Project 3 Group Retrospective

331-902 Team 3 or Group 9

Rushil Jayant
Riley Wenckens
Rahul Rajendran
Elvis Chen
Diego Martinez
Laith Bohsali

Executive Summary

Over the past month and a half, our group has worked on developing a new POS (point-of-sale) system for Rev's American Grill. The product we developed is a web-based application developed with a Vue framework front-end, and a Django back-end connecting to a PostGreSQL database. We have included many features in this project, including a menu board, customer and cashier order screen, a manager screen, a google translate service, and authentication with Google OAuth. Our aim was to make the system more efficient, accessible, and robust. Below will be an overview of our retrospective summary working on this project, giving an overview of this document.

The product analysis section talks about many important features, including an intuitive customer order screen and a visually pleasing manager analytics section. These are areas we consider well done, but we also found areas of improvement, like the kitchen and customer status page. Going forward we will use more industry standard API's and extend our caching to improve runtime. On top of this, we will aim to expand our user base and add many more features.

The work analysis section goes into detail about our navigation of an unfamiliar tech stack. We had to implement an Agile approach to quickly learn and implement on the go. Deploying our individual front and back end components and connecting them together required some problem-solving, research and a lot of teamwork. We had many lessons in time management, communication and technology skills that gave us relevant industry knowledge and skills. Both of which we will definitely use in our careers.

The going forward section reflects on our journey by highlighting our lessons again, emphasizing the importance of being an effective team member and how to translate that into working with a large scale team. Working with new methods and strategies is always difficult but there are ways to

manage the complexities and deliver a well designed and thought out result. As we move forward, we will refine our collaboration skills and use whatever new skills we learned in the future on future products.

Our retrospective meeting was held on May 1st. During this meeting we discussed our reflection on the project as a whole and promoted discussion about what went right and wrong. Some key points from the agenda, we were able to praise the parts of the project that were great, we evaluated the scrum master's performance, we reflected on personal growth and lessons learned, and we discussed workload distribution and concerns.

Our retrospective survey was given out to members of the team to derive valuable feedback and insights, giving us a deeper understanding of all the different perspectives. Many of the questions included information on the enjoyment and challenges of the project, the opportunities of improvement beyond base requirements, the lessons learned, the topics for discussion for our meeting, the suggestion of improvements, the evaluation of tools and tech used in the project, and suggestions for improving collaboration in the project. In response, many people highlighted the joy of learning new things and many voiced frustrations about the project or about collaboration together.

Product Analysis

Overall, we are pleased with the final product that was created. Some highlights of the product include the customer order screen and manager analytics page. We believe these features work really well, both from a functional and design perspective. Now, for a brief run down of these features. The customer order screen was designed to be an easy-to-use and appealing UI capable of handling a variety of payment methods. As a team, we really like how it turned out. Similarly, we are pleased with the design and functionality of the manager analytics page. The datatables are paginated nicely and present helpful information in a way that is not convoluted. The color scheme of the website is consistent with Texas A&M's styling, which we believe will make a great fit in Rev's American Grill. We put extra emphasis on these pages in response to user feedback and the previous iteration of the project.

Some features that could be improved include the kitchen and customer status page. Though the functionality of these pages is present, the styling is rather simple. Specifically, these features could be improved by providing additional information to the kitchen staff and customer respectively. For example, it would be nice for customers to see the ETA of their order. It would be computed and updated based on how many items are on their ticket and how busy the kitchen staff are.

For version 2.0, we hope to integrate more industry standard APIs. We believe this will give us more practice and better exposure to handling API calls. Some competing groups integrated Spotify API or OpenAI API. Though it may be slightly out of the project scope, we have learned to see APIs as a powerful tool in web development. We also would like to extend caching to more pages so the response time is minimized. We tried caching out for the manager page, and it did provide a significant reduction in the response time. Since we were limited by the overhead and response delay associated with the ORM and hosting platform, caching presented a good solution.

As a team, many valuable lessons were learned. We all had the opportunity to grow our critical team-skills such as communication, conflict-resolution, and time management. This will be of great benefit in future software projects and in the workplace. Though we all had python experience, nobody was familiar with Vue prior to the project. However, over the course of the project, we developed proficiency in Vue and have learned to appreciate its approachable and versatile nature. Having this framework in our toolbelt will be of great benefit in future hackathons and personal projects. This project allowed for high freedom of design and development, which served as an opportunity

Work Analysis

We started our project with a vision of making a web-based point of sale system that was web-based, accessible, robust, user friendly, and included everything that we did and learned from Project 2. Most of our team members were unfamiliar with the technology stack we chose (Django backend with Vue frontend), as well as the associated testing frameworks and libraries (pytest and vitest).

To begin, one of our biggest challenges as a team was to figure out how to deploy the front-end and backend components so that it works on anyone's computer and sets us up for future development progress. We initially used Vercel for both frontend and backend hosting, where the Django backend would act like an API for the front end to use, and the front end would be set up with Vite integration that Vercel has inbuilt. We initially thought this was an issue with Django settings during configuration, but after many changes on the Django configuration, the problem did not cease to exist. To solve this issue, we took a broader approach to the entire problem, where we treated the backend and the front-end to be separate components, and after a conversations with the TA during the sprint demo and finding online tools that help with hosting a Django backend, we were able to settle with render.com as a way to act like a host for our Django HTTPS API. Here endpoints are invoked by the Vue.js components and then display everything on the main webpage as well as other subpages.

Furthermore, we found learning the different technologies in regards to programming languages, and testing was difficult because as a group. Before this project, most of us did not know how to do these technologies for an application with formalized deliverables and standards. To meet this challenge head on, we took full advantage of the agile development methodology, where we were able to set actionable stories, which gave us targeted goals, and used the standup meetings to discuss not only blockers, but also learning resources for learning different technologies.

If we could do the project all over again, we would make sure that we gave ourselves more time to do many stories. As upper-division computer science students, we have multiple commitments that are in need of balancing, but we believe that we should have intra-sprint deadlines, rather than designate the first week of the sprint as planning, and the second as working weeks.

In terms of team communication, we used a private Discord server, which is a group chat with additional features, which include voice chat, screen sharing, and multiple chats with the same groups. We were able to use it fully to share project information, and to make our conversations more clear. It also enabled some team members who are very active on the app to respond faster to issues. Our main challenge with communicating with other team members was the lack of availability for team members for synchronous meetings outside of class. Although we were able to meet when absolutely necessary, we were unable to hold regular synchronous meetings outside of class, in-person or discord. We believe that we could have set up a time initially as a group when starting this project, preferably in-person through the sprint.

For balancing load between group members, we believe that we could have been more cognizant of team members' strengths and weaknesses, both technically and non-technical components. Currently, we are successful in having everyone in the team satisfied with the level of work that they are currently assigned, but improvements can be made. We should have assigned tasks to specific people as soon as the planning meeting ends, and then discussed who is doing a certain task based on their prior experiences. Also we should have been more accurate with our time estimations by discussing the amount of time a task should take as a group, then multiply that number by 2-3, just to be on the safer end of planning. We also think that we could have made the technical responsibilities of the scrum master more clear, rather than have the tasks for everyone decided on a volunteer basis.

Going Forward

There are many lessons that we learned as a group that we would carry for future projects, specifically in how we should organize team projects, learn new technologies fast, and iterate over feedback or issues. This project taught us how to work in a SCRUM team following the Agile Project

Methodology, and how we organized projects as a team so that we meet all the deliverables and deadlines throughout the past six weeks.

One of the most important things that we learned from this project is effective team organization. Specifically, we learned that using a Kanban board with stories that include a "definition of done" is important to clarify roles and responsibilities of team members in an Agile group. When everyone has stories assigned to them, it makes it clear what they need to do to complete a particular task, and what steps they need to take to do so effectively. In future projects, we plan on trying to implement Agile or a similar framework that emphasizes problem solving compartmentalization, and clear sub-tasks that make it easier for larger teams to work together to solve a larger goal at hand. Clear deliverables are important for team success because in this project it enabled everyone to work together and complete their agreed upon parts without much confusion on what completion of subtasks means.

Regardless of the kind of technical projects, whether individual or group, for a class or a personal project, we were able to reflect upon the fact that we learn new technologies (Django, Vue, Pytest, Vitest), in a relatively short amount of time without direct instruction. As a group, we felt we were able to succeed in this regard because we collectively had a growth mindset for grasping new technologies. While completing the stories that required knowledge of these technical skills, we believed that we could learn these skills by coding along and referencing publicly available documentation, and most importantly by asking each other for help. This growth mindset, which gives us the confidence to learn anything given enough time and hard work, enabled us to make a product that we did not know how to make a few months ago, from scratch.

The Agile Project Methodology is also one of our biggest takeaways from this project, and a skill we would plan on using for future projects, and most definitely for our future careers. Learning this methodology helped us bridge the gap between the way people in industry work in software projects at scale. With a focus on fast and iterative development, we plan on using the methodology for our needs in the future for projects that have fast and evolving deliverables to increase team throughput and productivity.

Appendix 1: Retrospective Meeting Agenda and Minutes

The following showcases our team's retrospective meeting, the agenda for the meeting, and the important key points discussed in the meeting (the minutes).

Date: May 1st
Time: 7:00 PM
Location: Zoom Meeting
Duration: Approximately 1 hour

Attendees:

- Laith Bohsali
- Elvis Chen
- Rushil Jayant
- Rahul Rajendran
- Diego Martinez
- Riley Wenckens

The following topics were discussed during the meeting in the order in which they appear:

1. Showing praise to certain aspects of our project, group members, or improvements we made compared to the last project.
2. Evaluating the Scrum Master's performance compared to the last project.
3. Where has everyone grown? Discuss what you have learned, what went wrong previously.
4. Workload and work distribution concerns. Any last minute concerns

```
The following are the taking of minutes during important parts or key
points in the meeting: these minutes were scribed by Diego Martinez
for the entirety of the meeting.

[7:00] Everyone was on time for the group call and present.

[7:05] After some chatter, the group decided to assign roles for the
scribe and the person who was responsible for completing the code
documentation. Diego Martinez was chosen as scribe and Rushil Jayant,
our scrum master, was chosen to complete the code documentation.

[7:08] Meeting has officially begun. Topic 1 was discussed. In
general lots of praise was given to certain key members. These
included both Elvis Chen and Rahul Rajendran.
```

[7:15] Topic 1 is still being discussed. Specifically, what everyone likes in the project. The most important thing that was shown to be liked was how modern it looked. Of course, there were specific aspects that were praised, like the kitchen screen.

[7:19] Topic 2 was then discussed. Our Scrum Master was put on the hot chair as we talked about his performance. Overall, there was a general consensus that he has improved dramatically compared to last time.

[7:29] Topic 2 has now ended and there was a small pause in the meeting due to a team member's connection cutting out.

[7:31] Topic 3 is now being discussed. We all went around in turns and said what has made us grow and what we have learned from our time working in a group. Notably, everyone feels more comfortable working in a group as well as working with technologies that we do not have experience with. We also discussed how we struggled with the deployments and what we could do in future projects to alleviate these concerns.

[7:45] Topic 4 is now being discussed. A few team members were expressing how they felt about being overworked or rather pulling too much weight compared to the others. This was noted. Others apologized for not contributing more time to the project. Overall, everyone was understanding and it was fairly constructive. Showing how the team could have done better and been more accommodating to others time. We also said that we are all good for the celebration of learning coming soon.

[7:55] Everyone said their last remarks and the meeting ended.

Appendix 2: Retrospective Survey

Survey Questions

- Name
- What parts of the project were fun and not so fun?
- Do you feel that this project provided room to innovate and go beyond stakeholder requirements? How so?
- Describe all personal lessons technical/non technical that you learned from this project and feel would be of value to share with the team
- What are topics you want addressed during the retrospective meeting?
- What were the aspects of the project you found most challenging? Why was it challenging?
- Were there any specific tools or tech that you felt were missing or improper in this project? Could any changes have been made?
- How could the team's collaboration and project management processes be improved?

Survey Responses

Name	Riley Wenckens
What parts of the project were fun and not so fun?	The coding portion of the project was a good experience. I liked working with Vue, Django, and tailwind to build a web application. The hidden requirements, unclear rubrics, and overload of unnecessary tasks made the project not so fun in certain areas.
Do you feel that this project provided room to innovate and go beyond stakeholder requirements? How so?	We designed our web application to be extremely dynamic. For example, if a manager wants to add an item to the menu it will automatically display in other areas, such as the menu board. Furthermore, the stakeholder will appreciate the use of provided accessibility.
Describe all personal lessons technical/non technical that you learned from this project and feel would be of value to share with the team	Prioritize time management and communication when working with a group. This allows team members to work together efficiently without a bottleneck.

What are topics you want addressed during the retrospective meeting?	We should discuss the approach we would take if we had to redo the project. I would also like to take another look at other groups' implementation because that will provide a good idea of how our project stands.
What were the aspects of the project you found most challenging? Why was it challenging?	Testing and changing requirements, it leaves the development team confused on what they should do. Since this project is part of a class and we have other classes, its sometimes challenging to balance the workload.
Were there any specific tools or tech that you felt were missing or improper in this project? Could any changes have been made?	I think the framework we chose was perfect, python is an easy language and vue allowed us to create good looking SPA. I really liked the REST framework provided with Django and thought it was easy to work with from the front end.
How could the team's collaboration and project management processes be improved?	We could have communicated more design details with our group members.

Name	Laith Bohsali
What parts of the project were fun and not so fun?	The development of the product was fun, the grading and assignments were not fun. Working with a team was fun at times but definitely frustrating at others.
Do you feel that this project provided room to innovate and go beyond stakeholder requirements? How so?	Somewhat. We were allowed to go above and beyond but what to do was never really specified and left very up in the air. Despite that we were able to implement some features.
Describe all personal lessons technical/non technical that you learned from this project and feel would be of value to share with the team	I learned a lot about working with a team and working together in with a cohesive mindset. I learned about how to connect a front and back end together and how to structure a full stack application.
What are topics you want addressed during	Overall result and quality of the product. How we could have worked together better. How we could have split up roles and work better.

the retrospective meeting?	
What were the aspects of the project you found most challenging? Why was it challenging?	Trying to work together was very challenging. It was tough since we had to overcome lots of hurdles and tribulations together.
Were there any specific tools or tech that you felt were missing or improper in this project? Could any changes have been made?	We could have implemented a better connection between back and front end. Our use of django was not the best and could have been improved to fit with vue better.
How could the team's collaboration and project management processes be improved?	We could have distributed roles more strictly and evenly. We could have also made sure to hold all the group members accountable.

Name	Elvis Chen
What parts of the project were fun and not so fun?	The most enjoyable part of the project was learning about new technology and trying to "engineer" creative and acceptable ways to solve problems. Unfortunately though, there were more painful parts of working on the project. Our task management was off, and we ended up with loads of work in our final sprint and there was a lot of task that was required along with the actual project itself, which I was not a big fan of.
Do you feel that this project provided room to innovate and go beyond stakeholder requirements? How so?	I feel we definitely went above and beyond on these projects. Mostly, due to the fact that our grading is quite strict and ambiguous, so we had to do a lot more to compensate for it.
Describe all personal lessons technical/non technical that you learned from this project and feel would be of value to share with the team	The biggest technical lesson I learned is with project management. I finally learned how to structure tasks and use a project board to allocate task and prioritize. The biggest lesson, overall, I learned was the fact that you need to allows overestimate your time frame for a specific task.

What are topics you want addressed during the retrospective meeting?	I would love to hear feedbacks on what I personally could have done better. It would also be nice to address what parts of the software can be improved upon if we were given more time and money.
What were the aspects of the project you found most challenging? Why was it challenging?	The most challenging part about the project has to be trying to all the edge cases for what is expected of you. The rubric doesn't give full transparency on what is to be expected, so we had to do a lot of work trying to figure it out. Also working in a team, and trying to separate into independent task was slightly annoying, but doable.
Were there any specific tools or tech that you felt were missing or improper in this project? Could any changes have been made?	I think in terms of tech we used a really good stack with django and vue js. It worked surprisingly good, but as someone who is more proficient with React I would have preferred if we went with something like Next.js.
How could the team's collaboration and project management processes be improved?	I think overall we did good in terms of management and collaboration. I would try to enforce specific task on people, and try to be more strict about deadlines though.

Name	Rahul Rajendran
What parts of the project were fun and not so fun?	The project was fun in the sense that we were able to see results in the database in real time based on interaction with the front-end. It was interesting to see how the front-end and back-end interacted in real time. One part of the project that was not so fun was front-end development. It took time to make everything look nice. The behavior of CSS did not always match what I thought it would. It also took a long time to understand axios, which was not fun.
Do you feel that this project provided room to innovate and go beyond stakeholder requirements? How so?	Yes, I believe this project did provide room to innovate and go beyond stakeholder requirements. This is because of the open-ended nature of the requirements. Aside from the minimum set, we were free to implement whatever we wanted.
Describe all personal lessons technical/non technical that you learned from this project and feel would	- Start early. You never know how long something will take until you start. This combined with other course work will often take more time than you expect. - Take the time to thoroughly learn and understand the frameworks and languages ahead of time. This will save time in the development

be of value to share with the team	process. - If you ever need clarification, don't be afraid to ask. - Be willing to take constructive criticism from peers and the TAs. If they disagree, have confidence in your answer and explain it nicely. - Don't procrastinate. Do a little work each day if possible.
What are topics you want addressed during the retrospective meeting?	During the retrospective meeting, I would like the members to discuss what they would do differently in future software development projects. Specifically, how to reduce the stress of meeting deadlines.
What were the aspects of the project you found most challenging? Why was it challenging?	I believe the most challenging aspect of the project was testing. Additionally, input validation required a decent amount of thought. There were many edge cases to consider. We had to consider a wide variety of changes in the back-end data tables influence the content of other screens took thought and effort. This is what made general functionality
Were there any specific tools or tech that you felt were missing or improper in this project? Could any changes have been made?	Using more familiar technology could have helped reduce the learning curve. Personally, I have never used Vue.js prior to this project. This made finding and using the appropriate testing framework time consuming. Although I ended up enjoying Vue and had a great learning experience, I think that change could've smoothened the development process.
How could the team's collaboration and project management processes be improved?	Other groups opted to meet in person, which I believe may have fostered better team collaboration. Project management was generally good. We rarely had two people work on conflicting tasks at the same time, which led to increased productivity and minimum conflict.

Name	Diego Martinez
What parts of the project were fun and not so fun?	The best part of the project was being able to learn how to use javascript a bit. Though kind of annoying, it was a good learning experience to get my feet wet in the crazy world of java development.
Do you feel that this project provided room to innovate and go beyond stakeholder requirements? How so?	Yes there was a lot of room for making improvements. In a sense there was a lot of freedom, but that could be equally bad as it was equally good.
Describe all personal lessons technical/non	I realized that I am way worse at taking criticism then I thought I was. So through out the rest of my college career and professional life, I

technical that you learned from this project and feel would be of value to share with the team	must be better at taking it lol. Something that I found my self exceeding in was general collaboration. I am now more confutable working in a team environment than I was before this class.
What are topics you want addressed during the retrospective meeting?	Just to show love to the group members that did there part and made good work. Especially those that went above and beyond.
What were the aspects of the project you found most challenging? Why was it challenging?	Learning how to properly use JavaScript was the most challenging part in my opinion.
Were there any specific tools or tech that you felt were missing or improper in this project? Could any changes have been made?	Everything seemed to flow well together. I feel like every decision that we made was pretty spot on for it being out first time working on a project of this magnitude.
How could the team's collaboration and project management processes be improved?	I feel like in the beginning of the semester, there was a lot of areas of growth that we needed but as time went on, we got better at this. Starting to fill those gaps. I feel like we are a significantly better team then where we started.

Name	Rushil Jayant
What parts of the project were fun and not so fun?	I believed the parts of the project that was that were the most fun were the parts involving learning a new programming language/framework and seeing that put into action and working after hours of hard work was both fun and rewarding. What was not so fun was configuring all the files for the deployment, and making the deployment environment work without issues.
Do you feel that this project provided room to innovate and go beyond stakeholder requirements? How so?	I believe that this project did not enable us to provide much room to go above and beyond the stakeholder requirements. Because of the intense nature of the requirements and the time deadline to deliver all the requirements, it made it harder to think of things that could be additional but not detract from the main focus of the product. Towards the end, we did find room we could go above and beyond by potentially making useful additions to the product that fit the context.

Describe all personal lessons technical/non technical that you learned from this project and feel would be of value to share with the team	I learned how to use Django and Vue js for the first time, and learned how to make an end-to-end application with it, which is by far my biggest technical achievement. I also feel that learning the Agile Process is also valuable as it enabled us to function better as a team together.
What are topics you want addressed during the retrospective meeting?	I want to make sure that during the retrospective meetings we cover on the key takeaways from this class and what we want to use, technical/non technical skills and how we all could better refine them. We also can share what we should do better on this project and what we could do if we could start over.
What were the aspects of the project you found most challenging? Why was it challenging?	I found that setting up the backend, as well as the autogeneration library for the documentation was the most difficult. This was because it required an understanding of the backend systems that are responsible for making this all happen seamlessly.
Were there any specific tools or tech that you felt were missing or improper in this project? Could any changes have been made?	I think that we should have used a different technology stack like Next.js because of its robust support for everything this project requires. Also it has both server and client side rendering, which makes it more efficient to do everything. I also felt we could have made more use of github projects and use more of the kanban board to better split tasks among group members.
How could the team's collaboration and project management processes be improved?	I think that the team's collaboration could be improved if we had more synchronous meetings outside of class to collaborate on homework assignments. This could have enabled us to finish tasks faster and more efficiently, than being stuck either last minute or during in class work. PM work could be more improved by doing more task assignments.