

Konzeption und Implementierung eines Schwarmverhaltens von mobilen Kleinrobotern anhand eines Verfolgungsszenarios

STUDIENARBEIT

für die Prüfung zum
Bachelor of Science
des Studiengangs Informatik
Studienrichtung Angewandte Informatik
an der
Dualen Hochschule Baden-Württemberg Karlsruhe

16. März 2017

Bearbeitungszeitraum	24 Wochen	
Name	Manuel Bothner	Simon Lang
Matrikelnummer	8359139	6794837
Kurs	TINF14B2	TINF14B2
Ausbildungsfirma	1&1 Internet SE	ifm ecomatic GmbH
	Brauerstr. 48	Im Heidach 18
	76135 Karlsruhe	88079 Kressbronn am Bodensee
Betreuer	Prof. Hans-Jörg Haubner	
Gutachter	Prof. Dr. Heinrich Braun	

Erklärung

(gemäß §5(3) der „Studien- und Prüfungsordnung DHBW Technik“ vom 29. 9. 2015)

Ich versichere hiermit, dass ich die Studienarbeit meiner Studienarbeit mit dem Thema: „Konzeption und Implementierung eines Sachwarmverhaltens von mobilen Kleinrobotern anhand eines Verfolgungsszenarios“ selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe. Ich versichere zudem, dass die eingereichte elektronische Fassung mit der gedruckten Fassung übereinstimmt.

Ort, Datum

Unterschrift

Ort, Datum

Unterschrift

Abstract

Zusammenfassung

Inhaltsverzeichnis

1	Einleitung	11
1.1	Ausgangslage	11
1.2	Zielsetzung	11
1.3	Erwartetes Ergebnis	11
2	Technische Grundlagen	12
2.1	Robotik	12
2.1.1	Grundlagen	12
2.1.2	Mobile Roboter	12
2.1.3	Antriebsarten	12
2.1.4	Sensorik	12
2.1.5	LEGO Mindstorm	12
2.2	App Entwicklung	13
2.2.1	Arten	14
2.2.2	Plattformübergreifende Programmierung	14
2.2.3	Mono	14
2.2.4	.Net Framework	14
2.3	Java	14
2.3.1	Grundlagen	14
2.3.2	Java Runtime Environment	14
2.4	Kommunikation	14
2.4.1	Grundlagen	14
2.4.2	TCP/IP	14
2.4.3	Wifi	14
2.4.4	Datenaustausch	14
2.5	Schwarmverhalten	14
2.5.1	Allgemein	14
2.5.2	Vorbilder aus dem Tierreich	14
2.5.3	Szenarien	14
2.5.4	Algorithmen	14
3	Projektorganisation	15
3.1	Projektablaufplan	15
4	Konzeption	16
4.1	Anforderungsdefinitionen	16
4.1.1	Softwarearchitektur	16
4.2	Use Cases	16

4.2.1	Connect	16
4.2.2	Synchronization	17
4.2.3	Szenario	20
4.2.4	Exception	20
4.3	Kommunikation	20
5	Lösungsansatz	26
6	Umsetzung	27
7	Evaluation	28
8	Zusammenfassung und Ausblick	29

Abbildungsverzeichnis

1	Connect	16
2	Connection	17
3	Synchronization	18
4	Robot list	19
5	Spectator	20
6	Spectator	21
7	Control	22
8	Control	23
9	Synchron	24
10	Follow	25

Tabellenverzeichnis

1 Einleitung

Heutzutage werden viele Arbeitsschritte in der Produktion, als auch Dienstleistungen von Maschinen verrichtet, da diese effizienter Arbeiten und weniger Kosten als Menschen verursachen. Da jede Maschine auf einen spezifischen Arbeitsschritt konfiguriert ist, müssen die verschiedenen Maschinen untereinander wie ein Schwarm agieren. Diese Verhaltensstrukturen kommen ursprünglich aus dem Tierreich, wie Fischeschwärme, Ameisen oder Bienen. Hierbei erledigt jedes Individuum seine zugewiesenen Aufgaben und hält die anderen Parteien auf dem aktuellen Stand.

In diesem Projekt werden diese Verhaltensmuster aus dem Tierreich aufgegriffen und anhand eines Verhaltensszenarios mit Kleinrobotern verwirklicht, die autonom agieren und kommunizieren, um zusammen ihr Ziel zu erreichen. Dabei sollen Konzepte, sowie Algorithmen für Schwarmroboter entstehen, die auch auf andere Szenarien angewendet werden können.

1.1 Ausgangslage

1.2 Zielsetzung

1.3 Erwartetes Ergebnis

2 Technische Grundlagen

2.1 Robotik

2.1.1 Grundlagen

2.1.2 Mobile Roboter

2.1.3 Antriebsarten

2.1.4 Sensorik

2.1.5 LEGO Mindstorm

2.2 App Entwicklung

Eine App, kurz für Application ist ein ausführbares Programm für mobile Geräte, wie Smartphones oder Tablets. Um eine App für ein Endgerät zu entwickeln, müssen im Voraus Zielpunkte abgeklärt werden, da je nach Art der Entwicklung andere Voraussetzungen und Möglichkeiten bestehen. Allgemein kennt die App Entwicklung drei verschiedene Arten, die native, web und hybride Entwicklung.

In der nativen Entwicklung werden die direkten Ressourcen des Gerätes verwendet. Dazu gehört die Laufzeitumgebung, Bibliotheken, Geräteschnittstellen und andere APIs. Der Vorteil hierbei ist, dass diese Entwicklungen für Betriebssystem optimiert sind, da sie auf die vorhandenen Schnittstellen aufbauen können und daher komplexere und rechenintensivere Anwendungen ermöglichen.(Zitat) die

Ein Beispiel ist hierfür eine native Android Entwicklung durch Java und Googles Android Studio, wobei die direkten Ressourcen von Android genutzt und somit auch die vorhandenen UI Elemente und Schnittstellen zur Verwendung kommen, die Android Nutzer gewohnt sind.

Die web Entwicklung arbeitet hingegen zur nativen Entwicklung mit systemübergreifenden Ressourcen und greift dabei auf gängige Webtechnologien, wie HTML, CSS und Javascript zurück. Die App wird hierbei nicht wie normale Anwendungen direkt auf dem System des Gerätes ausgeführt, sondern kommt in dessen Browser zur Ausführung.

2.2.1 Arten

Web Apps

Native Apps

Hybride Apps

2.2.2 Plattformübergreifende Programmierung

Xamarin

2.2.3 Mono

2.2.4 .Net Framework

2.3 Java

2.3.1 Grundlagen

2.3.2 Java Runtime Environment

2.4 Kommunikation

2.4.1 Grundlagen

2.4.2 TCP/IP

2.4.3 Wifi

2.4.4 Datenaustausch

2.5 Schwarmverhalten

2.5.1 Allgemein

2.5.2 Vorbilder aus dem Tierreich

2.5.3 Szenarien

2.5.4 Algorithmen

3 Projektorganisation

3.1 Projektablaufplan

4 Konzeption

In diesem Kapitel werden die Anforderungsdefinitionen des Projektes, mit Spezialisierung auf die verschiedenen Use Cases beschrieben.

4.1 Anforderungsdefinitionen

In diesem Abschnitt wird auf die Funktionalitäten und Use Cases des Projektes eingegangen.

4.1.1 Softwarearchitektur

4.2 Use Cases

4.2.1 Connect

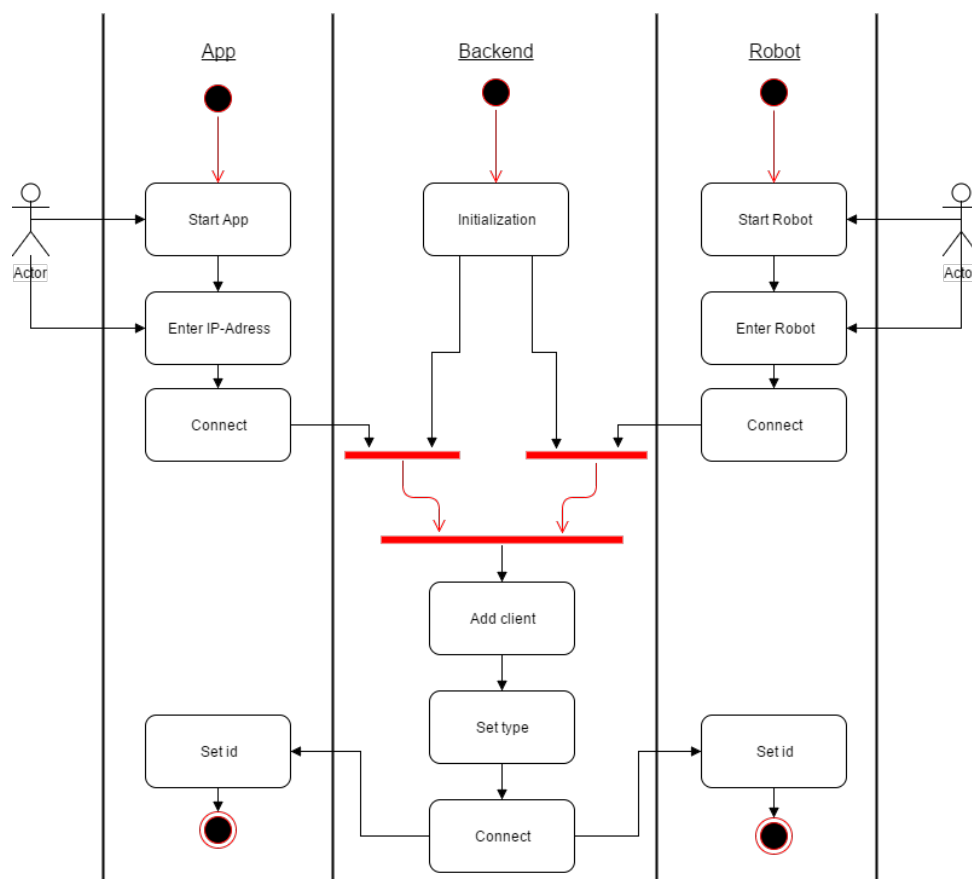


Abbildung 1: Connect

Im Use Case Connect wird eine erste Verbindung durch die Eingabe der IP-Adresse zum Backend aufgebaut. Dabei sendet die Komponente, ob Roboter oder App eine Abbildung seiner selbst als Objekt dem Backend. Daraufhin startet das Backend die Verbindung indem es der Komponente entsprechende Verbindungskommandos zusendet. Sobald eine

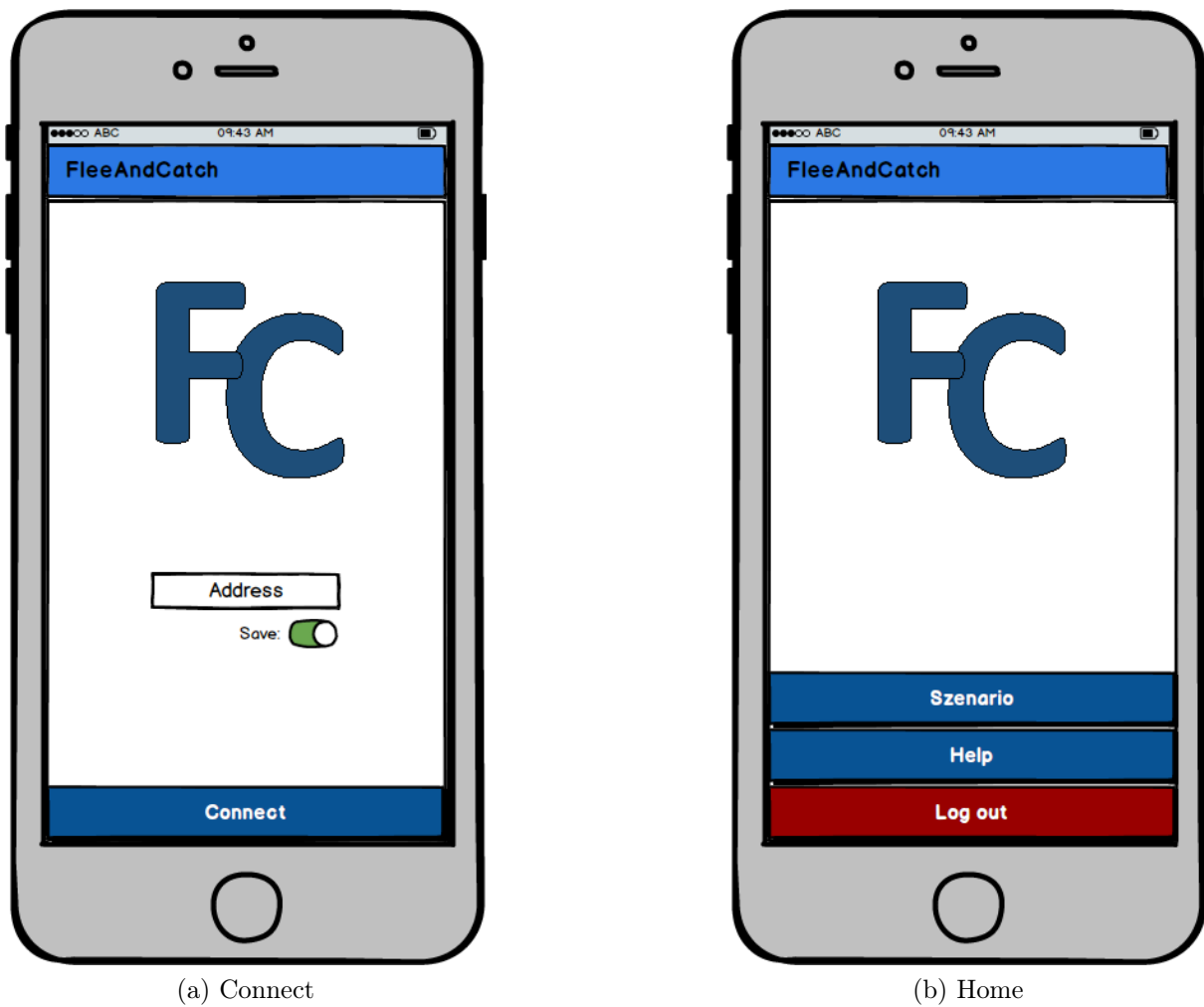


Abbildung 2: Connection

Reaktion in einer festgelegten Zeit erfolgt, akzeptiert das Backend die Verbindung und sendet die entsprechende Id für die Komponente. Ab diesem Moment ist die Komponente verbunden und ein Robot für Aktionen entsprechend verfügbar. Die Verbindungsinitialisierung dient hierbei der Verkürzung der Reaktionszeit, die bei einem Roboter sonst entsprechend hoch wäre.

4.2.2 Synchronization

Im Use Case Synchronization werden Daten entsprechend des gesetzten Typen zwischen den Komponenten übertragen. Dabei können einerseits die Roboter als Objekte, oder ganze Szenarien übertragen werden. Dies dient zur Gegenseitigen Synchronisierung der Daten.

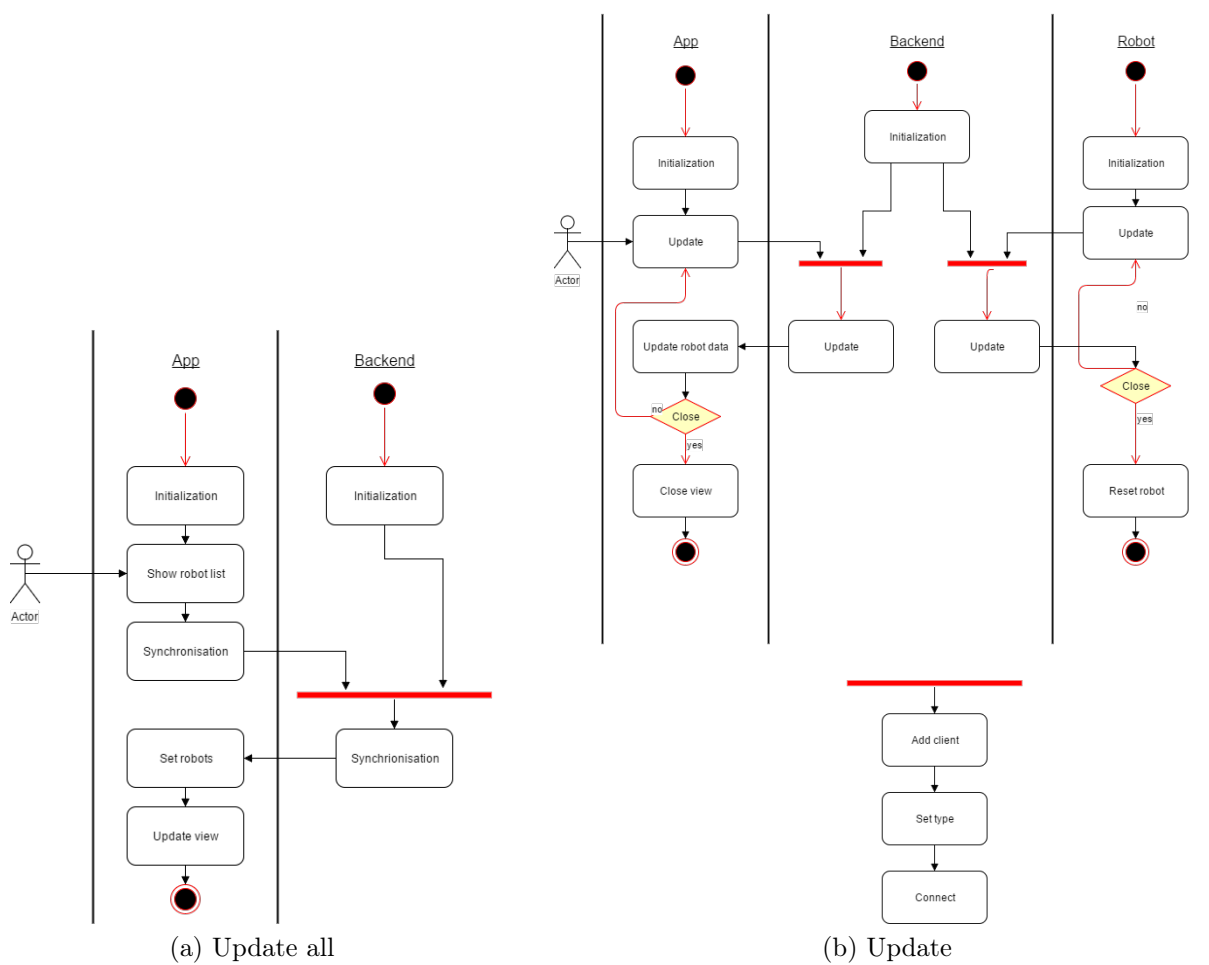


Abbildung 3: Synchronization

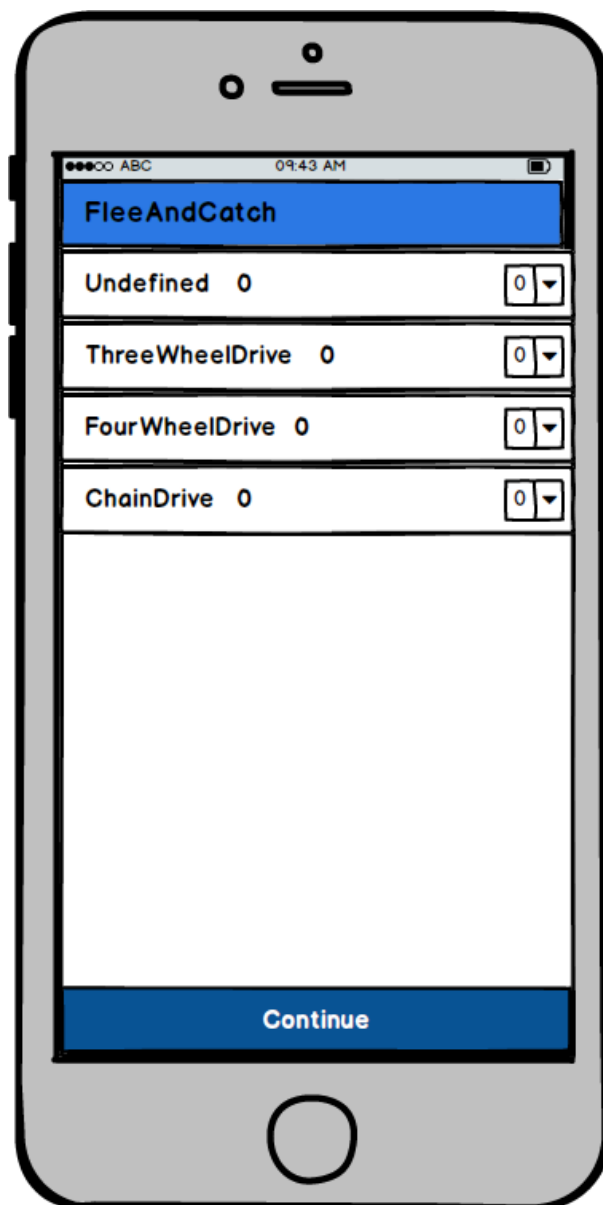


Abbildung 4: Robot list

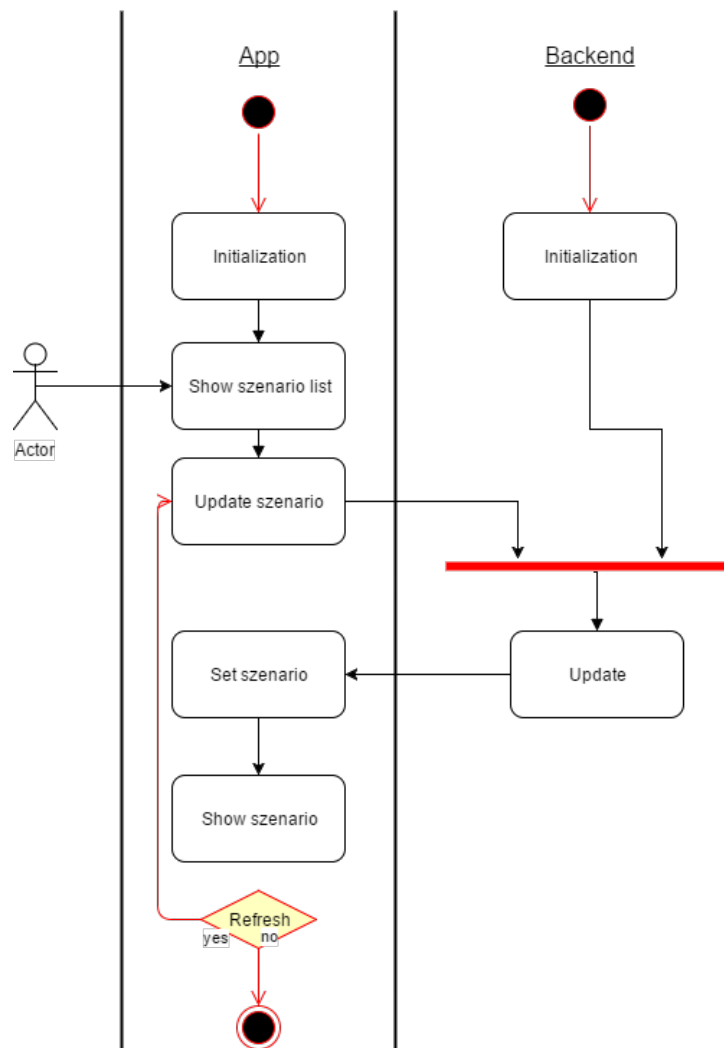


Abbildung 5: Spectator

4.2.3 Szenario

4.2.4 Exception

4.3 Kommunikation

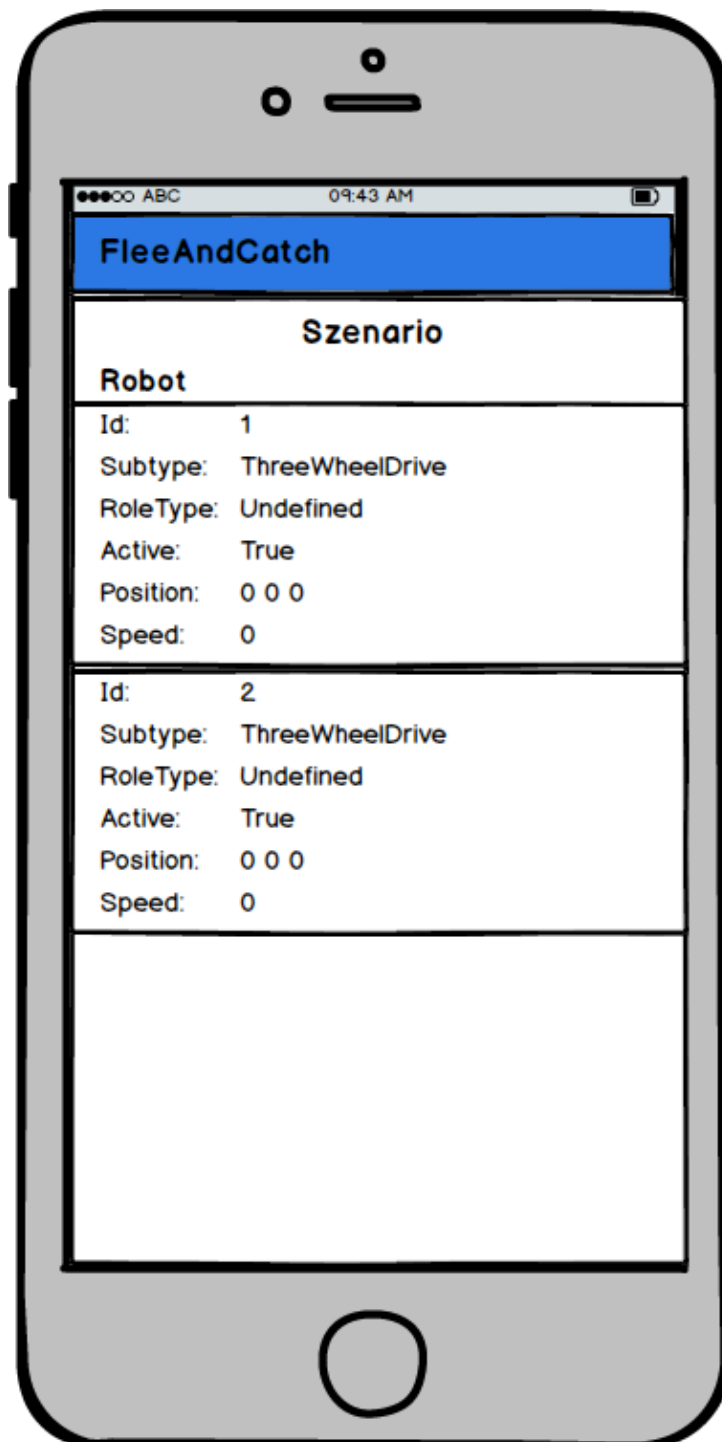


Abbildung 6: Spectator

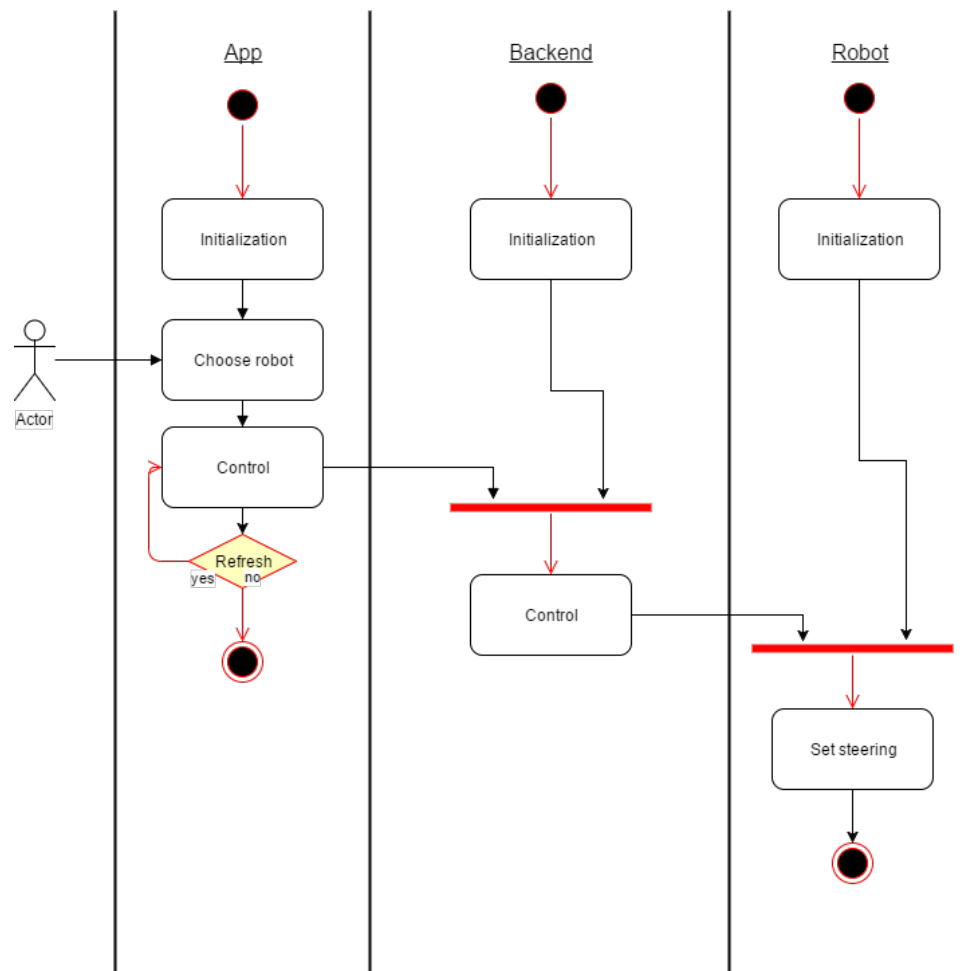


Abbildung 7: Control

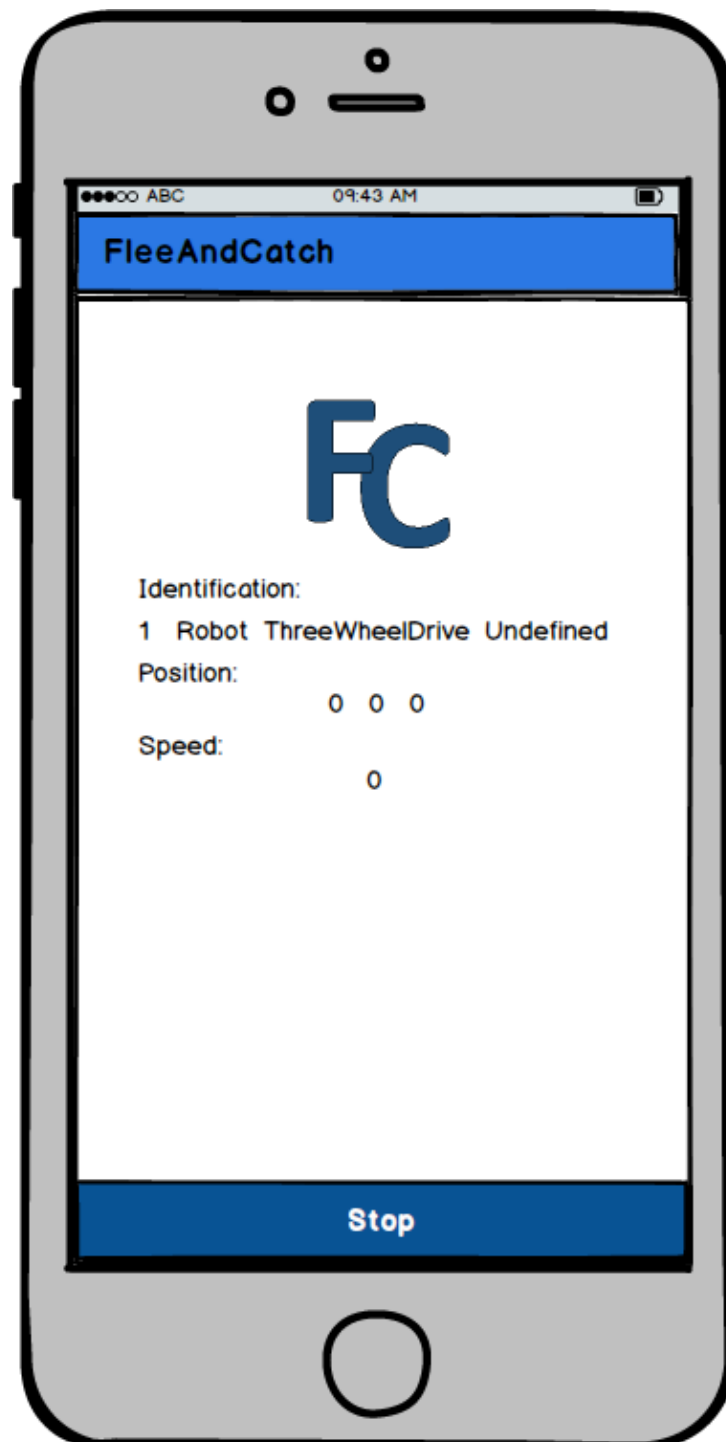


Abbildung 8: Control

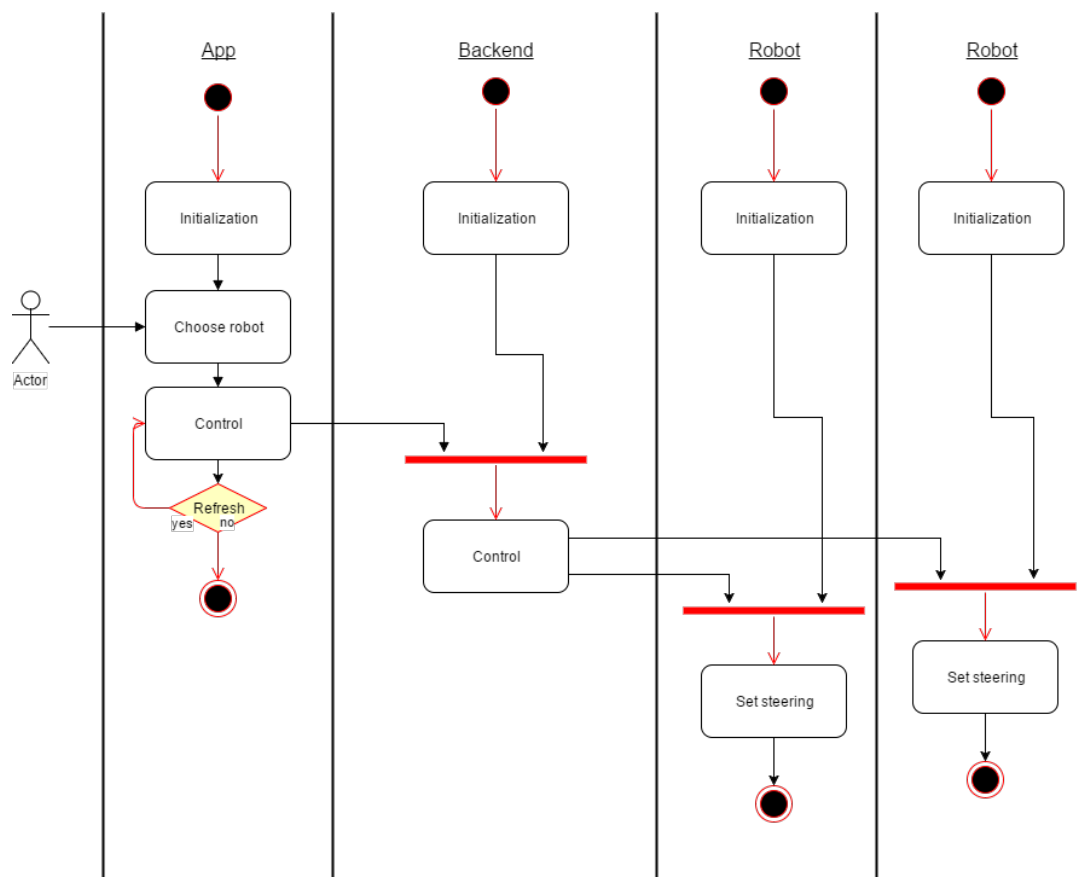


Abbildung 9: Synchron

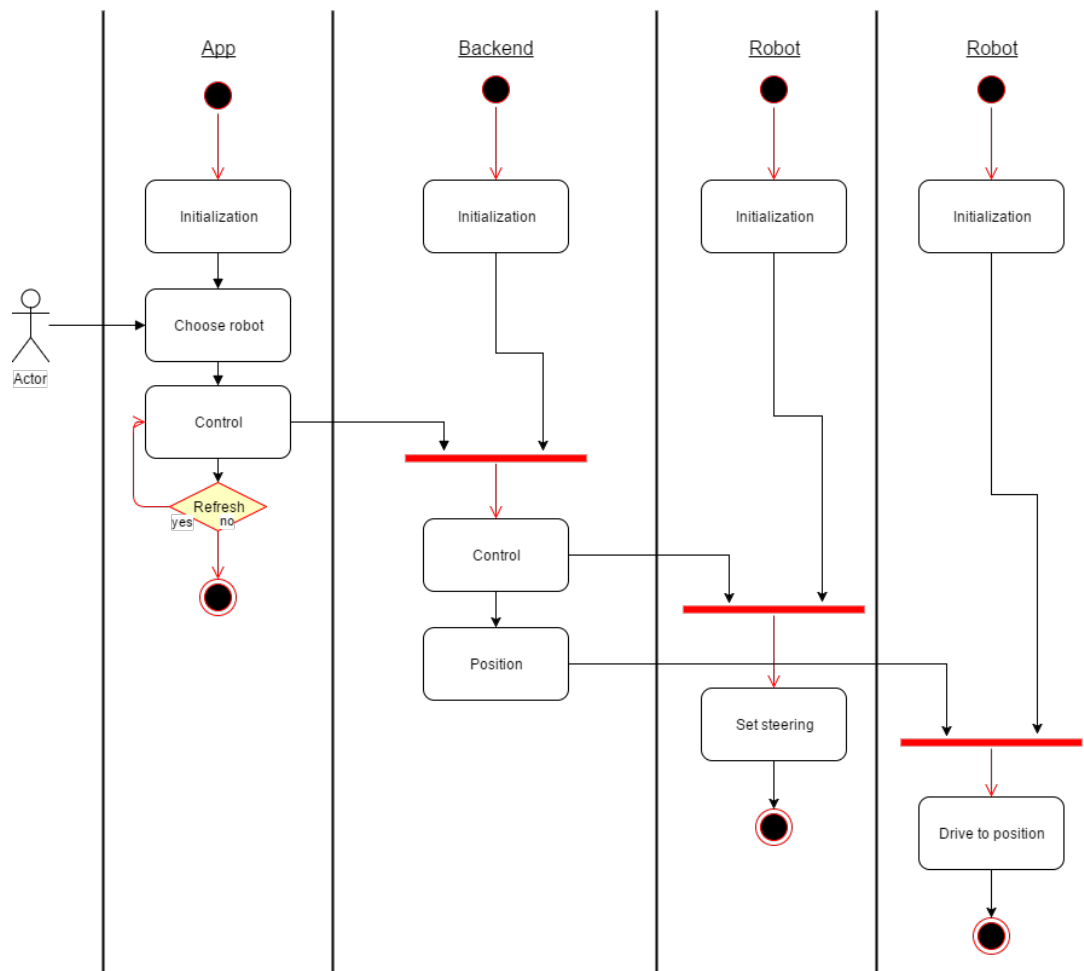


Abbildung 10: Follow

5 Lösungsansatz

6 Umsetzung

7 Evaluation

8 Zusammenfassung und Ausblick

Literatur

Anhang