

PIXEL, A VIRTUAL ASSISTANT WITH FACE RECOGNITION

Functional Specification

Institiúid Teicneolaíochta Cheatharlach



**INSTITUTE of
TECHNOLOGY
CARLOW**

At the Heart of South Leinster

Tataru Theodora

C00231174

Supervisor: Joseph Kehoe
30th of April, 2021

Declaration on plagiarism

<i>Student</i>	Theodora Tataru – C00231174
<i>Tutor</i>	Joseph Kehoe
<i>Institution</i>	Institute of Technology Carlow
<i>Assignment Title</i>	PIXEL, A VIRTUAL ASSISTANT WITH FACE RECOGNITION
<i>Submission Date</i>	30 th April 2021

I declare that this research project titled "PIXEL, A VIRTUAL ASSISTANT WITH FACE RECOGNITION" has been written by me under the supervision of Joseph Kehoe.

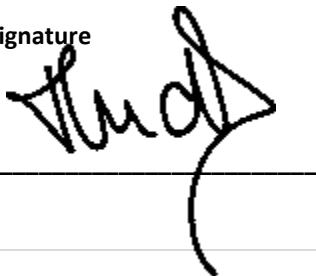
This document was not presented in any previous research papers for the award of a bachelor's degree to the best of my knowledge.

The work is entirely mine, and I accept full responsibility for any errors that might be found in this report. At the same time, the reference to publish materials had been duly acknowledged.

I have provided a complete table of references for all works and sources used in the preparation of this document.

I understand that failure to conform with the Institute's regulations governing plagiarism represents a serious offense.

Signature



Date:

30th April 2021

ABSTRACT

This document aims to provide a detailed representation of Pixel's Virtual Assistant functionalities. These functionalities are the core of the project and are the components that provide intelligence to the Virtual Assistant.

The functionalities of Pixel Virtual Assistant are validated with a survey completed by over 60 people, from different ages and backgrounds. This survey is provided and detailed in this document.

The main use case diagram is provided, which briefly outlines the functionalities of the project, followed by detailed use cases of each functionality.

In the last part of the document, the FURPS+ model is detailed.

CONTENTS

Abstract.....	2
Contents.....	3
Tables.....	5
Figures.....	6
Introduction.....	7
Application overview.....	8
Survey.....	9
General characteristics.....	9
Questions about "Pixel" Virtual Assistant.....	12
Users.....	21
Unregistered users.....	21
Registered users.....	21
Example.....	22
Main functionalities.....	23
Initiate interaction.....	25
Register.....	26
Get weather.....	28
Get time.....	29
Get date.....	30
Get COVID stats.....	31
Get help.....	32
Get definition.....	33
Get location.....	34
Call location.....	35
Get joke.....	36
Create list.....	37
Retrieve a list.....	38
Retrieve all lists.....	39
Add item to list.....	40
Delete item from list.....	41
Delete list.....	42
Restore backup.....	43
Create reminder.....	44
Retrieve reminder.....	46
Delete reminder.....	47
Create SOS contact.....	48
Retrieve SOS contact.....	49
Update SOS contact.....	50

Delete SOS contact.....	51
Send SOS	52
FURPS+	53
Functionalities.....	54
Core functionalities.....	54
Other functionalities	55
Usability	56
Reliability	57
Performance	58
Suportability.....	59
+	60
Security	60
Implementation	60
Interface.....	60
Testing.....	61
Similar products and inspiration	62
Conclusion.....	63
Bibliography	64

TABLES

Table 1 "Initiate interaction use case"	25
Table 2 "Registration use case"	27
Table 3 "Get weather use case"	28
Table 4 "Get time use case"	29
Table 5 " Get date use case"	30
Table 6 "Get COVID stats use case"	31
Table 7 "Get help use case"	32
Table 8 "Get definitions use case"	33
Table 9 "Get location use case"	34
Table 10 "Call location use case"	35
Table 11 "Get joke use case"	36
Table 12 "Create list use case"	37
Table 13 "Retrieve list use case"	38
Table 14 "Retrieve all lists use case"	39
Table 15 "Add item to list use case"	40
Table 16 "Delete item from list use case"	41
Table 17 "Delete list use case"	42
Table 18 "Restore backup use case"	43
Table 19 "Create reminders use case"	45
Table 20 "Retrieve reminders use case"	46
Table 21 "Delete reminder use case"	47
Table 22 "Create SOS contact use case"	48
Table 23 "Retrieve SOS contact use case"	49
Table 24 "Update SOS contact use case"	50
Table 25 "Delete SOS contact use case"	51
Table 26 "Send SOS use case"	52

FIGURES

Figure 1 "Age diversity - Survey"	9
Figure 2 "Computer skills background - Survey"	9
Figure 3 "Virtual assistant knowledge - Survey"	10
Figure 4 "Popular virtual assistants - Survey"	10
Figure 5 "Desire to purchasing a virtual assistant - Survey"	11
Figure 6 "Description of "Pixel" - Survey"	12
Figure 7 "Time functionality - Survey"	13
Figure 8 "Weather forecast functionality - Survey"	13
Figure 9 "Definition functionality - Survey"	14
Figure 10 "Compliment action - Survey"	14
Figure 11 "Joke action - Survey"	15
Figure 12 "Registration check - Survey"	15
Figure 13 "Registration process - Survey"	16
Figure 14 "Manipulating lists - Survey"	17
Figure 15 "Personalized output - Survey"	17
Figure 16 "WIFI connection - Survey"	18
Figure 17 "Pixel improvement - Survey"	19
Figure 18 "Pixel impact - Survey"	20
Figure 19 "Pixel Virtual Assistant use case diagram"	24
Figure 20 "Mobile data"	55
Figure 21 "Virtual Assistant display"	56

INTRODUCTION

This document focuses on the functionality that "Pixel" Virtual Assistant brings to its users.

"Pixel" is a virtual assistant encapsulated in a mirror that narrates and displays its answers to the mirror surface. In this document, the way users interact with "Pixel" is illustrated and described in detail.

The purpose of the document is to describe what the system can perform and what its constraints are.

Over 60 potential users completed a survey to validate that the functionality designed for "Pixel" meets the stakeholder's expectations and that the system is designed simple and accessible by any individual of any age and any level of computer skills.

The functionality, usability, reliability, performance, supportability, and other system attributes are described in this document.

APPLICATION OVERVIEW

"Pixel" Virtual Assistant is based on the traditional digital assistants that perform queries for the users such as weather, time, definitions, and others. This particular virtual assistant model targets the general segment of the market, people with poor computer skills and people with hearing disabilities.

People with hearing disabilities cannot benefit from the traditional virtual assistants as they cannot hear properly what the virtual assistant responds to their query. People with poor computer skills find it hard to engage with a virtual assistant as they cannot comprehend when the gadget is processing and what information is provided to their requests.

Therefore, "Pixel" is a redesigned virtual assistant encapsulated into a mirror, having at its core facial recognition. The mirror helps its users stay engaged for the whole conversation and is also used to provide real-time answers by displaying its responses on the mirror's surface along with the vocal response.

SURVEY

A survey was conducted using Google Forms to validate and improve the "Pixel" virtual assistant's functional design. The survey was completed by 67 people of different ages and background skills.

The complete results of the survey can be found at

https://docs.google.com/forms/d/1MbV3gCgHmoCn9v5nxpeJhIM9SXFndIKtY0_heSfgME4/edit#responses

GENERAL CHARACTERISTICS

What is your age?

67 responses

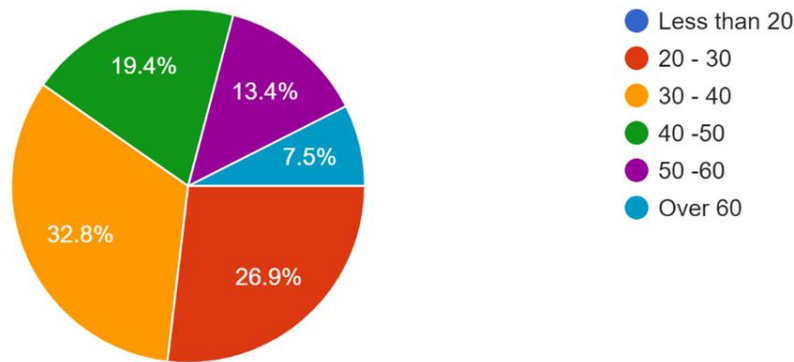


Figure 1 "Age diversity - Survey"

How would you describe your computer/mobile skills?

67 responses

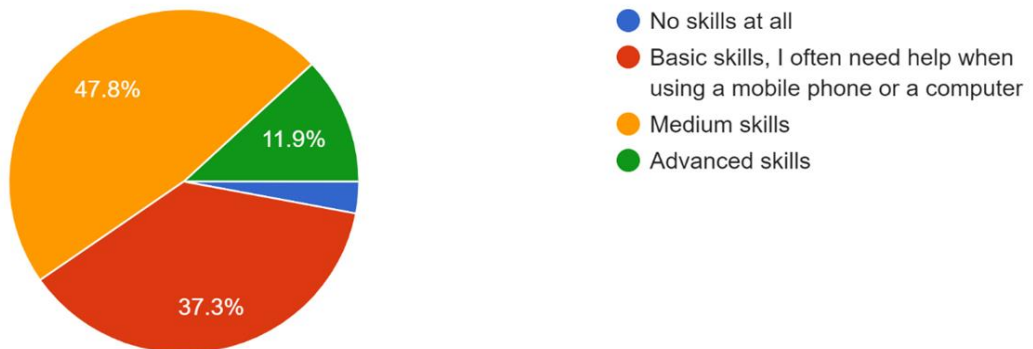


Figure 2 "Computer skills background - Survey"

As seen in **Figure 1** and **Figure 2**, the survey was completed by people of all ages and different computer skill levels. Most people declared that they have medium computer skills, followed by basic skills, where they often need help to perform actions on their devices. 12% of the people related that they have advanced computer skills, and 3% do not possess any computer skills.

Are you familiar with the term "Virtual Assistant", "Intelligent Assistant"?

67 responses

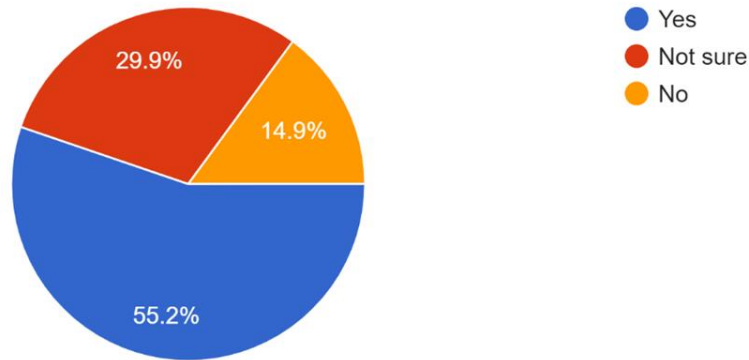


Figure 3 "Virtual assistant knowledge - Survey"

Have you heard of Alexa, Siri, Hi Google or Bixby?

67 responses

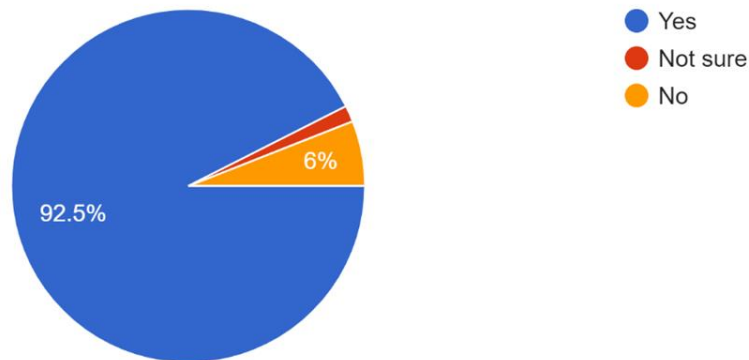


Figure 4 "Popular virtual assistants - Survey"

From **Figure 3** and **Figure 4**, it can be clearly observed that a significant percentage of people do not see any correlation between "Virtual Assistant" and Alexa, Siri, and others.

Would you like to have a Virtual Assistant?

67 responses

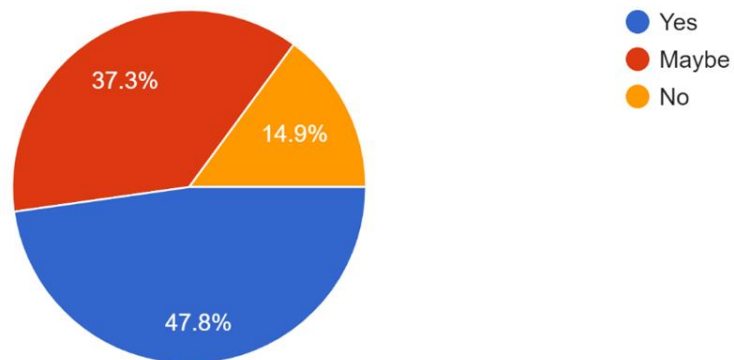


Figure 5 "Desire to purchasing a virtual assistant - Survey"

As seen in **Figure 5**, almost 48% of people would like to have a virtual assistant, followed by nearly 38% of people who would not like to possess a Virtual Assistant. For the purpose of a logical answer, people were asked to explain their choice.

The 48% of people who answered that they would like to have a virtual assistant justified their answer because it is a useful device that would help them better manage their time, and it is a fun device to have.

Most people who declared that they do not want such a device, or they are not decided on this matter, related that virtual assistants seem too complicated to be used or they are concerned about their data privacy. Both concerns of the last segment of the people are addressed directly by Pixel Virtual Assistant.

QUESTIONS ABOUT "PIXEL" VIRTUAL ASSISTANT

This part of the survey stated a brief description of "Pixel", as seen in **Figure 6**.

...

Pixel is a virtual assistant encapsulated in a mirror. To start a discussion with Pixel, you need to stand in front of the mirror. Nothing else. The virtual assistant listens to anything you have to say and provides you meaningful answers to most of your questions. The answers are returned to you by voice and on the mirror surface. To stop your dialog with Pixel, you need to get away from the mirror.

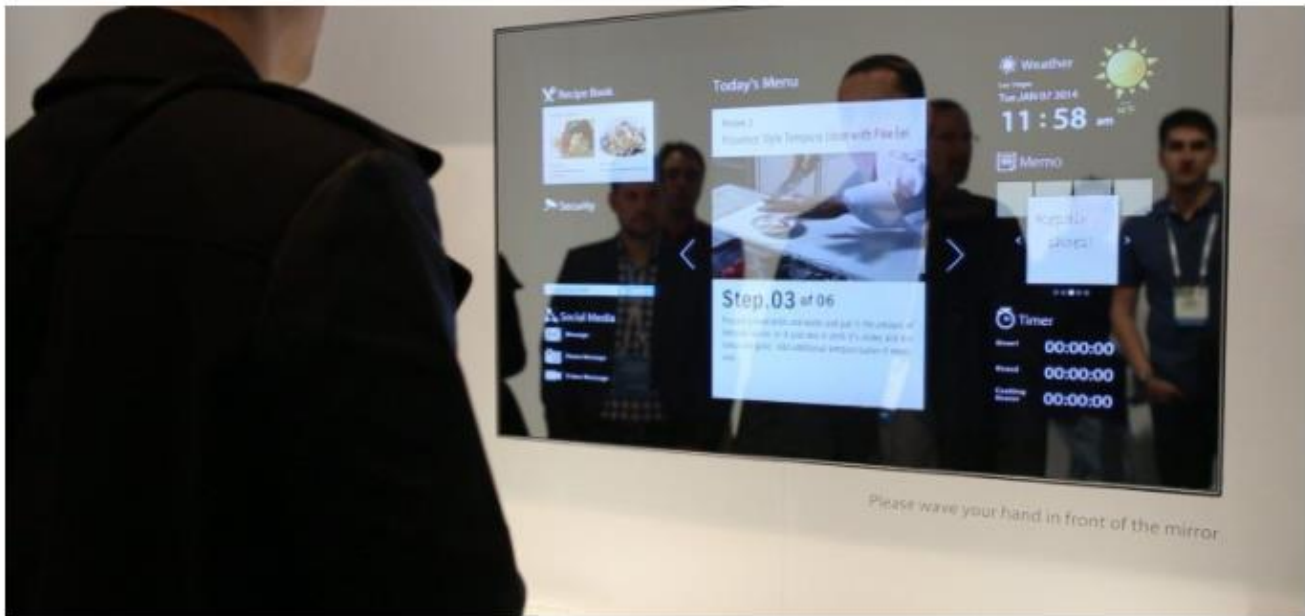


Figure 6 "Description of "Pixel" - Survey"

After this brief description, which leaves room for imagination, people were asked questions based on "Pixel" functionality.

To ask Pixel for time, you need to stay in front of the mirror, and ask a question similar to "What is the time?" Pixel will respond vocally and will display the time on the mirror surface. What do you think about this action?

67 responses



Figure 7 "Time functionality - Survey"

As seen in **Figure 7**, 3% of the people found this action too complicated, but most people found the action perfect or straightforward.

This question allowed the people to choose an answer that stated: "I would like the time displayed on the mirror all the time". Almost 36% voted for this answer, and this answer serves the user interface design.

To ask Pixel for the weather forecast, you need to stay in front of the mirror, and ask a question similar to "What is the weather outside?" Pixel will respond vocally and will display the weather forecast on the mirror surface. What do you think about this action?

67 responses

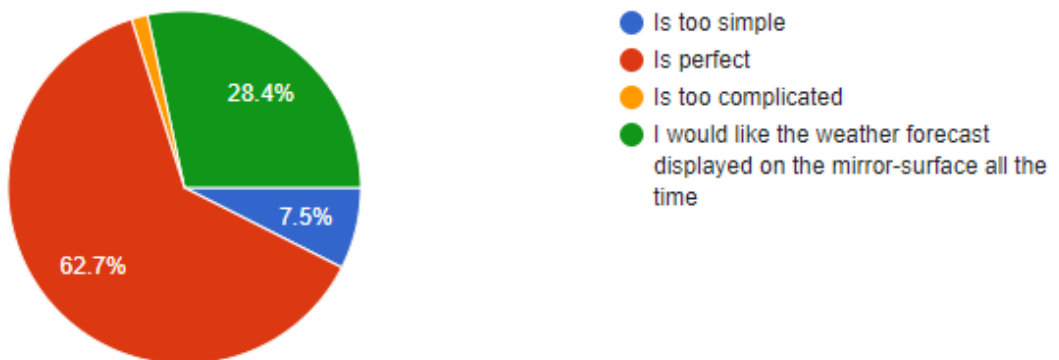


Figure 8 "Weather forecast functionality - Survey"

Figure 8 is very similar to the question in **Figure 7**, and their results are also very similar, resulting in a function seen by people as simple or straightforward.

To ask Pixel for a definition you need to stay in front of the mirror and ask a question. For example "What is a dinosaur?" Pixel will respond vocally and will display the answer on the mirror surface. What do you think about this action?

67 responses

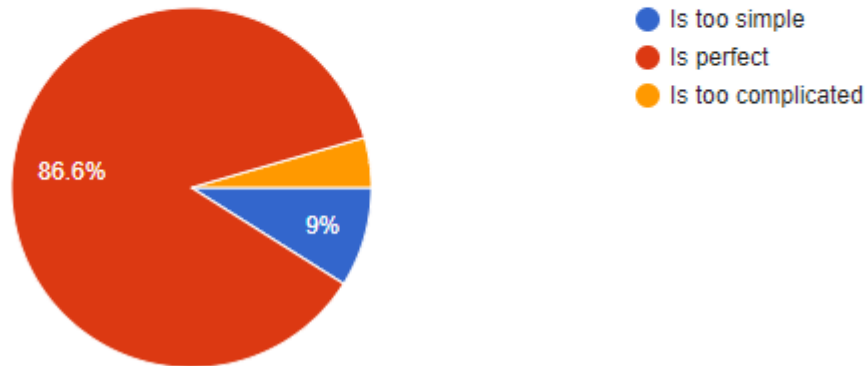


Figure 9 "Definition functionality - Survey"

Figure 9 shows that, as in the previous answers, 4.5% of people found this action too complicated to perform, but the majority voted the action as simple and straightforward.

If you stay silent in front of the mirror, Pixel will display on the mirror-surface a compliment or a positive quote. What do you think about this action?

67 responses

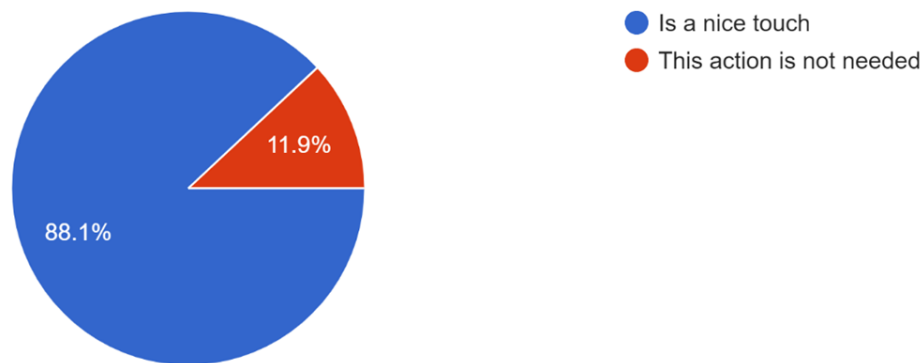


Figure 10 "Compliment action - Survey"

This question, seen in Figure 10, serves the user interface design and experience.

To ask Pixel for a joke, you need to stay in front of the mirror and ask a similar question to "Tell me a joke". The device will tell you vocally a joke and will display the joke as well on the mirror-surface.
67 responses

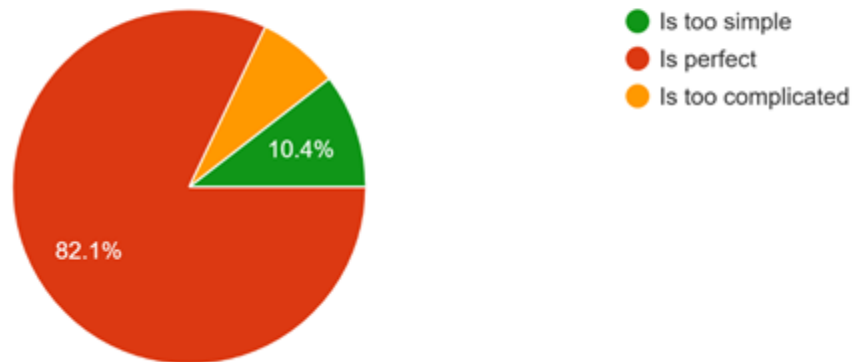


Figure 11 "Joke action - Survey"

As seen in **Figure 11**, this action was perceived as complicated by almost 8% of the people, where over 90% found the action straightforward or simple.

COMPLEX ACTIONS

To see if Pixel can recognize you, you need to stand in front of the mirror, and ask a similar question to "Who am I?" Pixel will let you know that you are not registered and does not know who you are. It will give you the option to register if you wish, by you vocally answering "yes".

67 responses

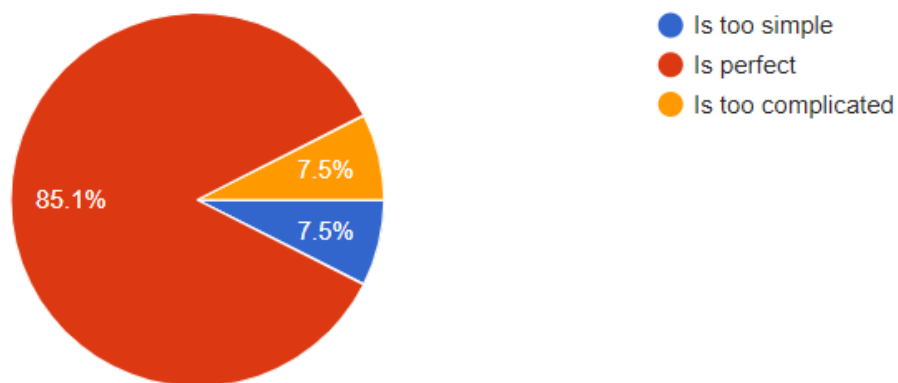


Figure 12 "Registration check - Survey"

For specific actions, the user needs to be registered. **Figure 12** question describe the user's process to check if it is registered on the device. As seen, 7.5% found the process too complicated, while the remaining found it straightforward or simple.

To register, all you need to do is to stay in front of the mirror, for 30 seconds, and perform different facial expressions (smiling, serious, surprised, confused). This way, Pixel will learn your face details and know who you are. At the end of the registration, Pixel will ask for your name, so it can address you by name in the future.

67 responses

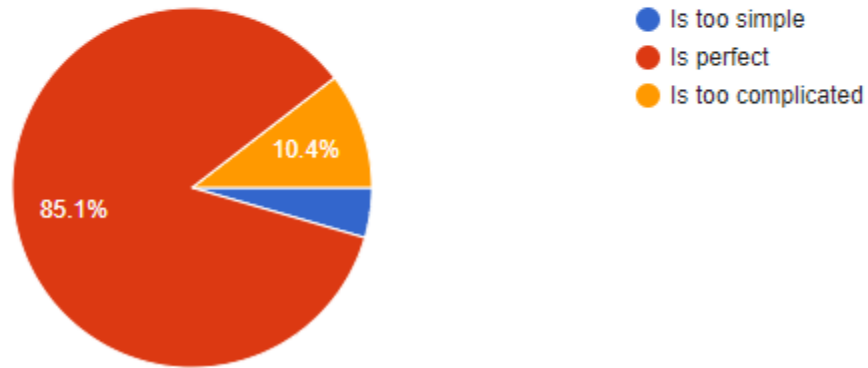


Figure 13 "Registration process - Survey"

The registration process was described as seen in **Figure 13**, and 10% found it too complicated. It is indeed an action that requires more complex tasks in contrast to the previous actions, but still, almost 90% of people found this process simple and straightforward.

As seen in **Figure 14** and **Figure 15** below, the next two questions focused on actions that need registration to be accessed. The majority of people found the registration needed to perform these actions, as these processes can contradict other user choices. For example, two users can create a list with the same name (grocery list, for example). Also, one user on a device might be very interested in fashion news and not interested in politics, while another user on the same device might feel the opposite.

Is useful to be registered on the device, as you can create lists to manage better your time. For example, you can create a grocery list with all the items you need to buy next time you go to the store. Would you register to get access to this functionality?

67 responses

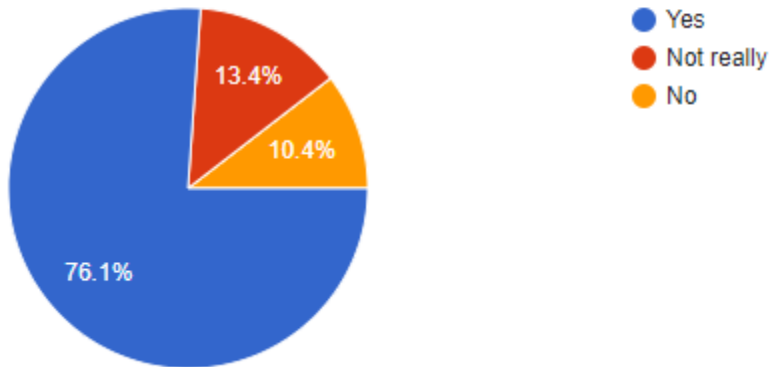


Figure 14 "Manipulating lists - Survey"

Another use of the registration is that notifications can be provided to you, about your favorite news, weather forecast, fashion, sports, and others. Would you register to get access to this functionality?

67 responses

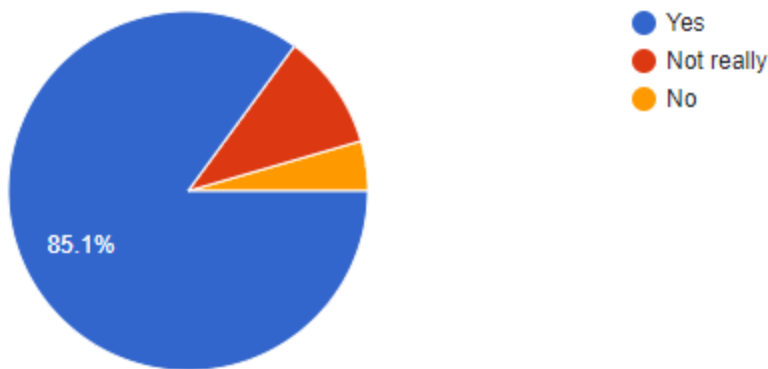


Figure 15 "Personalized output - Survey"

The statistics of the previous two questions from the survey show that over 76% of people consider that registration is necessary to keep their settings isolated from the other device users.

To connect Pixel to WIFI, a QR code is required. An internet connection is needed for Pixel to display the latest news and tell you the weather forecasts. QR code might not be something you know how to find on your phone. In this case, do you have a friend or relative that could help you with this process?

67 responses

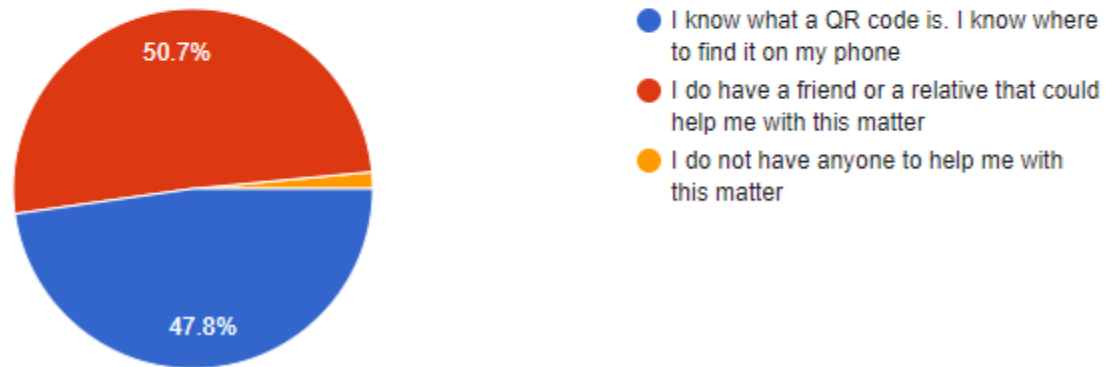


Figure 16 "WIFI connection - Survey"

WIFI connection is a must, as some actions are dependent on this connection. For example, the latest news or weather forecasts cannot be retrieved without a WIFI connection. As the device has only two input methods: the camera and the microphone, a QR code seemed to be the easiest way to establish an internet connection.

As just the term "QR code" is not very popular among people with basic or no computer skills, there is a need for another person with stronger computer skills to connect "Pixel" to the internet. This survey question was designed to observe if the users know how to access the WIFI QR code or have a person to help with this matter.

The result of this survey question was positive, as seen in **Figure 16**, as almost 48% of the people questioned knew where to find the QR code on their phone, and 50% have a friend or a relative that can help with this action. 1.5% answered that they do not know how to find the QR code and/or do not have a person to help them with this matter.

Keeping in mind how Pixel communicates with you: Is there anything you would change to make Pixel facilitate you in a different, easier way?

67 responses

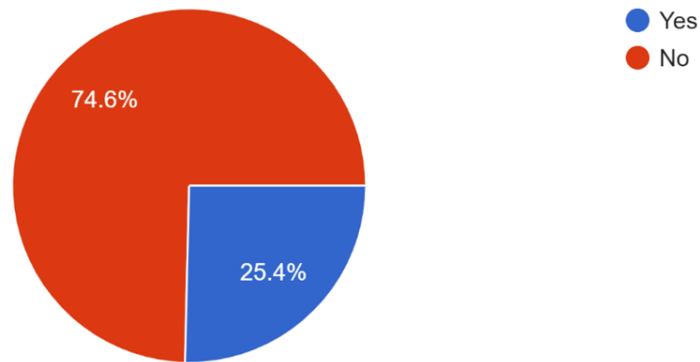


Figure 17 "Pixel improvement - Survey"

As seen in **Figure 17**, almost 26% of people felt that "Pixel" could be improved with different functionalities:

- Cooking receipts
- Voice recognition for simple tasks such as: get the time, get the forecast
- Calendar
- Music
- Others

These characteristics are considered depending on how the device's development progresses in the time frame allocated for this project.

Would you like Pixel Virtual Assistant to become the magic mirror, in your house?

67 responses

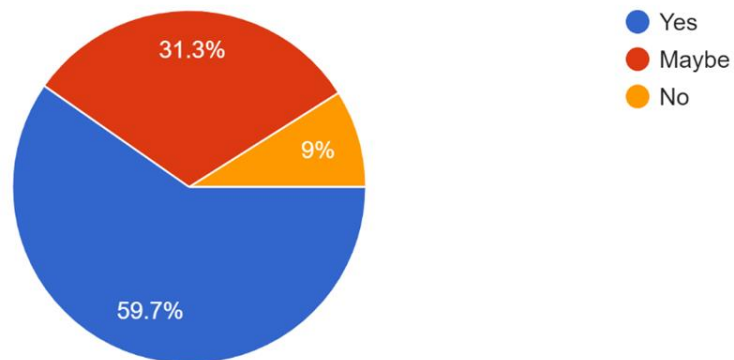


Figure 18 "Pixel impact - Survey"

As seen in **Figure 18**, "Pixel" is desired by almost 60% as a smart speaker encapsulated in a mirror in people's homes, which shows that the project's concept positively impacted the people questioned.

Also, this survey was designed to check if the device's functionality would positively impact users of different ages and different background skills.

The full survey results can be found at

https://docs.google.com/forms/d/1MbV3gCqHmoCn9v5nxpeJhIM9SXFndIKtY0_heSfgME4/edit#responses

USERS

"Pixel's" Virtual Assistant variety of users is narrowed down to two types: registered and unregistered members. The number of registered members is limited by its storage only.

The virtual assistant's end-users fall into two main categories: registered and unregistered.

UNREGISTERED USERS

"Pixel" Virtual Assistant can be used by any person that steps in front of the mirror. The device awakes the virtual assistant when a face is detected and is ready to receive requests and provide meaningful answers.

Queries about the weather, calling different locations, ask definitions, get Covid-19 statistics, get details about different locations and access the help menu can be accessed by any type of user.

REGISTERED USERS

70% of the virtual assistant's skills are available for both categories of users, but few features can be accessed only by registered users.

Creating different lists (e.g., shopping list), creating reminders, saving a next of kin contact, and sending an SOS can be accessed only by registered users. If an unregistered user requires the device to create a list, the device will inform the user to registers before proceeding.

These skills require registration to execute so that each user can have personal lists, reminders, and next of kin contact. Allowing these skills to be accessed by unregistered users could result in conflicting list names, reminders and of course, next of kin contact would be a problem, as when sending an emergency text, the device would not know to inform the next of kin who is in trouble, as it recognizes different faces by 128-d embeddings.

EXAMPLE

Imagine Angela, a 60 years old female that lives with her husband. Her husband, Jack, is hearing-impaired. Angela possesses a phone to chat on chat applications and watch YouTube recipe videos. Every time their daughter is visiting, Angela and Jack ask her to look at their devices that require attention, such as the TV that suddenly does not have sound or at their mobile phones that changed the text size. Therefore, Angela and Jack have the necessary computer skills to use smart devices, but when things get complicated, they need help from another person.

Their daughter bought as a Christmas present a virtual assistant encapsulated in a mirror that was installed in the bathroom.

Every morning when Angela goes to the bathroom to wash her teeth, face and put on her makeup, the mirror activates and greets Angela. Angela asks for the latest Covid-19 statistics in their area, as this is one of the most important news of the world at the moment. The mirror informs Angela about the number of new cases and deaths.

Angela asks the mirror how the weather would be that day. The mirror will inform Angela about the weather condition, also displaying the information on the mirror surface.

During his morning routine, Jack requires the mirror to call different locations to see if they are opened during the lockdown. He is a regular customer of several shops around the area, and he is anxious to find out when they will re-open their doors for customers again.

During the evening routine, the action repeats when Angela uses the mirror to modify her lists with things to do the next day. Jack does not use as much of the mirror during the evening routine.

Both spouses love how they get informed during their morning and evening routine, and they want to purchase another mirror so that they can interact with it outside the bathroom during the day.

From this example, it can be observed that the mirror benefits both spouses. Angela and Jack have difficulties in technology, but the mirror helps them stay engaged during the conversation, and it is easy to use at any time.

MAIN FUNCTIONALITIES

The functionalities of "Pixel" Virtual Assistant are very simple, as seen in **Figure 19**.

To initiate a dialog with the virtual assistant, the member needs to step in front of the mirror, a moment in which the camera scans the user's face and is ready to listen for a query. The list of queries contains a wider variety of functionalities for registered users than unregistered users. The registered members create personalized lists, reminders, next of kin contacts and send emergency messages.

A member is required to register just once, to access certain functionalities. Once the member is registered, it is recognized by the device every time it stays in front of the device.

Figure 19 provides a brief description of how the member can interact with the device, showcasing the system's functionality:

- Initiate interaction
- Query

As seen in **Figure 19**, there are five external systems that "Pixel" relies on:

- OpenCV for face recognition and facial detection
- STT for transforming speech to text
- gTTS to convert text to speech using Google's API
- GPS/GSM Module
- Cloud storage

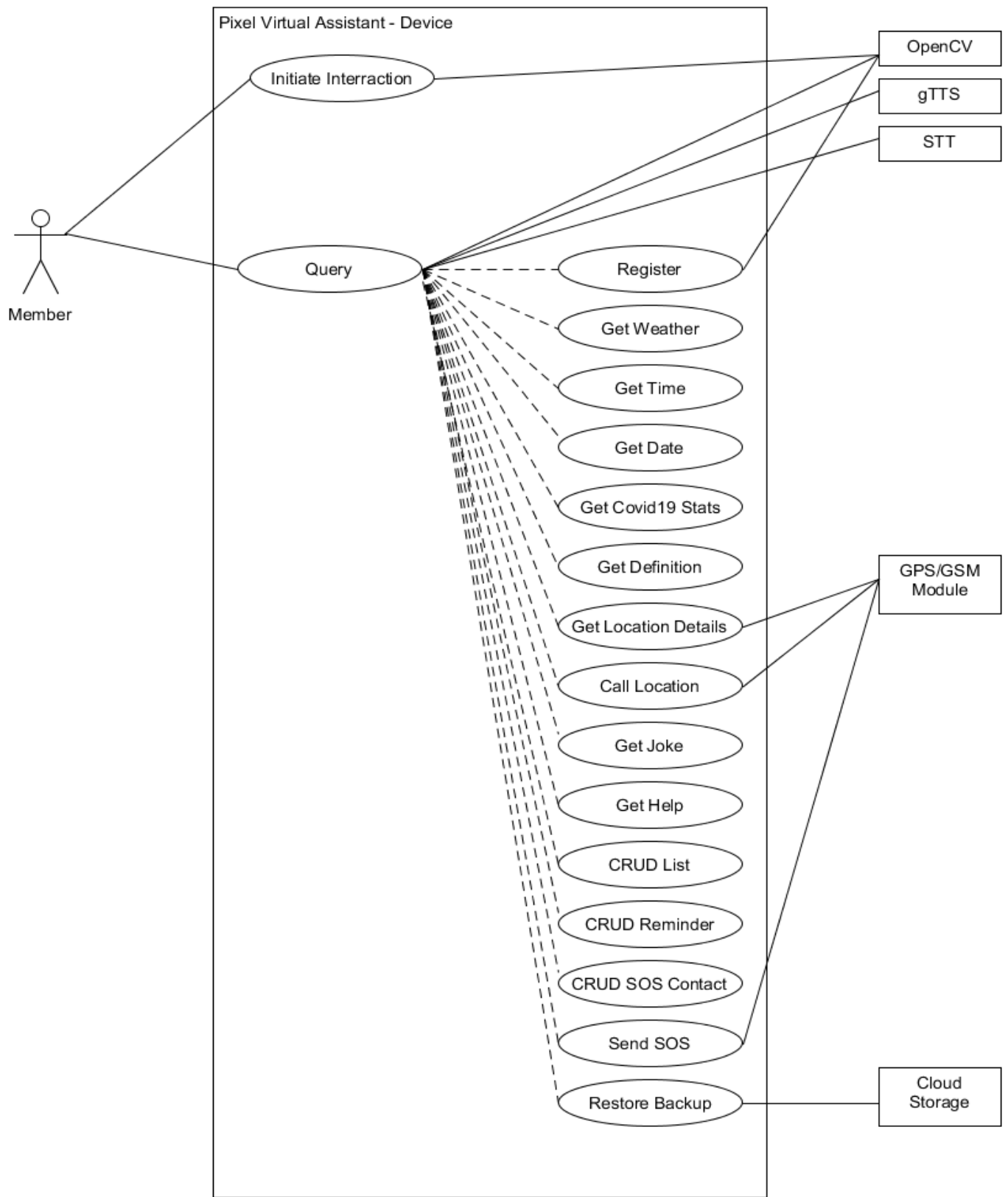


Figure 19 "Pixel Virtual Assistant use case diagram"

Source: Theodora Tataru, 2021

INITIATE INTERACTION

To begin a dialog with the device, the member needs to initiate an interaction with it.

The name of the functionality makes the action sound more complicated than it actually is. To initiate interaction, the member just needs to stand in front of the mirror, and within seconds, the mirror recognizes its presence. This action activates the virtual assistant to be ready to receive a request from the member.

The detailed use case can be seen in **Table 1**.

Name	Initiate interaction
Actors	Member
Preconditions	1. The device needs to be plugged in
Main Success Scenario	1. The member steps in front of the device 2. The device displays the time 3. The device greets the member vocally 4. The device displays the greeting on the mirror-surface
Alternatives	1a. The member stepped in front of the mirror and moved away too quick 1. The device goes on standby
Postconditions	1. The device is ready to receive a query from the member 2. The device stays active for a delimited amount of time

Table 1 "Initiate interaction use case"

Source: Theodora Tataru, 2021

After the user member initiates an interaction with the device, the device is ready to receive requests.

REGISTER

The first use case described in **Table 2** refers to the register functionality.

The use case starts when the member wishes to register on the device. The member queries the device to begin the registration with a question similar to: "Pixel, I want to register " or "Pixel, I want to sign up". Once the request was addressed to the device, the device gives the member the instructions that need to be followed to perform the registration process.

When the registration finishes, the device confirms the registration to the member vocally and displays the appropriate message confirmation on the mirror surface.

Registration is not required to interact with the device but enables the member to personalize settings. Also, the registration provides more functionality.

Name	Registration
Actors	Member
Preconditions	<ol style="list-style-type: none">1. The member is not registered2. The member needs to initiate interaction
Main Success Scenario	<ol style="list-style-type: none">1. The member requests the device to initiate registration2. The device asks the member vocally to pronounce his name3. The device displays on the mirror surface the message asking for the member's name4. The member pronounces his name5. The device asks the member vocally if the name is correct6. The device displays on the mirror surface the appropriate message7. The member confirms that the name is correct8. The device informs the member vocally that for the next 5 seconds, it will take pictures of him/her assuring the member that the pictures will be deleted at the end9. The device displays on the mirror surface the appropriate message10. The device informs the member vocally that registration begins11. The device displays on the mirror surface the appropriate message12. The device informs the member vocally how the registration is progressing during the registration process13. The device displays on the mirror surface how the registration is progressing during the registration process14. The device informs the member vocally that the process of registration is finished15. The device displays on the mirror surface the appropriate message
Alternatives	<ol style="list-style-type: none">2a. The member is already registered<ol style="list-style-type: none">1. The device informs the user that he/she is already registered2. The device is ready for another query6a. The member denies the name displayed on the mirror-surface<ol style="list-style-type: none">1. The device asks the member vocally to pronounce the name again2. The device displays the instruction on the mirror-surface3. The member pronounces his name again6b. The device informs the member vocally that the name is already in use<ol style="list-style-type: none">1. The device displays on the mirror surface that the name is already in use2. The device asks member vocally for a different name3. The device displays on the mirror surface the appropriate message4. The member provides a different name for the registrations6c. The member decides to cancel the registration<ol style="list-style-type: none">1. The device confirms that the registration is canceled

	<ol style="list-style-type: none"> 2. The device displays on the mirror surface the appropriate message 3. The device is ready for a query from the member until the member moves away from the mirror
Postconditions	<ol style="list-style-type: none"> 1. The new member is registered on the device 2. The device is ready for a query from the member until the member moves away from the mirror

Table 2 "Registration use case"

Source: Theodora Tataru, 2021

GET WEATHER

The use case in **Table 3** begins when the member queries the device about weather updates. To get the updates, the member must ask a question similar to "Pixel, how is the weather in Dublin?" or "Pixel, what is the temperature outside?". The device processes the query and displays the weather forecast on the mirror surface while narrating it simultaneously.

The detailed use case is presented in **Table 3**.

Name	Get Weather
Actors	Member
Preconditions	1. The member needs to initiate interaction before starting this query
Main Success Scenario	1. The member asks for weather conditions 2. The device processes the request 3. The device responds back by voice the weather updates 4. The device displays the weather updates on the mirror-surface
Alternatives	2a. The device does not recognize the location requested by the member 1. The device informs the member vocally that it did not understand the request 2. The device displays on the mirror surface that it did not understand the request
Postconditions	1. The weather updates are displayed on the mirror surface until another query is requested or the member leaves

Table 3 "Get weather use case"

Source: Theodora Tataru, 2021

GET TIME

The use case in **Table 4** describes how the member can query the time using the device.

The use case begins after the member has initiated an interaction and queries the time with a similar question: "Pixel, what is the time?" or "Pixel, tell me the time". The device processes the query and displays the time on the mirror surface while narrating it simultaneously.

The detailed use case can be observed in **Table 4**.

Name	Get Time
Actors	Member
Preconditions	1. The member needs to initiate interaction before starting this query
Main Success Scenario	1. The member asks the time 2. The device responds vocally to the member the current time 3. The device displays on the mirror surface the current time
Alternatives	-
Postconditions	1. The time is displayed on the mirror surface until another query is requested or the member leaves

Table 4 "Get time use case"

Source: Theodora Tataru, 2021

GET DATE

The use case in **Table 5** describes how the member can query the current date using the device.

The use case begins after the member has initiated an interaction and queries the date with a similar question: "Pixel, what is the date?" or "Pixel, tell me today's date". The device processes the query and displays the date on the mirror surface while narrating it simultaneously.

The detailed use case can be observed in **Table 5**.

Name	Get Date
Actors	Member
Preconditions	1. The member needs to initiate interaction before starting this query
Main Success Scenario	1. The member asks what the date is 2. The device responds vocally to the member the date 3. The device displays on the mirror surface the date
Alternatives	-
Postconditions	1. The date is displayed on the mirror surface until another query is requested or the member leaves

Table 5 "Get date use case"

Source: Theodora Tataru, 2021

GET COVID STATS

The use case described in **Table 6** refers to the COVID stats functionality.

The use case begins after the member has initiated an interaction and queries the COVID stats for a particular country, with a similar question "Pixel, how are covid-19 statistics in Ireland?" or "Pixel, tell me about covid-19 in France". The device processes the query and displays the COVID-19 statistics on the mirror surface while narrating it simultaneously.

The detailed use case can be observed in **Table 6**.

Name	Get COVID stats
Actors	Member
Preconditions	1. The member needs to initiate interaction before starting this query
Main Success Scenario	1. The member asks for Covid-19 statistics for a particular country 2. The device processes the request 3. The device responds back by voice the covid-19 statistics for that particular country 4. The device displays the covid-19 statistics on the mirror-surface
Alternatives	2a. The device does not recognize the location requested by the member 1. The device informs the member vocally that it did not understand the request 2. The device displays on the mirror surface the appropriate message
Postconditions	1. The COVID-19 statistics requested are displayed on the mirror-surface until the member queries a new request or the member leaves

Table 6 "Get COVID stats use case"

Source: Theodora Tataru, 2021

Table 7 describes in detail how a member can request the device for help.

To begin this use case, the member has to initiate an interaction with the device first. To query help, the member needs to ask a query similar to "Pixel, help me" , "Pixel, can you help me with the weather?" or "Pixel, how do I register?". The device processes the request and displays on the mirror surface instructions on how to interact with the device.

The detailed use case can be seen in **Table 7**.

Name	Get Help
Actors	Member
Preconditions	1. The member needs to initiate interaction before starting this query
Main Success Scenario	<ol style="list-style-type: none"> 1. The member asks the device for help 2. The device processes the request 3. The device vocally informs the member about multiple ways of interaction 4. The device displays on the mirror-surface instructions on how to interact with the device.
Alternatives	-
Postconditions	1. The help instructions are displayed on the screen until another query is requested or the member leaves

Table 7 "Get help use case"

Source: Theodora Tataru, 2021

GET DEFINITION

Table 8 describes the steps the member needs to follow to request a definition from the device.

To begin this use case, the member has to initiate an interaction with the device first. To query a definition, the member needs to ask a question similar to "Pixel, define dinosaur" or "Pixel, tell me about dinosaurs". The device processes the request and displays the definition on the mirror surface while narrating it.

The detailed use case can be seen in **Table 8**.

Name	Get Definition
Actors	Member
Preconditions	1. The member needs to initiate interaction before starting this query
Main Success Scenario	1. The member asks for a definition 2. The device processes the request 3. The device narrates the definition vocally 4. The device displays on the mirror surface the definition
Alternatives	-
Postconditions	1. The definition is displayed on the screen until another query is requested or the member leaves

Table 8 "Get definitions use case"

Source: Theodora Tataru, 2021

GET LOCATION

Table 9 describes the steps the member needs to follow to request location details from the device.

To begin this use case, the member has to initiate an interaction with the device first. To query a location, the member needs to ask a question similar to "Pixel, where is the cinema in Kilkenny?", "Pixel, what are the opening hours for SuperValu in Carlow?" or "Pixel, give me the address of Institute of Technology Carlow". The device processes the request and displays the location details on the mirror surface while narrating it.

The detailed use case can be seen in **Table 9**.

Name	Get Location
Actors	Member
Preconditions	1. The member needs to initiate interaction before starting this query
Main Success Scenario	<ol style="list-style-type: none">1. The member asks the device to provide details for a particular location2. The device processes the request3. The device vocally informs the member if the location is open at that particular moment in time, provides the location's main phone number and other details4. The device displays the address, contact information, and the opening hours for the entire week for that particular location
Alternatives	<ol style="list-style-type: none">2a. The device does not find the exact location the member requested for<ol style="list-style-type: none">1. The device finds a different location that has similar characteristics to the location requested initially by the member
Postconditions	1. The location details are displayed on the mirror surface until the member queries a new request or the member leaves

Table 9 "Get location use case"

Source: Theodora Tataru, 2021

CALL LOCATION

Table 10 describes the steps the member needs to follow to call a particular location from the device.

To begin this use case, the member has to initiate an interaction with the device first. To call a location, the member needs to ask a question similar to "Pixel, I want to call the cinema in Kilkenny" or "Pixel, call Institute of Technology Carlow". Once the request was addressed to the device, the device provides on the mirror surface while narrating the details about the location and requires permission from the member to proceed with the call.

The detailed use case can be seen in **Table 10**.

Name	Call location
Actors	Member
Preconditions	1. The member needs to initiate interaction before starting this query
Main Success Scenario	<ol style="list-style-type: none"> 1. The member asks the device to call a particular location 2. The device processes the request 3. The device narrates the details of the location requested 4. The device displays the appropriate message on the mirror-surface 5. The device request vocally the member for permission to create the call 6. The device displays the appropriate message on the mirror-surface 7. The member confirms that the device should proceed with the call 8. The device processes and confirms vocally the creation of the phone call 9. The device displays the appropriate message on the mirror-surface 10. The device waits for the phone call to finish 11. When the phone conversation is finished, the device confirms the end of the call
Alternatives	<ol style="list-style-type: none"> 7a. The member denies the call <ol style="list-style-type: none"> 1. The device is ready for a new query until the member moves away from the mirror
Postconditions	2. The location details are displayed on the mirror surface until the member queries a new request or the member leaves

Table 10 "Call location use case"

Source: Theodora Tataru, 2021

GET JOKE

Table 11 describes how the member can request a joke from the device.

To perform this query, the member needs to initiate an interaction first and then request the device for a joke with a similar question: "Pixel, tell me a joke" or "Pixel, do you know a good joke?". The device processes the request and displays the answer on the mirror surface while narrating it.

The detailed use case can be seen in **Table 11**.

Name	Get Joke
Actors	Member
Preconditions	1. The member needs to initiate interaction before starting this query
Main Success Scenario	1. The member asks for a joke 2. The device processes the request 3. The device responds vocally with a joke 4. The device displays the joke on the mirror-surface
Alternatives	-
Postconditions	3. The joke is displayed on the mirror surface until the member requests a new query or leaves

Table 11 "Get joke use case"

Source: Theodora Tataru, 2021

CREATE LIST

Any query involving a list requires the member requesting the list manipulation to be registered.

To create an example, the grocery list was used as a showcase, but the members are not restricted to this list. A member can create multiple lists, and their naming is up to the member's decision.

To begin this query, the member has to initiate the interaction first and needs to be registered.

The use case begins when the member asks a similar question to "Pixel, create a new list" or "Pixel, create a grocery list". The device asks for more details, and when the member provides all details, the device processes the query. When the list is successfully created, the device confirms the creation of the list vocally and is displayed on the mirror surface.

The detailed use case can be seen in **Table 12**.

Name	Create a list
Actors	Member
Preconditions	<ol style="list-style-type: none">1. The member needs to initiate interaction before starting this query2. The member needs to be registered
Main Success Scenario	<ol style="list-style-type: none">1. The member asks the device to create a new list2. The device asks vocally for the name of the list3. The device displays the appropriate message on the mirror-surface4. The member pronounces the name of the list5. The device requests vocally confirmation for the name of the list6. The device displays the appropriate message on the mirror-surface7. The member confirms the name of the list8. The device vocally confirms the creation of the list9. The device confirms the creation of the list on the mirror-surface
Alternatives	<ol style="list-style-type: none">2a. The member is not registered<ol style="list-style-type: none">1. The device informs the user that it needs to be registered to proceed7a. The member denies the name of the list<ol style="list-style-type: none">1. The device requests the member vocally the name of the list2. The device displays the appropriate message on the mirror-surface3. The member pronounces the name of the list7b. The device identifies another list with the same name<ol style="list-style-type: none">1. The device informs the member vocally about the existence of another list with the same name2. The device displays the appropriate message on the mirror-surface
Postconditions	<ol style="list-style-type: none">1. The list was created, and the member can add items to the list2. The name of the list is displayed on the mirror surface until the member requests a new query or leaves

Table 12 "Create list use case"

Source: Theodora Tataru, 2021

RETRIEVE A LIST

To retrieve a list and its items, the member needs to initiate an interaction and ask a question similar to "Pixel, show me the grocery list" or "Pixel I want to see the grocery list". The device processes the request and displays the list and its items on the mirror surface, along with vocal narration

The detailed use case can be seen in **Table 13**.

Name	Retrieve list
Actors	Member
Preconditions	<ol style="list-style-type: none">1. The member needs to initiate interaction before starting this query2. The member needs to be registered
Main Success Scenario	<ol style="list-style-type: none">1. The member requests to retrieve a particular list2. The device requests confirmation for the name of the list3. The device displays the appropriate message on the mirror-surface4. The member confirms the name of the list5. The device processes the request6. The device enumerates the list items vocally7. The device displays the list of items on the mirror-surface
Alternatives	<ol style="list-style-type: none">2a. The member is not registered<ol style="list-style-type: none">1. The device informs the user that it needs to be registered to proceed5a. The list pronounced by the member does not exist<ol style="list-style-type: none">1. The device informs the member that the list does not exist2. The device displays on the mirror surface the appropriate message
Postconditions	<ol style="list-style-type: none">1. The item lists are displayed on the mirror surface until the member queries a new request or the member leaves

Table 13 "Retrieve list use case"

Source: Theodora Tataru, 2021

RETRIEVE ALL LISTS

To retrieve all lists, the member needs to initiate an interaction and ask a question similar to "Pixel, show me all my lists" or "Pixel, what lists do I have". The device processes the request and displays the lists on the mirror surface, along with vocal narration.

The detailed use case can be seen in **Table 14**.

Name	Retrieve all list
Actors	Member
Preconditions	<ol style="list-style-type: none">1. The member needs to initiate interaction before starting this query2. The member needs to be registered
Main Success Scenario	<ol style="list-style-type: none">1. The member requests to retrieve all lists2. The device processes the request3. The device enumerates the lists available vocally4. The device displays all member's lists on the mirror-surface
Alternatives	<ol style="list-style-type: none">2a. There are no lists to retrieve<ol style="list-style-type: none">1. The device informs the member that there are no lists to be retrieved2. The device displays on the mirror surface the appropriate message3. The device is ready for another query from the member until the member moves away from the mirror2b. The member is not registered<ol style="list-style-type: none">1. The device informs the user that it needs to be registered to proceed
Postconditions	<ol style="list-style-type: none">1. The lists are displayed on the mirror surface until the member queries a new request or the member leaves

Table 14 "Retrieve all lists use case"

Source: Theodora Tataru, 2021

ADD ITEM TO LIST

To add an item to an existing list, the member first needs to initiate the interaction and then ask a question similar to “Pixel, add butter on the grocery list” or “Pixel put butter on my grocery list”. The device might ask for other details, and once the member provides these, the query is processed. When the item is successfully added to the list, the device confirms the action vocally to the member, along with the confirmation displayed on the mirror surface.

The detailed use case can be seen in **Table 15**.

Name	Add item to list
Actors	Member
Preconditions	<ol style="list-style-type: none"> 1. The member needs to initiate interaction before starting this query 2. The member needs to be registered
Main Success Scenario	<ol style="list-style-type: none"> 1. The member requests the device to add an item to a list 2. The device repeats the name of the list and the name of the item that needs to be added to the list 3. The device displays on the screen the name of the list and the name of the new item to be added 4. The device request vocally confirmation to add the item to the list 5. The device displays on the mirror surface the appropriate message 6. The member confirms the action 7. The device adds the new item on the particular list 8. The device confirms vocally the addition of the new item to the list 9. The device displays the items on the list on the mirror-surface
Alternatives	<ol style="list-style-type: none"> 2a. The member is not registered <ol style="list-style-type: none"> 1. The device informs the user that it needs to be registered to proceed 2b. The list where the item needs to be added does not exist <ol style="list-style-type: none"> 1. The device informs the member vocally that the list does not exist 2. The device displays on the mirror surface the appropriate message 3. The device asks the member if the list should be created 4. The member confirms the creation of the list 5. The device creates the list and adds the item to the list 6. The device confirms vocally that the list was created and the item was added to it 7. The device displays the appropriate message on the mirror-surface 4a. The member denies the action <ol style="list-style-type: none"> 1. The device is ready for another query from the member until the member moves away from the mirror
Postconditions	<ol style="list-style-type: none"> 1. The item is added to the list 2. The confirmation is displayed on the mirror surface until the member queries a new request or the member leaves

Table 15 "Add item to list use case"

Source: Theodora Tataru, 2021

DELETE ITEM FROM LIST

To delete an item from a list, the member must first initiate the interaction with the device, followed by request similar to "Pixel, delete butter from the grocery list" or "Pixel remove butter from the grocery list". The device might ask for some details for clarification, and once the member provides the details, the device processes the request. When the request was successfully processed, and the item was deleted from the list, the device confirms the procedure to the member vocally and displayed it on the mirror surface

The detailed use case can be seen in **Table 16**.

Name	Delete item from list
Actors	Member
Preconditions	<ol style="list-style-type: none">1. The member needs to initiate interaction before starting this query2. The member needs to be registered
Main Success Scenario	<ol style="list-style-type: none">1. The member asks the device to delete an item from a list2. The device repeats the name of the list and the item that needs to be deleted3. The device displays the list name and the item that is requested to be deleted4. The device asks vocally for confirmation5. The device displays the appropriate message on the mirror-surface6. The member confirms the deletion7. The device deletes the item from the list8. The device confirms the deletion of the item vocally9. The device displays the confirmation on the mirror-surface
Alternatives	<ol style="list-style-type: none">2a. The member is not registered<ol style="list-style-type: none">1. The device informs the user that it needs to be registered to proceed4a The device informs the member that the list does not exist<ol style="list-style-type: none">1. The device informs the member vocally that the list does not exist2. The device displays the appropriate message on the mirror-surface4b The device informs the member vocally that the item does not exist in that particular list<ol style="list-style-type: none">1. The device displays the appropriate message on the mirror-surface6a. The member denies the confirmation of the deletion<ol style="list-style-type: none">1. The device is ready for another query from the member until the member moves away from the mirror
Postconditions	<ol style="list-style-type: none">1. The confirmation of deletion is displayed on the mirror surface until the member queries a new request or the member leaves

Table 16 "Delete item from list use case"

Source: Theodora Tataru, 2021

DELETE LIST

To delete a list, the member must first initiate the interaction and then asks the device a similar query to "Pixel, delete the grocery list" or "Pixel I want to delete the grocery list". The device asks for the deletion confirmation, and when the confirmation is given, the device processes the request confirming back to the member the deletion of the list. The confirmation is narrated to the member and displayed on the mirror surface.

The detailed use case can be seen in **Table 17**.

Name	Delete list
Actors	Member
Preconditions	<ol style="list-style-type: none">1. The member needs to initiate interaction before starting this query2. The member needs to be registered
Main Success Scenario	<ol style="list-style-type: none">1. The member requests the device to delete a list2. The device request the name of the list to be deleted3. The device displays the appropriate message on the mirror-surface4. The member provides the name of the list5. The device vocally repeats the name of the list to the member requesting confirmation6. The device displays on the mirror surface the name of the list with the appropriate message7. The member confirms the name of the list8. The device deletes the list9. The device confirms the deletion of the list vocally10. The device displays the confirmation on the mirror-surface
Alternatives	<ol style="list-style-type: none">2a. The member is not registered<ol style="list-style-type: none">1. The device informs the user that it needs to be registered to proceed7a. The member denies the name of the list<ol style="list-style-type: none">1. The device request again the name of the list2. The device displays the appropriate message on the mirror-surface8a. The device informs the member that the list does not exist<ol style="list-style-type: none">1. The device informs the member vocally that the list does not exist2. The device displays the appropriate message on the mirror-surface
Postconditions	<ol style="list-style-type: none">1. The confirmation of deletion is displayed on the mirror surface until the member queries a new request or the member leaves

Table 17 "Delete list use case"

Source: Theodora Tataru, 2021

RESTORE BACKUP

The member can restore a backup of the device in case of failure or if the device is changed, and this action requires that the member to be registered on the device.

The use case begins after the user had initiated an interaction and asks a similar question to "Pixel, restore" or "Pixel, is any backup available". The device displays the available backups and asks the user what restore point is the one desired. The member answers accordingly. The device informs via voice and via the mirror-surface that some information might be lost and asks for confirmation to proceed. When the member confirms the action, the device restarts and restores the selected backup.

The detailed use case can be seen in **Table 18**.

Name	Restore Backup
Actors	Member
Preconditions	<ol style="list-style-type: none">1. The member needs to stand in front of the mirror2. The member needs to be registered
Main Success Scenario	<ol style="list-style-type: none">1. The member requires the device to restore a backup2. The device informs the member vocally if a backup is available3. The device displays on the mirror surface the available backup4. The device informs the member vocally that certain information might be lost5. The device displays on the mirror surface the appropriate message6. The device asks for confirmation before proceeding7. The device displays on the mirror surface the appropriate message8. The member confirms the action9. The device processes and installs the backup10. The device informs vocally the member that the backup had been restored11. The device displays on the mirror surface the appropriate message
Alternatives	<ol style="list-style-type: none">2a. The member is not registered<ol style="list-style-type: none">1. The device informs the user that it needs to be registered to proceed2b. The device informs the member vocally that there is no backup available<ol style="list-style-type: none">1. The device informs vocally that there is no backup available to restore2. The device displays on the mirror surface the appropriate message3. The device is ready for a new query8a. The member does not confirm the action<ol style="list-style-type: none">1. The device confirms vocally that the backup was not restored2. The device displays the appropriate message on the mirror-surface
Postconditions	<ol style="list-style-type: none">1. The confirmation of restoring and installing the backup is displayed on the mirror surface until the member queries a new request or the member leaves

Table 18 "Restore backup use case"

Source: Theodora Tataru, 2021

CREATE REMINDER

To begin this query, the member has to initiate the interaction first and needs to be registered.

The use case begins when the member asks a similar question to "Pixel, create a reminder" or "Pixel, remind me to take my medicine". The device asks for more details, and when the member provides all details, the device processes the query. When the reminder is successfully created, the device confirms the creation of the reminder vocally and is displayed on the mirror surface.

The detailed use case can be seen in **Table 19**.

Name	Create Reminder
Actors	Member
Preconditions	<ol style="list-style-type: none"> 1. The member needs to initiate interaction before starting this query 2. The member needs to be registered
Main Success Scenario	<ol style="list-style-type: none"> 1. The member asks the device to create a reminder 2. The device requests vocally the name for the reminder 3. The device displays the appropriate message on the mirror-surface 4. The member provides the name for the reminder 5. The device request vocally the confirmation for the name of the reminder 6. The device displays the appropriate message on the mirror-surface 7. The member confirms the name of the reminder 8. The device vocally ask the member for what date the reminder is to be set up 9. The device displays the appropriate message on the mirror-surface 10. The member communicates to the device the specific date 11. The device request vocally the confirmation for the date of the reminder 12. The device displays the appropriate message on the mirror-surface 13. The member confirms the date of the reminder 14. The device vocally ask the member for what time the reminder is to be set up 15. The device displays the appropriate message on the mirror-surface 16. The member communicates to the device the specific time 17. The device request vocally the confirmation for the time of the reminder 18. The device displays the appropriate message on the mirror-surface 19. The member confirms the time of the reminder 20. The device processes the information and creates the reminder 21. The device informs the member vocally that the reminder has been created 22. The device displays on the mirror surface the appropriate message
Alternatives	<ol style="list-style-type: none"> 7a. The member denies the name of the reminder <ol style="list-style-type: none"> 1. The device request vocally the member to provide a name for the reminder 2. The device displays the appropriate message on the mirror-surface 3. The member provides a new name for the reminder 13a. The member denies the date of the reminder <ol style="list-style-type: none"> 1. The device request vocally the member to provide a date for the reminder 2. The device displays the appropriate message on the mirror-surface 3. The member provides a new date for the reminder 19.a The member denies the time of the reminder <ol style="list-style-type: none"> 1. The device request vocally the member to provide a time for the reminder 2. The device displays the appropriate message on the mirror-surface 3. The member provides a new time for the reminder

Postconditions	<ol style="list-style-type: none">1. The reminder was created2. The reminder is displayed on the mirror surface until the member requests a new query or leaves

Table 19 "Create reminders use case"

Source: Theodora Tataru, 2021

RETRIEVE REMINDER

To retrieve a reminder, the member needs to initiate an interaction and ask a question similar to "Pixel, show me the reminders for tomorrow" or "Pixel, what reminder do I have today". The device processes the request and displays the reminders on the mirror surface along with vocal narration.

The detailed use case can be seen in **Table 20**.

Name	Retrieve Reminder
Actors	Member
Preconditions	<ol style="list-style-type: none">1. The member needs to initiate interaction before starting this query2. The member needs to be registered
Main Success Scenario	<ol style="list-style-type: none">1. The member requests to retrieve the reminders that exist2. The device asks the member to choose a specific date3. The device displays the appropriate message on the mirror-surface4. The member provides the specific date5. The device request vocally the confirmation for the date6. The device displays the appropriate message on the mirror-surface7. The member confirms the date8. The device narrates the name and details about the requested reminders9. The device displays the reminders on the mirror-surface
Alternatives	<ol style="list-style-type: none">1a. The member asks what reminders exist<ol style="list-style-type: none">1. The device informs the member that there are no reminders2. The device is ready for a new query7a. The member denies the date for the reminder<ol style="list-style-type: none">1. The device request vocally the member to provide a date for the reminder2. The device displays the appropriate message on the mirror-surface3. The member provides a new date for the reminder
Postconditions	<ol style="list-style-type: none">1. The reminders are displayed on the mirror surface until the member queries a new request or the member leaves

Table 20 "Retrieve reminders use case"

Source: Theodora Tataru, 2021

DELETE REMINDER

To delete a reminder, the member must first initiate the interaction and then asks the device a similar query to "Pixel, delete the reminder for tomorrow" or "Pixel I want to delete a reminder". The device asks for the deletion confirmation, and when the confirmation is given, the device processes the request confirming back to the member the deletion of the reminder. The confirmation is narrated to the member and displayed on the mirror surface.

The detailed use case can be seen in **Table 21**.

Name	Delete Reminder
Actors	Member
Preconditions	<ol style="list-style-type: none"> 1. The member needs to initiate interaction before starting this query 2. The member needs to be registered
Main Success Scenario	<ol style="list-style-type: none"> 1. The member requests the device to delete a reminder 2. The device request vocally from member to provide the specific date for the reminder 3. The device displays the appropriate message on the mirror-surface 4. The member provides the date for the particular reminder 5. The device vocally narrates the name of the reminders on that specific date 6. The device displays on the mirror surface the appropriate message 7. The device asks the member which of the reminder to delete 8. The device displays on the mirror surface the relevant message 9. The member pronounces the name of the reminder to be deleted 10. The device request vocally the confirmation for the name of the reminder 11. The device displays the appropriate message on the mirror-surface 12. The member confirms the name of the reminder 13. The device processes the information 14. The device confirms the deletion of the reminder vocally 15. The device displays the confirmation on the mirror-surface
Alternatives	<ol style="list-style-type: none"> 5a. There are no reminder available for that specific date <ol style="list-style-type: none"> 1. The device informs the member vocally that there are no reminder available to be deleted 2. The device displays the appropriate message on the mirror-surface 12a. The member denies the name of the reminder <ol style="list-style-type: none"> 1. The device request vocally the member to provide a name for the reminder 2. The device displays the appropriate message on the mirror-surface 3. The member provides a new name for the reminder
Postconditions	<ol style="list-style-type: none"> 1. The confirmation of deletion is displayed on the mirror-surface until the member queries a new request or the member leaves

Table 21 "Delete reminder use case"

Source: Theodora Tataru, 2021

CREATE SOS CONTACT

Table 22 describes how the member can query SOS from the device.

To perform this query, the member needs to initiate an interaction and then ask the device a similar query to "Pixel, I want to create an SOS" or "Pixel create an SOS contact". The device asks for more details, and when the member provides all details, the device processes the query. When the SOS contact is successfully created, the device confirms the creation of the SOS contact vocally and displays the information on the mirror surface.

The detailed use case can be seen in **Table 22**.

Name	Create SOS Contact
Actors	Member
Preconditions	<ol style="list-style-type: none"> 1. The member needs to initiate interaction before starting this query 2. The member needs to be registered
Main Success Scenario	<ol style="list-style-type: none"> 1. The member requires to create an SOS contact 2. The device requests a name for the SOS contact vocally 3. The device displays the appropriate message on the mirror-surface 4. The member provides the name for the SOS contact 5. The device repeats the name for the SOS contact vocally, requesting confirmation 6. The device displays on the mirror surface the appropriate message 7. The member confirms the name of the SOS contact 8. The device requests the phone number vocally for the SOS contact 9. The device displays the appropriate message on the mirror-surface 10. The member provides the phone number for the SOS contact 11. The device repeats the phone number of the SOS contact vocally, requesting confirmation 12. The device displays on the mirror surface the phone number of the SOS contact 13. The member confirms the phone number of the SOS contact 14. The device processes the information 15. The device sends a text message to the new next of kin of the member 16. The device confirms the SOS contact registration vocally 17. The device displays the appropriate message on the mirror-surface
Alternatives	<ol style="list-style-type: none"> 2a. The member is not registered <ol style="list-style-type: none"> 1. The device informs the user that it needs to be registered to proceed 7a. The member denies the name of the SOS contact <ol style="list-style-type: none"> 1. The device request vocally the member to provide a name for the SOS contact 2. The device displays the appropriate message on the mirror-surface 3. The member provides a new name for the SOS contact 13a. The member denies the phone number for the SOS contact <ol style="list-style-type: none"> 1. The device request vocally the member to provide a phone number for the SOS contact 2. The device displays the appropriate message on the mirror-surface 3. The member provides a new phone number for the SOS contact
Postconditions	<ol style="list-style-type: none"> 1. The SOS contact displayed on the mirror surface until the member queries a new request or the member leaves

Table 22 "Create SOS contact use case"

Source: Theodora Tataru, 2021

RETRIEVE SOS CONTACT

To retrieve the SOS contact, the member needs to initiate an interaction and ask a question similar to "Pixel, show me the SOS contact" or "Pixel, who is my next of kin". The device processes the request and displays the SOS contact on the mirror surface, along with vocal narration.

The detailed use case can be seen in **Table 23**.

Name	Retrieve SOS Contact
Actors	Member
Preconditions	<ol style="list-style-type: none">1. The member needs to initiate interaction before starting this query2. The member needs to be registered
Main Success Scenario	<ol style="list-style-type: none">1. The member requests the SOS contact2. The device processes the request3. The device narrates the name of the SOS contact4. The device displays the name of the SOS contact on the surface of the mirror
Alternatives	<ol style="list-style-type: none">2a. The member is not registered<ol style="list-style-type: none">1. The device informs the user that it needs to be registered to proceed3a. An SOS contact does not exist<ol style="list-style-type: none">1. The device informs the member that an SOS contact does not exist2. The device displays on the mirror surface the appropriate message
Postconditions	<ol style="list-style-type: none">1. The appropriate message is displayed on the mirror surface until the member queries a new request or the member leaves

Table 23 "Retrieve SOS contact use case"

Source: Theodora Tataru, 2021

UPDATE SOS CONTACT

To update the SOS contact, the member first needs to initiate the interaction and then ask a question similar to “Pixel, update my SOS contact ” or “Pixel, update my emergency contact”. The device asks the member for other details, and once these are provided, the query is processed. When the SOS contact is updated, the device confirms the action vocally to the member, along with the confirmation displayed on the mirror surface.

The detailed use case can be seen in **Table 24**.

Name		Update SOS Contact
Actors	Member	
Preconditions	<ol style="list-style-type: none"> 1. The member needs to initiate interaction before starting this query 2. The member needs to be registered 	
Main Success Scenario	<ol style="list-style-type: none"> 1. The member requests the device to update the SOS contact 2. The device asks for the new SOS contact name 3. The device displays on the mirror surface the appropriate message 4. The member provides the new name for the SOS contact 5. The device asks the member for confirmation of the new name 6. The device displays on the mirror surface the appropriate message 7. The member confirms the action 8. The device vocally request the phone number for the new SOS contact 9. The device displays on the mirror surface the appropriate message 10. The member provides the new phone number for the SOS contact 11. The device asks the member for confirmation of the new phone number 12. The device displays on the mirror surface the appropriate message 13. The member confirms the action 14. The device process the information and updates the new SOS contact 15. The device sends a text message to the number provided by the member 16. The device confirms the creation of the new SOS contact vocally 17. The device displays the confirmation on the mirror-surface 	
Alternatives	<ol style="list-style-type: none"> 2a. The member is not registered <ol style="list-style-type: none"> 1. The device informs the user that it needs to be registered to proceed 7a. The member denies the name of the SOS contact <ol style="list-style-type: none"> 1. The device request vocally the member to provide a name for the SOS contact 2. The device displays the appropriate message on the mirror-surface 3. The member provides a new name for the SOS contact 13a. The member denies the phone number for the SOS contact <ol style="list-style-type: none"> 1. The device request vocally the member to provide a phone number for the SOS contact 2. The device displays the appropriate message on the mirror-surface The member provides a new phone number for the SOS contact 	
Postconditions	<ol style="list-style-type: none"> 1. The new SOS contact is created 2. The confirmation is displayed on the mirror surface until the member queries a new request or the member leaves 	

Table 24 "Update SOS contact use case"

Source: Theodora Tataru, 2021

DELETE SOS CONTACT

To delete the SOS contact, the member must first initiate the interaction with the device, followed by request similar to "Pixel, delete my SOS contact" or "Pixel cancel my next of kin from the account". The device asks for some details for clarification, and once the member provides the details, the device processes the request. When the request was successfully processed, and the SOS contact was deleted from the account, the device confirms the procedure to the member vocally and displayed it on the mirror surface

The detailed use case can be seen in **Table 25**.

Delete SOS Contact	
Name	Delete SOS Contact
Actors	Member
Preconditions	<ol style="list-style-type: none"> 1. The member needs to initiate interaction before starting this query 2. The member needs to be registered
Main Success Scenario	<ol style="list-style-type: none"> 1. The member asks the device to delete the SOS contact 2. The device narrates the SOS contact available 3. The device displays the SOS contact that is available to be deleted 4. The device asks vocally for confirmation 5. The device displays the appropriate message on the mirror-surface 6. The member confirms the deletion 7. The device processes the information 8. The device confirms the deletion of the SOS contact vocally 9. The device displays the confirmation on the mirror-surface
Alternatives	<ol style="list-style-type: none"> 1a The device informs the member that an SOS contact does not exist <ol style="list-style-type: none"> 1. The device informs the member vocally that an SOS contact does not exist 2. The device displays the appropriate message on the mirror-surface 2a. The member is not registered <ol style="list-style-type: none"> 1. The device informs the user that it needs to be registered to proceed 6a. The member denies the deletion <ol style="list-style-type: none"> 1. The device confirms vocally that the SOS contact was not deleted 2. The device displays the appropriate message on the mirror-surface
Postconditions	<ol style="list-style-type: none"> 1. The confirmation of deletion is displayed on the mirror surface until the member queries a new request or the member leaves

Table 25 "Delete SOS contact use case"

Source: Theodora Tataru, 2021

SEND SOS

To send an SOS, the member must first initiate the interaction with the device and needs to be registered on the device. The member needs to ask a question similar to "Pixel, send an SOS". The device processes the request. When the request was successfully processed, the device confirmed the member's procedure vocally and displayed it on the mirror surface.

The detailed use case can be seen in **Table 26**.

Name	Send SOS
Actors	Member
Preconditions	<ol style="list-style-type: none">1. The member needs to initiate interaction before starting this query2. The member needs to be registered
Main Success Scenario	<ol style="list-style-type: none">1. The member asks the device to send an SOS2. The device asks the member for confirmation3. The device displays the appropriate message on the mirror-surface4. The member confirms the request5. The device processes the request6. The device sends an SOS message to the assigned SOS contact7. The device narrates the procedure to the member8. The device displays the appropriate message on the mirror-surface
Alternatives	<ol style="list-style-type: none">2a. The member is not registered<ol style="list-style-type: none">1. The device informs the user that it needs to be registered to proceed2b. The device informs the member that an SOS contact does not exist<ol style="list-style-type: none">1. The device informs the member vocally that an SOS contact does not exist2. The device displays the appropriate message on the mirror-surface3. The device informs the member vocally that Emergency Services will be called4. The device displays the appropriate message on the mirror-surface4a. The member denies the action<ol style="list-style-type: none">1. The device is ready for another query from the member until the member moves away from the mirror4b. The member does not confirm nor denies the action<ol style="list-style-type: none">1. The device sends the text message to the emergency contact2. The device displays the appropriate message on the mirror-surface3. The device informs the member vocally that Emergency Services will be called4. The device displays the appropriate message on the mirror-surface
Postconditions	<ol style="list-style-type: none">1. The device confirms vocally that an SOS has been send and displays on the mirror surface the appropriate message until the member queries a new request or the member leaves

Table 26 "Send SOS use case"

Source: Theodora Tataru, 2021

FURPS+ is a model that represents a document that classifies software attributes:

- **F**unctionality
- **U**sability
- **R**eliability
- **P**erformance
- **S**upportability
- + allows room to specify other constraints such as design, implementation, and physical constraints

The functionality of a software or a device is one of the main software attributes, but besides this main branch, there are other attributes as necessary (Wikipedia, 2020).

This model is designed to clarify, validate, and prioritize a software project's requirements after understanding the client's needs (COEPD, 2014).

FUNCTIONALITIES

The F in "FURPS+" stands for functionality, representing the complexity of processing within the system and the functionalities that span multiple use cases across the system and others (COEPD, 2014).

CORE FUNCTIONALITIES

The project has more than one core functionality, the functionalities described in the sections above, and others that support those.

The core functionalities of "Pixel" Virtual Assistant are:

- the intelligent assistant that performs tasks for the member
- the face recognition functionality that recognizes the member that stands in the front of the mirror
- the interface which is synchronized with the virtual assistant's answers

Bellow, the core functionalities are described in detail:

INTELLIGENT ASSISTANT

The project base is the intelligent assistant that receives requests from the member via voice, processes the requests, and returns a meaningful answer.

The fundamental requests that the intelligent assistant can perform are:

- Register
- Narrate and display whether
- Narrate and display time
- Narrate and display date
- Narrate and display COVID-19 stats
- Narrate and display definitions
- Narrate and display location details
- Call locations
- Narrate and display a joke
- Narrate and display options for help
- Manipulate lists
- Manipulate reminders
- Manipulate SOS contact
- Send SOS

FACE RECOGNITION

The member that interacts with "Pixel" Virtual Assistant is recognized by its particular facial features via facial recognition.

The virtual assistant is designed to make its users engage in their interactions by requiring them to stand in front of a mirror. This way, the device is active and ready to receive requests as long as the user stands in front of the mirror.

THE INTERFACE

The device displays on the mirror surface all the messages that are transmitted via voice to its user, keeping the user engaged in the discussion and allowing users that have hearing deficiencies to comprehend the answer to their request.

OTHER FUNCTIONALITIES

MOBILE DATA



As the device does not possess any other input method than the microphone and the camera, the simplest way of establishing an internet connection is by using the mobile data through the GPS module. Via voice, dictating a long WIFI password with low and capital characters, numbers and symbols would be a frustrating experience.

Figure 20"Mobile data"
(mariasmith636, 2014)

RETRIEVE LOCATION

For certain requests, the correct location of the device would benefit the quality of the intelligent assistant's answer to a request. (e.g., weather forecast, locations). This functionality enables the device to determine its accurate location through the GPS module.

BACKUP

The backup functionality enables the device to backup data for each registered member and their settings.

RESTORE BACKUP

The restored backup functionality allows the member to restore a backup if the device is malfunctioning or the device is changed.

USABILITY

Usability refers to the user experience when utilizing the device (COEPD, 2014).

The user presence is recognized within milliseconds 99% of the time. These metrics are covered by over 20 tests done for facial recognition with the camera, with more than ten different users.

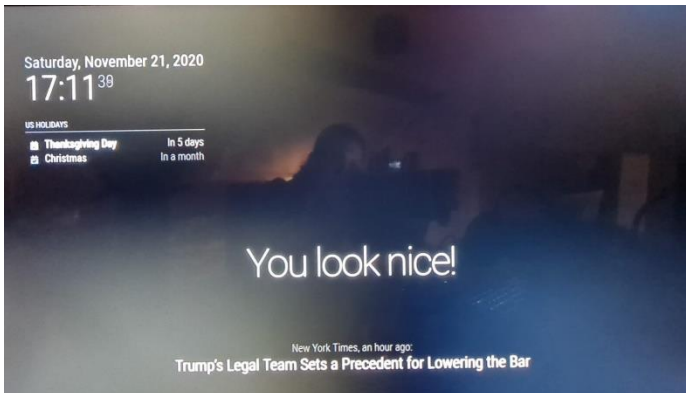
The answers returned to the user by the device are achieved under 2 seconds, 95% of the time. The metrics are covered by over 500 requests performed on the device.

The delay between the vocal response and the mirror surface is less than 0.5 seconds 95% of the time.

VOCAL RESPONSES

The device responds to the user's request vocally, narrating the answer and displaying it on the mirror surface.

DISPLAY



The device display is designed as simple as possible, as seen in **Figure 21**, and is hidden under a two-way acrylic mirror.

The user can see displayed on the screen, in real-time, the answers of the virtual assistant, along with other helpful information.

Figure 21 "Virtual Assistant display"

Source: Theodora Tataru, 2021

RELIABILITY

Reliability addresses the availability, accuracy, and recoverability of the system (COEPD, 2014).

The device response time to a user presence is measured in a few milliseconds 99% of the time.

"Pixel" Virtual Assistant performs a large variety of requests without an internet connection, but it does require access to the internet for some requests. The device recognizes a member, registers a member, and can perform some tasks without an internet connection.

An unexpected reboot of the system happens in less than 1% of the time, and its recovering time is less than 1 minute.

Accidental internet connection loss is recovered automatically as soon as the mobile data signal is back in place. If it cannot connect independently back to the internet connection, the device announces its user about the problem, and the recovering time depends on the user actions.

Other failures had not been encountered with "Pixel" Virtual Assistant.

PERFORMANCE

Performance includes the aspects of processing time, response time, start-up time, accuracy, and others (COEPD, 2014).

"Pixel" Virtual Assistant performance metrics:

- start-up time takes about 10 seconds, 99% of the time
- recognizing a user presence is achieved within milliseconds 99% of the time
- response time to a query takes about 2 seconds 95% of the time
- query responses are accurate 90% of the time
- the delay between the vocal response and the mirror surface is less than 0.5 seconds 95% of the time

SUPPORTABILITY

Supportability refers to the flexibility, maintainability, testability, instability, scalability, and compatibility of the system (COEPD, 2014).

"Pixel" Virtual Assistant:

- is 100% compatible with almost any Raspberry Pi. The compatibility is conditioned by the OpenCV library, as this module requires a minimum amount of resources for maximum functionality and accuracy: a minimum of 1GB of RAM and a minimum of 2GB of SWAP memory. As long as these requirements are met, there should be no issues in installing the software on any Raspberry Pi.
- The code is supplied with comments for each variable, function, and class that the system is using. This action was performed to ease any developer in changing, maintaining, or adding new features to the existing code.
- The project should scale easily, as the device's only interference with the outside computing world is the backup storage. All the actions performed by the device are processed internally.

The "+" in the FURPS+ model stands for constraints that include the design, the implementation, and the physical constraints (COEPD, 2014).

SECURITY

- Uploading backups for the device, a connection is required with the database.

The backup consists of saving the vectors related to the user's facial features and some of the user's settings. Before uploading, the data is encrypted to protect the user's sensitive data.

- Photos of the user are taken to learn the facial expressions and facial contour of the user. After the model was trained to recognize a person, the pictures are deleted from the device and never uploaded on the backup platform. This action is designed to protect users' sensitive and personal information.

IMPLEMENTATION

All the code used to develop this device is respecting the standards:

- Python respects the PEP 8 standards

Python was used to implement the device's backend: the logic and the processing happening behind the curtains.

TKinter was used to implement the GUI: the mirror-surface

INTERFACE

The device is 80% self-contained, as all the user's requests are processed on the device.

The remaining 20% refers to:

- OpenCV – used for facial recognition
- Backup storage – cloud storage needed to store user's backups

TESTING

The testing of "Pixel" Virtual Assistant was performed using manual testing. Each functionality was tested 50+ times with different inputs, finding vulnerabilities within the logic and the processing.

The manual testing was performed with ten different users, each user not having prior training by the developer on using the device.

Each test begins with the device being reset to its initial state, where no user had been registered on the device. The candidates were asked to interact with the device in the most natural way. Due to COVID-19 regulations, the ten candidates were testing the device separately from any other participants, respecting the COVID-19 safety regulations.

90% of the manual testing was a success, with the users being positively impacted by the interaction with "Pixel" Virtual Assistant.

10% of the manual testing brought confusion over the user's experience, only because some of the candidates testing the device had prominent accents not being native English speakers.

INTERFACE

- 80% of the users were delighted with the user interface, finding it easy to use and with an intuitive design
- 10% of the users found the interface too minimalistic
- 10% of the users found the interface elegant and easy to use

INTERACTION

- 30% of the users found strange the dialog with an object, but after a few conversations with the device, they were positively impacted by the experience
- 70% of the users were positively impacted by the interaction with the device

REGISTRATION

- 10% of the users found the registration too complicated
- 10% of the users felt embarrassed to perform different facial expressions in front of the mirror
- 80% of the users found the registration fun and easy

PERFORMANCE

- 100% of the users felt that the device is fast and accurate

The detailed testing report can be found in the "Final Document".

SIMILAR PRODUCTS AND INSPIRATION

"Pixel" is a virtual assistant (smart speaker) encapsulated in a mirror targeted at the general public, people with hearing impairments, and people with low computing skills.

A virtual assistant is a digital device that connects with users through voice commands. The assistant can carry out a series of tasks and offer helpful answers to the consumer's questions, including playing music, listening to audiobooks, managing lists, setting timers, and others (Wikipedia, 2021).

Smart devices are a helpful tool in customers' homes for monitoring their homes, entertainment, and better time planning; however, they interact with their users mostly by voice. This method of interaction excludes a significant portion of society from purchasing and utilizing these products.

Amazon Echo Show 10 and Google Nest Hub represent the two similar products available on the market that could be taken into consideration. There are several other devices in the industry, but the focus is on the closest versions to "Pixel."

Suppose a customer is on a video call, watching a movie, or following a recipe; in that case, the Echo Show 10 digital screen will rotate around its base, bringing the display and camera closer to their eyes. The display quality is not excellent, but it is a good fit for distance watching. When the user asks Alexa for a weather update, the device will respond audibly and show brief results on the screen (Skinner, 2021).

The Google Nest Hub is a great smart speaker featuring two powerful speakers that deliver incredible sound; however, the 10-inch monitor steals the show. The unit can stream video, follow step-by-step recipes, control many smart home devices, and estimate travel time, using either touch controls or going hands-free with voice instructions. The camera inside the unit can be utilized to make video calls (but only via Google Duo), has an advanced motion-tracking feature that holds the user in the center of the shot as they step around a space. It is a game-changer if the user is cooking and can't stay still in front of the monitor (Kozuch, 2021).

The two smart speakers described above bring digital assistants to another level, but even in this scenario, people with low computer skills and hearing impairments are not the focused target market. Also, both devices are based on their previous models where they are constantly listening for a keyword to be spoken by the user to begin a conversation.

Pixel, Virtual Assistant, responds to the user's question with a voice response while simultaneously displaying all the details on the mirror surface. Users with hearing impairments would be able to comprehend the device's information this way. This feature would also encourage people with limited computing skills to take their time, read the information, and decide what they have to do next. The idea that Pixel displays the entire conversation on the mirror surface is a crucial function.

Another important feature is that the device is idle all the time except when detecting a face; therefore, it does not listen to what is happening in the environment surrounding it.

The user interface for "Pixel, Virtual Assistant" is inspired by the open-source project "Magic Mirror", but it is designed from scratch using different technologies compared with the open-source project.

CONCLUSION

"Pixel" is a virtual assistant encapsulated in a mirror designed to meet the requirements of any individual by any age or any level of computer skills regarding a basic virtual assistant. This document states the functionality that "Pixel" brings to its users and how they interact with the device.

The main functionalities of "Pixel" are as follows: initiate interaction, register, get the weather forecast, get time, get the date, get COVID-19 stats, get definitions, get location details, call location, get joke, get help, manipulate lists such as grocery lists, manipulate reminder, manipulate SOS contact, send SOS and backup restore.

Each functionality was described in detail, emphasizing the interaction of the user with the device.

These functionalities were described in a survey that was completed by over 60 people, potential users. The purpose of the survey was to validate that the functionality meets the stakeholder's expectations, that interactions are simple to be performed by any individual, and that there exists a demand for the device.

Several system attributes were also discussed, such as functionality, usability, reliability, performance, supportability, and security.

Overall, the functionality of "Pixel" Virtual Assistant was decided, but the priorities are expected to change several times based on stakeholder feedback and by the time allocated for the development of the device. During its development, "Pixel" Virtual Assistant will be presented to a certain number of users that will interact with the device and provide feedback based on their experience. The feedback describes the device's impact on the user's experience, expectations, likes, and dislikes.

BIBLIOGRAPHY

COEPD, B. A. T. i. H. –, 2014. *What is FURPS+?*. [En ligne]

Available at: <https://businessanalysttraininghyderabad.wordpress.com/2014/08/05/what-is-furps/#:~:text=FURPS%20is%20a%20technique%20to,with%20client%27s%20needs%20and%20necessities.&text=The%20%E2%80%9C%2B%E2%80%9D%20of%20the%20FURPS%2B,%2C%20interface%2C%20and%20physi>
[Accès le 21 November 2020].

Kozuch, K., 2021. *The best Google Home speakers in 2021*. [En ligne]

Available at: <https://www.tomsguide.com/best-picks/best-google-home-speakers>
[Accès le 27 April 2021].

mariasmith636, 2014. *How your network operator differs in services offered?*. [En ligne]

Available at: <https://mobileservicesweden.wordpress.com/2014/07/19/how-your-network-operator-differs-in-services-offered/>
[Accès le 27 April 2021].

Saba, E., 2018. *Amazon Echo Show's new 'Tap to Alexa' accessibility feature is handy enough for anyone to use*. [En ligne]

Available at: <https://www.aftvnews.com/amazon-echo-shows-new-tap-to-alexa-accessibility-feature-is-handy-enough-for-anyone-to-use/>
[Accès le 27 April 2021].

Skinner, C., 2021. *Amazon Echo Show 10 review*. [En ligne]

Available at: <https://www.techradar.com/reviews/amazon-echo-show-10-review>
[Accès le 27 April 2021].

Wikipedia, 2020. *FURPS*. [En ligne]

Available at: <https://en.wikipedia.org/wiki/FURPS>
[Accès le 21 November 2020].

Wikipedia, 2021. *Amazon Alexa*. [En ligne]

Available at: https://en.wikipedia.org/wiki/Amazon_Alexa
[Accès le 27 April 2021].