Software Requirements Specification

for

Guess The Doodle

Prepared by:-

Name- Naman Jain(BT17CSE076)

Shantanu Shinde(BT17CSE115)

Indian Institute Of Information Technology, Nagpur

03-02-2019

Table of Contents

1. Introduction

1.1 Purpose

The purpose of this document is to give a detailed description of the requirements for the fun game “Guess The Doodle”. It will illustrate the purpose and complete declaration for the development of the system. It will also explain system constraints, interface, and interactions with other external applications. This document is primarily intended to be proposed to a customer for its approval and a reference for developing the first version of the system for the development team.

1.2 Product Scope

Guess The Doodle is a fun game that can predict a doodle drawn by the user. Our main aim is to efficiently use our Machine Learning Algorithms not only in big projects like Self Driving Car but also in fun games so that user can experience games like never before.

Guess The Doodle is a game that predicts doodles drawn by the user in its drawing pad. The software needs a good processor and graphics card to deeply train the datasets. So better is the hardware specifications, better will be its every prediction.

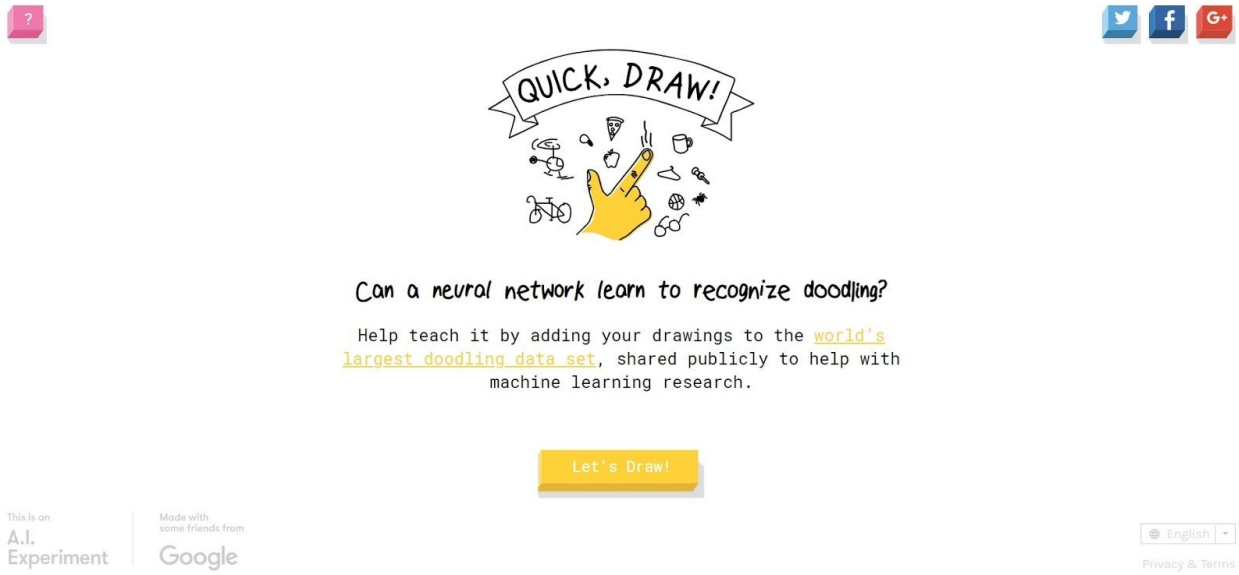


Fig. 1 - (Expected Software)

1.3 References

- IEEE Software Engineering Standards Committee, “IEEE Std 830-1998, IEEE Recommended Practice for Software Requirements Specifications”, October 20, 1998.
- Google Quick Draw Datasets.

2. Overall Description

2.1 Product Perspective

This system consists of interactive Web GUI which is accessible from any PC. It will take images drawn by the user and send it to the server. Then it converts it into Numpy Array. Then model evaluation will take place.
Now, this model will throw us the predicted result, and finally, this result will be displayed in the web browser.

Since this is a data-centric product it will take trained data from somewhere and will be stored somewhere. Every time the user draws a doodle, it will use this trained data and will predict what the user has tried to draw. Now, this current data will be trained again and will be used to improve our model. So next time it will predict results even better.

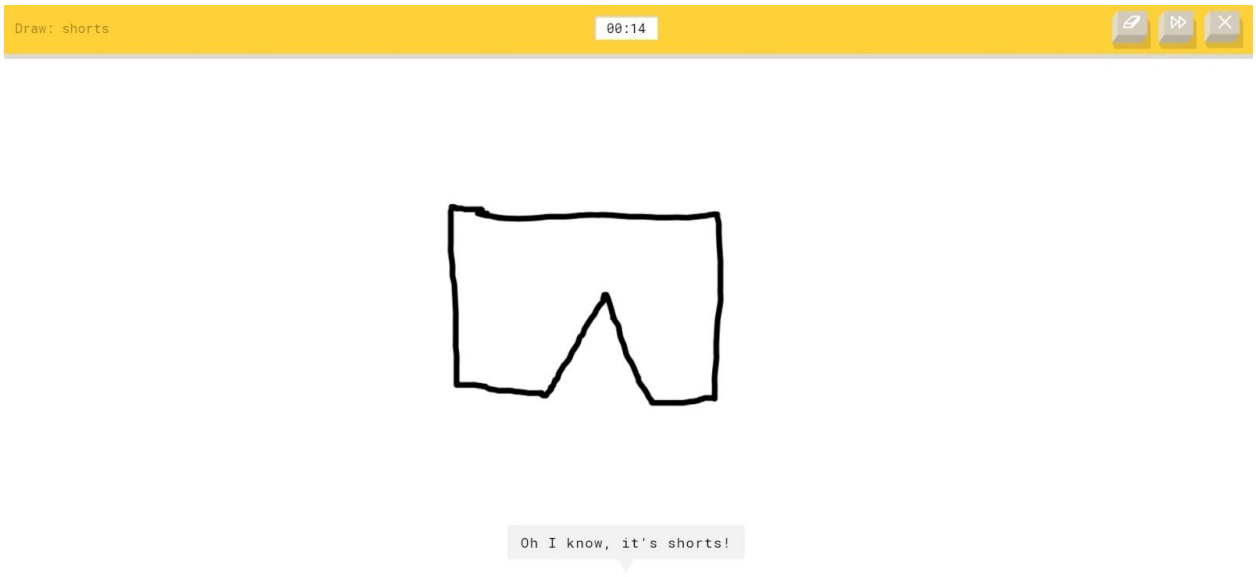


Fig. 2 - (Expected Software)

2.2 Product Functions

This software will use deep learning algorithms to predict the doodles. There will be a play button on the first page. On clicking the play button, The user will be able to draw a doodle. Now, this doodle will pass through several Machine Learning Algorithms and will predict what the user has drawn. The result will be shown in the browser itself, as well as what the user needs to draw, that too will be displayed in the browser after hitting the play button.

2.3 User Classes and Characteristics

The product can be used by anyone, as there are no such user classes. Same classes will be used by everyone. Since this is an interactive doodle prediction game, so everyone will use the same web app.

2.4 Operating Environment

The product can run on all hardware environments, but hardware with high-end CPU and Graphics card are recommended.
On the operating system side, the product will support Linux, Windows, OSx
One and only software requirement is the web browser.

2.5 Design and Implementation Constraints

Hardware constraints for training the model.
GPU RAM, CPU RAM should be large enough for the larger and deeper model. Larger the size of RAM, bigger will be the deep learning model. Which will improve the Doodle recognition and also the number of classes of the doodles to be recognized will increase.
The most important constraint is the web browser. The product interacts with the user in the web browser.

3. External Interface Requirements

3.1 User Interfaces

As the user runs the product in a Web browser, a play button will be displayed on the middle of the page. As the user hits the play button, the web browser will display on the upper section of the page what the user needs to draw. The lower section will display a scribble pad where the user will draw its doodle. Parallelly 10 seconds timer will also start. It's very important that the user must complete it's drawing in this span of time. After the time is complete, it will show whether the object drawn is correct or not. It will also display the predicted result.

3.2 Hardware Interfaces

It uses an extensive amount of CPU and Graphics Card to predict the doodle. So a good amount of both of these is expected.

3.3 Software Interfaces

The application interacts with the web browser so that it can efficiently interact with the user.

3.4 Communications Interfaces

The communication between the different parts of the system is important since they depend on each other. However, in what way the communication is achieved is not important for the system and is therefore handled by the underlying operating systems

4. System Features

"Guess The Doodle" needs a Web browser for the interaction with the user. A steady internet connection is expected.

5. Other Nonfunctional Requirements

5.1 Performance Requirements

This software needs an extensive amount of GPU and CPU for good performance of the software. Better trained model and larger dataset will help improve the accuracy and performance of the software.

5.2 Safety Requirements

This is a freeware, so there is no as such safety requirements for the product.

5.3 Security Requirements

The product does not take any personal and sensitive information of the user. Hence no security measures are required.

5.4 Software Quality Attributes

The software is available on all devices that have a web browser. This is a web app, and no licensing is required. Hence its portable from one system to another.
Since the product is open source, other developers can help to maintain and upgrade the software.
The model can also be trained to recognize written text, So it can be used as a Character recognition software.

5.5 Business Rules

This is a freeware and open source software.

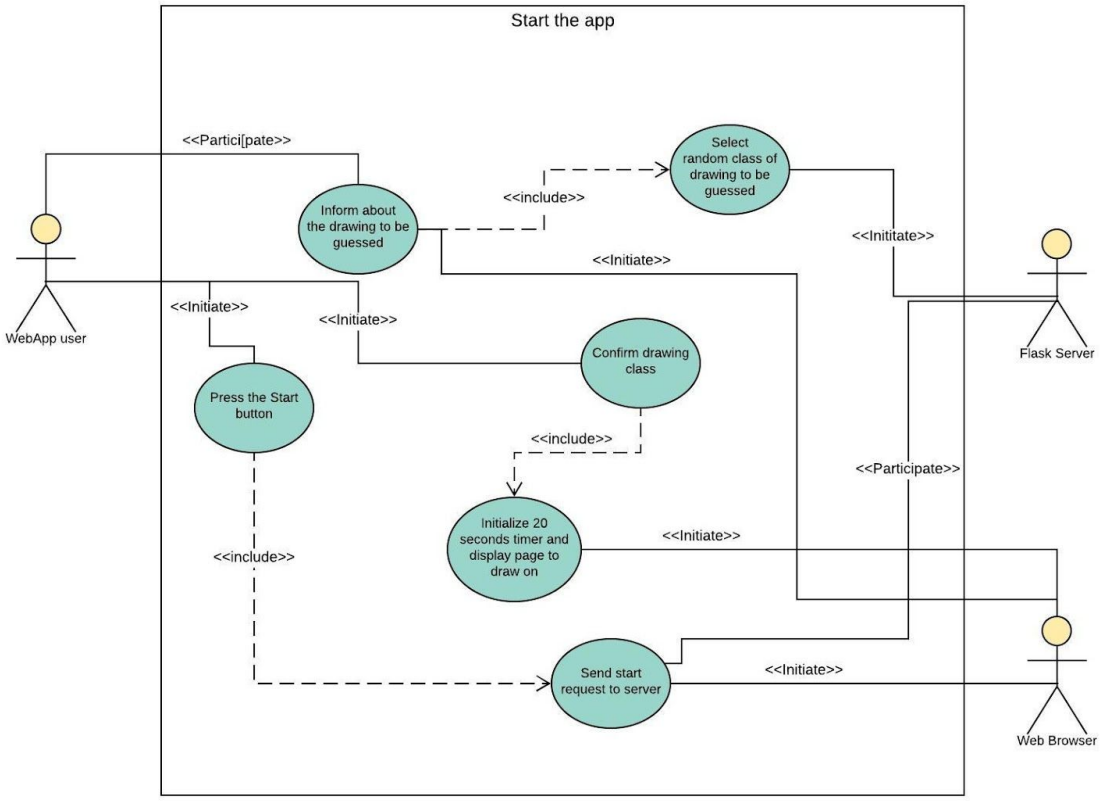
6. Other Requirements

All APIs, libraries, and databases are free and open source. So no legal requirements are necessary.

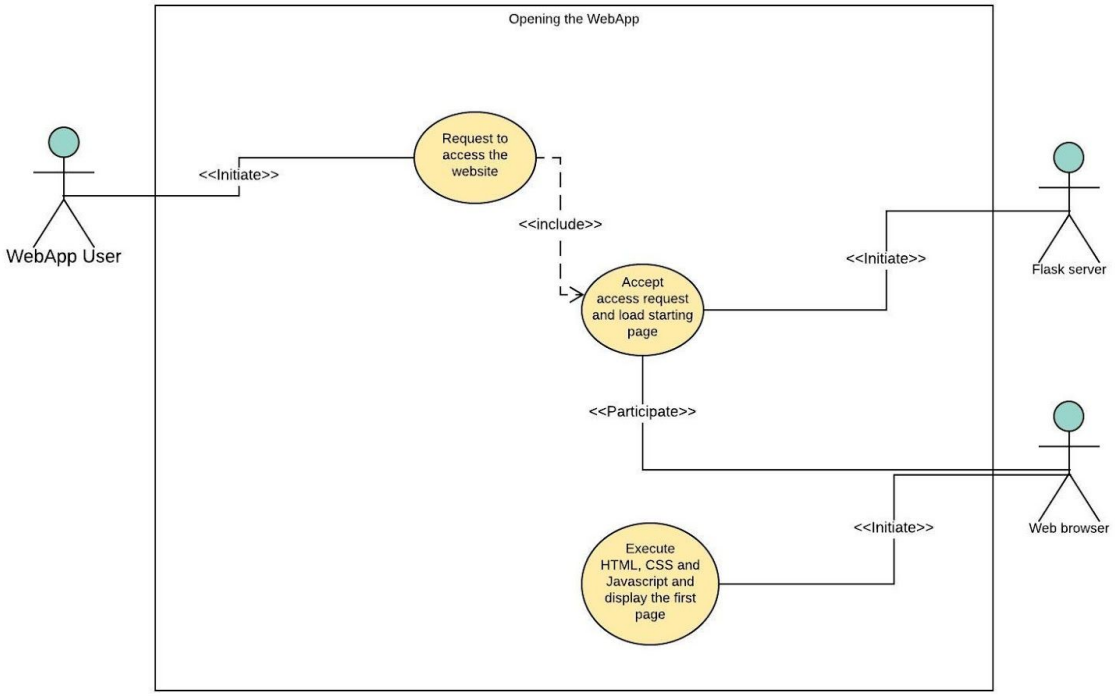
Analysis Models

UML diagram is attached below in this documentation(Fig. 3)

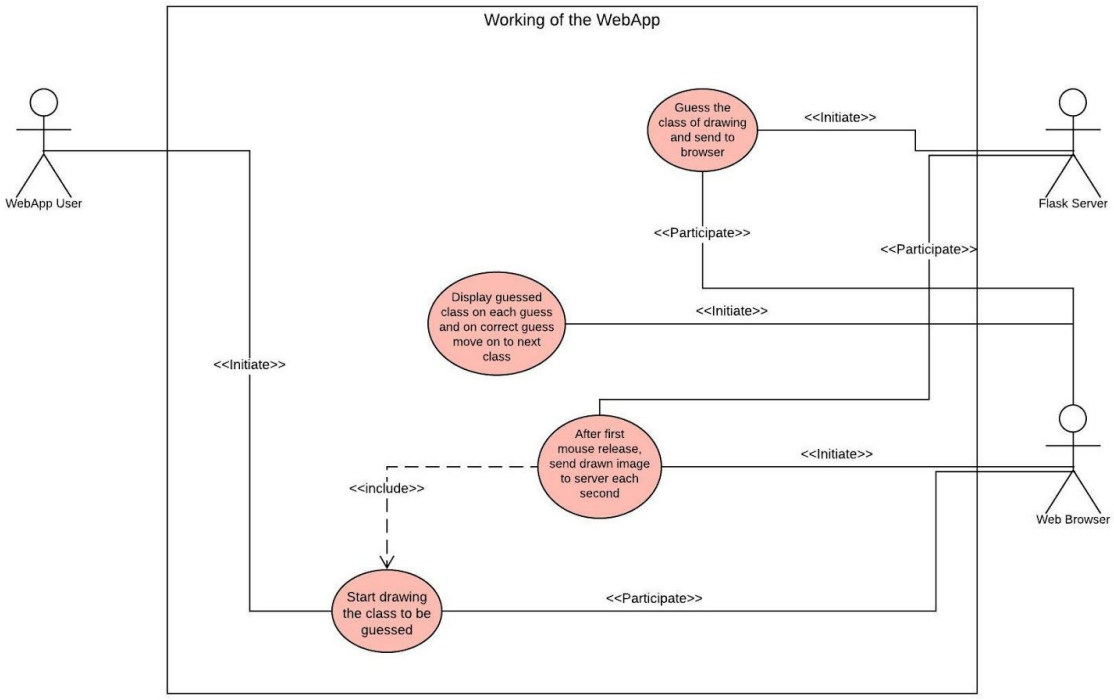
Fig. 3(UML Use Case Diagrams)



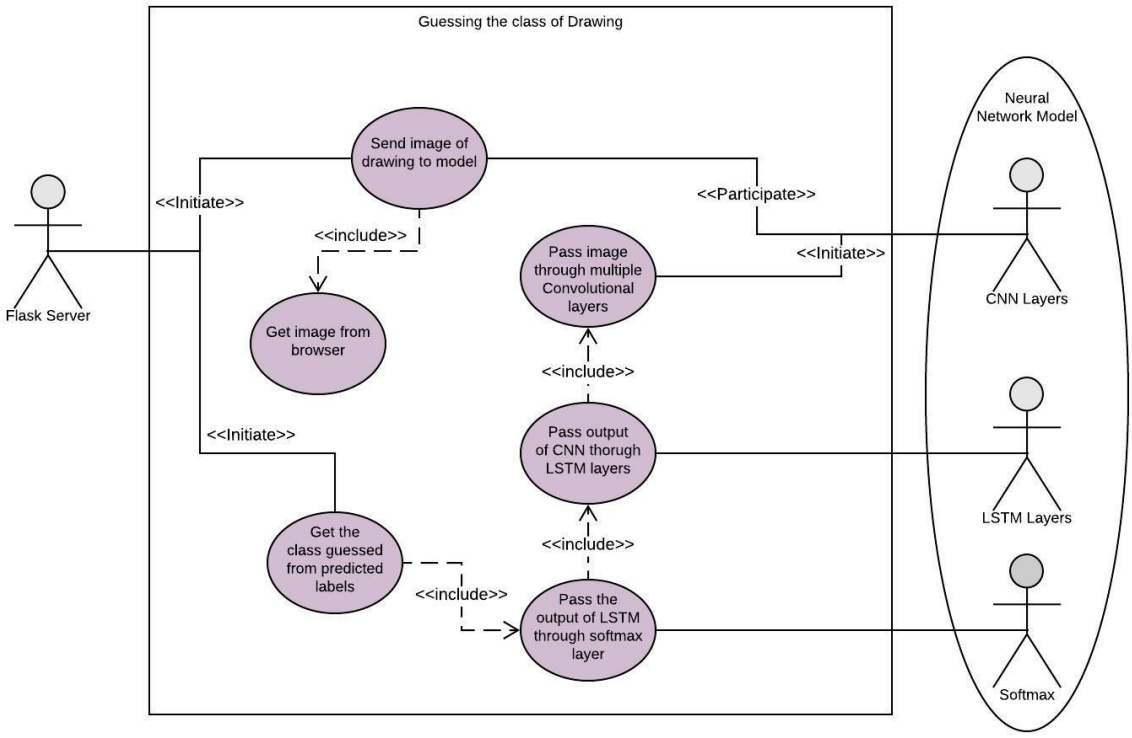
(a)


Opening the WebApp
WebApp User
Request to access the website
<<Initiate>>
<<include>>
Accept access request and load starting page
<<Initiate>>
Flask server
<<Participate>>
Web browser
<<Initiate>>
Execute HTML, CSS and Javascript and display the first page

(b)


Working of the WebApp
WebApp User
Guess the class of drawing and send to browser
<<Initiate>>
Flask Server
<<Participate>>
<<Participate>>
Display guessed class on each guess and on correct guess move on to next class
<<Initiate>>
<<Initiate>>
After first mouse release, send drawn image to server each second
<<Initiate>>
Web Browser
<<include>>
Start drawing the class to be guessed
<<Participate>>

(c)


Guessing the class of Drawing
Neural Network Model
Send image of drawing to model
<<Initiate>>
<<Participate>>
<<include>>
<<Initiate>>
Flask Server
Get image from browser
Pass image through multiple Convolutional layers
CNN Layers
<<include>>
<<Initiate>>
Pass output of CNN thorugh LSTM layers
LSTM Layers
Get the class guessed from predicted labels
<<include>>
<<include>>
Pass the output of LSTM through softmax layer
Softmax

(d)