Contents

[Introduction 2](#_Toc55771151)

[Customers 2](#_Toc55771152)

[Team 5](#_Toc55771153)

[Design 6](#_Toc55771154)

[Design Process 6](#_Toc55771155)

[Design Diagrams 6](#_Toc55771156)

[Use Case Diagrams 6](#_Toc55771157)

[Class Diagrams 9](#_Toc55771158)

[Sequence Diagram 11](#_Toc55771159)

[Bibliography 14](#_Toc55771160)

QuickML

|  |  |
| --- | --- |
| **Juan Lopez** | **Thony Yan** |

# Introduction

Quick Machine Learning (QuickML) is a computer application to introduce users to Machine Learning. Through this application users will have the power to use Machine Learning, specifically computer vision, without the need on knowing how to code and knowing the complexity of Machine learning.

## Customers

1. This application is mostly intended for educational purposes so anyone who wishes to have an introduction to the applications of computer vision in machine learning may use it.
2. Our team will design a user-friendly and straightforward software where anyone with basic computer knowledge will be able to train and implement their Machine Learning algorithm.
   1. Machine Learning and Artificial Intelligence have gained prominence in recent years, according to Indeed Machine Learning Engineers have a 344% growth in job posting in the year between 2015 to 2018 making it the rank 1 growth in rate. Another Machine Learning related job also ranks at number 13, Computer Vision Engineer (Daws, 2019). with companies like Google, Microsoft, and Amazon creating a cloud-based platform for Machine Learning. There are varieties of ways to use Machine Learning from image and voice recognition, solving complex problems, detecting anomalies, self-driving cars, autonomous systems, etc. Clearly, Machine Learning is a very important field that will help shape a better future. But, as it is a relatively new and exploding field with many fields put together like Calculus, Statistics, and Programming involve, everyday people will shy away from using Machine Learning because of its complexities. This is why we need a way to introduce individuals to Machine Learning as there’s a severe need for Machine Learning talent (Terra, 2020). Our product, QuickML, will help alleviate this gap by creating a software that will allow individuals to have the power of Machine Learning without needing to know the complex fields in it.
   2. Although there are other tools and applications in the market, we haven't found any straightforward ones. These applications were made with more complexity as they were made for users with more experience and knowledge in the field.
   3. QuickML would be a playground where you man changes parameters to make a model, train the model, and see and test the results. In this way users can start getting an interest in Machine Learning and will pursue more knowledge on it.
3. Contextual issues and external constraint:
   1. As we are making a Graphical User Interface (GUI) for QuickML through the Python programming languages in a Windows 10 operating system we do not how it will perform in another operating system like Linux or Mac. We will be using PyQt to create the GUIs and from the documentation, it says that it should be compatible with that other operating system, but we do not know to what extent.
   2. Also, we will most likely put some limitations to the parameters that users can input into the software in the machine learning model creation process as having a very large number will cause instability when training the machine learning algorithm. Special hardware known as Graphical Processing Unit (GPU) needs to be accessible for bigger models.
4. Schedule outline:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| QuickML  As a User I want to make and test a Machine Learning model | | | | |
| Epic | User Story | Task | Story point | Done? |
| Preprocessing GUI | As a user I want to organize data to get it ready for the training step. | Create folder directory structures. | 5 |  |
| Copy from one directory to corresponding directory | 8 |  |
|  | As a user I want to see the labels to know I put the right label name. | Let the user input the labels they want. | 1 |  |
| Show the labels the user has inputted | 2 |  |
| Let the user delete inputs | 3 |  |
| Model GUI | As a user I want to make different configuration to see the effects of the model. | Clearly put what type of layer can be put into model | 1 |  |
| Allow user to put inputs for configuration. | 3 |  |
| Show the model. | 2 |  |
|  | As a user I want to see the progress of the training so I can know how the process is going. | Put a progress bar. | 5 |  |
| Model Testing GUI | As a user I want to load my model to test if the training worked. | Create prompt for user to pick a file. | 1 |  |
|  | As a user I want to pass an image in the GUI and get a result. | Create prompt for user to pick a file. | 1 |  |
| Show result | 5 |  |

* 1. For the user story allocation, we use Story Points and order the importance using the Fibonacci sequence. Each GUI has their own Story Point sequence.
  2. The Sprint period is one week consisting of 4 sprints.
  3. Sprints
     1. **Sprint 1: Preprocessing GUI/**
        + Create GUI layout
        + Have a text input for user label inputs and directory paths
        + Make a text windows that show folder paths and label names
        + Create a way to delete Labels and corresponding folder path
        + Use inputs to create necessary directories to move images into
        + Move images from source to destination path
     2. **Sprint 2: Preprocessing GUI / Modeling GUI**
        + Fix any problems from preprocessing that might cause instability to the next GUI
        + Create GUI layout
        + Have a text window that shows the current model
        + When configuration is made, run the training process and create a progress bar to see progress.
        + Have an easy layout to put model structure building blocks
        + Once everything is done set some constraint to avoid creating unstable or impossible models.
     3. **Sprint 3: Model Testing GUI**
        + Fix any problems from modeling GUI that might cause instability to the next GUI
        + Create GUI layout
        + Create a way to load a model
        + Have a function to put input image
        + Have one image box showing the image
        + Have a text box showing the classification
     4. **Sprint 4: Wrap up**
        + Solve any necessary problems that can cause a software crash.
        + See if any previous GUI can be improved to be more user friendly.

## Team

1. Thony Yan – Programmer / Machine Learning Engineer
   1. Motivation
      1. My motivation is simply trying to give back to the community, my university Florida International University has always advertised the concept of lifelong learning and giving back to the community. Seeing as Machine Learning has exponentially become a popular subject. I feel like the only way we can advance in this field is to have a bigger community interested in this field to have more collective knowledge and improve Machine Learning for a brighter future. To do this we must first get the community to be interested in this subject so I believe letting people play around and see the results would ease their way in.
      2. This will be useful in my career as I plan to work as a Machine Learning Engineer. This would that I can be a candidate to be a Software Engineer or Machine Learning Engineer by making applications and gaining experience in the Software process.
   2. Role
      1. Programmer and Machine Learning engineer will be handling most functionality that deals with making the Machine Learning aspect functional
      2. Have about two years’ experience with Machine Learning mostly on Computer Vision
2. Juan Lopez – Programmer / GUI Designer
   1. Motivation
      1. This is a pretty good project to use because neural networking is very important. I may not have any experience right now with neural networking, but I have a lot to gain from it.
      2. Neural networking is meant to recognize the underlying relationship in data to solve issues. For example, in the company I work with, we often must process IDs and ensure they are not fake. If we had neural networks that processed them, it could speed up the processing of ID’s and eliminate any ID that is not valid or potentially fake. With a car, we would want it to identify what is a human and what isn’t a human so it can avoid any fatalities. Neural networks can be used in so many ways, so learning about them can be relevant for almost any career.
   2. Role
      1. The designer of the GUI makes the program look as pretty as possible. I must ensure the GUI also functions properly with the code in the background so that the neural network works properly.

# Design

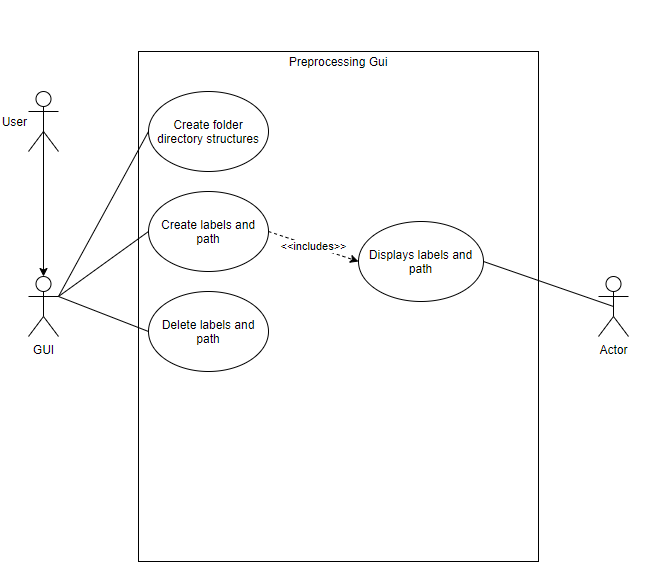
Software design is the process of defining the software’s’ architecture, components, interfaces, and other characteristics of a system or component. Also, software design is the software engineering life cycle activity in which software requirements are analyzed in order to produce a description of the software’s internal structure that will serve as the basis for its construction. QuickML software design will follow the Component-Based Design (CBD) strategy, this approach is creating well-defined interfaces and dependencies that can be composed and deployed independently (Pierre Bourque, 2014).

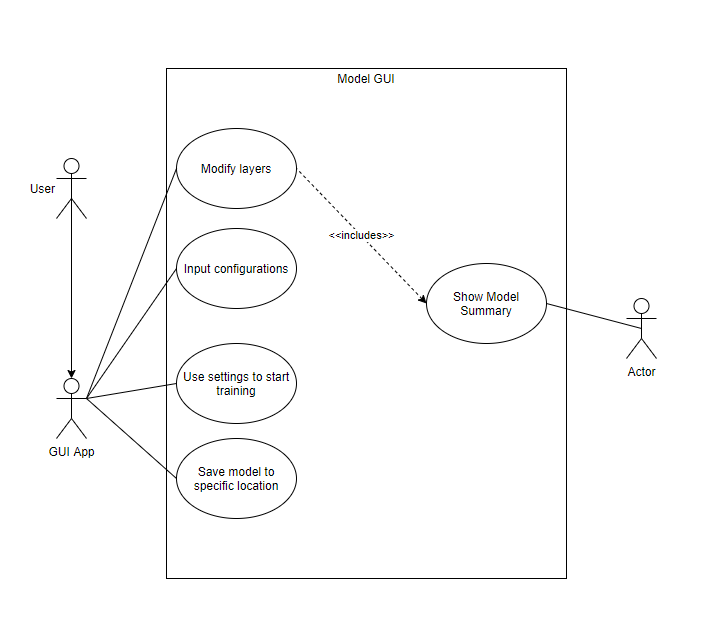
## Design Process

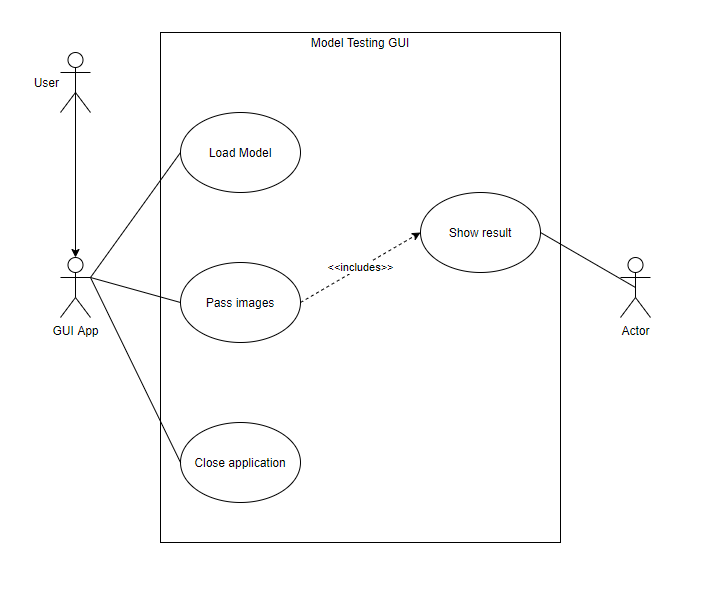
The way the design process went for QuickML was by looking at other similar applications first to see what they offer. From the information gathering, we found one application that would fit the criteria, but we believed it was too advance and not user-friendly for beginners. As a team, we first needed to share how the process should go. When doing computer vision with Machine Learning the process is usually: Preprocess data, make model, train model, and test model. Through this order, we decided to make three GUIs, the first one to preprocess the data, the second to make and train the model, and the last one to test the model. With that process in mind, we move on to create use case diagrams, sequence diagrams, and class diagrams.

## Design Diagrams

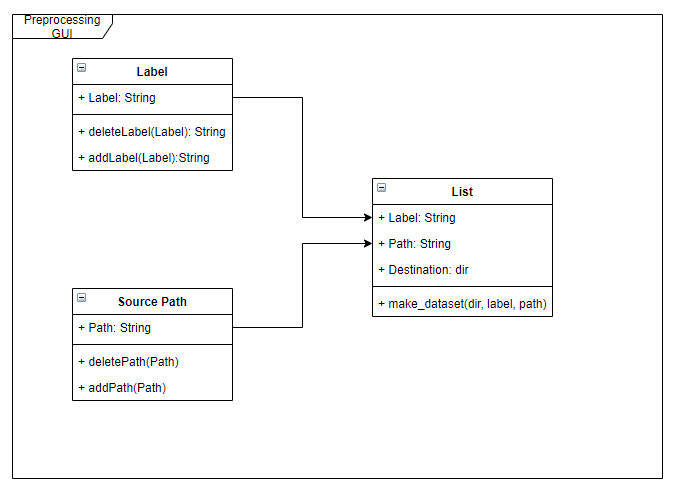
### Use Case Diagrams

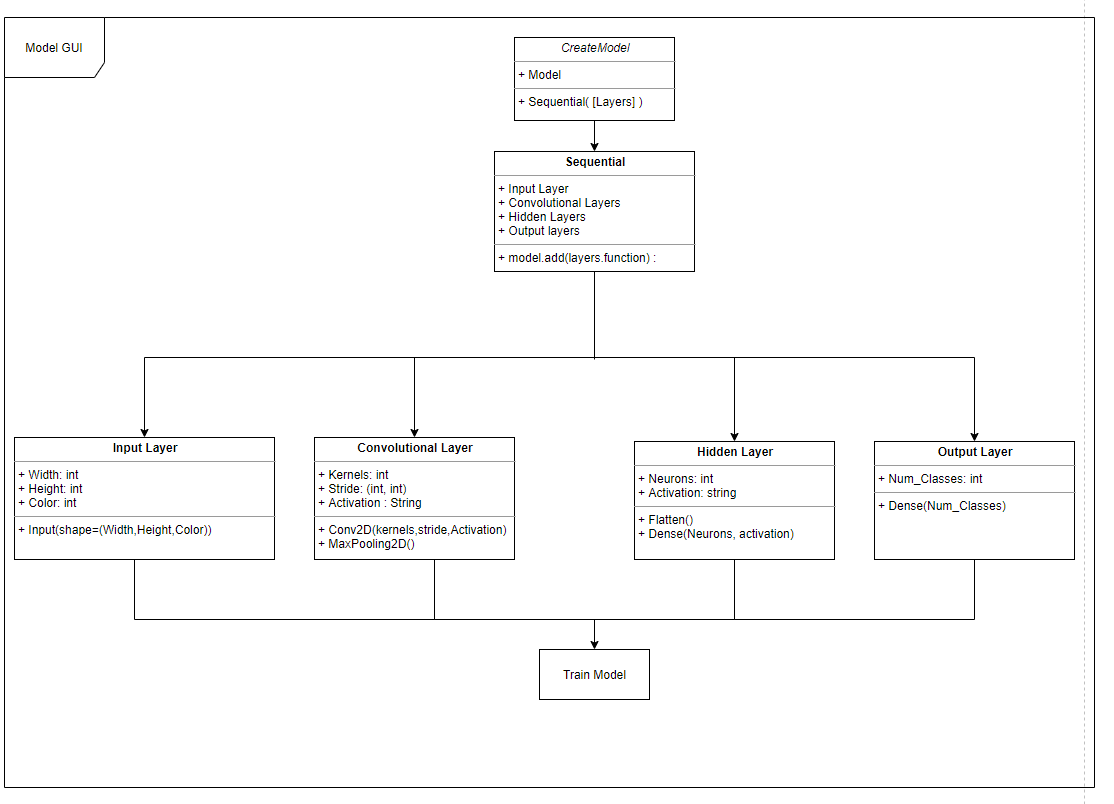


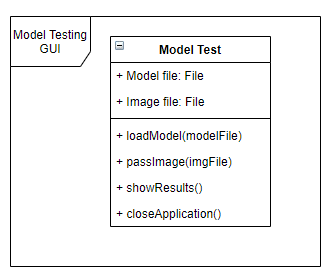




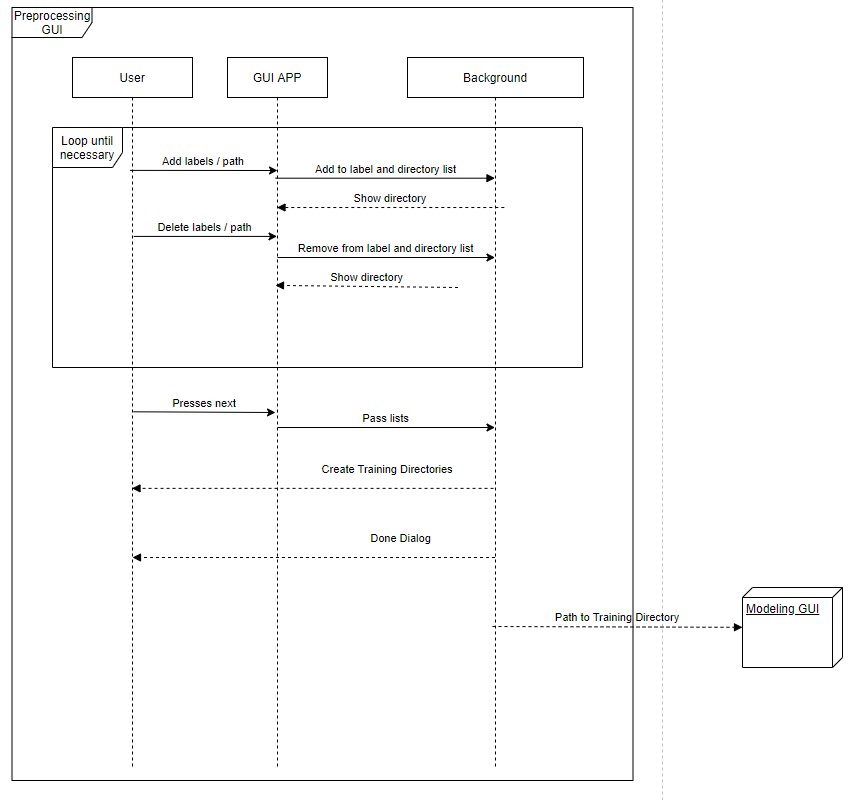
### Class Diagrams

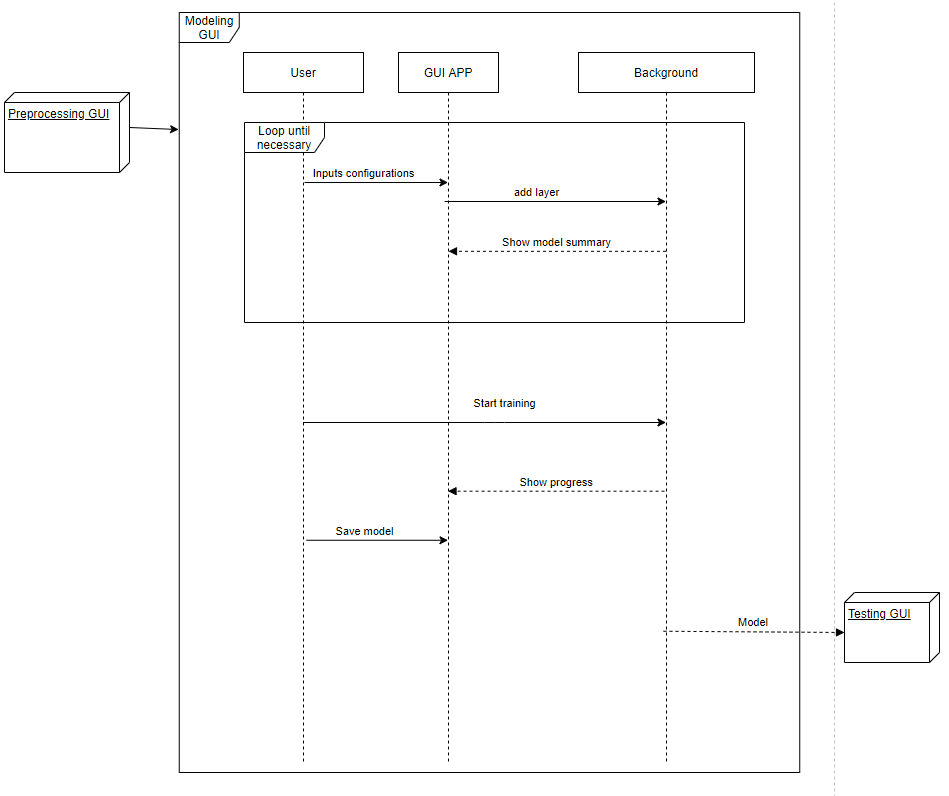


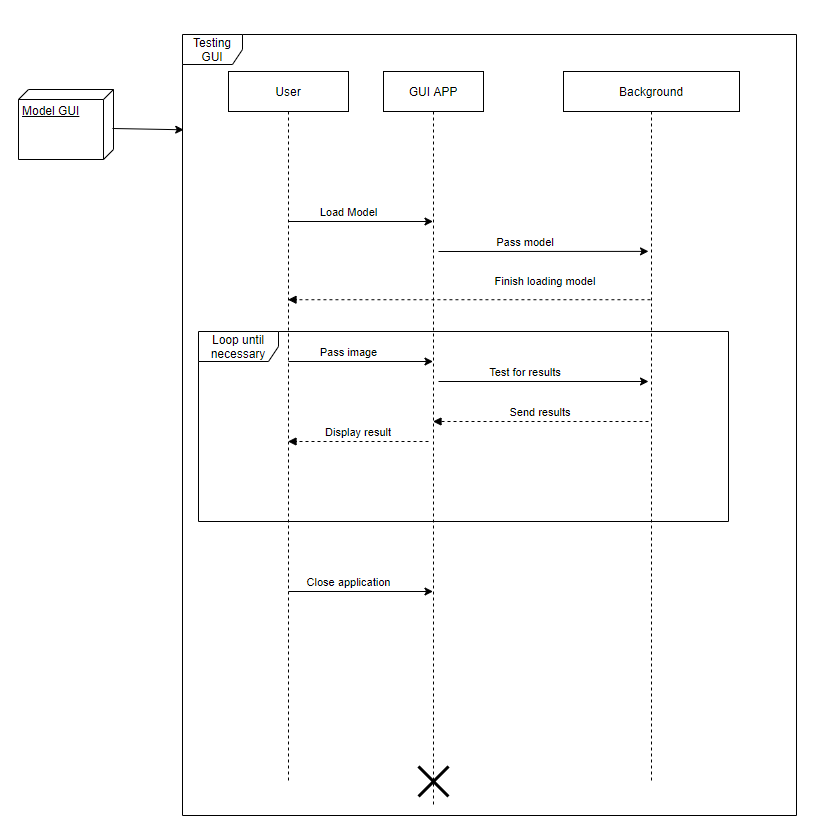




### Sequence Diagram







# Bibliography

Daws, R. (2019, March 15). *Surprise! Machine learning jobs are high-paying and in-demand*. (AINEWS) Retrieved from https://artificialintelligence-news.com/2019/03/15/machine-learning-jobs-high-paying-demand/

Pierre Bourque, R. E. (2014). *Guide to the Software Engineering Body of Knowledge, Version 3.0.* IEEE Computer Society. Retrieved from www.swebok.org.

Terra, J. (2020, June 4). *The Rise of Artificial Intelligence and Machine Learning Job Trends*. (Simplilearn) Retrieved from https://www.simplilearn.com/rise-of-ai-and-machine-learning-job-trends-article