ENSE 405 Project Report-out and Lessons Learned

Project Name and Team Members

Community of Meal Planners Forums, Kelly Holtzman

Project Sponsor

Dr. Tim Maciag (ENSE 405 Lecturer)

Business Need/Opportunity

The meal planning application market appears to consist of convenience tools rather than educational tools, limiting what platforms and content members of the community have available. There exists side platforms and content that only those “in the know” access and share. These two situations limit the demand for education and consumption practice change because there is no collective place for members to voice concerns and ask for change.

This project provides members of the meal planning community with a transparent space (forum) for relevant discussion and critique of recipes, practices, and tools that can be received by meal planning companies and tool developers.

Reflections on Project Planning

United Nations Sustainable Development Goals selected. This project aims to support sub-goals 12.7, 12.8, 12.A, and 12.B of Goal 12: Sustainable Consumption and Production patterns of the UN’s 17 Goals for Sustainable Development. Goal 12 was interesting to me as the COVID-19 pandemic has changed so drastically the way we live; particularly, how we produce and consume food has really changed: production has slowed down and the impact of reduced emissions and consumption led to a short period of wildlife and environment recovery. Though only for a limited time, the immediate recovery of the planet gave me hope we could all live sustainably at some point, if we made more efforts to do so.

Key findings from community research and understanding. Meal planning (a consumption practice) has become increasingly popular during the pandemic. I began investigating what was available for meal planners: tools, instructions, leading people, and found that most platforms support achieving personal goals, like eating a better variety of foods for weight loss, by providing users with convenience tools, like online recipe catalogs. Those platforms which show the importance of sustainability, such as including educational material alongside recipes, are missing the convenience tools of the more popular meal planning applications and vice versa.

The available meal planning apps are primarily of the Content and Individual Participation community orientations. The tools, recipes, and promises of personalized plans and advice comprise the Content orientation: advice being curated by the application developers and business management. The prioritized aspect of personalization makes most applications about Individual Participation and so the apps do not provide information about other people using it and how well they are faring.

Selected north star and carryover customers. I found that the most important customers were the meal plan consumers, the meal planners, and those persons developing the meal plans or apps, the developers. Obviously, the meal planners are the ones who drive market demand and are important sources of income for developers. Carryover customers include lurking people, who may or may not be planning but have come across the app, and people who enjoy educating others on how best to meal plan. Lurkers and educators are a secondary group of people who increase the usage statistics of meal planning applications and may or may not be sources of income.

Drafting an emerging picture. The gap between personalized tools and sustainable practices in meal planning apps is not caused by inability to add new tools to the platform, but rather caused by the inability *for the public* to add new tools to the platform for the benefit of everyone. Additionally, there exists no such tool where the user could switch between the tool and platform to achieve the same result as a fully-integrated platform.

The project began as an idea to merge the ability for people to suggest new tools for the platform with an existing platform. Unfortunately, all meal planning applications in use are proprietary and closed to modifications. The next best idea was to create from scratch a space for discussion, a forum, as exemplified in the Adobe XD prototype. The prototype revealed the work necessary for the first tools for the forum:

1. **People would need to communicate with one another.** A message system and database would be needed to manage and store messages appropriately. Figure 1.
2. **People would need to manage messages.** Messages should be labelled appropriately for searching and understandability purposes. Figure 1.
3. **People would need to manage themselves.** Inappropriate messages should be handled as appropriate for the community, introducing the need for community leaders to step up responsibility and guide action. Figure 2.
4. **People would need a way to become community leaders.** An application and vetting process would need to be designed and evolve over time. Figure 3.

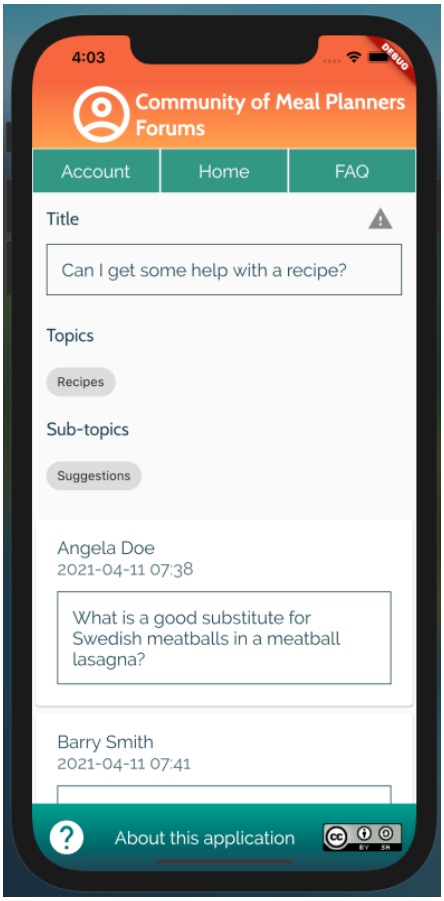

Figure 1. A way to communicate.

Each of these tools/processes comprising MVPs for the application were split accordingly in the Project Scope Statement and Sprints.

The technology used to implement the forum was chosen for convenience and ease of use. As discussed in the Business Case, the Google services Flutter and Firebase were optimal choices for a single-semester project; Flutter is cross-platform and it's popularity with the mobile application community means that the framework is mature and well-developed. Firebase is a fast and simple way to start up a database and maintain it's security and contents from a website over

self-hosting and maintaining a database, which is considerably more work.

The chosen MVPs and technology used to implement the project were not exchanged during development. The MVPs were reordered by priority of content most relatable to the course context, such as handling malicious information and building a leadership team, over useful end-points for integrating the application with existing apps.

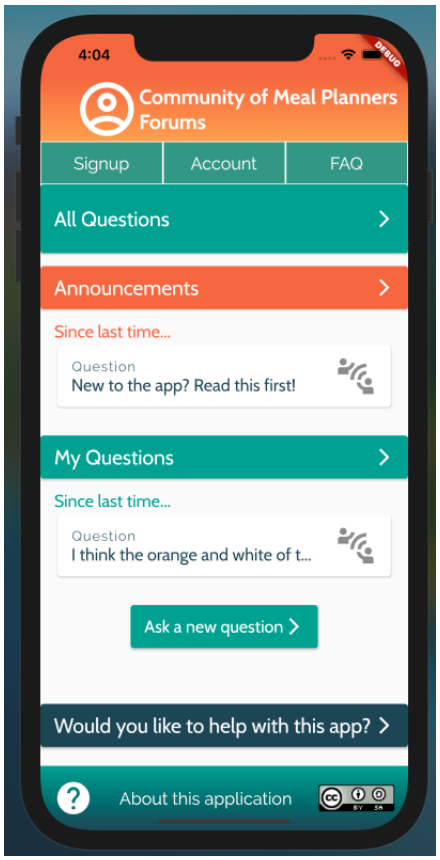

Figure 2. An option to join leadership.

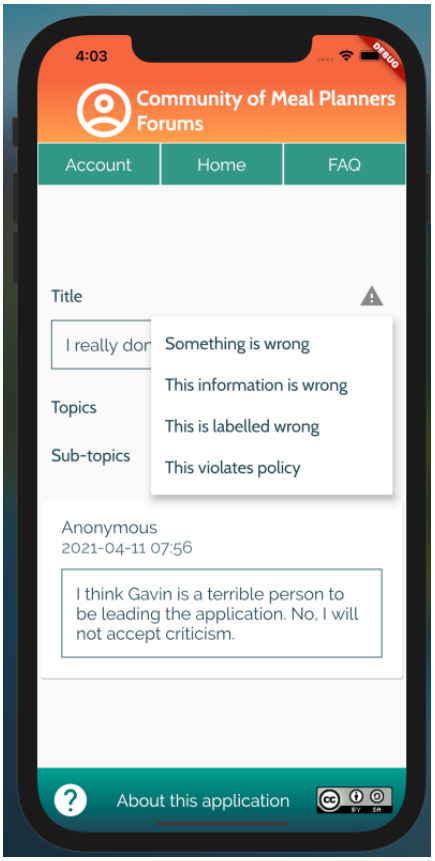

Figure 3. An option to flag a message for review.

Reflections on Project Results

Best project results. The use of the prototype to directly make the application's interface was most useful as interface development is time consuming and the appearance subjective as to what makes a good interface. In the case of this project, the prototype almost directly represented the application's final appearance.

The use of Google services to develop the application was a very good idea: with a short time frame for development after project planning, Google's services and the service's use of industry-familiar paradigms and algorithms was most convenient and applicable. The services used are so mature as to continue to be relevant after the project's completion.

The MVPs were refined from instructor and student feedback. Without incremental feedback during project development, the final project would not have been as effective or related to the course context.

Poor project results. Obviously, a single semester is not enough time to develop a fully integrated and useful product that people can use and effectively solve problems with. This is a limitation of the course that cannot really be remedied.

The most stand-out result of using Google's services was the minor lack in maturity of the Flutter framework for web development. In particular, a most frustrating issue was how bleeding-edge Flutter for web is:

1. A recurring issue was the updating of web packages and subsequent breakages caused by updates that forced me to investigate for solutions on my own. I had several outside party tasks I kept an eye on during development and waited for solutions.

2. The use of the same code for both Android, iOS, and web devices is frustrating and explosive to maintain. Consider the following solutions for development:

    a. For every screen, listen for the screen size changes and change build methods accordingly. Results in two or more builds per widget, and the duplication of widgets between different builds. Some extraction could be done, but increases complexity.

    b. For every widget, listen for the screen size changes and change the widget size on rebuild. Results in every widget having instructions for size. Frustrating to see what could have been a one-line widget become 5 lines or more. Causes more widget rebuilds, compromising Flutter best performance practices.

Software design activities and findings. The implementation of the app leadership/administration process is linked to course ideas of handling and resolving mis/disinformation: the community discovers infringing content and reports it at large - encouraging polite online manners and punishing people who abuse the application. Coming full circle, the more the people of the application are involved and feel involved, the more they will want to become leaders themselves, which enforces the Community Cultivation orientation that I believe would benefit meal planners and developers.

Current meal planning apps create a feeling of complacency. It is all too easy to become used to following app instructions and let it make decisions, decisions that could be poor for the environment in exchange for convenience. The Community Cultivation orientation relies on developers who plan and make the tools that the community uses. The developers should provide educational material to community members to help them break free of complacency and think of sustainable practices and critical thinking for a better future.

Minor course topics explored in the application design include some gamified processes: developers/leaders are encouraged to address calls for reviews by marking reviews as todos, and once the review is complete, they can delete the call from the list. A clean "inbox" of reviews is both aesthetically pleasing and lets the leaders set goals for themselves (i.e. "I will do this today"). People working towards joining the leadership team can consider answering questions and flagging inappropriate content as stepping stones towards their goal - the more they participate, the more they obviously care about the application, and the more appropriate they become for handling greater responsibilities.

What would you do the same on future projects? I often struggle with feeling I have not created sufficient documentation to start a project. On future projects I would definitely consider creating community research and understanding documents. The research documents are a strong way to show how different ideas, designs, and the final product could influence the community in the future where the project is complete.

I would continue to use Google's service to implement mobile applications. The ability to program in one language and compile for separate application deployments within the same framework is very valuable for short software projects. Google's popularity and framework documentation is top tier. None of Google's services sacrifice good design practices or higher programming concepts for simplicity.

What would you do differently on future projects? If I were to do a mobile and web deployment again, I would first dedicate myself to writing the same code twice for every widget. I did not really dedicate myself to making my web app work as well as the project mobile apps, as Flutter web development is not mature at the moment. Despite my frustrations with Flutter web now, web development will very soon become stable and mature. Perhaps the issues I outline in the poor project results will have better solutions in the future.

Opportunities and design ideas for future work. Some minor MVPs/tasks are outlined in my project GitHub issues. These include work that was not finished this semester and work that could be revised before the next round of app development. Such work includes:

1. **Adding account pages and interactions.** People using the app can read about other people and what's important to them.
2. **Minor quality of life improvements.** Improvements include the addition of questions to the FAQ page, and the addition of a WYSIWYG to questions once one is ready for web development.
3. **Integrating the application with an existing meal planning app.** No existing application is available for extension; this project would best be adapted by an existing app needing something new to boost user experience. This is the most important thing for the application to really reach people and be useful to them.