University of Essex
School of Computer Science and Electronic Engineering

CE903

Group 8

Robot Receptionist

Software Requirements Specification

Version: 1.0 **Date: 30/01/2019**

University of Essex
School of Computer Science and Electronic Engineering

Table of Contents

1. Introduction

1.1 Purpose

This software requirements specification (SRS) details the design and implementation of a robotic receptionist using the NAO Peppa robot. The following sections of this SRS contain detailed functionality and capabilities of the system, including all behaviours, the user interface and any limitations and constraints. The document also describes the intended users and any prior knowledge they may require to use the system.
At the end of this document the appendix contains a Gantt Chart representing the time scale of the project.

1.2 Scope

The software to be produced for this project is a controller for a robotic receptionist to be used at a University, and to assist with general queries along with general interaction. The robot will have the following capabilities, speech recognition and control, facial detection and tracking and physical gestures to enable interaction with users. The robot will have autonomous routines when idle to simulate social behaviour. There will be a user interface designed and developed for a built in tablet to also enable control of the robot and provide a visual display of information.
The robot will have basic autonomous navigation capabilities, such as the ability to follow a user, avoid obstacles and navigate around an area.
Overall, the system will be capable of detecting and tracking a human face, interacting with a user through speech, gestures and movement, and be able to assist the user with information requested, for example campus and laboratory information.

1.3 Definitions, Acronyms, and Abbreviations

SRS: Software Requirements Specification.
SDK: Software Development Kit.
API: Application Program Interface.

1.4 References

[1]. S. Howes, "First ever robot receptionist gets job in London", *mirror*, 2019. [Online]. Available: https://www.mirror.co.uk/news/uk-news/first-ever-robot-receptionist-gets-10738388. [Accessed: 01- Feb- 2019].
[2]. "Microphones — Aldebaran 2.5.11.14a documentation", *Doc.aldebaran.com*, 2019. [Online]. Available: http://doc.aldebaran.com/2-5/family/pepper_technical/microphone_pep.html. [Accessed: 04- Feb- 2019].
[3] "Tablet — Aldebaran 2.5.11.14a documentation", *Doc.aldebaran.com*, 2019. [Online]. Available: http://doc.aldebaran.com/2-5/family/pepper_technical/tablet_pep.html. [Accessed: 04- Feb- 2019].

1.5 Overview

The SRS is comprised of two major components, the overall description and the specific requirements of the project.
The overall description section of this report is concerned with a detailed overview of the system as a whole, giving an indepth description of the robots capabilities and limitations. This first section includes a description of all of the software, hardware and communications interfaces and how they will work synchronously, along with any constraints the robot will have. Finally, any assumptions and dependencies, along with the intended users and any related requirements are detailed.

The specific requirements of this report are most relevant to developers, detailing the exact functionality of the robot. This section includes all of the necessary functions that are required to create the robots capabilities and behaviours, as described in the overall description.
The specific requirements detail potential use cases for the robot, along with software attributes such as the system’s security and reliability, essentially an overview of the system from a technical perspective. The specific requirements also demonstrate potential use cases for the system.
The final section of this document is a brief description of the design of the system, this section details how the system will be built, the functions and overall architecture.

2. The Overall Description

2.1 Product Perspective

Pepper will act as a robotic receptionist for the CSEE department in the robotics laboratory and provide friendly interaction and useful information including campus info, events, directions and general info such as weather. Students can scan their campus card to enable pepper to tell them information regarding their timetable. Users can interact with pepper verbally or using the tablet on the front of her. Pepper will be able to answer questions, provide information and even play the game Rock - Paper - Scissors.

Greeting visitors and providing them with a range of useful information is an important role, but one that can be automated and performed by a robot, thus allowing human employees’ time to be used for tasks that cannot be so easily automated. A robotic receptionist fits this role, and can be used not only to perform required tasks as a receptionist, but also to entertain visitors, becoming a form of attraction for the location it is placed in.

This is not the first time a robot has been used for a receptionist role, in 2017 a version of pepper was placed at the Shoreditch offices of media agency Brainlabs, to act as a receptionist. The robot could greet people, talk to them and allowed the user to select from a variety of options on the touchscreen tablet. These options included contacting an employee of Brainlabs, providing information about the company, making the user a coffee and dancing[1]. The goal of using the robot as a receptionist in this scenario was to automate several jobs that a human employee would normally have to perform, such as registering guests and notifying the company of their arrival, thus freeing up their time to spend on jobs that cannot yet be performed by a robot.

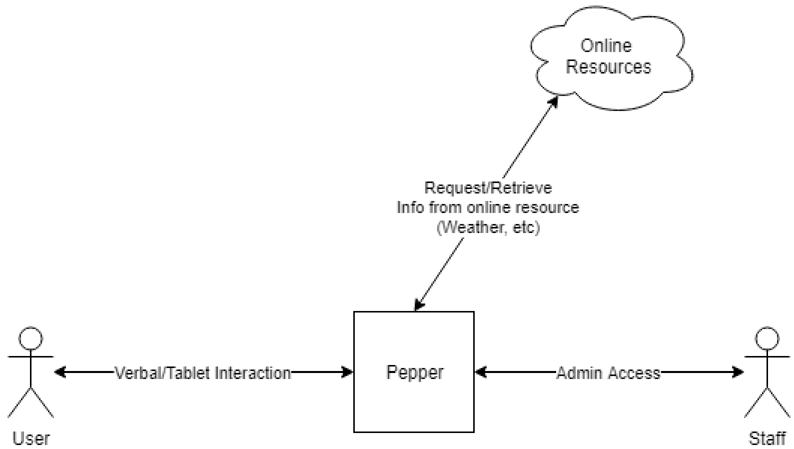

Figure 1: General Interaction Diagram.

2.1.1 User Interfaces

For accessibility purposes, all the functionalities of the robot can be executed by using any of the interfaces.

- Verbal Interface:
Verbal interface allows users to control and engage with the robot by talking. It translates speech to text, and performs an action or talks back according to the information received.
- Tablet Interface:
Tablet interface allows users to control and interact with the robot physically, by using the tablet located in front of the robot.
- Remote Interface:
Remote interface allows admins to connect to the robot and control it using a phone or a tablet. Remote connection mirrors the tablet, which makes it have the same functionalities.

2.1.2 System Interfaces

The system will be comprised of multiple software and hardware interfaces working synchronously to include all of the functionality of the robot, these can be defined as the following:

Software Interfaces:

The primary software interface for the system will be a speech to text, implemented using Google Cloud Speech to Text API.
This will be used to reliably convert the users speech to text, which can then be used for behavior selection or chatbot.

It will be used to access the conversational and weather information interfaces, using Amazon Lex API or Dialog-flow and the MET API respectively, this will provide human like responses for conversation and the ability to display current weather information to the user.
The speech to text interface will also be used to enable interactivity with the University information interface, using information from the university website in order to gather information about the University or campus. Finally a map interface using Google Maps will be implemented to display and give local information to the user.
The system will also include a databasing interface to store and retrieve information locally.

Hardware Interfaces:

The hardware interfaces required by the system are as follows, a navigation interface, using a lightweight navigation library that will control the motors and movement of the robot, enabling it to navigate in the environment safely and assist user.
Along with a barcode reading interface, this will be a hardware system capable of scanning and reading a student’s registration card barcode. This would enable the functionality of displaying a specific students information.

2.1.3 Communications Interfaces

The system will make use of TCP/IP to communicate wirelessly with a mobile Android device. Similarly, WiFi peer-to-peer will allow the robot to communicate more effectively.

2.1.4 Memory Constraints

According to Pepper’s datasheet, the RAM is limited to 4GB DDR3 while the flash memory to 32 GB eMMC. Likewise, the robot has a Intel HD graphics GPU that reaches up to 792 MHz. On the other hand, the tablet has two 512MB DDR3 RAM, a flash memory of 32GB eMMC, a CPU of 1.3 GHz quad-core, a cache of 512 KB and a micro SD card socket available.

2.1.5 Operations

The system will have the following modes of operation:

Autonomous behavior: Essentially the idle mode of Pepper, the autonomous behaviours exhibited when no user is present.
Interactive behavior: This behavior will be enabled when a user is interacting with Pepper.
Admin mode: For accessing Pepper’s settings or the behaviour log, this can also be used to alter behaviours. This mode can only be accessed by authorized users.

Reset Mode: If some area of the software were to crash, Pepper can be reset manually.

2.1.6 Site Adaptation Requirements

The system designed has no site adaptation requirements.

2.2 Product Functions

The following is a high level overview of the functionality of the robot:

2.2.1: Listen and search functionality:
The robot will be able to listen and search for a human in the environment without interruption from other behaviours.

2.2.2: Voice Detection:
The robot will be able to detect the human's speaking.

2.2.3: Visual Detection:
The robot will be able to detect a user when they enter a specified area.

2.2.4: Physical and Verbal Interaction:
If there are no users in the area the robot will start listening to the environment. Once a user has been detected through speech or visually, the robot will respond through verbal and physical interaction.

2.2.5: Retrieving information from the University website:
The robot will be able to access the university website and retrieve information from it.

2.2.6: Student ID reading:
The robot will be able to read a student's ID and retrieve information of the student i.e. course, timetable.

2.2.7: conversation and question answering through tablet:
The robot will respond to questions received through the tablet.

2.2.8: Verbal conversation and question answering:
The Robot will be able to converse with the human and answering general questions verbally. If the robot doesn't have the answer, it will give a default reply.

2.2.9: Physical interactions during communication:
The robot will show physical acknowledgement toward the user, through turning to look and moving its arms.

2.2.10: Movement:
The robot will have fully movement capabilities, forward, left, right and backward.

2.2.11: Information Retrieval:
The robot will be able to retrieve information from a variety of sources, both online and locally.

2.2.12: Hand gesture recognition
The robot will be able to recognize basic hand gestures of the user.

2.2.13: Physical Gestures:
The robot will be able to simulate hand gestures.

2.2.14: Play Rock, Paper, and Scissors Game
The robot will be able to play the game Rock, Paper, and Scissors with a user by recognizing the

basic hand gestures of the user and making a random gesture by its hand at the same time. It will be able to define the winner through this recognition.

2.2.15: Saving new behaviors:

The robot will save the most recent behaviors and actions. Additionally it will save the questions that it wasn't able to answer in order to increase its own informative capabilities.

2.2.16: Questions email:

The robot will send an email to the admins if it fails to answer a specific question with the following data:

- Name and email of the user.
- The question of the user.

2.2.17: Differentiate between users and admins

The robot will have the ability to recognize a pre-defined person from their face. (Administrator's only).

2.2.18: Controlling the Robot through mobile/laptop remotely (two way communication)

The admins will have the ability to control the robot to perform simple behaviors (such as saying a specific sentence, moving forward or backward, left or right) remotely from a tablet/phone.

2.2.19: Admin Setting:

The administrators will have higher level system access. These settings will enable the following:

- Activate/deactivate behaviors.
- Add/delete new admin.
- Retrieving the behavior log.

2.3 User Characteristics

This system is intended to be used by Applicant day visitors, CSEE Administrators and basically any user that gets close enough to communicate with pepper. Applicant day visitors and any user that interacts with pepper can only give voice commands and use the tablet that is attached to pepper; On the tablet the users can navigate through a user-friendly interface that provides many information that can be needed on such day, for example information about university accommodation, the weather, requirements needed for applicants and many more. As for the CSEE Administrators they are intended to program and control pepper through a software using Python, there they will manage all the tasks and functionalities that the robot pepper is made for, and they will have access to program the tablet. The administrators will also interact with pepper as any other user. Pepper will be programmed with functionalities that will fit the role of a receptionist from giving information to interacting with the users, so the functionalities chosen for pepper are based on an assumption of what the users will need on the applicant day.

2.4 Constraints

2.4.1: C++ python, Java and JavaScript are the only high-level languages that are supported.

2.4.2: Goal of the project is to develop a receptionist robot.

2.4.3: Health and safety while interacting with the robot has been considered, redundancy's will be in place to ensure an appropriate amount of care has been taken to avoid any safety issues.
2.4.4: Keeping the battery and motor temperatures in a safe operating range.
2.4.5: Main computer powering the robot is read only and cannot be modified.
2.4.6: On board computer cannot handle graphically demanding tasks.
2.4.7: The API's used have limitations on their informational and behavioural capabilities.

2.5 Assumptions and Dependencies

2.5.1 NAOqi SDK support:
The programming of the robot depends on the NAOqi SDK, the development kit support and updates are crucial for further development and maintenance, it is assumed that the current version (2.5) is used in all cases.
2.5.2 SoftBank product support:
In case of hardware failure or broken parts, SoftBank support is required to repair the robot or acquire spare parts.
2.5.3 User ability:
It is assumed that the user of this system will not need prior knowledge of software development, robotics or any other related technical skill.
2.5.4 SDK and API updates:
The software developed will be completed using the most recent version supported at the time of development.
2.5.5 Operational Environment:
The robot is designed to operate in indoor environments only. It is assumed there will be minimal noise and adequate lighting for the robots sensors to operate effectively.
2.5.6 Interaction:
The robot will interact with at most one person simultaneously, it is assumed that if there is no interaction within 20 seconds, there is no longer user engagement.
2.5.7 Language Support:
The robot will recognize the English language only.
2.5.8 Interruptions and Redundancies:
The robot will not respond to interaction from other users whilst engaged by the initial user.
The robot requires an internet connection at all times, if the internet is disconnected the robot will shutdown.
The robot will shutdown if there is any critical error from which it can not recover.

2.6 Apportioning of Requirements:

2.6.1 Critical Requirements:

Requirements that are critical for the overall functioning of the system and must be implemented and ready in the first version.
- Listen and search continuously for a human.

- Detect human’s voice.
- Visually detect humans.
- Retrieving information from the university websites.
- Conversation and question answering through tablet.
- Verbal conversation and question answering.
- Movement.
- Information retrieval from the web.
- State transition routine.

2.6.2 Secondary Requirements:

Requirements that are important for a more social and human like robot. These requirements are next on the list after the implementation of the critical requirements but can be delayed or cancelled depending on the time.

- Physical and verbal actions on state transitions.
- Student ID reading.
- Physical interactions during communication.
- Hand gesture recognition.
- Making hand gestures.
- Playing rock, paper, scissors.
- Saving behaviors.
- Differentiate admins from others.
- Controlling the robot remotely.
- Admin settings.
- Complex navigation with localization.
- Makaton.

3. Specific Requirements

3.1 External Interfaces

Voice Command

- Used as an input to interact with the robot.
- The sound file is recorded from the microphones of the robot.
- Microphone sensitivitiy: 250mV/Pa +/-3dB at 1kHz [2]
- Microphone Frequency range: 100Hz to 10kHz (-10dB relative to 1kHz) [2]
- Passed to the Gooogle text to speech API.
- Result is passed to the behavior selector for output.
- Performs similar jobs as tablet input.

Tablet Input

- Used as an input to interact with the robot.
- Depending on the user input, the output is either a command or a menu navigation action.
- If the input is a command, it is passed to the behavior selector for output.
- If the input is not a command, tablet displays the result of the menu action.

- The tablet display supports all the common media formats.[3]
- Performs similar jobs as voice commands.

Camera Input

- Used as an input to trigger the greeting function of the robot.
- Also used to orient the head of the robot towards the user.
- 2D Camera has 640*480 resolution at 30 fps.
- In one second, 5 frames are used to check faces.
- After a face is detected by the face detection module, the greeting behavior is activated.
- If the robot is already engaged with a user, coordinates of the face is passed to the head orientation function.
- Camera can output images in RGB, HSV and TIFF formats.

Remote Input

- Used to mirror tablet screen to interact with the robot.
- Computers or any type of electronic device that runs Android, iOS that have wireless connection capabilities can connect to the robot.
- Wireless hotspot is created on the tablet of the robot to provide remote connection.
- Wireless Protocol: IEEE 802.11 a/b/g/n
- Supported security: 64/128 bit: WEP, WPA/WPA2
- Works exactly like the tablet input interface.

3.2 Functional requirements

F.R.1: Navigation

F.R.1.1: The robot will be able to move forward, left and right.

F.R.1.2: The robots movement is limited to within a defined map.

F.R.2 Obstacle Avoidance

F.R.2.1: The robot will be able to detect and avoid collisions with static objects using both laser and sonar.

F.R.2.2: The robot will be able to detect and avoid collisions with dynamic objects using both laser and sonar.

F.R.3 Animation

F.R.3.1 Animation in neck articulation

F.R.3.2 Animation in waist articulation

F.R.3.3 Animation in arms articulation

F.R.3.4 Animation in hand/fingers articulation

F.R.4 Tablet output

F.R.4.1 The robot will be able to take and display pictures using the onboard camera’s.

F.R.4.2 The robot will be able to record and display video using the onboard camera’s.

F.R.4.3 The robot will be able to produce special layouts, for example email.

F.R.5 Tablet input

F.R.5.1 The tablet will accept single touch detection
F.R.5.2 The tablet will accept multi-touch detection
F.R.5.3 The tablet will have touch tracking
F.R.5.4 The tablet will be able to receive text input and display text output.

F.R.6 Reproducing audio

F.R.6.1 The robot will be capable of text to speech.
F.R.6.2 The robot will be capable of playing existing audio.

F.R.7 Camera input

F.R.7.1 The camera on the robot will be capable of facial detection.
F.R.7.2 The camera will be able to read an ID card using the cards barcode.
F.R.7.3 The camera will recognise hand gestures.
F.R.7.4 The camera will be capable of facial recognition.

F.R.8 Sensing audio

F.R.8.1 The robot will be able to recognise human speech and convert it to text using speech tagging.
F.R.8.2 The robot will be able to use tagged words as speech recognition for control.

F.R.9 Laser sensing

F.R.9.1 The robot will use a laser scanner for obstacle presence detection.
F.R.9.2 The robot will use the laser scanner to determine distance from objects.

F.R.10 Tactile sensing

F.R.10.1 The robot will use the built in tactile sensor recognise being touched on the head surface

F.R.11 Tactile sensing

F.R.11.1 The robot will use the built in tactile sensor recognise being touched on the hand surface

F.R.12 Motor position sensing

F.R.12.1 The robot will record odometry and velocity readings.
F.R.12.2 The robot will use the odometry and velocity readings for localisation.

F.R.13 Sonar Sensor Readings

F.R.13.1 The robot will use the built in sonar sensors for object detection.
F.R.13.2 The robot will use the built in sonar sensors for edge detection.

F.R.14 Backend actions

F.R.14.1 The following API access will be used, Android API, NAO Qi API, MET Office, Google Maps API, Google Text-to-Speech, Amazon-LEX, Dialog-Flow.

F.R.14.2 The robot will use a pre-existing map of its environment for navigation.

F.R.15 Database usage

F.R.15.1 The robot will capable of storing specific administrator faces for facial recognition locally and securely.

F.R.15.2 The robot will be capable of storing personal profile information locally and securely.

3.3 Performance Requirements

3.3.1 Autonomous behavior

Description: The robot should have autonomous movement to wander around while avoiding any present obstacle, proactive task-seeking and autonomous movement to wander around while avoiding any present obstacle.
Rational: In order for the robot to emulate a human-like idle behavior.
Dependency: None

3.3.2 Speech Recognition

Description: The robot should be able to recognize certain human commands.
Rational: In order for the robot to understand human voice commands.
Dependency: None

3.3.3 Socially Interactive (verbal)

Description: The robot should be able to react to human commands and to engage in human conversations through a chatbot.
Rational: In order for the robot to emulate a human-like verbal communication.
Dependency: None

3.3.4 Socially Interactive (physical)

Description: The robot should be able to recognize individuals through image face recognition.
Rational: In order for the robot to identify humans.
Dependency: None

3.3.5 Face detection

Description: The robot should be able to use body language (in the form of gestures) to communicate and to maintain eye contact while interacting with a human.
Rational: In order for the robot to incorporate gestures into verbal comunication.
Dependency: None

3.3.6 Providing information

Description: The robot should be able to provide general information about news, weather, transportation, etc. displayed on the tablet screen, as well as accepting user requests to access specific web services.
Rational: In order for the robot to provide useful information functionalities.
Dependency: None

3.3.7 Secured information

Description: The sensible information used by any user interacting with the robot (such as usernames and passwords) should be secured.
Rational: So that sensible information cannot be forcefully reached.
Dependency: None

3.3.8 User friendly applications

Description: The interactive application on the tablet should be intuitive to use.
Rational: To make the tablet interface easy for the user to interact with.
Dependency: None

3.3.9 Response Time

Description: Time taken between the human input and the robot response during an interaction with a human.
GIST: Fastness of the robot response.
SCALE: The response time of the robot while interacting.
METER: Measurements obtained from around 30 interactions while testing.
MUST: No more than 5 seconds on average.
WISH: No more than 2 seconds on average.

3.3.10 Extendibility

Description: The robot functionalities should be easy enough to extend for future projects.
Rational: So that the robot functionalities could be expanded or improved in future projects.
Dependency: None

3.3.11 Reusability

Description: The implemented robot functionalities should be modular and reusable.
Rational: So that other developers can reuse certain functionalities.
Dependency: None

3.3.12 Extendibility

Description: The robot functionalities should be easy enough to extend for future projects.

Rational: So that the robot functionalities could be expanded or improved in future projects.
Dependency: None

3.4 Logical Database Requirements

The following E-R diagram illustrates the logical database requirements. All other information (such as the available display information) is accessed remotely and does not need local database usage.

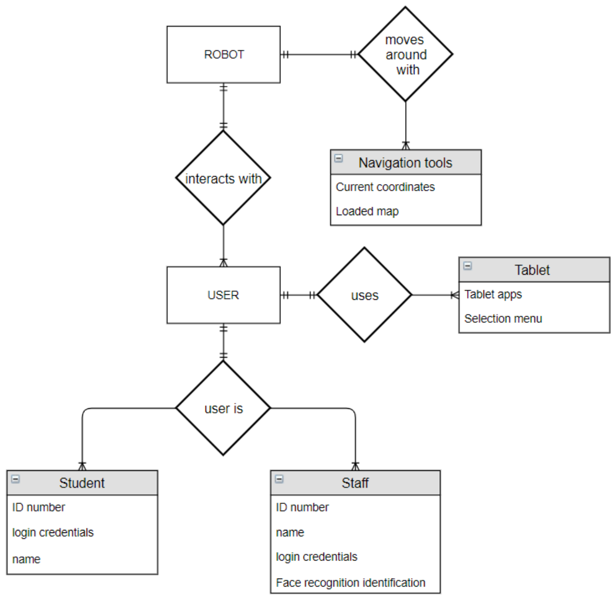

Figure 2: Logical Database Requirements Diagram.

3.5 Design Constraints

One of the biggest limitations of developing the robotic receptionist is the mobility that it needs to have when interacting with any human. Due to the unexpected nature of human movement, having a real-time reaction to what people do when interacting with Pepper can be a serious issue because in order to maintain a proper and human-like communication, the use of advanced computer vision is needed (particularly the 3D camera), as well as the analyses of all the surroundings using the sensors available. Even though this can very well be addressed making use of said features of the robot, the limited timeframe for which this project is to be developed does not allow the complete development of this task. Similarly, due to the limited mobility of the robot's arms and hands, only simple gestures and signalling can be performed when trying to communicate with a human using something other than voice output or the tablet. As for the software utilized for the project, making use of Choregraphe's built-in features can be a considerable limitation because some of them are not as efficient as others. This can be addressed by developing these types of behaviours completely from scratch in order to adjust their parameters to that of the project's. However, time is an issue again and thus this might be a task left for future implementation.

3.5.1 Standards Compliance

In regard to standards and regulations, mainting a clean and organized code is the main requirement as this is a necessity that must be met in order for the project to become fully understandable and to allow easy modifications to be implemented in the future. Concerning the report, following the IEEE standards is also essential due to it being the main source of information as to what the project is about and how the robot works; the correct description of the functionalities and behaviours implemented must be kept clear and concise. Finally, maintaining good references to data and information consulted, as well as previous researches regarding similar topics is a compulsory condition for the development of this project.

3.6 Software System Attributes

For the robotic receptionist to become fully deployable and for it to be able to have a clean and factual interaction each time it communicates with a human, certain requirements for different attributes must be met. This is described next.

3.6.1 Reliability

In order for the robot to become fully reliable at any given moment when interacting with people, two things are needed. First, the robotic receptionist must always have a stable and uninterrupted Internet connection. This will allow it to perform any web search necessary, as well as to inform the administrators in case there is an unexpected issue in real-time, allowing them to act accordingly as soon as possible. This also allows them to monitor the robot's performance on a regular basis. Secondly, the project needs to be developed making use of the same software versions of each program used. Doing this allows the code to be consistent throughout the complete project and for irregularities in the functionality to be addressed more easily.

3.6.2 Availability

Regarding the availability of the robotic receptionist, the system needs to have a set of development stages available at any given moment. This will allow for it to be tested in order to increase its reliability and performance. Having these type of project checkpoints also allows for easy recovery in case unexpected system errors occur. In likely manner, in order to make the robotic receptionist always available, it must be kept in its deployment zone at all times and connected to a power source when not being used. Another requirement the project must fulfill in order for the end result to be fully available is the use of multiple forms of inputs and outputs. Doing this allows the robot to not be limited as to what it can perform based on the information it receives.

3.6.3 Security

The main security feature needed for the project to be free of mailicious access is the use of cryptographic techniques that keep the data safe. However, this takes a large amount of time and effort, and a specific set of skills are required for it to be correctly developed. Therefore, in order to protect the information processed, only adminstrators will be able to access it after validating their identifiy. This can be done either through ID scan or direct user and password input. Similary, the communication between the robotic receptionist and the administrators must always be constant. This allows the administrators to monitor the use all data is given, as well as their integrity at any given time.

3.6.4 Maintainability

In order for the project development to be maintainted consistent throughout the whole implementation of it, the software utilized must be kept up to date at all times, especially the ones used the most such as Choregraphe and Chatbot. The tablet UI also needs to be regularly updated in order for the system's performance to be as optimal as possible. As for the robot's maintenance, this is done during the night by the staff in the Robotics Arena, which is the ultimate goal regarding the deployment zone of the robotic receptionist.

3.6.5 Portability

Due to the project being developed mostly in Choregraphe, the ease of porting the software is relatively simple, however if this is desired, the correct robot is required. Not any mobile robot can be a host for this project, as certain hardware conditions must be met. Similarly, software requirements such as having the same operating system are also compulsory for the portability of this project.

3.7 Additional Comments

Whenever a new SRS is contemplated, more than one of the organizational techniques given in 3.7 may be appropriate. In such cases, organize the specific requirements for multiple hierarchies tailored to the specific needs of the system under specification.

Three are many notations, methods, and automated support tools available to aid in the documentation of requirements. For the most part, their usefulness is a function of organization. For example, when organizing by mode, finite state machines or state charts may prove helpful; when organizing by object, object-oriented analysis may prove helpful; when organizing by feature, stimulus-response sequences may prove helpful; when organizing by functional hierarchy, data flow diagrams and data dictionaries may prove helpful.

In any of the outlines below, those sections called "Functional Requirement i" may be described in native language, in pseudocode, in a system definition language, or in four subsections titled: Introduction, Inputs, Processing, Outputs.

4. Design

4.1 System Architecture

ID	Behavior	Description
B1	Listening to the environment	The robot will be able to listen and search for a human beaning in the surroundings without stopping in all states.
B2	Recognising human voices	The robot will be able to detect the human's speaking.
B3	Obstacle Avoidance	The robot will advance while avoiding any present obstacles
B4	Visually detecting humans	The robot will be able to detect the human when the human enters the vision area of the robot. At any time of the operation the tablet will show the number of humans in the tablet.
B5	Socialize in state transitions	Once human detected verbally or visually, the robot will execute verbal and physical action.
B6	Retrieving information from the web	The robot will be able to access the university database and retrieve information from it.
B7	Retrieving local information	The robot will access the local memory to retrieve information.
B8	Student ID reading	The robot will be able to read the student ID and retrieve the information of this student
B9	Deliver information through tablet display	The robot will display a message for the user through the tablet display.
B10	Deliver questions verbally	The Robot will be able to converse with the human and answering general questions verbally. If the robot doesn't have the answer, it will give a reasonable reply.
B11	Gesturing during communication	The robot will move his face toward the human while speaking. Additionally it will move his arms and waste during the conversation.
B12	Hand gestures recognition	The robot will be able to recognize the basic hand gestures of the human
B13	Saving the behaviors	The robot will save the most recent behaviors and actions. Additionally it will save the questions that it wasn't able to answer.

B14	Sending emails	The robot will send an email to the indicated address.
B15	Recognising admins	The robot will has the ability to recognize a pre-defined persons from their faces (admins).
B16	Remote control	Controlling the Robot through mobile/laptop remotely (two way communication)
B17	Admin Setting	The admins will access a robot settings, allowing them to enable behaviors, managing admin profiles and retrieving the behavior log.
B18	Performing hand gestures on command	The robot will perform a specific hand gesture.
B19	Receiving a keyboard input	The robot will read a keyboard input from the user.
B20	Taking pictures	The robot will take a picture using the frontal cameras.

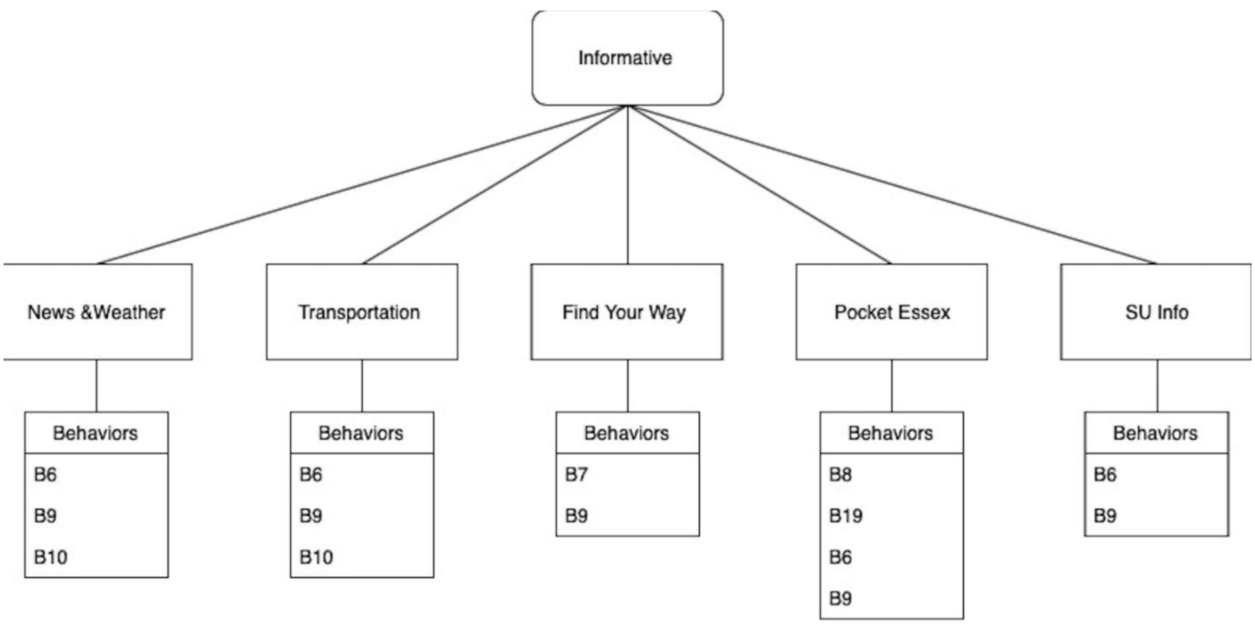

Figure 3.a: System Architecture Diagram.

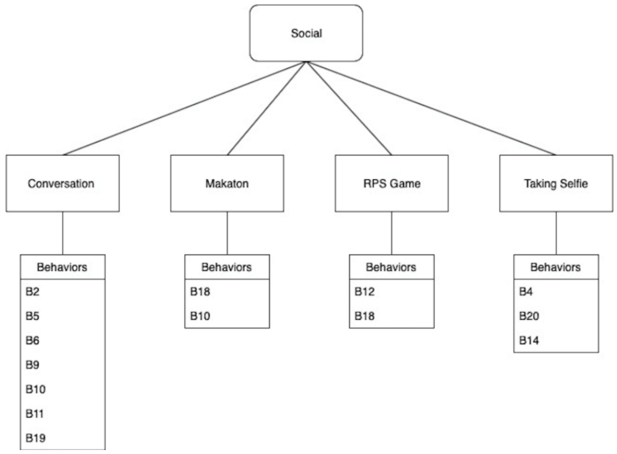

Figure 3.b.

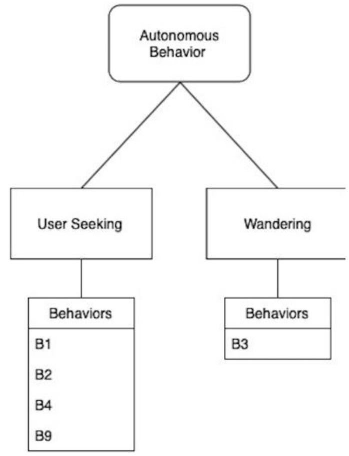

Figure 3.c.

4.2 System Design

Function Name:	Autonomous behaviour
Responsibilities	• Navigation when no user is present (wandering) • User seeking using speech and face recognition • Avoiding obstacles • Head, arm, hand and waist movements using animation to provide human-like gestures.
Functional Requirements:	F.R.1, F.R.2.1, F.R.3, F.R.7.1, F.R.7.4, F.R.8, F.R.9, F.R.10, F.R.11
Subsystem Design	20 Seconds of no interaction with the robot causes the robot to enter autonomous behaviour. The robot will then use navigation to wander the premises randomly, searching for a user using facial and audio speech recognition. It will also use animation to move its head and arms to appear more human-like.

Function Name:	Interactive behaviour
Responsibilities	• Recognising commands from the user, either verbal or from the tablet interface • Verbal communication with user • Displaying information on tablet display • Gestures using animation • Using gesture recognition to play rock paper scissors • Web scraping and using API's to obtain user requested information.

	• Storing a record of actions/responses using remote storage • Scanning a student's campus card to provide additional information
Functional Requirements:	F.R.3, F.R.4, F.R.5, F.R.6, F.R.7, F.R.8, F.R.10, F.R.12.1, F.R.13
Subsystem Design	Once the robot has detected a user, it uses animation and speech to give a greeting to the user. The robot will use facial recognition to look at the user during the interaction. The robot then responds to user commands (verbal or through tablet) user input must be either a predefined command or a phrase the robot can respond to using the chat-bot. Lack of user response for 20 seconds will cause the robot to revert to autonomous behaviour. Animation modules along with gesture recognition enable the robot to play rock – paper - scissors and to make the communication more interactive.

Function Name:	Administrator mode
Responsibilities	• Allowing access to settings and altering Pepper's behaviours and functions • Overriding existing functionality • Reviewing the behaviour logs and emails/feedback • Change administrator pins
Functional Requirements:	F.R.4, F.R.5, F.R.7.2, F.R.8, F.R.10, F.R.12.1, F.R.13
Subsystem Design	Mode can only be accessed by an authorised user, who must have an Admin ID and pin. An admin can enable/disable certain behaviours, for example remote access and movement. They can also add or remove admins. Functions can be modified, added or removed.

Function Name:	Reset Mode
Responsibilities	• Allows the robot to be shut down and completely reset
Functional Requirements:	F.R.10
Subsystem Design	Pressing the reset button the head of the robot completely shuts down and restarts the robot.

	Rebooting the robot is used when an unexpected problem occurs. Problems can then be reviewed by an administrator.

5. Change Management Process

Method – Hybrid Agile/Waterfall
This project will follow a hybrid methodology as our working process. The waterfall methodology allows for specific objectives to be set and managed, giving each stage a clear milestone, and a working prototype, whilst the agile methodology allows for any potential critical requirement changes and additional features to be added, without disrupting the entire project.

Requirements Gathering
As the project is following a hybrid methodology, a considerable amount of time was spent on gathering the requirements of the system, to be set into objectives in a given time frame shown in the attached Gantt Chart. The project will have a working prototype, where additional non-critical features can be implemented.
To set the requirements for the project, we discussed the overall project objectives with the project supervisor, along with a questionnaire that was sent to potential users to discover the types of functionality they may expect. We then used this information to set specific objectives for the project as described in the product functions and functional requirements of this report.

Requirement Changes
The nature of the project and the methodology we are using allows for requirement changes to an extent, as once there is a working prototype, additional functionality can be added iteratively through different versions. Any changes will be limited as much as possible however, to ensure the core functionality of the project is not altered. As such we have specified the requirements to be completed as critical functionality and secondary functionality. The critical functionality of the project is the essential components of the robot that need to be completed first, followed by the secondary functionality that will be implemented provided the critical functionality is completed.

6. Document Approvals

By signing this document, I approve the content of this SRS document.

Professor:
Name (printed): Dr. Vishwanathan Mohan

Signature:

Date: / /

7. Appendix

7.1 Gantt Chart