# Final Report

Purple Turtles

CSCE 606: Software Engineering

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.**
    a) CASTNXT is an all inclusive event/talent recruitment automation system. CASTNXT will host pages for events where people can apply / audition for roles, and then managers/ producers of the event can view the applications for their event and select applications to be sorted into groups. These groups can be contacted via email and their responses recorded to determine eligibility and fit. Finally an event admin will approve the selections made via the producers and finalized through more email communication with the talent. The main stakeholder for this product is our customer, who manages events and the recruitment of talent for them. This would automate their system and allow for a more streamlined process.
    b) Our solution to this problem is to simplify the problem and bring everything into one place, removing a lot of the tedium of the selection process. Our customer wanted a way for producers to be able to create recruitment applications, allow models to apply for these forms, and then for producers to filter through the submitted applications. We started with integrating Google Forms and Google Auth into our website and used them as the talent applications. Through our website a producer could link their talent application(s) and any model signed in can view and submit their application to that event. After a model has submitted their application, the producer for that event can see their application along with all the other applications to that event in a giant table. From there the producer, and other people involved with the event (such as the director) can pick which talents they want for each role in the event.
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.**
    a) Set up ruby / rails environment
        i) This was a 1 point chore because it was necessary for someone to do so that everyone could begin working on the project. It had to be done after we determined what api’s and gems we wanted to use because they had different requirements than the versions of rails and ruby we had used in class and what was defaulted to on the AWS machine. Therefore this story

consisted of setting up ruby and rails with all the versions / specs necessary for the rest of the project.

b) Create cucumber tests for user stories
    i) This was a 1 point chore during the first iteration where the routes for many of the view pages that were created were tested to ensure that they moved you correctly from one page to another.
c) Route user views together in correct orders
    i) This was a 1 point chore where the UI/ UX team created a group of view templates and linked them together in preparation for the backend functionality to be implemented later so that it could be used.
d) Create rspec test cases for user stories
    i) This was a 1 point chore during the first iteration where the routes for many of the view pages that were created were tested to ensure that they moved you correctly from one page to another.
e) Google Sign In account -> oath
    i) This was a 2 point feature because it involved the implementation of the google oauth and the restful creation and storage of users in a sqlite3 database.
f) Add UI templates to user pages
    i) This was a 1 point feature, which consisted of the integration of the UI/UX and the backend features that were being implemented at the time.
g) As a producer I should be able to access created forms to see their responses
    i) This was a 1 point feature where Google forms could be accessed, embedded, and then the results reviewed by site admins who were managing events.
h) As a producer I should be able to see a list of created forms
    i) This was a 1 point feature where Google forms could be stored on the site and viewed in a list. It was implemented by storing a couple important google form / sheets links in the database to be accessed.
i) As a model I should be able to access created forms to see their questions
    i) This was a 1 point feature where Google forms could be accessed, embedded, and then responded to either in another tab or embedded in the site for convenience.
j) Create basic UI templates for different user views
    i) This was a 2 point feature for adding more ui/ux integration with the backend and combining much of the project together. It consisted of taking the backend functionality and integrating it into the frontend ui that had been premade.
k) Created UI templates for login and new user registration pages

        i) This was a 1 point feature for adding a couple more focused web pages for specific uses like the intro splash page, and CSS/ JS to make the overall website look better and have a better UI.

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 stores listed above.**
    a) Our project was not a legacy project, and did not have any old code.
4) **List who held each team role, e.g. Scrum Master, Product Owner. Describe any changes in roles during the project.**
    a) Scrum Master : Jared Jones
    b) Product Owner : Adharvan Sai Jakkula
5) **For each scrum iteration, summarize what was accomplished and points completed.**
    a) Iteration 0
        i) The first iteration was the design and mock up phase. During this iteration we created lo-fi mockups of the ui's and dataflow, and interaction diagrams for the project to make sure that we understood what we were going to be working on for the remainder of the project before starting development.
    b) Iteration 1
        i) This iteration, after having some issues with communication, we realized that we did not fully understand the scope of the problem we were being asked to solve and had to do a total design rework to incorporate this into the project. This involved redoing the dataflow and interaction diagrams, and some minor changes to the lo fi ui mockup. This iteration we also set up the rails environment and github with versions of gems that would allow us to interact with the gems and api's we wanted to use for this project.
    c) Iteration 2
        i) During this iteration, we divided ourselves into three teams and each worked on different parts of the project in order to boost productivity since we felt behind having had to redo our design another iteration instead of beginning work earlier. The three teams were the google Oauth team, the forms team, and the UI/UX team. Each team worked on their respective functionality and then demoed their work to the client at the end of the iteration.
    d) Iteration 3
        i) The final iteration was more group work, but we worked together at the end to integrate all of the parts each group made together into the final version of our product. This involved putting much of the backend functionality created by the forms and oauth teams together and then

integrating it into the templates that had been created to hold the functionality and then testing it.

6) **List of customer meeting dates, and description of what happened at the meetings, e.g. what software/stories did you demo.**
    a) October 20th (Meeting 1)
        i) Our customer described to us the project in detail. We took notes on features, user groups, and functionality expectations.
    b) October 27th (Meeting 2):
        i) Due to a timezone mixup, our client did not attend the meeting. We met as a team for the full hour discussing. Two hours later, the customer joins the meeting and three of our team members join to have the weekly meeting with him.
        ii) We demoed our planned architecture for the product, and received new requirements and specifications for the product.
    c) October 27th (Meeting 3):
        i) After reworking the project architecture to better fit the requirements we demoed the new approach to ensure we understood the scope and generality of the project.
    d) November 3th (Meeting 4):
        i) This was our end of iteration demo where we completed rails general environment setup and outlined our ideas for progressing on to further iterations with plans for beginning work on forms and google authentication.
    e) November 10th (Meeting 5):
        i) This meeting we demoed progress on the work on google forms and the ability to add, remove, and store them in the database.
    f) November 17th (Meeting 6):
        i) This iteration was the demo of the work done by each group showing the google forms, user, and ui controllers.
    g) December 1st (Meeting 7):
        i) Demoed a few tweaks discussed in the previous meeting to each section of the project. Also show our integration progression for the overall project.
    h) December 8th (Meeting 8):
        i) Final full demo with the customer. Also talked about everything we need to get ready for the next team.

7) **Explain your BDD/TDD process, and any benefits/problems from it.**
    a) Using BDD/TDD and Agile development was great for dividing up work, but lead to issues when each person was focusing on a feature rather than the project as a whole. This leads to some issues when trying to combine all of the parts finding that they don't necessarily all fit together to fully complete a feature.

8) **Discuss your configuration management approach. Did you need to do any spikes? How many branches and releases did you have?**
    a) We had one spike this project taking a rather significant amount of time, so that we could research the functionality of the google forms and sheets api's in order to determine if we would be able to use them to implement this project or if we would have to use something else.
    b) We had 3 major releases (each iteration) that are listed tagged in our github repository that were the summarization of each iteration.
        i) Release 1: Built overall framework for the product
        ii) Release 2: Developing the UI, Google Auth, and Form submission components of the product.
        iii) Release 3: Integrating all of our components into one branch.
    c) We had multiple branches throughout the process that were created by the various teams that were used for individual development before merging them into the main branch for release. There were also a couple additional branches that were used for spike research into other types of form building and generation, before we decided on using google forms.

9) **Discuss any issues you had in the production release process to Heroku.**
    a) The current state of the project is not ready for a commercial / production release therefore we have held off releasing it on heroku.

10) **Describe any issues you had using AWS Cloud9 and GitHub and other tools.**
    a) Issue with space needed for some of the GEM files on the AWS instance. We are unsure whether this was an issue specifically with the GEM files we were trying to use or the AWS instance we were running, but when installing specific GEMs vital to the application, our instances would run out of storage space (or at least that was the error message that would pop up).
    b) We also had some issues with AWS instances disappearing with no explanation, thankfully our use of github made this more of an inconvenience rather than a massive issue that caused loss of progress and code.

11) **Describe the other tools/GEMs you used, such as CodeClimate, or SimpleCov, and their benefits.**
    a) Google Auth GEM
    b) Ruby 2.6.6
    c) Rails 5.2.2
        i) This version of ruby / rails was necessary because it was required to use google forms and sheets api gems.
        ii) They were also required for google oauth functionality to be integrated.

12) **Make sure all code (including Cucumber and RSpec!) is pushed to your public GitHub repo.**
    a) All included in the Github.

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.**
   a) Environment Setup:
      i) Create AWS Cloud9 environment
      ii) Run ssh-keygen -t ed25519 -C "something@email.com"
      iii) Add the ssh key to your github
      iv) Run git clone git@github.com:jared-jones280/CASTNXT.git
      v) Run git config --global user.name "My Name"
      vi) Follow CASTNXT/web/CASTNXT/README.rdoc for the rest of the development and deployment instructions
   b) Documentation:
      i) The documentation for this project is located in the CASTNXT/documentation/Fall2021/ folder.
      ii) This is where all of our iteration documents are.
   c) Code
      i) The code for this project is located in the CASTNXT/web/CASTNXT/ folder.
      ii) The structure of the code is very similar to the Ruby HWs. We have our models, routes, and controllers doing most of the heavy lifting.
         (1) Models are in the CASTNXT/web/CASTNXT/app/models/ folder
         (2) Controllers are in the CASTNXT/web/CASTNXT/app/controllers/ folder
         (3) Routes are in the CASTNXT/web/CASTNXT/config/ folder

14) **Links to your Pivotal Tracker, public GitHub repo, and Heroku deployment, as appropriate. Make sure these are up-to-date.**
   a) Pivotal Tracker:
      i) https://www.pivotaltracker.com/n/projects/2535733
   b) Github:
      i) https://github.com/jared-jones280/CASTNXT
      ii) Our customer never gave us a github repo to develop the project under. Future teams should fork this repo and can continue development from there.
   c) Heroku:
      i) n\a

15) **Links to your poster video and demo video.**
   a) https://youtu.be/F89AXkWQz-0