

# Follow-Me Fahrroboter

## ***Aktualisierte Anforderungsspezifikation***

Taylan Özer s85500@htw-dresden.de, Claire Schubert s85609@htw-dresden.de,  
Johann Marthaus s85567@htw-dresden.de, Jonathan Palermo s85287@htw-  
dresden.de, Danny Miersch s85567@htw-dresden.de, Jeremy Ziegert  
s85299@htw-dresden.de, Luca Kurth s85449@htw-dresden.de

28. Juni 2024

# Inhaltsverzeichnis

1. Vision: Follow-Me Fahrroboter .....	1
1.1. Einführung .....	1
1.2. Positionierung .....	1
1.3. Stakeholder Beschreibungen .....	2
1.4. Produkt-/Lösungsüberblick .....	3
1.5. Zusätzliche Produktanforderungen .....	4
2. Use-Case Model: Follow-Me Fahrroboter .....	5
2.1. Allgemeine Informationen .....	5
2.2. Identifizierte Use Cases .....	5
2.3. Use Case Diagramm .....	5
3. System-Wide Requirements: Follow-Me Fahrroboter .....	12
3.1. Einführung .....	12
3.2. Qualitätsanforderungen für das Gesamtsystem .....	12
3.3. Zusätzliche Anforderungen .....	13
4. Glossar: Follow-Me Fahrroboter .....	14
4.1. Einführung .....	14
4.2. Begriffe .....	14
4.3. Abkürzungen und Akronyme .....	19
4.4. Verzeichnis der Datenstrukturen .....	20
5. Domain Model: Follow-Me Fahrroboter .....	24

# 1. Vision: Follow-Me Fahrroboter

## 1.1. Einführung

Der Zweck dieses Dokuments ist es, den wesentlichen Bedarf und Funktionalität des Systems Follow-Me zu sammeln, zu analysieren und zu definieren. Der Fokus liegt auf den Fähigkeiten, die von Stakeholdern und adressierten Nutzern benötigt werden, und der Begründung dieses Bedarfs. Die Details, wie das System Follow-Me Fahrroboter diesen Bedarf erfüllt, werden in den Use Cases und der Supplementary-Specification beschrieben.

### 1.1.1. Zweck

Der Zweck dieses Dokuments ist es, die wesentlichen Anforderungen an das System aus Sicht und mit den Begriffen der künftigen Anwender zu beschreiben.

### 1.1.2. Gültigkeitsbereich (Scope)

Dieses Visions-Dokument bezieht sich auf das System Follow-Me Fahrroboter, das vom I01\_Follow-Me-Team entwickelt wird. Das System wird es Herr Prof. Dr.-Ing. Neugebauer erlauben, den Fahrroboter ihm folgen zu lassen mittels der eigens entwickelten Bilderkennungssoftware durch das o. g. Team. Die Hauptaufgabe besteht darin, demjenigen zu folgen, welcher visuell die Führung angibt.

### 1.1.3. Definitionen, Akronyme und Abkürzungen

Siehe [Glossar](#).

## 1.2. Positionierung

### 1.2.1. Fachliche Motivation

Die fachliche Motivation für das Projekt "Follow-Me Fahrroboter" liegt in der Synthese von zwei essenziellen Technologiebereichen: Robotik und Bilderkennung bzw. der nötigen Bilderkennungssoftware. Durch die Konstruktion eines Roboters auf Basis eines Raspberry Pi und der Integration einer von uns als Studiengruppe eigens entwickelten Bilderkennungssoftware streben wir an, ein effizientes und präzises System zu schaffen, das in der Lage ist, einer Person in einem Raum zu verfolgen.

Die Herausforderung besteht für uns im Allgemeinen noch darin, ein robustes und zuverlässiges System zu entwickeln, welches trotz der limitierten Hardware performant genug ist. Dies erfordert eine präzise Abstimmung zwischen der Hardware des Fahrroboters und der Software für die Bilderkennung sowie die Implementierung eigens entwickelter Algorithmen zur Personenerkennung.

Insgesamt stellt das Follow-Me-Fahrroboter-Projekt eine Machbarkeitsstudie für unseren Auftraggeber Herr Prof. Neugebauer dar und bietet uns als Gruppe von sieben Studierenden eine spannende Möglichkeit, uns auf dem Gebiet der Robotik und der Bilderkennung heranzutasten.

### 1.2.2. Problem Statement

Das Problem	Als Machbarkeitsstudie soll gezeigt werden, dass der Fahrroboter - basierend auf einem Raspberry Pi mit eigens entwickelter Bilderkennungssoftware - selbstständig in der Lage ist einer Person in einem Raum zu folgen.
betrifft	Prof. Dr.-Ing. Neugebauer HTW Dresden
die Auswirkung davon ist	Es wurde gezeigt, dass der Roboter in der Lage ist, einer Person in einem Raum zu folgen.
eine erfolgreiche Lösung wäre	Die Realisierung des Fahrroboters mittels einer Bilderkennungssoftware auf einem Raspberry-Pi und einem Roboter.

### 1.2.3. Positionierung des Produkts

Für	Prof. Dr.-Ing. Neugebauer
die	Als Machbarkeitsstudie einen Fahrroboter benötigen, welcher in der Lage ist, einer Person in einem Raum zu folgen.
die Lösung ist ein	Fahrroboter, basierend auf ROS II
Das	Herr Prof. Dr.-Ing. Neugebauer zeigt, dass mittels Bilderkennungssoftware der Roboter der Person folgt.
Im Gegensatz zu	bisherigen Lösungen haben wir einen neuartigen Ansatz gewählt, welcher eigens entwickelt wird.
Unser Produkt	Fahrroboter begleitet mittels Bilderkennungssoftware sein Ziel.

## 1.3. Stakeholder Beschreibungen

### 1.3.1. Zusammenfassung der Stakeholder

Name	Beschreibung	Verantwortlichkeiten
Prof. Dr.-Ing. Neugebauer	Als Auftraggeber möchte er eine Machbarkeitsstudie durchführen, dabei soll festgestellt werden, inwiefern es möglich ist, mit leistungsschwacher Hardware einer mittleren Bilderkennung Person zu folgen.	Herr Prof. Neugebauer legt den Rahmen und die Bedingungen an dieses Projekt fest und überwacht dieses demnach auch. Herr Prof. Neugebauer stellte auch den Fahrroboter Alphabot 2 und den Raspberry-Pi plus die hierfür benötigte Kamera zur Verfügung.
Benutzer	Person, welche das System benutzt	keine
Administrator	Person, welche das System einrichtet und verwaltet	keine

Name	Beschreibung	Verantwortlichkeiten
DSGVO	Im Allgemeinen ist hier ebenfalls der Schutz der personenbezogenen Daten zu betrachten. Der Alphabot 2 nimmt kontinuierlich Personen mit der hierfür geeigneten Kamera auf. Jene Personen sind sich vielleicht auch der Tatsache nicht bewusst, dass sie aktuell aufgenommen werden und so ist dies mit zu berücksichtigen. Demnach ist dies behutsam zu handhaben, sofern sich Personen im Blickfeld der Kamera befinden oder in Aufnahmen, die getätigt worden sind, ohne das Einverständnis der jeweiligen Personen.	Schutz der personenbezogenen Daten.
Entwickler	Streben die Entwicklung einer eigenen Software zum folgen von Personen an	Legen die Umsetzung und die Möglichkeiten in Verbindung mit Software und Hardware des Fahrroboters zusammen.

### 1.3.2. Benutzerumgebung

1. Anzahl der Personen, die an der Erfüllung der Aufgabe beteiligt sind. Ändert sich das?
  - 8 Personen; Mit Abschluss des 3. Semesters noch 7 Personen
2. Wie lange dauert die Bearbeitung der Aufgabe? Wie viel Zeit wird für jeden Arbeitsschritt benötigt? Ändert sich das?
  - Jede Iteration bzw. jeder Sprint ist auf 2 Wochen ausgelegt; Jeder Arbeitsschritt ist i. d. R. innerhalb von maximal zwei Tagen zu erledigen;
3. Gibt es besondere Umgebungsbedingungen, z.B. mobil, offline, Außeneinsatz, Touchbedienung, Nutzung durch seh- oder hörbeeinträchtigte Personen?
  - Es gibt keine besonderen Umgebungsbedingungen
4. Welche Systemplattformen werden heute eingesetzt? Welche sind es ggf. zukünftig?
  - Es wird Microsofts Visual Studio Code und Jet Brains PyCharm als allgemeine Entwicklungsumgebung genutzt und GitHub als Versionsverwaltungssoftware.
5. Welche anderen Anwendungen sind im Einsatz? Muss ihre Anwendung mit diesen integriert werden?
  - Das OS ROS II ist im Einsatz und muss mit dem Roboter integriert werden.

## 1.4. Produkt-/Lösungsüberblick

### 1.4.1. Bedarf und Hauptfunktionen

Bedarf	Priorität	Features	Geplantes Release
Optimierung Bilderkennung	hoch	Erkennung einer Person mittels eines performanten Bilderkennungsalgorithmus Face Detection	9. Iteration
erster Prototyp	hoch	Erkennung einer Person	5. Iteration
Aufsetzen des OS ROS II	hoch	Das System dient der allgemeinen Struktur für das Projekt	2. Iteration
Konstruktion des Alphabot 2	hoch	Plattform ist der Raspberry Pi mit einem Alphabot 2	1. Iteration

## 1.5. Zusätzliche Produktanforderungen

Anforderung	Priorität	Geplantes Release
Erkennung des Abstandes zu einer Person im Bild und Verfolgung dieser Person	hoch	-
einwandfreie bzw. einfache Nutzung	hoch	-

## 2. Use-Case Model: Follow-Me Fahrroboter

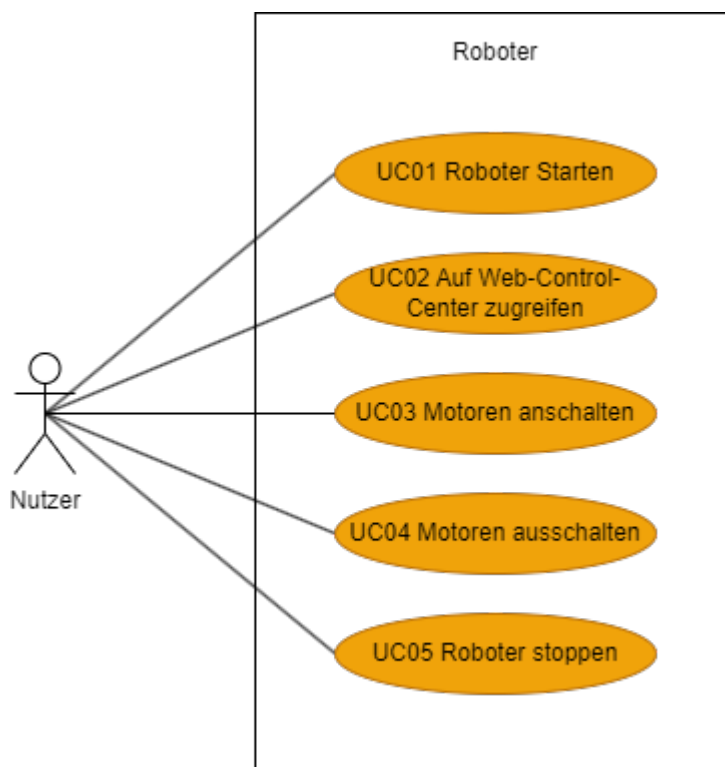
### 2.1. Allgemeine Informationen

Im nachfolgenden Abschnitt werden alle identifizierten Use Cases des Follow-Me-Fahrroboters beschrieben. Entscheidungen für etwaige Use Cases wurden retrospektiv im Entwicklungsprozess aufgrund offen definierter Anforderungen getroffen.

### 2.2. Identifizierte Use Cases

- **UC01:** Roboter starten
- **UC02:** Auf Web-Control-Center zugreifen
- **UC03:** Motoren anschalten
- **UC04:** Motoren ausschalten
- **UC05:** Roboter stoppen

### 2.3. Use Case Diagramm



## 2.3.1. Use-Case: UC01 Roboter starten

### Kurzbeschreibung

Der Benutzer schaltet den Roboter ein und startet die Software zur Personenverfolgung.

### Kurzbeschreibung der Akteure

- Benutzer

### Vorbedingungen

- Der Roboter ist ausgeschaltet und nicht mit dem WLAN verbunden.

### Standardablauf (Basic Flow)

1. Der Use Case beginnt, wenn der Benutzer den Roboter mit Strom versorgt.
2. Der Benutzer verbindet sich mit dem WLAN des Roboters.
3. Der Benutzer verbindet sich über SSH mit dem Roboter.
4. Der Benutzer führt die Startbefehle aus.
5. Die Software zur Personenverfolgung wird gestartet.

### Alternative Abläufe

#### Alternativer Ablauf 1: WLAN-Verbindung schlägt fehl

1. Der Benutzer überprüft die WLAN-Verbindung.
2. Der Benutzer stellt sicher, dass das WLAN-Netzwerk verfügbar ist.
  - Bei Scheitern des Verbindungsversuchs startet der Benutzer den Roboter erneut und folgt der Anwenderdokumentation.
  - Bei erneutem Scheitern wendet sich der Benutzer an den Systemadministrator.

#### Alternativer Ablauf 2: SSH-Verbindung schlägt fehl

1. Der Benutzer überprüft die Netzwerkverbindung.
2. Der Benutzer stellt sicher, dass die SSH-Konfiguration korrekt ist.
  - Bei Scheitern des Verbindungsversuchs startet der Benutzer den Roboter erneut und folgt der Anwenderdokumentation.
  - Bei erneutem Scheitern wendet sich der Benutzer an den Systemadministrator.

### Unterabläufe (Subflows)

#### Unterablauf 1: Verbindung mit WLAN

1. Der Benutzer gibt die WLAN-Zugangsdaten ein.
2. Der Benutzer verbindet sich mit dem WLAN-Netzwerk des Roboters.



## **Unterablauf 2: Verbindung über SSH**

1. Der Benutzer gibt die SSH-Zugangsdaten ein.
2. Der Benutzer verbindet sich mit dem Roboter.

## **Nachbedingungen**

- Minimal Garantie: Der Roboter ist eingeschaltet.
- Success Garantie: Die Software zur Personenverfolgung ist aktiv und betriebsbereit.

## **Besondere Anforderungen**

- Der Benutzer muss sich in einem WLAN-Bereich mit ausreichendem Signal befinden.

## 2.3.2. Use-Case: UC02 Auf Web-Control-Center zugreifen

### Kurzbeschreibung

Der Benutzer greift über einen Webbrowser auf das Kontrollzentrum des Systems zu, um die Bilderkennung zu überwachen.

### Kurzbeschreibung der Akteure

- Benutzer

### Vorbedingungen

- Der Roboter ist gestartet und Benutzer mit dem WLAN des Roboters verbunden.

### Standardablauf (Basic Flow)

1. Der Use Case beginnt, wenn der Benutzer die Adresse des Web-Control-Centers im Browser eingibt.
2. Der Benutzer ruft die Website des Kontrollzentrums auf.
3. Der Benutzer überwacht die Bilderkennung über das Web-Control-Center.

### Alternative Abläufe

#### Alternativer Ablauf 1: Website nicht erreichbar

1. Der Benutzer überprüft die Netzwerkverbindung.
2. Der Benutzer stellt sicher, dass die Web-Control-Center-Adresse korrekt ist.
  - Bei Scheitern des Verbindungsversuchs startet der Benutzer den Roboter erneut und folgt der Anwenderdokumentation.
  - Bei erneutem Scheitern wendet sich der Benutzer an den Systemadministrator.

### Unterabläufe (subflows)

#### Unterablauf 1: Zugriff auf Web-Control-Center

1. Der Benutzer gibt die Adresse des Web-Control-Centers im Browser ein.
2. Der Browser stellt eine Verbindung zum Web-Control-Center her.

### Nachbedingungen

- Minimal Guarantee: Der Benutzer hat versucht, auf das Web-Control-Center zuzugreifen.
- Success Guarantee: Der Benutzer hat erfolgreich auf das Web-Control-Center zugegriffen und kann die Bilderkennung überwachen.

### Besondere Anforderungen

- Der Webbrowser muss kompatibel mit dem Web-Control-Center sein.

### 2.3.3. Use-Case: UC03 Motoren anschalten

#### Kurzbeschreibung

Der Benutzer schaltet die Motoren des Roboters ein.

#### Kurzbeschreibung der Akteure

- Benutzer

#### Vorbedingungen

- Der Roboter ist gestartet.

#### Standardablauf (Basic Flow)

1. Der Use Case beginnt, wenn der Benutzer die Hardwaretaste zum Einschalten der Motoren betätigt.
2. Das System aktiviert die Bewegung der Motoren.

#### Alternative Abläufe

##### Alternativer Ablauf 1: Motoren starten nicht

1. Der Benutzer überprüft die Stromversorgung des Roboters.
2. Der Benutzer überprüft die Hardwaretaste auf Funktionalität.
3. Der Benutzer versucht erneut, die Motoren einzuschalten.
  - Bei wiederholtem Scheitern wendet sich der Benutzer an den Systemadministrator.

#### Unterabläufe (subflows)

##### Unterablauf 1: Betätigung der Hardwaretaste

1. Der Benutzer drückt die Hardwaretaste.
2. Das System empfängt das Signal zum Einschalten der Motoren.

#### Nachbedingungen

- Minimal Garantie: Der Roboter hat versucht, die Motoren einzuschalten.
- Success Garantie: Die Motoren des Roboters sind eingeschaltet und betriebsbereit.

#### Besondere Anforderungen

- Die Hardwaretaste muss leicht zugänglich und funktionsfähig sein.

## 2.3.4. Use-Case: UC04 Motoren ausschalten

### Kurzbeschreibung

Der Benutzer schaltet die Motoren des Roboters aus.

### Kurzbeschreibung der Akteure

- Benutzer

### Vorbedingungen

- Die Motoren des Roboters sind eingeschaltet.

### Standardablauf (Basic Flow)

1. Der Use Case beginnt, wenn der Benutzer die Hardwaretaste zum Ausschalten der Motoren betätigt.
2. Das System deaktiviert die Bewegung der Motoren.

### Alternative Abläufe

#### Alternativer Ablauf 1: Motoren schalten nicht aus

1. Der Benutzer überprüft die Hardwaretaste auf Funktionalität.
2. Der Benutzer versucht erneut, die Motoren auszuschalten.
  - Bei wiederholtem erneutem Scheitern wendet sich der Benutzer an den Systemadministrator.

### Unterabläufe (subflows)

#### Unterablauf 1: Betätigung der Hardwaretaste

1. Der Benutzer drückt die Hardwaretaste.
2. Das System empfängt das Signal zum Ausschalten der Motoren.

### Nachbedingungen

- Minimal Guarantee: Der Roboter hat versucht, die Motoren auszuschalten.
- Success Guarantee: Die Motoren des Roboters sind ausgeschaltet.

### Besondere Anforderungen

- Die Hardwaretaste muss leicht zugänglich und funktionsfähig sein.

## 2.3.5. Use-Case: UC05 Roboter stoppen

### Kurzbeschreibung

Der Benutzer stoppt den Roboter und beendet die Software zur Personenverfolgung.

### Kurzbeschreibung der Akteure

- Benutzer

### Vorbedingungen

- Der Roboter ist gestartet und die Software zur Personenverfolgung läuft.

### Standardablauf (Basic Flow)

1. Der Use Case beginnt, wenn der Benutzer den Prozess in der Kommandozeile mit STRG + C abbricht.
2. Der Benutzer trennt die Stromversorgung des Roboters.

### Alternative Abläufe

#### Alternativer Ablauf 1: Prozess wird nicht beendet

1. Der Benutzer überprüft die Kommandozeileingabe.
2. Der Benutzer wiederholt den Befehl zum Abbrechen des Prozesses.
  - Bei wiederholtem Scheitern wendet sich der Benutzer an den Systemadministrator.

### Unterabläufe (subflows)

#### Unterablauf 1: Beenden des Prozesses

1. Der Benutzer drückt STRG + C in der Kommandozeile.
2. Der Prozess wird abgebrochen.

#### Unterablauf 2: Trennen der Stromversorgung

1. Der Benutzer trennt die Stromversorgung des Roboters.
2. Der Roboter wird ausgeschaltet.

### Nachbedingungen

- Minimal Guarantee: Der Roboter hat versucht, den Prozess zu beenden.
- Success Guarantee: Die Software zur Personenverfolgung ist gestoppt und der Roboter ist ausgeschaltet.

### Besondere Anforderungen

- Der Benutzer muss Zugang zur Kommandozeile haben.

# 3. System-Wide Requirements: Follow-Me Fahrroboter

## 3.1. Einführung

In diesem Dokument werden die systemweiten Anforderungen für das Projekt Follow-Me Fahrroboter spezifiziert. Die Gliederung erfolgt nach der FURPS+ Anforderungsklassifikation:

- Systemweite funktionale Anforderungen (F),
- Qualitätsanforderungen für Benutzbarkeit, Zuverlässigkeit, Effizienz und Wartbarkeit (URPS) sowie
- zusätzliche Anforderungen (+) für technische, rechtliche, organisatorische Randbedingungen

## 3.2. Qualitätsanforderungen für das Gesamtsystem

### 3.2.1. Benutzbarkeit (Usability)

- NFAU-1: Die Bedienung des Roboters soll leicht und ohne Vorkenntnisse möglich sein.
- NFAU-2: Das System soll als portable Möglichkeit zur Mitführung gestaltet werden.

### 3.2.2. Zuverlässigkeit (Reliability)

#### Verfügbarkeit

- NFAR-1: Das System sollte möglichst eine Stunde mit vollgeladenem Akku laufen.
- NFAR-2: Der Roboter sollte in der Lage sein, durch eine handelsübliche Powerbank, sobald dieser an den Zusatzakku angeschlossen wurde, in Betrieb genommen zu werden.

#### Wiederherstellbarkeit

- NFAR-3: Bei einem Fehler soll das System einfach durch Error-Handling in der Lage sein, sich nicht vollständig zu beenden, sondern einen Fehler zu werfen und so weiterhin zu laufen.

### 3.2.3. Effizienz (Performance)

#### Antwortzeit

- NFAP-1: Die Antwortzeit der Bilderkennung soll schnell genug sein um Personen folgen zu können. Eine Verarbeitungszeit muss hier bei < 1 Sekunde sein.

#### Durchsatz

- NFAP-2: Der Roboter muss i. S. d. Datendurchsatzes in der Lage sein, innerhalb von 60 Sekunden eine korrekte Antwort zur Bilderkennung zu liefern.

#### Kapazität

- **NFAP-3:** Der Roboter muss in der Lage sein, Bewegungsdaten mit einer Rate von ca. 100-1000 Datenpaketen pro Sekunde zu verarbeiten.

#### **Zeiten zum Starten oder Beenden**

- **NFAP-4:** Der Roboter samt Software muss in der Lage dazu sein, innerhalb von 10 Minuten zu starten und innerhalb von 1 Minute sich vollständig zu beenden.

### **3.2.4. Wartbarkeit (Supportability)**

#### **Kompatibilität**

- **NFAS-1:** Durch die Integration mit dem Betriebssystem ROS II muss das System mit anderen Systemen kompatibel sein.

#### **Systeminstallation**

- **NFAS-2:** Die Installation des Systems muss einfach und gut dokumentiert sein.

## **3.3. Zusätzliche Anforderungen**

### **3.3.1. Einschränkungen**

- zu nutzende Komponenten (Alphabot 2 PI)
- Vorgaben für die Programmiersprache (Python)
- zu unterstützende Plattformen / Betriebssysteme (ROS II)
- Physische Begrenzungen für Hardware, auf der das System betrieben werden soll (Raspberry Pi 4 8GB)
- später Wechsel auf Jetson Nano und Arduino Uno mit Auftraggeber vereinbart

### **3.3.2. Organisatorische Randbedingungen**

- fester Projektzeitraum - Abgabe 28.06.2024
- Zwischenberichte und regelmäßige Iterations-Meilensteine
- Ressourcenverfügbarkeit (Laborräume, sofern nötig und digitale Infrastruktur)

### **3.3.3. Rechtliche Anforderungen**

- Datenschutz (Personen im Bild, evtl. ohne eigenes ausdrückliches Einverständnis)

# 4. Glossar: Follow-Me Fahrroboter

## 4.1. Einführung

In diesem Dokument werden die wesentlichen Begriffe aus dem Anwendungsgebiet (Fachdomäne) der Follow-Me-Fahrroboter definiert. Zur besseren Übersichtlichkeit sind Begriffe, Abkürzungen und Datendefinitionen gesondert aufgeführt.

## 4.2. Begriffe

Begriff	Definition und Erläuterung	Synonyme
ROS II	<i>Robot Operating System 2</i> - ein internationales Open-Source Projekt, welches ein universelles und freies Betriebssystem für alle Arten von Robotern schaffen soll.	-
Open CV	<i>Open Source Computer Vision</i> - eine freie, plattformübergreifende Bildverarbeitungsbibliothek für die Echtzeit-Bildverarbeitung.	-
Implementierung/ Implementation	Prozess, in dem Programmcode gemäß den Spezifikationen erstellt, getestet und in das System integriert wird.	Verwirklichung, Einsetzung
Philosophie	bezeichnet die grundlegenden Prinzipien und Werte, die die Gestaltung und von Softwareentwicklung leitet.	Grund-/ Leitprinzipien, Kernwerte
Architektur	beschreibt Philosophie, Entscheidungen, Einschränkungen, Nebenbedingungen, Begründungen, wesentliche Elemente und andere übergreifende Aspekte des Systems, die Einfluss auf Entwurf und Implementierung haben.	Struktur, Vorgehensweise



Begriff	Definition und Erläuterung	Synonyme
Architectural Mechanisms/ Architekturmechanismen	Standardisierte Lösungen für häufige Probleme, die Komplexität minimieren und Entwicklung konsistenter Dienste fördern	-
Product Owner	<i>Produktbesitzer</i> - Rolle im agilen Projektmanagement, die für die Definition und Priorisierung von Anforderungen verantwortlich ist.	Auftraggeber
Projekt-Coach	Person, die das Team bei der Umsetzung des Projekts unterstützt und berät.	Projektberater, Projektmentor, Projektbegleiter
Iteration	Zyklus wiederholter Prozesse, um schrittweise Fortschritte zu erzielen.	Zyklus, Durchgang, Durchlauf
Creative Lead	Person, die kreative Aspekte des Projekts leitet und innovative Ideen einbringt.	Kreativleiter, Kreativdirektor
Architect	definiert Softwarearchitektur und trifft Schlüsselentscheidungen, die Gesamtdesign und Implementierung des Systems bestimmen.	Architekt
Lead Developer	leitet die Entwicklung des Projekts und ist für die technische Umsetzung verantwortlich.	Hauptentwickler, leitender Entwickler
Entwickler	Person, die an Programmierung und Umsetzung des Projekts beteiligt sind.	Programmierer, Developer
Discord	Plattform für Chat, Sprach- und Videokommunikation, wird hier für den Austausch im Team verwendet.	-
GitHub	Plattform für Versionsverwaltung und Zusammenarbeit an Softwareprojekten.	-

Begriff	Definition und Erläuterung	Synonyme
Glossar	Dokument, welches wesentlichen Begriffe aus dem Anwendungsgebiet (Fachdomäne) der Follow-Me-Fahrroboter definiert.	-
Raspberry Pi OS	Raspberry Pi OS (ehemals Raspbian) ist die offiziell empfohlene Linux-Distributionen für den Raspberry Pi. Raspberry Pi OS basiert auf der Linux-Distribution Debian und enthält jede Menge vorinstallierter Software.	-
Waveshare	International Limited ist ein führender technischer Anbieter von Hardware-Innovationen. Mit personalisierten Dienstleistungen und breit gefächerten Lösungen, die unsere Kunden in die Lage versetzen, ihre Ideen und Projekte zu verwirklichen, sind wir bestrebt, ein wichtiger Enabler im Ökosystem der Forschung und Entwicklung zu sein.	-
ReadMe Datei	Eine Readme-Datei – häufig als Readme.txt oder Readme.md erstellt – enthält üblicherweise wichtige Informationen zu dem jeweiligen System, Projekt oder der jeweiligen Software. Damit Nutzer die Datei direkt sehen, liegt sie im Idealfall in der obersten Verzeichnisebene.	-
Pep8 Python	<i>PEP 8</i> ist der "Style Guide for Python Code", also die Richtlinie zum Formatieren von Python-Code, deren Einhaltung für das saubere Verfassen von Python-Programmen erforderlich ist.	-

Begriff	Definition und Erläuterung	Synonyme
PyCharm	PyCharm ist eine dedizierte integrierte Entwicklungsumgebung (IDE) für Python, die eine breite Palette an wichtigen Werkzeugen für Python-Entwickler bietet, die eng integriert sind, um eine komfortable Umgebung für die produktive Entwicklung von Python, Web und Data Science zu schaffen.	-
Integrated Development Enviroment	Kurz: <i>IDE</i> ; Ist eine integrierte Entwicklungsumgebung für Software und für die Anwendungsentwicklung, die gängige Entwicklertools in einer zentralen grafischen Oberfläche vereint.	-
Linux-Distribution	Eine Linux-Distribution ist die Auswahl bestimmter Software und einem Kernel von Linux, die aufeinander abgestimmt sind. Je nach Anwendungszweck werden verschiedene Anwenderprogramme hinzugefügt	-
Kernel	Der Kernel ist ein elementares, zentrales Modul eines Betriebssystems, mit anderen Worten: der Betriebssystemkern.	-
Linux	Linux ist ein kostenloses Betriebssystem, das unter der GNU General Public License (GPL) veröffentlicht wird.	-
Wireframe[s]	Wireframes repräsentieren Teile der Informationsarchitektur, der Benutzerführung und der Navigation in einem Softwareprojekt.	Drahtgittermodell

<b>Begriff</b>	<b>Definition und Erläuterung</b>	<b>Synonyme</b>
Use-Case[s]	Ein Use-Case bündelt viele Anwendungsszenarien und demonstriert diese auf akkurate Weise mit gleicher Anzahl an Anwendungsfällen, welche vorher definiert wurden für ein System und den dazugehörigen Akteuren bzw. Stakeholdern.	Anwendungsfall
Arduino	Arduino ist eine Plattform für Elektronik- und Mikrocontroller-Projekte. Diese Plattform umfasst zum einen die Arduino Elektronik-Hardware-Boards und zum anderen die Arduino-Software, mit deren Hilfe die Boards programmiert werden können.	-
YOLO (You Only Look Once)	YOLO ist eine Methode/Weg zur Objekterkennung. Es ist der Algorithmus bzw. eine Strategie zur Bilderkennung von Objekten.	-
LiDAR	LiDAR ist eine Lasertechnologie, die die Oberflächen eines Objekts oder eines Raums dreidimensional erfasst. Das Wort "LiDAR" ist eine Abkürzung für Light Detection And Ranging.	-
Punktwolkenverarbeitung	Um eine Punktwolke mit Photogrammetrie zu erstellen, muss der Raum zunächst mithilfe von Kameras aus allen Winkeln erfasst werden. Die aufgenommenen Bilder werden dann in einer speziellen Software verarbeitet, um den Raum in 3D zu rekonstruieren.	-
Powerbank	Eine Powerbank ist ein externer Zusatzakku, mit dem man mobile Geräte wie Tablets, Notebooks oder Handys aufladen kann, wenn keine Stromquelle verfügbar ist.	Zusatzakku, Akku

Begriff	Definition und Erläuterung	Synonyme
NVIDIA Jetson Nano	Nvidia Jetson ist eine Serie von Embedded-Computing-Boards von Nvidia. Die Modelle Jetson TK1, TX1 und TX2 sind alle mit einem Tegra-Prozessor (oder SoC) von Nvidia ausgestattet, der eine CPU mit ARM-Architektur integriert. Jetson ist ein energiesparendes System und wurde zur Beschleunigung von maschinellen Lernanwendungen entwickelt.	Kleinrechner, Entwicklungsboard, Edge-Computing-Gerät
Personenerkennung	Personenerkennung funktioniert durch Bilderkennung. Es darf sich nur eine Person im Bild befinden, da der Roboter nur einer Person folgen kann und nicht zwischen Personen unterscheidet. Die zugehörige Software wird genauer in der Entwicklerdokumentation beschrieben.	-

### 4.3. Abkürzungen und Akronyme

Abkürzung	Bedeutung	Erläuterung
ROS II	Roboter Operating System 2	<i>Roboter Operating System 2</i> - ein internationales Open-Source Projekt, welches ein universelles und freies Betriebssystem für alle Arten von Robotern schaffen soll.
IDE	Integrated Development Enviroment	Ist eine integrierte Entwicklungsumgebung für Software und für die Anwendungsentwicklung, die gängige Entwicklertools in einer zentralen grafischen Oberfläche vereint.

Abkürzung	Bedeutung	Erläuterung
YOLO	You Only Look Once	YOLO ist eine Methode/Weg zur Objekterkennung. Es ist der Algorithmus bzw. eine Strategie zur Bilderkennung von Objekten.
LiDAR	Light Detection And Ranging	LiDAR ist eine Lasertechnologie, die die Oberflächen eines Objekts oder eines Raums dreidimensional erfasst.
UC	Use Case	Abkürzung für einen oder mehrere Use Cases.
SSH	Secure Shell	SSH ist ein kryptografisches Netzwerkprotokoll für den sicheren Betrieb von Netzwerkdiensten über ungesicherte Netzwerke.
FAQ	Frequently Asked Questions	Ein FAQ ist eine Sammlung von häufig gestellten Fragen, die bei der Verwendung des Projektes aufkommen. Es dient als erste Anlaufstelle für Problembehandlung.
USB	Universal Serial Bus	<i>Universal Serial Bus (USB)</i> - ein standardisiertes Anschlusssystem, das die Verbindung, Kommunikation und Stromversorgung zwischen Computern und elektronischen Geräten ermöglicht.

## 4.4. Verzeichnis der Datenstrukturen

Bezeichnung	Definition	Format	Gültigkeitsregeln	Aliase
Topic	Eine Topic ist ein Kommunikationskanal für ROS II, der es ermöglicht Daten zwischen verschiedenen Knoten (Nodes) in einem ROS II Netzwerk auszutauschen.	Format: nicht festgelegt Syntax: beginnen immer mit "/" + ""	Müssen dem Ros II Message-Format entsprechen	-

<b>Bezeichnung</b>	<b>Definition</b>	<b>Format</b>	<b>Gültigkeitsregeln</b>	<b>Aliase</b>
Topic: /image_raw	topic, welche unverarbeitete Rohbilddaten der Kamera überträgt	sensor_msgs.msg Image Formate	"	-
Topic: /opencv_image	Überträgt bearbeitete Bilder von der Camera_OpenCV-Node. Diese Bilder sind das Ergebnis der Human-Detection-Function und werden anschließend von der Camera_Streamer-Node empfangen.	sensor_msgs.msg Image Formate	" Bild mit Koordinatensystem sowie einer Visualisierung des Trackings einer Person	-
Topic: /position_data	Liefert Informationen über die Position der Person im Bild, welche von der Human-Detection-Function erzeugt werden.	std_msgs.msg Int32MultiArray Formate Array mit 7 int Werten	" Die Werte für custom_x center of person, custom_y center of person, width of person, height of person müssen im Bild liegen, also kleiner als width of image / height of image sein.	-
Topic: /joystick	Überträgt die von der Joystick-Node empfangenen Joystick Signale an die Movement_Control-Node.	std_msgs.msg String	" Muss eines von 5 Steuersignalen sein: Center, Up, Right, Left, Down	-

Bezeichnung	Definition	Format	Gültigkeitsregeln	Aliase
Topic: /servo	Übergibt die von der Movement_Controller-Node erzeugten Steuerdaten an die Pan_Tilt-Node.	std_msgs.msg Int32MultiArray Format Array mit 2 int werten [0] - servo winkel x [1] - servo winkel y	" Der Wert für Winkel-x muss zwischen -90 und 90 liegen Der Wert für Winkel-y muss zwischen -80 und 30 liegen	-
Datastream	Übertragung von Daten innerhalb einer Node.	-	-	-
ROS 2 Image	Identisch mit dem von /image_raw übertragenen Bild.	sensor_msgs.msg Image Formate	siehe /image_raw	-
OpenCV-Image	ROS 2 Image, welches von der CV_Bridge-Function in ein OpenCV-Image Format umgewandelt wurde.	bgr8	siehe /image_raw	-
Processed Image	Das von der Human_Detection_Function bearbeitete Bild. Anfangs identisch mit dem von /opencv_image übertragenen Bild, wird dann in der Camera_Streammer-Node mithilfe von CV_Bridge in das OpenCV-Image Format umgewandelt.	Beginn: sensor_msgs.msg Image Formate Später: bgr8	siehe /opencv_image	



Bezeichnung	Definition	Format	Gültigkeitsregeln	Aliase
Koordinaten/Coordinates	Sind die von der Human_Detection-Function erzeugten Daten, welche dann als /Position_Data weitergegeben werden.	Array mit 7 int werten (siehe /position_data)	siehe /position_data	-

## 5. Domain Model: Follow-Me Fahrroboter

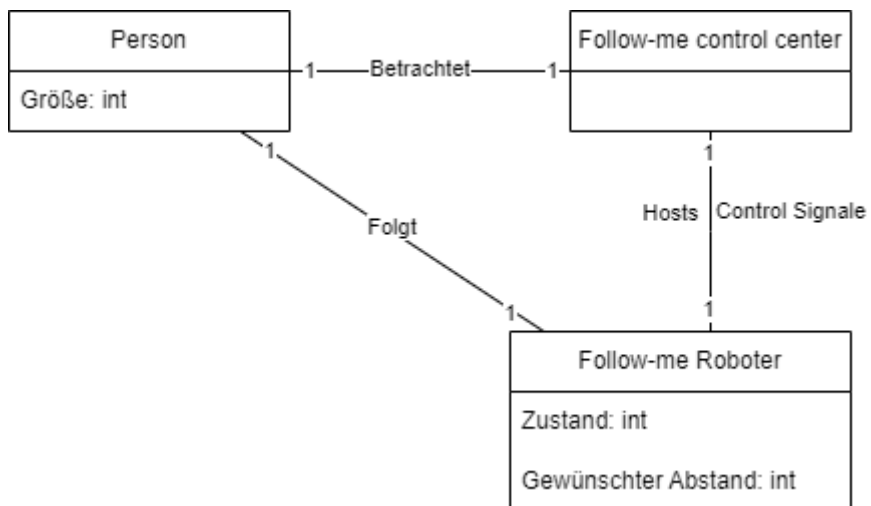


Abbildung 1. Domänenmodell nach Projektentwicklung