**HTL Saalfelden**

**Systemplanung und Projektentwicklung**

****

**Projektdokumentation**

**2023 / 2024**

|  |  |
| --- | --- |
| **Projektbezeichnung** | MBot2 |
| **Projektteam** | BOL |
| **Erstellt am** | 17.01.2024 |
| **Letzte Änderung am** | 06.06.2024 |
| **Status** | In Bearbeitung |
| **Aktuelle Version** | 1.5 |

**Änderungsverlauf**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Nr.** | **Datum** | **Version** | **Geänderte Kapitel** | **Art der Änderung** | **Autor** |
| 1 | 17.01.2024 | 1.0 | Kapitel 1 | Bearbeitung | Zvonimir Ljubic |
| 2 | 22.02.2024 | 1.1 | Kapitel 6.1 | Bearbeitung | Zvonimir Ljubic |
| 3 | 03.04.2024 | 1.2 | Alle | Bearbeitung | Zvonimir Ljubic |
| 4 | 25.04.2024 | 1.3 | Kapitel 2 | Erstellung | Zvonimir Ljubic |
| 5 | 29.05.2024 | 1.4 | Alle | Bearbeitung | Zvonimir Ljubic |
| 6 | 06.06.2024 | 1.5 | Alle | Fertigstellung | Zvonimir Ljubic |

**Inhalt**

[1. Allgemeines / Projektübersicht 4](#_Toc169653830)

[1.1 Projektbeschreibung 4](#_Toc169653831)

[1.2 Projektteam und Schnittstellen 4](#_Toc169653832)

[2. Funktionale Anforderungen 4](#_Toc169653833)

[2.1 Use Cases 4](#_Toc169653834)

[2.1.1 Verbindung des MBots mit dem Server 5](#_Toc169653835)

[2.1.2 User Login 5](#_Toc169653836)

[2.1.2 WASD-Steuerungsfunktion 5](#_Toc169653837)

[2.1.3 Geschwindigkeitseinstellung 5](#_Toc169653838)

[2.1.4 RGB-Einstellungsfunktion 5](#_Toc169653839)

[2.1.5 Linienverfolgungsfunktion 6](#_Toc169653840)

[3. Nichtfunktionale Anforderungen 6](#_Toc169653841)

[4. Projektplanung 6](#_Toc169653842)

[4.1 Variantenbildung 6](#_Toc169653843)

[4.2 Projektumfeldanalyse 6](#_Toc169653844)

[5. Softwarearchitektur 7](#_Toc169653845)

[5.1 Aktivitätsdiagramme 7](#_Toc169653846)

[5.1.1 Aktivitätsdiagramm 1 WASD-Steuerung 7](#_Toc169653847)

[5.1.2 Aktivitätsdiagramm n Name 8](#_Toc169653848)

[5.2 Sequenzdiagramme 8](#_Toc169653849)

[5.2.1 Sequenzdiagramm 1 User Login 8](#_Toc169653850)

[5.3 Komponentendiagramm 8](#_Toc169653851)

[5.4 Verteilungsdiagramme 9](#_Toc169653852)

[5.5 Softwarekomponenten / Programme 10](#_Toc169653853)

[5.5.1 SW Programme 10](#_Toc169653854)

[5.5.2 SW Komponenten 10](#_Toc169653855)

[6. Projektdurchführung 11](#_Toc169653856)

[6.1 Sprint 1 11](#_Toc169653857)

[6.1.1 Sprintplanung 11](#_Toc169653858)

[6.1.2 Sprint Demo 12](#_Toc169653859)

[6.1.3 Sprint Retrospektive 13](#_Toc169653860)

[6.1.4 Sprint Zusammenfassung 13](#_Toc169653861)

[6.2 Sprint 2 14](#_Toc169653862)

[6.2.1 Sprintplanung 14](#_Toc169653863)

[6.2.2 Sprint Demo 15](#_Toc169653864)

[6.2.3 Sprint Retrospektive 15](#_Toc169653865)

[6.2.4 Sprint Zusammenfassung 15](#_Toc169653866)

[6.3 Sprint 3 16](#_Toc169653867)

[6.3.1 Sprintplanung 16](#_Toc169653868)

[6.3.2 Sprint Demo 17](#_Toc169653869)

[6.3.3 Sprint Retrospektive 17](#_Toc169653870)

[6.3.4 Sprint Zusammenfassung 18](#_Toc169653871)

[6.4 Sprint 4 19](#_Toc169653872)

[6.4.1 Sprintplanung 19](#_Toc169653873)

[6.4.2 Sprint Demo 19](#_Toc169653874)

[6.4.3 Sprint Retrospektive 20](#_Toc169653875)

[6.4.4 Sprint Zusammenfassung 20](#_Toc169653876)

[6.5 Sprint 5 (Finaler Sprint) 21](#_Toc169653877)

[6.5.1 Sprintplanung 21](#_Toc169653878)

[6.5.2 Sprint Demo 21](#_Toc169653879)

[6.5.3 Sprint Retrospektive 22](#_Toc169653880)

[6.5.4 Sprint Zusammenfassung 22](#_Toc169653881)

[7. Installation / Software deployment 22](#_Toc169653882)

[8. Projektabschluß 23](#_Toc169653883)

[8.1 Projektzusammenfassung 23](#_Toc169653884)

[8.2 Attachments 23](#_Toc169653885)

# Allgemeines / Projektübersicht

## Projektbeschreibung

In diesem Projekt wurde ein MBot verwendet und es wurden verschiedenste Funktionen ausprogrammiert wie z.B., dass er einer Linie folgt, dass er Hindernisse vermeidet, uvm. Es wurde außerdem für dieses Projekt die Kommunikation zwischen Client und Server und MBot erstellt. Es wurde eine Website erstellt bei welcher sich der MBot steuern lässt. Anschließend wurde alles hier in dieser Datei dokumentiert.

## Projektteam und Schnittstellen

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Rolle(n)** | **Name** | **Telefon** | **E-Mail** | **Team** |
| Teammitglied | Zvonimir Ljubic | 06504012702 | zvonimir.ljubic@htl-saalfelden.at | BOL |
| Teammitglied | Thomas Oberhuber | 06645324155 | Thomas.oberhuber@htl-saalfelden.at | BOL |
| Teammitglied | Jonas Bogensberger |  | Jonas.bogensberger@htl-saalfelden.at | BOL |

# Funktionale Anforderungen

## Use Cases

Ein Bild, das Text, Diagramm, Reihe, Kreis enthält.

Automatisch generierte Beschreibung

### Verbindung des MBots mit dem Server

1. MBot verbindet sich mit dem WLAN
2. Wartet auf den Server
3. Server findet den MBot
4. Verbindung zum Server ist aufgebaut

### 2.1.2 User Login

1. User ruft die Seite auf
2. User wird zum Login gezwungen
3. User Logt sich ein
4. Die Login Daten werden an den Server geschickt
5. Der Server nimmt die Daten entgegen
6. Der Server überprüft die Daten
7. Der Login ist erfolgreich
8. Der User ist eingeloggt

### WASD-Steuerungsfunktion

1. Der User löst einen Steuerungsbefehl aus
2. Der Server nimmt den Befehl entgegen
3. Der Server leitet den Befehl an den MBot weiter
4. Der MBot nimmt den Befehl entgegen
5. Der MBot reagiert mit einer Bewegung auf den Befehl

### Geschwindigkeitseinstellung

1. Der User stellt die Geschwindigkeit ein
2. Der Server nimmt den Befehl entgegen
3. Der Server gibt einen Befehl an den MBot weiter
4. Der MBot nimmt den Befehl entgegen
5. Die Geschwindigkeit des MBots ist angepasst

### RGB-Einstellungsfunktion

1. Der User verändert die RGB-Einstellungen durch die Funktion
2. Es wird ein Befehl an den Server geschickt
3. Der Server nimmt den Befehl entgegen
4. Der Server sendet einen Befehl an den MBot
5. Der MBot nimmt den Befehl entgegen
6. Der LED-Einstellungen des MBots wurden geändert

### Linienverfolgungsfunktion

1. Der User wählt den Linienvefolgungsmodus aus
2. Der Server nimmt die Auswahl entgegen
3. Der Server leitet den Befehl an den MBot weiter
4. Der MBot nimmt den Befehl entgegen
5. Der MBot ist nun im Linienverfolgungsmodus

# Nichtfunktionale Anforderungen

Kommunikation zwischen Server und Mbot, soll über das WLAN erfolgen.

MongoDB soll zum Speichern von Daten verwendet werden.

Micropython wird für die Programmierung des Mbots verwendet werden.

# Projektplanung

Betriebssystem ist Windows. Programmiersprachen sind JS und Python. Micropython ist im Prinzip gesetzt, da der Mbot auf der Programmiersprache basiert. Es wird darauf abgezielt, die Termine einzuhalten und das Projekt erfolgreich abzuschließen.

## Variantenbildung

Wir hatten nicht viel Zeit zum Planen, somit haben wir uns auch direkt für die erste Variante entschieden nämlich einen Node JS Server zu programmieren und eine ganz normale html, CSS, JS, Webseite zu erstellen. Der Mbot soll mit Python programmiert werden. Als Datenbank wird MongoDB dienen.

## Projektumfeldanalyse

Da dieses Projekt im Rahmen eines Schulprojektes stattfindet ist es nicht auf den Markt fixiert, Stakeholder sind neben den Teammitgliedern die Lehrpersonen.

Lehrpersonen:

Prof. Falkensteiner

Prof. Eigner

Die Lehrpersonen, erwarten offensichtlich ein erfolgreich durchgeführtes Projekt und das alle geforderten Anforderungen erfüllt werden.

# Softwarearchitektur

Die Softwarearchitektur besteht aus einem Server, dem Client, der MongoDB und dem Mbot selbst. Der Server kommuniziert mit dem Mbot über das WLAN und verwaltet API-Endpunkte, um mit dem Client zu interagieren. Außerdem speichert der Server Sensordaten in der MongoDB Datenbank. Der Client wiederum kommuniziert mit dem Server und der Mbot erhält vom Server den Input und führt die Aktionen durch. Zur MongoDB wurde schon erwähnt, dass sie zur Speicherung der Sensordaten dient.

## Aktivitätsdiagramme

### Aktivitätsdiagramm 1 WASD-Steuerung

Anhand des Aktivitätdiagrammes sehen wir die WASD-Steuerung (etwas vereinfacht)

Wenn der Benutzer die Tasten W, A, S oder D drückt kommt es dazu, dass die WEB-Anwendung die Eingabe erfasst und einen Befehl an den Server sendet, welcher wiederum den Befehl empfängt und verarbeitet und an den Mbot einen Befehl sendet, dieser empfängt den Befehl vom Server und führt eine Bewegungsaktion durch.

Ein Bild, das Text, Screenshot, Diagramm, parallel enthält.

Automatisch generierte Beschreibung

## Sequenzdiagramme

### Sequenzdiagramm 1 User Login

Hier sehen wir ein Sequenzdiagramm zum User Login, wir haben 3 Actors nämlich User, Website und Server. Die Sequenz startet beim User, welcher dann die Website aufruft und sobald er die Website aufruft, muss er sich einloggen. Dieser Vorgang wiederholt sich so lange, bis er sich mit den richtigen Daten eingeloggt hat, währenddessen checkt der Server immer wieder die Daten und sendet einen Response an die Website. Wenn der Response true ist, also dass die Daten richtig eingegeben wurden, wird der User erfolgreich eingeloggt.

Ein Bild, das Text, Diagramm, Reihe, parallel enthält.

Automatisch generierte Beschreibung

## Komponentendiagramm

Hier sehen wir das Komponentendiagramm, welches aus folgenden Komponenten besteht:

* Server
* Mbot
* Database
* Client

Außerdem besitzen diese jeweils verschiedenste Interfaces z.B. beim Server die API-Endpoints, auf welche der Client sozusagen zugreifen kann, um mit dem Server zu kommunizieren. Beim Mbot sind die Interfaces die Sensoren, womit man ihn steuern kann.

Ein Bild, das Diagramm, Text, Plan, technische Zeichnung enthält.

Automatisch generierte Beschreibung

## Verteilungsdiagramme

Wie man am unten abgebildeten Verteilungsdiagramm entnehmen kann, besteht das Projekt aus vier Komponenten: Datenbank, Web-Server, MBot, Client.

Der Client muss in seinem Browser die Website ansteuern und kann anschließend mit dem Web-Server kommunizieren.

Der Web-Server empfängt Befehle der vom Benutzer und sendet diese über das WLAN an den MBot weiter. Zusätzlich speichert der Web-Server Daten in einer NoSQL Datenbank.

Der MBot basiert auf Micropython und wie bereits erwähnt erhält er Befehle des Servers und führt diese aus.

Ein Bild, das Text, Diagramm, Rechteck, Screenshot enthält.

Automatisch generierte Beschreibung

## Softwarekomponenten / Programme

### SW Programme

Visual Studio Code

* Version: 1.86.2

PyCharm

* Version: 2023.3.2

MBlock:

* Version:  V5.4.3

GitHub Desktop

* Version: 3.4.1

### 5.5.2 SW Komponenten

**Python:**

* Version:
* Hersteller: Python Software Foundation
* Bezugsquelle: [Python.org](https://www.python.org/downloads/release/python-3124/)
* SW-Lizenz: Python Software Foundation License

**Node.js:**

* Version: v20.14.0
* Hersteller: Node.js Foundation
* Bezugsquelle: [Node.js](https://nodejs.org/)
* SW-Lizenz: MIT License

**MongoDB:**

* Version:
* Hersteller: MongoDB Inc.
* Bezugsquelle: [MongoDB](https://www.mongodb.com/)
* SW-Lizenz: Server Side Public License (SSPL)

**MicroPython:**

* Version: 1.x
* Hersteller: Damien P. George
* Bezugsquelle: [MicroPython](https://micropython.org/)
* SW-Lizenz: MIT License

# Projektdurchführung

## Sprint 1

### Sprintplanung

Dauer: 07.02.2024-28.02.2024

Ausgewählte User Stories:

Als Entwickler möchte ich, dass der Roboter in der Lage ist, Befehle über die Anwendung über einen Server entgegenzunehmen und Daten zu senden, um die Fernsteuerung und Kommunikation zu ermöglichen. Story Points: 8

Als Anwender möchte ich die Möglichkeit haben, den MBot im Sicherheitsmodus zu betreiben, um versehentliche Kollisionen zu verhindern. Story Points: 4

Als Anwender möchte ich im automatischen Linienverfolgungsmodus die Steuerbefehle weiterhin über die Anwendung senden können, um den Roboter autonom einer Linie folgen zu lassen. Story Points: 5

Als Anwender möchte ich, dass die Fernsteuerung sowohl auf dem Computer als auch mobil funktioniert. Story Points: 2

Als Anwender möchte ich die Farben der LEDs auf der Controllereinheit über die Anwendung ändern können. Story Points: 2

Als Entwickler möchte ich eine umfassende Projektdokumentation erstellen. Story Points: 13

Als Anwender möchte ich, dass die Steuerung des MBots die IP-Adresse des Roboters anzeigt, um die Adresse zu kennen. Story Points: 2

Als Anwender möchte ich, dass die Anwendung eine Möglichkeit bietet, verschiedene Fahrmodi für den MBot2 auszuwählen. Story Points: 5

Anzahl Story points: 42

### Sprint Demo

In diesem ersten Sprint wurden erfolgreiche folgende Ziele umgesetzt:

Automatisches verbinden mit dem Schul-WLAN

Funktion für das Einstellen der Geschwindigkeit

Passwortgeschützte Seite

Linienverfolgungsmodus

IP-Adresse Anzeigen

Verschiedene Fahrmodi

Sicherheitsmodus

Die folgenden Ziele wurden in diesem Sprint nicht umgesetzt oder sind noch in Bearbeitung:

Kollisionsverhinderung

Fernsteuerung für Computer und Mobilgerät

Änderung von LEDs am Computer

Empfangen und senden von Daten des Roboters durch Server befehle

Die meisten dieser nicht umgesetzten Punkte sind nicht umgesetzt worden, da sich unser Team überschätzt hat.

### Sprint Retrospektive

Wenn man beachtet, dass es der erste Sprint war, lief es sehr gut vor allem gab es keine Schwierigkeiten bei der Kommunikation der Teammitglieder, genauso wie bei der Arbeitsteilung.

Ein negativer Punkt, welchen man in den nächsten Sprints umsetzen sollte, ist, dass man die fertigen Sachen auch commiten soll auf Github.

### Sprint Zusammenfassung

Burndownchart:

Ein Bild, das Text, Reihe, Diagramm, Zahl enthält.

Beschreibung automatisch generiert.

Man erkennt klar am Burndownchart, dass wir uns ziemlich überschätzt haben und somit einige Sotrypoints nicht fertiggestellt wurden. Dennoch lief der Sprint, dafür, dass es der erste war, relativ gut.

Ein Bild, das Text, Screenshot, Reihe, Diagramm enthält.

Automatisch generierte Beschreibung

Sprint Velocity: 24

Anhand der Sprint Velocity kann man noch nicht wirklich viele Informationen entnehmen, da dies der Erste Sprint war, was wir aber sehen können ist, dass man sich ziemlich in der Planung überschätzt hat. Es kann keine Aussage darüber getroffen werden, ob sich etwas am Endtermin verändert oder alles noch im grünen Bereich ist.

## Sprint 2

### Sprintplanung

Dauer: 06.03.2024-20.03.2024

Ausgewählte User Stories:

Als Anwender möchte ich, dass die Fernsteuerung sowohl auf dem Computer als auch mobil funktioniert. 5

Als Anwender möchte ich die Farben der LEDs auf der Controllereinheit über die Anwendung ändern können. 3

Als Anwender möchte ich das der M-Bot sich automatisch verbindet 5

Als Entwickler möchte ich, dass der Roboter in der Lage ist, Befehle über einen Server entgegenzunehmen und Daten zu senden. 8

Als Anwender möchte ich, dass die Steuerung des MBots eine Aus- und Einschaltfunktion bereitstellt, um eine bessere Fernsteuerung zu gewährleisten. 3

Anzahl Story points: 24

### Sprint Demo

Folgende Ziele wurden erfolgreich erreicht:

Server der Befehle entgegennimmt und Daten sendet

Fernsteuerung: Computer und Mobil

Farben der LEDs ändern

Server und MBot finden sich automatisch

In der Demo wurden, der Server, die Fernsteuerung und das Finden von Server und MBot gezeigt.

Folgende Ziele wurden nicht erreicht oder sind noch in Bearbeitung:

Ein- & Ausschaltfunktion

### Sprint Retrospektive

Lief alles gut es sind uns keine negativen Aspekte aufgefallen, dass einzig negative war vielleicht, dass ein Teammitglied oft abwesend war ansonsten lief alles gut.

### Sprint Zusammenfassung

Burndownchart:Ein Bild, das Reihe, Text, Screenshot, Diagramm enthält.

Automatisch generierte Beschreibung

Wie man am Burndownchart erkennen kann, wurden diesmal fast alle vorgenommenen Storypoints abgearbeitet sprich die Einschätzung hat sich verbessert im Vergleich zum 1.Sprint.

Sprintvelocity: 21

Ein Bild, das Text, Screenshot, Diagramm, Reihe enthält.

Automatisch generierte Beschreibung

Wie man erkennen kann, hat die Sprint Velocity nur leicht abgenommen, lag vor allem auch daran, dass ein Teammitglied den gesamten Sprint lang nicht da war und somit nichts weitergebracht hat. Da die Sprint Velocity etwas abgenommen hat kann es sein, dass man in Zukunft kleinere Pakte aus dem Sprintbacklog entfernen muss, wenn der Trend weiterhin abnimmt. Die Durchschnittliche Velocity liegt bei 22,5

## Sprint 3

### Sprintplanung

Dauer: 03.04.2024 – 17.04.2024

Ausgewählte User Stories:

Als Anwender möchte ich ein ansprechendes Websitendesign haben. 5

Als Anwender möchte ich den MBot über WASD steuern können. 3

Als Anwender möchte ich, dass die Steuerung des MBots eine Ausschaltfunktion bereitstellt, um eine bessere Fernsteuerung zu gewährleisten. 3

Als Anwender möchte ich die Werte aller Sensoren des MBots in der Anwendung angezeigt bekommen, um Echtzeitinformationen über den Roboter zu erhalten. 13

Als Anwender möchte ich, dass der M-Bot Kollisionen so vermeidet, dass er beim erkenne eines Hindernisses diesem ausweicht. 8

Als Anwender möchte ich eine API und eine Datenbank für die Sensoren des MBots haben. 8

Anzahl Storypoints: 39

### Sprint Demo

Folgende Ziele wurden erreicht:

Design der Website​

Steuerung über WASD​

API und Datenbank für Sensoren​

Folgenden Ziele wurden nicht erreicht oder sind noch in Bearbeitung:

Hindernisverhinderung​

Sensordaten an den Server schicken und anzeigen​

Ausschaltfunktion (Wir haben festgestellt, dass man es nicht umsetzen kann, deshalb haben wir es auch aus dem Backlog entfernt)

### Sprint Retrospektive

Auch in diesem Sprint lief alles gut es gab keine negativen Aspekte zumindest ist und nichts dergleichen aufgefallen. Vielleicht hätte man sich etwas mehr austauschen können über bestimmte Aufgaben.

### Sprint Zusammenfassung

Burndownchart: Ein Bild, das Text, Screenshot, Diagramm, Reihe enthält.

Beschreibung automatisch generiert.

Von den 39 Geplanten Storypoints wurden in diesem Sprint 25 Storypoints erledigt, es wurden 50% der Storypoints als erledigt dazugerechnet von den Zielen, welche zwar noch nicht fertig sind aber fast fertig sind und nur mehr noch testing oder kleine änderungen noch benötigen.

Sprint Velocity: 17

Ein Bild, das Text, Screenshot, Diagramm, Reihe enthält.

Automatisch generierte Beschreibung

Anhand der Sprint Velocity erkennen wir, dass auch in diesem Sprint weniger Storypoints abarbeitet, wurden als in den Sprints zuvor, dadurch mussten wir wieder einige User Storys in den nächsten Sprint verschieben, welche wir z.B. noch nicht komplett fertigstellen konnten, darunter unteranderem auch die Hindernisverhinderung und die Sensordaten anzeige. Dennoch nahm der Trend ab und somit liegt die durchschnittliche Velocity bei 20,66.

## Sprint 4

### Sprintplanung

Dauer: 24.04.2024 – 15.05.2024

Ausgewählte User Stories:

Als Anwender möchte ich die Werte aller Sensoren des Mbots in der Anwendung angezeigt bekommen, um Echtzeitinformationen über den Roboter zu erhalten. 13

Als Anwender möchte ich, dass die Anwendung mir eine Warnung gibt, wenn die Verbindung zum mBot unterbrochen wird. 2

Als Anwender möchte ich, dass der M-Bot eine „Suicide Prevention“ besitzt 5

Als Anwender möchte ich mittels eines Dropdown Menüs die Möglichkeit haben mehrere M-Bots auszuwählen und diese zu steuern. 13

Als Entwickler möchte ich eine umfassende Projektdokumentation erstellen. 13

Als Anwender möchte ich ein ansprechendes Websitendesign haben. 5

Als Anwender möchte ich im automatischen Linienverfolgungsmodus die Steuerbefehle weiterhin über die Anwendung senden können, um den Roboter autonom einer Linie folgen zu lassen. 8

Anzahl Storypoints: 46​

### Sprint Demo

Folgende Ziele wurden erreicht und auch in der Demo gezeigt:

Datenvisualisierung in Echtzeit​

Warnung bei Verbindungsabbruch​

Suicide Prevention​

Verbindung mit mehreren M-Bot

​Folgendes Ziel wurde nicht erreicht:

Dokumentation: Die Dokumentation konnte auch nicht als fertiggestellt gewertet werden, da sie erst nach Beendigung des Projektes als „erledigt“ gelten kann.

### Sprint Retrospektive

Auch in diesem Sprint sind keine Sachen schlecht gelaufen alles lief nach Plan wie man es auch aus der Zusammenfassung entnehmen kann.

### Sprint Zusammenfassung

Burndownchart:

Ein Bild, das Quittung, Reihe, Diagramm, Steigung enthält.

Automatisch generierte Beschreibung

Man beachte, dass der Burndownchart nicht wirklich aussagt, dass der Sprint 13 Storypoints offenließ, sondern sie sind nur als unerledigt markiert, da es sich hierbei um die Projektdokumentation handelt, welche schon zum größten Teil fertig war zu dem Zeitpunkt aber nur nicht präsentiert werden konnte, da die Person, welche die Projektdokumentation verfasst hat, nicht anwesend war am Tag der Präsentation. Selbst wenn man sie präsentiert hätte, wäre sie, als unerledigt bzw. als in Bearbeitung markiert worden, da die Dokumentation erst am ende des Projekts als vollständig fertig gelten kann, nachdem alle Inhalte dokumentiert werden konnten.

Sprint Velocity: 33Ein Bild, das Text, Screenshot, Zahl, Schrift enthält.

Beschreibung automatisch generiert.

Man kann klar erkennen, dass die Sprint Velocity bei diesem Sprint deutlich zugenommen hat, liegt vermutlich aber auch daran, dass dieser Sprint einen längeren Zeitraum hatte als die Sprints zuvor. Somit könnte der geplante Endtermin ohne Verluste erreicht werden, auch wenn man befürchtet, dass vielleicht paar User Stories dennoch gestrichen werden müssen. Die durchschnittliche Velocity steigt somit auf 23,74 an.

## Sprint 5 (Finaler Sprint)

### Sprintplanung

Dauer: 22.05.2025 – 05.06.2024

Ausgewählte User Stories:

Als Entwickler möchte ich eine umfassende Projektdokumentation erstellen. 13

### Sprint Demo

Diesmal gab es keine Klassische Sprint Review, sondern dies ist die Abschluss Demo des Projekts. Das gesamte Projekt wurde präsentiert und erklärt, zusätzlich wurden aus diesem letzten Sprint folgende Ziele gezeigt:

Dokumentation

### Sprint Retrospektive

Auch in diesem Sprint lief so gut wie alles wie geplant, dass einzige was sich negativ auf die Velocity ausgewirkt hat ist, dass wir neben den User Stories die wir behandeln wollten auch noch die Abschlusspräsentation fertigstellen mussten was ein zusätzlicher Aufwand war der auch einiges an Zeit in Anspruch nahm.

### Sprint Zusammenfassung

Sprint Velocity 26:

Ein Bild, das Text, Screenshot, Rechteck, Diagramm enthält.

Automatisch generierte Beschreibung

Im Finalen Sprint wurde ein Velocity von 26 erzielt, auch wenn nur wenige Arbeitspakete fertiggestellt wurden, dennoch ist diese Velocity im vergleich zu den vorherigen Sprints ziemlich gut und sie ist vor allem so hoch, da die Projektdokumentation miteingerechnet wurde. Es wurde außerdem in diesem Sprint die Finale Projektpräsentation erstellt, was dazu führte, dass wir nicht viel Zeit in die Fertigstellung anderer User Stories stecken konnten. Leider wurden nicht alle User Stories erfüllt. Die durchschnittliche Velocity des Projektes beträgt somit 24,2.

# Installation / Software deployment

Um das fertige System in Betrieb zu nehmen, muss man folgende Schritte machen:

1. Besuchen Sie das folgende GitHub Repository und laden Sie es herunter: <https://github.com/NebelNite/M-Bot.git>
2. Entpacken sie die zip Datei
3. Starten sie Visual Studio Code
4. Öffnen Sie den das Projekt
5. Starten Sie den Server
6. Starten Sie den MBot
7. Besuchen Sie die Seite
8. Logen Sie sich ein

Wenn die Verbindung zwischen Mbot und Server gegeben ist können Sie ganz einfach den Mbot über die Seite steuern bitte beachten Sie, dass auch alles richtig gestartet wurde.

# Projektabschluß

## Projektzusammenfassung

Dieses Projekt war sehr anspruchsvoll, es wurde viel Zeit investiert man hat viele neue Sachen gelernt und vor allem Erfahrung gesammelt. Großteil der behandelten Sachen am Projekt liefen gut nur es gab paar Punkte, die schlecht liefen, darunter die Fertigstellung des Linefollowers und anfangs gab es auch Probleme mit den Sensordaten, deshalb kam es zu einer Verzögerung. Allgemein kann man sagen, dass die Arbeitseinteilung verbesserungswürdig ist, und die Kommunikation zwischen den Teammitgliedern auch deutlich verbessert werden sollte, damit zukünftige Projekte effizienter und besser ablaufen können. Abschließend lässt sich sagen, dass man größtenteils vieles gleich machen würde, nur wie schon erwähnt sollte man beim nächsten Mal mehr Zeit in die Planung und Kommunikation investieren, damit man essenziell wichtige Anforderungen wie den Linefollower fertigbringt und dafür z. B. nicht so wichtige Funktionen streicht.

## Attachments

GitHub Repository: <https://github.com/NebelNite/M-Bot.git>