Final Report

PLANXT

1. Summary

PLANXT is an automation tool designed to help event planners schedule and plan events set up more efficiently and more easily. The previous team provided basic scheduling functionalities and a canvas for users to drop and drag elements (furniture, staff, etc.) for setup planning. In our project, the client wants us to make the WebApp more realistic by adding dependency checks and 3D preview features. The client also wants us to improve the user experience in several ways, such as editable venue days, highlighting elements when clicking the records, automatically adjusting the timeline of dependent events, etc.

In our project, we have added many features to improve the usability of the PLANXT app, including making the plans sharable and exportable, adding more realistic images to the canvas, highlighting the items in the canvas and table, adding the redo and undo buttons, adding the dependency check for groups, and a preliminary 3D preview functionality, etc. All these new features have made PLANXT more user-friendly, and more capable.

2. Description of all user stories

1. EVENT PLANNING PAGE

The legacy code had a static event planning page with a 4 days limit. Our customer asked for this page to be dynamic as well as the same changes should be applicable to the plan update page. Ruby by default does not support dynamic form. To achieve this, major work done was:

1.1 Update database for nested form
Points: 2, status: ***Complete***

To create a dynamic form, we updated the planning db to support a dynamic form. Created new controller event_steps_controller.

1.2 Incorporate Add steps button and use javascript for nested form functionality
Points: 2, status: ***Complete***
Added add steps button and wrote Java script code to support the dynamic page addition feature.

1.3 Add delete steps feature
Points: 2, status: ***Complete***
Added a delete button to remove unwanted planning days.

1.4 Add required fields for creating new plan page-2
Points: 2, status: ***Complete***
Added required fields to form, so users can not leave the mandatory fields blank and proceed to the canvas page.

1.5: Update Plan model UI for nested form format
Points: 1, status: ***Complete***
Changes made in UI to support the dynamic form UI requirement

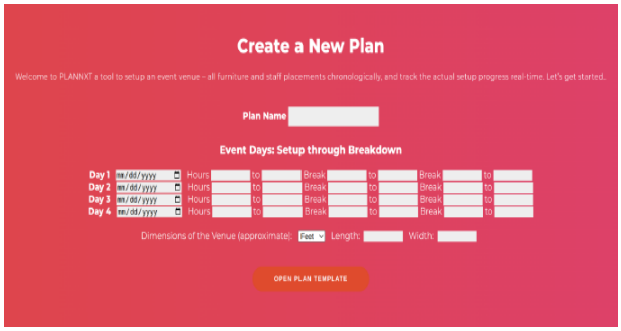
1.6: Create new plan page UI changes
Points: 2, status: ***Complete***
These UI changes were made as per customer requirements to make dynamic form pages more appealing.

1.7: Add navigation from the plan model page to update the nested form
Points: 3, status: ***Complete***

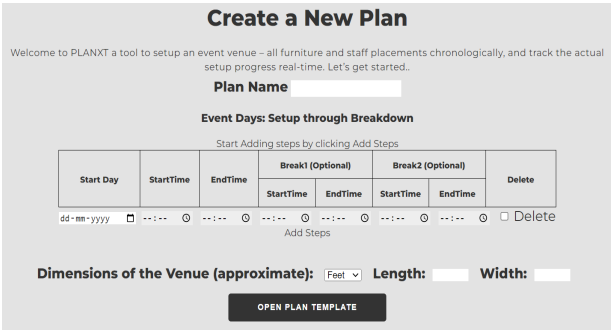
1.8 Improve Color Contrast
Points: 1, status: ***Complete***

All dynamic form changes from the planning page migrated to the update page, with added navigation features to switch between the correct pages.

Before:



After:



2. CANVAS PAGE

This is the core of the PLANXT application. We have made many changes and improvements here during this semester.

2.1 Improve Canvas Layout
Points: 2, status: ***Complete***
Move the item table to the bottom of the canvas page so that every important piece of information is shown at the same time.

2.2 Highlight Item Table Row on Click
Points: 3, Status: ***Complete***
Users can find the item information about the item they are moving on the canvas.

2.3 Add Back Arrow Button on Canvas Page
Points: 3, status: ***Complete***
Users can now return to the plan management page by clicking on it.

2.4 Add Navigation Buttons to List and Canvas Pages
Points: 1, status: ***Complete***
Users can navigate back to the list of plans from the canvas page.

2.5 Highlight Element on Click
Points: 2, Status: ***Complete***
Change the contour of an item if they are on click.

2.6 Add more symbols to the Electrical and Staff layer
Points: 2, Status: ***Complete***

2.7 Draggable Symbols
Points: 3, Status: ***Complete***
Several symbols are represented by an image. We made them draggable from the right side of the canvas page. Also, we make sure the right-click dropdown functions are working on those symbols.

2.8 Show details when clicking elements in the canvas
Points: 3, Status: ***Complete***
Show the item description on click, such as its dimension.

2.9 Clicking a table row will highlight the item in red on the canvas
Points: 3, Status: ***Complete***
A complimentary mechanism with 2.2.

2.10 Undo / Redo Function
Points: 5, Status: ***Complete***
Undo/redo an operation.

2.11 Select multiple items on Canvas at a time
Points: 2, Status: ***Complete***

2.12 Color Code Dependency Check
Points: 3, Status: ***Complete***
Users can easily find where the conflict is happening.

2.13 Add the `showAllItems` checkbox
Points: 2, Status: ***Complete***
Show every item for reviewing the whole floor plan.

3. ITEM GROUPING

For dependency checking and assigning group attributes, we developed a class that handles such functionality. Now, each item can be tagged with a group name.

3.1 Modify the item table for showing the group tag
Points: 1, Status: ***Complete***

3.2 Develop class `group_manager`
Points: 3, Status: ***Complete***
`group_manager` handles `group_id`-`group_name` mapping so that users do not have to deal with numbers. Also, users can easily modify group attributes in the item table.

3.3 Save/extract group info to/from DB
Points: 2, Status: ***Complete***

3.4 Group Dependencies Check
Points: 3, Status: ***Complete***
Check dependencies between groups and give a warning if there exists one.

4. 3D PREVIEW

4.1 3D Preview Research
Points: 8, Status: ***Complete***
Decided to use WebGL with JavaScript to create 3D Preview for PLANXT; managed to realize a basic WebGL-based 3D room with 2D floor planner (GitHub, demo).

4.2 Merge 3D Module in PLANXT
Points: 3, status: ***Complete***
Moved the 3D module sources to the project. Also, created a button and finished the routing for opening the 3D preview in a new tab.

4.3 Add 3D models: Camera, Podium, Staff, etc.
Points: 2, status: ***Complete***
Found (created or downloaded) 3D object models (.obj format); used Blender 2.7 and three js exporter to convert them to .json format; added them to items.js

4.4 Pass 2D Item Metadata to 3D Module and Show
Points: -, status: ***Incomplete***
Fetch 2D item metadata from plan_model and parse them into the data format that 3D needs.

5. PLAN MANAGEMENT

5.1 Fix Destroy Button
Points: 0, status: ***Complete***
The plans were not able to be destroyed properly.

5.2 Export / Import / Duplicate Plans
Points: 3, status: ***Complete***
Now users are able to do these plan operations on the plan management page.

5.3 Plan Sharing
Points: 3, status: ***Complete***
Being able to share and co-edit the plan with another user.

6. TEST:

Developed Cucumber/RSpec test cases for each iteration of developed features. These tests covered newly developed features as well as older features that were not covered previously.

Iter 2. Develop TDD test case for new feature_iter1
Points: 2, Status: ***Complete***

Iter 3. Add coverage report and new TDD test-2
Points: 2 Status: ***Complete***

Iter 4. Add new test for increasing coverage report
Points: 3, Status: ***Complete***

Iter 5. Iteration 5 test addition
Points: 4, Status: ***Complete***

7. AUTHENTICATION

7.1 Request from CRMNXT
Points: 2, status: ***Complete***
Provide methods for fetching user data.

7.2 Integrate Event 360 SSO login
Points: 5, status: ***Complete***

8. MISCELLANEOUS

1. Fix Legacy Deployment Issues and Re-write Document
Points: 0, status: ***Complete***
The deployment document we received from the legacy project was not clear and deployable. We spent some time figuring out the correct way to deploy the project, including server, db, and so on.

2. Upgrade Ruby
Points: 1, status: ***Complete***
Upgrade Ruby from 3.1.0 to 3.2.0.

3. Login Page Simplification
Points: 1, status: ***Complete***
We removed redundant information from the form.

4. Migrate the Project to the Client's Heroku Account
Points: 2, status: ***Complete***

5. Home page rework
Points: 2, status: ***Complete***
Our client wanted a new user interface, and we re-designed it based on his requirements.

6. Rework other page structures
Points: 1, status: **Complete**
Our client wanted a consistent and intuitive design matching the new home theme.

3. Legacy project: understanding the existing code and refactoring (Tian Liu)

- A good way to figure out which scripts control which parts of the App is to do a search for relevant keywords, such as “camera” or “Logout”. Then you can start to make changes based on the scripts you find.
- Another good way to understand the existing code and make modifications is to find similar functionalities that have already been implemented in the code, such as adding a button or adding a checkbox. Then you can follow the same way to implement new features.
- No significant refactoring of the codes has been made to the project.

4. Team Roles

- Amory Doerr 228007537 (Product Owner)
- Ashish Kumar 633002195 (Scrum Master)
- Tian Liu 525004380
- Stefan Wu 733000625
- Huixin Zhang 331007647

5. Summarize what was accomplished and points completed for each scrum iteration

0. 0 points
    a. Familiarize ourselves with the existing code base
    b. Attend initial client meeting
        i. Determine main goals for the semester
            1. Item dependencies

        2. 3D rendering
        3. FashioNXT integration
    ii. Establish fixed weekly meeting time
        1. Wednesdays @ 1:30 PM CT

1. 4 points
    a. Improve canvas/login pages
    b. Rework steps/days editing process
    c. Create initial OpenGL prototype
2. 16 points
    a. Implement import/export/copy plan
    b. Integrate dynamic event step creation/editing
    c. Highlight current selection on canvas page
    d. Move from OpenGL to WebGL
3. 39 points
    a. Grouping of items for dependencies
    b. Addition of symbols for items
    c. Allow highlighting multiple items
    d. Undo/redo functionality when editing
    e. Initial 2D layout in WebGL
4. 13 points
    a. Group dependency checking
    b. Home page redesign
    c. Plan sharing between users
5. 19 points
    a. Reach acceptable testing coverage
    b. Integrate 3D engine into main application
    c. Switch to Events360 OAuth for user management
6. 0 points
    a. Final meeting with client
    b. Finish presentation, demo, and report
    c. Ensure a smooth transition between teams

6. List of customer meeting dates, and description of what happened at the meetings

We meet with our client every Wednesday 1:30pm - 2:30pm.

1. Iteration 1:
    - The demo meeting for iteration 1 is on Feb 22, 2023, 1:30-2:30 pm
    - Customer feedback
        - Show item information on cursor hover
        - Make the item table movable and expandable
        - Add more items for each categories
            - Electric: light, projector, screen, wires, etc.
            - Staff: security, bartender, host
        - Allow planning across days for an event
        - Provide control panel for multiple events management
        - Be able to import / export a plan
            - User can replicate a plan if they are holding similar events again
            - Create a shared floor plan template database for each specific venue
2. Iteration 2:
    - The demo meeting for iteration 2 was on Mar 8th, 2023, 1:30-2:30 pm
    - Customer feedback
        - Add feature to enable selecting multiple items and highlight them at the same time
        - Get all the bugs fixed as soon as possible, so can work on more significant features
        - Find out how to make the conversion from 2D floor plan (json files) to 3D preview based on WebGL, such that the 3D preview feature can be deliverable
        - Refer to the NextFolio for code on showing details on hovering mouse over canvas
3. Iteration 3:
    - The demo meeting for iteration 3 was on Mar 29th, 2023, 1:30-2:30 pm
    - Customer feedback
        - Add timeline goals and feature owners for the project in the spreadsheet

    - Add more diverse symbols and icons to the canvas page to make the rendering as close to a real App
    - Enhance communication among team members for 3D preview feature development
    - Collaboration with CrmNXT
        - Provide API for CrmNXT to get users' name, email, and type
    - Collaboration with EventNXT
        - Matching Authentication System
4. Iteration 4:
    - The demo meeting for iteration 4 was on Apr 13th, 2023, 1:30-2:30 pm
    - Customer feedback
        - Move all the assets under FashioNxt accounts, such as Github and Heroku, immediately.
5. Iteration 5:
    - The demo meeting for iteration 5 was on Apr 26th, 2023, 1:30-2:30 pm
    - Customer feedback
        - Integrate 3D preview with the APP
        - Add appropriate furniture models (table and chair sets)

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

As a legacy project having no Initial test coverage, we started our TDD/BDD testing with two parallel approaches - Creating TDD/BDD test cases for currently proposed features and covering older code by understanding their functionality. For the new feature, we integrated the testing and development process together. We derived Behavioral-driven testing from user stories created from customer requirements and initial design proposals, and it was implemented using Cucumber. We used R-Spec, a Ruby unit test framework for developing Test-Driven development, where we write unit test cases before writing the actual development code itself. As our application is UI intensive we wrote extensive BDD test cases for the majority of features. TDD was mainly developed for features like login, and database testing. We integrated both happy path and sad path coverage in our test coverage. As a large chunk of our code is written in JS, we were not able to cover those functionalities. The main benefit of doing the TDD/BDD process by using RSpec and Cucumber is that as a Team we all were aware of the functionality and overall tasks to be completed.

8. Configuration management approach.

- Everyone is working on the same GitHub repo, and anyone working on a new feature will create a branch for each feature. After the feature has been finished and tested, it will then be merged into the main branch.
- Then the scrum master or Heroku App owner will push the code to the Heroku git repository to update the APP.

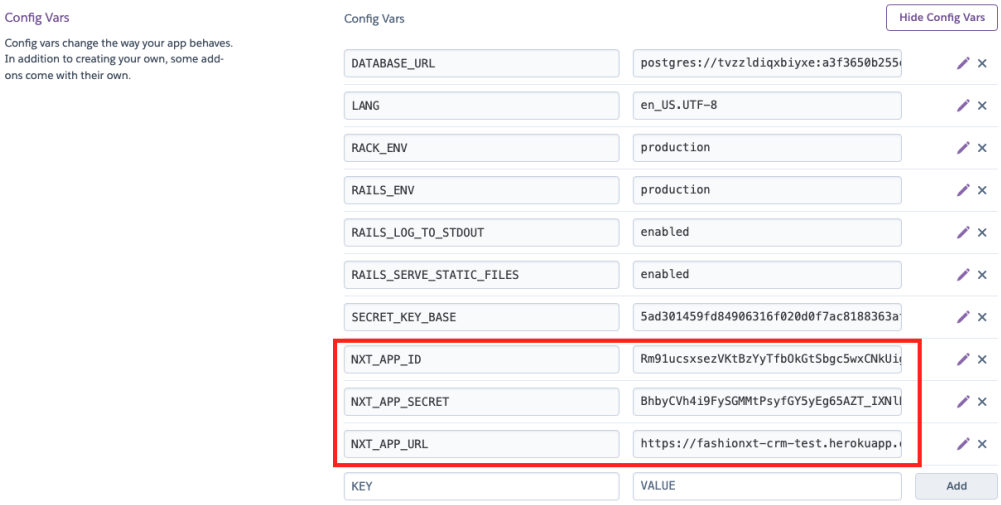
9. Issues you had in the production release to Heroku

- When we took over the project from the previous team, no clear guidelines on deploying the app to the Heroku platform were provided. To make the transition easier for the next team, we created detailed documentation on how to deploy an APP to Heroku, as well as how to link the current git repository to the Heroku git repository. Details can be found in the **README.md** of the git repository.
- If you make any changes to the database, make sure to run "heroku run db:migrate" to update the change to the Heroku database. This has wasted a significant amount of time starting the APP.
- To enable the PLANXT App to use the Event 360 SSO login, make sure you manually added the following environment variables on Heroku account

  NXT_APP_ID: 'Rm91ucsxsezVKtBzYyTfbOkGtSbgc5wxCNkUigmOQlU'

  NXT_APP_SECRET: 'BhbyCVh4i9FySGMMtPsyfGY5yEg65AZT_IXNlLd9Uso'

  NXT_APP_URL: 'https://fashionxt-crm-test.herokuapp.com/'



- If you are having troubles log in with different account, make sure you log out previous account first by clicking “log out” from FashioNXT

10. Describe any issues you had using AWS Cloud9 and GitHub and other tools.

- If you run the App in Cloud9 and start the preview, you will most likely encounter the “mismatch IP error”. Solutions are documented in the README.md of the git repository.
- When running the “RAILS_ENV=production rails s” to start the server in the production environment on the Cloud9 environment, you will have to set the secret key first. Detailed steps are documented in the **README.md** of the git repository.

11. Describe the other tools/GEMs you used, such as CodeClimate, or SimpleCov, and their benefits.

We used SimpleCov for our test coverage report generation. This gem helped us to:

- Generate overall test coverage report in percentage
- Provided each file coverage report
- Visual information of each line that is covered/uncovered

Omniauth was used for integration with Events360

- omniauth
- omniauth-oauth2
- omniauth-rails_csrf_protection

12. Make sure all code is pushed to your public GitHub repo.

- We developed all features using the git repository https://github.com/tian1327/planxt_Spring2023
- In the end, we synced all the codes to the git repository under the Fashionxtllc GitHub account: https://github.com/FashioNXT/planxt

13. Describe repo contents for deploying the project

- The current Git repository should contain all the necessary files to deploy the project
- We have written detailed step-by-step instructions on how to deploy the project in the README.md
- The code structures can be found in the file documentation/Spring2022/PlanNXT Server Guide.pdf

14. Resource Links

- Heroku deployment
- Pivotal Tracker
- Github
- FashioNxt Github
- Slack
- Project tracking sheet

15. Links to presentation video and demo video

- Presentation
- Presentation and App Demo video: https://youtu.be/RV3vojyLnYg