Chatbot Systems as Backends for Conversational Embodied Agents

Student Name	Lewis King
University	Aston University
Module Code	CS3010
Module Name	Individual Project
Student Number	159025495
Contact Email	kingl@aston.ac.uk
Supervisor	Ulysses Bernardet
Contact Email	u.bernardet@aston.ac.uk

Word Count	8637*
Submission Date	29/04/2019

* The word count excludes content preceding the introduction, and content following the conclusion.

Project Definition

Student Name Lewis King
Course Name Computing Science BSc
Project Title Chatbot systems as backends for conversational embodied agents.
What is the project about? Personal trainers are becoming increasingly popular for people at the gym, and therefore demand is becoming higher. This means that personal trainers are not always available and can be booked up weeks in advance. As well as this, sessions with personal trainers can be very expensive, especially if people have multiple sessions per week. The intended creation is a personal trainer chatbot that can be used on personal devices that will be designed to provide workout plans for the user along with exercise demonstrations. With the intended creation, there won't be any need to wait for available slots with personal trainers and it will be free of charge.
What is the project deliverable? The project deliverable will include analysis of existing chatbot frameworks, the creation of a personal trainer chatbot using one of the available frameworks and then integrating the chatbot into a graphical virtual human system. The virtual chatbot deliverable will represent a personal trainer for end users, through visually demonstrating exercises and workout plans, as well as motivating the user with visuals and audio cues. The deliverable will also include an evaluation of the effectiveness and usability of the chatbot, obtained through systematic feedback.
What is original about the project? From initial research, it appears that common chatbots are text based, and aren't represented through virtual humans. Therefore this project will provide the visual human representation in a chatbot, and the project will simulate the face-to-face interaction the user would have with a human personal trainer.
Timetable showing main stages in work plan

22/10/2018 - Research existing chatbot frameworks, e.g. Facebook.
25/11/2018 - Review findings and choose appropriate chatbot framework to use.
26/11/2018 - Begin chatbot design phase.
14/12/2018 - Submit Term 1 Report.
23/12/2018 - Complete design phase, begin chatbot creation.
31/01/2019 - Complete chatbot creation.
01/02/2019 - Begin integrating with graphical virtual human system.
31/03/2019 - Complete deliverable, focus on final report.
25/04/2019 - Begin demo preparation, finalise report.
29/04/2019 - Submit Report (original date was 07/05/2019).
10/05/2019 - Demonstration.

List of Contents

Relevant Links

Code Repository
https://github.com/lewisbenking/PT-Bot
^ This includes the Unity project, the exported chatbot JSON files and other useful documents.

JimBot Demo
https://www.youtube.com/watch?v=8Ze7yBey87Y

JimBot Instructions
https://www.youtube.com/watch?v=BI1W-TGtWrs

NUnit Test Results
https://www.youtube.com/watch?v=BTxQP4J89wc

Trello Board
https://trello.com/b/zk2OfJxu/jimbot

Introduction

This document contains the documentation of the creation of the CS3010 Individual Project.

From experiences at the gym over several years, it was noticeable that personal trainers are usually with clients, and sessions with them can be extremely expensive in the long term. Another reason why the solution was chosen was because some people may be fearful of being judged at the gym, meaning they would prefer to exercise in the comfort of their own home.

JimBot is an embodied conversational agent. Embodied conversational agents are computer generated virtual humans represented on a device that are capable of maintaining conversation with human users through either verbal and/or non- verbal communication methods used by humans, such as speech and hand gestures (Chatbots, 2018) (MIT Press, 2018). JimBot can simulate the interaction that a person would have with a personal trainer.

The sections of the report below consist of the project *background*, *preparation*, *deliverable*. This is followed by an *evaluation* and a *conclusion*.

Background

Personal trainers are becoming increasingly popular for gym-goers, and due to this the demand is increased and outweighing the supply. As a result, personal trainers are more likely to increase the cost for their services, which prices people out of having regular PT sessions. Initial research has shown that it appears costs vary depending on the workout, for example using statistics found on Bidvine for the local area, “Strength Training” can cost £35 per session and in comparison, “Body Weight Training” can cost £74 per session (Bidvine, 2018.).

The intended deliverable is called JimBot, and JimBot can act as a virtual human personal trainer chatbot that can be used on personal devices, initially on a computer. The chatbot is designed to provide workouts for 5 different areas of the body: “Arms”, “Back”, “Chest”, “Core”, “Legs”. In future iterations, it would be possible to extend this to include other areas of the body. In addition to workout lists, JimBot can motivate the user through speech and provide exercise demonstrations through existing video footage.

With the intended deliverable there will be no need to wait for personal trainers to become available for a workout session and moreover it will be free of charge.

Existing Solutions

Chatbots

It became apparent that there were already several products relating to virtual personal trainers to varying extents.

The most widely used existing solution is an app called BodBot. BodBot is initially free to install from the Google Play and Apple App Store. BodBot provides a daily workout summary and sends push notifications reminding you of the workout and timings, so there is no need to actually open the app if you already know the exercises (BodBot, 2017).

Despite its advantages, unfortunately the company has locked the main features of BodBot behind a paywall. These features included fitness tests and FitBit connectivity. When opening the app, an overlay appears advertising their upgraded premium "BodBot Plus" version for £9 per month, £54 per year or a £215 lifetime membership. To add to the disadvantages, the UI was difficult to navigate and there were bugs such as the BodBot Plus overlay appearing and not disappearing, and this stopped functionality from working. Roughly ⅓ of exercises available to choose from didn't have a video demonstration nor a text description, meaning for people would struggle to perform the exercise, or use external sources, if they were unfamiliar with it.

Alternatives to BodBot can be seen in *table 1* below, from research carried out, none of these pieces of software incorporated a virtual human, and most lacked any form of motivating techniques, which can be used to humanise the chatbot.

Table 1: BodBot alternatives and their main feature.

Software	Main Feature	Cost	Platform
Forksy AI (Forksy, 2018.)	Text based chatbot that lets users store a food diary of daily nutrition summary.	Free trial, contact the company for subscription pricing.	Facebook Messenger
Count.it (Green, 2016.)	App that gives a Count.It score based on physical activity, e.g. 10,000 steps is equivalent to 100 Count.It score. Has bad reviews on the App Store!	Up to $4 per user, per month.	FitBit Garmin Apple Watch
Daily FitBot	Gives push notifications and sends a 15-20-minute workout session.	Pricing is unavailable on their company website. Assumption is this chatbot is free to use on Messenger.	Facebook Messenger
Fitly (BotList, N.D.)	Offers specific and targeted training options, you can share workouts with friends on Messenger	Free trial, contact the company for subscription pricing.	Facebook Messenger
Peter (BotList, N.D.)	Specialises in motivation, can choose different types of character e.g. Yoda / Sergeant	Free trial**	Facebook Messenger
Gymi Personal Trainer (BotList, 2016.)	“Workout of the day”. Look up exercise demonstrations and routines for guidance, option to purchase workout products through native payments in Messenger.	Free trial**	Facebook Messenger

** These chatbots have allegedly since been deprecated.

Chatbot Frameworks

There are various chatbot frameworks being used at present, and several major companies have created their own frameworks, Adnan Rehan was helpful in describing some of the following “best” chatbot development frameworks (Rehan, 2018.):

Google

DialogFlow, formerly API.AI, was acquired by Google and rebranded. It is powered through Google’s machine learning to connect with Google Assistant, Amazon Alexa amongst other applications.

Microsoft

Microsoft developed the Bot Framework. This utilises features on Azure to build and maintain chatbots with active learning, and can be incorporated with Microsoft products, such as Office 365, and services such as Cortana.

IBM

IBM Watson was created on a neural network containing over one billion words on Wikipedia, and responds to natural language input by utilising the in-built machine learning algorithms.

Embodied Conversational Agents

Greta

Greta is a multilingual agent with a 3D model of a woman. She is able to communicate using a variety of both verbal and non-verbal behaviours, whilst showing facial expressions, gestures and head movements. Greta can be used for different roles such as a virtual tutor or an actor in games (Chatbots, 2002.).

Nadia

This virtual agent was created in 2017 by Soul Machines and was created for the benefit of Australians with disabilities; to help them access the National Disability Insurance Scheme. It’s voiced by Cate Blanchett and uses “Emotional Intelligence” through webcams to adjust her answers based on the user’s emotional state (MAACK, 2017.).

Connectome

This technology is in development from Couger. Couger have uploaded a video to their YouTube channel of an in-car embodied human agent showcasing natural spoken interaction with the passengers in the vehicle. An example of this was the human agent recommending places to eat in the vicinity and suggesting music to listen to (Couger, 2018.).

Summary

Based on the research conducted, there are a number of existing personal trainer chatbot solutions. However, these solutions did not meet the exact project requirements. This was because the existing solutions were solely text-based chatbots, meaning they didn’t incorporate interactive virtual human character models and there was no opportunity to send an audio input. Many solutions also locked features behind a paywall.

There has been a vast improvement in recent years regarding the capabilities of conversational embodied agents, with uses for people with disabilities or connecting to their cars. However, there is currently no existing solution that incorporates both a personal trainer chatbot, and a embodied human character.

It was clear based on the findings above that the preferred chatbot framework to use would be Facebook Messenger. Despite this, the chatbot framework chosen to use for this project was DialogFlow by Google. This was based on personal preference after comparing the different frameworks, and the personal uncertainty of using the Messenger API to create the chatbot.

Preparation

Requirements Analysis

This section describes the process of creating functional and non-functional requirements from collected user stories from 10 people, and then shows a comparison of generic chatbot features with the newly formed requirements.

User Stories

To begin *Requirements Analysis*, user stories were collated from 5 men and 5 women ranging from 21 to 60+, from different backgrounds and abilities. These 10 people will also be involved for the testing phase of the project, explained under the *Testing* heading.

They were asked these questions:

1) “What would you want from going to the gym?”
2) “What would you want from a personal trainer?”

Their responses were noted down, and then converted into user stories based on this generic template:

“As a… I want… So that…”

Some of the responses given from the 10 users were similar, and were therefore converted into a single story, and the user stories created can be seen below in *table 2*.

Table 2: User stories

ID	User Story
1	*As someone who is busy* *I want to interact with a PT in my own time* *So that I can work out whenever I feel like.*
2	*As someone who has started going to the gym* *I want to see demonstrations of exercises* *So that I can use the correct form.*
3	*As someone who goes to the gym* *I want workout plans*

	So that I can tailor my workouts based on my needs and abilities.
4	*As I have recently dislocated my shoulder* *I want relevant exercises* *So that they can help to rebuild and strengthen my shoulder.*
5	*As a user* *I want to see visual demonstrations of workout exercises and text instructions* *So that I can correctly perform the exercises with additional guidance.*
6	*As a user* *I want to see realistic animations and real-time responses* *So that my experience is satisfactory.*
7	*As a user* *I want to set my own goals with the Personal Trainer* *So that I have targets to aim for.*
8	*As a user* *I want to have fitness tests* *So that I can monitor my progress.*
9	*As a user* *I want nutritional recommendations* *So that I can change my diet.*
10	*As a returning user* *I want the chatbot to recognise me when I open it* *So that my user experience is satisfactory.*
11	*As someone who is embarrassed to go to the gym* *I want to work out in my home* *So that I can lose weight.*
12	*As someone who is busy* *I want to have spinning classes in my own time* *So that I can complete the class in my own time.*

It was explained to the respondents that following reflection, it would not be feasible to achieve every user story above in the time frame individually.

Requirements Specification

For the requirements specification, the user stories collated from *Requirements Analysis* were expanded upon to form functional and non-functional requirements.

Functional Requirements detail actions or activities that the system can perform, along with actions the user can do on the system, for example…
“Can I log in?”

Non-Functional Requirements
Non-functional requirements describe how well a system should behave, for example…
“Can I log in within 5 seconds?”

Following this, MoSCoW analysis was applied to the newly formed requirements in order to detail the most essential requirements for the current timeframe, as shown below.

MoSCoW Analysis

MoSCoW Analysis is a technique used to prioritise requirements. This technique is popular for projects adopting an agile development approach with a fixed deadline, meaning the focus should lie on the essential requirements. These requirements are separated into the following categories:

Must Have
These requirements will be deemed critical for the project to be a success. If all *Must have* requirements are met in the current timeframe, the project can be considered a success, therefore if at least one *must have* requirement is not met, then the project can be considered a failure. *Must have* requirements can be downgraded to *should have* providing all stakeholders have agreed, or new requirements have a higher importance.

Should Have
Should have requirements can weigh as highly as *must haves*, although they do not necessarily require completion within the current timeframe.

Could Have

These requirements are usually included in the current timeframe providing there are enough resources to do so. *Could haves* are not essential for the project to be deemed a success, and they could improve the end user experience.

Won't Have (this time)
Won't haves are agreed by stakeholders as inappropriate for the current timeframe, or deemed the least critical requirements.

Table 3: MoSCoW analysis

ID	Title	Functional / Non-functional	Description	MoSCoW
1	Open Application	Functional	Can I open the application?	M
1.1	-	Non-Functional	Can I open the application 24/7?	M
1.2	-	Non-Functional	Can the application open and load within 30 seconds?	M
2	General Communication	Functional	Can I communicate with JimBot?	M
2.1	-	Functional	Can JimBot communicate with me?	M
2.2	-	Non-Functional	Can communication occur in real-time?	M
2.3	-	Functional	Can JimBot motivate the user to keep exercising?	S
2.4	-	Non-Functional	Does JimBot look realistic?	M
3	Workouts	Functional	Can I request workouts?	M
3.1	-	Functional	Can JimBot output workouts?	M
3.2	-	Non-Functional	Can JimBot output workouts in real-time?	M
4	Exercise	Functional	Can I request exercise	M

	Descriptions		descriptions by interacting with the screen?	
4.1	-	Functional	Can JimBot output text exercise descriptions?	M
4.2	-	Non-Functional	Can JimBot output descriptions in real time?	M
5	Muscles Affected	Functional	Can I request the muscles affected by the exercise by interacting with the screen?	M
5.1	-	Functional	Can JimBot output the relevant muscles diagram?	M
5.2	-	Non-Functional	Can JimBot output diagrams in real-time?	M
6	Exercise Demonstrations	Functional	Can I request a demonstration of the exercise by interacting with the screen?	M
6.1	-	Functional	Can JimBot play a video of the exercise?	M
6.2	-	Non-Functional	Can JimBot play the video in real-time?	M
6.3	-	Functional	Can JimBot animate to show the exercise?	C
6.4	-	Non-Functional	Can these animations happen in real-time?	C
7	User Profile	Functional	Can I create my own user profile?	C
7.1	-	Functional	Can JimBot recognise different users?	C
7.2	-	Functional	Can I customise my profile with gender/height/weight?	C
8	Nutritional Guidance	Functional	Can I request nutritional advice?	C

8.1	-	Functional	Can JimBot provide nutritional advice?	C
9	Rehabilitation	Functional	Can I request rehabilitation exercises?	C
9.1	-	Functional	Can JimBot output rehabilitation exercises?	C

Generic Chatbot Requirements

After creating the above User Stories and requirements, it was necessary to gain an understanding of generic chatbots and their features.

From the research carried out, a diagram, *figure 1*, was created to illustrate the understanding of generic chatbots.

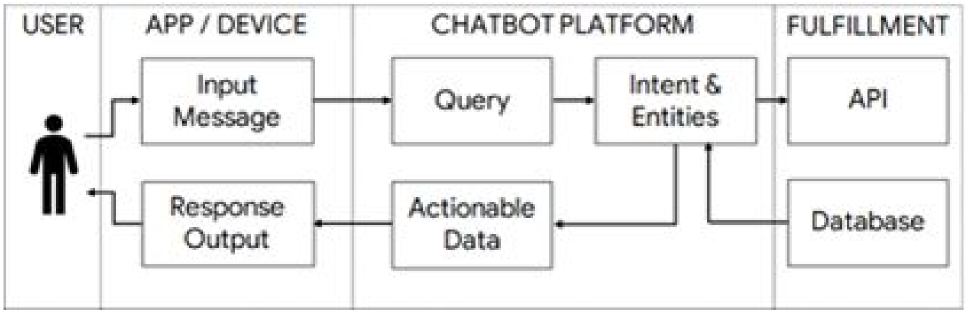
Users can query a chatbot by inputting a message through text or occasionally speech, on their personal device, whether that be a laptop or mobile phone.

The chatbot can use NLP to extract the user intent and entities from the input message, enlisting context mapping and all possible dialog paths to determine the context of the user input. This can help to control the chat direction and conversation flow.

By doing this, it helps to prevent the user from taking an unexpected path. When the chatbot has interpreted the intent/entities, it will perform external API calls which return a response / actionable data and the chatbot sends the response output to the users' device.

Depending on the chatbot framework, analytics can be enabled which will create crash logs to help identify bugs that when fixed, will help optimise the chatbot and improve the user experience. The logs can also help to identify features that aren't being used, so could be removed in future iterations for conservation of resources.

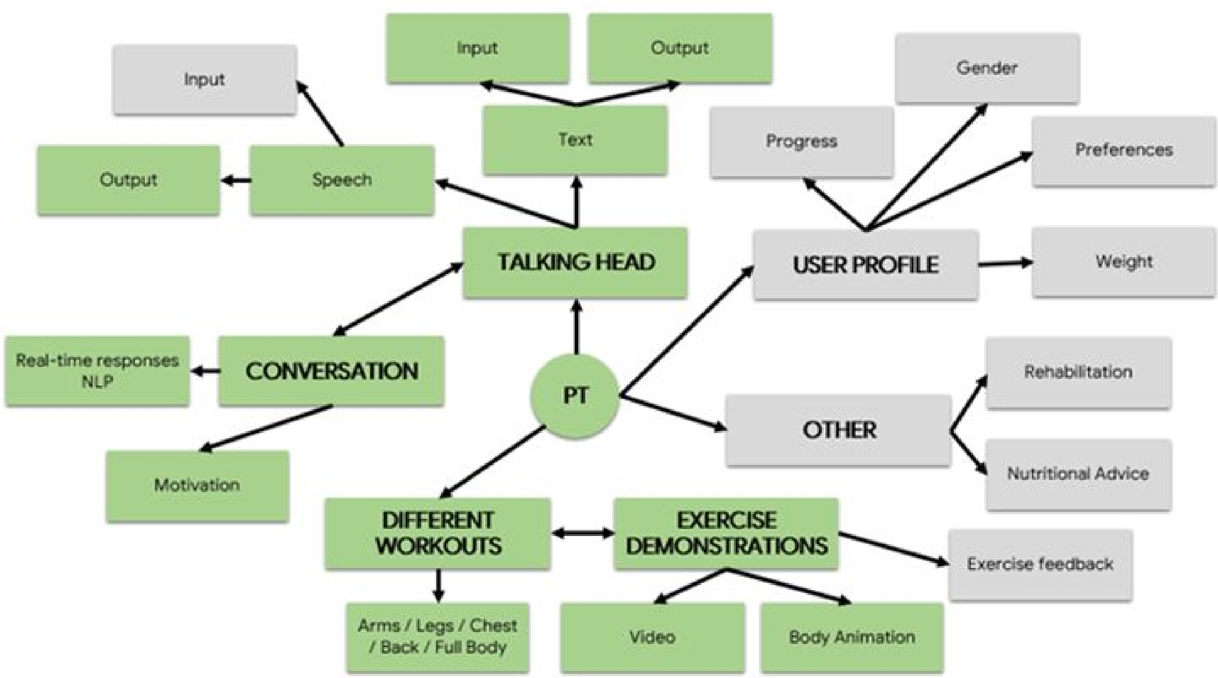
Figure 1: Generic chatbot diagram.



JimBot Chatbot Requirements

To demonstrate specific requirements for JimBot, another diagram, *figure 2*, was created. The requirements in the diagram below can be identified by their colour. The green colour indicates “must-have” features and the grey colour represents features that are “nice-to-have” (due to time constraints).

Figure 2: JimBot requirements.



Underneath this, *table 4* compares the JimBot Requirements mind map with the Generic Chatbot Requirements diagram.

Table 4: Comparing JimBot and generic requirements.

JimBot Requirements	Generic Chatbot Feature
Conversation Real-time responses Motivation	Input text or speech > NLP > Output text or speech
Talking head	Input text or speech > NLP > Output speech
Different workouts Arms / back / chest / core / legs	Call API > Query database > Get actionable data > Output response
Exercise demonstrations Video Animations	Call API > Query database > Get actionable data > Output response
User profile Progress / gender / weight / preference	Input data > Call API > Store on database
Other Nutritional advice	Output response

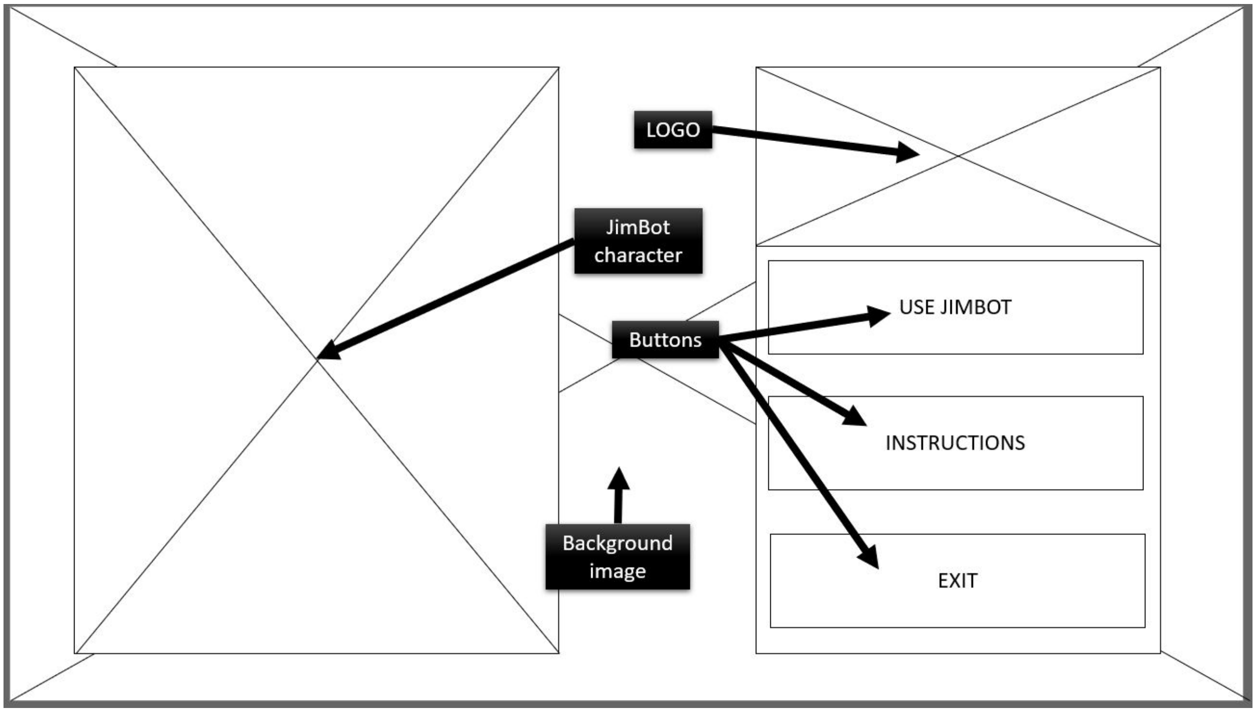
HCI Designs

These HCI designs will represent the human interaction with the user interface, therefore the front-end of the application. To create the HCI designs, a technique called wireframing was used.

Wireframes can be used to help visualise a future creation. They do not contain specific features, for example the presence of images is indicated with a cross through the box. These wireframes also do not specify the colour or font properties that will be used, but the layout of the final product has evolved from these wireframes.

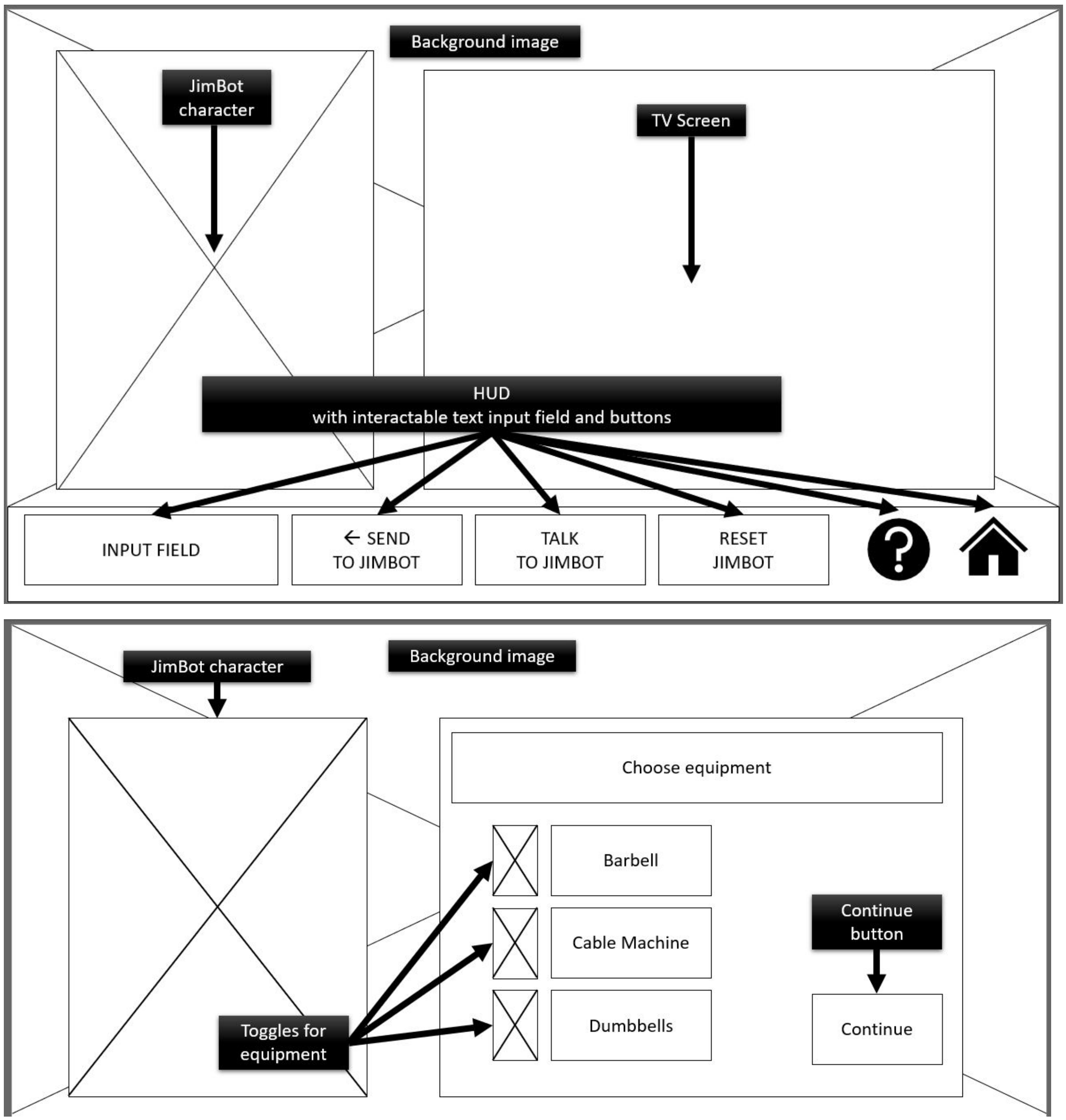
Figure 3 shows the starting screen wireframe diagram. This will be the first interaction the user has with the JimBot. JimBot will stand on the left pane of the screen and the menu bar with three selectable buttons will be displayed on the right of the screen.

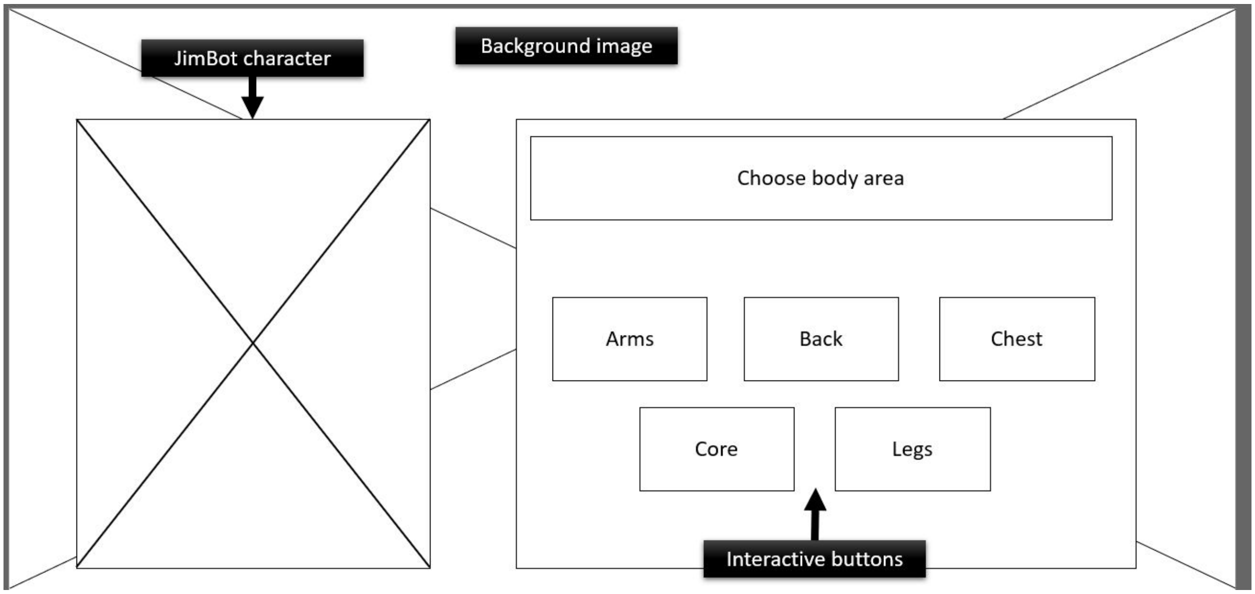
Figure 3: Starting screen wireframe.

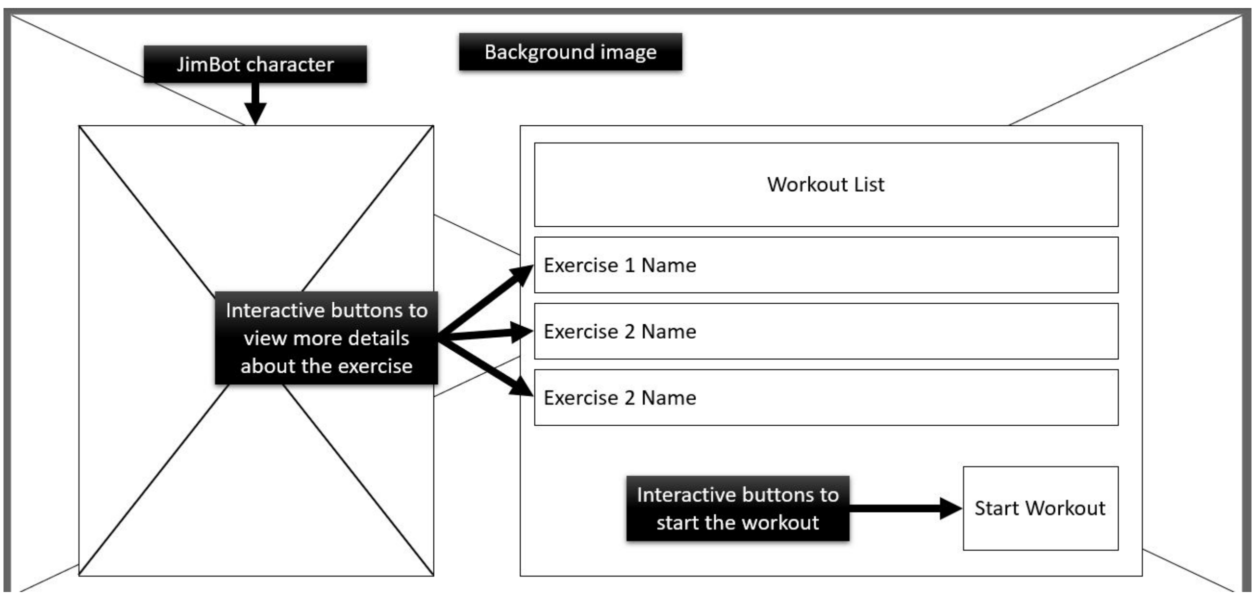


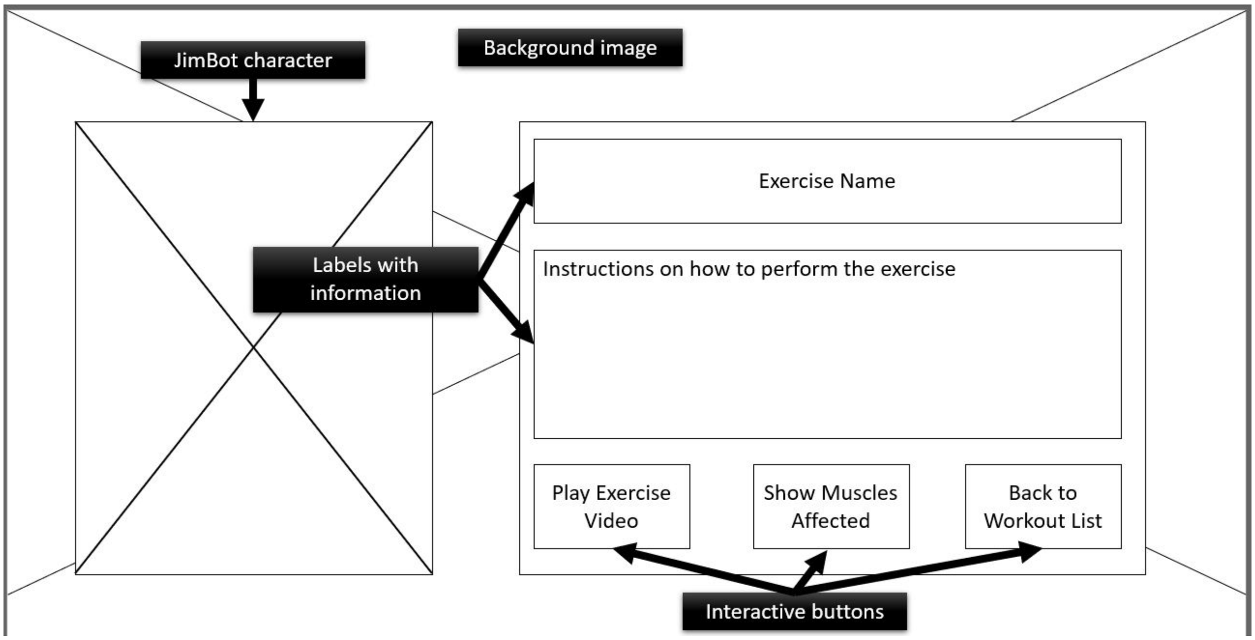
When the user selects Use JimBot from the menu shown above, they will be redirected to the JimBot scene. This screen contains JimBot, again standing on the left side for consistency purposes, and a large TV screen with interactive elements on the right side. Underneath the main content is a HUD with interactive elements for the user to interact with if they would prefer using this instead of the TV screen. For each choice the user makes on the TV screen or the HUD, the TV screen will update to display different information, with examples including exercise videos or the muscles affected diagram, as shown in *figure 4*.

Figure 4: JimBot scene wireframes.

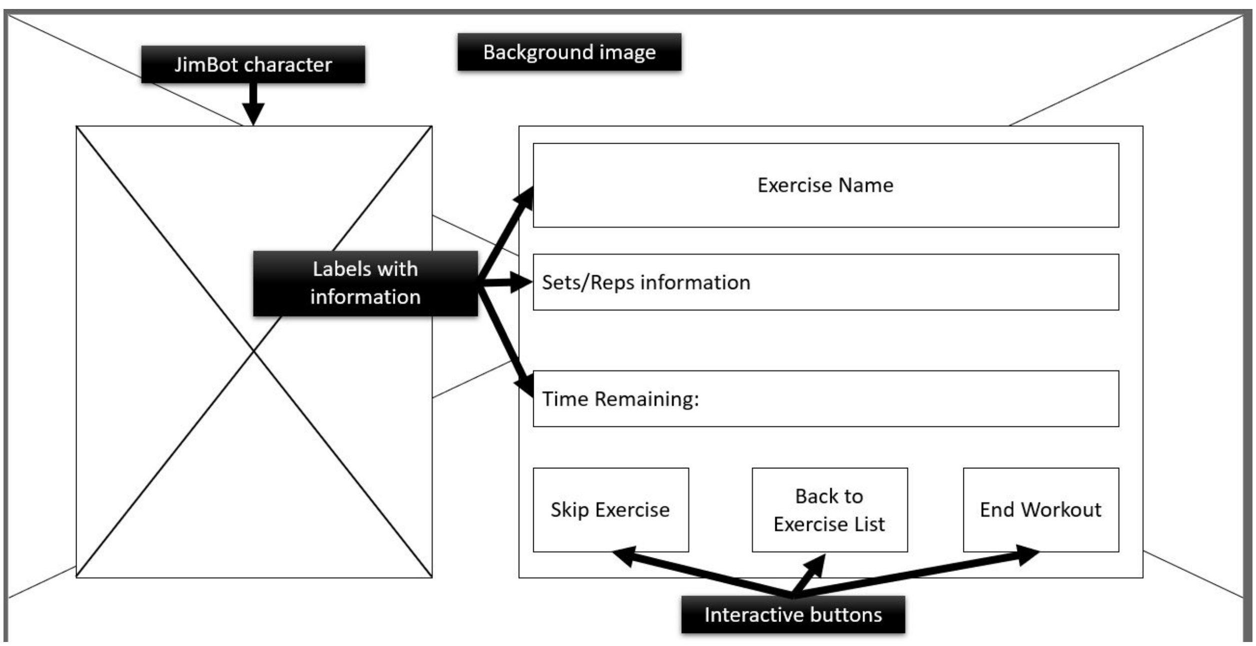



JimBot character
Background image
Choose body area
Arms
Back
Chest
Core
Legs
Interactive buttons

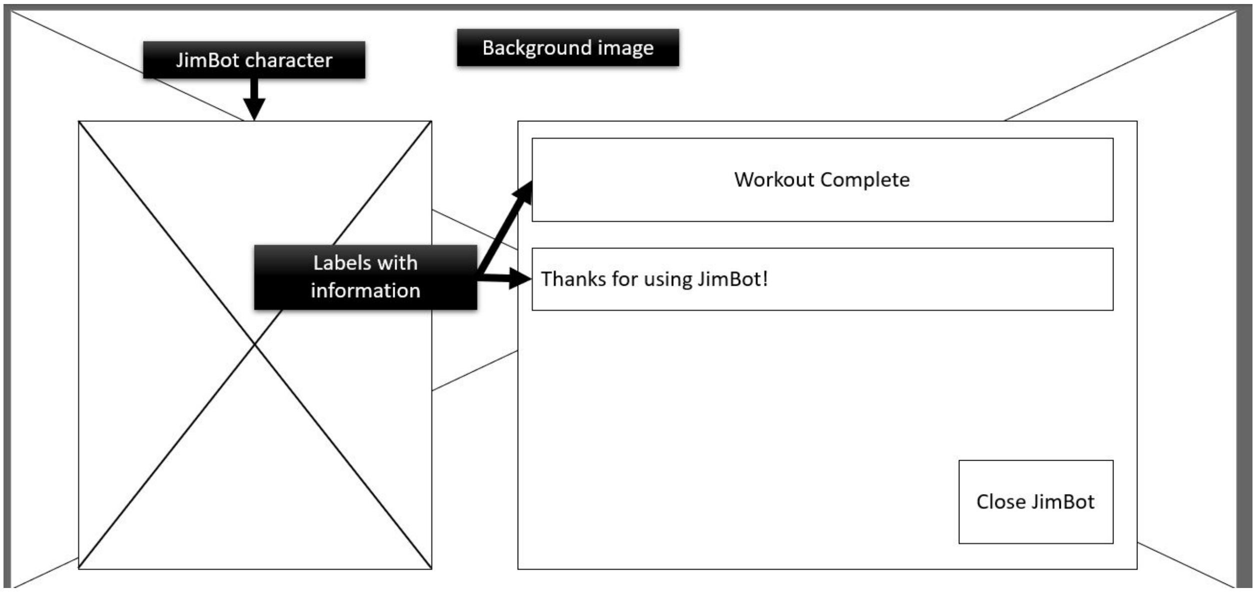

JimBot character
Background image
Workout List
Exercise 1 Name
Exercise 2 Name
Exercise 2 Name
Interactive buttons to view more details about the exercise
Interactive buttons to start the workout
Start Workout



JimBot character
Background image
Exercise Name
Labels with information
Instructions on how to perform the exercise
Play Exercise Video
Show Muscles Affected
Back to Workout List
Interactive buttons

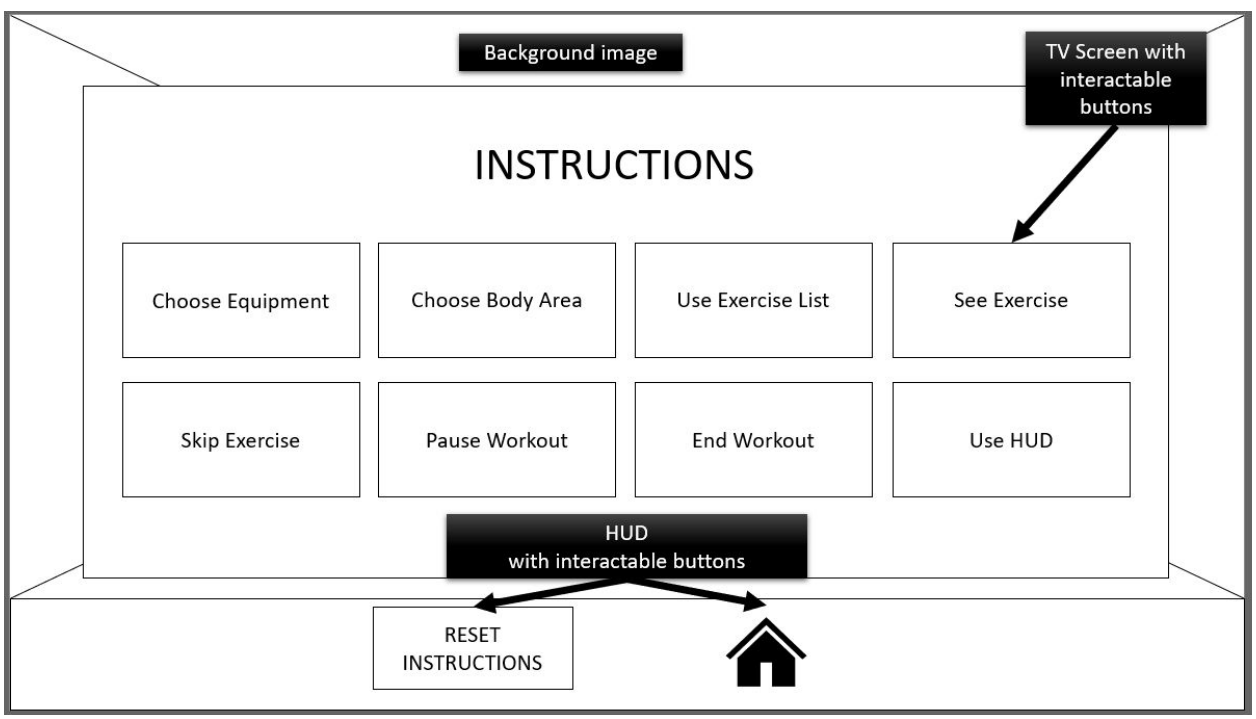


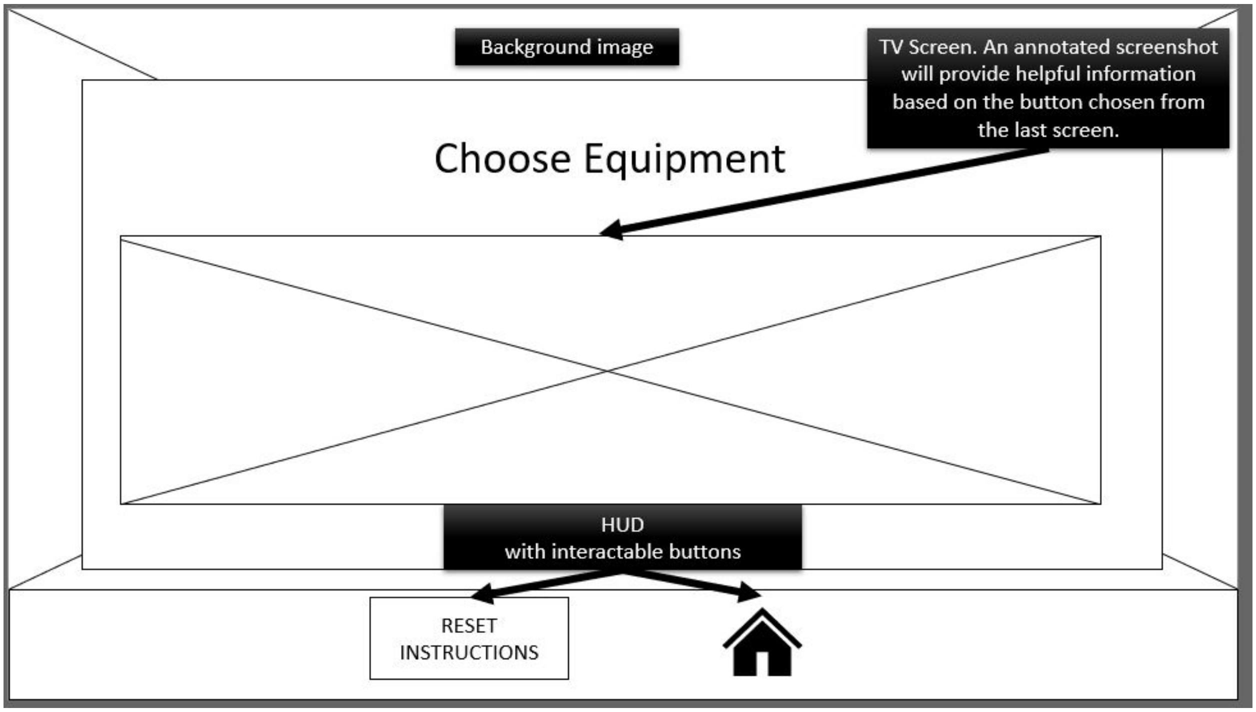
JimBot character
Background image
Exercise Name
Labels with information
Sets/Reps information
Time Remaining:
Skip Exercise
Back to Exercise List
End Workout
Interactive buttons



If the user decides to select the Instructions button from the starting screen, they will be directed to a different scene instead of the JimBot scene. The instructions screen at first displays interactive elements, and when one of these is selected, the screen will update to show specific advice on how to use a component of JimBot. All advice sections will share the same layout for consistency purposes, as shown in *figure 5*.

Figure 5: Instructions wireframes.





Deliverable

Software Methodology

Work leading to this deliverable product has adhered to industry recognised stages, and examples of these include analysis, design and implementation, testing and maintenance.

For creating the deliverable, it was decided an appropriate development approach to use would be Scrum.

Despite Scrum being best utilised by a larger team of people, there were benefits for this individual project, primarily due to the fact that tasks can be prioritised by order of importance. These tasks are then completed in sprints which for this project lasted 2 weeks. Furthermore, tasks can be flexible throughout the product development, which is advantageous if requirements are discovered to be unfeasible or require modification.

Documenting Tasks

The website Trello was used in this project to document tasks. By using a Trello board, tasks can be assigned to cards in different columns, or moved between columns. This can help show the progress of the project and also in a working environment stakeholders can have a clear overview of the project and progression.

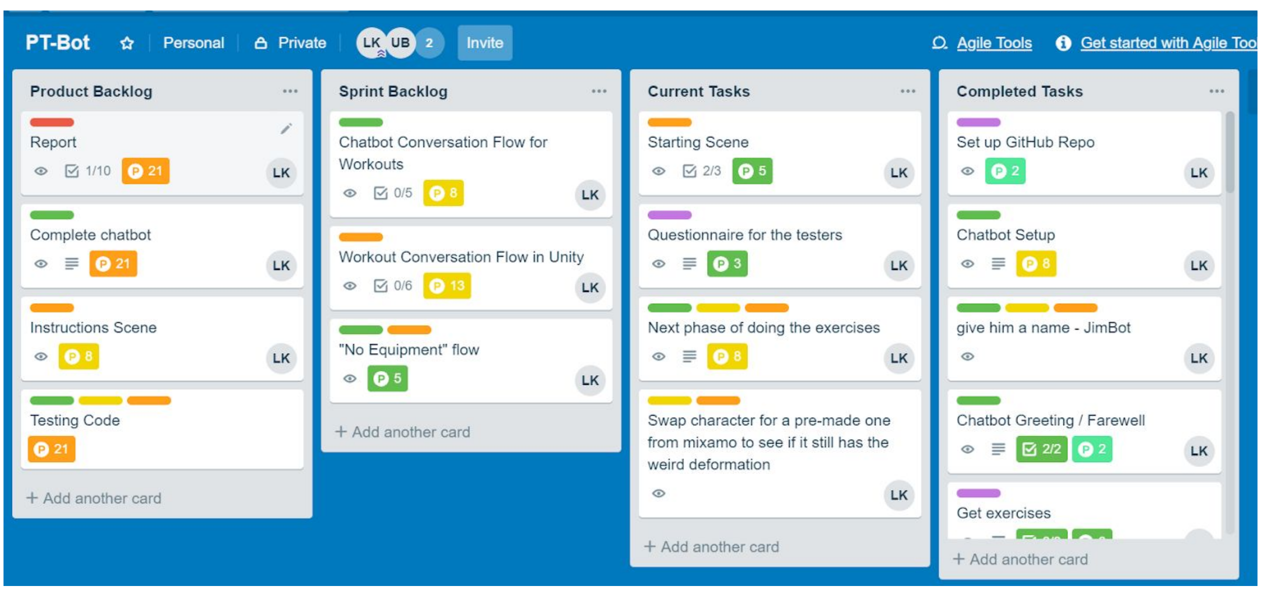
In terms of this project, and based on the scrum methodology, 4 columns were created for: 1) Product Backlog. 2) Sprint Backlog. 3) Current Tasks. 4) Completed Tasks.

For each card (stating a task), story points were assigned (representing the Fibonacci sequence), and these story points are there to indicate the complexity of a task, from 1 being least complex to 21 being most complex, as shown in *figure 6.*

The Trello board was updated at least weekly, primarily following the arranged weekly supervisor meetings, and the Trello board can act as a ‘project diary’.

This link will provide a clearer view of the entire Trello Board.
    https://trello.com/b/zk2OfJxu/jimbot.

Figure 6: Trello board documenting tasks.



Slack was used throughout the project as a means of communication, in order to ask questions and provide updates about progression outside of the regular weekly meetings.

Code Repository

GitHub was used to store the codebase. GitHub provides a clear layout with full history of the development progress, so was suitable to use for this project.

This link will provide access to the GitHub repository
https://github.com/lewisbenking/PT-Bot

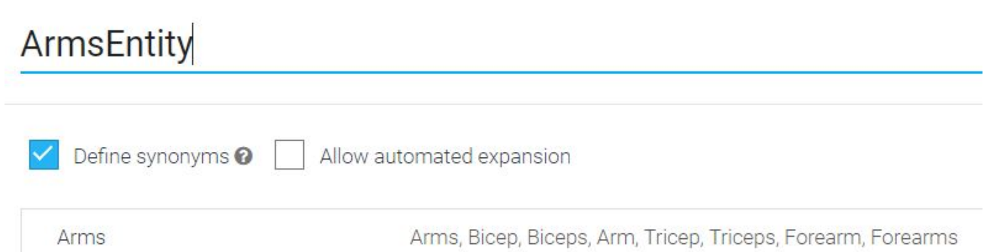
Development & Implementation

Chatbot

Upon reviewing existing chatbot frameworks, it was decided the most suitable framework to use for this project would be DialogFlow. DialogFlow is owned by Google, and incorporates their Google Assistant, which is currently the most accurate and successful assistant.

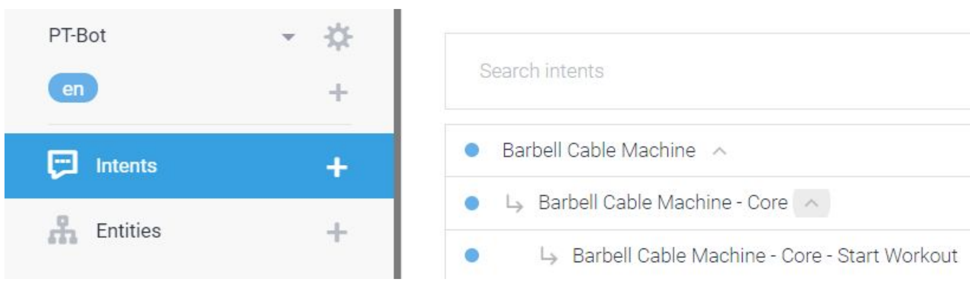
For this project specifically, it was important to create different entities representing the different areas of the body to train. By doing this, when the user asks for a specific workout, the chatbot will be capable of extracting the input entity and determining which type of workout to respond with. Five entities were created for arms, back, chest, core and legs, and each entity defines synonyms. An example of this can be seen below in *figure 7;* the "arms" entity contains "arms", "bicep", "biceps", "arm", "tricep", "triceps", "forearm", "forearms".

Figure 7: Entity for arms



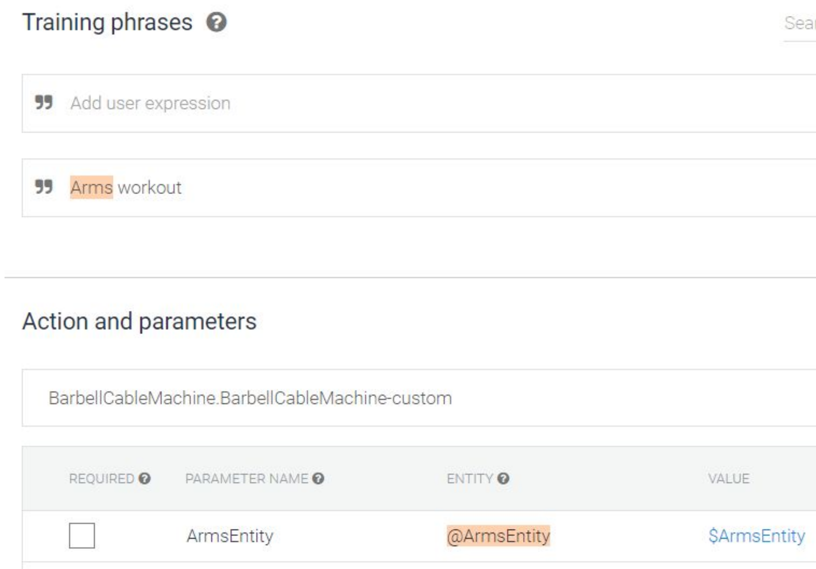
Following the creation of the above entities, it was necessary to create the intent flows. Throughout the creation of the project, the intent flows were modified to suit the changes in the Unity project. A benefit of using DialogFlow is that it is possible to use contexts to link the intents together, meaning the parameters from the first intent will get passed onto the next intent and so on. *Figure 8* shows the intent flows for one of the paths a user can take. For this example, they would have chosen ‘Barbell and Cable Machine’ from the interactive screen in the Unity project, and then chosen ‘Core’, and finally ‘Start Workout’.

Figure 8: One of the intent flows for the chatbot



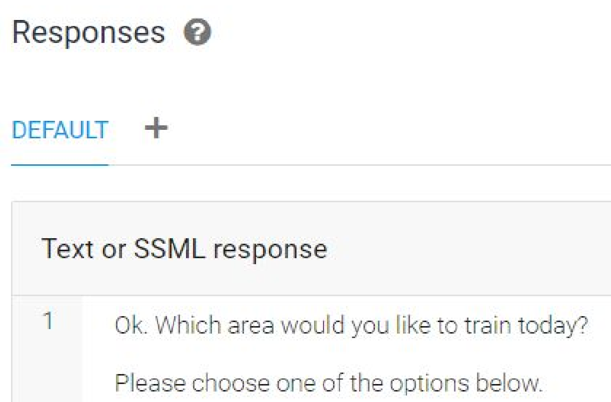
Inside each intent, specific training phrases can be assigned for the chatbot to extract from the user input. *Figure 9* shows an example of this for the arms workout intent. As the arms entity was created as shown above, DialogFlow recognises this in the *Action and parameters* section, so now the synonyms for arms can be used and can be recognised as well.

Figure 9: Training phrases on intents



Following *Training phrases* is the *Responses* section, and this is where suitable responses are entered for the chatbot to respond with based on the training phrases. *Figure 10* shows an example of this based on the equipment chosen by the user, for example ‘Barbell and Cable Machine’. In addition to the text response shown below, the chatbot also outputs an audio file, for this project it is a .wav file. This is extremely useful for this project to humanise the chatbot when it is integrated with the virtual human character model.

Figure 10: Example chatbot response



Virtual Human Character Model

Adobe Fuse CC is an application created by Adobe that enables the user to create their own custom 3D models or characters and then export for their projects, for example Photoshop. Adobe allow the user to select from existing assets and then adjust values with sliders and other options to dynamically update the appearance of the character model.

Fuse was a beneficial tool to use in this project because it parameters can be tweaked to generate a realistic and suitable virtual personal trainer character model, and the results can be exported into Mixamo.

Mixamo rigs the character imported from Fuse automatically meaning the user can have a fully skinned and weighted skeleton in the 3D mesh in minutes. Then a pre-made high quality animation from the store can be applied to the character, and the animation will instantaneously retarget to the 3D character. This means that the animation can be applied to any specific character model.

Unity Project

This phase of the project involved linking the DialogFlow chatbot to the virtual human character created from Fuse and Mixamo. The created Unity project contains three different scenes called “StartingScreen”, “JimBot” and “Instructions”. The User Interface has evolved from the wireframes created under *HCI Designs*.

It was important to control as much of the program flow as possible, and handle any expected and unexpected user input. The user journey commences on the starting screen, and then they proceed to the JimBot scene. Here, users are asked for their equipment preference, followed by their body area preference. A workout list of up to 3 exercises will be returned on the screen and these are interactive, with each providing an option to read the exercise instructions, view an exercise video or see which muscles are affected. There is an option to begin the workout with the workout list, and when the workout begins, a timer will count down from 10 minutes for each exercise, giving the user more than enough time so they can go at their own pace. There are options to skip the exercise or pause the workout if the user wanted to watch the exercise video again. The final screen when the workout finishes signals the end of the user journey.

Assets

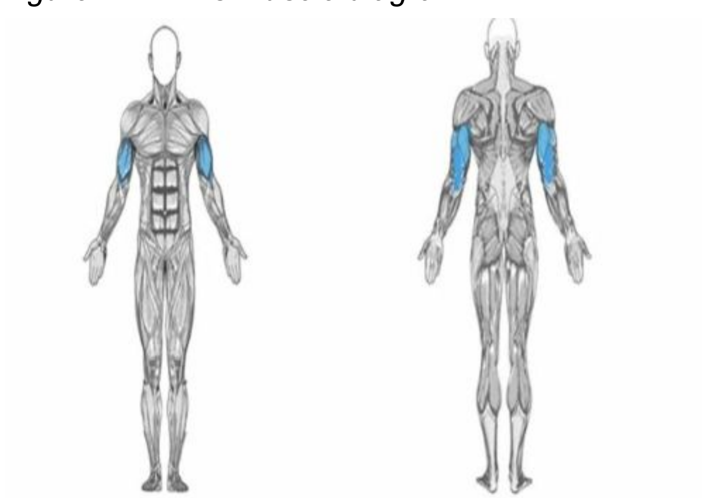
Sprites

After experimenting with different images, an an appropriate background image showing a gym was found on Pinterest (Pinterest, 2019), to fit with the planned aesthetic of the UI, as shown in *figure 11*. Window lighting was then applied to the background for a realistic effect.

Figure 11: Background image used.

Diagrams for the muscles affected for each workout (arms, back, chest, core and legs) were sourced from the same web pages used to paraphrase the exercise details from the BodyBuilding website (BodyBuilding, N.D.). *Figure 12* shows one of the diagrams used in the project.

Figure 12: Arms muscle diagram

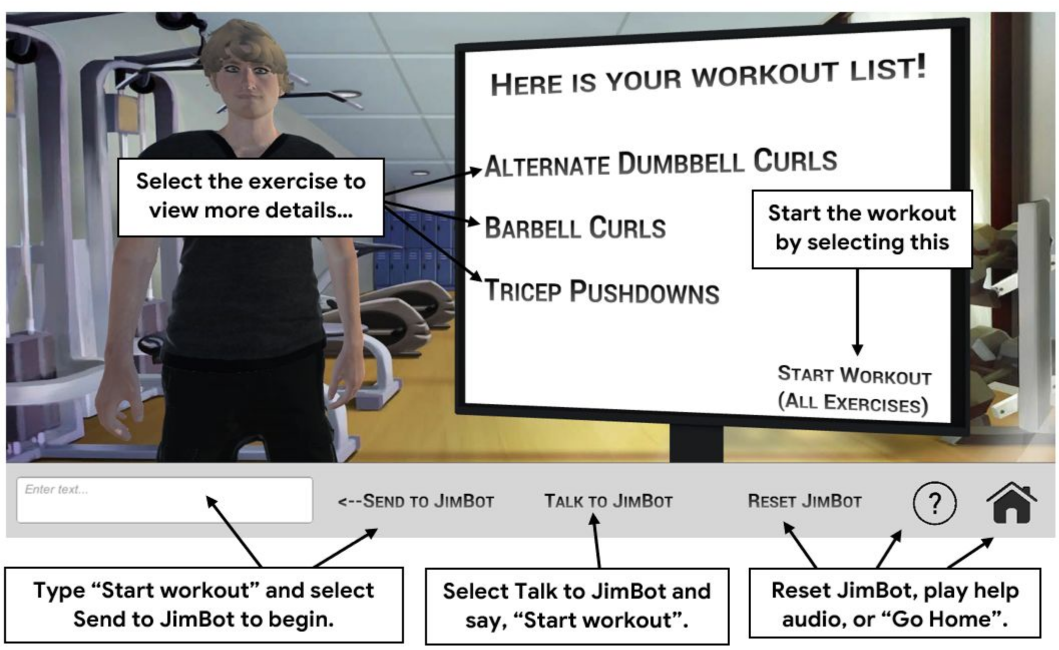


Icons used on the HUD for the home and help buttons were sourced from contributors on the KissPNG website (Duway, N.D.), (Akaeira, N.D.). The JimBot logo was also created using an image found from a Google search (SammlungFotos, N.D.), and was then personalised by adding text above and below the image, as shown in *figure 13*.

Figure 13: JimBot logo.

For the instructions scene, screenshots were self-taken from portions of the JimBot scene to guide the user through each stage of the application, and an example of this is shown in *figure 14*.

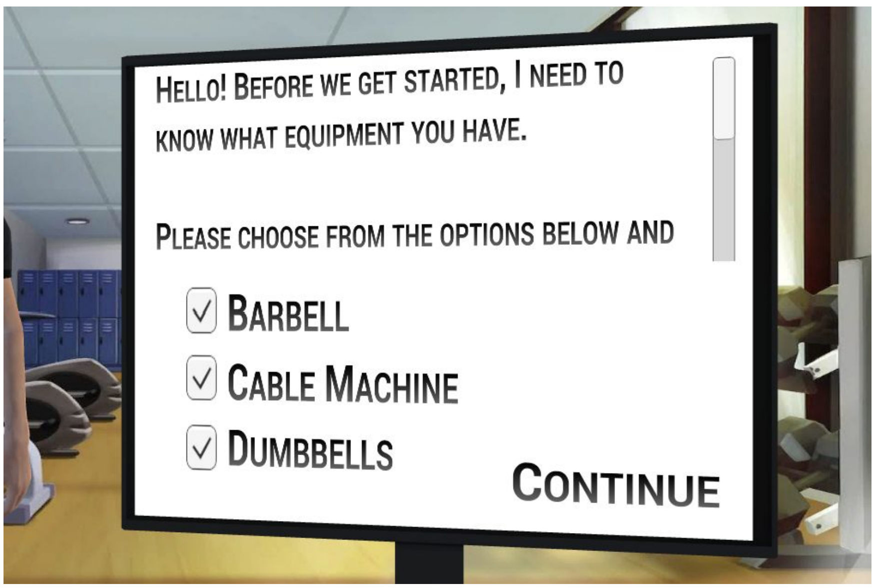
Figure 14: Instructions.



TV

Following the sprites mentioned above, a prefab on the Unity Asset Store called “Flat screen TV” by Rutger Klunder (Klunder, 2013). This was imported into the project and a white image was added for the background. Panels were added on top of this to represent the different sections the user would see, and these panels contained interactive elements such as input fields, toggles and buttons, as shown in *figure 15*. Directional lighting was then added to give a more realistic impression.

Figure 15: Flat screen TV with interactive elements.

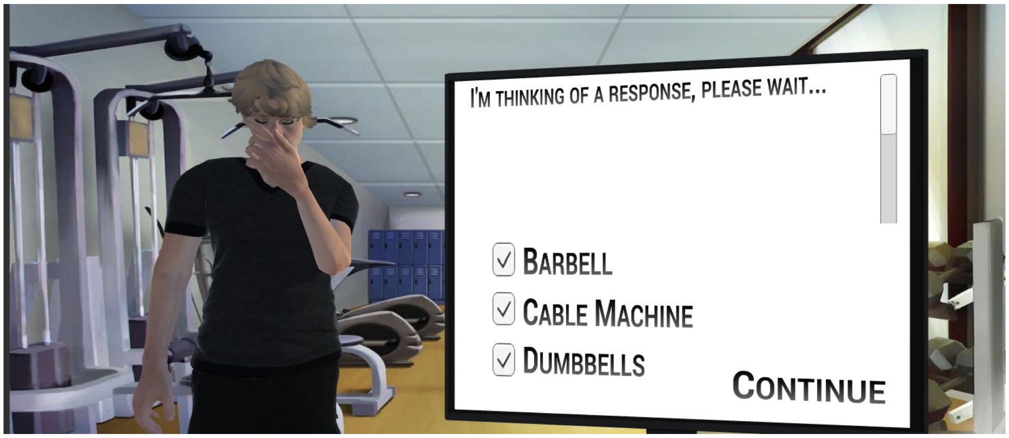


Animations

Several animations were created for the JimBot character. One was premade in Mixamo called “Breathing Idle”, which makes JimBot more realistic as he will sway and breathe in and out (Mixamo, 2019.).

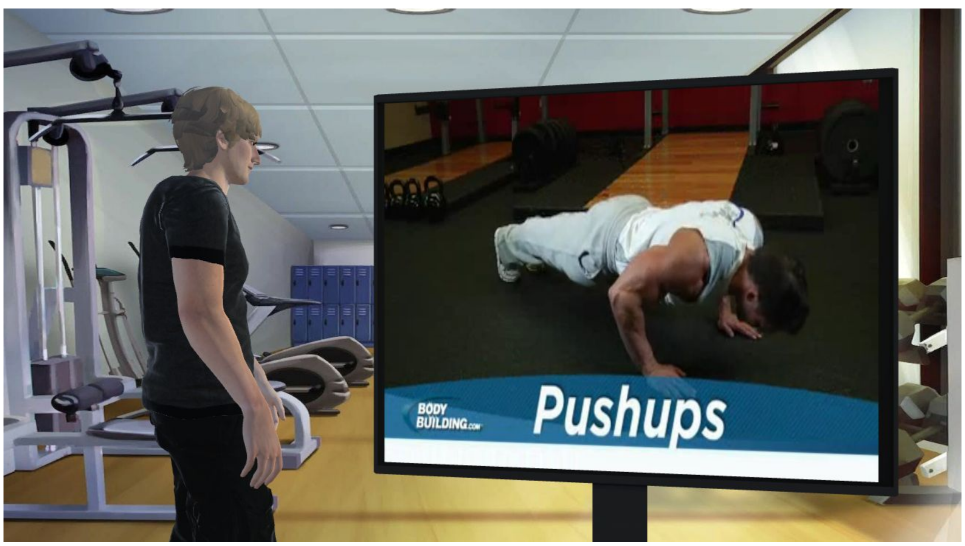
Other than this animation, more were created in Unity for specific actions. A prime example of this is for when the user sends an input to the chatbot, and waiting for a response from the DialogFlow API, an animation was made to simulate JimBot “thinking”, as shown in *figure 16*. The length of the animation primarily depends on internet connection speeds, and the length will be longer if the user sends a voice input to JimBot, as it takes longer to process this.

Figure 16: Thinking animation screenshot.



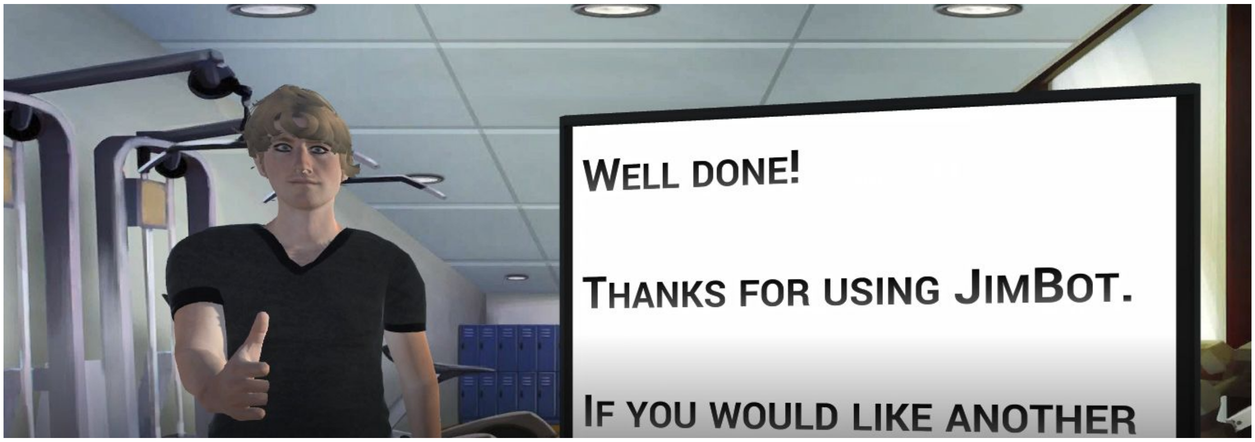
Another animation created was for when the exercise videos are playing, and JimBot will turn and point towards the TV to indicate to the user that they should observe the TV screen, see *figure 17.* When the exercise video finished, another animation will trigger for JimBot to return to facing the user.

Figure 17: Facing towards TV Screen.



The final animation created was for giving the user a “thumbs up” and is used for either when the user resumes a paused workout, when they are halfway through the workout, or at the end of the workout, as shown in *figure 18*.

Figure 18: Thumbs up animation screenshot.



Audio

Regarding audio clips, the only audio files for this project relate to the chatbot, so there is no background music. There are premade .wav files used for this project, and these include the following:

> "Default error response", "Halfway time", "Intro clip", "Next exercise", "Resume the workout", "Specific panel help", "Starting screen", "Start the workout", "The last exercise" and "Workout completed".

The other audio clip involves the response from the DialogFlow API, and a .wav file is updated every time a response is received. Lip-syncing was achieved through using an Asset called Salsa by Crazy Minnow Studio (Crazy Minnow Studio, 2014.).

C# Scripts (Classes)

Due to the complexity of the project, it was logical to begin by creating separate classes for each scene. A class was first created for the StartingScene, JimBot and Instructions scenes. The classes were updated based on the interactive elements on the screen, i.e. button clicks. As the classes became larger, it was imperative to separate logic from these classes into their own separate classes, to make the code more manageable and maintainable for bug identification purposes. *Figure 19* shows the hierarchy of scripts created for the project, and the source code can be found at:

> https://github.com/lewisbenking/PT-Bot

It is important to state that components of some scripts were influenced from online sources.

The two JsonData classes were completed with assistance from the Google documentation (Google, 2019).

Handling the response from the DialogFlow API, and capturing microphone inputs were inspired by a project from Zhang (Zhang, et al., 2019).

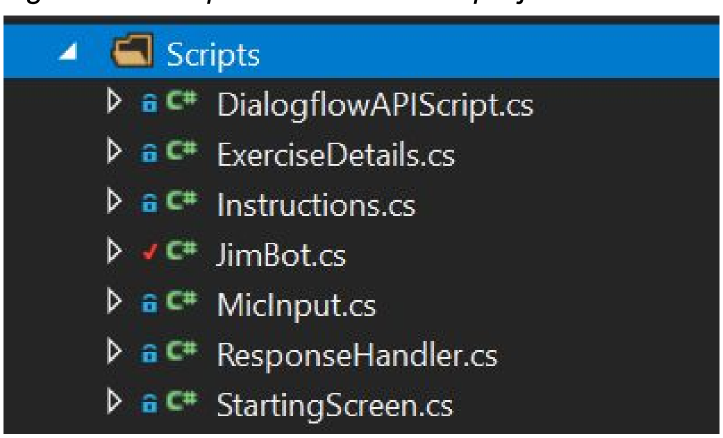
A tutorial by Alexander Zotov was helpful for creating a scrollable text area for displaying the long exercise descriptions (Zotov, 2017).

The Unity Forums provided information regarding playing videos at runtime (Unity, 2016), and a blog post from Gyanendu Shekhar was beneficial in loading images at runtime; in this case the muscles affected diagrams (Shekhar, 2017).

A post on the Unity forums was useful for pausing coroutines (Unity, 2012), and this was applied to the workout timer, enabling the user to pause the workout and return to the workout list.

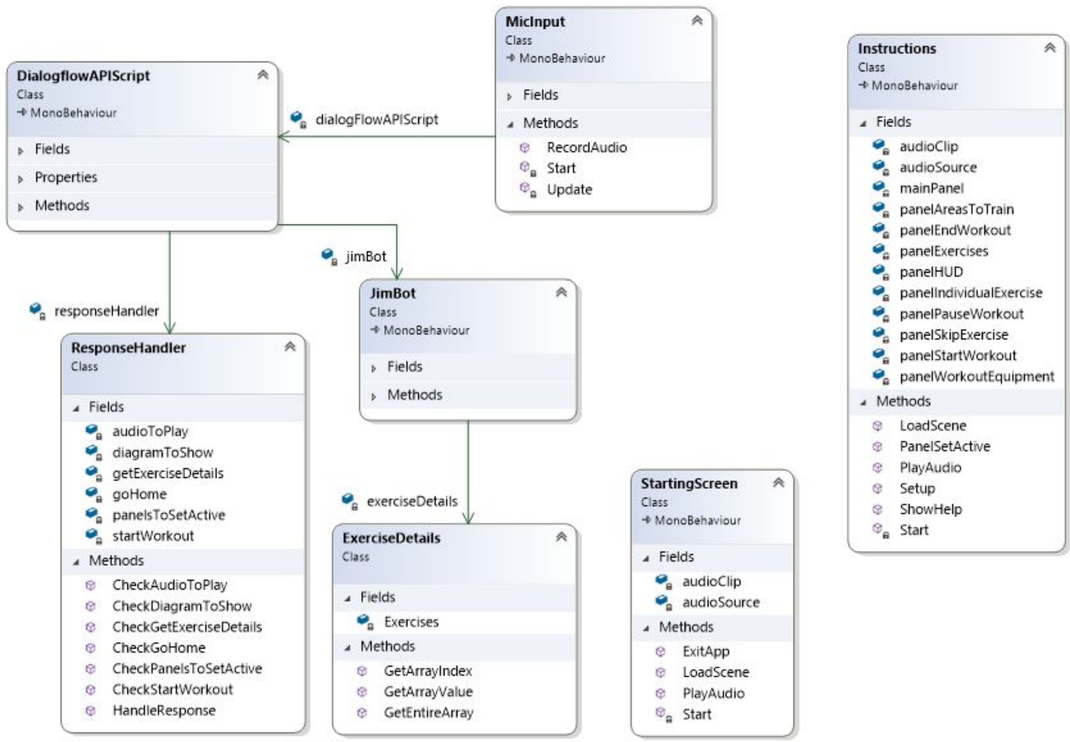
Some scripts were imported and have been referenced and acknowledged accordingly, for example the scripts from Crazy Minnow Studio for lip-syncing could not be modified in case any further issues were caused.

Figure 19: Scripts created for the project.



A class diagram was also created for the different classes. Due to the amount of fields and method defined in the classes, it was not possible to take a screenshot of the entire class diagram, as seen in *figure 20*. This can be found in the GitHub repository.

Figure 20: Class diagram.



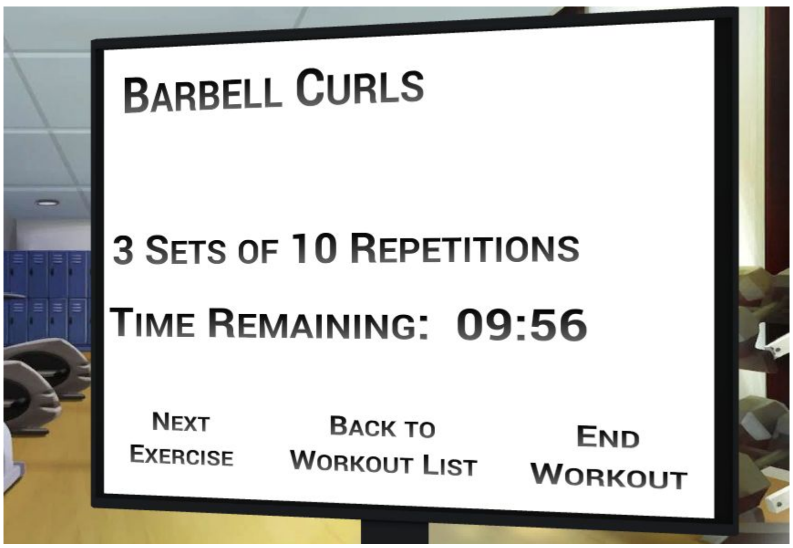
Text (Colour and Fonts)

Colour gradient of black to grey was used for the text on the UI. This was to make the UI more aesthetically pleasing rather than simply using black text on a light background. The font family and style was kept consistent across the UI to provide a sense of familiarity to the user when they view different sections of JimBot. This was applied through using an Asset from the Asset Store named TextMeshPro (Unity Technologies, 2017.).

The font used universally throughout the project is Roboto (Bold) and font sizes vary between body content and headers, and the text is presented in capital letters, making it easier for the user to read, and font sizes are appropriate for people with varying levels of eyesight. If the user struggles to clearly read the text, listening to the help audio may prove beneficial for them.

Figure 21 shows an example of the text used in the application. Related content is grouped together making the layout clear to the user, an example of this is shown with the buttons in *figure 21* grouped together at the bottom of the TV screen, and adequate space is given for the exercise name at the top of the screen, with more information displayed between the exercise name and buttons.

Figure 21: Text font, colour and size example.



Solving Specific Problems During Development

There were various issues arising throughout the development of JimBot and it was vital the major issues were resolved as quickly as possible.

An issue when first using the DialogFlow console to create the chatbot was that intents were not being recognised properly, particularly with synonyms. To resolve this issue, the intents were deleted and separate intents for the different body areas were created and the chatbot flow was altered. If this issue wasn't fixed, users may have received unexpected responses from the chatbot and this would negatively affect their user experience.

An issue identified in the later stages of the project was the JimBot character model in Unity was suffering from unusual deformations, and this was identified when background lighting was added to the scene. This issue was reduced through correctly scaling the character model, however it still suffered from these deformations. There may have been a mesh deforming issue with Adobe integrating the Fuse character with Mixamo and its autorigging feature. Therefore a decision was made to create a new character from scratch, and the same process was followed of importing the model from Fuse into Mixamo then applying the animation and importing into the Unity project. Fortunately this fixed the issue, and now JimBot looks more realistic than the previous character model used.

In the early stages, when first connecting the chatbot API from the Unity project and interpreting the response, a noise similar to white noise/static was playing. To fix this, the code was modified to write the audio output to a .wav file, load the file into an AudioClip and then play the AudioClip (DeadlyFingers, 2017.). This was vital to fix, otherwise JimBot would not be able to play audio responses from the chatbot API, and this would heavily affect the lip-syncing functionality.

Unfortunately not all issues could be fixed. One of these examples involved a random unpredictable delay in receiving a response from the DialogFlow API POST request. This was identified as being an issue with internet connection speeds, so unfortunately cannot be resolved. The “thinking” animation was created as to give the user the impression that JimBot is thinking of a response, but in actual fact it is waiting for a response from the DialogFlow API.

As required for the CS3270 module, it was required to use Behaviour Driven Development. This was extremely difficult to set up within Unity with scarce relevant documentation available. Details on this testing strategy will be provided under *Testing*.

Testing

Behaviour Driven Development (BDD)

BDD was carried out as a requirement for the *CS3270 Testing and Reliability in Software Engineering* module. BDD is an evolution of the more widely known Test Driven Development (TDD), however this methodology uses scenarios and the following template was used for this project:

Given [Original state]
When [Action to be tested]
Then [Expected output]

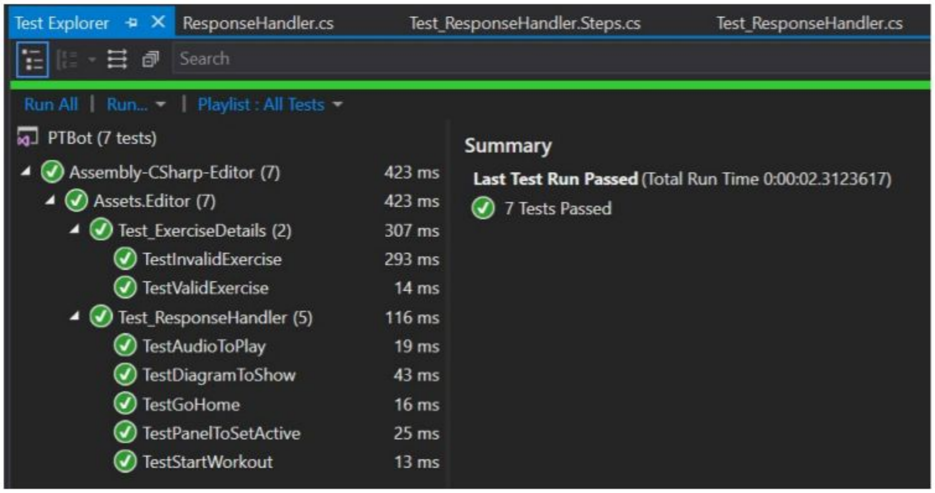
Through scenarios, the unit tests (viewed in the GitHub repository) were created for specific classes before any code was written. This is because the tests are supposed to fail at first, and then the code to make the tests pass will be written, otherwise there is no point to following this testing methodology. This could not be applied to every class for this project, because BDD is not currently supported with Unity objects, meaning any C# classes that derive from MonoBehaviour could not be tested through BDD. Despite this negative, a positive was that logic could be encapsulated into their own classes, meaning the code would eventually become more manageable and maintainable, with bugs being easier to locate. Another benefit of using BDD to create these classes was that it could serve to prove requirements defined above would be achieved.

"*Different types of testing, combined when necessary, were conducted on 2 classes. The test cases were derived from Scenario Testing, and updated following Exploratory Testing when notes were created from experimenting with various input values for method parameters. Black-box testing was demonstrated through the use of decision tables, and was utilised with condition coverage, a technique for White-box Testing, and they were used for their suitability for these tests when compared with other black-box and white-box testing methods. This testing strategy was beneficial to use, because through using and combining these types of tests, full coverage of both ExerciseDetails and ResponseHandler were achieved, ensuring all possible paths were testable and tested, with each test passing successfully, as shown from figure 17 and Appendix 7. Based on these results, it would be fair to say this test strategy was effective to use. Although, all projects are different, and the applied Q3 testing only covers a portion of the entire codebase, but it is a significant component.*" (King, 2019).

Appendix X shows relevant portions of the report submitted for the *CS3270* module (King, 2019). The entire report is available at:
https://github.com/lewisbenking/PT-Bot

The results for the tests can be seen below in *figure 22*, or viewed at:
https://www.youtube.com/watch?v=BTxQP4J89wc

Figure 22: Test explorer results



User Acceptance Testing (UAT)

UAT was carried out throughout the project and more details on this will be provided under the Systematic Feedback heading below.

Evaluation

Systematic Feedback

Obtaining Feedback

Throughout the duration of the project, it was important to obtain feedback from other people from different abilities, ages and backgrounds. This is because people are likely to interact with the product differently based on their ability, age and background, and it was important for them to see the product evolve throughout the weeks and months of work on the project, so they could see progress being made.

In order to obtain quantifiable feedback from the users, an anonymous questionnaire was created on Google Forms and provided to them twice throughout development. By doing this, issues with the product could be identified and hopefully rectified by the time the next questionnaire was handed out. By doing this, it helped to shape the product in a way that would meet user requirements.

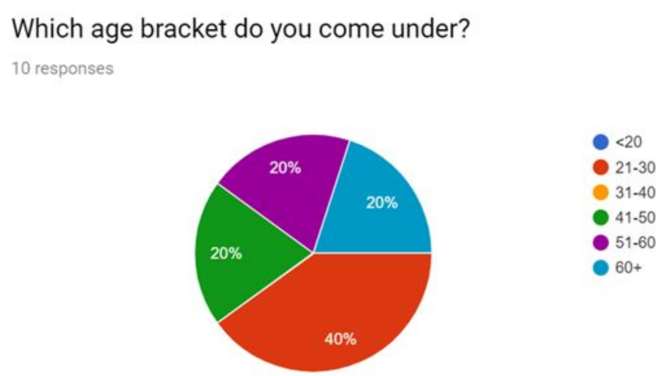
The questionnaire that was provided for the 10 users can be found as *appendix 1*. An advantage to using Google Forms for the questionnaire was the automatically generated charts with the results, and these will be displayed and discussed below.

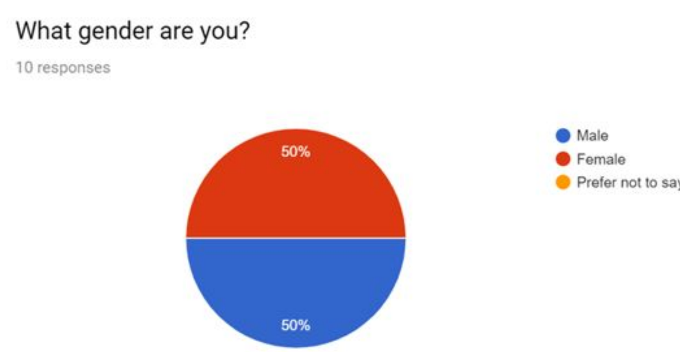
Questionnaire Results Analysis

Before analysing the results, it is important to acknowledge that the questionnaire was taken by 10 people, meaning the User Acceptance Testing stage of the project was conducted on a relatively small scale. It is also important to note, that they were able to test JimBot in their own time, under their own conditions.

In order to make a diverse test group, there was a 50/50 split of males and females, ranging from different backgrounds and age categories, with 4 users between 21-30, 2 users between 41-50, 2 users between 51-60 and the final 2 users being 60+, as shown in *figure 23*.

Figure 23: Question 1 and 2.



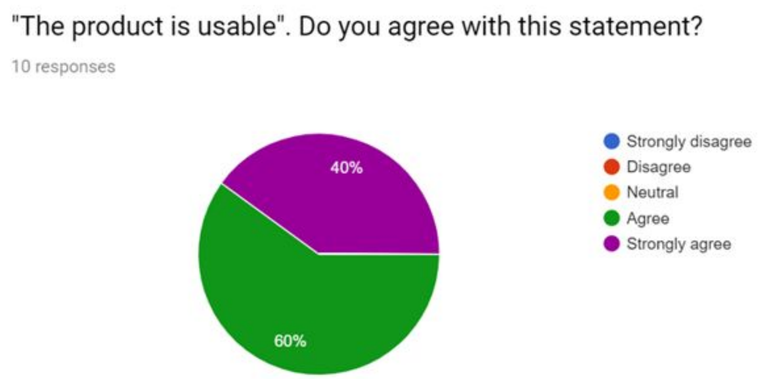


This is to demonstrate the suitability of JimBot for a variety of people. As shown from the questionnaire results that will now be discussed, the test group gave a variety of responses to the questions.

Usability

Regarding usability, as seen in *figure 24*, the responses to question 3 demonstrated that 60% of respondents agreed the product was usable. To back the opinion the product was usable, the other 40% responded with ‘Strongly Agree’. For this question, none of the users responded with ‘Disagree’ or ‘Strongly Disagree’, giving the indication the product was usable for the respondents.

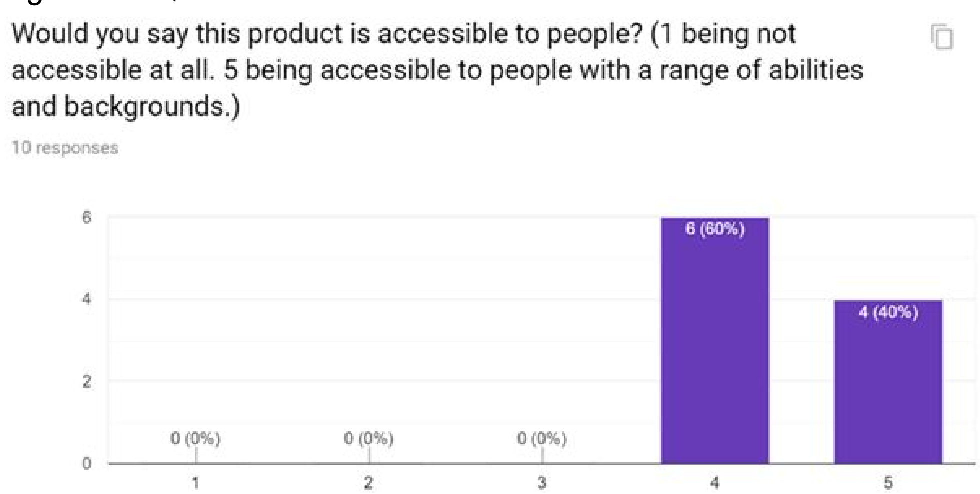
Figure 24: Question 3.



Accessibility

This question was designed to see how accessible users thought JimBot could be. *Figure 25* shows a score of 5/5 was given by 40% of respondents, and 60% gave a score of 4/5 for accessibility to users of different abilities/backgrounds, and 4/5 was the lowest score. These results suggest that, due to the different ways (speech or text), that users can interact with JimBot, this software could be accessible to people that are hard of hearing or have visual impairments.

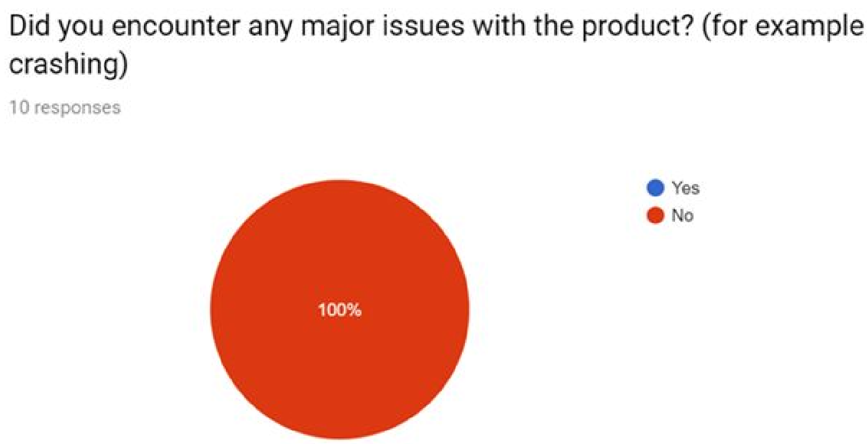
Figure 25: Question 5.



Major Issues

Through asking this question, potential major issues such as app crashes (meaning reliability) could be identified. Fortunately, *figure 26* shows, major issues did not occur for these 10 respondents, giving the indication that under these conditions JimBot is a reliable piece of software.

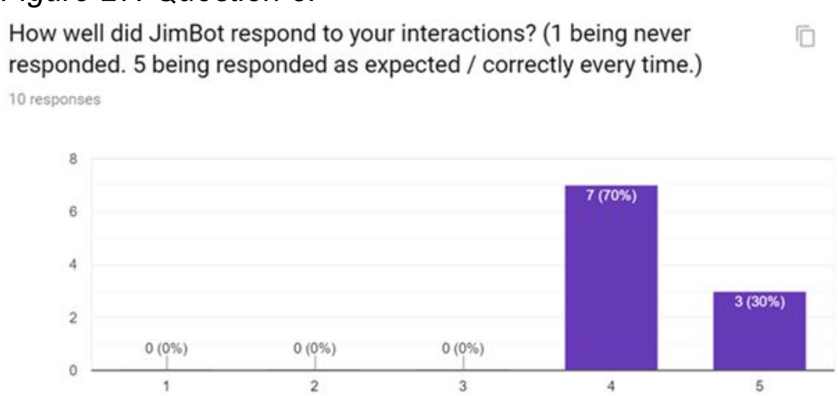
Figure 26: Question 6.



Responsiveness to Text Input

It was important to distinguish text and speech input into separate questions, because it was more likely that people would prefer to use one over the other and that speech input was less likely to be understood by the chatbot compared to text input. This is because the chatbot has to interpret the speech input from the dialect/accent of the user, which is more difficult than plaintext input. From the question regarding text input, as seen in *figure 27*, 70% of users gave JimBot a score of 4/5 for responding to their interactions, whether that being through using interactive elements on the TV screen, or by sending a text message on the HUD to JimBot. This was the lowest score, as 30% of users gave a score of 5/5, suggesting that JimBot is responsive to text in the correct way the majority of the time.

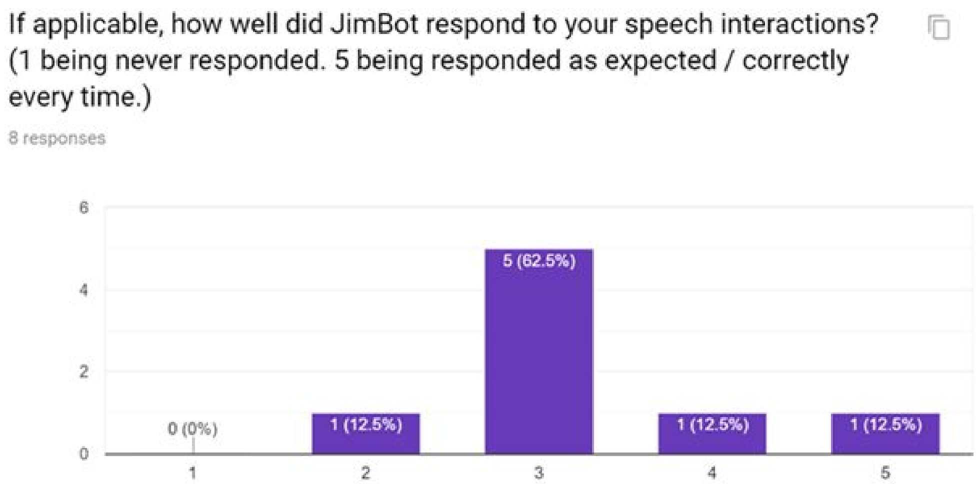
Figure 27: Question 8.



Responsiveness to Speech Input

This question was not applicable to every user. This is because before using JimBot, a user mentioned they had a tracheotomy so would have been difficult for them to use speech interaction (this person couldn't be identifiable from the results because another person decided not to use speech interaction). *Figure 28* shows the varied responses from the 8 users that used speech to interact with JimBot. The mean score was 3/5, with 5 users giving this score, the other scores included 2/5, 4/5 and 5/5. These results were predictable and expected, because it was mentioned to the users before testing that it may be difficult for JimBot to understand their dialect/accent at first, and they need to speak clearly. However with continued use, JimBot is more likely to understand them. These results suggest that regarding responsiveness to speech input, it will depend on how the person speaks to JimBot to determine the effectiveness in responding correctly, and with continued use the effectiveness is likely to improve, as the chatbot improves in interpreting the input.

Figure 28: Question 9.



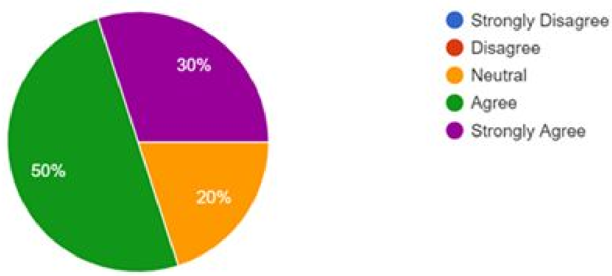
Does JimBot Solve the Problem?

For this question, users were informed about the problem the project is trying to solve, and as shown from *figure 29*, 50% of users agreed JimBot would or could solve the aforementioned problem. Responses also included strongly agreed and also 20% of users were unsure if JimBot would or could solve the problem. Therefore, none of the respondents stated JimBot wouldn't or couldn't solve the problem.

Figure 29: Question 10.

The product was designed as a potential solution to the following problem: "The demand for personal trainers has increased, and sessions are costly to the consumer. JimBot aims to fill the demand free of charge." Would you say based on the statement above, JimBot solves the problem, or would solve with some modifications?

10 responses



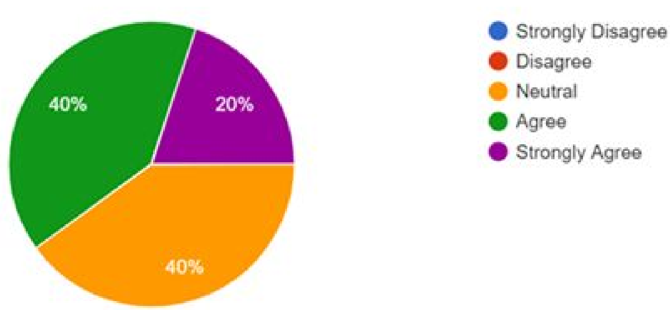
Could People Eventually Use JimBot?

This question was designed to see if users could envisage people eventually using an iteration of JimBot either in their own homes or at the gym. *Figure 30* shows that 40% of users were unsure that people would actually or eventually use JimBot, and 60% of users either stated “agree” or “strongly agree” to this question. These varied responses, although none negative, were understandable, given the future is difficult to predict!

Figure 30: Question 11.

Do you agree with the following statement: "I could envisage people using either the current version of JimBot, or a future iteration, either at home or at the gym."

10 responses



Describing Identifiable Patterns

From looking at the raw questionnaire response data, as shown in *figure 31*, it was interesting to see how people in an age group would share similar opinions to others in their group, and other age groups may occasionally have different opinions. This was indicated in question 5, as the lowest score was 4/5 across the 10 users. Although the different age groups did not always share the same opinion and example of this can be seen for question 11, with more people over the age of 41 being inclined to give a 'neutral' response whereas people under 30 responded with either 'agree' or 'strongly agree'.

Figure 31: Raw questionnaire response data.

Q1	Q2	Q3	Q4	Q5	Q5	Q7	Q8	Q9	Q10	Q11
21-30	Male	Strongly agree	-	5	No	-	5	4	Strongly Agree	Strongly Agree
21-30	Female	Strongly agree	-	5	No	-	5	5	Strongly Agree	Strongly Agree
21-30	Male	Strongly agree	-	4	No	-	4	3	Strongly Agree	Agree
21-30	Female	Agree	-	4	No	-	4	3	Agree	Agree
41-50	Female	Agree	-	4	No	-	4	3	Agree	Agree
41-50	Male	Agree	-	4	No	-	4	2	Neutral	Neutral
51-60	Male	Agree	-	4	No	-	4		Agree	Neutral
51-60	Female	Agree	-	4	No	-	4	3	Neutral	Neutral
60+	Male	Strongly agree	-	5	No	-	4		Agree	Neutral
60+	Female	Agree	-	5	No	-	5	3	Agree	Agree

Reflection

Unfeasible Requirements

Upon reflection, when creating the requirements, time constraints were not carefully considered. This oversight led to unrealistic requirements being created at the beginning of the project.

Given the timeframe the unfeasible requirements, which can be seen in *table 3*, regarding MoSCoW analysis included JimBot animating to perform the exercises himself, user profiles and nutritional advice. Fortunately, they were all deemed as *could have* requirements meaning not including them wouldn't be detrimental to the success of the project. Based on evidence above from the systematic feedback, it would be fair assessment that the *must* and *should have* requirements have been met, meaning the project could be considered a success.

User Response

It was beneficial to obtain user feedback from UAT, to prove the product could meet the defined requirements stated previously in this document, and these users could reflect potential users in the future. Based on the above *Systematic Feedback*, the responses received from the 10 users were generally positive. It was interesting to see the different identifiable patterns from the raw data shown above. Reflecting on UAT, despite the positive responses it was clear there would be room for improvement in the future, and this will be identified below, and the product limitations will be acknowledged next.

Product Limitations

It would be fair to say overall, the systematic feedback obtained from the questionnaire was largely positive and left room for improvement, however there are naturally limitations to JimBot.

Internet Connectivity

As the chatbot is online-based using cloud services, and therefore not built in to the application itself, it requires an internet connection. Throughout personal testing, on WiFi connection at home there has been delays of up to 30 seconds in posting requests to the API endpoint. However, in comparison, when testing at the university using the Aston Connect WiFi there is an almost instantaneous response. Therefore, internet connection is a limiting factor in the efficiency of the chatbot.

Platform Compatibility

The current iteration of JimBot is currently only designed for a laptop, and as most users are extremely unlikely to want to take their laptop to the gym with them, it may not be suitable for them, meaning a future iteration will certainly need to be compatible with mobile devices.

DialogFlow

The chatbot is limited in that Google only allow for so many intents and entities to be created, meaning the chatbot may not be able to specifically answer every question from the user, whereas a human personal trainer is more likely to know the specific answers.

Microphone Quality

The DialogFlow chatbot can unfortunately struggle in recognising the correct user intent from the input audio and this was obvious when collecting the systematic feedback. This may depend on the quality of the microphone being used by the user. An example of this occurred when a user spoke “Give me a legs workout” to the chatbot, and the chatbot interpret an incorrect input as “Give me a like out”. Due to this, the chatbot extracted the incorrect “like” intent as responded with “I like you too.”

Human Personal Trainers

In spite of the advantages to this product, some people may always prefer face-to-face interaction with a human personal trainer instead of the virtual human. In addition to this, it would be reasonable to make the assumption that human personal trainers are highly likely to have specialist knowledge that I am lacking in, therefore the chatbot may not cover every precise query the user has regarding fitness and nutrition that a personal trainer could answer.

Unfortunately, unlike when compared to human personal trainers, JimBot cannot actually see the user, meaning there is no way for JimBot to understand anything about the user. Personal trainers are likely to look at their client and guide them through the exercise. Since JimBot doesn’t use cameras or other sensors to observe the user and track posture, there is no way for JimBot to guide the user through the exercises and correct their form or movement.

Knowledge Gained

The main knowledge gained from this project revolves around chatbots, and their creation. This also includes their architecture and generic requirements for creating a chatbot.

Behaviour Driven Development was a new approach adopted for creating specific classes in the Unity project, as also required for *CS3270 Testing and Reliability in Software Engineering*. New understanding was gained regarding connecting to the external DialogFlow API via requests, and the general connection to an external API and handling their responses could be used for future projects.

In addition to this, knowledge about animating the JimBot character in Unity was gained, through using different rotations and translations for the character as shown above.

In terms of testing, knowledge about conducting User Acceptance Testing was gained as well as being able to interpret the responses given, and identifying any patterns or trends in the responses.

The new knowledge gained from this project could certainly be applied to future projects, or in the workplace.

Discussing Future Ideas

Due to time constraints, it was unfortunately not possible to implement every initial idea from the requirements specification. In addition to this, throughout the project it became apparent that several new features could be implemented into JimBot in the future.

An important feature for a future iteration of JimBot would be to make JimBot cross-platform compatible, meaning that JimBot could function on mobile devices and tablets as well as laptops.

To make JimBot more realistic regarding lip-syncing, this would require lip-syncing individual phonemes and visemes, which unfortunately does not occur in the current iteration. However, admittedly, more knowledge around this field is a necessity before any development could occur.

Given more time, it would be beneficial for JimBot to provide a variation of exercises based on the user journey rather than the current single workout list for each path. To elaborate on this, JimBot could randomly choose from a set of 3 workout lists for each path. This would be feasible but due to time constraints wasn’t possible in the current iteration.

JimBot could also be expanded to provide exercise classes such as “spinning” classes or “fat burning” classes, and could also be modified to animate the exercises himself in addition to having current exercise videos as the current iteration does. Although this would require extensive research and planning, and would be extremely time consuming.

Conclusion

JimBot was created as a personal trainer chatbot connected to an animated virtual human character to provide a free service containing suitable workouts to end users based on their equipment and body area preference.

Reflecting on the work, it was clear the majority of systematic feedback obtained was positive, and that the idea of JimBot is a useful one. Despite this, there are naturally limitations to JimBot, either due to time constraints or the fact JimBot cannot actually see the end user whereas a human personal trainer can, so they can guide the client through their movement or form.

It was never intended for JimBot to replicate the exact wealth of specialised knowledge that personal trainers have within the current timeframe and with existing resources. More development would be required on a future iteration to be anywhere near to achieving this goal, and with the virtual human component JimBot provides an advantage over the existing text-based solutions. There certainly is potential to expand and improve to make JimBot more realistic regarding animations and lip-syncing phonemes and visemes, and also animate the exercises himself to complement the exercise videos.

This document has covered the process of creating JimBot, including the *background*, *preparation*, *deliverable* and an *evaluation*.

References

Akaeira, N.D.. *Computer Icons, Icon Design, Desktop Wallpaper, Text, Black And White PNG.* [Online]
Available at:
https://www.kisspng.com/png-computer-icons-icon-design-help-icon-3414045/
[Accessed 20 03 2019].

Bidvine, 2018. *How much does a personal trainer cost?.* [Online]
Available at: https://www.bidvine.com/personal-training/price-guide
[Accessed 01 11 2018].

BodBot, 2017. *BodBot Personal Trainer: Workout & Fitness Coach.* [Online]
Available at: https://www.bodbot.com/
[Accessed 10 10 2018].

BotList, 2016. *Gymi Personal Trainer.* [Online]
Available at: https://botlist.co/bots/gymi-personal-trainer
[Accessed 10 10 2018].

BotList, N.D.. *Daily FitBot.* [Online]
Available at: https://botlist.co/bots/daily-fitbot
[Accessed 10 10 2018].

BotList, N.D.. *Fitly.* [Online]
Available at: https://botlist.co/bots/fitly
[Accessed 10 10 2018].

BotList, N.D.. *Peter.* [Online]
Available at: https://botlist.co/bots/peter-2
[Accessed 10 10 2018].

Chatbots, 2002. *Greta.* [Online]
Available at: https://www.chatbots.org/chatbot/greta/
[Accessed 01 11 2018].

Chatbots, 2018. *Embodied Conversational Agent.* [Online]
Available at: https://www.chatbots.org/embodied_conversational_agent/
[Accessed 01 11 2018].

Crazy Minnow Studio, 2014. SALSA with RandomEyes. [Online]
Available at:
https://assetstore.unity.com/packages/tools/animation/salsa-with-randomeyes-16944
[Accessed 20 02 2019].

Couger, 2018. *If driving with Virtual Human Agent....* [Online]
Available at: https://www.youtube.com/watch?v=zmWgJ8z239o
[Accessed 01 11 2018].

DeadlyFingers, 2017. Unity WAV. [Online]
Available at: https://github.com/deadlyfingers/UnityWav
[Accessed 20 02 2019].

Duway, N.D.. *Computer Icons, Home, Ico, Angle, Symbol PNG.* [Online]
Available at:
https://www.kisspng.com/png-computer-icons-home-clip-art-black-home-icon-609973/
[Accessed 15 03 2019].

Forksy, 2018. *Automated Nutrition Coaching with AI.* [Online]
Available at: https://getforksy.com/
[Accessed 10 10 2018].

Google, 2019. *DialogFlow API.* [Online]
Available at:
https://cloud.google.com/dialogflow-enterprise/docs/reference/rest/v2-overview
[Accessed 15 02 2019].

Google, 2019. *Overview.* [Online]
Available at: https://dialogflow.com/
[Accessed 30 11 2019].

Green, A., 2016. *Our Story.* [Online]
Available at: https://blog.countit.com/our-story-2d04b4444e83
[Accessed 10 10 2018].

King, L., 2019. *Final Year Project Testing Strategy and Test Plan,* Birmingham: N/A.

Klunder, R., 2013. *Flatscreen TV.* [Online]
Available at:
https://assetstore.unity.com/packages/3d/props/electronics/flatscreen-tv-9721

[Accessed 20 02 2019].

MAACK, M. M., 2017. *Meet Nadia, the scarily 'human' chatbot who can read your emotions.* [Online]
Available at:
https://thenextweb.com/artificial-intelligence/2017/03/24/say-hello-to-nadia-the-terrifyingly-human-chatbot-with-emotional-intelligence/
[Accessed 01 11 2018].

MIT Press, 2018. *Embodied Conversational Agents.* [Online]
Available at: https://mitpress.mit.edu/books/embodied-conversational-agents
[Accessed 01 11 2018].

Mixamo, 2019. *Breathing Idle on Remy.* [Online]
Available at: https://www.mixamo.com/#/?page=1&query=breathing+idle
[Accessed 05 03 2019].

Pinterest, 2019. *N/A.* [Online]
Available at:
https://i.pinimg.com/originals/4b/f5/29/4bf5297d2589e3c9f8bb46f131c93400.jpg
[Accessed 01 02 2019].

Rehan, A., 2018. *9 Best Chatbot Development Frameworks to Build Powerful Bots.* [Online]
Available at: https://geekflare.com/chatbot-development-frameworks/
[Accessed 30 11 2018].

SammlungFotos, N.D.. *Galerie Bilder und Informationen: Gebogene Hantel Clipart.* [Online]
Available at: http://sammlungfotos.online/brandsbdwn-bent-barbell-clipart.htm
[Accessed 05 03 2019].

Shekhar, G., 2017. *Load Image at Runtime in Unity.* [Online]
Available at: http://gyanendushekhar.com/2017/07/08/load-image-runtime-unity/
[Accessed 01 03 2019].

Unity Technologies, 2017. TextMeshPro. [Online]
Available at:
https://assetstore.unity.com/packages/essentials/beta-projects/textmesh-pro-84126
[Accessed 05 03 2019].

Unity, 2012. *How to "pause" a Coroutine?.* [Online]
Available at: https://forum.unity.com/threads/how-to-pause-a-coroutine.161479/
[Accessed 01 03 2019].

Unity, 2016. *Correct Video Format for Playback?.* [Online]
Available at:
https://forum.unity.com/threads/correct-video-format-for-video-playback.446011/#post-2886983
[Accessed 01 03 2019].

Zhang, L., Sun, A. & Pecis Bonder, I., 2019. *uniFlow: Dialogflow V2 Unity Plugin Prototype.* [Online]
Available at: https://itp-xstory.github.io/uniFlow/#introduction
[Accessed 20 02 2019].

Zotov, A., 2017. *How to make scrollable UI (user interface) text area in Unity game? Quick tutorial..* [Online]
Available at: https://www.youtube.com/watch?v=ArH0S2Cdptk
[Accessed 05 03 2019].

Bibliography

BodyBuilding, N.D.. *Air Bike.* [Online]
Available at: https://www.bodybuilding.com/exercises/air-bike
[Accessed 01 02 2019].

BodyBuilding, N.D.. *Alternate Dumbbell Curls.* [Online]
Available at: https://www.bodybuilding.com/exercises/dumbbell-alternate-bicep-curl
[Accessed 01 02 2019].

BodyBuilding, N.D.. *Alternate Dumbbell Curls Incline.* [Online]
Available at: https://www.bodybuilding.com/exercises/alternate-incline-dumbbell-curl
[Accessed 01 02 2019].

BodyBuilding, N.D.. *Barbell Bench Press Medium Grip.* [Online]
Available at: https://www.bodybuilding.com/exercises/barbell-bench-press-medium-grip
[Accessed 01 02 2019].

BodyBuilding, N.D.. *Barbell Curls.* [Online]
Available at: https://www.bodybuilding.com/exercises/barbell-curl
[Accessed 01 02 2019].

BodyBuilding, N.D.. *Barbell Curls Lying on Incline.* [Online]
Available at:
https://www.bodybuilding.com/exercises/barbell-curls-lying-against-an-incline
[Accessed 01 02 2019].

BodyBuilding, N.D.. *Barbell Deadlift.* [Online]
Available at: https://www.bodybuilding.com/exercises/barbell-deadlift
[Accessed 01 02 2019].

BodyBuilding, N.D.. *Barbell Full Squat.* [Online]
Available at: https://www.bodybuilding.com/exercises/barbell-full-squat
[Accessed 01 02 2019].

BodyBuilding, N.D.. *Barbell Lunge.* [Online]
Available at: https://www.bodybuilding.com/exercises/barbell-lunge
[Accessed 01 02 2019].

BodyBuilding, N.D.. *Barbell Shrug.* [Online]
Available at: https://www.bodybuilding.com/exercises/barbell-shrug
[Accessed 01 02 2019].

BodyBuilding, N.D.. *Barbell Side Bend.* [Online]
Available at: https://www.bodybuilding.com/exercises/barbell-side-bend
[Accessed 01 02 2019].

BodyBuilding, N.D.. *Close Grip Front Lat Pulldown.* [Online]
Available at: https://www.bodybuilding.com/exercises/close-grip-front-lat-pulldown
[Accessed 01 02 2019].

BodyBuilding, N.D.. *Concentration Curls.* [Online]
Available at: https://www.bodybuilding.com/exercises/concentration-curls
[Accessed 01 02 2019].

BodyBuilding, N.D.. *Decline Dumbbell Bench Press.* [Online]
Available at: https://www.bodybuilding.com/exercises/decline-dumbbell-bench-press
[Accessed 01 02 2019].

BodyBuilding, N.D.. *Dumbbell Bench Press.* [Online]
Available at: https://www.bodybuilding.com/exercises/dumbbell-bench-press
[Accessed 01 02 2019].

BodyBuilding, N.D.. *Dumbbell Flyes.* [Online]
Available at: https://www.bodybuilding.com/exercises/dumbbell-flyes
[Accessed 01 02 2019].

BodyBuilding, N.D.. *Dumbbell Lunges.* [Online]
Available at: https://www.bodybuilding.com/exercises/dumbbell-lunges
[Accessed 01 02 2019].

BodyBuilding, N.D.. *Dumbbell Shrug.* [Online]
Available at: https://www.bodybuilding.com/exercises/dumbbell-shrug
[Accessed 01 02 2019].

BodyBuilding, N.D.. *Dumbbell Side Bend.* [Online]
Available at: https://www.bodybuilding.com/exercises/dumbbell-side-bend
[Accessed 01 02 2019].

BodyBuilding, N.D.. *Hammer Curls.* [Online]
Available at: https://www.bodybuilding.com/exercises/hammer-curls
[Accessed 01 02 2019].

BodyBuilding, N.D.. *Hanging Leg Raise.* [Online]
Available at: https://www.bodybuilding.com/exercises/hanging-leg-raise
[Accessed 01 02 2019].

BodyBuilding, N.D.. *Pullups.* [Online]
Available at: https://www.bodybuilding.com/exercises/pullups
[Accessed 01 02 2019].

BodyBuilding, N.D.. *Pushups.* [Online]
Available at: https://www.bodybuilding.com/exercises/pushups
[Accessed 01 02 2019].

BodyBuilding, N.D.. *Romanian Deadlift.* [Online]
Available at: https://www.bodybuilding.com/exercises/romanian-deadlift
[Accessed 01 02 2019].

BodyBuilding, N.D.. *Russian Twist.* [Online]
Available at: https://www.bodybuilding.com/exercises/russian-twist
[Accessed 01 02 2019].

BodyBuilding, N.D.. *Seated Barbell Twist.* [Online]
Available at: https://www.bodybuilding.com/exercises/seated-barbell-twist
[Accessed 01 02 2019].

BodyBuilding, N.D.. *Seated Cable Rows.* [Online]
Available at: https://www.bodybuilding.com/exercises/seated-cable-rows
[Accessed 01 02 2019].

BodyBuilding, N.D.. *Triceps Dips on Bench.* [Online]
Available at: https://www.bodybuilding.com/exercises/bench-dips
[Accessed 01 02 2019].

BodyBuilding, N.D.. *Triceps Pushdown.* [Online]
Available at: https://www.bodybuilding.com/exercises/triceps-pushdown
[Accessed 01 02 2019].

Appendices

Appendix 1: Questionnaire

1) Which age bracket do you come under?

<20 21-30 31-40 41-50 51-60 61+

2) What gender are you?

Male Female Prefer Not To Say Other

^The first two questions are there to identify any patterns between age groups/genders.

3) "The product is usable". Do you agree with this statement?

Strongly Disagree
Disagree
Neither
Agree
Strongly Agree

4) If you answered strongly disagree or disagree could you please explain?
(This will help to improve a future iteration).

____

5) Would you say this product is accessible to people?
(1 being not accessible at all. 5 being accessible to people with a range of abilities and backgrounds.)

1 2 3 4 5

6) Did you encounter any major issues with the product?
(For example, crashing)

Yes No

7) If you encountered any major issues, could you please provide details?
(This will help to improve a future iteration.)

____

8) How well did JimBot respond to your interactions?
(1 being never responded. 5 being responded as expected / correctly every time.)

1 2 3 4 5

9) If applicable, how well did JimBot respond to your speech interactions?
(1 being never responded. 5 being responded as expected / correctly every time.)
Please note, the more you talk to JimBot the more likely he is to understand what you are saying.

1 2 3 4 5

10) The product was designed as a potential solution to the following problem: "The demand for personal trainers has increased, and sessions are costly to the consumer. JimBot aims to fill the demand free of charge."

Would you say based on the statement above, JimBot solves the problem, or would solve with some modifications?

Strongly Disagree
Disagree
Neither
Agree
Strongly Agree

11) Do you agree with the following statement:
"I could envisage people using either the current version of JimBot, or a future iteration, either at home or at the gym."

Strongly Disagree
Disagree
Neither
Agree
Strongly Agree