California State University
Long Beach
Computer Science

Nao Sorter Specification

Submitted By:
Fair N. Aboshehwa
Christopher Yip
Anthony Guerra
Chau Trieu
Truc Tran

Submitted To:
Dr. Penzenstadler
Asst. Professor
Dept. of CECS

May 12, 2019

Contents

1 Executive Summary **2**
1.1 Abstract 2

2 Stakeholder Model **3**
2.1 Stakeholder Description 3
2.2 Stakeholder Matrix 3
2.3 Stakeholder Diagram 4

3 Goal Model **5**
3.1 Description 5
3.2 Model Diagram 5

4 System Vision **6**
4.1 Description 6
4.2 System Vision Diagram 7

5 Usage Model **8**
5.1 Use Case Overview 9
5.2 Description 9

6 Non Functional Requirements **10**
6.1 Process Requirements (Three) 10
6.2 Deployment Requirements (Five) 10
6.3 System Constraints Requirements (Five) 12
6.4 Quality Requirements (Five) 14

1 Executive Summary

1.1 Abstract

The NAO Sorter is a robot companion that can pick up and organize your mess for you. It knows where things are supposed to be placed and helps by picking up misplaced objects and putting them where they should be. The NAO Sorter provides various features that can be expanded upon. It offers vision recognition which allows the robot to classify and recognize objects which it has learned. The NAO Sorter will also offer a localization package that allows the robot to know where in the room it is as it walks around the room. The picking up and sorting objects feature can also be expanded upon and has tremendous commercial potential in allowing 3D sorting and placement of storage items.

2 Stakeholder Model

2.1 Stakeholder Description

The following stakeholder matrix and stakeholder diagrams are representations of entities that have an active interest in the project's success. The stakeholder matrix is a diagram that categorizes those entities into their function in the project as well as what type of interest the entity supports. The stakeholder diagram is a graphical representation of the matrix and the relationship between stakeholders and the NAO Sorter product.

2.2 Stakeholder Matrix

Stakeholder	**Role**	**Function**	**Active Interests**	**Priority**
Professor	1. Sponsor 2. Owner	1. Decision Maker 2. Information Provider	1. Provides NAO Robot 2. Would not like the robot to be damaged	- High
Developers	1. Staff	1. Programmer 2. End User 3. Decision Maker 4. Post-implementation Support	5. Creating functionality of the robot 6. Testing functionality of the robot. 7. Meeting all of the specified requirements.	- High
Customers	1. Users 2. Buyers	1. End Users	1. NAO robot to sort objects efficiently. 2. Provided functionality to work smoothly.	- High
Creators of NAO	1. Supplier 2. Potential Partner	1. Information Provider 2. Potential Decision Maker	1. Indirect interest in our product as it may potentially attract more to buy their robot.	- High.

Table 1: Stakeholder Table

2.3 Stakeholder Diagram

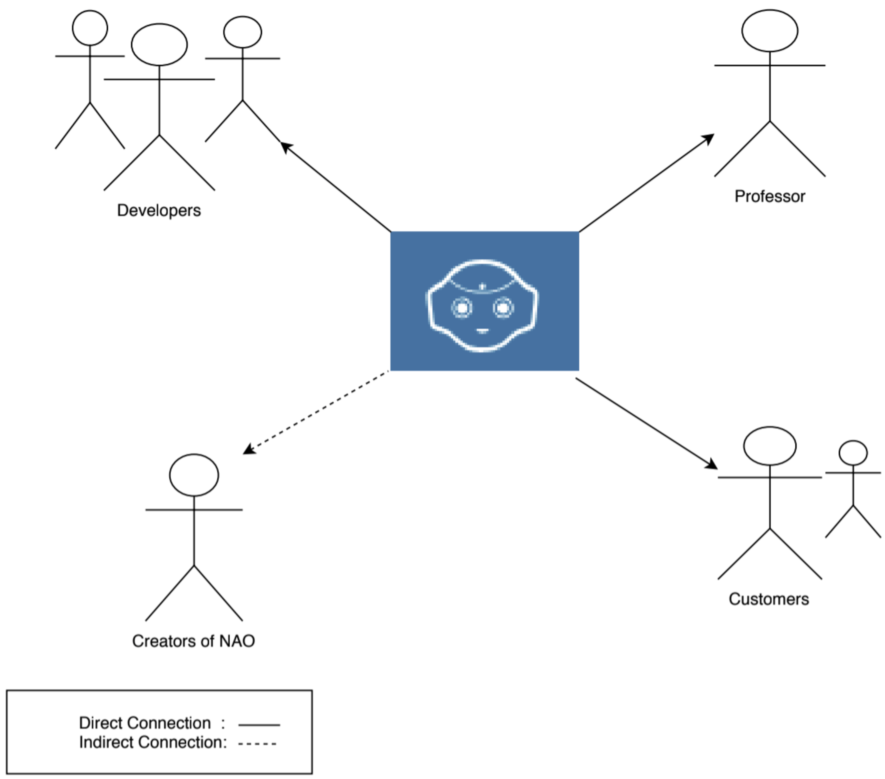

Figure 2. Stakeholder Diagram

3 Goal Model

3.1 Description

The Goal Model Diagram is representation our goals derived from stakeholders, organized into system goals, usage goals and business goals.

3.2 Model Diagram

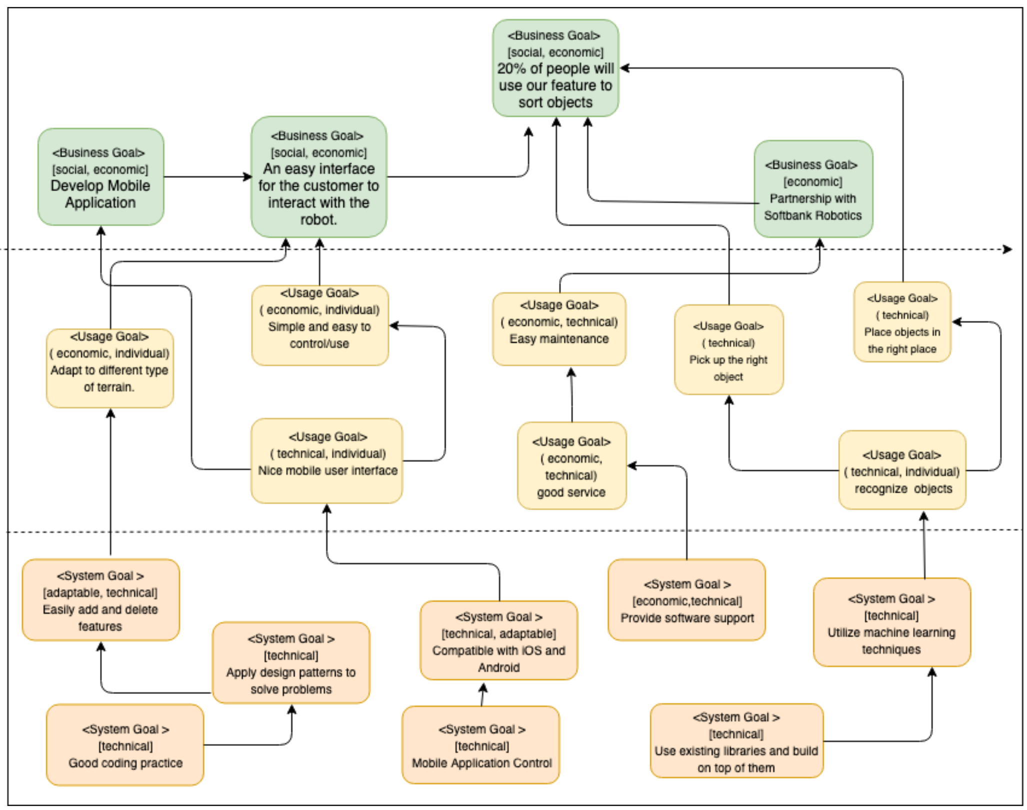

Figure 4. Goal Model

4 System Vision

4.1 Description

System designers are those who contribute to the development of the NAO Robot program; this includes researchers, advisers, and developers. The software program allows the NAO Robot to detect objects (square objects, circular objects) with different colors, pick them up, and place them in a specified location or container. Legal factors relevant to the program include copyright, EPA rules/guidelines, and other legal restrictions. After legal factors are dealt with, the robot will be ready for users to operate. The customer service team will support the users when they have questions or need some help with the product. The customer service team will forward users' feedback to the system designers. The system designers are the ones responsible for fixing errors and updating the software.

4.2 System Vision Diagram

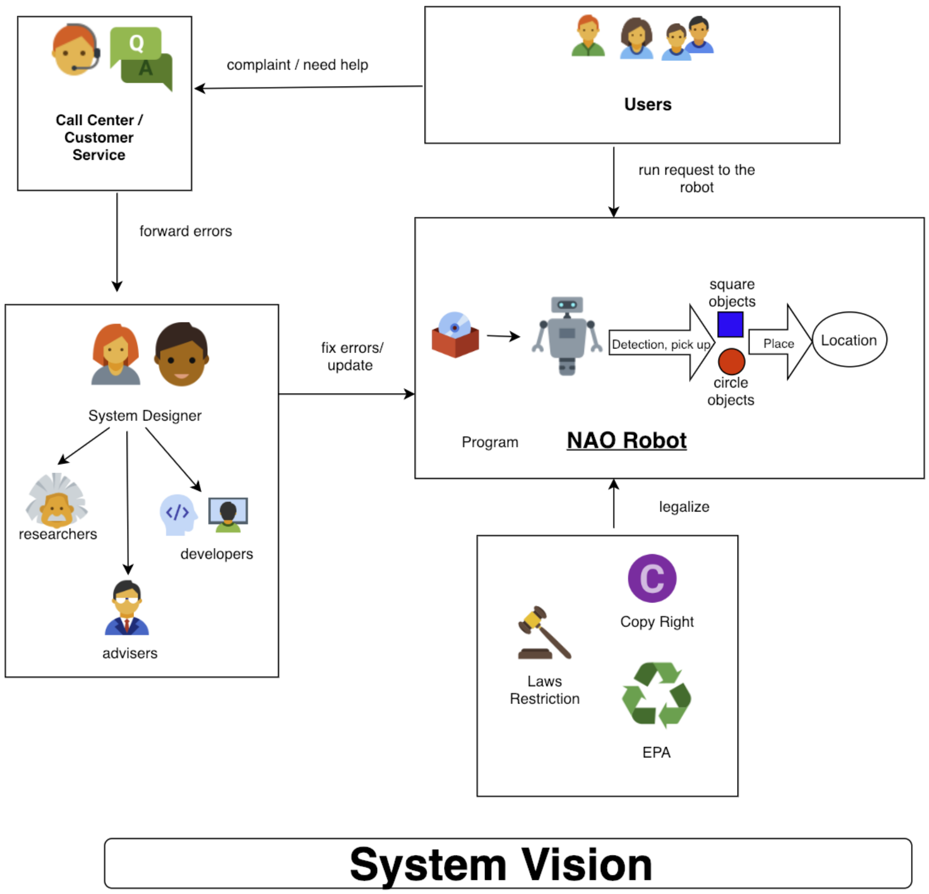

Figure 1. System Vision

5 Usage Model

Use Case	Characteristic Information	Main Success Scenario	Extensions
Pick Up Object	- Goal in Context: Go to and pick up the object that is recognized. - Scope: App and Robot - Level: Primary task. - Preconditions: Robot recognizes object to be picked up. - Success End Condition: Robot has object in its grasp. - Failed End Condition: Robot cannot reach object or pick up object. - Primary Actor: App User. - Secondary Actor: None - Trigger: User clicks app button. (Manual Mode) - Trigger: User clicks app button. (Manual Mode)	i. Robot looks with a certain radius for object. ii. Robot recognizes assigned object to be picked up. iii. Robot maneuvers to object. iv. Robot picks up object successfully.	- Types of objects that can be recognized are: Ball, Cube, Pyramid, etc...
Store Object	- Goal in Context: After picking up an object, move to receptacle with indicated label and place object inside receptacle. - Scope: Mobile app and robot. - Level: Primary task. - Preconditions: Robot is active, has picked up an object, a receptacle exists with a label that the robot can recognize. - Success End Condition: The object the robot was carrying is deposited within the indicated receptacle, and the robot has stepped away from the receptacle. - Failed End Condition: The robot has dropped the object but it is not in the indicated receptacle, or robot is inactive, or robot fails to find the receptacle. - Primary Actor: User. - Trigger User taps the "Store Object" button on the mobile app.	i. User taps the "Store Object" button on the mobile app. ii. Robot scans the area for the labelled receptacle corresponding to the shape of object picked up. iii. Robot identifies the appropriate receptacle. iv. Robot moves to receptacle. v. Robot stops at the receptacle in a position that it can drop the carried object in. vi. Robot drops object into receptacle. vii. Primary Actor: User viii. Robot steps back from receptacle.	- Square label, step 3, user chooses the "Square" label as receptacle of choice: Robot identifies the receptacle with the "Square" label. - Circle label, step 3, user chooses the "Circle" label as receptacle of choice: Robot identifies the receptacle with the "Circle" label. - Triangle label, step 3, user chooses the "Triangle" label as receptacle of choice: Robot identifies the receptacle with the "Triangle" label.
Tidy Room	- Goal in Context: Robot searches for recognizable objects, picks them up, and stores them in labeled container. - Scope: Mobile app and robot. - Level: Primary task. - Preconditions: Robot is active. Someone pressed the button on app. - Success End Condition: Robot stores every object to correct location. - Fail End Condition: There are objects in the room not stored. There are objects that are stored in the incorrect container. - Primary Actor: User - Trigger User presses button on app.	i. User presses tidy room button on app. ii. Robot starts walking around geofenced area looking for recognizable objects. iii. Robot recognizes object. iv. Robot picks up object. v. Robot looks for labeled container with the same label as the object. vi. Robot drops off object to labeled container. vii. Repeat steps 3-6 until all objects are in respective containers.	- Specify number of objects in app. - Specify type of objects to look for.

Table 2: Use Cases Matrix

5.1 Use Case Overview

5.2 Description

The Use-Case diagram is a graphical representation of the main functions that a user would interact with when using our product. It models the functionality of our system when (in this case) a user gives a command to the robot through one of the ways that it can be interacted with.

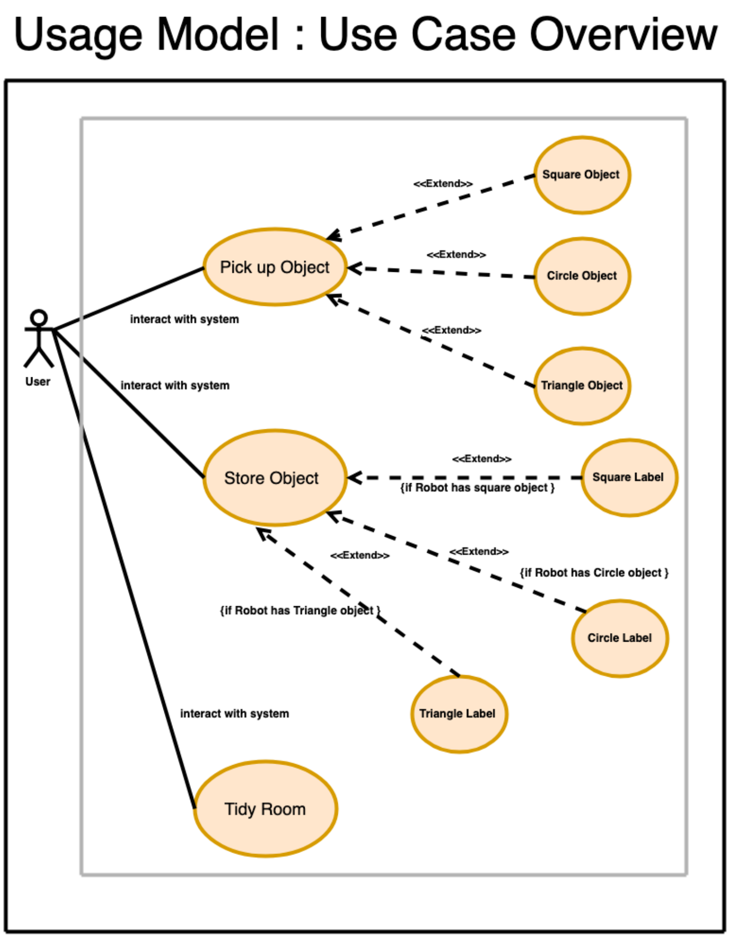

Figure 3. Use Case Diagram

6 Non Functional Requirements

6.1 Process Requirements (Three)

NFR	The software development process must be object oriented.
Rationale	Make software easy to extend and fix.
Satisfaction criterion	Every programmer can easily extend or fix the program.
Measurement	All programmer can extend functionality or add features to the software by simply adding more classes (and objects).
Risk	Extend or fix program cost too much time.

NFR	There must be a way of monitoring software changes.
Rationale	Software is easy to update and revert back to a previous state.
Satisfaction criterion	Changes to software is easy to track.
Measurement	Everyone in the team is able to see any modification to the software at any given time.
Risk	The developers will have a hard time knowing what changes to the software were made by another member, which will make software harder and riskier to modify.

6.2 Deployment Requirements (Five)

NFR	Software development must be use-case driven.
Rationale	By developing base on use-cases, the software is guaranteed to meet its requirements.
Satisfaction criterion	All the use-cases have been implemented.
Measurement	Assign each use case to a student and have them test the software if it fulfills the use case assigned.
Risk	The requirements are not met.

NFR	Publish the mobile app.
Rationale	Easier for user to tell robot to sort specific type of objects that are available.
Satisfaction criterion	People/ all classmates will be able download the app in Android/IOS app store.
Measurement	The app name will appear in the app store when people search.
Risk	Robot cannot receive the task require from users.

NFR	Upload/install software program to the robot.
Rationale	Make the program more efficiently to connect, update or install separately.
Satisfaction criterion	Be able to install the software to the robot.
Measurement	After installing, robot can run the program (access the object images the robot see and recognize the shape of its.)
Risk	Robot cannot sort objects. The whole project fail.

NFR	Mobile app must be compatible with Android and IOS.
Rationale	Anybody with an Android or IOS device can use the mobile app.
Satisfaction criterion	The mobile app will work how it's suppose to on both Android and IOS devices.
Measurement	Download the app on a random IOS and Android device and test its functionality.
Risk	The control of robot is limited to devices running a specific OS.

6.3 System Constraints Requirements (Five)

NFR	The robot must be connected to an available WiFi network to function.
Rationale	The robot receives instructions over WiFi to function.
Satisfaction criterion	The robot successfully connects to WiFi networks.
Measurement	The robot is able to connect to any WiFi network that it has access to.
Risk	The robot does not function.

NFR	The user's device must be connected to an available WiFi network.
Rationale	Control of the robot occurs through instructions sent from the controlling app on the user's device, over WiFi.
Satisfaction criterion	The user's device successfully connects to WiFi networks.
Measurement	The user's device is able to connect to any WiFi network it has access to.
Risk	Control of the robot is not possible.

NFR	The robot and the app must be connected to the same network.
Rationale	In order to establish a connection the devices must be on the same network.
Satisfaction criterion	Both the app and the robot connect successfully.
Measurement	Both devices connect within a certain time-frame (3 secs).
Risk	No communication between devices.

NFR	The WiFi must allow communication between devices.
Rationale	There need to be allowed communication between the app and the robot. CSULB WiFi for example wouldn't allow it without a static address.
Satisfaction criterion	Effective communication is established between devices.
Measurement	Both the app and the robot show that they are able to connect with each other.
Risk	Without communication we would not be able to control the robot.

NFR	The robot and the app device must be paired.
Rationale	In order to establish effective communication the devices must be paired to each other.
Satisfaction criterion	Both devices connect to each other.
Measurement	Both the app and the robot show that they are connected to each other. (Validated by both).
Risk	Without communication we would not be able to control the robot.

6.4 Quality Requirements (Five)

NFR	Nice mobile user interface.
Rationale	A pleasing interface results in greater user satisfiability.
Satisfaction criterion	Most users find the UI pleasing.
Measurement	Test with classmates as users.
Risk	An unpleasing UI can lead to a drop in users.

NFR	Easy to use and control.
Rationale	Performance is tied to ease of use.
Satisfaction criterion	90% of users have a handle on the controls available to control the robot.
Measurement	Test with fellow students from class.
Risk	Without ease of use the main bridge between user and robot is broken.

NFR	Easy maintenance.
Rationale	The robot should be easy to maintain.
Satisfaction criterion	User can get the robot fix or repair cheap and quick.
Measurement	All the robot parts should be common and standardized replacement parts.
Risk	Certain parts are difficult to get to, it increase labor costs.

NFR	Recognize and pick up the right object.
Rationale	It's unsatisfied if the robot cannot recognize or can not pick up the right object.
Satisfaction criterion	Robot is able to recognize and pick the right object.
Measurement	Compare the object robot pick up with the object user command.
Risk	The robot might recognize and pick up the wrong object.

NFR	The objects that are picked up and placed are not damaged.
Rationale	Damaging the objects would be costly for the user and damaging to reputation.
Satisfaction criterion	The objects the robot interacts with are not damaged.
Measurement	Physically checking all objects it interacts with after the simulation is done for signs of tear.
Risk	The robot might damage something expensive or important to the user.