Capstone Project Proposal

AFFORDABLE

January 2019

1 Background

AFFORDABLE is building a system to connect and match patients with charity opportunities and billing assistance. Our technology stack is made from a front-end webapp using jQuery and websockets connecting to a C++ linux based webserver, all running on Amazon Web Services. AFFORDABLE still is looking to finish up several parts of this system and could mentor students through learning how to build complex systems. Sean Pannella, formally a trading systems developer for Susquehanna International Group, is the lead software developer for the AFFORDABLE platform. He will help answer questions and guide students through designing challenges during the semester. Alex would also interact with these students overseeing their progress and working as mentors.

2 Design Premise

“HUGs” are small sponsor microgrants (Healthcare Utilizing Grants) that can be delivered to individuals for various healthcare needs. Once the money is raised, the donating parties can select the optimal beneficiaries for their fundraiser. These funds are collected for a specific service or commodity, such as pharmaceuticals, healthcare treatment, etc. Hence, the distribution of these funds must be carefully audited to both limit fraud and to maintain the confidence of the donors that the funds are being utilized appropriately. As a third party between the donors and recipients, we can distribute the funds directly to the user and verify the expenditures for that grant in a simple and efficient manner that doesn’t hamper the user experience. Our approach is to transfer funds piece-wise with each relevant expenditure and have a running ledger of each transfer.

Your goal in this project, is to develop and optimize this management system to 1) upload an image of their receipt/invoice and subsequently store this information, 2) have a grant recipient manually log in expense details, 3) have image recognition of the cost totals to limit fraud and curtail manual surveillance measures and 4.) Have an administrator management option for modifying both user response and software image recognition failures and 5) validate image recognition with a data set. Your management system should be able to stand alone, and integrate well into the platform of others, such as AFFORDABLE.

We have included a image of our needs for such software. The receipt upload and recognition would exist within a greater management system where users are accessing funds from multiple sources. The user needs to be able to log all their expenses and submit for payment within AFFORDABLE such that we can deliver the funds, and the donors can both view and approve expenses. This is where we need your system for a user to be able to quickly upload a photo and log in data in a user-friendly way. The average user will be doing this process many times, and thus its accuracy and efficiency is of the utmost importance!

Based on the success of your image recognition, you can expand the fields of recognition to include other information such as the date. The further you can push this concept, the more powerful your project will become for a multitude of other purposes.
2.1 Deliverables

2.1.1 Weeks 1-3 Milestones

Propose a plan and request needed clarifications on the milestones. Make two webpages:
1. A webpage that allows a user to upload an image of a receipt and information about this receipt like the purchase total price, the store name and item name.
2. A webpage that allows an admin user to view all of the submissions and change the total price, the store name and the item name. The admin user should also be able to mark the receipt purchase as accepted or rejected and this information should be save for next visit.

These webpages could be supported with any backend you like, however, the data should be stored in a SQL based database like sqlite or mysql or any other SQL variant. (If needed Sean can provide a database and/or a AWS EC2 box to run the backend) The frontend could use any framework but React is preferred. The goal is to have a non-polished demo with basic functionality done by the end of this period. Finally, start collecting receipts of your own to build a training set / testing set. (Validation set would come from us)

2.1.2 Weeks 4-6 Milestones

Clean up the implementation made during the first milestone so that at the end of their period a polished version could be presented and deployed. Work with Sean on setting it up so there is a version Warren and Sean can access and start setting up the validation set. Propose a plan on how to recognize the following on these receipts: total price, date, store name, possible item names (total price should be a priority). This plan should include making a training set, an algorithm for recognizing, a scoring system on how well you are recognizing, some relevant research to backup why you think your algorithm would work, and assumptions on the dataset. The plan should also stipulate what you do not support, so test cases of that sort can be removed.

2.1.3 Weeks 6-9 Milestones

Implement a working version of your proposed plan. Communicate with Sean and Warren if the algorithm is not working as planned and make a proposal of a new algorithm with new constraints. Start to quantify how successful the algorithm and bucket where it seems to work better and where it seems to work not as well. Be able to present us some preliminary results on your training set. Alert us if you hit any slow downs or need to change your proposal.

2.1.4 Week 10-12 Milestones

Polish up your implementation. Prepare your final presentation. Work with Sean and Warren for testing it against their validation set. Write up a document about roughly how it works, what improvements you would recommended and any lines of thought that you attempted but they did not work.

2.2 Learning Goals

• Learn to request clarifying information on goals
• Learn how to counter-propose a plan that fits into the scope they feel they are able to complete
• Organize as a team to complete a simple CRUD application with image upload
• Learn to research known solutions for problems like the receipt recognition problem
• Setup a training dataset and how to evaluate their machine learning algorithm
• Possibly learning how to propose a solution if their first approach fails
• Reflect back on possible improvements and report on ideas that failed
• Practicing communicating with a stakeholder and keeping them up to date on your progress