SOFTWARE REQUIREMENTS SPECIFICATION

YOLO

YOUR OWN LIVING OBJECT

Şeyma ALAN
2380053

Derya TINMAZ
2380947

Table of Contents

Contents

List of Figures

List of Tables

1 Introduction

This document is the Software Specification Requirement (SRS) of a social robot designed to stimulate creativity in children which is called as YOLO, developed by Patrícia Alves-Oliveira, Patrícia Arriaga, Ana Paiva and Guy Hoffman.

1.1 Purpose of the System

Purpose of the system is producing of a social robot that will aid children's creativity during playing. YOLO project merges traditional play markers such as physical, free, and outdoor play with a storytelling practice in which children utilize the robot as a character in their stories. Creativity is supposed to be stimulated by interaction with the robot. As a result, YOLO is a technological toy with the potential to enhance creativity. The robot offers stimulus for youngsters to construct fresh storylines for the stories they create throughout the play process.

1.2 Scope

YOLO is a non-anthropomorphic social robot created to encourage children's creativity. This robot was designed to be utilized by kids during free play, when they may use it as a character in their own stories. YOLO employs creativity approaches to encourage the invention of new storylines while playing. Especially, YOLO can boost both divergent and convergent thinking.
YOLO can also have multiple personalities, allowing it to engage in socially sophisticated and engaging actions. The robot's design was influenced by psychological theories and models of creativity, design research with children that included user-centered design approaches, and input from experts in the field of creativity.
YOLO uses non-verbal expressive modalities like lights and motions that are equipped with sensors that monitor children's playing activities to communicate with them.
Scope of the YOLO project can be listed as:

- Get info from touch sensor which recognizes the physical contact.
- Get info from optical sensor which detects the movement direction.
- Identify and classify the movement using shape recognition sensors and software.
- Provide proper navigation info to the robot and move the robot using wheels.
- Provide lights to the robot using LEDs.
- Understand the storyline and create proper reaction to the story.

1.3 System Overview

This section of the document will provide detailed information about the system including all components.

1.3.1 System Perspective

YOLO's system based on the general field of Human-Robot Interaction(HRI) and Child-Robot Interaction(cHRI). HRI is a research area dedicated to the design and evaluation of robotic systems that interact with humans. Related to that YOLO can communicate and interact with people in a personal way. Its interaction modalities range from emotional to color, motion, and sound. It has a non-humanlike shape. YOLO's system focuses on playful robotic technology dedicated to children. This allows children to gain benefits such as positive achievements, as an increase in motivation for learning and improvement in collaborative learning. With its system children can learn different styles with different paces through playing. YOLO's cHRI system provides children parents' to a confident, safe, and beneficial playground for their children. Moreover, It provides findings about children behavior and child-robot interaction to related researchers.

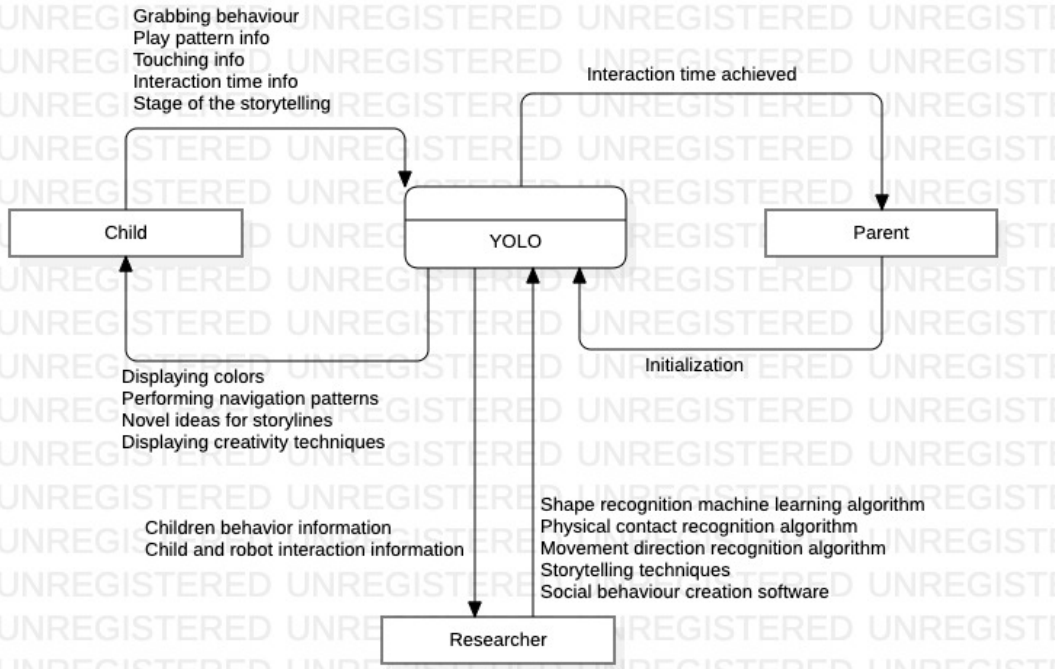

Figure 1: Context Diagram

1.3.2 System Functions

Functionalities (use cases) of YOLO is summarized as what they do in the table below. More advanced and detailed version can be found in Functions section (3.2) with corresponding complete description tables.

Function	Summary
Initialize	Lets parents to assemble the components of the robot, set up the software and integrate with YOLO hardware.
Touch	Lets user the touch it.
Touch for the first time	Lets user the touch it for the first time.
Perform hello behaviour	YOLO shows hello behaviour.
Touch again	Lets user the touch it again.
Perform puppeteer behaviour	YOLO shows puppeteer behaviour.
Display white lights	YOLO shows white lights.
Manipulate	Lets user the manipulate it allowing the user to grab and move the robot.
Supervise	Lets parents the supervise while their children are playing.
Introduce the characters of the story	Lets user start telling story by allowing the user to introduce the characters
Perform mirror technique	YOLO provides stimuli that are connected with the storyline that children are exploring, leading to the elaboration and convergence of story ideas.
Reach the point of the greatest tension at the story	Lets user reaches the top point of the story that was introduced by user.
Perform contrast technique	YOLO presents stimuli that are not related to the storyline that the users are now providing.
Reach the end at the story	Lets the user reach the end of the story that was told.
Do not play anymore	Lets the user stop playing with YOLO and do not touch the robot anymore.
Perform attention call behaviour	YOLO acts attention call behaviour in response not to touch
Turn off	YOLO turns off itself after goodbye behaviour in response not to touch after attention call behaviour
Perform goodbye behaviour	YOLO performs good bye behaviour because the user does not react attention call
Get information about children behaviour	Lets researchers make observations and gain experiences about children behaviour using reactions of children to the YOLO.
Get information about children and robot interaction	Lets researchers make observation about children and robot interaction

Table 1: System Functions

1.3.3 Stakeholder Characteristic

There are three main stakeholders of the YOLO system which are children, parents and researchers.

Children are the most frequent users of the YOLO. They play with YOLO and manipulate it. They use YOLO as a playmate of their play or story. Child should be able to grab, manipulate and play with YOLO. In other words, the user should have basic motor skills.

Another stakeholders are parents. They should initialize and install the robot so that children could start playing. Assembling YOLO is a process that involves interacting with mechanical tools and machinery. There is a detailed installing guide for YOLO. Parents should be able to read and understand technical document then implement the stated technical steps. So, parents should have some technical skills. Also, during the usage of YOLO, parents should supervise their children.

The last stakeholders are researchers. Researchers conduct an academic research and observation about children behaviours and child - robot interaction using YOLO. They should be capable of creating and conducting experiments safely.

1.3.4 Limitations

- **Regulatory policies:** YOLO should be considered as a toy. As with any toy, YOLO should be used under parental supervision.
- **Hardware limitations:** The hardware used in the YOLO is an embedded system that as space restrictions for the software used in YOLO movements which makes the hardware options limited.
- **Interface to other applications:** There is no interface to other applications.
- **Parallel operation :** In YOLO system AI algorithms and sensors work concurrently. So that, system analyzes given shapes and the story-line then YOLO reacts based on the output of the algorithms.
- **Audit functions :**There is no audit function in the system.
- **Control functions :** The control of the YOLO system is done by Raspberry Pi system in the robot. It controls all of the components of YOLO.
- **Higher-order language requirements :** YOLO software is written in Python. Using appropriate libraries, Python allows not only coding of robot control system but also machine learning algorithms. With the benefit of Python libraries, whole YOLO project is coded with Python.

- **Signal handshake protocols :** All communication protocol between YOLO Raspberry Pi and router uses HTTPS protocol for sending and receiving information from the web. server and client servers must meet WoT W3C standards.
- **Quality requirements :** There is no such limitation.
- **Criticality of the application :** There is no such limitation.
- **Safety and security considerations :** Due to YOLO is considered as a toy for children, its components should be appropriate for child development. There is no possible major hazards while using and playing YOLO. However, parents always supervise their children. Also, there is an assemble guideline that should parents use while initializing YOLO and obey guards.
- **Physical/mental considerations:** Physically/mentally disabled children can also use and play with YOLO.
- **Limitations that are sourced from other systems:** There is no such limitation

1.4 Definitions

Term	Definitions
ML system	Machine learning system. It implements shape recognizer algorithm. It is responsible for detecting the shapes that users draw while playing with YOLO.
API	Application Programming Interface
YOLO	Your own living object. It is the name of the robot

Table 2: Definitions

2 References

This document is prepared with respect to IEEE 29148-2011 standard:

29148-2011 - ISO/IEC/IEEE International Standard - Systems and software engineering – Life cycle processes –Requirements engineering.

Other sources:

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

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

3 Specific Requirements

3.1 External Interfaces

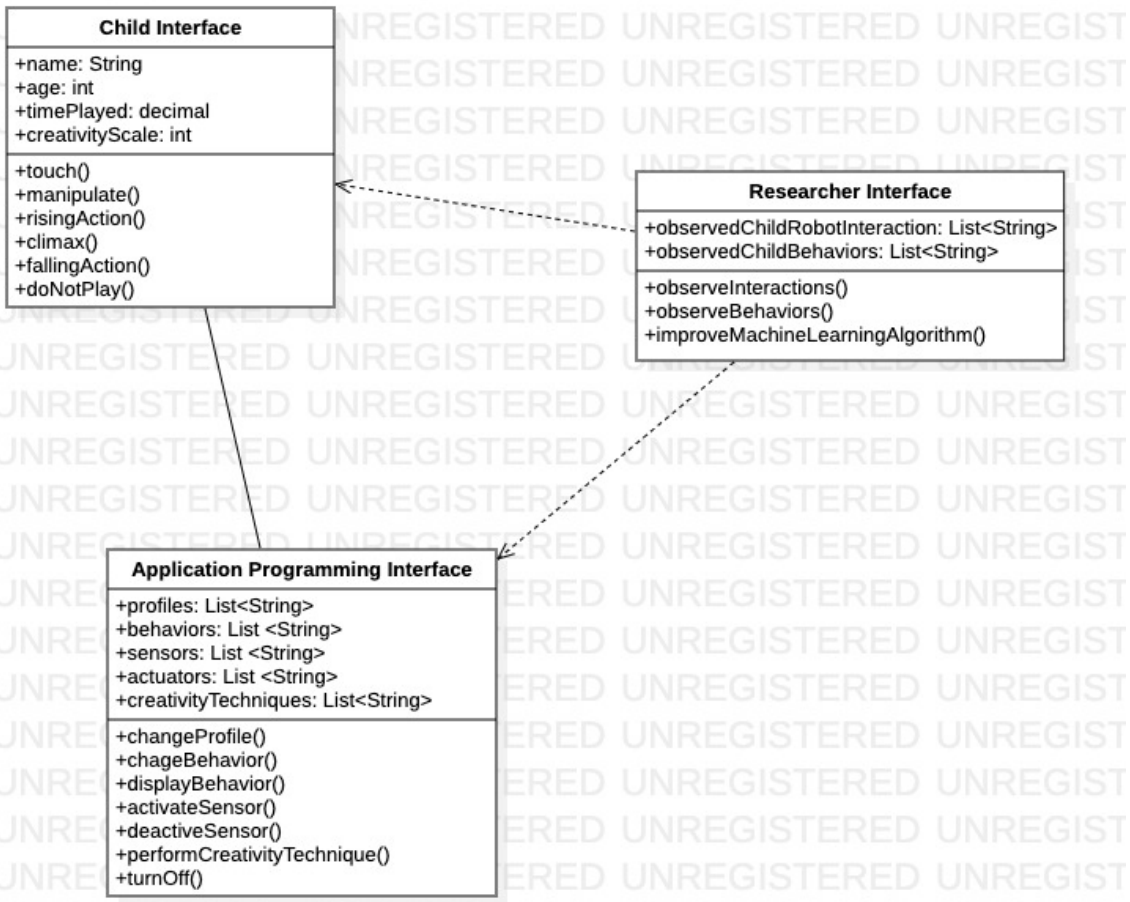

Figure 2: External Interfaces Class Diagram

3.2 Functions

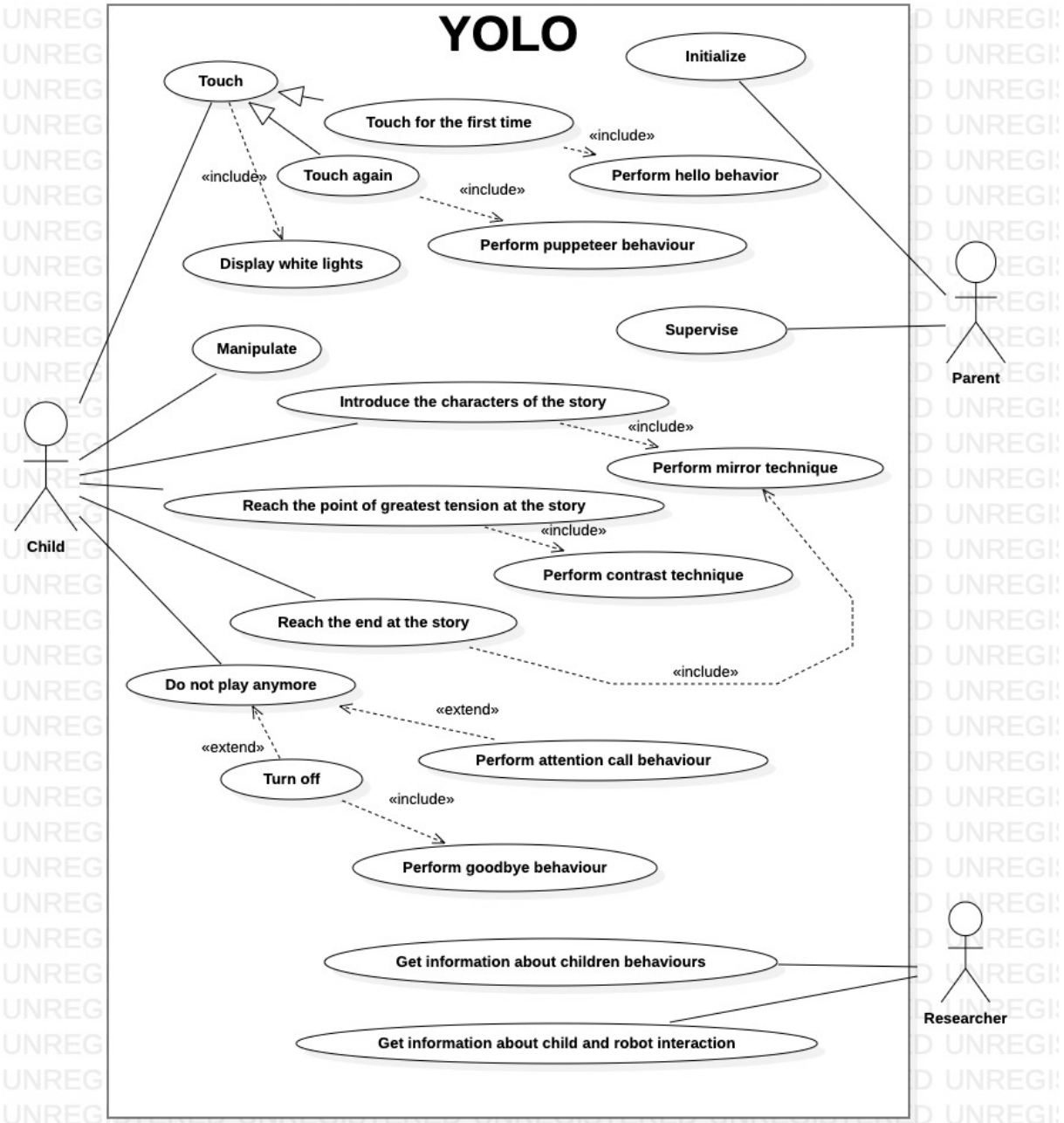

Figure 3: Use-Case Diagram

Use case name	Initialize
Actors	Parent
Description	Parents assemble the components of the robots, make set up of YOLO software and integrate with YOLO hardware.
Data	Information of materials' usages
Preconditions	User has the components of the robot. The robot's battery should be sufficiently charged. The robot should be within range of the router. Wire connection should be strong.
Stimulus	User wants to turn on the robot.
Basic Flow	1- Parent assembles the components of the robot. 2- Parent connects the Raspberry Pi and to the software program via wi-fi using router. 3- Parent initializes the YOLO's software program using computer or laptop.
Alternative Flow	-
Exception Flow	The robot's battery is not enough, or the robot out of the router range, or wiring connections are weak, the robot is not turned on.
Post conditions	The robot is turned on.

Table 3: Initialize

Use case name	Touch
Actors	Child
Description	Child touches the robot so that the capacitive touch sensor is activated.
Data	-
Preconditions	The robot is in a reachable place.
Stimulus	User wants to touch the robot.
Basic Flow	1.Child touches the robot. 2.Touch sensor is activated. 3.Robot becomes deactivated. 4.Child plays with the robot as he/she does with conventional toys.
Alternative Flow	1.Child touches the robot. 2.Touch sensor is activated. 3.Robot becomes deactivated. 4.Child plays with the robot as he/she does with conventional toys. 5.Child releases his/her hand. 6.Robot becomes autonomous.
Exception Flow	2.Touch sensor is not activated.
Post conditions	The robot performs idle behaviour.

Table 4: Touch

Use case name	Touch for the first time
Actors	Child
Description	Child touches the YOLO for the first time.
Data	-
Preconditions	The user should not have touched the robot before.
Stimulus	User wants to touch the robot for the first time.
Basic Flow	1.Child touches the robot. 2. Hello behaviour is performed by the YOLO.
Alternative Flow	-
Exception Flow	Touch sensor is not activated.
Post conditions	The robot performs idle behaviour.

Table 5: Touch for the first time

Use case name	Perform hello behaviour
Actors	Child
Description	YOLO performs hello behaviour, which is declared in the behaviour set of the robot.
Data	Sensor data from the capacitive touch sensor and counter value of the touch numbers.
Preconditions	YOLO should be touched for the first time.
Stimulus	-
Basic Flow	1. User touches YOLO 2. Counter starts counting the number of touches and save them. 3. YOLO system checks whether this touch is the first time or not. 4. If first time, then performs hello behaviour.
Alternative Flow	-
Exception Flow	1. Touch sensor may not give right outputs. 2. Counter counts wrong number for the touch value. 3. Hello behaviour will not be performed.
Post conditions	Upon performing hello behaviour, now YOLO will be ready for puppeteer behaviour.

Table 6: Perform hello behaviour

Use case name	Touch again
Actors	Child
Description	Child physically touches the robot again.
Data	-
Preconditions	The user should have touched the robot before.
Stimulus	User wants to touch the robot again.
Basic Flow	1.Child touches the robot. 2. Puppeteer behaviour is performed by the YOLO.
Alternative Flow	-
Exception Flow	Touch sensor is not activated.
Post conditions	The robot performs idle behaviour.

Table 7: Touch again

Use case name	Perform puppeteer behaviour
Actors	Child
Description	YOLO performs puppeteer behaviour. In other words, robot refrains any movement and it is used as traditional toys with the full control of children.
Data	Sensor data from the capacitive touch sensor and counter value of the touch count.
Preconditions	YOLO must have been touched again.
Stimulus	-
Basic Flow	1. User touches YOLO 2. Counter starts counting the number of touches and save them. 3. YOLO system checks whether this touch is the first time or not. 4. If not, then performs puppeteer behaviour.
Alternative Flow	-
Exception Flow	1. Touch sensor may not give right outputs. 2. Counter counts wrong number for the touch value. 3. Puppeteer behaviour will not be performed.
Post conditions	Upon performing puppeteer behaviour, user can use YOLO as like a non-autonomous traditional toy.

Table 8: Perform puppeteer behaviour

Use case name	Display white lights
Actors	Users
Description	YOLO displays white lights in their glowing fibers.
Data	Sensor data from capacitive touch sensor.
Preconditions	User should be touching the YOLO.
Stimulus	-
Basic Flow	1.User touches YOLO 2. Capacitive touch sensor is activated 3. Glowing fibers start lightning.
Alternative Flow	-
Exception Flow	Capacitive touch sensor may not get the right outputs.
Post conditions	The robot performs idle behaviour.

Table 9: Display white lights

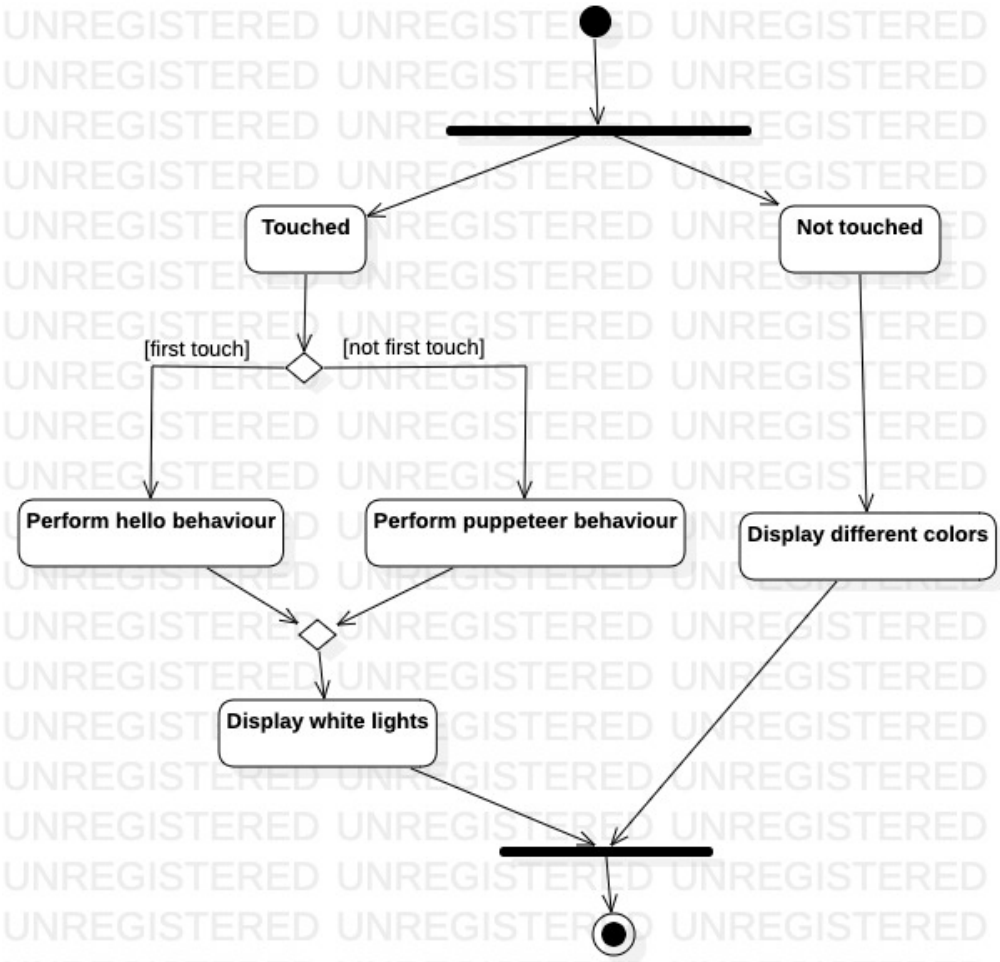

Figure 4: Activity Diagram of "Touch"

Use case name	Manipulate
Actors	Child
Description	Child grasps and manipulates the robot.
Data	-
Preconditions	In order to manipulate the robot, children should be grasping (in other words, touching) the robot. The robot should have optical sensor in order to recognize of the direction of the play patterns.
Stimulus	User wants to manipulate the robot.
Basic Flow	1. User performs movements patterns with the robot. 2.The robot imitates movements patterns.
Alternative Flow	1. User performs movements patterns with the robot. 2.The robot does a different movement.
Exception Flow	-
Post condition	The robot performs a movement.

Table 10: Manipulate

Use case name	Supervise
Actors	Parents
Description	Parents supervise their children while they are playing with YOLO.
Data	Child and robot interaction observation.
Preconditions	-
Stimulus	-
Basic Flow	1. Child plays with YOLO. 2.Parents will observe he/she all the time.
Alternative Flow	-
Exception Flow	Parent may not observe children or parent would leave room for a short time.
Post conditions	-

Table 11: Supervise

Use case name	Introduce the characters of the story
Actors	Child
Description	Child starts to introduce story characters and builds the story while moving the robot and manipulating it.
Data	Movement data that is collected from the optical sensor
Preconditions	User should be manipulating the YOLO
Stimulus	-
Basic Flow	1.User grasps the YOLO 2. User manipulates the YOLO 3.Optical sensor collects data actively. 4. Shape recognition starts identifing and categorizing each movement.
Alternative Flow	-
Exception Flow	Optical sensor may not work properly and not receive the first characters. or shape recognition cannot capture the objects and cannot identify the characters. or YOLO is not close enough to the router so system cannot start to process machine learning algorithms.
Post conditions	Now YOLO system is ready for analyzing the given story and process the story in machine learning algorithm.

Table 12: Introduce the characters of the story

Use case name	Perform mirror technique
Actors	Child
Description	YOLO performs mirror technique that stimulates the convergent thinking of the user as a response to the arc of the story.
Data	Movement and shape recognition data that is collected from the optical sensor and shape recognition algorithm
Preconditions	User should have already introduced the characters of the story.
Stimulus	-
Basic Flow	1. User introduces characters of the story. 2. Shape recognition and machine learning algorithms process the story. 3.Algorithm decides the user is in the first arc of the storyline 4. YOLO performs mirroring technique.
Alternative Flow	-
Exception Flow	Algorithms may not work properly and cannot decide the right part of the story so that we may observe different techniques.
Post conditions	-

Table 13: Perform mirror technique

Use case name	Reach the point of the greatest tension at the story
Actors	Child
Description	Child reaches tension point of the story that he/she is telling.
Data	Movement and shape recognition data that is collected from the optical sensor and shape recognition algorithm
Preconditions	Child should be already introduce the story.
Stimulus	-
Basic Flow	1. User keeps manipulating the YOLO 2.Optical sensor collects data actively. 3. Shape recognition starts identifing and categorizing each movement.
Alternative Flow	-
Exception Flow	Optical sensor may not work properly and not receive the first characters. or shape recognition cannot capture the objects and cannot identify the characters. or YOLO is not close enough to the router so system cannot process machine learning algorithms.
Post conditions	Now, YOLO system is ready to perform corresponding technique for the tension part of the story.

Table 14: Reach the point of the greatest tension at the story

Use case name	Perform contrast technique.
Actors	Child
Description	Robot performs contrast technique at the greatest tension of the story, which boosts the divergent thinking of the user.
Data	Movement and shape recognition data that is collected from the optical sensor and shape recognition algorithm
Preconditions	Child should be already reached the greatest tension point of the story.
Stimulus	-
Basic Flow	1. User keeps performing the story. 2. Shape recognition and machine learning algorithms process the story. 3.Algorithm decides the user is in the second arc of the story-line 4. YOLO performs contrast technique.
Alternative Flow	-
Exception Flow	Algorithms may not work properly and cannot decide the right part of the story so that we may observe different techniques.
Post conditions	-

Table 15: Perform contrast technique

Use case name	Reach the end at the story
Actors	Child
Description	Child reaches the story that he/she performing.
Data	Movement and shape recognition data that is collected from the optical sensor and shape recognition algorithm
Preconditions	Child should be already passed the greatest tension point of the story and YOLO performed contrasting technique.
Stimulus	-
Basic Flow	1. User keeps manipulating the YOLO 2.Optical sensor collects data actively. 3. Shape recognition starts identifying and categorizing each movement.
Alternative Flow	-
Exception Flow	Optical sensor may not work properly and not receive the characters. or shape recognition cannot capture the objects and cannot identify the characters. or YOLO is not close enough to the router so system cannot process machine learning algorithms.
Post conditions	-

Table 16: Reach the end at the story

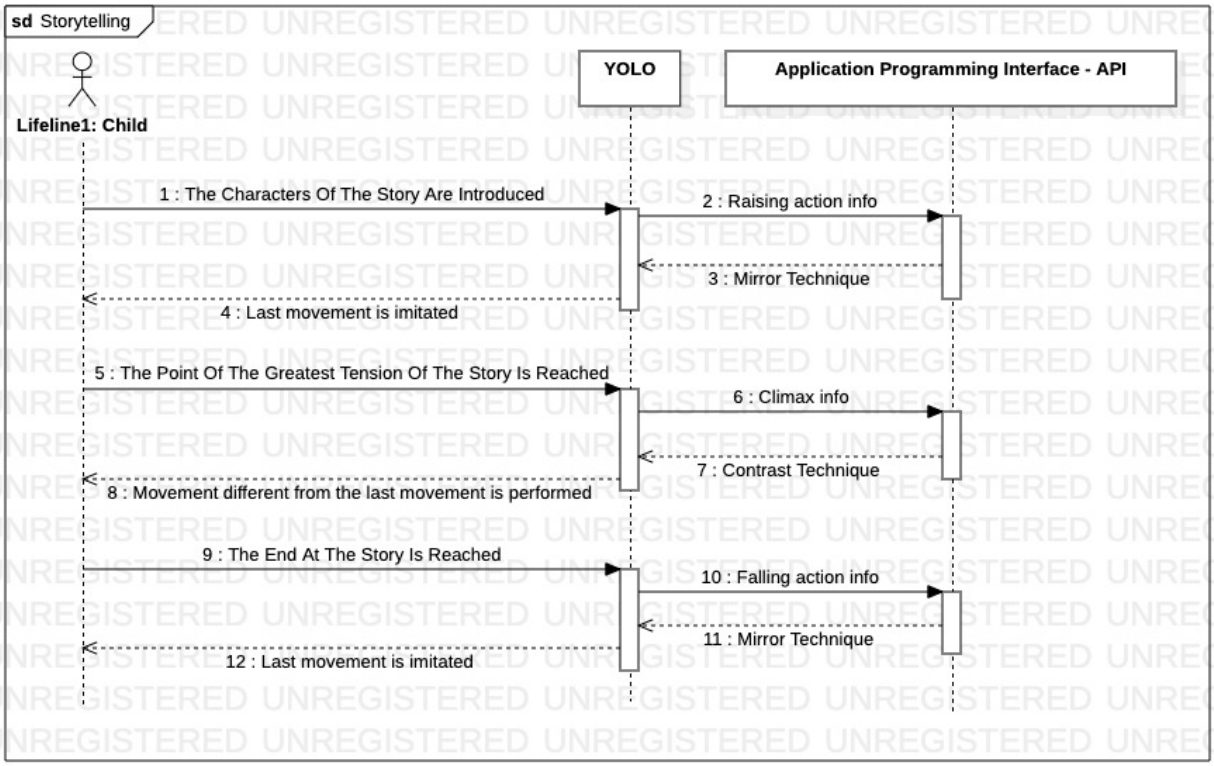

Figure 5: Sequence Diagram of "Storytelling"

Use case name	Do not play anymore.
Actors	Child
Description	Child stops playing with YOLO. In other words, he/she do not touch the robot anymore.
Data	-
Preconditions	Child should have been playing with the robot before.
Stimulus	Child does not want to play with the robot anymore.
Basic Flow	1. Child stops playing with YOLO. 2. Robot performs goodbye behaviour and turns off.
Alternative Flow	-
Exception Flow	-
Post conditions	The robot turns off.

Table 17: Do not play anymore

Use case name	Perform attention call behaviour.
Actors	Child
Description	Robot performs attention call behaviour in response not to touch, which is defined in the software of behaviour set.
Data	Timer value of the last touch.
Preconditions	Child must stopped playing with the YOLO. Also, the time value since recorded the last touch should be greater than the attention call threshold.
Stimulus	-
Basic Flow	1. Child stops playing with YOLO. 2. Robot performs attention call behaviour.
Alternative Flow	-
Exception Flow	-
Post conditions	The robot performs idle behaviour.

Table 18: Perform attention call behaviour

Use case name	Turn off.
Actors	Child
Description	Robot terminates.
Data	Timer value of the last touch.
Preconditions	Child should not be touching the robot anymore, or max interaction time is achieved.
Stimulus	It should have been a while since the robot is being touched , or the child played a lot with the robot
Basic Flow	1.Child stops playing with YOLO. 2.Robot performs attention call behaviour and child still does not touch. 3.Robot performs goodbye behaviour and turns off.
Alternative Flow	1.Child plays with YOLO. 2. Max interaction time is achieved. 3. Robot performs goodbye behaviour and turns off.
Exception Flow	-
Post condition	The robot is not active anymore.

Table 19: Turn off

Use case name	Perform goodbye behaviour
Actors	Child
Description	Robot performs goodbye behaviour in response to the turn off decision.
Data	Timer value of the last touch.
Preconditions	YOLO should have already decided to terminate interaction, i.e. turn-off.
Stimulus	The robot is about to turn off.
Basic Flow	1.Child stops playing with YOLO. 2.Robot performs goodbye behaviour and turns off.
Alternative Flow	1.Child plays with YOLO. 2. Max interaction time is achieved. 3. Robot performs goodbye behaviour and turns off.
Exception Flow	-
Post conditions	The robot turns off.

Table 20: Perform goodbye behaviour

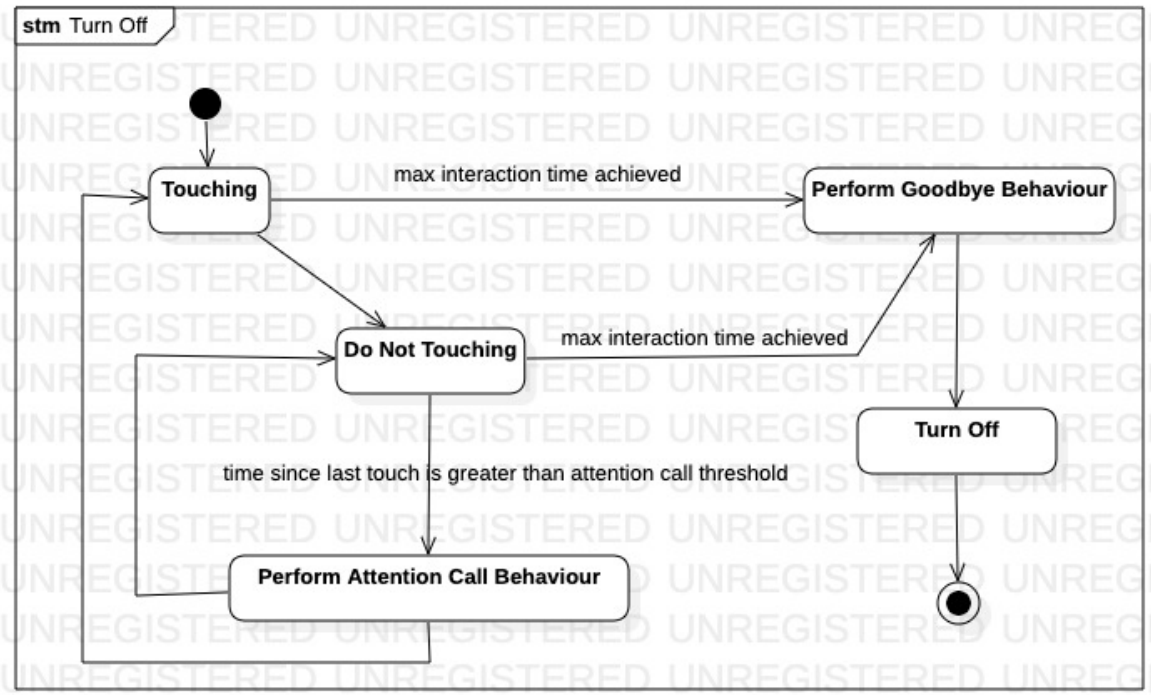

Figure 6: State Diagram of "Turn Off"

Use case name	Get information about children behaviours
Actors	Researcher
Description	Researchers observe the behaviours of the children and get information about children attitudes.
Data	Observation of children behaviours.
Preconditions	Experiment environment has been set up Needed consent forms have been signed.
Stimulus	-
Basic Flow	1.Children play with YOLO 2.Researchers observe children behaviours.
Alternative Flow	-
Exception Flow	-
Post conditions	Researchers gain experience on children attitudes. So that, they improve YOLO and other scientific researches.

Table 21: Get information about children behaviours

Use case name	Get information about children and robot interaction
Actors	Researcher
Description	Researchers observe the behaviours of the children while they were using YOLO. So that, they can gain information and experience on children and robot interaction
Data	Observation of children and robot interaction.
Preconditions	Experiment environment has been set up properly Needed consent forms have been signed.
Stimulus	-
Basic Flow	1.Children play with YOLO 2.Researchers observe children's attitude towards YOLO and their interaction with robots.
Alternative Flow	-
Exception Flow	-
Post conditions	Researchers gain experience on children and robot interactions. So that, they improve YOLO and other scientific researches.

Table 22: Get information about children and robot interaction

3.3 Usability Requirements

- User shall be able to initialize and successfully connect the software of YOLO to the router.
- User shall be able to grab and play with the robot.
- User shall be able to manipulate the robot.
- YOLO shall be react to the storyline of the user.
- YOLO shall be move based on the software system decisions.
- YOLO shall be light its LEDs upon touching.
- YOLO shall be perform specific behaviours based on the storyline.

3.4 Performance Requirements

- Router should be able to stay connected while the user is playing with YOLO.
- Battery average life is between 5 and 7 hours. It should be properly charged.
- Router range is important. Depending on the type of the router, if the router range is wide, child can play outside. Otherwise, not.
- Wiring connections should be as strong as possible.

3.5 Logical Database Requirements

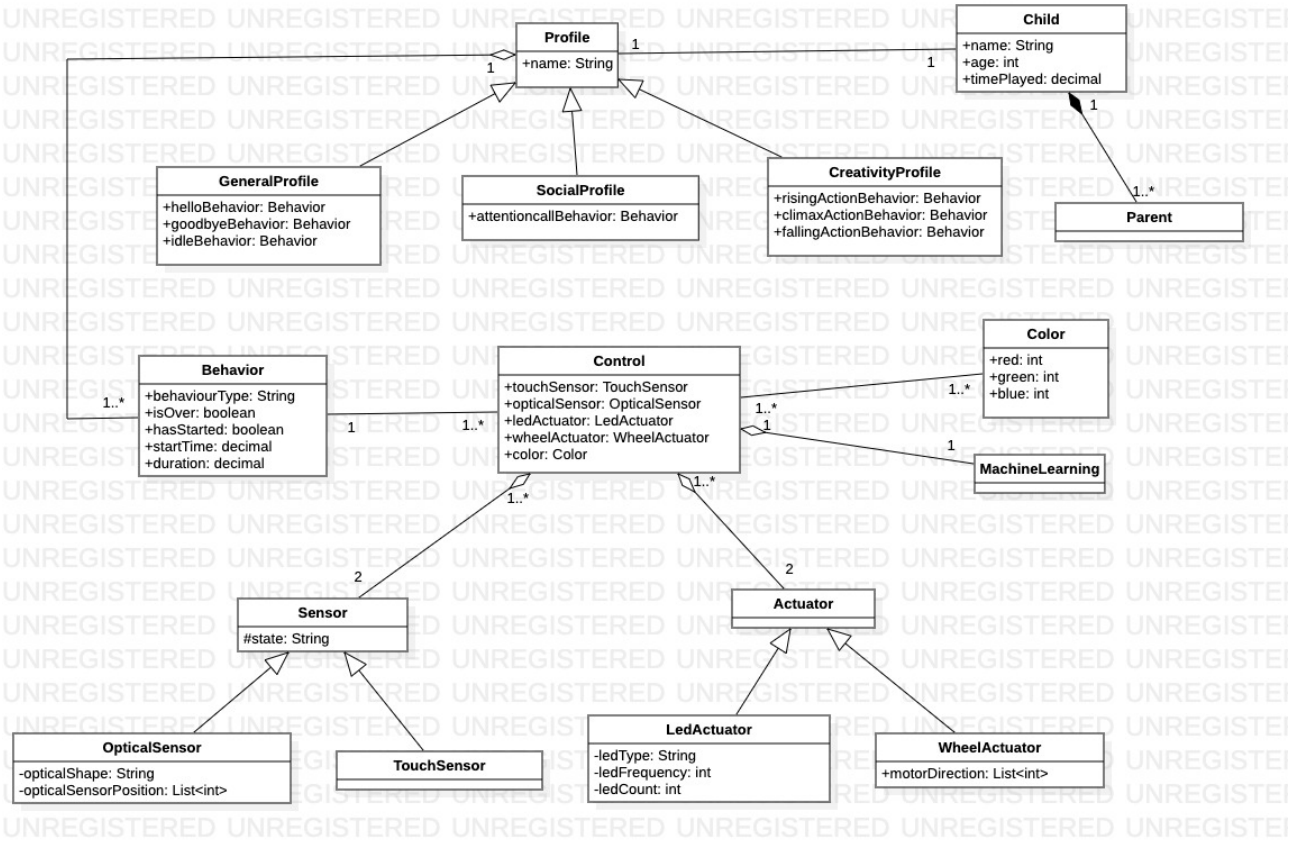

Figure 7: Logical Database Requirements Class Diagram

- A child can have one or more parents that observes its interactions.
- A child can have an interaction with robot having one profile at a time.
- When YOLO is initialized and connected to the router, active connection entry shall be created with unique IDs.
- Every active/closed connection has an exactly one event log table which stores performed events.
- Whenever a profile is created, General Profile, Social Profile and Creativity Profile tables are created.
- A profile can have one or more behaviour.
- One behavior can have a connection with one or more control table.

- Sensor, actuator and machine learning tables use data from control tables. So, sensor, actuator and machine learning tables are weak entities for control table.
- Touch Sensor and Optical Sensor tables are children of Sensor table. These two inherit attributes of Sensor table.
- Led Actuator and Wheel Actuator tables are children of Actuator table. These two inherit attributes of Actuator table.
- Existence of the sensor and actuator tables depends on control table.

3.6 Design Constraints

YOLO is a robot made for children. Due to its target group and playful application nature, there should not be any possible major hazards when operating and playing with it. Also, all of YOLO's components should be appropriate for children health and development.

YOLO should be connected to the router during the whole playing session. Otherwise, It will not react to the story of user.

3.7 System Attributes

Reliability

- In case of a sensor shutdown, other sensors shall continue their operations.
- All of the hardware and software code of the system must be open-source.
- In case of a connection failure robot becomes unresponsive but user shall be able to continue to use it in non-autonomous mode.

Availability

- After initialization, YOLO shall be available immediately and start responding.
- In the absence of an internet or router connection, YOLO shall be available to use it in non-autonomous mode.

Security

- The system components shall not cause any possible major hazards due to its safe design for children.

Maintainability

- YOLO software must be able to be updated via the connected computer in any time.

- Open-source designs and codes of system shall be updated regularly and shall help to use of system.

Portability

- User shall be able to connect to YOLO and use all related functions of the system via router whenever there is an internet connection.
- Libraries which are used in the software shall be applicable for different programming languages.

3.8 Supporting Information

YOLO is an open-source social robot project. A user can assemble the 3D-printed YOLO components and hardware, along with the embedded software and machine learning algorithms provided by creators on Github. Furthermore, users with technical background can contribute to the project as an open-source developer.

4 Suggestions to Improve The Existing System

- Although it is an open-source project, users may struggle to assemble whole hardware components. Pre-assembling some of the parts and letting user to assemble the bigger parts of the system would be helpful for them.
- Parents may want to know how their children interact with robot and how long they play with YOLO. Providing these kind of necessary information to parent would be impactful. Moreover, tracking this kind of playing data would be useful for research developments. Tracking and collecting usage data then presenting it to user with an interface would improve the scope of the system.