Fall 2024 PlanXT Final Report

Document Date: Nov 27, 2024
Document Status: In Progress
Important Links

- Application Page: https://planxt.herokuapp.com/
- Github: https://github.com/CSCE-606-Event360/Fall2024-PlaNXT
- Pivotal Tracker: https://www.pivotaltracker.com/n/projects/2721605
- Slack: plantxcorp.slack.com

Customer Meeting Time: Nov 22, 2024 3:00 PM on Zoom

Submit the link to your team's project Git repository.

Place in the public GitHub directory `./documentation/Summer2024`:

- **PPTX** (not PDF, etc) of your slides. (This is graded under Presentation and Demo).

1. For project presentation and demos, each team will prepare a 5-7 minute video presentation and a 10-15 minute video demo. I recommend only one person present, only one person demo the project, use screen-capture software, and utilize voice over narration (be sure that captions are created and accurate). Do not edit the videos too tightly, with quick jumps between screens. The viewer must be able to read what is on the screen.
2. In your presentation and live demo, you should describe the application and walk through the major use cases. You may also explain some of the design choices you made while working on your project. The presentation should introduce your application, your design, and challenges and lessons you have learned about the software engineering process.
3. **PRACTICE/EDIT YOUR PRESENTATION AND DEMO TO STAY ON TIME.**
4. Combine the presentation and demo into one video (i.e. [15 - 20 minute video] = [[presentation][demo]]).

In the textbox place the location of your video. Vimeo and YouTube are popular options to store and share video.

- **PDF** of your report, containing**:**

1. Two paragraph summary of the project *as implemented*, including the main customer need and how the application meets it, including who the stakeholders are. This will contrast to what you wrote in Iteration 0.
2. Description of *all* user stories (including revised/refactored stories in the case of legacy projects). For each story, explain how many points you gave it, explain the implementation status, including those that did not get implemented. Discuss changes to each story as they went. Show lo-fi UI mockups/storyboards you created and then the corresponding screen shots, as needed to explain to stories.
3. For legacy projects, include a discussion of the process for understanding the existing code, and what refactoring/modification was performed on the code, in addition to the user stories listed above.

4. List who held each team role, e.g. Scrum Master, Product Owner. Describe any changes in roles during the project.
5. For each scrum iteration, summarize what was accomplished and points completed.
6. Include a table that tells how many stories and points each member of the team completed, in total.
    1. optionally (recommended): also break it down by iteration.

6. A table that tells how many stories and points each member of the team completed, in total.

Name	Sprint 1	Sprint 2	Sprint 3	Sprint 4	Total Story points	Percentag e
Govind…	2 stories 10 points	2 stories 7 points	1 story 3 points	2 stories 6 points	7 stories 26 points	25%
Mahim…	2 stories 10 points	1 story 5 points	2 stories 8 points	1 story 3 points	6 stories 26 points	25%
Sai Siv…	3 stories 13 points	0	1 story 3 points	2 stories 10 points	6 stories 26 points	25%
Ankitha…	3 stories 13 points	0	1 story 5 points	2 stories 8 points	6 stories 26 points	25%
Total	10 stories 46 points	3 stories 12 points	5 stories 19 points	7 stories 27 points	25 stories 104 points	100%

7. List of customer meeting dates, and description of what happened at the meetings, e.g. what software/stories did you demo.

7. List of customer meeting dates, and description of what happened at the meetings, e.g. what software/stories did you demo.
Sprint 1 - Customer Meeting Time: Oct 4, 2024 3:00 PM Oct 11, 2024 3:00 PM on Zoom

In the meetings, we demoed the project running locally on our machines. We presented our ideas for adding value to the product in the next sprint. Specifically, we mentioned that we would add a visual indicator to the items on the 2D canvas to represent that they were being set up, were already set up, or were being broken down. Additionally, we mentioned that we would work on adding a dependency feature between items. Tito mentioned that we do not want to chase low hanging fruit, and instead focus on the big picture features. He wants us to prioritize the dependency feature, and in fact he would first like to see the chronology bar in 3D. He is not married to the existing 3D rendering tool, and would like us to explore other options and quickly decide on a new one (or choose the existing solution). We also presented the idea of item groups, and while he liked the idea, he wants to see the product completed with the main features first.

Sprint 2 - Customer Meeting Time: Oct 18, 2024 3:00 PM Oct 25, 2024 3:00 PM on Zoom

In the meetings, we showcased the progress made during the current sprint by demonstrating the recent changes we implemented. The client provided valuable feedback, highlighting areas for improvement and additional features needed. The client expressed a desire for live scrolling functionality in the 3D timeline scrubber developed during this sprint, which would enable real-time updates in the interface. The client also emphasized the importance of prioritizing a dependency feature between the objects. We informed the client about the development changes made for 2D-3D synchronization, acknowledging that further work is needed in this area for the next sprint. Additionally, the client also encouraged us to consider the intensity involved when upgrading the current legacy code or libraries. Lastly, he requested real-world performance testing to assess the system's ability to handle a large load (more than 500 items) and to confirm that effective rendering is maintained under such conditions.

Sprint 3 - Customer Meeting Time: Nov 1, 2024 3:00 PM Nov 8, 2024 3:00 PM on Zoom

During our meetings, we demoed the syncing 3D to 2D feature and the live 3D scrolling feature and had several discussions about the next direction in which to take this application. The client requested several new features and we documented all of the additional functionality in which the client was interested. There were a whole host of features including an online collaboration and access control functionality with the ability to share plans with other users, leave comments on different aspects of the plan, and so on. We also presented a survey of existing floor planning tools at the request of the client, and the client picked his favorite tool and the specific features that he would like to see from these existing commercial applications in our application. However, we set up realistic expectations with the client and picked a few primary user stories to focus on for the next two weeks. Specifically, we want to add all the functionality unique to Tito's vision. The live timeline bar feature is complete, and therefore the next priority is the dependency feature. Minor UI changes have been added as pending user stories. All other changes have been documented, but will not be focused on, at least for the upcoming sprint.

Sprint 4 - Customer Meeting Time: Nov 15, 2024 3:00 PM Nov 22, 2024 3:00 PM on Zoom

During our meetings, we demoed the dependency feature and the modifications done to the items table and had several discussions about the next direction in which to take this application. The client was happy with our progress and all the changes done till now as we had completed the two major features expected for this semester plan - the synchronization of the 2D and 3D view and the dependency feature. He has a few features in mind that the future team could implement and we documented all of the additional functionality in which the client was interested. There were a whole host of features including an online collaboration and access control functionality with the ability to share plans with other users, leave comments on different aspects of the plan, and so on. He also wanted an additional chat-like feature where he could send the orientation document to the team. All these changes were documented and will be shared with the future team.

8. Explain your BDD/TDD process, and any benefits/problems from it.

8. Explain your BDD/TDD process, and any benefits/problems from it

We integrated both Behavior-Driven Development (BDD) and Test-Driven Development (TDD) in our development to create robust, high-quality software that aligns closely with clients needs and technical standards.

BDD Process - We begin with BDD to establish a shared understanding of the desired application behavior between the client and us. We had meetings with the client every week to discuss requirements and define behaviors. Then the requirements are broken down into user stories, focusing on specific functionalities from the user's perspective. We used the *As a - So that - I want* format to write clear, concise scenarios that describe the context, action, and expected outcome of each behavior.

TDD Process - Once we had a clear set of behaviors, we moved into the TDD cycle. We first create a test that defines the desired functionality. We then implemented the code to make the test pass. And then we improved the code structure while ensuring all tests still passed.

Benefits:

- BDD allowed for better understanding between the client, product owner and the developers by using a human readable language that described the behaviour of the features..
- TDD led to cleaner, more reliable code with reduced redundancy. This allowed us to only include only the necessary features and reduce the likelihood of the bugs.
- Both BDD and TDD ensured that development is driven by actual requirements and user expectations.
- By using TDD we were able to identify any bugs and correct any issues at the beginning itself.

Problems:

- Writing tests before code and creating detailed behavior scenarios slowed down the development initially.
- As the project evolved, maintaining a large suite of tests and behavior scenarios was a bit time-consuming.

9. Discuss your configuration management approach. Did you need to do any spikes? How many branches and releases did you have?

9. Discuss your configuration management approach. Did you need to do any spikes? How many branches and releases did you have?

Configuration Management Approach

Our team implemented a robust configuration management strategy using Git for version control. This approach allowed us to effectively manage our codebase, track changes, and collaborate efficiently throughout the development process.

We utilized Git as our version control system, which provided us with the following benefits:

- Distributed development, allowing team members to work independently
- Easy branching and merging capabilities
- Efficient collaboration through pull requests and code reviews

Branching Strategy - We adopted a feature branching workflow, which aligned well with our agile development process:

- Most of the user stories and features were developed in its own branch (we had around 20 branches)
- The main branch was kept stable and always contained the latest production-ready code
- Pull requests were used to merge feature branches back into the main branch, ensuring code quality through peer reviews

To streamline our workflows and enhance collaboration, we frequently released updates and each release was thoroughly tested and validated before being merged into the main branch. This enabled us to gather user feedback early and often, and be able to continuously improve the application.

During our project, we did not need to implement any formal spikes. Our team was able to address uncertainties and technical challenges within the scope of our regular user stories and sprint planning.

10. Discuss any issues you had in the production release process to Heroku.

10. Discuss any issues you had in the production release process to Heroku.

Our team's deployment to Heroku was largely smooth, with only one notable issue related to database changes. After making changes to our database schema locally, we noticed that these changes were not reflected in the Heroku production environment. We then had to manually run database migrations on Heroku after deploying code changes that included database schema updates.

11. Describe the tools/Gems you used, such as GitHub, CodeClimate, SimpleCov, and their benefits and problems.

12. Make sure all code (including Cucumber and RSpec!) is pushed to your public GitHub repo.
13. Make a separate section discussing your repo contents and the process, scripts, etc., you use to deploy your code. Make very sure that everything you need to deploy your code is in the repo. We have had problems with legacy projects missing libraries. We will verify that everything is in the repo.
14. Links to your Project Management tool page, public GitHub repo, and Heroku deployment, as appropriate. Make sure these are up-to-date.

Application Page: https://planxt.herokuapp.com/
Github: https://github.com/CSCE-606-Event360/Fall2024-PlaNXT
Pivotal Tracker: https://www.pivotaltracker.com/n/projects/2721605

15. Links to your presentation video and demo video

Presentation:
https://drive.google.com/file/d/1bb5izMkgzRPkzpzwro24i6VO-dl0Il0b/view?usp=drive_link

Demo:

[Link TBA]

WRITE THE REPORT AS IF IT IS BEING READ BY A TEAM WHO WILL FOLLOW ON TO YOUR PROJECT. INCLUDE WHAT YOU WOULD WANT TO KNOW IF TAKING OVER THE PROJECT. IF YOU NEED MORE SPACE, USE IT.

Each group must send their customer their Final Report and the Customer Satisfaction Survey

.

The customer will email the completed survey to the instructor.

- **SEND IT TO THE CUSTOMER AT THE SAME TIME AS YOU SUBMIT YOUR FINAL REPORT.**
- **SEND YOUR FINAL REPORT TO YOUR CUSTOMER WITH THE SURVEY.**
- **THE CUSTOMER EVALUATION IS WORTH 5% OF YOUR PROJECT GRADE.**

Note that you are responsible for ensuring that all project material necessary to facilitate future work on your project is uploaded to the GitHub repository for your project or is otherwise available to both the instructor and the next team(s).

- Introduction - 2para
- Design - 3 major features
- Cucumber tests - Ankitha
- Flowchart for ppt - Govind

Improvements:

1. Sync between 2d and 3d
2. Chronology bar
3. Dependency feature
4. Tabular modifications
5. UI and UX changes

Difficulties:

- Previous ppt contents - CHECK!
- The 3D Rendering method did not have any documentation and it was difficult to make changes to the underlying library at the beginning.
- All JS for 2d floor plan canvas is in single view file, confusing, and not well documented

Future improvements:

- Access control for different users
- Commenting on different items and aspects
- Chat feature in the app
- Adjustable walls

- Hovering items in the table highlights the item and its dependencies in the 2d and 3d canvas
- Seamless integration between the 2D and 3D view
- Being able to stack items vertically
- Upgrade the 3D library

Sprint 1
In this sprint, we achieved the following:

- All members of the team got the development environment setup on their machines.
- All members of the team spent time understanding the legacy code, and the developers made a minor change just to solidify our understanding of the code base
- We got a better understanding of the priorities of the project after our second meeting with Tito

+ Feature: Add a Date and Time widget for input
+ Feature: Add measurements hint to the input

Sprint 2
In this sprint, we achieved the following:

- Explored various 3D rendering tools and libraries, comparing and evaluating them to finalize a suitable solution.
- Developed and integrated a timeline scrubber in the 3D preview, allowing items to appear and disappear according to their start and end times.
- Synchronized the timeline scrubber with 3D data to accurately reflect the venue's layout over time.
- Implemented functionality to update the 2D top view automatically when changes are made in the 3D view, ensuring consistency between both views.
- Completed unit tests to verify the accuracy of the chronology bar and the synchronization between 3D and 2D views.
- Implemented client suggestions and ensured a clear understanding of project priorities.

Sprint 3
In this sprint, we achieved the following:

- Few UI and backend changes for the dependency feature
- Syncing the deletion, rotation, width and height changes between 3D and 2D view
- Displaying items table in 3d view
- A few UI changes overall for user experience
- Live scrolling feature for the timeline bar
- A few cosmetic changes for the timeline bar
- UI changes to the 2-D plan page

Sprint backlog

- The user story "Create Dependency between items" is moderately complex and involves multiple changes across the system. Due to its scope, the developer anticipates that work on this story will extend beyond the current sprint into the next sprint as well. The team will consider breaking this story down into smaller, more manageable pieces that can be completed within a single sprint if possible.

- The user story “Reflect the changes of 3D in 2D top view” is completed
- The user story “Real world scenario testing” is completed
- The user story “Cosmetics for the 3D timeline” is completed.
- The user story “Modify the layout of 2D plan page” is completed.

Sprint 4

In this sprint, we achieved the following:

- Added the dependency between items and be able to view and edit and delete the items
- View items on 2d plan by layers
- Select the duration of setup and breakdown instead of end times
- Sorting and filtering table in 2d view
- A few UI changes overall for user experience
- Adding a table to display the count of each item

Sprint backlog

- The user story "Create Dependency between items" is completed.
- The user story “UI and backend Changes for Dependencies” is completed
- The user story “Filtering and search feature for tables” is completed.
- The user story “Dependency Visualization” is completed.
- The user story “Automatic Timeline Updates” is completed
- The user story “Customizable wall shapes and sizes in 2D and 3D”
- The user story “Exploring the possibility of customizable wall shapes and sizes in 2D and 3D” is completed.
- **+** The user story “Add an item at the current scroll bar time” is completed (This user story was added mid-sprint at the client’s request. The team agreed to assign the user story 3 points)
- **+** The user story “View items on 2-D plan by layers” is completed (This user story was added mid-sprint at the client’s request. The team agreed to assign the user story 5 points)