MACHINE LEARNING MODELS FOR GROSS-MOTOR SKILLS

Benoit Bossavit

CS2013 – CS3013: Sample Outline for a Project Plan

Table of Contents

1. Project Goals and Objectives

1.1. Background

(Describe the context of the project and the activities of the sponsor/client)

Our client, Benoit Bossavit, is working on a research project for SCSS. He is focusing on the ability of young children to perform gross-motor skills. Our client is using a Microsoft XBOX Kinect to collect motion capture data to track the movement of test subject's joints. With this motion capture data, he was able to map each joint's position to create a 3-D 'skeleton' for each frame based on the XYZ coordinates of each joint in each frame.

For the client's research he has a set of six motor-skills that he examines. He would like for us to design and create a set of machine learning models for identifying the gross-motor skill being performed and the amount of times it is successfully completed in each file. We have been instructed that we have free reign on the design of the models. If time is an issue our client has specified that he would be satisfied with a set of machine learning models for recognizing the jog, hop-right and hop-left motor skills.

1.2. Objectives

(Describe how will the successful completion of this project assist the sponsor/client. Comment on the impact this project will have)

Our involvement in and successful completion of this project will allow the client to focus on his research and devote more time to the motion capture rather than having to spend time developing a machine learning model himself. The objective is to save our client time and effort and allow our client to dedicate more attention to the research at hand rather than having to learn about machine learning model development which he has no prior experience in. By having our team develop the machine learning models it means he does not have to build them himself which would take away from his work.

1.3. Goals

(Describe what will the project deliver)

Our goal is to deliver to our client a useful and accurate application built around the use of machine learning models that will benefit our client in his research. We aim to produce machine learning models that can identify gross-motor skills based on a series of motion capture frames and to determine the number of times each motion is performed per file. The final deliverable project will be in the form of an application our client will be able to run on his computer locally as this is where he has the data stored from his research. As well as providing the final product, we will also be providing the client with any documentation we have produced in order to allow further development of our models. For example, if our client decides to incorporate a new motion into his research, having our code documentation will allow the new team to understand the approach we took and how to go about training a new model that will be compatible with our model or updating our model

2. Project Scope

2.1. Scope

This project consists of developing an efficient and accurate machine learning model that can identify what movement is being performed when a json file containing motion capture data is passed through the model.

The machine learning model will be presented with a motion capture data file and will determine which motor-skill it is relating to. It will look to determine how many times the skill was performed for each motion capture data file. In order for a motion to be considered 'successful' the machine learning model determines if it was performed sufficiently, i.e. if the motor-skill being checked for is a hop and the child jumped off one foot but landed on two, this is not considered a successful iteration of the motor-skill.

2.2. Project Deliverables

(Describe the deliverables of the project: code and documentation)

Code:

The code we will be providing as part of our deliverables includes a code bundle with the following:

- A fully functioning application where our client can upload files via a GUI and have these files be analysed to determine the movement and iterations pertaining to each file
- Machine learning models with the ability to accurately identify which motion is being performed

Documentation:

The documentation we will be providing as part of our deliverables includes the following documents

- Requirements Document:

  This is a document signed off by the client detailing what we plan to deliver at the end of the module and how we intend to deliver on these requirements

- Requirements Presentation

  This is a power presentation that details the product we plan to deliver and our plan on how to do so which was presented during class.

- Project Plan

  This is a document detailing the timetabling and scheduling for how we intend to complete our project

- Software Design Specification

  This is a document detailing an in-depth/high-level evaluation of the design approaches and decisions made on how the application will function

- Final Presentation

The final presentation is a PowerPoint that details our final products development and includes a live demonstration to be presented in class

- Development Report

  This document is to be completed by the 2nd years and details the development choices made for this project

- Management Report

  This document is to be completed by the 3rd years and will detail the management choices they made during this project

- Individual Reflection

  This is an individual document each member of the team must produce detailing their thoughts and experiences of the project

2.3. Project Boundaries

(Outline what is in scope and out of scope the project boundaries.)

2.3.1. In Scope

- The ability to upload files to the application via drag and dropping into GUI or select file window
- The ability to validate the file type uploaded is of. json type
- Machine learning models be able to identify which of the six motions was performed
- Machine learning models count only successfully completed motor-skills
- The ability to display results to the GUI
- The ability to export the results as a .csv file
- The machine learning models should have at least a 90% accuracy rate

2.3.2. Out of Scope

- The ability to process files other than .json files
- The ability to select the format to export the results as (e.g. csv, word, notepad etc.)
- Update the application to be able to decompress files that have been uploaded in a compressed format

2.4. Product Backlog

(Provide the product backlog which should include: requirements, new features, changes to existing features, bug fixes, infrastructure changes etc.

(Include the Requirements' Document and the Software Design Specification in Appendices 1 & 2 respectively))

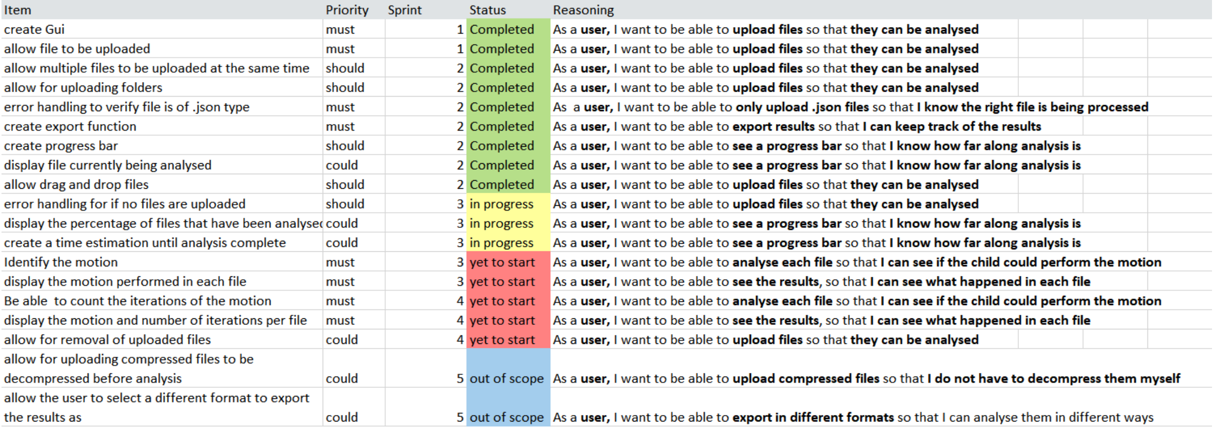
Our project had no prior or existing infrastructure, or feature, because of this we did not have to worry about fixing any existing bugs or changing existing features or infrastructure.

The requirements of our product include:

- upload files
- determine what motion is captured in each file
- determine how many times the motion is successful performed
- produce results to be displayed

The features our product will have include:

- upload files by drag and drop or file select window
- upload single or multiple files or folders
- Analyse files to determine motion and iterations
- Display progress of analysis
- export results



Item	Priority	Sprint	Status	Reasoning
create Gui	must	1	Completed	As a **user,** I want to be able to **upload files** so that **they can be analysed**
allow file to be uploaded	must	1	Completed	As a **user,** I want to be able to **upload files** so that **they can be analysed**
allow multiple files to be uploaded at the same time	should	2	Completed	As a **user,** I want to be able to **upload files** so that **they can be analysed**
allow for uploading folders	should	2	Completed	As a **user,** I want to be able to **upload files** so that **they can be analysed**
error handling to verify file is of .json type	must	2	Completed	As a **user,** I want to be able to **only upload .json files** so that **I know the right file is being processed**
create export function	must	2	Completed	As a **user,** I want to be able to **export results** so that **I can keep track of the results**
create progress bar	should	2	Completed	As a **user,** I want to be able to **see a progress bar** so that **I know how far along analysis is**
display file currently being analysed	could	2	Completed	As a **user,** I want to be able to **see a progress bar** so that **I know how far along analysis is**
allow drag and drop files	should	2	Completed	As a **user,** I want to be able to **upload files** so that **they can be analysed**
error handling for if no files are uploaded	should	3	in progress	As a **user,** I want to be able to **upload files** so that **they can be analysed**
display the percentage of files that have been analysec	could	3	in progress	As a **user,** I want to be able to **see a progress bar** so that **I know how far along analysis is**
create a time estimation until analysis complete	could	3	in progress	As a **user,** I want to be able to **see a progress bar** so that **I know how far along analysis is**
Identify the motion	must	3	yet to start	As a **user,** I want to be able to **analyse each file** so that **I can see if the child could perform the motion**
display the motion performed in each file	must	3	yet to start	As a **user,** I want to be able to **see the results**, so that **I can see what happened in each file**
Be able to count the iterations of the motion	must	4	yet to start	As a **user,** I want to be able to **analyse each file** so that **I can see if the child could perform the motion**
display the motion and number of iterations per file	must	4	yet to start	As a **user,** I want to be able to **see the results**, so that **I can see what happened in each file**
allow for removal of uploaded files	could	4	yet to start	As a **user,** I want to be able to **upload files** so that **they can be analysed**
allow for uploading compressed files to be decompressed before analysis	could	5	out of scope	As a **user,** I want to be able to **upload compressed files** so that **I do not have to decompress them myself**
allow the user to select a different format to export the results as	could	5	out of scope	As a **user,** I want to be able to **export in different formats** so that I can analyse them in different ways

3. Project Approach

(Provide a high-level description of the project approach)

For this project, the most important part is the machine learning models, we decided to use TensorFlow as it contains a variety of machine learning language libraries, and so it made sense to choose it as the core technology of the program we created. Therefore, we also chose python as the programming language as it is easy to use with TensorFlow as they are very compatible. When beginning our project we decided it would be best to divide the development process into two parts; the front end, which will be a GUI the our client can use to upload files and export results, and the back end which will parse and process the files uploaded using the machine learning models we will produce. We divide our team into two, three in each to work on the respective project sections.

To develop our front end, we chose to use PyQt5 libraries to create a user interface. We chose PyQt5 as a tool as it is the most popular powerful cross-platform GUI library which fits in with what we are looking to develop. Our intended functionality of the UI is to allow the user to upload files via file select window or simple drag and drop actions. Our client stores his files in. json format and so we also want to make sure the UI has the ability to check and verify that the files being uploaded are of. json type and throw an error message if they do not match this type. Once the client clicks analyse, the files will be passed to the machine learning back end for processing. After the program runs and all the files uploaded have been processed, the results will appear on the UI in a text display screen. Our client will then have the opportunity to export the results as a CSV file if he would like to save the results.

For the development of the back end of our application, we designed and created a machine learning model using TensorFlow. The purpose of the model is to identify exactly what the motion the motion capture file has recorded in each input file. Our client provided us with the test data he has collected and we used this data to train our machine learning models to be able to identify which of the possible six motor-skills it represents, When we have both front-and and back-end functioning, we will integrate the two into a single system and make sure they work together. We will then work on refining the application by working on improving our machine learning models accuracy and adding functionality to the UI that will add to the user experience for example having a progress bar displayed.

3.1. Scrum Sprints

(Provide diagrams for the scrum sprints for the projects and outline the rationale for the duration of the sprints.

For each scrum sprint you should clearly indicate the overall objectives of the sprint, the start and end dates of the sprint, and provide dates for the following meetings: sprint planning; scrum; sprint review; sprint retrospective; and backlog refinement.)

We set out a template we would all follow when taking our respective turns as scrum master to ensure consistency in our scrum sprint documentation. Some sprints were longer than others due to what had to be completed. We chose not to include the scrum planning, review, retrospective and backlog refinement as scrum dates.

Each of the scrums have a one-day overlap as we used the last day of the previous scrum as the first day of the next scrum as this is the day we had our weekly scrum meetings and it made sense to review the scrum and plan the next one taking into account what we learned from the previous scrum review and retrospective

3.1.1. Week 1

For the first week, we did not have a project or client and so there was no need for a sprint, the primary objective of this week was to set up a good foundation for our team. We used this week to:

- assign team roles
- make our project preference list
- set up GitHub for version control
- make our first git commits to make sure everyone had access
- set up a google drive
- set up a Facebook group chat
- set up a GitHub projects folder

3.1.2. Sprint 1- completed

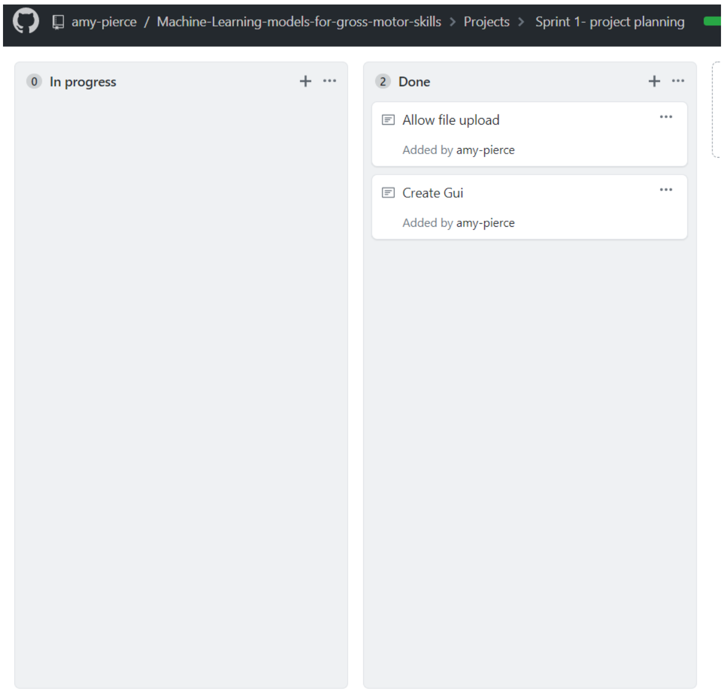
Our second sprint was focused on planning and preparation for the project. During this sprint we

- met with our client
- decided on what technologies and languages to use
- took time to learn the skills required
- completed our requirements document and presentation
- developed and designed our proposed systems using mock-ups and UML diagrams
- Begin User interface design

Stage	Project Planning

Objective	understand what is required by the client and plan how we will deliver on it
Start date	January 27th
End date	February 14th
Duration	14 days
Key Dates	**Date:**
Sprint Planning	January 27th
Scrums	● January 31st ● February 7th
Sprint Review	February 11th
Sprint retrospective	February 14th
Backlog refinement	February 14th

Sprint backlog:



amy-pierce / Machine-Learning-models-for-gross-motor-skills > Projects > Sprint 1- project planning

0 In progress

2 Done

Allow file upload
Added by amy-pierce

Create Gui
Added by amy-pierce

3.1.3. Sprint 2-completed

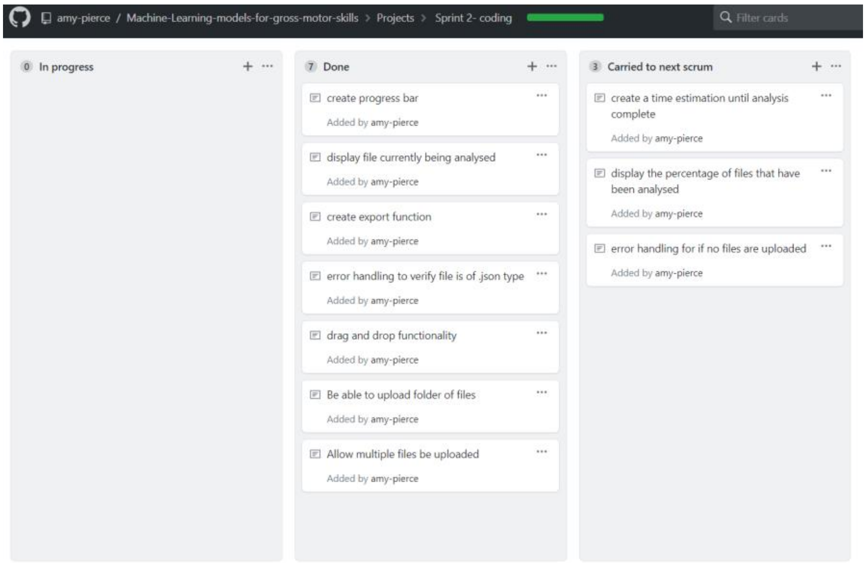
Our 3rd sprint was focused on developing the software. We kept the front-end and back-end development separate for this sprint and had two different teams working on the two different sections. During this sprint we worked on

- Developing a functioning user interface with uploading and export functionality
- Parsing the data into a form that could be passed to our machine learning models
- Training the machine models with test data the client provided

- Testing our machine learning models

Stage	Coding
Objective	being the software development phase getting a solid base foundation system in place.
Start date	February 14th
End date	March 9th
Duration	24 days
Key Dates	**Date:**
Sprint Planning	February 14th
Scrum	• February 21st • February 28th
Sprint Review	February 28th
Sprint retrospective	March 9th
Backlog refinement	March 9th

Sprint backlog:



amy-pierce / Machine-Learning-models-for-gross-motor-skills > Projects > Sprint 2- coding

0 In progress

7 Done

- create progress bar
  Added by amy-pierce
- display file currently being analysed
  Added by amy-pierce
- create export function
  Added by amy-pierce
- error handling to verify file is of .json type
  Added by amy-pierce
- drag and drop functionality
  Added by amy-pierce
- Be able to upload folder of files
  Added by amy-pierce
- Allow multiple files be uploaded
  Added by amy-pierce

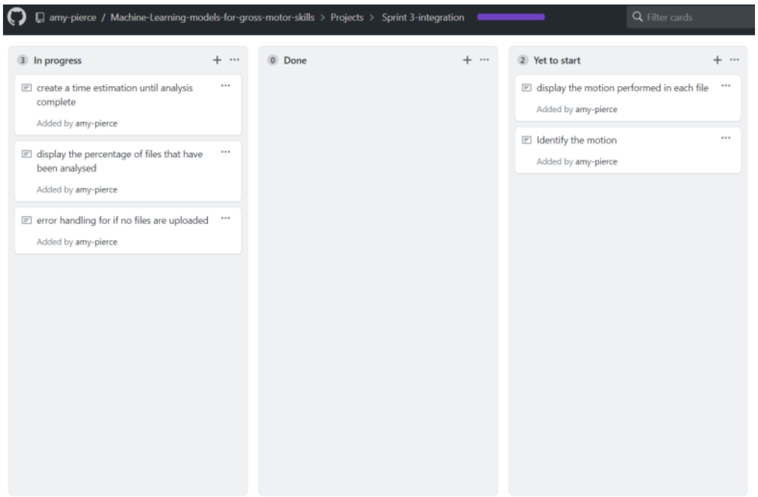
3 Carried to next scrum

- create a time estimation until analysis complete
  Added by amy-pierce
- display the percentage of files that have been analysed
  Added by amy-pierce
- error handling for if no files are uploaded
  Added by amy-pierce

3.1.4. Sprint 3- current

Stage	Integration
Objective	connect up the front-end and back-end of our system
Start date	March 9th
End date	March 20th
Duration	12 days
Key Dates	**Date:**
Sprint Planning	March 9th
Scrum	● March 13th
Sprint Review	March 20th
Sprint retrospective	March 20th
Backlog refinement	March 20th

Sprint backlog:



amy-pierce / Machine-Learning-models-for-gross-motor-skills > Projects > Sprint 3-integration

3 In progress

create a time estimation until analysis complete
Added by amy-pierce

display the percentage of files that have been analysed
Added by amy-pierce

error handling for if no files are uploaded
Added by amy-pierce

0 Done

2 Yet to start

display the motion performed in each file
Added by amy-pierce

Identify the motion
Added by amy-pierce

Sprint 4 will be used to work on the integration of the front end and the backend of our application. We plan to use this sprint to:

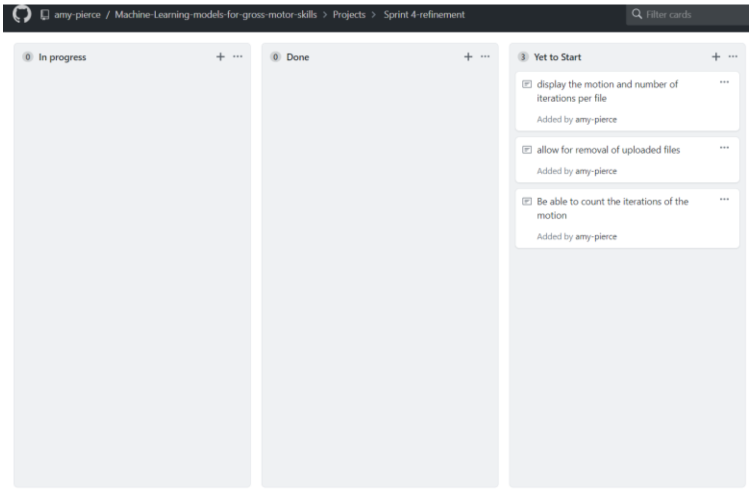
- Connect the machine learning models to the user interface
- test that a user can upload files to the user interface and have the results displayed at
- Complete the project plan and software design specification

3.1.5. Sprint 4- planned

The focus of this spring is to carry out testing on our application. We will use this spring to:

Stage	Refinement
Objective	add any finished features to our application before testing
Start date	March 20th
End date	April 3rd
Duration	14 days
Key Dates	**Date:**
Sprint Planning	March 20th
Scrum	● March 27th
Sprint Review	April 3rd
Sprint retrospective	April 3rd
Backlog refinement	April 3rd

Sprint backlog:


amy-pierce / Machine-Learning-models-for-gross-motor-skills > Projects > Sprint 4-refinement
Filter cards
0 In progress
0 Done
3 Yet to Start

display the motion and number of iterations per file

Added by amy-pierce

allow for removal of uploaded files

Added by amy-pierce

Be able to count the iterations of the motion

Added by amy-pierce

3.1.6. Sprint 5-planned

This final sprint is intended to be used for the final testing of the application and fixing any bugs that may be found. We will use this sprint to:

- Run tests on the file upload functionality

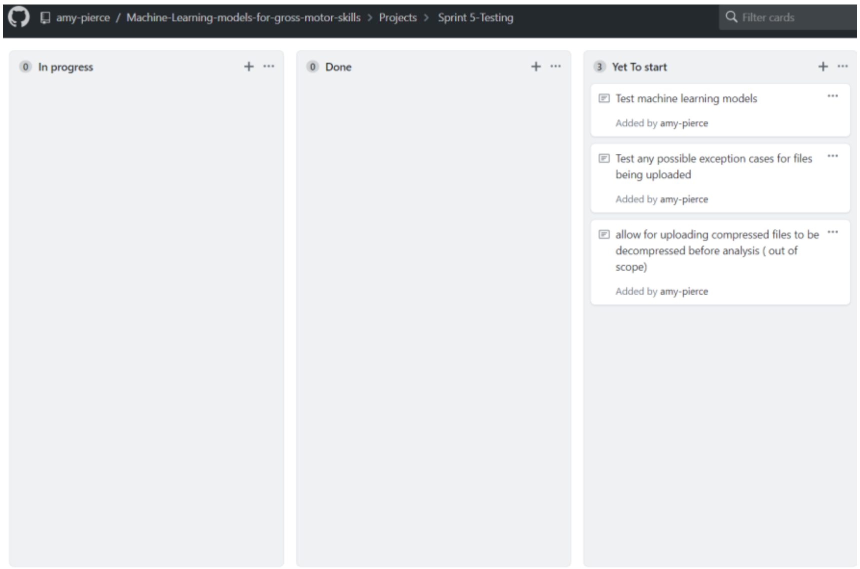
- Run tests on the accuracy of the machine learning models
- Run test on the file exporting functionality
- Run tests on the smooth integration between the front and back end
- Run test on the functionality of the progress bar

If we have the capabilities to look at adding one the features outside of the scope of this project we may do so, however this is unlikely. The out of scope features we will look to tackle is

- adjusting our application to make it so that it can decompress files that have been uploaded in a compressed form
- make it so the user can choose what file format to export the results as

Stage	Testing
Objective	test the product and resolve any errors/bugs found
Start date	April 3rd
End date	April 17th
Duration	14 days
Key Dates	**Date:**
Sprint Planning	April 3rd
Scrum	● April 10th
Sprint Review	April 10th
Sprint retrospective	April 17th
Backlog refinement	April 17th

Sprint backlog:


amy-pierce / Machine-Learning-models-for-gross-motor-skills > Projects > Sprint 5-Testing
Filter cards
0 In progress
0 Done
3 Yet To start

Test machine learning models
Added by amy-pierce

Test any possible exception cases for files being uploaded
Added by amy-pierce

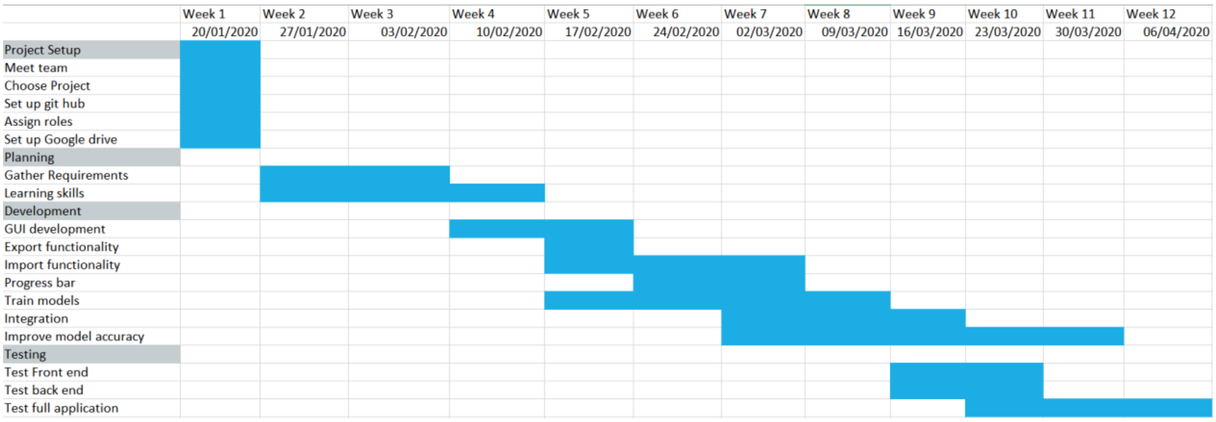
allow for uploading compressed files to be decompressed before analysis (out of scope)
Added by amy-pierce

3.2. Milestones

Milestones	Date
Begin Coding	February 14th
Functioning user interface	February 28th
Trained machine learning models	March 9th
Integrated front and back end	March 20th
Fully functioning application	April 10th

3.3. Gantt Chart and Work Breakdown Structure

The following is a breakdown of our intended work flow structure. We have set this out so that we have rough guidelines to follow in terms of getting tests done. We created this Gantt chart at the beginning of the project but have since decided that the product backlog is a better way to monitor our work and set goals for ourselves.



	Week 1	Week 2	Week 3	Week 4	Week 5	Week 6	Week 7	Week 8	Week 9	Week 10	Week 11	Week 12
	20/01/2020	27/01/2020	03/02/2020	10/02/2020	17/02/2020	24/02/2020	02/03/2020	09/03/2020	16/03/2020	23/03/2020	30/03/2020	06/04/2020
Project Setup	■											
Meet team	■											
Choose Project	■											
Set up git hub	■											
Assign roles	■											
Set up Google drive	■											
Planning												
Gather Requirements		■	■									
Learning skills		■	■	■								
Development												
GUI development				■	■							
Export functionality					■							
Import functionality					■	■	■					
Progress bar						■	■					
Train models					■	■	■	■				
Integration							■	■	■			
Improve model accuracy							■	■	■	■	■	
Testing												
Test Front end									■	■		
Test back end									■	■		
Test full application										■	■	■

4. Project Organisation

4.1. Staff

(Provide a list of the team members and their prior experience in projects and prior technical skills)

Team Member	Prior experience	Prior technical skills
Amy Pierce	● Created a one-time-two-factor	● Java ● C

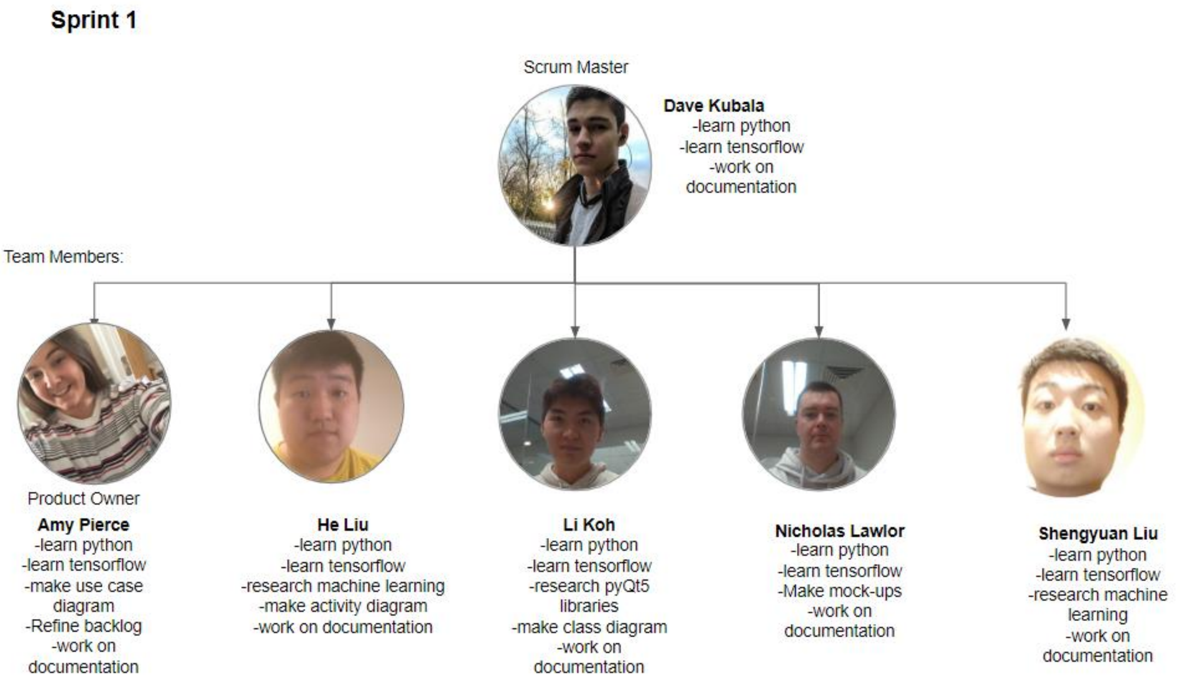
	authentication login system ● Created a blog and FAQ page for a start up	● JavaScript ● HTML & CSS ● React.js ● UI design
Nicholas Lawlor	● Created a website for the intention of use in the hotel industry	● Java ● Python
He Liu	● Built a chrome extension for alerting users of earthquakes nearby	● Python ● Java ● Chrome Extension ● Java script
Li Koh	● Open flow protocol project to build a web browser	● Python ● PyQt5 ● React ● UI design ● Java
Dave Kubala	● 3D animation for competition (2nd place)	● Unity ● C# ● Java ● iPi Mcap Studio
Shengyuan Liu	● Built a web-based application for creating Data Flow Diagrams.	● React ● Java script ● C ● Java ● Python

4.2. Staff Chart

(Provide a scrum staff chart outlining the following people for each scrum sprint in the project: Product owner; Scrum master; team members.

For each sprint staff chart outline what each member of the team will be working on.)

4.2.1. Sprint 1



Sprint 1

Scrum Master

Dave Kubala
-learn python
-learn tensorflow
-work on documentation

Team Members:

Product Owner

Amy Pierce
-learn python
-learn tensorflow
-make use case diagram
-Refine backlog
-work on documentation

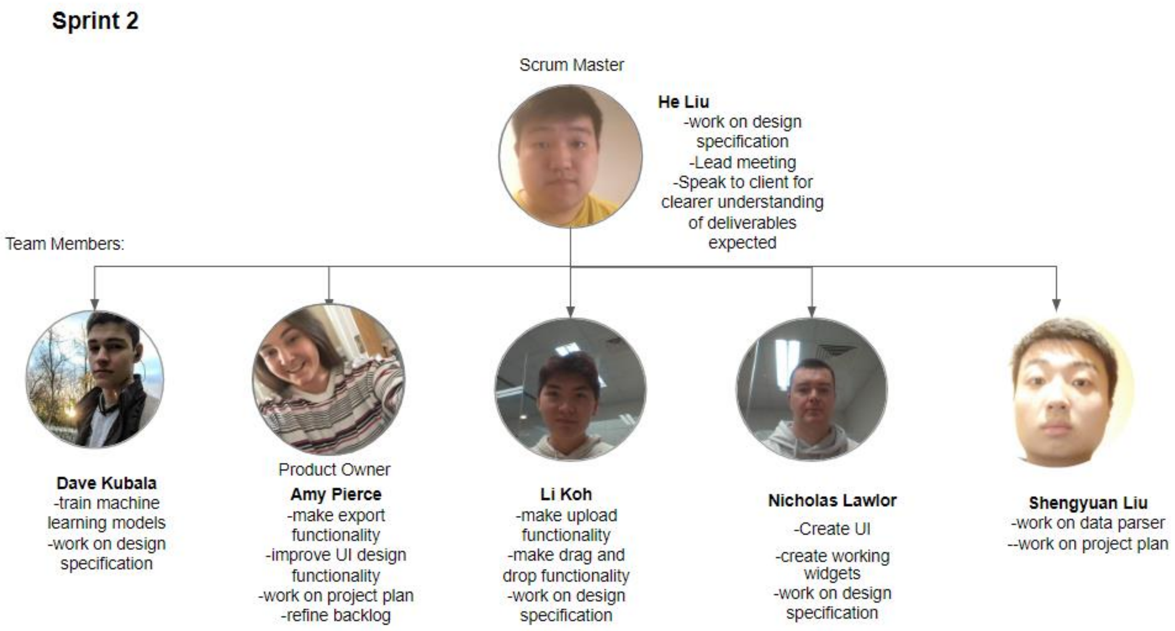
He Liu
-learn python
-learn tensorflow
-research machine learning
-make activity diagram
-work on documentation

Li Koh
-learn python
-learn tensorflow
-research pyQt5 libraries
-make class diagram
-work on documentation

Nicholas Lawlor
-learn python
-learn tensorflow
-Make mock-ups
-work on documentation

Shengyuan Liu
-learn python
-learn tensorflow
-research machine learning
-work on documentation

4.2.2. Sprint 2



Sprint 2

Scrum Master

He Liu
-work on design specification
-Lead meeting
-Speak to client for clearer understanding of deliverables expected

Team Members:

Dave Kubala
-train machine learning models
-work on design specification

Product Owner

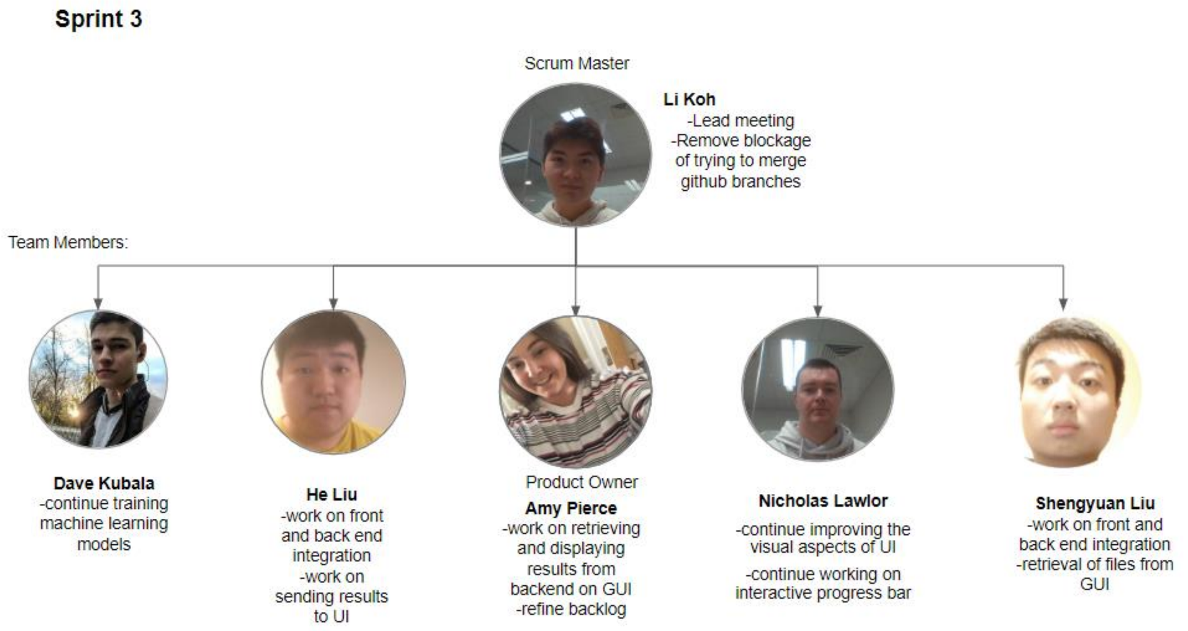
Amy Pierce
-make export functionality
-improve UI design functionality
-work on project plan
-refine backlog

Li Koh
-make upload functionality
-make drag and drop functionality
-work on design specification

Nicholas Lawlor
-Create UI
-create working widgets
-work on design specification

Shengyuan Liu
-work on data parser
--work on project plan

4.2.3. Sprint 3



Sprint 3

Scrum Master

Li Koh
-Lead meeting
-Remove blockage of trying to merge github branches

Team Members:

Dave Kubala
-continue training machine learning models

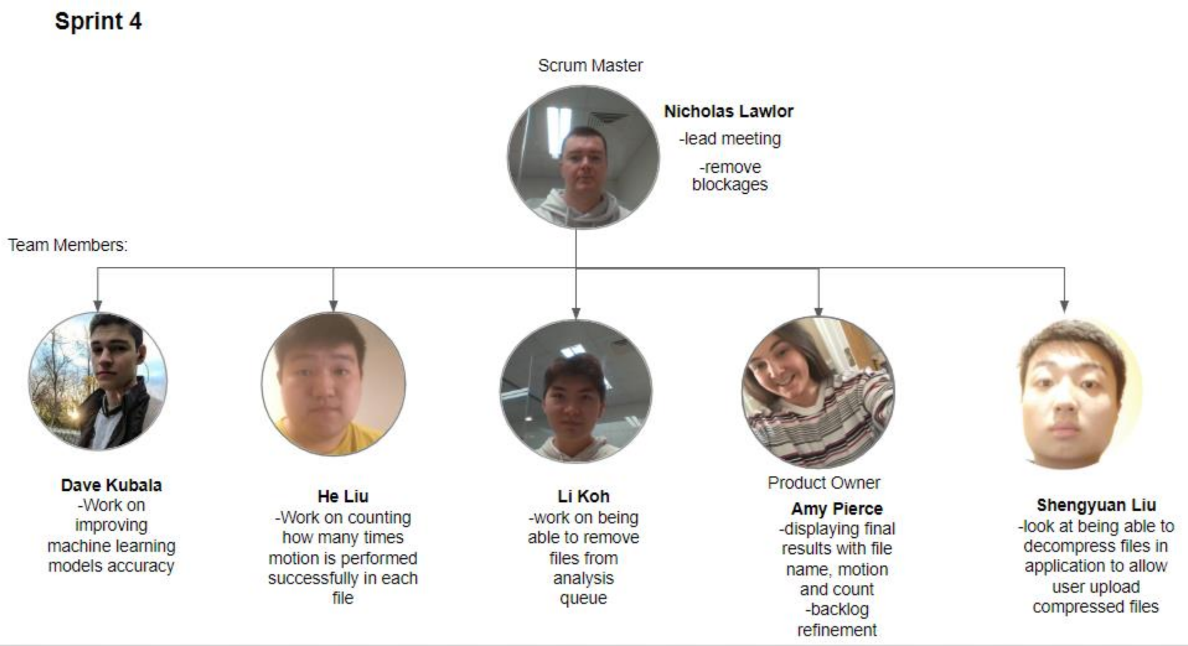
He Liu
-work on front and back end integration
-work on sending results to UI

Product Owner
Amy Pierce
-work on retrieving and displaying results from backend on GUI
-refine backlog

Nicholas Lawlor
-continue improving the visual aspects of UI
-continue working on interactive progress bar

Shengyuan Liu
-work on front and back end integration
-retrieval of files from GUI

4.2.4. Sprint 4



Sprint 4

Scrum Master

Nicholas Lawlor
-lead meeting
-remove blockages

Team Members:

Dave Kubala
-Work on improving machine learning models accuracy

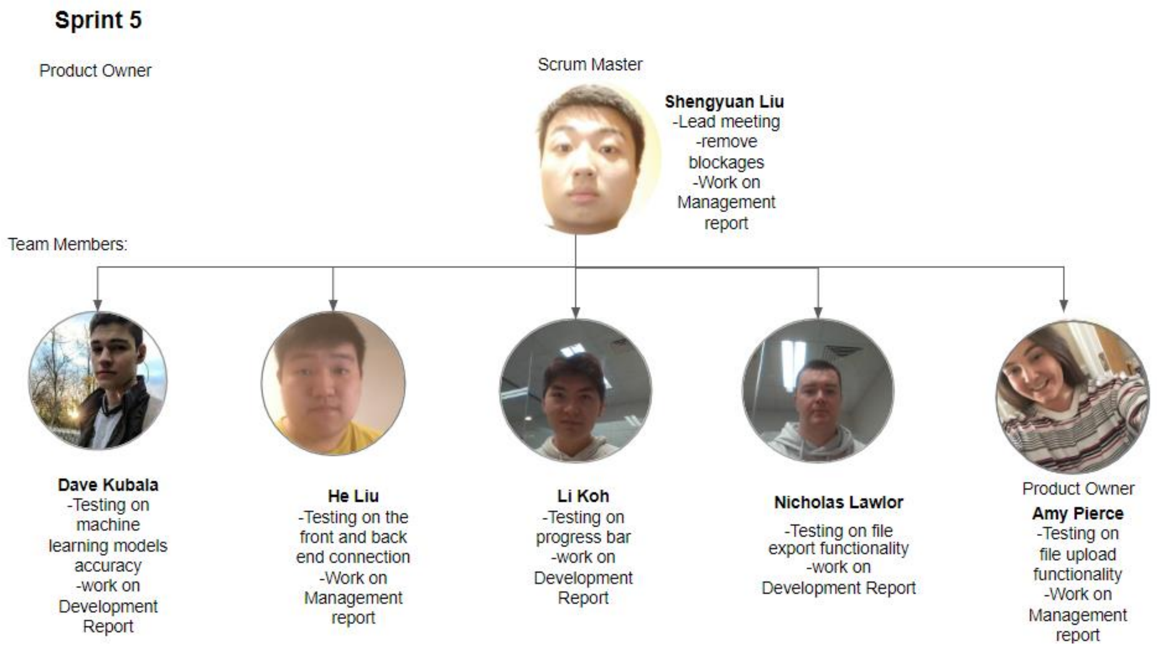
He Liu
-Work on counting how many times motion is performed successfully in each file

Li Koh
-work on being able to remove files from analysis queue

Product Owner
Amy Pierce
-displaying final results with file name, motion and count
-backlog refinement

Shengyuan Liu
-look at being able to decompress files in application to allow user upload compressed files

4.2.5. Sprint 5


Sprint 5
Product Owner
Scrum Master
Shengyuan Liu
-Lead meeting
-remove blockages
-Work on Management report
Team Members:
Dave Kubala
-Testing on machine learning models accuracy
-work on Development Report
He Liu
-Testing on the front and back end connection
-Work on Management report
Li Koh
-Testing on progress bar
-work on Development Report
Nicholas Lawlor
-Testing on file export functionality
-work on Development Report
Product Owner
Amy Pierce
-Testing on file upload functionality
-Work on Management report

5. Risk Analysis

5.1. Risk Analysis

(Identify the risks to the successful conclusion of the project and calculate their Risk Factor)

Risk Element	**Impact (1 to 5)**	**Likelihood (1 to 5)**	**Risk Factor (I*L)**
Team member illness	1	3	3
Lack of knowledge	3	3	9
Clashing time tables	2	3	6
Last minute change of core design	5	1	5
Slow progress	4	2	8

5.2. Risk Mitigation

(Present the risks (ordered by their risk factor) and identify actions to reduce them)

Risk	**Measures to Reduce Risk**
Team member illness	If a team member falls ill, they will make the team leader and current scrum master aware as soon as possible. Their work will be divided equally among the rest of the team members.

Lack of knowledge	Take enough time to develop and learn about the technologies and languages required to complete the project before beginning development.
Clashing time tables	We will use the second on Tuesdays that is scheduled for the module to have our weekly scrum meetings as we all have that time slot free. Working around time tables we have found other available time slots that we have agreed to meet during.
Last minute change of core design	By keeping in constant communication with our client we can reduce the risk of last-minute changes being requested. By checking in with the client regularly we can make sure we are working towards the correct end goal. Planning, documentation and structure will help eliminate this risk.
Slow progress	By planning properly and following scrum plans with daily 15-minute sprints, we can eliminate this risk by monitoring our progress and making sure we are on schedule and.

6. Project Controls

(Outline the mechanisms and tools used to control all aspects of the project execution, progress, and quality, deliverables, deadlines and communication.)

- **Execution:**
  - **Python**

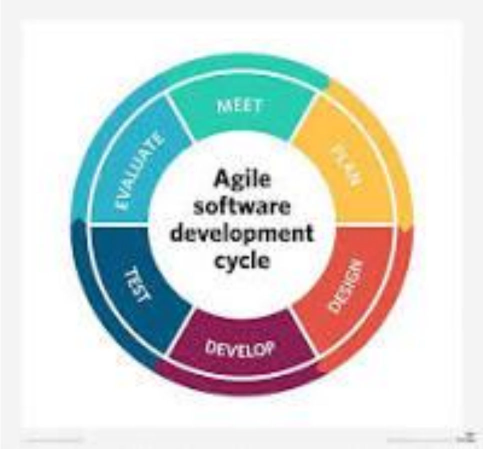
We use Python as our basic program development language. Because it has a perfect performance in the machine learning engine. We choose to use it as it has many TensorFlow libraries which is the core of our machine learning model and PyQt5 which can let us to build a beautiful and easy to operate UI. Another important reason for choosing it is that it is easy to use, and for those of us who have some development experience, it doesn't take much extra time to learn and we can focus on our code.

  - **TensorFlow**

We choose TensorFlow as our machine learning model core as it has a lot of existing models in itself. And some of them are specially used to learn and detect human’s motion. At the same time, it can be written in python code. Our code is reprocessed based on these models, so that the program can analyse and learn specific by our input, it will be more sensitive to the data we input for the six motions and finally build the machine learning model which we want. Also, the TensorFlow is an open source platform so we can offer it.

  - **Agile Method**

We are following the agile method to execute our project development. This allows us to constantly develop and test smaller aspects of the project as we go. We chose the agile method as it allows us to tackle smaller problems individually and constantly test and develop aspects which means we can find and solve issues as we experience them early on in the development rather than finishing development and discovering issues that will be difficult to resolve if we were to use the waterfall method.



- **Progress:**
  - GitHub

Our GitHub allows us to use version control to monitor our progress in terms of who is making commits and adding to the project.

  - Meetings

Our weekly scrum meetings allow us to track and monitor the progress we make each week. We use these meetings to discuss any problems that may be blocking us and to check in with everyone to see what progress they have made. The weekly meetings with our demonstrator also allow him to monitor our progress and make sure we are on track.

- ○ Product backlog

We are using a product backlog to keep track of what tasks need to be completed, have been completed and are currently in progress. This allows everyone to see what stage we are at for each of the tasks relating to the project

- **Quality:**
  - ○ Pair Programming

In order to maintain a high quality of code we have decided to use pair programming, each third year will program with a second year. This means that all code written will be read and examined by another team member before being committed to GitHub to ensure it is of a high quality. By verifying the quality of the code before commuting to GitHub, it will ensure high quality coding standards are constipating throughout all the program files.

- Coding standard

In order to help the everyone to produce code of an acceptable quality, we decided to set out a coding standard for everyone to follow. This will ensure code of a high standard is produced and will make it easier to integrate the different aspects when merging on GitHub. The standard we set out is as follows:

- All variables must be in lower camel case
- Comments must be included next to any code not obviously understandable
- Indentation must be used
- All variable and function names must be easy to understand

- **Deliverables:**
  - ○ Google drive

We will use Google drive to keep all of our documents in as we can all access it and make changes to the documentation as we complete it. The client can easily be granted access to the google drive and this is how they

- GitHub

GitHub is where we are keeping all of our code for this project. We will be able to keep all the coding packages we plan to deliver to the client in a folder on our GitHub. We will also have a folder in our GitHub repository to store all our documentation in.

- **Deadlines:**
  - Messenger

Our Facebook group chat is very useful to remind each other of deadlines coming up. We set events in the chat to alert us the day a submission is due. It is a simple and effective method for keeping on track of our deadlines

- Meetings

Our weekly meetings are vital to staying on top of deadlines. We use these meetings to discuss and work through documentation together. During each meeting we take note of what submission is up next and when. The make sure we don't forget to submit the weekly scrum videos; we submit them together immediately after taking them

- Trello

We have a trello set up that we use for our documentation and submission deadlines. By having a list of things that need to be complete it will help us keep on top of what aspects of the project are due at what time which is vital to keep on top of the various deadlines we have.

- Gantt Chart

We made a Gantt chart at the beginning of the project to outline the milestones that we need to keep on track of in order to keep up with deadlines. We tend to use the product backlog more but it was a good guide to start off with when we did not know what a product backlog was in the first two weeks of the project.

- **Communication:**
  - Messenger

Our main means of communication was a Facebook group chat as it was easy and formal and as everyone usually had their phone or laptop with them was an easy way to contact all member in the group

- GitHub

Git hub was a useful way to communicate our code with one another, it allowed us to add comments to the code we uploaded and was an effective way to share the work we had completed when we were not in a meeting or in college.

- GitHub Projects

GitHub Projects allowed us to communicate what jobs and tasks needed to be done or had been completed. We could add and remove tasks as we pleased and move them into the appropriate stage of completion: in progress, done or yet to start. This allowed us to keep up communications regarding what needs to be done

- Google Drive

Google drive was used extensively to complete our documents, it allowed us to communicate with each other and leave comments on the documentation we were working on. It was very useful to be able to all work on the same document together without being in the same place, and we could also open up chat windows as we did so to further our communication

7. Communication

7.1. Client Communication

(Outline the schedule of meetings with the client and any requested reports)

Meetings	**Date**	**Purpose**
Client Meeting	29/01/2020	Whole team met the client to introduce themselves and get a brief description of the project and various test data.
Requirements Meeting	07/02/2020	Check on progress Ensure clients' needs are being met. And see if anything the client wants to change. Get more various data.
Midway Point Meeting	TBC	Our team will meet the client to demonstrate program development progress and answer customer questions about the program

Client Demo Meeting	TBC	After finishing the whole program, our team will meet the client to demo program working results and get the handover or proof from the client.
Final Handover Meeting	TBC	We will meet with the client to hand over all the deliverables and get proof of handover.
Document	**Date Due**	**Description**
Requirements Document	15/02/2020	A detailed description of the requirements and scope of the project
Software Design Specification	09/03/2020	A document detailing an in-depth evaluation of the design approaches and decisions made on how the application will function
Development Report	17/04/2020	A detailed report of the development choices made throughout the project

*Our client requested only certain documents that pertain to the development choices of our project and so we do not need to deliver to him the Project Plan or Management Report, if at a later stage they are requested they will be provided.

7.2. Project Team Meetings

(Outline the schedule of team meetings and any requested reports)

Meetings	**Date**	**Purpose**
Week 1	22/01/2020	Introductions and decisions made as to project preferences, assign group roles.
Week 2	28/01/2020	Meet the demonstrator for the first time, discuss the assigned project and planned client meeting.
Week 2	31/01/2020	weekly scrum meeting; debrief after client meeting, discuss what languages and

		technologies to use. Take the group meeting video.
Week 3	04/02/2020	weekly meeting with demonstrator; discuss progress
Week 3	07/02/2020	weekly scrum meeting; Discuss possible technologies to use and divide the team into front end and back end groups. Start learning new technologies and languages. Take the group meeting video.
Week 4	10/02/2020	discuss requirements presentation and document
Week 4	11/02/2020	debrief after presentation; discuss what is left to do in requirements document and progress check on software development
Week 4	14/02/2020	weekly scrum meeting; Last discussion on requirements document before submission, discuss the next stage for development and starting coding. Take the group meeting video.
Week 5	18/02/2020	Discuss approach to development of the application and assign roles. Take the group meeting video. meet demonstrator
Week 5	21/02/2020	weekly scrum meeting: Coding session, work on linking the front and backend up. Take the group meeting video.
Week 6	25/02/2020	Check the whole program processing and discuss the task during the study week. Take the group meeting video. Meet demonstrator
Week 6	28/02/2020	weekly scrum meeting; divide up coding tasks to be done

		over the break and discuss progress made, divide up documentation work. Take the group meeting video.
Week 7	no meeting	no meeting due to reading week
Future Meetings	**Proposed Date**	**Proposed Purpose**
Week 8	09/03/2020	To discuss project plan and software design specification documents before submitting and debrief after reading week.
Week 8	10/03/2020	Debrief after meeting with Macu, discuss overall progress and timeline projections.
Week 8	13/03/2020	Weekly scrum; merge all coding elements together, getting ready for moving into the final development phase. Take the group meeting video.
Week 9	17/03/2020	weekly meeting with demonstrator
Week 9	20/03/2020	Weekly scrum; progress check, preparation for live demo. Take the group meeting video.
Week 10	24/03/2020	weekly meeting with demonstrator
Week 10	27/03/2020	Weekly scrum meeting; Take the group meeting video.
Week 11	31/03/2020	weekly meeting with demonstrator
Week 11	03/04/2020	Weekly scrum meeting; Take the group meeting video.
Week 12	07/04/2020	weekly meeting with demonstrator
Week 12	10/04/2020	Weekly scrum; final preparations for hand over and final documents submissions Take the group meeting video.

Documents	**Date Due**	**Description**
Project Selection	32/01/2020	Submit project preferences
Requirements Presentation	10/02/2020	PowerPoint including overview, objectives, functional & non-functional requirements, diagrams etc to be presented in class
Signed-Off Requirements Document	14/02/2020	Document including overview, objectives, functional & non-functional requirements, diagrams etc
Project Plan	09/03/2020	A plan of our project and an idea of how we will implement the given project.
Software Design Specification Document	09/03/2020	In-depth evaluation of the design approaches and decisions made on how the application will function
Final Presentation	23/03/2020 **or** 30/03/2020 -TBC	PowerPoint detailing our final product to be presented in class with a live demo
Project Code Bundle	17/04/2020	A folder/ Bundle of all the code for our final deliverable product.
Development Plan	17/04/2020	A document completed by the 2nd years detailing the development choices made throughout the project
Management Report	17/04/2020	A document completed by the 3rd years detailing the management decisions made throughout the project
Individual Reflective Essays	17/04/2020	Each person individual Essay to document their experience with the project and their contributions to the group.
Client Handover Proof	17/04/2020	Proof from client that they have received the deliverables

8. Appendices

8.1 Appendix 1- Requirements Document

CS2013: Outline for a Requirements Document

CONTENTS

1. Introduction
    1.1. Overview – Purpose of system
    1.2. Scope
    1.3. Objectives and success criteria
    1.4. Definitions, abbreviations
    1.5. References

2. Current system

3. Proposed System
    3.1. Overview
    3.2. Functional Requirements
    3.3. Non-functional requirements
    3.4. System prototype (models)
        3.4.1. User interface mock-ups
        3.4.2. Use cases (including text narratives)
        3.4.3. Object model
        3.4.4. Dynamic model

4. Client Sign-Off

1. Introduction

1.1. Overview – Purpose of system

The purpose of this group project for CS2013/CS3013 is to design and create machine learning models for tracking gross-motor skills for our client in the SCSS department. Our client is doing research on the development of gross-motor skills in young children and has asked that we create a machine learning model to help track certain movements to be able to determine whether the children can perform the motor-skills, and hence the overall purpose of our project is to create machine learning models that can recognize such motor-skills.

This project will help our client spend more time collecting data for his research rather than spending time developing the machine learning aspects himself.

1.2. Scope

The scope of this project consists of developing an efficient and accurate machine learning model that can identify what movement is being performed when a json file containing motion capture data is passed through the model.

The machine learning model will be presented with a motion capture data file and will determine which motor-skill it is relating to. It will look to determine how many times the skill was performed for each motion capture data file. In order for a motion to be considered 'successful' the machine learning model determines if it was performed sufficiently, i.e. if the motor-skill being checked for is a hop and the child jumped off one foot but landed on two, this is not considered a successful iteration of the motor-skill.

1.3. Objectives and success criteria

Our objective is to fully deliver the items described in the scope as well as any additional time permitting functions. Our success criteria include a fully functional and tested machine learning model that can be integrated and used on the clients existing system to compare the results he has previously gotten. Our main objective is to produce something useful for the client that actually works and will benefit his research within the given time-frame.

Our client has data collected for six different motions, however due to the limited time frame they have acknowledged that making six machine learning models may be quite a feat. To ensure that we successfully deliver a system of high quality, we have been guided to focus primarily on 3 of the motions: jog, hop left and hop right. Once we have completed these models to a high and accurate standard, we may then have time to train additional models.

Our main objective is to deliver to our client machine learning models that process these three motor-skills. We want our system to be accurate and extensible, that is that we want our client to be able to easily train the model with new data as well as getting consistent and accurate results on test data.

1.4. Definitions, abbreviations

Deep learning

Deep learning is an artificial intelligence function that imitates the workings of the human brain in processing data and creating patterns for use in decision making. Deep learning is a subset of machine learning in artificial intelligence (AI) that has networks capable of learning unsupervised from data that is unstructured or unlabeled. Also known as deep neural learning or deep neural network.

Gross-Motor Skills

Gross motor (physical) skills are those which require whole body movement and which involve the large (core stabilising) muscles of the body to perform everyday functions, such as standing and walking, running and jumping, and sitting upright at the table.

Machine learning

Machine learning is an application of artificial intelligence (AI) that provides systems the ability to automatically learn and improve from experience without being explicitly programmed. Machine learning focuses on the development of computer programs that can access data and use it to learn for themselves.

The process of learning begins with observations or data, such as examples, direct experience, or instruction, in order to look for patterns in data and make better decisions in the future based on the examples that we provide. The primary aim is to allow the computers learn automatically without human intervention or assistance and adjust actions accordingly

Neural networks

Neural networks are a set of algorithms, modeled loosely after the human brain, that are designed to recognize patterns. They interpret sensory data through a kind of machine perception, labeling or clustering raw input. The patterns they recognize are numerical, contained in vectors, into which all real-world data, be it images, sound, text or time series, must be translated.

1.5. References

Deep Learning

https://www.investopedia.com/terms/d/deep-learning.asp

Gross-motor Skills

https://childdevelopment.com.au/areas-of-concern/gross-motor-skills/

Machine Learning

https://expertsystem.com/machine-learning-definition/

Neural Networks

https://pathmind.com/wiki/neural-network

2. Current system

Our client’s current system is focused primarily on recording motion capture data. The system's intended use is to record movement patterns on motor skills. The client is using a Microsoft XBOX Kinect to collect infrared motion capture data to track each individual joint of a test subject while they move. Each joint’s XYZ coordinate positional tracking data is compiled into a set of joints,or a ‘skeleton’.

The existing system being used in our client-s research to collect raw positional and rotational tracking data (3D position of each joint of the body over a period of time) from 40 test subjects who each performed a number of different tracked motions, e.g. hop on left and right foot, jumping side to side, jumping with both feet, and jumping forward.

There are no machine learning models that work concurrently with the system; producing these models are our tasks for this project. Using the motion capture data acquired from the current system, we will train our machine learning models to recognize each individual motion.

Before our involvement in this project, our client had developed a non-machine learning algorithm for analysing the data that has been gathered. To test our machine learning models to their fullest extent, we will compare the accuracy of our models to that of the results of our clients current system as we near the finalized models.

3. Proposed System

3.1. Overview

Our machine learning model will implement both a frontend and a backend which will allow for ease of use by the client. The frontend will implement a simple file selection system, which will allow the client to assign .json files to be read by the machine learning model. The backend will take in files selected in the selection system and parse the .json files into a readable data format. The system will have a pre-baked machine learning model that predicts motions at approximately 90% to 95% accuracy. After the .json files have been parsed, they will be sent into the pre-baked model, and will output a list of the motions that have been inputted to the frontend, followed by which motion they are most likely to represent. These results will then be displayed to a frontend panel, which will be capable of exporting the result into a variety of file formats (e.g. .XLS, .CSV, .XML, .JSON, etc.)

3.2. Functional Requirements

*A **Functional Requirement** (FR) is a description of the service that the software must offer. It describes a software system or its component. A function is nothing but inputs to the software system, its behavior, and outputs. It can be a calculation, data manipulation, business process, user interaction, or any other specific functionality which defines what function a system is likely to perform.https://www.guru99.com/functional-requirement-specification-example.html*

- The system will be able to process .json motion capture files uploaded by the user.
- The system will parse the files for the machine learning model to process.
- The machine learning model will be able to read the data parsed from the .json file and output the most likely motion that the file represents.
- The system will be able to accurately predict the iterations of each motion from the .json motion capture file (e.g. if the motion is determined to be a Hop-Left, the system will be able to predict how many times a Hop-Left was done in the motion capture.).
- The system should store the results in a file document.
- The machine learning model should have an accuracy of at least 80%.

3.3. Non-functional Requirements

*A **Non-Functional Requirement** (NFR) defines the quality attribute of a software system. They judge the software system based on Responsiveness, Usability, Security, Portability and other non-functional standards that are critical to success of the software system. Example of nonfunctional requirement, how fast does the website load? Failing to meet non-functional requirements can result in systems that fail to satisfy user needs.https://www.guru99.com/non-functional-requirement-type-example.html*

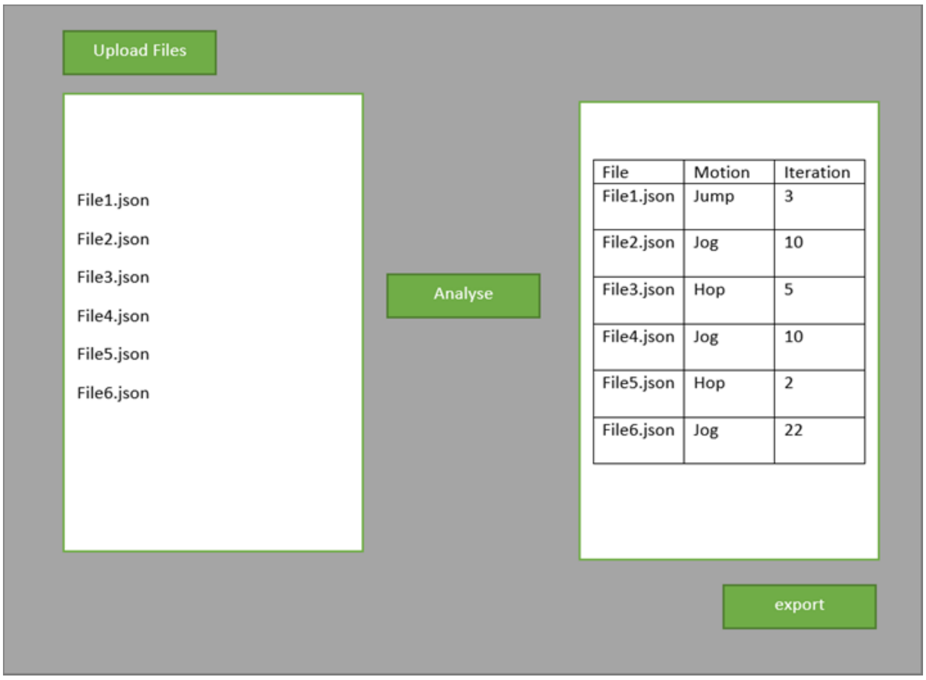
- The frontend will be able to load multiple motion capture files at once to be read by the machine learning model and be able to output multiple results.
- The front end GUI will be user friendly and easy to navigate.
- The frontend will be able to export results of the processed data into one of several file formats.

- The model should be able to be refined (e.g. swapped for a more refined model) using a series of supervised machine learning techniques.

3.4. System prototype (models)

3.4.1. User interface mockups

In our mockup phase, we have produced two different potential user interface design which would allow the client to easily use the system. A well designed GUI is important for this task; our client should be able to easily access and send the data to the system without much hassle. Our first is a simple table-to-table GUI. The user would simply append the files into the first window, click ‘analyze’, await analysis, and be able to view the results in tabulated form within minutes.



File1.json

File2.json

File3.json

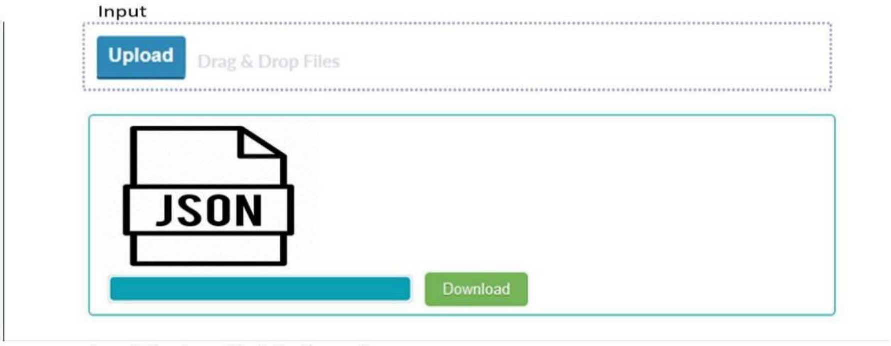
File4.json

File5.json

File6.json

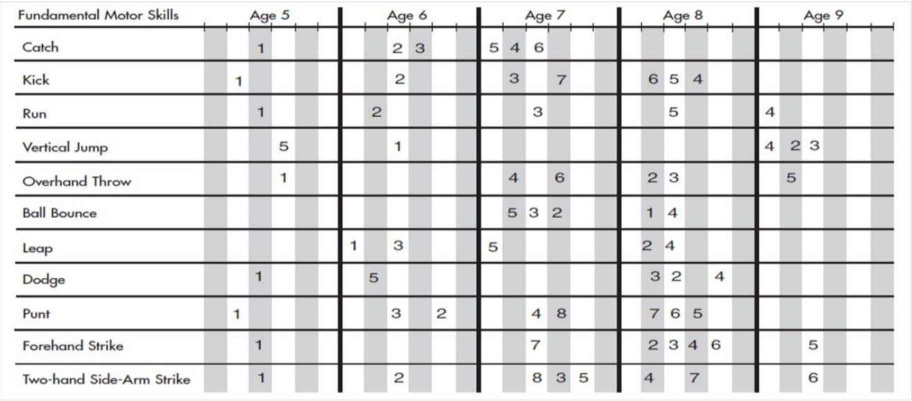
File	Motion	Iteration
File1.json	Jump	3
File2.json	Jog	10
File3.json	Hop	5
File4.json	Jog	10
File5.json	Hop	2
File6.json	Jog	22

Json file input



Input the json file into the software

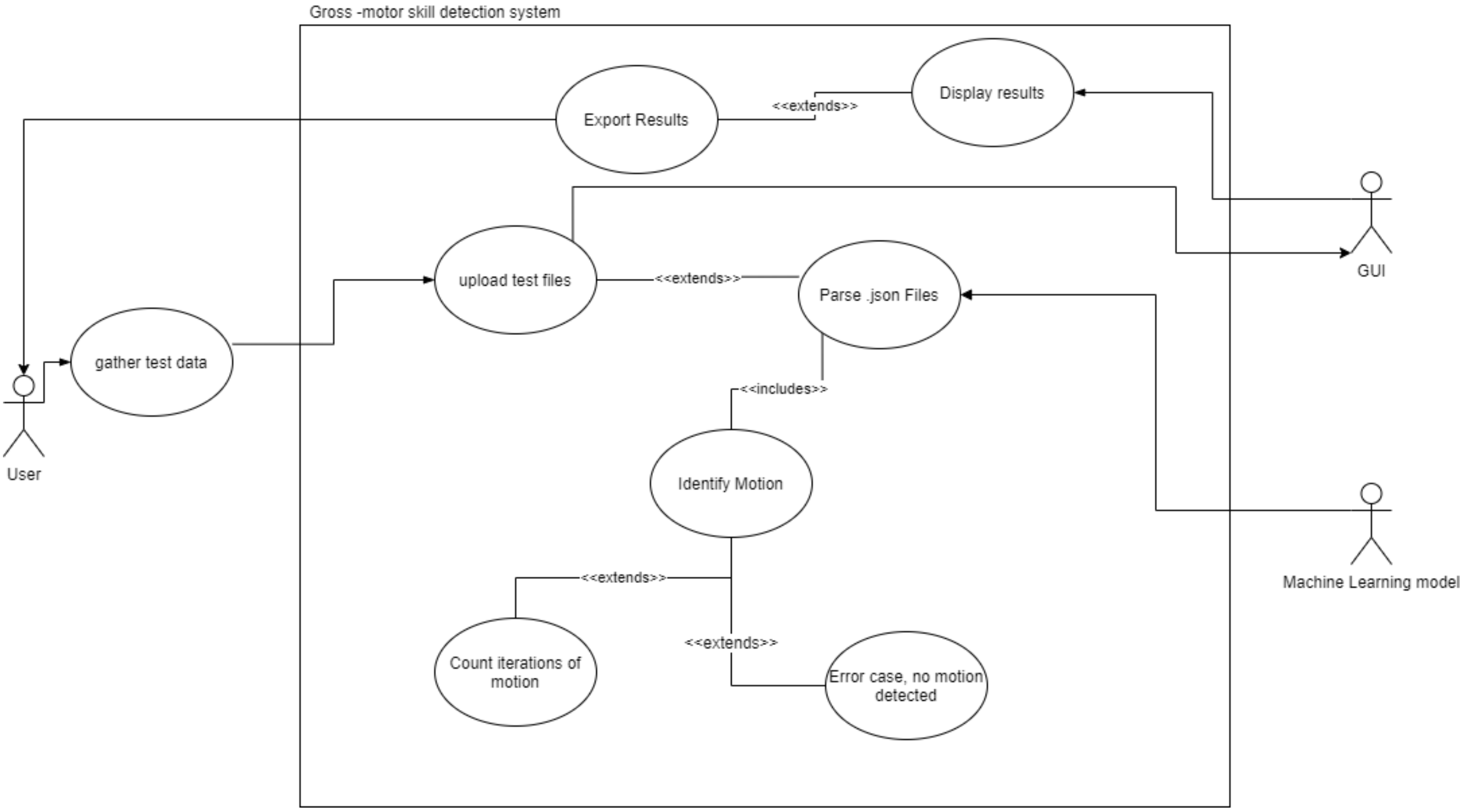
Output the result table:



Fundamental Motor Skills	Age 5						Age 6						Age 7						Age 8						Age 9					
Catch			1						2	3			5	4	6															
Kick		1							2					3		7				6	5	4								
Run			1					2							3						5				4					
Vertical Jump				5					1																4	2	3			
Overhand Throw				1										4		6				2	3					5				
Ball Bounce														5	3	2				1	4									
Leap							1		3				5							2	4									
Dodge			1					5												3	2		4							
Punt		1							3		2				4	8				7	6	5								
Forehand Strike			1												7					2	3	4	6				5			
Two-hand Side-Arm Strike			1						2						8	3	5			4		7					6			

3.4.2. Use cases (including text narratives)

Use Case Diagram



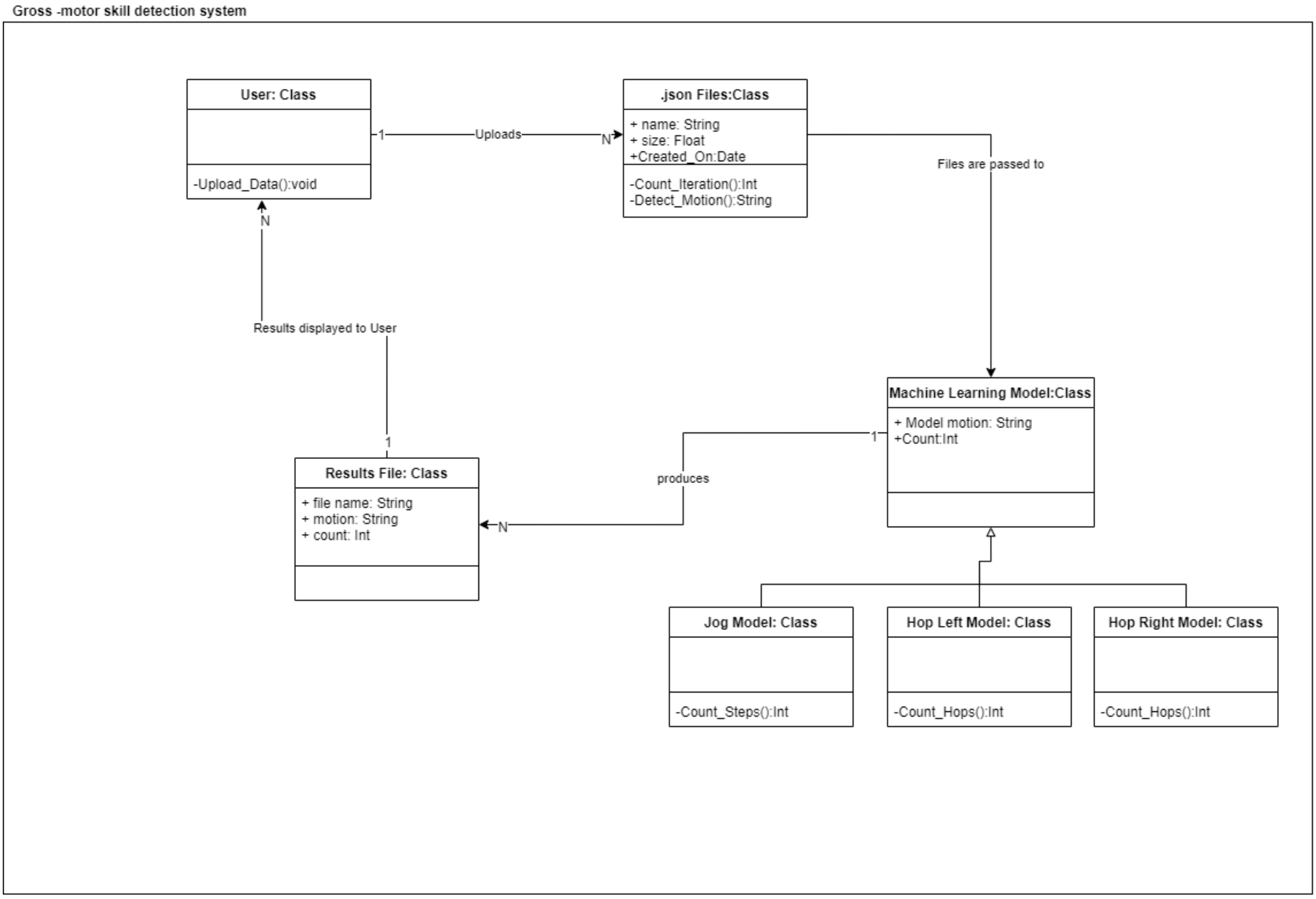
Textual Descriptions

Name: Upload test files Participating actors: User, GUI Entry condition: the user has test files to upload Exit condition: the files are uploaded to the GUI Normal scenario: the user uploads the files they want to process Error scenario: the user tries to upload non json files, told the file type is incompatible.	**Name: Gather Test Data(outside system)** Participating actors: User Entry condition: User has test subject and access to kine Exit condition: Data on coordinates frames is collected a json Normal scenario: The test subjects movements a captured and mapped by the kinect and stored as a json of coordinate frames Error scenario: The test subjects movements could not mapped or recorded

Name: Parse .JSON Files Participating actors: Machine Learning Models Entry condition: files are uploaded via the GUI Exit condition: The data is parsed into a processable format Normal scenario: The files uploaded are parsed to be passed to machine learning model Error scenario: the files are unable to be parsed	**Name: Export Result** Participating actors: GUI, User Entry condition: The files were successfully processed a results were produced Exit condition: The user has an exported copy of the resu Normal scenario: The user clicks "export" and the resu are exported as a file Error scenario: The export fails, user told to retry
Name: Identify Motion Participating actors: Machine Learning Models Entry condition: Files have been parsed and user clicks 'analyse' Exit condition: The machine learning model identifies the motion in each file Normal scenario: The user clicks the 'analyse' and the testing data is passed to the learning model to identify each files motion Error scenario: motion could not be identified	**Name: Display Result** Participating actors: GUI Entry condition: The files were successfully processed a results were produced Exit condition: The outcome displayed on the screen a written into a file Normal scenario: The machine learning model produc results in a file and are displayed to the User on the GUI Error scenario: The machine learning model could r produce results.
Name: Count iterations of motion Participating actors: machine learning model Entry condition: the motion has been identified Exit condition: the number of times the motion was completed in each files is recorded Normal scenario: the model counts how many times in each file, the relevant motion is fully completed Error scenario: The model is unable to detect if the motion is not being performed properly	

3.4.3. Object model

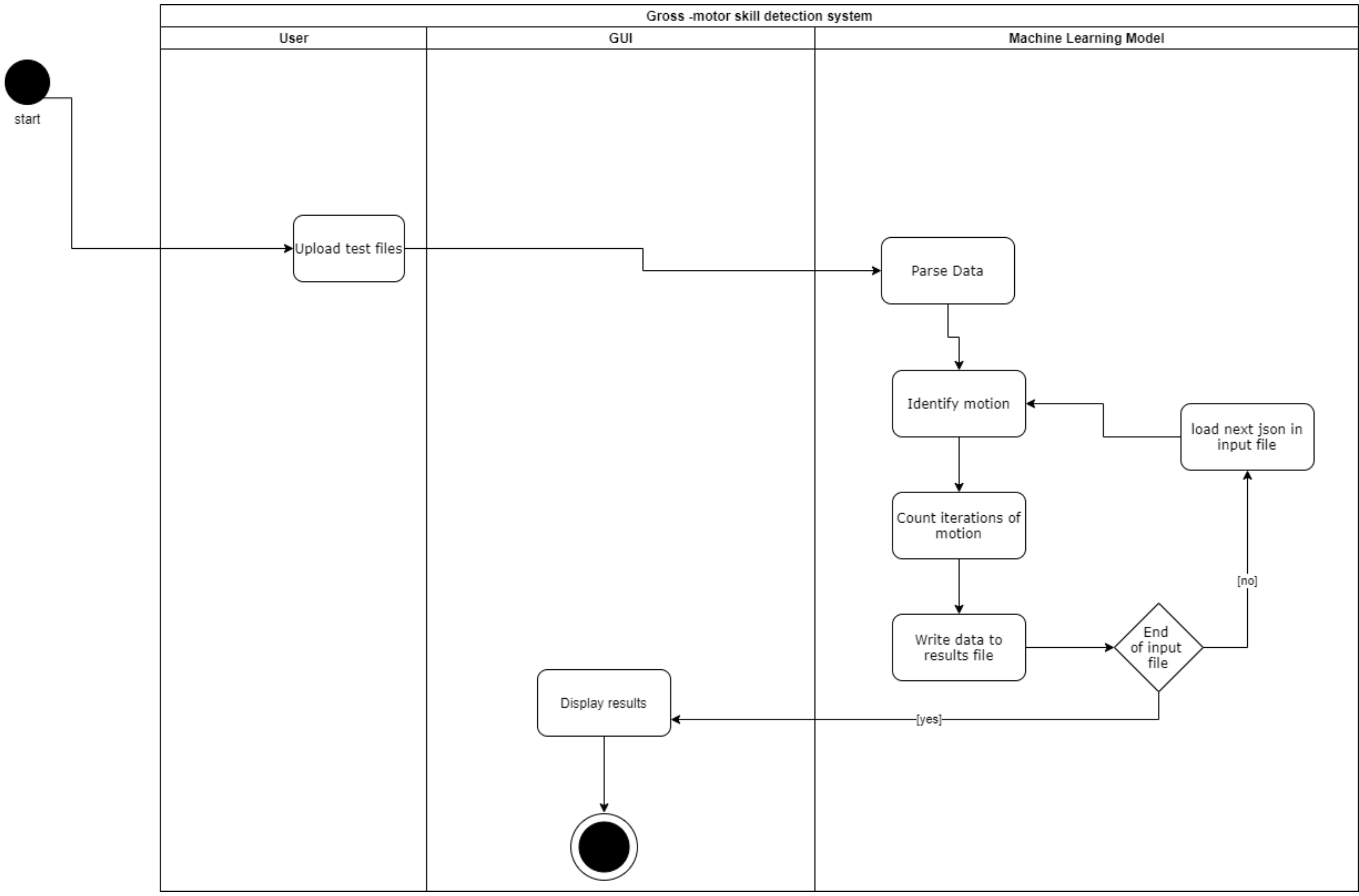
Class Diagram



The class diagram above shows how we expect the elements of our system to interact

- The user uploads multiple .json files
- These files are then passed to the machine learning model
- There are 3 machine learning models; jog, hop left and hop right (indicated by inheritance)
- The machine learning model then produces multiple results files
- The results are then displayed to the user

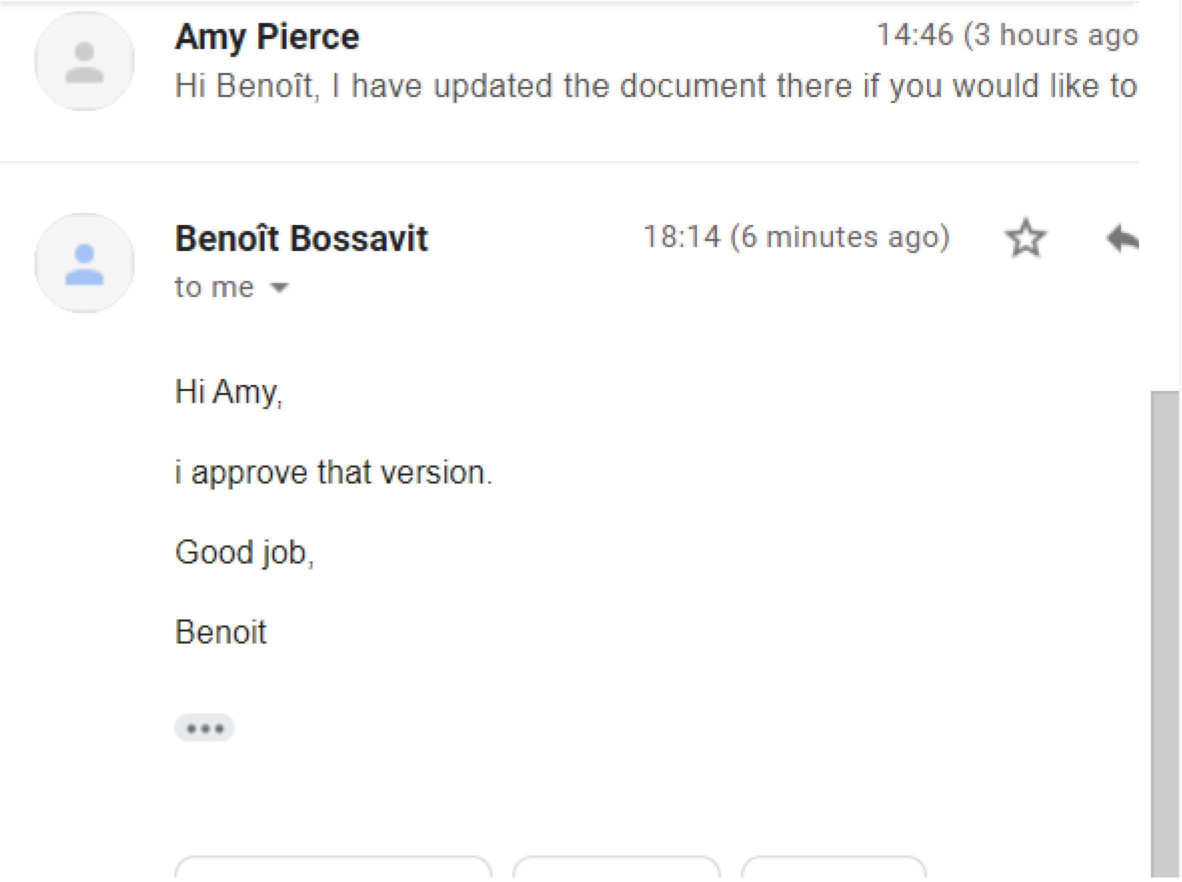
3.4.4. Dynamic model



Our Dynamic model (Activity diagram) shows the happy case order of events for our system, i.e. it ignores error cases such as the user uploading incorrect data or the machine learning model crashing.

- A user uploads research data which is then parsed.
- The first file in the data set is processed to identify the movement and see how many complete movements were recorded.
- If there are no more files in the data set, the process is over. Otherwise, the system moves to the next file and repeats until all files have been processed.
- When the end of the file is reached, the results are displayed.

4.Client Sign-Off



Amy Pierce 14:46 (3 hours ago

Hi Benoît, I have updated the document there if you would like to

Benoît Bossavit 18:14 (6 minutes ago)

to me

Hi Amy,

i approve that version.

Good job,

Benoit

8.2 Appendix 2- Software Design Specification Document

CS2013: Software Design Specification Outline

1.Introduction

1.1 Overview – purpose of systems

The purpose of this group project for CS2013/CS3013 is to design and create machine learning models for tracking gross-motor skills for our client Benoit Bossavit, a post-graduate researcher in the SCSS department. Our client is doing research on the development of gross-motor skills in young children and has asked that we create a machine learning model to help track certain movements to be able to determine whether the children can perform the motor-skills, and hence the overall purpose of our project is to create machine learning models that can recognize such motor-skills.

This project will help our client spend more time collecting data for his research rather than spending time developing the machine learning aspects himself.

1.2. Scope

This project consists of developing an efficient and accurate machine learning model that can identify what movement is being performed when a json file containing motion capture data is passed through the model.

The machine learning model will be presented with a motion capture data file and will determine which motor-skill it is relating to. It will look to determine how many times the skill was performed for each motion capture data file. In order for a motion to be considered 'successful' the machine learning model determines if it was performed sufficiently, i.e. if the motor-skill being checked for is a hop and the child jumped off one foot but landed on two, this is not considered a successful iteration of the motor-skill.

1.4. Definitions, abbreviations

Deep learning

Deep learning is an artificial intelligence function that imitates the workings of the human brain in processing data and creating patterns for use in decision making. Deep learning is a subset of machine learning in artificial intelligence (AI) that has networks capable of learning unsupervised from data that is unstructured or unlabeled. Also known as deep neural learning or deep neural network.

Gross-Motor Skills

Gross motor (physical) skills are those which require whole body movement and which involve the large (core stabilising) muscles of the body to perform everyday functions, such as standing and walking, running and jumping, and sitting upright at the table.

Machine learning

Machine learning is an application of artificial intelligence (AI) that provides systems the ability to automatically learn and improve from experience without being explicitly programmed. Machine learning focuses on the development of computer programs that can access data and use it to learn for themselves.

The process of learning begins with observations or data, such as examples, direct experience, or instruction, in order to look for patterns in data and make better decisions in the future based on the examples that we provide. The primary aim is to allow the computers learn automatically without human intervention or assistance and adjust actions accordingly

Neural networks

Neural networks are a set of algorithms, modeled loosely after the human brain, that are designed to recognize patterns. They interpret sensory data through a kind of machine perception, labeling or clustering raw input. The patterns they recognize are numerical, contained in vectors, into which all real-world data, be it images, sound, text or time series, must be translated.

1.5. References

Deep Learning

https://www.investopedia.com/terms/d/deep-learning.asp

Gross-motor Skills

https://childdevelopment.com.au/areas-of-concern/gross-motor-skills/

Machine Learning

https://expertsystem.com/machine-learning-definition/

Neural Networks

https://pathmind.com/wiki/neural-network

2. System Design

2.1. Design Overview

2.1.1 High-level overview of how the system is implemented, what tools, frameworks and languages are used etc.

The tools and technologies we have chosen are based on what we believe to be most suitable with some advice from our demonstrator. The main programming language we used for our project is Python as well as different libraries for the front-end and back-end.

The front-end will be able to load multiple motion capture files at once to be read by the machine learning model and be able to output multiple results and will be able to export results of the processed data into one of several file formats (e.g. .XLS, .CSV, .XML, .JSON, etc.)
.

We are going to be using the following tools to construct our front end.

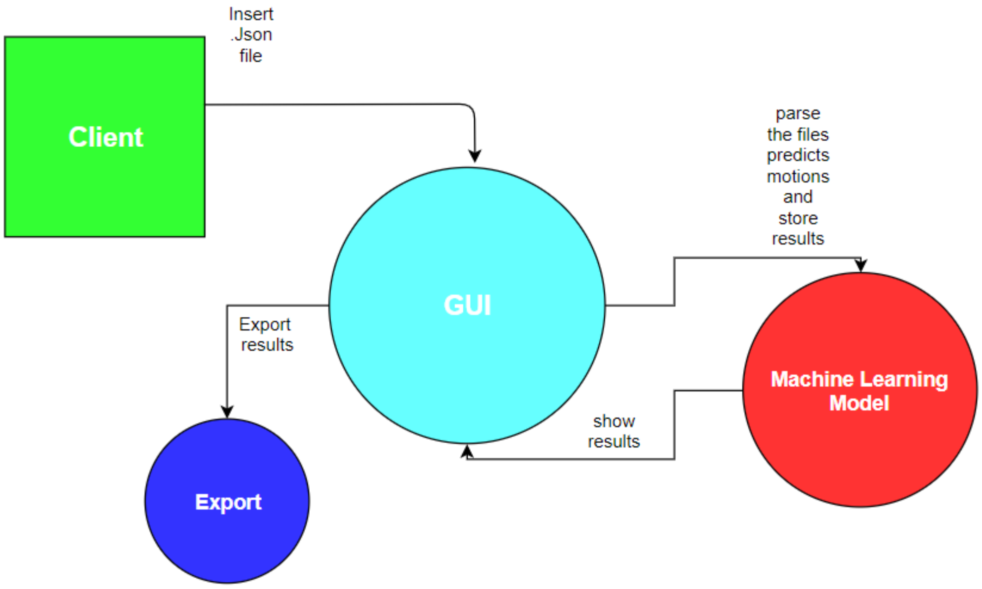
- PyQt: is a Python binding of the cross-platform GUI toolkit Qt, implemented as a Python plug-in.
- Tkinter : is the standard GUI library for Python. Python when combined with Tkinter provides a fast and easy way to create GUI applications

The backend will take in files selected in the selection system and parse the .json files into a readable data format. The system will have a pre-baked machine learning model that predicts motions at approximately 90% to 95% accuracy. After the .json files have been parsed, they will be sent into the pre-baked model, and will output a list of the motions that have been inputted to the frontend, followed by which motion they are most likely to represent.

We are going to be using the following tool to construct our back end.

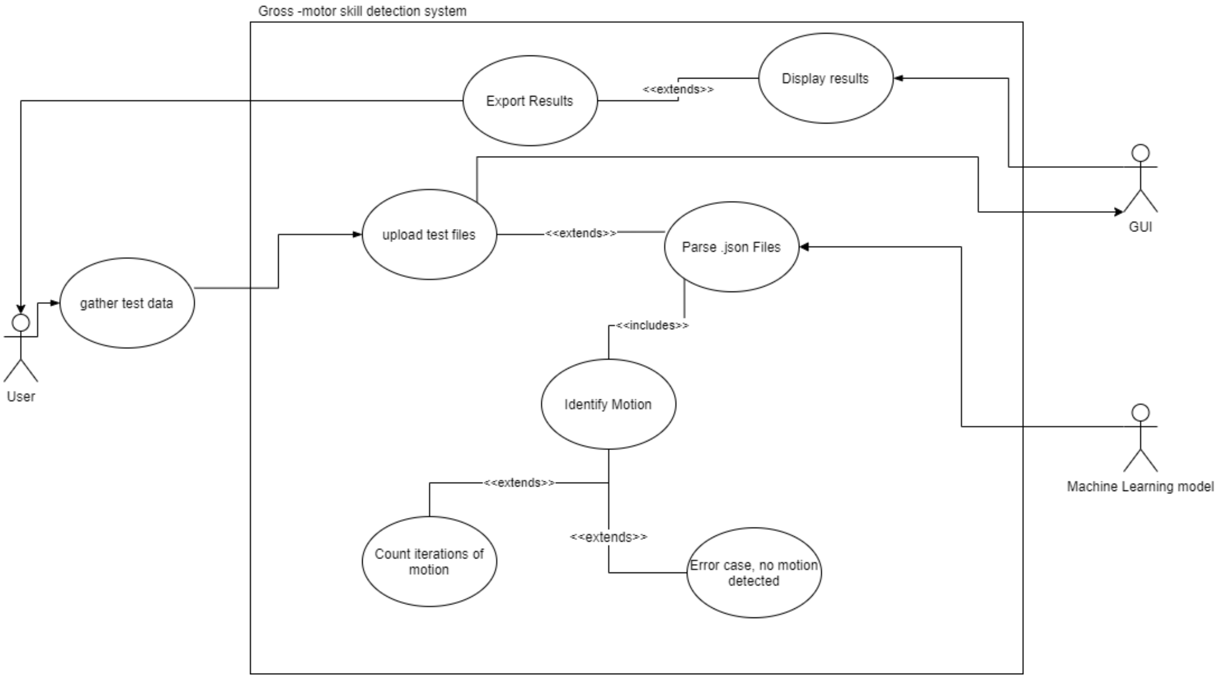
- TensorFlow: This is an open source library used for numerical computation and large-scale machine learning. Our machine learning model will be able to accurately predict the iterations of each motion from the .json motion capture file (e.g. if the motion is determined to be a Hop-Left, the system will be able to predict how many times a Hop-Left was done in the motion capture.). The results will be stored in a file document.

2.2.1. System Context



This context diagram outlines how each of the product's aspects interact as a system. The Gui interacts with all of the other components. The user is able to upload/insert .json files to the GUI which then passes them to the machine learning model which parses the files and performs the necessary analysis. The results are returned to the GUI and can then be exported

2.2.2 Use Case Diagram (from Requirements) ju

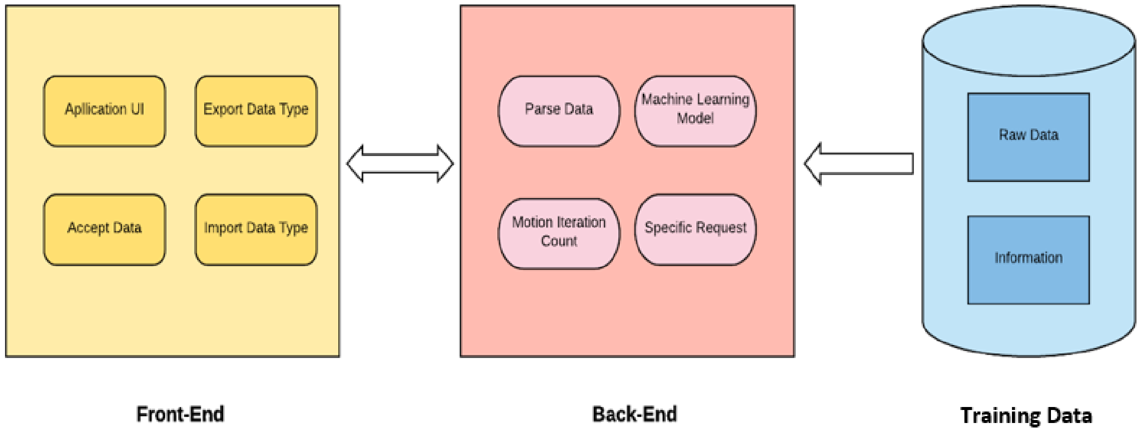


Textual Descriptions

Name: Upload test files	**Name: Gather Test Data(outside system)**

Participating actors: User, GUI Entry condition: the user has test files to upload Exit condition: the files are uploaded to the GUI Normal scenario: the user uploads the files they want to process Error scenario: the user tries to upload non json files, told the file type is incompatible.	Participating actors: User Entry condition: User has test subject and access to kinect. Exit condition: Data on coordinates frames is collected as a json Normal scenario: The test subjects movements are captured and mapped by the kinect and stored as a json file of coordinate frames Error scenario: The test subjects movements could not be mapped or recorded
Name: Parse .JSON Files Participating actors: Machine Learning Models Entry condition: files are uploaded via the GUI Exit condition: The data is parsed into a processable format Normal scenario: The files uploaded are parsed to be passed to machine learning model Error scenario: the files are unable to be parsed	**Name: Export Result** Participating actors: GUI, User Entry condition: The files were successfully processed and results were produced Exit condition: The user has an exported copy of the results Normal scenario: The user clicks "export" and the results are exported as a file Error scenario: The export fails, user told to retry
Name: Identify Motion Participating actors: Machine Learning Models Entry condition: Files have been parsed and user clicks 'analyse' Exit condition: The machine learning model identifies the motion in each file Normal scenario: The user clicks the 'analyse' and the testing data is passed to the learning model to identify each files motion Error scenario: motion could not be identified	**Name: Display Result** Participating actors: GUI Entry condition: The files were successfully processed and results were produced Exit condition: The outcome displayed on the screen and written into a file Normal scenario: The machine learning model produced results in a file and are displayed to the User on the GUI Error scenario: The machine learning model could not produce results.
Name: Count iterations of motion Participating actors: machine learning model Entry condition: the motion has been identified Exit condition: the number of times the motion was completed in each files is recorded Normal scenario: the model counts how many times in each file, the relevant motion is fully completed Error scenario: The model is unable to detect if the motion is not being performed properly	

2.2.3 System Architecture



This architecture diagram details the interaction between the different parts of the application.

The front end has the functionality to :

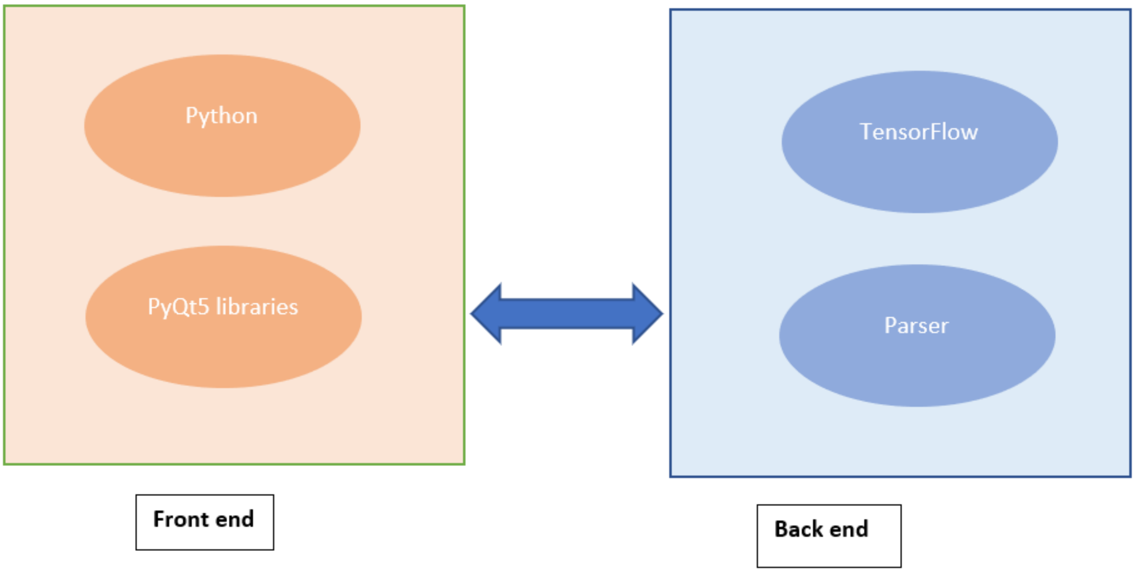
- import data
- export data
- accept data from the back end
- act as a user interface application where the user can drag and drop or manually select files to upload.

The back end has the functionality to

- parse data into a format that can be analysed
- count motion iterations
- use machine learning models to identify the motion
- make specific requests for certain motions for example if the motion is jump, it can request to compute the distance.

The training data is fed to the back end to train the machine learning models:

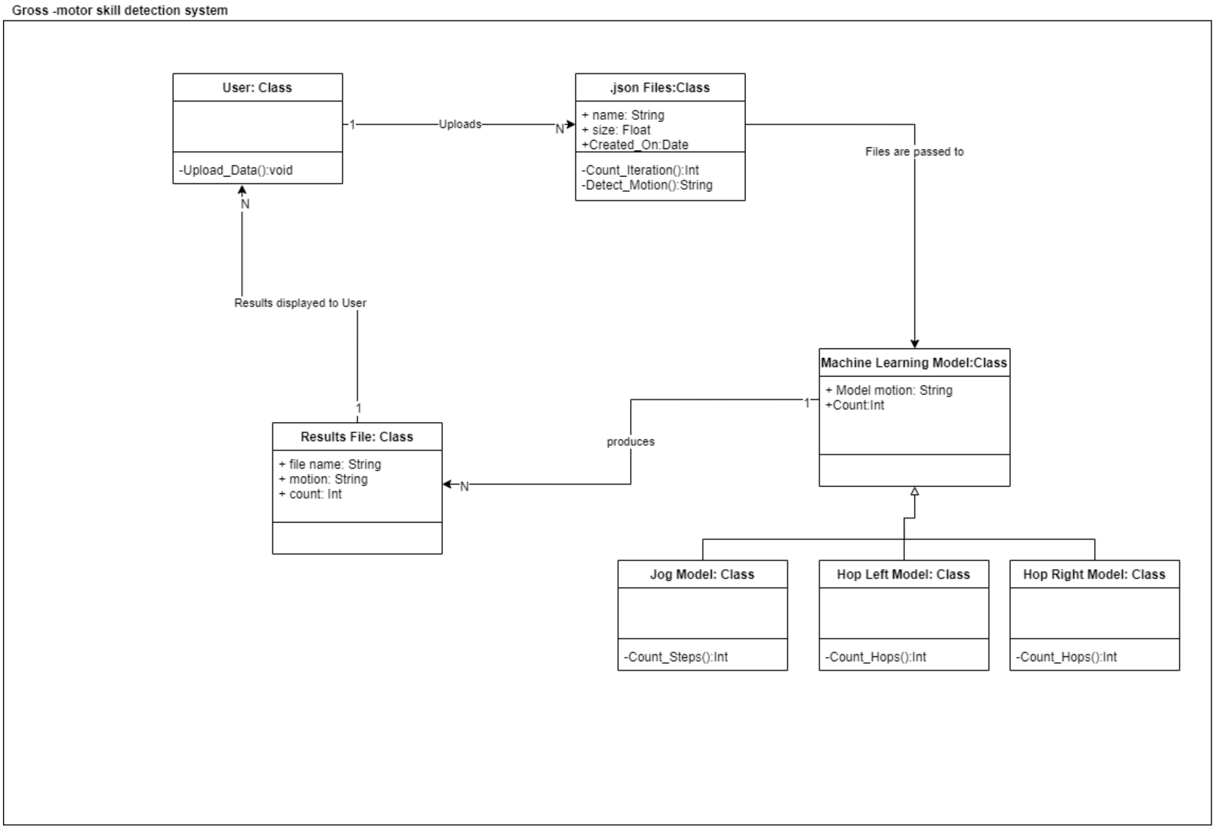
- We got the raw data from our client, which are the data frames about the motions.
- Create some new training data based on the original data by shifting the coordinates.
- We get the guide line from our client, demonstrator and mostly doing online research.



Above is a very simple depiction of the main technologies used in each aspect of the application.

- The front end GUI is made using Python and uses PyQt5 libraries to add functionality
- The back end machine learning models uses tensorFlow and a parser to allow the data to be converted into an analysable format.

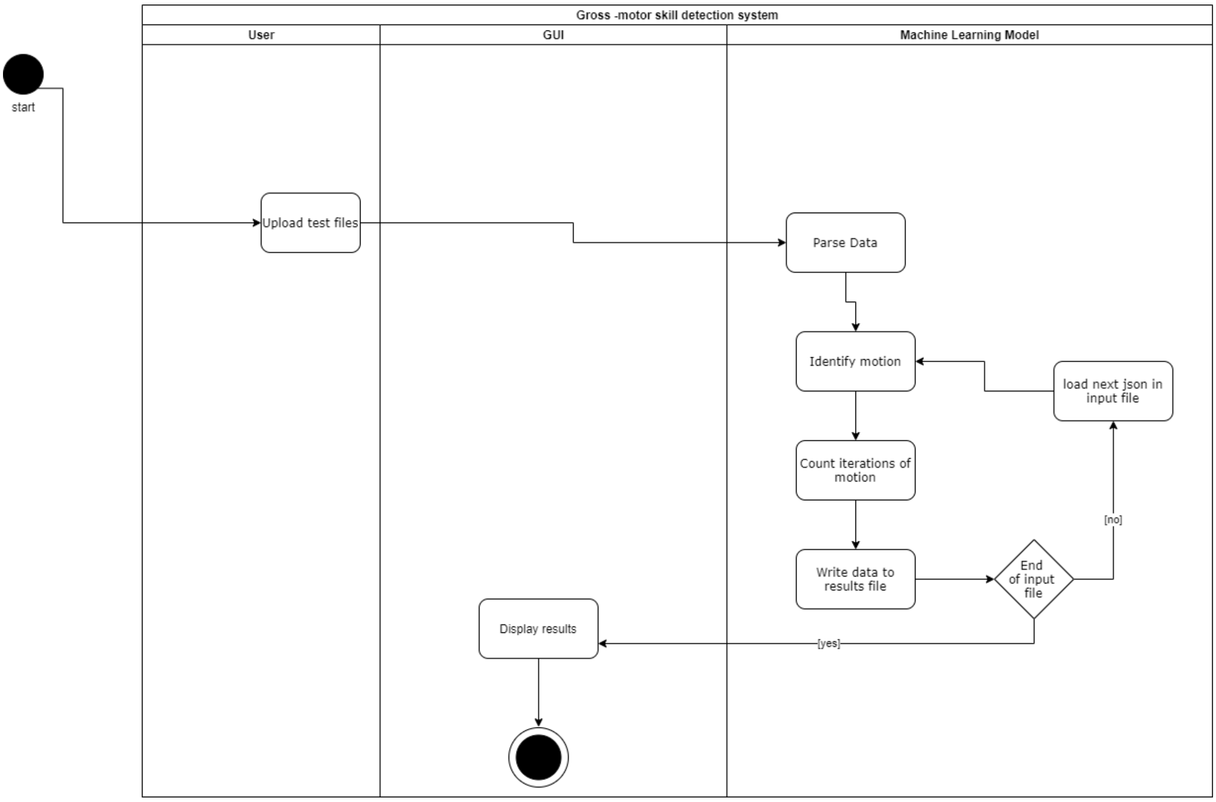
2.2.4 Class Diagram



The class diagram above shows how we expect the elements of our system to interact

- The user uploads multiple .json files
- These files are then passed to the machine learning model
- There are 3 machine learning models; jog, hop left and hop right (indicated by inheritance)
- The machine learning model then produces multiple results files
- The results are then displayed to the user

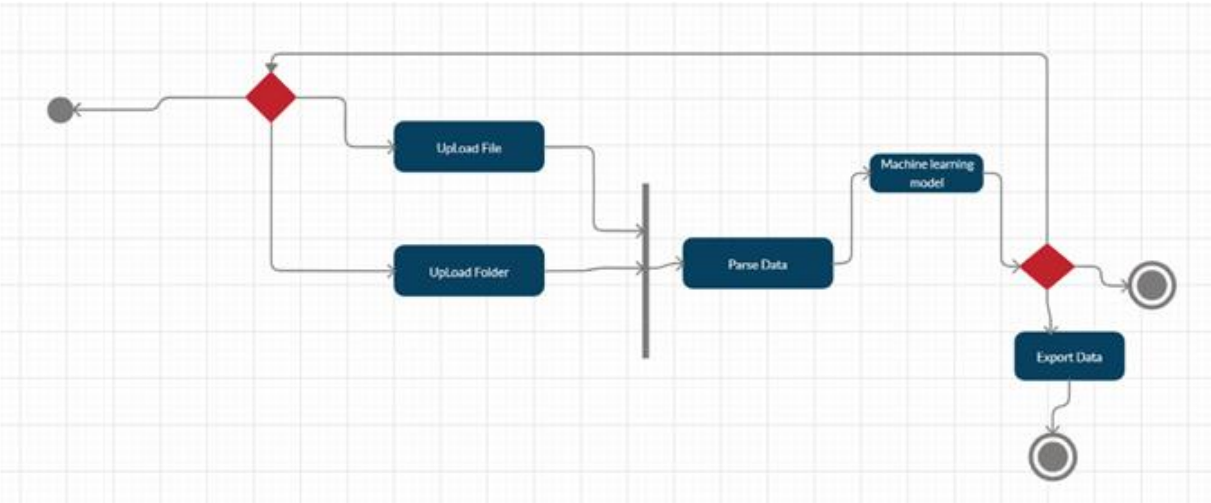
2.2.5 Sequence Diagrams



This sequence diagram outlines the happy case flow through the system.

1. When a file is uploaded it is passed to the machine learning model to be parsed
2. The machine learning model will work to identify the motion
3. The machine learning model then works to count how many times that motion was successfully performed
4. The results are then output
5. If there is no more files to be analysed then the results are displayed and the program is ended
6. Otherwise, steps 2 to 4 are repeated until all files uploaded have been analysed and the results are then displayed and the program ends

2.2.6 State Diagrams

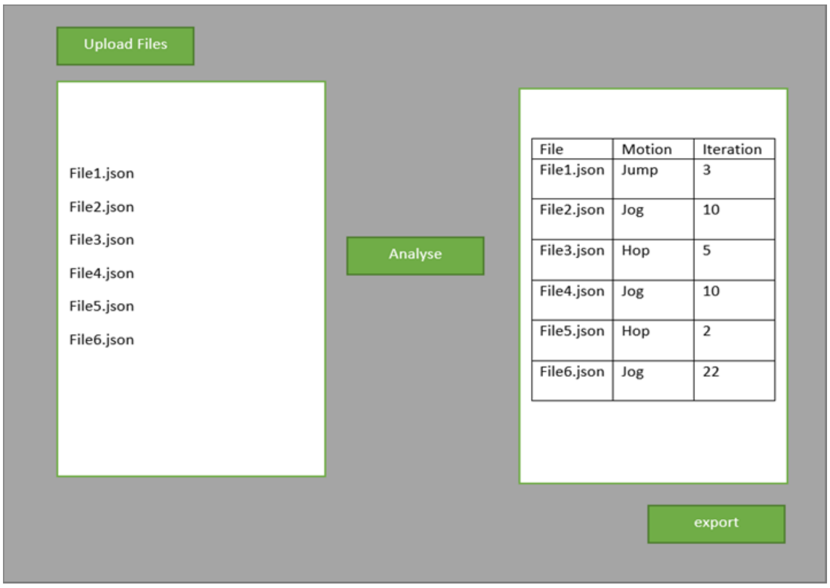


This state diagram depicts a simple flow of events in our system.

- The client inserts or uploads a file or folder to the GUI
- The files will then be parsed
- then the machine learning model will accept the parsed files and provided there were files to analyse
- the results will be produced and exported back to the client.
- If there is no files to be parsed, the program will then terminate
- The program will continue to allow the user to upload files until they choose to terminate

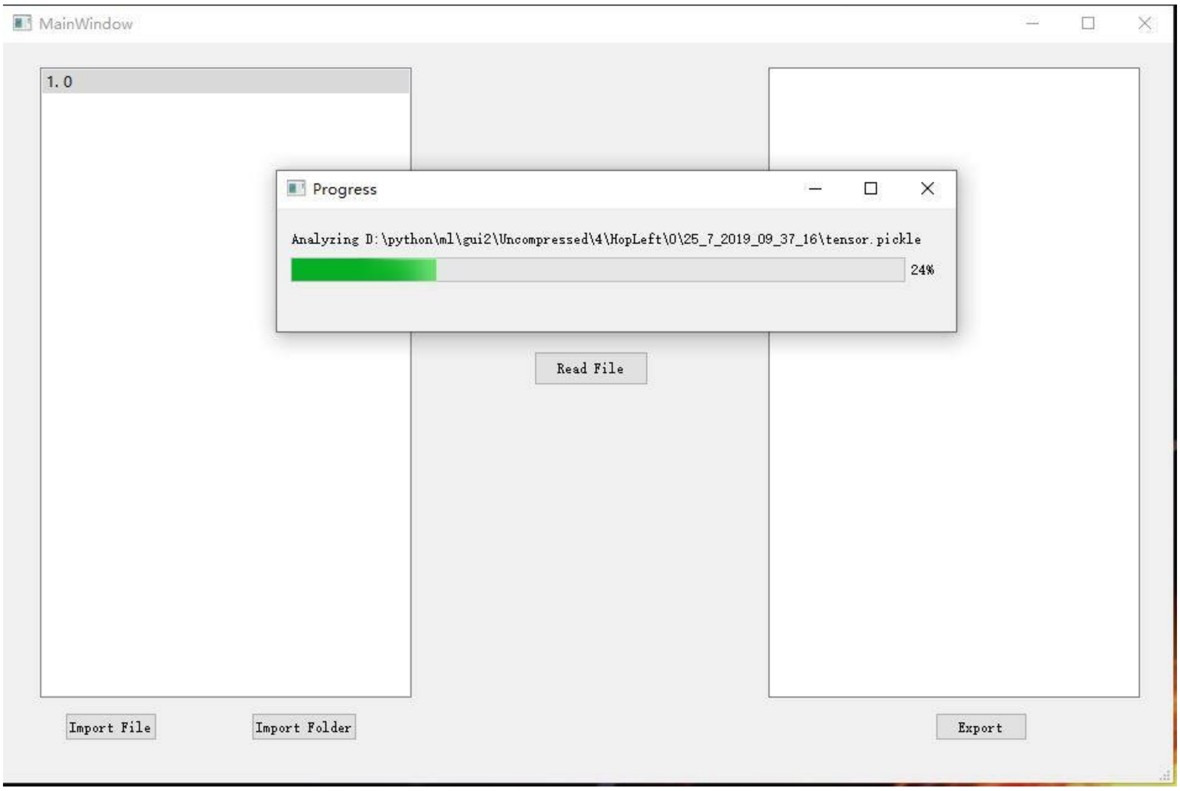
2.2.7 Interface Mockups

When we began this project we drew up some user interface mockups and this was the design we decided to work towards:



File	Motion	Iteration
File1.json	Jump	3
File2.json	Jog	10
File3.json	Hop	5
File4.json	Jog	10
File5.json	Hop	2
File6.json	Jog	22

After completing some of our coding tasks, we now have a user interface that works and we can see that we have a fairly similar UI design to what we had been looking for:



```
Analyzing D:\python\ml\gui2\Uncompressed\4\HopLeft\0\25_7_2019_09_37_16\tensor.pickle
```