# System Specification

| | |
|---|---|
| Project Name | Project NAO |
| Document Owner | Streibl Viktoria, Melanie Mühleder, Sabina Brantner |
| Created on | |
| Last modified | 30/01/2016 |
| State | X in Work<br>vorgelegt<br>Released |
| Dokumentablage | SystemSpecification.doc |

# Content

# 1 Initial Situation and Goal

## 1.1 Initial Situation

This project is about working with NAO´s, small humanoid robots. We planned to develop an app which allows easier handling with those. That means the user is able to connect with the NAO via the network and control it with a control button, or prefabricated actions like ‘sit down’ or ‘stand up’.

### 1.1.1 Application Domain

Our project solves these problems:
NAO´s could only be used if someone has programming skills and knows how those robots work.
Another problem is if somebody wants to show what a NAO is able to do, for example in a meeting, it costs a lot of time to switch on a computer.

For those problems we develop an app for smartphones:
The app helps users to control their robot and work with them. That works like this:
if the smartphone is in the same network as the NAO, the app is able to connect with the robot. So the user could choose some basic actions like ‘stand up’, the robot executes the action and stands up. Furthermore if the practitioner needs some other actions, he is able to download those from the data server.

### 1.1.2 Glossary

| | |
|---|---|
| NAO | Humanoid robot |
| actions | Things a NAO could do. For example: ‘stand up’ or ‘sit down’ |
| data server | A server which provides actions |
| network | System of several computers/device which communicate together |

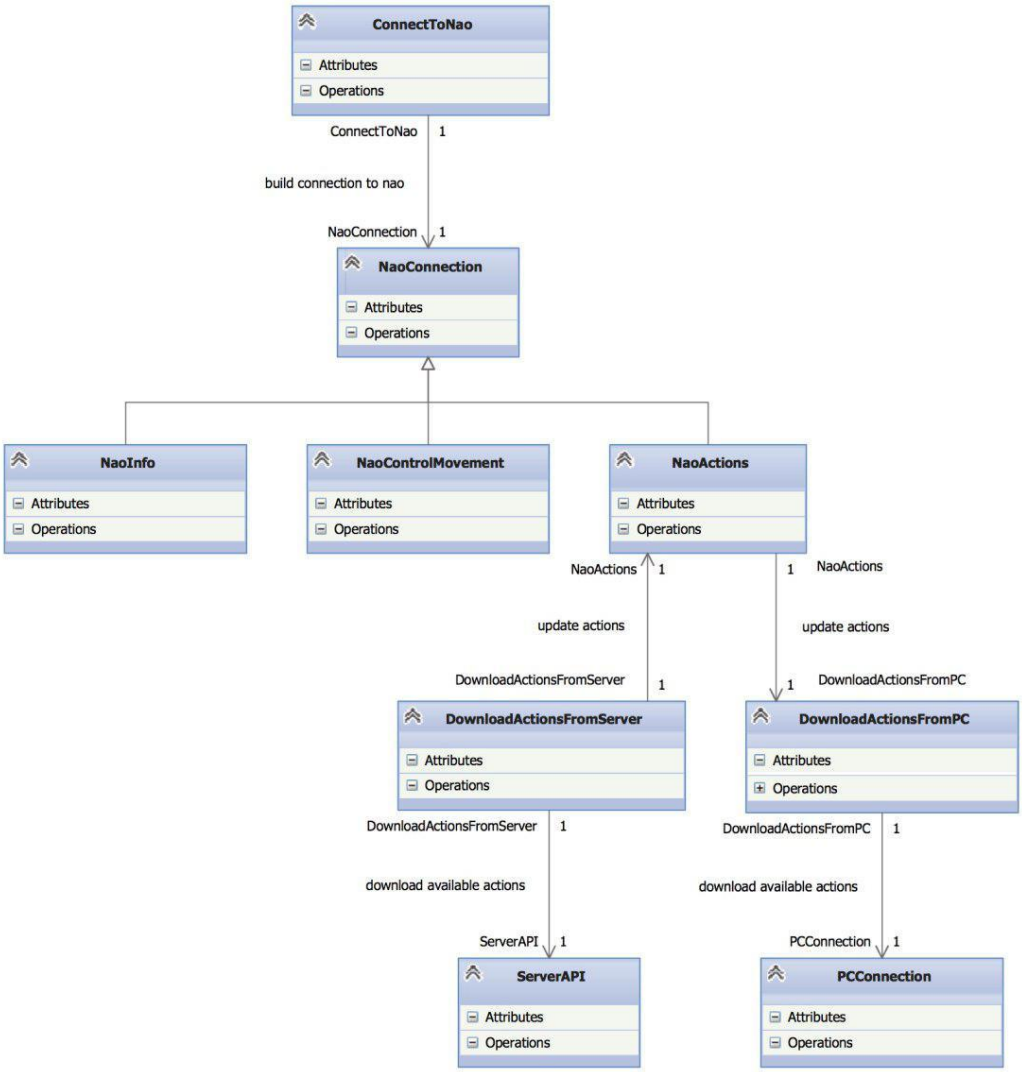
### 1.1.3 Model of the Application Domain

First of all, the app has to be connected to the NAO. Then the app has 3 different abilities.

The status which is the battery status, the temperature, the signal strength and the appearance, of the NAO can be checked.

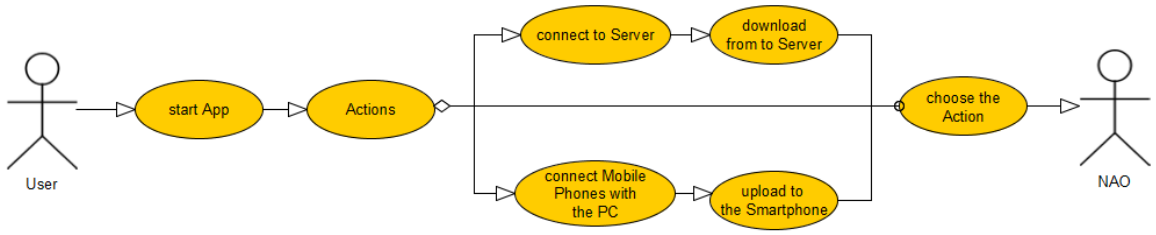
There is also the possibility to control the NAO with a navigation ball.

The third function of the app is sending actions to the NAO which executes them. In the app there is a local library which synchronizes frequently in case the user downloads new actions. Actions can be downloaded from the server or the PC. When the user downloads actions from the server the app calls the server API. In case of downloading from PC the mobile phone has to be connected with it by a cable.

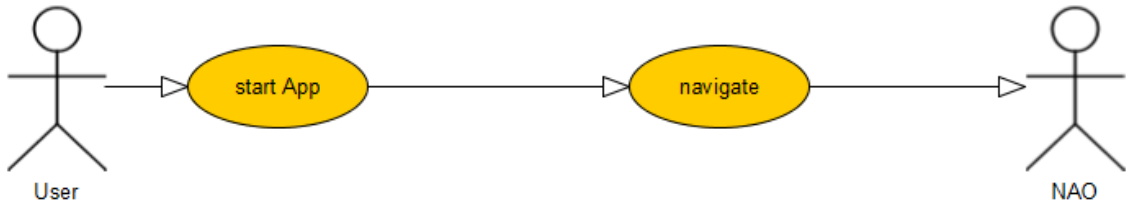

ConnectToNao
Attributes
Operations
ConnectToNao
1
build connection to nao
NaoConnection
1
NaoConnection
Attributes
Operations
NaoInfo
Attributes
Operations
NaoControlMovement
Attributes
Operations
NaoActions
Attributes
Operations
NaoActions
1
1
NaoActions
update actions
update actions
DownloadActionsFromServer
1
1
DownloadActionsFromPC
DownloadActionsFromServer
Attributes
Operations
DownloadActionsFromPC
Attributes
Operations
DownloadActionsFromServer
1
DownloadActionsFromPC
1
download available actions
download available actions
ServerAPI
1
PCConnection
1
ServerAPI
Attributes
Operations
PCConnection
Attributes
Operations

## 1.1.4 Overview of the Business Processes

Process-ID: 01 Action Selection




Process-ID: 02 Navigate the NAO

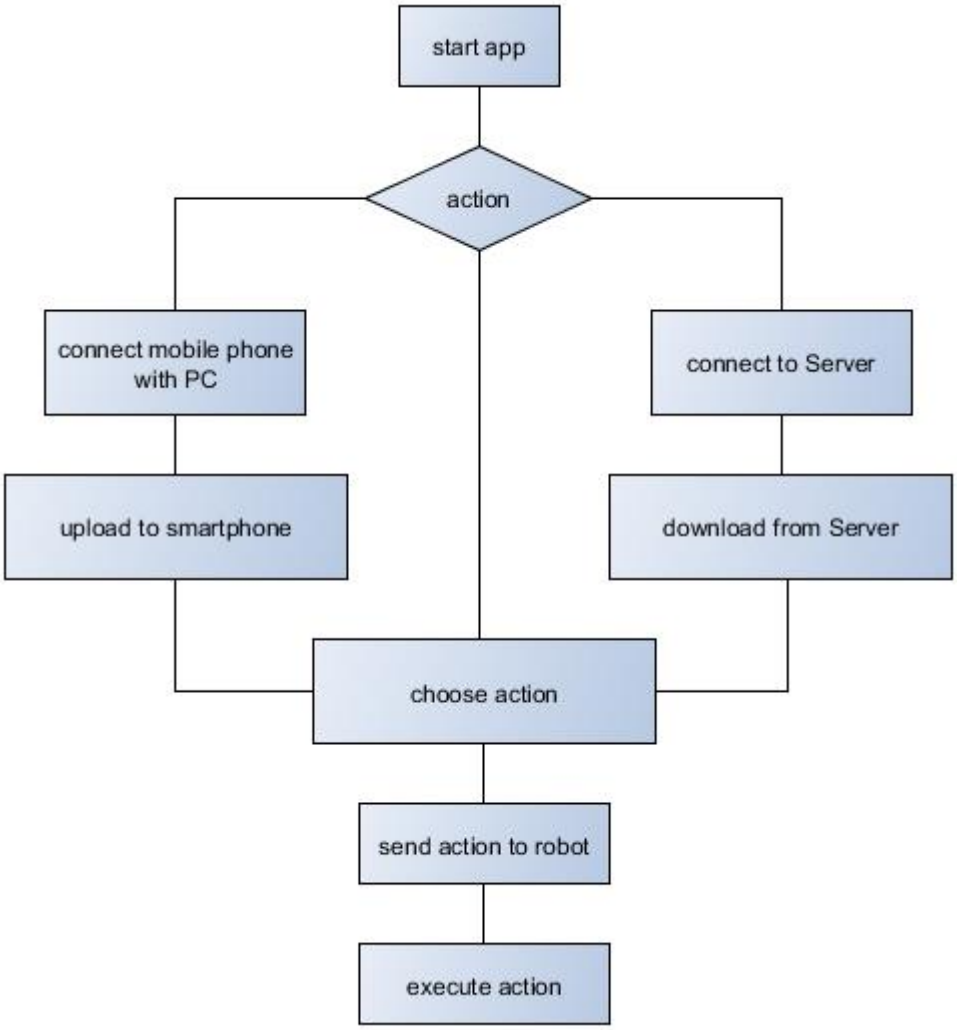



## 1.1.5 Description of the Business Processes

### Description of 01: Action Selection

| | |
|---|---|
| Triggering Event: | If the user want to use one of the Actions, for example to sit down. |
| Result: | The Code behind the Action, which is chosen, sends to the robot and the NAO accomplished the job. |
| Contributors: | The User which interact with the NAO and work with the Application.<br>The App which is the Communication interface between the user and the robot.<br>The NAO, a humanoid robot, which accomplished the tasks. |

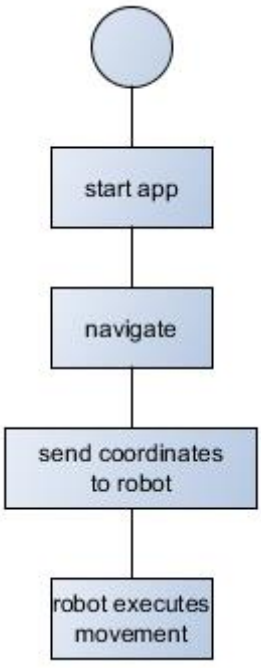
Diagram to Process-ID: 01



## Description of 02: Navigate the NAO

| | |
|---|---|
| Triggering Event: | If the user want to navigate the NAO without writing code or active an action. |
| Result: | If the user pulls the control button in a direction. The NAO should react like the button. That means, if the user pulls the button forward, the NAO will also walk forward. The speed depends on the power the user pulls the button. |
| Contributors: | The User which interact with the NAO and work with the Application.<br>The App which is the Communication interface between the user and the robot.<br>The NAO, a humanoid robot, which walk around. |

Diagram to Process-ID: 02

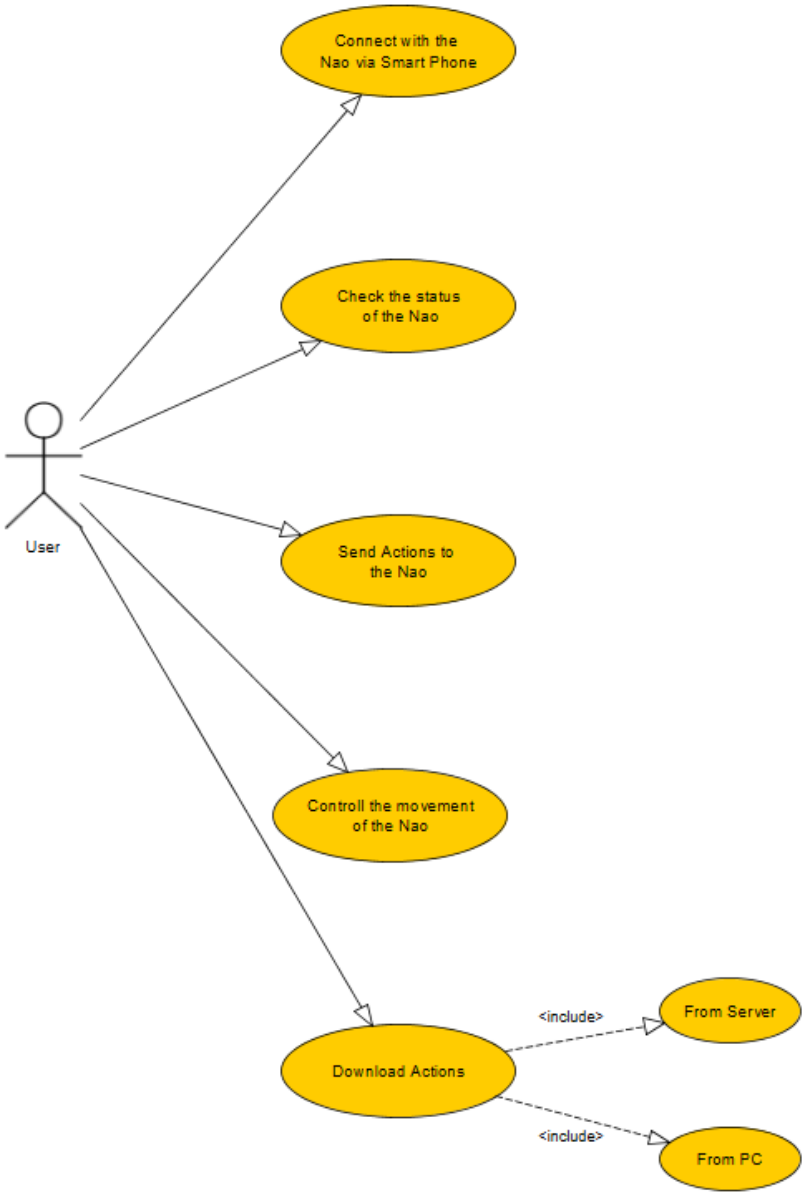


## 1.2 Goal Definition

The goal of this project is to develop an app which helps people who have no programming skills to work with NAO´s.
With prefabricated actions, NAO´s could be easier used in other areas of life, for example in hospitals to cheer up sick children, as the users only have to click on the action and the robot executes it.
The target group are mainly programmers. They develop solutions afterwards they are able to upload them on the app. But also people who have no programming skills could profit, as they only have to select their required program.
That means the time to connect with a robot is reduced by 10 minutes via smartphone and existing wireless point.

# 2 Functional Requirements - APP

## 2.1 Use Case Diagrams



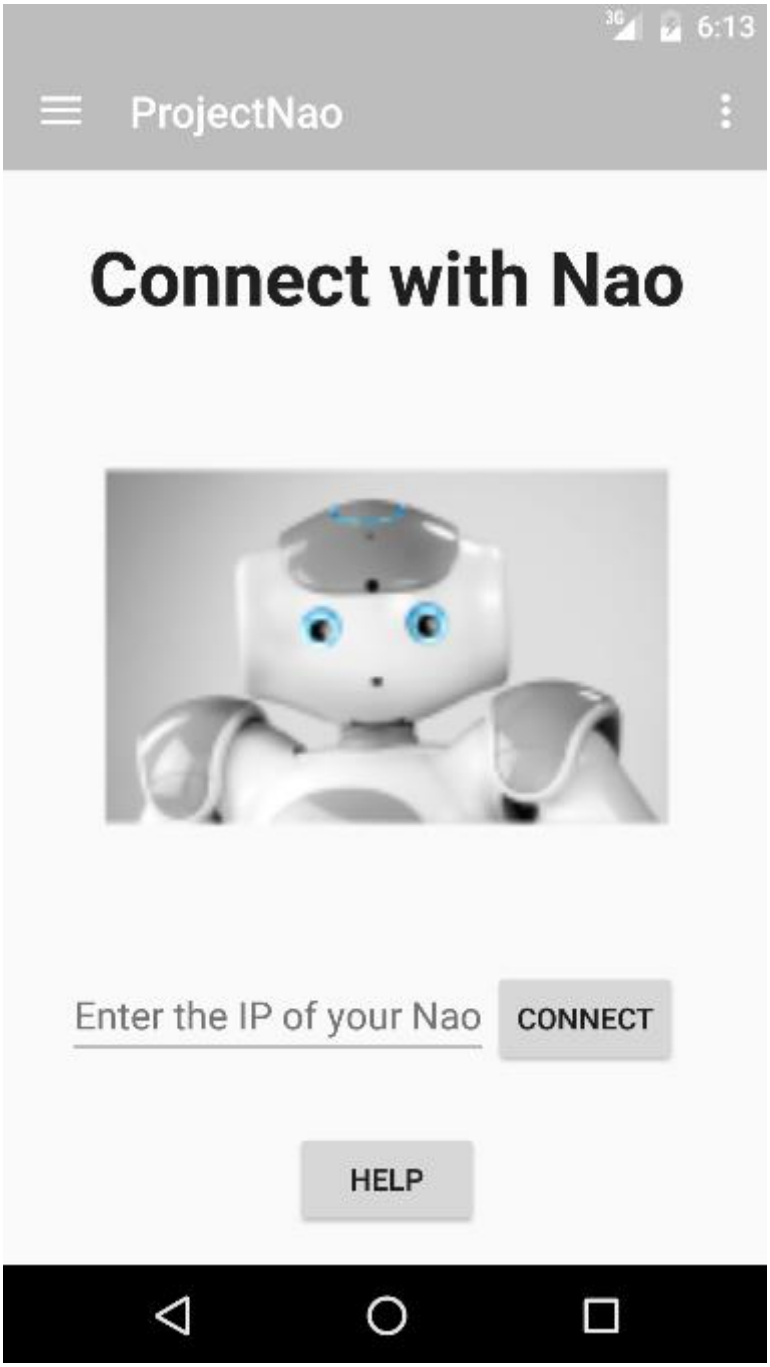
## 2.2 Connect with the NAO via Smartphone

This use case describes how to connect your mobile phone with the NAO. The connection process works through the network. There is a text field where the IP address of the NAO should be entered.

### 2.2.1 Characteristic Information

| | |
|---|---|
| Superior business process: | Process-ID: Connect with the NAO |
| Goal: | Start a connection with the NAO via Smartphone |
| Precondition: | Own a NAO, knowing the IP address of the NAO, Smartphone and NAO have to be in the same network |
| Postcondition: | Status of the NAO is available and actions can be executed |
| Involved User: | User |
| Triggering Event: | - |

### 2.2.2 GUI to call the use case



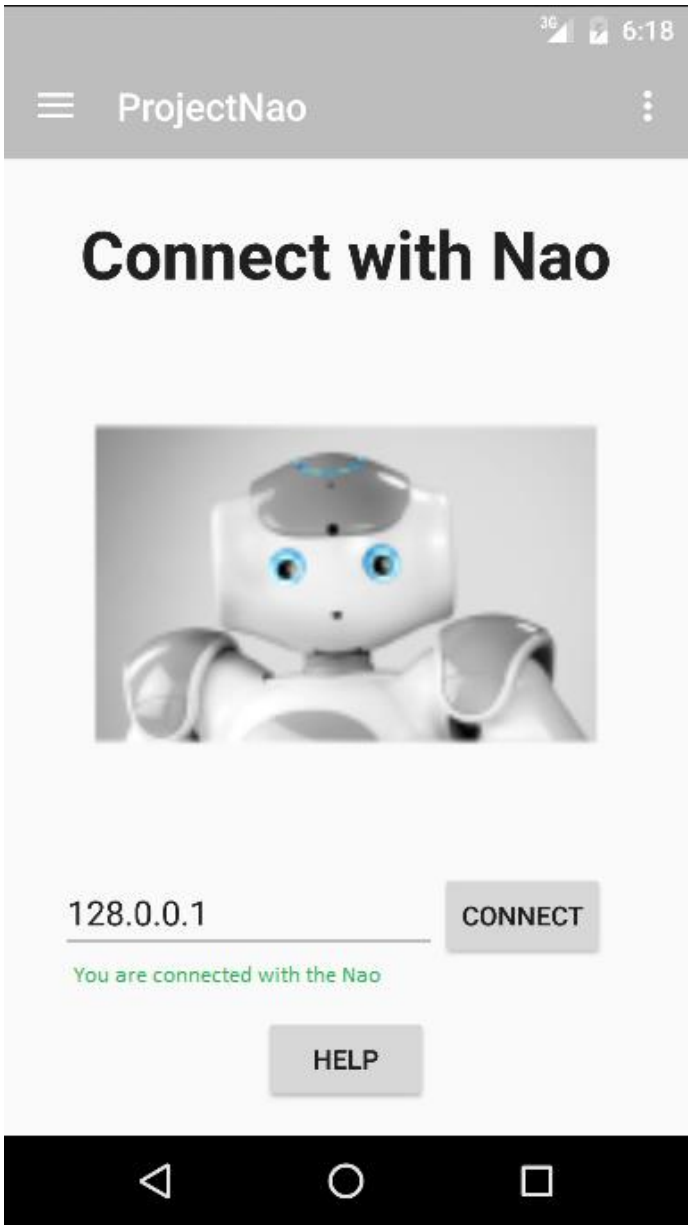

| Input field | Valid inputs |
|---|---|
| Enter the IP of your NAO | The valid input is a IP- Adress |

### 2.2.3 Scenario for the standard use (good case)

| Step | User | Activity |
|---|---|---|
| Input field IP address | User | Enter the correct IP of your NAO |
| Click Connect Button | User | The smartphone is connected with the NAO |
| Click Help Button | User | The user gets a description how to connect with the NAO successfully |

### 2.2.4 GUIs for the standard use

In this case a valid IP-address (IP of your NAO) is entered. Then the ‘Connect’ Button get pressed and the text ‘You are connected with the NAO’ is shown. This means that the connection with the NAO is established successfully.

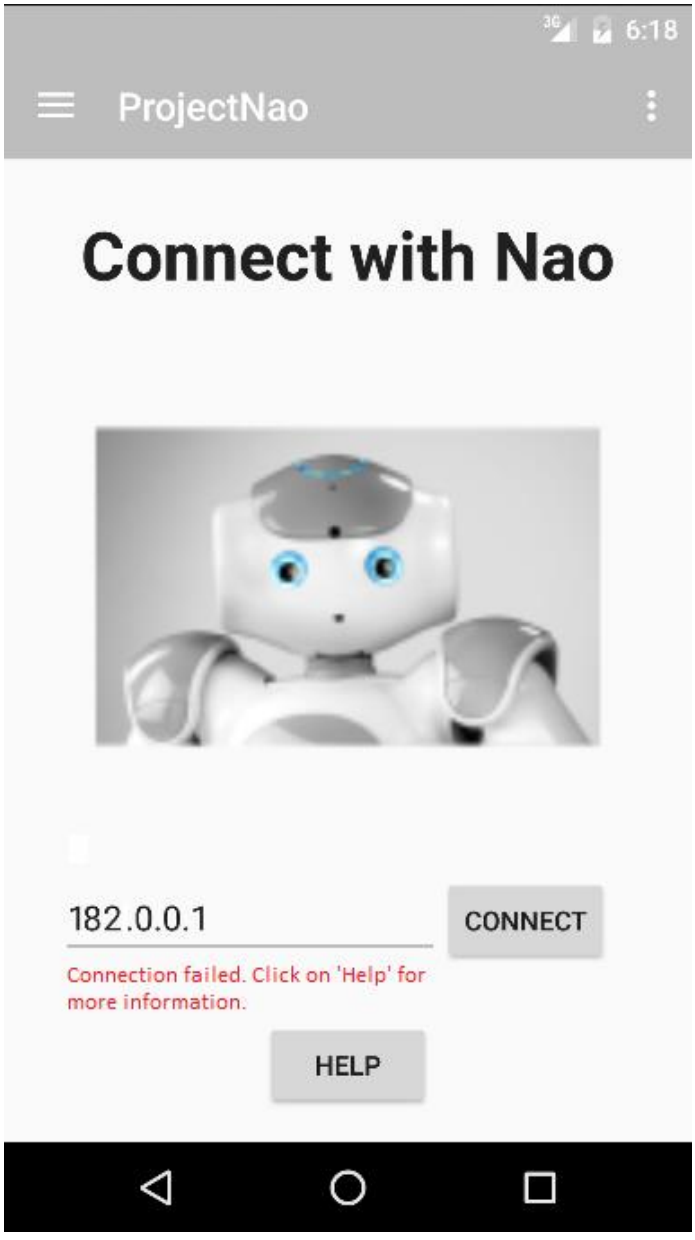


### 2.2.5 Scenarios for non-standard uses (bad cases or work around cases)

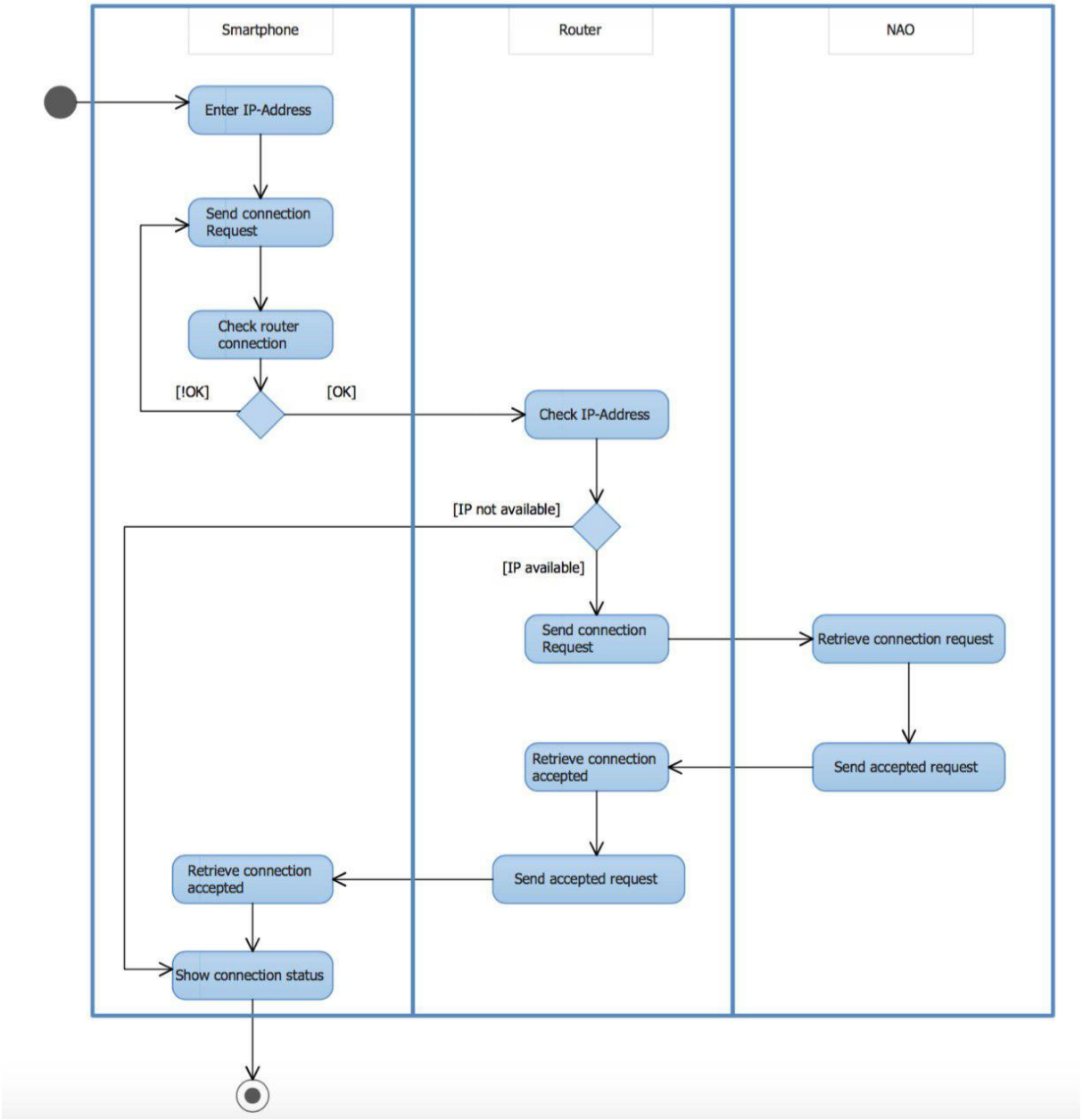
| Step | User | Activity |
|---|---|---|
| Input field IP adress | User | Enter an incorrect IP |
| Click Connect Button | User | The smartphone is not connected with the NAO |

### 2.2.6 GUIs for the non-standard uses

In this case an invalid or a valid IP-address (but not from the NAO) is entered. Then the ‘Connect’ Button get pressed and the text ‘Connection failed. Click on ‘Help’ for more information’. That means that the connection with the NAO is not established successfully.



## 2.2.7 Workflow




## 2.2.8 Open Points

- Actually, there are no open points.

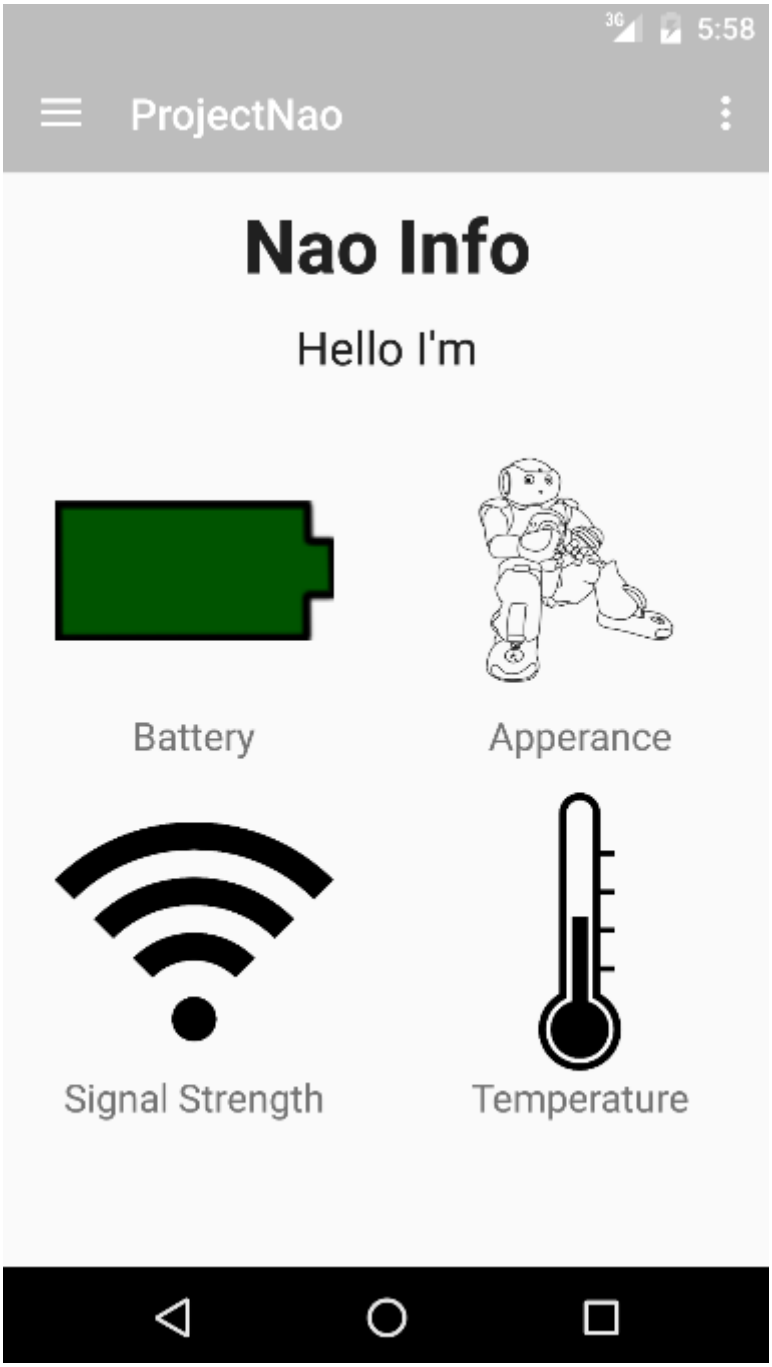
## 2.3 Check the status of the NAO

This use case describes the status of the NAO. That includes the battery, the signal strength, the temperature and the appearance. There are 4 pictures which show the current status of each field.

### 2.3.1 Characteristic Information

| | |
|---|---|
| Superior business process: | Process-ID: Status of NAO |
| Goal: | It shows the current status and the name of the NAO |
| Precondition: | Mobile phone is connected with the NAO |
| Postcondition: | - |
| Involved User: | User |
| Triggering Event: | - |

### 2.3.2 GUI to call the use case



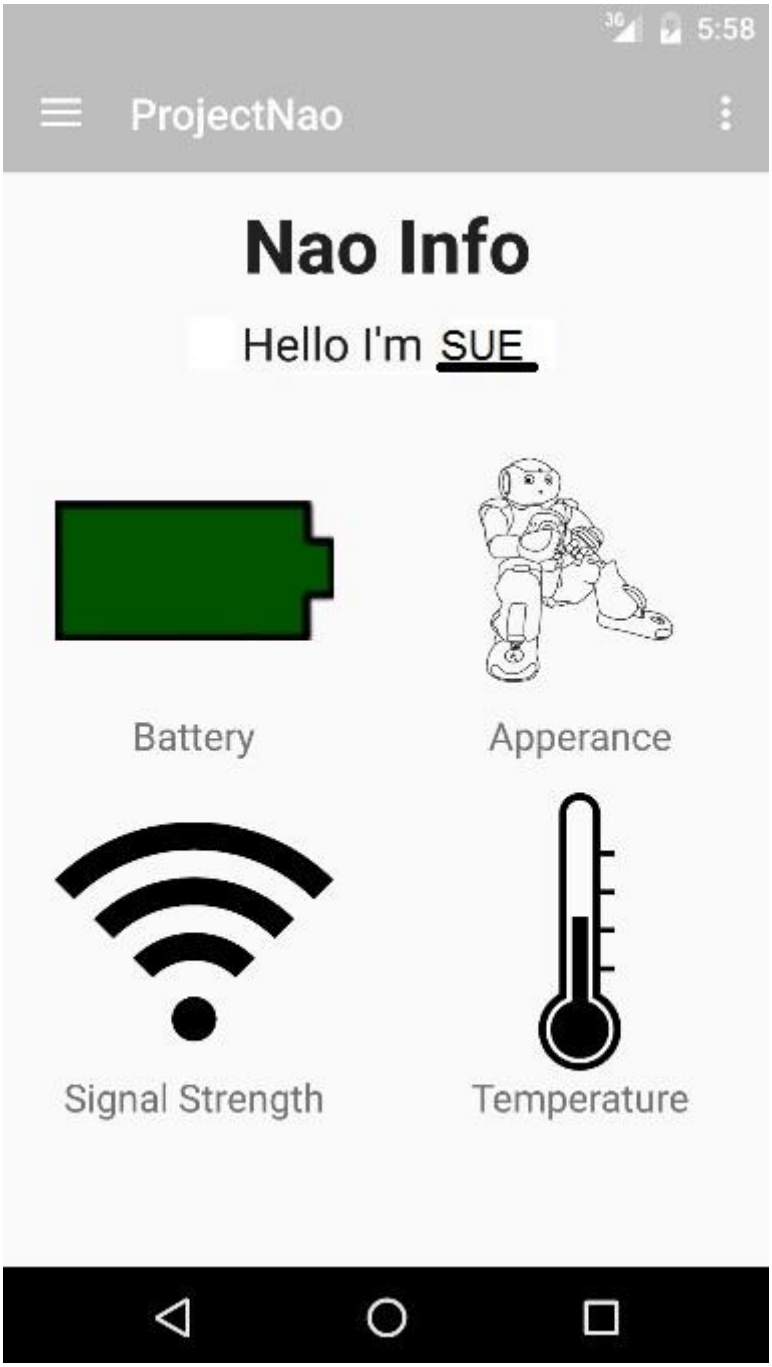

| Input field | Valid inputs |
|---|---|
| - | - |

### 2.3.3 Scenario for the standard use (good case)

| Step | User | Activity |
|---|---|---|
| Got to Status of NAO view | User | Status and name of NAO is shown |

### 2.3.4 GUIs for the standard use

In this case the user goes to the Status of NAO view and he sees the current battery status, temperature status, signal strength and appearance. The name of the NAO (in this case ‘SUE’) is also shown.

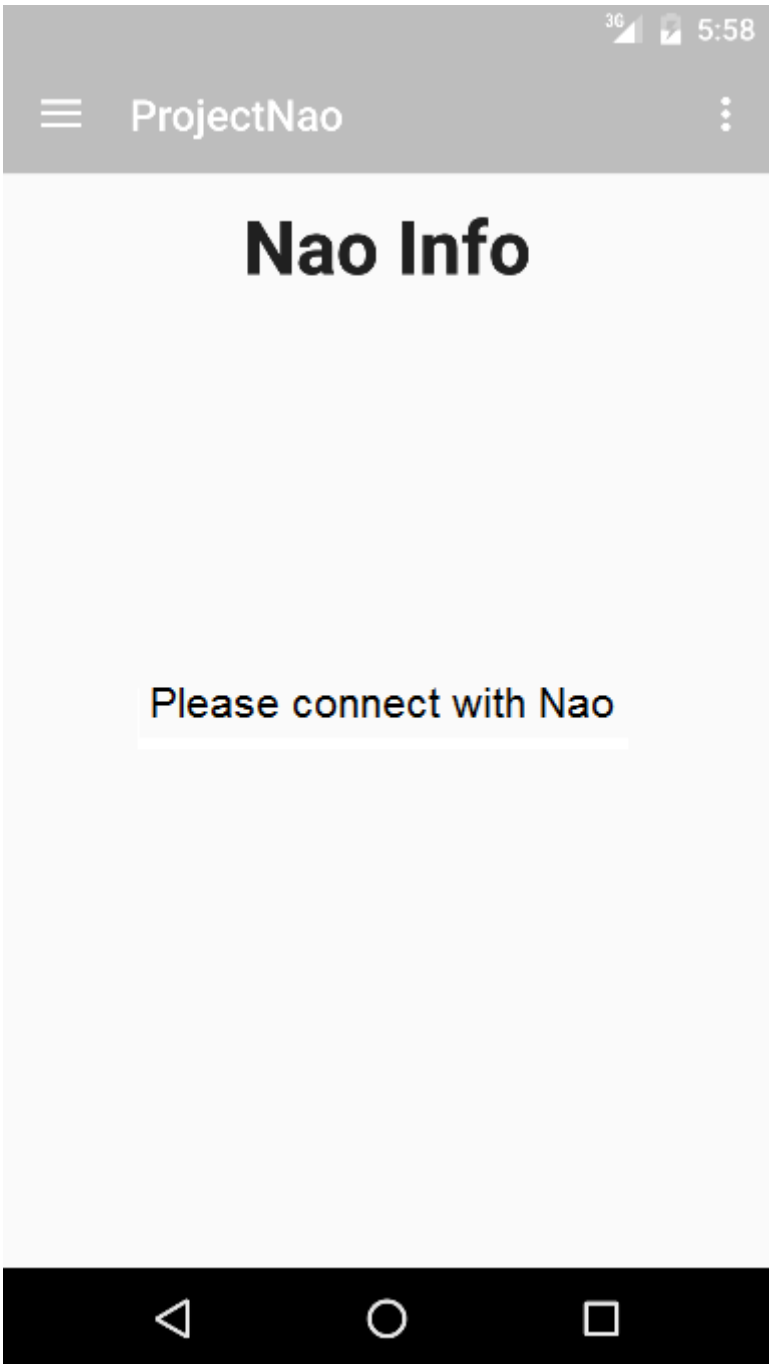


### 2.3.5 Scenarios for non-standard uses (bad cases or work around cases)

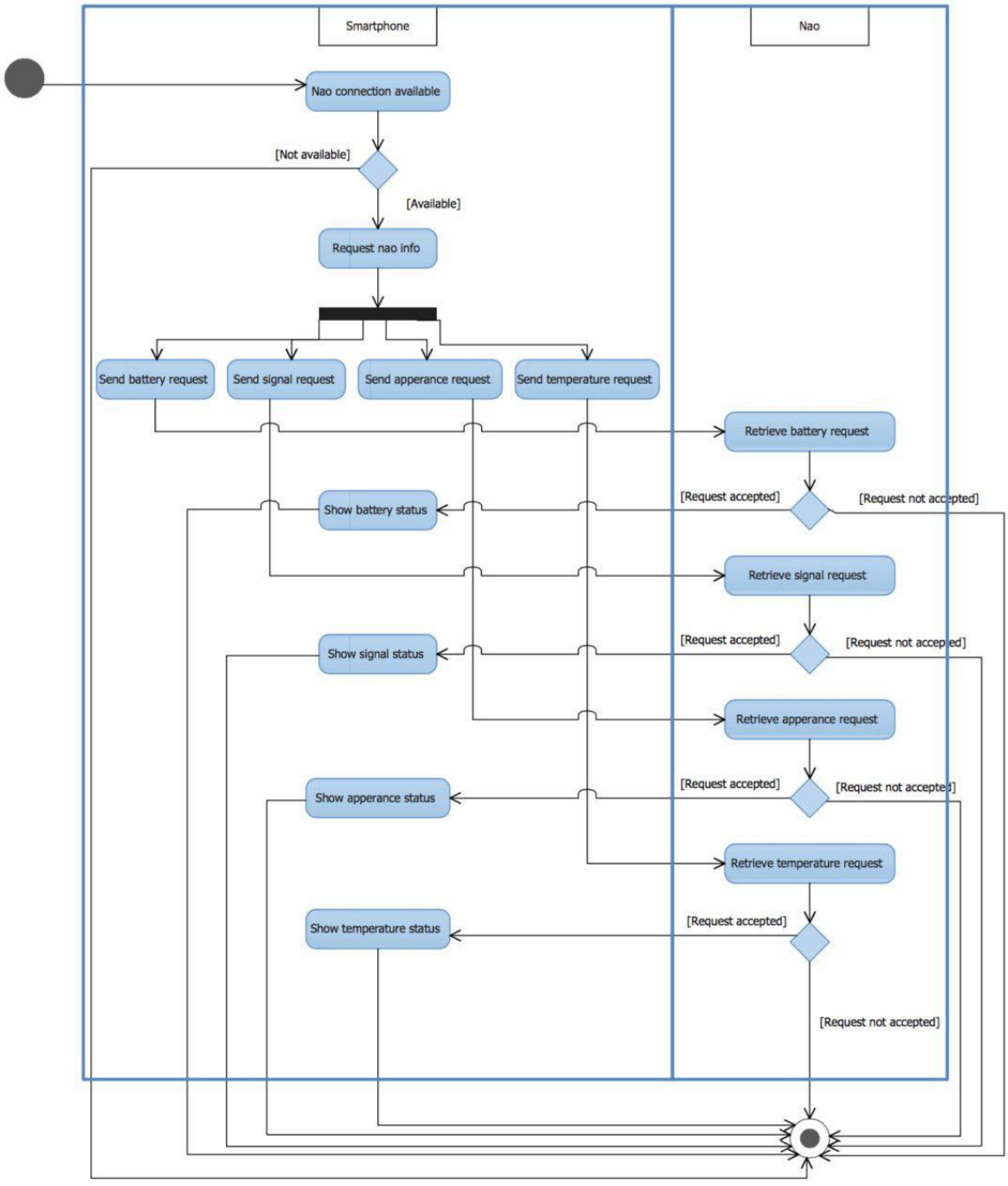
| Step | User | Activity |
|---|---|---|
| Go to Status of NAO View | User | Text ‘Please connect with NAO’ is shown |

### 2.3.6 GUIs for the non-standard uses

In this case, the user goes to the status of NAO view and the battery status, signal strength, temperature, appearance and name are not shown. Instead, the text ‘Please connect with NAO’ is shown.



## 2.3.7 Workflow




## 2.3.8 Open Points

Actually, there are no open points.

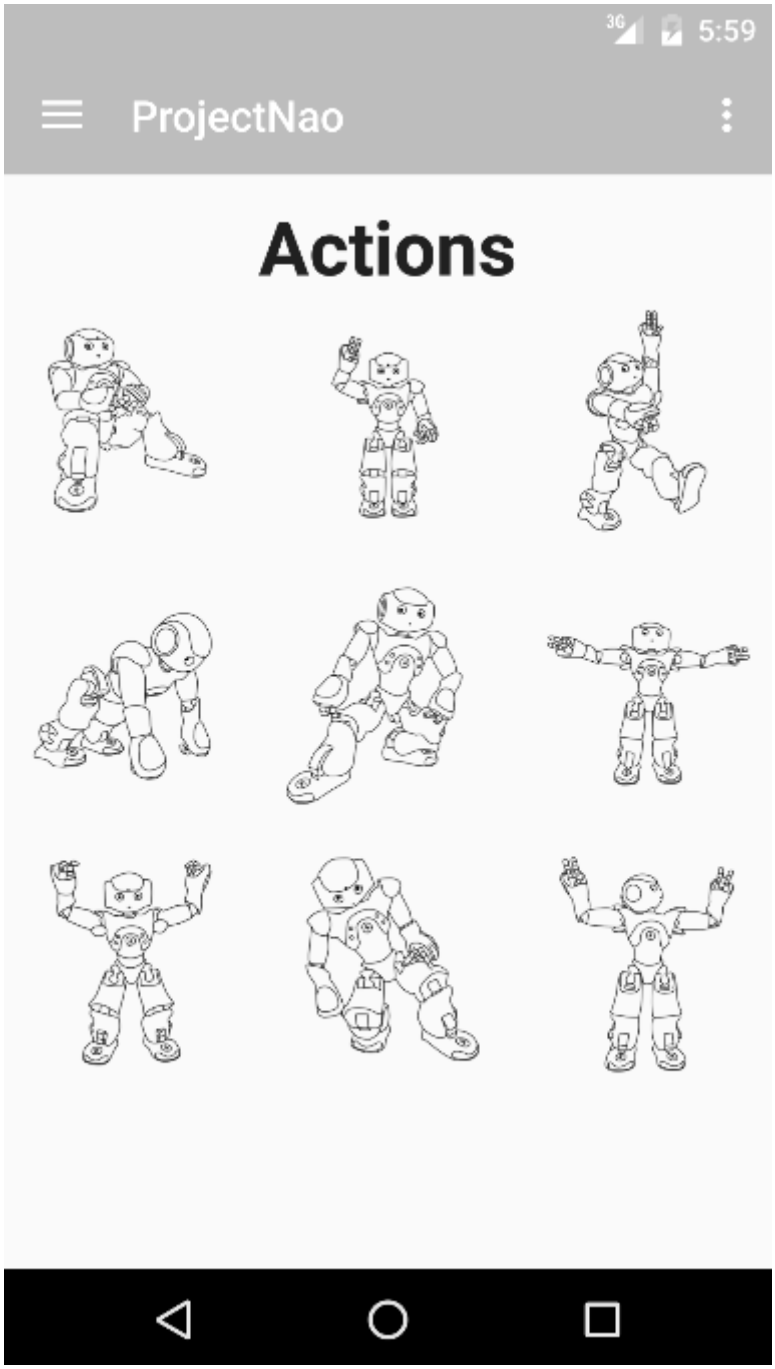
## 2.4 Send Actions to the NAO

This use case shows all the actions which the NAO can execute. All the actions are shown by a picture and when one of them get selected the action is sent to the NAO. If there are new action available (Actions from Server/PC) the list refreshes automatically. After the app is closed the actions are still on the phone.

### 2.4.1 Characteristic Information

| | |
|---|---|
| Superior business process: | Process-ID: Send Actions to NAO |
| Goal: | Sends the chosen action to the NAO successfully. |
| Precondition: | Mobile phone is connected with the NAO. |
| Postcondition: | The NAO executes the action. |
| Involved User: | User |
| Triggering Event: | - |

### 2.4.2 GUI to call the use case



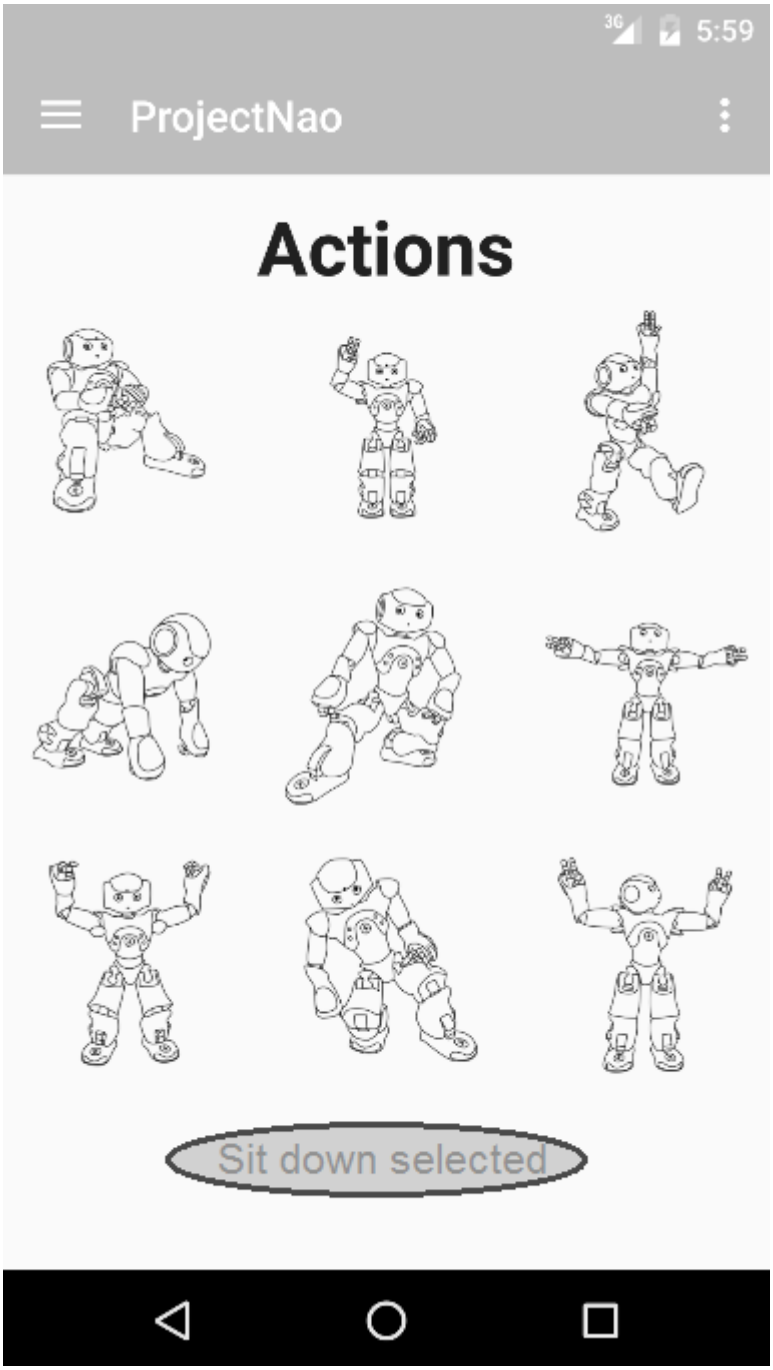

| Input field | Valid inputs |
|---|---|
| - | - |

### 2.4.3 Scenario for the standard use (good case)

| Step | User | Activity |
|---|---|---|
| Action get selected | User | Text '<Action Name> selected' is shown. Action is sent to the NAO. NAO executes the Action |

### 2.4.4 GUIs for the standard use

In this case the user selects one of the given actions. The name of the action + selected is shown on the display. Few seconds later the NAO executes the action.

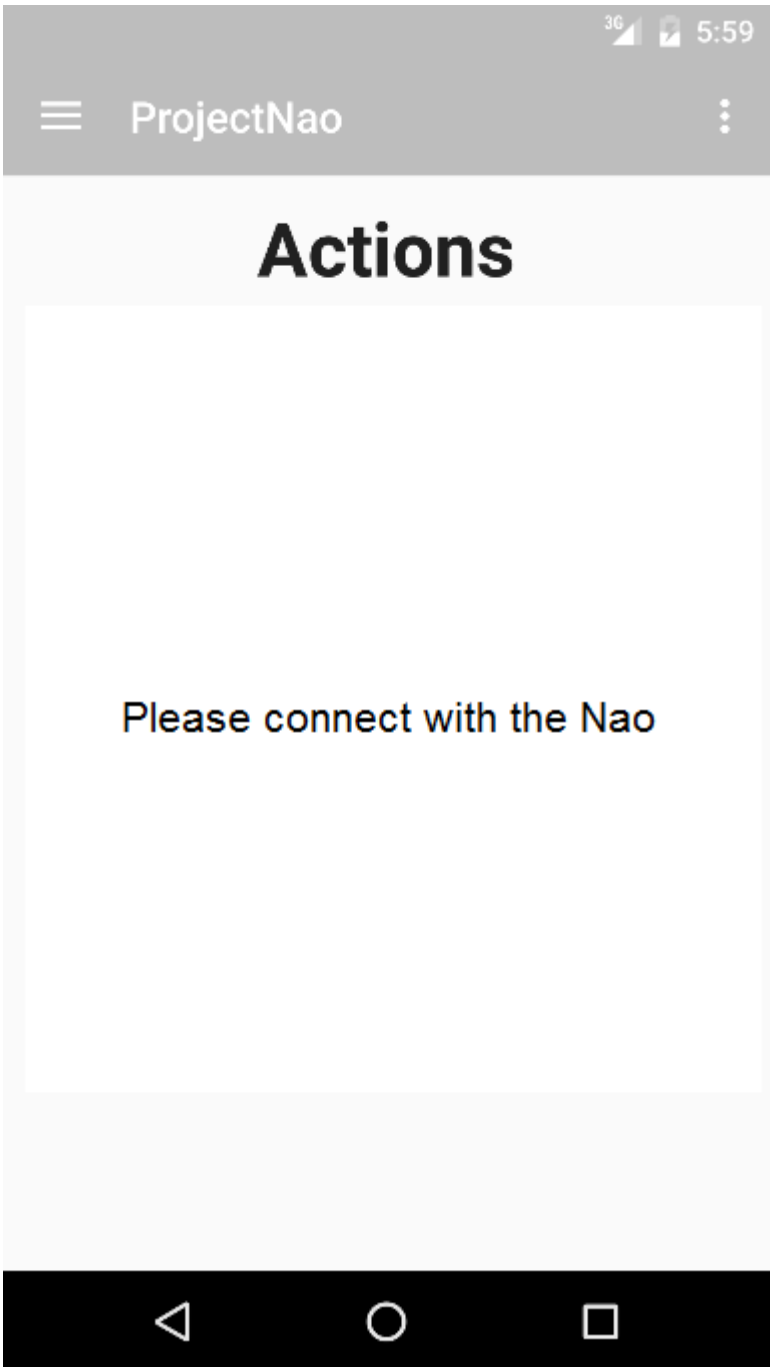


### 2.4.5 Scenarios for non-standard uses (bad cases or work around cases)

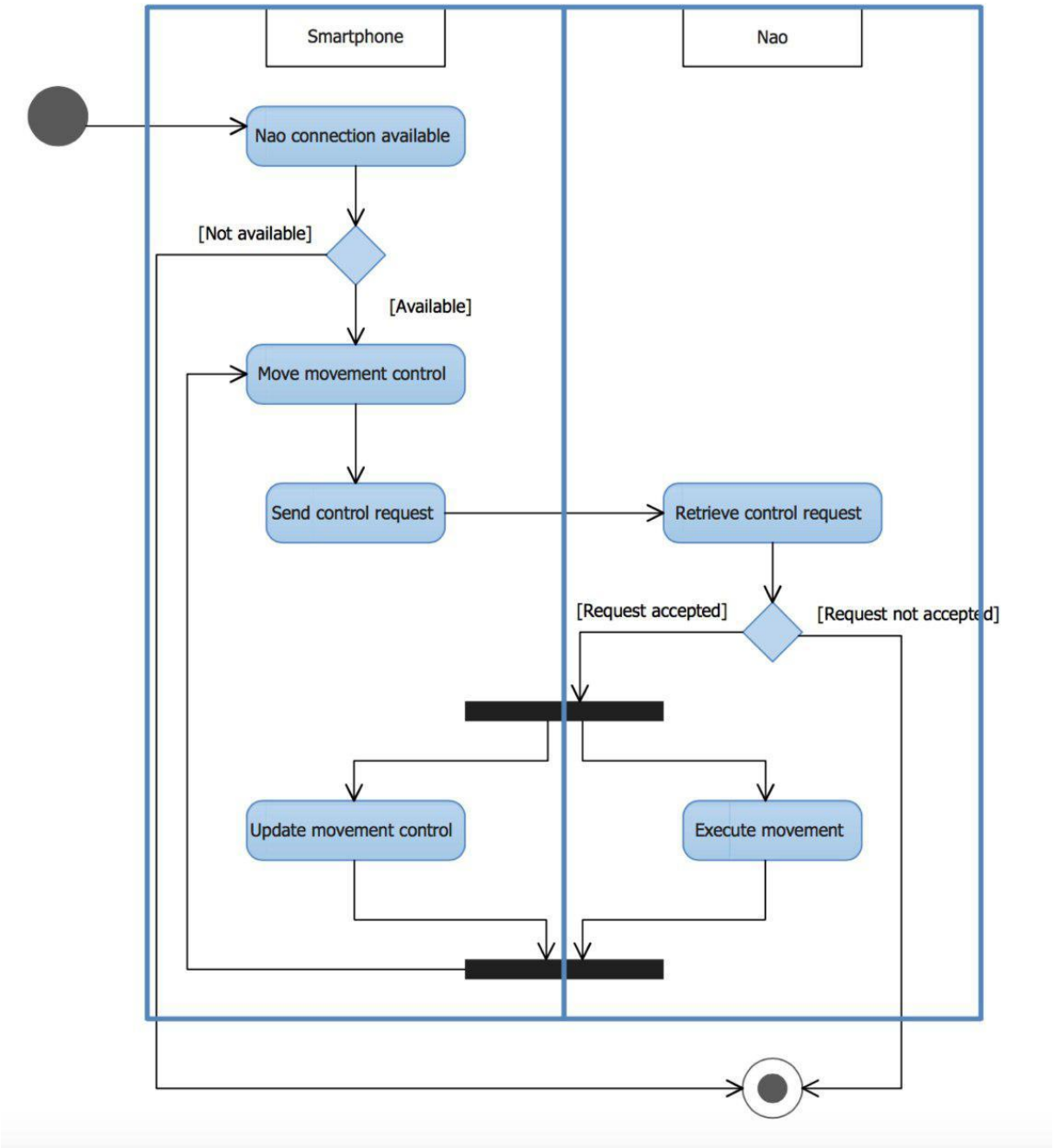
| Step | User | Activity |
|---|---|---|
| Action get selected | User | Smartphone is not connected to the NAO. |

### 2.4.6 GUIs for the non-standard uses

In this case the mobile phone is not connected with the NAO. The result of this is an empty view with a headline and the tip that the phone should be connected with the NAO.



### 2.4.7 Workflow




### 2.4.8 Open Points

Actually, there are no open points.

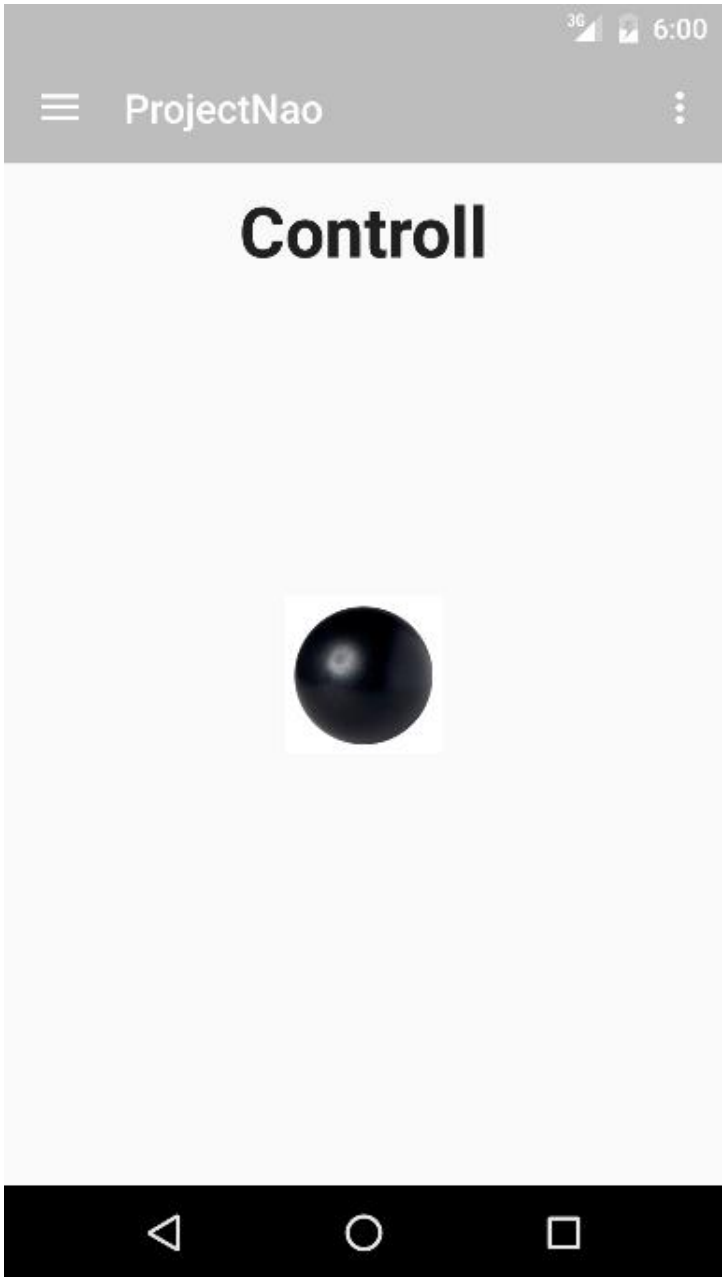
## 2.5 Control the movement of the NAO

This use case describes how to control the movement of the NAO with a drag and drop ball on the screen. In each direction the ball is moved the NAO also moves in that direction.

### 2.5.1 Characteristic Information

| | |
|---|---|
| Superior business process: | Process-ID: Control the movement |
| Goal: | The user should be able to control the movement of the NAO with a drag and drop ball. |
| Precondition: | The smart phone has to be connected with the NAO. The NAO has to be in an erect position. |
| Postcondition: | The NAO moves in the direction in which the ball is moved |
| Involved User: | User |
| Triggering Event: | - |

### 2.5.2 GUI to call the use case



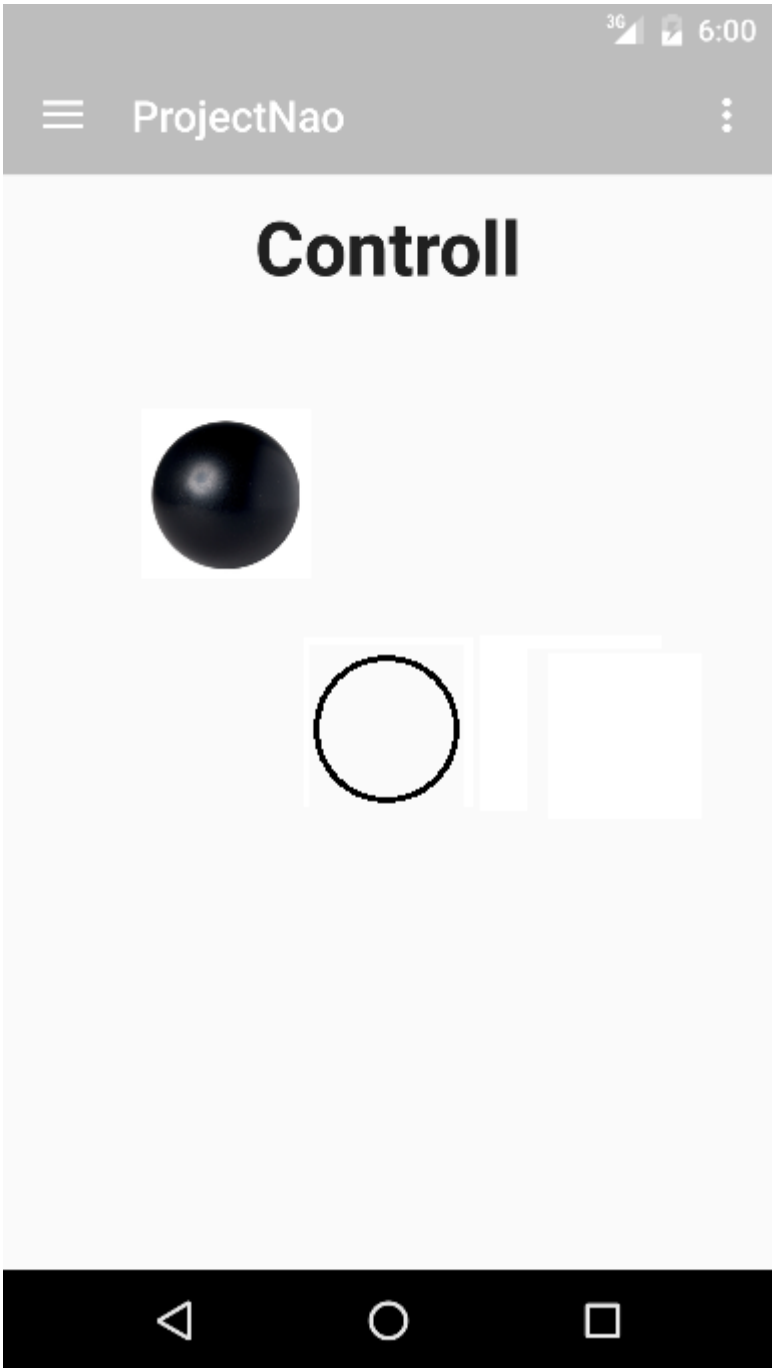

| Input field | Valid inputs |
|---|---|
| - | - |

### 2.5.3 Scenario for the standard use (good case)

| Step | User | Activity |
|---|---|---|
| Ball gets dragged over the screen | User | NAO moves like the ball |

### 2.5.4 GUIs for the standard use

In this case the user touches the ball and moves it over the display.



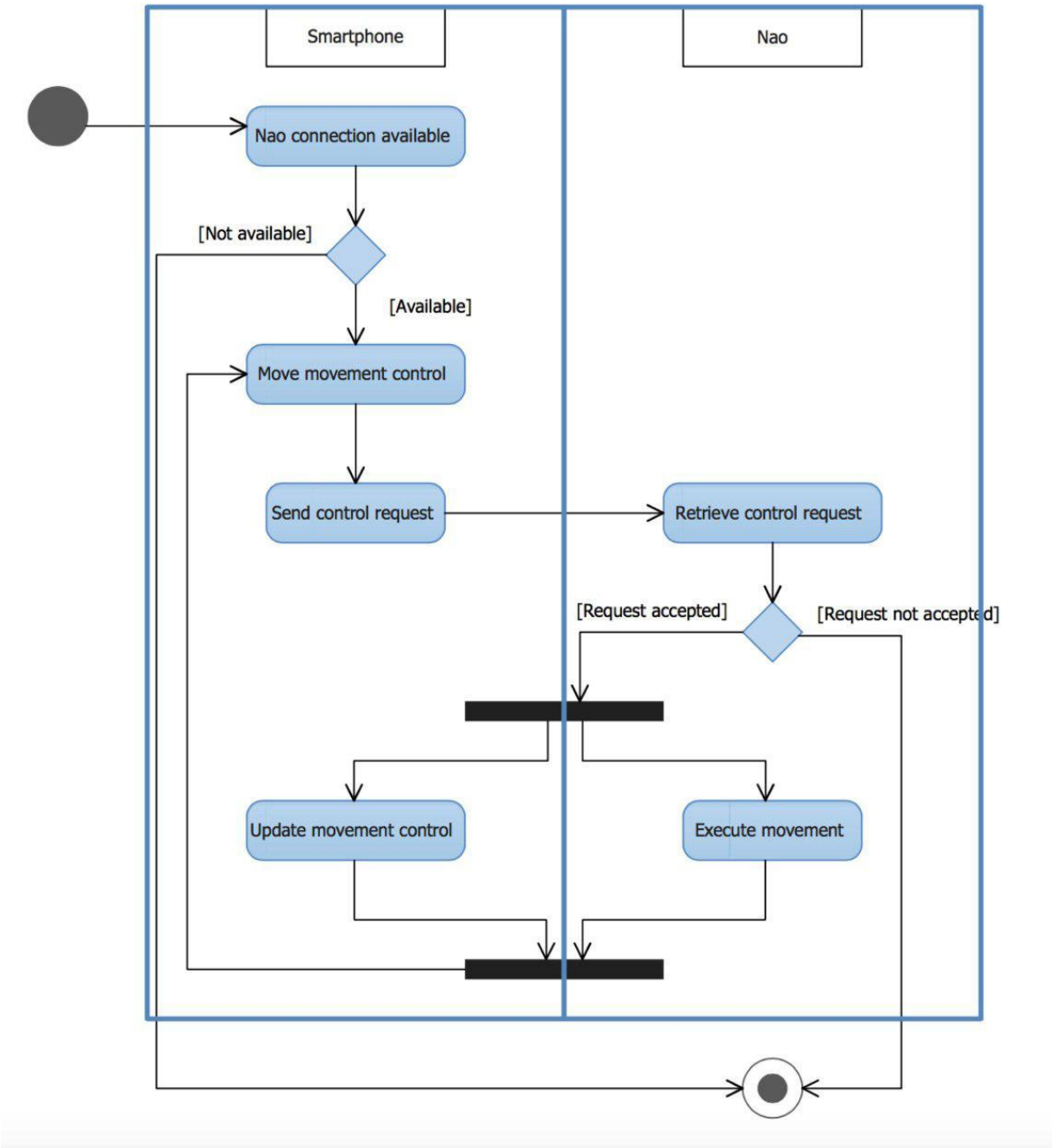
### 2.5.5 Scenarios for non-standard uses (bad cases or work around cases)

| Step | User | Activity |
|---|---|---|
| Go to Control view | User | Mobile phone is not connected with the NAO |

### 2.5.6 GUIs for the non-standard uses

In this case the mobile phone is not connected with the NAO. The result of this is an empty view with a headline and a message which says that the phone is not connected with the NAO.

### 2.5.7 Workflow




### 2.5.8 Open Points

Actually, there are no open points.

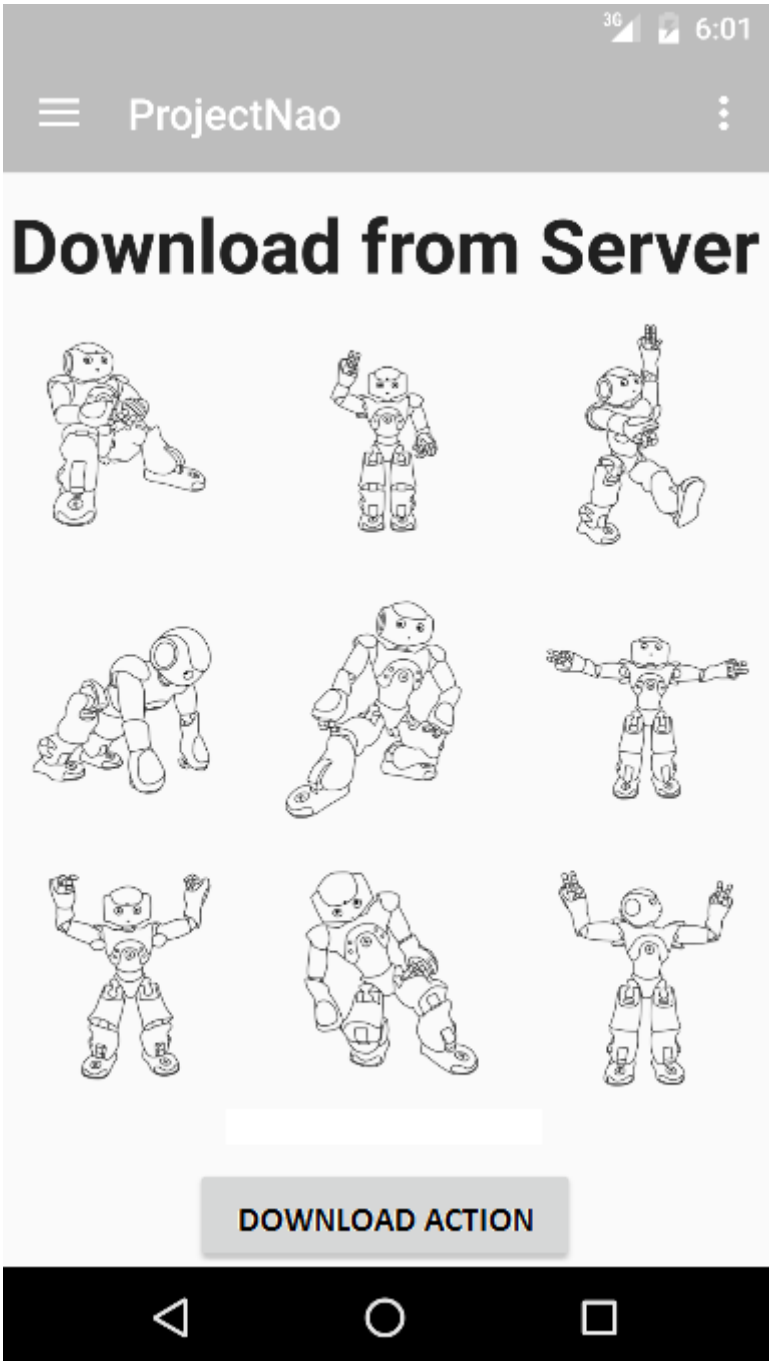
## 2.6 Download Actions from Server

This use case describes how to connect with the server and download actions. There is a button ‘Download Action’ and when it gets clicked the selected Action is stored in the local storage. The view ‘Action’ will be synchronised and the action can be sent to the NAO.

### 2.6.1 Characteristic Information

| | |
|---|---|
| Superior business process: | Process-ID: Download Server |
| Goal: | The user is able to download actions from the Server. |
| Precondition: | Smartphone has an internet connection and enough memory for the actions. |
| Postcondition: | New actions are available in the app. |
| Involved User: | User |
| Triggering Event: | - |

### 2.6.2 GUI to call the use case



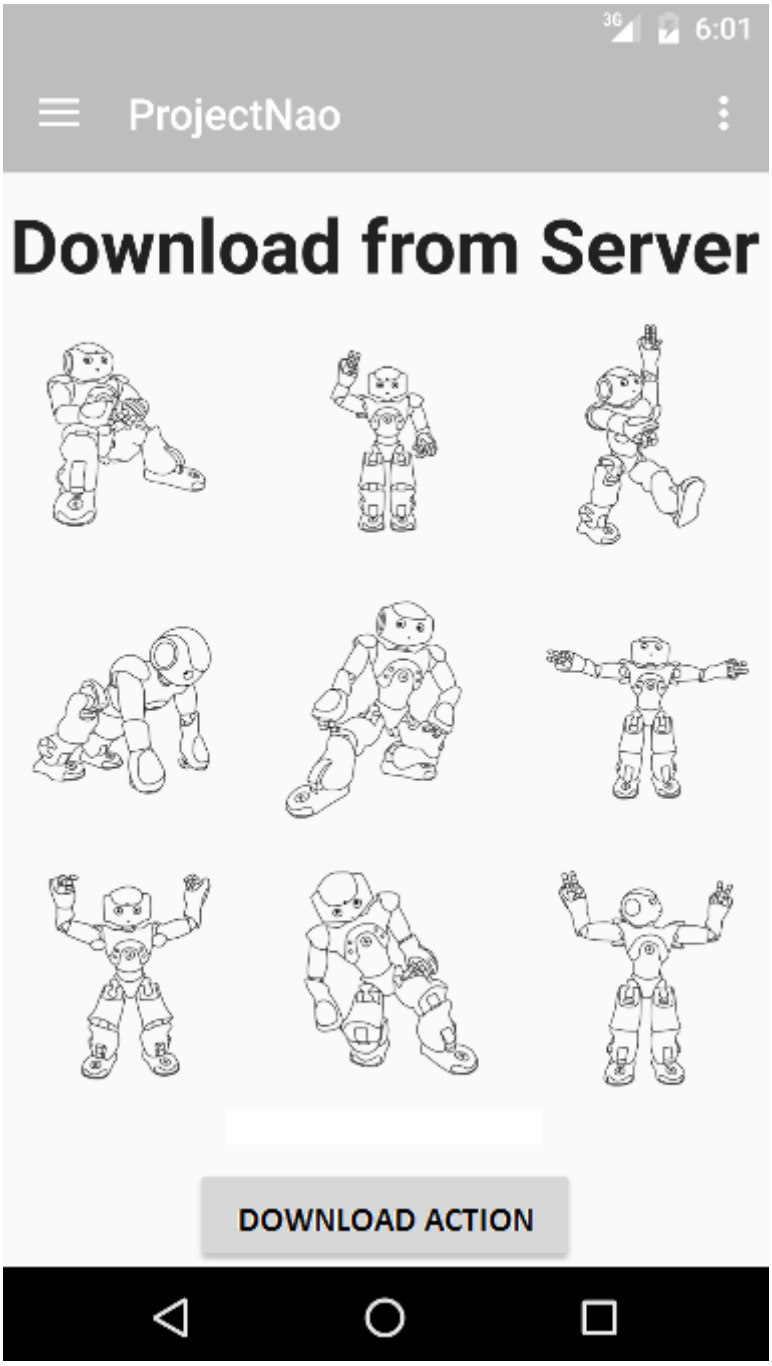

| Input field | Valid inputs |
|---|---|
| - | - |

### 2.6.3 Scenario for the standard use (good case)

| Step | User | Activity |
|---|---|---|
| Press ‘Download Action’ button | User | New Actions are available in the app |

### 2.6.4 GUIs for the standard use

In this case the smartphone has an internet connection and is able to connect to the server successfully. There are different actions which can be downloaded.

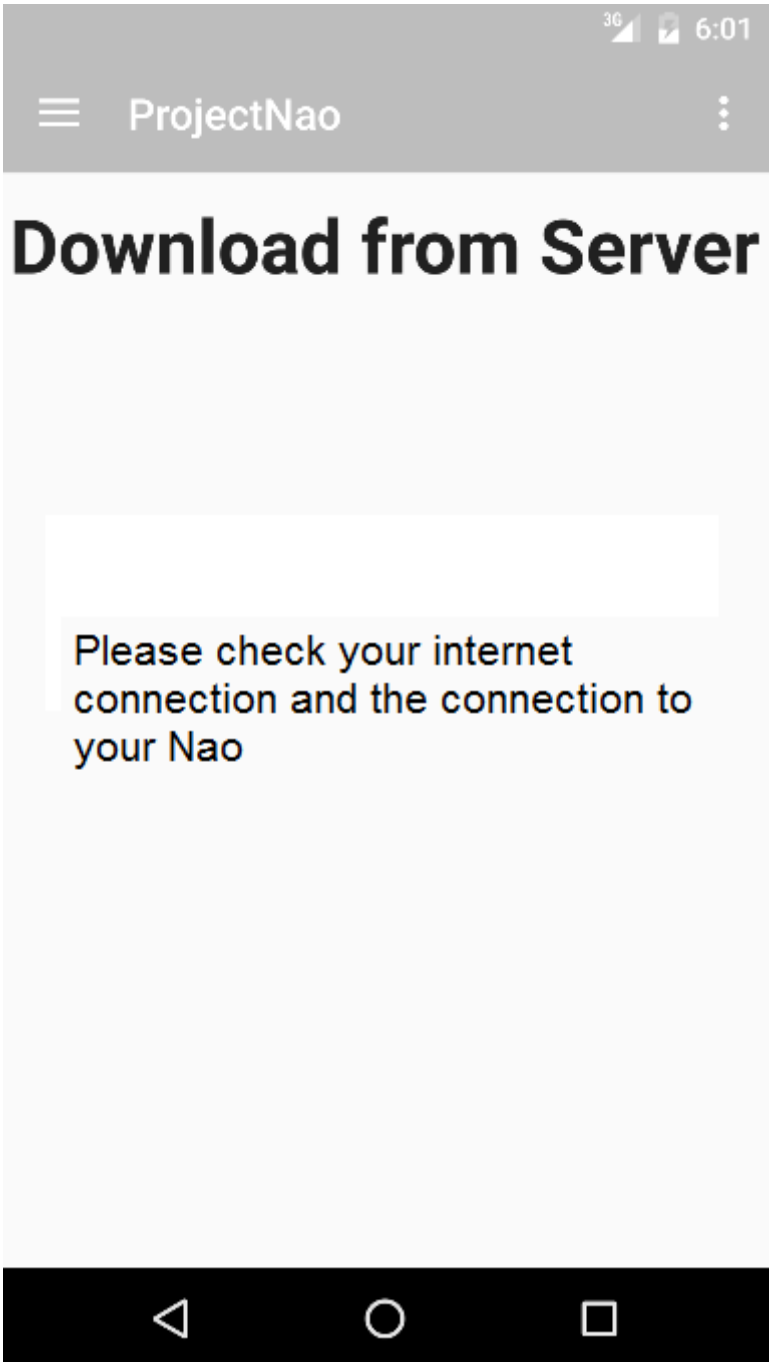


### 2.6.5 Scenarios for non-standard uses (bad cases or work around cases)

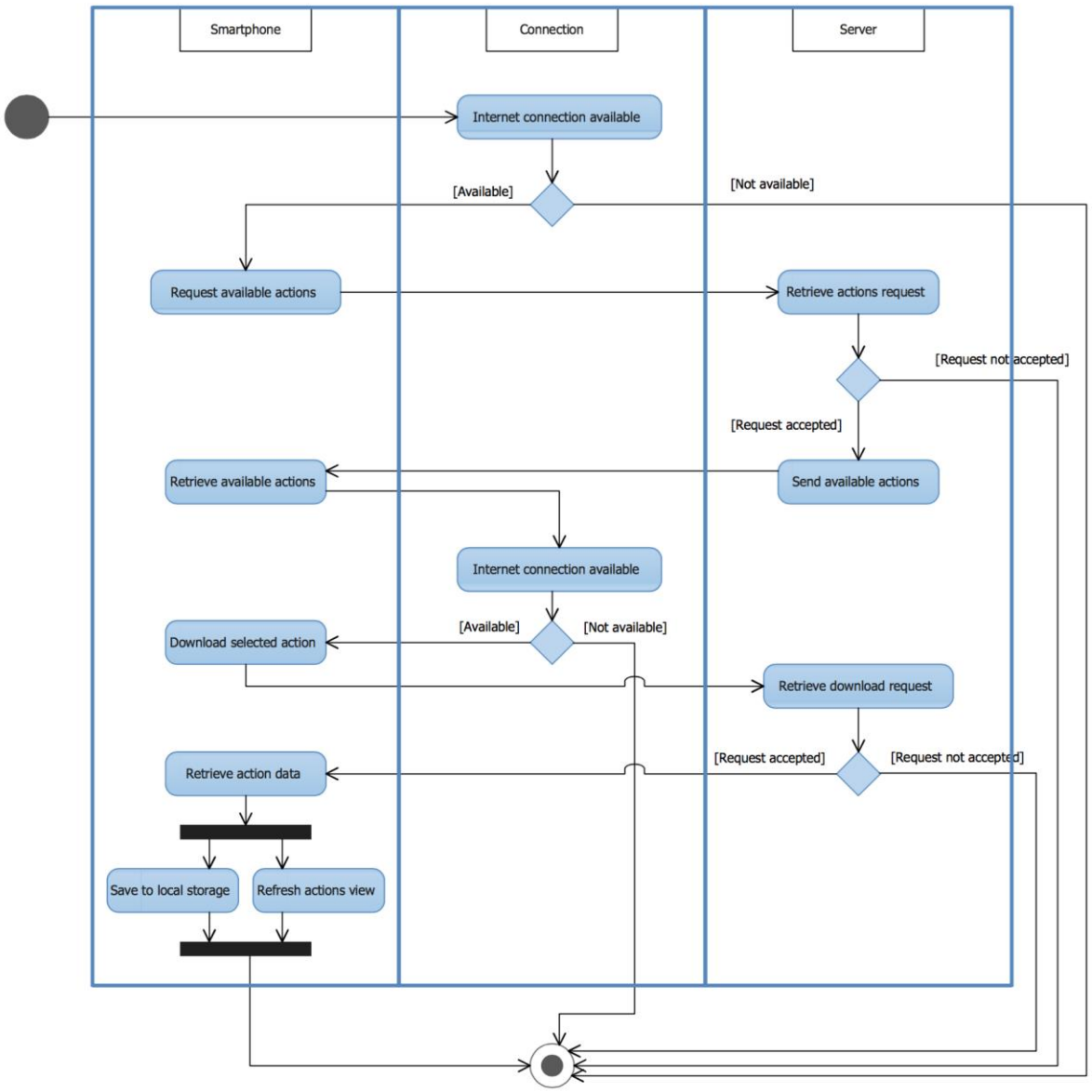
| Step | User | Activity |
|---|---|---|
| Press ‘Download Action’ button | User | An error message is shown on the bottom of the view |

### 2.6.6 GUIs for the non-standard uses

In this case the smart phone has no internet connection. Therefore it can’t connect to the server successfully and no actions are shown or can be downloaded.



## 2.6.7 Workflow




## 2.6.8 Open Points

Actually, there are no open points.

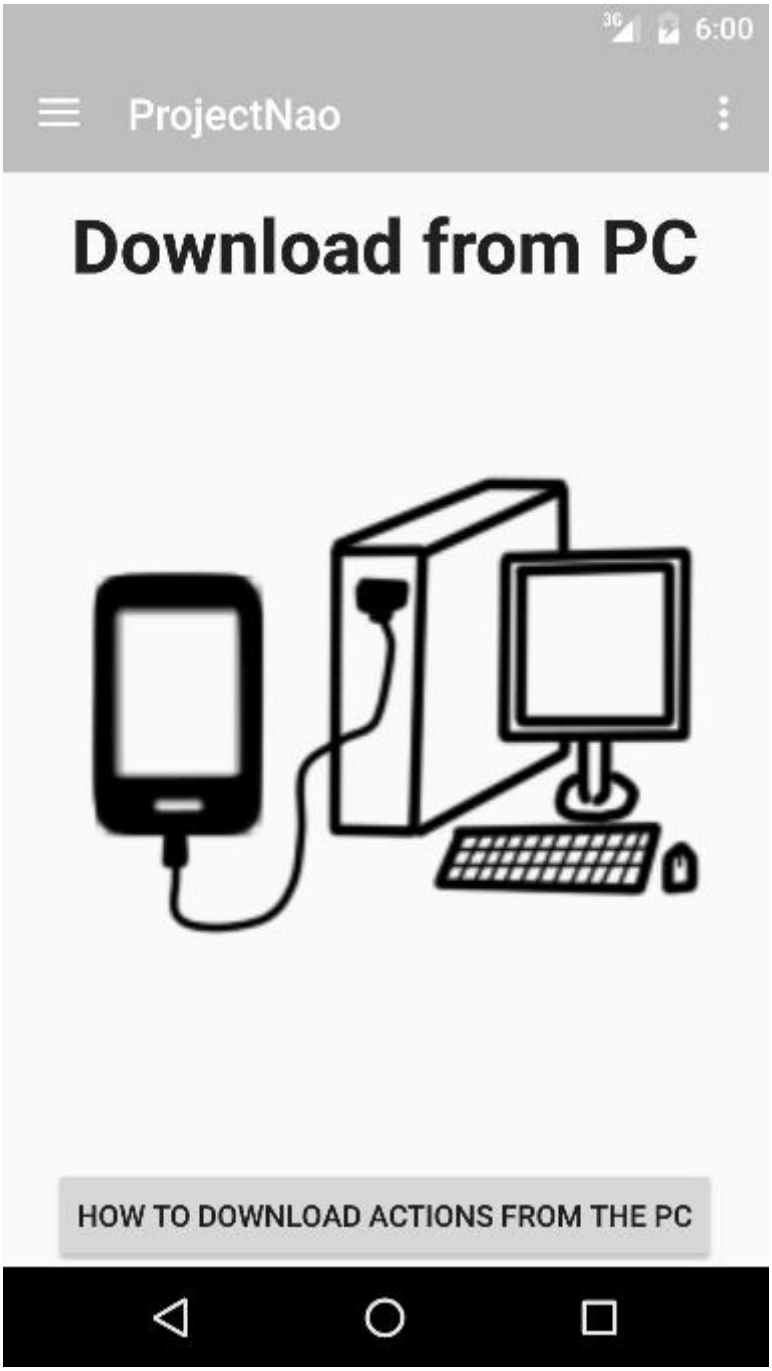
## 2.7 Download Actions from PC

This use case describes how to connect with the PC and download actions. There is a button ‘How to download Actions from the PC’. When it gets clicked the user gets an instruction. The Actions are stored in the local storage. The view ‘Action’ will be synchronised and the action can be sent to the NAO.

### 2.7.1 Characteristic Information

| | |
|---|---|
| Superior business process: | Process-ID: Download PC |
| Goal: | The user is able to download actions from the PC. |
| Precondition: | Cable to connect the smart phone with the PC. Enough memory for the actions. |
| Postcondition: | New actions are available in the app. |
| Involved User: | User |
| Triggering Event: | - |

### 2.7.2 GUI to call the use case




| Input field | Valid inputs |
|---|---|
| - | - |

### 2.7.3 Scenario for the standard use (good case)

| Step | User | Activity |
|---|---|---|
| Press ‘How to download Actions from the PC’ button | User | User gets linked to another view in which the introduction is |

### 2.7.4 GUIs for the standard use

In this case the user has a cable to connect with the PC. The actions have to be copied in the app structure. Then, the view ‘Action’ gets refreshed autonomic and the user can send them to the NAO.

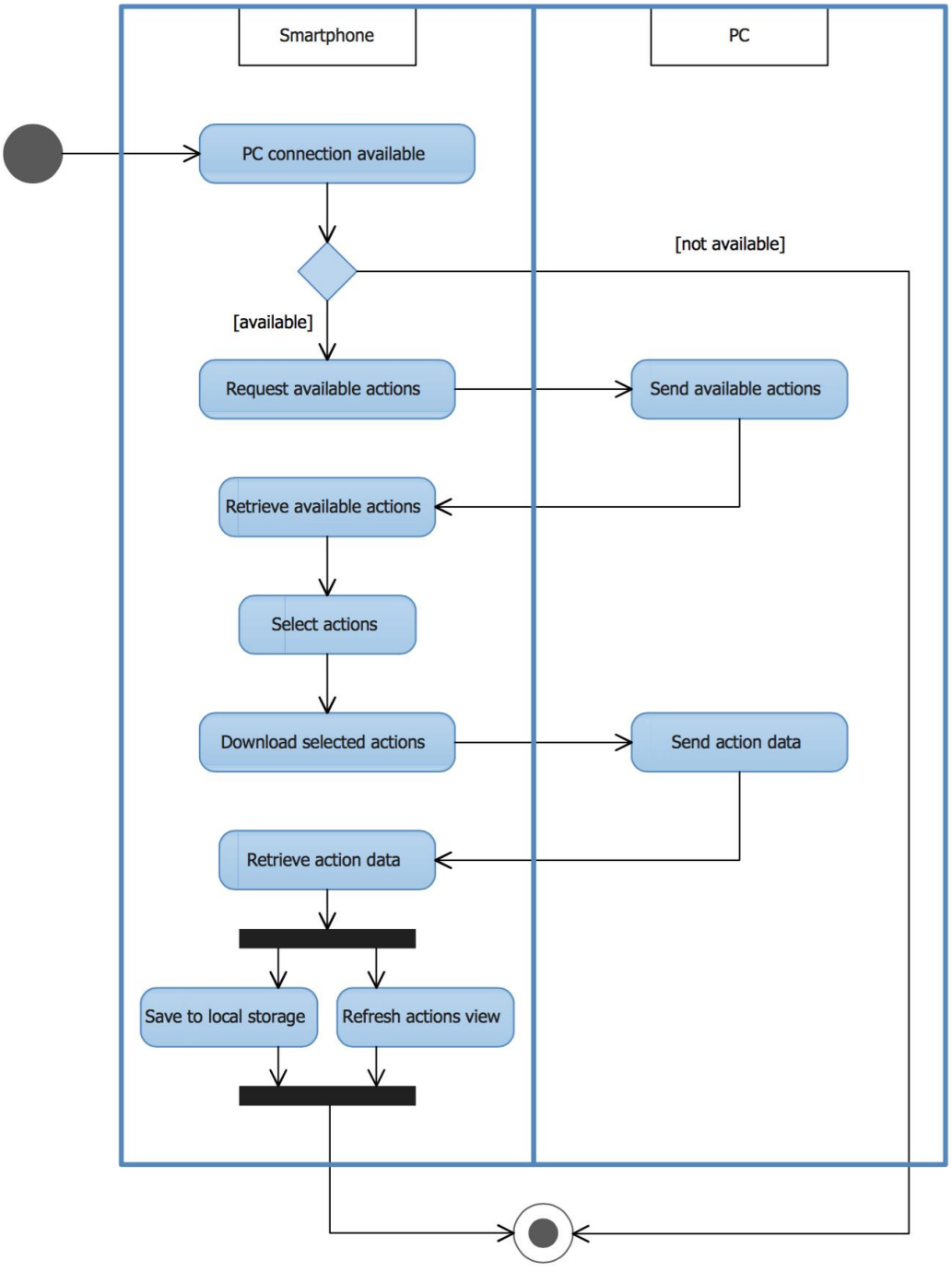
### 2.7.5 Scenarios for non-standard uses (bad cases or work around cases)

Actually, there is no non-standard use.

### 2.7.6 GUIs for the non-standard uses

Actually, there is no non-standard use.

### 2.7.7 Workflow



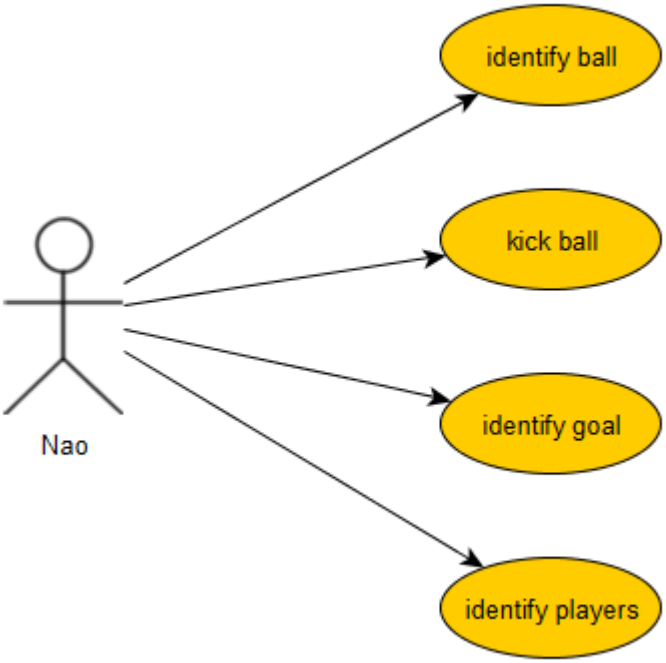
### 2.7.8 Open Points

Actually, there are no open points.

# 3 Functional Requirements – Soccer

Muss noch ausgearbeitet  werden

## 3.1 Use Case Diagrams



## 3.2 Identify Ball

### 3.2.1 Characteristic Information

### 3.2.2 GUI to call the Use Case

### 3.2.3 Scenarios for the standard use (good case)

### 3.2.4 GUIs for the standard use

### 3.2.5 Scenarios for the non-standard use (bad cases and work around cases)

### 3.2.6 GUIs for the non-standard use

### 3.2.7 Workflow

### 3.2.8 Open Points

## 3.3 Kick Ball

### 3.3.1 Characteristic Information

### 3.3.2 GUI to call the Use Case

### 3.3.3 Scenarios for the standard use (good case)

### 3.3.4 GUIs for the standard use

### 3.3.5 Scenarios for the non-standard use (bad cases and work around cases)

### 3.3.6 GUIs for the non-standard use

### 3.3.7 Workflow

### 3.3.8 Open Points

## 3.4 Identify Goal

### 3.4.1 Characteristic Information

### 3.4.2 GUI to call the Use Case

### 3.4.3 Scenarios for the standard use (good case)

### 3.4.4 GUIs for the standard use

### 3.4.5 Scenarios for the non-standard use (bad cases and work around cases)

### 3.4.6 GUIs for the non-standard use

### 3.4.7 Workflow

### 3.4.8 Open Points

## 3.5 Identify Players

### 3.5.1 Characteristic Information

### 3.5.2 GUI to call the Use Case

### 3.5.3 Scenarios for the standard use (good case)

### 3.5.4 GUIs for the standard use

### 3.5.5 Scenarios for the non-standard use (bad cases and work around cases)

### 3.5.6 GUIs for the non-standard use

### 3.5.7 Workflow

### 3.5.8 Open Points

# 4 Non-functional Requirements - APP

| ID: | NFR_001 |
|---|---|
| Name: | Upload speed |
| Type: | EFFIC |
| Description: | The App send the data of the Action to the NAO, this should be done under 5seconds. |
| Assigned use cases: | Send Actions to NAO |

| ID: | NFR_002 |
|---|---|
| Name: | Navigation reaction |
| Type: | EFFIC |
| Description: | In the App, there is a function called Navigation, where the user can control the NAO. If the user pull the control ball, the NAO react under 500ms. |
| Assigned use cases: | Control the movement |

| ID: | NFR_003 |
|---|---|
| Name: | Easy Handling |
| Type: | USE |
| Description: | It's important that the App is easy to handle. Everyone, also laymen of programming, can control the NAO with this Application. |
| Assigned use cases: | Send Actions to NAO |

| ID: | NFR_004 |
|---|---|
| Name: | Compatibility |
| Type: | USE |
| Description: | The user can easily add new actions, no matter if they are from the server or from the pc which the user made on his/her own. |
| Assigned use cases: | Downlaod PC, Download Server |

# 5 Non-functional Requirements – Soccer

Muss noch ausgearbeitet werden

| | |
|---|---|
| ID: | NFR_005 |
| Name: | |
| Type: | |
| Description: | |
| Assigned use cases: | <Use Case-ID> |

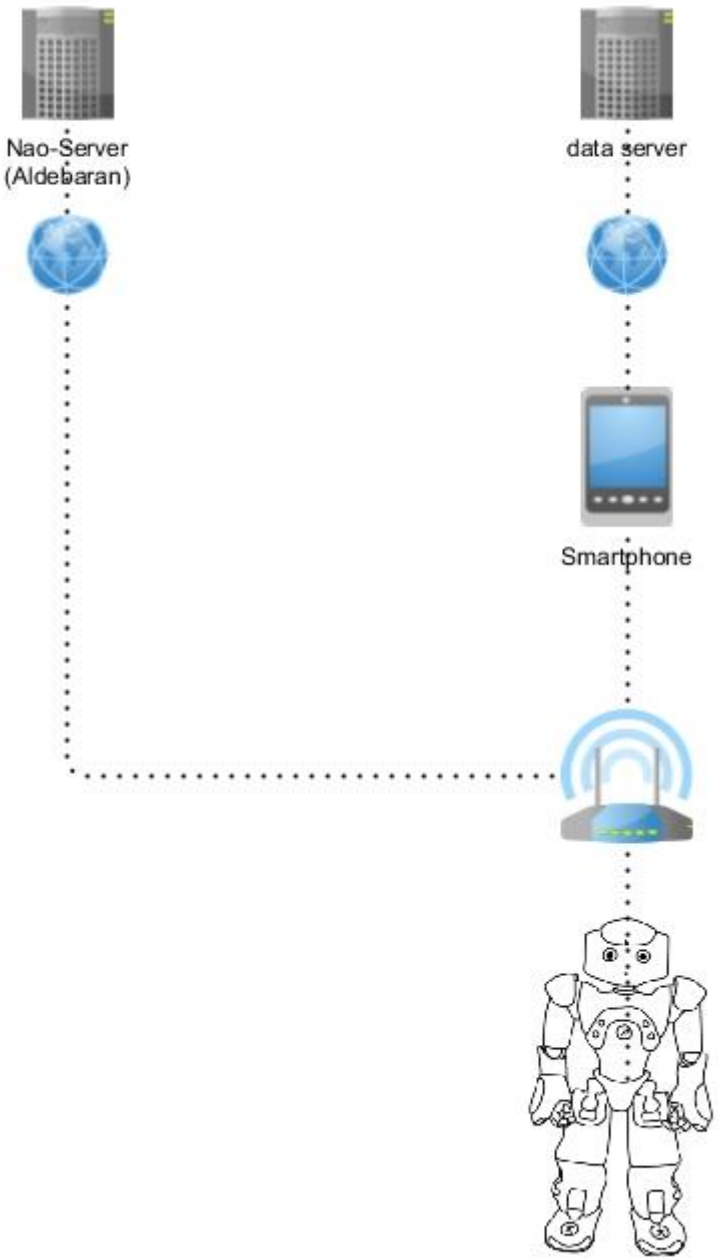
# 6 Quantity Structure

Wurde noch nicht ausgearbeitet

# 7 System Architecture and Interfaces

Aldebaran provides some basic actions for the robots to work. The project also contains developing a server which provides some actions that the robot should do.

If internet is available the robot connects with the NAO-Server.
The smartphone of the user is connected with the data server when it has an internet connection. Then the user is able to download actions. These actions can be sent to the NAO if it's in the same network as the mobile phone.



# 8 Acceptance Criteria

Muss noch ausgearbeitet werden

## 8.1 <AC_001> - <Name of Acceptance Criterion>

| Test Step | Expected Behaviour | |
|---|---|---|
| | | |
| | | |
| | | |