3 Design Specification

Identify the requirements of the design and collate them as a design specification.
Use quantified statements when possible. Label each statement as a ‘Wish’ or a ‘Demand’.

Date: 11/04/20

Demand/Wish	Requirements	Keywords
1. D	A scenario database that can be expanded by uploading past medical diagnoses and patient conditions.	Variability, Exposure
2. D	Able to change the conversational flow by reacting to the user’s input	Adaptability, Dynamic
3. D	Able to accurately depict the nature and location of pain through facial expression and impaired motion	Accurate depiction

3 Design Specification

Identify the requirements of the design and collate them as a design specification.
Use quantified statements when possible. Label each statement as a ‘Wish’ or a ‘Demand’.

Date: 11/04/20

Demand/Wish	Requirements	Keywords
4. D	Able to realistically reconstruct 3D model of a human patient, with special attention to bone structures and joint movements.	3D modelling, Anatomy, Orthopedic parameters
5. D	Able to simulate the sense of touch to improve realism of physical examinations conducted by the Resident.	Realism, Sense of touch

3 Design Specification

Identify the requirements of the design and collate them as a design specification.
Use quantified statements when possible. Label each statement as a 'Wish' or a 'Demand'.

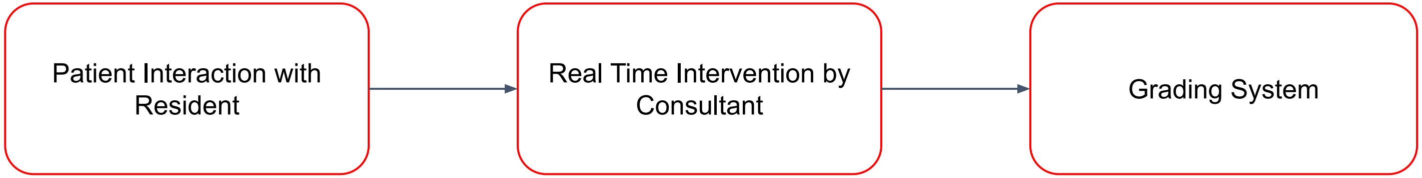
Date: 11/04/20

Demand/Wish	Requirements	Keywords
6. D	Enable the Consultant to deliver instant feedback and demonstrate correct procedures to Residents at the point of error.	Instant feedback
7. W	Able to provide key metrics of Resident's performance to both Residents and Consultant, identifying areas of improvement and strengths.	Grading system
8. W	Able to allow multiple users to collaborate while at different locations	Collaboration, Different locations

4 Function Analysis

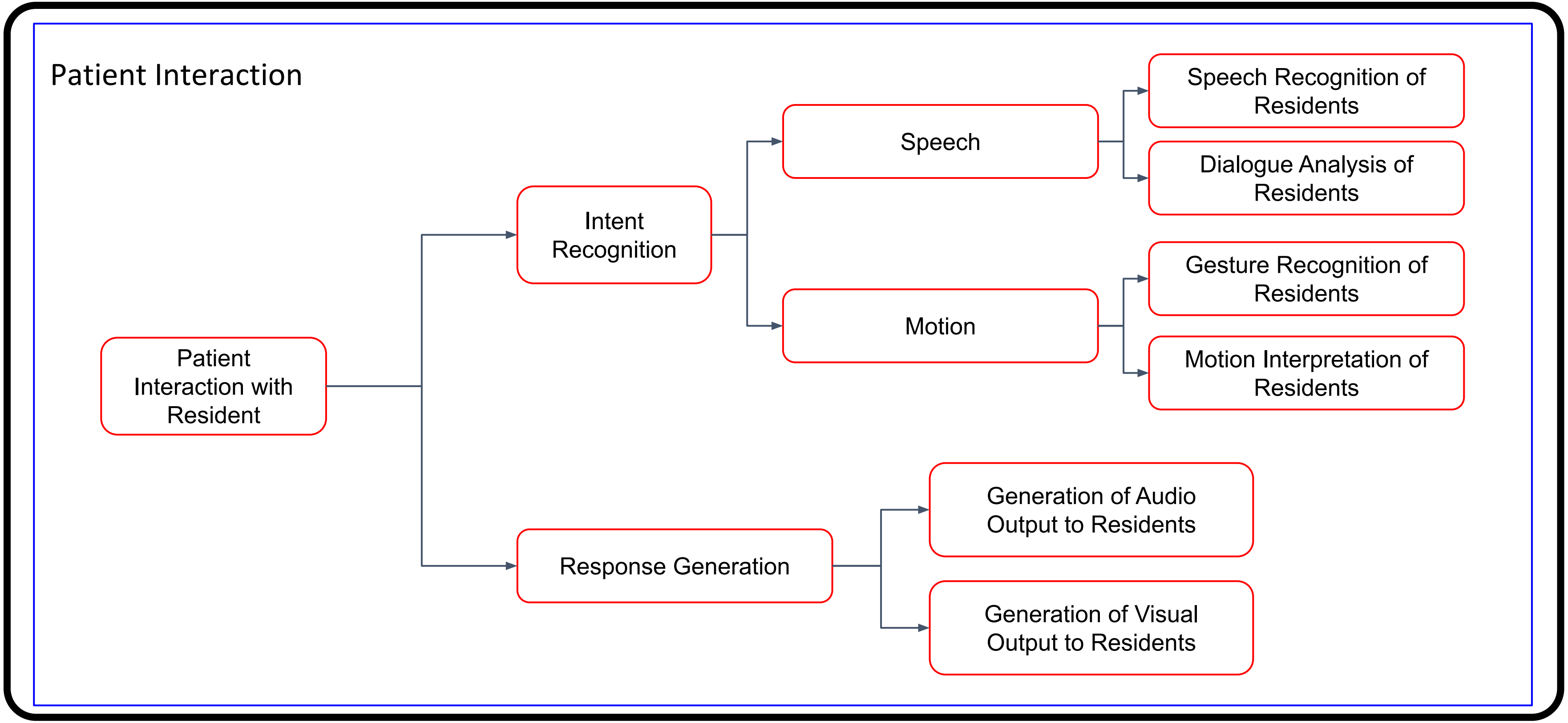
Identify the overall function and break it down into smaller functions.

We identified 3 core functions for our solution and the flow of the solutions are shown below:



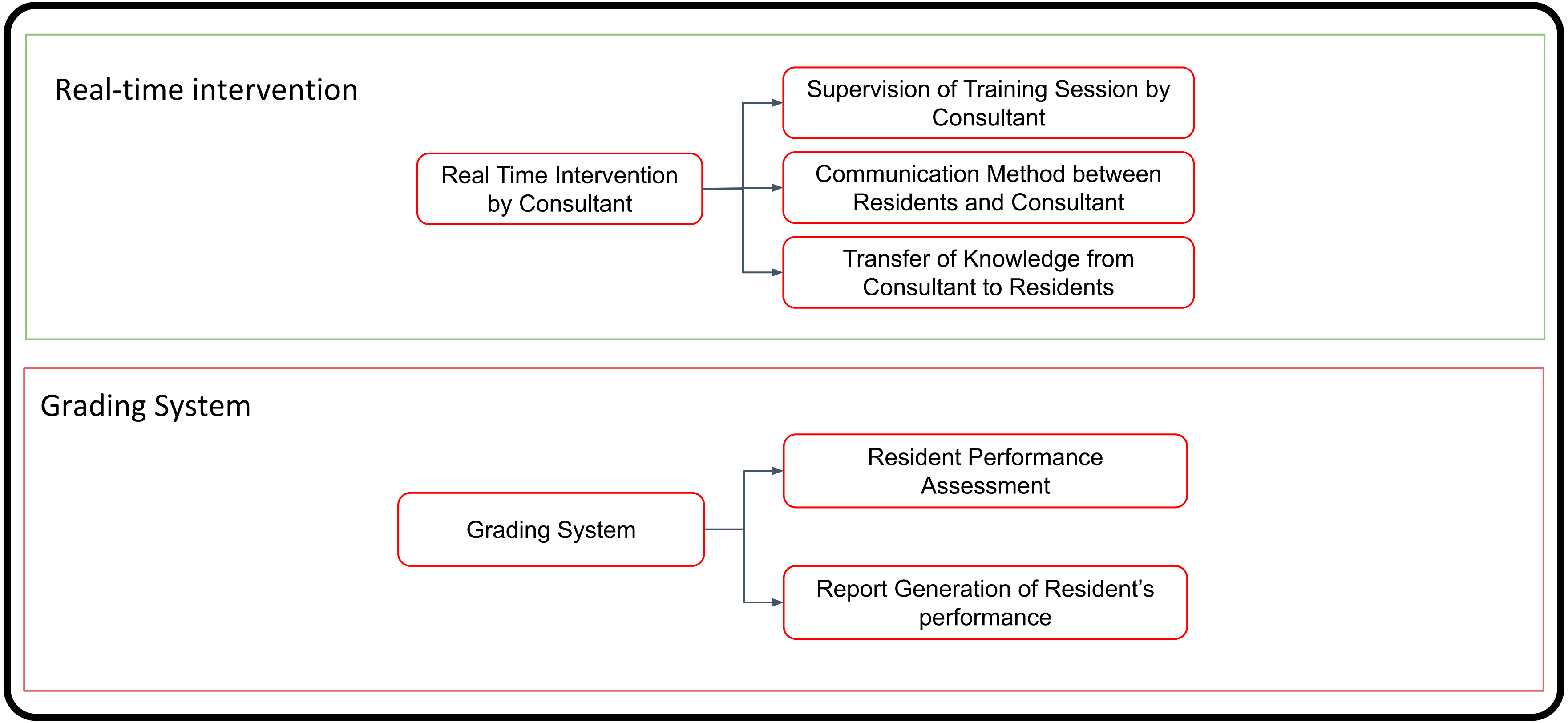
4 Function Analysis

Identify the overall function and break it down into smaller functions.



4 Function Analysis

Identify the overall function and break it down into smaller functions.



5 Concept Generation

Generate different concepts to address the functions identified.

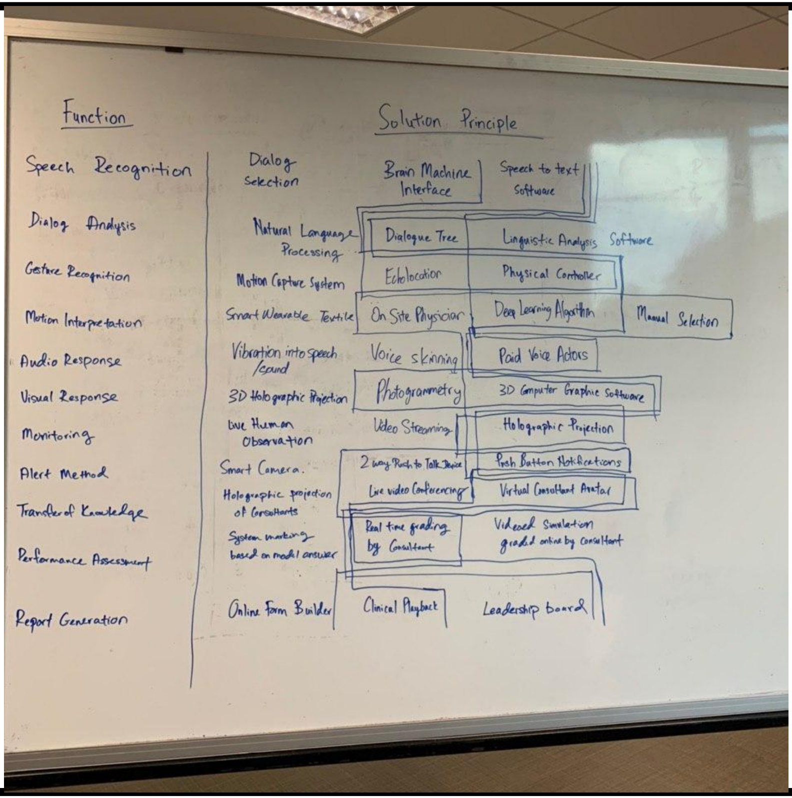


Figure 1: Combining various solution principles to come up with a solution

5 Concept Generation

Generate different concepts to address the functions identified.

Function	Solution Principles			
Speech Recognition of Residents	Dialog selection	Translation of neural signals into speech	Speech to text software	Human interpretation assisted by speech to text software
Dialogue Analysis of Residents	Natural Language Processing	Dialogue tree	Analysis of transcripts for context and keywords	Human interpretation assisted by Natural Language Processing
Gesture Recognition of Residents	Motion capture system	Emission and reflection of ultrasound signals to detect movements and body position	Physical controller input	Translation of neural signals into motion
Motion Interpretation of Residents	Smart wearable textiles with electromyography sensors to interpret muscle signals	Manual selection of motion	Deep learning algorithm to predict Resident’s intents for various motions	Interpretation by real Human

5 Concept Generation

Generate different concepts to address the functions identified.

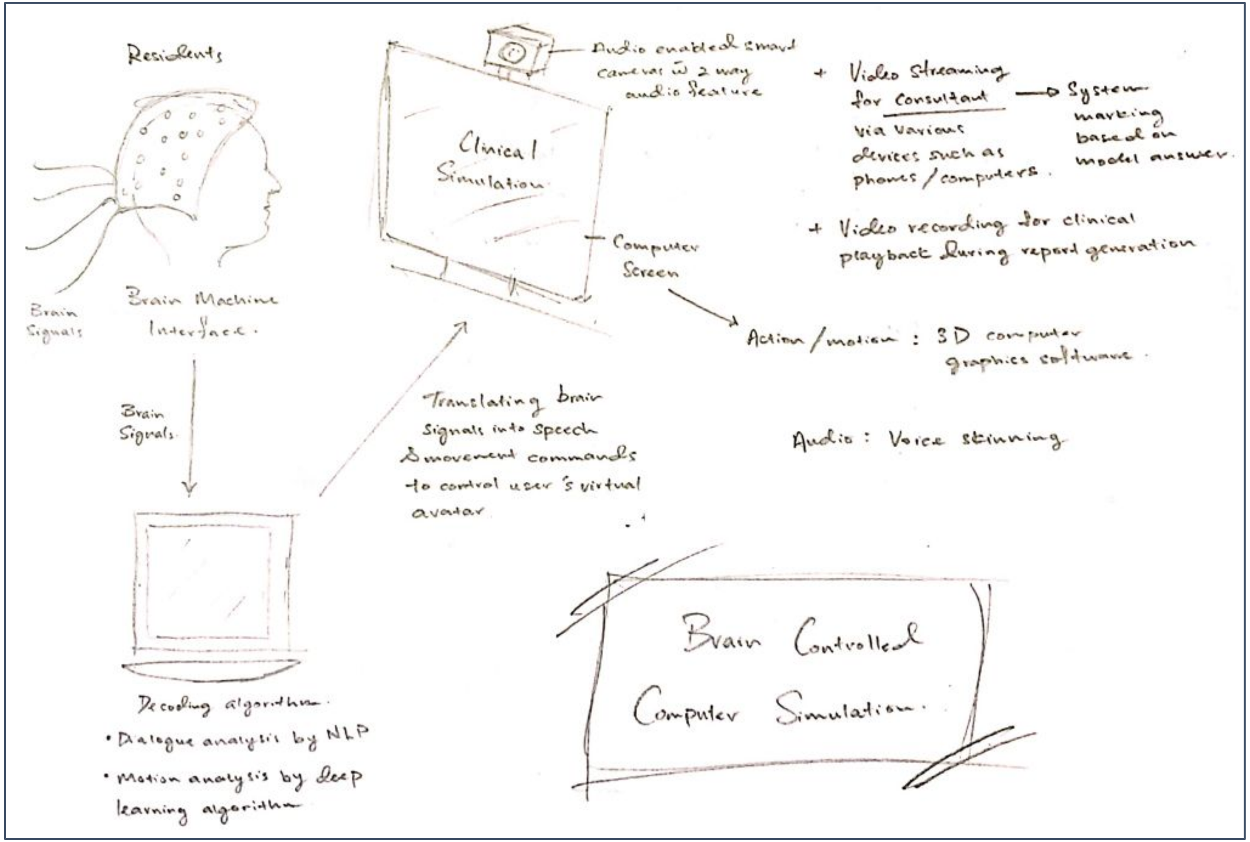
Function	Solution Principles			
Generation of Audio Output to Residents	Text to speech software	Voice skinning using generative adversarial deep neural networks (GAN) to recreate different voices and languages	Hire voice actors to pre-record audio clips that can be played back	Human voice
Generation of Visual Output to Residents	Use professional actors to pre-record video clips that can be played back	Recreating virtual 3D avatar from multiple photographic images (Photogrammetry)	3D graphic design software	
Supervision of Training Session by Consultant	Live human observation	Video streaming	Holographic projection of the clinical session	
Communication Method between Residents and Consultant	Using audio enabled smart cameras with 2-way audio feature	2-way push-to-talk device	Push-button notification system	

5 Concept Generation

Generate different concepts to address the functions identified.

Function	Solution Principles			
Transfer of Knowledge from Consultant to Residents	Holographic projection of Consultant that is capable of mimicking his actions in real time	Live video conferencing	Virtual Consultant avatar guiding the Residents	
Resident Performance Assessment	System marking based on model answers	Real-time grading by specialists	Entire simulation is videoed down and saved to portal where a pool of qualified Consultants can mark the simulation	
Report Generation of Resident's performance	Online form builders that will use template and populate variable fields	Metrics, comments and visualizations will be presented in the clinical playback to help the Resident visualize his performance	System to view individual and peer learning progression, ie leadership board	

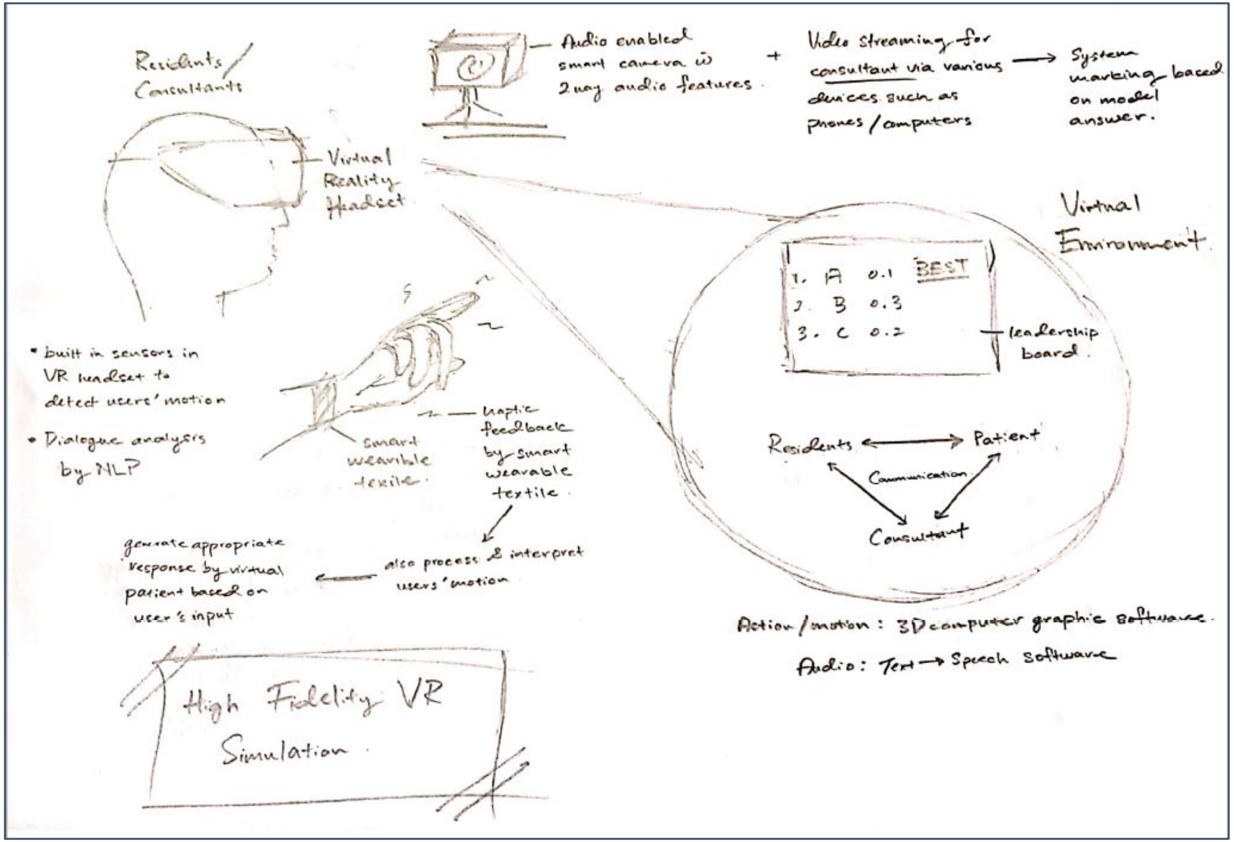
Solution A : Brain Controlled Computer Simulation



- Resident controls simulation using a brain computer interface
- Neural Signals from Resident's brain translated into movement and speech commands for Resident's virtual avatar in the simulation as they interact with the virtual patient
- Third person view of entire simulation
- Virtual patient controlled by deep learning algorithm

Figure 2: Sketch of solution A

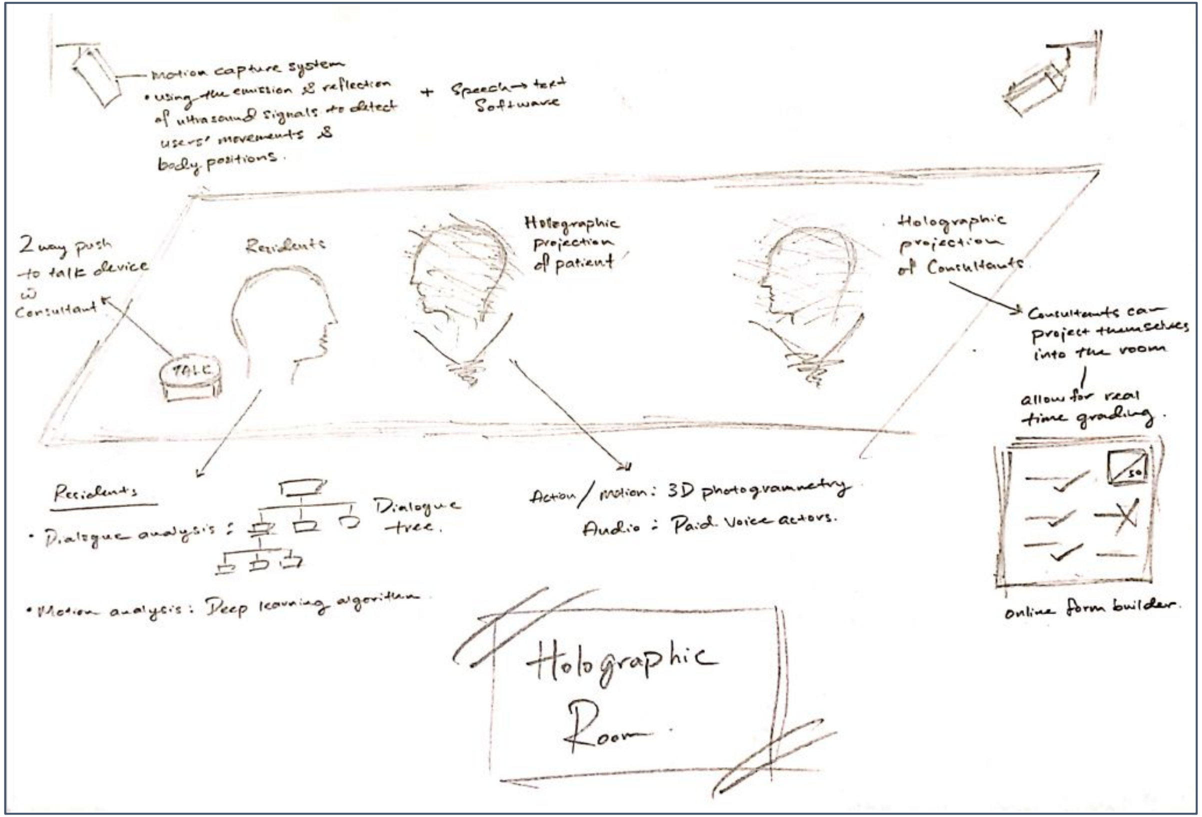
Solution B : High Fidelity Virtual Reality Simulation



- Resident enters virtual reality simulation using a VR headset
- Built in sensors in the VR headset to detect Resident's motion
- First person interaction with virtual patient using Point of View (POV) of virtual avatar
- First person view of entire simulation

Figure 3: Sketch of solution B

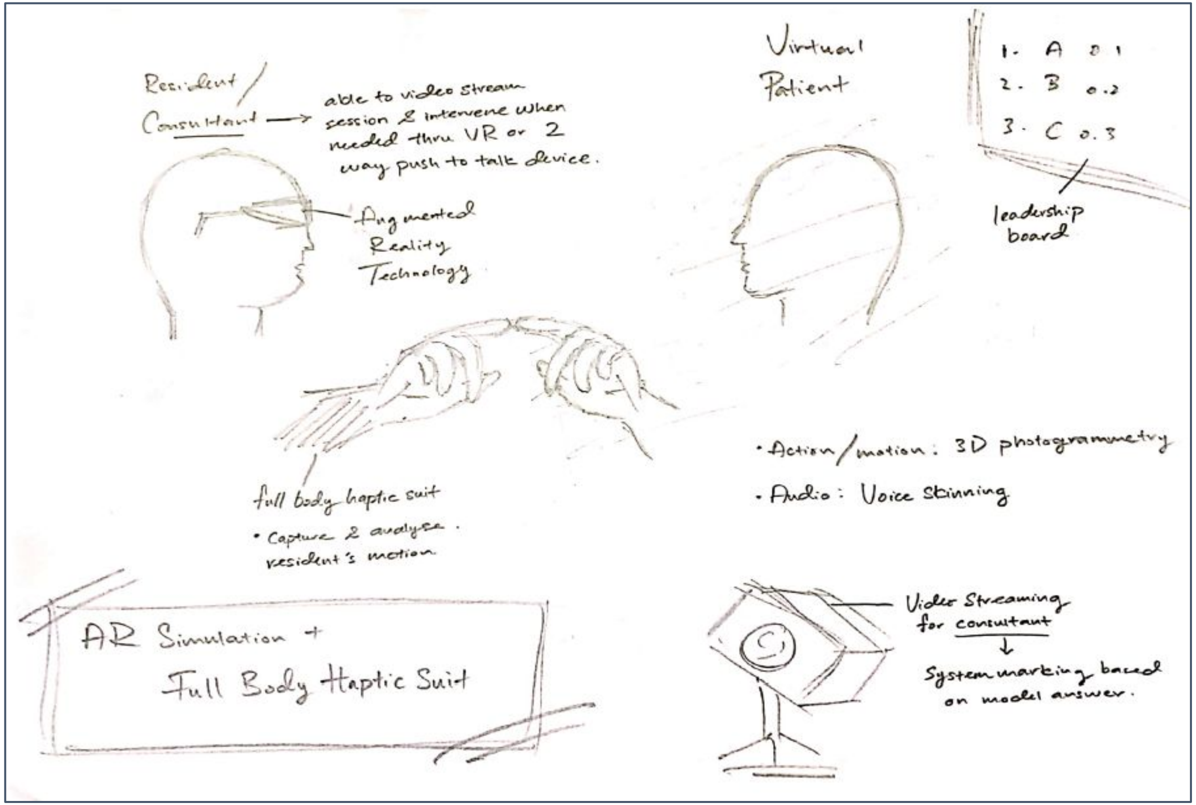
Solution C : Holographic Room



- Resident interacts directly with holographic patient in a special room set up with holographic projection equipment
- Resident's movements and speech are extracted by motion capture systems and speech to text software.
- Subsequent analysis will be done on the Resident's movements and speech to generate an appropriate response by the holographic patient

Figure 4: Sketch of solution C

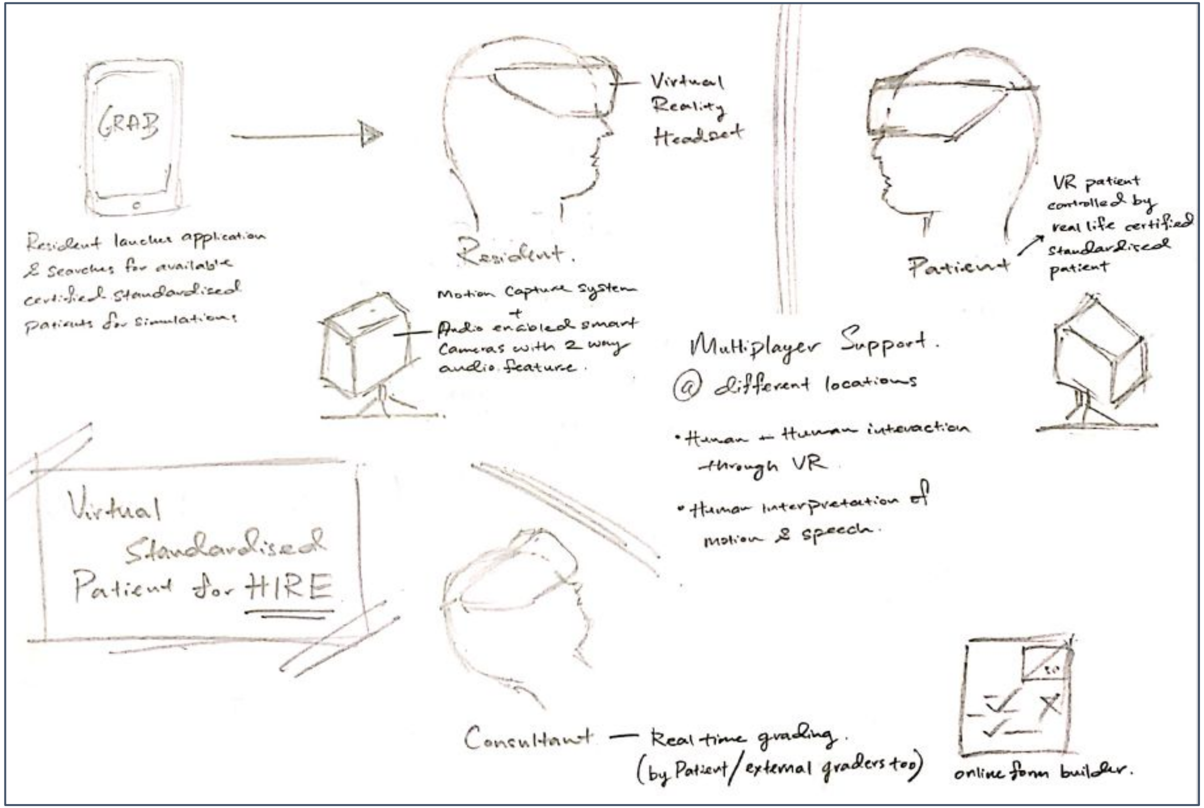
Solution D : AR Simulation + Full Body Haptic Suit



- Resident views and interacts with a virtual patient in a physical room tailored to mimic a clinical setting using Augmented reality technology
- Resident's movement will be captured and analysed by the full body haptic suit
- Full body haptic suit will produce the haptic feedback generated from the interaction with the virtual patient

Figure 5: Sketch of solution D

Solution E: Virtual Standardized Patient for Hire



- Resident first launches application and searches for available certified standardized patients to start the simulation
- Once the certified actor is matched to the Resident, both the Resident and the certified actor will put on VR headsets and enter the virtual reality simulation
- Simulation is carried out in a multiplayer setting, with the certified actor controlling the virtual patient avatar in real time.

Figure 6: Sketch of solution E

Functions	A: 2D Brain Controlled Computer Simulation	B: High Fidelity VR Simulation	C: Holographic Room	D: AR + Full Body Haptic Bodysuit	E: Virtual Standardized Patient for Hire
Speech Recognition of Residents	Translation of neural signals into speech	Speech to text software	Speech to text software	Speech to text software	Human assisted by speech to text
Dialogue Analysis of Residents	Natural Language Processing	Natural Language Processing	Dialogue Tree	Natural Language Processing	Human assisted by Natural Language Processing
Gesture Recognition of Residents	Translation of neural signals into motion	Motion capture system	Emission and reflection of ultrasound signals to detect movements and body position	Motion capture system	Motion capture system
Motion Interpretation of Residents	Deep learning algorithm	Smart Wearable Textile	Deep learning algorithm	Smart Wearable Textile	Human interpretation
Generation of Audio Output to Residents	Voice skinning	Text to speech software	Paid voice actors	Voice skinning	Human Voice
Generation of Visual Output to Residents	3D computer graphics software	3D computer graphic software	3D photogrammetry	3D photogrammetry	3D photogrammetry
Supervision of Training Session by Consultant	Video streaming	Video streaming	Holographic Projection	Video streaming	Video streaming
Communication Method between Residents and Consultant	Audio enabled smart cameras with 2-way audio feature	Audio enabled smart cameras with 2-way audio feature	2 way push to talk device	2 way push to talk device	Audio enabled smart cameras with 2-way audio feature
Transfer of Knowledge from Consultant to Residents	Live video conferencing	Virtual Consultant Avatar	Holographic projection of Consultants	Virtual Consultant Avatar	Virtual Consultant Avatar
Resident Performance Assessment	System marking based on model answer	System marking based on model answer	Real time grading by Consultant	System marking based on model answer	Real time grading by certified patient and external grader
Report Generation of Resident's performance	Clinical Playback	Leaderboard system	Online form builder	Leaderboard system	Online form builder

6 Concept Selection

Evaluate the concepts according to the key design specifications.

		Concepts									
		A		B		C (REF)		D		E	
SELECTION CRITERIA	Weight (%)	Rating	Weighted Score	Rating	Weighted Score	Rating	Weighted Score	Rating	Weighted Score	Rating	Weighted Score
Expandable Database	15	5	0.75	4	0.6	3	0.45	2	0.3	1	0.15
Adaptive Conversational Flow	15	4	0.6	4	0.6	3	0.45	4	0.6	5	0.75
Accurate Patient Behaviour	20	1	0.2	4	0.8	3	0.6	4	0.8	5	1
Accurate Anatomical Features	15	2	0.3	2	0.3	3	0.45	3	0.45	3	0.45
Sense of Touch Simulation	10	1	0.1	5	0.5	3	0.3	5	0.5	4	0.4
Immediate Intervention	15	1	0.15	3	0.45	3	0.45	3	0.45	3	0.45
Useful Grading System	10	5	0.5	5	0.5	3	0.3	5	0.5	3	0.3
	Total Score	2.6		3.75		3		3.6		3.5	
	Rank	5		1		4		2		3	
	Continue?	No		Yes		No		No		No	

6 Concept Selection

Evaluate the concepts according to the key design specifications.

		Explanation of Concepts Ratings									
		A		B		C (REF)		D		E	
SELECTION CRITERIA	**Weight (%)**	**Rating**	**Weighted Score**	**Rating**	**Weighted Score**	**Rating**	**Weighted Score**	**Rating**	**Weighted Score**	**Rating**	**Weighted Score**
Expandable Scenario Database	10	5	Database for computer games are easily updated and expanded. 3D graphic software will be better to create new scenarios, compared to 3D photogrammetry , which requires mulitple real life photos	4	3D graphic software will be better to create new scenarios, compared to 3D photogrammetry , which requires mulitple real life photos	3	3D graphic software will be better to create new scenarios, compared to 3D photogrammetry , which requires multiple real life photos	2	AR requires actual rooms and objects to overlay with 3D graphics. 3D graphic software will be better to create new scenarios, compared to 3D photogrammetry , which requires mulitple real life photos	1	This will be heavily based on the ability of the hired standardised patient to carry out the scenarios. 3D graphic software will be better to create new scenarios, compared to 3D photogrammetry , which requires mulitple real life photos
Adaptive Conversational Flow	15	4	NLP allows the generation of more natural and realistic responses according to the conversation history	4	NLP allows the generation of more natural and realistic responses according to the conversation history.	3	Dialogue tree is a set of predetermined responses.	4	NLP allows the generation of more natural and realistic responses according to the conversation history	5	Actual human conversation is the more natural. Human-human interactions are also more realistic.
Accurate Patient Behaviour	20	1	As it is a computer game, simulation is not fully immersive	4	It is fully immersive due to the recreation of the entire simulation environment	3	It is semi-immersive as the simulation environment is not recreated. Furthermore, the generation of audio output is via pre-recorded audio files	4	It is fully immersive due to the recreation of the entire simulation environment	5	Human-human interactions are also more realistic.
Accurate Anatomical Features	15	2	Generation of anatomical features using graphic software is not as accurate and reaslistic as 3D model reconstruction from photos	2	Generation of anatomical features using graphic software is not as accurate and reaslistic as 3D model reconstruction from	3	Generation of anatomical features using graphic software is not as accurate and reaslistic as 3D model reconstruction from photos	3	Generation of anatomical features using graphic software is not as accurate and reaslistic as 3D model reconstruction from photos	3	Generation of anatomical features using graphic software is not as accurate and reaslistic as 3D model reconstruction from photos
Sense of Touch Simulation	5	1	Low quality visual haptic feedback as it is a 2D computer simulation	5	High visual and audio feedbackas users is immersed in the virtual environment. Smart wearible textile can simulate the sense of touch and force feedback	3	Medium quiality visual haptic feedback as it is semi-immersive smulation	5	High visual and audio feedbackas users is immersed in the virtual environment. Smart wearible textile can simulate the sense of touch and force feedback	4	High visual and audio feedback as users is immersed in the virtual environment
Immediate Intervention	15	1	Live video conferencing only allows audio explanation. Consultants are unable to demostrate proper techniques in computer games	3	Virtual consultant avatar allows consultants to interact and demostrate proper techniques on the virtual patient within the simulation	3	Virtual consultant avatar allows consultants to interact and demostrate proper techniques on the virtual patient within the simulation	3	Virtual consultant avatar allows consultants to interact and demostrate proper techniques on the virtual patient within the simulation	3	Virtual consultant avatar allows consultants to interact and demostrate proper techniques on the virtual patient within the simulation
Useful Grading System	10	5	Standardised marking system with no biasness.	5	Standardised marking system with no biasness.	3	Real-time marking may results in errors and missing details. Performance biasness may be present due to human biasness	5	Standardised marking system with no biasness.	3	Real-time marking may results in errors and missing details. Performance biasness may be present due to human biasness

7 Prototyping

Describe the prototypes made and highlight the key findings.

Upon selecting concept B (High Fidelity VR Simulation), we took a modular approach to break down our prototype into 2 key independent functions. This allows both components to be improved on simultaneously without affecting each component's development.Through prototyping, we can gain a better understanding of the limitations of existing software and the feasibility of our concept design.

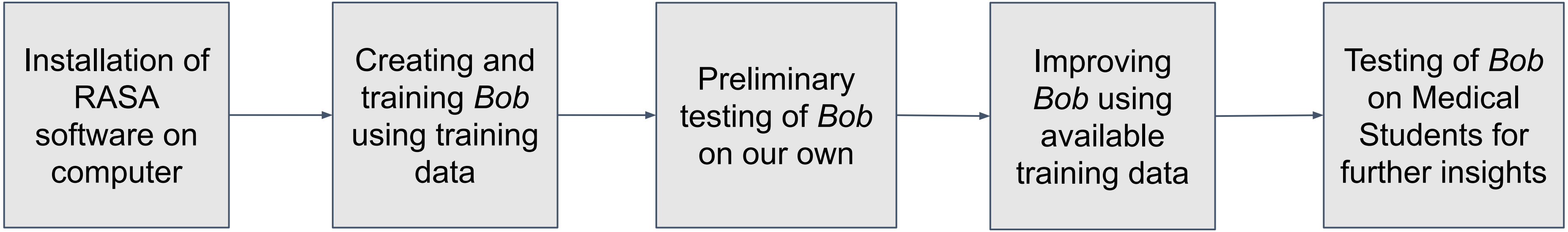
Prototyping Areas		**Function/Objective**	**Software used**
Conversational agent		Analyses speech inputs and generates verbal and physical responses from the virtual patient	RASA conversational agent
Animation	Motion animation	Animates the virtual patient's motions for generating visual output to residents	1. Blender 2. Microsoft Kinect 360 3. Unity
	Facial animation	Animates facial expressions and emotions to accurately depict the nature and location of the patient's pain	1. Blender 2. Unity

7 Prototyping

Conversational Agent: Overview

We utilized the tools and infrastructure provided by RASA open source to create *Bob*, a contextual assistant that we hope will eventually be able to mimic the behaviour of an actual patient.

Prototyping Process



7 Prototyping

Conversational agent: Workflow

Start

Input by Resident through speech

"Hello! My name is Eldred, and I will be the doctor attending to you today"

→

***Bob*'s thought process**

Bob utilizes a pipeline known as Supervised_embeddings* to determine the intent of the Resident's input

"Hello! My name is Eldred, and I will be the doctor attending to you today."

↓

Supervised_embeddings pipeline

↓

"**Introducing_yourself**" intent identified

→

Bob then uses different policies to determine the response to the intent

"**Introducing_yourself**" intent

↓

Augmented Memoization Policy, Keras Policy, Mapping Policy and Fall back Policy

↓

Bob identifies "**utter_chit_chat_introduction**" as the most appropriate response

→

End

Output of "**utter_chit_chat_introduction**" by *Bob*

" Hi! I'm *Bob*, an AI patient simulation. Today I would be simulating a 60 year old uncle. Please be kind with me! Thank you so much!"

*Further details found in appendix

7 Prototyping

Conversational agent: Training of *Bob*

Training was done by keying in the necessary details into the data files

```
## intent:chit_chat_introduction
- who are you?
- what are you?
- who is this?
- tell me about yourself.
- Who are you?
- Who r you?
- Hello! My name is Eldred. It is Nice to meet you!
- Hi! I am Doctor Eldred, and I will be the Doctor attending to you
- Hello! I am Eldred, and I will be the Doctor treating you today
```

Training data to help *Bob* identify the intent of "**chit_chat_introduction**"

```
## Training story 1

* chit_chat_introduction: Hi! I'm Eldred, the doctor treating you today
    - utter_chit_chat_introduction

## Training story 2

* chit_chat_introduction: Hi! I am Eldred, the person that will be helping you today!
    - utter_chit_chat_introduction

## Training story 3

* chit_chat_introduction: Hello! I am Eldred. Nice to meet you!
    - utter_chit_chat_introduction
```

Training stories to help *Bob* to link the intent detected "**chit_chat_introduction**" to the appropriate response "**utter_chit_chat_introduction**"

```
utter_chit_chat_introduction:
- custom:
    text: Hi! I'm Bob, an AI patient simulation. Today I would be simulating
      a
      60 year old uncle. Please be kind with me! Thank you so much!
```

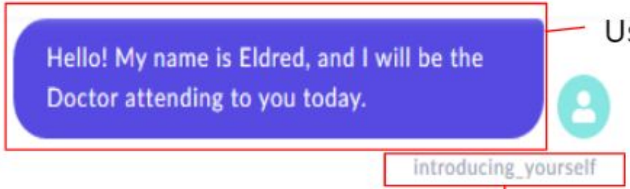
The predefined "**utter_chit_chat_introduction**" response

Figure 7: *Bob's* training data

7 Prototyping

<u>Conversational agent: Preliminary testing of *Bob*</u>

We spoke to *Bob* on RASA-X* to see if he was able to recognise our intent and give the correct reply



User Input

Intent *Bob* identifies from the Resident's input

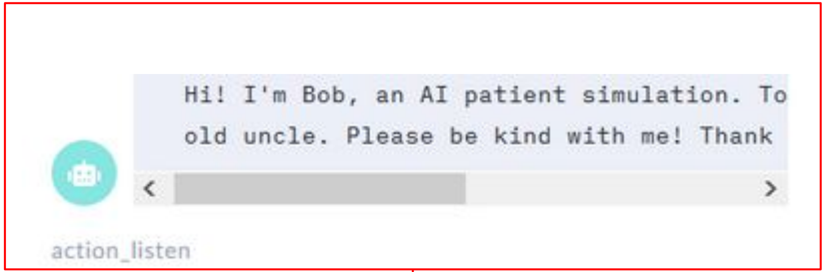


Figure 8: *Bob's* response

Bob outputs "**utter_chit_chat_introduction**" action in response to identify Resident's intent. The full text of the response is: "Hi! I'm *Bob*, an AI patient simulation. Today I would be simulating a 60 year old uncle. Please be kind with me! Thank you so much!"

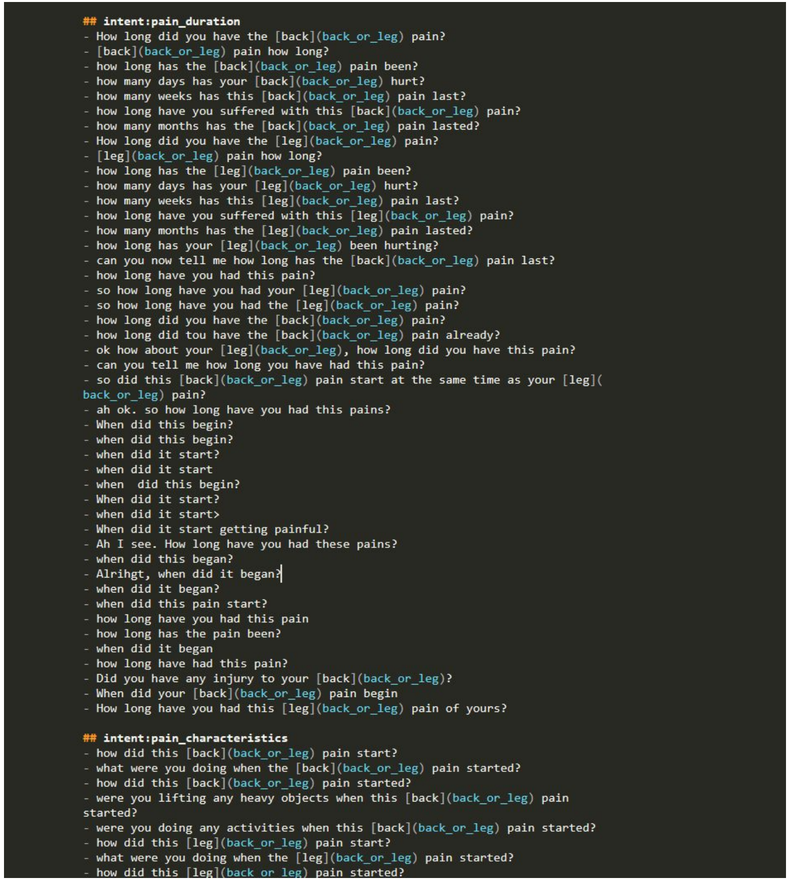
**RASA-X is an interface we use to communicate with Bob, further details can be found in appendix*

7 Prototyping

Conversational agent: Improving *Bob*

Bob was then trained further by utilizing an existing database containing a large amount of training data. This data was provided by Dr Tan Jun Hao from National University Hospital (NUH) Orthopedics, and compiled by Aiden Koh (Year 3 NUS IDP student)

Bob was modelled after a 60 year old Patient with L4/5 degenerative spondylolisthesis (slippage of spinal discs due to bone degeneration).



```
## intent:pain_duration
- How long did you have the [back](back_or_leg) pain?
- [back](back_or_leg) pain how long?
- how long has the [back](back_or_leg) pain been?
- how many days has your [back](back_or_leg) hurt?
- how many weeks has this [back](back_or_leg) pain last?
- how long have you suffered with this [back](back_or_leg) pain?
- how many months has the [back](back_or_leg) pain lasted?
- How long did you have the [leg](back_or_leg) pain?
- [leg](back_or_leg) pain how long?
- how long has the [leg](back_or_leg) pain been?
- how many days has your [leg](back_or_leg) hurt?
- how many weeks has this [leg](back_or_leg) pain last?
- how long have you suffered with this [leg](back_or_leg) pain?
- how many months has the [leg](back_or_leg) pain lasted?
- how long has your [leg](back_or_leg) been hurting?
- can you now tell me how long has the [back](back_or_leg) pain last?
- how long have you had this pain?
- so how long have you had your [leg](back_or_leg) pain?
- so how long have you had the [leg](back_or_leg) pain?
- how long did you have the [back](back_or_leg) pain?
- how long did tou have the [back](back_or_leg) pain already?
- ok how about your [leg](back_or_leg), how long did you have this pain?
- can you tell me how long you have had this pain?
- so did this [back](back_or_leg) pain start at the same time as your [leg](
back_or_leg) pain?
- ah ok. so how long have you had this pains?
- When did this begin?
- when did this begin?
- when did it start?
- when did it start
- when  did this begin?
- When did it start?
- when did it start>
- When did it start getting painful?
- Ah I see. How long have you had these pains?
- when did this began?
- Alrihgt, when did it began?
- when did it began?
- when did this pain start?
- how long have you had this pain
- how long has the pain been?
- when did it began
- how long have had this pain?
- Did you have any injury to your [back](back_or_leg)?
- When did your [back](back_or_leg) pain begin
- How long have you had this [leg](back_or_leg) pain of yours?

## intent:pain_characteristics
- how did this [back](back_or_leg) pain start?
- what were you doing when the [back](back_or_leg) pain started?
- how did this [back](back_or_leg) pain started?
- were you lifting any heavy objects when this [back](back_or_leg) pain
started?
- were you doing any activities when this [back](back_or_leg) pain started?
- how did this [leg](back_or_leg) pain start?
- what were you doing when the [leg](back_or_leg) pain started?
- how did this [leg](back_or_leg) pain started?
```

Figure 9: Short snippet of database used to train *Bob*

7 Prototyping

Conversational agent: Results

After *Bob* was trained with the existing database, we tested *Bob*'s capabilities on RASA-X again.

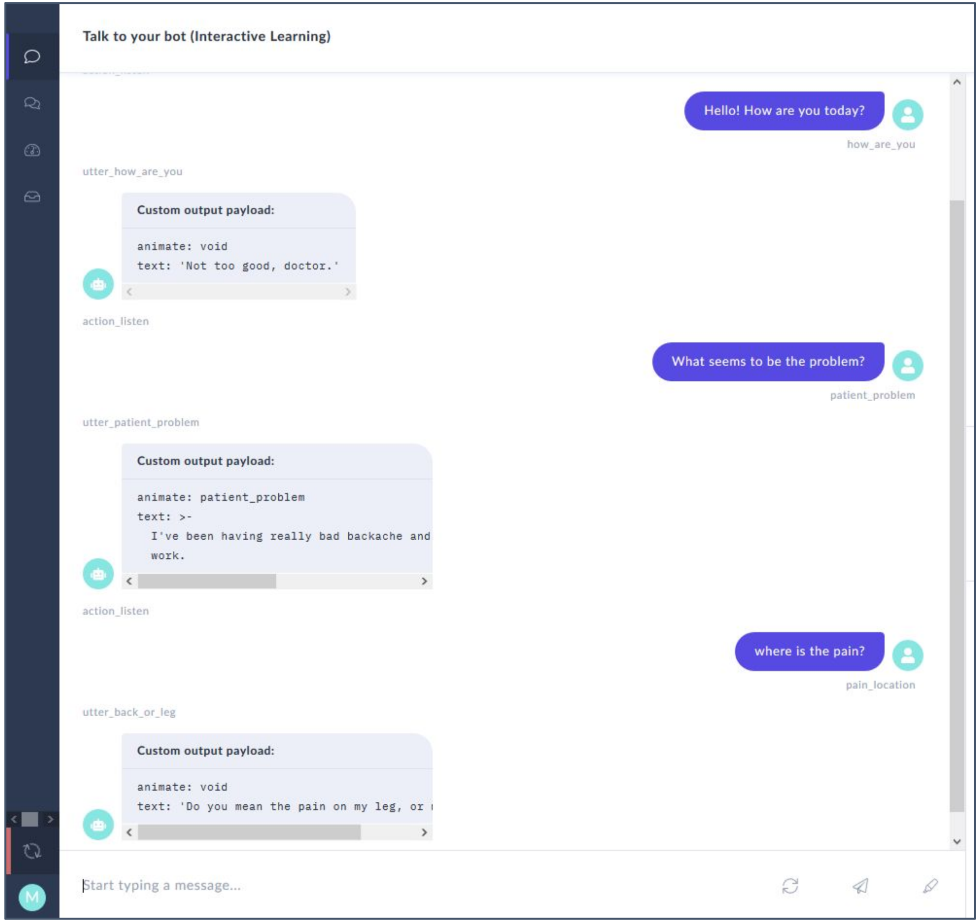


Figure 10: Testing of improved *Bob*

7 Prototyping

<u>Conversational agent: Testing *Bob* with Medical Student</u>

Shiwei is a Year 3 NUS Medical Student who had studied orthopedic clinicals (both history taking and physical examinations*).

How testing was conducted:

- Shiwei communicates with *Bob* through a telegram bot**
- *Bob* generates a response for Shiwei
- In the event that *Bob* fails to generate an acceptable response, the team steps in and enters a custom response to ensure the conversation with Shiwei continues to flow

Figure 11: Testing conducted with Year 3 Medical Student, Xiao Shiwei

**Refer to appendix for more details on the structure of Orthopedic clinicals.*
***Refer to appendix for more details on how the telegram bot is managed.*

7 Prototyping

<u>Conversational agent: Testing *Bob* with Medical Student</u>

During the trial test with Shiwei, there were instances where *Bob* failed and we had to step in to input our own custom responses

Figure 12: Interaction between *Bob* and Shiwei

7 Prototyping

Conversational agent: Key Findings

Key findings identified from the user testing of *Bob* with Shiwei	
Strengths	*Bob* was able to understand most of the questions asked and generate realistic responses accordingly. This gives us validation that *Bob* has achieved what we intended it to do
Limitations (listed according to insightfulness)	● The medical student highlighted that *Bob*'s database needs to be further expanded to respond to inputs related to cancer, weakness and numbness of body, as well as Activities of Daily Living (ADL) independency *(most insightful)* ● *Bob* is unable to understand sentences when there are multiple possible answers (e.g. “When do you feel the most pain?” can be answered with the time of the day - morning/night, or when the leg is in a specific position - flexed/extended) ● *Bob* is unable to understand user inputs if they are phrased very differently from the training data ● *Bob* is unable to understand a user input with multiple intents *(least insightful)* (i.e. “Did the **back pain** and **leg pain** start at the same time?”)
Evaluation	*Bob* is a feasible conversational agent that can be incorporated into our Virtual Reality Simulation Tool. However, considerable work has to be done to expand *Bob*'s database and train him to be able to handle the rigour of an actual simulation

7 Prototyping

Animation: Overview

The aim of this animation process is to model a realistic 3D virtual patient, *Bob*, that can simulate the actions required during clinicals. Unity is an essential software for our project to assemble all the 3D models and animations exported from Blender and Kinect 360. This will help to explore the feasibility of building a virtual reality simulation tool.

Motion and facial animation involves 3 steps:

1) **Rigging**: Process of creating bone structure of a 3D model. By manipulating the bone structure of *Bob*, we are able to transform and animate him.
2) **Animation Creation**: Recording and animating *Bob*'s movements and facial expressions
3) **Model Control**: Controlling *Bob* based on user input.

Function	Software used
Rigging	Blender
Animation Creation	Blender, Kinect 360
Model Control	Unity

7 Prototyping

Animation: Results of Animation Creation*

*More details on Animation creation process using Blender and Kinect 360 can be found in the appendix

Bob is able to demonstrate a posture of back pain accompanied by a sad and painful facial expression.

Figure 13: Side view

Figure 14: Side view

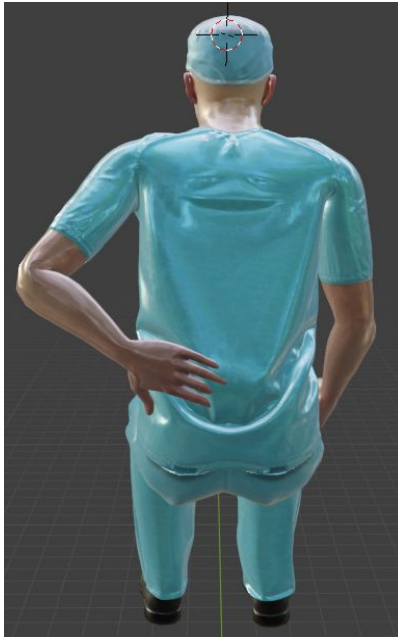


Figure 15: Back view

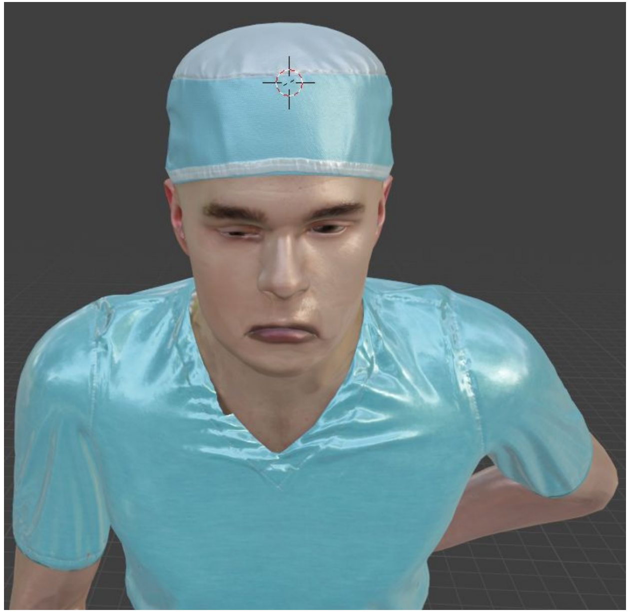


Figure 16: Sad and painful facial expression

7 Prototyping

Animation: Results of Model Control in Unity

Figure 17: Blender animation controlled by Unity to check for *C5 motor functionality by testing shoulder abduction strength.

Figure 18: Kinect animation controlled by Unity to check for *C6 motor functionality by testing plantar flexion strength.

*More details on C5/C6 motor functionality can also be found in the appendix

7 Prototyping

Animation: Evaluation of Animation Creation

In process of creating our custom animations, we have evaluated each process as shown below:

	Blender (Motion and Facial)	Xbox Kinect (Motion only)
Strengths	Allows more precise control over motions.	Able to generate many frames of custom animation at a fast pace (a 20 second video captured at 30fps generates 600 frames).
Limitations	Very time consuming to individually rotate each bone per frame to make the entire animation. It lacks the realism of creases in the skin and muscles during facial expression and movements.	Motion captured may be inaccurate especially when the subject's body part is behind another (e.g. hand behind the leg). Requires additional hardware to run i.e. Kinect 360 sensor.
Evaluations	This should be used as a post-processing tool to clean up inaccurate motions from Kinect.	This is feasible to create custom animations quickly.

7 Prototyping

Animation: Evaluation of Model Control

After creating our custom animations, we imported it to Unity to control the animation. The 2 different ways of controlling *Bob* in Unity are evaluated below:

	Animation Tab*	**Script for Custom Animation***
Strengths	Quicker to animate as it has a graphic user interface (GUI) to drag and drop animations.	Allows for more customizability, and can be extended to speech inputs, rather than just a fixed number of keyboard inputs.
Limitations	Limited in the ability to customize and control our character animations.	It is more time consuming to code for individual animations.
Evaluations	This should be done for quick testing to check if the animation is correctly imported into Unity.	This should be done for our final prototype as it is compatible with more types of user inputs. This will allow better integration with *Bob* when assembling our prototype.

* Details of the two methods used to trigger animations in Unity can be found in the appendix

Appendix (Background) Done by Group

Orthopedic Spine Clinicals

Orthopedic Spine Clinicals are typically conducted in two phases, the history taking phase and the physical examination phase. Based on our observations from clinicals conducted at NUH Orthopedic Spine Clinic:

The history taking phase involves the resident rapidly asking a set of questions to the patient to narrow down their condition.

The physical examination phase involves the resident testing the sensory and motor functionality of the spinal nerves shown in the diagram (Figure 19).

We aim to replicate the history taking process with the conversational agent, and some of these physical tests in our animation process.

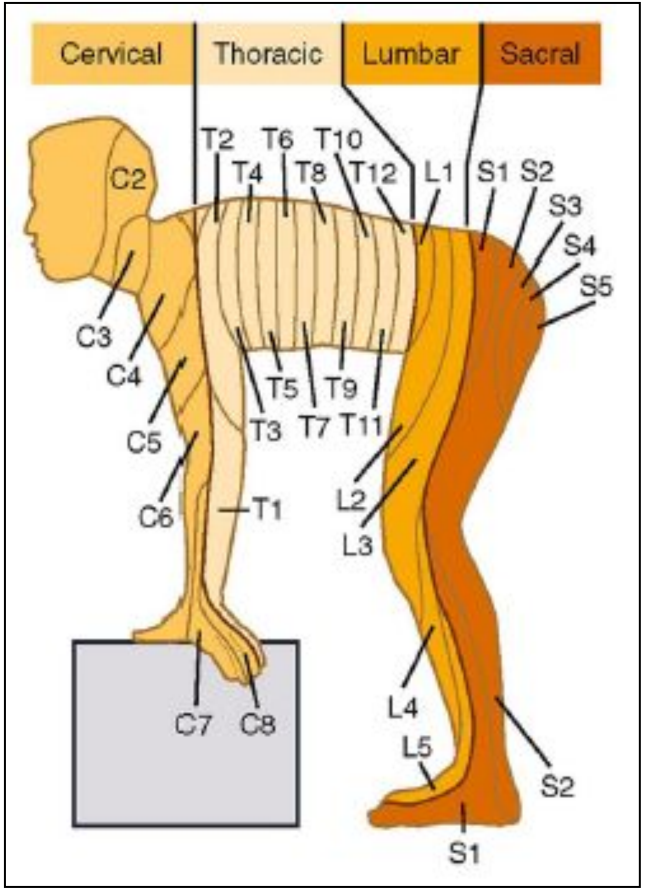


Figure 19: Representation of spine anatomy

Appendix (RASA) Done by Eunyu

RASA Component Analysis: Pipelines

A sequence of processing steps which are used to extract specific text features and train certain components which allows the model to understand patterns

Supervised_embeddings configuration

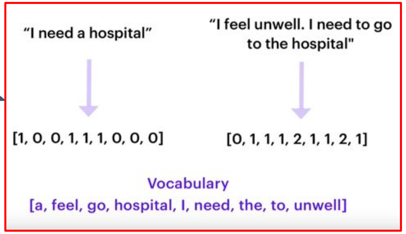
```
language: "en"

pipeline:
- name: "WhitespaceTokenizer"
- name: "RegexFeaturizer"
- name: "CRFEntityExtractor"
- name: "EntitySynonymMapper"
- name: "CountVectorsFeaturizer"
- name: "CountVectorsFeaturizer"
  analyzer:"char_wb"
  min_ngram: 1
  max_ngram: 4
- name: "EmbeddingIntentClassifier"
```

Explanation of key components

- **WhitespaceTokenizer** : Looks for a white space and creates a division

- **CountVectorsFeaturizer** : Breaks each text input into an array of words and counts how many times each word appears in each text input



- **EmbeddingintentClassifier** : Takes array output from CountVectorsFeaturizer and uses it to determine most probable intent based on training data

Figure 20: Examples of pipelines

Appendix (RASA) Done by Eunyu

RASA Component Analysis: Policies

- Component responsible for training a dialogue model to decide how the assistant should respond next.
- Each policy in the configuration makes its own prediction about the next best action. The policy that predicts the next action with the highest confidence level determines the assistant's next action.

Types of policies:

1. Augmented Memoization Policy

- Mimics the stories it was trained on
- Match the current stories with training provided in the data file
- If current conversation matches existing training data exactly, it predicts output with a probability of 1
- If there is no exact match, output is predicted with probability 0
- Needs to be used in conjunction with other policies

2. Mapping policy

- Maps an intent to a specific action
- Useful when you know that an intent should always be followed by a certain response, regardless of what has happened previously in the conversation (eg asking whether it is a bot)
- Too specialised, so it needs to be used with other policies

Appendix (RASA) Done by Eunyu

RASA Component Analysis: Policies

3. Keras policy

- Machine learning (learns from training data)
- Takes into account:
  - What was the last action
  - What intents were predicted for current user input

4. Fallback policy

- Activated when the assistant can't predict an intent or next action with certainty above a certain threshold or when confidence levels of two intents are very close together
- Default action is then triggered

Appendix (RASA) Done by Eldred

RASA-X Deployment

After Bob was trained using the training data, we deployed Bob onto RASA-X, an online interface that allows us to more easily communicate with him

Deployment Process

Appendix (RASA) Done by Eldred

RASA-X Deployment

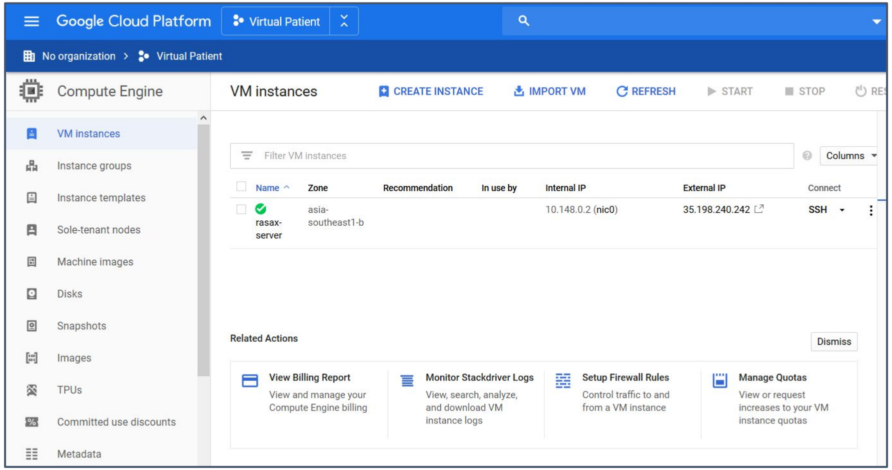


Figure 21: Google Cloud interface used to control the Virtual Machine instance hosting the RASA-X server

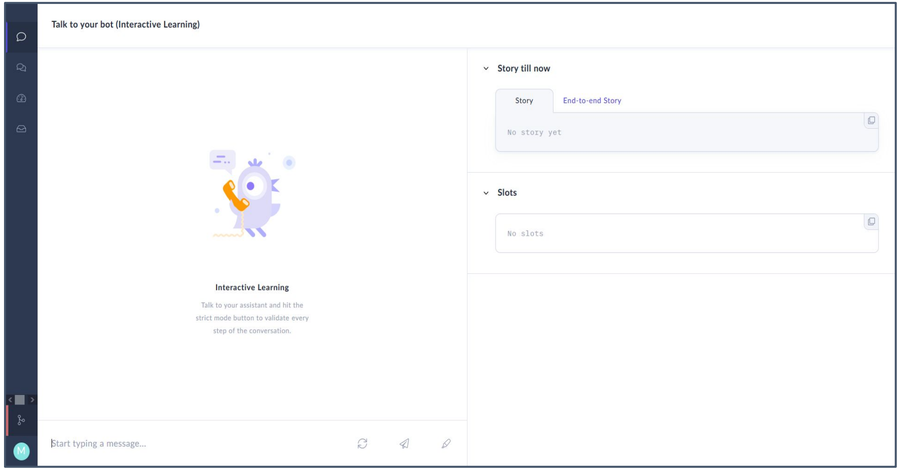


Figure 22: RASA-X interface used to communicate with *Bob* easily

Appendix (RASA) Done by Eldred

Testing *Bob* with Medical Student

We do the testing via a telegram bot rather than allowing Shiwei to interact directly with *Bob* via the RASA-X interface. This is to allow us to enter a custom response in the event that *Bob* fails to generate a reply, ensuring that the conversation flows smoothly.

Testing Process

Appendix (Animation) Done by Augustine and Alvin

Rigging: Blender

Blender was used for rigging the face and the body segments of *Bob* (25 bones for the face and 52 for the body). This is the bare minimum required to create the necessary motions for the residents to conduct clinicals.

Figure 23: Rigging of face

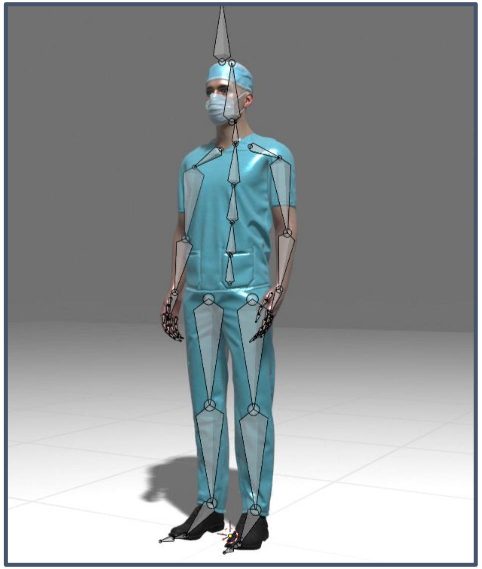


Figure 24: Rigging of body

Appendix (Animation) Done by Alvin

Animation Creation (Facial): Blender

The transition from 0th to 20th position is saved as our custom animation before exporting to Unity.

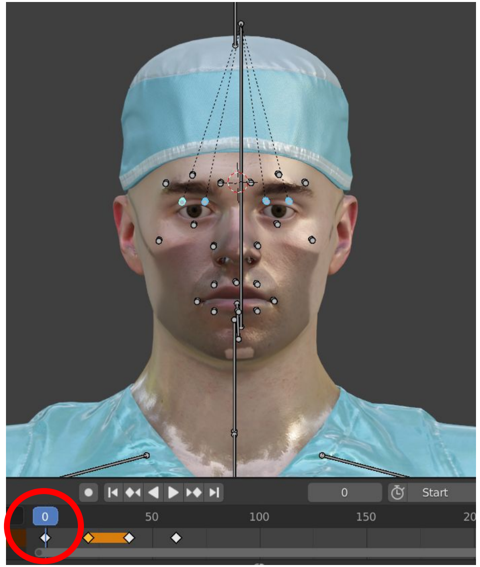


Figure 25: position of joints at 0th frame

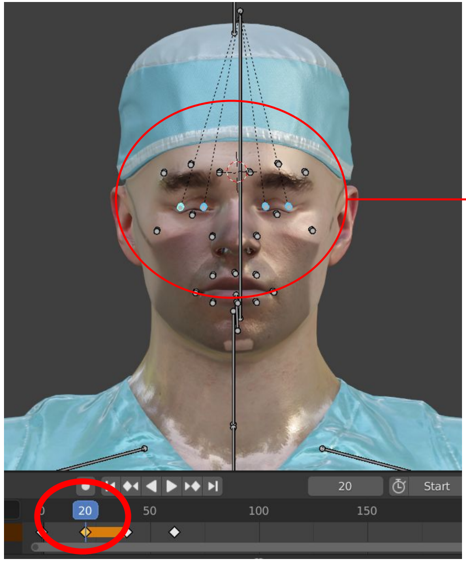


Figure 26: position of joints at 20th frame

We created our own animations by individually moving the joints to its desired position.

Appendix (Animation) Done by Augustine

Animation Creation (Motion): Blender

The transition from 0th to 20th position is saved as our custom animation before exporting to Unity.

Figure 27: position of joints at 0th frame

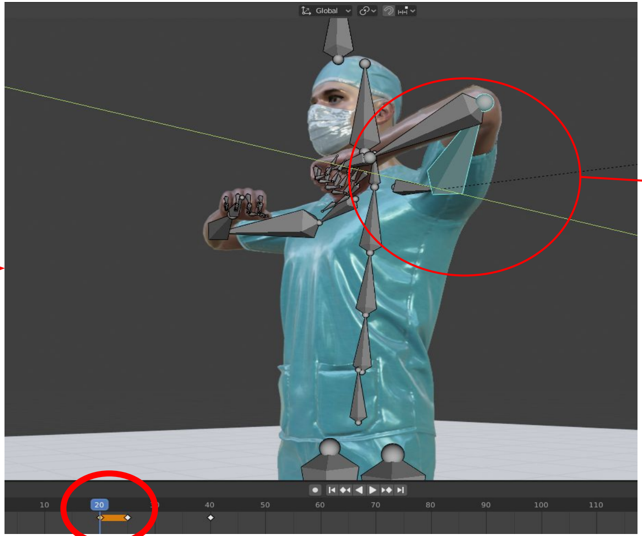


Figure 28: position of joints at 20th frame

We created our own animations by individually rotating the joints to its desired position.

Appendix (Animation) Done by Augustine

Animation Creation: Kinect 360

We also used a Xbox 360 Kinect sensor with iPi Recorder to record custom animations as an alternative method to Blender.

iPi Mocap Studio was used to map the recorded model to the bones of our virtual patient model. This was exported to Unity.

Figure 29: Hardware setup used to capture the animation.

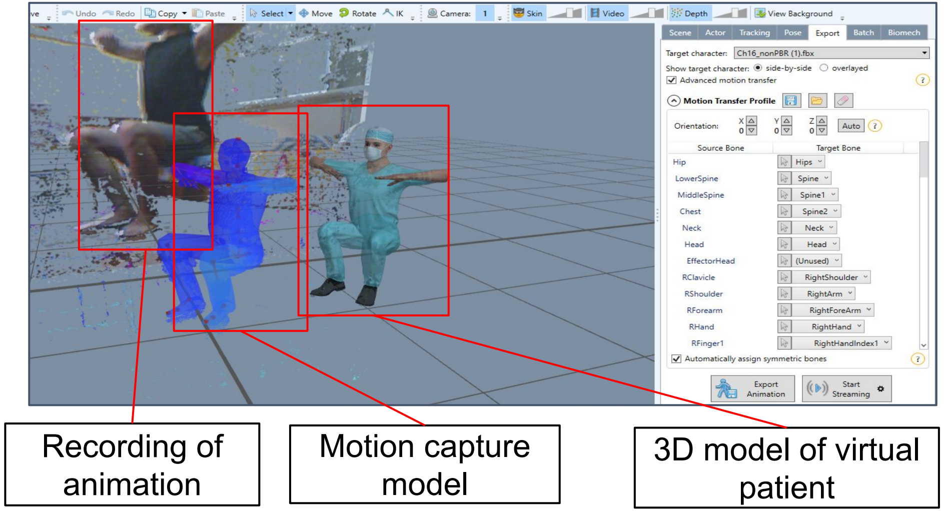


Figure 30: iPi Mocap Studio software setup

Appendix (Animation) Done by Augustine and Alvin

Model Control: Unity

Upon creation of the animation, the following processes are carried out in Unity

1. Importing of the model and animation → 2. Controlling the model (2 different ways)

a) Use of Graphic User Interface (GUI) to sequence the animations according to the resident's needs

OR

b) Scripting our own C# programming language for the triggering of the actions

1: Importing the model and the animation
A 3D virtual model was downloaded from mixamo.com. The animation that we imported will be displayed through this model.

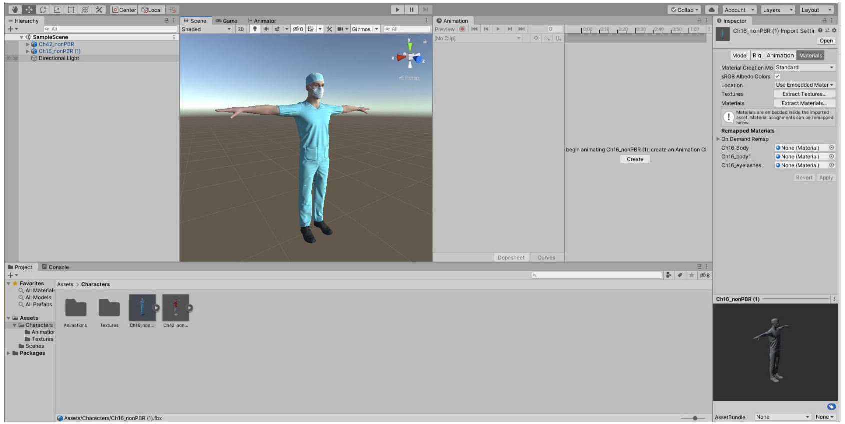


Figure 31: Default layout in Unity with our character model

Appendix (Animation) Done by Alvin

<u>Model Control: Unity</u>

a) Using the animator tab, we were able to use the Graphic User Interface (GUI) to sequence the animations according to the resident's needs.

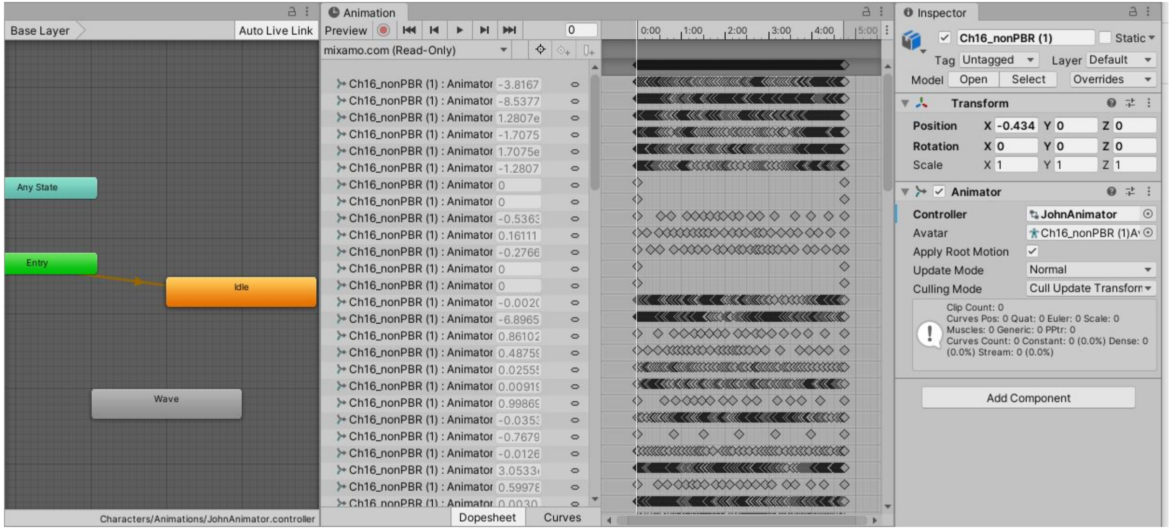


Figure 32: Unity's animation tab

Figure 33: *Bob* in "Idle" position as sequenced in Unity's animator tab

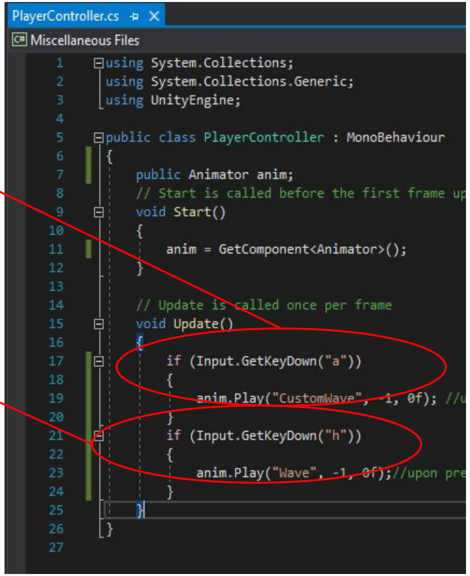
Appendix (Animation) Done by Augustine

Model Control: Unity

b) Scripting using C# (programming language) allows us to assign predefined user inputs (e.g. keyboard, microphone input) to trigger patient response.

Assigns the key “a” to trigger animation “CustomWave”

Assign the key “h” to trigger animation “Wave”

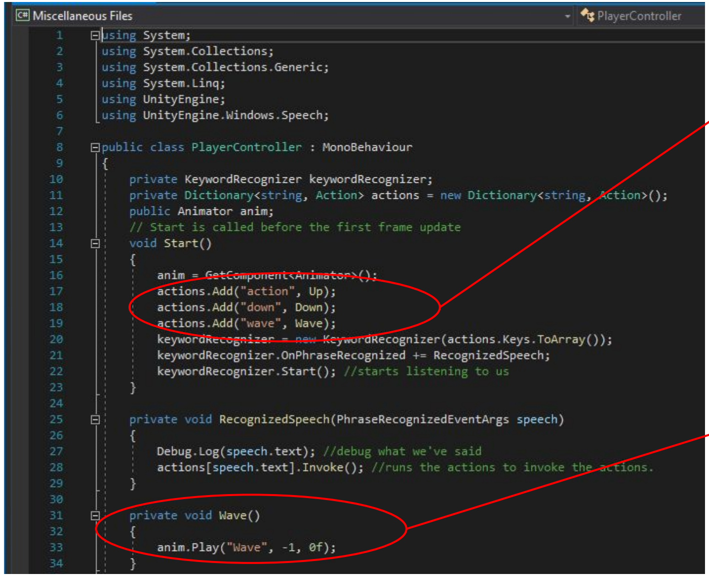

PlayerController.cs
Miscellaneous Files

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class PlayerController : MonoBehaviour
{
    public Animator anim;
    // Start is called before the first frame up
    void Start()
    {
        anim = GetComponent<Animator>();
    }

    // Update is called once per frame
    void Update()
    {
        if (Input.GetKeyDown("a"))
        {
            anim.Play("CustomWave", -1, 0f); //u
        }
        if (Input.GetKeyDown("h"))
        {
            anim.Play("Wave", -1, 0f);//upon pre
        }
    }
}
```

Figure 34: Assigning keyboard control to trigger motion


Miscellaneous Files
PlayerController

```
using System;
using System.Collections;
using System.Collections.Generic;
using System.Linq;
using UnityEngine;
using UnityEngine.Windows.Speech;

public class PlayerController : MonoBehaviour
{
    private KeywordRecognizer keywordRecognizer;
    private Dictionary<string, Action> actions = new Dictionary<string, Action>();
    public Animator anim;
    // Start is called before the first frame update
    void Start()
    {
        anim = GetComponent<Animator>();
        actions.Add("action", Up);
        actions.Add("down", Down);
        actions.Add("wave", Wave);
        keywordRecognizer = new KeywordRecognizer(actions.Keys.ToArray());
        keywordRecognizer.OnPhraseRecognized += RecognizedSpeech;
        keywordRecognizer.Start(); //starts listening to us
    }

    private void RecognizedSpeech(PhraseRecognizedEventArgs speech)
    {
        Debug.Log(speech.text); //debug what we've said
        actions[speech.text].Invoke(); //runs the actions to invoke the actions.
    }

    private void Wave()
    {
        anim.Play("Wave", -1, 0f);
    }
```

Figure 35: Assigning speech control to trigger motion

Calls animation functions when the words are recognized by our system

Triggers the animation “Wave” when the user says the word “wave”.