CastNXT Sprint 1 Plan

Team Roles:

1. Product Owner - Shweta Kumaran
2. Scrum Master - Jose Salazar
3. Developers - Alea Nablan, Fatemah Zare, Milad Saadat, Rahul Baid, Rituparna Mandal, Harsh Shah

Customer meeting:

The first customer meeting was held on Tuesday, September 24, 2024, at 7 PM CST. We plan to meet weekly mostly at this time, depending on the client's weekly availability.

Summary:

The main goal for this project is to have the application production ready. We will add more features to the application and ensure that these features are user-friendly to improve the user experience. The stakeholders for this project are going to be people who are involved in the talent recruitment process. These are models that want to audition to be part of a fashion show, designers who want to show off their new items, and producers which take care of the logistics for fashion shows (including hiring and securing other resources needed)

User Stories:

1. deploy the app to Heroku:

   As a developer,

   I want to deploy the application to Heroku,

   so that it is accessible to users via the web and I can ensure the app works as expected in the production environment.

2. Update the library versions

   As a user,

   I want to secure my data by keeping libraries up-to-date

   So that it is not exploited by malicious actors.

3. Connect MongoDb to the App

   As a user,

I want the app to be connected to a database

So that I do not have to face any internal issues when trying to register or log in to the system.

4. Update Feature and Code Documentation

   As a user,

   I want to access up-to-date documentation about the functionalities,

   so that all stakeholders (clients and admins) get the correct flow for features.

5. Upload images to a User

   As a user,

   I want to upload images to my user page from my mobile device,

   so that it is possible to add more images.

6. Send payments

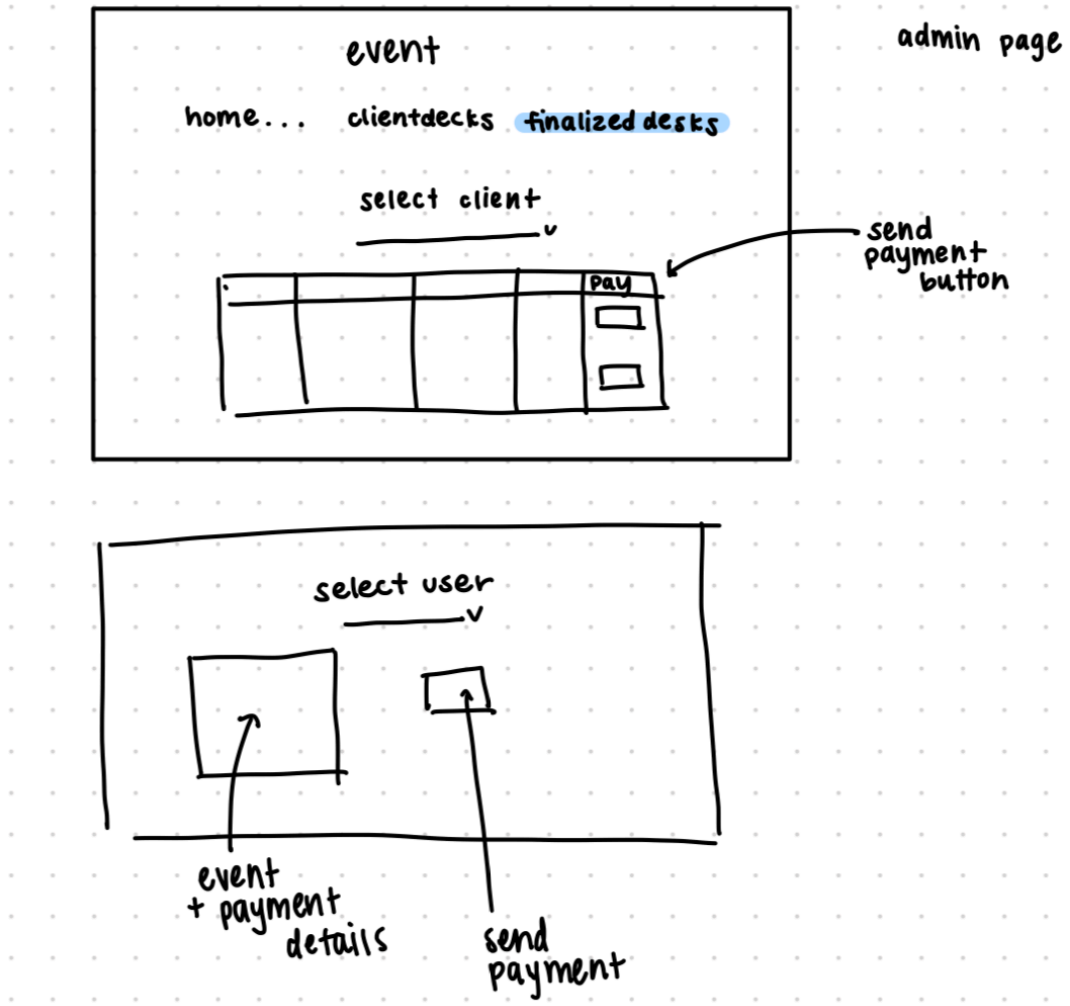
   As a admin,

   I want to send payment through the portal,

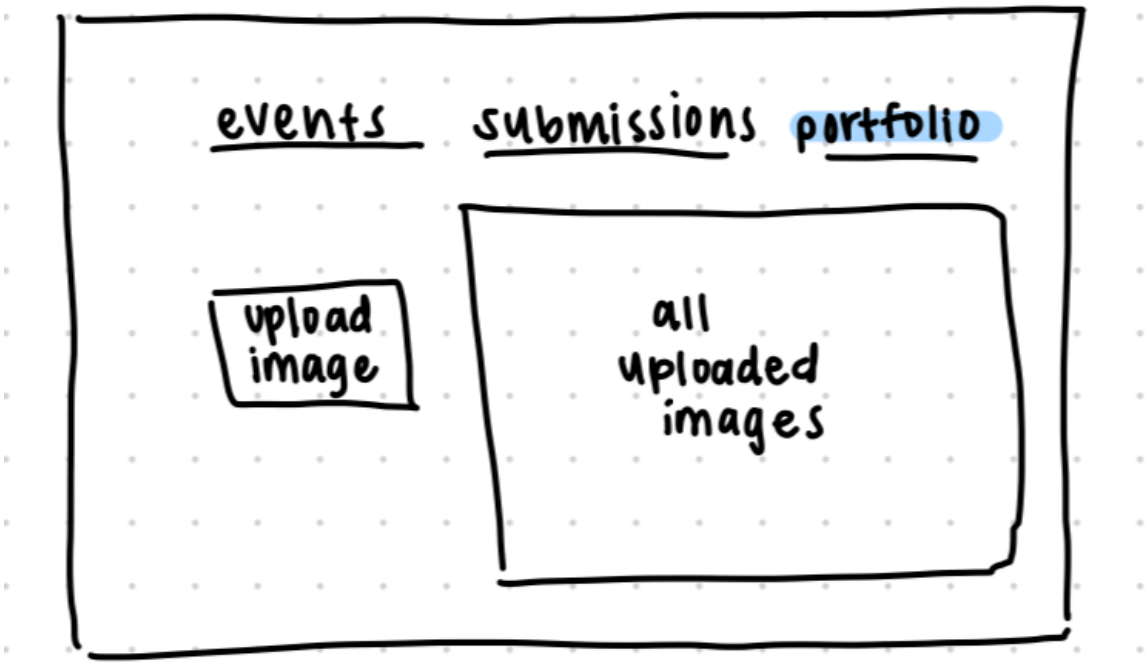
   so that it is easily viewable for the user.

<u>User interface:</u>

User Story 6:


event
admin page
home...
clientdecks
finalized desks
select client
send payment button
pay
select user
event + payment details
send payment

User Story 5:



Sprint backlog:

The sprint goal for this sprint is to ensure successful deployment and builds and to gain a greater understanding of the code base. All of the tasks will be to each developer to ensure a successful developer environment for future sprints.

Tasks:

1. Deploy to Heroku
    a. Points: 3
2. Set up database to heroku
    a. Points: 2
3. Set up project and database locally
    a. Points: 4
4. Update Ruby version
    a. Points: 3
5. Update packages and versions
    a. Points: 3
6. Review of codebase
    a. Points: 4

Total Points: 19

Strategy for learning/improving the legacy project: The assumption we have about the codebase is that the code for this project was developed following test-driven development and behavior driven development. What is most important about understanding this legacy code, is understanding how it's supposed to work and behave. So we will look through the acceptance tests and the unit tests written thus far to see what the intended behavior is for the application. As part of this, since we will inevitably add new features to our app, we will write the associated unit tests and acceptance tests for these features, to make sure we understand how the application should work. We will look at the existing documentation as well to understand the intended flow of the application. In addition to this, we will each get the app running locally, so that we can use the app ourselves, pretending as though we are the user. We will test out and press every single button to see what the user sees. Once we do this, we will get an understanding of the pain points that currently exist, and that will provide us insight into how we can improve the legacy code ourselves.

Once we begin writing code, we will first get a high level overview of what each of the methods and functions do in the codebase. Once we start writing code, if we notice that our code has a dependency on some other part of the codebase being correct, we will look through the implementation details of that part of code. A useful strategy we will use, if we don't understand why some code is written the way it is, is to make minor changes to the codebase in the section we are reading, and test them locally to see their effects. When we're writing code that will eventually be pushed to the repository, we plan on writing in the same style that the existing code is written in. We don't want to get wrapped up in making continuous changes and refactoring the whole code base, or breaking features that already work if it is not necessary. If we do need to change existing code in the codebase, then we will make a user story on it, and we will only change the minimal amount of code necessary to meet the user story. In other words, we will not rewrite code if it is not the goal of the user story being worked on. The code will be rewritten to not rely on knowing the implementation details of other parts of the codebase. We will repeat this cycle as many times as needed, when writing code for the codebase.

While it is not necessarily a replacement story, user stories 1 and 3 is a repeat of a user story done in the past by other teams, since we do need to ensure everyone on our team can access the codebase and application locally. However, user stories 2 and 4 are improvement stories, since we feel that the documentation and codebase is out of date. For our project, we need to ensure data security, and it is likely that updates to the code are needed in order to patch security flaws which may have been found since the code was written.

Links:

GitHub: https://github.com/shweta-kumaran/CastNXTfall24

Slack: https://join.slack.com/t/slack-kll6283/shared_invite/zt-2r9q79t74-y1P3FWeX92YvA0_yAfUXng