Life Cycle Plan (LCP)

Image Processing Platform

Team 04

Name	First Role	Second Role	Third Role
Hao Wu	Requirements Engineer	Software Architect	Implementer
Junran Liu	Operational Concept Engineer	Software Architect	Implementer, Trainer
Meiyi Yang	Project Manager	Life Cycle Plan	Implementer
Vinny DeGenova	IIV & V	Quality Focal Point	Implementer
Xiangchen Zhao	Life Cycle Plan	Prototyper	Implementer
Xinhui Liu	Feasibility Analyst	Operational Concept Engineer	Implementer, Tester
Yifan Liu	Prototyper	Requirements Engineer	Implementer

12/04/2016

Version History

Date	Author	Version	Changes made	Rationale
10/08/16	Meiyi Yang	1.0	Wrote a draft for DCR ARB	For draft of DC package
10/14/16	Meiyi Yang	1.1	Updated project plan	For DC package
12/04/16	Meiyi Yang	2.0	Add Iteration Plan	For As Built Package

Table of Contents

Table of Tables

Table of Figures

1. Introduction

1.1 Purpose of the LCP

The LCP will identify the milestones of the projects, identify responsibilities and skills, and also estimate project effort and schedule to make sure everybody knows what need to be done by when or by who.
"The purpose of a development project's LCP is to:

- Serve as a basis for monitoring and controlling the project's progress
- Help make the best user of people and resources throughout the system's life cycle
- Provide evidence to other key stakeholders that the developers have thought through the major life cycle issues in advance."

1.2 Status of the LCP

The status of the LCP is currently Foundation Phase version number 1.1. This version is modified after the DCR ARB (10/10/2016) from the version 1.0: draft for DC package. The is the second version in the Foundation Phase, which will be the final delivery version for the DC package.

1.3 Assumptions

- The duration of the project is 12 weeks from August to December.
- All of our team member will work on this project for one semester.
- Our team has 7 team members, including 6 local students and 1 remote student.

2. Milestones and Products

2.1 Overall Strategy

Because the duration of this project is 12 weeks. Our team is following the NDIs-intensive (Non-Developmental Items) Process. Here is the information for each phase in our project.

Exploration phase:

Duration: 08/22/2016 - 09/26/2016
Concept: Analyze current system; Explore alternatives; Assess and plans to mitigate risks; Identify responsibilities and skills; Detail project plan; Record project individual effort; Record project progress;
Deliverables: Client Interaction Report; Win Conditions Report; Progress Report; Project Plan; Risk Report
Milestone: Valuation Commitment Review
Strategy: One Incremental Commitment Cycle, Win-Win negotiation session.

Valuation phase:

Duration: 09/27/2016 - 10/05/2016
Concept: Identify objectifies, constraints and priorities; Develop operational concept; Explore alternatives; Provide project feasibility evidence; Prototyping; Assess and plans to mitigate risks; Plan and manage project; Perform WinWin negotiation; Define quality and configuration policy; Shaper;
Deliverables: Top Risk Prototype; Progress Report; Project Plan; Risk Report;
Milestone: Foundation Commitment Review
Strategy: One Incremental Commitment Cycle; Win-Win negotiation session;

Foundation phase:

Duration: 10/05/2016 - 10/18/2016
Concept: Assess project status; Plan and manager project; Manage project quality; Prototyping; Develop software architecture;
Deliverables: DC package; DCR ARB; Progress Report; Project Plan; Risk Report
Milestone: Development Commitment Review
Strategy: One Incremental Commitment Cycle

Development phase:

Duration: 10/19/2016 - 12/07/2016
Concept: Construction iteration; Transition iteration;
Deliverables: CCD Report; TRR; Final Deliverables; Product Release; On-Campus Technical Debt Report; QFP Technical Debt Report; Progress Report; Project Plan; Risk Report
Milestone: Operation Commitment Review; CCD (Core Capability Drivethrough); TRR (Transition Readiness Review)

Strategy: One Incremental Commitment Cycle

2.2 Project Deliverables

The following tables are the project deliverables in each phase for our project and its due date, format, and medium.

2.2.1 Exploration Phase

Table 1: Artifacts Deliverables in Exploration Phase

Artifact	Due date	Format	Medium
Client Interaction Report	09/16/2016	.doc, .pdf	Soft copy
Win Conditions Report	09/26/2016	.pdf	Soft copy
Risk Report	Bi-weekly	.xls	Soft copy
Project Plan	Bi-weekly	.mpp	Soft copy
Progress Report	Bi-weekly	.xls	Soft copy
Jira	Every Monday	Jira ticket	Jira website

2.2.2 Valuation Phase

Table 2: Artifacts Deliverables in Valuation Phase

Artifact	Due date	Format	Medium
Top Risk Prototype	09/30/2016	.pptx	Soft copy
Risk Report	Bi-weekly	.xls	Soft copy
Project Plan	Bi-weekly	.mpp	Soft copy
Progress Report	Bi-weekly	.xls	Soft copy
Jira	Every Monday	Jira ticket	Jira website

2.2.3 Foundations Phase

Table 3: Artifacts Deliverables in Foundation Phase

Artifact	Due date	Format	Medium
DCR ARB Presentation	10/10/2016	.pptx	Soft copy
Draft for DC Package	10/10/2016	.doc, .pdf	Soft copy
DC Package	10/17/2016	.doc, .pdf	Soft copy
Risk Report	Bi-weekly	.xls	Soft copy
Project Plan	Bi-weekly	.mpp	Soft copy
Progress Report	Bi-weekly	.xls	Soft copy
Jira	Every Monday	Jira ticket	Jira website

2.2.4 Development Phase

Table 4: Artifacts Deliverables in Development Phase

Artifact	Due date	Format	Medium
Risk Report	Bi-weekly	.xls	Soft copy
Project Plan	Bi-weekly	.mpp	Soft copy
Progress Report	Bi-weekly	.xls	Soft copy
On-Campus Technical Debt Report	Bi-weekly	.xls	Soft copy
QFP Technical Debt Report	Bi-weekly	.xls	Soft copy
Jira	Every Monday	Jira ticket	Jira website
CCD (Core Capability Drivethrough)	11/16/2016	.pptx	Soft copy
TRR (Transition Readiness Review)	11/30/2016	.pdf, .doc	Soft copy
Final Deliverables	12/05/2016	code	Soft copy
Project Release	12/07/2016	code	Soft copy

3. Responsibilities

3.1 Project-specific stakeholder's responsibilities

Our clients in the project includes Eder Figueroa and Ripple Goyal, who are both from Armorway. Our developers are Hao Wu, Junran Liu, Meiyi Yang, Xiangchen Zhao, Xinhui Liu, and Yifan Liu. Vinny DeGenova, who is our remote student, will be as a role of IIV & V and Quality Focal Point. The details of the responsibilities for above stakeholders will be list in the Table 4.

3.2 Responsibilities by Phase

Table 5: Stakeholder's Responsibilities in each phase

Team Member / Role	Primary / Secondary Responsibility				
	Exploration	**Valuation**	**Foundations**	**Development-Construction Iteration**	**Development-Transition Iteration**
Name: Hao Wu Role: Requirements Engineer; Software Architect; Developer;	**Primary Responsibility** WinWin Negotiation; Capture the requirement;	**Primary Responsibility** WinWin Negotiation; Capture the requirement; Win Conditions Report	**Primary Responsibility** Define requirement; **Secondary Responsibility** Develop Software Architecture	**Third Responsibility** Implement the System	**Third Responsibility** Transition the system;
Name: Junran Liu Role: Operational Concept Engineer; Software Architect; Developer;	**Primary Responsibility** Analyze the current system	**Primary Responsibility** Identify objectives, constraints and Priorities	**Primary Responsibility** Identify objectives, constraints and Priorities **Secondary Responsibility** Analyze the architect;	**Third Responsibility** Implement the System	**Third Responsibility** Transition the system

Name: Meiyi Yang Role: Project Manager; Life Cycle Plan; Tester;	**Primary Responsibility** Project Plan and manage; **Secondary Responsibility**; Identify team member's skills;	**Primary Responsibility** Project Plan and manage;	**Primary Responsibility** Project Plan and manage; **Secondary Responsibility** Assess Life Cycle Content	**Primary Responsibility** Project Plan and manage; **Third Responsibility** Testing	**Primary Responsibility** Project Plan and manage;
Name: Vinny DeGenova Role: IIV & V; Quality Focal Point; Developer	**Primary Responsibility** Interact with client;	**Primary Responsibility** Verify and Validate Work Products	**Primary Responsibility** IIV & V presentation **Secondary Responsibility** QFP presentation	**Third Responsibility** Implement the System	**Third Responsibility** Transition the system
Name: Xiangchen Zhao Role: Life Cycle Plan; Prototyper; Developer;	**Primary Responsibility** Project Plan and manage; Analyze project risk and project progress; Manage Jira	**Primary Responsibility** Project Plan; Analyze project risk and project progress; **Secondary Responsibility** Prototyping;	**Primary Responsibility** Plan the project **Secondary Responsibility** Prototyping; Product Demo;	**Primary Responsibility** Project Plan and manage; **Third Responsibility** Implement the System	**Third Responsibility** Transition the system
Name: Xinhui Liu Role: Feasibility Analyst; Operational Concept Engineer; Trainer;	**Primary Responsibility** Explore Alternatives; **Secondary Responsibility** Identify objectives, constraints and Priorities	**Primary Responsibility** Explore Alternatives;	**Primary Responsibility** Assess Feasibility Evidence	**Third Responsibility** Implement the System;	**Third Responsibility** Provide Training
Name: Yifan Liu Role: Prototyper; Requirements Engineer	**Primary Responsibility** Evaluate COTS; **Secondary Responsibility** Analyze current system;	**Primary Responsibility** Prototyping;	**Primary Responsibility** Prototyping;	**Third Responsibility** Implement the System	**Third Responsibility** Transition the system

Name: **Eder Figueroa** **Role:** Client	**Primary Responsibility** Convey project	**Primary Responsibility** Win-win negotiation	**Primary Responsibility** Verify our work	**Primary Responsibility** CCD; Assess Development Iteration;	**Primary Responsibility** Verify out work

3.3 Skills

Table 6: Team Member Skills

Team members	Role	Skills
Hao Wu	Requirements Engineer; Software Architect;	**Current Skills:** Python, Django, **Required Skills:** MySQL
Junran Liu	Operational Concept Engineer; Software Architect	**Current Skills:** Python, JQuery, ReactJS **Required Skills:** Google TensorFlow
Meiyi Yang	Project Manager; Life Cycle Plan	**Current Skills:** C++, Python, HTML, CSS **Required Skills:** Google TensorFlow
Vinny DeGenova	IIV & V; Quality Focal Point	**Current Skills:** Python **Required Skills:**
Xiangchen Zhao	Life Cycle Plan; Prototyper	**Current Skills:** C++, Python, **Required Skills:** Google TensorFlow
Xinhui Liu	Feasibility Analyst; Operational Concept Engineer	**Current Skills:** HTML, CSS, JavaScript **Required Skills:** ReactJS
Yifan Liu	Prototyper; Requirements Engineer	**Current Skills:** C/C++, Java, Python, Scala, SQL, ReactJS **Required Skills:** Django

4. Approach

4.1 Monitoring and Control

The approach we are using in monitoring and controlling our project is Progress Report, Project plan and Github.

4.1.1 Closed Loop Feedback Control

Our team use Google Drive for temporary file storage to review the work. The peer review meeting will be held once or twice a week including on-campus meeting and remote meeting.

4.1.2 Reviews

Our team has one fixed time meeting with our client on Thursday afternoon. We will also hold a group review in this meeting period. Except this meeting, we will also conduct a peer review after the CSCI577a class on Monday, Wednesday, Friday if it is needed. A remote peer review will be held by GoToMeeting and Github if it is emergent.

4.2 Methods, Tools and Facilities

Table 7: Methods, Tools and Facilities

Tools	Usage	Provider
Github	A version control system	Github
Google Drive	A sharing folder for reviewing the documents.	Google
Jira	A tool to create and track effort for each team member	USC
Winbook	A tool to identify the win conditions	USC
GoToMeeting	A remote meeting software	Citrix
FaceTime	For communication with our client	Apple
COCOMO 2	A tool for resource estimation	USC
Microsoft Office	A software to plan the project	Microsoft
Slack	Group chat software for communication with clients	Slack
WeChat	Group chat software for communication with group member	Tencent
Gmail	Formal communication or report for the project	Google

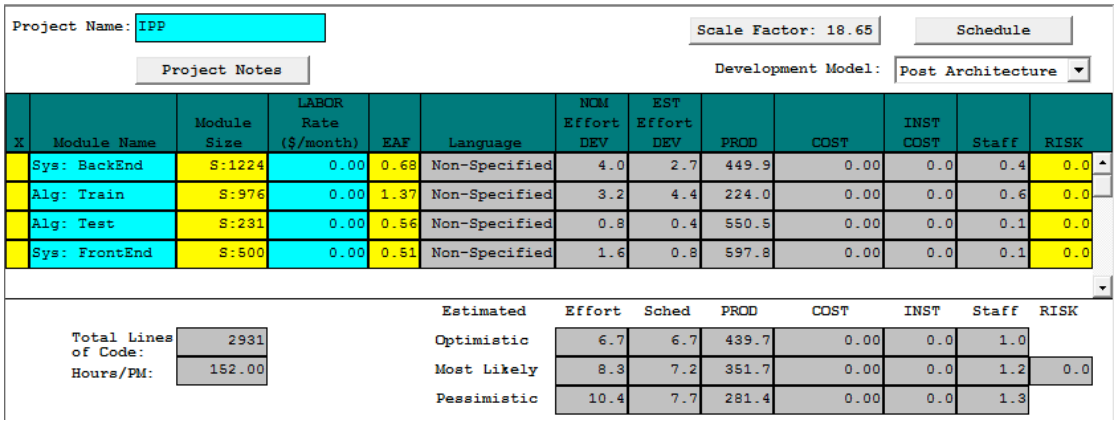
5. Resources

Identify the following information in order to estimate the software cost:

- Method: COCOMO II
- Estimated Effort: 7 team members at 20 hours/week for 12 weeks
- Budget information: $0
- Project duration: 12 weeks
- Component modules in the project: System (Back-End); System (Front-End); Algorithm (Training Phase); Algorithm (Testing Phase)
- Programming language used: Python, HTML, CSS, JavaScript, ReactJS

Here is the estimation result and all the parameters setting:

Figure 1: COCOMO II Estimation Result



Project Name: IPP | Scale Factor: 18.65 | Schedule | Project Notes | Development Model: Post Architecture

X	Module Name	Module Size	LABOR Rate ($/month)	EAF	Language	NOM Effort DEV	EST Effort DEV	PROD	COST	INST COST	Staff	RISK
	Sys: BackEnd	S:1224	0.00	0.68	Non-Specified	4.0	2.7	449.9	0.00	0.0	0.4	0.0
	Alg: Train	S:976	0.00	1.37	Non-Specified	3.2	4.4	224.0	0.00	0.0	0.6	0.0
	Alg: Test	S:231	0.00	0.56	Non-Specified	0.8	0.4	550.5	0.00	0.0	0.1	0.0
	Sys: FrontEnd	S:500	0.00	0.51	Non-Specified	1.6	0.8	597.8	0.00	0.0	0.1	0.0

Total Lines of Code: 2931
Hours/PM: 152.00

Estimated	Effort	Sched	PROD	COST	INST	Staff	RISK
Optimistic	6.7	6.7	439.7	0.00	0.0	1.0	
Most Likely	8.3	7.2	351.7	0.00	0.0	1.2	0.0
Pessimistic	10.4	7.7	281.4	0.00	0.0	1.3	

Table 8: COCOMOII Scale Driver

Scale Driver	Value	Rationale
PREC	LOW	Though our team has experience on deep learning and image processing algorithm. We still need a long time to learn Google TensorFlow.
FLEX	NOM	All the module are independent and flexible.
RESL	NOM	The architecture of the project has been designed. And the most of the risks in the system integration have been reduced by prototype development.
TEAM	NOM	Our team members have enough experience to cooperate and develop on this project.
PMAT	HIGH	Our project is using a deep learning framework Google TensorFlow. This framework is mature enough and we have developed a prototype on this.

Table 9: COCOMOII Cost Driver (System: Back-End)

Cost Driver	Value	Rationale
RELY	LOW	Back-end module is mostly independent from the algorithm module.
DATA	HIGH	The image dataset and model storage will happen in this module.
DOCU	LOW	This module do not need a lot of documentation.
CPLX	LOW	This module is for data storage and query. The complexity is low.
RUSE	LOW	The reusability of the data query will not happen frequently.
TIME	NOM	There is no certain time constraint in this module.
STOR	HIGH	The storage constraint depends on the image dataset storage.
PVOL	NOM	The platform using here is Unix.
ACAP	HIGH	Our team members have a high analyst capability.
PCAP	HIGH	Our team members have a high programming capability.
PCON	NOM	2 of our team members will focus on this part.
APEX	LOW	Our team members have little experience in Django, which is the framework for the back-end system. Our team members have enough experience on MySQL and Python.
LTEX	NOM	Our team members have enough experience on SQL and Python.
PLEX	NOM	Our team members have enough experience on UNIX.
TOOL	NOM	Django and MySQL will be used in this module for system integration and data storage. All tools in this module are not hard.
SITE	NOM	All the development will happen in near USC campus.
SCED	NOM	No expected acceleration or expansion

Table 10: COCOMOII Cost Driver (Algorithm: Model Retraining Phase)

Cost Driver	Value	Rationale
RELY	NOM	The training phase depends on the dataset uploading.
DATA	HIGH	The processing of the image dataset will happen here.
DOCU	NOM	This module needs some documentation for the algorithm setting and dataset selection.
CPLX	HIGH	This module is for training the image dataset. The complexity is very high.
RUSE	HIGH	The reusability of the training will happen frequently.
TIME	NOM	This module needs a long time training, but the constraint time of the training is not very hard to implement.
STOR	HIGH	The storage constraint depends on the image dataset storage.
PVOL	NOM	The platform using here is Unix.
ACAP	HIGH	Our team members have a high analyst capability.
PCAP	HIGH	Our team members have a high programming capability.
PCON	NOM	3 of our team members will focus on this part.
APEX	LOW	Our team members have little experience in Google TensorFlow.
LTEX	NOM	Our team members have enough experience on Python.
PLEX	NOM	Our team members have enough experience on UNIX.
TOOL	NOM	Google TensorFlow and Python will be used in this module for training phase.
SITE	NOM	All the development will be taken place in near USC campus.
SCED	NOM	No expected acceleration or expansion

Table 11: COCOMOII Cost Driver (Algorithm: Image Recognition Phase)

Cost Driver	Value	Rationale
RELY	NOM	The testing phase depends on the training phase and front-end uploading image.
DATA	LOW	This phase only needs one image data per time.
DOCU	LOW	This module needs little documentation.
CPLX	LOW	This module is for testing the deep learning algorithm. The complexity is low.
RUSE	LOW	The reusability of the testing module will not happen frequently.
TIME	NOM	This module doesn't need a long time. But has a short time constraint.
STOR	NOM	This module doesn't need a large data storage. But all the data storage should be cached.
PVOL	NOM	The platform using here is Unix.
ACAP	HIGH	Our team members have a high analyst capability.
PCAP	HIGH	Our team members have a high programming capability.
PCON	NOM	3 of our team members will focus on this part.
APEX	LOW	Our team members have little experience in Google TensorFlow.

LTEX	NOM	Our team members have enough experience on Python.
PLEX	NOM	Our team members have enough experience on UNIX.
TOOL	NOM	Google TensorFlow and Python will be used in this module for training phase.
SITE	NOM	All the development will be taken place in near USC campus.
SCED	NOM	No expected acceleration or expansion

Table 12: COCOMOII Cost Driver (System: Front-End)

Cost Driver	Value	Rationale
RELY	LOW	Front-end module is mostly independent from the algorithm module.
DATA	LOW	This module doesn't need too much data.
DOCU	NOM	This module needs some documentation for User Interface.
CPLX	NOM	This module is for User Interface. The complexity is not very high.
RUSE	NOM	The reusability of the user interface will happen frequently.
TIME	NOM	There is no certain time constraint in this module.
STOR	NOM	There is no certain storage constraint in this module.
PVOL	NOM	The platform using here is Unix.
ACAP	HIGH	Our team members have a high analyst capability.
PCAP	HIGH	Our team members have a high programming capability.
PCON	NOM	2 of our team members will focus on this part.
APEX	NOM	Our team members have enough experience on the Front-End framework.
LTEX	HIGH	Our team members have enough experience on the HTML, CSS, JavaScript, but little experience ReactJS and integration with back-end.
PLEX	NOM	Our team members have enough experience on UNIX.
TOOL	NOM	All the tools including ReactJS in this module are not hard.
SITE	NOM	All the development will happen in near USC campus.
SCED	NOM	No expected acceleration or expansion

6. Iteration Plan

6.1 Plan

6.1.1 Capabilities to be implemented

Table 13: Construction iteration capabilities to be implemented

ID	Capability	Description	Priority	Iteration
OC-01	Upload images	Users are able to upload images from local dictionary.	HIGH	2
OC-02	Preprocess images	The pipeline can preprocess images.	MED	1
OC-03	Add topic	Trainers could add new topic.	LOW	1
OC-04	Train/Retrain the model	The pipeline could retrain the model.	HIGH	2
OC-05	Show results	The users could see images with labels.	MED	1
OC-06	Recognize images	The pipeline could use the model to recognize uploaded images.	HIGH	2
OC-07	Show Training Process Bar	The website could show the process bar.	LOW	2

6.1.2 Capabilities to be tested

Table 14: Construction iteration capabilities to be tested

ID	Capability	Description	Priority	Iteration
TC-01	Uploading images	Users are able to upload images from local dictionary.	HIGH	2
TC-02	Model Retraining	The pipeline could retrain the model.	HIGH	2
TC-03	Image Recognition	The system could upload images with different size and format and the result page can show the initial images with labels.	HIGH	2
TC-04	Integration of Tensorflow	The system could use Tensorflow algorithm to train different dataset.	HIGH	2
TC-05	Different Unix based system	The system could be run in the Unix system	LOW	1

6.1.3 Capabilities not to be tested

For OC-02 Preprocess images, this module has been implemented by the Tensorflow algorithm.
For OC-05, OC-06, the performance of OC-06: Recognize images will be shown in the OC-5: Show Results. So these two capabilities will be both tested in TC-03.
Also for OC-01: Upload images and OC-03: Add topic, these two capabilities will be both tested in TC-01.
OC-07: Show Training Process Bar is also tested with OC-04: Train/Retrain the model in the TC-02: Model Retraining.

6.1.4 CCD Preparation Plans

We prepared for two strategies: local PC and AWS for CCD to avoid. Both need one PC.

6.2 Iteration Assessment

6.2.1 Capabilities Implemented, Tested, and Results

Table 15: Capabilities implemented, tested, and results

ID	Capability	Test Case	Test Results	If fail, why?
01	Uploading training images	TC-01-01	Pass	N/A
02	Uploading Vast Training Images	TC-01-02	Pass	N/A
03	Uploading Oversize Training Images	TC-01-03	Pass	N/A
04	Uploading invalid training images	TC-01-04	Pass	N/A
05	Uploading same name training images	TC-01-05	Pass	N/A
06	Sending training request from front end to the back end and starting training process.	TC-02-01	Pass	N/A
07	After Sending training request to start the training process, turn to training process bar page.	TC-02-01	Pass	N/A
08	After training process is done, the front end should get training result.	TC-02-01	Pass	N/A
09	The user cannot train twice at the same time.	TC-02-01	Pass	N/A
10	If the user kills the web page while training is processing, the training process is still continuous. Reopen the page will lead to processing bar page.	TC-02-01	Pass	N/A
11	Test Model Without Uploading Image	TC-03-02	Pass	N/A
12	Test Multiple Images at One Time	TC-03-02	Pass	N/A
13	Test Image with Different Format	TC-03-02	Pass	N/A
14	Test Image with Different Size	TC-03-02	Pass	N/A
15	Test Image Which can’t be Classified to One Topic in the Model	TC-03-02	Pass	N/A
16	Error information for wrong format of input image	TC-04-01	Pass	N/A
17	Check the output of the test part	TC-04-01	Pass	N/A
18	Error information for only one class	TC-04-01	Pass	N/A
19	Error information for small dataset	TC-04-01	Pass	N/A
20	Check the output of the training part	TC-04-01	Pass	N/A

21	Ubuntu14.04 & Use GPU	TC-05-01	Pass	N/A
22	Check the output of the training part	TC-05-01	Pass	N/A
23	Check the output of the training part	TC-05-01	Pass	N/A
24	Check the output of the training part	TC-05-01	Pass	N/A
25	Check the output of the training part	TC-04-01	Pass	N/A

6.2.2 Core Capabilities Drive-Through Results

Positive Feedbacks:

Nice UI;
Good performance for uploading images from local;
Good performance for training model;
Good result for recognize images;

Improvements needed/suggested:

Table 16. Core Capabilities Drive-Through Feedback

Functionality	Improvements needed / Changed to-be considered	Risks / Risk Migration
Select Model	Add some description here for two models or parameters	N/A
Show Results	Add "go back" button; Add image information like "size";	N/A
Select Label	Change the "Enter" icon to "Configuration" icon	N/A
Show Labels	Delete the "Description" column	N/A
Upload Images	Delete the "Search" button	N/A
Retrain Model	Add a functionality of choosing training iteration times to reduce time for training.	It is hard to send parameter to the Tensorflow from Django. And validation for this parameter is hard to implement. The risk migration here is we used four modes for training, and user could only select which mode he/she want. These four modes have different training iteration times, which could make the training time from 5 minutes to 2 hours.

6.3 Adherence to Plan

Our project has zero cost and already fits all the requirements from our client after TRR. After TRR, our team focused on transition and will migration our system before 12/07/2016. All the

preparation of the transition has been done. And all the documentations for transition, training, and user manual have been finished and delivered to our client.

For more details of Transition Plan and Training Plan, please check
http://greenbay.usc.edu/csci577/fall2016/projects/team04/Development/TM_TRR_F16a_T04_V1.0.pdf
http://greenbay.usc.edu/csci577/fall2016/projects/team04/Development/TP_TRR_F16a_T04_V1.1.pdf