SOFTWARE REQUIREMENTS SPECIFICATION

for

YOLO Project

Serra Zehra Dane - 2380251
Hana Medhat Elboghdady - 2455566

April 15, 2022

Table of Contents

List of Figures

List of Tables

Revision History

Revision	Date	Author(s)	Description
1.0.0	08.04.2022	SZD , HME	SRS First Draft
1.0.1	15.04.2022	SZD , HME	SRS Final Bundle Release

1 Introduction

1.1 Purpose of the System

Since children generally learn through exploration and exposure, certain objects can help them in this process of understanding the world around them. Social robots have become these common objects as technology has evolved over time in the sense that has led to more interactive and digitised play-time. Lots of research has shown technology can be beneficial for children's learning and so YOLO, a non-anthropomorphic robot, has emerged as a project for the purpose of stimulating creativity in children. It is worth mentioning that in developed communities innovation and collaborative problem solving is being deemed superior to standard means of acquiring knowledge, and are setting a path towards a phenomenon called "creative economies".

1.2 Scope

This robot is called YOLO, short for Your Own Living Object In the scope of this system, the social robot should benefit children's creativity during playtime. Children can play with this robot while still maintaining some elements of traditional play for example could be an outdoor activity, since the robot is portable.

The specific sequence of events entails a storytelling activity in which children use the robot as a character for their stories. creativity is supposed to be stimulated during interaction with YOLO.

During the process of play, the robot provides stimuli for children to develop new story-lines for the stories they create. The robot does so by using techniques of creativity training. Particularly, it stimulates two core elements of the creative thought: divergent and convergent thinking, two modes of creative thinking that usually are naturally stimulated through play.
The main interaction elements of this system are:

- Lights and movement
- Abstract shape as imagination trigger
- Touch for shared control

YOLO can be used as a tool for research studies with children,
researchers can use it as a platform to explore the design of behaviors for a robot aimed at interacting with children.

Another usage of this robot by the social and cognitive sciences field as a controllable and programmable tool, to study the developmental aspects of children when interacting with robots YOLO as a research tool for STEAM activities aimed at promoting robotics knowledge among young children and adults

1.3 System Overview

1.3.1 System Perspective

YOLO Robot is device intended for the use of children to enhance and shape their creativity. YOLO becomes a character in the stories the children come up with during play time.The social robot makes use of creativity techniques that promote the creation of original story-lines. The preparation of the YOLO is simple for non-experts such as parents or educators. After starting up the robot child can interact with it in several ways including speaking to the robot, touching it and moving t around. In response to these actions YOLO can light its LEDs with a variety of colours and move around using its Omni wheels. The way it behaves varies depending on the technique used, like contrasting or mirroring, and at which stage of the story they are at. Also, developers can reach open-access code of the YOLO Robot and create new behaviors through new code additions.

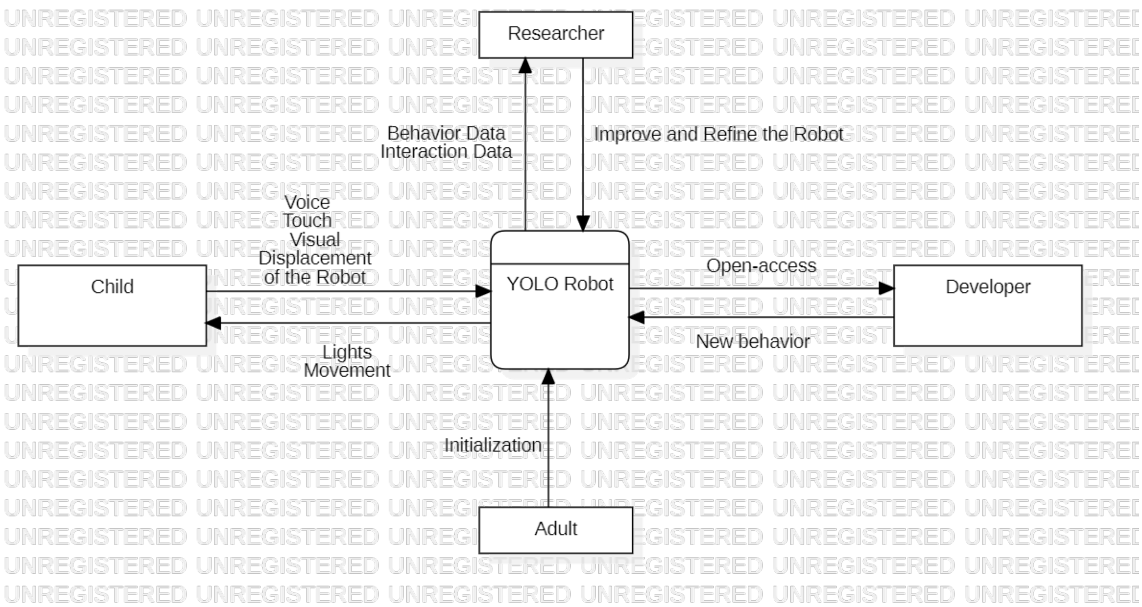

Figure 1.1: System Context Diagram for YOLO

1.3.1.1 System Interfaces

The YOLO robot consists of a combination of small physical and software parts. With the help of interfaces these parts communicate among themselves and with the user. The following are the interfaces through which interaction occurs:

- User interfaces
- Hardware interfaces
- Software interfaces
- Communication interfaces

Users interacts with the robot through physical input and interactive output as the user interface. Software interface enables any user the opportunity to design personalized behaviors for YOLO. The YOLO robot communicates with the physical tools such as RaspberryPi, using the hardware interface.

1.3.1.2 User Interfaces

In order to start using the robot, first a parent or an older figure with some programming background should set up the robot. They should first download Raspbian Stretch Lite OS image into a computer, install Raspbian onto the SD that should be connected to the computer and configure the raspberry pi. Then they should install the project code and some imported libraries in python, the language of choice in which the software was created. They should run an API use case from the examples file which is triggered by clicking on the touch sensor. If all goes well, there will be no error detected and that means the user can start experimenting with more of the examples in the directory to get familiar with the characters and behaviours.

1.3.1.3 Hardware Interfaces

First of all, the system needs a computer to install it on. Then hardware components needed to execute the system are a Micro SD Card with an SD adaptor with 16GB of storage, a Raspberry-Pi Zero W, a router ,micro USB charger and micro USB male to USB female. Ofcourse the raspberry pi needs to be connected to the WIFI to be able to work. Additional hardware in the system is within the robot itself such as the optical sensors which detect coordinates and touch sensors that detect physical contact. Also omni wheel actuators and LED actuators to provide the output and interaction with the user such as different movement and lights.

1.3.1.4 Software Interfaces

YOLO robot runs on python scripts for the main part executed on raspberry pi. Operating systems supporting the set up of YOLO include Windows, Linux and Mac. There are pre-sets already developed with this software, but it is possible to define new behaviors and make it fit personalized needs with the use of Python script-based language and Raspberry-Pi's specifications and the API available on the internet on github.

1.3.1.5 Communication Interfaces

In order to utilize the main functionalities of the YOLO Robot such as playing, moving etc. there needs to be a router, to connect to a WI-FI. Because the router range is wide, children can play with YOLO both indoors and outdoors. The performance of the robot is dependent on router range which is a point to keep in mind.

1.3.1.6 Memory Constraints

There aren't many memory constraints if the user will use the 16GB SD card, it will be enough.

1.3.1.7 Operations

Most of the operations performed by the YOLO system user-initiated but others are self initiated too.
User-initiated:

- Activation
- Displacement of the Robot
- Physical contact
- Communication interfaces
- Movement and shape recognition

Self-initiated:

- Time Behavior of the Robot
- Behavioral Mode Selection
- Mirroring Movement
- Contrasting Movement

Also developer operations:

- Acquiring open-source API

Researcher operations:

- Acquiring Behavior and Interaction Data

Details of these operations are extensively covered in Section 3.2.

1.3.2 System Functions

You can find the System Functions in Table 1.1.

Function	Summary
Activation	Adult accesses to the code and the guide to install and execute the software.
Displacement of the Robot	The child has the ability to move the robot around to add to the effect of story telling.
Physical Contact	The robot displays white lights while being touched, refrains from performing any behavior. When not sensing touch, the robot displays colors associated with its different social behaviors.
Time Behavior of the Robot	YOLO Robot can monitor stages of the story child created and process its changes. If there is no change it can initiate a movement and contribute to the story.
Changing LED Lights	Depending on the current behaviour selected, the robot flashes led lights in different colours, animations and brightness levels. the three modes are exuberant, aloof and harmonious.
Movement and Shape Recognition	The YOLO robot senses how children move it and recognize the manipulation patterns of children while grabbing with its movement and shape sensors.
Behavioral Mode Selection	The YOLO robot senses how children move it and recognize the manipulation patterns of children while grabbing with its movement and shape sensors.
Mirroring Movement	When child moves the YOLO Robot, it can imitate the previous movement made by children thus elaborating on a given story-line. This technique is used to stimulate convergent thinking.
Contrasting Movement	When child moves the YOLO Robot, it can perform a different movement, setting an intention to change the course of the story. This technique is used to stimulate divergent thinking.
Acquire Open-Source API	A primary function of this software is to serve as an Application Programming Interface that enables any user the opportunity to design personalized behaviors for YOLO, consequently providing the possibility to generate new behaviors and interaction modes.
Acquire Behavior and Interaction Data	YOLO has the potential to be used as a research tool for academic studies, and as a toy for the community to engage in personal fabrication. A researcher can acquire the behavior of the robot and the child and child-robot interaction data to improve the robot in order to be better in providing creativity in children.

Table 1.1: System Functions

1.3.3 Stakeholder Characteristics

There are three main users of the YOLO Project which are children, parents/teachers, and programmers.

Children are the ones who creates a story and give a character to the YOLO Robot according to the flow of the story. They are required to move the YOLO Robot in order the robot to contribute their story and display behaviors, movements, lights. Therefore, children just need to have the ability to play with the YOLO Robot.

Parents/Teachers are in charge of helping children and activating the YOLO Robot. They need to have good communication skills, know the system in detail and be able to install the software to activate the YOLO Robot.

Programmers are responsible for creating new behaviors to the YOLO robot by writing code. Therefore, they should have high computer and coding skills.

Researchers are the ones who improve and refine the robot that is why they need to have research abilities and background knowledge about child psychology and human-robot interaction.

1.3.4 Limitations

- **Regulatory Policies:** The YOLO Project is an open source hardware and software project. Therefore the project files and codes are accessible for everyone.
- **Hardware Limitations:** The YOLO Project detect the shape and duration of the movement which child creates by playing the YOLO Robot with its optical and time sensor. Therefore, these hardware devices shall send data to the system without any delay. For users, fully built YOLO Robot and software installed in is enough.
- **Interfaces to Other Applications:** The YOLO Project shall be compatible with hardware management system and operating systems.
- **Parallel Operation:** Parallelization takes an important role in YOLO Project because a single child can play with multiple YOLO Robots at the same time and multiple children can play with one YOLO Robot. Therefore multiple YOLO Robots is able to function concurrently.
- **Audit Functions:** YOLO Project does not have any audit functions.
- **Control Functions:** YOLO Project is controlled not by a center but by every user therefore control functions does not exist.
- **Higher-order Language Requirements:** The main system is written in Python, Git version controlling system is used and sensor components are written in microprocessor specific assembly.

- **Signal Handshake Protocols:** YOLO Project does not have any signal handshake protocols.
- **Quality Requirements:** Stability, maintainability and ease-of-usage are important priorities for YOLO Robot. The robot shall be easily installed and used globally.
- **Criticality of the Application:** The YOLO Project is not a critical system. It won't have huge effects if it fails.
- **Safety and Security Considerations:** The app is open source therefore anyone who knows how to read code can detect vulnerabilities however it does not harm anybody since every connection is establishments are not centralized.
- **Physical/Mental Considerations:** Any child can play with a traditional toy can also play with YOLO Robot.

1.4 Definitions

- YOLO, your own living object
- KNN, k-nearest neighbor
- ML, machine learning
- LED, light-emitting diode
- API, application programming interface
- App, application
- STL, standard tessellation language
- CC, creative commons

2 References

[1] Alves-Oliveira, P., Arriaga, P., Paiva, A., & Hoffman, G. (2019). Guide to build YOLO, a creativity-stimulating robot for children. HardwareX, 6, e00074. https://doi.org/10.1016/j.ohx.2019.e00074

[2] Alves-Oliveira, P., Arriaga, P., Paiva, A., & Hoffman, G. (2021). Children as Robot Designers. Proceedings of the 2021 ACM/IEEE International Conference on Human-Robot Interaction. https://doi.org/10.1145/3434073.3444650

[3] Alves-Oliveira, P., Gomes, S., Chandak, A., Arriaga, P., Hoffman, G., & Paiva, A. (2020). Software architecture for YOLO, a creativity-stimulating robot. SoftwareX, 11, 100461. https://doi.org/10.1016/j.softx.2020.100461

3 Specific Requirements

3.1 External Interfaces

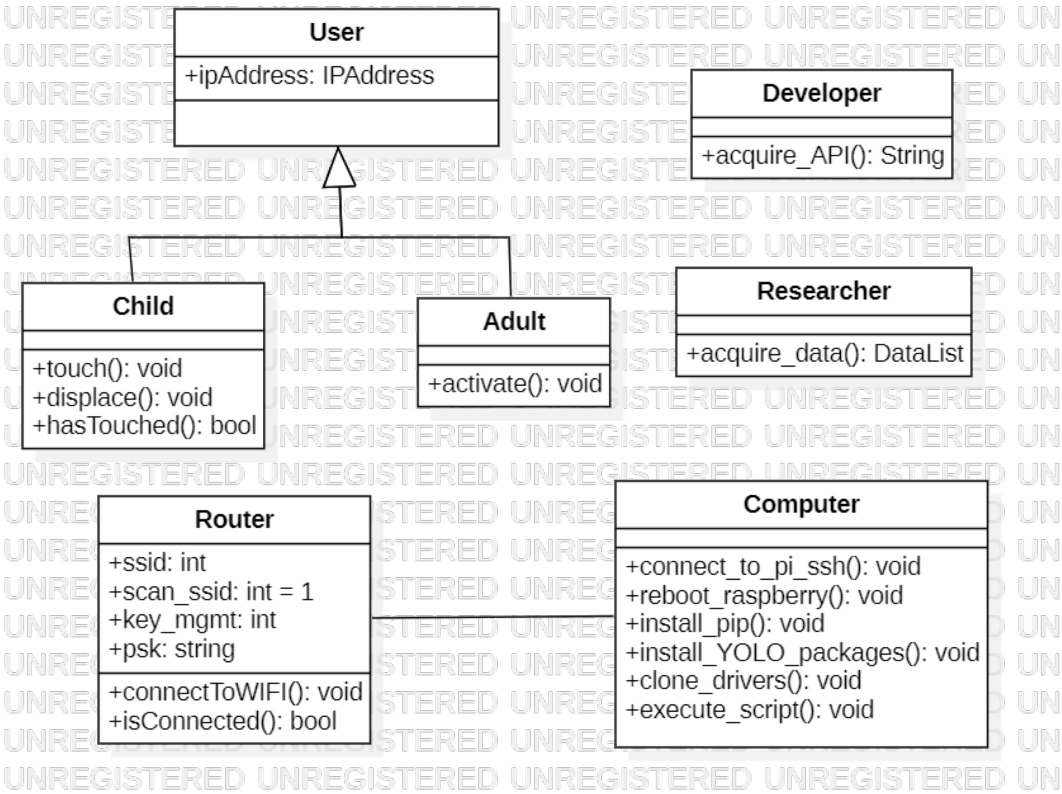

Figure 3.1: External Interfaces Class Diagram

3.2 Functions

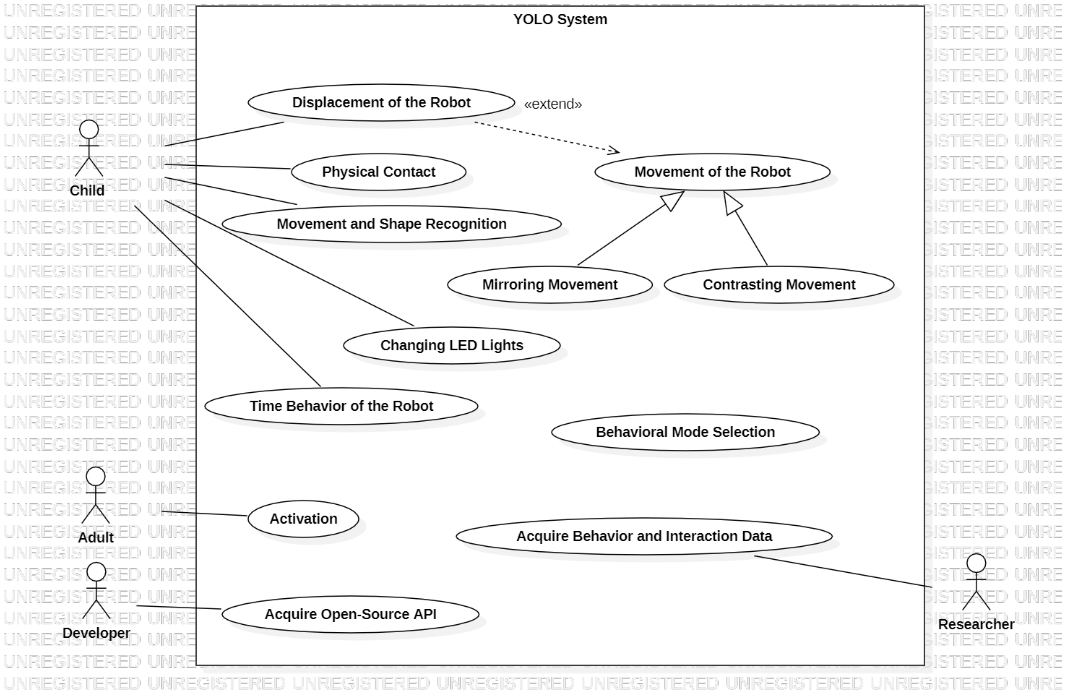

Figure 3.2: Use Case Diagram for YOLO

Use-Case Name	Activation
Actors	Adult
Description	Adult accesses to the code and the guide to install and execute the software.
Data	-
Preconditions	-
Stimulus	-
Basic Flow	**Step 1** - An adult access the code and the guide to install and execute the software **Step 2** - Adult follows the instructions given in the guide **Step 3** - Activation of the YOLO System is done and ready to play
Alternative Flow #1	-
Alternative Flow #2	-
Exception Flow	-
Post Conditions	Child starts to create a story.

Table 3.1: Activation Function

Use-Case Name	Displacement of the robot
Actors	Child
Description	The child has the ability to move the robot around to add to the effect of story telling.
Data	Displacement of robot
Preconditions	-
Stimulus	Child shall displace the robot
Basic Flow	**Step 1** - A child starts playing with YOLO Robot **Step 2** - Child moves the robot **Step 3** - Robot doesn't move on it's own letting the child control the story flow **Step 4** - Release of robot may set off new reactions/movements of the robot on its own without the child being involved physically **Step 5** - The robot continues with Mirroring Movement
Alternative Flow #1	**Step 5** - The robot continues with Contrasting Movement
Alternative Flow #2	-
Exception Flow	-
Post Conditions	-

Table 3.2: Displacement of the Robot Function

Use-Case Name	Physical Contact
Actors	Child
Description	The robot displays white lights while being touched, refrains from performing any behavior. When not sensing touch, the robot displays colors associated with its different social behaviors.
Data	Sensor input
Preconditions	-
Stimulus	Child should touch the robot
Basic Flow	**Step 1** - A child approaches YOLO Robot **Step 2** - Child touches the robot **Step 3** - Child may move the robot or not **Step 4** - YOLO will display white lights. **Step 5** - YOLO Robot doesn't move until physical contact is broken
Alternative Flow #1	-
Alternative Flow #2	-
Exception Flow	**Step 2** - Child does not touch the robot **Step 3** - The robot will display colors depending on the social behaviour currently being executed.
Post Conditions	-

Table 3.3: Physical Contact Function

Use-Case Name	Time Behaviour of the Robot
Actors	Child
Description	YOLO Robot can monitor stages of the story child created and process its changes. If there is no change it can initiate a movement and contribute to the story.
Data	The stage of storytelling
Preconditions	-
Stimulus	Child might displace the robot
Basic Flow	**Step 1** - During playtime, Robot monitors stage of story **Step 2** - Child changes arc of story **Step 3** - YOLO processes stage change **Step 4** - YOLO moves OMNI wheels and flashes LED lights as a form of creativity technique. **Step 5** - Repeat
Alternative Flow #1	**Step 2** Child doesn't change arc **Step 3** - YOLO responds to no change by initiating some action to get the story going.
Alternative Flow #2	-
Exception Flow	-
Post Conditions	-

Table 3.4: Time Behaviour of the Robot Function

Use-Case Name	Changing LED lights
Actors	Child
Description	Depending on the current behaviour selected, the robot flashes led lights in different colours, animations and brightness levels. the three modes are exuberant, aloof and harmonious
Data	Current behaviour mode, shapes and motion of child
Preconditions	-
Stimulus	Child drawing shapes and moving the YOLO Robot
Basic Flow	**Step 1** - A child starts playing with YOLO Robot **Step 2** - Robot processes shapes and movement **Step 3** - Robot integrates processed information with current behaviour mode **Step 4** - If exuberant, vibrant purple and red colors with high brightness levels are displayed.
Alternative Flow #1	**Step 4** - If aloof, cold colors such as green and blue with low brightness levels are displayed.
Alternative Flow #2	**Step 4** - If harmonious, yellow and orange at medium brightness levels are displayed.
Exception Flow	**Step 1** - If there is not any touch white color is displayed.
Post Conditions	The story continues as child wants

Table 3.5: Changing LED lights Function

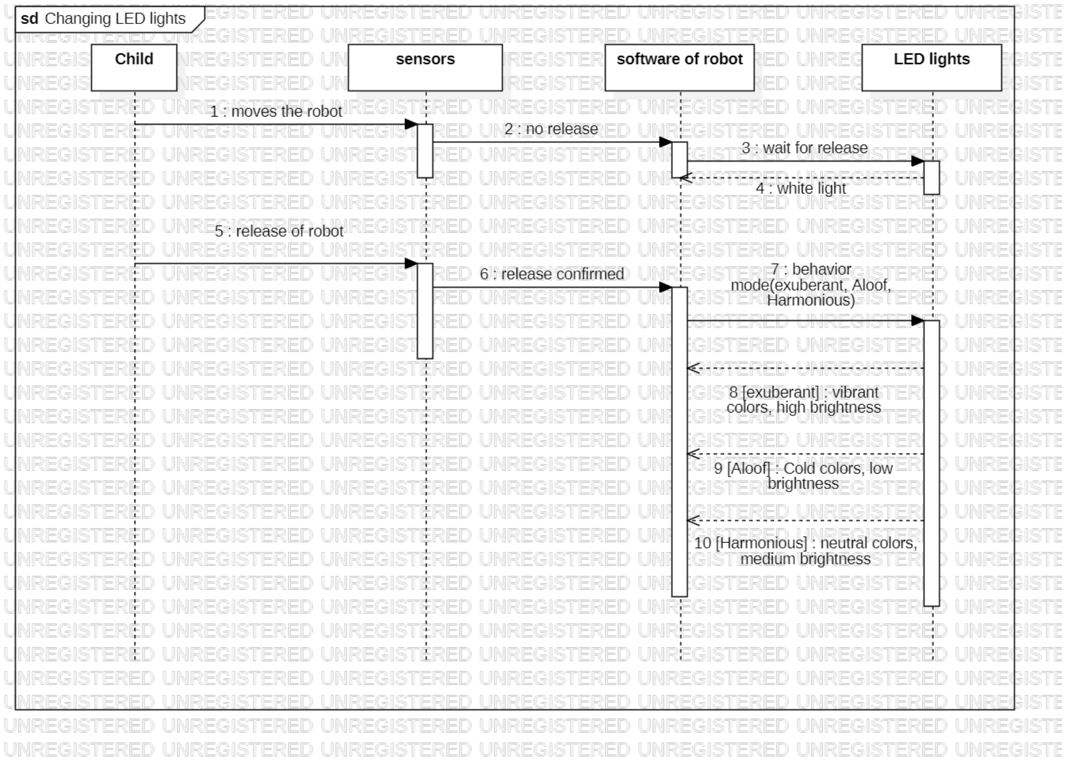

Figure 3.3: Sequence Diagram for Changing LED Lights Function

Use-Case Name	Movement and Shape Recognition
Actors	Child
Description	The YOLO robot senses how children move it and recognize the manipulation patterns of children while grabbing with its movement and shape sensors.
Data	Recognized direction of the play patterns of children while manipulating the robot.
Preconditions	The motion and shape sensors should work.
Stimulus	Child should have displace the robot.
Basic Flow	**Step 1** - A child starts playing with YOLO Robot **Step 2** - Child gives a role to YOLO within the story that s/he creates **Step 3** - Child moves the robot in any direction that visualize the story **Step 4** - YOLO Robot find rectangle, circle or triangle bounds using convex hull **Step 5** - YOLO Robot generate feature vector. **Step 6** - YOLO Robot runs the feature vector through pretrained ML KNN model in **Step 7** - YOLO Robot recognizes the pattern of the movement
Alternative Flow #1	-
Alternative Flow #2	-
Exception Flow	**Step 3** - Child does not move the YOLO **Step 4** - YOLO, without being previously manipulated by children, starts moving around to call their attention for playing.
Post Conditions	The story continues as child wants

Table 3.6: Movement and Shape Recognition Function

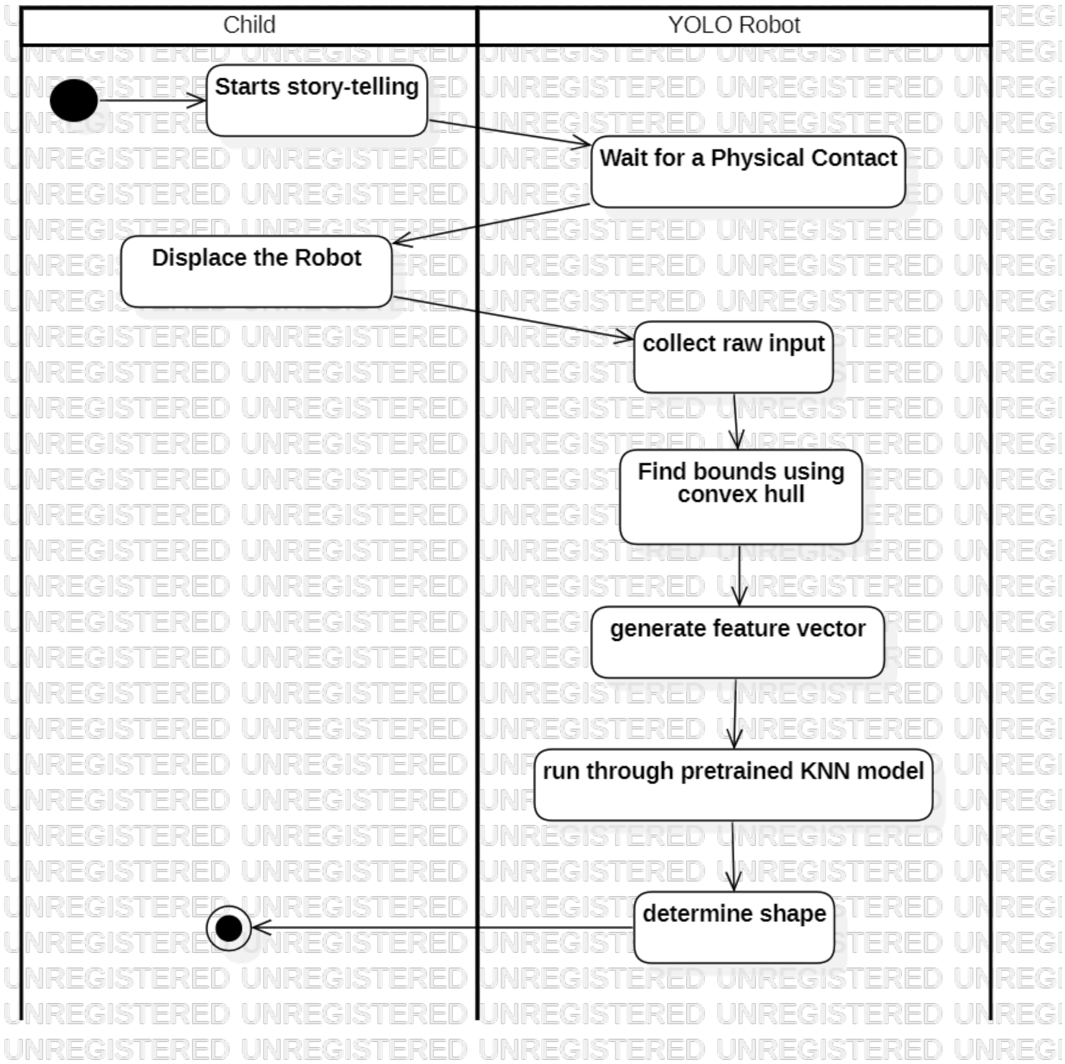

Figure 3.4: Activity Diagram for Movement and Shape Recognition Function

Use-Case Name	Behavioral Mode Selection
Actors	-
Description	YOLO expresses different social profiles to exhibit social behaviors. The profiles are named Exuberant, Aloof, and Harmonious. These social behaviors appear as presets when YOLO is turned on and can be used interchangeably, making the robot a flexible character in the children's stories.
Data	Information from the environment
Preconditions	-
Stimulus	-
Basic Flow	**Step 1** - A child starts playing with YOLO Robot **Step 2** - Child gives a role to YOLO within the story that s/he creates **Step 3** - YOLO System choose between three modes according to the story. **Step 4** - If the mode is exuberant, YOLO reacts to every social interaction in an "enthusiastic" manner. Movements are fast and have a high amplitude.
Alternative Flow #1	**Step 4** - If the mode is aloof, YOLO is less "socially reactive" and is a "shy robot". In this mode, the robot exhibits low amplitude, slow movements.
Alternative Flow #2	**Step 4** - If the mode is harmonious, YOLO acts in a moderated fashion, presenting behaviors that are in-between the extreme versions of Exuberant and Aloof. As Harmonious, YOLO exhibits medium speed, movements with medium amplitude.
Exception Flow	**Step 3** - Child does not move the YOLO **Step 4** - YOLO, without being previously manipulated by children, starts moving around to call their attention for playing.
Post Conditions	The story continues as child wants

Table 3.7: Behavioral Mode Selection Function

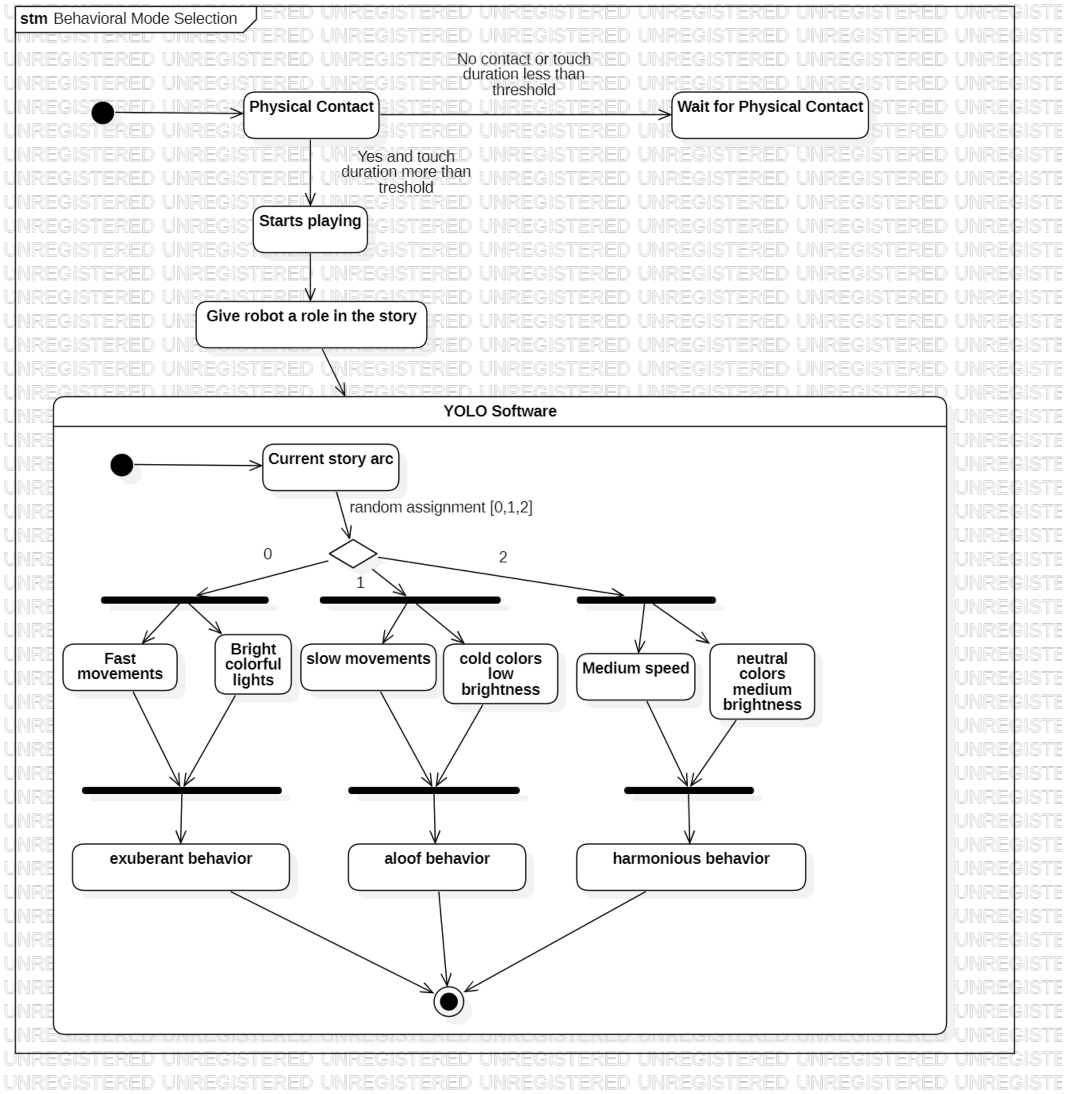

Figure 3.5: State Diagram for Behavioral Mode Selection Function

Use-Case Name	Mirroring Movement
Actors	Child
Description	When child moves the YOLO Robot, it can imitate the previous movement made by children thus elaborating on a given story-line. This technique is used to stimulate convergent thinking.
Data	The pattern of the child's move
Preconditions	-
Stimulus	Child should have displace the robot
Basic Flow	**Step 1** - A child starts playing with YOLO Robot **Step 2** - Child gives a role to YOLO within the story that s/he creates **Step 3** - Child moves the robot in any direction that visualize the story **Step 4** - YOLO System choose to move in a mirroring manner. **Step 5** - YOLO Robot recognizes the shape using its ML model **Step 6** - YOLO Robot imitates the previous movement made by children
Alternative Flow #1	-
Alternative Flow #2	-
Exception Flow	**Step 3** - Child does not move the YOLO **Step 4** - YOLO, without being previously manipulated by children, starts moving around to call their attention for playing.
Post Conditions	The story continues as child wants

Table 3.8: Mirroring Movement Function

Use-Case Name	Contrasting Movement
Actors	Child
Description	When child moves the YOLO Robot, it can perform a different movement, setting an intention to change the course of the story. This technique is used to stimulate divergent thinking.
Data	-
Preconditions	-
Stimulus	Child should have displace the robot
Basic Flow	**Step 1** - A child starts playing with YOLO Robot **Step 2** - Child gives a role to YOLO within the story that s/he creates **Step 3** - Child moves the robot in any direction that visualize the story **Step 4** - YOLO System choose to move in a contrasting manner. **Step 5** - YOLO Robot performs a different movement
Alternative Flow #1	-
Alternative Flow #2	-
Exception Flow	**Step 3** - Child does not move the YOLO **Step 4** - YOLO, without being previously manipulated by children, starts moving around to call their attention for playing.
Post Conditions	The story continues as child wants

Table 3.9: Contrasting Movement Function

Use-Case Name	Acquire Open-Source API
Actors	Developer
Description	A primary function of this software is to serve as an Application Programming Interface that enables any user the opportunity to design personalized behaviors for YOLO, consequently providing the possibility to generate new behaviors and interaction modes
Data	open source code
Preconditions	-
Stimulus	-
Basic Flow	**Step 1** - Software is open source **Step 2** - Developer can find the code and use the API **Step 3** - Developer can create new behaviours and functionalities for the robot **Step 4** - New behaviours are shared child users can play with them
Alternative Flow #1	**Step 4** - new behaviours used to study child-robot interaction
Alternative Flow #2	-
Exception Flow	-
Post Conditions	new additions to YOLO project

Table 3.10: Acquire Open-Source API Function

Use-Case Name	Acquire Behavior and Interaction Data
Actors	Researcher
Description	YOLO has the potential to be used as a research tool for academic studies, and as a toy for the community to engage in personal fabrication. A researcher can acquire the behavior of the robot and the child and child-robot interaction data to improve the robot in order to be better in providing creativity in children.
Data	Behavior and interaction data
Preconditions	-
Stimulus	-
Basic Flow	**Step 1** - A researcher arranges interviews and observation **Step 2** - A researcher involve children in the design of the social behaviors during story-telling. **Step 3** - A researcher collect behavioral data via directly observing children while playing **Step 4** - Behavior observation and analysis is done with the experimental study with children as users.
Alternative Flow #1	-
Alternative Flow #2	-
Exception Flow	-
Post Conditions	Researcher improve and refine the system

Table 3.11: Acquire Behavior and Interaction Data Function

3.3 Usability Requirements

- An adult shall install and download project without heavy tech-field knowledge.
- A child shall play with YOLO Robot that enables his/her to create a story by simply moving it.
- The YOLO Robot shall immediately response the child with moving its omni wheels and lighting its lights.
- The YOLO system shall correctly choose between its moods as defined like exuberant, aloof and harmonious.
- A developer shall easily access the open-source software and hardware codes and create new behaviors.
- An adult shall reach the design files are in STL format to 3D print the YOLO robot.

3.4 Performance Requirements

- The YOLO Robot shall be played by multiple children.
- The YOLO System shall perform interactively.
- The YOLO Robot shall boost creativity and originality in users.
- The YOLO Robot shall not move when being held by child.
- The robot motion sensors capture coordinates in 3 s intervals.
- The current ML model, that was trained with the physical robot, shall recognize with an 80% success rate the following shapes: circle, rectangles, loops, curls, spikes, and a straight line.

3.5 Logical Database Requirements

- The YOLO Robot has a touch sensor which has the ability to recognize when the robot is being touched.
- The YOLO Robot has an optical sensor which recognizes the direction of the play patterns of children while manipulating the robot.
- The YOLO Robot has omni wheel actuator that helps it to move around according to its behavior and the arc of the story.
- The YOLO Robot has LED actuator so that it can display different colors of lights and several brightness levels according to its current mode.

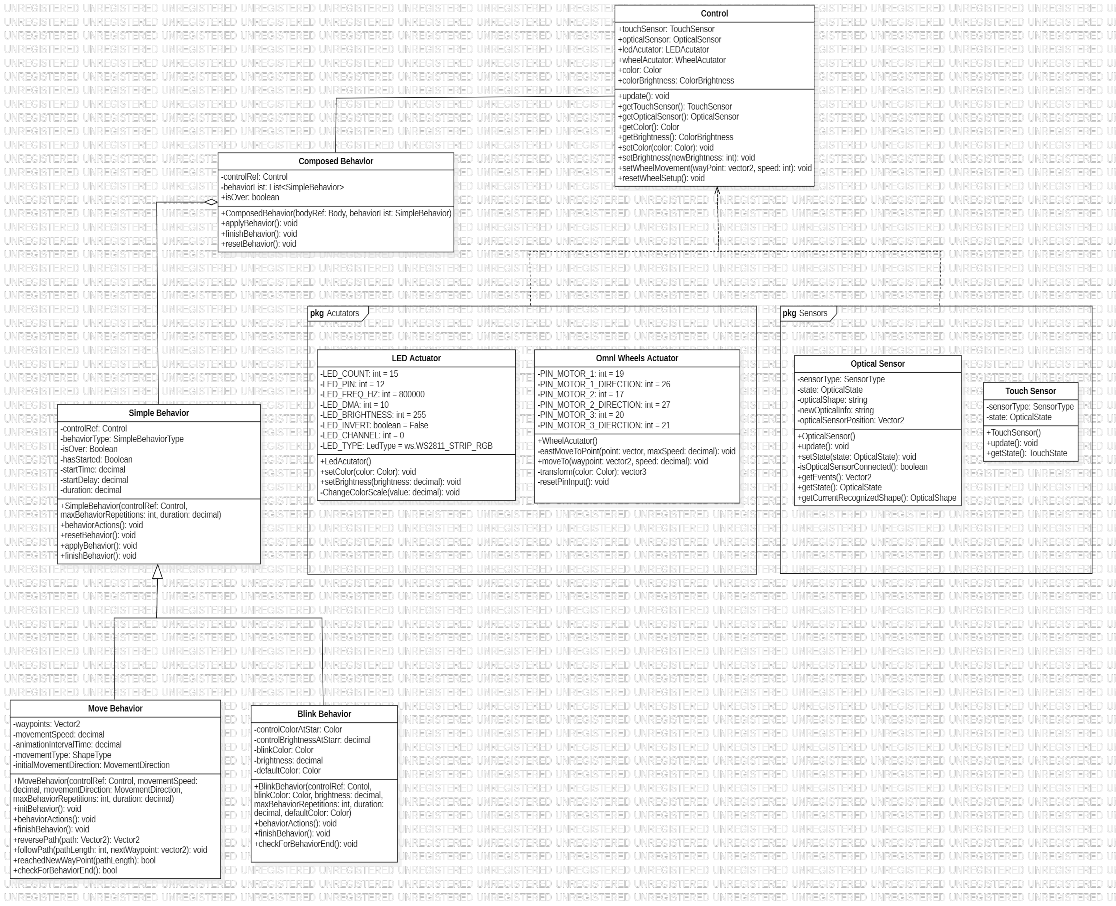

Figure 3.6: Logical Database Requirements Class Diagram

- There are several behaviors of the robot. One of them is the composed behavior. It is the composition of several simple behaviors that are displayed. Composed behaviors can be used to define the social behaviors which YOLO exhibits, such as Exuberant, Aloof, and Harmonious.

- Another one is the simple behaviors. These behaviors consist of assigning different light behaviors (different colors, animations, and brightness) to different movement configurations (different movement patterns at varying speed).

- The Simple behaviors also separated into two different behaviors. One of them is move behavior which includes the path and the shape of the movement, other one is the blink behavior that includes the color and brightness info of the LEDs.

- With those behaviors, actuators and sensors the YOLO Robot can have the control of the system.

3.6 Design Constraints

Open source hardware designs and open source software development methods are chosen in order to serve users in a cheap and free way and licensed under CC Attribution 4.0 International. The design process has been carried out in accordance with The Code of Ethics of the World Medical Association (Declaration of Helsinki) for experiments involving humans and informed consent was obtained for experimentation with human subjects

3.7 System Attributes

3.7.1 Reliability

- For full performance, YOLO's batteries need to be properly charged.
- Performance also depends on the distance between the robot and the router.
- Small-scale and light-weight robot meant for children's hands' size and easy manipulation

3.7.2 Availability

- Downloading an available version of the software with original pre-sets that have been developed.
- The average battery life is between 5 and 7h though it fluctuate depending on the playing behavior of children.
- Children can play with YOLO both indoors and outdoors and the router range is wide. could need a smaller distance between the robot and the router if it becomes unresponsive.

3.7.3 Security

- The robot's interior components (such as screws and standoffs) are made of nylon to avoid shorts between electrical boards.
- The circuitry and electronic boards were assembled in a compact and robust layered design in order to be safely manipulated by children.

3.7.4 Maintainability

- YOLO's software can be developed and personalized according to the needs and goals of the developers, i.e software is always up to date.

3.7.5 Portability

- The robot has a robust internal power system, providing energy to all internal components.
- It is independent of power outlet, so it can move around easily.

- Light-weight meaning it can be displaced easily

3.8 Supporting Information

- The YOLO Project aims to stimulate creativity in children.
- If one or more omni-wheels start not to move, substitute the 9V battery, as there might be a power shortage. Therefore, if the robot is non-responsive, recharge the power bank and try again when it is full.
- If YOLO continues non-responsive, check the wiring connections as they might need extra soldering as the unrestricted movements during children's play can weaken the connections.
- There are no major hazards when operating and playing with YOLO, however, like any other technological toy, children should be supervised by an adult at all times.
- A responsible adult should be in charge of initializing YOLO.

4 Suggestions to Improve the Existing System

1. The articles of the YOLO Project mention that the YOLO Robot randomly choose between its behaviors. In order to improve the existing system one should update the code that the behavior of the robot is chosen by a machine learning algorithm.

2. The YOLO Robot could also have a microphone or try to detect sound in some way, that can help take the accuracy of the responses to a higher level. That entails having ML for sound detection and processing to follow up with the story.

3. The experience could be more personalised for example the robot should be able to store and remember the name of the user, or users, who will play. This will engage the child more.

4. Although YOLO Robot is a complex project there are only two outputs of the Robot; movements and lights. There could be more different actuators in order to make it more interactive.

5. It is possible to have also a black version of the YOLO Robot, or any other color for the matter, since it will make the experience generally more fun and appealing when the child chooses the color of the robot. It can also contribute to the idea of personalization.