

# Sore Vision



Phase 2

Project Report – 03/20/2018

**Project Team – #17**

*Team Members*

Raj Anantharaman (#3)

## Table of Contents

Table of Contents .....	2
1. Project Objectives.....	3
a. Significance.....	3
b. Features.....	3
c. Standard Use Case for “Sore Vision” .....	3
d. Sequence Diagram for “Sore Vision” .....	4
2. Approach .....	5
a. Data Sources .....	5
b. Algorithms .....	5
3. Related Work.....	6
4. Application Specification .....	6
a. Iteration 1 .....	6
b. Iteration 2 .....	6
5. Iteration 1 - Implementation .....	7
a. Webpage .....	7
b. Clarifai Custom model for Mouth sore.....	10
6. Iteration 2 - Implementation .....	11
a. Webpage .....	11
b. Random Forest Classifier model for Mouth sore.....	13
7. Project Management.....	14
a. Iteration 1 Report from Zenhub .....	14
b. Iteration 2 Report from Zenhub .....	17
c. Issues/Concerns.....	20
Appendix A – JSON for recreating Use Case Diagram.....	21

## 1. Project Objectives

### a. Significance

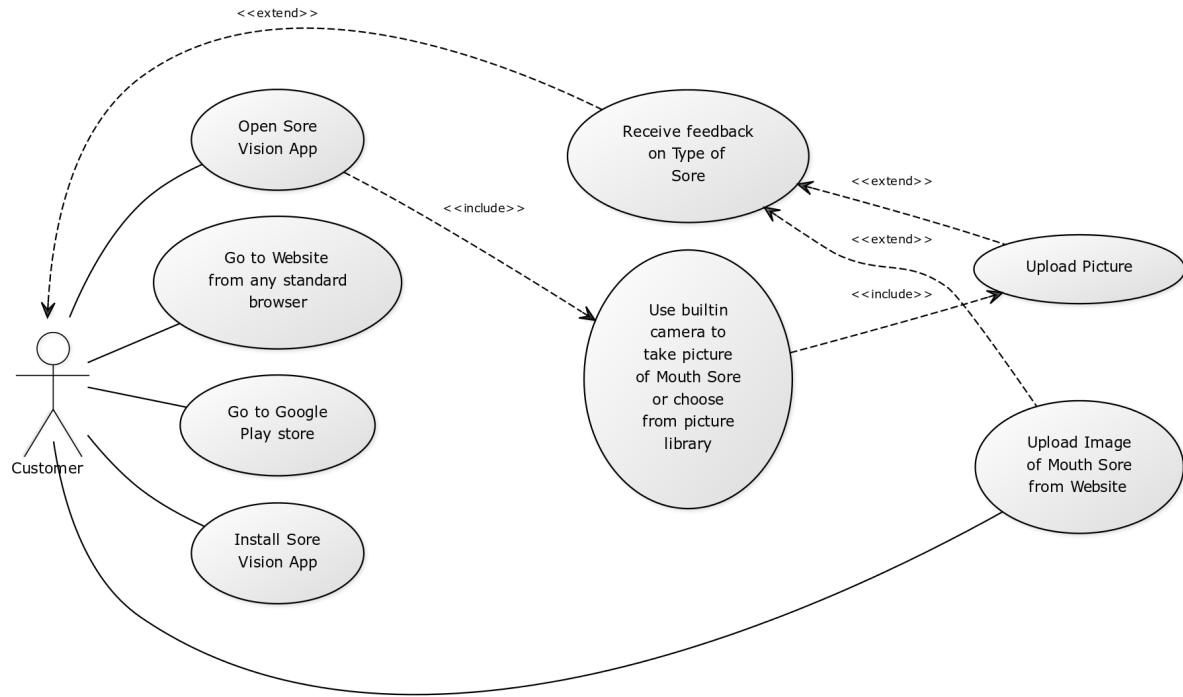
The main objective of “**Sore Vision**” is to provide an easy way for users to perform self-examinations of mouth sores using their Android phone. While it is an understatement that mouth sores do create extreme discomfort and pain for the person, that is not the worst part. One variety of mouth sore, referred to as the “**cold sore**” is highly contagious and an infected person can easily pass on the infection to another person just through skin to skin contact. The other variety of mouth sore, referred to as a “**canker sore**” is also painful but luckily is not contagious. The objective of the “**Sore Vision**” app is to provide immediate feedback to the user on the kind of mouth sore that he/she is battling when presented with an image of the same. Sore Vision uses deep learning to distinguish a cold sore from a canker sore.

### b. Features

The primary features of “Sore Vision” include:

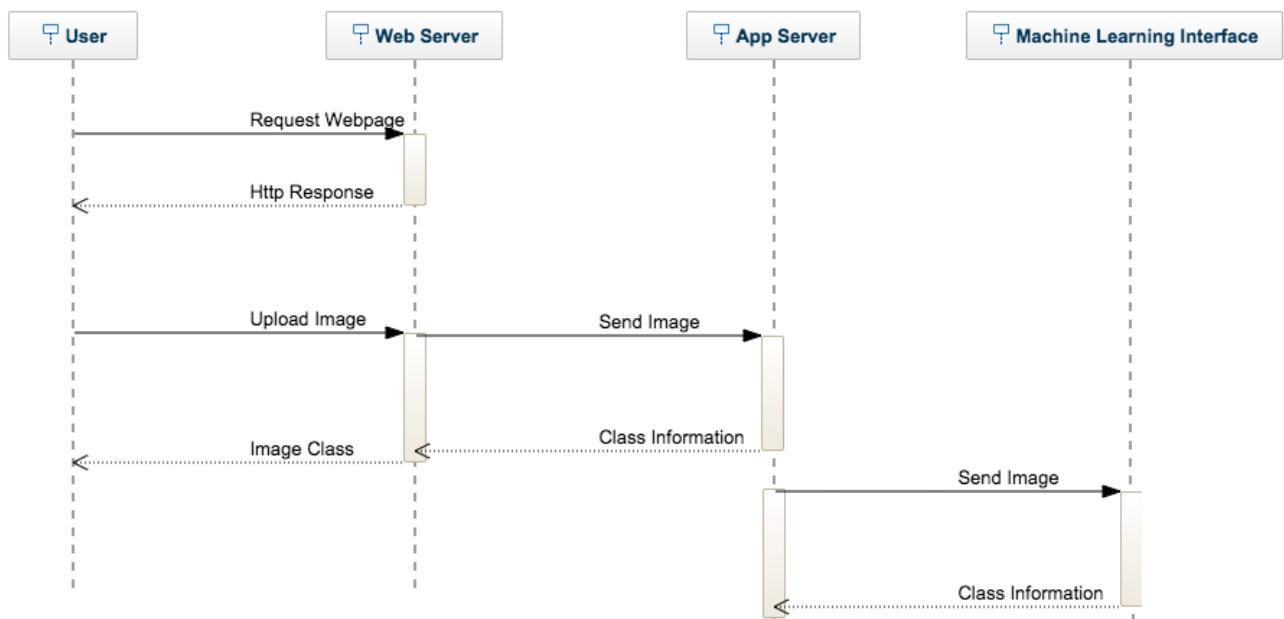
- A model that is trained adequately to identify and classify mouth sores. Three separate models will be designed and developed. First based on Clarifai API, second based on Spark Machine learning and the third based on Deep learning using Tensorflow.
- A web application that provides the user with the ability to upload an image and receive feedback right away through the web UI.
- An installable Android application that allows the user to capture and upload a picture right from within the application.

### c. Standard Use Case for “Sore Vision”



This use case diagram was created using the Use Case diagram creator by Pocketworks and can be accessed at <https://yuml.me/diagram/usecase/draw>. The JSON for creating the use case diagram is given in Appendix A.

#### d. Sequence Diagram for “Sore Vision”



## 2. Approach

When the project has been successfully completed, Sore Vision will be available to the user as a web application as well as an Android app. Image classification will be handled through three separate means.

- 1) External API – We will use Clarifai API for this purpose
- 2) Machine Learning(Spark)
- 3) Deep Learning (Tensor Flow)

During Iteration 1, we focused on finishing one aspect of the user interface and one image classification scheme. At the end of iteration 1, we had a working prototype of the website with the necessary integration with Clarifai API to effectively and efficiently classify the image uploaded by the user.

During Iteration 2, we focused on using Spark machine learning to create a classification model. We will build a simple HTTP server to receive test images from a web client and pass it on to the app server for classifying the images into two classes, Canker sore and Cold sore.

### a. Data Sources

Data for training the model will be mainly sourced from <http://images.google.com>. The goal is to find at least 50 to 75 images that provide close up shots of cold sores and canker sores. With the aid of a dentist, I will classify them into the right classes to begin with. Other than this, there is not much of analytical tasks that needs to be performed on the dataset.

### b. Algorithms

In iteration 1, We will use Clarifai's visual recognition algorithm to achieve our goal. That being said, the models that are available in Clarifai will not be sufficient. During this iteration of the project, I plan on creating a new model and training it with adequate data.

In iteration 2, We will use the Random Forest Classifier algorithm to create a model, save this model and eventually use this model to classify images into the two classes.

### 3. Related Work

A company called i-Nside is using Clarifai API to provide an application that enables doctors to effectively diagnose and treat ear problems. Being a endoscope manufacturer, they already had a library of over 100,000 images that they could readily use to build and train their own custom visual recognition model in Clarifai.

### 4. Application Specification

#### a. Iteration 1

For iteration 1, the application will essentially consist of four layers.

- 1) The UI layer which is a standard HTML5 mobile responsive web page. For this iteration, a few test images are provided with the application for testing. Javascript is used to transfer the image from the UI layer to the web server.
- 2) The web server which receives uploaded images from the web page. The web server has the responsibility of receiving the image, storing a local copy of the image and passing it on to the application server for processing.
- 3) An application server which receives the image from the web server and invokes the Clarifai API. The application server is responsible for calling the “**Mouth-sore**” model and passing the image to the model.
- 4) The custom visual recognition model built into Clarifai.

#### b. Iteration 2

For iteration 2, the application will continue to consist of four layers.

- 1) The UI layer which is a standard HTML5 mobile responsive web page. For this iteration, a few test images are provided with the application for testing. Javascript is used to transfer the image from the UI layer to the web server.
- 2) The web server which receives uploaded images from the web page. The web server has the responsibility of receiving the image, storing a local copy of the image and passing it on to the application server for processing.
- 3) An application server which receives the image from the web server and passes on the image to the Random Forest model.
- 4) The visual recognition model built using the Random Forest classifier algorithm built using Spark MLLib.

In future iterations, the UI layer will be replaced with an android app and the visual recognition model will be replaced with Deep Learning in Tensor Flow.

The use case and the sequence diagram provided in section 1 above provide the application interaction.

## 5. Iteration 1 - Implementation

Implementing Google cardboard and Google conversation API did not make sense for this project as the user will not able to upload an image through speech. Google cardboard might be an option. I will investigate it in the next iterations.

### a. Webpage

Below is a snapshot of the webpage that will be publicly available to the end user.

## Sore Vision

### Sore Vision - Is it a cold sore or a canker sore?

Cold sores are a cluster of blisters that first appear clear then become cloudy. First infection may be inside the mouth, but cold sores generally appear outside the mouth on the lips.

Canker sores are gray or white sores surrounded by a red inflamed area. They occur only inside the mouth, on the tongue or the insides of the cheeks, lips or throat.

Despite their name, cold sores (also known as fever blisters) are not caused by the common cold. Cold sores typically result from a viral infection called herpes simplex virus (HSV). HSV is contagious and may be transmitted by any contact with the blisters until the sore is healed.

[Learn more about Cold sores](#)



Cold sore?

Select an Image:



Response received from the application when the “Cold sore?” button is pushed with the first image. The model is providing feedback that there is a 85.54% chance that the image has a cold sore.

infection called herpes simplex virus (HSV). HSV is contagious and may be transmitted by any contact with the blisters until the sores are healed.

[Learn more about Cold sores](#)



Cold sore?

There is a 85.54% chance  
that this is a cold-sore.

Response received from the application when the “Cold sore?” button is pushed with the Second image (which is clearly a canker sore). The model is providing feedback that there is 98.10% chance that the image has a canker sore.

Despite their name, cold sores (also known as fever blisters) are not caused by the common cold. Cold sores typically result from an infection called herpes simplex virus (HSV). HSV is contagious and may be transmitted by any contact with the blisters until they are healed.

[Learn more about Cold sores](#)

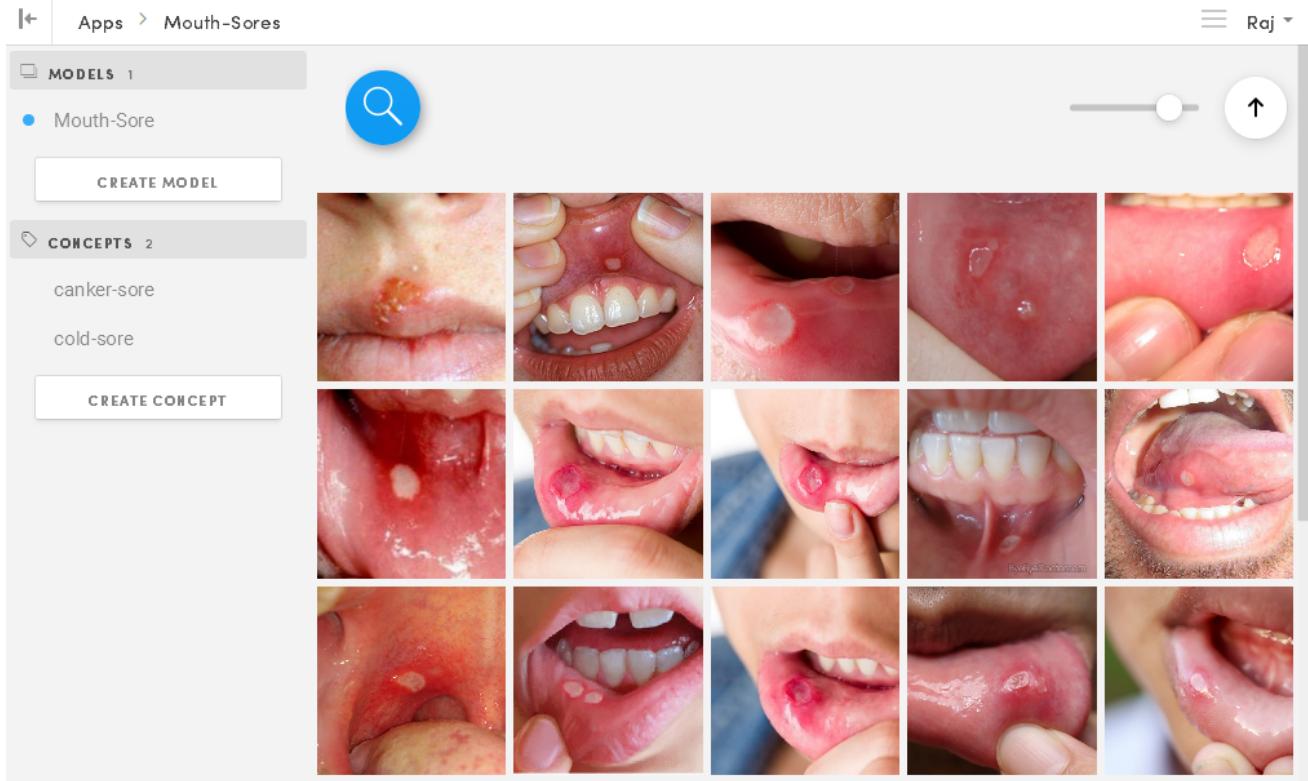


Cold sore?

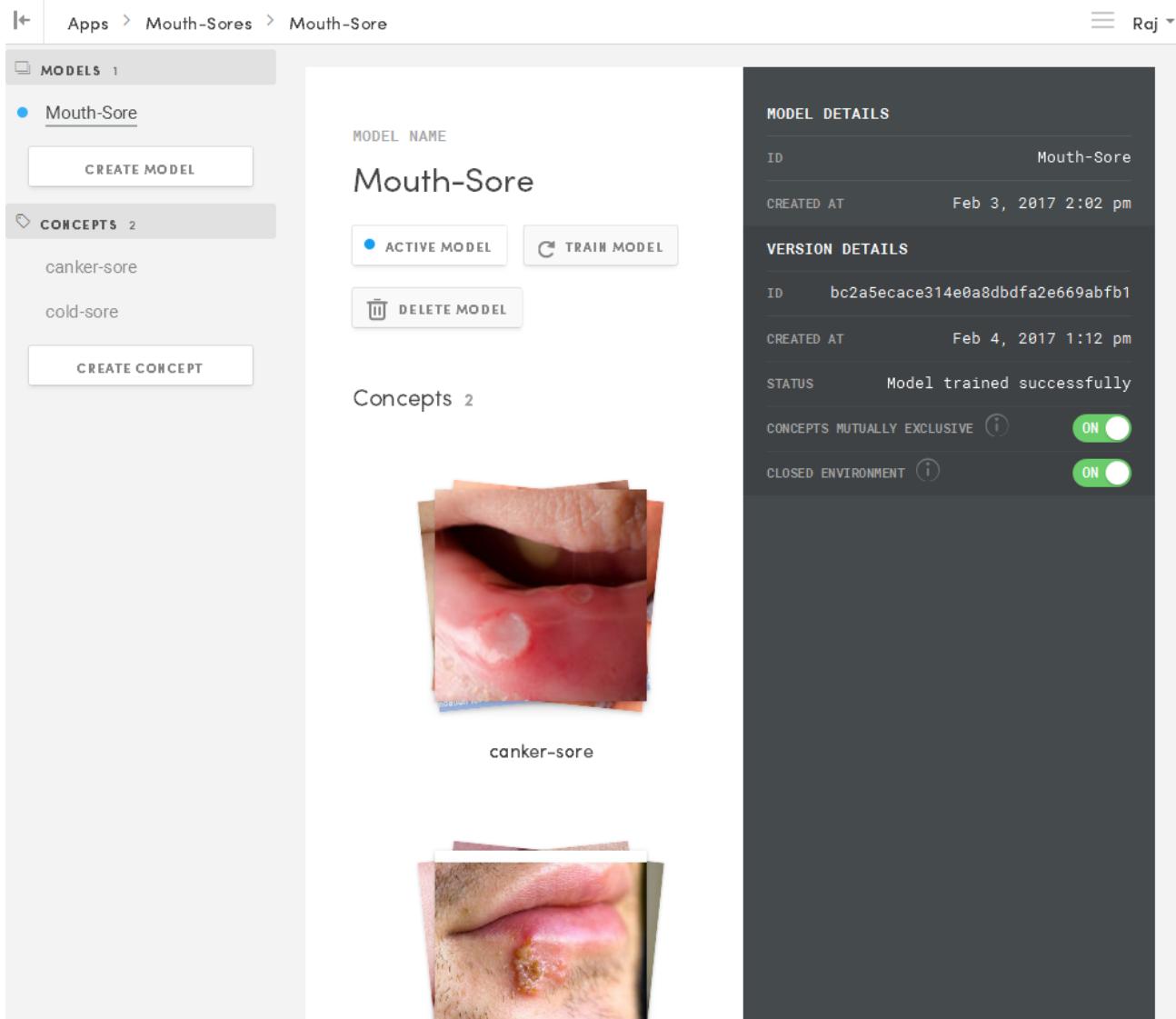
There is a 98.10% chance  
that this is a canker-sore.

### b. Clarifai Custom model for Mouth sore

The custom visual model has 2 concepts and has 75 images. A few are shown below.



The model was trained successfully and having the “Concepts mutually exclusive” checked provided great results.

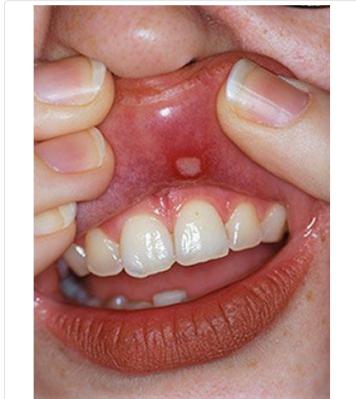


## 6. Iteration 2 - Implementation

### a. Webpage

Below is a snapshot of the webpage that will be publicly available to the end user.

## Mouth Sore Prediction



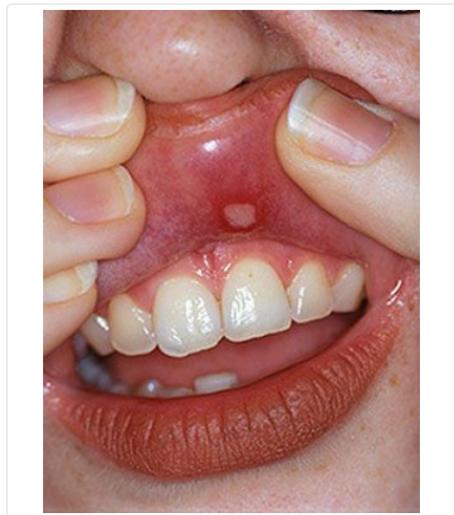
Predict

Select an Image:



Response received from the application when the “Predict” button is pushed with a few test images are below. The model is clearly not very reliable. In both instances below, its prediction is off target.

## Mouth Sore Prediction



Predict

Test image predicted as :  
Coldsore

## Mouth Sore Prediction



Predict

Test image predicted as :  
Coldsore

### b. Random Forest Classifier model for Mouth sore

The model was trained with images and was tested with 3 images for cold sore and 3 for canker sore. Results of the test classification are provided below.

file:/C:/Users/Raj  
Anantharaman/Documents/CS5542/Iteration2/cam\_image\_classification/data/test/Cankersore/shutterstock\_312119459.jpg

Predicting test image : Cankersore as Cankersore

file:/C:/Users/Raj  
 Anantharaman/Documents/CS5542/Iteration2/cam\_image\_classification/data/test/Coldsore/Rach  
 el-Cold-Sore.jpg

Predicting test image : Coldsore as Coldsore

file:/C:/Users/Raj  
 Anantharaman/Documents/CS5542/Iteration2/cam\_image\_classification/data/test/Coldsore/sonja  
 M-day1.jpg

Predicting test image : Coldsore as Cankersore

(1.0,0)  
 (0.0,0)  
 (0.0,0)  
 (1.0,1)  
 (1.0,1)  
 (1.0,1)  
 0.833333333333334

|===== Confusion matrix =====

2.0 1.0  
 0.0 3.0  
 0.833333333333334

An accuracy rate of 83.33% as indicated by the confusion matrix but in reality the model is not as reliable. The clarifai model from iteration 1 is far better than this model.

## 7. Project Management

### a. Iteration 1 Report from Zenhub

Iteration 1 contained 5 issues of varying complexity. The total number of hours estimated for Iteration 1 was 32 hours. All the five issues have been closed. Iteration 1 has been delivered on time.

Below is a snapshot of the Iteration 1 report. The same can be accessed at

<https://github.com/rah59/Project-Sore-Vision/issues#reports?report=burndown&milestoneId=2354945>

The screenshot shows a GitHub project board titled "rah59 / Project-Sore-Vision". The top navigation bar includes links for Code, Issues (0), Pull requests (0), Boards (selected), Reports, Projects (0), and Wiki. Below the navigation are filters for Show one, Labels, Milestones, Assignees, and Epics. A "filters" link is also present. The board itself has three columns: "Progress" (0), "Done" (0), and "Closed" (5+). The "Done" and "Closed" columns each contain 5 items, all of which are labeled "Closed" and have a small circular icon with a checkmark. The items are described as follows:

- Project-Sore-Vision #5: Create an app server to connect to Clarifai (Iteration 1)
- Project-Sore-Vision #4: Create an HTTP Server to receive image from Web Client (Iteration 1)
- Project-Sore-Vision #3: Create Web UI to allow users to upload images (Iteration 1)
- Project-Sore-Vision #2: Create new visual model in Clarifai (Iteration 1)
- Project-Sore-Vision #1

This repository Search Pull requests Issues Gist ToDo + ⚙️

rah59 / Project-Sore-Vision ⚙️ Watch 0 Star 0 Fork 0

Code Issues 0 Pull requests 0 Boards Reports Projects 0 Wiki

**Burndown** Velocity tracking

## Iteration 1

This marks the completion of iteration 1

Labels Hide Pull Requests Burn Pipelines

Start: Feb 1, 2017 Edit Due: Feb 27th, 2017 Edit

weekends ideal completed

32 Total Story Points  
32 Completed Story Points / 0 Remaining Story Points

5 Total Issues and Pull Requests  
5 Completed Issues and PRs / 0 Remaining Issues and PRs

Remaining Issues and Pull Requests		Story points
No issues to show based on your filters.		

Completed Issues and Pull Requests		Story points
Collect image data for cold sores and canker sores Project-Sore-Vision #1	(3)	
Create new visual model in Clarifai Project-Sore-Vision #2	(8)	
Create Web UI to allow users to upload images Project-Sore-Vision #3	(8)	
Create an HTTP Server to receive image from Web Client Project-Sore-Vision #4	(5)	
Create an app server to connect to Clarifai Project-Sore-Vision #5	(8)	

## b. Iteration 2 Report from Zenhub

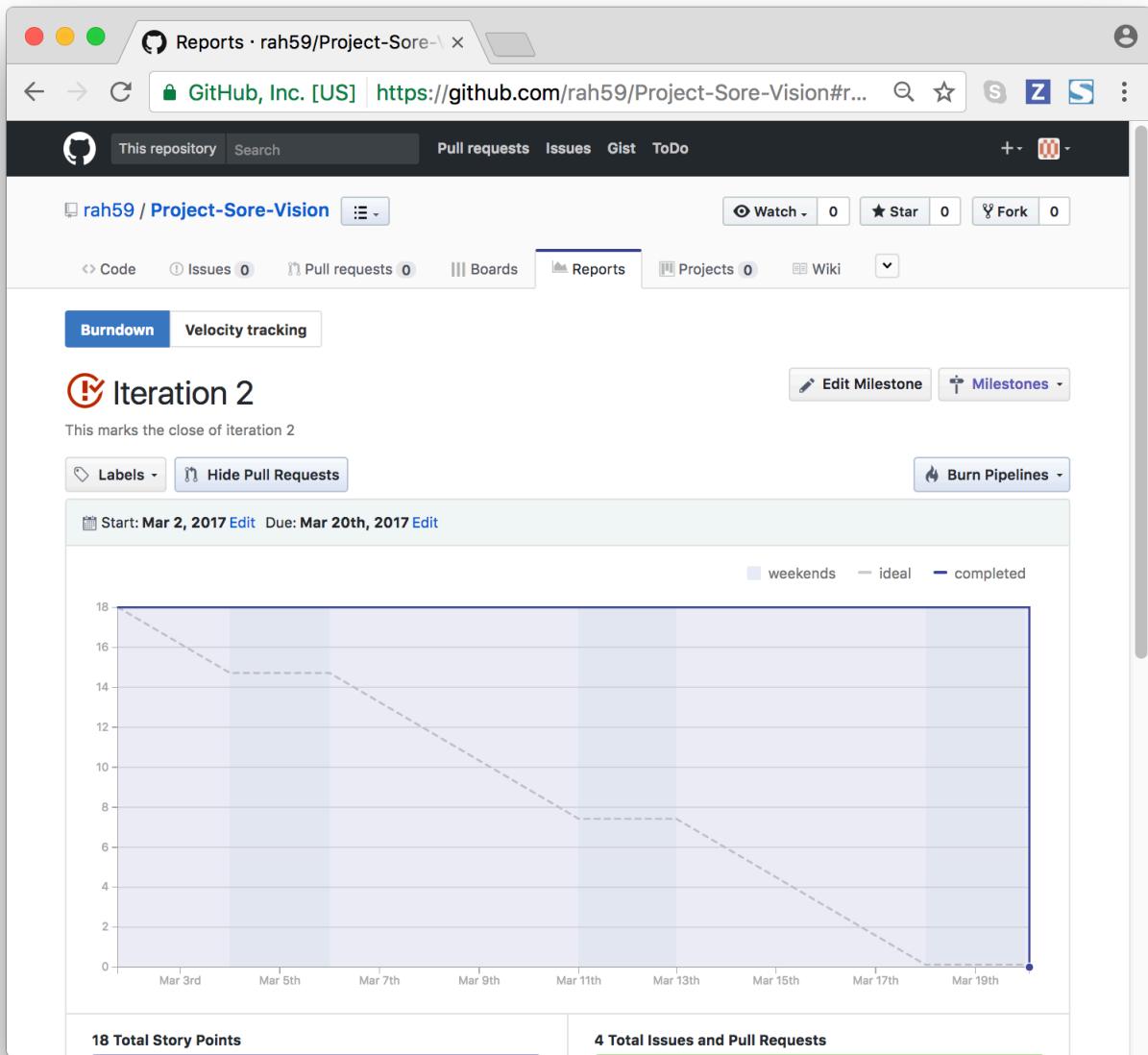
Iteration 2 contained 4 enhancements. The total number of hours estimated for Iteration 2 was 18 hours. All the four enhancements have been closed. Iteration 2 has been delivered on time and closed now.

The screenshot shows the Zenhub interface for the repository 'rah59 / Project-Sore-Vision'. The top navigation bar includes links for 'Pull requests', 'Issues', 'Gist', and 'ToDo'. Below the navigation is a search bar and buttons for 'Watch', 'Star', and 'Fork'. The main area features a 'Boards' tab selected, followed by 'Reports', 'Projects', and 'Wiki'. A 'New Issue' button is visible on the right.

Under the boards section, there are several columns: 'New Issues' (0), 'In Progress' (0), 'Icebox' (0), 'Backlog' (0), and 'Review/QA' (0). To the right of these columns is a large 'Done' column containing 18 closed issues, each with a detailed description and a blue 'enhancement' label.

- Project-Sore-Vision #9:** Create an HTTP Server to receive image from Web Client  
Iteration 2
- enhancement**
- Project-Sore-Vision #8:** Repurpose the Web UI to allow users to upload images  
Iteration 2
- enhancement**
- Project-Sore-Vision #7:** Create an app server that uses a Spark random forest model to classify image  
Iteration 2
- enhancement**
- Project-Sore-Vision #6:** Create Training and Testing dataset of Cold sores and Canker sores  
Iteration 2
- enhancement**

A 'Load more issues...' button is located at the bottom of the issue list. A dashed box labeled 'Add a Pipeline ...' is visible on the right side of the board.



Iteration 2 Report – Project “Sore Vision”

<b>18 Total Story Points</b>	<b>4 Total Issues and Pull Requests</b>
18 Completed Story Points / 0 Remaining Story Points	4 Completed Issues and PRs / 0 Remaining Issues and PRs

Remaining Issues and Pull Requests	Story points
No issues to show based on your filters.	

Completed Issues and Pull Requests	Story points
⌚ Create Training and Testing dataset of Cold sores and Canker sores [enhancement] Project-Sore-Vision #6	(3)
⌚ Create an app server that uses a Spark random forest model to classify image [enhancement] Project-Sore-Vision #7	(8)
⌚ Repurpose the Web UI to allow users to upload images [enhancement] Project-Sore-Vision #8	(2)
⌚ Create an HTTP Server to receive image from Web Client [enhancement] Project-Sore-Vision #9	(5)

Iteration 2 had 18 days while Iteration 1 had 27 days. The velocity chart below indicates the velocity with which work is being completed.



Milestone	Start Date	End Date	Duration	Story points
Iteration 1	Feb 1, 2017	Feb 27, 2017	27 days	32 of 32 completed
Iteration 2	Mar 2, 2017	Mar 20, 2017	19 days	18 of 18 completed
Iteration 3	No start date	Apr 24, 2017	--	Not estimated
Iteration 4	No start date	May 1, 2017	--	Not estimated
Iteration 5	No start date	May 5, 2017	--	Not estimated

### c. Issues/Concerns

The project requires the incorporation of Google cardboard and/or Conversational UI. However, there is no direct way to incorporate Conversational UI as it is nearly impossible to upload an image through speech. The use of Google cardboard will be investigated in later iterations. An android app will need to be developed in Iteration 3 to capture an image and not rely on test images alone.

## Appendix A – JSON for recreating Use Case Diagram

```
{ "customisations": "plain;dir:TB", "type": "usecase", "dsl": "[Customer] - (Go to Website from any standard browser), [Customer] - (Upload Image of Mouth Sore from Website), (Receive Feedback on Type of Sore) < (Upload Image of Mouth Sore from Website), [Customer] - (Go to Google Play store), [Customer] - (Install Sore Vision App), [Customer] - (Open Sore Vision App), (Open Sore Vision App) > (Use builtin camera to take picture of Mouth Sore or choose from picture library), (Use builtin camera to take picture of Mouth Sore or choose from picture library)>(Upload Picture), (Receive feedback on Type of Sore) <(Upload Picture), [Customer] < (Receive feedback on Type of Sore)", "digest": "625c2361", "file_only": false, "mime": "image/png", "extension": "png" }
```