2015 enPiT BizApp

Global Service Application Development Course

Robot Service Application Project

BuddyBot

System Requirements Specification

Revision 1.1

October 29th, 2014

Advanced Institute of Industrial Technology

Universiti Brunei Darussalam

Unitec Institute of Technology

Japan

Advanced Institute of Industrial Technology

Kaoru Nagase

Tomoki Kawayama

Takushoku University

Kentaro Kurakata

enPiT

Yuta Yaguchi

Kei Ogasawara

Brunei Darussalam

Universiti Brunei Darussalam

Matthew Jerickson Nazareno Vejerano

Ak Ahmad Al-Khairi bin Pg Haji Tejudin

Abdul Aziz bin Hj Abdul Kadir

Irzy Tuah

Dk Nur Afiqah Jacwati Puteri Pg Md Caesar Perkasa Putera

Nur Farwezza Nadhirah bte Mohd Kamransah

Md Shohidul Islam

Mohamad Fauzi bin Zaini

New Zealand

Unitec Institute of Technology

Adam Deery

Josh Prow

Dila Beisembayeva

Wafaa Che Rose

**Revision History**

|  |  |  |  |
| --- | --- | --- | --- |
| Date | Description | Author | Comments |
| 25th August 2014 | Rev. 1.0 | Yuji Kaneko |  |
| 29th October 2014 | Rev. 1.1 | Yuji Kaneko | Fixed Yuta’s name |
|  |  |  |  |

Contents

[1 Introduction 1](#_Toc396721585)

[1.1. Purpose 1](#_Toc396721586)

[1.2. Scope 1](#_Toc396721587)

[1.3. Definitions, Acronyms, and Abbreviations 2](#_Toc396721588)

[2. General Description 2](#_Toc396721589)

[2.1. Product Perspective 2](#_Toc396721590)

[2.2. Product Functions 3](#_Toc396721591)

[2.3. User Characteristics 4](#_Toc396721592)

[2.4. General Constraints 4](#_Toc396721593)

[2.4.1. EV3 Limitations 4](#_Toc396721594)

[2.4.2. RSNP Limitation 6](#_Toc396721595)

[2.4.3. Human Limitations 6](#_Toc396721596)

[2.4.4. Other Limitations 7](#_Toc396721597)

[2.5. Assumptions and Dependencies 7](#_Toc396721598)

[3. Specific Requirements 7](#_Toc396721599)

[3.1. External Interface Requirements 7](#_Toc396721600)

[3.1.1. User Interfaces 7](#_Toc396721601)

[3.1.2. Hardware Interfaces 8](#_Toc396721602)

[3.1.3. Software Interfaces 9](#_Toc396721603)

[3.1.4. Communications Interfaces 9](#_Toc396721604)

[3.2. Functional Requirements 9](#_Toc396721605)

[3.2.1. Register the missions 9](#_Toc396721606)

[3.2.2. Complete the missions 12](#_Toc396721607)

[3.2.3. Point System 13](#_Toc396721608)

[3.2.4. Additional robot features 15](#_Toc396721609)

[3.3. Use Case 15](#_Toc396721610)

[3.3.1. Send mission to server 15](#_Toc396721611)

[3.3.2. Send mission to EV3 16](#_Toc396721612)

[3.3.3. Interaction with child 17](#_Toc396721613)

[3.3.4. Send log data to server 18](#_Toc396721614)

[3.4. Non-Functional Requirements 18](#_Toc396721615)

[3.5. Design Constraints 20](#_Toc396721616)

[4. Change Management Process 20](#_Toc396721617)

# 1 Introduction

## Purpose

The purpose of this document is to state the detailed requirements for the BuddyBot system, which cares for and educates children with chronic illnesses. It will explain the reason this system is required in New Zealand and will show mandatory and optional system requirements, use cases, system constraints and human interfaces. This document is intended to be a reference to develop the system for ourselves, the Global PBL development team.

## Scope

BuddyBot aims to develop a robot that can assist children with chronic illnesses. By providing entertainment and management techniques, BuddyBot can help relieve pressure on parents by asking the child to follow certain instructions. Vital 　information can be made available for people such as parents, medical professionals or welfare provider although BuddyBot will not take the place of a parent/caretaker.

The final product will have the ability to encourage certain activities and follow a specific child around. The software will have monitoring and scheduling built into it to observe the child’s health and wellbeing. This will be done by developing a mission themed application to encourage the child to follow instructions such as taking medication, bed times and other useful features.

The software will be used in this robot for testing although it could be adapted to help every child or elderly people. The overall goal is to provide a user friendly application that can interact with children of various ages and abilities while having a low fault rate and high entertainment value. The ability for a non-technical person to be able to add extra functions to provide new games and challenges to the device is important.

BuddyBot is to be designed in a child friendly way with appealing and useful functions while being robust enough for everyday use.

## Definitions, Acronyms, and Abbreviations

***Caretaker***: A human who takes care of the child, including parents, doctors, teachers and caretakers.

***EV3***: LEGO® MINDSTORMS® Education EV3

# General Description

## Product Perspective

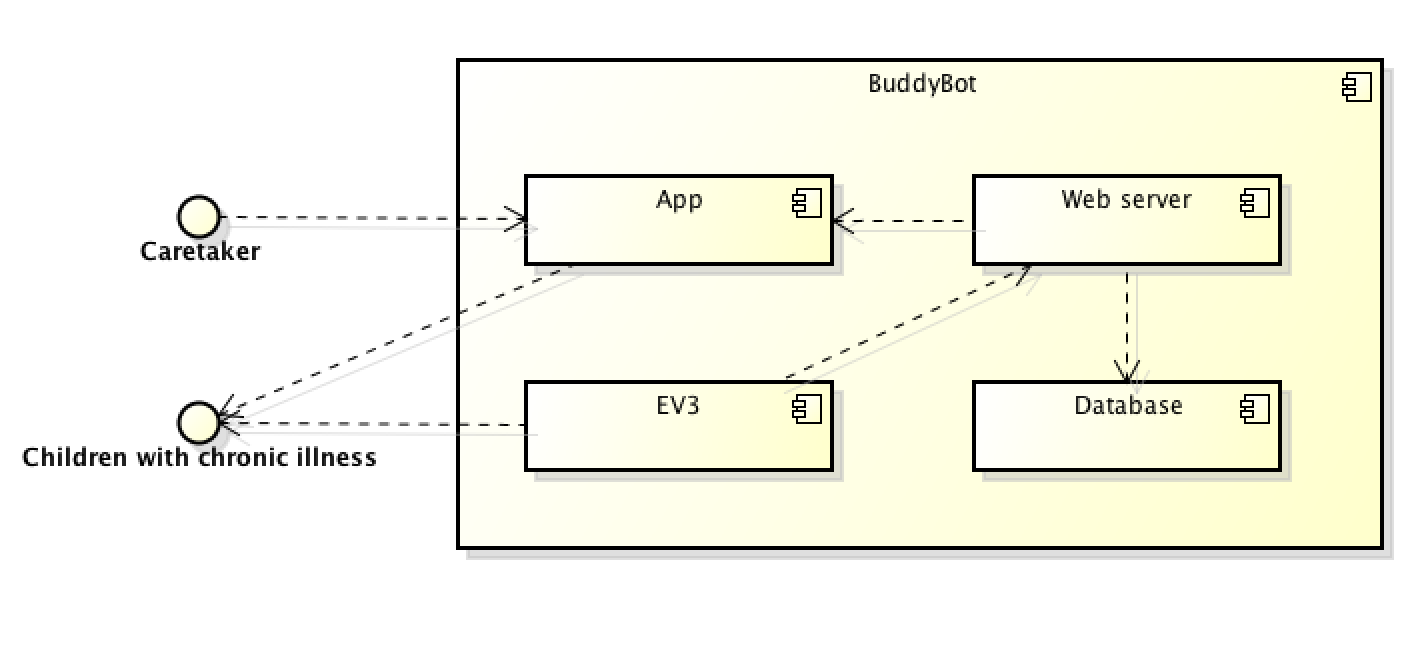


Figure 1 - Component diagram

The main components of BuddyBot are the application, web server, EV3 and a Database. The caretaker and children with chronic illnesses are outside of the system. The caretaker cannot register tasks (missions) without access to the web server. The application then sends the tasks to a web server where the EV3 can retrieve tasks from web server. If a child completes a mission, the application sends confirmation to the server.

## Product Functions

Using the BuddyBot, a child who has a chronic illness will be able to learn and understand their illness. The child will build their knowledge of their illness through the missions that are sent by the caretaker.

The caretakers use an application to send missions to the server, and then the server sends these missions to the EV3 which interacts with the child according to the type of mission it is. Once the user completes the missions, the BuddyBot sends the data to the server. Caretakers can then close the missions through the application. By completing the missions, the users get visible points which can be exchanged with items for additional parts or functions for the BuddyBot.

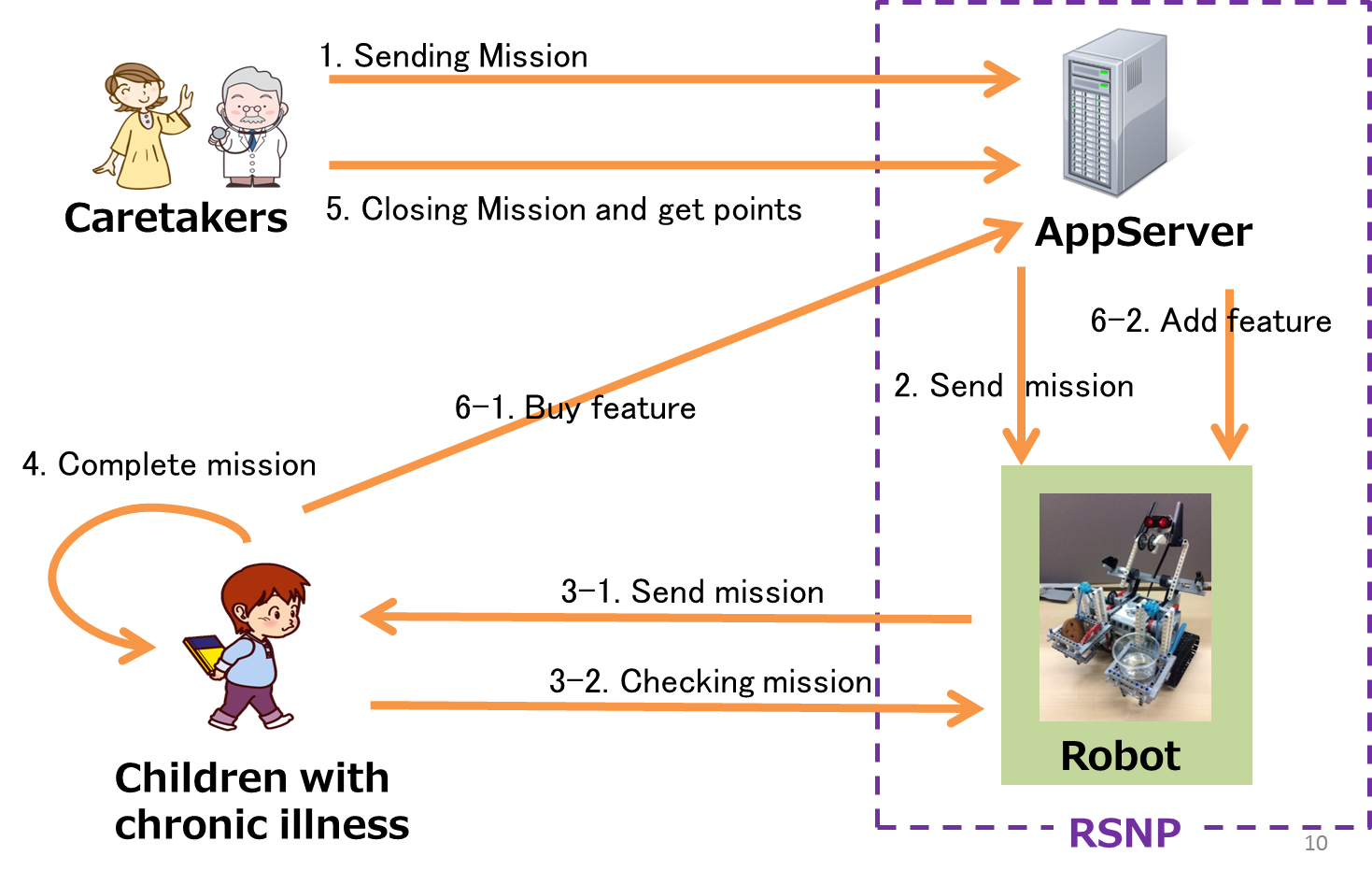


Figure 2 - Sequence Diagram

## User Characteristics

Users of the robot are classified into 2 categories: Children and Caretakers.

a) Children are the main users of this robot system. They are primarily school-age children with a chronic illness. Many do not know how they can get over their difficult situation or require assistance in everyday life. They have to take medicine every day by themselves and go to doctor, but often refuse medical treatment.

b) Caretakers are guardians of the child, and they don't always have technical knowledge of medicine, but have enough knowledge to care for their children. They recognise the importance of daily medical care, but can often have issues convincing the child in their care the importance of the treatment.

c) Doctors are experts in medicine. They have technical knowledge of medicine and healthcare, but they cannot care for every individual child every day. They need help to make the children happy at the hospital and to teach the children the knowledge required for daily care.

## General Constraints

There are many constraints that limit the scope and build of the product, these are categorized these into 4 major categories:

1. EV3 Limitations
2. RSNP Limitations
3. Human Limitations
4. Other Limitations

### EV3 Limitations

The EV3 is the basic structure upon which the BuddyBot is being built, it is a Lego product and runs its own propriety Linux-based OS. Each unit has a collection of sensors, ranging from ultrasonic sensor to press buttons, and 3 motors. Being a Lego product the “brick” or processing unit is designed to be attached to a hull constructed of Lego Technic parts, however using custom parts one can use the unit along-side more powerful and stronger materials.

1. **Sensors**

The EV3 comes with a few default sensors, depending on the version purchased. These include:

* Touch sensors
* Colour sensors
* Gyroscopic sensors
* Ultrasonic sensor
* Infrared sensor (home edition only)

Many official and non-official parts can also be purchased.

The sensors on the EV3 unit are basic and provide very limited information. Therefor the expansion of the unit to work alongside a more advanced unit such as a tablet or RaspberryPi is common.

Also of note is that the EV3 is fully compatible with the motors and sensors from the previous MindStorms robot version; the NXT, and with a converter can also interact at a basic level with the RCX (predecessor to the NXT) pieces.

The EV3 has 4 sensor ports which connect to the sensors using a modified RJ-12 port (which is incompatible to standard RJ cables and must be used with NXT or EV3 cables only).

1. **Motors**

The EV3 has 4 motor ports which like the sensors are RJ-12 type cables, but the basic kit for the EV3 comes with only 3 motors default. There are 2 kinds of (official) EV3 motors – the medium motor and the large motor, the maximum output of these motors isn’t stated but from experience with the NXT motors it is safe to assume that the motors are only designed to move small / light-weight units.

The motors also can measure rotation in degrees. The sensor by default works in 360 increments but can be forced to report up to 720, thus meaning the unit can measure 0.5 degree rotations.

The EV3 is also compatible with the prior MindStorms motors (the RCX motors need a converter) however these motors provide less torque than the EV3’s own motors.

1. **Processor**

The EV3 runs off a small 300MHz processor, with 64MB of RAM. This means it can easy process basic requests but has very little ability when it comes to processing multiple functions and will often fall behind in large calculations.

Unfortunately there is very little documentation on the efficiency of the processor so this may have to be tested with as the robot is being developed.

1. **Communication**

The EV3 can communicate over a High Speed USB2 connection, Bluetooth, Wi-Fi (via an external Wi-Fi dongle), and IR if the correct sensor is attached.

1. **Battery**

Switching on EV3 and running the sensors all the time can easily deplete the battery.

### RSNP Limitation

RSNP is the service into which the BuddyBot is going to be plugging into. The exacting amount interaction with the device is as of yet unknown, and the API itself needs researching as to the limitations and capabilities of the services it provides.

More information can be found at this website; [www.robotservices.org/wiki/en](http://www.robotservices.org/wiki/en). However much of the information is incomplete and can only be found on the Japanese version [www.robotservices.org/wiki/jp](http://www.robotservices.org/wiki/jp)

### Human Limitations

The BuddyBot is to be used by young children as well as be used as a utility for doctors and caretakers. This means the administration of the device must be easy enough for it to be used by people with very little to no technical know-how but while also being complex and strong enough to be entertaining to a young child.

Via the administration the caretaker will need to be able to program new games and challenges into the device with which to entertain the child. This needs to be an easy to operate system as the caretaker may have absolutely no experience in programming or computing as a whole. The other service of monitoring the child also needs to be user friendly as this information should be available from anywhere and to people who again may have no experience with computers and robotic devices.

The design of the robot itself is limited by the requirement to stand up to daily usage by a young child. The robot will need to be stable enough to survive being knocked down and dropped while also being fast enough and having enough functions to keep up and entertain the child. The design of the unit will need to inspire the child to use the robot as a play thing and the challenges and missions need to be able to retain the interest of the user.

### Other Limitations

Due to the purpose behind the project; to test the RSNP service working alongside the EV3, it is important that the RSNP interact with the EV3 directly. This means that although the EV3 has compatibility with tablets and other devices better suited to the processing of the required functions these devices need to be handled via the EV3.

## Assumptions and Dependencies

1. EV3 is capable of network communication.
2. Caretaker can operate smartphone and PC.
3. The application is provided by English.

# Specific Requirements

## External Interface Requirements

### User Interfaces

1. **Application**

The application is provided as a web application in English. The application has interfaces as following:

* Registering and closing missions
  + Caretakers register missions from this application. They can write mission texts and choose difficulty to register the missions.
  + Caretakers can close a mission when the child completes the mission. This application shows the mission, which has been registered, and it can be closed from here.
* “Buying” new feature and specification documents of the robot
  + The child can “buy” new feature and specification documents of the robot by spending points from this application. It shows the points that the child has and also shows a list of features and specification documents.
* Seeing specification documents of the robot
  + The application shows a list of specification documents that the child has already bought. The child can see them whenever they want.

1. **EV3**

* Asking missions
  + The child can see the mission which the robot has received through the robot’s monitor.
* Commanding actions
  + The child can play the feature which he/she has bought by sending specific command to the robot. The command is different by the feature. Ex) Touching the touch sensor.

### Hardware Interfaces

1. **Application**

Caretaker/child uses the application via web browser. Display, keyboard and mouse (PC), touch screens and some physical buttons (Smartphone) are controlled by the Operating System installed in the device. And the application can use these hardware by the libraries which are provided by the OS.

1. **EV3**

The EV3 has 4 input ports which can be used by some sensors, and 4 output ports which can be used by motors. The software can receive signals from the sensors, and can send commands to the motors by EV3 Linux based OS.

### Software Interfaces

1. **Operating System**

Required Operating Systems for this project:

* + Operating system includes web browser to provide the application
  + EV3 Linux based OS to provide robot interface
  + Linux OS to provide a server which contains the database

1. **Database**

The server has a database to store information of users, robots and points. It will have some implement of a Relational Database System such as MySQL or PostgreSQL.

### Communications Interfaces

1. **The application and the server**

The application and the server communicate by HTTP or HTTPS protocol via the network by cellular communications or Wi-Fi from the internet access point.

1. **EV3**

The EV3 and the server communicate by RSNP protocol via the network by Wi-Fi from the internet access point.

## Functional Requirements

### Register the missions

1. **Introduction**

Caretakers register missions.

The content of a mission helps the children to learn about their illness.

For example, missions are:

* Go to the doctor
* Take medicine regularly
* Do homework
* Pass an examination about chronic illnesses
* Help their mother

1. **Inputs**

Missions include following items:

* Description

Explaining what children have to do.

This content is text and is sent to EV3, and children do as it explains.

* Experience points

The points gained when children complete their mission.

Children can get points based on the registered content.

1. **Processing**

a. User registers mission’s contents by web browser application of each device.

b. User presses a register button.

c. Web application server store mission in a database.

d. Web application server sends above mission to RSNP server.

e. RSNP server sends mission to EV3.

d. Web application servers send a result about registration to user's devices.

1. **Outputs**
2. Server’s Database
3. EV3
4. Error Handling

a. Caretaker cannot register mission by network trouble.

b. Web application server can't send a mission to EV3.

### Complete the missions

1. Introduction

Children complete mission by web browser application with their caretaker.

So children notify orally to their caretaker.

1. Inputs

Caretaker press finish button on web application.

1. Processing
2. Children notify orally their caretaker that they completed mission.
3. Caretaker login web application.
4. Caretaker press finish button appropriate mission.
5. Web application server register data that a mission was completed with database. (Children get point according to difficulty)
6. Web application server sends above data to RSNP server.
7. RSNP server sends the command to delete completed mission to EV3.
8. EV3 execute to delete the completed mission.
9. Outputs

a. Server’s Database

b. EV3

1. Error Handling

a. Caretaker cannot register completed mission by network trouble.

b. Web application server can't send command to EV3.

c. EV3 can’t delete completed mission.

### Using robot features

1. Introduction

Children can “buy” new robot features for Lego EV3 using the points the child has accumulated.

Robot feature include robot blueprint, the motion of robot.

If Children have many robot motions, children can change robot feature by web application.

1. Inputs
2. Buy new robot feature

Children press buy button on web application.

1. Change robot feature

Children press change button on web application.

1. Read robot’s blueprint

Children press view button on web application.

1. Processing
2. Buy new robot feature

a-1. Children press buy button appropriate feature.

a-2. Bought feature can be to select press view mode button or change mode button.

1. Change robot feature

b-1. Children press change button appropriate feature.

b-2. Web application server send data to RSNP server.

b-3. RSNP server sends the command to change feature to EV3.

b-4. EV3 execute to change the feature.

1. Read robot’s blueprint

c-1. Children press view button appropriate feature.

c-2. Web application server show appropriate robot blueprint on web browser.

1. Outputs

a. Server’s Database

b. EV3

1. Error Handling
2. Web application server can’t send data of feature to EV3.
3. Web application server can’t show robot blueprint.

### Point System

1. Introduction

Children can buy new robot feature by point.

Caretaker defines difficulty of mission, when they register mission.

1. Inputs

Point system has 3 ways of inputs.

1. Caretaker defines difficulty of mission.
2. Children get point according to the difficulty.
3. Children use point for to buy a new robot feature

Following sentence describes each way.

1. Processing
2. Caretaker defines difficulty of mission

a-1. refer to 3.2.1-c) Processing

1. Children get point according to the difficulty.

b-1. refer to 3.2.2-c) Processing

1. Children use point for to buy a new robot feature

c-1. refer to 3.2.3-c) Processing

1. Outputs

a. Server’s Database

b. EV3

1. Error Handling

Point system used by several situations.

So point system’s error handling depends on situations.

We describe about error handling at following section:

a. 3.2.1-e) Error Handling

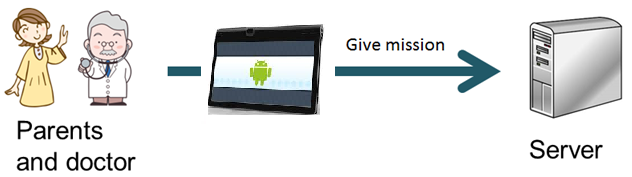
b. 3.2.2-e) Error Handling

c. 3.2.3-e) Error Handling

## Use Case

### Register mission

1. Diagram



Caretaker

Figure 5 – Send mission to server

1. Brief Description

Caretakers register a mission with the web application server.

1. Processing
2. Caretakers register a mission about health or the child’s illness with web application server.
3. Caretaker uses web browser for register mission with web application server.

### Send mission to EV3

1. Diagram

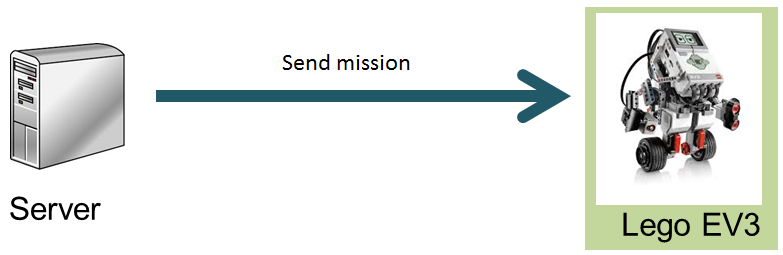
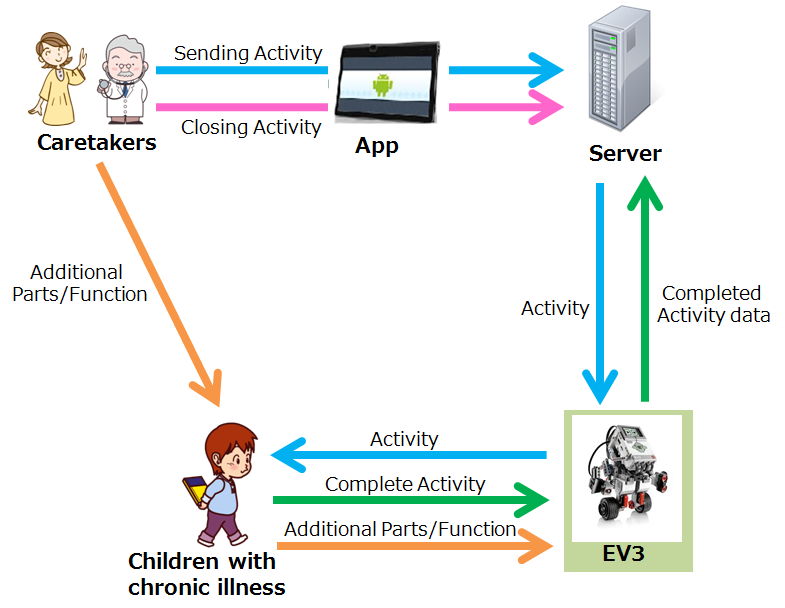


Figure 6 – Send mission to EV3

1. Brief Description

RSNP Server sends the registered mission to the EV3.

1. Initial Step-By-Step Description
2. Server relays the mission to EV3.
3. Mission is sent from server to EV3 using RSNP connection.

### Interaction with child

1. Diagram

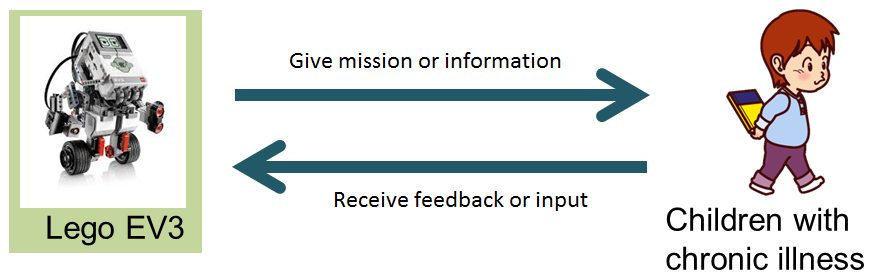
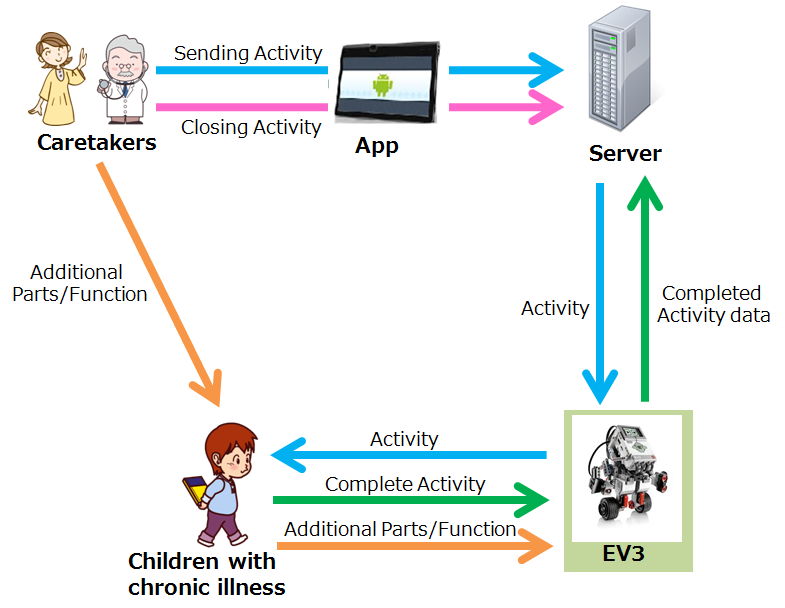


Figure 7 – Interaction with child

1. Brief Description

EV3 gives the mission to the children. When mission complete, mission is delete from EV3.

1. Initial Step-By-Step Description
2. EV3 notifies the child about the incoming mission.
3. When mission complete by web application server, mission is delete from EV3.

## Non-Functional Requirements

The main functions of this robot are to entertain a young child whilst monitoring it to ensure the health and safety of the child. The BuddyBot needs to be built to be child safe and easy to use by a non-technical user and the non-functional requirements need to reflect the needs of both user groups.

* + 1. Child End Requirements.

For the device to be child safe it must be easy to play and interact with, whilst still being durable and robust.

* + 1. Fun

For the BuddyBot to fulfill its purpose, it must hold the attention of the young child and keep them entertained, therefore this is a primary requirement

* + 1. Mobile

The BuddyBot needs to be mobile, both self-propelled and manually moved. It should be aware of its surroundings and be able to change its movements and behavior to suit

* + 1. Durable

Any item used by a young person has the high potential for getting damaged by dropping or incorrect usage. The BuddyBot should therefore be spill proof, drop proof, and be able to stand up to a moderate level of physical stress.

* + 1. Robust

The BuddyBot is designed to be a lasting companion to children, and therefore must be able to last for extended periods of time without need for repair.

* + 1. Caretaker

The other main user type is the caretaker who needs to setup and interact with the information system in the robot.

* + 1. Easy to use

These users may have no or very little skill in computing or working with technology, therefore the system must be easy to use and understand.

* + 1. Correct

As with any information reporting device the information given must be correct

* + 1. Customizable

The BuddyBot will have to work in many environments and with children with different illness. This means the service it provides must be tailored to fit the individual child. This customization will often take form in the type of challenges and games utilized by the system and the device will need an interface from which new games can be uploaded and even created.

## Design Constraints

* Internet connection
  + The application and the EV3 cannot work without an internet connection.
* Limited mission time
  + Only one mission can be carried out at a time on the EV3, which could cause the schedule to overload crash or miss important missions.
* Difficulty in creating mission/ game
  + The application to develop the missions will have to be easy to use and have a simple interface. The design of this will be incredibly important in the effectiveness of the robot.