Project 2 Group Retrospective

Team 51

(🐕‍🦺)

Anna Brooks

Peter Just

In Harini Kumar

Casey Pei

Alan Perez

Matthew Wang

# Executive Summary

Our project aims to design a Point of Sale(POS) system for the popular on-campus restaurant Rev American Grill. The team has made tremendous progress to create a functional POS system that can serve the needs of not only servers but as well as busy managers.

For the product analysis section, some of the features that our team is particularly proud of is our graphical user interface which intuitively helped everyone navigate through the POS system. Although this assisted us in many ways, we feel as though the product can be improved in the next version by optimizing some key features to make it much faster and more efficient. Some of the technologies that were used to create our working product version include Java, SQL, and JavaFX.

In the work analysis section, our team faced some challenges during the beginning stages of the product. However, we were able to quickly address issues and bring up solutions and strategies for better communication and workflow. Some successful actions our team took include proactive team meetings and dividing work among team members.

Throughout the lifetime of this project, we faced challenges and learned lessons along the way. Examples which include organization and planning. The code for our project could have been more structured to find bugs easier and planning to help each other resolve common issues. Going forward we will be more careful planning our project and ensure common issues are addressed much earlier in the project.

In our retrospective meeting, we dedicated and scheduled to meet on March 22nd at 4pm for approximately 1 hour. The meeting’s agenda focused on discussing various topics about improving the project, as well as the team. As a result of this meeting, we also created some survey questions that each of our team members filled out. Interestingly, we found that there was a need to discuss planning during the next meeting, as all team members felt that this area was especially lacking. Lastly, the results of the survey found that our group finds pair programming to be especially useful for resolving issues and advancing our project forward.

# Product Analysis

Some parts of the project that work well include having a clean user interface with pictures of menu items and having all of the required functionalities working. Some features that could be improved include better error handling (for instance, to prevent inventory decrementing below zero), showing more error messages on the GUI, and fixing some minor bugs with inconsistent spacing. We could also populate the database with more orders.

Next steps for version 2.0 could include cleaning up and refactoring our code, followed by adding functionality to the item customizations page. Including a way to return to the login page could also be useful. Bigger problems that could be addressed include making our GUI function more quickly, as some of the features currently take a long time to query the database.

We gained more experience working with Java, JavaFX, SQL, and git. While the first three would mainly be applicable to future projects using these specific technologies, the knowledge we gained about version control with git could be applied to a wide variety of projects in the future. We especially want to focus on avoiding merge conflicts and accidentally overwriting work in the future by utilizing multiple branches.

# Work Analysis

As we worked together to complete Project 2, we experienced some challenges initially with understanding the requirements for each deliverable. This was due to our failure to take enough time to form a plan based on the correct instruction page. If we could do it all over again, we agree that we would start by reading the correct requirements document thoroughly and then assign each member ownership of the tasks on the requirements list. We addressed this issue by Phase 4.

By the end of this project, our team became successful at meeting altogether to put our individual parts together and complete the project as a team. We used Discord to effectively communicate and meet outside of class; this also allowed us to share screens and ensure the master version of the project was up to standard. We didn’t take the time to do this for phase 2, which was reflected in our scores for that phase. However, once we got our workflow going, communication and synchronization became much easier and more effective.

We split up the workload by project requirements or functions, which was a good strategy for the most part. It did lead to some confusion when it came time to combine our work and test the final product, since naming and fxID’s had some variation between parts. We also had challenges with using GitHub for version control and this set us back quite a bit. Looking back, we should have been more careful with GitHub pushes and created the appropriate branches at the start of the project. Now that we have more familiarity with the program, it should be easier to address this issue during the next phase of the project.

# Going Forward

There are quite a few lessons we have learned from this project, with the largest one being organization. We had problems causing our project to become difficult to work with, and by the end correcting bugs was a huge pain. Before the next project, we are going to clean up the repository and organize the code better. This includes a better separation of frontend from backend, as well as making it easier for each team member to pull, compile, and test the project. Knowing how to keep the project clean is very important to learn, as if we had known the lesson from the start, our time would have been spent much more efficiently.

Another major lesson we have learned is the importance of testing at every step. The project was difficult to run on some members’ computers, so a lot of code was being added with not a lot of testing being done. This left us with buggy, nonfunctional code, and without any idea of whose contribution caused it. Going into the next project, we are going to more clearly separate workspaces so that testing is very easy to do, and bugs that come up can be clearly identified.

One last lesson we will keep in mind going forward is the importance of having a clear plan. We were not sure of the requirements, so we left our plans vague, but that meant that we were missing features when it came time to test our project. In the future, we are going to try and be more proactive about having a plan so that even if we don’t meet the exact requirements, we will at least have something that just needs some changes. If we had done this from the beginning of the project, it would have prevented us from slowly getting further behind with each checkpoint.

Appendix 1: Retrospective Meeting Agenda and Minutes

**Meeting Agenda - Retrospective Meeting**

| *Meeting information* | | |
| --- | --- | --- |
| **Date:** 3/22/2023 | **Time:** 4:10-5:00 pm | **Location:** Zach 596 |

| *Member* | *Attending?* |  | *Discussion Topics* |
| --- | --- | --- | --- |
| Anna Brooks  Peter Just  Harini Kumar  Casey Pei  Alan Perez  Matthew Wang | ✔️  ✔️  ✔️  ✔️  ✔️  ✔️  ✔️ |  | **Pre-Meeting:**   1. Questions for the Retrospective Survey 2. Fill out Survey   **Meeting:**   1. Review Retrospective Survey results; discuss:    1. What did you enjoy or not enjoy about the project?    2. What would you improve about the project if you could do it again from the beginning?    3. What did you learn from working on this project?    4. Is there anything you would like to address with the team in our retrospective meeting?    5. Was the project fun and interesting?    6. Did it provide opportunities to innovate?    7. What went well? What didn't go well?    8. What lessons did you learn that you would share with your team?    9. What one topic do you want to make sure we address in the retrospective meeting? 2. Assign who does what portion of the Group Retrospective Report |

| **AGENDA** |
| --- |
| * We will meet on 3/22/23 during lab at 4:10 - 5:00 pm. * Everyone attended the meeting. * During the meeting, we will discuss –   + Making survey questions   + Going over our answers to the survey questions and discussing what we did well and what we could improve on for the next project   + Assigning parts for the retrospective report * The assignments we decided on for the report are as follows: |

| **Meeting Notes** | |
| --- | --- |
| **Topic** | **Notes** |
| What did you enjoy or not enjoy about the project? | Didn’t enjoy feeling like requirements were not clear |
| What would you improve about the project if you could do it again from the beginning? | Figuring out branching, |
| What did you learn from working on this project? | **Teamwork is probably the most important factor when coming to team success.** |
| Was the project fun and interesting? |  |
| Did it provide opportunities to innovate? |  |
| What went well? What didn't go well? | We were able to |
| What lessons did you learn that you would share with your team? |  |
| What one topic do you want to make sure we address in the retrospective meeting? | **Peter** is planning on refactoring the code base for Project 3 |
| Deciding assignments | Shown below |

| **Retrospective Report Assignments** | |
| --- | --- |
| **Group Member** | **Retrospective Report Portion** |
| Anna Brooks | Make meeting agenda, “Work Analysis” |
| Peter Just | “Going Forward” |
| Harini Kumar | “Product Analysis” |
| Casey Pei | Make meeting agenda template + take notes |
| Alan Perez | “Executive Summary” |
| Matthew Wang | Survey Questions |

Appendix 2: Retrospective Survey

## Survey Questions

What did you enjoy or not enjoy about the project?

What would you improve about the project if you could do it again from the beginning?

What did you learn from working on this project?

Is there anything you would like to address with the team in our retrospective meeting?

Was the project fun and interesting?

Did it provide opportunities to innovate?

What went well? What didn't go well?

What lessons did you learn that you would share with your team?

What one topic do you want to make sure we address in the retrospective meeting?

## Survey Responses

What did you enjoy or not enjoy about the project?

* I did not enjoy ANYTHING
* I think it was enjoyable to work with other people but didn’t enjoy the lack of instructions and direction from the TA and instructor.
* I enjoyed having the opportunity to build a full-scale project from the ground up, much like in the workplace, but I did not enjoy the way the deadlines were paced. Nothing of substance was done in the first half, while the latter half was way too rushed.
* I enjoyed my team but did not enjoy the scheduling. Too much time was given for planning with not enough specific feedback, then the actual implementation was very rushed.
* I did not enjoy the lack of clarity in some parts of the project, but I enjoyed working with my group and seeing our final product come together at the end.
* I enjoyed working with my team to build a project with proper frontend + backend. I did not enjoy the little guidance on some parts of the lab with git.

What would you improve about the project if you could do it again from the beginning?

* We would align our database design and GUI plans better with the project requirements and desired functionalities. Making sure everyone could run our project would have helped too.
* I think the easiest way to improve was to have better planning. I remember that a lot of our hardest bugs came with the lack of planning(aka compilation problems(this lowkey came from a lack of knowledge IMO), GUI and backend call conflicts, and lack of understanding of the problem statement).
* I think we would start with the requirements sheet now that we know which one was for the actual instructions for each phase.
* I would have tried to organize the project better, so that team members weren’t constantly experiencing problems. Also, dividing the front end from the back end better in the code would have helped, especially in testing.
* Redesign some parts of the codebase and use different technologies

What did you learn from working on this project?

* SQL
* I learned how to use Scenebuilder and javaFX
* I learned how to use SQL
* I learned a lot about how not to do a project.
* I learned a lot about SQL.

Was the project fun and interesting?

* It was cool when we had an idea of what we were supposed to do
* It was a cool idea but I did not have fun
* The fast pacing (having several deadlines each week) in the second half made the project stressful
* The project itself was interesting. Not what I’d call fun, though.
* Yea it was fun sometimes.

Did it provide opportunities to innovate?

* No
* It could have but it did not
* Not at all. Most creative direction was stifled by the requirements being hard enough to keep up with.
* We were restricted a lot so no.
* Not really, most of the emphasis was on meeting requirements for the demos

What went well? What didn't go well?

* Planning went poorly, everything else went well
* Our last phase went well and when we met on Discord that went well. Our first few phases didn’t go too well because we didn’t know about the specific requirements.
* I agree that our Discord meetings with pair/group programming and debugging were useful. Meeting deadlines and understanding project requirements did not initially go well.
* Our team kept a good line of communication throughout the whole project, and that really carried us, but our lack of experience ended up hindering our efficiency.
* Pair programming went well. Lack of knowledge for configuring the project did not work well.

What lessons did you learn that you would share with your team?

* Planning needs to better
* Planning and meeting altogether is good
* Clearly delegating tasks was helpful, as was setting an earlier deadline for individual tasks so that we had time to debug and make sure everything was working together
* If there’s a problem, speak up early before it’s too late to try and change it.
* Address issues earlier in the project

What one topic do you want to make sure we address in the retrospective meeting?

* Planning
* Finding the requirements
* Planning fixes/additional features
* Refactoring the code for the next project
* Planning