Final Report - CS 160 Final Report

Hoang Vu

Project: Dashboard with general apps

Table of Contents

Acknowledgements

Even though I work by myself on the project, I do take suggestions from the professor what should be included in the project. I do get explanations from my brother about tensorflow and neural network. Importantly, I use stackoverflow, geekforgeek, w3school, real python site to get answers and tutorials as the guidelines to complete the project. What's more? To stay high concentration, I need help from my best friend.

Executive Acts

As the only member, I use freecamp.com and notes are the main tools for the project management. I have to show persistence and disciplines to work consistently 100 hours per week. To plan the project, I first choose the target first. Since I work on independent applications, I skim through lists of ideas of applications that I can find on the Internet. At the same time, I ask around from classmates about the project ideas. After two days, I pick 6 apps

and put these apps in the project proposal. They are finance app, accounting app, calculator, login and register, weather app, and comment and advertisement. When I continue to work on the project, I was inspired to do additional apps. At this point, the ideas just flow naturally into my mind; I search Google for how to do apps and then choose more 20 apps. Once I pick the target, I may drop the target later. For example, I first choose to work on the profile and notification features, but I have to drop the two features because I run out of time. I totally have removed 6 or 7 apps from my project.

I initially ask to form a group with any person. At first, I simply tell the mates what I can do, and I did get accepted to 3 or 4 groups. Nevertheless, I was absent from class on next two days for personal emergency; during the time, some groups get disbanded while other groups have already chosen the topics. Finally, I had no choice but worked on the project alone.

I do keep in touch with members of other teams about the requirements of the project. For example, in one week before the mid-term report, one week before the final presentation, and one week before the project proposal, I ask for contacts of classmates, and ask questions about how I can do best on the project presentation.

Background & Introduction

I have very little knowledge of how projects, or works of a programmer and of a developer look like. Since my major is Finance, I chose to work on accounting and finance software. Nevertheless, I realized that businessmen prefer Excel to other web accounting and finance software. In fact, in the survey of people in investing group of San Jose State University, out of 56 people, 40 people want to use Excel, and only 4 people use web software. When I read about viewpoints of developers, I found out that AI and personal customization and motivation are only two competitive features of online accounting and finance software. As it turns out, like calculator, the accounting and finance software is the tool while financial analysts and accountants need skills to work with people and regulations. Also, the books only use accounting and finance as simple examples to implement codes in real life. Finally, I decide that like many programmers, I will do a simple accounting and finance software for personal use.

How did I choose the project topic? Since I want to rush on the project with little coding background, I plan to work on independent apps that later can be put together to use in a big

project. Then, when I see the waiting screen of Mac computer, I thought I should bring every desktop item to website. The idea works if each website is a widget, or a desktop item, and if there is a website to maintain the connections between these websites; certainly, users will not have to type passwords or remember the sites of their banks, or install software to their computer. For me, I can learn more about javascript and web framework by working on simple apps at the beginner level. Since I was unsure about the scope of the project, I only included in the project proposal 6 apps: login register, weather app, accounting app, finance app, and comment or advertisement app.

Purpose & Motivation

Originally, I plan to put my code sources in the sharing mode. In other words, I will contribute my project to github. To do so, I need to keep my project clean so that people can understand my codes.

At first, I thought not so many accounting and finance software is published as open source. However, I was wrong. Then, I decide that I will work hard on my project just by learning new lessons about coding daily. Surely, to get deep insight to every detail of coding, I need to start with creating simple apps.

Honestly, although I find it quite interesting reading and exploring new aspects of coding issues, such as installations, machine learning, cost analytics of apps, error handling, and so on, I does not feel motivated, or inspired to do many apps. Nevertheless, to follow the deadlines of each task strictly, I have to be disciplined.

Differentiator

Each of my app has its own distinct features from similar apps. For example, the game app is originated as a console game; it seems that the game has not been published as an online game before. Secondly, even though the calculator app not only performs finance calculations, but it actually solves homework problems of real estate courses. Put it simply, using my calculator app, finance students can find the answer of the last step rather than go through each step to get the answer. Thirdly, for mail classification, I will classify the email before it is sent to the mailbox; if

the mail is a spam, then the mail will be deleted, and the user email may get blocked. Fourthly, one unique feature of my movie recommendation is that if user agree to give at least 35 movie ratings, the user has a chance to have high correlation with another user. If it is high positive correlation (score > 0.7) between 2 users, then one user should watch favorite movies of the other user. On the other hand, if it is negative correlation score < -.07), then one user should watch movies that the other user does not like. Fifthly, both accounting and finance app are equivalent to a bookkeeper or finance analyst at an entry-level because the software automate most of the basic accounting activities.

Methodology

I have tried many different ways to get the result that I want:

- Researching, following tutorials and looking information from credible sources, such as stackoverflow, javatutorpoint, geekforgeek, youtube videos. For example, I have to include the {% csrf_token %} for security matter in django, or follow installaion instructions of sudo chown -R $USER /usr/local, or J-connector of mySQL and many other instructions.
- I myself have to come up with a similar problem, and use that problem as an experiment and practice on what I have read. After I understand every code line, I start to code what I understand and remember. For example, the movie recommendation is based on the Euclidean score and coefficient correlation. I do find the formula easy to understand, and use the matrix transform library of python for matrix calculation.
- Making random guesses of the codes: After I read the documentation, and see examples of a library of java or python, I develop a hunch of how codes should be written, and when the codes go wrong.
- Reading articles and theories: I read MDN to see the difference between a module and a pure text/javascript. Moreover, I have read about view-controller model, request and response in the server side, web frameworks, python tricks.
- Asking questions: when I get stuck, I post questions on stackoverflow, or ask my Brother. For demonstration, I asked my brother about the tensorflow and neural network. I asked stackoverflow about the potentials of multiplayer browser game.
- Tracking the errors and debugging: on average, it takes me 45 minutes to fix a bug.

- Problem-solving: In each app, I encounter different problems, I apply the design patterns of strategy, listener, composition, decoration,…. One other option is to repeat the codes and display the result in a simple way. For instance, in accounting app, I use 12 if-else conditions to convert the type to the string.

Problem Statement

Since the project is consisted multiple mini independent apps, the testing is quite easy for me. I simply use console to make sure that the result met the requirement. Nevertheless, I decide to use selenium, a web automation, as a proof for testing.

In each app, I need to figure out specific requirements and the scope. First, I look up from the Internet how a technology, such as Django, or Servlet, or Node.js can be applied to each specific app.

One challenge is about writing a coherent and comprehensible descriptions and reports about the project. Since there are so many important details, and writing a summary that reflect all main points of the project consumes much time, I may need to do the report informally.

The login and register app is divided into 2 parts which are self-created login and register interface, and gmail login. At this point, I need to deal with the authentication system and database. In the issue of authentication, I need to deal with email verification, forgot password, strong password, code resend, 2-factor authentication, encode and decode, prevent go back, URL redirect, profile edit, session tracking, account deletion, restricted access, and RECAPTCHA. For database, I have to deal with middleman attack and SQL injection.

For the accounting app, I will not put in delete or edit feature because according to accounting concepts, if any difference or edits take place, ones must not delete or edit, but create a new entity called *Reconciliation* and add to that entity the difference in the amount. Certainly, the financial statements should be simplified.

For the mail classification, I also need to do the mail sending feature which send auto reply. Also, I may need to check whether the email has existed.

For the advertisement app, I have to combine all of possible CSS features. To reduce the design effort, I should look for free ads banners that have been designed; the problem is that such free banners from many websites cannot be embedded into my website.

I will need to employ code reuse for features, such as notes, clock, stock chart, forum, news

For the game app, I have to encapsulate all of the data, use UML diagram to come up with classes and methods. For instance, the class Game uses the class cardDeck and class Player. The game has many states, and the interface drawing should not be canvas but images.

For the chat app, there is a tutorial of creating a group chat using socket.io and express.

For weather map and movie recommendation, I may need to understand the advanced terms, such as minimum rain, counterVectorize, SVM, kernel, gammar, margin, verbose.

For apps based on machine learning, I just need to implement simple apps at the entry level. In short, I just need to use the SVM classifier which is based on the matrix calculations.

Implementation & results

For the dashboard, I apply the strategy pattern to drag, show and hide apps.

For the login and register, because of so many confusing features of authentication, I have dropped the reinvented authentication interface, and deployed the GoogleOath for my login page. Next, I have tried to employ 2 following options. In one way, I encode and decode the user id, pass the information to each app through url. In another way, I send request of token access to google and retrieve the current user id. Unfortunately, Google only allows me to use the first way.

For the calculator app, I take the solutions of finance problems, and translate each step of solutions in to the algorithm in the calculator app. I use scriptlet and put the algorithm in methods.

For weather app, I use openweathermap and darksky and django to retrieve the data.

For accounting and finance, I use scriptlet, servlet, jsp, and mySQL to store data and display the results.

For movie recommendation, I use dataset from heroku for content-based system and dataset from movielens for user-based system. For the user-based, I use pandas to read csv file, then use sklearn to transform words and features into matrix. Then, I use numpy to calculate the eucleadian distance, coefficient correlation.

For digit handwritten, I have to rewrite the dataset in python, and make canvas in javascript

For the chat app, I add the feature of profanity filter so that no bad words can be displayed. I use express to import static files, such as images, and socket.io to update the server.

For the card game in javascript, to ensure the encapsulation, I create and put classes in different JS files. In addition, I use the random name generator module from the Internet.

For dictionary app, I use BeautifulSoup, a similar technology to Selenium, to look for elements of Oxford Learner English Dictionary, then and I retrieve the definitions.

For advertisement app, I add all CSS effects to my website except those that are already present in the banner.

For news, I use the news.api and javascript to retrieve data.

For clock, forum, unit converter, scoring board, I use code reuse and weebly.com website

For debug, I use console, and ensure that each app works before I work on the next part. Also, I add test cases using selenium.

I do drop some features. For example, I have dropped the paypal donation button, the rock-paper-scissor game, the notes and notifications.

Result: Here is the list of what I have created: (12.5 / 18 apps)

1. Dash Board: => 95%
2. Login page: => 100%
3. Calculator App: => 100%
4. Weather App: => 100%
5. Accounting app, including Transactions and View Transactions, Bills, Customers, Product, Balance Sheet, Income Statement, CashFlow => (90%)
6. Finance app, including Personal Budget and View Personal Budget => 80 %
7. Mail Classification (aka Contact Customer) => 75%, including mail sending, write dataset, train model with new parameters, model optimization, and post result
8. Movie Recommendation: => 90%, including the interface, the entire user-based, and half of content-based (I made movie & index retrieval, and coded the formula of cosine similarity
9. Review and Rating app (aka digit handwritten recognition) : => 85%, including canvas, database, login form, dataset, and post result on web
10. Card Game => 90%
11. Chat app => 90%

12. Dictionary app => 100%
13. Advertisement app => 80%
14. News App, including the loading scrolling bard, and additional news at the bottom (70%)

Here is the list of features that I embed to my app:

15. Scoring Board => 0%
16. clock => 7%
17. Unit Converter => 0%
18. Forum => 0 %

+Drag element of dash board

+ Quizz of accounting app

+Stock & interest of Finance App

+ numpy and sklearn of email classification

+ numpy and sklearn, and model optimization of movie recommendation

+ numpy and sklearn, and model optimization of digit handwritten recognition

+ Filter module of chat app

+ Random name generator & shuffle algorithm of game app

+ In advertisement app, did not create the ads on the left & the animation package

+ Headline RSS feeds of news app

Conclusions

I am the only member of the project. For the project, 6 technologies that I have used are node, django with python, selenium testing, mySQL database, java with jsp and servlet, machine learning, and Javascript with html and CSS.

I myself have created 12.5 app of 18 apps.

Afterall, I gained knowledge of managing projects, skills of independent researches, and discipline attitudes.

For future project, I plan to do the serial and sync of codes, stress testing, client interface optimizations, and code maintenance.

Appendices

a.links to your source code on github

https://github.com/hoangvu27/cs160project

Alternatively, just type in the search box, hoangvu27/cs160project

b. Final report of progress & agile management tools:

Scrum master: Hoang Vu – only member

Sprint: June 26

+ calculator app & advertisement app
+ Read server-side, drop Php, drop C# and ASAP.NET, but learn mySQL

Sprint: July 3:

+ Created dashboard, weather app, dictionary app, news app
+ Drop spring MVC web framework, but learn JSP

Sprint July 10,

+ accounting and finance app
+ debugging, read costs of finance app, what best accounting app

Sprint July 14 (mid-term report)

+ movie recommendation app
+ read neural network and tensorflow

Sprint July 26:

+ Created game app, rating app

+ Learn Django + CSS

Sprint July 31

+ chat app, email classification app

+ Learnt socket.io and express

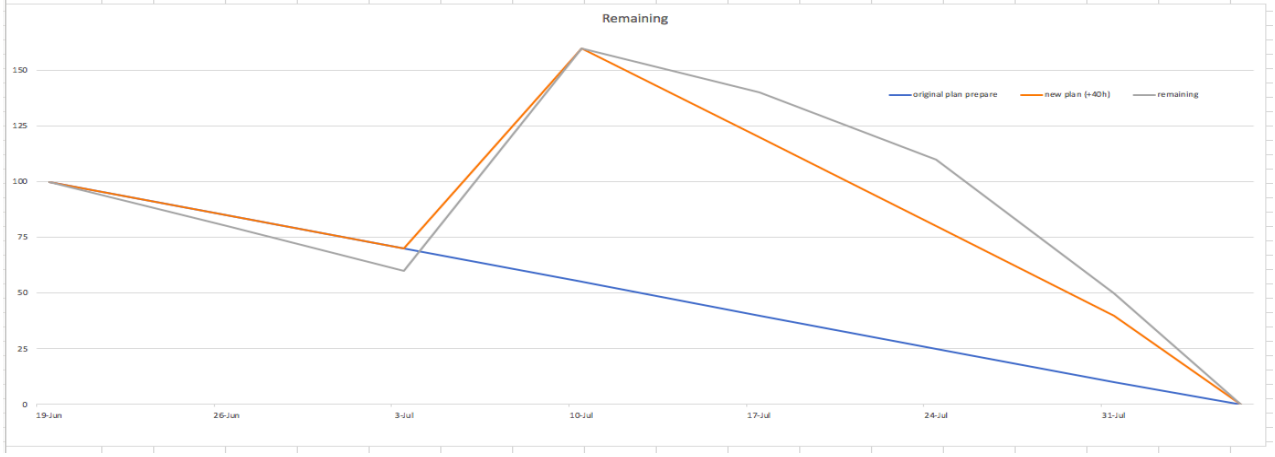
Sprint Aug 5:

+ Unit converter, scoring board, login page, clock, forum, selenium

+ presentation

*** If these images are unclear, I attach the report with several image files that show specific start time and complete time of each task. I use freecamp website.

	12-Jun	19-Jun	26-Jun	3-Jul	10-Jul	17-Jul	24-Jul	31-Jul	5-Aug
original plan	prepare	100	85	70	55	40	25	10	0
new plan (+40h)		100	85	70	160	120	80	40	0
remaining		100	80	60	160	140	110	50	0



	Task	Hour	Remaining
1	DashBoard	5	0
2	Login & register	10	0
3	Calculator	4	0
4	weather app	4	0
5	accounting app	10	0
6	finance app	5	0
7	email classification	5	0
8	movie recoomendation	10	0
9	chat app	5	0
10	game	30	0
11	digit written recognition	10	0
12	news	3	0
13	advertisements	6	0
14	dictionary	2.5	0
15	clock	1	0
16	forum	0.1	0
17	scoring board	0.5	0
18	unit converter	0.1	0
19	bug testing	24	
20	read & research	24	
	total	159.2	

Link to my demo

https://www.youtube.com/watch?v=08MNCl1sAJ0&list=PLGMekoYYYfY61yrEFNA8Xy4Q7XJO8vSsH&index=151&t=0s

Alternatively, type in youtube search box: Hoang Vu - cs160 project - SJSU summer 2019

References

Drag element of dashboard:

https://www.w3schools.com/howto/howto_js_draggable.asp

Login Page:

https://developers.google.com/identity/sign-in/web/sign-in

Finance:

Interest rate: https://www.mortgagecalculator.org/rates-widgets/mortgages/

Stock: https://www.tradingview.com/

Weather:

https://openweathermap.org/

https://darksky.net/forecast/40.7127,-74.0059/us12/en

Movie recommendation:

https://medium.com/code-heroku/building-a-movie-recommendation-engine-in-python-using-scikit-learn-c7489d7cb145

Digit handwritten

https://www.youtube.com/watch?v=7aK7tVdcVbY

Email classification

https://www.youtube.com/watch?v=exHwwy9kVcg

Chat app:

https://medium.com/@noufel.gouirhate/build-a-simple-chat-app-with-node-js-and-socket-io-ea716c093088

Machine learning topics:

https://medium.com/machine-learning-101/chapter-2-svm-support-vector-machine-theory-f0812effc72

Advertisement:

https://bannernow.com/samples

News:

https://feedgrabbr.com/

Scoring board:

http://www.tablesleague.com/

Dictionary:

https://www.oxfordlearnersdictionaries.com/us/

Clock:

https://www.w3schools.com/graphics/canvas_clock_start.asp

Forum:

https://www.createaforum.com/

Django:

https://realpython.com/get-started-with-django-1/