Change Log

|  |  |  |
| --- | --- | --- |
| Change | Date of Change | Author |
| Baseline | 11/6/22 | Pauline Wade |
|  |  |  |
|  |  |  |

Project Turnover Report

Aggie Pregnant and Parenting Student Organization

Parent Portal

Dillon Lee   
Frank Martinez   
Mualla Argin   
Macy Drew   
Daniel David   
Nathan Patterson

CSCE 431 – Software Engineering   
Fall 2022 

Texas A&M University

Department of Computer Science and Engineering

Contents

[1 Project Turnover 4](#_Toc118710053)

[1.1 Customer Turnover 4](#_Toc118710054)

[1.1.1 User Acceptance Test Results 4](#_Toc118710055)

[1.1.2 Customer Feedback Results 4](#_Toc118710056)

[1.1.3 Turnover of Artifacts to the Customer 4](#_Toc118710057)

[1.2 Instructor Turnover 4](#_Toc118710058)

[1.2.1 Turnover of Artifacts to the Instructor 4](#_Toc118710059)

[2 Configuration Management 5](#_Toc118710060)

[2.1 Configuration Items 5](#_Toc118710061)

[2.2 Version Control 7](#_Toc118710062)

[2.3 Change Control process 7](#_Toc118710063)

[3 Project Presentation 7](#_Toc118710064)

[3.1.1 Lessons Learned 8](#_Toc118710065)

[3.1.2 Acknowledgement of ‘Acts of Service’ 9](#_Toc118710066)

[4. Recommendations for Future Work 9](#_Toc118710067)

[5. Acknowledgements 9](#_Toc118710068)

[6. References 9](#_Toc118710069)

[7. Annex: UAT form and/or Customer Feedback Filled out by the Customer 10](#_Toc118710070)

# Project Turnover

After the successful completion of User Acceptance Test & Maintenance, we finally closed the project! We now need to start Project Turnover which is described in the sections below.

For the User Acceptance Testing phase, we made sure to guide the customer throughout the process, providing them with documentation needed to understand the form and fill it out. We just wanted to make their process of filling out the form as straightforward as possible. We managed to pass all the UAT tests. This made us feel very fulfilled as we were able to prove that our application worked for our customer. We did not need little maintenance to the application, at most we just had to change some labels and remove seed data (fake data for testing) to be ready for customer use. We already had an initial meeting with the customer representation and have started the process of having them sign up for GitHub Student pack and redeeming the Heroku credits needed to transfer the application without a need payment. Due to our initial customer rep’s situation and having another customer representative take that person’s place, the process of fully transferring the application for free is taking a lot longer than expected, but we expect it should be done promptly. The customer representatives are all extremely busy and we are doing our best to accommodate their needs and be patient. We made to create all the documentation needed for the organization to get started using the application and will be following our extended support plan as outlined in our Deployment & Support Plan. We also created a 2-minute demo and reflection presentation which discussed our overall thoughts about the project experience. We celebrated the successful turnover by sleeping soundly.

## Customer Turnover

Throughout customer turnover, we made sure to meet with the customer an additional time outside of our sprint review meetings to answer any questions and onboard them with GitHub Student Pack and the free Heroku credits associated with that. The customer was very receptive and kind. This made us happy because of the positivity we had achieved. We were both patient with each other and were understanding of our situations. We could not have asked for a better customer (student organization), which helped us professionally by exemplifying the dynamic we should strive for in our careers going forward.

### User Acceptance Test Results

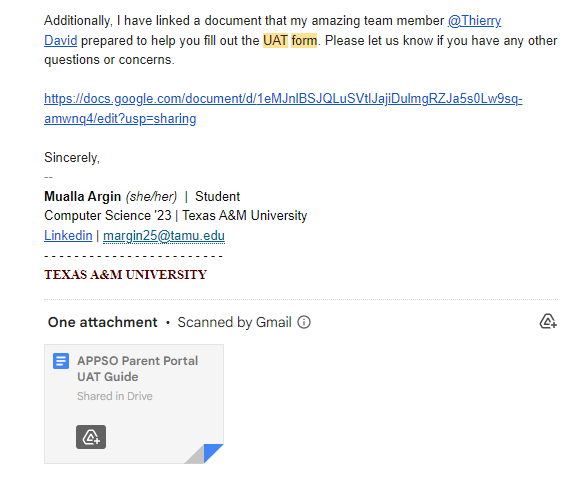
The customer returned the filled out UAT form to the instructor team, indicating which user stories and acceptance criteria met their expectations. See the Annex for the UAT form filled in by the customer.

<https://docs.google.com/spreadsheets/d/1xxfJJTCd1jzxOCiJx7wUbHksPSLgkJf1/edit?usp=sharing&ouid=100630313886753670985&rtpof=true&sd=true>

During the user acceptance, we had the customer fill out a document where they determined whether our product covers all user story requirements and acceptance criteria. Our UAT form was designed using the SunnyDay Form Example given by Professor Wade. Once we completed developing the form, we had professor Wade review it for any errors/issues. After a few iterations of the document, we had our final product which we sent to our customers with a UAT guide that gave them instructions on how to fill it out. We also gave the customer the option to do a Teams call if they had any confusion when they were filling out the form.

APPSO Parent Portal UAT GUIDE: <https://docs.google.com/document/d/1eMJnlBSJQLuSVtIJajiDulmgRZJa5s0Lw9sq-amwnq4/edit?usp=sharing>

Email Sent to the Customer:



We are incredibly happy with the customer feedback we received through the UAT Form. The customer did not mention any pressing concerns and was able to navigate through the different account types effectively. We agree with the customer’s assessment.

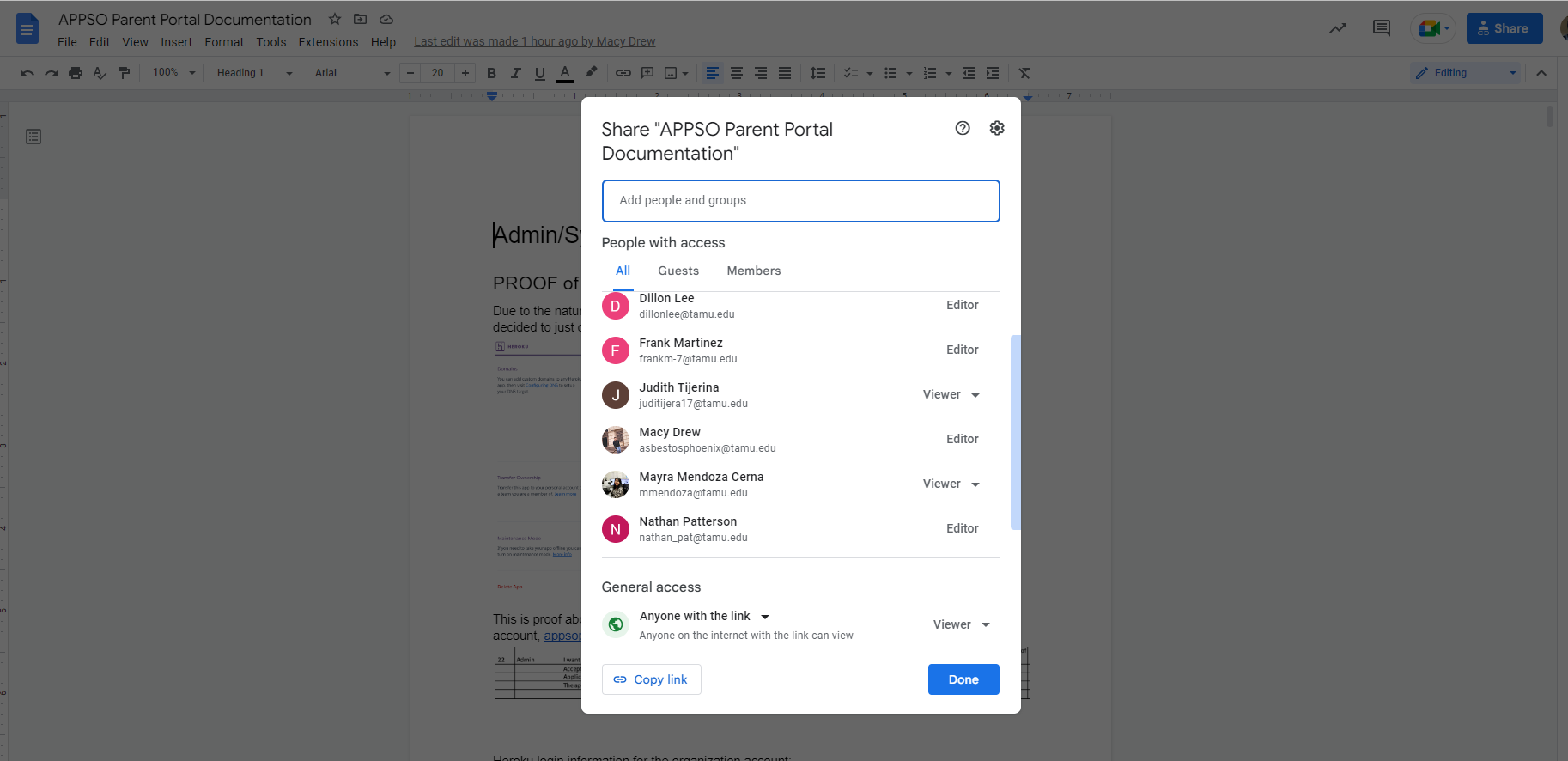
### Customer Feedback Results

The instructor gave our customer a customer feedback form to rate our team on professionalism, and to give their overall feedback on the delivered product. See the Annex for the customer feedback form filled in by the customer.

As of yet, we have not received the customer feedback form, because Professor Wade did not originally have the correct point of contact. (Our original customer representative is now busy with personal needs, also known as her second child).

### Turnover of Artifacts to the Customer

Training was conducted in a Microsoft Teams meeting with the current customer representative, Mayra Cerna. During the meeting, the team answered all her questions regarding the training documentation we created before the meeting. These documents were also presented to the customer during the UAT phase.

User/Admin/System Documentation: <https://docs.google.com/document/d/1mKVl6haz5PUv8I0OmKrQSLZu_gtnL2Ah1lyAzhNzpsw> 

Artifact above shows evidence that the user/admin/system documentation is shared with the customer representatives, Judith Tijerna & Mayra Cerna.

Training Meeting with Customer Representative: <https://tamucs.sharepoint.com/teams/Team-FA22-CSCE431-SoftwareEngineering-AggiePPStudentOrg/Shared%20Documents/Recordings/Meeting%20in%20_APPSO-Parent-Portal_-20221130_150223-Meeting%20Recording.mp4?web=1>

## Instructor Turnover

The artifacts we turned over to the instructor team included our code and several various video presentations. We also were required to convey the user’s filled-out UAT form to the instructor, which encountered a slight hiccup due to our customer representative having changed halfway through the project. (When we took the Aggie Pregnant and Parenting and Students Organization as our customer, we did not expect that our representative would be both pregnant and parenting, but we were very excited to be able to see how our app would benefit members like Judith.)

### Turnover of Artifacts to the Instructor

For the code portion of the turnover, we created a GitHub Classroom repo and cloned our project repo to the new one. For the presentation and demo videos, as well as this report and supporting documents, we compiled them into the MS Teams file repository.

# Configuration Management

A big part of Configuration Management includes identification of the project’s configuration items, and a description of our version and change control process, which are all included in the following sections.

## Configuration Items

Below is the itemization of all our project assets and their location (all gems are **defined** in the Gemfile and are only installed when calling bundle install)

**Environment**

* Docker Engine v20.10.17
* Docker container v4.12.0
* Heroku

**Program**

* Ruby 3.0.2p107
* Rails 6.1.4.1
* Rspec-rails 5.1.2
* PostgreSQL 13.3
* Devise 4.8
* Oauth2 2.0.9
* Omniauth 2.1.0
* Omniauth-oauth2 1.8.0
* Omniauth-rails\_csrf\_protection 1.0.1
* Capybara 3.37.1
* Faker 2.23.0

**Tools**

* GitHub
* RuboCop 1.36.0
* Simplecov 0.21.2
* Brakeman 5.3.1
* Jira
* Visual Studio Code

**Document**

* Data Design Final Version (located in Teams under our team’s channel -> Files -> Turnover (Project Turnover) -> Documents -> APPSO\_DataDesign.docx)
* Figma v9.0
* User Manual (Final version): <https://docs.google.com/document/d/1mKVl6haz5PUv8I0OmKrQSLZu_gtnL2Ah1lyAzhNzpsw/edit?usp=sharing>

**Data**

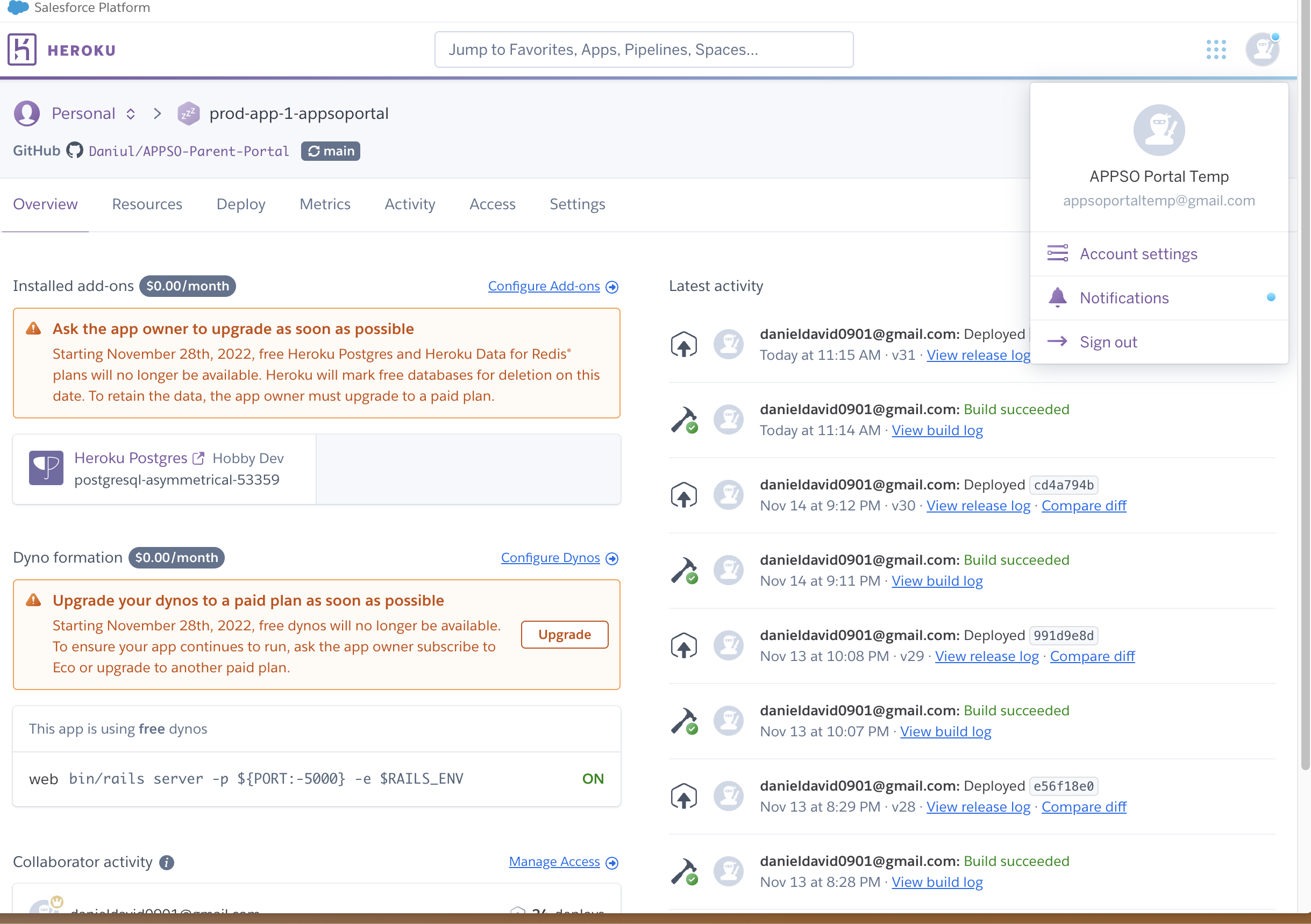
* Final Test Data is on GitHub (<https://github.com/FA22-CSCE431-classroom-8am/sprint-3-appso-parent-portal/tree/main/spec>).
  + By following the link above:
  + Integration test Data is in feature/integration\_spec.rb
  + Unit Test Data is in unit\_spec.rb

Our code is maintained in a classroom GitHub Repository, with the latest code in the ‘Main branch.’ See more information in the section below on “Version Control.”

<https://github.com/FA22-CSCE431-classroom-8am/sprint-3-appso-parent-portal>

Our product and sprint backlog can be found in Jira, with organization name N/A (no organization name) and project name Parent Portal

Our application is deployed in our Heroku account https://stage-appso-portal.herokuapp.com and in the customer’s Heroku account



In our MS Teams repository, subfolder “Turnover/Documents” are the following:

* Sprint 1 report v2 (latest)
* Sprint 2 report v3 (latest)
* Sprint 3 report v2 (latest)
* Project Turnover report v2 (latest)
* Scope v3 (latest)
* Team Working Agreement v1 (latest)
* Data Design v4 (latest)
* Risk Plan v2(latest)
* Security Risk Plan v2 (latest)
* UAT Form v3 (latest)
* Configuration Item Identification v1 (latest)
* Application Show & Tell video v1 (latest)
* Reflection video v1 (latest) and corresponding slide deck v1
* UAT Results
* Customer Feedback Results
* User/Admin/System Documentation (all-in-one document) v2 (Latest)
* Efficiency Summary v1 (latest)
* FK to PK Connection Diagram v1 (latest)

## Version Control

In software engineering it is necessary to track and control the source code developed. In our project, we decided to use GitHub, which offers functionalities, such as integrated issue tracking, collaborative code review, team management, and highlighting of syntax. It allowed us to work on separate features of the application, track bugs, and manage coding tasks.

All team members created separate, feature branches corresponding to the user stories that we were assigned. We made sure to only work in our respective branches. When one member was done implementing a user story, they created a pull request. The pull request includes Acceptance Criteria and Definition of Done, both of which needed to be completed for the feature to be officially done from a customer and developer standpoint. An additional team member was required to conduct a code review of the changes made to the branch, and check that there are no merge conflicts. GitHub CI/CD checks and unit/integration tests were also reviewed to ensure that they all passed. If there was a merge conflict, a more extensive code review is done with one team member and the author to ensure that no expensive overwritten will occur. The process of extensively using GitHub to conduct Continuous Integration and Continuous Delivery, along with maintain Version Control allowed us to have organization in terms of the code we wrote. It felt more professional and organized when we utilized GitHub.

Included in the classroom GitHub is a ReadMe file describing the following:

-Requirements

-External Dependencies

-Installation

-Tests

-Execute your code

-Environmental Variables/Files

-Deployment

-CI / CD process, including description of GitHub actions code

-Support

## Change Control process

Change control involves controlling changes to the baseline of the project which includes software, data, and documentation.

Whenever there someone requested to make changes to either the database design or the code itself, they were required to tell the whole team (not just the product owner), regarding the proposed changes to the target documentation. However, the product owner has the final say and checks to make sure the proposed changes do not seriously affect our project deliverables. If there was a scenario where it did change one of the deliverables, the product owner would then contact the customer representative and engage in discussion regarding the proposed changes. There was a loose chain of command, where everyone can listen and add input to changes a member wants to make on the baseline of the project, and then the product owner will then make an informed decision after the group discussion.

# Project Presentation

We produced 2 video recordings:

1. Video 1: Show and Tell – highlights of the product – 2 minutes in duration  
   Main purpose is for other members of the class to view asynchronously a demo of our application, prior to the in-class presentation on our reflection (see below).
2. Video 2: Reflection -5-7 minutes in duration
   * Lessons Learned (e.g., what we did right, what can be improved, etc.)
     + Timeline Retrospective
   * Acknowledgement of ‘Acts of Service’

This video was played in class, followed by a live Q&A

The following sub-sections describe the content of each segment for Video 2, followed by a reflection of how we felt as we were preparing and presenting the topic.

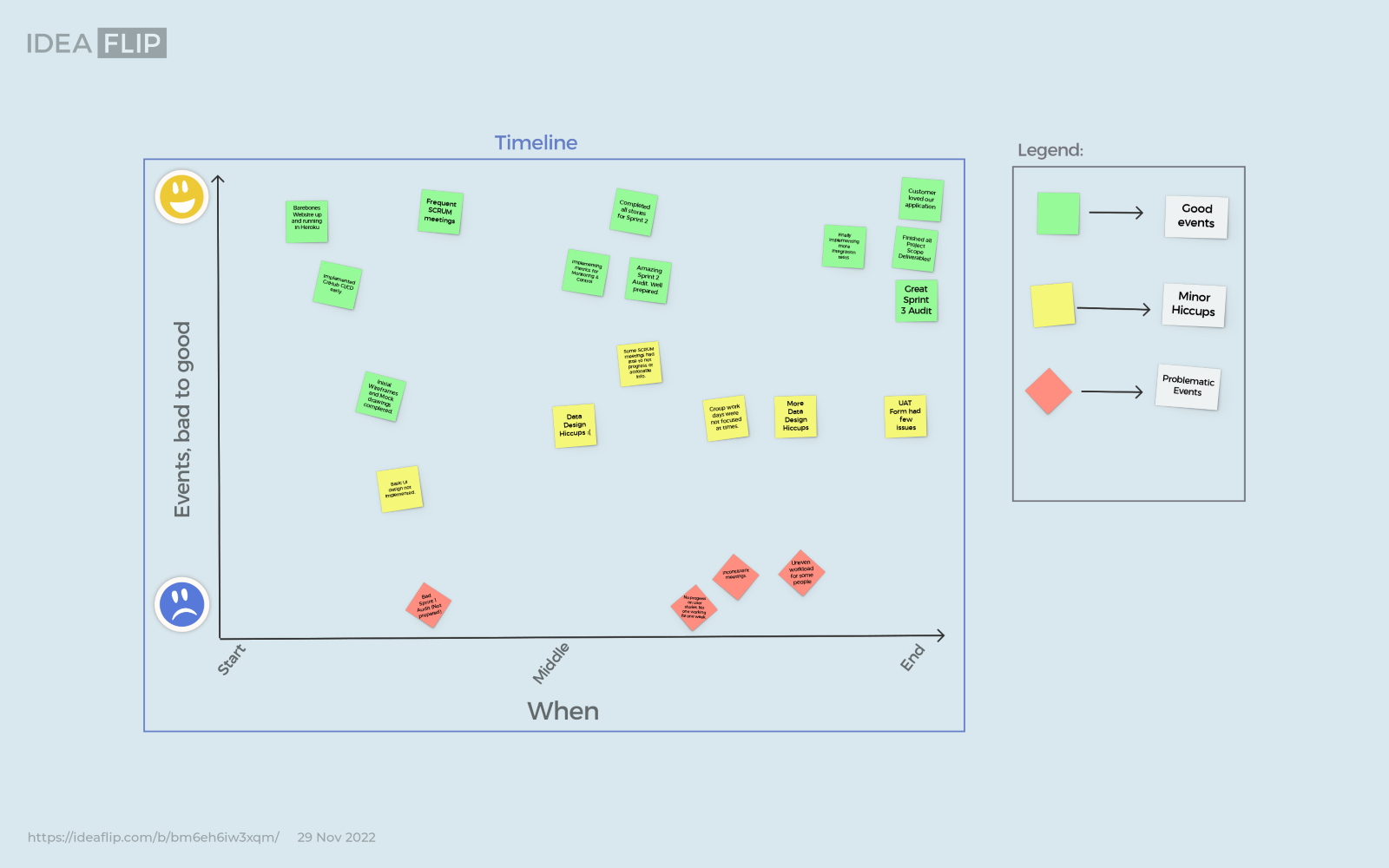
### Lessons Learned

Lessons learned can include the following:

* Team dynamics: what worked and what did not
* Key activities / processes that helped us grow professionally / personally
* How effective was the agile process? What would we change?
* What would we improve about the environment (e.g., hardware, software, infrastructure, working agreement, team space)?
* Were stakeholders’ requirements met? Why or why not?
* What did we do right? How did we celebrate?
* What would we do differently (i.e., in the event of an EPIC FAIL)? Remember that it is best to FAIL – FAST & NOW (in Sprint 1 vs. 3), instead of FAIL – SLOW or LATER (e.g., UAT in front of the customer)
  + How does your experience with failing help you avoid the same mistake next time?
* [other topics can be discussed]

**Timeline Retrospective**

The [timeline retrospective](https://trello.com/c/zHeiCpz8/49-timeline-retrospective) below is a visual mapping of important events, achievements, highs & lows, across the life of the project.



Although we did not present our timeline retrospective visual to the class (due to time constraints), creating the visual map was a great learning experience. We were able to remember good memories, minor hiccups, and adverse events that we had to push through. As we added the events to the board, we started to see that despite having a lot of hiccups and some problematic events, we were able to also have a lot of good events. Our road to completing this project was not perfect, we experienced both happiness and pain during this process. We learned that it is not about the destination, but the **journey** itself, which is what this timeline retrospective board helps realize.

### Acknowledgement of ‘Acts of Service’

We acknowledged team members who went over and above what was expected, describing their specific contribution, during our presentation. We acknowledged them by mentioning members who had outstanding contributions to the project and briefly described such contributions. We included this at the end of the project to express our gratitude to these individuals for helping us complete a successful project. We also acknowledged them with verbal and written recognition in person and on Discord for making outstanding contributions to the project.

# Recommendations for Future Work

* Make sure each team member knows how to run code coverage reports and can understand its output to make code closer to error free.
* Make sure to schedule sprint reviews with your customer at least a week in advance.
* Please, please, please review Data Design Entity Relationship Diagram concepts before taking this class!
* Please, please, please review your data design a multitude of times before submitting a copy to Professor Wade.
* When picking out dates and times for scrum meetings set a reoccurring schedule that works for all members 95% of the time. It can get tricky to have full participation during scrum meetings if a routine is not present.
* To avoid merge conflicts and performing duplicate work on the same user story or feature, communicate with your teammates whenever you begin a new task.

# Acknowledgements

[Acknowledge team members in this section who exhibited exceptional teamwork, over and above what was expected of them. Be specific about their contribution.]

* Thierry David
  + TLDR; truly went above and beyond as a back-end lead, on-call developer, and team player throughout the project.
  + Created a phenomenal User/Admin/System Documentation for customer handoff
  + Made sure to fully practice the Monitoring & Control metrics within Sprint 2 and 3, creating documents summarizing our practice utilizing such metrics that heavily influenced our sprint audit scores
  + handled deployment through Heroku and the GitHub Workflow CI/CD tests implementations.
  + Created the Training document and the application documentation for both the admin and officers.
  + He implemented all the integration tests (around 10+ tests) at the end of sprint 3.
  + Assisted all team members with code issues (pair programming and advice)
* Frank Martinez
  + TDLR; was always available whenever Pull Request reviews and code reviews were needed asap. Made sure to deliver on user stories early or on time. Frank was also a part of the back-end team.
  + One of the things our team struggled with in the beginning of the project development was how to run the development environment, he wrote a guide to help familiarize his teammates with running Docker and editing code from within it.
  + Because of his additional efforts to learn Ruby on rails in Sprint 1 and Sprint 2, he gained an advanced conceptual understanding of ruby on rails. This allowed him to teach most of his teammates the Ruby on rails infrastructure

# References

Cite any external references, source of code / other artifacts you used but was developed by someone else.

See examples below:

* <https://www.atlassian.com/software/jira>
* [https://www.unleashed-technologies.com/blog/2014/08/01/what-github-and-how-can-it- benefit-your-development-team](https://www.unleashed-technologies.com/blog/2014/08/01/what-github-and-how-can-it-benefit-your-development-team)
* GitHub repos (.Honesty is the best policy!)
* [List other references here.]

# Annex: UAT form and/or Customer Feedback Filled out by the Customer

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **#** | **User Story with its Acceptance Criteria** | | | **For Customer Use only** | | | | |
|  | **Persona (Who)** | **Requirement (What)** | **Value (Why)** | **Critical?** | | **Test Result** | | **Comments** |
|  |  |  |  | **Yes** | **No** | **Accept** | **Reject** |  |
|  |  |  |  |  |  |  |  |  |
| **1** | User (Member) | I need my information represented in an organized way | so that it can be viewed and manipulated efficiently. |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
|  |  | **Acceptance Criteria** |  |  |  |  |  |  |
|  |  | Data recorded (creating new events, new users, etc), is retained through login/logout |  | x |  | x |  | I can see my information as soon as I log in |
|  |  |  |  |  |  |  |  |  |
| **2** | User (Member) | I want to be able to fill out a member sign-up form | in order to apply as a member of the organization |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
|  |  | **Acceptance Criteria** |  |  |  |  |  |  |
|  |  | Members can create a unique login and password |  | x |  | x |  |  |
|  |  | Form is easily seen and accessible from the existing members' login screen |  | x |  | x |  |  |
|  |  | Members cannot create an admin account without permission from an officer |  | x |  | x |  |  |
|  |  |  |  |  |  |  |  |  |
| **3** | User (Member) | I need to be able to log into my member account | so that I can view my participation data and scholarship application |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
|  |  | **Acceptance Criteria** |  |  |  |  |  |  |
|  |  | Members can log in with their username (or google email) and password |  | x |  | x |  |  |
|  |  | Members can recover their passwords via email if forgotten or lost |  | x |  | x |  |  |
|  |  | Members are not able to login as officer |  | x |  | x |  |  |
|  |  |  |  |  |  |  |  |  |
| **4** | Officer | I need to be able to log into my admin account | so that I can view membership information and scholarship records |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
|  |  | **Acceptance Criteria** |  |  |  |  |  |  |
|  |  | Officers can visually see the Google log-in page when asked to sign in (with Google) |  |  | x | x |  |  |
|  |  | Officers can login with their Google tamu.edu account. |  | x |  | x |  |  |
|  |  | Members are not able to login as an officer. |  | x |  | x |  | They can choose the officer login but they won't have the approval, maybe that should be stayed when choosing the role |
|  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
| **5** | Officer | I would like to see a list of members | so that I can know who is currently in the organization as a whole |  |  |  |  |  |
|  |  | **Acceptance Criteria** |  |  |  |  |  |  |
|  |  | Officer can see an entire list of members. |  | x |  | x |  |  |
|  |  | List is sorted by alphabetical order (last name) |  |  | x | x |  | maybe give more option on ordering the list, like from recent to oldest |
|  |  | Only Valid letters for first/last names are displayed (no special characters, symbols, etc.) |  |  | x |  |  | was not able to test this but if possible I would leave it open, as many international student use their foreign letter in their names. |
|  |  | List is centered in the screen |  |  | x | x |  |  |
|  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
| **6** | User (Member) | I need to have a safe storage location for the data for my website | so that I can keep track of member and scholarship information. |  |  |  |  |  |
|  |  | **Acceptance Criteria** |  |  |  |  |  |  |
|  |  | Members can access their member data (Events attended, scholarship application) |  | x |  | x |  |  |
|  |  | Members scholarship information auto-saves |  | x |  | x |  |  |
|  |  | Members don’t have access to private admin data |  | x |  | x |  |  |
|  |  |  |  |  |  |  |  |  |
| **7** | User (Member) | I need to be able to fill out questions/information for the scholarship application | so that I can be in the running for the scholarship |  |  |  |  |  |
|  |  | **Acceptance Criteria** |  |  |  |  |  |  |
|  |  | Places where a user can enter their responses for each question are present |  | x |  | x |  |  |
|  |  | User is able to save the scholarship application |  | x |  | x |  |  |
|  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
| **8** | Officer | I need to filter completed scholarship applications | so that I can choose scholarship winners |  |  |  |  |  |
|  |  | **Acceptance Criteria** |  |  |  |  |  |  |
|  |  | Officer is able to view a list of members that have completed their scholarship application. |  | x |  | x |  |  |
|  |  | Officer can see the name and email of each user who completed the application. |  | x |  | x |  |  |
|  |  |  |  |  |  |  |  |  |
| **9** | User (Member) | I need to be able to save answers and edit them later | so that I can conveniently start and stop my application around me and my kids' schedule. |  |  |  |  |  |
|  |  | **Acceptance Criteria** |  |  |  |  |  |  |
|  |  | Edit button is visible on member’s dashboard page to get back into the scholarship application. |  | x |  | x |  |  |
|  |  | Save button is present within the application. |  | x |  | x |  |  |
|  |  | Fields edited and saved by the user should repopulate upon renavigation to their application. |  | x |  | x |  |  |
|  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
| **10** | User | I need to be able to log-in to Parent Portal securely | in order to keep my username and password from being exposed. |  |  |  |  |  |
|  |  | **Acceptance Criteria** |  |  |  |  |  |  |
|  |  | Users can log-in with their existing Google Account |  |  |  |  |  |  |
|  |  |  |  | x |  | x |  | I was able to log using the sign up, but when tipping my mail and password I couldn't access |
|  |  |  |  |  |  |  |  |  |
| **11** | Officer | I need to be able to check off if a member attended a event/meeting | in order to keep track of member attendance |  |  |  |  |  |
|  |  | **Acceptance Criteria** |  |  |  |  |  |  |
|  |  | Officer is able to assign a member as present/absent for an event. |  | x |  | x |  |  |
|  |  | Clicking "Present" or "Absent" should update member’s participation information to include such an event. |  | x |  | x |  |  |
|  |  | No other inputs should be allowed. |  | x |  | x |  |  |
|  |  |  |  |  |  |  |  |  |
| **12** | User (Member) | I'd like to be able to see my personal participation stats tracked in an easy-to-read format | so that I can know how to improve my standing |  |  |  |  |  |
|  |  | **Acceptance Criteria** |  |  |  |  |  |  |
|  |  | Member sees the dashboard display immediately upon login |  | x |  | x |  |  |
|  |  | Dashboard includes application completion and link to edit application |  | x |  | x |  |  |
|  |  | Dashboard includes attendance data |  | x |  | x |  |  |
|  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
| **13** | Officer | I need to be able to see specific and overall member rankings based on various stats | so that I can make a data-driven decision on the winners |  |  |  |  |  |
|  |  | **Acceptance Criteria** |  |  |  |  |  |  |
|  |  | Officer can see member name, role, scholarship progress/completed, events attended, and notes. |  | x |  | x |  |  |
|  |  | Officer can sort by name name, role, events, and other criteria |  | x |  | x |  |  |
|  |  |  |  |  |  |  |  |  |
| 14 | Officer | I need to be able to visualize all members' involvement rankings | so that I can help decide scholarship winners |  |  |  |  |  |
|  |  | **Acceptance Criteria** |  |  |  |  |  |  |
|  |  | Officer should be able to click a filter to see a page ranking all the members based from highest involvement to lowest involvement. |  | x |  | x |  |  |
|  |  | Should only include valid members that have registered in the site. |  | x |  | x |  |  |
|  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
| 15 | User (Member) | I need to be able request to be an admin, or officer and go through signup process | so that I can ensure that I have the right member role for the portal website |  |  |  |  |  |
|  |  | **Acceptance Criteria** |  |  |  |  |  |  |
|  |  | Option should be available to choose role within organization (Member, Officer, Faculty, Admin) |  | x |  | x |  | maybe explaining what each role mean |
|  |  |  |  |  |  |  |  |  |
| 16 | APPSO president/admin | I need to be able approve/deny new user registrations | so that I can verify member identity and roles |  |  |  |  |  |
|  |  | **Acceptance Criteria** |  |  |  |  |  |  |
|  |  | Admin can see on a list, all new members who have requested for registration |  | x |  | x |  |  |
|  |  | Admin can view the email and the role that user requested. |  | x |  | x |  |  |
|  |  | Admin can choose to approve or deny a certain user |  | x |  | x |  |  |
|  |  |  |  |  |  |  |  |  |
| 17 | Officer | I need to be able to create events | in order to keep track of user attendance on different days. |  |  |  |  |  |
|  |  | **Acceptance Criteria** |  |  |  |  |  |  |
|  |  | Admin or officer is able to click a button to create an event with name, time, and description. |  | x |  | x |  |  |
|  |  |  |  |  |  |  |  |  |
| 18 | User (member) | I would like to be able to see my own application completion percentage | so that I can budget my time for completing it. |  |  |  |  |  |
|  |  | **Acceptance Criteria** |  |  |  |  |  |  |
|  |  | Completion percentage appears on member dashboard in an easily understood format |  | x |  | x |  |  |
|  |  | Percentage is automatically updated upon changes to application |  | x |  | x |  |  |
|  |  | Complete range is only 0-100 |  | x |  | x |  |  |
|  |  | Completion bar is visible on live website. |  | x |  | x |  |  |
|  |  |  |  |  |  |  |  |  |
| 19 | Scholarship Committee (mix of Officers and School Faculty) | I would like be able to see members' application completion percentage | so that I can remind them to finish applying |  |  |  |  |  |
|  |  | **Acceptance Criteria** |  |  |  |  |  |  |
|  |  | Members' completion status appears in a list to the officer when they navigate to an appropriate view |  | x |  | x |  |  |
|  |  | Complete range is only 0-100 |  | x |  | x |  |  |
|  |  | Only scholarship committee can see this view, no one else should have access |  | x |  | x |  |  |
|  |  |  |  |  |  |  |  |  |
| 20 | Scholarship Committee (mix of Officers and School Faculty) | I would like to view completed scholarship applications | so that I can start to make a decision on who is the scholarship winner |  |  |  |  |  |
|  |  | Acceptance Criteria |  |  |  |  |  |  |
|  |  | Scholarship Committee can view a member's completed application from a list of members |  | x |  | x |  |  |
|  |  |  |  |  |  |  |  |  |
| 21 | Scholarship Committee (mix of Officers and School Faculty) | I would like to be able to make notes on members' applications | so that I can remember my thoughts when considering scholarship applicants |  |  |  |  |  |
|  |  | Acceptance Criteria |  |  |  |  |  |  |
|  |  | Clicking on the notes should not be obtrusive to the committee member |  | x |  | x |  |  |
|  |  | Option to add notes to each member |  |  | x | x |  |  |
|  |  |  |  |  |  |  |  |  |
| 22 | Admin | I want the application installed in my own Heroku account | so that the organization has full ownership of the application |  |  |  |  |  |
|  |  | Acceptance Criteria |  |  |  |  |  |  |
|  |  | Application can be seen on the Heroku Dashboard |  | x |  | x |  |  |
|  |  | The app owner within the Heroku settings states the organization's admin e-mail or organization e-mail. |  | x |  | x |  |  |
|  |  |  |  |  |  |  |  |  |
| 23 | Admin | Documentation / online help is sufficient | so that I can understand how to use the system |  |  |  |  |  |
|  |  | Acceptance Criteria |  |  |  |  |  |  |
|  |  | Admin is able to set up and maintain the application without the active presence of the development team |  | x |  | x |  |  |
|  |  |  |  |  |  |  |  |  |
| 24 | User (Member) | Documentation / online help is sufficient | so that I can understand how to use the system |  |  |  |  |  |
|  |  | Acceptance Criteria |  |  |  |  |  |  |
|  |  | Able to use the application without the active presence/help of the dev team. |  | x |  | x |  |  |
|  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
| 25 | Admin | Training provided is sufficient | so that I can train others in the administration/maintenance of the system |  |  |  |  |  |
|  |  | Acceptance Criteria |  |  |  |  |  |  |
|  |  | Can train others without help from the development team |  | x |  | x |  |  |
|  |  |  |  |  |  |  |  |  |
| 26 | User (Member) | Training provided is sufficient | so that I can train others on how to use the system |  |  |  |  |  |
|  |  | Acceptance Criteria |  |  |  |  |  |  |
|  |  | Can train others without help from the development team |  | x |  | x |  |  |
|  |  |  |  |  |  |  |  |  |

Product Backlog

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Number** | **User Story NOT Implemented** | | | **For Customer Use only** | | | | |
|  | **Persona (Who)** | **Requirement (What)** | **Value (Why)** | **Critical?** | | **Agreed?** | | **Comments** |
|  |  |  |  | **Yes** | **No** | **Yes** | **No** |  |
| **1** | **All User Stories Implemented** |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
| **2** |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
| **3** |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
| 4 |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |

The Customer Feedback Form is currently unavailable.