Service robots' feasibility in the hotel industry: A case study of Hotel Presidentti

Linh Hahkio

2020 Laurea

Laurea University of Applied Sciences

Service robots' feasibility in the hotel industry: A case study of Hotel Presidentti

Linh Hahkio
Business Information Technology
Bachelor's Thesis
May, 2020

Laurea University of Applied Sciences **Abstract**
Degree Programme in Business Information Technology
Bachelor of Business Administration

Linh Hahkio

Service robots’ feasibility in the hotel industry: A case study of Hotel Presidentti

Year 2020 Pages 39

Service robots have become a trend in the hotel industry in the past few years. This thesis project investigates the technology and value a service robot might bring to a case hotel Original Sokos Hotel Presidentti. Based on the results, general conclusions are drawn for the hotel industry.

A pilot test using a humanoid robot, Pepper, was held in the Original Sokos Hotel Presidentti in 2019. The robot was installed with a reception application developed to show what a service robot can do in a hotel. The test aimed to capture customers’ opinions on using the robot through their expressions and behaviours.

The conclusion of the thesis is based on the data gained from the test. Overall, there are jobs in the hotel industry that can be done and assisted by service robots. Customers and employees of the hotel both show interest in the use of service robots in the hotel industry. The main value service robots can bring to the hotel industry comes from its entertaining features; however, more values are expected to be added as customers get used to using the robots for different purposes. Regarding service robots’ technology, there are limitations in speech recognition in noisy environments which aggravates user experience. There are also accidents coming from unexpected user behaviours that service robots cannot handle. Therefore, service robots are not yet ready to work fully unsupervised.

The results suggested new areas of research. There was not enough data to address the question of whether the customers’ excitement in using service robots is long-term or short-term. However, as the thesis was done from the perspective that there are limitations in time and resources, the thesis can be considered a success. The project found that it would be feasible to use service robots semi-automatically in the hotel industry and this research works well as a foundation for further research on the same topic.

Keywords: Service robot, Pepper robot, hotel industry, web-scraping, low-code flatform

Table of Contents

1 Introduction

Robots are machines developed to assist and substitute human and human action. Robots have been used in industrial environment to increase productivity and to manipulate the tasks in dangerous environment. Due to the development of robotics in cognition, manipulation and interaction, the use of robots has expanded from industrial use to service and entertainment usage.

Hotel industry including tasks that potentially be automated by robotics. Recognizing the increasing trend of using robotics and potential value it might bring to the business, Sokos Original Hotel Presidentti has initiate a project to studying the possibility of service robot in the industry by carrying out a pilot test using Pepper robot with receptionist application. Customer behaviours, emotions and reactions are observed during the two-week period test. Data is then analysed to make conclusions on our research question: Whether using service robot in Presidentti hotel is technically possible and what values it might attribute.

Overall, hotel business earn money by providing temporary, usually short-term basis accommodations for guests. The price range and service provided varies between hotels. Competence of each hotel depends heavily on its pricing model, marketing strategy and range of service it provides. Hotel industry is a section of the service industry, which covers not only hotels but also other forms of overnight place rental such as motels, hostels, inns and guesthouses. Hotel industry operating environment is therefore closely related to the travel and tourism industry (Revfine n.d.).

Operating within the hotel industry, Original Sokos Hotel Presidentti is one of the popular hotels in Kamppi, Helsinki. Due to the need of an international standard hotel for guests, which would come for the Olympic game in Moscow in 1980, Presidentti hotel was opened by the president Urho Kekkonen and prime minister Mauno Koivisto. It was originally called Ramada hotel. During 2016 and 2018, the hotel has undergone a major renovation and refurbishment. The new concept for the hotel was designed by Paolo Suhunen from Ivana Helsinki. The hotel design is influenced by Finland and Finnish nature. Presidentti is now one of the largest meeting and event hotels in Helsinki (Sokos hotel n.d.).

To gain advantage and stand out in the competition, companies need to be up to date with the latest trends and it is the same in the hotel industry. One of the on-going trends in the hotel industry is the hotel technology trend. Current trending innovations are smart hotels, service automation, facial recognition, robots, guest app, voice search, chatbot, artificial intelligence, big data, virtual reality and augmented reality (Revfine n.d.). Presidentti, as a member working in the hotel industry, wants to analyse the possibility of these technology trends in its own working environment: how it is perceived in the hotel setting and if it can be of any assistance in the hotel. This thesis project is a part of this big research, which is conducted specifically to research service robots. The project is held with an agreed mindset that there are limitations in service provided but it will give enough valid data to find out how the staff and different customers welcome service robots and what kind of interest it will generate. The focuses of this project are to examine service robots' technologies and their applicability and to investigate how service robots are welcomed and what are the benefits service robots might bring. The data is gained from the case hotel and from that general conclusions are drawn for the hotel industry.

Before launching the project, the hotel needs to choose the robot model and decide what functions it performs. In order to choose which robot to test out, the hotel research team has analysed a few robot models including Pepper, Savioke, Starship and robots from Pulurobotics oy. They are different in nature. Some perform logistic functions while others focus on interaction and communication. In the end, the hotel decided to use Pepper, a humanoid robot, as it is a well-known product and it is in the genre of mature technology. Pepper as a robot with communication capability, it is the easiest approach now for the hotel and it has the most touchpoints with end-clients. Also, the fact that Laurea has a Pepper robot with an academic licence that makes the coding easier was a reason for choosing it. Regarding the tasks for Pepper robot, since the hotel industry is a large area, the research team narrowed it down to reception tasks. The reason is that the reception section well presents the environment of the hotel which heavily deals with customers. It is also the first stage and usually the first-place customers go through in the hotel. Thus, it maximizes the exposal of the robot to the target users during testing. Moreover, Pepper robot is a social robot and its main core is about engaging and interacting with humans, so the reception section suits well with the nature of the robot.

Having all the requirements decided, the hotel then participates with Laurea university of applied sciences to develop the reception application for Pepper robot. The project team includes an application designer and a developer. The team also works closely with the representative from Presidentti hotel and Laurea to make sure the project is on track. The designer is responsible for putting together necessary data, creating the flows of application, user experience, user interface based on requirements given by the hotel. I work as the developer role, developing the application based on designer work (description, wireframe, prototype). The description of the application and how it is implemented is discussed in the section.

2 Theoretical Background

2.1 Service robot

Service robots, according to The International Organization for Standardization (IFR), performs tasks under partial or fully autonomous state. Functions of service robots are meaningful to human’s being and usually more cognitive compared to industrial automation tasks.

Service robots vary in form and function. There are service robots which focus on interacting with people while others involve tasks which operate in difficult and dangerous settings such as in space or laboratory. Service robots are developed and deployed in many industries and applications. Some main highlighted types of service robots are agriculture robots, construction robots, customer service robots, defence robots, demolitions robots, exoskeleton robots, field robots, humanoid robots, industrial cleaning robots, inspection robots, logistics robots and medical robots (Robotic Industries Association n.d.).

The existence of a wide range of service robots is due to the demand of automation outside of factory settings. The market of service robots has risen significantly in the past few years and the trend is expected to continue further. According to the International Federation of Robotics (2019), the sale of service robots for professional use has reached 9.2 billion U.S dollars in 2018, 32% up from the previous year (Figure 1). Logistic robots, medical robots and field robots contributed most to this sale. At the same time, sale of exoskeleton is predicted with a high potential growth. The rising trend is recorded also in personal use service robots with 3.7 billion U.S dollars in sales, up about 15% compared to 2017 (Figure 2). Main contributors in personal use service robots are vacuuming and floor cleaning robots. Future forecast shows that sales of service robots in general will positively reach a total of 37 billion U.S dollars through 2021, indicating a growth rate of 21%. The International Federation of Robotics (2018), in its report also demonstrates that approximately 700 service robot manufacturers identified with America and Europe are 2 regions taking the most market share about 45% each (Figure 3).

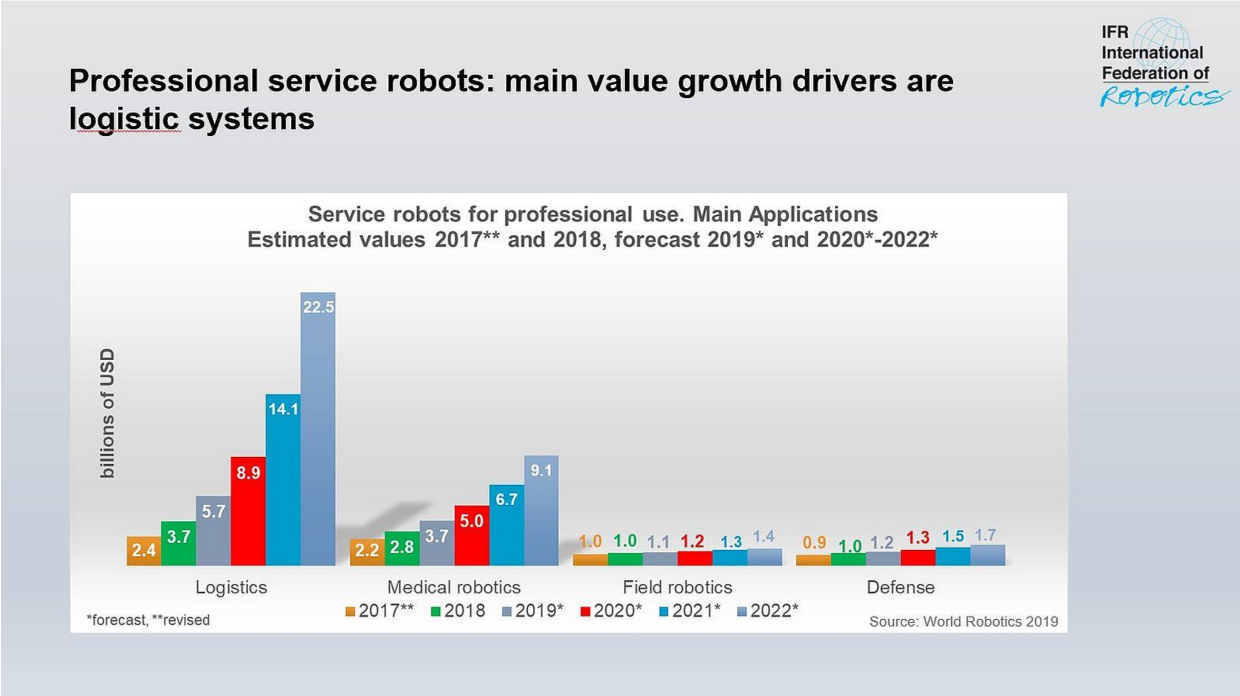


Figure 1 Value of professional service robots by main applications during 2017-2022 (International Federation of Robotics 2019)

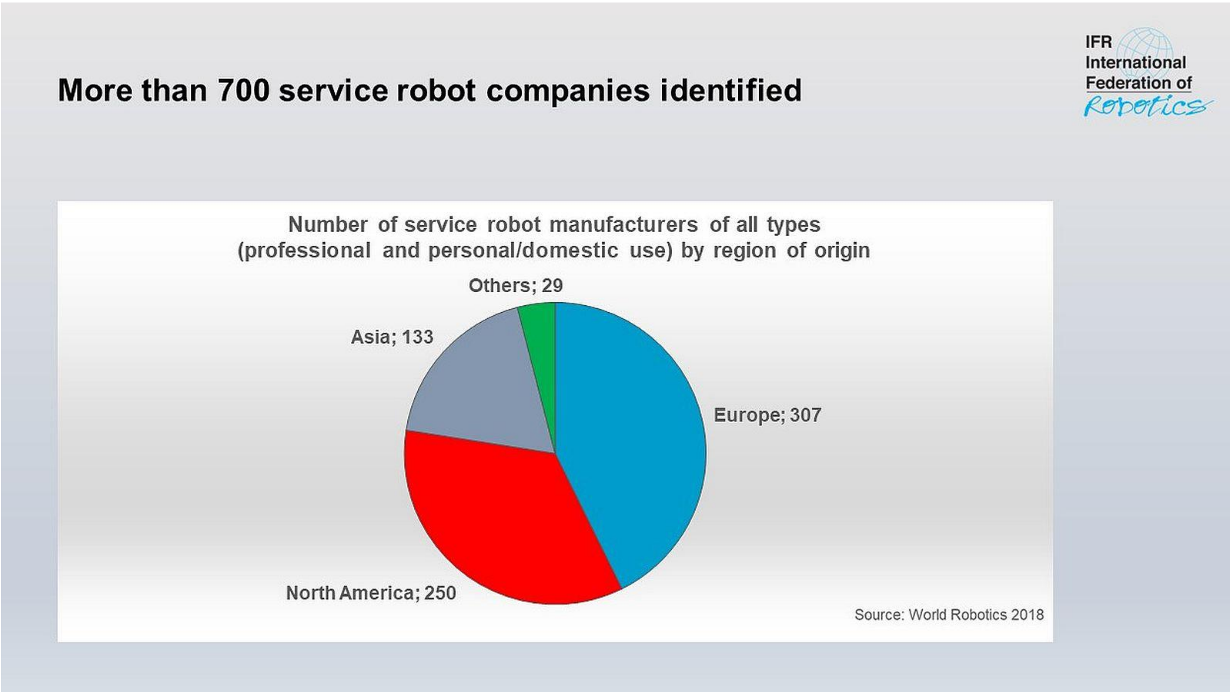


Figure 2 Number of service robot manufacturer by region of origin (International Federation of Robotics 2018)

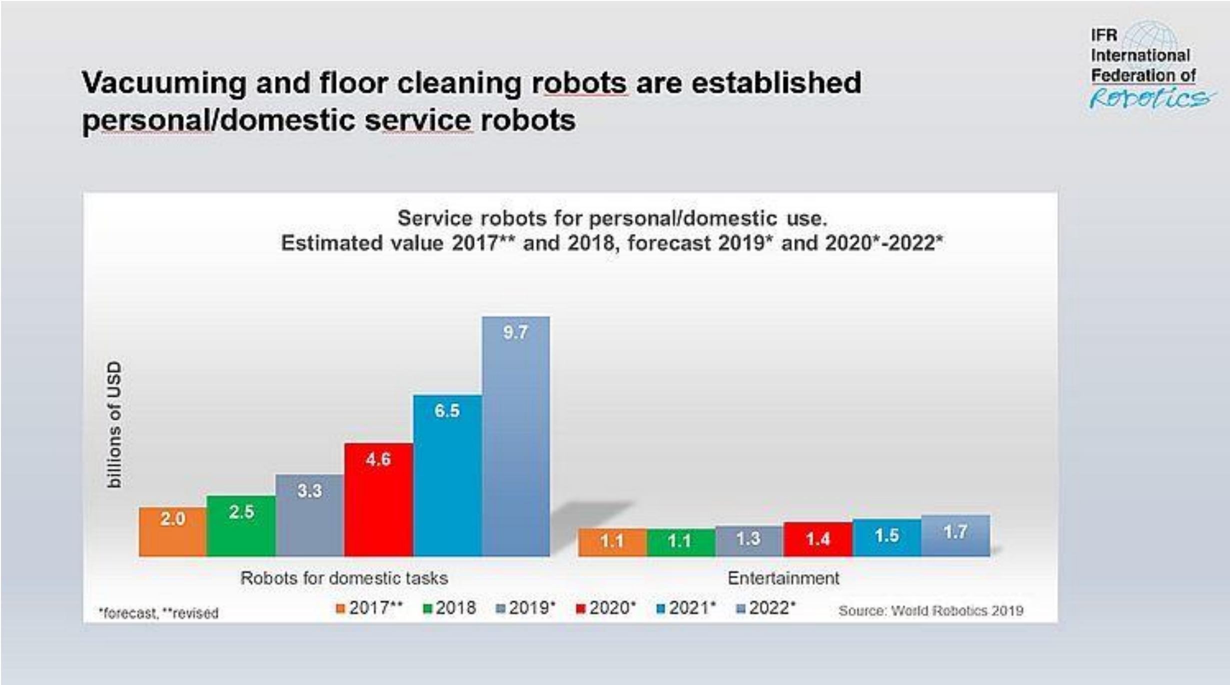


Figure 3 Value of personal service robots by main applications during 2017-2022 (International Federation of Robotics 2018)

Humanoid robots, which Pepper is an example, is a relatively new field of professional service robots. A humanoid robot takes the similar form of a human and it is developed to imitate a human's gesture, behaviour and interaction. It has a wide range of applications and can be operated in different settings. Many tasks can be automated by humanoid robots from laborious, routine and dangerous tasks to tasks that require interacting with humans like guiding, accompanying and potentially being used in the medical field as host to grow organs for transplanting (Robotics Industries Association n.d.). Due to a variety of functions and fast adapting to new technology for advancement, a rapid growth for humanoid robots is reasonably expected. A total of 3.9 billion U.S dollars is predicted for humanoid robot’s market value in 2023 with compound annual rate as 52.1% during 2017-2023 period according to Robotics Industries Association (no date). The market for humanoid robots is likely to expand more along with the enhancement of technology.

2.2 Pepper robot

2.2.1 SoftBank Robotics

Pepper is a humanoid robot made by SoftBank Robotics, one of pioneers in manufacturing humanoid robots. SoftBank Robotics is a French company, originally named Aldebaran. In 2015, it was acquired by Japanese organisation SoftBank Group and changed its name to SoftBank Robotics later in 2016 (SoftBank Robotics n.d.).

Pepper was first introduced in 2014 in a conference. It is known as the first emotional robot. Before Pepper, SoftBank Robotics has 2 other humanoid robots: Nao, the company's first robot which was launched in 2006 and Romeo which was launched in 2009 (Figure 4). While Pepper and Nao are intensively advertised and have already been deployed in many industries, projects or research, Romeo is mainly used in Romeo projects for researching purposes.

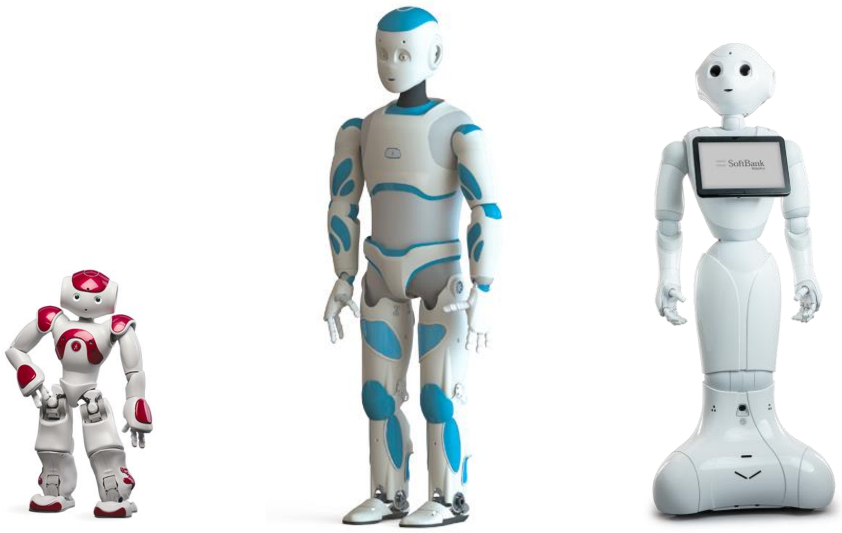


Figure 4 Nao, Romeo and Pepper robot
(Softbank Robotics n.d.)

In the next sub chapter 2.3.2, the technical aspects of Pepper that are put into consideration when developing and testing the application are presented. The features which enable Pepper ability are shown briefly while features more directly influence the project are explained in more detail.

2.2.2 Pepper technical overview

2.2.2.1 Dimensions

As a humanoid robot Pepper mimics human looks. However, Pepper is not a bipedal robot, it comes with wheels for a more stable stand and movement. Pepper is 1210mm tall, approximately the height of a 7 to 10-year-old kid. The height is desired to prevent intimidation. Figure 5 and figure 6 below shows Pepper’s dimensions.

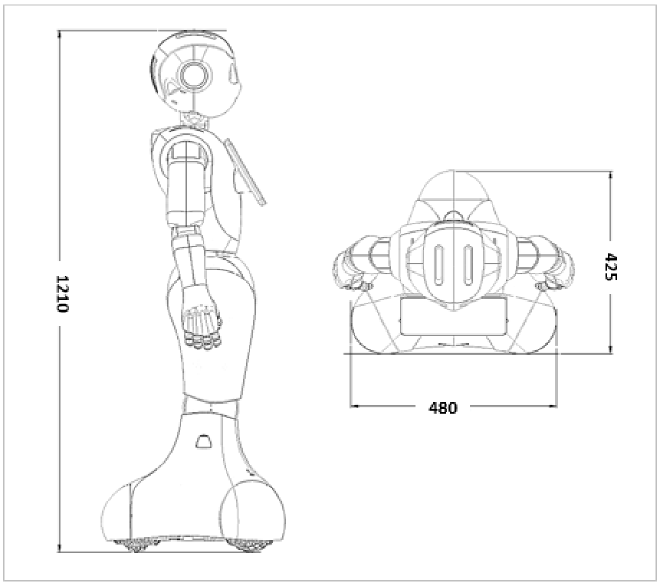


Figure 5 Dimensions of Pepper in initial standing pose
(Softbank Robotics n.d.)

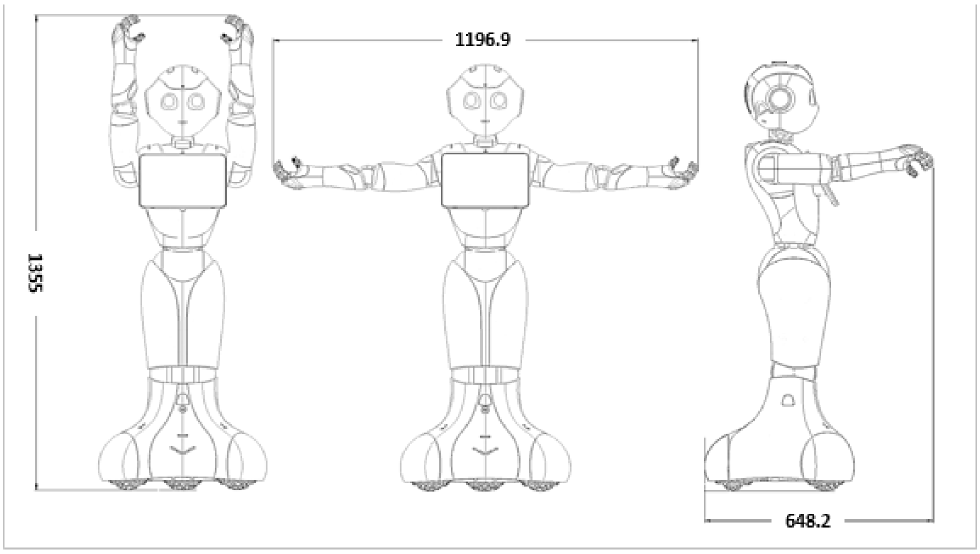


Figure 6 Max dimensions of Pepper with arms spread out
(Softbank Robotics n.d.)

2.2.2.2 Hardware

As stated in Pepper documentation, Pepper is equipped with:

- 26 motors enable Pepper movement, gesture and animation (Figure 7).

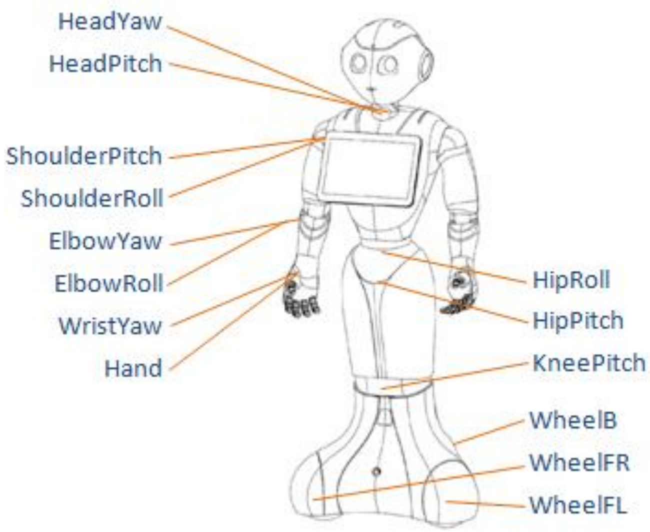


Figure 7 Pepper's motors map
(Softbank Robotics n.d.)

- 2D camera (Figure 8) and 1 3D camera (Figure 9) capture surroundings for perceptual module and navigation.

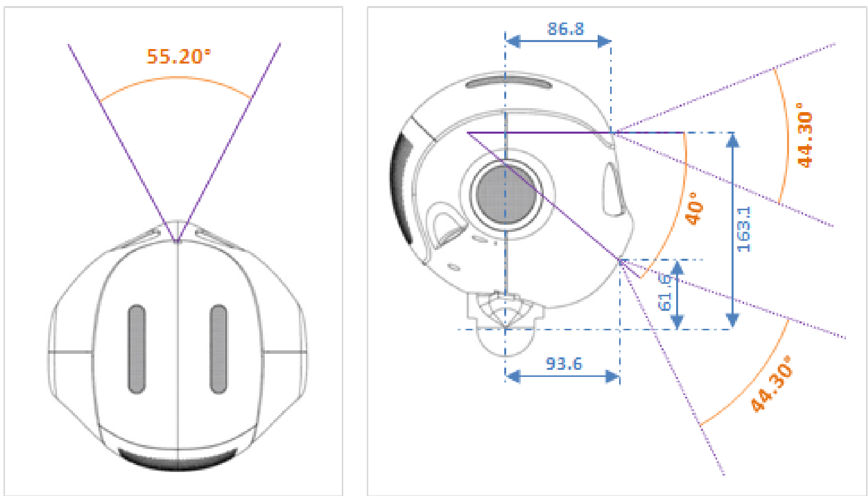


Figure 8 Pepper's 2D camera location and range
(Softbank Robotics n.d.)

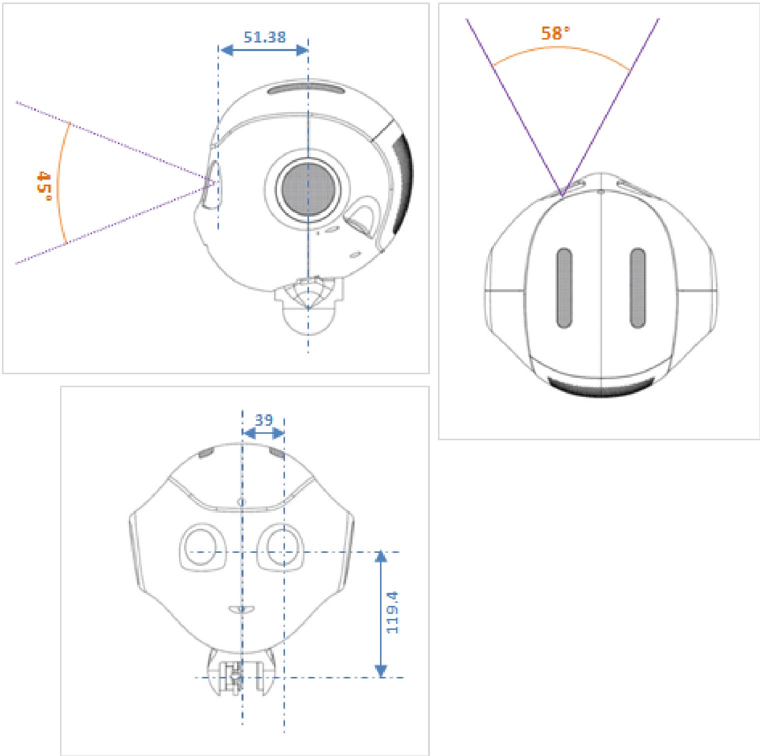


Figure 9 Pepper's 3D camera location and range
(Softbank Robotics n.d.)

- An inertia unit provides accelerometers and geometers which are used to measure the base speed and attitude of movement.
- 40 LED lights (8 on each eye, 10 on each ear and four on each shoulder) gives notifications e.g. blue eyes when Pepper is listening.
- 6 laser line generators scan surroundings to support movement.
- infra-red sensors sense surroundings to support movement.
- 2 ultrasonic sensors to measure distance to obstacles.
- 30 Magnetic Rotary Encoders support the motor rotation measurement to reach desired position.
- 2 contact buttons (power button and stop button), 3 tactile sensors (head and in 2 hands), 3 bumpers (bottom of stand).
- A 795 Wh battery – approximately 12 hours in use
- 4 microphones on the head and 2 loudspeakers in the ears. (Figure 10)

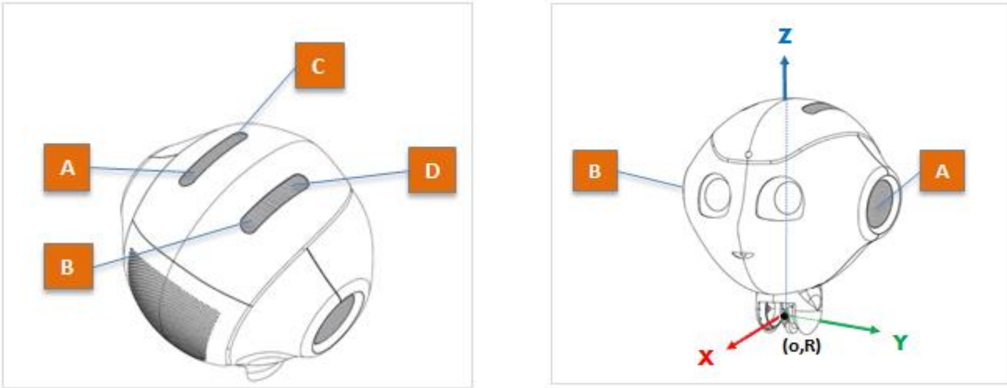


Figure 10 Pepper's loudspeakers location (Softbank Robotics n.d.)

- An Intel Atom E3845 processor

- A tablet in front of the chest. It does not have an operating system; It can connect to WIFI. Its dimension is 246 x 175 x 14.5mm and its resolution is 1280 x 800. It sup-ports multiple video formats (e.g. wmv and mp4) and audio format (e.g. wma, flac).

The information received by Pepper's hardware can be retrieved through API calls and later be used to build the reception application. The APIs and method to code with Pepper is discussed later in chapter. Dimensions of the robot, the position of microphone, loudspeakers are used to define where Pepper should be placed in the lobby (reception area) and the optimal surrounding distance for the robot to operate normally and safely.

2.2.2.3 Operating system and framework

All robots made by Softbank Robotics run Naoqi operating system (Naoqi OS), which is a program used to control the robots and manage all other applications on the robots. Naoqi OS is a GNU/Linux distribution developed based on Gentoo by the company to best fit its robot models (Softbank Robotics n.d.).

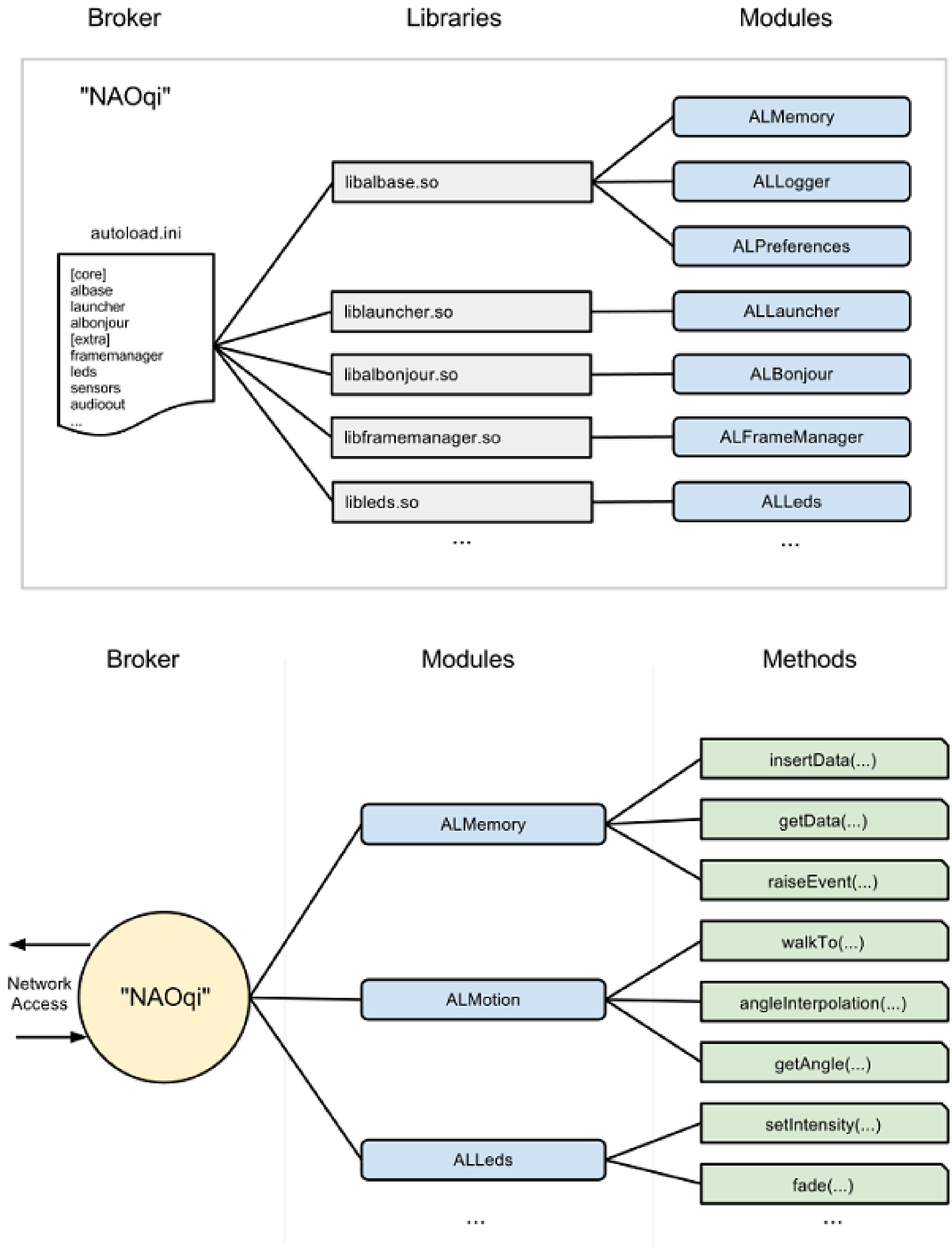


Figure 11 Naoqi process model
(Softbank Robotics n.d.)

Main software running under Naoqi OS is Naoqi. It automatically starts when robots launch, before any other applications can operate. Naoqi can also be installed locally on computers for working with stimulated robots.

As Figure 11 describes, Naoqi provides a list of modules with corresponding default methods and allows those methods to be called from outside the process. When Naoqi starts, a list of libraries is loaded and modules which are classes of those libraries are automatically instantiated. By creating a proxy, an object of the module, all the functions listed in the module can be used. Figure 12 is an example of using an ALProxy object to create a proxy to Text-to-speech module which makes the robot say a given string.

```
from naoqi import ALProxy
tts = ALProxy("ALTextToSpeech", "<IP of your robot>", 9559)
tts.say("Hello, world!")
```

Figure 12 Example code of creating a proxy to a module
(Softbank Robotics n.d.)

The programming framework used to program robots with Naoqi is the qi framework. With this framework, the available methods and syntax to call them are the same for both local or remote IP connected robots and regards to the programming language used. The framework is cross-platform so it can be used in computers with either Windows, Linux or Mac OS. Several languages are supported by the framework. The list is shown in figure 13.

Programming Languages	Bindings running on		Choregraphe support	
	Computer	Robot	Build Apps	Edit code
Python	✔	✔	✔	✔
C++	✔	✔	⊘	⊘
Java	✔	⊘	⊘	⊘
JavaScript	✔	✔	✔	⊘
ROS	✔	⊘	⊘	⊘

✔	OK
⊘	Not available

Figure 13 Programming languages supported by qi framework
(Softbank Robotics n.d.)

By installing an appropriate software development kit (SDK) which contains all necessary tools such as compiler, debugger and framework, an application for Softbank Robotics' robots can be programmed in the language of your choice. In this project, Choregraphe software is used to create the reception application. The software is discussed more in detail in section 4.3.1.

2.2.2.4 Speech recognition engine

Speech recognition engine is the core that empowers Pepper's ability to under-stand human language. The engine depends on technologies provided by Nuance (Softbank Robotics n.d.). There are 26 languages supported by the engine, for in-stance English, Finnish and Swedish. The full list of possible languages is published on Softbanks Robotic website.

To make Pepper understand speech, in Pepper's application, ALSpeechRecognition or ALDialog module is used. Using ALSpeechRecognition, Pepper can recognize words or phrases. First a list of words and phrases that should be recognized is sent to the module. Once Pepper successfully captures a human input through microphones, what the robot hears will be compared to the provided list to find out the word/phrase in the list with the highest confidence number. The confidence number indicates the possibility that the word/phrase is what the user said. The best match word/phrase is stored in WordRecognized key and WordRecognizedAndGrammar key so the word/phrase can be referred to later using the key's name (Softbank Robotics n.d.). In the meanwhile, ALDialog module utilizes speech recognition technique as well as facilitates conversation making. The content of potential conversation that will be fed to ALDialog is programmed using QiChat rules, sub-rules and syntax.

User rule

```
u: (Hello Nao how are you today) Hello human, I am fine thank you and you?
```

Proposal rule

```
proposal: Have you seen that guy on the TV yesterday?
   u1: (yes) He was crazy, no?
   u1: (no) Really, I need to tell you.
```

Option word

```
u:([hi hello]) [hello hi] human
```

Execution

```
> hello
hello human
> hello
hi human
> hi
hello human
```

Figure 14 Qichat example: user rule, proposal rule and optional word (Softbank Robotics n.d.)

Qichat rules and sub-rules help manage the flows of conversation and make the conversation more context based. There are 2 types of rules: user rule where a human input links to a specific robot output and proposal rule where robot output is given without a user input. Figure 14 shows an example of Qichat rule. QiChat reduces the effort in adding similar word/phrases in user input and robot responses by introducing syntax such as concept (group of word and phrase that works similarly), choices (The robot reacts as one of the word in the list matches user input) and option word (The robot reacts if user phrase contains the word or not).

3 Methodology

3.1 Pilot study

Pilot project or study is a trial or an initial small study prior to a larger scale research. It is used to evaluate the feasibility, reliability or validation of an aspect of a research e.g. refining research questions, methods and procedures (Thabane, Ma & Chu). The information gathered from a pilot study is then used to assess if it is worth to hold a further research on the topic as well as works as a guideline for related researches to prevent potentially fatal issues on large-scale.

According to Thebane, Ma and Chu (2010), there are several motivations for holding a pilot study, which can be classified based on its subjective. The reasons for carrying out a pilot project can be categorized in 5 groups as follow:

Process: Feasibility and adequacy of procedure and method of main research are analysed step by step, for example, testing the effectiveness of a data analyse technique.

- Resources: This focuses on time and finance assessment e.g. measuring the time to fill out a survey.
- Management: This includes the measurement of human resource availability and optimization of personnel and data usage and management
- Scientific: This deals with the potential outcome such as collecting and estimating the variability of preliminary data.
- Training and funding: Research topic's feasibility or research team's competence is measured to convince investors and other stakeholders that the research is worth funding and support. Throughout the pilot project, the researchers get familiar with elements of the main research e.g. process and method.

Pilot study is a useful guideline for a larger-scale study, however, the result from pilot study should be analysed and implemented carefully. As Van Teijlingen & Hundley (2001) stated in their article, the limitations of pilot study are misinterpreted conclusion, contamination data/process and funding issues. Data collected from pilot projects is usually on a small basis and without a statistical foundation. Furthermore, some problems and issues are only disclosed in large-scale research. As a result, pilot study results might represent some features of responses from main research but does not guarantee it. Therefore, conclusions and assumptions drawn from pilot study can be inaccurate. Moreover, the main research can be contaminated by using erroneous data from pilot study or by reusing test samples/participants. The problem is coming from the potential modification for the main research (e.g. to the research tool) after the pilot test, which makes the data from pilot study no longer valid. By having the same sample or participants from pilot study to main study, the re-search is exposed to overlearn risk. The experienced participants are likely to react differently from their first time. Lastly, a failed pilot test is problematic, especially when it has already consumed a large amount of finance. Investors may withdraw even if changes are made to adapt to the problem. Pilot study, there-fore, increases the likelihood of success of a following study, however, does not as-sure it (Van Teijlingen & Hundley 2001).

3.2 Observation

- Observation is a research technique in which data is gathered through watching, recording and analysing events. Observation is often used for case study. In this project, observation is implemented to capture the general impression and behaviours of all guests and the situation of background when guests try out the robot.

- There are 4 important points that need to be considered when using observation technique (Dawson, 2009). Following is how they are executed in this project.

- Arrangements: The robot is located in the lobby of the hotel, with the reception desks to the left and elevators to the right. It is under testing for 8 days. The experiment started from 8.4.2019. There is an instruction paper on the desk next to the robot and attached on the column next to the robot to inform guests that they are being observed.

- Recording the observation: The data is recorded real time by taking note.

- Participation: The observer does not actively participate during the test. However, assistance is given to tester if needed e.g. Inappropriate behaviours or accidents happens

- The Hawthorne effect refers to the influence of observers to the testing situation. The testers might react differently because they are aware of being watched. To avoid this situation, we try to lessen our existence during the test and come to assist if the testers seem to be struggling or have an extreme impression such as over-excited.

3.3 Software development life circle (SDLC)

Software development life cycle (SDLC) including all stages in developing a software (Daw-son 2009). Basic stages of all SDLC models include requirements, design, build, test and implementation. Some popular SDLC models are Waterfall, Scum and Agile. In order to create an application to test out the robot in this project, Waterfall model (Figure 15) is chosen. Waterfall model is best matched with our small project. It is simple, easy and fast to implement so it suits a small-scale project like this one. In addition, it works best with projects with fixed requirements as it is very difficult and time consuming with the Waterfall model to come back to each stage for changes. For large, complicated projects with flexible requirements, they are better to use Scum or Agile models.

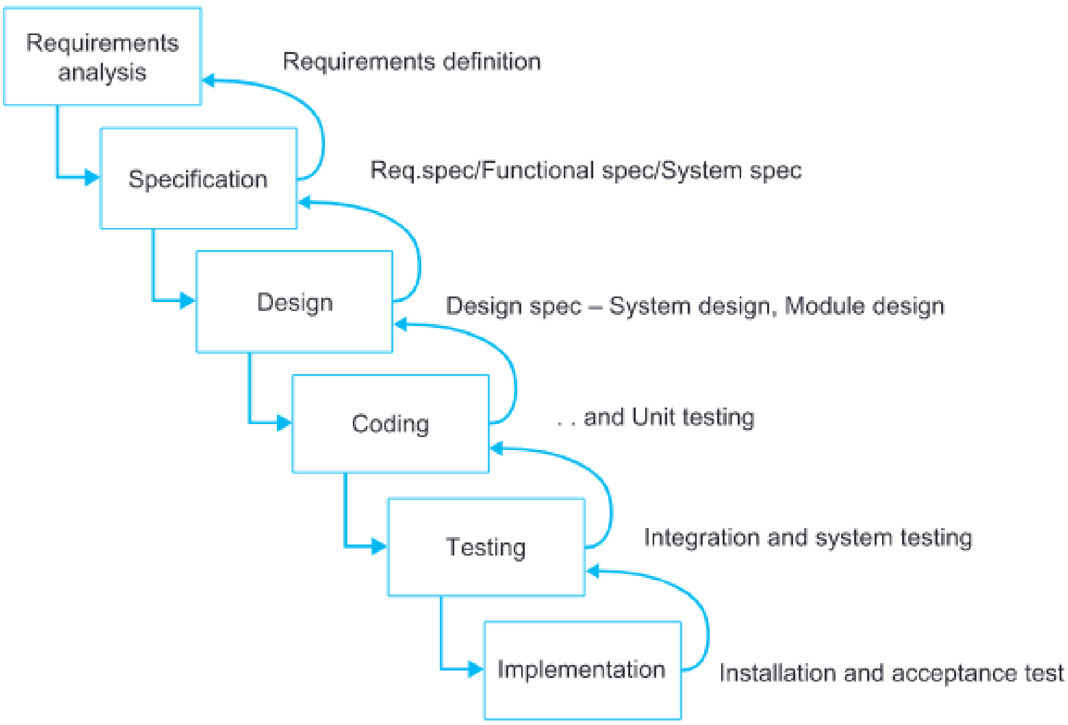


Figure 15 Waterfall model
(Dawson 2009)

As the main purpose of this project is not developing an application, the structure of the research will not follow the phases of the software development model. Instead, the upcoming sections will focus on explaining the main features of the application, what technologies are used and how they are utilized to enable those features. Through mentioned information, the technical aspect of Pepper robot is presented and analysed.

4 Reception Application

4.1 Application requirements

To analyse the feasibility of Pepper robot, a reception application is developed for the robot to give testing targets, guests of the hotel, the brief idea of what a service robot can do. The hotel defined the desired functions of the Pepper based on its main capability which is to interact with humans through communication and with the assistance of its tablet.

The requirements for the application are:

- The application is in English or Finnish based on user choice.
- The robot can make conversation with guests regarding basic information of the hotel and answer QA questions (e.g. hotel introduction, robot information, greetings, location of main facilities). This information is prepared by the Presidentti hotel.
- The robot can show popular sights around Helsinki, this information is fetched from myhelsinki.fi page. Pepper should show the image of an attraction on the tablet and give details about the place through speech.
- The robot can show a list of 5 restaurants including Bistro Manu, BBQ house, Loiste, Frans and Amelie and Grill-it Helsinki. It provides information about the menu, address map and some restaurant's images with the help of the tablet. The information is gathered from the raflaamo.fi page.
- The robot can perform an entertaining function: a simple dance.
- The application should be done at least a week before piloting session so there is time for all team members, especially the representative from Presidentti hotel to test it out. Pre-testing is done to assure the application match requirements.

The designer then designs the application wireframe aligning with these requirements. The wireframe and its detailed description are used by the developer to create the application. The whole process takes roughly 2-3 months in total including the pre-test.

4.2 Technology

4.2.1 Low-code platform: Choregraphe

Low code development has been used only a few years ago. According to Marvin (2018), the concept, however, is not new. It was first mentioned in 2014 in a report of Forrester Research, a tech research and analysis company. The company defined the term “low-code platform” as a platform which is used to fast create a business app with minimal manual coding and prior cost of configuration, training and deployment (Marvin 2018). Some well-known examples of low-code platform are Quick Base, Zoho Creator and Microsoft PowerApps.

A low-code platform can be used by both low technical people and tradition developers. A low-code platform contains existing templates with multiple prebuild piece of codes, usually visualized as elements, forms and objects that can be connected to build an app. By having most common features and components pre-coded, an app does not need to be built from scratch and therefore resources are optimized.

Choregraphe is a desktop application, created by Softbank Robotics, consists of multiple platforms, one of which is a low-code platform created specify to develop applications for robots operating with Naoqi OS. Choregraphe provides tools to create and test robot applications, monitor and control the robots. Main panels of the software are: project panel (It lists all elements of the current project), Box libraries panel (It is a list of pre-made code boxes), Flow diagram panel (It is where the behaviour of the robot, the application is built-ed), Robot View (It is a 3D view of the robot and objects, people or zones from camera view), Dialog panel (It shows user speech that robot captured and its responses), robot application panels (It is a list of all applications installed in the connected robot) (Figure 16).The code boxes in Choregraphe is in Python language and it can be extended by adding new boxes with new block of code or by modifying existing boxes. Boxes in Choregraphe are categorised into 4 types: python box, flow diagram box, timeline box and dialog box. A python box contains a python script and it is often used to add new functions that aren't in the box library such as connecting to a third-party program. A flow diagram box has a flow diagram within it which contains many linked boxes. A timeline box is concerned with the movement of the robot. Finally, a dialog box is widely used in making a Pepper application as it enables the robot conversational skills. The syntax to call the functions in the code box are explained earlier in section 2.3.2.3. The list of available functions of the robot can be found in documentation published on the company’s website.

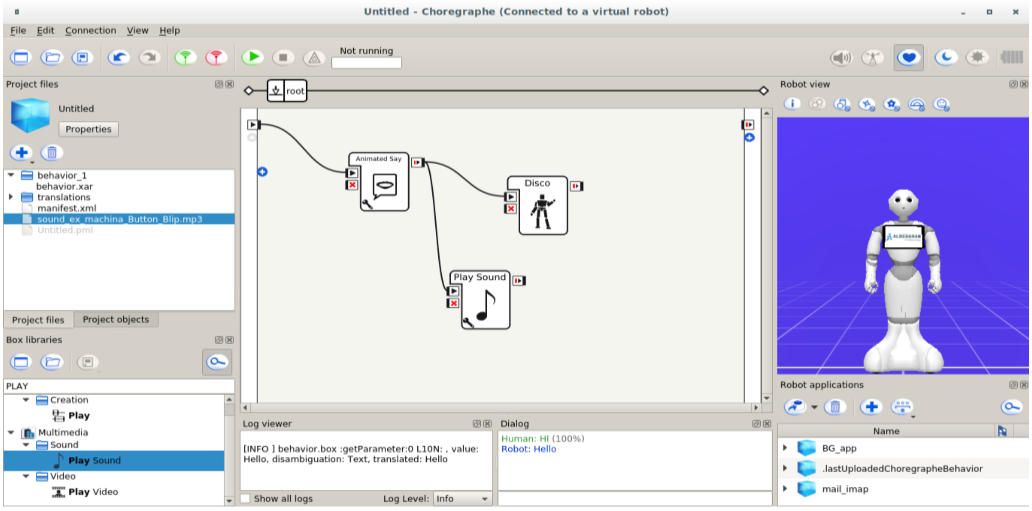


Figure 16 Choregraphe software interface

With the use of Choregraphe, it is easier and faster to make, test and manage our application. All stakeholders can understand the application better with this visualisation and be more involved in the process.

4.2.2 Web-scraping

Nowadays with the booming of Internet, website is one of the main sources for information. Usually, in order to fetch data from a website, a standard way is to use APIs. API, short for Application Programming Interface, works as intermediary for interconnection between da-ta and applications. However, there are many cases where API is not applicable. API is not provided by all website and there is no cohesive API for pulling data across multiple websites. Moreover, the data you want to retrieved might be uncommon or valuable/protected hence it is not included in the API or it is available with limited access and rate. In addition, the actual website is more regularly maintained than its API. Web-scraping exists as an alternative method for those situations.

Web scrapers extract data from the web in a systematic and autonomous way (except from your IP address). By using web-scaping, you can avoid the problems mentioned above. Nonetheless, it has its own constraints. The first concern is legal issues. According to Lawson (2015), the use of web-scraped data for publishing purposes might be illegal based on copyright law. Personal use of the retrieved data is normally accepted but otherwise it is recommended to check if the websites prohibit web scraping. Lawson also pointed out that, based on real court cases, facts such as company’s address and telephone listing can be republished while own opinions or reviews are likely not allowed. However, there is no solid law on the mat-ter. It is wiser to ask for permission beforehand. Second problem of web scraping arises from websites with poorly mark-up format. It makes web-scraping more difficult and time consuming.

Based on works of Lawson (2015) and Glez-Peña, Lourenço, López-Fernández, Reboiro-Jato & Fdez-Riverola (2014) the process of building a web-scraper includes 4 phases: website research and inspection, site access, HTML parsing and data extraction, data utilization. The implementation of these steps in our project will be presented in section 4.4 Development.

4.3 Development

This section explains the structure of the application development and web-scraping process in detail.

4.3.1 Application development structure

The main application is developed using Choregraphe software. The data needed to be shown on the tablet such as images is, however, visualized as an element of a website created by using HTML and Javascript SDK. Javascript SDK enables the connection and interaction between the website shown and main application. The structure of the application development is shown in figure 17.

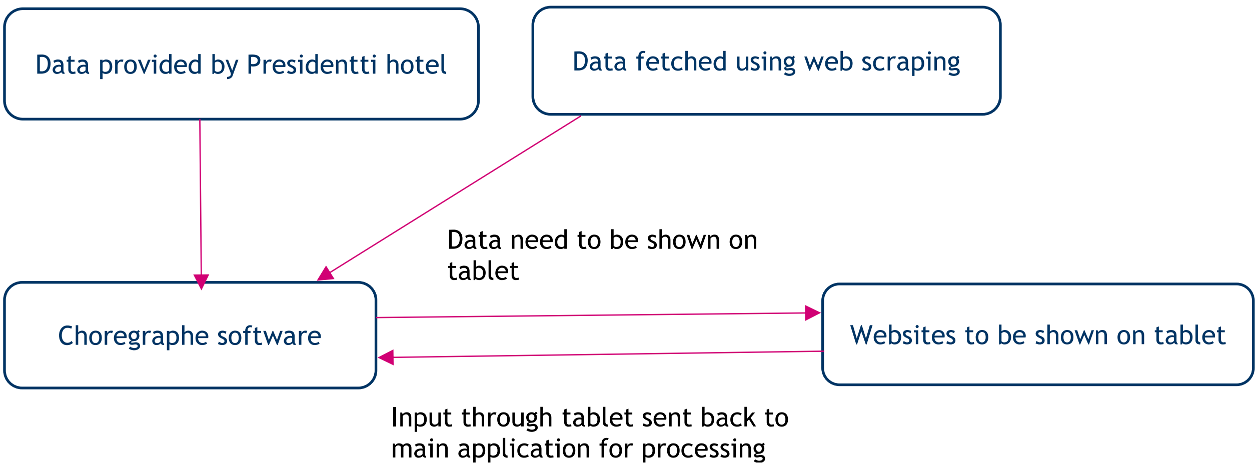


Figure 17 Structure of reception app development

The main steps included in building main application are:

- Create new boxes to pull the data from raflaamo.fi and myhelsinki.fi using RoboBrowser library.
- Create a box to show the website on tablet.
- Using a “Dialog” box to create conversation in both English and Finnish.

- Using a "Animated Say" box to make Pepper speak outside a conversion.
- Using a "Play sound" box to play a sound when something is clicked on the tablet.
- Using a "Subscribe to event" box to receive signal when an event is triggered. It is used to detect and receive information from the sites shown on the tablet when an element is clicked.

The most time-consuming task in this stage is data extraction using web-scraping. The others are easily done by using available boxes in Choregraphe. Next section 4.4.1 will, therefore, focus in representing the steps used to pull data with web-scraping.

4.3.2 Pulling data using web-scraping

As mentioned in the web-scraping introduction in section 4.3.2, the web-scraping process includes 4 phases: website investigation, website connection, HTML parsing and data extraction, data utilization. In this section, implementations of each phase will be discussed in more details.

4.3.2.1 Website inspection

First, the targeted websites are investigated. Main concerns for inspection are the HTML structure of the page, the HTML elements that contain interesting data, resources that are uploaded or downloaded. Website inspection can be done using a developer tool provided by most browsers. In this project I used the Firefox Developer tool. The tool can be accessed by right click then choose Inspect Element or Ctrl + Shift + I or F12 in Linux and Windows computer or Cmd + Opt + I for macOS.

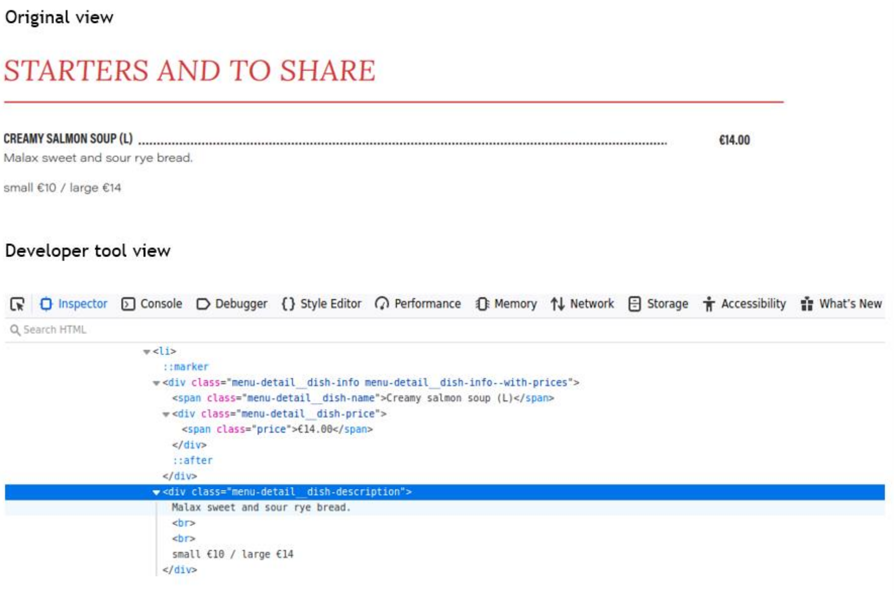


Figure 18 Website view and inspection view of same website

Figure 18 is an example of the HTML structure of the wanted data: the dishes' name is found under the "span" tag with class "menu-detail_dish-name". This information is needed to address the elements we want to fetch. After listing down all information we want and its HTML features, a connection to the targeted website is established through HTTP protocol. The setup is done using python with RoboBrowser library.

4.3.2.2 Library installation

SSH and virtual environment are used to install the library in the robot.

Pepper robot has python installed but RoboBrowser package is not included so it needs to be installed separately. The robot operating system is Linux native so python libraries can be set up through bash shell. In order to install a library on the robot, the connection to the robot is established first using SHH. SHH is an abbreviation for secure shell protocol. It was invented by Tatu Ylönen in 1995 and is widely used for secure file transfer and system administration. SSH protocol enables secure connection between client and server by encrypting all data including user authentication, commands, outputs and transferred files (SSH.COM n.d.). SHH protocol is shown in figure 19 below.

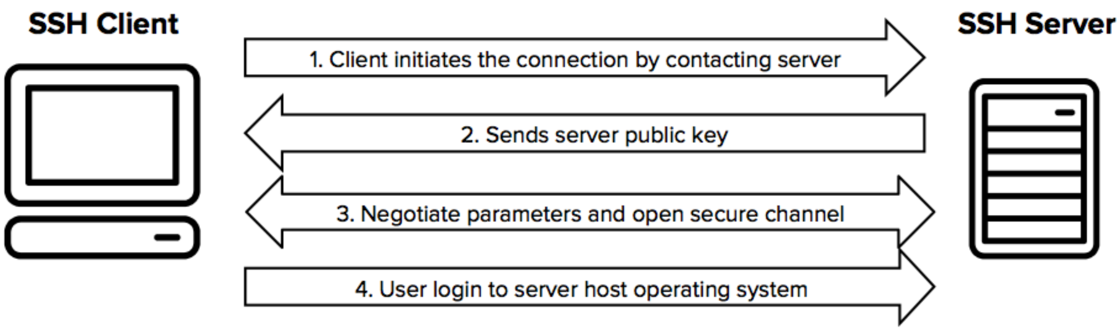


Figure 19 SSH protocol
(SSH.COM n.d.)

The remote access utilizing SSH protocol is done by using client software such as PuTTy or Tectia. Linux/ Unix base system does not need additional installation as it comes originally with SSH client and can be called directly via terminal command using syntax as figure 20 below shows.

```
$ ssh nao@192.168.8.100
```

Figure 20 Command to connect to a server using SSH

In the command "ssh" is the syntax to call ssh client, "nao" is the user account (nao is the default account for nao/Pepper robot) and after the @ sign is the IP address of the robot. The connection requires password authentication. Besides the user account "nao", as a Linux based system, Pepper also has an administration user "root". However, the usage is limited to shutdown action only. It leads to the problem of missing permission to install Python packages. The problem is solved by employing a virtual environment.

Python virtual environments' core is to create an isolated Python environment for Python projects. By default, every project shares the same directory where site packages or third-party libraries are stored (Real Python n.d.). In the long term, dependencies of some project might conflict with each other and for this reason the usage of virtual environments becomes important. Each project can have its own virtual environment containing needed site packages and it is disconcerted from any other projects. Tools to create virtual environments are, for example, virtualenv and venv. Once a virtual environment is set up, the libraries are installed using pip package manager which is already available in Pepper. Figure 21 shows the command to install RoboBrowser with pip.

```
$ pip install robobrowser
```

Figure 21 Command to install RoboBrowser using pip

After libraries are installed, for it to work, the virtual environment needs to be activated during application running time. Virtual environments can be activated within your application code using python.

4.3.2.3 Access site and extract data in the application

As all required packages are installed and the virtual environment is activated, they can be imported into your code in Choregraphe for use. RoboBrowser functions are listed in detail in RoboBrowser documentation which can be accessed online at https://robobrowser.readthedocs.io/en/latest/readme.html.

First, connection to a wanted website is done by creating a browser and using the browser to open the website. As the website opens, it is automatically parsed to HTML format. All the elements of the websites now are accessible via its attributions e.g. tag, id or class name.

```
# Activate virtual environment
activate_this_file = "path_to_project/lib/bin/activate_this.py"
execfile(activate_this_file, dict(__file__=activate_this_file))

import re
from robobrowser import RoboBrowser

#Set up browser
self.browser = RoboBrowser(history=True, parser = "html.parser")

#Access and parse url
url = "https://www.raflaamo.fi/en/helsinki/"
self.browser.open(url)

#Find element
header = self.browser.find("h1", class_="menu-detail__header")
```

Figure 22 Set up library and fetch data using Python and RoboBrowser library

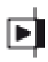
An example of finding wanted components is illustrated in Figure 22. Because the data is found using its attributions which we acknowledge through website inspection, the application collapses if these attributions' names change. It is an unavoidable problem of using webscraping together with the ones mentioned in section 4.3.2.

4.3.2.4 Data utilization

The data received from the website is delivered to the audience using Pepper mainly through his speech or tablet. Pepper conversation can be developed just with Choregraphe while making tablet visualization requires web making tools which are HTML, CSS and Javascript.

Pepper can speak to the customer using ALTextToSpeech or ALDialog module. The modules enable customization of Pepper speech, such as voice, volume, language and speech speed and content. There are premade code boxes for speech in the box library in Choregraphe. The content of the speech/dialog can be manually programmed, or it can receive content from other boxes, in this case the python box we used to retrieve information from the website.

A box has one or many input and output which enable its ability to commute with the boxes connected to it. There are 5 natures for the input which are onStart, onStop, onEvent, ALMemory input and onload (table 1). The most common use ones are onStart, onStop and onEvent. As the names imply, the box is started once the onStart input is triggered and the box is stopped when onStop input is stimulated. OnEvent box does not have any effect on the box but it is used to call a specific function under the function's name onInput <input_name> in the box script. Unlike onStart, onStop and onEvent, the last two natures are used only in a diagram box. ALMemory input and onLoad cannot be activated from outside of the diagram box and works only when the diagram is loaded. ALMemory input receives a signal every time the value of the data stored in ALMemory is updated. The data mentioned are for example, data received from Pepper sensors and devices. Output is simpler with two natures onStopped and punctual (table 2). The effect of onStopped varies based on the box type. In a Diagram box or a timeline box, onStopped output corresponds to onStop input being triggered and stops the box. However, in a dialog or a python box, onStopped does not stop the box but it deactivates the dialog topic, or it has no specific effect (Softbank robotics n.d.).

Input natures		
Nature	On the box	Within the diagram box
onStart		
onStop		

onEvent		
ALMemory input		
OnLoad		

Table 1 Input natures
(Softbank Robotics n.d.)

Output natures		
Nature	On the box	Within the diagram box
onStopped		
punctual		

Table 2 Output natures
(Softbank Robotics n.d.)

A box can receive and pass data through input and output. The input and output data type comprise bang, number, string and dynamic. They are separated by colour of the output as shown table 3. Bang type contains information whether the output is stimulated but it does not carry any other data. Number output indicates float, int or an array of number data type. String output represents string or array of strings. Dynamic output covers all the data types mentioned above.

Type	Input	Output
Bang		
Number		
String		
Dynamic		

Table 3 Input/Output data types
(Softbank Robotics n.d.)

A local site for the robot is created with HTML, CSS and Javascript to show the retrieved data on Pepper tablet. Showing data on the tablet supports users in understanding robot speech and delivers information which cannot easily be shown in speech such as image. Thus, it ensures that user-robot communication is delivered correctly, wholly and lively. The HTML site commutes with the robot using QiMessaging. Websites can access QiMessaging services through Javascript binding that QiMessaging provides. QiMessaging is hosted on the robot and it is added to the HTML file as presented in Figure. Next, the connection is established by creating qiSession object and then it is used to call services using Javascript (Figure 23).

```
<script src="/libs/qimessaging/2/qimessaging.js"></script>
```

Figure 23 Syntax to add Qimessing to HTML file

ALMemory is the module used to transfer the data to the robot site. The data is obtained using Choregraphe then it is stored in ALMemory for later use and remote access using insertData function. Robot site fetches data from ALMemory using the module's getData function (Figure 24).

```
// Create session object
var session = new QiSession();

// Using session to call ALMemory module and use its getData function to get data
session.service("ALMemory").then(function(ALMemory) {
    ALMemory.getData('data_key').then(function(data_key){
      // Do something to the data
    });
```

Figure 24 Get data using Qimessing and ALMemory module in Javascrip

The web-scraping task can be done with Javascript as well, however it is not optimal. The tablet CPU has low capacity so "heavy" work such as animations and events which requires external frameworks or strong and stable internet connection might not work as browsers normally does. Therefore, all tasks are prioritized to be done with Python in Choregraphe.

4.4 Testing

4.4.1 Pre-test

Before the user test was launched, the application was tested first by the development team and representatives of Sokos hotel. This phase was implemented to make sure that all requirements mentioned in section 4.2 were met. The environment where the test occurs was an office room with less than 10 people at the testing time and only one person tried out the robot at a time. The testers had full or semi knowledge on interacting with Pepper. The test had a great result with all functions worked as expected. The testing environment does not resemble a real case scenario which is a crowded and noisy hotel lobby or real testers which are users which likely have not used a humanoid robot before. However, this phase is an important step to ensure the robot works for the real user testing phase.

4.4.2 User test and result data

The user testing phase lasted for 8 days and is divided into 2 periods. Splitting the test into 2 was for small errors missed during the pre-test to be fixed during the break. It minimized the effect of bad application to the overall user experience of the robot. The method to collect data was observation. The setting for testing and collecting data has been described in section 3.2. Below is the collected result of the test from our team observation (Figure 25, table 4 and observation note).

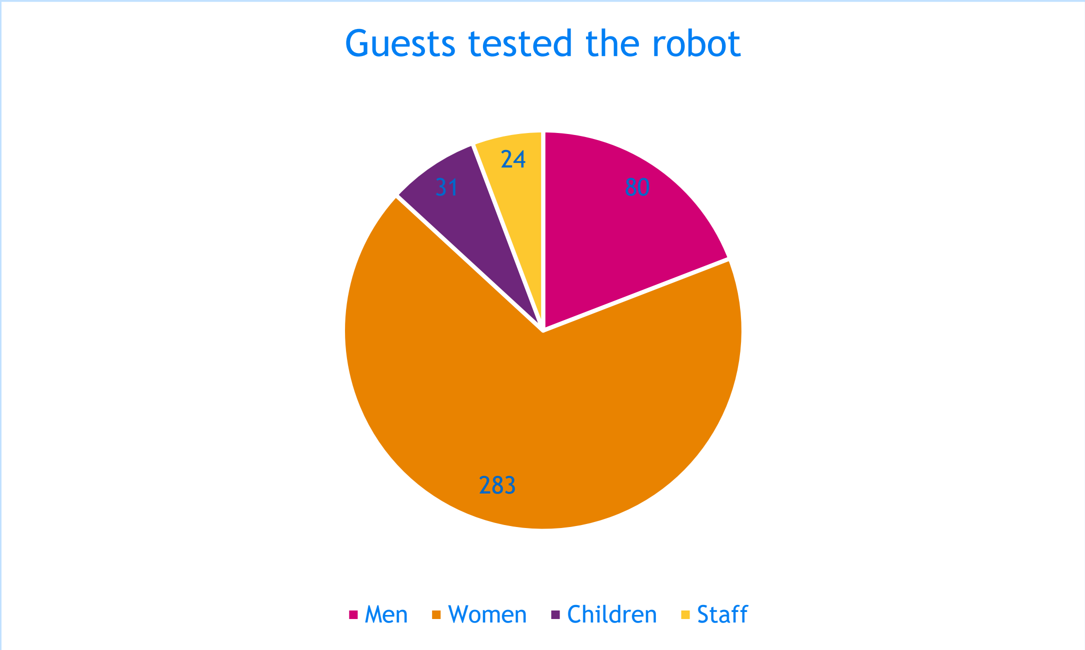


Figure 25 Number of guests tested the robot

User	Age (optional)	Comment
Woman	45	I'm afraid it will grab me by the hand.
Reception girl		Oh my god! (nice way)
Girl	18	I love Pepper
Woman	70	My granddaughter is coming here tomorrow, hope it's still here. It's super!
Croatian Skate team coach		Where can we get one for the Croatian Ice-Skating federation

Group of skater girl		It is a nice robot. We will come back to play with it later.
Russian girl		My English is too bad for it to understand

Table 4 Collected comments during testing

Observation note:

- Some men tried to tease the robot rather than using it (10/80) and 2 men just curiously whispered in the background.
- Most people seem to be interested in it. Especially kids, they are super excited.
- The robot has difficulty in recognizing people with different accents or bad English.
- Most people tend to use Pepper tablets rather than having conversations with it.
- Young girls or kids seem to ignore the instructions given by the robot.
- Second test period starts with little people interacting with the robot (downtime).
- Kids seem to be reasonably excited, but the affordability of our interface is not good enough. They don't realize what the buttons are doing.
- A big middle-aged man got excited about dancing with a robot, it was fun.
- A baby puts a finger in Pepper's mouth.
- Main functions used are dance and attraction.
- The staff said they had not experienced the robot so far disturbing at all, but also fun.

4.4.3 Result analysis

Data collected from user tests shows that Pepper robot got attention from all targeted users including hotel's customers and hotel's staff. Hotel's customers were categorized into women, men and children groups. Women and men were the main tester. Together they constituted 86% of the total 283 participants. The result was reasonable as the robot is mostly for customer use and the number of adult customers often surpasses the number of adolescent customers.

In general, women were more interested in using the robot than men. The number of women approached and used the robot are 3.5 times higher than the number of men (Figure 25). A portion of men hesitated and did not take the robot as a serious product. Nearly 1 in 6 men just made fun of the robot without giving it a try. On the other hand, women properly went through the available functions and many of them praised it (Table 4)

Adolescent users including both teenagers and young kids showed most excitement when seeing and using the robot. They are mostly attracted to the robot itself and its entertaining function which is playing music and dancing as this function is eye and ear catching. One noticeable behaviour of this user group was that they often ignored the instruction given by the robot. As the robot works under semi-unsupervised settings, without much of our interference, the robot was programmed with warning to alert users when it made movement. It was to avoid the user staying too close to the robot when it starts moving. However, some children failed to notice the caution. Fortunately, the robot arms did not have much force, so no serious accidents occurred. In addition, kids might have unexpected actions such as trying to poke or put a finger in the robot mouth. These behaviours are difficult to predict and prevent and it requires consideration in future development.

Employees, another important stakeholder of the hotel, also joined the user test. The robot was enjoyable to them and it did not bother their work. Even though there are rumours about human fear of losing jobs to robots, it was not shown at all in our case.

The robot and its application successfully delivered basic functions which gave users a brief idea on the use of robots within the hotel industry, however there were some flaws remained. The user interface of the application and the way to use the robot confused some people, especially kids. Some users did not recognize that they could talk to the robot. It led to the overuse of tablets as the main method to interact with the robot, instead of its initial support function. Kids did not know what the function of the buttons shown on the tablet are. Moreover, the robot had problems understanding different accents or incorrect English, especially in the hotel lobby with a lot of background noise. It made users hesitate to keep talking to the robot when they do not get the response from the first try. Furthermore, for Pepper to properly make conversation, users need to establish eye contact with Pepper first. Once

Pepper detects that there is a person in front of it through its cameras, it starts listening to speech and it indicates that it is listening by having navy blue eyes. However, this step is not commonly known by random users and it is not easy to quickly notice that by yourself during a relatively short trial.

5 Conclusion

Based on the result analysis of the study case (section 4.5.3), we can draw some general conclusions for the hotel industry.

Overall, there is work in the hotel industry that service robots can assist. The reception application in the case hotel is an example. Pepper entertained customers, especially young ones and also provided customers information that is often asked at the reception such as facility location and popular attractions around the hotel. The robot usage gained interest from both customers and personnel of the hotel. There were 418 people who tried the robot out during the 8-day testing period. Users included different age ranges from underage kids to retired adults and include both men and women. The most attractive aspect of the robot from the test case was its entertaining function. It brought enjoyment to the customers and it was so far the largest value the robot contributes to the hotel. The reason for its popularity is because it did not require much action from the users to use and kids could easily use it too. Therefore, once customers get more used to using the robot for different tasks, other values of service robots in the hotel industry will be better shown.

As regards to the technical aspect of service robots, in the case hotel, the robot successfully performed desired functions, but in my opinion, service robots are not ready to work fully unsupervised in the hotel industry. It is noticeable from the result that there were unexpected accidents which Pepper robot could not handle by itself. Pepper is certainly not the most advanced humanoid robot in the market and there are plenty of features the robot can be improved. Pepper's ability to communicate with humans through conversation did not work well in the hotel setting. People switched to mainly using the robot's tablet as the result and there was no benefit from using a humanoid service robot anymore. The robot communication enhancement can be done through upgrading hardware or using additional software. The robot can be installed with a better microphone to better capture user speech in noisy environments. Some better speech recognition services can be added to the robot are, for example, Google Cloud Speech-to-Text or IBM which supports many different languages. Consequently, service robots become better as related technologies become more advanced and they can eventually work well without intensive care in the hotel industry. So even though there are limitations left, there is high hope in using service robots to assist work in the hotel industry in the future.

6 Discussion

There were many customers interested in using the robot, however, the excitement can be short-term. From the hotel case result, there was a downtime at the start of the second test period. Only a few people interacted with the robot during that time. The testing period is relatively short so there is not enough data to conclude whether customers will keep interest in the use of service robots in the hotel industry, particularly when service robots are no longer a curiosity to them. Therefore, further research might be needed to investigate this matter.

This research was done with the mindset that there are limitations. So even though there is an unanswered discussion, I consider the research a success. The thesis successfully analyses service robots' feasibility in the hotel industry and this information works well as a foundation for a larger-scale research on the same topic.

References

Electronic sources

Dawson, C. 2009. Projects in computing and Information Systems. 2nd edition. England: Pearson Education Limited.

Glez-Peña, D., Lourenço A., López-Fernández, H., Reboiro-Jato, M. & Fdez-Riverola, F. 2014. Web scraping technologies in an API world. Briefings in Bioinformatics. 15 (5), 788–797. https://doi.org/10.1093/bib/bbt026

International Federation of Robotics. 2018. Press release world robotics 2018 service robots. Accessed December 2019. https://ifr.org/news/service-robots-global-sales-value-up-39-percent

International Federation of Robotics. 2019. Press release world robotics 2019 service robots. Accessed 15 December 2019. https://ifr.org/news/service-robots-global-sales-value-reaches-12.9-billion-usd

Lawson, R. 2015. Web scraping with Python. Packt Publishing Ltd.

Marvin, R. 2018. The Best Low-Code Development Platforms of 2019. Accessed 15 December 2019. https://uk.pcmag.com/cloud-services/89789/the-best-low-code-development-platforms

Real Python. No date. Python Virtual Environments: A Primer. Accessed 10 May 2020. https://realpython.com/python-virtual-environments-a-primer/

Revfine. No date. Hotel industry: everything you need to know about hotels. Accessed 15 December 2019. https://www.revfine.com/hotel-industry/

Revfine. No date. Hotel technology trend: 11 upcoming innovations you must know. Accessed December 2019. https://www.revfine.com/hotel-technology/

Robotic Industries Association. No date. Humanoid robots. Accessed 15 December 2019. https://www.robotics.org/service-robots/humanoid-robots

Robotic Industries Association. No date. What are professional service robots. Accessed 15 December 2019. https://www.robotics.org/service-robots/what-are-professional-service-robots.

Sokos hotel. No date. History Orginal Sokos hotel Presidentti. Accessed 15 December 2019. https://www.sokoshotels.fi/en/helsinki/sokos-hotel-presidentti/history

Softbank Robotics. No date. Softbank Robotics Documentation. Accessed 15 December 2019. http://doc.aldebaran.com/2-5/home_pepper.html

SSH.COM. No date. SSH (Secure Shell). Accessed 10 May 2020. https://www.ssh.com/ssh/

Thabane, L., Ma, J., Chu, R. et al. 2010. A tutorial on pilot studies: the what, why and how. BMC Med Res Methodol. 10 (1).

Van Teijlingen, E.R. & Hundley, V. 2001. The importance of pilot studies. Social research updated. 35.

Figures

Tables