CS 682 Project Lists – Fall 2018

1.	**Title:**	A Web Platform to Connect Educators, Organizations and Students
	Admin:	Esi Adeborna, OGR LLC, gr.gbordeme@gmail.com
	Description: This project seeks to provide a platform for educators, organizations and students to connect and work on mutually beneficial projects, creating real world experience in the classroom. The platform allows educators such as college professors to post classroom assignments/projects (e.g. software development) with specified criteria. Organizations interested in leveraging the skills of students in these classroom assignments, submit actual project that meets the criteria specified by the educator. This project would develop a database-driven web based application, using any programming language of choice. The application would require logon for all three players (educators, organizations and students). The application would allow educators to post classroom assignments with specified eligibility criteria. It should also allow organizations to post projects in response to class assignments.	
	Resources: • Resources can be made available as needed.	
2.	**Title:**	Credit Underwriting Software
	Admin:	Rambeau, Elevate Export Finance Group, srambeau@elevatefinance.ca
	Description: Elevate Export Finance lends to foreign buyers of Canadian manufactured goods with an Export Development Canada guarantee. Our products allow Canadian SMEs grow by reducing their payment risk and allowing them to maintain their payment risk. **Details** Through this project students will be expected to consider our scoring system and integrate it into a custom application that allows us to determine credit worthiness based on pre-determined financial criteria. **Goals** A credit application that incorporates our mandatory requirements and computes a score/result for each potential borrower.	
	Resources: • Resources can be provided as needed	
3.	**Title:**	The Influence Factors and Risk of Older Adults Falls
	Admin:	Wei Ding, UMass Boston, wei.ding@umb.edu
	Description: This is a machine learning application project about falls among old people. Falls have been detected as one of the most important reason of death in older adults. Therefore, identifying the risk of falls and taking care of older adults with high risk is very important. Students who join this project will be joining an ongoing research project in Professor Wei Ding's Knowledge Discovery Lab, a collaboration with Dr. Suzanne G. Leveille from the nursing department at the Umass Boston: given a large, annotated dataset of readings taken by wearable sensors and walking on Gaitrite Mat. Can we detect what kind of relationship among pain, gait and falls? Furthermore, can we predict the risk of falling for individuals? Several state-of-the-art current machine learning models will be run first. And we also plan to perturb the original data to check the performance of models. Moreover, some paradoxes was found during previous research via domain knowledge, they would be our entry points of further	

	studies. Then we plan to design a complex model which can track the performance of each sample from data set in the prediction models. Our final goal is to use this machine learning model to help domain researchers detect the patients who need special care. Models will be designed and coded in Python or R. Students will help code model designs and create a flexible framework to feed data to those models, validate them during training, and test their efficacy. Expect requirements to evolve as research progresses; new avenues of investigation may take the code in unforeseen directions!	
	Resources: • Existing project code and documentation. • Ongoing support and discussions with students and the faculty. • Other computing resources can be made available as needed.	
4.	**Title:**	Memory Efficient Voxel Processing Library
	Admin:	Todd Sparks, Product Innovation and Engineering, toddesparks@mopine.com
	Description: The frustration: Voxel based geometry allows for simple and robust morphological processing. However, the memory footprint scales with the cell size of the volume being described. This tends to occupy considerably more memory than something like a tessellated surface model. The project: I need a compressed voxel data structure (think three dimensional JPG or PNG) for storing *float* or *int* data (not just *boolean*) that I can use for morphological operations (e.g. erode, dilate, marching cubes), math operations. The objective is to be able to work on structures where the "full fat" embodiment may actually be larger than the available system memory. I envision this working by marching a ROI around the compressed structure so that the entire work volume need never be uncompressed to do any work on it. The impact of lossy vs lossless compression is of interest too. Language: I need a library I can use in Python 3.6, but something lower level for the library is probably smart. Features: • Compressed 3d array data structure • Castable to and from a *numpy.array* instance in Python 3.6 • Addressable similar to a numpy array • Morpological operations such as erode and dilate • Math operations (per-element add/subtract/divide) • Isosurface computation (e.g. marching cubes) References: The solution I currently use is from scipy's ndimage. Documentation can be seen here: https://docs.scipy.org/doc/scipy/reference/ndimage.html#morphology	
	Resources: • Resources can be made available as needed.	
5.	**Title:**	Software Development
	Admin:	Joe Weiss, IDM Systems, joe@idmsystems.co
	Description: We provide a data management solution to large building and hotels maintenance departments. Our system replaces the old way of collecting data on paper and a clipboard and the data captured using a smart device. The data is then automatically analyzed against the	

acceptable building parameters and items that are out of range are flagged. The data is then compiled and displayed on a digital sign display within their office for easy viewing and identification.

Currently we are using Google Forms and Sheets to collect the data. A custom database, app, and program is needed to streamline the on-boarding process so that the company keep up to growth demand.

Creating a responsive web application to replace and enhance current means of data collection.

The program will need to:

- Have the ability to easily create, copy, and edit data collection forms and templates.
- If forms are filled out with poor or no service the form is held in cache and once service is restored then it is submitted automatically.
- Allow multiple user levels within each facility and the entire database.
- Must have search field within results and be able to print desired results.
- A given date range of data from each form must be complied to fit 16:9 display. That data is then posted to the web.
- Have conditional formatting abilities to format data based on a desired value.
- Daily logbook and pass down feature. Have ability to select if comments written on forms should be added to daily log from computer or mobile device.
- Allow for work order creation.
- Keep audit of user input and changes.
- Document Storage at two levels of viewing (manager and associate)
- Have the ability to import or copy and paste Excel, Word, PDF documents into forms and templates (such as rooming list).
- Be able to quickly view last 5 days worth of submitted data and be able to append to the comments field.
- Have a walk-thru feature for quickly creating work tickets and project lists.
- Ability to see important documents in storage such as Emergency Response Manual and or Equipment Manuals from a mobile device.
- Have a Preventive Maintenance feature for repeating scheduled maintenance tasks to be completed on mobile device.
- Catch repeating problems and flag them.
- Create QR codes.
- Access site specific forms and templates from either the app tab or by scanning a QR code. QR code can only be accessed by trusted devices/users.

A fully functional system as listed above.

NOTE: Some work has already been done by previous CS 682 Students. This project will be a continuation of existing work.

Resources:

- Access to current system documents and forms.
- One on one meetings and program direction from CEO.

6.	**Title:**	New Meal Delivery Program Operations Software System
	Admin:	Mona Lee, Project Angel Food, mlee@angelfood.org

Description: Project Angel Food is the only non-profit in Southern California providing free food and nutrition services for people living with critical illnesses. We have delivered 11 million free meals for over 20,000 Los Angeles residents since our creation in 1989. Each week, we deliver 11,000 meals throughout LA County. Each year, our delivery fleet travels at least 100,000 miles, distributing our food across 35 major geographical routes – from South and East Los Angeles and Long Beach to San Gabriel Valley and San Fernando Valley to the rural towns of the Antelope Valley.

Project Angel Food is seeking a unified program operating software system, essentially a custom-meal delivery CRM. In conceiving our needs, it may be helpful to envision a small manufacturing operation with a heavy customer support/interaction aspect. We need a robust cloud based system, with security and encryption that is e-PHI and HIPPA compliant; it should be scalable and afford logging as well as several levels of file and data accessibility. The ability to communicate with various other systems, for example, a proprietary government agency data system, is important.

Summary of the end-to-end process modules:
We have identified 4 major modules of our Program Database – these module must work together to ensure our Program can complete our mission in an effective and efficient manner. The Agency Departments that will be using the Program Database are identified as: Client Services, Nutrition Services, Kitchen, Dispatch, and Development/Finance. For this project, we seek that students will develop an overall software design that includes all the 4 modules and focus to implement completely the client profiles module.

Client profiles: used primarily by Client Services and Nutrition Services. Each client file will have a unique sequential number and have capacity to record client information including among other things: addresses and phone numbers, demographic data, contacts and medical provider info, medical and social economic indicators, ongoing record of any communication and interactions and alerts to communicate between departments and for future follow-ups, nutrition screens and medically tailored diet selections. We need the ability for staff to activate / inactivate accounts and to suspend /resume future deliveries. The database should have search capabilities to find profiles using different parameters as needed.

MTM (Medically Tailored Meal) Matrix: used primarily by Nutrition Services and the Kitchen. Our services are about providing meals that are specifically tailored to meet the needs of unique critically ill clients with many different diseases and treatment requirements. These specific needs are facilitated by ability of Nutrition to select several special diet categories for each client. These special diet categories are then used to select the most appropriate meal from several the Kitchen creates each day.

Routing and Delivery : used primarily by Client Services and Dispatch. We are a home delivery program, so each client will be assigned to a delivery route – currently routes go out weekly, however the database should allow for flexibility for various delivery models. Our clients require some predictability in delivery day and times while balancing the agency's need for the most efficient use of resources. Dispatch requires outputs to facilitate advanced packing of each MTM delivery. Dispatch drivers need access to address, phone and other delivery instructions in the field. The database needs to incorporate capability to track meal inventories,

verify the accuracy of the meal packing, and report delivery outcomes, while also facilitating in-field communication and flexibility to adjust and amend deliveries at will.

Reporting and Billing: Used primarily by Nutrition Services, Client Services and Development/Finance. In order to fundraise and support our meal program, we need to be able to pull many different data points about the persons we serve, have mechanism to bill fee-for-service contracts, and ability to report outcomes of our nutrition interventions to demonstrate our programs effectiveness. This module may also be wrapped into the Client profile module if a total of three modules are preferred.

A successful outcome is a mobile-friendly/cloud based software that fully integrates the end-to-end functions of all Program Departments: Client Services, Nutrition Services, Kitchen, Dispatch and Finance (for tracking contract clients and invoicing customers of our social enterprise sales). Functions would include real-time meal delivery tracker, on the fly van routing, and route optimization. It would provide options to reconfigure user interface, page layout and other functions as required. In addition, the software would include a bar scan system to maintain a detailed inventory of all meal types and alert staff when inventory levels exceeded or fell below predesignated quantities.

The students would significantly increase the efficiency and effectiveness of our program operations. The UMass Boston team and talent would be instrumental to helping us develop a data-driven culture across all departments at Project Angel Food.
NOTE: Some work has already been done by previous CS 682 Students. This project will be a continuation of existing work.

Resources:

- We would provide the team references to our current AIMS software, information about our program operations and flow, and time with key staff.
- We would also provide an office space with a computer and phone for one or two students, if they would like to work on-site in our Hollywood facility.

7.	**Title:**	AI Meets the Classics
	Admin:	Christopher Stephenson, TOPOS LABS - Boston, cstephenson@toposlabs.com

Description: Given your understanding of language, your creativity, and your passion for charting exciting new ground, use our cognitive AI engine ("Edison") to analyze a classic novel or world-known piece of literature.

About Topos Labs

Topos Labs is a Boston, MA-based, AI-powered text mining startup, focused on extracting intelligence from unstructured text. Our mission is to build a user-friendly reading machine that brings structure to, and extracts intelligence from, human-readable text to help solve big problems.

Our cognitive engine, called 'Edison', can be trained on new domains of knowledge in any language, quickly and without programming or data science. We convert language to math. Edison is AI for the masses.

Edison is not a shrink-wrapped application designed to answer a specific business need. Rather, it is a horizontal platform designed to enable the building of applications to support a broad range of compelling use cases across many markets. The platform is currently being used at several large, well-known brands in retail, media, research, government, healthcare, biotechnology, finance and security.

Project Framework

1. First step will be to select a well-known or classic novel/piece of literature. Discuss and negotiate with your teammates to select a culturally significant novel, such as Anna Karenina; The Great Gatsby; The Adventures of Huckleberry Finn; Hamlet; Moby Dick; Ulysses; Pride and Prejudice; The Odyssey. A more challenging option could include: The Bible, The Torah, and The Koran. Another approach could be to select one from the HRR Tolkien series, Harry Potter series, or Game of Thrones series.
2. Next students will build the ontology used to train Edison, to establish the baseline of domain knowledge, which will be adjusted iteratively throughout the project. Ontologies are developed in the form of simple spreadsheets that can be uploaded easily to train Edison on any topic quickly.
3. Next students will feed book text into Edison for processing.
4. Next students will create a simple chatbot or NLQ UI that allows an end user to query Edison about the text to learn more. Effectively, the students will be creating Literary 'Sherpa' for the text, which can be reused for other texts in the future.

NOTE: For simplicity, it may be best to generate a pre-defined list of questions (or FAQs) to take a prototype approach vs. attempting to build a truly conversational chatbot…Topos team can offer suggestions and guidance.

Project Goals

Topos hopes the student team will think outside the box, show us (and the world) something we didn't know about the book, let us ask questions to learn more. Visualize the trends and insights in new and interesting ways.

Successful Project will:

1. Expose some new and interesting things about the novel that we didn't know before.
2. Allow an end user to query Edison to learn more about the story. Who is the protagonist? What are his/her chief characteristics? What are the dominant themes in this novel? How does the lead characters' view of X theme, Y topic, or Z person throughout the book. How many times does Frodo experience the ring's evil powers? Etc.
3. Establish a newly created ontology (terms, topics and themes, etc.)

Tools/Languages

There are no requirements or hard rules about what tools and/or languages to use for the project. Suggestions include the usual suspects: Java, Python, SQL, JS.. More sophisticated and ambitious teams could use S2S, which is specifically designed for highly interactive chatbots.

Resources:

- Resources can be made available as needed.

8.	**Title:**	Recommendation platform for international higher education
	Admin:	Shawn Lestage, Enrolled – Vancouver, shawn.lestage@enrolled.com

Description: Enrolled is an education technology company that has designed a platform to increase the effectiveness and decrease the cost of international student engagement, recruitment, and applications with AI technology and machine learning. We match students, based on their academic profile and personal preferences, with 'best fit' institutions, degree subjects, cities, and countries. We are committed to helping students make the best possible education decisions.

Details

Through this project students will be expected to: survey potential customers (i.e. international students), build a working recommendation engine capable of matching prospective students

with their 'best fit' higher education opportunities, conduct user testing, identify and document UX red flags, and recommend features for future platform iterations.

Goals

The final goals are: for the students to provide a demonstration of the recommendation engine they have built and their proposed improvements (if time and money weren't an issue).

Resources:

- Access to Enrolled research and material.

9.	**Title:**	Cloud Gadget Container
	Admin:	Ron Ijack, Upstream Works Software, rijack@upstreamworks.com

Description: We would like to create a standalone cloud based container that will allow the loading of external and internal gadgets and for communication between those gadgets. The application will be hosted on Amazon Web Services (AWS) and will allow companies to subscribe for the service. Each company can then create different departments and users and and assign different gadget configurations to each department or user.

Business Requirements:

- Cloud based SAAS model (AWS)
- Users must log in and be authenticated
- Design must support roles and permissions for users. The following are some of the known roles:
  - User (agent)
  - Supervisor
  - Administrator
  - Site Administrator (super user)
- Subscribers will have the following model:
  - Organization
    - Business Unit
      - Department
        - User
        - Supervisor
- Each type of application user will have different properties/profile
- Must allow for multi tenanting
- Administration portal is required to manage:
  - Users - roles and permissions
  - Layout configuration
    - Per organization, business unit, department and user
    - Layouts can have tabs and each tab has gadgets assigned to it
    - Gadgets can be tab level or page level
      - Tab level - only displayed when the gadget is active
      - Page level - Displayed on all tabs
    - Grid based design capability for layout management
- Have ability to load gadgets based on URL
- Gadgets should have a mechanism to communicate amongst each other using the container
  - Gadgets should be able to publish events and data and the ability to subscribe to these events and get the data
- Must work in secure and non secure modes (http & https)

- Multi language support

Technical Considerations:

- All code must include unit tests (minimum 80% coverage)
- Micro service architecture
- Code stack to be determined by the team with documentation explaining reasoning for choice. Final approval will be needed by Upstream
- All code will be managed through GitHub
- Architecture plan is required prior to development that will be reviewed and approved by UWS
- To be developed using responsive design principles (adaptable to many screen sizes)
- Development team is responsible for quality assurance and ensuring the product is delivered with highest quality
- Third party plugins can be used but must be documented and under the correct shareware license (list to be provided)

Process:

- NDA must be signed between UWS (Upstream Works Software) and the team
- Kickoff meeting with the team to explain the project goals and process
- Use Kanban for the development process
  - Stories and project/process management will be using Jira
- Stories will be groomed and prioritized for the team
- Regular demos will take place bi-weekly
- Will have regularly scheduled time with the team to answer any questions they may have
- Will attempt to have a face to face meeting with the team if possible
- Defects will be dealt with as soon as they are found

Project Deliverables:

- Working application in AWS
- Documentation
  - Architectural diagrams - system design
  - All mockups
  - Style guide
  - User guides/manuals
  - Environment details / Hardware requirements
- All source code

Resources:

- https://www.youtube.com/user/UpstreamWorks - UWS Product videos
- https://shindig.apache.org/about_shindig_overview.html - Current container used by Cisco where UWS gadgets reside
- https://aws.amazon.com/ - AWS for hosting
- https://github.com/ - Code repository
- https://www.atlassian.com/software/jira - Process management