

Review-Dokumentation CarDuinoDroid



Inhaltsverzeichnis

	Seite
1. Einleitung	3
2. Phase 1	4
a. Probleme und Lösungen	5
3. Phase 2	6
a. Grobkonzept	6
b. Zeitplan	7
c. Frontend	8
d. Backend	9

Einleitung

Um einen möglichst einfachen und schnellen Überblick über unser Softwareprojekt zu bekommen und das entstandene Projekt nutzen zu können, erstellen wir ein Dokumentationsdokument. Neben diesem Dokument, befindet sich auch im programmierten Code Kommentare, um diesen möglichst einfach nachvollziehen zu können. Dies ist auf der einen Seite notwendig um unser Projekt leichter bewertbar zu machen, andererseits hilfreich für uns selber, sowie für die kommenden Änderungen des Carduinodroid. Weiterhin werden auch die einzelnen Phasen unseres Softwareprojektes dokumentiert und ausformuliert. Dies dient dazu, um Erfahrungen zu sammeln, die wir für kommende Projekte nutzen können, sowie um Fehler besser zu erkennen und um Erwartungswerte im Nachhinein überdenken zu können.

Phase 1 - Planungsphase

- die Wahl und die Beschreibung des Vorgehens und des Projektplans

Da wir an einem vorhandenem und funktionierendem Projekt weiterarbeiten und dieses weiterentwickeln sollen, haben wir uns gemeinsam auf das agile Vorgehen geeinigt.

Dieses zeichnet sich dadurch aus, das wir agil, also beweglich in unserem Projekt sind. Dies ist besonders deswegen notwendig, da wir bis jetzt keinerlei Erfahrung mit Softwareentwicklung gemacht haben und somit nur schwer Arbeitsaufwand von Teilprojekten abschätzen können.

Außerdem macht es uns das agile Vorgehen möglich, die ganze Gruppe in das Projekt einzubeziehen und den Lern- und Erfahrungswert zu steigern.

Allerdings soll der Quellcode ausführlich dokumentiert werden und die Testphase einen großen Teil des Projektes einnehmen.

Der Projektplan wurde nur sehr klein gehalten, da uns noch die nötigen Informationen für den Ablauf fehlten und durch das agile Vorgehen wurde er auch mit niedriger Priorität eingestuft.

- ein zu erstellendes Pflichtenheft

Unser Pflichtenheft wurde von uns gemeinsam erarbeitet und mit Hilfe von Skype und GoogleDocs erstellt und später in LaTeX überführt.

- die Festlegung der verwendeten Softwareentwicklungswerkzeuge

Ideen: Java , Google Web Toolkit, GitHub, LaTeX, Eclipse

- die Teamorganisation

Keine Teamaufteilung, da auch noch keine Teilprojekte vorhanden waren.

Auf die nächste Phase geschoben.

Ergebnisse vom Treffen mit dem Tutor:

Überdenkung der Werkzeugs: Google-Web-Toolkit

Übermäßige Serverauslastung → nicht unsere Aufgabe

Hintergrundclient wird benötigt (Möglichkeiten via PHP, Java, C...)

Änderungen am Pflichtenheft:

- Sensor-Aktor-Interaktion wird zuviel

- Streamingtechnik fällt weg → Einzelbildübertragung funktioniert wunderbar

- Wunschkriterium: Möglichkeit den Stream zu drehen

Agile Vorgehen → ersten Prototypen zur 2. Phase

Sonstiges:

Visualisierung der gedrückten Tasten im GUI ist die Bestätigung, dass das Handy den Befehl empfangen hat.

IP wird per Hand eingegeben - nicht automatisch

Probleme und Lösungen

Probleme:

- auf dem Server muss ständig ein Programm laufen. Dies ist nicht mit einem standard Webserver möglich, da er nur anfragen beantwortet und dann schläft.
- Dynamische IP des Carduinodroids
- Sven Köhler zeigt sehr geringe Beteiligung an der Planung und Entwicklung des Projektes.

Lösungen:

- eigenständiges Programm, welchem im Hintergurnd läuft (daemon) und die kommunikation übernimmt.
- Sven Köhler hat freiwillig am 15.05.2013 das Team und die Arbeit am Projekt verlassen.

Phase 2 - Grobkonzept

Datenbankentwurf

Wir haben uns für eine MySQL Datenbank entschieden, aufgrund der einfachen Nutzung und standardisierten und einfachen Sprache. Als Auswahl stand noch Hibernate zur Entscheidung, doch wir haben uns dagegen entschieden, weil die Möglichkeit Objekte zu speichern nicht von uns benötigt wird. Im folgenden ist unser relationales Datenbankschema zu sehen, sowie den grafischen Datenbankentwurf.

User = {userID, nickname, password, rightFlag }

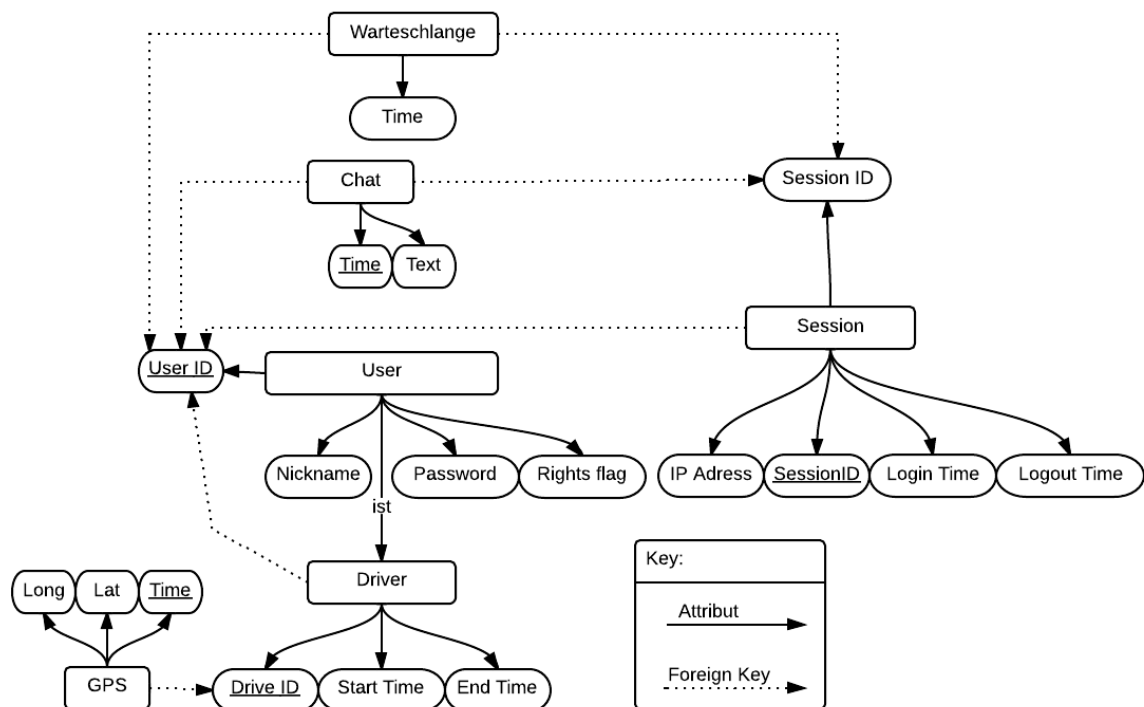
Driver = {driveID, userID->User, startTime, endTime }

Session= {sessionID, userID -> User, loginTime, logoutTime, IP }

GPS = {time, driveID -> Driver, long, lat }

Chat = {userID->User, Time, text, sessionID->sessionID }

Queue = {queueID, userID->User, sessionID->Session, time }



Zeitplan

	Woche 1	Woche 2	Woche 3	Woche 4	Woche 5
Alex	IDE / Controller	Kamera	Kamera	Admin-bereich	Testen
Christoph	GUI Javascript	GUI Javascript	Chat Bot	Chat Client	Debugging
Jenja	GUI Design	GUI Implementierung	Java Adminbereich	*Bedarf*	Debugging
Marc	Informations-sammlung	Quellcode-doku	Quellcode-doku	Doku-formatierung	Kontrolle
Michael	DB-Initialisierung	Queue	Queue	Chat Server	Codecleanup
SvenW	Kommunikation Server-Android	Komm. Server-Android	Auto-steuerung	*Bedarf*	Debugging
Vincenz	IDE / DB Verbindung	DB Verwaltung (Admin)	DB Tests / Optimierung	Frontend Finalisierung	Testen

Frontend

Das Frontend besteht im wesentlichen aus 2 Komponenten. Der Landing Page und der Hauptseite.

Die Landing Page besteht aus dem Login für die registrierten User sowie den Admin worüber sie sich authentifizieren.

GUI des registrierten Nutzers:

GUI sollte Kamera auf Vollbild setzten können

GUI sollte sich an das Browserfenstergröße anpassen

GUI soll Lenkwinkel und Geschwindigkeitsregler enthalten

- Links: Chat

- Mitte: Kamerabild

- Rechts: Warteschlange

 - unten Links: Lenkwinkel, Geschwindigkeitsregler, Auflösung

 - unten Mitte: Steuerung

 - unten Rechts: Licht, Hupe

GUI des Admins:

GUI sollte Kamera auf Vollbild setzten können

GUI sollte sich an das Browserfenstergröße anpassen

- Links: neuen User erstellen

- Rechts: Globale Einstellungen

GUI des Gastes

GUI sollte Kamera auf Vollbild setzten können

GUI sollte sich an das Browserfenstergröße anpassen

- Links: Chat

- Mitte: Kamera Bild

Backend

Im Backend soll der Admin Einstellungen vornehmen können, welche in eine INI-Datei gespeichert werden. Inhalt dieser INI-Datei soll die maximale Fahrtdauer, die Verbindungsdaten zur Datenbank, sowie das Zeitintervall zur Speicherung der GPS Daten. Außerdem wird im Hintergrund ständig ein Daemon laufen, der die dauerhafte Verbindung zwischen dem Server und dem Android Smartphone aufrecht erhält.

Im Backend auf Seite des Users gibt es hauptsächlich Loggingfunktionen, wie das Logging des Chats, der Fahrtzeiten und den Sessions. In diese kann der Admin Einsicht nehmen und er hat die Möglichkeit User zu erstellen, sowie globale Einstellungen vorzunehmen, die in der oben genannten INI-Datei gespeichert werden. Sämtliche Funktionen werden in Java implementiert.