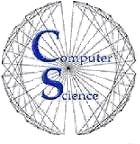
**“My Friendly Pintos”**

**A Multi-Service Robot Cluster**

Version 1.2

Software Functional Specifications

|  |  |
| --- | --- |
| **Team:** | Abhishek Bais, Chris Rehfeld, Charles Le and Akshay Hegde |
| **Team Names:** | Plz Send The Codes |
| **Date:** | 07/08/2012 |



July 08, 2012

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| Date | Version | Description | Author |
| June 15, 2012 | 1.0 | Functional Specifications and Use Case descriptions of “My Friendly Pintos”,  A Multi Service-Robot Cluster. | Plz Send The Codes team |
| June 30, 2012 | 1.1 | 1. Incorporated changes to highlight where “*My Friendly Pintos*” can be put to use.  2. Changes in the document structure.  3. Added Business Rules | Plz Send The Codes team |
| July 08, 2012 | 1.2 | 1. Updated the CRC Cards. | Plz Send The Codes team |

# Table of Contents

[Table of Contents 2](#_Toc77487619)

[List of Figures 3](#_Toc77487620)

[1.0. Introduction 4](#_Toc77487621)

[1.1. Goal 4](#_Toc77487622)

[1.2.Summary 4](#_Toc77487623)

[1.3. References 5](#_Toc77487625)

[2.0. System Functional Requirements 6](#_Toc77487627)

[2.1 Overview – Senior Home Environment 6](#_Toc77487628)

[2.2 System Environment 7](#_Toc77487628)

[2.3 Functional Requirements Specification 8](#_Toc77487629)

[2.3.1 Elderly Use Case 8](#_Toc77487630)

[Feature: Issue Request/Communicate](#_Toc77487631) 9

[Feature: Add Item](#_Toc77487631) 10

[Feature: Get Item](#_Toc77487633) 11

[Feature: Cancel Request](#_Toc77487633) 12

[Feature: Help Desk](#_Toc77487633) 13

[Feature: Check Status](#_Toc77487633) 14

[Feature: View History](#_Toc77487633) 15

[Feature: Erase History](#_Toc77487633) 16

[2.4 User Characteristics](#_Toc77487648) 16

[2.5 Non-Functional Requirements](#_Toc77487649) 17

[3.0. Detailed Use Cases](#_Toc77487650) 17

[3.1 External Interface Requirements](#_Toc77487651) 17

[3.2 Detailed Use Case Descriptions 18](#_Toc77487652)

[3.2.1 Use Case: Add Item](#_Toc77487653) 18

[3.2.2 Use Case: Get Item](#_Toc77487654) 19

[3.2.3 Use Case: Cancel Request](#_Toc77487656) 21

[3.2.4 Use Case: Help Desk](#_Toc77487656) 22

[3.2.5 Use Case: Check Status](#_Toc77487656) 24

[3.3 Detailed Non-Functional Requirements](#_Toc77487665) 26

[3.3.1 Logical Structure of the Data](#_Toc77487666) 26

[4.0. Business Rules 28](#_Toc77487650)

[5.0. System Design 30](#_Toc77487650)

[5.1 CRC Cards](#_Toc77487651) 30

[5.2 UML Diagram 32](#_Toc77487652)

[5.3 Sequence Diagrams](#_Toc77487651) 33

[5.3.1. Use Case: Add Item](#_Toc77487653) 34

[5.3.2. Use Case: Get Item](#_Toc77487654) 35

[5.3.3. Use Case: Cancel Request](#_Toc77487653) 36

[5.3.4 Use Case: Help Desk](#_Toc77487654) 37

[5.3.5 Use Case: Check Status](#_Toc77487654) 38

[5.0. Glossary 39](#_Toc77487650)

[6.0. Attachments](#_Toc77487650) 40

[6.1 Approval Signatures](#_Toc77487651) 40

List of Figures

[Figure 1 - System Environment](#_Toc77487669) 8

[Figure 2 – Functional Specification Diagrams 10-17](#_Toc77487670)

[Figure 3 – Detailed Use Case Diagrams](#_Toc77487671) 19-25

[Figure 4 - Logical Structure of the Pinto Manager Data](#_Toc77487672) 26

[Figure 5 – System Design - UML Diagram](#_Toc77487672) 33

[Figure 6 – System Design - Sequence Diagram - Use Case: AddItem](#_Toc77487672) 34

[Figure 7 – System Design – Sequence Diagram - Use Case: GetItem](#_Toc77487672) 35

[Figure 8 - System Design – Sequence Diagram - Use Case: Cancel Request](#_Toc77487672) 36

[Figure 9 - System Design – Sequence Diagram - Use Case: Help Desk](#_Toc77487672) 37

[Figure 10 - System Design – Sequence Diagram - Use Case: Check Status](#_Toc77487672) 38

# 1.0. Introduction

## 1.1. Goal

#### The goal of this document is to present a detailed description and the functional requirement specifications of “*My Friendly Pintos*” - A multi service-robot cluster system. It elaborates the features of the system, the interfaces of the system, what the system does, and the environmental constraints under which the system operates. It also presents how the system responds to external stimuli.

#### This document is intended for both the stakeholders and the developers of the system The designed system will be presented as a cost-effective assistance and support solution to the elderly society of San Jose County for their due consideration and adoption in their homes and also to the management team of various senior homes and hospitals in the city.

#### The second chapter of this document, “*The System Functional Requirements*” section, gives an overview of the functionality of the system. It describes the functional requirements and is used to establish a context for the “*Detailed Use Case*” descriptions in chapter three.

#### The third chapter of this document, the “*Detailed Use Case*” description section, is written primarily for the developers of the system. It describes in technical terms how the system responds to different external stimuli.

#### 

## 1.2. Summary

The concept of service robots, machine equivalents to humans, “*humanoids*”, capable of interacting and performing multiple physical tasks with little or no human support has always intrigued mankind. For several decades, such service robots have been the subject of novels, movies, cartoon, and “*visionary* *futuristic projects”* of technology companies. Only recently have they been introduced as consumer products.

As service robots become more and more commercially viable, they are being introduced in a number of different setting such as homes, senior homes and hospitals. In these settings they perform varied tasks such as vacuum cleaning, keep homes secure and mow the lawn etc.

In the past decade continual and rapid advancements in the field of medicine have led to increased average lifespan and a sharp rise in health care costs. There has also been a shift towards individualistic societies. We at “***Plz Send The Codes***”, a fast rising startup, envision a huge demand for service robots in the years to come.

Having analyzed the nature of requirements pertaining to assistive robotics designed for the elderly, we assert that traditional "***industrial***" robot designs are neither commercially viable, nor do they meet usability standards required for the “***home, senior home or hospital*** “demography.

This software system aims to design and develop “*Pinto*”, an intelligent service robot to assist and support the elderly in their daily needs in a senior home setting. With its advanced navigation and collision avoidance capabilities coupled with the ability to naturally communicate and perform elderly support and care tasks independently with no human assistance, we believe that “*Pinto*” will be an unprecedented success- becoming a household name.

## 

## 1.3. References

[Meng, Q, Lee, M.H.](http://ieeexplore.ieee.org/search/searchresult.jsp?searchWithin=p_Authors:.QT.Meng,%20Q..QT.&newsearch=partialPref)Dept. of Computer. Sci., Univ. of Wales, Aberystwyth, UK

[Learning and control in assistive robotics for the elderly](http://ieeexplore.ieee.org/xpl/login.jsp?tp=&arnumber=1438894&url=http%3A%2F%2Fieeexplore.ieee.org%2Fiel5%2F9824%2F30991%2F01438894.pdf%3Farnumber%3D1438894)

**This paper appears in:**

[Robotics, Automation and Mechatronics, 2004 IEEE Conference on](http://ieeexplore.ieee.org/xpl/mostRecentIssue.jsp?punumber=9824)  
**Date of Conference:** 1-3 Dec. 2004

**Volume:**1   
**Page(s):** 71 - 76 vol.1   
**Product Type:** Conference Publications

**2.0. System Functional Requirements**

***2.1 Overview – Senior Home Environment***

The demography of a senior home is worlds away from the laboratory, space or battlefield – the most common domain for human-robot interaction. Most assumptions and requirements for these domains do not readily translate to the senior home. In some cases they’re wholly inappropriate.

In particular, we must be attentive to the material culture of the senior home, where there are not “*users*” but rather “*moveable* *objects*” like pets, people, other pintos and “*immovable objects*” like furniture literally in co-habitation with each other.

In the following chapters, this document will provide a detailed description of the use model of a service robot by the elderly in a senior home setting, the social complexities that exist, and how the “*My Friendly Pintos*” – A multi robot cluster system, delivers in their presence.

The team at “***Plz Send The Codes***” understand that the demography of a “*true*” senior home will consist of several elderly, some support staff, several “*Pintos*”, a larger number of movable and immovable obstacles and will involve tons of other complexities.

The system we describe below is a “*pilot system* ” to present to the management team of various senior homes and hospitals in the city and to the elderly society at large in the San Jose County to demonstrate how, commercially viable - “Pintos” can changes their lives.

Our system, therefore, has one primary actor, the” *Elderly*” who has multiple “*Pintos*” at his disposal to perform various tasks for him. This elderly also performs duties like “***Add Item***” to the Senior Home himself, which in a “*true*” senior home setting would be performed by the “*Senior Home Support Staff*”. There are other home occupants but they do not interact with the “*Pintos*”.

## 2.2 System Environment

Figure 1 - System Environment

#### The “*My Friendly Pintos*” – Multi Service-Robot Cluster software system has two active actors and one cooperating system.

The primary actor, here “*Elderly*” accesses the Multi Service-Robot Cluster through the Communication System, here “*Pinto Manager*”. Any elderly communication with the system happens through textual or voice commands.

The Pinto Manager has access to all Service-Robots – “Pintos”.

The Multi Service-Robot Cluster is constrained by the senior home environment.

A senior home environment has various obstacles, “*movable*”, such as other home occupants, other Pintos, and “*immovable*”, such as furniture.

The Pintos navigate obstacles and retrieve “*Items*” for the main Actor.

The secondary actor- us at [*www.Plz\_Send\_The\_Codes.com*](file:///C:\Documents%20and%20Settings\abais\My%20Documents\Downloads\www.Plz_Send_The_Codes.com), provides customer support which includes the ability to file usability complaints of the system, the ability to register the system with us, and the ability to get periodic software and feature updates.

The following chapters elaborate the various features of the system modeled as Elderly Use Cases.

## 2.3 Functional Requirements Specification

#### This section outlines the main features that the system must provide. Each feature is presented as an interaction between the primary actor, the “*Elderly*” and the system, “*My Friendly Pintos*”.

#### The system broadly “*provides*” the primary actor three types of features categorized below:-

1. **Interaction Features to provide for “*Elderly – My Friendly Pintos”* communication.**

#### The system must “*provide*” the primary actor, the “*Elderly”,* access to each one of the service request features listed under (ii) to get “*Assistance and Support”.*

#### The “*Issue Request/Communicate*” features serve this purpose. It supports two modes of communication between the “*Elderly and My Friendly Pintos*” namely, “*voice input*” and “*textual input*”.

1. **Service Features to provide for “*Elderly - Assistance and Support*”.**

#### The system must “provide” the primary actor, the “*Elderly*”, various service features to get “*Assistance and Support*”.

#### The “*Add Item”, “Get Item”, “Cancel Request”, “Help Desk”, “Check Status”, “View History”, “Erase History*” are the services “*My Friendly Pintos*” must provide.

1. **Helper Features to provide for “*Elderly – Customer Support”.***

#### The system must “*provide*” the primary actor, the “*Elderly”* various helper features for customer support as and when the need arises.

#### The “*Help Desk”* feature broadly provides the “*Elderly*” the ability to *“File Complaint”, “Register System”* and “*Get Software Update*” from the secondary actor, us at [*www.Plz\_Send\_The\_Codes.com*](http://www.Plz_Send_The_Codes.com).

### 2.3.1 Elderly Use Case

**Feature:**  Issue Request/Communicate

**Diagram:**

Elderly

Issue Request

**Brief Feature Description**

The elderly accesses the Multi Service-Robot Cluster system with a service/support request. The system must provide an easy to use interface for the elderly to communicate with the system.

The system must support “*voice*” and “*text*” input as modes to accept these requests. The system must be able to establish the connection with the elderly or respond why it cannot establish a connection with the elderly.

**Initial Step-By-Step Description**

Before this feature can be used, the Elderly has read the “User Guide” and familiarized himself with the various features the Multi Service-Robot Cluster System provides.

1. The Elderly establishes a connection the system by voice command or text input.
2. The system responds with a greeting, “*Welcome!! Pintos at your service. How can we help you?*”
3. The Elderly chooses to view items in his home by name, category, or keyword.
4. The system displays the choices to the Elderly.
5. The Elderly responds, “*Great, This Works. I do not need anything now. I’m done!”*
6. The system responds with a greeting, “*You’re welcome. Good Bye.*”

**Feature:**  Add Item

**Diagram:**

Elderly

Add Item

**Brief Feature Description**

The elderly accesses the Multi Service-Robot Cluster system to add/replace an item in his home that he may in the future ask Pinto to fetch. The system must provide an easy to use interface for the elderly to register an item with the system. The system must be able to add the item to its list, or respond why it cannot add item to its list.

**Initial Step-By-Step Description**

Before this feature can be used, the Elderly has read the “User Guide” and familiarized himself with the various features the Multi Service-Robot Cluster System provides

and

The Elderly has already established a connection with the Multi Service-Robot Cluster System using the “*Issue Request/Communicate*” feature.

and

The Elderly has brought the item to his house.

1. The Elderly chooses to register a new item to the list of items the Multi Service-Robot Cluster system recognizes.
2. The system responds with “*Please enter item name, description and location.*”
3. The Elderly provides the name, description and location of the item.
4. The system responds with “*Your item has been added to the list.*”

**Xref:** Section 3.2.1, Detailed Use Case: Add Item

**Feature:** Get Item

**Diagram:**

Elderly

Get Item

**Brief Feature Description**

The elderly accesses the Multi Service-Robot Cluster system to get an item in the senior home. The system must provide an easy to use interface for the elderly to instruct the system which item to retrieve. The system must be able to get the item requested, or respond why it cannot get the item.

**Initial Step-By-Step Description**

Before this feature can be used, the Elderly has read the “User Guide” and familiarized himself with the various features the Multi Service-Robot Cluster System provides

and

The Elderly has already registered the item with the Multi Service-Robot Cluster System using the “*Add Item*” feature.

and

The Elderly has already established a connection with the Multi Service-Robot Cluster System using the “*Issue Request/Communicate*” feature.

1. The Elderly chooses to get an item from the list of items the Multi Service-Robot Cluster system recognizes.
2. The system provides feedback indicating how soon it can start fulfilling the request.
3. The system brings the item to the Elderly, prompting him to take it.

**Xref:** Section 3.2.2, Detailed Use Case: Get Item

#### Feature: Cancel Request

**Diagram:**

Elderly

Cancel Request

**Brief Feature Description**

The elderly accesses the Multi Service-Robot Cluster system to cancel a previously submitted request. The system must provide an easy to use interface for the elderly to instruct the system which request to cancel. The system must be able to immediately cancel the previous request and respond back to the Elderly when the cancellation is done or respond why it cannot cancel the previous request.

**Initial Step-By-Step Description**

Before this feature can be used, the Elderly has read the “User Guide” and familiarized himself with the various features the Multi Service-Robot Cluster System provides

and

The Elderly has already registered the item with the Multi Service-Robot Cluster System using the “*Add Item*” feature.

and

The Elderly has already established a connection with the Multi Service-Robot Cluster System using the “*Issue Request/Communicate*” feature.

and

The Elderly has already submitted a request with the Multi Service-Robot Cluster System using one of the service or help features.

1. The Elderly chooses to *Cancel* a previously submittedrequest.
2. The system immediately halts the previous request.
3. The system responds “*Ok, canceled request*” when the task cancellation is complete.

**Xref:** Section 3.2.3, Detailed Use Case: Cancel Request

#### Feature: Help Desk

**Diagram:**

Elderly

Help Desk

**Brief Feature Description**

The elderly accesses the Multi Service-Robot Cluster system to get assistance/customer support. The system must provide an easy to use interface for the elderly to get assistance. The system must be able service this request at any time. The system must be able to record the Elderlys’ complaint and take action on it. The system must be able to register the software system and download latest features/software updates from [www.Plz\_Send\_The\_Codes.com](http://www.Plz_Send_The_Codes.com), if the help desk has been contacted for that purpose.

**Initial Step-By-Step Description**

Before this feature can be used, the Elderly has read the “User Guide” and familiarized himself with the various features the Multi Service-Robot Cluster System provides

and

The Elderly has already established a connection with the Multi Service-Robot Cluster System using the “*Issue Request/Communicate*” feature.

1. The Elderly chooses to contact *Help Desk*, to “*File Complaint*”/”*Register System*”/”*Get Software Update*”.
2. The system responds “*Please record your message*”.
3. The Elderly records his message.
4. The system sends the message and diagnostic information to the help desk.

**Xref:** Section 3.2.4, Detailed Use Case: Help Desk

#### Feature: Check Status

**Diagram:**

Elderly

Check Status

**Brief Feature Description**

The elderly accesses the Multi Service-Robot Cluster system to check the status of the system, to gain info on queued and executing requests. The system must provide an easy to use interface for the elderly to query the system for submitted requests. The system must be able service this request at any time. The system must be able to respond with the detailed status of the submitted requests or respond why it cannot get the status.

**Initial Step-By-Step Description**

Before this feature can be used, the Elderly has read the “User Guide” and familiarized himself with the various features the Multi Service-Robot Cluster System provides

and

The Elderly has already registered the item with the Multi Service-Robot Cluster System using the “*Add Item*” feature.

and

The Elderly has already established a connection with the Multi Service-Robot Cluster System using the “*Issue Request/Communicate*” feature.

and

The Elderly has already submitted a request with the Multi Service-Robot Cluster System using one of the service or help features.

1. The Elderly chooses to *Check Status* of the system.
2. The system displays detailed information of all currently executing and queued requests.

**Xref:** Section 3.2.5, Detailed Use Case: Check Status

#### Feature: View History

**Diagram:**

Elderly

View History

**Brief Feature Description**

The elderly accesses the Multi Service-Robot Cluster system to views the history of all requests he has submitted so far. The system must be able to must be able to respond with a history report or respond why it cannot display the history report.

**Initial Step-By-Step Description**

Before this feature can be used, the Elderly has read the “User Guide” and familiarized himself with the various features the Multi Service-Robot Cluster System provides

and

The Elderly has already established a connection with the Multi Service-Robot Cluster System using the “*Issue Request/Communicate*” feature.

1. The Elderly chooses to *View History* of all requests previously submitted.
2. The system displays all requests previously submitted in a neat format sorted by year, month and date with delivery status.

#### Feature: Erase History

**Diagram:**

Elderly

Erase History

**Brief Feature Description**

The elderly accesses the Multi Service-Robot Cluster system to erase the history of all requests he has submitted so far. The system must be able to must be able to respond with a cleared out history report and a message that history has been cleared or respond why it cannot display the cleared history report.

**Initial Step-By-Step Description**

Before this feature can be used, the Elderly has read the “User Guide” and familiarized himself with the various features the Multi Service-Robot Cluster System provides

and

The Elderly has already established a connection with the Multi Service-Robot Cluster System using the “*Issue Request/Communicate*” feature.

1. The Elderly chooses to *Erase History* of all submitted requests.
2. The system erases the history of all requests it has processed so far.
3. The system responds with a message “*Your request history has been cleared*”.

## 2.4 User Characteristics

#### The Elderly is expected to be “*Basic Gadget Literate*” to be able to communicate with the Multi Service-Robot Cluster System.

## 2.5 Non-Functional Requirements

## 2.5.1 Robustness

### The system needs to ensure it is able to either complete a task, or exit gracefully and predictably. Any exceptional circumstances should trigger a graceful exit with user feedback having been provided.

### 2.5.2 Performance

The system must offer concurrency in executing assigned tasks. The system must ensure each Pinto is able perform his task seamlessly, and in the shortest possible time.

### 2.5.3 Ease of Use

The systemis being designed and developed for the Elderly. It is important that the system be extremely user friendly and has simple user input methods, either “*command line text*” or “*voice input*”. Interaction with the system should be intuitive and natural. Ideally, the system is “fun” to use.

### 2.5.4 Extensibility

The system mustprovide “*Assistance and Support*” to the elderly. As it becomes popular and more elders begin to use it, their needs and requirements may grow.

The system should be easily extensible so that more features and options can be easily added to it in times to come.

# 3.0. Detailed Use Cases

# *3.1 External Interface Requirements*

# The only link to an external system is the link to the “[www.Plz\_Send\_The\_Codes.com](file:///C:\Documents%20and%20Settings\abais\My%20Documents\Downloads\www.Plz_Send_The_Codes.com)” website to contact the help desk to Register the Multi Service-Robot Cluster System, Check for periodic software updates or File Complaints and/or receive customer support for operational issues as and when encountered.

## 3.2 Detailed Use Case Descriptions

The Logical Structure of the Data is contained in Section 3.3.1.

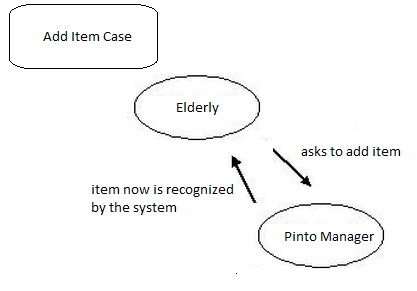
### Use Case: Add Item

**Brief Description**

The Elderly accesses the Multi Service-Robot Cluster system to set the location of an item, so that it may be retrieved by the system sometime in the future.

**Xref:** Section Feature: Add Item

|  |  |
| --- | --- |
| **Use Case Name** | Add Item |
| **Actors** | Elderly |
| **Trigger** | An Elderly Verbally addresses the System by saying "Pinto Manager, note the location of <item name and description>" |
| **Precondition** | The Elderly is audible by the Pinto Manager.  The item is in the house. |
| **Basic Path** | 1) The Elderly says "Pinto Manager, note the location of <item name and description>"  2) Pinto Manager adds the item to its list of recognized items.  3) Pinto Manager says "<item name> recorded. I can now retrieve it for you anytime you want."  4) end |
| **Postcondition** | The list of recognized items has a new entry |
| **Other** | A Pinto can transport most items that an Elderly would be capable of lifting, so all items are accepted to the list. |

****

The above is a diagram representation of the Elderly –My Friendly Pintos Add Item interaction

### Use Case: Get Item

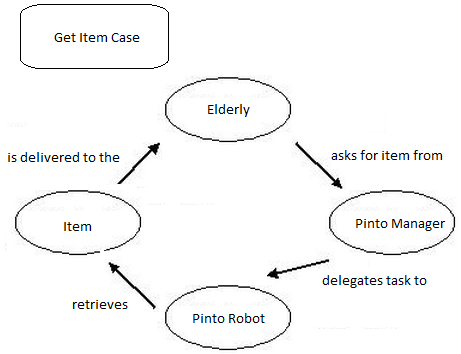
**Brief Description**

The elderly accesses the Multi Service-Robot Cluster system and issues a request to have an item brought to him. The system guides the user through the interaction, supporting usability.

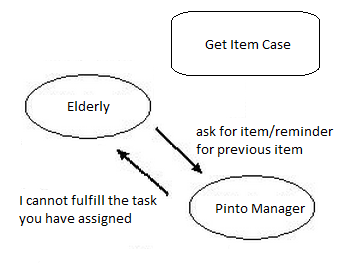
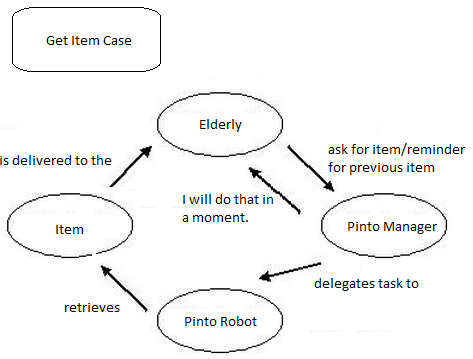
**Xref:** Section Feature: Get Item

|  |  |
| --- | --- |
| **Use Case Name** | Get Item |
| **Actors** | Elderly |
| **Trigger** | An Elderly Verbally addresses the system by saying "Pinto Manager, get me <item>" where <item> is the name of an item. |
| **Precondition** | The Elderly is audible by the Pinto Manager  The item is in the Pinto Managers list of retrievable items. |
| **Basic Path** | 1) Elderly says "Pinto Manager, get me <item>"  2) Pinto Manager internally verifies that it is capable of retrieving that specific item  3) Pinto Manager accepts the task as something it can fulfill  4) Pinto Manager enqueues the task  5) Pinto Manager verifies there is an idle Pinto to perform the task.  6) Pinto Manager says "Ok, I will get that right now"  7) A Pinto retrieves the item, bringing it directly to the location of the Elderly.  8) Pinto Manager says "Take your item"  9) The Elderly takes the item from the pinto  10) The Pinto returns to the staging area and becomes idle  11) end |
| **Alternative Path A** | In Step 5, there are no idle Pintos  6A1) Pinto Manager says "I will do that in a moment, all my Pintos are busy right now"  6A2) Some time passes, and a Pinto becomes idle  6A3) Continue with step 7 |
| **Alternative Path B** | In Step 7, the Pinto cannot retrieve the item(unforeseen reason)  7B1) Pinto Manager says "I'm sorry, but I cannot get your <item>"  7B2) The Pinto returns to its staging area, and becomes idle  7B3) The task is cancelled.  7B4) end |
| **Alternative Path C** | In Step 7, the Pinto cannot reach the Elderly  7C1) Pinto Manager says "I'm sorry, but I cannot reach you. I will wait with your item at <location>"  7C2) some time passes, and eventually either the flow continues with step 9, or the Elderly initiates the "Cancel Task" use case.  7C3) end |
| **Postcondition** | The Pinto is idle |
| **Other** | The Pinto will attempt to move to wherever the current location of the Elderly is once it has picked up the item. This is useful if the Elderly moves around while waiting for the item retrieval to complete. |

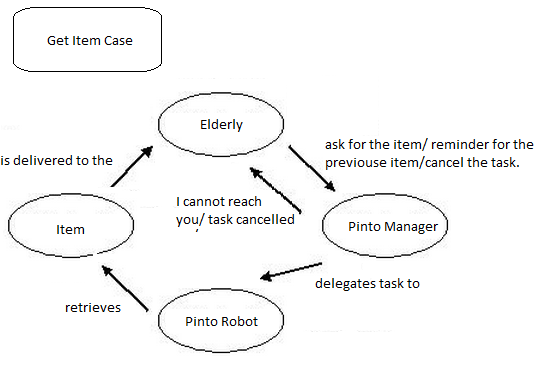
**Base Case**

. 

**Variation A Variation B**



**Variation C**

****

The above is a diagram representation of the Elderly –My Friendly Pintos Get Item interaction

### Use Case: Cancel Request

**Brief Description**

The Elderly cancels a request that was previously submitted

**Xref:** Section Feature: Cancel Request

|  |  |
| --- | --- |
| **Use Case Name** | Cancel Task |
| **Actors** | Elderly |
| **Trigger** | An Elderly Verbally addresses the System by saying "Pinto Manager, cancel <description of task>" |
| **Precondition** | The Elderly is audible by the Pinto Manager.  The task being canceled is either currently queued, or being actively fulfilled |
| **Basic Path** | 1) The Elderly says "Pinto Manager, cancel <description of task>"  2) Pinto Manager says "Ok. Done"  3) end |
| **Alternative Path A** | In step 1, the task requested to be canceled is executing, and the item has already been picked up  2A1) Pinto Manager says "That task is being fulfilled right now. If you cancel it, my Pinto will just leave your item on the ground. Do you still want to cancel it?"  2A2) If the user says no, then this use case aborts.  2A3) The Pinto leaves the item on the ground.  2A4) The pinto returns to its staging area and becomes idle.  2A5) end |
| **Postcondition** | The task is canceled. |
| **Other** | In general, the Pinto will try to return to the staging area if it cannot complete its task. |

****

The above is a diagram of the Elderly –My Friendly Pintos Cancel Request interaction

### Use Case: Help Desk

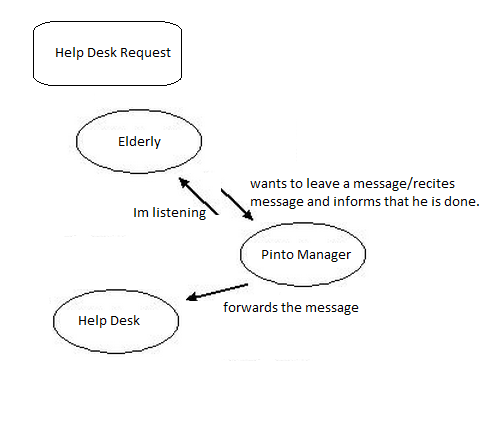
**Brief Description**

The Elderly needs assistance with the system, and can submit a request for help directly through the system. This automatically sends beneficial system diagnostic info to the support team, to facilitate usability through improved speed and reduced complexity of the support and resolution process.

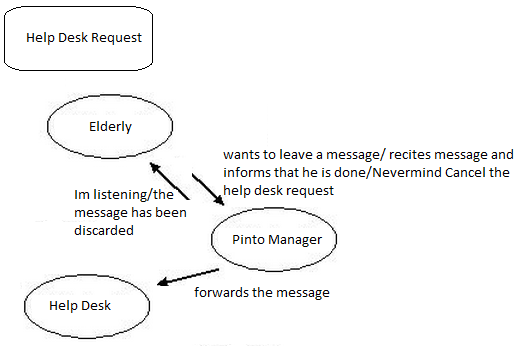
**Xref:** Section Feature: Help Desk

|  |  |
| --- | --- |
| **Use Case Name** | Help Desk Request |
| **Actors** | Elderly |
| **Trigger** | An Elderly Verbally addresses the System by saying "Pinto Manager, I want to send a message to the help desk". |
| **Precondition** | The Elderly is audible by the Pinto Manager. |
| **Basic Path** | 1) "Pinto Manager, I want to send a message to the help desk"  2) Pinto Manager says "Ok. I'm listening closely, and recording. Please go ahead. Just tell me when you're done."  3) The Elderly recites their message  4) The Elderly says "I'm done"  5) Pinto Manager sends the message to the help desk, along with recent system diagnostic information.  6) end |
| **Alternate Path A** | At any point, the Elderly says "Never mind. Cancel this help desk request".  A1) The Pinto Manager discards any recorded message that may exist  A2) end |
| **Postcondition** | If the process was completed, the help desk will have the request for support, and will call the Elderly sometime soon to assist them. |
| **Other** | An alternate trigger would be saying "I need help with the system". |

**Base Case**

****

**Variation A**

****

The above is a diagram for the Elderly –My Friendly Pintos Help Desk interaction

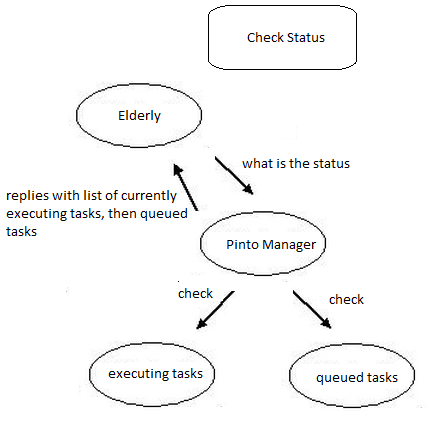
### Use Case: Check Status

**Brief Description**

The Elderly checks the status of the system, to gain info on queued and executing requests.

**Xref:** Section Feature: Check Status

|  |  |
| --- | --- |
| **Use Case Name** | Cancel Task |
| **Actors** | Elderly |
| **Trigger** | An Elderly Verbally addresses the System by saying "Pinto Manager, cancel <description of task>" |
| **Precondition** | The Elderly is audible by the Pinto Manager.  The task being canceled is either currently queued, or being actively fulfilled |
| **Basic Path** | 1) The Elderly says "Pinto Manager, cancel <description of task>"  2) Pinto Manager says "Ok. Done"  3) end |
| **Alternative Path A** | In step 1, the task requested to be canceled is executing, and the item has already been picked up  2A1) Pinto Manager says "That task is being fulfilled right now. If you cancel it, my Pinto will just leave your item on the ground. Do you still want to cancel it?"  2A2) If the user says no, then this use case aborts.  2A3) The Pinto leaves the item on the ground.  2A4) The pinto returns to its staging area and becomes idle.  2A5) end |
| **Postcondition** | The task is canceled. |
| **Other** | In general, the Pinto will try to return to the staging area if it cannot complete its task. |

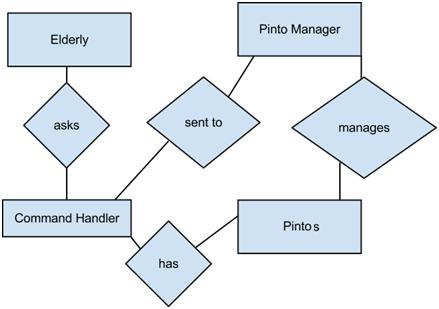
****

The above is a diagram for the Elderly –My Friendly Pintos Check Status interaction

## 3.3 Detailed Non-Functional Requirements

### 3.3.1 Logical Structure of the Data

The logical structure of the data to be stored in the internal Article Manager database is given below.



**Figure 4 - Logical Structure of the “My Friendly Pintos”**

The data description of each of these data entities is as follows:

**Elderly Data Entity**

|  |  |  |  |
| --- | --- | --- | --- |
| **Data Item** | **Type** | **Description** | **Comment** |
| Name | String | Name of the elderly | The manager knows the name of the elderly |
| Command | Pointer | Command Handler data entity |  |

**Command Handler Data Entity**

|  |  |  |  |
| --- | --- | --- | --- |
| **Data Item** | **Type** | **Description** | **Comment** |
| Listens | Pointer | Elderly entity | Gets the command either in text or speech form |
| Sends command | Pointer | Pinto Manager | Sends the command to the Pinto Manager so that it can command the robots to act |

**Pinto Manager Data Entity**

|  |  |  |  |
| --- | --- | --- | --- |
| Data Item | Type | Description | Comment |
| Name | String | Name of the Pinto Manager |  |
| Location | Integer | Location of a pinto | Manager knows all the locations of all the objects in the grid |
| Manages | Pointer | Manages the robots and assigns them tasks | Handles all the actions of all the robots simultaneously |
| Avoids collisions | Pointer | Avoids collision with other pintos and objects | Uses a special algorithm to figure out |
| Reports | Pointer | Elderly entity | Reports the Elderly if the command is finished |

**Robot Data Entity**

|  |  |  |  |
| --- | --- | --- | --- |
| Data Item | Type | Description | Comment |
| Location | Integer | Pinto knows it’s own location | Pinto can update the manager with its location |
| Priority ID | Integer | Each pinto has it’s own priority ID | The manager gives the pinto with the greatest number ID, the higher priority. |
| Listens | Pointer | Pinto Manager entity | Listens to the Pinto Manager for an action |
| Act | Pointer | Pinto Manager entity | Carries out a task given by the Pinto Manager |

## Business Rules

We at “Plz Send The Codes” understand that the “*My Friendly Pinto*” –A multi robot cluster, is primarily designed for Elderly Support and Care and is meant to provide a cost effective solution to achieve their needs.

Our business rules are therefore customer centric and are illustrated in the following table:-

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Status\ Behavior | Purchases 1 robot cluster | Purchases 5 robot cluster | Rents 1 robot cluster | Rents 5 robot cluster |
| Home User | Pays $1000 | Pays $3000 | Pays $100/month with a 2 year contract | Pays $300/month with a 2 year contract |
| Senior Home | Pays $3000 | Pays $9000 | Pays $300/month with a 2 year contract | Pays $900/month with a 2 year contract |
| Hospital | Pays $5000 | Pays $15000 | Pays $500/month with a 2 year contract | Pays $1500/month with a 2 year contract |

**These business rules are described underneath:**

**Purchase/Rent of a 1 robot cluster:-**

1. “**If**” customer “**is**” Home User “**and**” makes a purchase of 1 robot cluster, “**then**” he pays $1000 with lifetime of free support.
2. “**If**” customer “**is**” Home User “**and**” makes a rental of 1 robot cluster, “**then**” he pays $100/month with a 2 year contract with free support for that period. If he chooses to break the contract intermediately, “**then**” he pays a penalty of $10/month left in contract.
3. “**If**” customer “**is**” Senior Home “**and**” makes a purchase of 1 robot cluster, “**then**” he pays $3000 with lifetime of free support.
4. “**If**” customer “**is**” Senior Home “**and**” makes a rental of 1 robot cluster, “**then**” he pays $300/ month with a 2 year contract with free support for that period. If he chooses to break the contract intermediately, “**then**” he pays a penalty of $10/month left in contract.
5. “**If**” customer “**is**” Hospital “**and**” makes a purchase of 1 robot cluster, “**then**” he pays $5000 with lifetime of free support.
6. “**If**” customer “**is**” Hospital “**and**” makes a rental of 1 robot cluster, “**then**” he pays $500/ per month with a 2 year contract with free support for that period. If he chooses to break the contract intermediately, “**then**” he pays a penalty of $10/month left in contract.

**Purchase/Rent of a 5 robot cluster:-**

1. “**If**” customer “**is**” Home User “**and**” makes a purchase of 5 robot cluster, “**then**” he pays $3000 with lifetime of free support.
2. “**If**” customer “**is**” Home User “**and**” makes a rental of 5 robot cluster, “**then**” he pays $300/ month with a 2 year contract with free support for that period. If he chooses to break the contract intermediately, “**then**” he pays a penalty of $10/month left in contract.
3. “**If**” customer “**is**” Senior Home “**and**” makes a purchase of 5 robot cluster, “**then**” he pays $9000 with lifetime of free support.
4. “**If**” customer “**is**” Senior Home “**and**” makes a rental of 5 robot cluster, “**then**” he pays $900/month with a 2 year contract with free support for that period. If he chooses to break the contract intermediately, “**then**” he pays a penalty of $10/month left in contract.
5. “**If**” customer “**is**” Hospital “**and**” makes a purchase of 5 robot cluster, “**then**” he pays $15000 with lifetime of free support.
6. “**If**” customer “**is**” Hospital “**and**” makes a rental of 5 robot cluster, “**then**” he pays $1500/month with a 2 year contract with free support for that period. If he chooses to break the contract intermediately, “**then**” he pays a penalty of $10/month left in contract.

We do not “currently” provide/support for more than 5 robots per cluster for purchase or rent for any period of time.

# 5.0 System Design

**5.1 CRC Cards**

Given below is a brief description of the system classes, a list of their responsibilities and a list of classes each class collaborator with. All class names are given in bold.

|  |  |
| --- | --- |
| **PathFinder** | |
| **Responsibilities** | **Collaborators** |
| * Populates a map of the senior home setting given positions of obstacles. * Computes all paths between the pinto docking station and each item. * Finds shortest path between the pinto docking station and “a” item. * Dynamically updates regions of avoidance for pintos out fetching items. | * **EnvironmentMap** * **Vertex** * **Edge** * **Pinto** |

|  |  |
| --- | --- |
| **Vertex** | |
| **Responsibilities** | **Collaborators** |
| * Keeps track of adjacencies. * Keeps track of minimum distance to reach it from pinto docking station. * Keeps track of its immediate previous vertex to track the path. | * **Edge** |

|  |  |
| --- | --- |
| **Edge** | |
| **Responsibilities** | **Collaborators** |
| * Keeps track of target vertex * Keeps track of weight to help pick a “next step” vertex and also to avoid collisions. | * **Vertex** |

|  |  |
| --- | --- |
| **EnvironmentMap** | |
| **Responsibilities** | **Collaborators** |
| * Knows the static layout of the house including obstacles and items. * Knows co-ordinates of pinto docking station. * Records co-ordinates of elderly. * Records co-ordinates of items. * - * Provides a walkability matrix for pintos. | * **MovableObject** * **Item** * **PathFinder** * **MapFeatures** |

|  |  |
| --- | --- |
| **MovableObject** | **Subclass: Pinto** |
| **Responsibilities** | **Collaborators** |
| * Knows its name and type * Keeps track of its co-ordinates. | * **EnvironmentMap** |

|  |  |
| --- | --- |
| **Item SuperClass: Movable Object** | |
| **Responsibilities** | **Collaborators** |
| * Knows its name * Knows its co-ordinates. | * **Movable Object** |

|  |  |
| --- | --- |
| **PintoManager** | |
| **Responsibilities** | **Collaborators** |
| * Processes Commands. * Queries Pathfinder for a shortest collision free path to the item. * Dispatches pintos to getItems on shortest collision free paths. * Keeps track of pintos, their availability and their task status. * Has a polling scheme to periodically watch for queued getItem tasks and spawns them as pintos become idle. * Relays messages to userInterface for delivery to the elderly via callbacks through commands. | * **Pinto** * **PathFinder** * **Command** * **EnvironmentMap** |
| **Pinto** | **Superclass: MoveableObject** | |
| **Responsibilities** | **Collaborators** | |
| * Executes Commands. * Accepts command from PintoManager. * Creates pathfinders to to find shortest paths from docking station to the item, item back to the elderly, and from the elderly back to the docking station. * Updates PathFinder its co-ordinates. * Delivers items to external actor. * Keeps track of its status IDLE| BUSY | * **Item** * **PintoManager** * **Command** * **PathFinder** * **EnvironmentMap** | |

|  |  |
| --- | --- |
| **UserInterface** | |
| **Responsibilities** | **Collaborators** |
| * Reads elderly input in text or voice. * Relays messages from the system back to the elderly. * . * Provides callbacks for pintos to communicate back to the elderly. * Dispatches commands * Sends user input to CommandParser to validate tasks to perform. * Displays the houseMap. * Displays the user guide for the system. | * **CommandParser** * **PintoManager** * **EnvironmentMap** |

|  |  |
| --- | --- |
| **CommandParser** | |
| **Responsibilities** | **Collaborators** |
| * Parses input. * Generates commands for the system to perform. * Validates commands. | * **Command** * **UserInterface** |
| **Command** | |
| **Responsibilities** | **Collaborators** |
| * Knows item name. * Knows command type [add\_item|get\_item|cancel\_item|help\_desk|get\_item\_status| show\_map] * Provides the callback interface for pintoManager to interact with the userInterface. | * **CommandParser** |

|  |  |
| --- | --- |
| **MapFeatures** | |
| **Responsibilities** | **Collaborators** |
| * Reads an colored image file which depics the housemap with static obstacles and wall boundaries. * Translates the pixilated image into the housemap with blocked tiles marked obstacles and free spaces as walkable locations. | * **Point** * **BufferedImage** * **EnvironmentMap** |

## 5.2 UML Diagrams

To visualize these classes and their different relationships between their collaborations, a UML diagram describing their interaction is shown below along with their respective multiplicities.

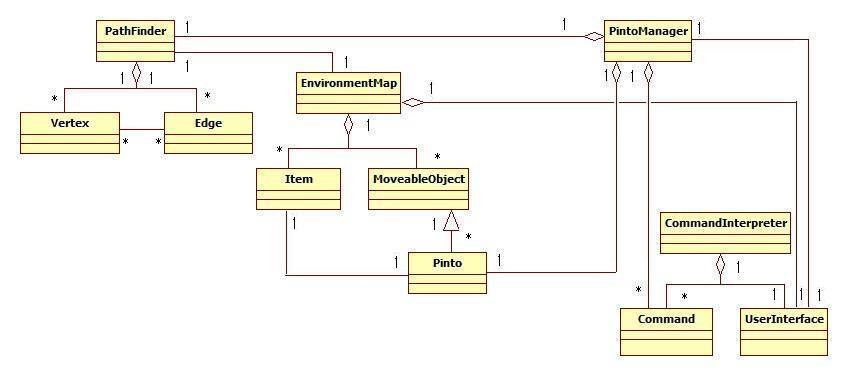
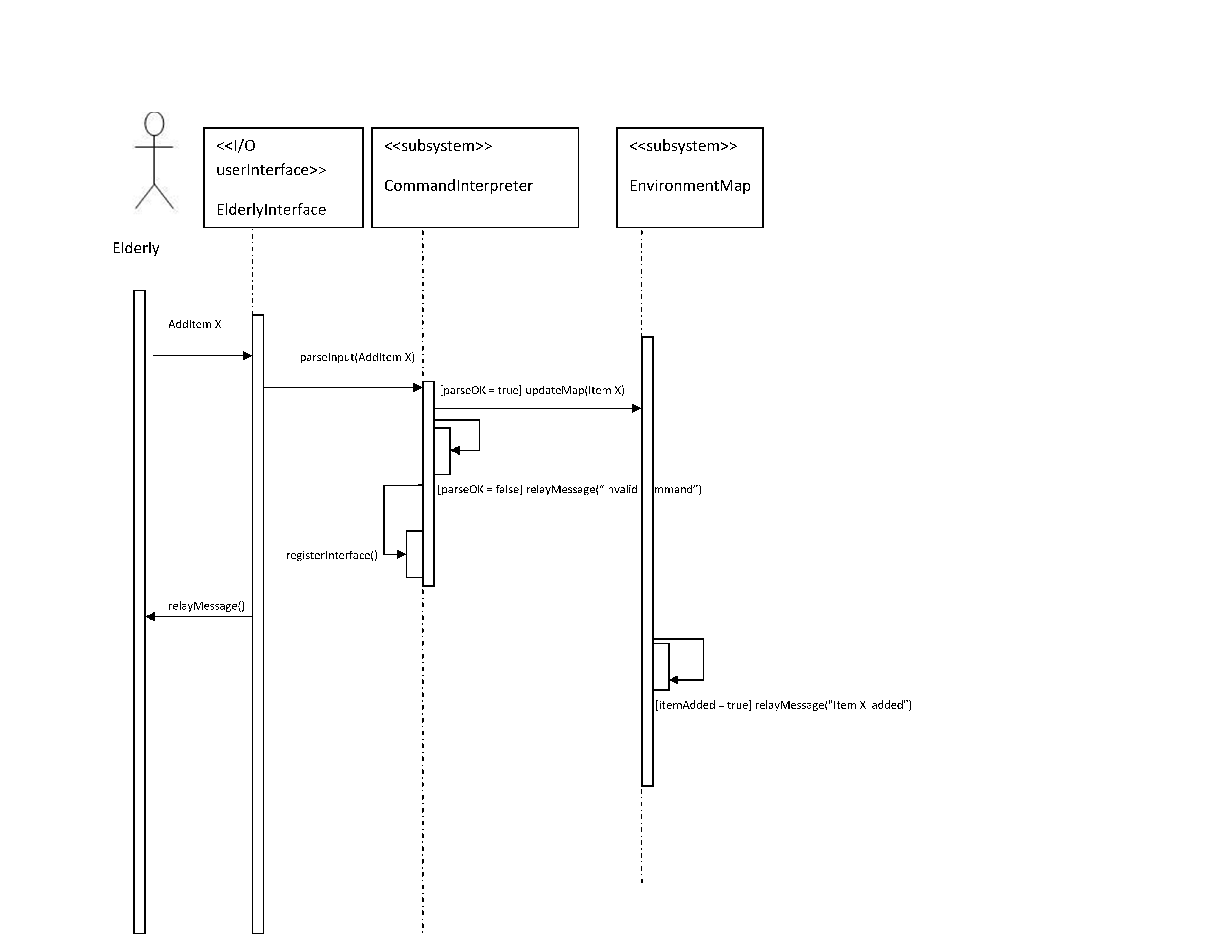


Figure 5 - System Design - UML

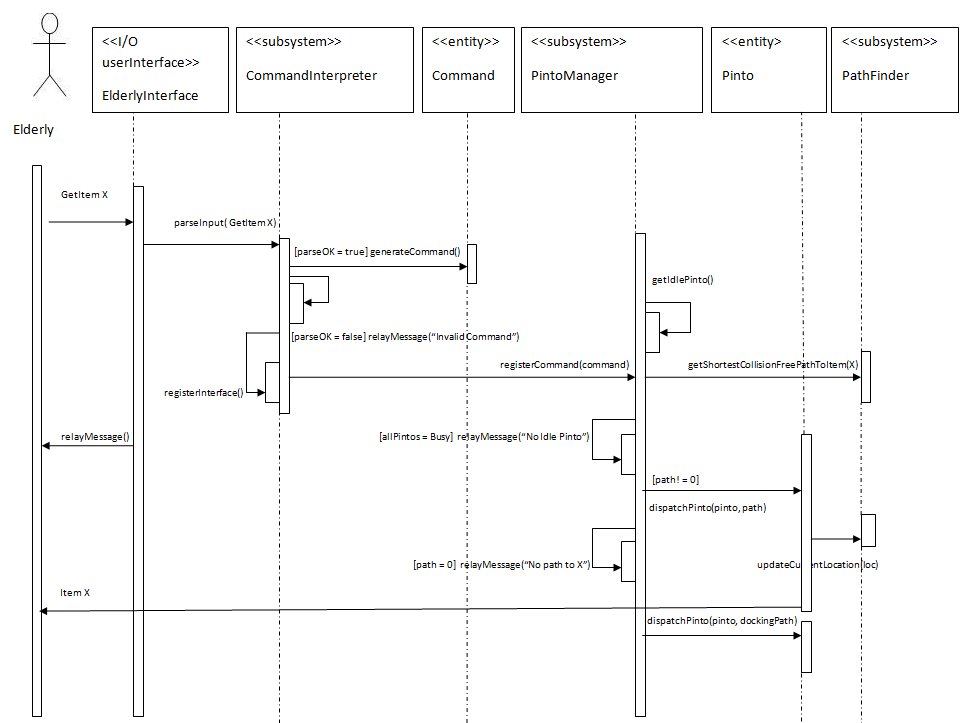
**5.3 Sequence Diagram**

**5.3.1 Use Case: Add Item**



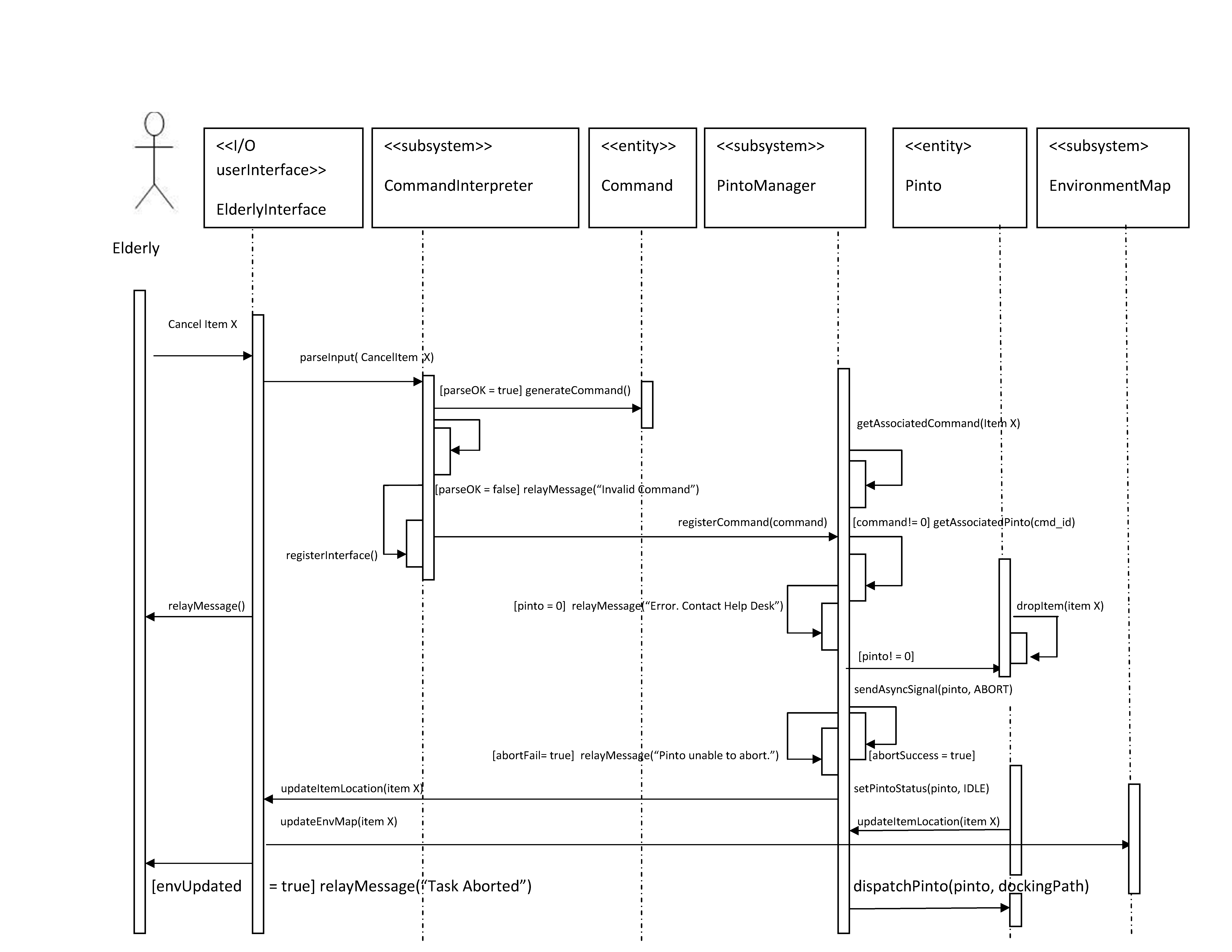
**Figure 6: Sequence Diagram for Use Case: Add Item**

**5.3.1 Use Case: Get Item**

****

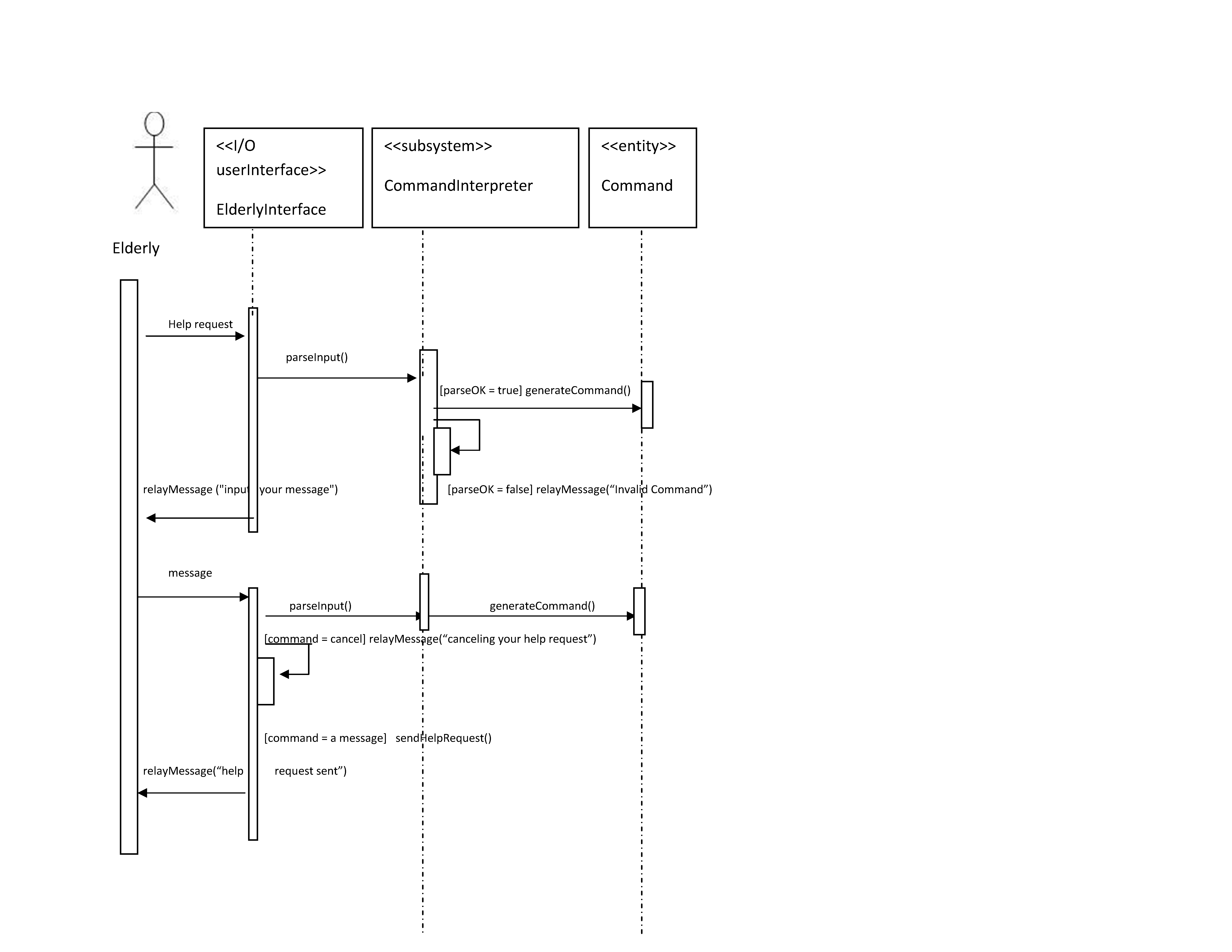
**Figure 7: Sequence Diagram for Use Case: Get Item**

**5.3.3 Use Case: Cancel Request**

****

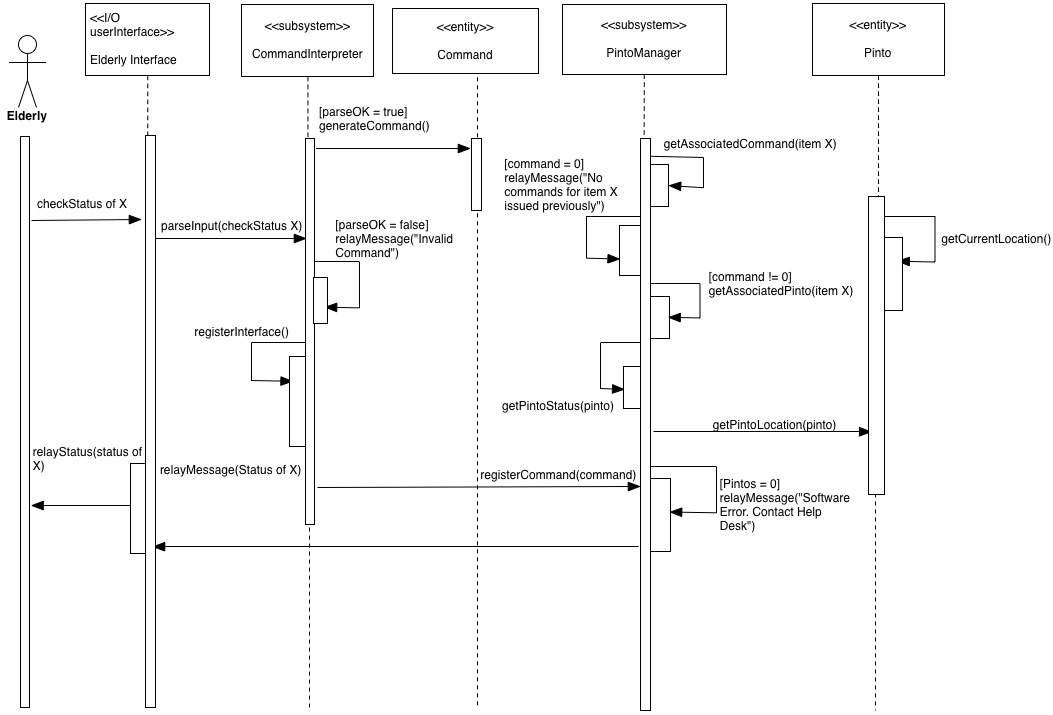
**Figure 8: Sequence Diagram for Use Case: Cancel Request**

**5.3.4 Use Case: Help Desk**

****

**Figure 9: Sequence Diagram for Use Case: Help Desk**

**5.3.5 Use Case: Check Status**

****

**Figure 10: Sequence Diagram for Use Case: Check Status**

## 6.0 Glossary

|  |  |
| --- | --- |
| **Term** | **Definition** |
| Active Article | The document that is tracked by the system; it is a narrative that is planned to be posted to the public website. |
| Actors | Someone or something outside the system that either acts on the system – a primary actor – or is acted on by the system – a secondary actor. For example, An *Elderly (primary actor) submits a job to retrieve an Item.* [*www.Plz\_Send\_The\_Codes.com*](http://www.Plz_Send_The_Codes.com) *provides software updates and customer support, and is the (secondary actor).* |
| Software Functional Specifications | A document that completely describes all of the functions of a proposed system and the constraints under which it must operate. For example, this document. |
| Stakeholder | Any person with an interest in the project who is not a developer. |
| System | The Multi Service-Robot cluster with which Actors interact. |

7.0 Attachment A – Approval Signatures

* \_Abhishek Bais\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_07/08/2012\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Signed: Date:

*< Software Developer >*

* \_Chris Rehfeld \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_07/08/2012\_\_\_\_\_\_\_\_\_\_\_\_\_

Signed: Date:

*< Software Developer>*

* \_Charles Le\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_07/08/2012\_\_\_\_\_\_\_\_\_\_\_\_\_

Signed: Date:

*< Software Developer >*

* \_Akshay Hegde\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_07/08/2012\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Signed: Date:

*< Software Developer >*