ENSE 405 Project report-out & lessons learned

Disclaimer: This template should be used to guide your discussion. The course facilitator asks that you be as open, honest, and professional in your responses as possible. Please know it is OK to agree/disagree with any concept, process, or idea discussed in this course. The knowledge gained from your open, honest, educated, and professional responses may be discussed at a future Engineering Faculty and/or "Canadian Engineering Education Association (CEEA)" event.

Project name

Provide your team's "name" and team member(s)

- Name: Team "Cellar Recipes"
- Team Members: Alok Paranjape

Project sponsor & course facilitator

Dr. Tim Maciag (ENSE 405 professor)

Business need/opportunity

- Can be re-stated/copied from your business plan document

Quoted From Business Case:

"The opportunity here is to link the two systems. Making a way to use the items in a food hamper more efficiently by recommending and planning out different recipes specifically for them. While software for managing a food bank and finding recipes from ingredients (or generating them with language models), this application is trying to fill a gap between them. In particular, trying to make managing food inventory in a portable (non-organization level) format, and having stored recipes rather than searching the internet for them."

Reflections on project planning (3-5 pages)

- State and discuss the United Nation's (UN) Sustainable Development Goals (SDGs) selected and your "why" for selecting the one(s) you did

  For my project, the UN SDGs that I went with were #2 (Zero Hunger), also partially #12 (Sustainable Consumption). Personally, I went with UN SDG #2 because I felt like it was most relevant to my project (particularly sub goal 2.1, about improving access by people in vulnerable situations to sufficient and nutritious food), and also relevant to the community of practice that it was being designed for (the Regina Food Bank).

  I also thought that #12 was related, but less so, considering the consumption practices wouldn't be as affected directly by that community, at least in the sense of food banks reducing food wastage as a general institution (particularly in how they manage food deals from larger grocery chains). If it was related to my project, I think it

would be more indirect. As well, Prof. Petry mentioned shooting for multiple goals at once, so I didn't just want to have one goal.

- Discuss key findings from your community research and understanding/requirements gathering (Community characteristics and technology configuration inventory)
    - Discuss your professional opinion of the processes and documentation used in this course for project planning. Did they help/hinder and how?

There were a lot of different options when considering the technology inventory. I probably should have expected it, but there's a lot of different software solutions for both food banks, and individuals organizing their food consumption. I initially started the project inspired by the "find recipe by ingredient" sites like(https://www.supercook.com) , but there were way more examples to draw from than that, such as entire software suites designed for managing an entire food bank (https://www.link2feed.com), or how recent developments in language models means they can autonomously develop recipes from just a prompt and some ingredients. Looking at all of the potential options, it really gave me perspective with how many different solutions there were to solve the problems I was looking, and how much larger the scope of those solutions could be compared to what I was attempting to make.

For the community characteristics, I mostly just went on the Regina Food Bank's website and looked through a lot of their activities and press releases, and tried to map each action to a specific community orientation, like we talked about in class. For instance, how their public mission lent itself to a service context, or how their frequent fundraisers mapped to a project orientation. I had to make a lot of assumptions though, since I had more trouble with fitting an established organization with multiple fronts into the categories we talked about in class.

Overall, I'd say that looking at the community characteristics was less helpful than the technology inventory, like, my project was sort of serving a "content" orientation, which was an extension of some their current practices with included recipes, but that wasn't something I was able to accurately put in an orientation without talking to Mr. Bailey afterwards. I sort of felt like a lot of our community orientations were too focused on digital habitats to fully apply something more physical like the RFB (even with their digital presence and communications systems). In contrast, the technology inventory was more useful, since it meant I could how high the ceiling for solutions was, and get some additional inspiration for how I could do my project.

- State selected north star & carryover customers. Why are these customers important to your project's golden circle (why, how, what)?

  I'd say that the north star customers for my project would be clientele who the Regina Food Bank serves, with the carryover customers being the RFC staff themselves. With respect to the golden circle, I'd say that they're part of how I defined each section. "Why" would be something like (The current system means that people who use the RFB get a bunch of food without context, and potentially aren't using it as efficiently as possible), with the "How" as (a way to more efficiently sort that food and find ways to use it), with "What" being (an application or web service that tracks those two factors). As well, since the food is being managed/distributed by the staff of the Regina Food Bank, there are potentially some ways they could benefit as well (such as knowing which items are in higher demand, or building on their current food classes and recipe systems)

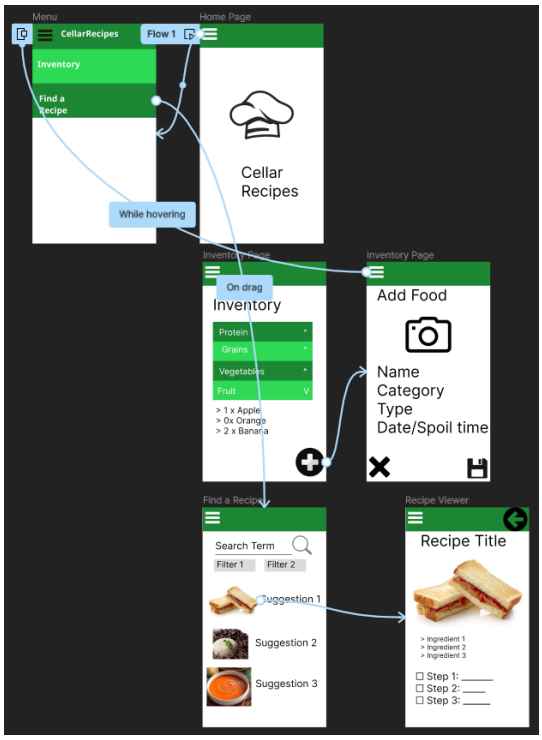
- Summarize assumptions made and constraints uncovered, re: drafting an emerging picture

  Like with the community orientations, I made a bunch of assumptions about the Regina Food Bank's operations when writing the "drafting an emerging picture" document. Like, looking at all of their website and activities (such as using a separate service for volunteer communications and donations), I tried to slot a bunch of what I could find into each polarity (like, what tools were synchronous/asynchronous, participatory/reified, etc). For instance, I assumed and put the volunteer hours asynchronous, and the newsletters as asynchronous, a lot of the actual volunteering is participatory, with the documentation and reports as reified. I also made a big assumption that, in spite of the current participatory/synchronous focus I assumed, they'd still be tolerant of a more individual and reified tool, like what I was proposing.

  The three constraints that I found were Time, Scope, and Technology/Knowledge, which were pretty accurate, all things considered.

- Discuss initial & the evolution of your technology stack selection, drafted prototypes, and initial Minimum Viable Products (MVPs)

  I made a few high/low fidelity prototypes, such as some simple sketches, and also some more active high-fidelity prototypes in Figma. They were sort of useful, but kind of ran into a wall when it came to recreating the layouts navigation in Flutter.
  (Image below)



I was initially thinking I'd try and develop and Android app after talking to Mr. Bailey, since he said that it would probably fit the clientele best (and not require internet access like a web application), and I was also generally curious about Android development. Unfortunately, I ran into some issues after installing it, with actually compiling and running projects on my laptop, even after subsequent re-installations. I changed my technology stack to Flutter/Dart after that, which was able to install and run more smoothly. I did have some issues with relearning it (since I took ENSE 375 a year back), especially with how nested widgets can get. There was also some debate about using SQFLITE or some other database, but I ultimately just went with a static set of views and check-boxes, since I figured that would be more representative of the concept while still being feasible to set up.

My MVPs in the initial documentation were kind of vague, like, mentioning being able to different user stories (like, storing food or viewing different recipes, or searching with different factors). I probably should have been more detailed and split them up into something more manageable.

- Provide images/screenshots where appropriate/needed

Reflections on project results (4-5 pages)

- Discuss what you created. Provide key images/screenshots illustrating core functionality
    - Review your initial "Planning and initialization" video created for the first deliverable. How close did you come to realizing the solution/product you initially envisioned?

I wouldn't say that I got very close to the software solution I initially envisioned. Especially with regards to the food tracking system which is more of a static checkbox rather than the database with additional information that I was aiming for (without quantities or spoil times, or tags, there's just categories). I think I was able to get somewhere with the recipe viewing and ingredient screen, but even if it is expandable, it's still a bunch of static pages, and not a database of recipes like I initially wanted.

- Summarize software design activities and findings. Ensure you discuss how you/your team either linked or envision links to design ideas back to topics discussed in class lectures

The two main topics that I was trying to bring from the lectures was the "Data to Knowledge Process" and "Rhizomatic learning".

I'd say that the main goal of my app was taking data and trying to get knowledge from it. Like, when I was talking with Mr. Bailey, part of the information he was able to share with me was a spreadsheet of common types items that get included in each bundle. I'd sort of count this (or at least, the contents of a single bundle) as data, since it's just a list without much information or other context.

A	B	C	D	E	F	G	H	I	J	K	L	M
Animal Protein	Halal Meat	Grains	Vegetable	Fruit	Bread/Baked Good	Dairy	Candy	Miscellaneous	Soup/Stew	Vegan Foods	Premade Meals	Baby F
ground beef	chicken	lentil	canned (all)	apple	bread loaf	milk	cookies	mustard	soup cans	V pepperoni	TV dinner	formul
turkey whole	turkey	oat	carrot	orange	buns hotdog	cheese	popcorn	ketchup	soup dry	sausage	pizza	snacks
ham whole	ground beef	split pea	onion	banana	buns hamburger	sour cream		relish		V ground	agency meals (CK)	baby f
chicken whole	lamb	chickpeas	celery	grape	bread miscellaneous	yogurt		mayo		V pizza		
chicken pieces		oat groat	peppers bell	waterme	pastry	egg		syrup corn				
deli meat large		pasta	peppers hot	cantelou	cake			syrup maple				
deli meat sliced		rice	broccoli	honeyde	pie			energy drink				
canned meat		flour	cauliflower	canned (all)				pickle				
sausage		cereal bar	onion green	pie filling				bbq sauce				
hot dog		granola bar	okra	juice large				salad dressing				
duck		canned bean	pumpkin	fruit cups				oil canola				
lamb		cereal breakfast	zucchini	apple sauce				oil olive				
fish whole			yam/sweet potato					oil other				
fish portioned			canned veg					vinegar				
seafood			frozen					sugar				

I guess what I was trying to do with the application was try and add context to the list of food (i.e with the additional database information, quantity, spoil times, etc), and eventually try and get some knowledge out of it (the interactions with different combinations leading to different recipes, and the specific knowledge of preparing ingredients in each recipe). I wasn't able to get a proper database working other than with the names and dates (though, they weren't properly displayed), but I was at least able to get the recipes working.

For Rhizomatic learning, I'd say that it sort of coloured the way I looked at developing my application with specific features and techniques with how I developed the project iteratively. Like, I'd try and Plan out different features (and the relevant learning material for implementing them), Do by putting time in and trying to implement those features, Study by evaluating if what was made was sufficient to meet expectations (or otherwise able to built on to eventually do so), and Act by deciding to continue iterating on that spefic method later on (or else, trying in a different direction). As well, I'd say that the Cynefin framework was particularly

accurate when when going through the process, since it things went from Complex to Complicated to Chaotic and back as I went through different planning features, figuring out how to implement them, actually trying and then evaluating.

- Summarize how you felt about this project (likes/dislikes), from your experiences with the technology stack selected, translating prototypes into real solutions, and the creation/realization of your MVPs
    - Summarize what went well during the project

Even if things didn't go as well as I hoped there's things that happened I'm still happy about. For instance, I'm glad that I was actually able to talk with somebody from the RFB, and get some actual context about their operations after making all those assumptions. I'm glad that I was able to relearn Flutter and make a properly scaling and navigable application, especially since its more complicated than what we covered in 375.

    - Summarize what not went well during the project

A bunch of things didn't go as well. For instance, a lot of work didn't end up in the final product. I spent the better part of two weeks trying to get Android Studio to work and run properly, and had nothing to show for it due to the sheer number of errors. Even when I got Flutter to work for page navigation and selection, I still spent a bunch of time trying to get a SQFLITE database to work (and it did, but it was the simplest possible CRUD operations in version cr3). I ultimately had to drop it in favour of just doing a static page/UI instead. As well, there were a bunch of slip ups with the documentation portions, such as when I missed some of the files for Scrum #1, or assumed the wrong deadline in Scrum #2.

    - What would you do the same on future projects?

I'd probably want to continue using Flutter, especially considering that it's still quite popular as an cross platform SDK, and generally useful for a software engineering career. As well, I think this was a good opportunity to see how important it is to identify a community/client and interact with them directly. I also felt like a lot of of the documentation was more than useful enough to justify the work, such as the Business Case/Project Requirements/Technology inventory, so I'd probably try a lot of the project initialization documents as well.

    - What would you do differently on future projects?

Obviously, not try and learn a new language/development platform in such a short time frame, considering how the time I lost trying to get Android Studio to work. I already mentioned how useful identifying and interacting with Mr. Bailey was, but I probably could have gotten more feedback than the singular meeting and email exchange. I would also spend more time requirements testing and doing refactoring, since I mostly focused on getting it to work without the code being CLEAN or being properly organized (for instance, how many redundant cases are in the card widgets). Finally, I felt like some of the goals/planning I did initially had to be abandoned

once I got to actually got to development, so I'd probably try and take steps to properly assess the feasibility (for instance, some examples of it actually running) of a goal in the actual environment before committing to it.

- Discuss opportunities and design ideas for future work

  There's a lot of potential for future work with this project, especially with what I wanted to compared to how things ended up. For one thing, I'd like to implement a proper, working inventory system with actual quantities and dates, compared to the context-less, writable list database that I made for version cr3, as well as actually integrating it with the recipe selection screen. In the recipe selection screen, having collapsible, non-boolean ingredient tallies, as well as a search bar/filtering system the way some of the sites I was inspired by have (such as different ingredient restrictions, or substitution). Obviously, for the recipes, having a single page which pulls from a database to display recipes, rather than having the static page setup that's currently implemented. Overall, the main ideas for future work are fully implementing what I initially wanted to make.

General reflections on the class & project experience (3-5 pages)

- Before taking ENSE 405, were you aware of the UN SDGs?
  - Yes/No – Please elaborate

  Before taking 405, I was vaguely aware of UN SDGs from them some mentions in previous classes (and possibly Prof. Petry bringing them up), but it was not nearly anything to this extent. As well, in high school, the We Foundation had a pretty strong presence, so a lot of assemblies and fundraisers were in service of specific goals every year (if I remember correctly, the last two were #6 and #4, respectively). I definitely didn't think of them to the extent we talked about in this class though, at least in regards to them potentially being guides for engineering projects. Before this class, they were more like disparate humanitarian fundraising efforts rather than something to be designed for in a project in an engineering context.

- Typically, before taking this class, when you engineered software solutions, were you concerned with areas encompassing the UN SDGs?
  - Yes/No
    - If yes, provide some past examples and explain
    - If no, do you have examples of past engineered works that you (co)created that could address one or more of the UN SDGs

  Before taking this class, I wouldn't say that I was really thinking about UN SDGs with respect to my projects. I might be reaching a bit, but I'd say that the laser telegraph system I made for ENEL 351 back in the winter was

the one project that could of related to SDG #9. I wasn't really designing it for power constraints, or to work in an embedded context, but it could theoretically work as a low power signalling system in an isolated community (in the same vein as a signalling lamp), which is sort of related to #9.c's goal of improving access to communications technology. Other than that, I wouldn't say that the other projects I worked on were really designed in with the SDGs in mind or even generally applicable to UN SDGs (such as the recipe site for CS 215, or the battleship game for ENEL 350 or the music website for ENSE 271).

- Did learning about the UN SDG(s) help you understand better understand your role and responsibility as an engineer to society?
    - Yes/Neutral/No – Please elaborate

I'd probably say neutral leaning towards yes. Like, on one hand, engineers have a duty to the public (and the environment), and UN SDG goals are all about the problems the public (at it's greatest global extent) are facing now and in the future, at the highest possible level. On the other hand, I also feel like the SDGs can sometimes be too broad to properly fit with a given engineering project (like, even massive projects might not necessarily make an impact towards a goal). Like, the greater decisions surrounding the project and its direction are useful for understanding whats at stake as an engineer, but they're less relevant for immediate decisions.

- What was your experience(s) in engineering your specific software solution to address the UN SDG(s) selected?

I'd say that the UN SDGs were more important while initially planning my project than anywhere else. Like, I was coming up with project ideas, and felt that Zero Hunger was relevant, but I wouldn't say it went much further than that, especially with the scope of the project. Like, even in the most optimistic case, where everything in my project worked like I wanted it to, it wouldn't really be much more than some people who use the Regina Food Bank (or any other similar program)'s services using food slightly more efficiently than they were before, contributes infinitesimally towards the larger goal.

As well, I'd say that the UN SDGs were less relevant to development processes, when it got to coding and in-app desgin. Like, It didn't really matter that the larger direction was about Zero Hunger when I switched from Android Studio to Flutter, or any of the decisions I made with databases and layouts within Flutter.

- As a future engineer, what are your thoughts on the UN SDGs as a whole? Do you think they can help or hinder our work as software engineers?

As said previously, I feel like UN SDGs are extremely relevant when considering about an engineer's responsibilities to the public and the environment (and the projects they choose to pursue, and in what direction), because the SDGs are all about that, on the ultimate global scale. That being said, I have some concerns about the scope of SDGs, in the sense that a most engineering projects aren't nearly big enough to have a quantifiable impact on the goals/subgoals/indicators of a given UN SDG.

I think that they're definitely helpful when pointing engineers in a direction when picking a project and figuring out requirements for the larger direction of that project. On the other hand, SDGs seem less useful and more of a hinderance when it comes to actual development, since the moment-to-moment engineering decisions simply aren't the right scope to affect anything covered in an SDG.

- Should we use the UN SDGs to guide our work or is our work dependent on customer requests, regardless of the UN SDGs?

Reflecting on what I said earlier, while UN SDGs are relevant to engineering as a greater guide, I still think that they're also too broad for specific projects, including dealing with customer requests regarding those projects. For instance, a project about diagnosing medical conditions from medical imaging might be generally about #3 ("good health and wellbeing"), but I think that the subgoals under it aren't enough to properly direct a project, and that user interaction and feedback (for instance, what the customers say about how useful the product works, or some UI needs) are way more necessary to guide how project goes with later development. I don't think that looking over SDGs is analogous to handling customer requests, in that they're at totally different levels of development.

- Will you use your understanding of the UN SDGs in engineering solutions in the future?
    - Yes/No/Maybe – Please elaborate

  Maybe. For one thing, I'll probably try and categorize projects I work on to try and fit under a certain SDG when applicable, but it really depends more on what projects I'm doing and in what career. There are definitely engineering jobs where SDGs are more relevant, but not to the point where I'd say there's a guarantee about definitely using them in my engineering solutions.

- Will your experience learning about the UN SDGs inform your career path decisions in the future?
    - Yes/No/Maybe – Please elaborate

  Maybe. With that being said, it might be possible to try finding those jobs and going on a career path that more heavily involves UN SDGs. I'm planning to get further education in software engineering, and part of that may include some of those roles. For instance, a project that I was proposing was talking about security in power

systems and other critical infrastructure, which seems relevant to SDG 7 (particularly 7.b). I don't know if it will go anywhere, but it's at least more SDG related than the previous projects I've worked on.

- Provide any other comments on the project