



Pflichtenheft

Intergalactical Polish Expansion - IPE

komo-296d & dix_-0815 & rauh-091e

E3FI1 – AK2025

Heinrich-Hertz-Schule

Präambel	2
1. Zielbestimmung	2
1.1. Muss-Kriterien.....	2
1.2. Wunschkriterien.....	4
1.3. Abgrenzungskriterien	4
2. Funktionale Anforderungen	5
2.1. Softwarearchitektur	5
2.2. Anwendungsfall.....	5
2.3. Datenmodell	6
2.4. UML-Diagramme	8
2.5. Betriebsbedingungen	9
3. Nicht funktionale Anforderungen	10
3.1. Einsatzumgebung Software	10
3.2. Einsatzumgebung Hardware.....	10
3.3. Einsatzumgebung Orgware.....	10
3.4. Produktschnittstellen	10
4. Projektablauf	12
4.1. Gantt-Diagramm	12
4.2. Netzplan.....	12
5. Benutzerschnittstelle	13
5.1. Bediener der Bodenstation.....	13
5.2. Endbenutzer.....	13
6. Abnahmekriterien	13
6.1. Qualitätsziele	13
6.2. Testszenarien	13

Präambel

1. Grund des Projekts

Mit den innovativen Funktionalitäten, die dieses Projekt bietet, werden wir in der Lage sein, neue Exoplaneten zu erforschen und deren Geheimnisse zu entschlüsseln. Unser Ziel ist es, nicht nur die Grenzen des bekannten Universums zu erweitern, sondern auch sicherzustellen, dass die Menschheit langfristig über ausreichend Lebensraum und wertvolle Ressourcen verfügt. Durch diese bahnbrechenden Entdeckungen wollen wir die Grundlage für eine nachhaltige Zukunft legen und neue Horizonte für kommende Generationen eröffnen.

2. Grund des Dokuments

Dieses Dokument stellt das Pflichtenheft dar. Es beschreibt welche Funktionalitäten wir geplant haben und wie wir diese umsetzen wollen.

Zusätzlich zu den Grundbausteinen eines Pflichtenhefts gehen wir hier auch auf die Projektplanung (siehe [Abschnitt 4: Projektablauf](#)) ein. Dieses Kriterium wurde im Lastenheft festgelegt.

3. Gliederung

Das Pflichtenheft ist in 6 Hauptthemen aufgeteilt:

1. Zielbestimmung
2. Funktionale Anforderungen
3. Nicht funktionale Anforderungen
4. Projektablauf
5. Benutzerschnittstellen
6. Abnahmekriterien

4. Disclaimer

Wir haben uns viel Mühe gegeben dieses Dokument humoristisch-ernsthaft zu formulieren, als ob dieses Projekt tatsächlich in der Realität eingesetzt werden soll. Damit erhoffen wir den Lesern gelegentlich ein Lächeln abzuknüpfen und die Trockenheit eines Pflichtenhefts ein wenig aufzubrechen.

1. Zielbestimmung

1.1. Muss-Kriterien

Die Solution wird simulieren, dass ein oder mehrere Roboter auf fremde Welten entsandt werden, diese erkunden, indem sie sich über die Oberfläche dieser Welten bewegen, die Umgebungswerte (Bodenbeschaffenheit und Temperatur) messen und diese Daten an eine Bodenstation auf der Erde sendet. Die Bodenstation wird diese Daten in einer SQL-Datenbank speichern, damit auch unautorisierte Nutzer sich die bisher unerkundeten Welten per Zugriff auf eine Webseite darstellen lassen können.

In den folgenden Abschnitten werden die funktionalen Kriterien für jede einzelne Komponente nochmals genauer beschrieben.

Funktion des Rovers

Die Rover werden sich, nachdem sie von der Bodenstation auf einen neuen Planeten entsendet wurden, auf eine Verbindung und respektive Befehle der Bodenstation warten, um sich über die Planetenoberfläche bewegen und nach

der Ladung jedes Feld nach dessen Eigenschaften scannen, bevor es befahren wird, da wir uns nicht sicher sein können, dass jedes Feld eine sichere Umgebung für die weitere Funktionalität der Rover sein wird.

Um diese Funktionen umzusetzen, werden die Rover:

- Das Feld unmittelbar vor sich untersuchen können
- Sich um 90° drehen können
- Sich ein Feld nach vorne bewegen können
- Die eigene Position nachverfolgen können, um sicher zu stellen, dass sie in finiter Zeit alle erreichbaren Felder eines Planeten scannen
- Melden können, dass ein Planet vollständig gescannt wurde
- Mit jedem neu gescannten Feld die Daten (Eigenschaften und Position des Feldes) unmittelbar an die Bodenstation senden können, um die Gefahr des Datenverlustes bei Zerstörung eines Rovers zu minimieren

Zusätzlich wird der Rover auch über einen Autopiloten verfügen, der in finiter Zeit alle erreichbaren Felder eines Planeten untersucht.

Die Übertragung der Daten wird grundsätzlich im JSON-Format erfolgen, um die Daten direkt strukturiert zu übermitteln. Für die Kommunikation zu den Rovern und für das Management dieser ist ein Server verantwortlich, der zwischen der Bodenstation und den Rovern agiert.

Funktionen der Bodenstation

Die Bodenstation wird die Schnittstelle darstellen, die es den Nutzer ermöglicht mit den Robotern zu kommunizieren. Sie ist dafür verantwortlich mit einem Rover Verbindungen einzugehen und diese zu halten, indem sie eine Verbindung zum zentralen Roververwaltungsserver pflegt. Nun stellt die Bodenstation sicher, dass alle Befehle des Nutzers an den Rover-Server weitergeleitet werden kann. Hierzu stellt sie ein Rover-Manager-Objekt, dass für die Verbindung zum Rover-Server verantwortlich ist, sowie für den Datenfluss von und zu diesem Server fließen. Die Antworten des Servers werden daraufhin dem Nutzer dargestellt, sofern vorhanden.

Für die Zwischenspeicherung der Daten des Rover-Manager stellt die Bodenstation einen gemeinsamen, synchronen Speicherbereich, der sowohl von der Bodenstation als auch von dem Rover-Manager bearbeitet werden kann. Falls die Daten, die der Rover an die Bodenstation schickt, speicherwert sind, speichern wir diese in der dazugehörigen Datenbank. Hierzu stellt die Bodenstation eine Verbindung zur Datenbank, über die Daten an die Datenbank übermittelt werden. Für diese Verbindung und Verarbeitung ist ein Datenbank-Manager-Objekt zuständig, dass ebenfalls von der Bodenstation gestellt wird.

Prinzipiell werden Daten der Bodenstation im JSON-Format verwaltet. Sowohl im synchronen Speicher der Bodenstation als auch auf jeglichen Datenübertragungen, sowohl in Richtung Rover als auch in Richtung Datenbank, erfolgen im JSON-Format, um Daten übersichtlich zu strukturieren.

Datenbank und Website

Die Datenbank wird von den Ergebnissen des Roboters befüllt und wird diese abspeichern. Sie wird Logs für jeden Rover und dessen Standorte führen. Es wird protokolliert werden, wenn ein Roboter crasht und auf welchem Grund er sich befinden wird. Diese Daten werden künftig von der HTML-Seite mit Java-Servlets entnommen, die Daten mit JavaScript vorbereitet und anschließend mit HTML und CSS dargestellt werden. Dies wird es dem Zuschauer ermöglichen, eine visuelle Präsentation des Rovers zu sehen und welcher Boden untersucht wurde. Das Konzept wird für Temperatur- und jegliche andere Daten, die der Rover erhalten wird, ausbaubar sein.

1.2. Wunschkriterien

Um diese Solution für die Kartographierung neuer Welten resilienter gegenüber lebensfeindlichen Planeten zu gestalten, planen wir zusätzliche Features, die verschiedene Vorteile mit sich bringen sollen.

Wir planen die zweite Generation der Rover direkt mit einem verbesserten Autopiloten auszustatten. Hierbei möchten wir den Lebenszyklus der Rover möglichst lang halten, weshalb sie zusätzlich zu den funktionalen Anforderungen auch ihren eigenen Zustand und die betriebliche Erhaltung berücksichtigen sollen. Dazu werden die Rover stetig ihren eigenen Batteriezustand überprüfen, bei Bedarf die Solarmodule auf einem geeigneten Feld ausfahren, um die Batterie wieder aufzuladen und kein Feld betreten, welches die eigene Zerstörung zur Folge haben könnte, nachdem sie erfolgreich auf einem Planeten gelandet sind.

Um die Schnittstelle für den Bediener der Bodenstation möglichst intuitiv und einfach zu gestalten, erstellen wir eine grafische Oberfläche zum Einsatz kommen, die alle wesentlichen Funktionen der Bodenstation in einer GUI zugänglich macht.

Zudem werden wir eine Log-File hinzufügen, die alle Eingaben und Ausgaben, die bestehen abspeichern, um in einem Fehlerfall eine bessere Fehlerauswertung anzubieten.

Die Datenbank wird zukünftig nicht nur die Position des Rovers protokollieren, sondern auch mit einem Zeitstempel versehen, um die zeitliche Abfolge der Bewegungen exakt nachzuverfolgen. Der Rover selbst wird visuell dargestellt, wobei sein zurückgelegter Weg grafisch aufbereitet und angezeigt wird. Während der Rover den Boden erkundet, werden die einzelnen Felder, die er aufdeckt, nach und nach sichtbar gemacht. Zusätzlich wird das visuelle Drehen des Rovers simuliert, um seine aktuelle Ausrichtung anschaulich darzustellen. Alle erhobenen Daten, wie Positionen, Bewegungen und Bodenbeschaffenheit, werden grafisch aufbereitet, um eine intuitive und übersichtliche Darstellung für den Nutzer zu gewährleisten.

1.3. Abgrenzungskriterien

Um dieses Projekt noch beliebter bei möglichen Käufern, wie NASA, ESA und SpaceX zu machen, können noch viel mehr Kriterien in dieser Lösung umgesetzt werden. Dennoch müssen wir Abstand von folgenden Funktionalitäten und Entwicklungskriterien nehmen, um den Kostenfaktor des Projekts nicht zu sprengen:

- Live-Kamera-Feed von der Oberfläche der zu erkundende Welt
- Rückkehrfunktion der Rover zum Raumschiff und/oder zur Erde
- Tools zur Kommunikation mit eventuell anzutreffenden intelligenten Lebensformen
- Funktionalitäten zur Befestigung von gesicherten Landeplätzen für zukünftige Besuche entdeckter Welten
- Funktionen, um schon von der Erde geeignete Landeplätze für die Rover zu finden, um noch weniger Drohnen bei der Entsendung zu verlieren
- Autorisierungsverfahren, um nur ausgewählten Nutzern die Darstellung der erkundeten Welten zu gewähren, um die Allgemeinheit nicht mit Informationen zu verunsichern, die sie nicht einordnen können
- die wirkliche Umsetzung in die Realität

2. Funktionale Anforderungen

2.1. Softwarearchitektur

Die Rover verwendet das State-Machine-Pattern, um verschiedene Betriebsmodi zu implementieren und flexibel zwischen diesen zu wechseln. Dadurch kann der Rover je nach Bedarf unterschiedliche Verhaltensweisen annehmen, was eine effiziente Anpassung an wechselnde Bedingungen ermöglicht.

Die Bodenstation hingegen folgt dem Singleton-Pattern, um sicherzustellen, dass nur eine Instanz der Bodenstation während des gesamten Betriebs existiert, was eine konsistente Kommunikation und Steuerung gewährleistet.

Die Daten werden in einer MariaDB-Datenbank gespeichert, welche mit DBeaver verwaltet und bearbeitet wird. Für die Entwicklung der Website kommt primär VSCode zum Einsatz, wobei bei Bedarf auch Eclipse verwendet wird, um spezifische Anforderungen zu erfüllen.

2.2. Anwendungsfall

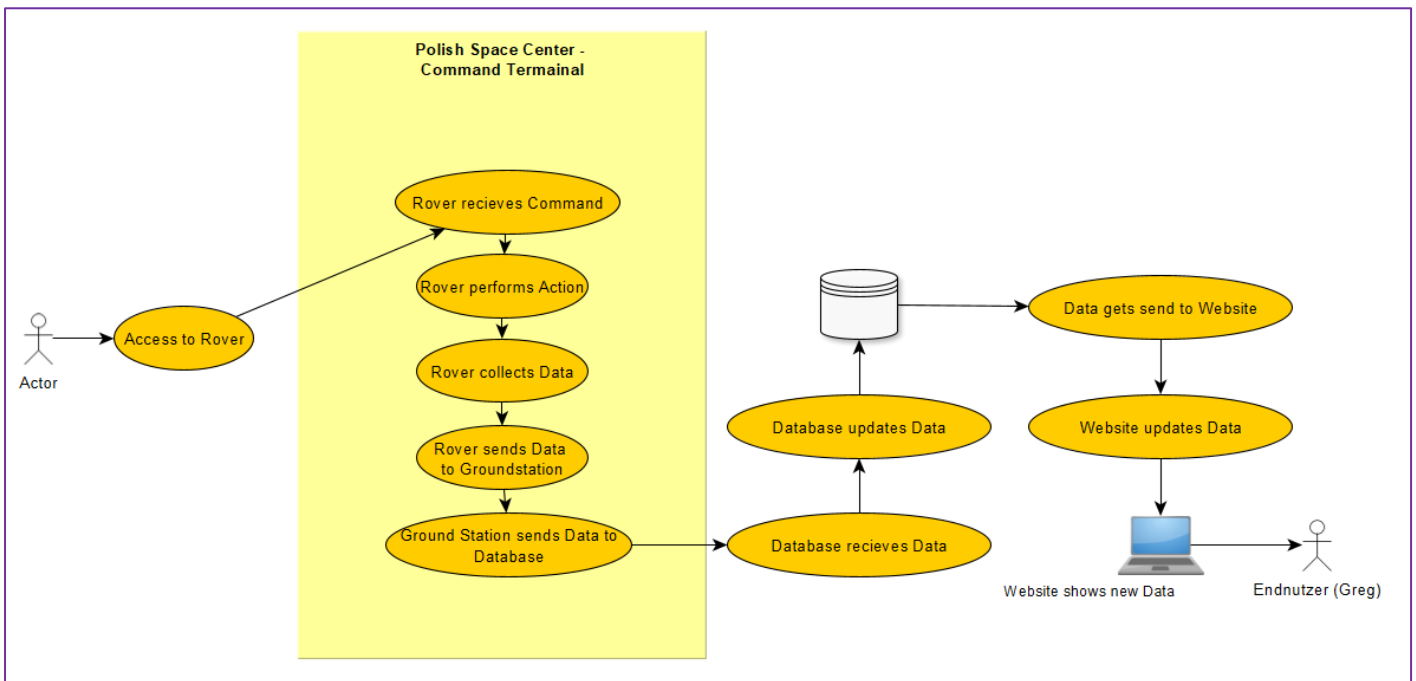


Abbildung 1: Use-Case Diagram

2.3. Datenmodell

Aus den geforderten Funktionen ergeben sich einige Daten, die teils erzeugt werden und teils durch die Untersuchung der Exoplaneten anfallen.

Per Definition des Lastenhefts, sollen zwar nur die Daten der Exoplaneten in der Datenbank gesichert werden, wir haben uns jedoch dazu entschieden noch zusätzliche Daten als speicherwürdig anzuerkennen, um die Missionen der Rover besser nachvollziehbar zu machen.

Anforderungsanalyse:

Robot: <ul style="list-style-type: none"> • <u>R-ID</u>: int • Name: text • P-ID: int (FK) • Last Activity: datetime • Last Input: text • ClientID: int • G-ID: int (FK) • D-ID: int (FK) • PosX: float • PosY: float • Timestamp: datetime • Textprotocoll: text 	Planet: <ul style="list-style-type: none"> • <u>P-ID</u>: int • Name: text • Size: int • R-ID: int (FK) 	Direction: <ul style="list-style-type: none"> • <u>D-ID</u>: int • Name: text
LastActivity: <ul style="list-style-type: none"> • <u>A-ID</u>: int • Name: text • Duration: float • Successful: boolean 	StatusHistory: <ul style="list-style-type: none"> • <u>S-ID</u>: int • R-ID: int (FK) • C-ID: int (FK) • A-ID: int (FK) • Messdatenhistorie: json • IsCrashed: boolean • ErrorProtocoll: text • LastError: text 	Ground: <ul style="list-style-type: none"> • <u>G-ID</u>: int • Type: text

Robot <> Planet

Jeder Roboter ist einem Planeten zugeordnet, dargestellt durch die Beziehung über die Planeten-Identifizier (P-ID). Ein Planet kann mehrere Roboter haben, die ihn erkunden.

Robot <> Ground

Die Bodentypen eines Planeten werden durch den Boden-Identifizier (G-ID) mit den Robotern verbunden. Ein Roboter kann sich auf genau einem Bodentyp bewegen, während ein Bodentyp von mehreren Robotern genutzt werden kann.

Robot <> Direction

Die Bewegungsrichtung eines Roboters wird über den Ausrichtungs-Identifizier (D-ID) in der Tabelle Direction festgelegt. Jeder Roboter kann nur eine Richtung gleichzeitig haben.

Robot <> LastActivity

Die letzte Aktivität eines Roboters wird über den Aktivitäten-Identifizier (A-ID) in der Tabelle LastActivity protokolliert. Ein Roboter kann mehrere Aktivitäten durchführen.

StatusHistory <> Robot, LastActivity

Die Statushistorie verbindet Roboter und LastActivity. Dies ermöglicht die Nachverfolgung von Messdaten, Erfolgen und Fehlern.

Datenbankmodelle:

Aus der Anforderungsanalyse haben wir folgende Datenbankmodelle erarbeitet.

Robot: (R-ID, Name, Planet (P-ID), last_Activity (ID), Messdaten-Historie (Textdokument), last_Input (letzter Input ins Textdokument), Connected Client (String), Ground (G-ID), Direction (D-ID), PosX, PosY, Timestamp, Text-Protokoll (Text-Datei), crashed (ID))

Planet: (P-ID, Name, Size, RobotID)

Ground: (G-ID, Type)

Direction: (D-ID, Name)

LastActivity: (A-ID, Name, Dauer, Status)

StatusHistory: (S-ID, Rover (R-ID), Action (A-ID), Messdatenhistorie

Abbildung 2: Relationsschreibweise

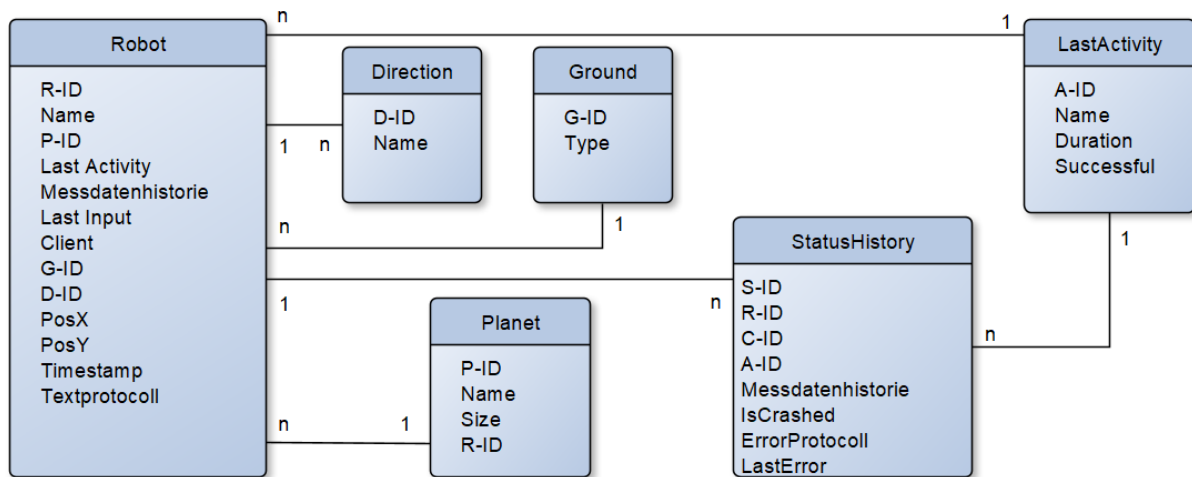


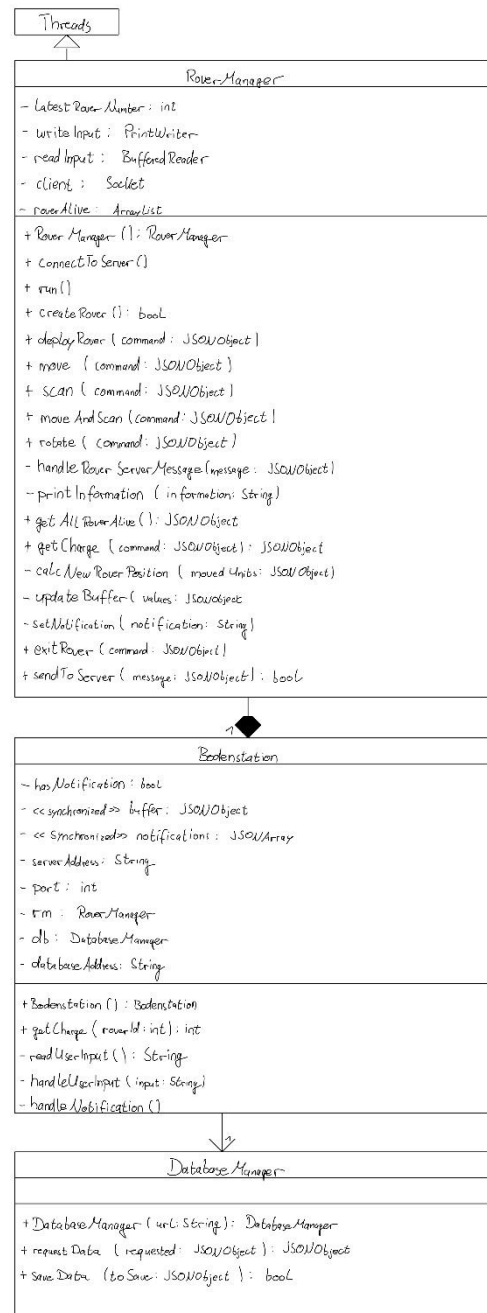
Abbildung 3: Entity-Relationship-Model

2.4. UML-Diagramme

2.4.1. Rover*



2.4.2. Bodenstation*



2.5. Betriebsbedingungen

Für den Betrieb des Systems ist es erforderlich, dass die Datenbank geöffnet und für alle relevanten Komponenten erreichbar ist, um einen reibungslosen Zugriff auf die gespeicherten Daten zu gewährleisten. Zudem müssen die Verbindungen zwischen den einzelnen Systemkomponenten korrekt konfiguriert sein, damit eine stabile Kommunikation und fehlerfreie Datenübertragung möglich ist. Diese Voraussetzungen sind essenziell, um die einwandfreie Funktion des Gesamtsystems sicherzustellen.

3. Nicht funktionale Anforderungen

3.1. Einsatzumgebung Software

Um die gewünschten Kriterien zu erfüllen, werden die einzelnen Programmkomponenten in Java programmiert. Dadurch werden die entsprechenden Entwickler Eclipse oder IntelliJ als integrierte Entwicklungsumgebung nutzen.

Folgende Bibliotheken werden zusätzlich zu den bereits mitgelieferten Abhängigkeiten benötigt:

Bibliothek	Zweck
Java.io	Umsetzen von Datenflüssen
Java.net	Herstellen von TCP/IP-Verbindungen
Java Database Connectivity	Verbindungsherstellung zu einer Datenbank/ Verwalten von Datenbank
org. JSON	Nutzung von JSON-Objekten
Json.awt	Erstellen einer GUI

Die Datenbank zur Sicherung der Ergebnisse des Rovers wird über MariaDB als SQL-Datenbank gehostet und mithilfe von DBeaver eingesehen und erweitert (zu Testzwecken).

3.2. Einsatzumgebung Hardware

Das Projekt wird mithilfe von drei Entwickler-Laptops verwirklicht, die als Arbeitsgeräte für das Entwicklerteam dienen. Für die Ausführung des Systems werden drei Rechner benötigt, die spezifische Aufgaben übernehmen: Ein Rechner hostet die Datenbank, ein weiterer simuliert die Bodenstation, und der dritte Rechner repräsentiert die Rover. Diese Aufteilung gewährleistet eine klare Trennung der Systemkomponenten und eine reibungslose Durchführung der Simulation.

3.3. Einsatzumgebung Orgware

Das Entwickeln größerer Projekte bringt meist besondere Hürden in Blick auf Kommunikation, Kollaboration und Dokumentation mit sich. Diese haben wir früh identifiziert und werden deshalb bestimmte Hilfsmittel bei der Organisation einsetzen.

Um den Programmcode nachverfolgbar, wartbar und zugänglich für alle Mitarbeitenden zu machen, werden wir Git und die Plattform GitHub benutzen.

Für die Zusammenarbeit in Dokumenten und zur Kommunikation von Verantwortlichkeiten wird Teams benutzt.

3.4. Produktschnittstellen

Da diese Lösung aus mehreren Komponenten besteht, müssen diese auch ordnungsgemäß miteinander verknüpft werden.

Exoplanet <-> Rover

Der Rover befindet sich unmittelbar auf dem Exoplaneten. Diese Verbindung wird zum Zweck der Simulation als Client-Server Relation realisiert, sodass die Rover als Client bei der Untersuchung und einzelnen Bodenflächen auf einem Exoplaneten Informationsanfragen an den Exoplaneten-Server senden.

Rover <-> Bodenstation

Die Rover werden durch einen zentralen Rover Server gesteuert. Der Server hat einen direkten Zugriff auf die Rover. Er ist in der Lage Rover zu erstellen, zu steuern und zu verlassen. Dieser Server stellt eine Netzwerkverbindung zur Bodenstation via TCP-Protokoll her. Über diese Schnittstelle fließen alle Informationen in jeweils beide Richtungen.

Bodenstation <-> Datenbank

Die Bodenstation stellt einen Datenbankmanager, der die Möglichkeit besitzt auf die Datenbank, via MySQL-Protokoll, zuzugreifen, dass auf den TCP/IP-Protokoll Stack basiert. Daher erfolgt auch hier eine Netzwerkverbindung. Hierdurch ist die Datenbank in der Lage Daten in der Datenbank zu speichern und abzufragen.

Datenbank <-> Webseite

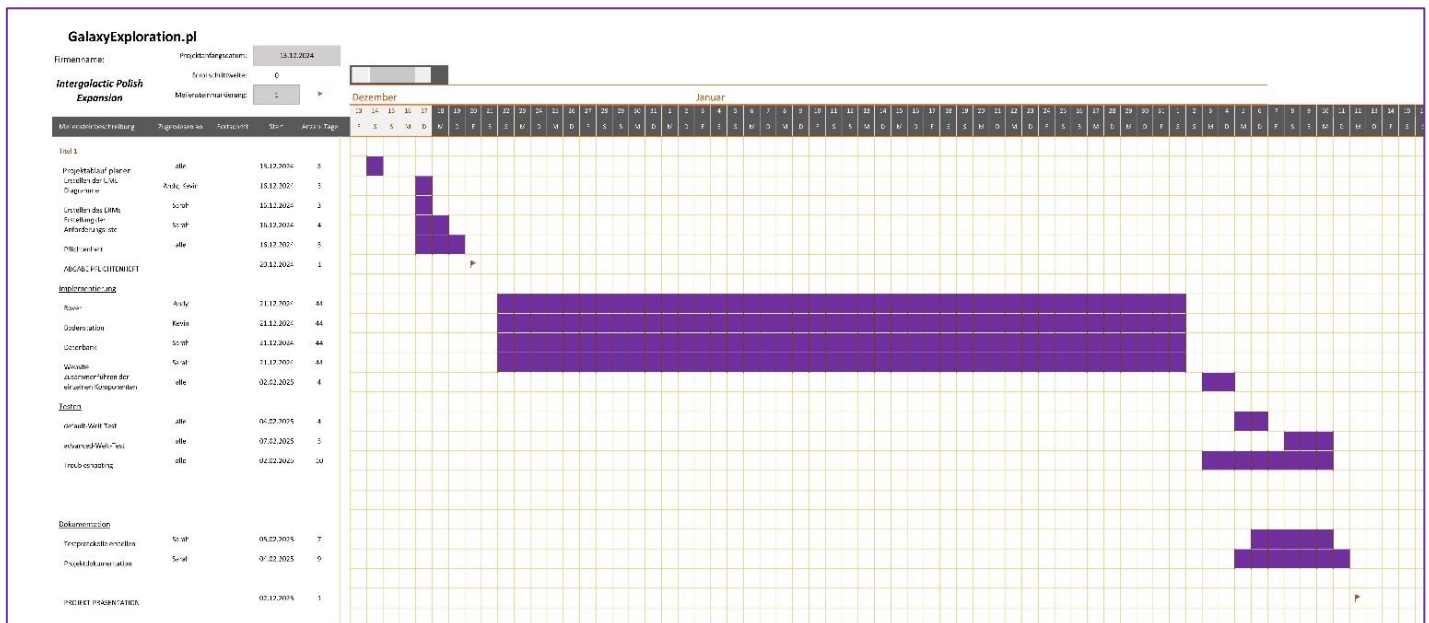
Die Schnittstelle zwischen der Datenbank und der Website spielt eine zentrale Rolle bei der Verarbeitung und Darstellung der Daten, die von den Rovern zur Erforschung von Exoplaneten gesammelt werden. Sie ist verantwortlich für die effiziente Übertragung und Speicherung der gewonnenen Informationen für die Endnutzer. Die Datenbank ist angebunden an die Bodenstation und leitet die Daten an die Website weiter, welche für die Visualisierung der Daten verantwortlich ist. Mit Java-Servlets ist dies leicht umzusetzen und mit JavaScript, HTML und CSS werden die Daten für den Endnutzer dargestellt.

4. Projektablauf

Dieses Projekt erstreckt sich über mindestens 60 Personenzeitstunden (in Präsenz), aufgeteilt auf drei Entwickler und achteinhalb Wochen. Damit wir zur Deadline am 12.02.2025 einen funktionierenden Prototyp abliefern können, wird der Ablauf der Entwicklung geplant und die Verantwortlichkeiten eindeutig aufgeteilt.

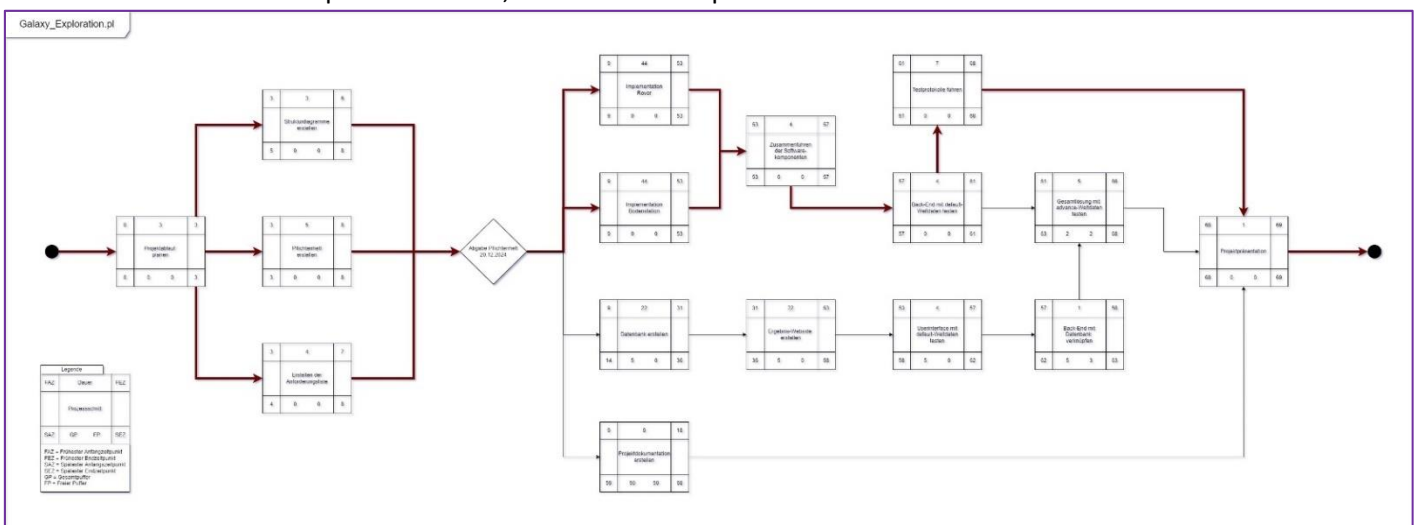
4.1. Gantt-Diagramm*

Um zielgerecht kollaborieren zu können haben wir die Arbeit an diesem Projekt zeitlich eingeplant und in Verantwortlichkeiten aufgeteilt, wie hier in diesem Diagramm zu sehen ist:



4.2. Netzplan*

Auch mit der Identifikation von zeitkritischen Teilprozessen haben wir uns schon in der Planungsphase beschäftigt, um herauszufinden, wo, wann und wie lange wir eventuelle Zeitpuffer für Verzögerungen im Entwicklungs- und Dokumentationsablauf einplanen können, wie hier im Netzplan beschrieben:



Der rot eingefärbte Pfad stellt den so genannten „kritischen Pfad“ dar. Verzögerungen bei den dort befindlichen Prozessen gefährden die per Definition die terminliche Einhaltung des Projekts, während alle anderen Prozesse einen gewissen Pufferzeitraum haben, bevor sie das Gesamtprojekt verzögern könnten.

Was nicht außer Acht gelassen werden darf, ist, dass wir die Hauptkomponenten der Lösung in unserer Freizeit erstellen werden, um diese kritische Phase zu entzerren, damit wir die Zeit, in der wir uns persönlich treffen, für die Zusammenführung der einzelnen Komponenten und die daraus eventuell erforderliche Fehlersuche und -behebung, effektiver nutzen können. Somit kann die Kritikalität höchstens die Umsetzung zusätzlicher Wunschfunktionen beeinflussen, damit die Implementierung der Grundfunktionen gesichert ist.

5. Benutzerschnittstelle

5.1. Bediener der Bodenstation

Der Bediener an der Bodenstation hat die Möglichkeit, eine Vielzahl von Aktionen über die Kommandozeile an die Rover zu senden. Dazu gehört das Erzeugen eines neuen Rovers, das Entsenden eines Rovers zu einem bestimmten Planeten und das Auswählen eines Rovers zur Befehlsannahme. Zudem kann der Bediener den Rover anweisen, ein Feld vorwärtszufahren – vorausgesetzt, dies würde den Rover nicht beschädigen –, sich um 90° nach links oder rechts zu drehen oder eine Bodenprobe zu entnehmen. Weiterhin ist es möglich, Zustandsinformationen eines Rovers abzufragen oder den Autopiloten zu aktivieren. Während all dieser Aktionen sendet der Rover kontinuierlich Daten zurück, die den Bediener stets auf dem neuesten Stand halten und gleichzeitig die Datenbank aktualisieren.

5.2. Endbenutzer

Der Endnutzer kann sich die Ergebnisse der Erkundung ganz einfach über eine Webseite darstellen lassen. Hier wird eine Tabelle angezeigt, die die verschiedenen Oberflächen der Planeten farblich darstellt. Außerdem werden die Eckdaten zu jeder Tabelle mitgeliefert (Größe, Name, Durchschnittstemperatur)

In zukünftigen Versionen wird hier auch der Pfad dargestellt der Rover über die Planetenoberflächen dargestellt.

6. Abnahmekriterien

6.1. Qualitätsziele

Die niedrigschwellige Anforderung dieses Projekts wird die Erkundung einer Default-Welt darstellen, indem die Bodenstation bestimmten Rovern eindeutige Befehle erteilt. Die Funde werden danach in einer Datenbank gespeichert und über eine Webseite farblich darstellbar sein.

In weiteren Szenarien werden die Rover weitere verschiedene Welten dieser Galaxie (Advance- und expert-Level) entweder autark oder über ein eigens entwickeltes User-Interface von der Bodenstation aus gesteuert werden, jedoch soll der Erhalt der Funktionalität der Rover oberste Priorität haben. Auf der Webseite zur Veranschaulichung der gesammelten Daten soll dann auch der Pfad der Rover dargestellt werden.

6.2. Testszenarien

Um die Qualitätsziele zu erreichen, wird es unumgänglich sein, Rover per Trial-and-Error zuerst auf bekannt Planeten zu entsenden, um deren Funktion zu testen.

Die ersten Tests werden sich darauf beschränken, ob die entsandten Rover noch von der Bodenstation erreichen lassen. Parallel kann bereits getestet werden, ob Platzhalterdaten in der Datenbank sich wie gewünscht auf der Webseite darstellen lassen.

Schritt zwei wird sein, den Rovern von der Bodenstation aus Bewegungsbefehle zu geben. Danach werden sie angewiesen Bodenproben zu nehmen und diese zu übermitteln. Wenn diese Daten bei der Bodenstation angekommen sind, wird getestet, ob die automatisierte Einspeisung derer in die Datenbank problemlos möglich ist.

Im letzten Testschritt, bevor wir die Rover zu lebensfeindlichen Welten entsenden, müssen Funktionalitätserhaltende Maßnahmen der Rover überprüft werden: Ermitteln der Umgebungstemperaturen der Felder vor dem Befahren, Kollisionsüberprüfung mit anderen Rovern, Batterieladungsstand berücksichtigen, Markieren geeigneter Felder für das Laden der Batterie per Solarzellen und das automatische Aufsuchen dieser bei kritischem Ladestand.