und aufzurufen. Hierfür wird die Verbindung zur URL hergestellt und über die setRequest-Methode die GET-Methode aufgerufen.

```
public void fetchXML(){
    Thread thread = new Thread(new Runnable(){
        @Override
        public void run() {

            try {
                URL url = new URL(urlString);
                HttpURLConnection conn = (HttpURLConnection)
                    url.openConnection();
                conn.setReadTimeout(10000 /* milliseconds */);
                conn.setConnectTimeout(15000 /* milliseconds */);
                conn.setRequestMethod("GET");
                conn.setDoInput(true);
                conn.connect();
                InputStream stream = conn.getInputStream();
            }
        }
    });
}
```

Abbildung .5: fetchXML-Methode der Datei 'HandleXML.java'

Das Relevante in der 'MainActivity.java' ist die open-Methode. Dort wird der manuell eingegebene Ortsname in einer URL zusammengesetzt. Anschliessend wird ein neues HandleXML-Objekt erzeugt, welches für das Parsen der XML-Datei zuständig ist und die Ressourcen werden in der Anwendung angezeigt.

```
public void open(View view){
    String url = location.getText().toString();
    String finalUrl = url1 + url + url2;
    country.setText(finalUrl);
    obj = new HandleXML(finalUrl);
    obj.fetchXML();
    while(obj.parsingComplete);
    country.setText(obj.getCountry());
    temperature.setText(obj.getTemperature());
    humidity.setText(obj.getHumidity());
    pressure.setText(obj.getPressure());
}
```

Abbildung .6: open-Methode der Datei 'MainActivity.java'

Die Anwendung wurde durch Eingabe von 20 verschiedenen Städtenamen getestet. Die resultierenden Werte wurden mit den Werten auf der Webseite verglichen. Das Ergebnis erwies sich als positiv. Alle Werte stimmen überein, sodass die Anwendung ihren Zweck erfüllt. Dieses Proof-Of-Concept konnte erfolgreich abgeschlossen werden.

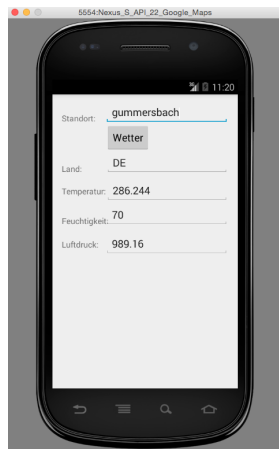


Abbildung .7: Anwendung in Ausführung - Erhalt der Wetterdaten

1.3 Kommunikation über Google Cloud Messaging

Um die Kommunikation der Nutzer untereinander zu gewährleisten soll eine Google Cloud Messaging API eingebunden werden. Dieser Proof-Of-Concepts soll über Google Cloud Messaging realisiert werden. Für diese Realisierung musste zunächst ein Projekt über die Google API Console(5) registriert werden. Nach dem Erstellen eines Projekts bekommt man eine Projektnummer. Diese Nummer braucht man im späteren Verlauf für die Clients. Weiterhin muss man die gewünschte API aktivieren die über **APIs u.auth.->Google Cloud Messaging for Android** zu erreichen ist. Außerdem braucht man einen Server-Schlüssel, der erstellt werden muss über **APIs u. auth -> Zugangsdaten Öffentlicher API-Zugriff->Server Schlüssel erstellen**. Über developer.android.com(6) bekommt man eine ausführliche Anleitung fürs Implementieren von Server und Client. Es wird über OpenSourceFile(7) die Source Dateien bereitgestellt die frei zugänglich sind.

Leider ist das Proof of Concept nicht erfüllt, da die gewünschten Funktionen nicht realisiert werden konnten. Das Problem ist darauf zurück zu führen, das mehrere Fehler aufgetreten sind und diese nicht korrekt aufgehoben werden konnten.

2 Mensch-Computer-Interaktion

2.1 Benutzermodelle

2.1.1 Identifizierung der Stakeholder

Um eine möglichst präzise Benutzermodellierung zu erlangen, soll die Identifizierung von potentiellen Stakeholdern erfasst werden, welches für die menschenzentrierte Gestaltung laut ISO 9241 Teil 210 (3) notwendig ist.

Stakeholder werden als

Einzelperson oder Organisation, die ein Anrecht, einen Anteil, einen Anspruch oder ein Interesse auf ein beziehungsweise an einem System oder an dessen Merkmalen hat, die ihren Erfordernissen und Erwartungen entsprechen (4)

definiert. Dementsprechend werden zunächst die potentiellen Stakeholder aufgelistet, die für den Meilenstein 2 notwendigerweise erfasst werden müssen, um das Kommunikationsmodell qualitativ gestalten und erfassen zu können. Die Benutzermodellierung wird anschließend in Meilenstein 3 fortgesetzt. Nun zu den Stakeholdern:

Reiter

Reiter sollen die primären Benutzer des Systems darstellen. Sie sollen mit dem System direkt interagieren. Die Anwendung soll hauptsächlich zu ihrem Nutzen sein. Sie sollen alle Funktionalitäten des Systems nutzen können, weshalb die Gestaltung besonders zu ihrem Interesse geplant und vorgenommen werden sollte.

Reit-Interessierte

Hierbei handelt es sich um Benutzer, die Interesse an der selben Aktivität teilen. Reit-Interessierte Menschen stellen unsere sekundären Benutzer dar. Sie können die Anwendung zu sonstigen Zwecken nutzen, wie z.B. interessenbedingt als Kommunikationsmittel. Selbstverständlich wird die eigentliche Funktionalität ihrerseits nicht nutzbar sein.

Sponsoren

Durch die Anwendung könnten Sponsoren der Domäne Aufmerksamkeit erreichen. Die Anwendung soll nur auf den Reiter spezialisiert sein. Das bedeutet, dass Sponsoren nichts mit den Funktionalitäten der Anwendung anfangen können. Somit gelten sie als tertiäre Benutzer, die kein Interesse an dem System teilen, sondern als Zweckmäßigkeit nutzen können.

2.1.2 User Profiles

User Profiles stellen Charakterisierungen von Stakeholdern dar. Um die, in Meilenstein 2 identifizierten Stakeholder möglichst genau charakterisieren zu können benötigt man Merkmale. Im nächsten Abschnitt sollen detaillierte Charakterisierung der aufgelisteten Stakeholder gruppiert in User Profiles, anhand von Merkmalen durchgeführt und ihre Ausprägung empirisch ermittelt oder geschätzt werden.

Die Werte, die im Folgenden erfasst wurden, beziehen sich nur auf Deutschland und sind durch Interviews mit Zeynep M. (Reiterin), Watlozs (Reiterin) und den Statistiken der Webseite für Deutsche Reiterliche Vereinigung - Bundesverband für Pferdesport und Pferdezucht (1), zustande gekommen. Sind die Merkmale nicht aussagekräftig genug, können die User Profiles um verschiedene Merkmale ergänzt werden.

Die User Profiles wurden in folgende Gruppierungen aufgesplitten:

- Erfahrene und Profi-Reiter
- Amateur-Reiter
- Anfänger
- Reit-Interessierte
- Sponsoren

Die Begründung für die Aufspaltung der allgemeinen Gruppe Reiter beruht darin, dass Erfahrene- beziehungsweise Profi-Reiter andere Merkmale und Erfahrungen aufweisen wie die Amateur-Reiter oder der Anfänger. Erfahrene Reiter haben ganz andere Bedürfnisse und Anforderungen wie die anderen User Profiles. Die Ergebnisse der Merkmalsausprägungen unterscheiden sich auch dementsprechend. Aus dem Grund ist die Unterteilung der Reiter in Untergruppen zwingend notwendig gewesen, um die Eigenschaften präziser erfassen zu können.

User Profile - Erfahrene und Profi-Reiter

- **demographische Charakteristiken:** circa 8 bis 60 Jahre, häufiger reiten Frauen mehr als Männer
- **Reit-Erfahrungen:** reitet mehr als 15 Jahren regelmäßig in der Woche
- **Turnier-Erfahrungen:** an keinem bis an mehreren Turnieren teilgenommen
- **beherrschte Techniken:** Dressurreiten, Springreiten, Distanzreiten, Voltigieren, Schrittreiten

- **Wissen über die Umwelt:** für das Pferd schädliche Pflanzen
- **Kenntnisse über Ortschaften:** geritten an vielen verschiedenen Orten Deutschland weit und Landschaften
- **Vorkenntnisse über Richtlinien:** mäßig bis gute Vorkenntnisse
- **Wissen über Pferde:** Gesundheitszustand des Pferdes, Ausrüstung
- **Fähigkeiten oder Einschränkungen:** hohe Fähigkeiten, generell keine Einschränkungen
- **Training:** von generell Schritt-Training bis zu mehreren Disziplinen erlernen

User Profile - Amateur-Reiter

- **demographische Charakteristiken:** circa 8 bis 60 Jahre, häufiger reiten Frauen mehr als Männer
- **Reit-Erfahrungen:** reitet mehr als 3 Jahre regelmäßigen Abständen
- **Turnier-Erfahrungen:** an keinem bis an mehreren Turnieren teilgenommen
- **beherrschte Techniken:** Dressurreiten, Springreiten, Distanzreiten, Voltigieren, Schrittreiten
- **Wissen über die Umwelt:** kein Wissen bis gutes Wissen
- **Kenntnisse über Ortschaften:** an noch nicht vielen Orten geritten bis zu an vielen verschiedenen Orten Deutschland-weit geritten
- **Vorkenntnisse über Richtlinien** von keine bis gute Vorkenntnisse
- **Wissen über Pferde:** kennt sich im Umgang mit dem Pferd mäßig bis sehr gut aus (Gesundheitszustand des Pferdes, Ausrüstung, pflege)
- **Fähigkeiten oder Einschränkungen:** hohe Fähigkeiten, generell keine Einschränkungen
- **Training:** von generell Schritt-Training bis zu mehreren Disziplinen erlernen

User Profile - Anfänger

- **demographische Charakteristiken:** circa 8 bis 60 Jahre, häufiger reiten Frauen mehr als Männer
- **Reit-Erfahrungen:** noch nie geritten bis zu 2 Jahre
- **Turnier-Erfahrungen:** keine Turniererfahrungen

- **beherrschte Techniken:** keine oder Schrittreiten
- **Wissen über die Umwelt:** keine Erfahrung bezüglich der Umwelt
- **Kenntnisse über Ortschaften:** wenig bis mäßige Kenntnisse über die Ortschaft
- **Wissen über Pferde:** keine bis mäßiges Wissen
- **Fähigkeiten oder Einschränkungen:** keine bis hohe Fähigkeiten (starker Orientierungssinn), generell keine Einschränkungen
- **Training:** Schritt-Training

User Profile - Reit-Interessierte

- **demographische Charakteristiken:** circa 8 bis 30 Jahre, häufiger sind Mädchen/Frauen interessierter als Jungs/Männer
- **Fähigkeiten oder Einschränkungen:** keine bis hohe Fähigkeiten, generell keine Einschränkungen
- **Wissen über das Reiten:** häufig kein vorhandenes Fachwissen über das Reiten bis auf einige wenige Ausnahmen
- **Wissen über Pferde:**
- **Motivation:** möchte reiten, Gleichgesinnte kennenlernen, Pferde vom Naha betrachten und sich über die Rubrik informieren

User Profile - Sponsoren

- **demographische Charakteristiken:** circa 20 bis 60 Jahre, Frauen und Männer
- **Berufserfahrungen:** überdurchschnittlich
- **formale Qualifikation:** mittlere bis hohe Qualifikation
- **Informationen zum Unternehmen:** kleine bis große Unternehmen
- **Fähigkeiten oder Einschränkungen:** keine bis hohe Fähigkeiten, generell keine Einschränkungen
- **Computer Kenntnisse und -Erfahrungen:** durchschnittlich bis hohe Kenntnisse
- **Fachwissen:** Sponsoren, die in der Branche tätig sind haben ein hohes Fachwissen was das Reitsport und die Pferde angeht. Sponsoren, die nicht in der Branche tätig sind haben generell keinen bis wenig Kenntnisse
- **Motivation:** möchte werben und die Reiter, Gemeinschaften, Vereine fördern

2.1.3 Real Person

Damit das Endergebnis die realen Anforderungen im weiterführend Kapitel standhalten kann, sollten die realen Benutzer im Projekt involviert werden. Diese Benutzer werden reale Benutzer genannt. Die Eigenschaften, Merkmale, Anforderungen und Bedürfnisse dieser Benutzer sind aussagekräftiger für das Endergebnis als die Persona. Anschließend sollen Personae erstellt werden, da nur zwei Interviews mit realen Benutzern geführt worden sind, die aber nicht ausreichen würden, um die Anforderungen adäquat zu erfassen.

Die Real Persons und die Personae gelten als Repräsentanten der User Profiles. Dementsprechend wird die Zuordnung zum jeweiligen User Profile vorgenommen.

Zeynep M. - Erfahrene Reiterin

Zeynep M. ist 26 Jahre alt und Studentin der Uni Köln. Sie studiert im 6. Semester Erziehungswissenschaften und ist fast fertig mit ihrem Studium. Zeynep ist in der Türkei geboren und kam mit 6 Jahren nach Deutschland. Sie kam in einem kleinen Dorf am Rande der türkischen Stadt Maras auf die Welt. Die Eltern besaßen zu der Zeit einen Bauernhof in Mitten einer großen Weide. Sie lebten sehr abgelegen. Zeyneps Vater Ali M. hatte an einer renommierten Uni in Ankara, Landwirtschaft und Agrarwirtschaft studiert. Seinen Traum hatte er mit seiner kleinen Familie erfüllt, in dem er ein Bauernhof betrieb. Auf dem Bauernhof pflegte er mehrere Tiere und hatte mehrere Felder. Zeynep war immer ein aufgewecktes Kind und hat sich beim ersten Anblick in die Pferde verliebt. Ihr Vater hat sie schon mit drei Jahren auf ein kleines Pferd gesetzt und ist mit ihr Runden gelaufen. Mit fünf Jahren konnte sich Zeynep schon selbst auf ein Pferd schwingen und es beim Lauf führen. Die Liebe zu den Pferden erfüllte Zeynep, und sie hatte ein Traum eines Tages irgendwann eigene Pferde zu besitzen. Zeyneps Familie sind Aleviten (eine muslimische Minderheit, die in der Ausübung des Islams, sich von der Mehrheit der Sunniten unterscheidet). 1995 kam es zu Aufständen gegen die Aleviten in mehreren Teilen der Türkei, unter Anderem auch in dem Dorf von Zeyneps Familie. Der Vater hatte Angst um die Sicherheit seiner Familie, verkaufte seinen Bauernhof und somit all seine Tiere. Sie sind in die Großstadt geflohen. Aber auch dort wurden sie nicht in Ruhe gelassen. Daraufhin bekam er die Chance mit seiner Familie nach Deutschland zu fliehen. In Deutschland angekommen hat die Familie ein neues Zuhause bezogen und der Vater hat in einem Familien-Betrieb in der Landwirtschaft ein guten Job erhalten. Zeynep hat zu der Zeit ihren Traum von ihren Pferden nicht losgelassen. Ihr Vater hat ihr zum Geburtstag, über seinen Chef, das lang ersehnte wieder sehen mit Pferden ermöglichen können. Der Besitzer der Pferde hat Zeyneps Leidenschaft zu den Pferden gesehen und Zeynep - war zu der Zeit 8 Jahre alt - angeboten mit den Pferden zu reiten, wenn sie die Pferde ab und an pflegen würde. Seither verbringt Zeynep ihre Freizeit auf dem Hof und reitet fast jeden Tag. Sie nimmt auch an diversen Turnieren teil. Ihr Ziel ist es in paar Jahren ihre eignen Pferde zu besitzen mit denen sie kranken Kindern hilft.

Tina Watlozs - Amateur-Reiter

Tina Watlosz ist 32 Jahre alt geboren in Dormagen. Gebürtige Polin. Ihre Familie stammt aus Thorn eine Stadt in der Nähe der deutschen Grenze. Ihre Eltern kamen nach ihrer Hochzeit nach Deutschland und gründeten hier ihre Familie. Tina hat noch drei Brüder, die älter sind als sie. Tina selbst ist ledig und wohnt in einer kleinen zwei Zimmer-Wohnung in Dormagen. Sie selbst spricht fließend Polnisch und besucht ihre Großeltern wann sie kann. Ihre Großmutter ist Tinas Vorbild. Sie teilen die Leidenschaft zu den Pferden. Die Großmutter Patrizia ist 74 Jahre alt. Sie war in dem Alter von 24 Jahren Künstlerin und trat mit Pferden auf. Sie hat Kunststücke auf Pferden gezeigt und arbeitete 8 Jahre lang in einem Zirkus, bis sie ihren heutigen Mann kennenlernte und sesshaft wurde. Diese Leidenschaft erbte Tina von ihr. Tinas Liebe zu Tieren und ganz besonders zu Pferden, entstand in den Sommerferien in Polen. Jedes Jahr in den Ferien, war sie mit ihrer Familie bei ihren Großeltern. Die Oma brachte die Kinder auf ein Bauernhof, auf dem sie Reitstunden bekamen. Die Brüder hatten nie wirklich Lust zu reiten, außer Tina. Also nahm die Oma irgendwann nur noch Tina mit zum Hof. Tina verbrachte jeden Tag sehr viel Zeit mit den Pferden, bis der Vater ihr dann auch Reitstunden in Deutschland bezahlte. Seither reitet Tina. Tina hatte mit 21 einen Reit-Unfall und konnte 4 Jahre aus Angst nicht mehr auf ein Pferd steigen. Durch Therapien entwickelte Tina sich wieder zu einer selbstbewussten Reiterin. Für sie ist die Sicherheit beim Reiten sehr wichtig. Sie sieht und erkennt vorausschauend, welche Gefahren auf Strecken lauern und verliert niemals die Kontrolle über ihr Pferd. Sie liebt ruhige und einsame Land-Spaziergänge auf ihrem Pferd.

2.1.4 Personae

Personae stellen die Verkörperung von prototypischen Benutzern dar, die die Anwendung als Ergebnis für die individuellen Bedürfnisse nutzen sollen. Die Erfassung der Personae ist von essentieller Bedeutung, da die Aktivitäten, die Perspektive, die Bedürfnisse und Wünsche der Personae den Design-Prozess bereichern.

Zu den Stakeholdern Reiter und Reit-Interessierte werden je zwei Personae erfasst. Zu den Sponsoren soll ein Persona genügen, da es sich um die tertiären Benutzer handelt. Allgemein handelt es sich bei den Personae um fiktive Persönlichkeiten, dessen Eigenschaften bewusst, der Eigenschaften der realen Stakeholder ähneln soll.

Persona: Erfahrene Reiterin

Name: Frida Paul

Alter: 32

Beruf: Grafikerin

Ausbildung: Hochschuleabschluss in Web-Design

Einkommen: 30.000 € im Jahr

Familienstand: ledig

Hobbys: reisen, reiten

Ziel: Ein erfülltes Leben führen

Frida arbeitet in einer Werbeagentur als Grafikerin. Sie lebt alleine in einem kleinen Haus außerhalb der Stadt. Sie reitet seit dem sie 16 ist. In ihrer Freizeit ist sie eine leidenschaftliche Reiterin. Sie reist sehr gerne um die Welt und unternimmt vieles mit Pferden auch im Urlaub. Ihre Reisen plant Frida so, dass sie auf Pferden neue Landschaften entdecken kann. Frida ist Mitglied im Reiter Verein e.V. Dresden und organisiert dort auch viele Spendenaktionen für Pferde. Sie besitzt selber ein Pferd und kümmert sich fürsorglich jeden Tag um die Pflege von ihrem Pferd. Außer der Pflege jeden Tag trainiert Frida ihr Pferd vier mal die Woche zum Distanzreiten. Sie hat ihr Pferd drei Jahre trainiert und traut sich dieses Jahr am Turnier teilzunehmen. Beim Distanzreiten müssen Reiter mit Pferd eine bestimmte Strecke unter einer Zeit zurücklegen dabei wird das Pferd von Tierärzten vor und nach dem Lauf kontrolliert (u.a. Atmung, Puls, Herzschlag) und erst bei keiner Beanstandung oder Probleme kann man erst gewinnen. Dabei geht es Frida um den Spaß und die enge Beziehung zu ihrem Tier.

Persona: Erfahrener Reiter

Name: Michael Berg

Alter: 28

Beruf: Tischler

Ausbildung: Ausbildung zum Tischler-Meister

Einkommen: circa 60.000 € im Jahr

Familienstand: verheiratet, eine Tochter

Hobbys: mit der Familie Zeit verbringen, Pferde

Ziel: die Zufluchtsstätte der Pferd aufrecht halten

Michael Berg ist 28 Jahre alt und wohnt in Hamburg. Er ist verheiratet und hat eine kleine Tochter von 2 Jahren. Michael arbeitet als Tischler im eigenen Familien Unternehmen. Er hat mit acht Jahren angefangen zu reiten und genießt die Zeit mit seinen Pferden. Michael besitzt mit seiner Frau zusammen drei Pferde. Über seine liebe zu Pferden lernte Michael seine Frau kennen, die Tierärztin ist. Michael setzt sich stark für den Tierschutz ein. Mit 26 erbte Michael das Anwesen seiner Eltern auf dieser erfüllt Michael sein Wunsch nach einer Zufluchtsstätte für Pferde die durch schlechte Haltung und Pflege heruntergekommen sind. Er baute einen Stahl und kann durch Spenden und andere Tierliebhaber diese Aufrecht erhalten. Er unternimmt Reitausflüge mit seiner Frau und anderen Reitern durch den Wald. Einmal im Monat besuchen sie Deutschland weit andere Reiterhöfe und setzten sich beide für mehr Sicherheit auf Reitwegen ein.

Persona: Reit-Interessierte

Name: Anke Nadel
Alter: 18
Beruf: Abiturientin
Ausbildung: voraussichtlich Abitur
Einkommen: Taschengeld 200 € pro Monat
Familienstand: ledig
Hobbys: Musik, mit Freunden was unternehmen
Ziel: Reiten lernen

Anke Nadel ist 18 Jahre alt und wohnt in Köln. Sie ist Schülerin der Klasse 12 des Albertus Magnus Gymnasiums in Ehrenfeld. Anke war zu ihrem 18. Geburtstag bei ihren Großeltern eingeladen, die in Königswinter wohnen. An dem Wochenende, wo sie zu Besuch da war, haben die Freunde ihrer Großeltern ein Reitturnier veranstaltet, auf die sie eingeladen waren. Anke kam noch nie so nah in den Kontakt mit Pferden wie an dem Tag auf dem Turnierplatz. Anke liebt Musik und mit ihren Freundinnen einkaufen gehen. Sie hat selbst keine Haustiere. Doch kam sie an dem Tag nicht mehr aus dem Staunen raus. Sie durfte an dem Tag auch unter Begleitung eines Betreuers paar Runden auf dem Pferd reiten. Nach dem Wochenende bei ihren Großeltern hat Anke das erlebte mit ihren Eltern geteilt, die ihr dann zu gesprochen haben, Reitstunden zu bezahlen.

Persona: Sponsoren

Name: Rudolf Grün
Alter: 65
Beruf: Rentner und Sponsor
Ausbildung: Ausbildung zum Sattler
Einkommen: 80.000 € im Jahr
Familienstand: verheiratet, 2 Kinder
Hobbys: reisen
Ziel: weiter als Sponsor im Reitsport tätig sein

Rudolf Grün ist 65 Jahre alt und lebt in München. Er lebt mit seiner Frau und seinen zwei Hunden zusammen im Herzen von München. Bis vor drei Jahren arbeitet Rudolf in seinem eigenen mittelständigen Unternehmen zur Herstellung von Reitersätteln. Seit knapp 30 Jahren engagiert sich Rudolf im Reitsport und unterstützt finanziell angehende Profireiter. Rudolf reist schon sein Leben lang um die Welt aus Leidenschaft und aus beruflichen Gründen. Trotz das Rudolf seit drei Jahren selbst nicht mehr arbeitet engagiert er sich trotzdem noch für Reiter und Pferde. Er ist auf Turnieren und Reiterhöfen ein sehr gern gesehener Gast und gehört bei den meisten zum Sport dazu.

2.1.5 Szenarien

Szenario 1: Frida möchte eine neue Route entdecken

Frida geht wie jeden Sonntag morgens um acht Uhr in den Stall um ihr Pferd zu pflegen. Die Pflege ihres Pferdes nimmt meistens eine Stunde bis zwei Stunden in Anspruch. Erst mal Füttern dann Bürsten und Hufen reinigen. Am Samstag hat sich Frida am Abend spontan für eine neue Strecke entschieden sie möchte diesmal was neues Entdecken. Sie hat am Freitag auf der Arbeit von einem Kollegen der Fotograf ist, erfahren das es einen schönen Aussichtspunkt 30 km entfernt von dem Reiterhof gibt.

Frida hat eine Karte mitgenommen um den Weg zum Aussichtspunkt über geeignete Reitwege zu finden. Nach dem sie ihr Pferd ritt fertig gemacht hat. Reitet Frida los. Sie rechnet mit einer Ankunft von circa 3,5 Stunden am Ziel. Frida hat eine Stoppuhr dabei um die Zeit zu messen. Nach circa 15 km die Hälfte der Strecke, erhöht Frida das Reittempo und misst die Zeit nochmal. Auf dem Weg begegnet Frida Fußgängern und Hundehalter und muss ihr Tempo verringern. Frida hat vorher nicht bedachtet das eine Straße ihr Weg kreuzte, diesen darf sie mit ihrem Pferd nicht betreten, sie musste ein Umweg von weiteren 5 km vornehmen. Am Ziel angekommen merkt Frida das sich nicht mehr genügend Wasser für ihr Pferd hat und muss jeden der ihr entgegen kommt fragen wo sie was zu trinken her bekommen könnte. Ein aufmerksamer Passant schenkt ihr dann seine Wasserflasche. Im Stall wieder angekommen ist Frida erschöpft von dem anstrengenden Ausflug und ist nur noch froh über die heile Ankunft.

Analyse des Szenarios:

Der Nutzer wird im Laufe der Route immer wieder von Hindernissen überrascht und muss dementsprechend die Routenplanung ändern. Diese kann zur Folge haben, dass das Tier und der Reiter überfordert wären und gefährliche Situationen nicht meiden könnten.

Szenario 2: Michael will einen problemlosen Ritt im Wald vornehmen

Michael hatte eine anstrengende Woche in seinem Job. Und möchte an seinem freien Tag mit seinem Pferd durch den Wald spazieren. Michael macht sein Pferd fertig und reitet los. Er kennt alle Wege um sein Anwesen herum und macht sich nicht weiter Gedanken über seine Route. Er plant ein kleines zwischen Stopp und hat deshalb noch Proviant mit. Er kennt alle verbotenen Wege und geht deshalb die gewohnte Strecke von ihm. Nach einer Stunde entscheidet sich Michael spontan die Route zu ändern um neue Teile des Waldes zu entdecken. Er reitet immer mehr in den Wald hinein und merkt nicht wie spät es geworden ist. Als er dann, an einem Fluss das Ende der Strecke erkennt muss er umdrehen. Beim Blick in den Himmel merkt er, dass ein Unwetter aufzieht und es gleich anfängt zu regnen. Michaels Pferd hat panische Angst vor Gewittern und er weiß das er die Strecke nicht rechtzeitig vor dem Gewitter schaffen kann. Also

versucht Michael aus dem Wald zu kommen, um seine Frau anzurufen, die ihn und sein Pferd abholen soll. Michael muss Passanten Fragen wo er ist um seiner Frau sein genauen Standort zu sagen. Michael geht mit seinem Pferd zu einem umliegenden Reiterhof und kommt dort erst mal ins trockene seine Frau war nach zwei Stunden erst da um die zwei abzuholen.

Analyse des Szenarios:

Bei wechselhaftem Wetter ist der Reiter und sein Pferd im Freien ausgeliefert und hat meistens nicht die Möglichkeit bei längeren Entfernungen rechtzeitig Schutz zu bekommen. In den, für den Reiter unbekannten Regionen, gibt es im Wald oder an Straßen wenig Orientierungspunkte für die Umgebung.

Szenario 3: Anke möchte reiten lernen

Anke hat sich im Internet mehrere Reiterhöfe in der Umgebung rausgesucht um Reitstunden zu bekommen. Dabei traut Sie sich nicht wirklich alleine Reitstunden zu nehmen. Da Anke sehr schüchtern ist. Sie fragt ihre beste Freundin Stefanie, der Sie ihre Faszination zu Pferden erzählt hat, ob sie nicht gerne mit ihr Reitstunden nehmen möchte. Ihre beste Freundin teilt das Interesse und möchte auch reiten lernen. Die Mädchen setzten sich dann nochmal gemeinsam hin und suchen im Internet nach geeigneten Reiterhöfen. Nach längerem hin und her entscheiden Sie sich für eine Reitgruppen für Anfänger, melden sich vorab für eine Probestunde an. Gemeinsam gehen die Mädchen zur ersten Reitstunde und lernen dort neue Leute kennen und kommen allmählich aus sich raus.

Analyse des Szenarios:

Wenn Anfänger und Reit-Interessierte Informationen einholen möchten. Werden Sie von mehreren Informationsquellen überflutet. Diese vielen Quellen müssen sie vergleichen und ihren Bedürfnissen nach einordnen, dies nimmt sehr viel Zeit und Motivation in Anspruch.

2.1.6 Identifizierung der Nutzungskontexte

Die Identifizierung und Erstellung von Nutzungskontexten dient zur Ermittlung von Anforderungen der Benutzer. Zuvor sollen die User-Needs erarbeitet werden. Diese ergeben sich aus den erzielten Ergebnissen der User-Profiles, den Personae, der Szenarien und den Use Cases.

Reiter:

- Pferd pflegen
- Pferd trainieren
- neue Ortschaften und Routen entdecken
- neue Freunde mit demselben Interesse kennenlernen
- neue Rittarten erlernen

Reiter-Interessierte:

- Reiten erlernen
- soziale Kontakte knüpfen
- mit Gleichgesinnten Unternehmungen durchführen
- Informationen erlangen

Sponsoren:

- für das eigene Unternehmen werben
- die Domäne unterstützen und fördern

Die Nutzungskontexte werden nun tabellarisch anhand der Aufgaben, des physischen und sozialen Umfelds und den Arbeitsmittel der Stakeholder ermittelt.

Stakeholder	Aufgaben	physisches und soziales Umfeld	Arbeitsmittel
Reiter	Pferd pflegen und trainieren, Reitrouuten planen und aufzeichnen, neue Ortschaften entdecken, Informationen mit anderen Benutzern teilen	Reithöfe, Bauernhöfe, landesrechtlich erlaubte Straßen und Landwege	Reiter-Ausstattung
Reit-Interessierte	Informationen sammeln, mit Benutzer kommunizieren und Kontakt aufnehmen	Unterwegs, auf der Arbeit oder Zuhause	..
Sponsoren	Finanzierung, Versicherung, Support,	Unternehmen	Auto, Verträge

Tabelle .1: Identifizierung der Nutzungskontexte

2.2 Benutzungsmodelle

Die Benutzungsmodellierung ist wichtig für die Gebrauchstauglichkeit des Systems, da der Design hauptsächlich auf der Benutzungsmodellierung aufbaut. Es besteht die Möglichkeit, dass das System ein hervorragendes User Interface bietet. Jedoch nützt dies nicht viel, wenn das System den Aufgaben des Anwenders nicht gerecht wird. Deshalb ist die Benutzungsschnittstelle ein wesentlicher Faktor für die prototypische Umsetzung des Endprodukts.

Es sind verschiedene Alternativ-Methoden zur Arbeitsermittlung vorhanden. Mit den Task Scenarios werden narrative Beschreibungen der Benutzer-Aufgaben erfasst. Jedoch ist dies für den Verlauf des Projekts nicht von grosser Bedeutung, da das System keine all zu komplexe Benutzung aufweist. Die Karten werden automatisch geöffnet und die Wetterdaten sind durch eine einfache Eingabe des Ortes aufrufbar.

2.2.1 Essential Use Cases

Der Essential Use Case nach Constantine und Lockwood analysiert auf der einen Seite die abstrakte Sicht des Benutzers und auf der Anderen Seite die Forderung des Systems an den Benutzer. Die erstellten Essential Use Cases werden im weiteren Verlauf mit Concrete Use Cases präzisiert. Die Concrete Use Cases beschreiben die Aufgaben der Benutzer mit dem System. Das wichtige ist hierbei die Präzision der Beschreibung. Um ein benutzergerechtes Endsystem zu liefern, die die Anforderungen des Benutzers erfüllt, ist das Essential Use Case und das Concrete Use Case eine optimale Wahl zur Arbeitsermittlung.

User Intention	System requirements
Benutzer möchte Route erstellen	System erfordert das Starten der Karte; System erfordert eine textuelle Beschreibung zu der Route und zu eventuellen Barrieren, Gefahren an
Benutzer möchte Route finden	System erfordert das Starten der Karte; System erfordert die Eingabe des erwünschten Ortes; System erfordert eine textuelle Beschreibung zu der Route und zu eventuellen Barrieren, Gefahren an
Benutzer möchte Route durch manuelle Eingabe des Ortes finden	System fordert das Starten der Karte; System fordert die Eingabe des erwünschten Ortes;
Benutzer möchte Route teilen	System erfordert das Starten der Karte; System fordert die Wahl der aufgezeichneten Route; System erfordert eine Beschreibung zur Route;
Benutzer möchte aktuelle Wetterdaten auf dem Smartphone erhalten	System fordert die Aktivierung der PUSH-Benachrichtigung
Benutzer möchte aktuelle Wetterdaten im System aufrufen	System fordert das Starten der Wetter-Oberfläche; System fordert die Eingabe des erwünschten Ortes
Benutzer möchte einem anderen Benutzer schreiben	System fordert die Wahl einer Route; System fordert einen Aufruf des Benutzerprofils über die Route
Benutzer möchte Umgebungsinformationen auf der Karte abrufen	System erfordert das Starten der Karte; System erfordert die Wahl der angezeigten Informationen

Tabelle .2: Essential Use Case

2.2.2 Concrete Use Cases

Im folgenden werden die im Essential Use Case erstellten Aufgaben in Concrete Use Cases unterteilt. Dadurch sollen die einzelnen Benutzer-Aufgaben präzisiert werden.

Benutzer möchte Route erstellen

User action	System response
Benutzer wählt die Option Karte	System öffnet die Karte und zeigt den aktuellen Standort
Benutzer wählt die Option 'Route starten'	System beginnt die Route aufzuzeichnen
Benutzer beendet die Aufzeichnung der Route	Fenster erscheint; System fordert die Eingabe eines textuellen Berichts in Form von Beschreibung, Gefahren, Ebene

Tabelle .3: Concrete Use Case für 'Benutzer möchte Route erstellen'

Benutzer möchte Route finden

User action	System response
Benutzer öffnet Karte	System öffnet die Karte
Benutzer gibt den Namen der erwünschten Stadt im Eingabefeld ein	Karte wird aktualisiert und der erwünschte Ort wird angezeigt
Benutzer wählt die erwünschte Route	System zeigt die Informationen zu der Route an

Tabelle .4: Concrete Use Case für 'Benutzer möchte Route finden'

Benutzer möchte Route durch manuelle Eingabe des Ortes finden

User action	System response
Benutzer wählt die Option Karte	System öffnet die Karte und zeigt den aktuellen Standort
Benutzer gibt den Namen der gewünschten Stadt im Eingabefeld	die Karte aktualisiert und findet den gewünschten Ort
Benutzer wählt die Option 'Route starten'	System beginnt die Route aufzuzeichnen
Benutzer beendet die Aufzeichnung der Route	Fenster erscheint; System fordert die Eingabe eines textuellen Berichts in Form von Beschreibung, Gefahren, Ebene

Tabelle .5: Concrete Use Case für 'Benutzer möchte Route durch manuelle Eingabe des Ortes finden'

Benutzer möchte Route teilen

User action	System response
Benutzer ruft über den Reiter 'Profil' seinen Profil auf	System öffnet das Profil
Benutzer wählt die Route	System öffnet die Karte und zeigt Route an
Benutzer wählt die Route auf der Karte	System öffnet die Informationen (Beschreibung, Gefahren, Ebene)
Benutzer wählt 'teilen'	System macht die Route für andere Benutzer öffentlich

Tabelle .6: Concrete Use Case für 'Benutzer möchte Route teilen'

Benutzer möchte aktuelle Wetterdaten im System aufrufen

User action	System response
Benutzer ruft Wetter unter dem Reiter 'Wetter' auf	System öffnet das Wetter-Fenster
Benutzer gibt den erwünschten Ort in das Textfeld ein und klickt auf den Button 'Wetter'	System ruft die Daten über die URL auf und zeigt es an

Tabelle .7: Concrete Use Case für 'Benutzer möchte aktuelle Wetterdaten im System aufrufen'

Benutzer möchte einem anderen Benutzer schreiben

User action	System response
Benutzer wählt das Profil	System zeigt das erwünschte Profil an
Benutzer klickt auf 'Nachricht'	System öffnet ein Fenster mit den Eingabefeldern für Überschrift und Text
Benutzer schickt die Nachricht über den Button 'Senden' ab	System leitet die Nachricht weiter an den anderen Benutzer

Tabelle .8: Concrete Use Case für 'Benutzer möchte einem anderen Benutzer schreiben'

Benutzer möchte Umgebungsinformationen auf der Karte abrufen

User action	System response
Benutzer tippt die angezeigten Informationspunkte auf der Karte an	Das System zeigt die Artikel in einem neuen Fenster im voll Text an und lässt die Route im Hintergrund laufen
Der Benutzer kann über ein Pfeil nach rechts weitere Artikel lesen	Das System stellt mehrere Artikel hintereinander zur Verfügung
Der Benutzer möchte die Ansicht verlassen	Das System schließt das Fenster und öffnet die Karte.

Tabelle .9: Concrete Use Case für 'Benutzer möchte Umgebungsinformationen auf der Karte abrufen'

3 Anforderungen

Analyse der Anforderungen

Im folgendem Abschnitt wird anhand aus der Analyse des Nutzungskontext resultierten Anforderungen der Nutzer an die Anwendung, die funktionalen, quantitativen und organisatorischen Anforderungen ermittelt.

3.1 Analyse der Erfordernis

3.1.1 Smartphone mit Android Betriebssystem

Um die App nutzen zu können, benötigt der Nutzer einen Smartphone mit lauffähigem Android Betriebssystem. Da es für die jetzige Entwicklung, den Rahmen der Veranstaltung sprengen würde, wurde die Entwicklung eingeschränkt. Im Ausblick könnte die Realisierung auch für unterschiedliche Endgeräte(u.A. Tablets)und Betriebssysteme wie IOS oder Windows festgehalten werden.

3.1.2 Funktionale Anforderungen

F10 Standort erfassen

- F10.1 automatische GPS erfassen
- F10.2 manuelle Standort Erfassung

F20 Route erstellen

- F20.1. Ziel eingeben
 - F20.1.1. Route starten
 - F20.1.1.1 Route ändern
 - F20.1.1.2 Route löschen
 - F20.1.1.3 Route teilen
 - F20.1.2 Route beenden

F30 Informationen lesen

- F30.1 Anzeige auf der Karte an tippen
- F30.1.2 Artikel wird angezeigt
- F30.1.3 Weiter Artikel lesen
- F30.1.4 Artikel verlassen

F40 Wetter Anzeige

- F40.1 Automatische Wetter Anzeige
- F40.1.1. Aktualisierung der Wetterdaten

F50 Nachrichten an andere Nutzer senden/erhalten

- F50.1 Profil über die Karte aufrufen
- F50.1.1 Nachricht verfassen
- F50.1.1.2 Nachricht senden
- F50.1.2 Nachricht erhalten
- F50.1.2.1 Nachricht löschen
- F50.2 Profil verlassen

3.1.3 Qualitative Anforderungen

- Q10 Präzise Standort Erfassung
- Q20 Übersichtliche Karte
- Q30 Aktuelle Wetterdaten
- Q40 Schnelle Erfassung und anzeige der Umgebung
- Q50 Geringer Akkuleistung
- Q60 Verständliche Darstellung von Informationen
- Q70 Datensicherheit und Datenschutz
- Q80 Leichte Kommunikation zu anderen Nutzern

3.1.4 Organisatorische Anforderunge

- O.1 Genaue Wegbeschreibungen
- O.2 Überblick der Umgebung
- O.3 Notfall Hilfe
- O.4 Kommunikation zu anderen Nutzern

Abbildungsverzeichnis

.1	Datei 'google-maps-api.xml' mit dem generierten API-Schlüssel .	2
.2	Ausschnitt aus der Datei 'activity maps.xml'	3
.3	OnCreate-Methode der Datei 'MapsActivity.java'	3
.4	Anwendung in Ausführung - GPS-Lokalisierung	4
.5	fetchXML-Methode der Datei 'HandleXML.java'	5
.6	open-Methode der Datei 'MainActivity.java'	5
.7	Anwendung in Ausführung - Erhalt der Wetterdaten	6
.8	Projektplan	29

Tabellenverzeichnis

.1	Identifizierung der Nutzungskontexte	19
.2	Essential Use Case	21
.3	Concrete Use Case für 'Benutzer möchte Route erstellen'	22
.4	Concrete Use Case für 'Benutzer möchte Route finden'	22
.5	Concrete Use Case für 'Benutzer möchte Route durch manuelle Eingabe des Ortes finden'	23
.6	Concrete Use Case für 'Benutzer möchte Route teilen'	23
.7	Concrete Use Case für 'Benutzer möchte aktuelle Wetterdaten im System aufrufen'	24
.8	Concrete Use Case für 'Benutzer möchte einem anderen Benutzer schreiben'	24
.9	Concrete Use Case für 'Benutzer möchte Umgebungsinformatio- nen auf der Karte abrufen'	24

4 Literaturverzeichnis

- [1] <http://www.pferd-aktuell.de/fn-service/zahlen--fakten/zahlen--fakten> - Sichtungsdatum: 03.05.2015
- [2] <https://code.google.com/apis/console> - Sichtungsdatum: 05.05.2015
- [3] Norm, Ergonomie der Mensch-System-Interaktion - Teil 210: Prozess zur Gestaltung gebrauchstauglicher interaktiver Systeme (ISO 9241-210:2010), Stand Januar 2011 http://www.iso.org/iso/home/store/catalogue_tc/catalogue_detail.htm?csnumber=52075 - Sichtungsdatum: 22.04.2015
- [4] German UPA e.V., Arbeitskreis Qualitätsstandards, German UPA Qualitätsstandard für Usability Engineering, April 2012, Seite 77 http://www.germanupa.de/data/mediapool/n070_qualitaetsstandard_der_german_upa.pdf - Sichtungsdatum: 22.04.2015
- [5] <https://console.developers.google.com/> - Sichtungsdatum: 21.04.2015
- [6] <http://developer.android.com/google/gcm/index.html> - Sichtungsdatum: 25.04.2015
- [7] <https://github.com/google/gcm> - Sichtungsdatum: 28.04.2015

5 Projektplan

Datum / KW	Aktivität	1. Unteraktivität	2. Unteraktivität	Workload geplant /	Workload gesamt	Workload Derya Erguc	Workload tatsächlich Sinem Kaya
14	Exposé	Ideenfindung	Brainstorming	3		3	3
	Dokumentaufbau	Layout / Struktur	Latex	3		0	4
	Projektplan			2			1,5
					6	3	7
13.04.2015	Meilenstein 1	Nutzungsproblem		1		1	1
		Zielsetzung		1		1	1
		Verteilte Anwendungslogik		1		1	1
		Wirtschaftliche und gesellschaftliche Relevanz		1		1	1
						4	4
27.04.2015	Meilenstein 2	Zielhierarchie	Strategische Ziele	1		2	2
			Taktische Ziele	1		2	2
			Operative Ziele	1		1	1,5
		related-works	ReiterApp	1		0	1
			Cavalio-Retcoach	1		1	0
			sonstige	1		0	1
		Ableistungsmerkmale		1		0,5	1
		Methodischer Rahmen (MCI)		15		8	9
		Kommunikationsmodell		5		8	3
		Risiken		3		2	2
		Spezifikation der POCs		1		1	3
		Architekturdiagramm/ Architekturbegründung	Architekturdiagramm	3		2	2
			Architekturbegründung	3		0	4
					MS2 gesamt IST:	27,5	31,5
	Meilenstein 3	Dokumentation der POC's		5		1	3
		Benutzermodelle		10		6	4
		Benutzungsmodelle		6		1	4
		Anforderungen		5		3	0
					MS3 gesamt IST:	11	11
	Meilenstein 4	Datenstrukturen		5			
		WBA-Modellierung		15			
		Prototypen UI		15			
					MS4 gesamt IST:	0	0
	Meilenstein 5	funktionale Prototypen		20			
		Evaluationsergebnisse UI		10			
		narratives Konzept für firmische Präsentation		6			
					MS5 gesamt IST:	0	0
	Meilenstein 6	Prozessassessment		10			
		Fazit		4			
		Installationsdokumentation		3			
					MS6 gesamt IST:	0	0
		Programmierung GESAMT		150		15	28
			insgesamt Soll:	319	insgesamt Ist:	80,5	81,5

Abbildung .8: Projektplan