

Present our Plans to finish the Project and deliver a fully functional product on time

Assuming that Deliverable 4 represents the final day for everything to be completed, the only remaining task is finishing the website. Since the core features of the system are already implemented, it's now just a matter of connecting them to the web page. Given that most of the work has been done, we're confident that we will finish on time.

Remaining Features that have been added

1. Implemented User roles and permissions - Separate Logins
2. Creating GUI - Not fully completed
3. Sales report
4. Stock notification - included in sales report

Plans for writing/producing documentation/videos/etc

We plan on making a video presentation of a full run-through of our Inventory System. This video will highlight every feature used, making sure every function is demonstrated and works as intended. We'll walk through the core functions of adding, updating, and removing inventory products, and we'll go over the search functionality. The presentation will also showcase how the system interacts with the database, showing real-time updates and changes.

Additionally, we will add a README.md document, which will display the instructions on how to run and clone our application. The documentation will also include step-by-step instructions for key features such as product management, search, and login. We will also have developer documentation, which will provide technical insights, API references, system architecture, and code documentation.

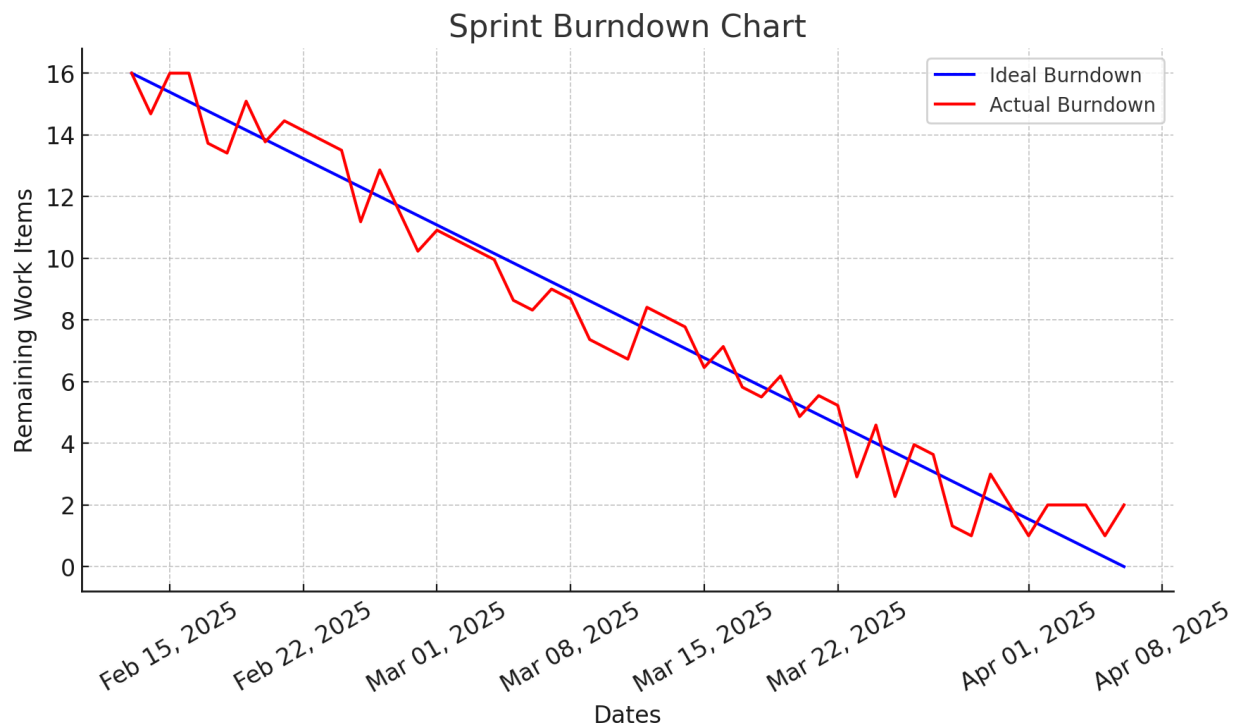
Ensure the customer is content with the progress and direction for the remainder of the project

Talk about the project board

What was completed? What tasks did you fail to complete? Why?

We were able to complete all the core features for the inventory system, including the back-end and front-end logic. However, we didn't fully complete some non-core features such as stock notifications and a fully developed GUI. The non-core features are incomplete due to the time constraints, and our main focus was on getting the functionality working to make sure that the core system was fully functional and tested. The GUI is incomplete as it was introduced very late into the project, and by that point, we still had a lot of iteration 3 incomplete, which meant that we had to prioritize functionality rather than perfecting the user interface.

Burndown-Chart



Final Velocity

With a team of 5 members, we tackled a total of 12 tasks, spread across 3 iterations, each lasting 2 weeks (10 working days).

4 tasks per iteration for the team of 5 members.

3 iterations x 10 working days = 30 total working days

Total team effort = 0.85 velocity x 5 members x 30 working days = 127 task-days of effort

Show Final Demo

Results of testing

Overview of some of your system and unit testing strategy

All team members were involved in the testing process. Each member took responsibility for testing different modules or components based on what they were comfortable with. We conducted both system testing and unit testing to cover all areas of the project. Each feature was tested individually to ensure that individual functions and components were working correctly. We used manual testing for user interface and experience testing, where team members interacted with the system to identify any usability issues.

Name a few (at least two) interesting bugs you were able to find. (There are always bugs!)

A bug that we encountered was that when we were retrieving the access types from the login, it was not properly giving the correct UI menu. This meant that regular clerk employees had access to the full UI, meaning they could remove or add products to the database. Another bug we encountered during development was an unexpected white bar that appeared at the top of our GUI. It seemed to appear randomly, which made it very difficult to trace when we first discovered it. Once we figured out what was causing it, we tweaked and fixed it to make sure everything was aligned correctly.

Agile Retrospectives

Overall Retrospective

This retrospective covers everything from the start to the end of the project. During these three iterations, we have completed 12 core functions with some additional non-key functions. Our overall goal for this project was to build a simple and reliable inventory management system. We will reflect on what went well and what didn't during the whole project, making sure we can learn and improve in upcoming projects.

General Team Observations about the Project

Our group has successfully completed 12 core functions and maintained a velocity of 0.85, staying on track with the goals we have set. Our collaboration and communication were quite strong as we would help each other when it was necessary. However, we all agree that time management and testing were challenging, which is due to the UI development taking longer than expected. Which then caused us to have less time to test and debug. Moving forward, we plan to schedule the more complex tasks earlier and as well as set a proper time for testing framework and set clearer deadlines to ensure smoother iterations.

Individual Reflections

Group Members	What went well?	What didn't go well?	Suggestions for Improvement
Vithunan	Everyone was able to work well with each other. Some helped others when they were falling behind	I think we could have done better with our time management and struggled a bit with our testing phases for our features	Scheduling more challenging tasks early into the iteration and completing them first
Tony	Coding went smoothly, with very minimal errors and the need for	Errors required some changes to be made to smaller parts of	We could create a more flexible testing framework so that

	adjustments.	the code	we would not have to make corrections each time something new is added.
Tisha	Communication with the group was quite strong, and everyone was ready to help when needed.	Some parts of the interface took longer than we had initially planned, which delayed integration with the other features.	Setting more explicit deadlines for UI development and deadlines for testing, so it's not all rushed.
Omar	Good efforts from the whole team tackling every issue that was faced with good attention to detail and documentation.	Linking Front End to Back End proved difficult.	Need to consider more time for bugs and errors.
Abdullah	Our team collaborated effectively, with members supporting each other when someone needed help	We faced some challenges with merging code from different team members, sometimes pushing your code can override the code that someone else is working	Always pull the latest changes from the main branch before pushing your code.

Group Members	Regarding the Process? Regarding Technology?	Lessons Learned?	The most important thing to take away
Vithunan	Using the project board in GitHub felt very effective and kept everyone on track	I felt like time management was very important, felt like we were chasing deadlines instead of staying on par with them	Testing features early makes life a lot easier, waiting till the end brings up the possibility of unexpected problems, which we won't have enough time to fix
Tony	Breaking tasks up into smaller iterations felt like it avoided everyone from feeling overwhelmed and allowed a steady flow of progress	Make sure to plan for bugs, they take a lot longer to fix than expected, and make sure to leave enough time for debugging	Keeping code organized made it way easier to add new features or make any changes
Tisha	UI development needs to have clearer deadlines and must be part of the plan way before. In my opinion, it seemed very rushed since it was introduced in the second-to-last meeting	Know when to take breaks, sometimes we get stuck on certain features not working, instead of taking a step back, most of the time we try to brute force it into working	Setting realistic timelines for tasks makes things run more smoothly and helps avoid the stress of rushing to finish
Omar	Front-end development was very rushed and stressful because it was only brought up in the last meeting, even when we asked many times if a developed GUI was needed	Getting feedback from both group members and customers can help a lot. They helped me spot issues that I missed, and I was able to improve the product because of it	Taking the time to plan things out and document everything in the beginning makes it so much easier to fix problems later on without needing to worry
Abdullah	Having regular team meetings helped clarify any problems or	Never ignore the small issues, tend to fix them quickly,	Talking things through and making sure everyone's on

	misunderstandings the group had, making sure everyone was on the same page	since it is most likely to become a greater issue down the line	the same page helps a ton. It cuts down on confusion and prevents a lot of unnecessary mistakes.
--	--	---	--