Prospective Projects for Fall 2018 CS 499

**Project 1 –**

Project:

Shared Cloud Development Environment

Google Docs is an online word processor that enables easy collaboration between users. The goal of this project is to create an online code editor with collaboration features similar to Google Docs. In order to do this, the application should show which users are currently viewing the document, show where their cursor is in the document, and show their edits in real-time.

This project will likely require the students to implement (or adopt) a file system backed by a public cloud storage solution. CodeMirror (an open source web-based code editor) can then be used as a starting point for the web page. This web page will need to be integrated with the filesystem so that CodeMirror can write to the stored documents. CodeMirror itself may need to be altered to show the cursors of other users editing the same document. A server will likely be necessary to accept edits from CodeMirror and broadcast them to other users viewing the same document. If possible, the addition of revision control and/or snapshotting would improve the usefulness of the application and make it more similar to Google Docs. The final goal would be to host the finished application publicly.

The customers for this project would be Donald Venus (May 2018 UK alumnus) and Travis Heppe (Google). We will both be fully remote for this project and likely will not be able to attend anything in person.

donaldmvenus@gmail.com heppe@google.com

**Project 2 –**

Real time Task Tracking Scoreboard

 The Nick Ratliff Realty Team is a small team of real estate agents that focuses on customer service and creating consistency in the real estate market. In an attempt to measure productivity of both team members and the marketing efforts, there is a need to regularly monitor the daily activities of the agents. We are looking for areas where we need to increase efficiency and overall productivity as well as identify opportunities for growth and training.

 In hopes of finding these answers, we are interested in developing a phone app, both iPhone and Android compatible, that would help the team members track their daily activities. It is important to understand that these team members are mostly real estate agents which are independent contractors and do not have regular office hours. They are often busy and multi-tasking so an easy way of inputting the data is vital to encouraging regular, prolonged use.

**We have a number of potential trackable ideas…**

* Track phone activity and interact with the user after each phone call / text to quantify trackable calls and conversations.
* Have a database of client contact information to automatically log phone calls or text conversations
* Hourly / Shift / Daily Log entries
* Potential to set or log hours of working each day to calculate efficiency
* Track client appointments
* Track client showings
* Track client consults
* Track income based by client
* Track contracts with projected income
* Project income by month
* Eventual build out of projected activities required for an income
* Ability to project based on seasonal activity

Presentation

* Scoreboard with Agent’s ranking based on whatever we track
* Easy Visual platform of projected income by agent and for team overall

Overall the concept is basic. The ability to obtain data that allows a team leader to track a customizable list of actions. Truth is that task can be done with a simple Google form, but the challenge comes in a simple user-friendly app. It must not be invasive to their work flow yet increasing competitiveness among team members while maintaining high levels of accuracy in reporting. Every business owner wants to know how to increase profits, this app should help clear that up for the owner while introducing a level of fun for the end user.

nick@lexingtonkyhomesearch.com

**Project 3 –**

Crash Recreation – 3 options

 **Delta V Innovations Inc.**

**University of Kentucky CS499 Senior Project Descriptions**

Delta V Innovations is committed to providing software specializing in Computer Aided Design for crash and crime scene recreation, storage of captured data, and sharing amongst users: data, analysis and simulation of events. With law enforcement in mind, Delta V Innovations also intends to offer this software application to numerous users including but not limited to insurance agencies, private investigators, and car manufacturers.

There are currently 3 areas of development:

- A simple mobile application used to collect measurements, global positioning system data and photos to pass along to a 3D software at a separate location.

- A desktop application which will be graphics based and coupled with a physics engine library to complete a simulation of the crash or crime scene.

- A cloud-based database used to link the two platforms together. This will take input from the mobile application and allow users to share it with the desktop application or other users.

We are currently seeking motivated teams for each of these categories.

**Mobile Application**

The current mobile application is in both Android and iOS format. Originally created by the CS 499 spring class of 2018, it has a simple to use interface allowing the end-user to search for reference materials commonly used by crash investigators in the field. This application, Delta V Field Light, has a simple-to-use interface which communicates with the existing database, hosted in AWS. Future functionality of the mobile application includes allowing the user to access a list of crash reconstruction formulas, take photos of crash or crime scenes, utilize global positioning system data for geo-referencing, and make general notes. Future teams will need to further design the current application, make improvements to the existing program and develop innovated solutions to advance the application’s goals.

**3-D Desktop Application**

The 3D desktop CAD software allows the user to recreate the crash or crime scene by using basic CAD tools and realistic graphics. The previous CS499 team has created the communication protocols allowing the transfer of data from the mobile application, through the database hosted in AWS and display on the desktop program. Future team members are being asked to develop a report page to be used on the desktop application which will allow the user to select formulas for calculating speed and other data, enter variables, see the solutions and display/print the report. This function will be integrated into the current desktop program UI.

**Database Support**

The design and integration of the database ties the multiple different (both current and future) software applications together. The database consists of a SQL database hosted on AWS with supporting stored procedures that allows for simple access and functionality by all applications. The current database has been designed with current and future functionality of the mobile and desktop applications. The existing reference material which has been imported into the database supports the current mobile application. Future team members need to further design the database, update the existing data, and develop methods which will allow the user to search for information.

Thank you,

Mike Flamm

Delta V Innovations Inc.

President

513-706-1893

DeltaVInnovationsInc@gmail.com

**Project 4 –**

Beyond Birth Addiction Medicine Clinic

The Beyond Birth Clinic is an outpatient addiction recovery program for early postpartum women with substance use disorders.

We are interested in the opportunity to offer our patients a system that helps them track moods, receive suggestions on healthy ways to respond to negative stress and anxiety. By tracking scheduled recovery activities, prompting women to set daily priorities, and manage weekly responsibilities, we believe recovery from substance use disorders becomes stronger.

Ideas include:

Prompting users to enter an emotion

Offering access to pre-recorded guided meditation and Jin Shin Jytsu videos (2 min in length)

The opportunity to post encouraging notes for others who have successfully navigated a day of high anxiety, etc.

Include a meeting locator for recovery meetings or recovery social activities

Offer a “checklist” journal entry about how they felt upon waking, what resources were used during the day, and how they felt by bedtime. Possible additional items to record include # of cigarettes smoked, # of times a person “vaped”.

A person could view their own data as well as aggregate group data.

They can provide feedback (visible to others) about what features or postings were helpful to them.

Add a daily gratitude feature “Today I am thankful for \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_”

Cause phone to prompt the person to respond to the question: How are you doing? Reminder so individual can take a moment to check in with themselves. Could be preset to go off as often as the person wants. Could be a checklist or 1-10 scales for anxiety, stress, etc.

Insight Timer: posting and meditation timers

Breathe - to inhale and exhale to a picture showing color movement with breath - can be adjusted to own speed

Some apps to look at are:

Sleep Cycle - checklists, sleep analysis

Holly Dye
Transformation Manager, Beyond Birth Services
University of Kentucky
College of Nursing
Perinatal Research Center

Holly.dye@uky.edu

**Project 5 –**

marry two APIs to create real time object recognition using the camera of a smart phone

In the past several years Google and Apple have released APIs to help developers write apps for machine learning and augmented reality on mobile devices.  As if not to confuse us Google has released ML Kit and ARCore while Apple as released ARKit and Core ML.  In this project we would like to marry the two APIs to create real time object recognition using the camera of a smart phone.  This project will be investigative in nature but its practical applications for Lexmark could include providing a HUD (heads up display) for our service technicians in the field working on our printers.

For this project in particular the goal would be to create an app to recognize poker hands.  The user would be able to open the app and show it a hand of cards by moving the phone to put the cards into focus of the camera.  Then in real time the app would label the hand as straight, flush, full house, etc.

Patrick McDaniel, Lexmark, patrick.mcdaniel@lexmark.com

**Project 6 –**

Title: Restoration of compressed scanned document images

Archival of printed documents often requires retention of the digitized copy of the document. When there is a large quantity of documents, storage size becomes a critical concern.

Lossy compression offers a solution, but the quality of the reconstructed image may suffer. For example, JPEG compression artifacts such as ringing or blocking can distort text within the image, making it difficult to read.

The goal of this project is to restore and enhance the lossy document image to resemble the original, scanned document as closely as possible. The project will explore machine-learning approaches to transform the compressed image into a higher-quality image.

In addition to developing a single, multi-purpose model (i.e., for all types of images), the project may consider additional models customized for specific types of compression or image content.

In addition to subjective visual evaluation (i.e., looking at the image), the project should consider one or more objective quality criteria. Examples include fidelity metrics (such as SNR and SSIM) and functional metrics (such as OCR accuracy).

The customer will provide full-resolution (ground-truth) scanned document images. Using these images, the project team will develop training data.

Some training samples should use JPEG compression across a range of quality factors. Other training samples should apply simple spatial and/or tonal resolution reduction (i.e., lower resolution and/or quantization of tone levels). The project team may wish to include other lossy compression algorithms in addition.

In all cases, test images need to include an appropriate range of compression.

The customer will evaluate and approve the proposed collection of training samples.

The deliverable product consists of functional machine-learning models that the customer can run and evaluate. The project team should clearly document and explain their approach so that the customer can readily duplicate their results.

Patrick McDaniel, Lexmark, patrick.mcdaniel@lexmark.com

**Project 7 –**

AI for large option trading data set for Fishback Management and Research

Artificial Intelligence is currently at the forefront of the financial world. We at FMR have already created research engines that perform robust industry-standard analytics. Our endpoint objective is to accelerate this style of analysis by creating an AI learning platform using TensorFlow. This semester's project will be a proof-of-concept towards that end goal. Because it is a proof-of-concept, the number of variables in the analysis will be limited. But the project will still be dealing with Big Data. This is a great opportunity to work closely with UK Industry Partners and get exposure to some of the hottest topics in the computer engineering field: Big Data, AI and Tensor Flow.

Don Fishback odds@donfishback.com

**Project 8 –**

SQS –

SQS:

Our project will continue regard enhancing the training web application for SQS Quality Assurance New Hires. Previous CS499 groups have developed a web application for new SQS employees to use as a training platform for both manual quality assurance and test automation. This web application is built on PHP, Apache, and SQL. It needs to be further enhanced to add and update functionality to meet current day web development standards. The web application is modeled on the idea of a skills bank for employees to track their status, while identifying existing (purposeful) defects within the site.

Lance Radebaugh Lance Radebaugh <Lance.Radebaugh@sqs.com>

**Project 9 –**

GE Appliances

Vision Station

Description: We would like to design a PC based solution to do visual inspections on an assembly line.

Terms used throughout the project description:

* Image source - it can be a folder with stored images or a vision device such as web camera, GigE camera or 3D sensor.
* Inspection – a process of analyzing images
* Communication (comms) – a way to send and receive control signals and results.
* Job – a logical connection between image source and the inspection process. When a “job” is loaded, the system connects to an image source and conducts a selected inspection process.

The system should be able to do the following:

* Select image sources: read and load images from a local or remote folder or acquire images from vision devices connected to the PC or located in the network.
* Perform visual inspection and analysis of the images
* Load various vision inspection jobs, i.e., one job can check for presence/absence. Another job checks for dimensions.
* Provide the results of the inspection both to an external party and visually.
* UI
	+ to display the results of the findings on a screen.
	+ Provide a way to select the source of images: files, cameras.
	+ Default job selection comes from the config file
	+ to select a job from a list of jobs present in a config file

The vision inspection should be able to load either of the tools “OpenCV” or “Halcon”. OpenCV is an open source tool, temp licenses for Halcon will be provided.

The communication module to should be able to receive signals from a “folder”, “UI” or “keyboard”, as well the network. At the same time, the communication module should be to send control signals and results.

Incoming signals are as follows:

* reset
* load job (by job id number)
* trigger

Outgoing signals are:

* ready (control signal)
* fault (control signal)
* pass /fail (result)
* alpha numeric values (result)

The following diagram may explain better the overall flow of the system.

Job determines a combination of device, vision process and communication path.

Device provides an image to the vision process. The vision process provides a result to the communication block. The communication block passes the data and control signals to the external objects. In our case the external objects are “local directory” or a network server.

Please note the communication module should be able to receive control signals from external objects. The commands like “reset”, “trigger”, or change jobs. The communication object sends all current communication actions to the UI for visibility.

The UI can initiate a job change or provide a mechanism to issue control signals to the job (trigger, reset).

Hakim Sultanov Hakim.Sultanov@geappliances.com

**Project 10 –**

Better classifiers for OCR

I have an OCR program written in C. It can be trained to recognize

letters from any alphabet, any font. The classifier I use is

nearest-neighbor in a 26-dimensional Euclidean space, where each

dimension is the fill percent of one of the 5x5 regions of the rectangle

surrounding the letter, and one dimension represents the ratio of height

to width. I would like to give the user an option to choose other

classifiers. In particular, I would like a deep-learning classifier.

The classifier can be written in Python, but it must integrate with the

existing C code. The code itself is in ~raphael/projects/ocr/ ,

available in the MultiLab.

Dr. Finkel raphael@cs.uky.edu

Project 11

 Solar Car Team

Lap Time Recording Utility

Goal: A graphical interface for manual, semi-automatic, and automatic recording of solar race

car lap times.

During a solar car race, recording lap times is a critical responsibility for each team. At the

minimum, each team needs to record their own times for official documentation purposes.

However, knowing other teams’ lap times during a race allows for more advanced strategy and

can provide an upper-edge against other teams.

The goal of the project is to create a software solution for recording lap times during a solar car

race. Here are some high-level requirements:

1. Simple to use graphical interface, clearly display each team’s:

a. Last lap time

b. Average lap time

c. Slowest lap time

d. Fastest lap time

2. Ability to record multiple teams’ lap time information.

3. Manual entry mode, allowing insertions of new lap times or edits to previously recorded

lap times.

4. Intuitive semi-automatic entry mode, for example through the use of three buttons (one

start-timer button, one lap button, one stop-timer button).

5. Robust automatic entry mode, for example through the use of a camera vision system

and detect teams passing the finish line.

6. Save data locally to disk and upload to remote/cloud storage, for example Google drive.

7. Provisions for maintaining a second clock, which is manually synchronized with the

event’s “Official” timing clock (which may be offset by an arbitrary amount). All lap times

should be timestamped with both this time and the best available local time estimate

(from the OS)

8. Preserve recorded data in the case of a program crash or unexpected loss of power

(flush buffers to file after every write operation)

weilian.song@uky.edu

Project 12

 Solar Car Team

Vision-based Cloud Coverage Forecast

Goal: Utilizing a generic webcam and local weather information, predict cloud coverage for the

next eight hours in one-minute increments.

During a solar car race, cloud coverage is one of the main factor that reduces the solar array

output, thus impacting the overall performance of the solar car. This weather variable is visible,

and can change very rapidly in a small geographic area; therefore a local, vision-based

algorithm is ideal for prediction.

Some high-level requirements:

1. The algorithm must accurately calculate the current percentage of cloud coverage in sky.

2. The algorithm must be able to run on a laptop or embedded computer (raspberry pi, etc.)

3. The algorithm must be able to output predictions for the next eight hours in one-minute

increments.

4. The algorithm must utilize sky imagery obtained from a generic webcam for prediction.

5. The algorithm may utilize a limited set of local weather information, such as ambient

temperature, wind speed/direction, humidity, weather forecast from an included API, etc.

The algorithm should be capable of functioning with just the webcam with reduced

functionality.

6. The team may want to utilize a deep neural network for prediction, using imagery and

weather data as input and percentage of cloud coverage as output.

At each reporting interval, the team may also predict the optimal angle to position the solar array

for maximum solar power. This is of particular interest because on a cloudy day, the most

optimal angle may not be directly at the sun, especially near dawn/dusk.

The team may also attempt to predict other attributes we can collect labels for, including the

probability of precipitation.

weilian.song@uky.edu

Project 13 –

Jupyter notebooks have become widely popular.  While iPhython and Jupyter notebooks were initially used by the coding community to create notebooks suitable for interactive programming, particularly in python, their usage has exploded in recent years to include a long list of "languages" (e.g., R, Matlab, bash, PowerShell, Java/Javascript, C/C++, PHP, Perl, Go, and others) as well as markup/down capabilities.  Almost any type of interactive document can now be represented and shared via Jupyter notebooks.

However, a key challenge toward sharing these notebooks has been recreating the environment where the notebook was developed.  Many notebooks require that a custom environment with specific software, libraries, and data/files be installed before a notebook can be shared and used by others.

One solution to this problem is the Binderhub project that combines several state of the art technologies to make notebooks shareable. Binderhub leverages Kubernetes containers running in Google Cloud along with JupyterHub and Github to dynamically download, "compile", and launch a custom environment needed to execute a Jupyter notebook, making it accessible over the web.

The goal of this project is to construct and deploy a Binderhub system and all its components, including working with private and public cloud resources, and then tailor it to support one or more specialized notebooks.  This involves working with containers, cloud services, orchestration systems, git repositories, and a long list of web services.  The next step is to develop control software that simplifies the creation of custom binderhub environments, along with the development of some example environments.  While there is flexibility in the example environment to be developed, an immediate need is to develop an environment capable of supporting notebooks for big data analysis and machine learning.

Customer: James Griffioen (CCS/CS), Tony Elam (CCS),  and Jurek Jaromczyk (CS)

griff@netlab.uky.edu