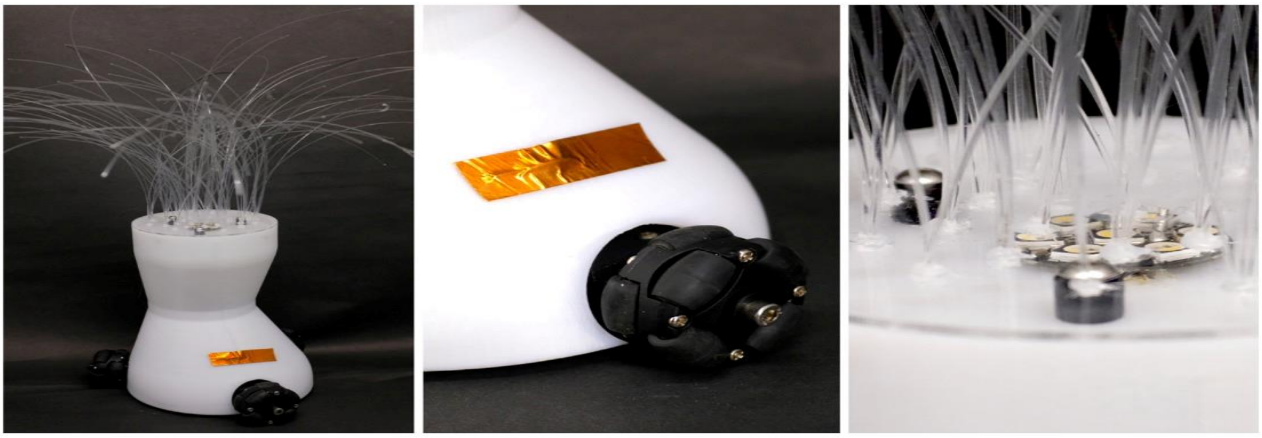

Software Architectural Descriptions

YOLO

GROUP 16

Fazlı Balkan 2380178

Tunahan Dündar 2380319

Table of Contents

List of Figures

List of Tables

1. Introduction

1.1. Purpose and objectives of the YOLO social robot

The purpose of the system is to design and develop a toy in order to encourage creativity among children by allowing them to create stories via YOLO. Moreover, YOLO's AI system responds to children's emotional state such that it chooses among Exuberant, Aloof, and Harmonious state. Furthermore, by detecting the children's stories YOLO's AI system chooses contrast or mirroring action.

1.2. Scope

In the scope of this system, it aims to contribute to the children's psychological development by allowing them to explore their deep emotions. For example, imagine a child that is behind from his/her peers in emotional development, YOLO can be used for children to understand their emotions and imagination. It can also be used by psychologists or pedagogue to observe children's psychological state.

YOLO accomplished to contribute to the children's psychological development by its advanced AI and control system. For example, in order to simulate a different perspective of a story YOLO acts in contrast action such that it moves different from the children's movements.

This scope can be extended. For example, in a story which a child created YOLO can simulate different social profiles to exhibit social behaviors, which can contribute child to experience distinct emotional actions. It also enriches the story of the child.

1.3. Stakeholders and their concerns

The target stakeholders can be divided into two groups as developers and users. Firstly, developers use that device in order to improve and add new functionalities. In this way, they personalize the YOLO on their purpose. On the other hand, users, children, use YOLO to create stories which helps them to stimulate their imaginations and emotional development.

Children that are using YOLO might be negatively affected by the device. For example, since the device choose the next move according to children' movements, it might act in a way that hurts the children physically or psychologically. Moreover, developers that change or add functionalities of the device might disrupt the existing device. For example, when a developer tries to add new emotional state, it might damage the existing emotional states.

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.

[1] P. Alves-Oliveira, P. Arriaga, G. Hoffman, A. Paiva, Guide to build YOLO, a creativity-stimulating robot for children, HarwardX 6 2019.e00074, http://www.elsevier.com/locate/ohx

[2] P. Alves-Oliveira, P. Arriaga, A. Chandak, G. Hoffman, A. Paiva, Software architecture for YOLO, a creativity-stimulating robot, HarwardX 6 2020.100461, http://www.elsevier.com/locate/softx

3. Glossary

Term	**Definition**
AI System	Artificial intelligence system. In an abstract way, it is the brain of the YOLO. It decides which action should be taken in any moment.
Controller	It performs the action that is decided by the AI system
API	Application Programming Interface
YOLO	Your Own Living Object

Table 1: Glossary

4. Architectural Views

4.1. Context View

4.1.1. Stakeholders’ use of this view

Stakeholders can use this view to see the system and its environment. It can be used to see the relationships, interactions of the system between its environment.

4.1.2. Context Diagram

This is the context diagram for the YOLO. Actors and their relation with the YOLO is shown. There are 2 actors, one of which is User and the other is Developer. User gives input to the information system by touching and movement. YOLO acts according to the given user input. On the other hand, developer adds new functionality or changes the existing ones. YOLO’s information system feedbacks the developer.

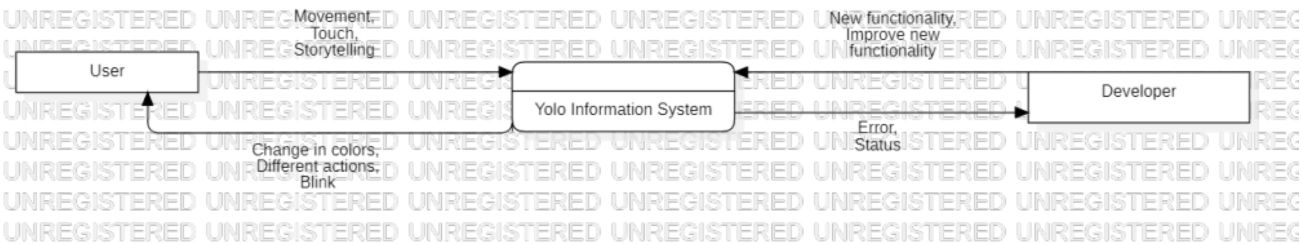

Figure 1: Context Diagram

4.1.3. External Interfaces

There are 3 external interfaces as User Interface, API, and Github. User Interface can be defined as an interface that the YOLO user can manage their account. API is an interface that can be used by developer or user in order to access or change the YOLO. API is the endpoint that makes this possible for users and developers.

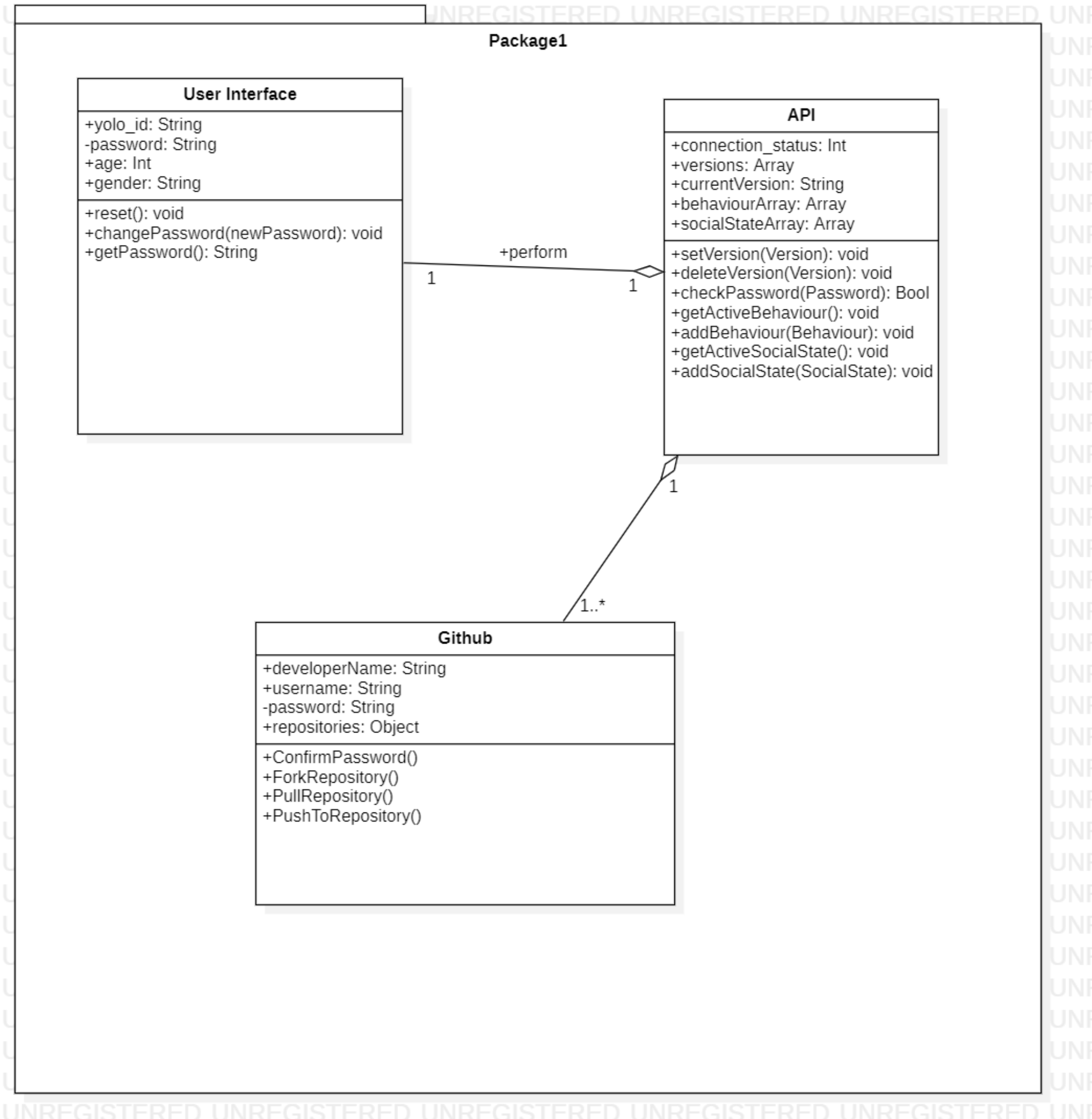

Figure 2: External Interfaces Class Diagram

4.1.4. Interaction scenarios

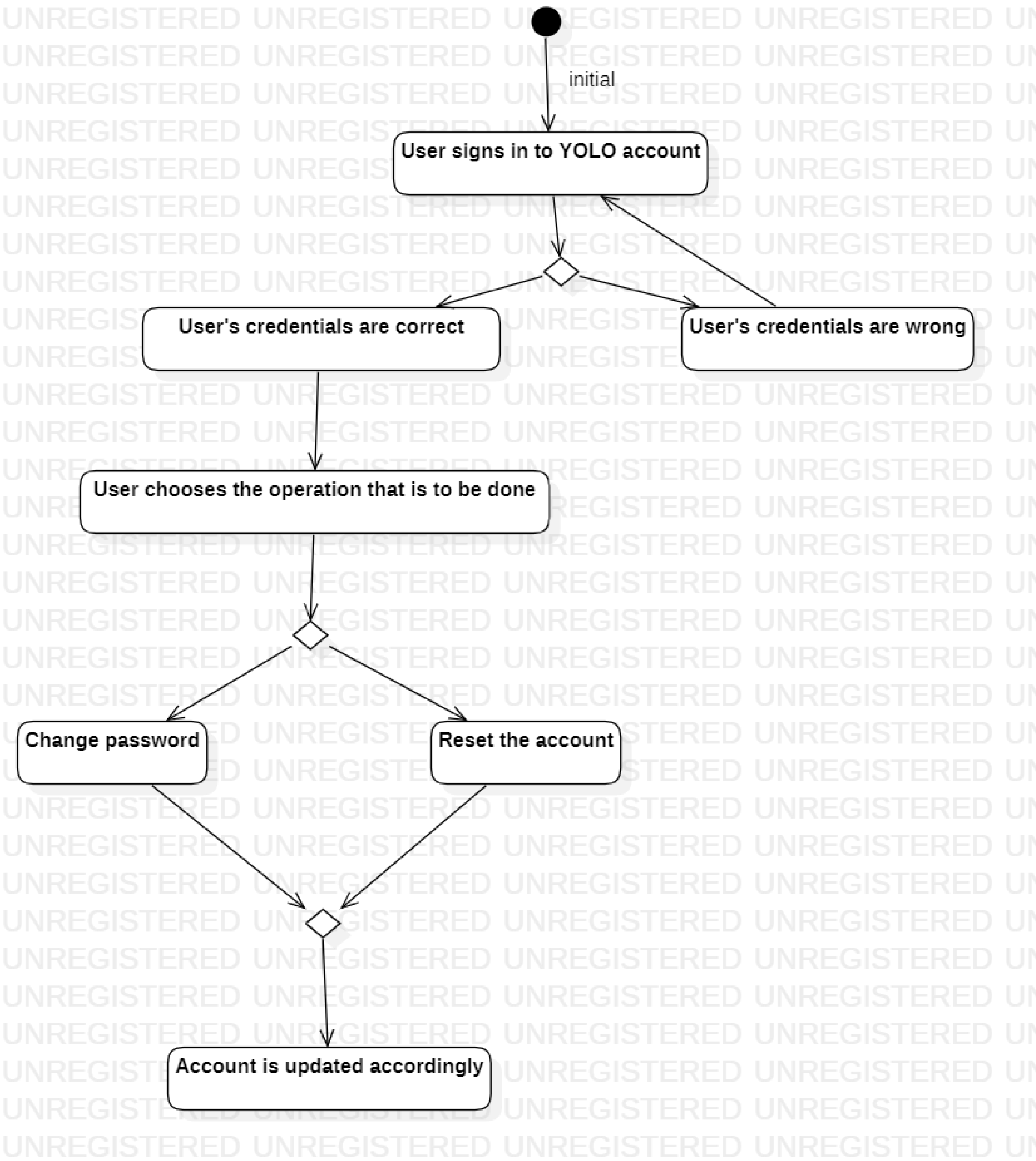

Figure 3: Activity Diagram 1

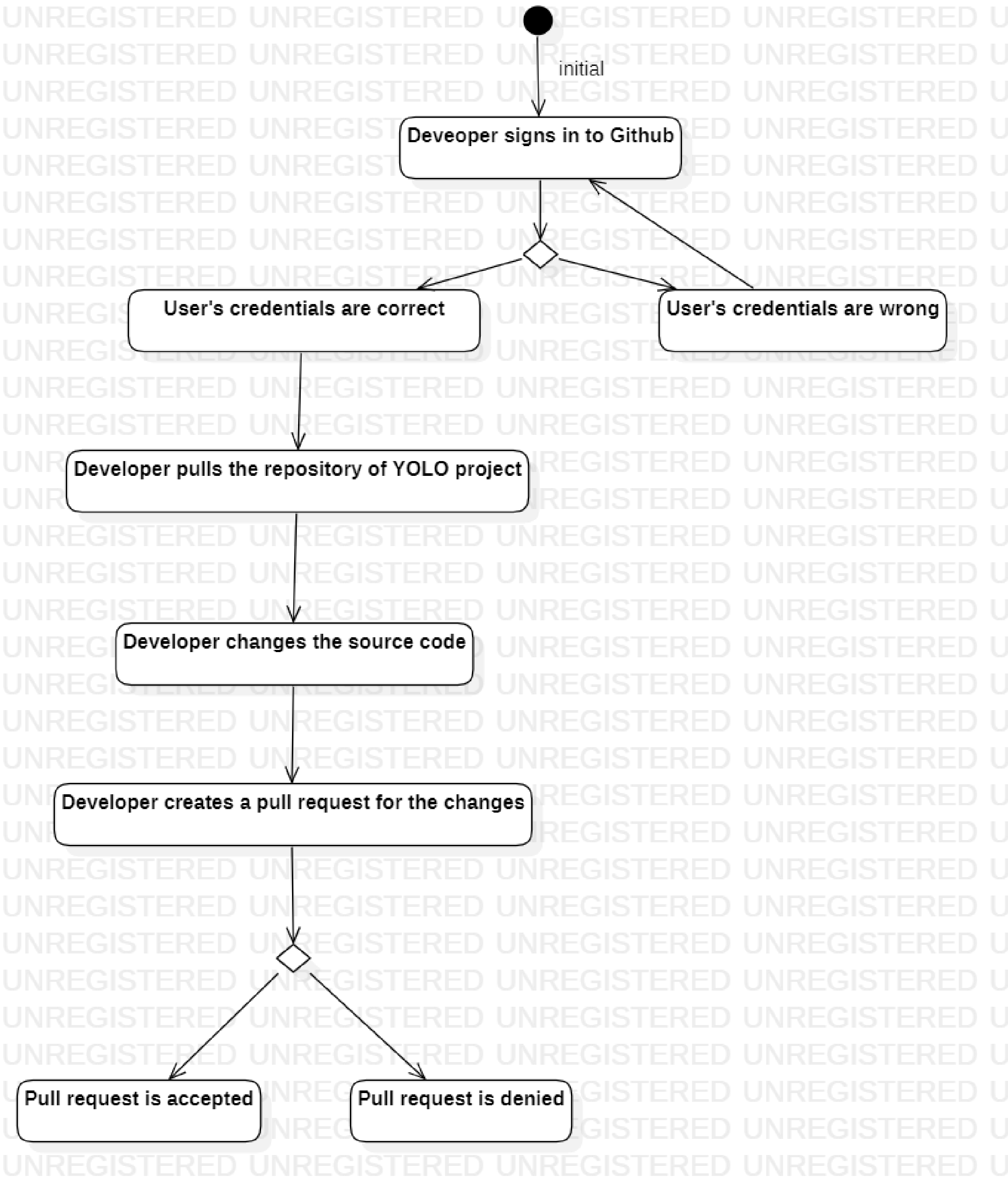

Figure 4: Activity Diagram 2

4.2. Functional View

4.2.1. Stakeholders' use of this view

This view is important for Stakeholders because it helps them understand the functionality of the subsystems of the YOLO project. The stakeholders use this view to understand and analyze the subsystems and the interaction between them.

4.2.2. Component Diagram

This is the component diagram for the YOLO. There are 5 components as Sensors, AI System, Control System, Router and Computer. Basically, it starts with sensors getting the information from user and it transfers this information to AI System. AI System interprets the data and sends decision for emotional and behavior type to Control System. Control System acts accordingly. Router connects the API with Wifi.

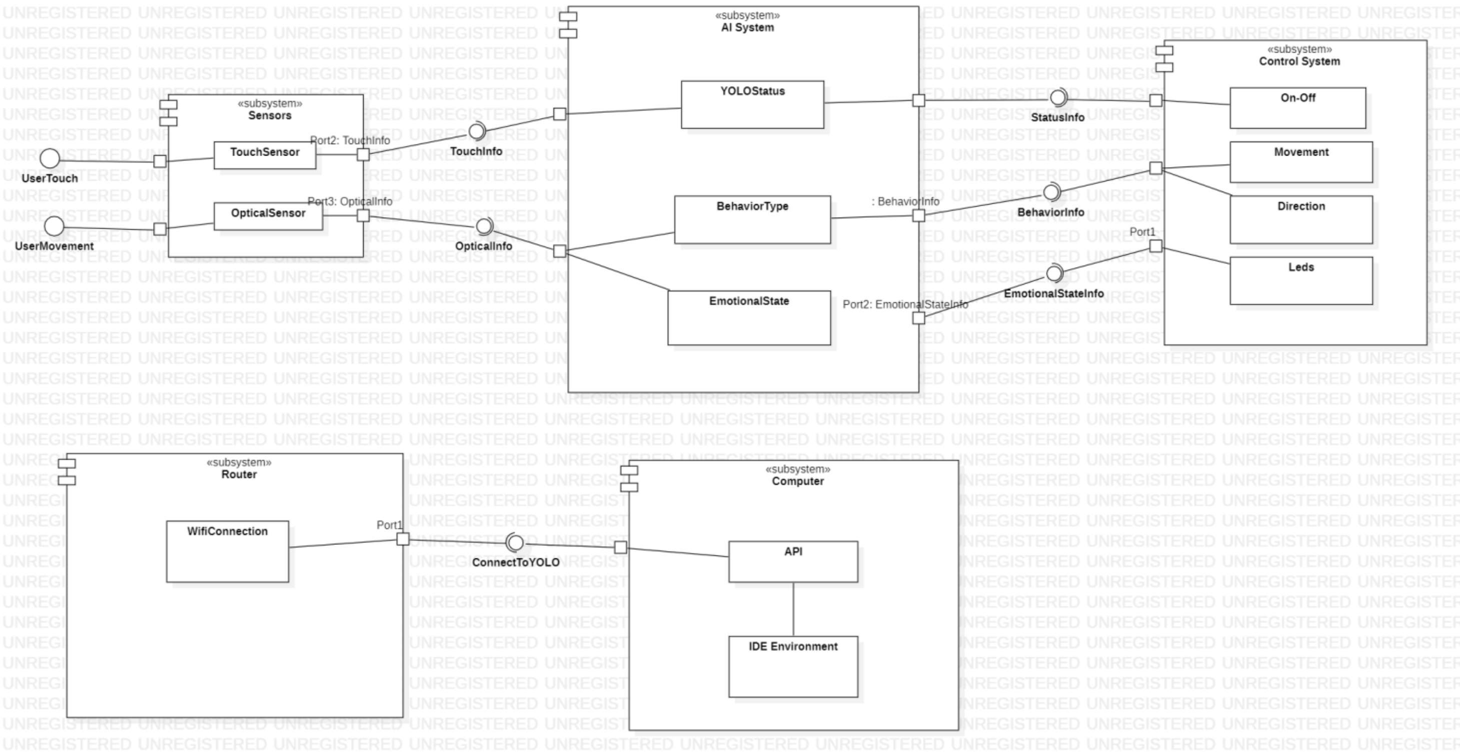

Figure 5: Component Diagram

4.2.3. Internal Interfaces

This is the internal interface class diagram. There are 4 internal interfaces as YOLO Control System, AI System, Communication System, Sensor. Functions' and attributes' names explain itself.

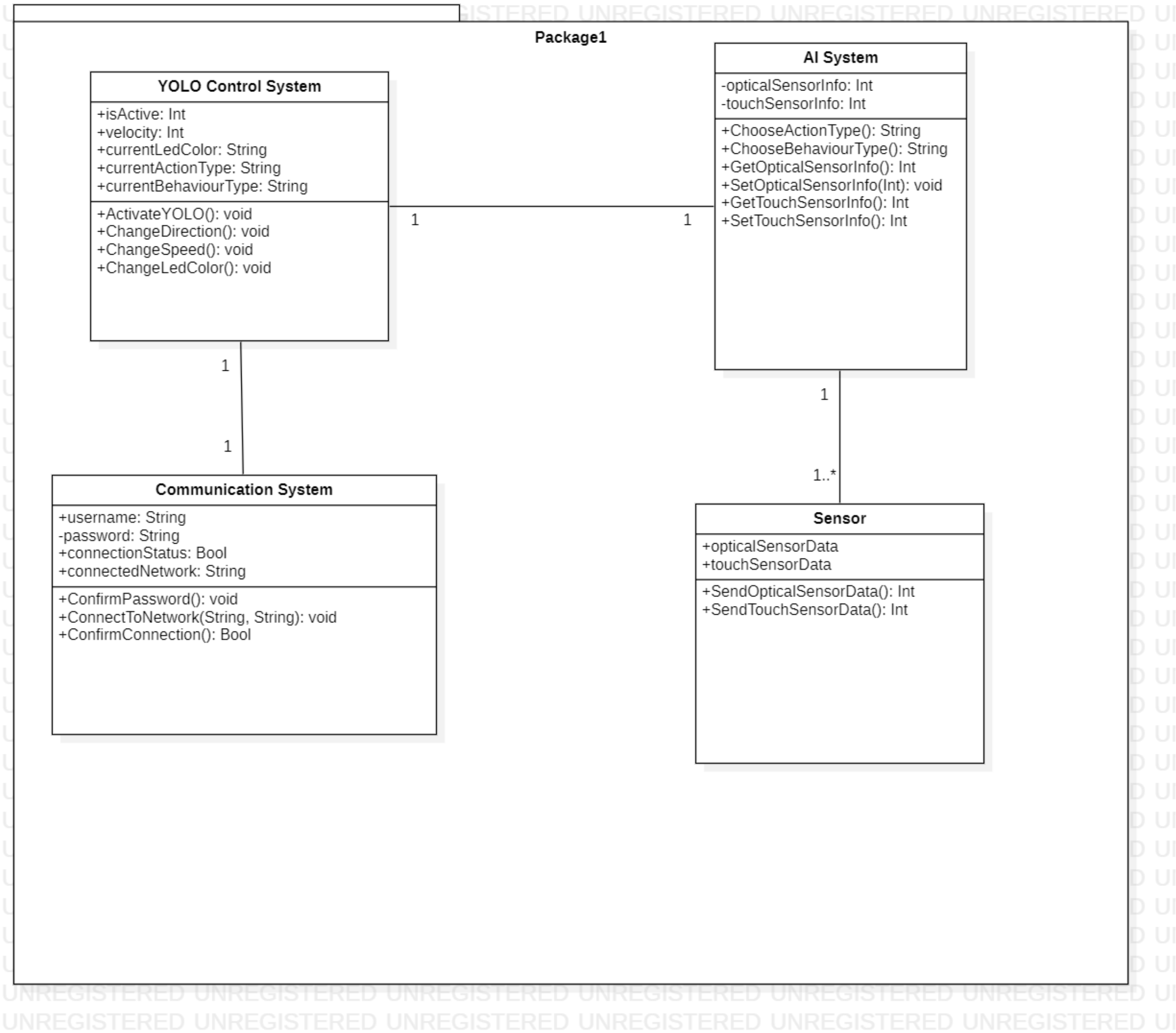

Figure 6: Internal Interface Diagram

4.2.4. Interaction Patterns

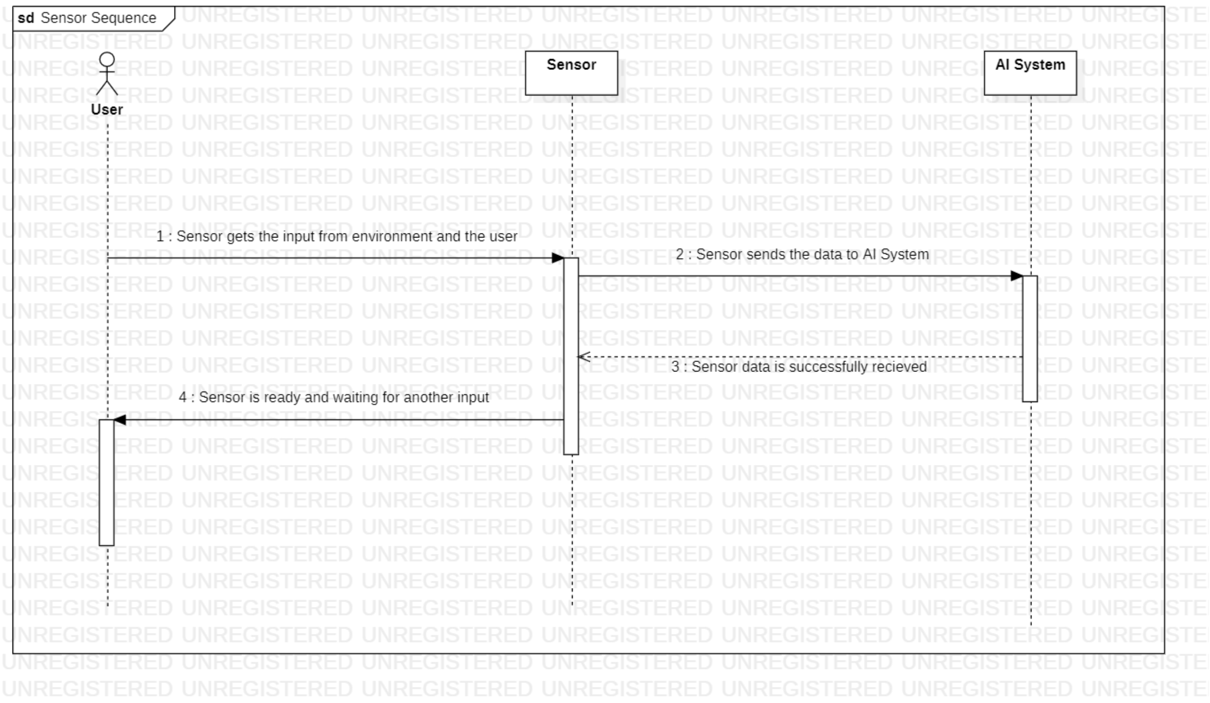

Figure 7: Sequence Diagram 1: Sensor Sequence

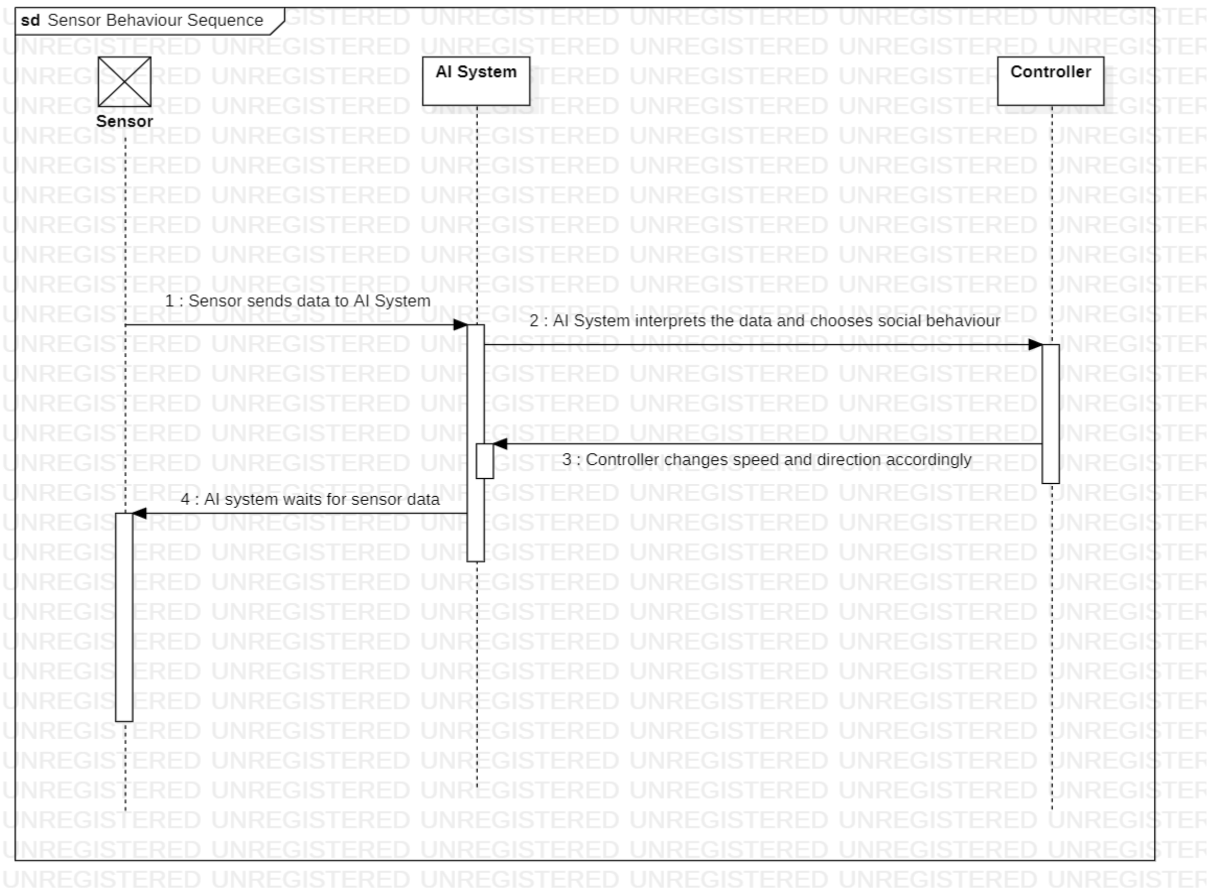

Figure 8: Sequence Diagram 2: Sensor - Behavior Sequence

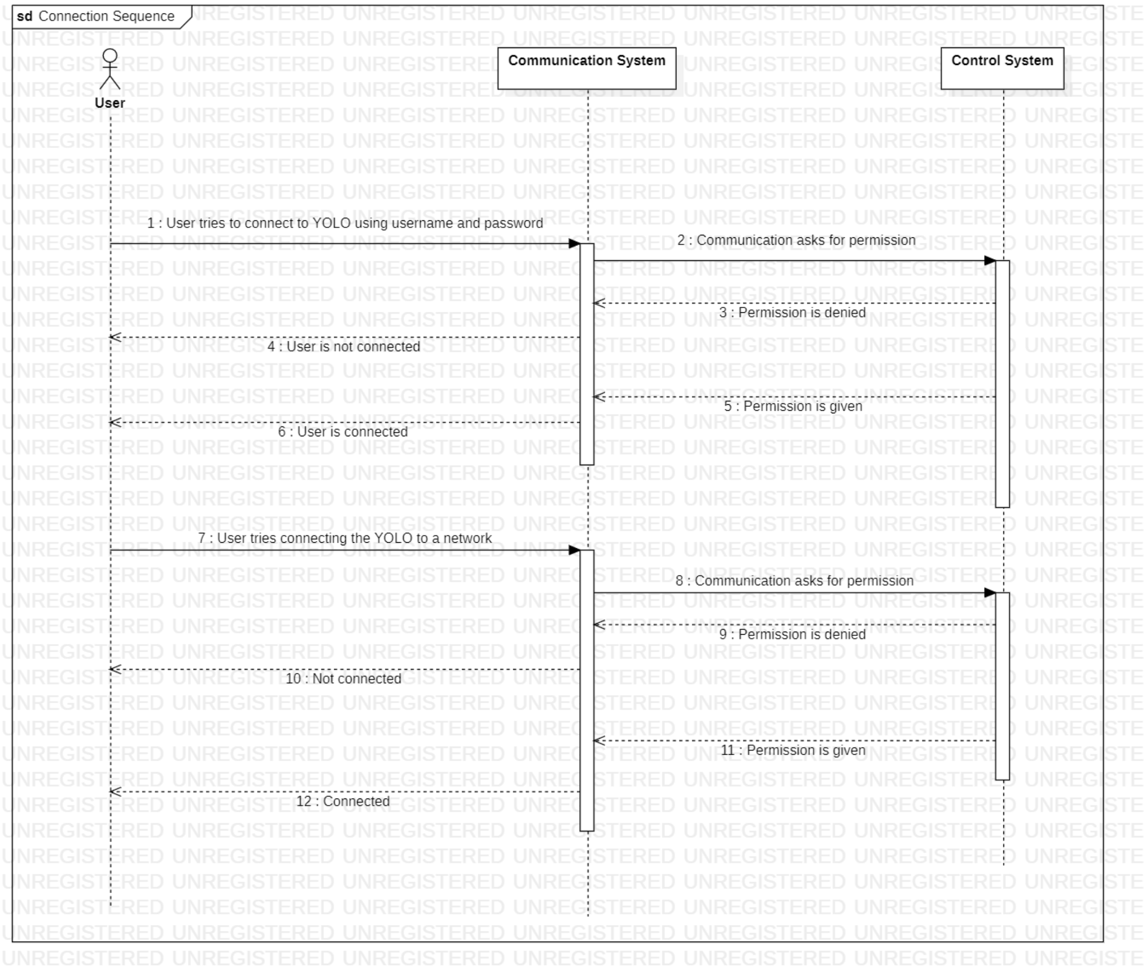

Figure 9: Sequence Diagram 1: Connection Sequence

4.3. Information View

4.3.1. Stakeholders' use of this view

Stakeholders can use information view to get a sense of the orientation of the database. Stakeholders can also see the collection of classes, interfaces, and associations. It can also be used to see the constraints between the classes.

4.3.2. Database Class Diagram

In this section, you can see the Database class diagram for the YOLO project.

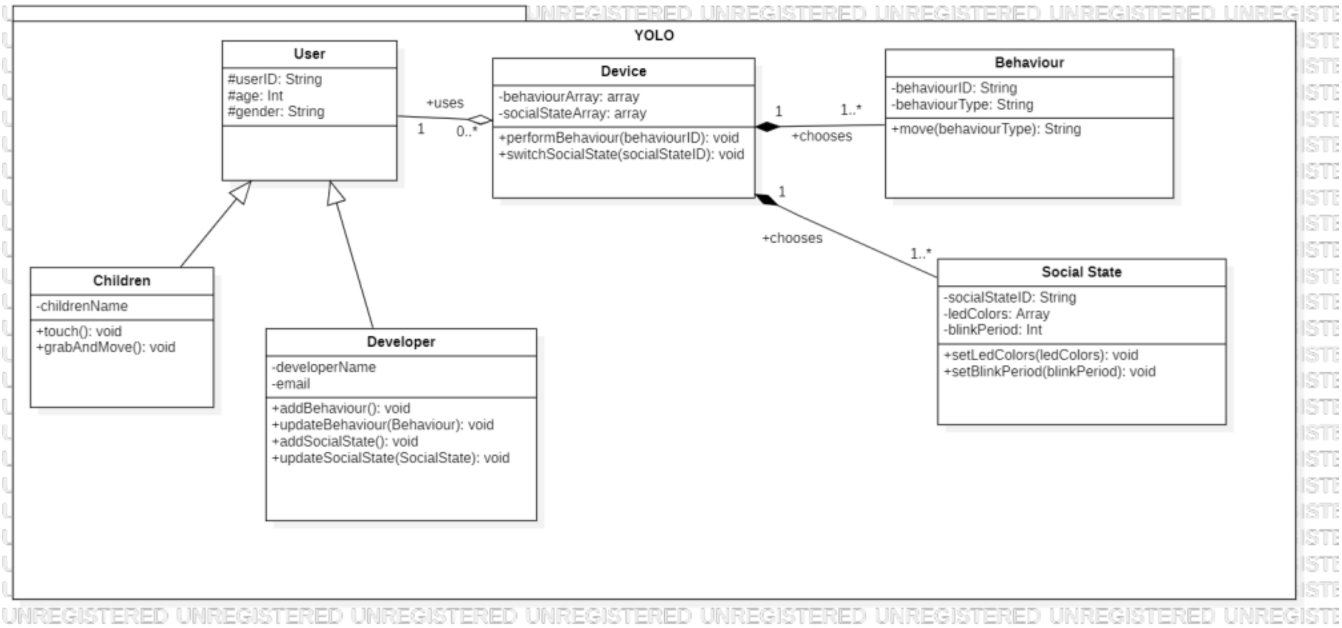

Figure 10: Database Class Diagram

4.3.3. Operations on Data

touch()	This function is used when user touches the YOLO.
grabAndMove()	This function is used when user grabs and moves the YOLO.
addBehavior()	This function is used when developer adds a behavior to YOLO.
updateBehavior()	This function is used when developer updates a behavior to YOLO.
addSocialState()	This function is used when developer adds a social state to YOLO.
updateSocialState()	This function is used when developer updates a social state to YOLO.
performBehavior()	This function is used when YOLO performs the behavior.
switchSocialState()	This function is used when YOLO switches social state.
move()	This function is used when YOLO moves according to Behavior.
setLedColors()	This function is used when YOLO sets Led Colors.
setBlinkPeriod()	This function is used when YOLO sets Blink Period of Leds.

Table 2: Operations on Data

4.4. Deployment View

4.4.1. Stakeholders' use of this view

Stakeholders, especially developers, use this view a lot. When they want to add a feature or modify the source code that they use this view to understand how the YOLO is executed and what is the architecture of this execution. They can also use this view to see the hardware and software environment for execution.

4.4.2. Deployment Diagram

This is the deployment diagram of YOLO. YOLO is running on Raspberry Pi. It is developed in the Linux operating system. Moreover, to be more specific when it is developing an IDE is used in order to ease the developing process. This device is connected to an API in order to access or change source code.

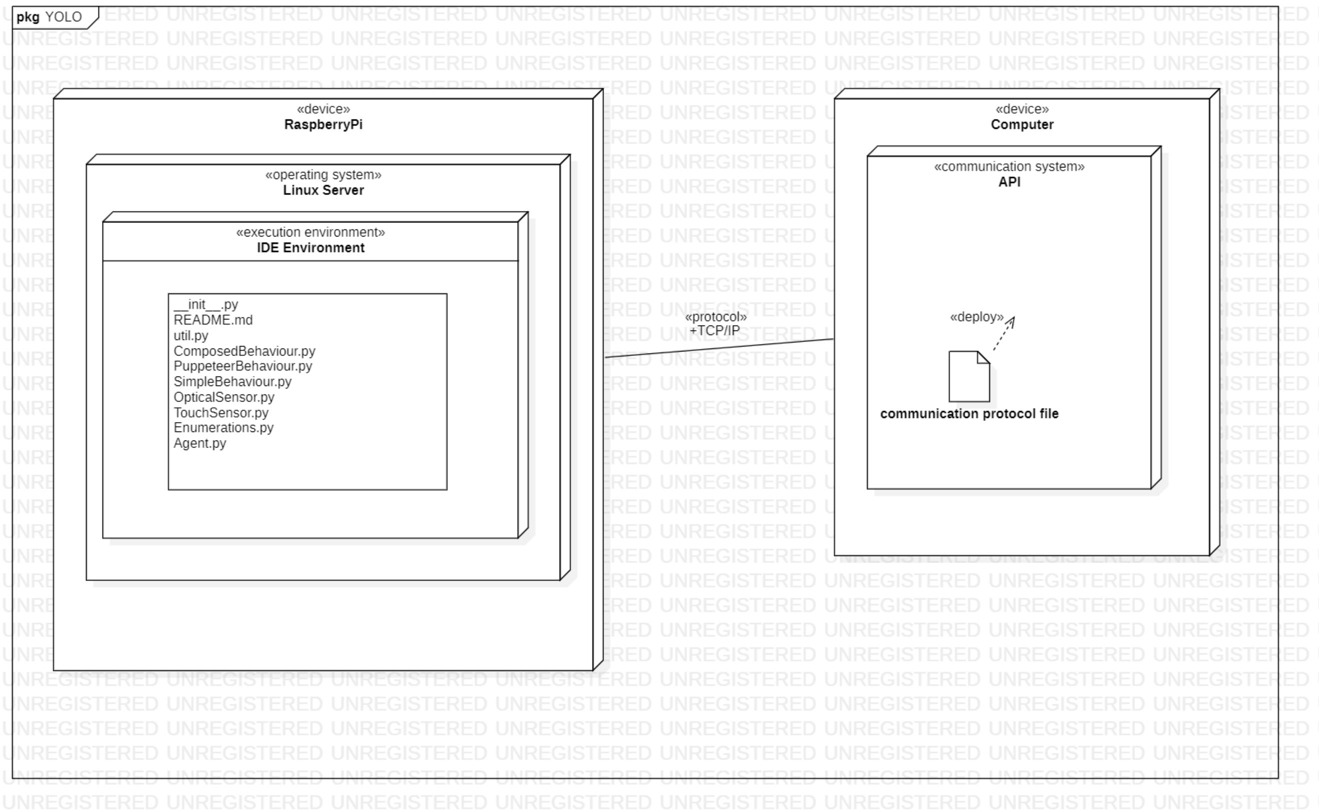

Figure 11: Deployment Diagram

4.5. Design Rationale

Context View:

- External interfaces are implemented in order to connect the internal interfaces with the environment.
- There are 2 actors in the context view, one of which is User and the other is Developer.

Functional View:

- Internal interfaces are implemented in order to accomplish the required actions by the developer.
- There are mainly 4 interfaces namely Sensors, AI System, Control System, and Communication System.

Information View:

- Database is implemented in order to store and change, if needed, relevant information which might be related to developer, user or components in the YOLO.

Deployment View:

- The YOLO is working on the Raspberry Pi and it is written using Python as a programming language.