



HOLY SPIRIT UNIVERSITY OF KASLIK
SCHOOL OF ENGINEERING

FINAL REPORT

Robot Pepper at the commencement ceremony

Completed by

Maria El Murr (202100236)

Paul Chbat (202100228)

Anthony Massaad (202101048)

Elio El Khoury (202001967)

Georges Chamoun (202100639)

Presented to: Br. **Elie Saad**

Module: GIN456

Spring 2023-2024

Table of Contents

Acknowledgements.....	3
Report Structure.....	4
Introduction.....	5
Use Case 1: Select Language.....	6
Use Case 1 and story board:	6
Technical Part:	8
Use case 2: Stating Rules.....	10
Use case 2 and story board:	10
Technical part:	12
Use Case 3: Scan QR Code	16
Technical Part:	18
Use case 4: Direct Guests	20
Use case 4 and story board:	20
Technical part:	22
Use case 5: Take a Selfie	24
Personas	30
Persona 1:.....	31
Persona 2:.....	32
Persona 3:.....	33
Persona 4:.....	34
Challenges.....	36
Desing Patterns	37
Conclusion	38

Acknowledgements

We want to express our deepest gratitude to the Holy Spirit University in Kaslik for the opportunity to participate in this course and for providing us with all the resources required, particularly the Pepper robot, that helped us complete this project.

A special acknowledgment goes to Br. Elie Saad for his guidance, which has been helpful for creating our project and construct meaningful interactions with Pepper.

We also extend our appreciation to the entire team for their collaborative efforts and dedication. Working together have contributed significantly to the success of our project.

In closing, we sincerely hope that our efforts reflect the hard work and dedication we have made throughout the semester.

Report Structure

Our report begins with an introduction, providing an overview of the project and its objectives.

Following this, we will explain each case individually:

Starting with the language selection feature, we will explain how Pepper was programmed to offer language options.

Moving forward, we explore Pepper's role in communicating the ceremony rules, outlining the methods employed to ensure clear and effective communication.

Subsequently, we discuss Pepper's ability to scan QR codes, explaining the process of integrating this functionality.

We also allow Pepper robot to highlight the zone where the guest is seated.

Lastly, we let guests enjoy the interaction with the robot by a fun feature, where Pepper will do gestures so that the guests can take a selfie with it.

We will also introduce the key personas involved in the project.

Towards the end of the report, each member of the team will write about the challenges that he faced while completing the project.

Finally, we conclude by summarizing the project's outcomes.

Introduction

Our report highlights the function of a Pepper robot for a graduation ceremony.

We programmed Pepper to fulfill several important roles during the event:

Firstly, it provided guests with the option to choose between English and French languages for communication, ensuring accessibility for all attendees.

Additionally, Pepper informed guests about the ceremony's rules and guidelines.

It also could scan QR codes, aiding in the organization of seating arrangements by directing guests to their designated zones.

Lastly, Pepper added a touch of excitement by taking selfies with guests, enhancing the overall experience of the ceremony.

Through this project, we aimed to demonstrate how robots like Pepper can contribute to the smooth execution and enjoyment of special events like graduation ceremonies.

Use Case 1: Select Language

Use Case 1 and story board:

UC-ID and Name	UC-01 Select Language		
Created By	Anthony Massaad, Elio Khoury	Creation Date	2/13/2024
Actors	<ul style="list-style-type: none"> ➤ Ceremony guests (Parents, cousins, friends...) ➤ Robot (Pepper) <p>Roles: The robot proposes the language to be used, and the guest chooses the language that he wants.</p>		
Trigger	The guest chooses the proposed language and communicates with the robot using the chosen language.		
Description (Objective s/Goals)	<ul style="list-style-type: none"> ➤ The robot presents a list of available languages or prompts the user to specify their preferred language. ➤ Enable the robot to identify the language preferences of the user. ➤ Ensure clear and effective communication between the user and the robot. ➤ The robot communicates with the user using the selected language for the interaction. 		
Preconditions	<ol style="list-style-type: none"> 1. The robot is powered on and in an idle state, displaying its default interface or awaiting interaction. 2. The user is there, close to the robot, and has the required interfaces or input devices to use the language selection capability of the robot. 		
Postconditions	<ol style="list-style-type: none"> 1. The user can choose a language from the list of possibilities. 2. The robot remains ready for further interactions in the chosen language. 		
Action Sequence (Success Scenario)	<ol style="list-style-type: none"> 1. The user selects a language from the available options. 2. The robot accepts the language selection. 3. The robot's interface elements switch to display in the chosen language. 4. The robot's voice prompts in the selected language. 5. The user proceeds with interactions in the selected language. 		
Extensions	<p>3a. No matching language option is found:</p> <p>3a1. The Robot alerts the User that the selected language option is not available or recognized.</p> <p>3a2. The User chooses to either select another available language option, exit the language selection process, or return to step 2 of the normal flow to reattempt language selection.</p> <p>5a. The User encounters difficulties in using the selected language:</p>		

	5a1. The Robot provides additional assistance, such as simplifying interactions, or switching back to a previously used language.
Requirements	<ol style="list-style-type: none"> 1. The robot must support multiple language options. 2. The language selection feature must be easily accessible for the user. 3. Assistance options must be available for users encountering difficulties in using the selected language (ex: language support, ...).
Storyboards	Figure below.
Priority	1
Related Use Cases	<ol style="list-style-type: none"> 1. UC-02: greet guests. 2. UC-05: direct guests.
Assumptions	<ol style="list-style-type: none"> 1. The user has basic familiarity with the language selection process or interface. 2. The robot can understand and communicate in the languages offered as options.
Open Issues	<ol style="list-style-type: none"> 1. Dealing with situations where the user's preferred language is not supported by the robot. 2. Handling potential misunderstandings or ambiguities in language selection.



Figure 1: Use case 1 storyboard.

Technical Part:

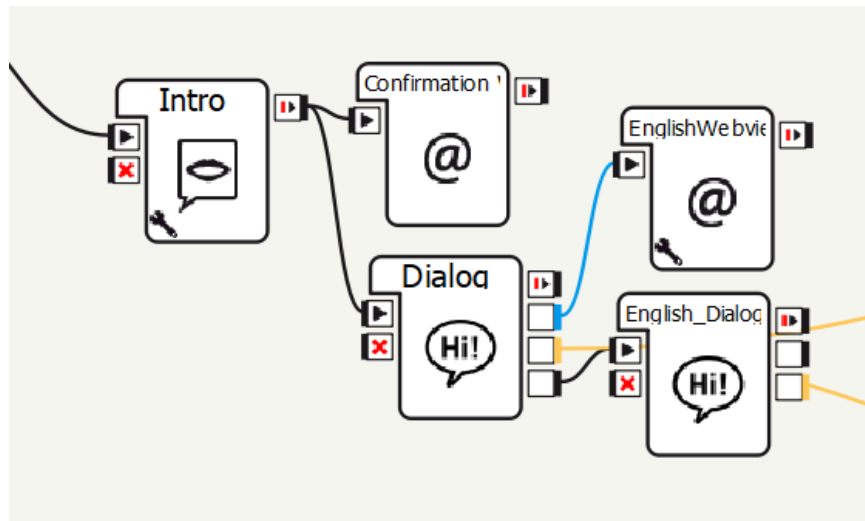


Figure 2: language selection boxes.

We utilized various blocks in our project to achieve the required functionality. Initially, we introduce the first use case, UC01: Select Language, as depicted in Figure 1. In this use case, the first block (Intro) involves launching our project with an introduction and an animation. Following that, we have two other blocks (Confirmation WebView and Dialog) where we start our project. This block includes displaying the webpage of UC01 and offering a brief greeting after the preferred language is selected. (Note: We only support the English language due to limited access to other languages.)

```
topic: ~confirmation()
language: enu
include: lexicon_enu.top

proposal: %englishProposal ~joyful you selected english Language $englishOutput=1 great $english=1 $onStopped=1
proposal: %frenchProposal ~neutral You selected french language \pau=200\ I don't actually know how to speak french

u: (english) ^gotoReactivate(englishProposal)
u: (french) ^gotoReactivate(frenchProposal)

u: (e:englishAnswer) ^gotoReactivate(englishProposal)
u: (e:frenchAnswer) ^gotoReactivate(frenchProposal)
```

Figure 3: Language selection dialog.

After selecting the English language, we proceed with an interaction between the robot and the guest, exchange greetings, and gathering some information about the guest.


```

topic: ~English_Dialog()
language: enu

proposal: \rspd=85\What is your name \pau=500\?
        ul: (_[elio anthony maria paul Georges]) Welcome $1 $name=$1 \pau=500\ ^nextProposal
proposal: \rspd=85\ what is the relationship between you and the graduating student?
        ul: (*) \rspd=90\ Great $name \pau=500\Most Welcome \pau=500\ ^nextProposal
proposal: \rspd=85\What are you most looking forward to about today's ceremony?
        ul: (*) \rspd=90\ Wow \pau=500\That's wonderful\pau=500\ ^nextProposal
proposal: \rspd=85\ Is this the first time you visit USEK?
        ul: (Yes) \rspd=90\ Ok Great $name \pau=500\ ^nextProposal
        ul: (No) \rspd=90\ Most Welcome $name \pau=500\ ^nextProposal
proposal: \rspd=85\ Thank you \pau=100\ for joining us today. \pau=500\ $onRules=1

#u:(hello) \rspd=85\ Hello , Now I want to take some information \pau=500\^nextProposal
u:(e:onStart) \rspd=85\ Hello , Now I want to take some information \pau=500\^nextProposal

```

Figure 4: Language selection proposals.

In this dialog, we prompt the user for their name, inquire about their relationship to a graduating student, ask about their anticipation for the ceremony, and inquire whether they've visited the location before.

Use case 2: Stating Rules

Use case 2 and story board:

UC-ID and Name	UC-02 Stating Rules		
Created By	Paul Chbat, Maria El Murr	Creation Date	2/13/2024
Actors	<ol style="list-style-type: none">1. Ceremony Guests (Family and friends) arrive at the event.2. Receptionist robot initiates the rules.		
Trigger	The robot states the rules of the ceremony.		
Description (Objectives/Goals)	<ol style="list-style-type: none">1. The robot states the rules of the ceremony.2. The ceremony guests enjoy the interaction and get to know the rules to be followed.		
Preconditions	<ol style="list-style-type: none">1. The ceremony guest has chosen one of the available languages in the previous interaction described in UC-01 (Select Language).2. The robot, after processing the choice of the ceremony guest, is ready to engage with the ceremony guest.		
Postconditions	The robot will start the process of scanning the QR code provided by the ceremony guest.		
Action Sequence (Success Scenario)	<ol style="list-style-type: none">1. The robot asks the guests if they are ready to know the rules.2. The guests respond to the robot.3. The robot starts by stating the rules.4. The guests can navigate to the next rule by saying next or by pressing the next button.5. After finishing all rules, the robot will give the user a quick fact about the ceremony.		

	6. The robot will tell the user the next step of the process.
Extensions	
Requirements	<p>1. The robot must have access to the rules to be followed while attending the ceremony.</p> <p>2. The robot must be able to complete this interaction using the selected language.</p>
Storyboards	Figure below
Priority	1
Related Use Cases	<p>1. UC-01 Select Language.</p> <p>2. UC-03 Scan QR Code.</p>
Assumptions	
Open Issues	

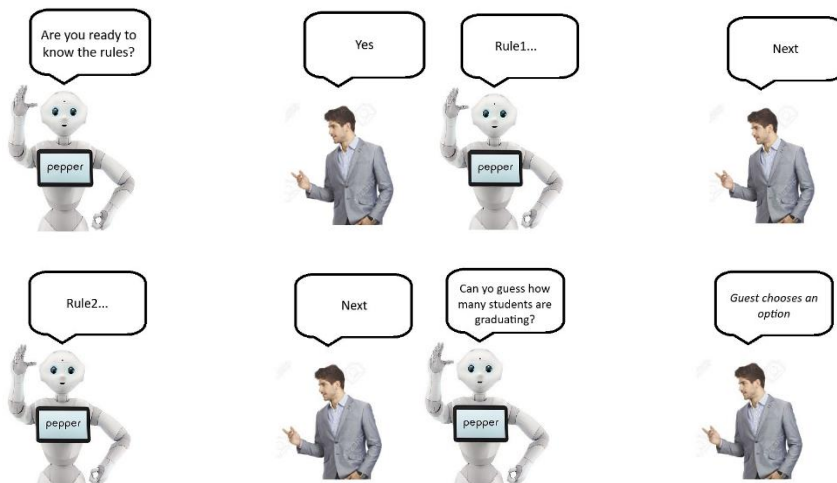


Figure 5: Use case 2 story board.

Technical part:

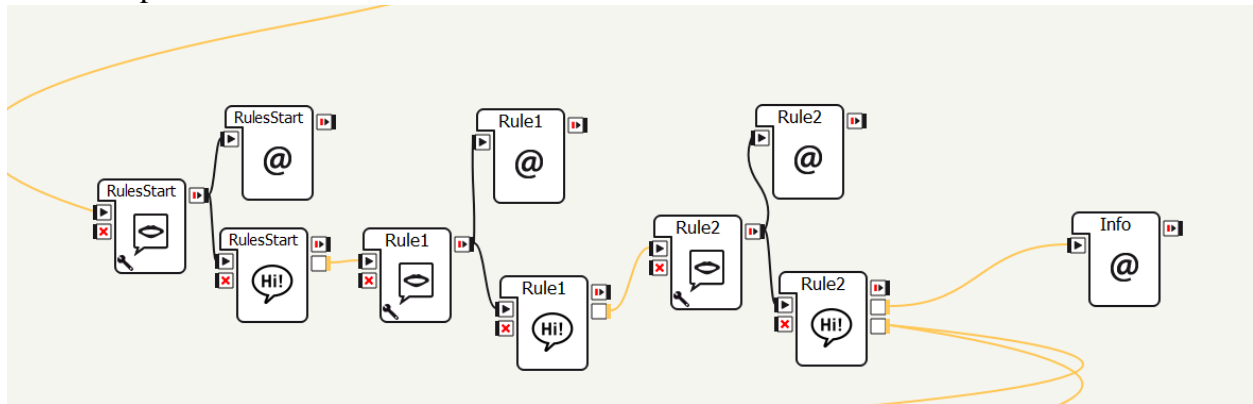


Figure 6: Use case 2 boxes.

As we can see from the figure above, we have a certain workflow, which will discuss each part of it in details:

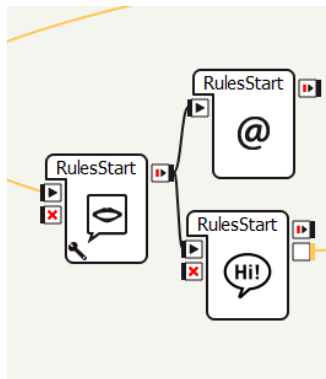


Figure 7: Stating the rules introduction.

In this part, the robot will talk and display an image on the tablet at the same time.

```

if appName:
    if tabletService:
        # use default robot IP address from the tablet
        robotIP = "198.18.0.1" # tabletService.getRobotIp()

        # Set page details
        pageText = "Are you ready to hear the rules?"
        pageImage = "rules.png"
        # Set the URL
        url = "http://{ip}/apps/{appName}/pages/start.html?pageText={pageText}&pageImage={pageImage}".format(ip=robotIP,
        appName=appName, pageHeading=pageHeading, pageText=pageText, pageImage=pageImage)

        self.logger.info(url)

```

Figure 8: Passing parameters to html page.

Concerning what we are displaying on the tablet while the robot is talking, as we can see from the picture above, we are passing the “pageText” and “pageImage” to the URL, this URL is the path of the html page called “start.html”.

According to the above figure, we will display a text “Are you ready to know the rules?” and below an image.

```

topic: ~start()
language: enu

concept: (greetings) [Hello Hi Greetings "Good evening"]
concept: (approval) [yes sure "why not" absolutely]
concept: (denial) [no "another time"]

proposal: %rules Okay, we will move on to the rules, say next after each rule to move on. \pau=200\ $rule1=1 $onStopped=1

u: (e:.onStart) \pau=200\ Now are you ready to hear the rules?
ul: (~approval) ^gotoReactivate(rules)

```

Figure 9: Start state rules dialog.

This is the dialog box, the robot will ask the guest if he's ready to know to rules, if the guest answers by any of the words in the approval concept, gotoReactivate will start the proposal named "rules" and rule1=1 will allow us to go to the next box.

```

function displayPageInformation() {
    document.getElementById("pageHeading").innerHTML = decodeURI(getUrlParam("pageHeading", ""));
    document.getElementById("pageText").innerHTML = decodeURI(getUrlParam("pageText", ""));

    imageName = getUrlParam("pageImage", "");
    if (imageName) {
        document.getElementById("pageImage").src = "../pics/" + imageName;
    }
}

```

Figure 10: Display page information function.

This is the displayPageInformation function that will retrieve the page heading, the page text and will image name.

After the guest said yes to know the rules, the robot will start stating the rules, and to move one from one rule to another, the guest can press the next button or he can say next, we will now see the logic behind pressing the next button.

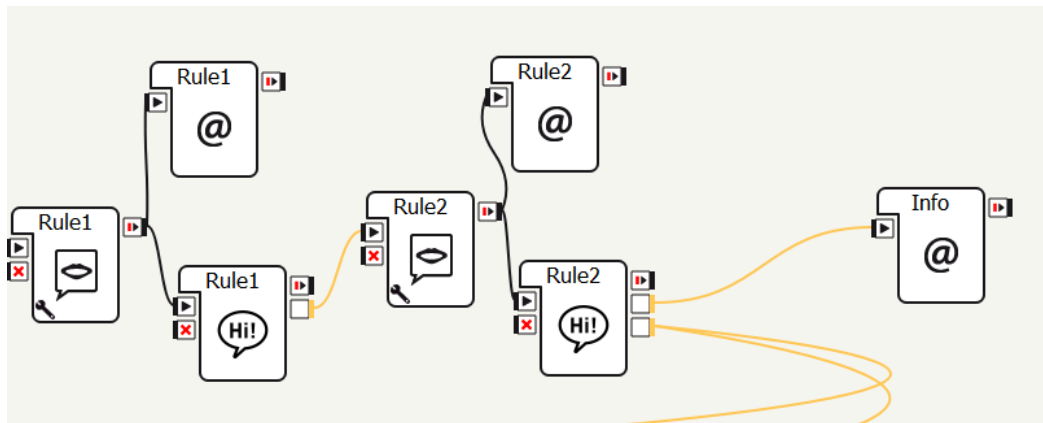


Figure 11: Rules boxes.

```

<div class="w3-center">
  <button type="button" id="next" onclick="raiseConfirmationEvent('rule1next')" class="fa rate-button rounded-rect-btn green-btn" style="font-size:40px;">
    <span class="default-font">Next</span></button>
</div>

```

Figure 12: Next button.

In this image we can see the button next which on click will raise a confirmation event, the name of the event will be passed as parameter. And what this function does is that it set rule1next=1 so that we can go to rule 2 and so on.

```
topic: ~rule1()
language: enu

proposal: %rule1prop Stay respectful to other guests attending this ceremony, and avoid shouting while the ceremony is
underway.\pau=1000\
  ul:(next) $rule2=1 $onStopped=1

u:(e:onStart) Rule 1 \pau=250\ ^gotoReactivate(rule1prop)
u:(e:rule1next) $rule2=1 $onStopped=1
```

The dialog above shows that when event rule1next is raised we will set rule2=1 to move on to the next rule.

After stating all rules, the robot will ask the guest a trivia question displaying 4 buttons(options) to choose from:

```
topic: ~rule2()
language: enu

proposal: %rule2prop Avoid throwing trash on the floor, and no food is allowed in the ceremony room.\pau=1000\
  ul:(next) ^gotoReactivate(info)

proposal: %info $info=1 \pau=250\ Before moving on to more serious stuff, \pau=100\
  Can you guess how many students are graduating this year? Choose an option.

proposal: %end Great, now don't forget to follow the rules, \pau=250\ now we will move on to scanning your qr code.
  $scanqr=1 $onStopped=1

u:(e:incorrectButton) Incorrect, 1200 students are graduating this year. \pau=250\ ^gotoReactivate(end)
u:(e:correctButton) Correct \pau=250\ ^gotoReactivate(end)
u:(e:onStart) Rule 2 \pau=250\ ^gotoReactivate(rule2prop)
u:(e:rule2next) ^gotoReactivate(info)
```

```
<div class="">
  <div class="">
    <div id="informationSection" class="">
      <h2 id="pageText"></h2>
      <hr />
    </div>
    <div class="">
      <div class="">
        <button type="button" id="yesButton" onclick="raiseConfirmationEvent('correctButton')" class="" style="font-size:40px;">
          <span class="">1200</span><br>
        </button>
        <button type="button" id="noButton" onclick="raiseConfirmationEvent('incorrectButton')" class="" style="font-size:40px;">
          <span class="">1300</span><br>
        </button>
      </div>
    </div>
    <div class="">
      <div class="">
        <button type="button" id="noButton" onclick="raiseConfirmationEvent('incorrectButton')" class="" style="font-size:40px;">
          <span class="">1500</span><br>
        </button>
        <button type="button" id="noButton" onclick="raiseConfirmationEvent('incorrectButton')" class="" style="font-size:40px;">
          <span class="">1000</span><br>
        </button>
      </div>
    </div>
  </div>
  <div class="">
    <img id="pageImage" class="" alt="Pepper Pic" width="600" height="650">
  </div>
</div>
```

All logic applied for the buttons and dialog was used before.

```

def onInput_onStart(self):
    # We create TabletService here in order to avoid
    # problems with connections and disconnections of the tablet during the life of the application
    tabletService = self._getTabletService()

    # Get the app name
    appName = self.packageUId()

    if appName:
        if tabletService:
            # use default robot IP address from the tablet
            robotIP = "198.18.0.1" # tabletService.getRobotIp()

            # Set page details
            pageText = "Guess how many students are graduating"
            pageImage = "usek.png"
            # Set the URL
            url = "http://{ip}/apps/{appName}/pages/info.html?pageText={pageText}&pageImage={pageImage}".format(ip=robotIP, appName=appName,
pageText=pageText, pageImage=pageImage)

            self.logger.info(url)

            # Show the web view on the tablet
            tabletService.showWebView(url)
        else:
            self.logger.warning("Couldn't find tablet service, so can't set application: %s" % appName)

    # self.onStopped()

```

To be able to pass only pageText and pageImage without pageHeading in the Info webview we modify the displayPageInformation function to get the following:

```

function displayTextImageInformation() {
    document.getElementById("pageText").innerHTML = decodeURI(getUrlParam("pageText", ""));

    imageName = getUrlParam("pageImage", "");
    if (imageName) {
        document.getElementById("pageImage").src = "../pics/" + imageName;
    }
}

```

Use Case 3: Scan QR Code

UC-ID and Name	UC-03: Scan QR Code		
Created By	Anthony Massaad, Elio Khoury	Creation Date	2/17/2024
Actors	<ul style="list-style-type: none"> ➤ Ceremony guests (Parents, cousins, friends...) ➤ Robot (Pepper) <p>Roles: The robot should be capable of scanning QR codes using its built-in camera, and guests should scan their event tickets to enter the ceremony.</p>		
Trigger	The guest presents QR Code for Check-in, and the robot scans this QR Code using his build in camera.		
Description (Objectives/Goals)	<ul style="list-style-type: none"> ➤ The guests attending the ceremony can present their QR codes for identification and registration. ➤ A faster check-in process at the ceremony reception area reduces errors and minimizes guest wait times. ➤ It provides a secure method of authentication, safeguarding guest information throughout the registration process. 		
Preconditions	<ol style="list-style-type: none"> 1. Guests must have received or have access to QR codes containing relevant information. 2. The robot must be equipped with functional QR code scanning hardware and software. 3. The robot should be placed strategically in the check-in area. 		
Postconditions	<ol style="list-style-type: none"> 1. The guest's QR code has been scanned successfully by the robot's camera. 2. The event management system is updated with the latest guest information obtained from the QR code scanning process. 3. The robot remains available to assist guests with any additional inquiries or tasks beyond the check-in process. 		
Action Sequence (Success Scenario)	<ol style="list-style-type: none"> 1. Guests approach the reception area where the robot is stationed. 2. The robot greets the guests and prompts them to present their QR codes for check-in. 3. Guests retrieve their QR codes and hold them up for the robot to scan. 4. The robot validates the QR code data to confirm guest registration and check-in status. 5. The robot displays a confirmation message to the guests. 6. Use case ends. 		
Extensions	<p>3a. No QR Code Presented:</p> <p>3a1. The robot alerts the guest that no QR code was detected.</p> <p>3a2. The guest may choose to retrieve and present their QR code again.</p> <p>5a. QR Code Validation Failure:</p> <p>5a1. The robot notifies the guest that their QR code could not be validated.</p>		

	5a2. The guest may choose to rescan their QR code or seek assistance from event staff.
Requirements	<ol style="list-style-type: none"> 1. Each QR code presented for check-in should be single use to prevent unauthorized access or duplication of entry. 2. The robot must validate QR code data to ensure guest registration and check-in status. 3. The robot should handle errors gracefully, providing clear instructions to guests in case of QR code scanning failures.
Storyboards	Figure below
Priority	1
Related Use Cases	<ol style="list-style-type: none"> 1. UC-01: Select language. 2. UC-02: greet guests. 3. UC-04: take selfie. 4. UC-05: direct guests.
Assumptions	<ol style="list-style-type: none"> 1. The QR codes provided to guests contain valid and accurate information required. 2. The lighting conditions at the check-in area are suitable for the robot's camera to scan QR codes effectively. 3. Guests are familiar with QR codes and can present them correctly for scanning.
Open Issues	<ol style="list-style-type: none"> 1. Handling situations where guests do not have access to their QR codes due to technical issues or misplacement. 2. Providing an alternative check-in process for guests who cannot use QR codes (e.g., manual registration). 3. Ensuring the privacy and security of guest information contained within the QR codes. 4. Handling duplicate or invalid QR codes that may be presented by mistake or maliciously. 5. Providing clear error messages and troubleshooting guidance when QR code scanning or validation issues occur.

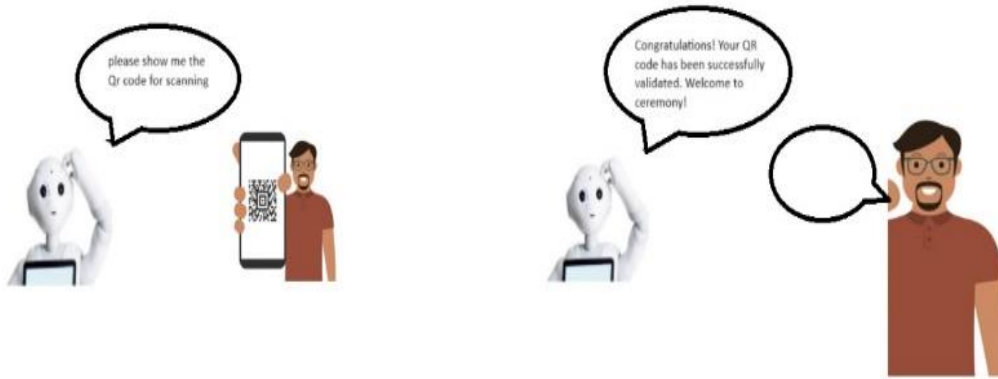


Figure 13: Use case 3 story board.

Technical Part:

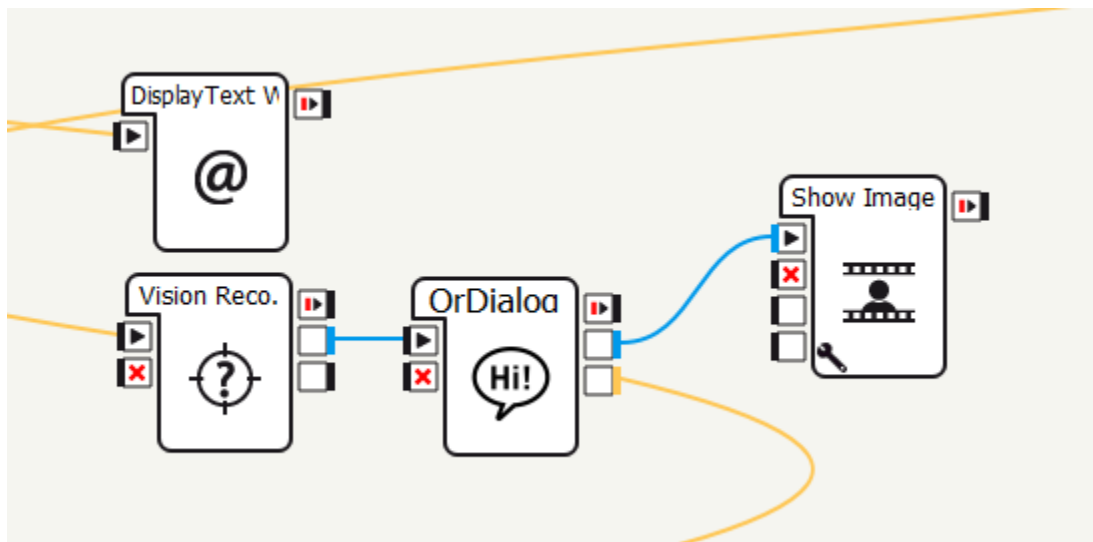
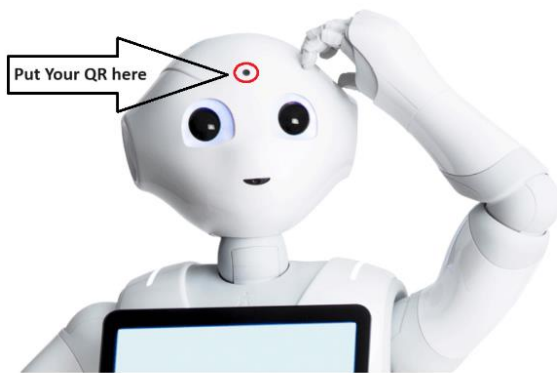


Figure 14: Scan QR Code boxes.

In these boxes we display first a page that helps the user to know how to scan the QR Code using the camera on the head of the robot (in the DisplayText WebView box).



Scan QR Code



Figure 15:displayText WebView.

After that we use vision recognition to Scan the QR Code of the guest.

```
class MyClass(GeneratedClass):
    def __init__(self):
        GeneratedClass.__init__(self, False)

    def onLoad(self):
        self.nPicturesDetected = -1

    def onUnload(self):
        #puts code for box cleanup here
        pass

    def onInput_onStart(self, p):
        self.memory_proxy=self.session().service("ALMemory") #create a session
        if(len(p) > 1):
            self.nPicturesDetected = len(p[1])
            labels = []
            for s in p[1]:
                labels.append(s[0])
            self.onPictureLabel( labels )
            self.memory_proxy.raiseEvent("PictureLabel",s[0][0]) #raise for the event
            self.logger.info("event picture label raised :"+ s[0][0])# show event in logs
        else:
            if(self.nPicturesDetected != 0):
                self.nPicturesDetected = 0
                self.onNoPicture()
```

Figure 16:Vision Recognition Code

First, we create a session with the "ALMemory" service, which is a memory management service. After that, we raise an event named "PictureLabel" with the first label from **p[1]** using the ALMemory service. Here, **s[0][0]** is used to access the first element of the first sub-list within **p[1]**, which is assumed to be the label or name of the detected picture or object. Finally, we log a message to the application's logs indicating that a "PictureLabel" event has been raised, along with the specific label or name of the detected picture or object (**s[0][0]**).

After Scanning the QR Code, we show the invitation cart of the guest using the ShowImage box.

Use case 4: Direct Guests

Use case 4 and story board:

UC-ID and Name	UC-04 Direct Guests		
Created By	Paul Chbat, Maria El Murr	Creation Date	2/16/2024
Actors	<ul style="list-style-type: none"> - Ceremony Guest (Family and friends) wait to know where the zone is where they are seated. - Receptionist robot displays the image of the whole theater on its screen and highlights the zone of the guest. 		
Trigger	The robot processes the QR code and displays the theater image and then asks the user which zone he's seated in.		

Description (Objectives/Goals)	<p>1- The robot, after processing the QR code, directs the guest by displaying a map of the ceremony hall.</p> <p>2- The guest enters the ceremony hall.</p>
Preconditions	<p>1- The robot will have scanned the QR code and provided the guest with the name of the zone where he's seated based on his QR code.</p>
Postconditions	<p>The robot, after guiding the guest, goes back into the idle state and back to the default language (English) and waits for a new guest to arrive.</p>
Action Sequence (Success Scenario)	<ol style="list-style-type: none"> 1. The robot displays a map of the ceremony hall on the tablet. 2. The robot asks the guest in which zone he's seated. 3. The guest selects the zone. 4. The robot highlights the zone where the guest is seated. 5. The robot goes back to idle state.
Extensions	
Requirements	<p>1- The robot should have access to a map of the ceremony hall with highlighted zones.</p>
Storyboards	Figure below
Priority	
Related Use Cases	<p>1- UC-03 Scan QR code</p> <p>2- UC-04 Take a Selfie</p> <p>3- UC-01 Select Language</p>
Assumptions	

Open Issues

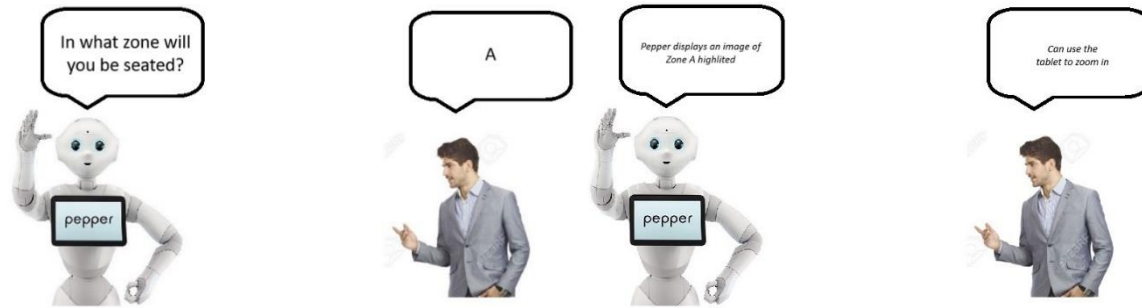


Figure 17: Use case 4 story board.

Technical part:

After scanning the QR code, the guests were told in which zone they are seated, now the robot will ask them about the zone name and will highlight it so that they know where this zone is in the theater.

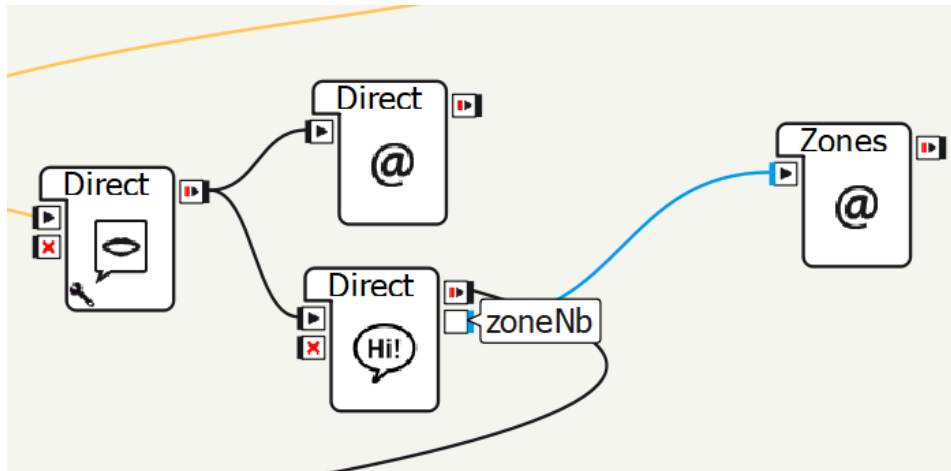


Figure 18: Direct Guest workflow.

As we can see from the picture above, we have two webview boxes, first one which is named "Direct" will show an image of the theater and will show four buttons since we have four zones (A, B, C and D).

The guest will click on the button according to which zone he's seated in and the other webview will show a picture of the theater but with the zone highlighted.

```

topic: ~directGuest()
language: enu

proposal: %question In which region will you be seated?
  ul: (A) $zoneNb="1" ^gotoReactivate(speak)
  ul: (B) $zoneNb="2" ^gotoReactivate(speak)
  ul: (C) $zoneNb="3" ^gotoReactivate(speak)
  ul: (D) $zoneNb="4" ^gotoReactivate(speak)

proposal: %speak Now you can zoom in using the tablet to choose any seat, say or press continue to move forward.
  ul: (continue) $onStopped=1

u: (e:.onStart) Now let me direct you to your seat. \pau=250\ ^gotoReactivate(question)
u: (e:continueButton) $onStopped=1
u: (e:choose) $zoneNb=$choose ^gotoReactivate(speak)

```

Figure 19: Direct Guest dialog.

In the dialog of the direct guests, the robot will go to the proposal name “question” and will ask the guest ‘in which region will you be seated?’, the guest can say the name of the zone or can click on the buttons, when he says the name of the zone we set “zoneNbr” to the predefined zone number that we chose for each zone (1-A, 2-B, 3-C, 4-D) so we display the corresponding html page.

When the guest clicks on the desired button, we raise an event named “choose” and set its value to the according zone number. In both cases, if the guest has talked to the robot or clicked the button, we will activate the proposal named “speak” while displaying the appropriate html file.

```

<div class="...">
  <div class="...">
    <div id="informationSection" class="...">
      <hr />
      <h1 id="pageHeading" style="font-size: 45px;padding-top:50px"></h1>
      <hr />
      <h3 id="pageText"></h3>
    </div>
    <div class="...">
      <div class="...">
        <button type="button" id="next" onclick="raiseEvent('choose', '1')" class="..." style="font-size:40px;">
          <span class="...">A</span><br>
        </button>
        <button type="button" id="next" onclick="raiseEvent('choose', '2')" class="..." style="font-size:40px;">
          <span class="...">B</span><br>
        </button>
        <button type="button" id="next" onclick="raiseEvent('choose', '3')" class="..." style="font-size:40px;">
          <span class="...">C</span><br>
        </button>
        <button type="button" id="next" onclick="raiseEvent('choose', '4')" class="..." style="font-size:40px;">
          <span class="...">D</span><br>
        </button>
      </div>
    </div>
  </div>
  <div class="...">
    <img id="pageImage" class="..." alt="Pepper Pic" width="600" height="650">
  </div>
</div>

```

This is the direct.html file, as discussed it contains 4 buttons for each zone, the page heading, page text and image are passed as parameters from webview box in choregraphe.

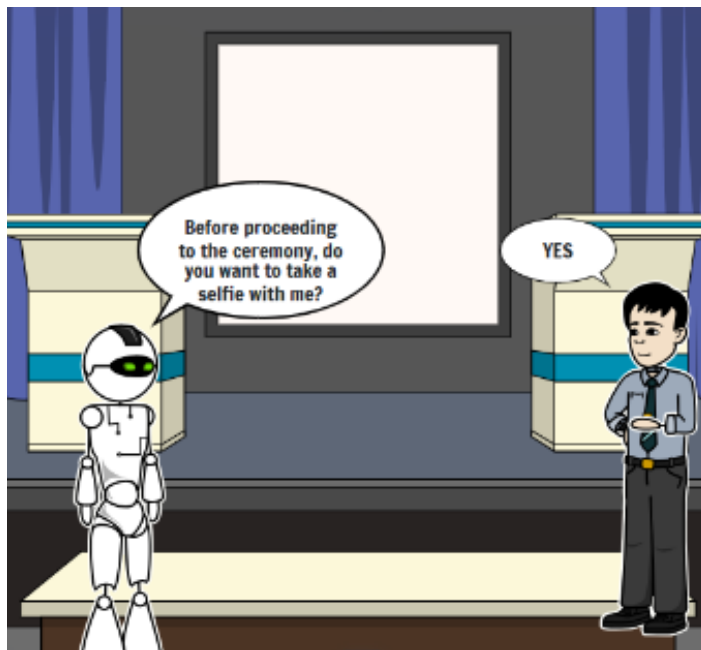
After choosing a zone and checking it out on the tablet the guest can either say or press a Continue button to move on. (All previously explained logic is applied in this case to create the continue button)

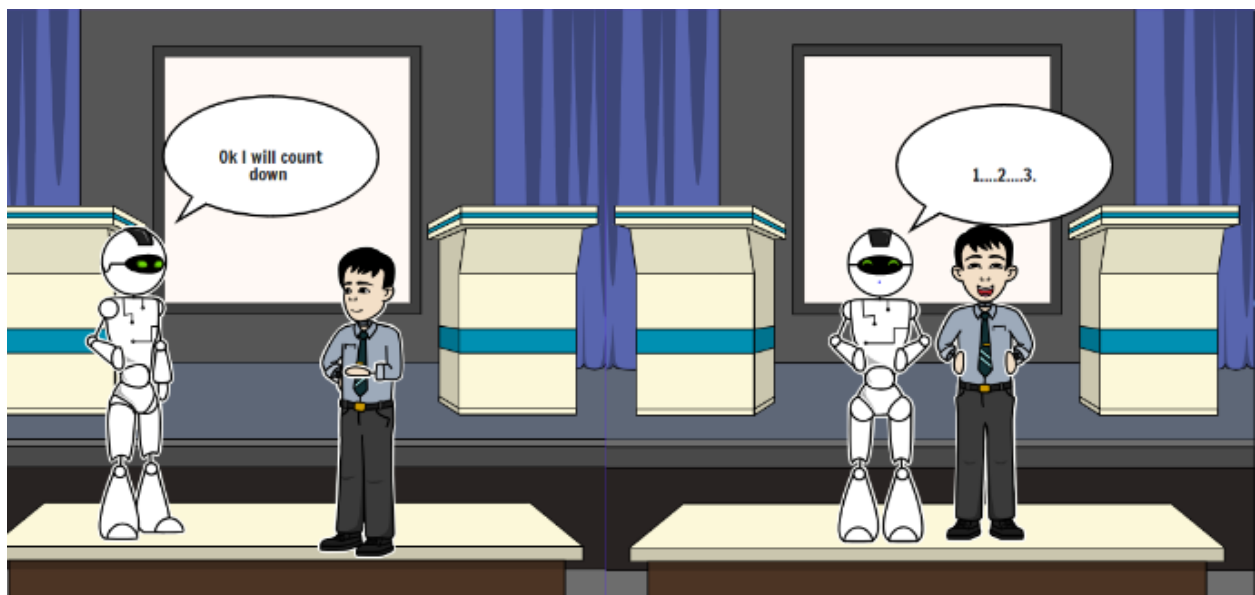
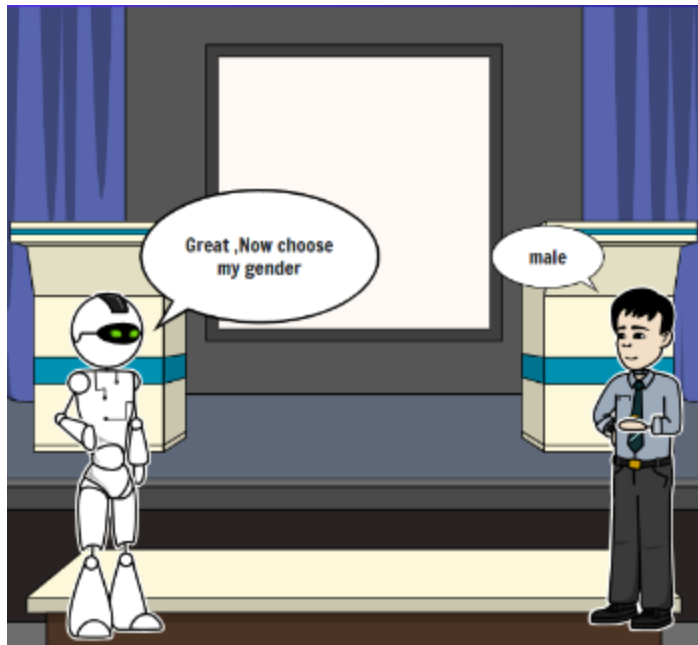
Use case 5: Take a Selfie

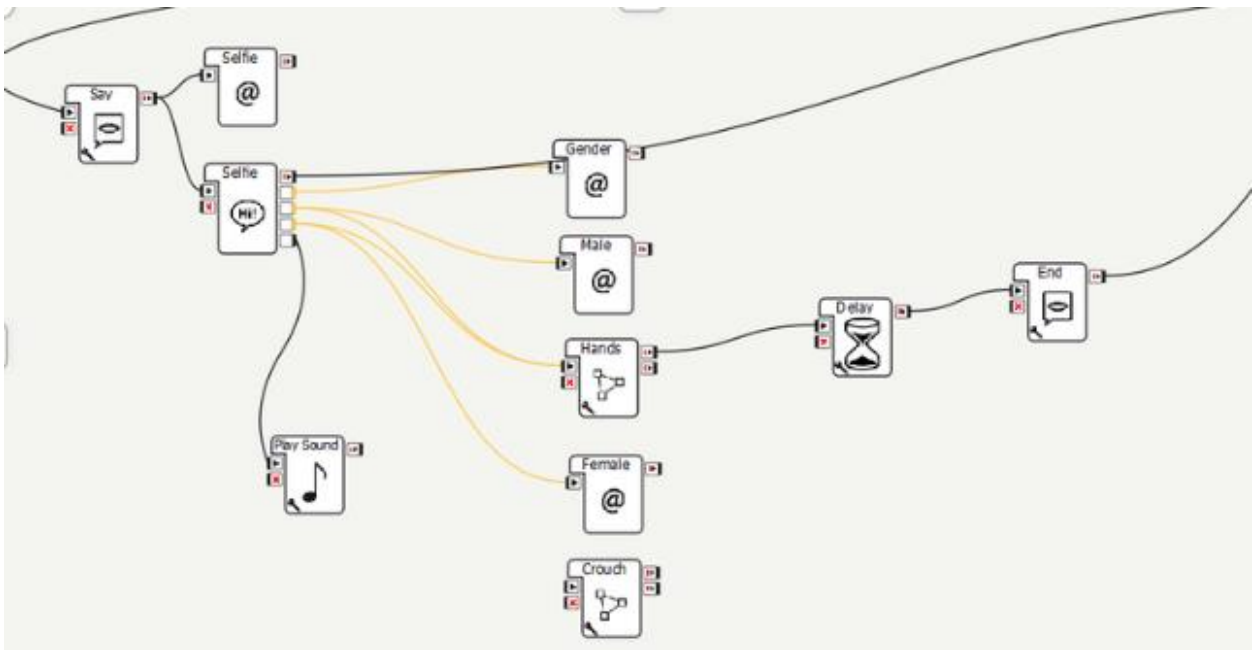
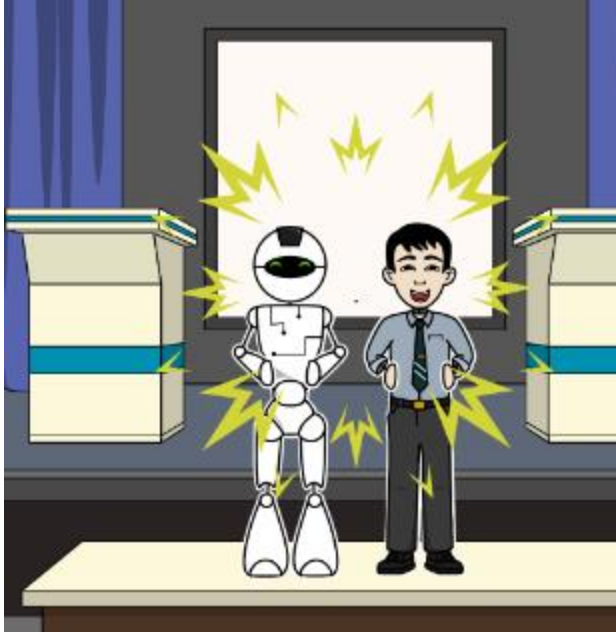
UC-ID and Name	Take Selfie		
Created By	Georges chamoun	Creation Date	13/2/2024
Actors	The main actor is the ceremony guest		
Trigger	The ceremony guest desires to capture a moment.		
Description (Objectives/Goals)	The objective is to enable ceremony guests to take selfies to capture memorable moments during the event.		
Preconditions	First Pepper should be on, and taking the position desires to take the selfie from the tablet when the eyes are green, we will take the picture		
Postconditions	So, the robot will take back the standard position and will ask the user if its good and he give him choice if he wants to retake a picture		
Action Sequence (Success Scenario)	<ol style="list-style-type: none"> 1. The ceremony guest indicates to take a picture as verbal, or he can push trigger on the pad 2. After that peper will tell the user to choose the gender 3. After that it will display peper gender image 4. It will take posture to be ready for the image with counting down 		

Extensions	
Requirements	<i>The selfie-taking process should be intuitive and user-friendly</i>
Storyboards	
Priority	Medium
Related Use Cases	
Assumptions	
Open Issues	Ensuring proper lighting and positioning for quality selfies.

Story board







Here is the complete workflow for the "take selfie" use case designed for the Pepper robot. The main dialog box includes all the robot's prompts and user interactions, with buttons for user input. The process starts with the robot asking if the user wants to take a selfie. If the user agrees, the robot guides them to a

new web view telling him to choose the gender of pepper male or female after that through positioning, counts down, takes the photo,

```
topic: ~selfie()
language: enu

concept: (approval) [yes sure "why not" absolutely]
concept: (denial) [no "another time"]

#selfie
proposal: %noselfie Have a great ceremony \pau=500\ $onStopped=1

u:(e:.onStart) Before proceeding to the ceremony, do you want to take a selfie with me?
  ul:(~approval) Great $choosegender=1 ^gotoReactivate(gender)
  ul:(~denial) Too bad ^gotoReactivate(noselfie)
u:(e:yselfie) Great $choosegender=1 ^gotoReactivate(gender)
u:(e:nselfie) Too bad ^gotoReactivate(noselfie)

#gender
proposal: %gender \pau=200\ Now choose my gender
  ul:(male) ^gotoReactivate(male)
  ul:(female) ^gotoReactivate(female)

proposal: %male $maleoutput=1 Ok i will count down, 3 \pau=500\ 2 \pau=500\ 1 \pau=500\ $picsound=1
proposal: %female $femaleoutput=1 Ok i will count down, 3 \pau=500\ 2 \pau=500\ 1 \pau=500\ $picsound=1

#button inputs
u:(e:maleB) ^gotoReactivate(male)
u:(e:femaleB) ^gotoReactivate(female)
```

we used dynamic implementations for all the images in the use case as shown in the picture

```
<body>
  <div class="" style="max-width:1100px">
    <div class="">
      <div class="">
        <img id="pageImage" class="" alt="Pepper Pic" width="600" height="650">
      </div>
      <div class="">
        <div id="informationSection" class="">
          <hr />
          <h1 id="pageHeading" style="font-size: 45px;padding-top:20px"></h1>
          <hr />
          <h3 id="pageText"></h3>
        </div>
        <hr />
        <div class="">
          # Set page details
          pageImage = "male.jpg"
          # Set the URL
          url = "http://(ip)/apps/(appName)/pages/displayimage.html?pageImage={pageImage}".format(ip=robotIP, appName=appName,
pageImage=pageImage)
```

Every button is activated to raise the event so its implemented in the code

```

<div class="w3-col m6 w3-padding-large">
  <div id="informationSection" class="w3-center">
    <hr />
    <h1 id="pageHeading" style="font-size: 45px;padding-top:20px"></h1>
    <hr />
    <h3 id="pageText"></h3>
  </div>
  <hr />
  <div class="w3-center">
    <button type="button" id="yesButton" onclick="raiseConfirmationEvent('yselfie')" class="fa rate-
      <span class="default-font">Yes</span><br>
    </button>
    <button type="button" id="noButton" onclick="raiseConfirmationEvent('nselfie')" class="fa rate-
      <span class="default-font">No</span><br>
    </button>
  </div>
</div>
</div>
<hr />
</div>
</body>
<script type="text/javascript">
  displayPageInformation();
</script>
</div>

```

```

u:(e:yselfie) Great $choosegender=1 ^gotoReactivate(gender)
u:(e:nselfie) Too bad ^gotoReactivate(noselfie)

```

```

</div>
<div class="w3-col m6 w3-padding-large">
  <div id="informationSection" class="w3-center">
    <hr />
    <h1 id="pageHeading" style="font-size: 45px;padding-top:20px"></h1>
    <hr />
    <h3 id="pageText"></h3>
  </div>
  <hr />
  <div class="w3-center">
    <button type="button" id="maleButton" onclick="raiseConfirmationEvent('maleB')" class="fa rat
      <span class="default-font">Male</span><br>
    </button>
    <button type="button" id="femaleButton" onclick="raiseConfirmationEvent('femaleB')" class="fa
      <span class="default-font">Female</span><br>
    </button>
  </div>
</div>
</div>
<hr />
</div>
</body>
<script type="text/javascript">
  displayPageInformation();
</script>
</html>

```

```

#button inputs
u:(e: maleB) ^gotoReactivate(male)
u:(e: femaleB) ^gotoReactivate(female)

```

So as we said after asking you if you want to take a picture with pepper if the user said yes or no the button yes it will raise the event and took us to another web page telling you to choose your gender if the user push the button male or female he will raise the event and display another web view showing you the picture needed and in same time it will activate the hands posture and start the count down

Personas

Persona 1:

 <h2>"I want to experience graduation again"</h2>		
Demographics	Name	Marie-Jeanne
	Age	25
	Major	Psychology
	Status	Single
	Work	Has her own clinic in Achrafieh and Broummana.
Motivation	Hobbies	Studying body language, reading, arts, meditation.
	Sports	Yoga and Tennis
	Future Goals	She wants to open a clinic in Dubai and Mozambique, and help the poor deal with insecurities and mental health.
	Personal challenges	She's struggling to overcome past trauma caused by her abusive ex-husband.
Social Environment	Background	Marie-Jeanne is a volunteer with MSF and volunteers every month in Africa to help kids.
	Relationships	Has trust issues because of her past experiences. Has only one friend whom she can tell all her secrets.

Persona 2:



“I can’t wait to graduate”

Demographics	Name	Jean-Philippe
	Age	15
	Class	Grade 10
	Field	Scientific
	School	Central College Jounieh
Motivation	Hobbies	Scout, interested in video games, geek
	Sports	Chess
	Clubs	Participates in the astronomy, robotics, ai clubs in his school and attends workshops.
	Personal challenges	Challenges himself to achieve academical excellence by studying hard. Struggles with social anxiety, doesn’t like to talk to people.
Social Environment	Background	Jean-Philippe lost his father at young age and sees a father in his older brother.
	Parents	After the loss of his father, Jean-Philippe is overly attached to his mother, so he works hard to make her proud.


Persona 3:



“I’m proud of my son”

Demographics	Name	Raymond
	Age	50
	Work	Sales Manager
	Address	Sarba-Jounieh.
	Number of Childrens	2
Motivation	Hobbies	Hunting, playing chess.
	Sports	Swimming and gym
	School	He graduated with a bachelor’s degree from Sagesse University.
	Personal challenges	He challenges himself to work overtime to make money for his family and educate them in top universities.
Social Environment	Background	He has a very high position in the company where he is working. He had a hard childhood, that’s why he wants his kids to experience a better one.
	Experience with technology	He hasn’t interacted before with a live Robot, because he isn’t up to date with all technology nowadays.

Persona 4:

 <p>“My brilliant cousin impresses me.”</p>		
Demographics	Name	Cesar
	Age	28
	Work	Software Engineer
	Address	New York City
	Number of Children	0
Motivation	Hobbies	Photography, hiking.
	Sports	Rock climbing, cycling.
	School	He graduated with a master’s degree in computer science from MIT. He challenges himself to continually learn new programming languages and emerging technologies.
	Personal challenges	Moving across the country for his dream job but being far from family.
Social Environment	Background	Experience with technology. He has been coding since he was a kid and even developed some popular mobile apps while in college.
	Experience with technologie	He focuses on robotics and artificial intelligence, and he is passionate about pushing the boundaries of robot capabilities.



“I want to experience graduation again”

Demographics	Name	Nassif el hachem
	Age	22
	Major	Computer science
	Status	Single
	Work	Has her own store
Motivation	Hobbies	Studying coding , reading, arts, Tennis .
	Sports	Swimming
	Future Goals	He want to have multiple branches across Lebanon
	Personal challenges	He have challenge to get the breakeven for his first branche
Social Environment	Background	He is a member in ieee
	Relationships	He is so social and like people and he love to organize events

Challenges

"Initially, I found working with the Pepper robot challenging because it was a bit unfamiliar. The error messages in Choregraphe weren't always clear, which made it difficult to diagnose issues when running projects. Sometimes, no error message appeared, but nothing would work, and the only solution was to reboot Pepper." ~ Maria El Mur

"First, the Pepper robot uses a different type of language, which is familiar but not popular. For this reason, it's hard to conduct research about interacting with Pepper. When designing the Scan QR Code use case, I found that the robot couldn't differentiate between two very similar photos, so I want to change the photos. Sometimes, when I'm running my project, the robot continues running the project individually, and I can't stop it except by rebooting it." ~ Anthony Masaad

"The work was intriguing, but I faced several challenges with various functionalities. For example, passing parameters between boxes was like a game—sometimes "<str>" worked for strings, and other times I had to use '<str>'. I also found the Pepper robot to be unstable and often needed to reboot it for it to work correctly. ~ Paul Chbat

"Previously, I struggled with Pepper's camera, not always recognizing similar images, like QR codes. So, I made each QR code a different color to help. Additionally, allowing the 'showImage' box to dynamically take the URL of the image as a parameter was a little bit difficult, but now it's working. These adjustments have significantly improved Pepper's image recognition capabilities. " ~ Elio El Khoury

Desing Patterns

In our project, we utilized several design patterns to enhance the interaction flow with the robot. The first one is the Q&A Pattern, where the robot asks the guest questions, and each party waits for the other to finish speaking before responding. This ensures a smooth and respectful conversation. The second pattern is the Interaction Initiation Pattern, where the robot initiates the interaction by waving and welcoming the guest. Lastly, we implemented the Guided Interaction Pattern, where the guest always has clear indications on the tablet to know the next steps in the interaction. This helps maintain a seamless and intuitive experience for the user.

Conclusion

To sum up, the Pepper robot project we worked on for the graduation ceremony demonstrated the possibilities and difficulties of utilizing robotics in these kinds of situations.

We successfully integrated important functionalities using Choregraphe, allowing the robot to greet guests, select their preferred language, communicate the ceremony rules, scan QR codes, direct guests, and even take selfies.

Overall, this project demonstrated the exciting possibilities of integrating robots into social events and set the stage for further innovation in the field.