[**Condospective-Sprint 1 2**](#_heading=h.ktbkstqas9p2)

[Introduction 2](#_heading=h.7oer3hpuh8oz)

[What went wrong 2](#_heading=h.6rw3by1s24e5)

[**What went right 4**](#_heading=h.qoosofn56hvp)

[**Conclusion 6**](#_heading=h.k35gjngcy0al)

[**Condospective-Sprint 2 6**](#_heading=h.vfhwng4fl6p9)

[Introduction 6](#_heading=h.hdruqlipay42)

[What went wrong 7](#_heading=h.d1445058n1p8)

[What went right 9](#_heading=h.s9wfyv3ow26e)

[Conclusion 11](#_heading=h.mmbgnqtii4lt)

[**Condospective-Sprint 3 11**](#_heading=h.s2ayyr7jjgvk)

[Introduction 12](#_heading=h.m8f01fdh7e89)

[What went wrong 12](#_heading=h.pststig5gev8)

[What went right 15](#_heading=h.k98cakvpm8eh)

[Conclusion 17](#_heading=h.42kvd41mm9f0)

[**Condospective-Sprint 4 17**](#_heading=h.6e2999wkhnbp)

[Introduction 17](#_heading=h.gjdgxs)

[What went wrong 18](#_heading=h.30j0zll)

[What went right 20](#_heading=h.1t3h5sf)

[Conclusion 22](#_heading=h.lnxbz9)

[**Condospective-Sprint 5 22**](#_heading=h.dbm5jqynf41u)

[Introduction 22](#_heading=h.j0hih4d01ec1)

[What went wrong 23](#_heading=h.5w8z41lmm21l)

[What went right 25](#_heading=h.ty2jnnhh6z1x)

[Conclusion 27](#_heading=h.bvx0lnbetxhi)

[**Lessons Learned 28**](#_heading=h.qq9vq5svbm6y)

[Negative Lessons 28](#_heading=h.t386s0mksxlo)

[Positive Lessons 28](#_heading=h.a1dwlsmvtwyt)

# Condospective-Sprint 1

## Introduction

Firstly, coming into this sprint, our initial expectation was to establish a clear vision of our product vision and the overall mission behind this project. To do so, it was important to identify the stakeholders and their interests. Secondly, it was important to ensure that all team members understood the requirements and constraints this project was to be built on. User stories helped immensely with that since they highlighted the key aspects of our project. Finally, multiple team meetings and significant research have been conducted to establish which tools and architecture our project was going to be based on.

## What went wrong

1. New tools and technologies

One of the most challenging parts of this sprint was the choice of the appropriate technologies, databases, and frameworks to use. Due to the vast and constantly evolving landscape of available tools, it was challenging to navigate through the multitude of options and determine all together which ones best suit the project’s requirements.

The difficulty in adapting to new tools has led to long meeting hours as well as delays in development, impacting our project timelines. To address this issue, the team made sure all members spent additional time becoming comfortable with the new technologies needed for this project.

A method that should have been implemented from the beginning is pair programming. Encouraging pair programming sessions within the team would have facilitated faster adoption of the new tools and promoted mutual learning.

1. User stories

One aspect that did not work well during the development is the development of user stories. This issue occurred, because of a possible lack of prioritization during the sprint planning sessions regarding the most valuable features for the end user. The user stories seem to be divided into small distinct parts instead of providing one clear objective.

This led to complicating the sprint planning sessions, making it difficult for the team to accurately estimate the effort required for each task and allocate resources accordingly. To fix this issue, the team conducted refinement sessions to review and determine how to refine the existing user stories in the upcoming sprints. The granular user stories will be broken down into more manageable and clearer chunks.

A method that could have been used to avoid this issue is the use of story mapping techniques. These techniques could help visualize the flow of features and user interactions, allowing the team to identify and prioritize high-level user activities rather than getting bogged down in overly detailed tasks.

1. Codebase and Testing

Another aspect that presented challenges during this sprint was the effectiveness of the codebase. Indeed, the codebase was not as appropriate and efficient as anticipated, leading to challenges in testing and quality assurance efforts. This aspect occurred due to a lack of code reviews and limited resources for refactoring.

This issue caused increased time and effort required for testing and reduced overall test coverage. To address this, the team conducted a code quality assessment to identify areas for improvement and prioritize refactoring efforts.

Improvement could have been made by allocating more time for code refactoring and optimization from the start of code-writing.

1. Uno Platform

The Uno Platform is an aspect that did not work for the development of this first sprint. This happened due to this platform’s insufficient documentation, making it difficult for the team to understand and utilize its features effectively.

The impact included increased time and effort trying to adapt to the new platform and the likelihood of errors or misuse of Uno’s platform’s features. To address this issue, the team conducted a thorough review of existing documentation to update the outdated information about the platform.

More in-depth research on the platform’s documentation could have been done before choosing the platform.

1. C# experience

A challenge that the team encountered was the lack of experience of some team members with the new technologies introduced for this project. This occurred due to factors such as the limited exposure to C# in previous projects and the limited time assigned to achieving the deliverables.

This aspect has led to slower adoption and learning curves, delays in task completion, as well as increased workload on specific team members. To fix this issue, the team has leveraged online resources and documentation to provide additional support. Furthermore, the team has assigned tasks based on team members’ strengths and areas for growth, allowing for gradual hands-on experience.

Improvements could have been made by conducting a skills assessment at the outset of the project to identify the gaps within the team. Based on that assessment, the choice of language used for this project could have differed.

## What went right

1. Good pace

One aspect that did work well during the development was the pace at which our team worked during the sprint. This good pace in executing tasks can be attributed to effective sprint planning and task breakdown. Initially, our team invested sufficient time to define clear sprint goals, prioritize tasks, and break them down into manageable units by assigning them to all team members.

The good pace in executing tasks resulted in increased productivity, allowing our team to deliver more value within the sprint timeframe. This positively impacts overall project progress and contributes to meeting project milestones and deadlines. To address this aspect, our team prioritized tasks based on their importance and urgency, focusing on high-priority items first to ensure maximum value delivery within the sprint timeframe.

While the good pace in executing tasks is commendable, there is always room for improvement. Identifying and mitigating risks that could potentially impact task execution could’ve helped maintain a more consistent pace throughout the sprint.

1. Continuous Monitoring

An aspect that worked well during the development is the continuous monitoring and adjustment. This aspect was well executed due to clear and open communication among team members. Regular meetings and updates on task progress helped keep everyone aligned and focused on their respective responsibilities. This could include reallocating resources, re-prioritizing tasks, or providing additional support to team members as needed.

By doing so, our team was able to adapt quickly to evolving requirements, priorities, and challenges. It enhanced our ability to stay aligned with project goals and respond to changes in the project environment. To address this aspect, daily stand-up meetings were conducted. These meetings have provided an opportunity for team members to share updates, discuss progress, and raise any issues or concerns.

However, a better and more constant use of the Jira platform by all team members could’ve helped document our progress.

1. Agile Methodology Implementation

The implementation of Agile methodologies worked well during the development. This aspect occurred due to clear understanding and commitment from the team. Agile principles such as iterative development, frequent feedback, and adaptive planning were embraced and integrated into the team’s workflow.

The impact of effective Agile methodology implementation included increased productivity within the team, as well as improved collaboration among all team members. To ensure this implementation, the team established daily stand-ups and sprint planning meetings. Furthermore, regular retrospectives were held to keep up with the team’s processes and identify areas that needed more work.

This implementation could have been better by strengthening communication channels between the different parts of the sprints. For example, if some members finished earlier than anticipated, and other members needed help with their parts, teammates should be able to move from part to part to help each other.

1. Team Communication

Good communication was a key aspect during this sprint. Clear and transparent communication channels were established through the use of Slack and Discord. These channels encouraged team members to communicate openly and proactively share updates, questions, and concerns.

This aspect improved coordination between team members, reduced misunderstandings, and enhanced overall team cohesion. Also, it ensured everyone was aligned with the sprint goals and aware of their responsibilities, which led to higher productivity. Team communication was addressed by setting communication expectations from the start of the project and encouraging active participation in team meetings and discussions.

To further improve communication, the team could have implemented more structured guidelines for providing constructive feedback.

1. Collaborative Problem Solving

An aspect that did work well during the development is Collaborative Problem Solving. Each member of the team brought unique perspectives and problem-solving approaches to the table.

This aspect increased creativity, faster problem resolution, and a stronger sense of ownership and accountability among team members. To address Collaborative Problem Solving, the team organized brainstorming sessions and workshops to generate ideas collaboratively.

To further enhance Collaborative Problem Solving, the team could organize more opportunities to share knowledge and expertise such as in-person meetings.

## Conclusion

In conclusion, the team has encountered both successes and challenges in equal measure. One of the main takeaways from this sprint is the importance of effective communication and collaboration within the team. Despite facing obstacles such as technical complexities and time constraints, the team’s ability to work together, share knowledge, and support one another has been instrumental in overcoming obstacles. Additionally, the team has gained a deeper understanding of the technologies and processes involved, highlighting areas for improvement and growth as this project continues. Finally, this sprint has highlighted the significance of adaptability when encountering hurdles. As the team navigated through unfamiliar territories, all team members demonstrated their ability to learn quickly, pivot when necessary, and remain focused on the project’s requirements.

# Condospective-Sprint 2

## Introduction

Firstly, coming into this sprint, our team knew that multiple elements needed an update. Our product vision statement, requirements and user stories needed minor changes to make them more specific and clearer to what we wanted for our application. Secondly, our team took a collective decision to change the environment we had set up for our application. Indeed, for our front-end, we decided to switch from the Uno Platform to using React.

## What went wrong

1. Transition of Front-End Tools

One of the more challenging parts of this sprint was the change of tools done for our front-end part. Our team encountered multiple issues and limitations with the Uno Platform, initially chosen, such as compatibility issues, performance concerns, or lack of community support. Therefore, our team decided to choose a different route and utilize React instead for the continuation of the project.

However, the shift in tools necessitated the rewriting of code, and the adjustment of development processes, resulting in a temporary slowdown of project progress. To address this challenge, our team recalibrated project milestones, revised timelines and allocated additional resources to facilitate the transition to React.

This transition could have been avoided by choosing the right tool for all team members at the beginning of the project. To do so, our team could have conducted a more thorough evaluation of the potential tools’ considering factors such as platform compatibility, learning curves, long-term maintenance requirements, integration capabilities, etc.

1. React

An aspect that went wrong for some, but not all team members, was the utilization of React for our front-end. Since React introduces new concepts such as JSX syntax, virtual DOM, and component-based architecture, it was challenging for some team members that haven’t had hands-on experience with it.

The complexity of React concepts led to confusion, frustration, and slower adoption among team members, affecting productivity and overall motivation. To address the issue, the team provided comprehensive documentation and tutorials to familiarize team members with React concepts gradually.

Implementing a structured pair programming where experienced React team members are paired with those new to React could have facilitated this issue for the whole team.

1. C# experience

A challenge that the team continued to deal with during this sprint is the lack of experience of some team members with C#, used for this project. This occurred due to factors such as the limited exposure to C# in previous projects and the limited time assigned to achieving the deliverables.

This aspect has led to slower adoption and learning curves throughout sprint #2. To fix this issue, the team continues to leverage online resources and documentation to provide additional support. Furthermore, the team has assigned tasks based on team members’ strengths and areas for growth, allowing for a gradual hands-on experience.

Improvements could have been made by conducting a skills assessment at the outset of the project to identify the gaps within the team. Based on that assessment, the choice of language used for this project could have differed.

1. 80% code coverage

The most challenging aspect of this sprint was the implementation of necessary tests to achieve 80% code coverage. The failure of doing so may have been caused due to our team facing time constraints that led to deprioritizing testing efforts.

Without sufficient test coverage, the project is susceptible to undetected bugs and regressions, potentially leading to degraded product quality and increased maintenance efforts. To address this issue, our team has conducted a retrospective analysis to identify the root causes of the testing gaps. Furthermore, we’ve reevaluated project priorities and allocated dedicated time and resources for test implementation.

An approach that may have been implemented from the start is a test-driven development (TDD). This approach could have ingrained testing as a fundamental aspect of the development process, ensuring comprehensive coverage from the outset.

1. Pressing Deadline

An aspect that our team struggled with during this second print, but not the first, was completing the requirements for this sprint before the deadline initially imposed (before the extension was given). This was caused by high-pressure of trying to finish everything on time while dealing with other academic work and personal life.

Team members felt compelled to rush certain tasks or skip thorough testing and validation processes to meet the deadlines. To address this issue, the team prioritized tasks based on criticality and potential impact.

Setting realistic and achievable deadlines based on thorough project planning and estimation could have prevented excessive pressure on team members and allowed for a more balanced allocation of resources.

## What went right

1. Good pace

One aspect that continued to work well during the development was the pace at which our team worked during the sprint. This good pace was reached by the effective sprint planning and task breakdown our team maintained from the first print. This time around, the prioritizing and the assignment of tasks was done more quickly.

The good pace in executing tasks resulted in increased productivity, allowing our team to deliver more value within the sprint timeframe. This positively impacts overall project progress and contributes to meeting project milestones and deadlines. To address this aspect, our team prioritized tasks based on their importance and urgency, focusing on high-priority items first to ensure maximum value delivery within the sprint timeframe.

While the good pace in executing tasks is commendable, there is always room for improvement. Indeed, when it comes to the implementation of code, identifying and mitigating risks concerning the implementation could’ve avoided the large amount of time that was spent on resolving coding issues on the back-end part.

1. Continuous Monitoring

An aspect that continued to work well during the development is the continuous monitoring and adjustment. Regular meetings and updates on task progress helped keep everyone aligned and focused on their respective responsibilities once again. This included reallocating resources, re-prioritizing tasks, or providing additional support to team members as needed.

By doing so, our team was able to adapt quickly to evolving requirements, priorities, and challenges. It enhanced our ability to stay aligned with project goals and respond to the bigger changes that were made in the project environment. To address this aspect, daily stand-up meetings were conducted. These meetings have provided an opportunity for team members to share updates, discuss progress, and raise any issues or concerns.

However, the integration of monitoring tools with automated incident response systems could have helped us more. By automating the response to common alerts and predefined scenarios, our team could have reduced manual intervention, minimized downtime, and improved overall system reliability.

1. Agile Methodology Implementation

The implementation of Agile methodologies remained something that worked well during the development. This aspect occurred due to clear understanding and commitment from the team. Agile principles such as iterative development, frequent feedback, and adaptive planning were embraced and integrated into the team’s workflow.

The impact of effective Agile methodology implementation included increased productivity within the team, as well as improved collaboration among all team members. To ensure this implementation, the team established daily stand-ups and sprint planning meetings. Furthermore, regular retrospectives were held to keep up with the team’s processes and identify areas that needed more work.

However, this implementation could have been better by putting more effort in strengthening communication channels between the different parts of the sprints. For example, if some members finished earlier than anticipated, and other members needed help with their parts, teammates should be able to move from part to part to help each other. This aspect is something our team still must work on.

1. Jira

The use of the Jira Platform was improved during this sprint and was a result of something working well for our team. The effort in using a user-friendly platform like Jira originated from our team’s need to efficiently organize tasks, assign them to each other, and track our progress throughout the development cycle.

This team effort significantly improved team coordination and helped in meeting project deadlines effectively. To ensure every member of the team put in this effort, regular stand-up meetings were held where team members discussed their task’s status, identified any blockers, and collaborated on solutions.

While the task management aspect was efficient, providing more detailed task descriptions and acceptance criteria upfront could have improved clarity and reduced the need for frequent clarifications.

1. Collaborative Problem Solving

An aspect that did work well and was maintained during the development of sprint #2 is

our team’s collaborative problem solving. This was reached, because each member of the team brought unique perspectives and problem-solving approaches to the table.

This aspect increased creativity, faster problem resolution, and a stronger sense of ownership and accountability among team members. To address collaborative problem solving, the team organized brainstorming sessions and workshops to generate ideas collaboratively.

However, for future sprints, our team could organize more opportunities to share knowledge and expertise such as in-person meetings to further enhance this aspect.

## Conclusion

In conclusion, the team has encountered both achievements and challenges in equal measure during this second sprint. One of the main takeaways from this sprint is the importance to remain consistent in communicating and collaborating with one another. Despite facing obstacles such as technical complexities in the back end and time constraints, the team’s ability to work together, share knowledge, and support one another has been instrumental in overcoming obstacles. Additionally, the team has gained a deeper understanding of the technologies that were retained from sprint #1. However, when it comes to the new technologies that were installed, there is room for improvement and growth for all team members as this project continues. Finally, this sprint has highlighted the significance of adaptability when encountering hurdles. As the team navigated through unfamiliar territories implemented in this sprint, all team members demonstrated their ability to adapt quickly and remain focused on the project’s requirements and deadlines to respect.

# Condospective-Sprint 3

## Introduction

Firstly, coming into this sprint, our team’s focus was to focus on testing to achieve at least 80% of code coverage. To do so, we collectively agreed to put aside the implementation of new features and dedicate more time and effort on back-end and front-end unit testing the increase. Furthermore, based on the sprint #2 feedback, a deployment diagram was added to our system architecture document.

## What went wrong

1. Transition of Front-End Tools

One of the more challenging parts that was found again in this sprint was the change of tools done in sprint#2 for our front-end part. Indeed, our team encountered multiple issues and limitations with the Uno Platform, initially chosen, such as compatibility issues, performance concerns, or lack of community support. Therefore, our team decided to choose a different route and utilize React instead for the continuation of the project.

However, the shift in tools necessitated the rewriting of code, and the adjustment of development processes, resulting in a temporary slowdown of project progress. While many of these concerns were resolved during sprint#2, many members of the team are still accommodating to this transition. To address this challenge, our team recalibrated project milestones, revised timelines and allocated additional resources to facilitate the transition to React.

This transition could have been avoided by choosing the right tool for all team members at the beginning of the project. To do so, our team could have conducted a more thorough evaluation of the potential tools’ considering factors such as platform compatibility, learning curves, long-term maintenance requirements, integration capabilities, etc.

1. React

An aspect that went wrong once again for some, but not all team members, was the utilization of React for our front-end. Since React introduces new concepts such as JSX syntax, virtual DOM, and component-based architecture, it was challenging for some team members that haven’t had hands-on experience with it.

The complexity of React concepts led to confusion, frustration, and slower adoption among team members, affecting productivity and overall motivation. To address the issue, the team provided comprehensive documentation and tutorials to familiarize team members with React concepts gradually.

Implementing structured pair programming where experienced React team members are paired with those new to React could have facilitated this issue for the whole team.

1. C# experience

A challenge that the team continued to deal with during this sprint is the lack of experience of some team members with C#, used for this project. This occurred due to factors such as the limited exposure to C# in previous projects and the limited time assigned to achieving the deliverables.

This aspect has led to slower adoption and learning curves throughout sprint #3, especially with addition of test cases. To fix this issue, the team continues to leverage online resources and documentation to provide additional support. Furthermore, the team has assigned tasks based on team members’ strengths and areas for growth, allowing for a gradual hands-on experience.

Improvements could have been made by conducting a skills assessment at the outset of the project to identify the gaps within the team. Based on that assessment, the choice of language used for this project could have differed.

1. Documentation

An aspect that our team struggled with during this third print, but not the first two, was not completing the documentation at the last minute and before the deadline imposed. This was caused by the high-pressure and extreme focus of successfully achieving 80% code coverage on time.

By focusing on testing, team members neglected the importance of the documentation aspect of our sprint and felt compelled to leave it to the last day. To address this issue, the team agreed to find a balance between achieving the testing or the implementation of new features and the writing of documents. To do so, we’ve decided to include specific days where every member of the team would work on the documentation rather than implementation/testing.

Finally, setting realistic and achievable deadlines based on thorough project planning and estimation could have prevented excessive pressure on team members and allowed for a more balanced between implementation/testing and completion of documentation.

1. Jira

The use of the Jira Platform was not as efficient during this sprint as it was during sprint#2. Despite its capabilities, we fell short in maximizing its benefits for organizing tasks, assigning responsibilities, and monitoring progress effectively throughout the development cycle.

Acknowledging Jira's value to our team, we understand that our usage during this sprint didn't reflect its full potential. To address this issue, we’ve agreed to provide more detailed task descriptions and acceptance criteria in future sprints to improve clarity and streamline our processes effectively.

This issue could have been avoided by setting frequent reminders for every member of the team to participate in Jira and to hold them accountable if they haven’t shown consistent participation.

## What went right

1. 80% Code Coverage

In this new sprint, our team had effortlessly achieved an impressive 80% code coverage, marking a significant milestone for our team. The implementation of necessary tests that were not achieved in prior sprints was seamless, reflecting our team's proactive approach to quality assurance.

By ensuring 80% code coverage, we can now state that are our project is well-protected against undetected bugs and regressions. This ensures that our product maintains its high quality and reduces the need for extensive maintenance efforts in the future. To reach this level of code coverage, our team focused on prioritizing testing and allocating dedicated time and resources effectively to address any potential gaps in our testing strategy.

However, this achievement could have been reached in earlier sprints if our team hadn’t underestimated the importance of testing in our project. Testing should have been a priority over the implementation of new features from the start of the project. Moving forward, we will continue to uphold this standard of excellence in testing, leveraging our newfound success as motivation to maintain and even surpass our current levels of code coverage.

1. Continuous Monitoring

An aspect that continued to work well during the development is the continuous monitoring and adjustment. Regular meetings and updates on task progress helped keep everyone aligned and focused on their respective responsibilities once again. This included reallocating resources, re-prioritizing tasks, or providing additional support to team members as needed.

By doing so, our team was able to adapt quickly to evolving requirements, priorities, and challenges. It enhanced our ability to stay aligned with project goals and respond to the bigger changes that were made in the project environment. To address this aspect, daily stand-up meetings were conducted. These meetings have provided an opportunity for team members to share updates, discuss progress, and raise any issues or concerns.

However, the integration of monitoring tools with automated incident response systems could have helped us more. By automating the response to common alerts and predefined scenarios, our team could have reduced manual intervention, minimized downtime, and improved overall system reliability.

1. Agile Methodology Implementation

The implementation of Agile methodologies remained something that worked well during the development. This aspect occurred due to clear understanding and commitment from the team. Agile principles such as iterative development, frequent feedback, and adaptive planning were embraced and integrated into the team’s workflow.

The impact of effective Agile methodology implementation included increased productivity within the team, as well as improved collaboration among all team members. To ensure this implementation, the team established daily stand-ups and sprint planning meetings. Furthermore, regular retrospectives were held to keep up with the team’s processes and identify areas that needed more work.

However, this implementation could have been better by putting more effort in strengthening communication channels between the different parts of the sprints. For example, if some members finished earlier than anticipated, and other members needed help with their parts, teammates should be able to move from part to part to help each other. This aspect is something our team still must work on.

1. Good Pace

One aspect that continued to work well during the development was the pace at which our team worked during the sprint. This good pace was reached by the effective sprint planning and task breakdown our team maintained from the first print. This time around, the prioritizing and the assignment of tasks was done more quickly.

The good pace in executing tasks resulted in increased productivity, allowing our team to deliver more value within the sprint timeframe. This positively impacts overall project progress and contributes to meeting project milestones and deadlines. To address this aspect, our team prioritized tasks based on their importance and urgency, focusing on high-priority items first to ensure maximum value delivery within the sprint timeframe.

While the good pace in executing tasks is commendable, there is always room for improvement. Indeed, when it comes to the implementation of code, identifying and mitigating risks concerning the implementation could’ve avoided the large amount of time that was spent on resolving coding issues on the back-end part.

1. Collaborative Problem Solving

An aspect that did work well and was maintained during the development of sprint #2 is

our team’s collaborative problem-solving. This was reached, because each member of the team brought unique perspectives and problem-solving approaches to the table.

This aspect increased creativity, faster problem resolution, and a stronger sense of ownership and accountability among team members. To address collaborative problem solving, the team organized brainstorming sessions and workshops to generate ideas collaboratively.

However, for future sprints, our team could organize more opportunities to share knowledge and expertise such as in-person meetings to further enhance this aspect.

## Conclusion

In conclusion, the team has encountered both achievements and challenges in equal measure during this third sprint. One of the main takeaways from this sprint is the importance of testing and consistently reaching at least 80% code coverage. Despite facing obstacles such as learning curves of front-end tools and lack of use of the Jira platform, the team’s ability to work together, share knowledge, and support one another has been instrumental in overcoming obstacles. Additionally, the team has continued to gain a deeper understanding of the technologies that were retained from sprint #1. However, when it comes to the new technologies that were installed in sprint#2, there is always room for improvement and growth for some team members as this project continues. Finally, this sprint has highlighted the significance of prioritizing testing and not underestimating it when having to implement new features. As the team navigated through unfamiliar territories implemented in this sprint, all team members demonstrated their ability to adapt quickly and remain focused on the project’s requirements and deadlines to respect.

# Condospective-Sprint 4

## Introduction

Firstly, coming into this sprint, our team’s focus was to implement all features to have a lighter succeeding and final sprint as it coincides with finals. To do so, we collectively agreed to spend less time on testing and dedicate more time and effort on back-end and front-end implementation. Furthermore, based on the sprint #3 feedback, the missing scenarios in the Use-Case diagram and Activity diagram were added, additional acceptance tests were provided, and the overall content of the code management document was improved.

## What went wrong

1. Pressing deadline

An aspect that our team struggled with during this fourth sprint was completing the requirements for this sprint before the imposed deadline (before and after the extension was given). This was caused by the high pressure of trying to finish everything on time while dealing with final exams season.

Team members felt compelled to rush certain tasks to meet deadlines or leave things to the last-minute due to other more urgent coursework. To address this issue, the team prioritized tasks based on criticality and potential impact.

Setting realistic and achievable deadlines based on thorough project planning and estimation could have prevented excessive pressure on team members and allowed for a more balanced allocation of resources and efforts.

1. React

An aspect that continued to be challenging for some, but not all, team members, was the utilisation of React for our front-end. Since React introduces new concepts such as JSX syntax, virtual DOM, and component-based architecture, it was challenging for some team members who haven’t had hands-on experience with it.

The complexity of React concepts led to confusion, frustration, and slower adoption among team members, affecting productivity and overall motivation. To address the issue, the team provided comprehensive documentation and tutorials to familiarize team members with React concepts gradually.

Implementing structured pair programming where experienced React team members are paired with those new to React could have facilitated this issue for the whole team.

1. C# experience

A challenge that the team continued to deal with during this sprint was the lack of experience of some team members with C#, used for this project. This occurred due to factors such as the limited exposure to C# in previous projects and the limited time assigned to achieving the deliverables.

This aspect has led to slower adoption and learning curves throughout sprint #3, especially with the addition of test cases. To fix this issue, the team continues to leverage online resources and documentation to provide additional support. Furthermore, the team has assigned tasks based on team members’ strengths and areas for growth, allowing for a gradual hands-on experience.

Improvements could have been made by conducting a skills assessment at the outset of the project to identify the gaps within the team. Based on that assessment, the choice of language used for this project could have differed.

1. Documentation

An aspect that our team struggled with during this sprint was not completing the documentation at the last minute and before the imposed deadline. This was caused by the high-pressure and extreme focus of successfully implementing all features, so that our final sprint can only be dedicated to polishing and fixing existing elements. By focusing on the completion of our features, the team neglected the importance of the documentation aspect of our sprint and felt compelled to leave it to the last day.

To address this issue, the team agreed to find a balance between achieving the implementation of new features and the writing of documents. To do so, we’ve decided to include specific days where every member of the team would work on the documentation rather than implementation.

Finally, setting realistic and achievable deadlines based on thorough project planning and estimation could have prevented excessive pressure on team members and allowed for a balance between implementation and completion of documentation.

1. Jira

The use of the Jira Platform was not as efficient during this sprint as it was during our two first sprints. Despite its capabilities, we fell short in maximizing its benefits for organizing tasks, assigning responsibilities, and monitoring progress effectively throughout the development cycle.

Acknowledging Jira's value to our team, we understand that our usage during this sprint didn't reflect its full potential. To address this issue, we’ve agreed to provide more detailed task descriptions and acceptance criteria in future sprints to improve clarity and streamline our processes effectively.

This issue could have been avoided by setting frequent reminders for every member of the team to participate in Jira and holding them accountable if they haven’t shown consistent participation.

## What went right

1. Deployment

In this new sprint, our team successfully achieved the deployment of our website, marking a significant milestone for our team. Our streamlined deployment process ensured that all necessary updates and features were seamlessly integrated into the live site.

By doing so, it minimized downtime and disruptions for our users, enhancing their overall experience. To address potential issues, we conducted thorough testing across various environments, allowing us to identify and resolve any issues prior to deployment. Also, clear communication channels were established to promptly address any unforeseen challenges that arose during the process.

While the deployment was executed smoothly, there is always room for improvement. Enhancing automation tools and further refining our testing protocols could expedite future deployments and enhance overall efficiency.

1. Continuous Monitoring

An aspect that continued to work well during the development was the continuous monitoring and adjustment. Regular meetings and updates on task progress helped keep everyone aligned and focused on their respective responsibilities once again. This included reallocating resources, re-prioritizing tasks, or providing additional support to team members as needed.

By doing so, our team was able to adapt quickly to evolving requirements, priorities, and challenges. It enhanced our ability to stay aligned with project goals and respond to the bigger changes that were made in the project environment. To address this aspect, daily stand-up meetings were conducted. These meetings have provided an opportunity for team members to share updates, discuss progress, and raise any issues or concerns.

However, the integration of monitoring tools with automated incident response systems could have helped us more. By automating the response to common alerts and predefined scenarios, our team could have reduced manual intervention, minimized downtime, and improved overall system reliability.

1. Agile Methodology Implementation

The implementation of Agile methodologies remained something that worked well during the development. This aspect occurred due to clear understanding and commitment from the team. Agile principles such as iterative development, frequent feedback, and adaptive planning were embraced and integrated into the team’s workflow.

The impact of effective Agile methodology implementation included increased productivity within the team, as well as improved collaboration among all team members. To ensure this implementation, the team established daily stand-ups and sprint planning meetings. Furthermore, regular retrospectives were held to keep up with the team’s processes and identify areas that needed more work.

However, this implementation could have been better by putting more effort into strengthening communication channels between the different parts of the sprints. For example, if some members finished earlier than anticipated, and other members needed help with their parts, teammates should be able to move from part to part to help each other. This aspect is something our team still needs to work on.

1. Camaraderie

One aspect that was highlighted throughout this sprint was the remarkable teamwork and support within our team when faced with challenges. This camaraderie stemmed from a shared understanding that our collective success depended on each member’s contributions. If one team member encountered difficulties with their tasks or features, others readily stepped in to help and ensure progress was not impeded. This proactive approach ensured that no individual was left struggling alone and that critical project milestones were consistently met.

To address potential bottlenecks, regular check-ins were made by the team leader and status updates were conducted, allowing us to redistribute workload as needed and provide additional support where necessary.

However, while our team’s willingness to assist each other was commendable, there’s always room for improvement. Establishing clearer escalation procedures and refining task prioritization methods could further enhance our ability to respond swiftly to challenges and optimize project efficiency.

1. Collaborative Problem Solving

An aspect that did work well and was maintained during the development of sprint #2 is

our team’s collaborative problem solving. This was reached, because each member of the team brought unique perspectives and problem-solving approaches to the table.

This aspect increased creativity, faster problem resolution, and a stronger sense of ownership and accountability among team members. To address collaborative problem solving, the team organized brainstorming sessions and workshops to generate ideas collaboratively.

However, for future sprints, our team could organize more opportunities to share knowledge and expertise such as in-person meetings to further enhance this aspect.

## Conclusion

In conclusion, the team has encountered both achievements and challenges in equal measure during this fourth sprint. Despite facing obstacles such as learning curves for front-end and back-end tools and a lack of use of the Jira platform, the team’s ability to work together, share knowledge, and support one another has been instrumental in overcoming obstacles. Additionally, the team has continued to gain a deeper understanding of the technologies that were retained from sprint #1. However, when it comes to the new technologies that were installed in sprint #2, there is always room for improvement and growth for some team members as this project continues. One of the main takeaways taken from this sprint is the importance of teamwork and communication when implementing new complex features. Finally, as the team navigated through the unfamiliar territories implemented in this sprint, all team members demonstrated their ability to adapt quickly and remain focused on the project’s requirements and deadlines.

# Condospective-Sprint 5

## Introduction

Firstly, coming into this sprint, our team’s focus was to complete the features implemented in our previous sprints and fix some bugs that were found in our project. Secondly, our aim was to polish elements in our website such as the user profile UI, the properties profile UI, and the amenities page’s UI. Finally, our team collectively worked on the writing of final deliverables and the preparation for our project’s presentation.

## What went wrong

1. Pressing deadline

An aspect that our team struggled with during this fourth sprint was completing the requirements for this sprint before the imposed deadline (before and after the extension was given). This was caused by the high pressure of trying to finish everything on time while dealing with the final exams season.

Team members felt compelled to rush certain tasks to meet deadlines or leave things to the last-minute due to other more urgent coursework. To address this issue, the team prioritized tasks based on criticality and potential impact.

Setting realistic and achievable deadlines based on thorough project planning and estimation could have prevented excessive pressure on team members and allowed for a more balanced allocation of resources and efforts.

1. React

An aspect that continued to be challenging for some, but not all, team members, was the utilisation of React for our front-end. Since React introduces new concepts such as JSX syntax, virtual DOM, and component-based architecture, it was challenging for some team members who haven’t had hands-on experience with it.

The complexity of React concepts led to confusion, frustration, and slower adoption among team members, affecting productivity and overall motivation. To address the issue, the team provided comprehensive documentation and tutorials to familiarize team members with React concepts gradually.

Implementing structured pair programming where experienced React team members are paired with those new to React could have facilitated this issue for the whole team.

1. Bugs

An aspect that went wrong that our team had to deal with during this sprint was bugs. These bugs were caused due to multiple factors throughout the development phase. For example, some updates to the backend API endpoints were not properly communicated to the frontend development team. Additionally, a lack of comprehensive error handling on the frontend contributed to the entire page crashing when encountering unexpected responses from the backend.

This aspect has affected the overall user experience, causing elements to appear misaligned or improperly sized on certain devices or screen resolutions. This inconsistency diminished the professionalism of our application and could potentially deter users from engaging with our product. Upon identifying the issue, we immediately conducted a thorough investigation to pinpoint the root cause. The frontend and backend teams collaborated closely to synchronize API changes and ensure compatibility between the frontend UI components and backend data responses.We also implemented robust error handling mechanisms on the frontend to gracefully handle unexpected responses and prevent the entire page from crashing in similar scenarios in the future.

Better coordination and communication between the frontend and backend teams could have prevented this issue. Additionally, implementing more comprehensive automated testing, including integration tests between frontend and backend systems, could have helped detect compatibility issues early in the development process.

1. Final Deliverables

An aspect that our team struggled with during this sprint was not completing the final deliverables at the last minute and before the imposed deadline. This was caused by the high-pressure and extreme focus of successfully completing all features, fixing remaining bugs and polishing existing elements. By focusing on the completion of our features, the team neglected the importance of the documentation aspect of our sprint and felt compelled to leave it to the last day.

To address this issue, the team agreed to work collectively in order to complete all documentation instead of dedicating the work to one or two team members. To do so, we evenly separated sections of the documentation work.

Better coordination and communication between the frontend and backend teams could have prevented this issue. Additionally, implementing more comprehensive automated testing, including integration tests between frontend and backend systems, could have helped detect compatibility issues early in the development process.

1. Jira

The use of the Jira Platform was not as efficient during this sprint as it was during our two first sprints. Despite its capabilities, we fell short in maximizing its benefits for organizing tasks, assigning responsibilities, and monitoring progress effectively throughout the development cycle.

Acknowledging Jira's value to our team, we understand that our usage during this sprint didn't reflect its full potential. To address this issue, we’ve agreed to provide more detailed task descriptions and acceptance criteria in future sprints to improve clarity and streamline our processes effectively.

This issue could have been avoided by setting frequent reminders for every member of the team to participate in Jira and holding them accountable if they haven’t shown consistent participation.

## What went right

1. Deployment

In this final sprint, our team continued to successfully deploy our website, marking a significant milestone for our team. Our streamlined deployment process ensured that all necessary updates and features were seamlessly integrated into the live site.

By doing so, it minimized downtime and disruptions for our users, enhancing their overall experience. To address potential issues, we conducted thorough testing across various environments, allowing us to identify and resolve any issues prior to deployment. Also, clear communication channels were established to promptly address any unforeseen challenges that arose during the process.

While the deployment was executed smoothly, there is always room for improvement. Enhancing automation tools and further refining our testing protocols could expedite future deployments and enhance overall efficiency.

1. Continuous Monitoring

An aspect that continued to work well during the development was the continuous monitoring and adjustment. Regular meetings and updates on task progress helped keep everyone aligned and focused on their respective responsibilities once again. This included reallocating resources, re-prioritizing tasks, or providing additional support to team members as needed.

By doing so, our team was able to adapt quickly to evolving requirements, priorities, and challenges. It enhanced our ability to stay aligned with project goals and respond to the bigger changes that were made in the project environment. To address this aspect, daily stand-up meetings were conducted. These meetings have provided an opportunity for team members to share updates, discuss progress, and raise any issues or concerns.

However, the integration of monitoring tools with automated incident response systems could have helped us more. By automating the response to common alerts and predefined scenarios, our team could have reduced manual intervention, minimized downtime, and improved overall system reliability.

1. Agile Methodology Implementation

The implementation of Agile methodologies remained something that worked well during the development. This aspect occurred due to clear understanding and commitment from the team. Agile principles such as iterative development, frequent feedback, and adaptive planning were embraced and integrated into the team’s workflow.

The impact of effective Agile methodology implementation included increased productivity within the team, as well as improved collaboration among all team members. To ensure this implementation, the team established daily stand-ups and sprint planning meetings. Furthermore, regular retrospectives were held to keep up with the team’s processes and identify areas that needed more work.

However, this implementation could have been better by putting more effort into strengthening communication channels between the different parts of the sprints. For example, if some members finished earlier than anticipated, and other members needed help with their parts, teammates should be able to move from part to part to help each other. This aspect is something our team still needs to work on.

1. Camaraderie

One aspect that was highlighted throughout this sprint was the remarkable teamwork and support within our team when faced with challenges. This camaraderie stemmed from a shared understanding that our collective success depended on each member’s contributions. If one team member encountered difficulties with their tasks or features, others readily stepped in to help and ensure progress was not impeded. This proactive approach ensured that no individual was left struggling alone and that critical project milestones were consistently met.

To address potential bottlenecks, regular check-ins were made by the team leader and status updates were conducted, allowing us to redistribute workload as needed and provide additional support where necessary.

However, while our team’s willingness to assist each other was commendable, there’s always room for improvement. Establishing clearer escalation procedures and refining task prioritization methods could further enhance our ability to respond swiftly to challenges and optimize project efficiency.

1. Collaborative Problem Solving

An aspect that did work well and was maintained during the development of sprint #2 is

our team’s collaborative problem solving. This was reached, because each member of the team brought unique perspectives and problem-solving approaches to the table.

This aspect increased creativity, faster problem resolution, and a stronger sense of ownership and accountability among team members. To address collaborative problem solving, the team organized brainstorming sessions and workshops to generate ideas collaboratively.

However, for future sprints, our team could organize more opportunities to share knowledge and expertise such as in-person meetings to further enhance this aspect.

## Conclusion

In conclusion, this sprint highlighted the critical importance of effective time management, proactive planning, and clear communication in navigating project challenges and achieving success. From grappling with pressing deadlines amidst academic commitments to managing technology transitions and bug resolutions, the team learned the significance of balanced workload distribution and thorough training to ensure seamless project execution. The successful deployment of the website and continuous monitoring processes underscored the value of agile methodologies and collaborative problem-solving, emphasizing the benefits of mutual assistance and accountability within the team. Moving forward, these lessons will guide our approach to future projects, driving our commitment to excellence and continual improvement in our work.

# Lessons Learned

## Negative Lessons

1. **Tool Evaluation**:Initially selecting the Uno Platform led to unforeseen compatibility issues and concerns regarding performance. This underscored the importance of conducting thorough evaluations of potential tools before committing to them. Such evaluations should consider factors like compatibility, community support, and long-term maintenance to prevent similar issues in the future.

# 

1. **Technology Adoption:** The introduction of new technologies, such as React, highlighted the necessity of providing adequate training and support to team members. Some struggled with the unfamiliar concepts introduced by React, leading to confusion and slower adoption. Pairing experienced team members with those new to the technology could have eased this transition and improved overall productivity.
2. **Skills Assessment**: The absence of a skills assessment at the project outset also proved to be a stumbling block. Some team members lacked experience with C#, impacting their ability to contribute effectively. Conducting a skills assessment early on could have identified knowledge gaps and allowed for tailored training and task assignments to address these shortcomings.
3. **Test Coverage**: Our failure to achieve the target of 80% code coverage during some of the sprints revealed gaps in our testing efforts. Prioritizing testing from the project's inception and considering the adoption of test-driven development practices could have ensured more comprehensive coverage and reduced the risk of undetected bugs and regressions.
4. **Deadline Pressure**: The pressure to meet deadlines under external pressures was felt in some sprints more than others. Setting realistic deadlines based on thorough planning and prioritizing tasks based on criticality could have alleviated this pressure and allowed for a more balanced allocation of resources and efforts.

## 

## 

## Positive Lessons

1. **Effective Sprint Planning**: Our effective sprint planning and task prioritization facilitated a consistent pace of progress throughout the project. By prioritizing tasks based on importance and urgency, we ensured maximum value delivery within the sprint timeframe.This approach facilitated seamless coordination and kept the team focused on achieving our goals efficiently.
2. **Continuous Monitoring:** Our continuous monitoring and adaptation to evolving requirements and challenges was a crucial aspect of our project. Regular meetings and updates helped us quickly identify and address challenges as they arose. This proactive approach enabled us to stay aligned with project goals and respond effectively to changes in the project environment.
3. **Agile Methodology:** Our embrace of Agile principles, including iterative development and frequent feedback, fostered increased productivity and collaboration within the team. Daily stand-up meetings and sprint planning sessions provided opportunities for open communication and collaboration. By embracing a culture of transparency and accountability, we ensured that everyone was aligned and focused on achieving our objectives efficiently.
4. **Task Management:** Efficient task management using the Jira platform streamlined our workflow and improved team coordination. By organizing tasks and tracking progress in a centralized system, we minimized confusion and duplication of effort. Regular stand-up meetings and task status updates helped keep everyone informed and aligned, ensuring smooth progress throughout the project.
5. **Collaborative Problem Solving:**Our collaborative problem-solving approach was instrumental in overcoming challenges and driving innovation. Leveraging diverse perspectives and expertise within the team, we were able to tackle complex problems and find creative solutions. Regular brainstorming sessions and knowledge-sharing initiatives fostered a culture of continuous learning and improvement, enabling us to overcome obstacles efficiently.

In conclusion, this project provided valuable insights into the importance of thorough tool evaluation, continuous skill development, and effective collaboration. By learning from both successes and challenges, our team will most definitely apply these lessons to future projects, fostering growth and improvement.

# 