Sprint 1 Retrospective

Introduction

This first retrospective takes place at the end of the first sprint for the development of a condo management system (CMS). During the sprint, we produced mostly all the necessary documentation and planning for the project. We also developed a log in/sign up page for users as well as the corresponding profile pages. We are still in the early development stage and so we mostly focused on documenting, planning and developing UI prototypes for future sprints.

The expectations when the sprint began were high as we thought we could finish all our tasks and possibly advance more in the project because we knew that it is a heavy project. However, upon starting to work on the project, we realized it was very heavy on documentation. We did spend time on some of the technical aspects of the project by developing some pages as well as developing UI prototypes for the next sprint, however, most of our resources were spent on documentation and planning for the coming sprints.

As for the results, we managed to complete all of our tasks and so the results are satisfactory. As a team, we have also become more familiar with the project and between team members and so we are confident to be more productive in the next sprint.

For the development of the CMS, we have been using GitHub to collaborate as well as Jira to manage all the tasks to do. As mentioned previously, we mainly focused on documentation and so we did not use that many tools, but we do have plans to implement pipelines for automatic testing. We have also decided to use firebase to store our data because it allows us to abstract away the authentication process, making it faster to implement

What went wrong

1 - Time management of the team

The team struggled with time management because we started working on the project later than what would have been ideal. This happened because the channels were not very active and the team members had other priorities at the time. It was the beginning of the semester and everyone was still adjusting to their courses and workloads.

The impact of this was that we had a lot less time to produce all the deliverables required for sprint 1. We addressed this issue by communicating more on slack and setting up meetings to determine what to do as well as just using the text channels. As for the future, we know now to meet early in the sprint to make sure everyone is on the same page and knows what to do.

2 - Late Task distribution

Another problem we had was that we did not know what to do initially. We waited to meet with our TA and did not discuss or distribute the tasks immediately after that. This slowed the production of our deliverables. This happened because we did not initially take the initiative to plan.

The impact of this was that we were slowed down. We addressed this by starting to use a management software where each person assigned themselves to a task. In the future sprints, we know we can do better by planning before the sprint starts. We have already done that for sprint 2 and we have high hopes it will go smoother.

3 - Availability constraints within team

The team also struggled with availabilities. Our team is comprised of 10 members with different schedules and different workloads, we found it initially very hard to find times to meet where everyone was available which caused us to have our first meeting very late.

The impact of these availability constraints was that we could not start working and it created confusion as to what to do. We addressed this by having to make compromises with our schedules in order to meet as well as using our slack channel more. We could have been better at planning for these meetings, but now we know and we will be communicating through slack, smaller team meetings as well as setting team meetings in advance.

What went right

1 - Using Jira (management software)

As a team, we initially struggled with the distribution of tasks. We decided to start using Jira because it is a tool very popular and commonly used in professional settings, and it is very good at displaying what we need to do, what is being done and what is done.

The addition of this tool had a positive impact with the team because it provided them with a clear overview of what was to be done. It was also very easy to assign oneself to a task as well as see who was doing what. Most of the team was able to easily use this tool. As for what could have been done better, we could possibly write the tasks in more details for future sprints.

2 - Team initiative in little time to finish work

Another positive aspect was the team initiative when we realized we did not have much time left in the sprint. Most of the team really put their all and worked hard to get the deliverables ready by the deadline. This happened because we started discussing things more and the project was already being set up through Jira and GitHub to facilitate collaboration.

The impact of this initiative is that it allowed us to communicate more between team members, helping each other out and clarifying any confusions because we all became more involved. We do think that we could have done this before, but we are glad to have found this rhythm and have higher expectations for the next sprint.

3 - Setting up codebase to facilitate the work between team members

We were able to setup the codebase on GitHub, with all the configuration for backend and frontend done. The pages were setup in a way where we can develop in parallel without impeding on one another, which has made development a lot easier and faster. We could work on further separating the code into components to increase cohesion and our ability to work in parallel.

Conclusion

By the time we finished Sprint 1, we learned about all the preparations needed to develop an application. We did some development during this sprint, but we mostly focused on the documentation, diagrams and planning for the whole project. It was a very documentation-heavy sprint, but all team members have a better understanding of what the system has to do as well as all the tasks we will have to do in order to complete development.

We also learned how to work in a big team. Although we struggled in the beginning because of schedule conflict and poor time management, we were eventually able to get a good momentum going and we learned to work efficiently with each other. It is not often that we get to work with so many people in a school setting and it brings a lot of challenges, but it is something similar to what a real life job would be like and it is providing good practice for our futures. We were able to work well together through better communication as well as management and collaboration softwares.

As the main takeaway, we learned that communication within the team is essential. With all the different people in the team, we have different strengths and it has been working very well to ask in the slack channel and receive help from the people who can help in the team. We aim to continue this type of communication as well as prioritizing it in future sprints to ensure a smooth development.

Sprint 2 Retrospective

Introduction

As Sprint 2 is ending, many things are starting to be clearer. The team has a lot of strong qualities like being able to deliver a high-quality work under little time and helping each other in times of need, which are qualities that must be preserved. On the other hand, the team is lacking on small behaviors that would highly increase efficiency if addressed correctly for the remaining sprints.

What went wrong?

All 3 of the following sections are intertwined together, since if the task distribution is done at the very beginning of the sprint's timeline, there would be an overall better time management. Also, if every member would attend the team meeting corresponding to the task distribution, it would be clear from the beginning what is expected from each member.

1- Late task distribution

This section was also a problem seen in the last sprint. The team meeting specifically created for the task distribution was done a bit late in the timeline of this sprint. This could be due to leaving a couple of days of rests before starting to think again about the project. One way of tackling this issue is to have a meeting within 24 hours of the previous sprint deliverance. In that meeting, the team would go over all the necessary tasks, make sure no tasks are missing and divide all the tasks within members. Then, the would be a few general meetings to see how each task sections are going and to enhance the work environment.

2- Time management

There is a slight problem with the time management of the overall team, but this is mainly due to the late task distribution. Even though it went a bit better than the previous sprint, it is because of the reading week that helped taking a break from the rest of the work and focus more on getting the work done as early as possible. Another problem was the confusion of the due date, at the end it turned out to give us more time to get the tasks done. A solution to this problem is to set personal due dates to tasks which would make sure tasks are done a couple of days before the sprint's deliverance date. Also communicating more between members of a small task section to make sure no one is delaying the other.

3- Showing up to team meetings

Not all meetings required all the members to attend, some meetings were held between the documentation team or between the backend team or the frontend team. In these meetings, only the members of the specific section are required to meet and discuss, but when it comes to the general team's meetings, every member of the team is required to attend and participate. This was an issue since we would have a consensus on the date and time of the meeting, but not all members would show up. We would end up postponing the meeting and come up with another date for the meeting.

What went Right

1- Using Jira

Jira is a great way of communicating for teams. Everything takes place in Jira except physical meetings are done on Zoom. The task division feature helps a lot to divide the tasks and know who is doing what without the need of wasting time asking each other all the time.

2- Availability constraints

For this section, there was a lot of improvements compared to the previous sprint. When asking about each one's availabilities to have a meeting, someone would always create a poll on Jira, and we could always find 1 or 2 slots in which every member was available.

3- Good communication between members

In general, there was a good communication between task sections to understand what needs to be done between each other and between members of a same tasks section it seems that members are interacting well with each other, making sure everyone knows what they are doing and helping in times of need.

Conclusion

For the next sprint, the team should work on assigning tasks at the beginning do the timeline of sprint 3 and analyze how the rest of the issues develop since it seems to be the root cause to most issues.

Sprint 3 Retrospective

Introduction

This retrospective seals the third sprint for the development of a condo management system. During this sprint, we advanced in the front-end, back-end and more importantly in the testing. We focused on developing a reservation system which would allow users to reserve a facility like the gym or the pool, on an employee role management system which would organize the two types of employees and assess their roles and responsibilities. Lastly, we developed a request management system which would allow the communication between the users and the employee.

With those, testing was the most important section that needed to be completed since our progress for testing was slow and behind the planned timeline. As for the results of this sprint, we successfully completed many tests all the while advancing as planned for the implementation of the new features. This made us gain back our confidence and assured us that we will be able to finish the project, with all its planned features, in time.

What went wrong

1- Time Management

This is the only problem that persists until now. The team in general struggles with time management, but its not anymore because of the task division. This time it was due to lack of knowledge towards the due date and that there was a lot to do in a relatively short time. In the future, we will assign someone to remind the whole team of the due date every here and there, so that it doesn't come as a surprise a few days before the deadline.

2- Implementation

In this sprint User Stories 4 and 7 were implemented while all user stories until 7 (included) should have been implemented by the end of the third sprint. This is due to time management and underestimating the load of work for the implementation. For the next sprint, we will need to implement user stories 5,6 and 8, so that in the last sprint will be less of a load than the coming because realistically speaking we will have much less time during the fifth sprint due to finals.

What went Right

1- Using Jira

Nothing changed in that section. Jira is a great platform for task divisions.

2- Good communication between members

For the most part, there was a good communication between each other, whenever someone would ask to do a meeting, most members would answer right away. There was always a team member that would answer right away when someone asks a question. At each conversation, we made sure that everyone understood what was said, no one would come out not understanding what was said.

3- Task Division and Meetings

As stated in the previous retrospective, this section was the main cause for everything that went wrong last sprint, but it was fixed right away. The team had a meeting right after the sprint 2 demo and talked about the vision for the future sprints and the tasks that needed to be completed for this sprint. Then, the tasks were divided among members and a summary was written on the team's chat for those who could not attend the meeting.

4- Tests and Code Management

As tests were very lacking in the previous sprints, more members were assigned to write code tests in this sprint. This allowed an increase in tests compared to the previous sprints. Also, being aware of the mistakes we made previously, all reports, whether for the documents or for the codes, were clearly written. This includes testing, code style and quality, and bug reports. This allows for any reader that was not part of the code development to understand what the code is, and what’s good/bad about it.

Conclusion

In this sprint, we saw the benefits of doing a meeting at the beginning of the sprint, sensibilizing everyone about the tasks that need to be completed by the end of this sprint and dividing the tasks in that moment. That gave everyone the full number of days to complete their part without being confused or left without a task. On the other hand, we misjudged the load of work that was ahead of us. For the future, we would consider the amount of work a task takes and increases its planned time by a couple of hours/days.

Sprint 4 Retrospective

Introduction

This retrospective seals the fourth sprint for the development of a condo management system. As the semester is coming to an end, so is our project and the final submission of our semester long product. This marks the end of the fourth sprint and the start of the fifth and final sprint.

By now, we should have implemented all our features and hosted the website according to the initial release plan. As the sprints went by, we were falling short on some work and ended up with having to implement 3 features in this sprint instead of 2. Thankfully, we were able to implement all 3 features which are the development of a reservation system which would allow users to reserve certain products like the pool or the game room, an employee role management system which would allow to organize employees into 2 categories customer service and financial, and a Notification page which would store all recent notifications received. Above these, we progressed in our testing and in the organization of the documents.

What went Wrong

1- Time Management

By now, this has been the only section that is always lacking. Unlike the previous sprint, where it had gone a bit better, we did not set up a team meeting right away after demonstrating the previous sprint, we let close to a week pass before scheduling our first team meeting for the fourth sprint. This, therefore, gave us the opportunity to only meet one other time for this sprint.

In the last sprint we should do as we did in the previous sprint and meet right away, in the following days of the sprint 4 deadline.

2- Tasks Planning and Divisions

This section goes in hand with the previous sprint, but it is more important in the upcoming sprint since some tasks have been removed and new tasks have been added. Task planning and division is usually done in the first team meeting, but even though every team member has a position, we tend to overpower one team section while leaving another more important section with very few workers. This issue should be addressed in the following sprint to complete the heavier tasks ahead of time instead of stressing to submit some work before the deadline.

What went Right

1- Updated with the documents' submissions

Due to a misunderstanding, in the first 2 sprints, we were missing some documents or in m ore precise terms, our documents were not well organized. We started organizing our documents in sprint 3 and had to do some modifications and improve the writing skills for these documents which leads us to the submitted documents for sprint 4 in which all documentations are placed in well-labeled documents/files and no required information is missing.

2- Tests and Code Management

This section has improved since the last sprint, we increased test cases and widened our test coverage, all reports, including the code management document and the bug report, have been updated with the work of this sprint. This allows anyone who wants to be informed with our work to skim through the code and read thoroughly our documents and they should understand the coding with ease.

3- App Hosting

We achieved a very important event in this sprint which was hosting our app/website over the internet. This success would not have been realized without working well on the frontend to have a presentable, pleasing to the eye, product to the users. Also, would not have been possible without a tailored backend which is linked to the frontend and allows the user to perform certain activities. The hosting is an important achievement in our project, and it is even more important that we were able to do so without tardiness to the planned release date.

4- Compromising the lost time

Even though we were late in the meetings and task divisions for this sprint, we were able to work efficiently to complete all the required works for this sprint and the missing works from the previous sprint. This leads us to be on track with the overall timeline of the project which is also another great achievement for the team.

Conclusion

Even though we repeated some of our mistakes in this sprint after dealing with them in the past, we were still able to get the job done and developed the planned features and the complementary tasks. The plan for the next sprint is to meet at the beginning of it decide on the plan for the upcoming sprint and set fake deadlines to guide us not to leave everything to the end. We will mainly focus on updating and enhancing all the existing features and if time permits, we will implement some of the additional and bonus features. Overall, we matured in comparison to the beginning of this project, each learned to develop in their own way to bring more to the table, yet no one is perfect and therefore we still commit the same errors sometimes. We learn and move on; we will focus on those errors in the last sprint and hopefully it will make us more resilient in the future.

Sprint 5 Retrospective

Introduction

At the writing of this retrospective, the team has completed the work for the fifth and final sprint of this project. By now, all demanded features have been implemented and are functioning. Above that, the test coverage is either close or above the 80% mark.

What Went Wrong

1- Time Management

As previously mentioned, this section seams to be the main problem in our group. Between scheduling team meetings, discuss the matters and come to a conclusion, we never end up having a lot of time in our hands to progress at a comfortable rate. This forces us to stress over the matter and cram everything at the last few days before the submission deadline. Add to it the finals season that each student was going through, it really did not help fix this problem, as a matter of fact it amplified this issue.

2- Communication

This section was always one of the group's best characteristics. Most members of the group used to be very active on the group chat and always stayed updated with the latest decisions. Unfortunately, Sprint 5 was in the middle of the finals season which caused our communication to weaken since everyone was more focused on their final exams and would not be consistently checking the group chat. This had a negative impact on our time management and the delay of the completion of tasks.

What Went Right

1- Task Division

The division of tasks was done right away on our first team meeting for this sprint, everyone had a task and we focused on completing on the mandatory features rather than to divide ourselves and trying to complete the additional and bonus features too. This was definitely a good idea since we were able to complete all implementations, all documentations, and all testing as it was demanded.

2- Implementation

For this sprint we had planned to complete the US 8, 9 and 10, but we were also behind on some features, mainly for US 5 and 6. Taking the advice from our teacher and TA, we decided to focus on perfecting the mandatory US so 5, 6 and 8. While working on those, it happens that we were able to implement a small part of the bonus/additional features. So, maybe we did not reach our goal of implementing all the additional features, but we were able to complete all the mandatory ones.

3- Testing

The testing section went very well this sprint, just like last sprint. One difficulty we did face is that the tool we used to calculate the percentage of coverage was making mistakes and it would not count some of the implemented tests, therefore the percentage was lower than the actual coverage. We ended up being able to overcome that obstacle and developed some many tests since we were also a bit behind on the testing, but now it shows all the test coverages and we have excelled the 80% coverage mark as demanded.

4- Documentation

The documentations were all completed since last sprint. Up until sprint 3, we were missing some documents which we were unaware off, but after reading the comments given by the teacher and understanding which documents we were missing and which documents needed to change format, we worked on those on sprint 3 and 4 and we were able to complete them all. For this sprint, it was a matter of adding the sections for the end of sprint 5 and the end of this project. By now, all documents have been updated to include the sections of sprint 5.

Conclusion

In conclusion, this sprint went very well in terms of tasks and completion of features. We were able to complete all development concerning the mandatory features, partially implement one or two of the additional/bonus features, attained the 80% overall test coverage and completed all required documentations. On the other hand, how biggest problem was, surprisingly, communicating with each other, which we proved to be able to overcome any obstacle.

Overall Retrospective