6/10/2024

PRJ Milestone: 1

Robotics Group: 4

Karabo Linala (577593)
Kelo Letsoal (577613)
Johannes Gerhardus Van Wyk (578007)
Kamahelo Mototo (577715)
Jordan Miguel Barradas (577848)
Jamie Sepp Butow (577588)

Executive Summary

This document outlines the project execution plan (PEP) for the final year robotic arm project for Robotics Group IV. The plan first elaborates on the project scope which is to create a robotic system which makes use of a robot arm to detect via computer vision technologies a variety of objects and be able to sort these objects by picking them up and placing them down without damage. In addition the plan outlines the overall objectives and deliverables of the project which include the design and development of an object detection system, the design and development of a gripping mechanism, the creation of a systems design document which contains the designs for both the object detection system and the gripping mechanism, the creation of a user manual which contains operational as well as maintenance information about the system as well as a demonstration of this system with industry captains in November 2024. Moreover, the plan gives details on the chosen agile software development methodologies and tools for this project which are Scrum and Kanban (boards) as well as giving examples of the various roles, documents, templates, technologies and charts required for both these methodologies. Furthermore, the plan elaborates on risk and risk management by identifying and classifying identified risks via a qualitative risk assessment using a risk matrix and then specifying various mitigation strategies in order to reduce/ eliminate the impact of these risks on project outcomes.

Table of Contents

Table of Figures

Project Scope

Scope Description: the robotic arm system this project aims to program should be able to distinguish between objects based on visual characteristics (size and shape) and be able to sort these objects (from smallest object to largest) by picking them up and dropping them off without damaging them.

Acceptance Criteria:

- Robotic system should make use of computer vision technologies to distinguish between objects based on visual characteristics (functional object detection system).
- Robotic system should be able to pick up and place down objects without damaging them in any way (development of a gripping mechanism).
- Robotic system should also be able to sort these objects based on visual characteristics from smallest object to largest.

Key Deliverables:

- **1X Robotic Arm system** that contains an object detection system as well as gripping mechanism which allows the robot arm to detect, pick up and place down as well as sort objects based on visual characteristics (should be fully functional by deadline for milestone 3: 01/11/2024).
- **1X System Design document** that includes elaboration on both hardware and software components (system design should be completed before deadline for milestone 2: 06/09/2024).
- **1X User Manual and Technical Document** that includes information for maintaining and operating the system (should be completed by the deadline for milestone 3 : 01/11/2024).
- **1X Project Presentation** of the system at the 2024 Project Expo held at Belgium Campus in November to demonstrate the systems capabilities (completed before week of 2024-11-18 to 2024-11-22).
- **1X Project Execution Plan** should be developed and finalized before 13/07/2024.

Project Exclusions:

- Use of more than one robot arm model.
- Use of technologies in the detection system that cannot be described as visual detection technologies.

Project Constraints:

- Time constraint: time limit of 3 months.
- Human Resource constraint: 6 Bachelor of Information Technology students with intermediate knowledge of robotics and programming.
- Budget constraint: project is constrained to the monetary resources allocated to it by the university.
- Equipment: project will use of robot arm located at the Pretoria Campus of Belgium Campus.

Project Objectives

Overall objective statement: within the allotted 3 months, create a Robotic System that is capable of detecting objects using an object detection system and then sort a variety of these objects by picking them up and placing them down without damaging them.

Specific and Measurable Goals

1. **Development of a Gripping Mechanism** – Part of Deliverable: *Robotic Arm System*
    - Must be capable of picking up and placing down objects without damaging them.
    - Deadline: 01/11/2024
2. **Development of an Object Detection System** – Part of Deliverable: *Robotic Arm System*
    - Must be capable of detecting and quantifying objects based on visual characteristics (size) using computer vision technologies (image sensors).
    - Deadline: 01/11/2024
3. **System Design Documentation –** Part of Deliverable: *System Design Document*
    - **Hardware Components** : Provide thorough specifications and schematics for each piece of hardware that the system uses.
    - **Software Components** : Provide thorough documentation that explains the rationale and flow of the software’s algorithms and code.
    - Deadline: 06/09/2024
4. **User Manual and Technical Documentation –** Part of Deliverable: *User Manual & Technical document*
    - **Operating Instructions** : Write a User Guide that provides detailed instructions on how to utilize the Robotic System.
    - **Maintenance Guide** : Provide a Technical Handbook that describes routine maintenance practices and troubleshooting techniques.
    - Deadline: 01/11/2024
5. **Project Presentation/Demonstration –** Part of Deliverable: *Project Presentation*
    - **Stakeholder Presentation** : Create and give a presentation that highlights the features of the system and includes recorded or live examples of item identification, picking, and placing.
    - **Feedback Integration** : During the presentation, get input from the stakeholders; then include pertinent recommendations in the final documentation and system enhancements.
    - Deadline: 2024-11-18 to 2024-11-22

Requirements Analysis

Functional requirements and non-functional requirements for the Robotic System are as follows:

Functional requirements:

- **Object Detection and Classification:**
    - The system must detect objects using computer vision techniques and classify them based on predefined visual characteristics (size of the object).
- **Gripping Mechanism:**
    - The robotic arm must have a gripping mechanism capable of picking and placing objects of various shapes and sizes without causing damage.
- **Integration and Control:**
    - The object detection system and the gripping mechanism must be seamlessly integrated with the robotic arm, ensuring smooth mechanical and electrical interfacing.
- **User Interface:**
    - A user-friendly interface must be provided to allow operators to start and stop robotic system sorting operation – done through a mobile application using an API (application program interface) to connected to the robotic arm system.

Non-functional requirements:

- **Performance:**
    - The system must perform object detection and classification in real-time with minimal latency. The sorting process should be efficient, ensuring the robotic arm operates at optimal speed without compromising accuracy.
- **Reliability:**
    - The system should demonstrate high reliability with minimal downtime, using durable components capable of sustained operation.
- **Scalability:**
    - The system should be designed to allow for future upgrades or modifications, supporting the addition of new object categories or enhancements to the gripping mechanism.
- **Usability:**
    - The user interface should be intuitive and easy to use, even for individuals with minimal technical expertise. The system should provide clear feedback regarding its status and any issues that arise.

- **Safety:**
  - The system must include safety features to prevent accidents or injuries during operation. The robotic arm should stop operation if it detects an obstacle or potential collision.
- **Compatibility:**
  - The system should be compatible with existing robotic arm models and standard interfaces, ensuring seamless integration with common operating systems and development environments.
- **Maintainability:**
  - The system should be easy to maintain, with accessible components and clear maintenance procedures. The documentation should include troubleshooting guides and maintenance schedules.
- **Compliance:**
  - The system must comply with relevant industry standards and regulations for robotics and automation, adhering to best practices for electrical and mechanical safety

These requirements ensure that the robotic arm will be efficient, reliable and user-friendly for the automated object sorting.

High Level Design

Section is outdated as of milestone 3. For updated software and hardware components as well as updated high level design please see section 1 of milestone 2 document for updated contents necessitated by university hardware availability.

The following secti his project as well as their interaction

Hardware Com

Kuka Robot Ar

- The KUKA robotic Arm's hardware consists of parts including motors, sensors (like the proximity sensors, vision systems, and force/torque sensors), end-effectors, and other mechanical pieces that combine to al he robot arm to carry out its duties efficiently.

Kuka Smart Pad

- Users may communic ith and programme the robot via the Sm ad, which serves as the control interface for KUKA robo rms. It is designed to be simple t e and straightforward, giving you control over the robot's a ns and real-time feedba he following are some significant characteristics:
    - Interface via Touchscre he Smart Pad a touchscreen display that makes navigating through settings and men ple.
    - Ergonomic Design: The Smart sign makes it easy to handle and use for extended periods of time.
    - Customisable Buttons: De ing o eir requirements, users may set up the Smart Pad's buttons to carry o articular task
    - Real-time Monitori ith the Smart Pad, s can keep an eye on the robotic arm's condition and fu nality at all times.
    - Programmin d Debugging: Using the Smart Pa ers may directly programme the robot and ug their apps.

(Kuka, 2024)

Image Sensors

- An essential part of robotic vision systems are image sensors. They take in visual data from their surroundings, which may be processed for a number of uses, including navigation, localization, and object identification. This is a summary of how they are used in the KUKA arm:
    - ➢ Detection: The KUKA arm can identify items in its working area thanks to image sensors. For example, a robotic arm in an assembly line can recognise components and carry out exact tasks according to their placement and orientation.

Section is outdated as of milestone 3. For updated software and hardware components as well as updated high level design please see section 1 of milestone 2 document for updated contents necessitated by university hardware availability.

➢ Actual As an illustration, the sensors from the Sony IMX series are widely utilised in numerous fields, such as robotics. These sensors are appropriate for object detection and tracking in dynamic situatio ght performance.

- *How image*
  - ○ Tak ng light.
  - ○ Ligh verting light into elec
  - ○ Processing the Image: The robot's control system processes the digital image to extract relevant ata, like item identification and location.
- *Example of use o ge Sensors in KUKA arm*
  - ○ Identify Item nd and identify items that are moving down a veyor belt.
  - ○ Establish Positio etermine each object's exact position orientation.
  - ○ Direct Motions: Giv e robotic arm input so that it c recisely grip objects and transfer them to the correct sp

Figure 1: Example of Kuka Robot Arm

Software Components:

Section is outdated as of milestone 3. For updated software and hardware components as well as updated high level design please see section 1 of milestone 2 document for updated contents necessitated by university hardware availability.

Kuka Work Visual

- Kuka Robo
- An integra d to programme, configure, olkit to effectively work with

➢ *How it*

- **Project Setup:** Using WorkVisual, users establish a new project and enter parameters such as the controller robot model.
- **Programming Configuration:** Using the offered code develop and system configuration tools, users creat configure robot programmes.
- **Testing and Simulati** The software enables users to eva the program's functionality and make necessary improve ts by simulating the behavi of the robot in a simulated setting.
- **Deployment:** After the prog me is complete, it m e uploaded and run on the actual robot.
- Monitoring and Maintenance: arantee opti performance, WorkVisual provides tools for keeping an eye on the robot's op ons, id ying problems, and carrying out maintenance procedures.

(Kuka, 2024)

Kuka Sim

- KUKA Sim is a powerful simula software de ped by KUKA for the purpose of designing, simulating, and optimizing r systems and produ lines in a virtual environment. This tool allows users to create de d simulations of robotic pr ses, ensuring efficiency and accuracy before implementati the real world.

➢ *How it works:*

- **Create an tup:** Using a 3D simulation environment, us reate the robotic cell or produ line, arranging robots, tools, and other equipment as ssary.
- **Programming:** The user-friendly programming interface of KUKA.Sim is used to programme the robots. Within the simulation, users may design and test robot movements and functions.
- **Simulation and Optimisation:** The robot programmes are tested, collisions are checked, cycle durations are examined, and pathways and operations are optimised through simulation runs.
- **Validation:** To make sure the robot programmes will function as planned in the actual world, the simulation results are verified.
- **Deployment:** The completed robot configurations and programmes are moved from the simulation environment to the real KUKA robots so they may be used.

(Kuka, 2024)

Interactions:

Command flow

Section is outdated as of milestone 3. For updated software and hardware components as well as updated high level design please see section 1 of milestone 2 document for updated contents necessitated by university hardware availability.

- From MES t nverts them into precise moti
- Robot to M data and status updates bac

Sensor feedback loop

- Real-Time Adjustments: The controller receives constant data from the sensors, enabling dynamic changes to the robot's activities.
- Error detection: Quick identification and remediation of abnormalities (such as obstructions and misalignment).

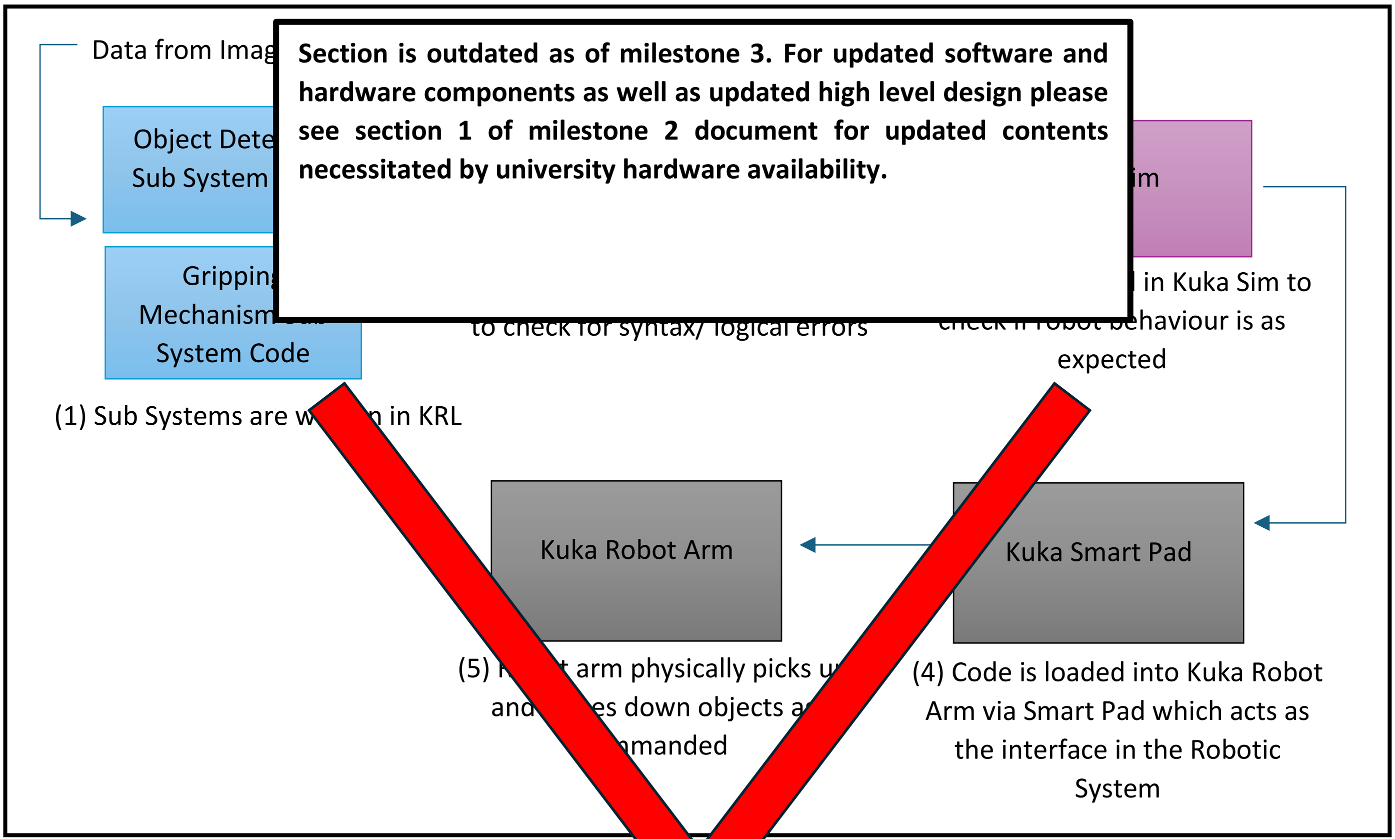
Safety interactions:

- Emergency Stops: When the stop button is pressed, operations are immediately stopped.
- Light Curtains: To ensure operator safety, the robot arm will automatically stop if the light curtain's beam breaks.

Overall Design Approach:

- **Object Detection Sub-system:**
  - KRL code that uses data sent by image sensors that pertains to the size of objects to calculate the sorted order of the objects.
- **Gripping Mechanism sub-system:**
  - KRL code that uses data supplied to it by the object detection subsystem to order the Kuka Robot Arm to physically pick up and place down an object in the sorted order.

The following diagram illustrates the overall design approach for the project:



Section is outdated as of milestone 3. For updated software and hardware components as well as updated high level design please see section 1 of milestone 2 document for updated contents necessitated by university hardware availability.

Data from Imag
Object Dete
Sub System
Gripping
Mechanism Sub
System Code
(1) Sub Systems are written in KRL
to check for syntax/ logical errors
im
in Kuka Sim to
check if robot behaviour is as
expected
Kuka Robot Arm
Kuka Smart Pad
(5) Robot arm physically picks up
and places down objects as
commanded
(4) Code is loaded into Kuka Robot
Arm via Smart Pad which acts as
the interface in the Robotic
System

Figure 2: Overall Design Approach

Modular Architecture:

- Scalability: The ability to add or remove parts (such as sensors and end-effectors) without having to completely rebuild the system.
- Maintainability: By limiting problems to certain modules, it makes diagnosis and maintenance easier.

Safety Systems with Redundancy:

- Conformity: Following global safety regulations (such as ISO 10218-1 and ISO 13849).
- Multiple Layers of Safety: Makes sure that there are backup safety systems in place to keep operators safe in the event that one fails.

Project Timeline

The following section details how the project will be broken down into sub-tasks and milestones (***please note: these milestones differ from the PRJ marking milestones)***

Milestone 1: Part 1 - Design Object Detection System

Objective : Research and develop a design for an object detection system that can differentiate and quantify objects based on their size using computer vision technologies (image sensors).

- Conduct necessary research into available sensors and computer vision technologies.
- Conduct necessary research into efficient sorting algorithms.
- Iterate over the design as necessary to ensure compatibility with the robot arm and other sub-systems (i.e. the Gripping mechanism).
- Continually add all work done to the *Systems Design Document* as this milestone will form part of that deliverable.
- **Deadline: 06/09/2024**

Milestone 1: Part 2 - Design Gripping Mechanism

Objective : Research and develop a design for a Gripping Mechanism that is capable of physically moving a variety of objects by picking them up and placing them down without damaging them.

- Conduct necessary research into available and compatible grips for the robot arm.
- Conduct necessary research into manipulating the grip with software as to grip firmly onto objects but not so firm as to break, scratch or damage the object.
- Iterate over the design as necessary to ensure compatibility with the robot arm and other sub-systems (i.e. the Object Detection System).
- Continually add all work done to the *Systems Design Document* as this milestone will form part of that deliverable.
- **Deadline: 06/09/2024**

Milestone 2: Part 1 - Development of Object Detection System

Objective : Implement the design finalized in Milestone 1 and physically code the Object Detection System and test it rigorously.

- Conduct any training or research necessary on the software packages and programming language required to code the Kuka Robot Arm and interface with it.
- Code the sub-system in KRL (Kuka Robotic Language) to the specifications listed in the design document, ensuring that it meets project acceptance criteria.

- Continuously test this sub-system for expected functionality as well as integration with other sub-systems (i.e. Gripping Mechanism) throughout the milestone.
- During this milestone continuously add information about the operation and maintenance instructions to the *User Manual & Technical Information* document as this milestone forms part of that deliverable.
- **Deadline: 01/11/2024**

Milestone 2: Part 2 – Development of Gripping Mechanism

Objective: Implement the design finalized in Milestone 1 and physically code/ implement the Gripping Mechanism and test it rigorously.

- Conduct any training or research necessary on the software packages, hardware and programming language required to code the Kuka Robot Arm and interface with it (attach and manipulate the arm).
- Code the sub-system in KRL (Kuka Robotic Language) to the specifications listed in the design document, ensuring that it meets project acceptance criteria.
- Continuously test this sub-system for expected functionality as well as integration with other sub-systems (i.e. Object Detection System – does it move correctly according to the commands given to it by that sub-system etc) throughout the milestone.
- During this milestone continuously add information about the operation and maintenance instructions to the *User Manual & Technical Information* document as this milestone forms part of that deliverable.
- **Deadline: 01/11/2024**

Milestone 3: Final Testing & Project Demonstration

Objective: Conduct final tests on the Robotic System ensuring that all project requirements are met. In addition, prepare for and conduct project demonstration in front of supervisors and captains of industry.

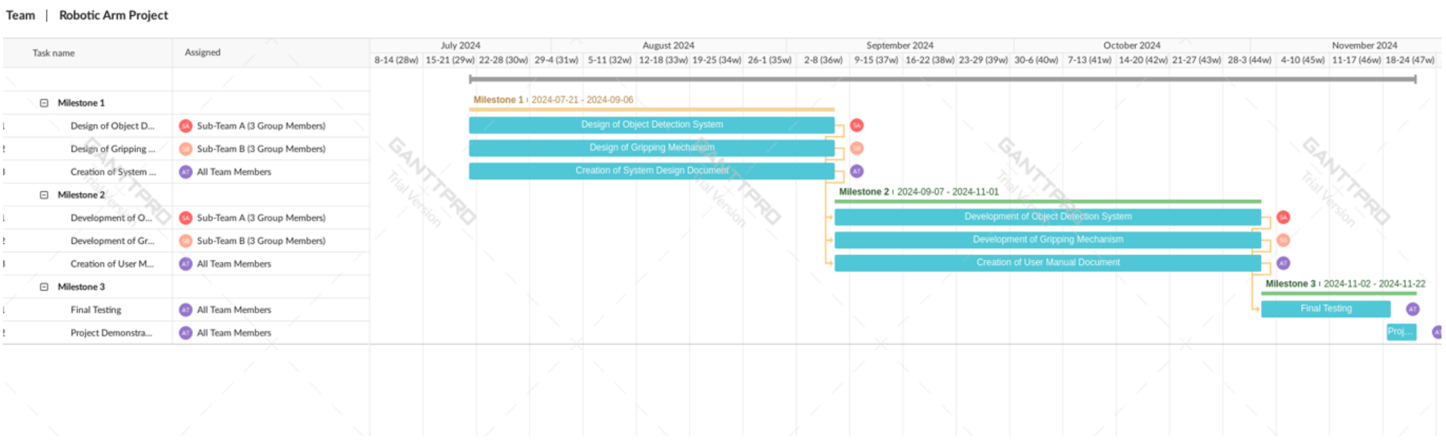
- Conduct final tests on the Robotic System and make corrections as and when necessary.
- Prepare system before demonstration (set robot system up at correct booth and ensure the area is clean and safe for spectators).
- Create and prepare any and all necessary documentation and presentations before demonstration.
- **Deadline: Week of 2024-11-18 to 2024-11-22**

Work Breakdown Structure (Table Form)

Robotic Arm Project

- **Milestone 1**
    - Part 1
        - Design Object Detection System
            - Research Computer Vision Technologies
            - Research Sorting Algorithms
            - Add to *System Design Document*
    - Part 2
        - Design Gripping Mechanism
            - Research Robotic Arm Attachments
            - Research into robot arm manipulation
            - Add to *System Design Document*
- **Milestone 2**
    - Part 1
        - Develop Object Detection System
            - Conduct Training and Research necessary to program Kuka Robot
            - Physically code the Object Detection System to specifications finalized in Milestone 1
            - Add to *User Manual Document*
    - Part 2
        - Develop Gripping Mechanism
            - Conduct Training and Research necessary to program Kuka Robot
            - Physically code the Gripping Mechanism to specifications finalized in Milestone 1
            - Add to *User Manual Document*
- **Milestone 3**
    - Conduct final tests on completed Robotic System
    - Prepare for project demonstration
    - Do *presentation/ demonstration of Robotic System*

Project Gannt Chart



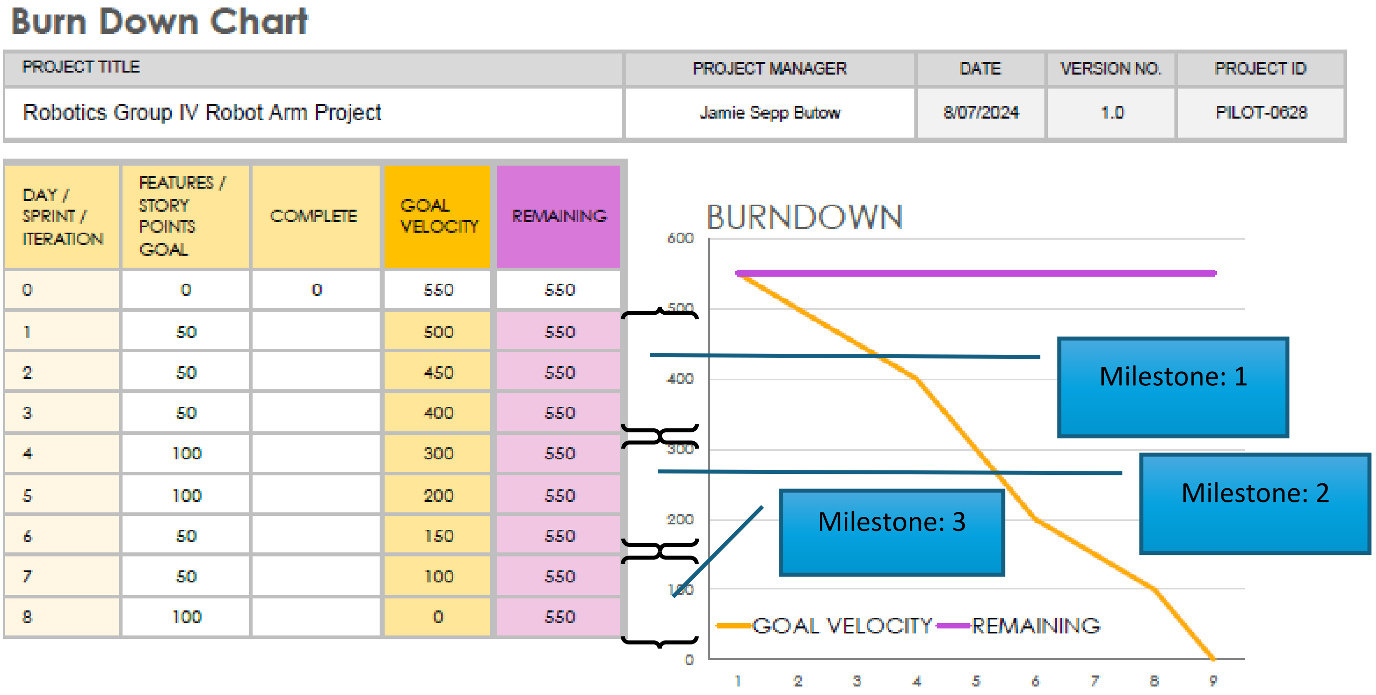
Team | Robotic Arm Project

Task name	Assigned
Milestone 1	
Design of Object D...	SA Sub-Team A (3 Group Members)
Design of Gripping ...	SB Sub-Team B (3 Group Members)
Creation of System ...	AT All Team Members
Milestone 2	
Development of O...	SA Sub-Team A (3 Group Members)
Development of Gr...	SB Sub-Team B (3 Group Members)
Creation of User M...	AT All Team Members
Milestone 3	
Final Testing	AT All Team Members
Project Demonstra...	AT All Team Members

July 2024
August 2024
September 2024
October 2024
November 2024
8-14 (28w) 15-21 (29w) 22-28 (30w) 29-4 (31w) 5-11 (32w) 12-18 (33w) 19-25 (34w) 26-1 (35w) 2-8 (36w) 9-15 (37w) 16-22 (38w) 23-29 (39w) 30-6 (40w) 7-13 (41w) 14-20 (42w) 21-27 (43w) 28-3 (44w) 4-10 (45w) 11-17 (46w) 18-24 (47w)
Milestone 1 | 2024-07-21 - 2024-09-06
Design of Object Detection System
Design of Gripping Mechanism
Creation of System Design Document
Milestone 2 | 2024-09-07 - 2024-11-01
Development of Object Detection System
Development of Gripping Mechanism
Creation of User Manual Document
Milestone 3 | 2024-11-02 - 2024-11-22
Final Testing
Proj...
SA
SB
AT
GANTTPRO
Trial Version

Figure 3: Gannt Chart of Robotic Arm Project

Project Burn Down Chart Template



Burn Down Chart

PROJECT TITLE	PROJECT MANAGER	DATE	VERSION NO.	PROJECT ID
Robotics Group IV Robot Arm Project	Jamie Sepp Butow	8/07/2024	1.0	PILOT-0628

DAY / SPRINT / ITERATION	FEATURES / STORY POINTS GOAL	COMPLETE	GOAL VELOCITY	REMAINING
0	0	0	550	550
1	50		500	550
2	50		450	550
3	50		400	550
4	100		300	550
5	100		200	550
6	50		150	550
7	50		100	550
8	100		0	550

Figure 4: Project Burn Down Chart Template

Methodology Selection

For the purposes of this project the agile methodologies of Scrum and Kanban (boards) will be used to help develop and deliver project objectives. This section will give an overview and explanation of these methodologies, the relevant roles, tools and techniques used as well as various templates as necessary.

Why scrum?

Scrum is incremental and iterative which means that it allows us to prototype, learn from experience, and receive feedback quickly which inevitably means we are not only able to develop features with a clear goal in mind but that we are also able to develop much more quickly.

Why Kanban?

Kanban helps us keep track of the development of the project. Using kanban boards lets us know which features need to be implemented, which features have already been implemented and which features are currently being implemented. Kanban in combination with scrum is great for planning and executing sprints. Helps with clear visualization.

Scrum

Overview of Scrum:

Scrum is a management framework that promotes self-organizing teams that work together towards a common goal. It specifies a set of tools, techniques and roles for efficient and effective project management. With scrum, much like a real sport match, teams work together, learn from experience and adapt to change as necessary (Amazon Web Services, 2024).

Explanation of key scrum roles, tools and activities:

Product backlog

These are the requirements or features that need to be implemented to deliver/ complete the project. These are the features the robot will have.

Sprint planning

Where the scrum master will decide which feature for the robot in the product backlog will be selected for the upcoming sprint.

Daily Standup

These are meetings (take place at the beginning of the day and are about 15 min) that take place daily during the sprint to discuss the progress of the current sprint, the events of the previous day as well as potential improvements based on what worked and did not work the previous day (continuous improvement).

Sprint Backlog

The selected user story items from the product backlog that the development team will tackle during a given sprint.

User Story

A single requirement/ feature for the software/ system written from the position of an end user.

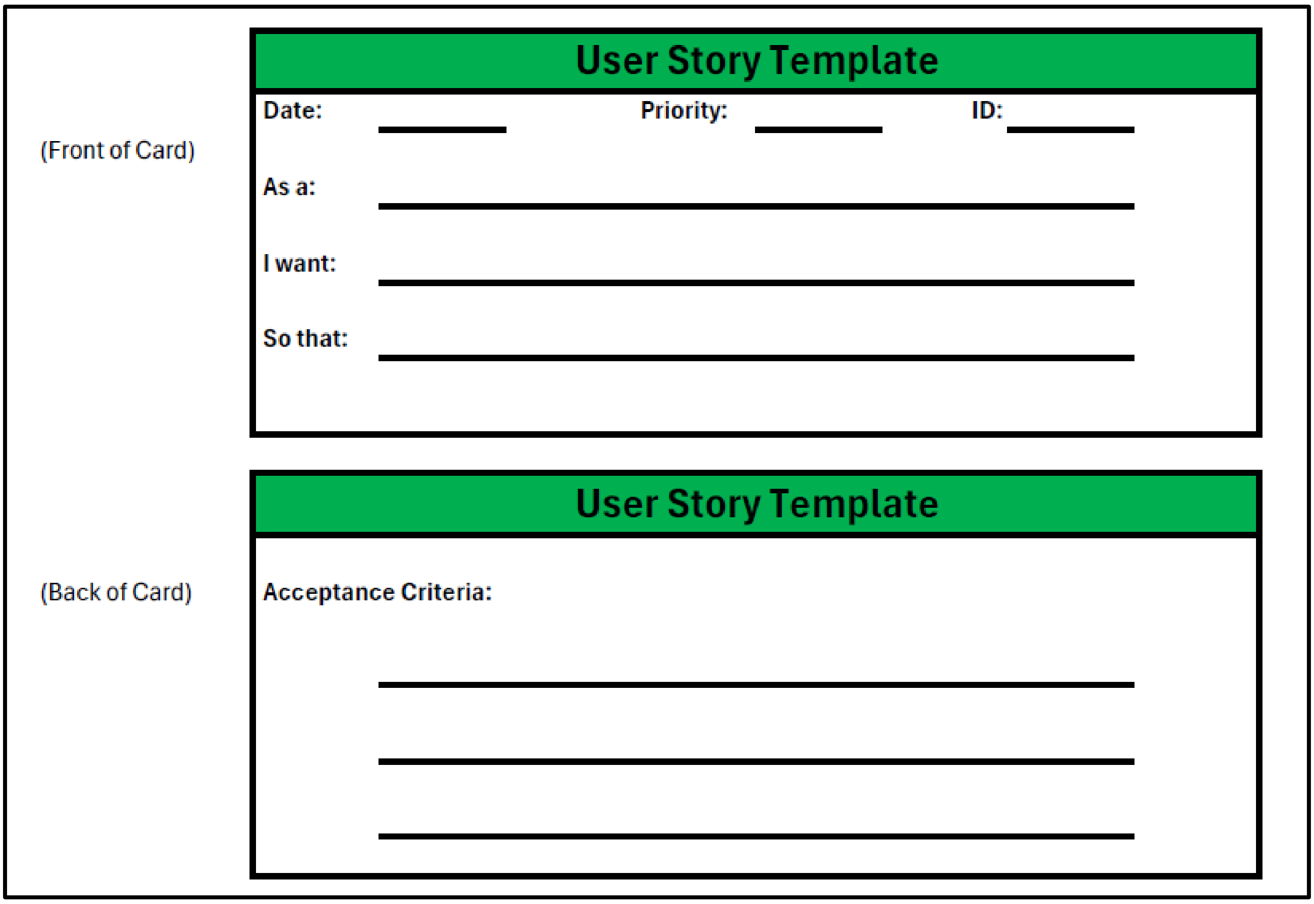
Stakeholders

- Project owner: Usually the customer. They are there to give feedback on the project.
- Scrum Master: Planner of events (selects items from the product backlog for the development team to tackle during the upcoming sprint).
- Development Team: in charge of implementing the software features of the robot (implementing user story items).

Scrum Templates:

User Story Template:

The following is the template that will be used to write user stories for this project. The card can be used virtually, but if printed should be printed on either side of a card.



(Front of Card)

User Story Template

Date: ________ Priority: ________ ID: ________

As a: ________________________

I want: ________________________

So that: ________________________

(Back of Card)

User Story Template

Acceptance Criteria:

____

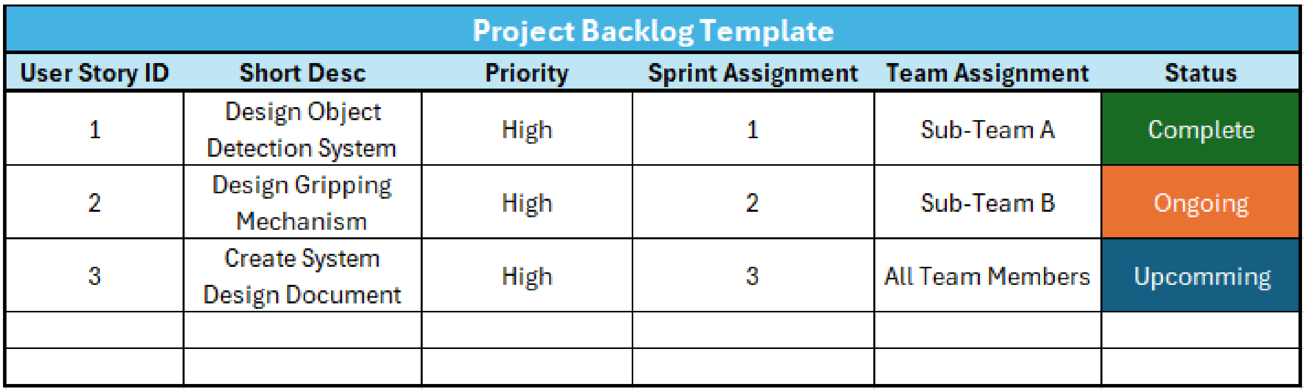
____

____

Figure 5: User Story Template

Project Backlog Template

The following is the template that will be used to keep track of the product backlog for this project. The User Story ID corresponds to the ID input on the User Story Template (see above). The template has been filled in with the backlog for Milestone 1 of the project (***Project Milestone NOT PRJ Marking Milestone***) for illustration purposes.



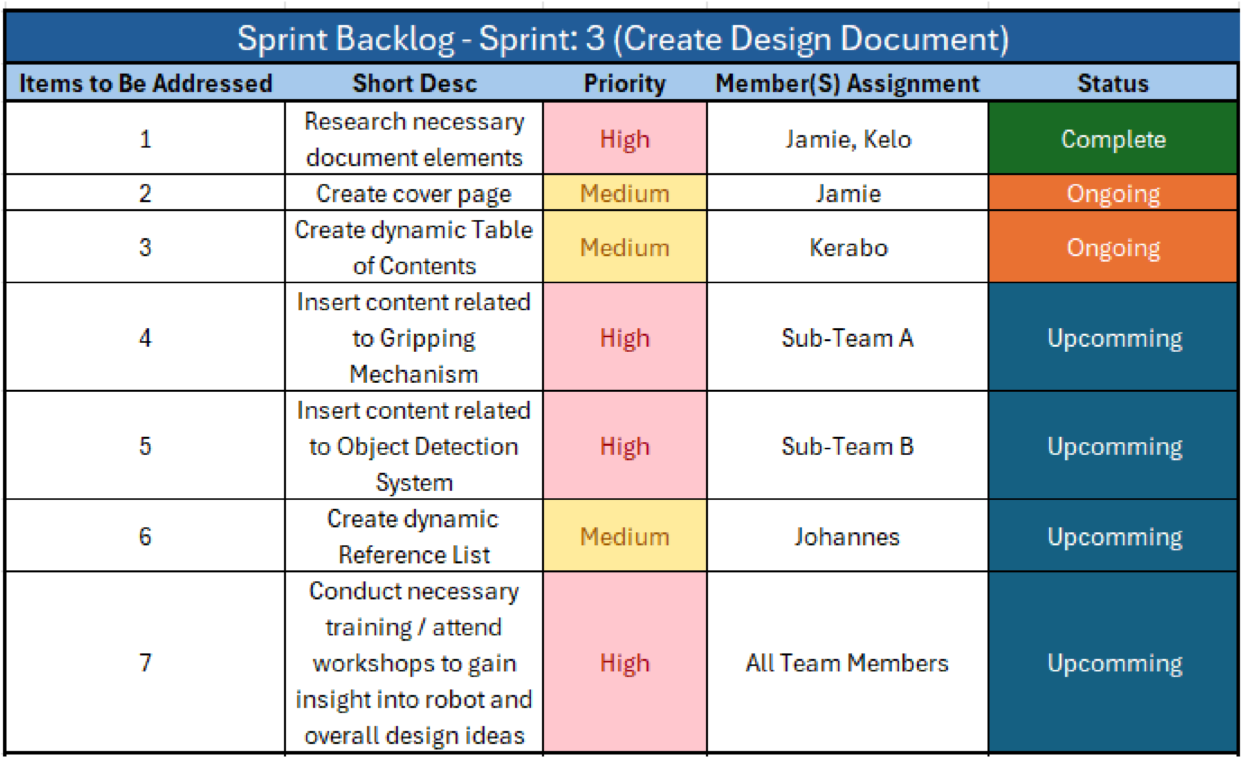
Project Backlog Template

User Story ID	Short Desc	Priority	Sprint Assignment	Team Assignment	Status
1	Design Object Detection System	High	1	Sub-Team A	Complete
2	Design Gripping Mechanism	High	2	Sub-Team B	Ongoing
3	Create System Design Document	High	3	All Team Members	Upcomming

Figure 6: Product Backlog Template

Sprint Backlog Template

The following is the template that will be used to keep track of the status of individual sprints. The template has been filled in with items related to User Story ID in the above example for illustration purposes. Although the project team will use Kanban Boards to further visualize sprint backlogs, an excel master file for each sprint like the example below will also be used.



Sprint Backlog - Sprint: 3 (Create Design Document)

Items to Be Addressed	Short Desc	Priority	Member(S) Assignment	Status
1	Research necessary document elements	High	Jamie, Kelo	Complete
2	Create cover page	Medium	Jamie	Ongoing
3	Create dynamic Table of Contents	Medium	Kerabo	Ongoing
4	Insert content related to Gripping Mechanism	High	Sub-Team A	Upcomming
5	Insert content related to Object Detection System	High	Sub-Team B	Upcomming
6	Create dynamic Reference List	Medium	Johannes	Upcomming
7	Conduct necessary training / attend workshops to gain insight into robot and overall design ideas	High	All Team Members	Upcomming

Figure 7: Sprint Backlog Template

Kanban

Overview of Kanban:

A visual tool/ method for managing workflow at the individual, team and even organizational level. It relies on the principle of firstly knowing where you currently are in order to get to where you would like to end up (Lynn, 2024).

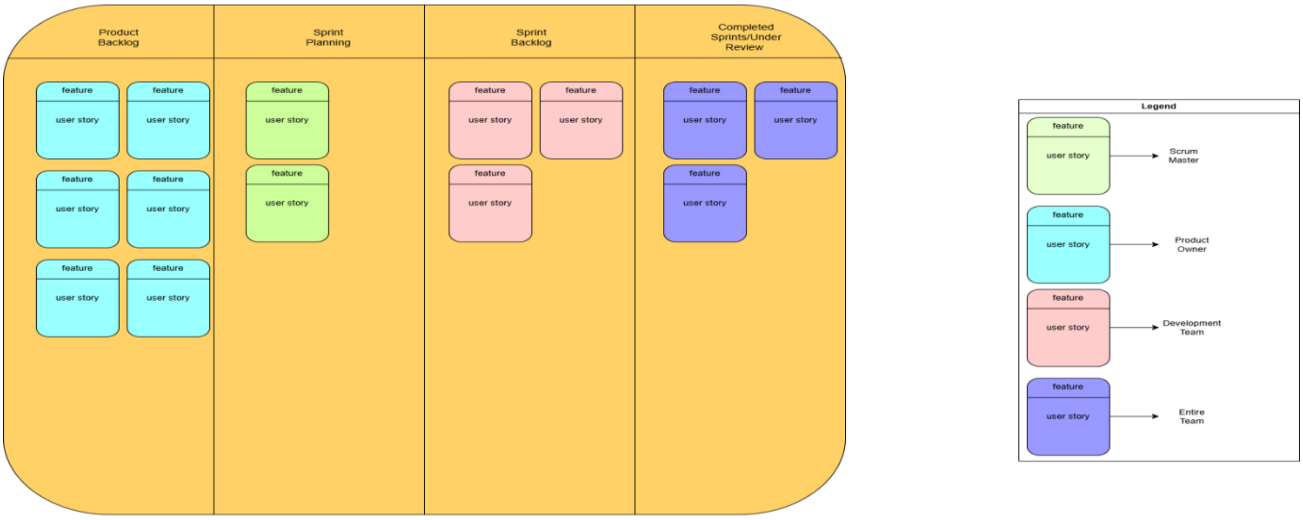
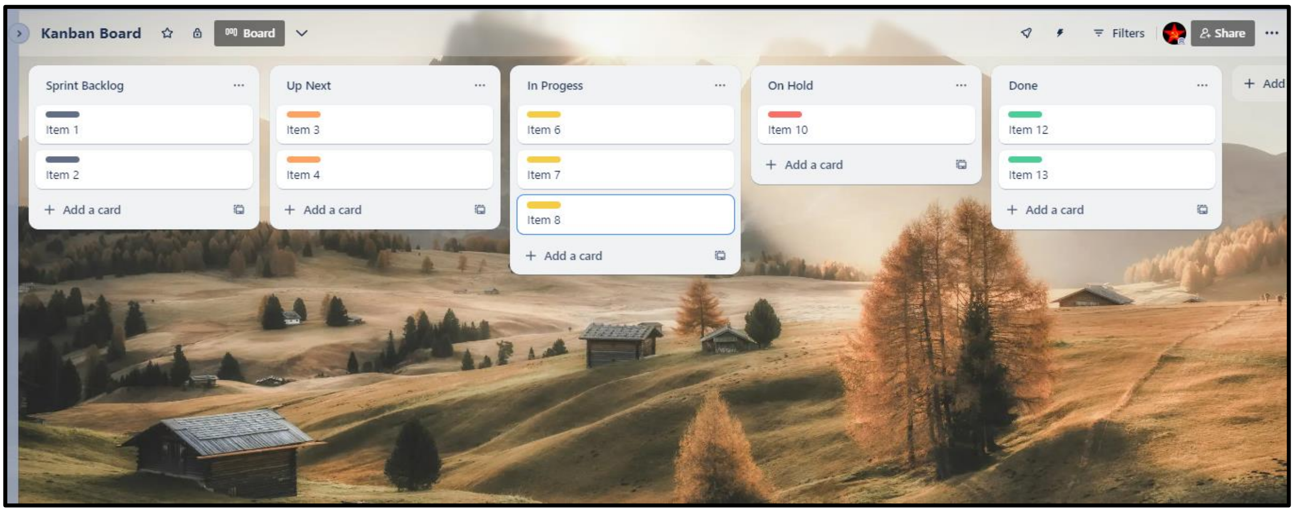


Figure 8: Example of a Kanban Board

This project will make use of a *virtual (software) implementation* of a Kanban Board in order for the team to better visualize the work (virtual Kanban Boards will be used to help the team visualize *sprint backlogs*). The software chosen for this will be a Trello board from Tello.com.

Justification:

- Provides cards that can be configured to contain all necessary information about a user story item.
- Up to 10 collaborators allowed on the free plan.
- Robust label system that will help the team to visualize the work further.



Sprint Backlog: Item 1, Item 2

Up Next: Item 3, Item 4

In Progess: Item 6, Item 7, Item 8

On Hold: Item 10

Done: Item 12, Item 13

Figure 9: Kanban Trello Board Example

Resource Management

Personnel Requirements

The project group consists of 6 members, each with unique roles and responsibilities:

Project Manager: [Jamie Butow]

Responsibilities:

- Supervise all project parts.
- Coordinate teamwork.
- Ensure completion of project milestones.
- Communicate with stakeholders.
- Manage the project timeline.
- Manage the project resources.

Sub-Team A – Gripping Mechanism: [Karabo Linala, Kelo Letsoalo, Jordan Barradas]

Responsibilities:

- Focus on mechanical and electrical compatibility of the system.
- Development of control software for the robotic arm.
- Compatibility of gripping mechanism with objects of different shapes and sizes.
- Integration of gripping mechanism with mechanical arm.
- Documentation of system design and implementation.
- Continuously review work of Team B.

Sub-Team B – Object Detection: [Johannes van Wyk, Kamohelo Mototo, Jamie Butow]

Responsibilities:

- Development and implementation of the object detection system.
- Development of the identification and classification techniques used by the system.
- Optimization of the detection system.
- Integration of different software components.
- Communication between detection systems and robotic arm.
- Documentation of system design and implementation.
- Continuously review work of Team A.

Supervisors: [Mr. A. Magaudini & Mr. G. Modimola]

The supervisors are not members of the project group but are assigned to provide guidance and quality assurance to the students and their work on the project.

Equipment Requirements

Robotic arm

- The group will make use of a robotic arm provided on the campus of Belgium Campus ITversity.

Components of gripping mechanism

- A three-finger gripper that will attach to the robotic arm. The three-finger variant is needed to adapt to objects of different shapes and sizes.

Cameras and sensors

- Used for object detection and classification.

Workstations

- Group members will use their personal computers for development of the software needed in the project.

Software

- Visual studio code: An Integrated development environment to support programming.
- OpenCV: A library for computer vision tasks. Needed to process and detect objects through the cameras and sensors.
- Microsoft Office Suite: Needed for documentation, analysis and communication.
- GitHub: Used for version control and teamwork.

Budget Requirements

Personnel Costs:

Personnel consists of the project group contributing time and effort as part of their educational curriculum. This results in no direct personnel costs being associated with their work.

Equipment Costs:

All necessary equipment needed for the project will be provided by the campus:

- Robotic Arm
- Gripping mechanism
- Vision hardware

- Other mechanical components (e.g. mounting brackets and cables)

Software Costs:

All software used are freeware, open source or provided by campus:

- Visual studio code – Free
- OpenCV – Free
- ROS – Free
- Microsoft Office Suite – Available under educational license

Miscellaneous Expenses:

- Calibration tools, test objects, and measurement tools – Estimated at R300
- Contingency Fund – R500

Risk Management

Risk Identification by Risk Category:

- **Technical Risks:**
  - Integration issues between the Grip Mechanism and the Object Detection System.
  - Challenges arising from the use of new software technologies (the use of unfamiliar programming languages such as KRL in programming the robot) and the use of new hardware technologies (interfacing with an unfamiliar robot arm).
  - Performance of robot is not expected/ insufficient to fulfil acceptance criteria.
- **Organizational and Resource Related Risks:**
  - (Hardware) Resource risks – robot arm is unavailable at Pretoria Campus or breaks down.
  - (Informational) Resource risks – lack of informational resources leading to slowed/ halted development.
  - (Software) Resource risks – breakdown of communication or other vital software during the development process leading to the halting of the project.
  - A breakdown in organization among the team leading to the hampering of project progress.
  - The requirements of the project being drastically changed by important stakeholders (i.e. Belgium Campus).
- **External / Environmental Risks:**
  - Sustained power outages which lead to the halting/ slowing down of project development.
  - Sustained internet outages which affect project team communication and thereby hinder project progress.
  - Mass (country wide) political or social upheaval causing the project to be put in jeopardy.
- **Project Management Risks:**
  - PEP (Project Execution Plan) is inadequate for the project leading to ambiguity, unrealistic expectations and or timelines leading to the project being put in jeopardy.
  - Unknown unknowns – risks that were not accounted for or foreseen occurring and delaying/ halting the project.

Qualitative Risk Analysis – Risk Matrix filled in with Identified Risks

Impact:

Rating	Potential Impact on Robotic Arm Project
5	**Catastrophic:** most objectives cannot be achieved, or several are severely affected
4	**Major:** one objective severely affected, or most objectives threatened
3	**Moderate:** some objectives affected, may take considerable effort to rectify
2	**Minor:** effects on project easily remedied with objectives still being able to be achieved
1	**Negligible:** very small impact, no action required to remedy

Likelihood:

Rating	
5	**Almost Certain:** will occur (perhaps several times in one year)
4	**Likely:** likely to happen at least once a year
3	**Possible:** reasonably likely it will occur
2	**Unlikely:** could occur but not very likely
1	**Rare:** very unlikely but not impossible

Risk Matrix:

Scale: (Green): low risk – lowest priority, (Yellow): medium risk - medium priority, (Red): high risk – highest priority.

	Low		Impact		High
High	🟨	🟨	🟥	🟥 PEP does not account for all eventualities	🟥
	🟩	🟨	🟨 Lack of informational resources	🟥 Challenges arise from the use of unfamiliar software and hardware technologies	🟥
Likelihood	🟩	🟨	🟨 Sustained Power Outage	🟨 Integration Issues between Gripping Mechanism and Detection System	🟥
	🟩	🟩	🟨 Sustained Internet Outage	🟨 Breakdown in Organization of Team	🟨 Robot becomes Unavailable or suffers from a Breakdown
Low	🟩	🟩 Mass Social Upheaval / Unrest	🟩 Breakdown in Communication and Documentation Software	🟩 Breakdown in Development Software	🟨 Performance of Robot is Unexpected, Change of Project requirements

Risk Mitigation Strategies by Risk Level

Risk Name/ Description	Risk Level	Mitigation Strategies
PEP does not account for all eventualities	**HIGH**	• Continually update project plan as project progresses and adapt as necessary, procedures, templates, methodologies and risk management strategies to deal with unknown unknowns as and when they appear
Challenges arise from use of unfamiliar technologies (both hardware and software)	**HIGH**	• Research thoroughly about technologies chosen by the team before implementing them in the project. Seek guidance from all informational resources including EBSCO, the internet and project supervisors.
Lack of informational resources	**MEDIUM**	• If one avenue of informational resources (EBSCO for instance) does not provide enough information, make use of alternatives (backup) information repositories found on the web (Udemy, Free Code Camp, Code Academy etc)
Sustained Power Outage	**MEDIUM**	• Make use of invertor to power internet and computer (for loss of power up to 2 days) • Make use of generator to power internet and computer (for loss of power more than 2 days)
Sustained Internet Outage	**MEDIUM**	• Make use of internet cafes to ensure project meetings and deadlines are not missed. • Make use of other internet connection alternatives (5G cell towers if fibre is lost for instance)
Integration Issues between Gripping Mechanism and Detection System	**MEDIUM**	• Research beforehand if chosen designs for respective elements of the project are compatible and continuously test integration between these elements during development and testing phases of the project
Robot becomes Unavailable or suffers from a Breakdown	**MEDIUM**	• Have contact with other robotic teams who are using the same robot model and have agreements with them in place so that if there is an issue with the robot, the team can make use of the other robot.
Performance of Robot is Unexpected / insufficient to fulfil project acceptance criteria	**MEDIUM**	• Create and strictly enforce acceptance criteria on every task and milestone so that the project always aligns with what is expected. Rigorously test project with a variety of objects to ensure it meets the requirements.

Risk Name/ Description	Risk Level	Mitigation Strategies
Breakdown in Organization of Team	**MEDIUM**	• Assign a project manager per phase, assign scrum master's for tasks, hold daily standups and meet with the entire development team and project supervisors regularly (bi monthly at minimum) in order to ensure the team is always organized.
Breakdown in Development Software	**LOW**	• Prepare backup IDEs in case of prolonged IDE software breakdown. For example, be prepared to use Visual Studio 2019 in case Visual Studio Code suffers a prolonged outage.
Breakdown in Communication and Word Processing Software	**LOW**	• Prepare backup communication and word processing software. For backup communication software use Discord as a stop gap in case of Microsoft Teams outage. For backup word processing software, install Free Office as backup.
Mass Social Upheaval / Unrest	**LOW**	• Continually monitor local and international news and take action as necessary. For example, if there are ongoing demonstrations/ riots near or along the route to campus, work from home or take alternative routes.

References

Amazon Web Services, 2024. *What is Scrum?.* [Online]
Available at: https://aws.amazon.com/what-is/scrum/
[Accessed 8 7 2024].

Kuka, 2024. *KUKA Robot Guide.* [Online]
Available at: https://www.kuka.com/en-gb/robot-guide
[Accessed 8 7 2024].

Lynn, R., 2024. *INTRODUCTION TO KANBAN.* [Online]
Available at: https://www.planview.com/resources/guide/introduction-to-kanban/
[Accessed 10 7 2024].