

Team: ANTE

*Elias Tracy - Product Owner | Tuong Tran - SCRUM Master
Andrew Imwalle - Developer | Nick McDonough - Developer*

Pivotal Tracker: <https://www.pivotaltracker.com/n/projects/2598148>

GitHub: <https://github.com/tamu-edu-students/fighting-aggies-platform>

Slack: <https://app.slack.com/client/T04GX9UU3V3/C04H08B4N0J>

Heroku: <https://fighting-aggies-platform.herokuapp.com/>

Coach login:

Username: faapplicationuser@gmail.com

Password: password2023

Admin login:

Username: faapplicationmanager@gmail.com

Password: password2023

Customer Meeting Date:

1/13/2023

Time:

8:05 AM

Location:

Ibis Singapore on Bencoolen

Meeting Notes:

- Start off meeting with project demo
 - Change admin ui
 - Make lighter screen overall
 - 12manfoundation.com
 - Bad gray scale input box
- Coach view
 - Coach can see subset of players
 - Admin setups who they can see
 - Graphs
 - For a given route, and a given players, what's their accuracy
 - Single page app
 - When you click a player's route, shows their player info on the same page
 - Components?
 - Page with all of the routes
 - Should be several clicks away
 - Main importance is summary
 - Player info page
 - Should have the most recent report
- Views

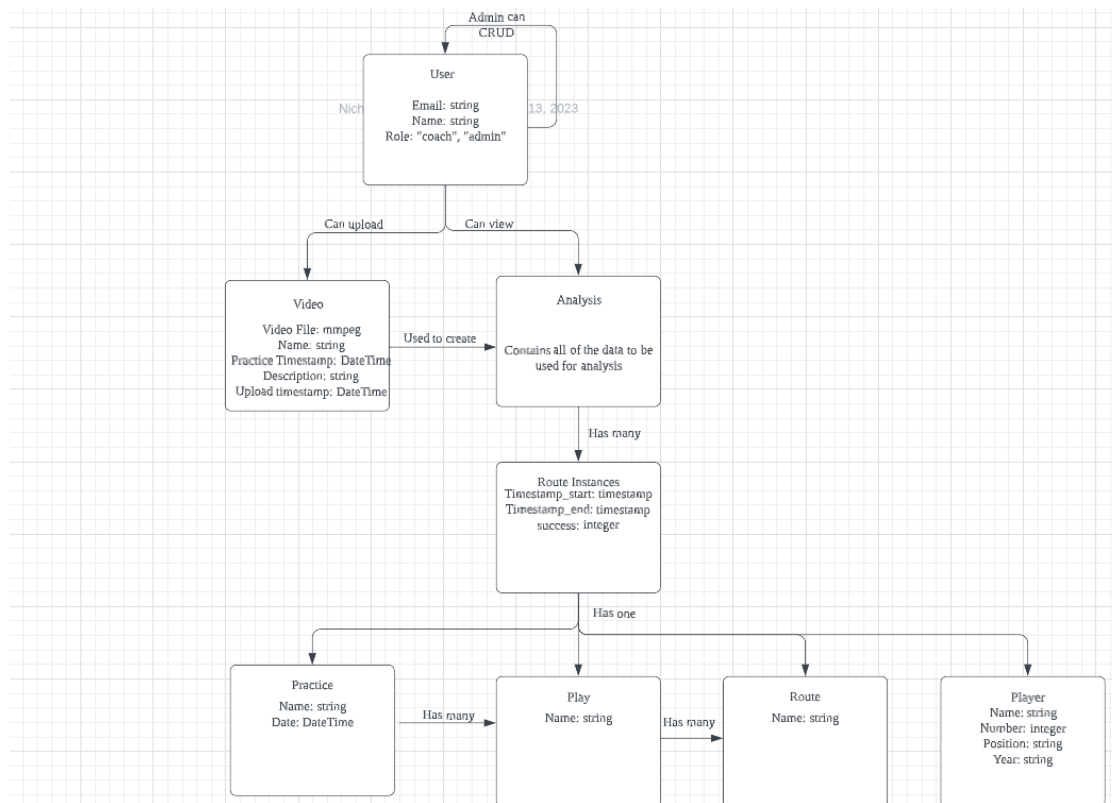
- Make it feel like it's all on one page
- Several workflows
 - Uploading videos
 - Data repository
- Data exploration view
 - Load data
 - Specify what you want
- Report Generation View
 - Load report card from templates from data from data exploration view
- Report generation content
 - Mostly tabular data
 - 1-2 charts/plots
 - Coach should be able to design report how they want
 - They can put whatever they want
 - Some may want just a table
 - Some may want fancy charts of everything
 - Some may want report cards for each player
 - Design a domain specific language?
 - Creating a simpler SQL
 - Simple excel graph builder
 - Custom report card templates
 - Coaches can make their templates and save them
 - Make a few pre made templates
 - Player report
 - Play report
 - Run report
 - PRactice report
 - Basic one
 - Players with routes on opposite curves
 - Intersection is accuracy
- Prof advice
 - Make several templates to see what we need in the back end
 - Find common similarities to see what we need on the front end
 - Will take several semesters
 - “Show me these players on these routes”
 - Generates table of players, routes, accuracy
 - Make tableau?
 - Choose data -> choose data format -> make report
 - Blg focus
 - Make it easy to pass on
- By next client meeting
 - Choose players
 - Choose routes
 - Choose time range or choose practices

User Stories

User stories successfully implemented this iteration:

User Story	Points	Completed By
Feature: Drop down role assignment	1	Eli
Feature: User should be able to login with a one time passcode	3	Andrew
Feature: Confirmation before deleting a user	1	Eli
Feature: Log out Dashboard	1	Nick
Feature: Log out Admin	1	Nick
Feature: Restrict Admin Page Access	1	Nick/Andrew
Feature: Stylized Admin Dashboard	2	Tuong
Feature: User does not have to sign in repeatedly / Restrict Page Access	2	Andrew

Design Diagram

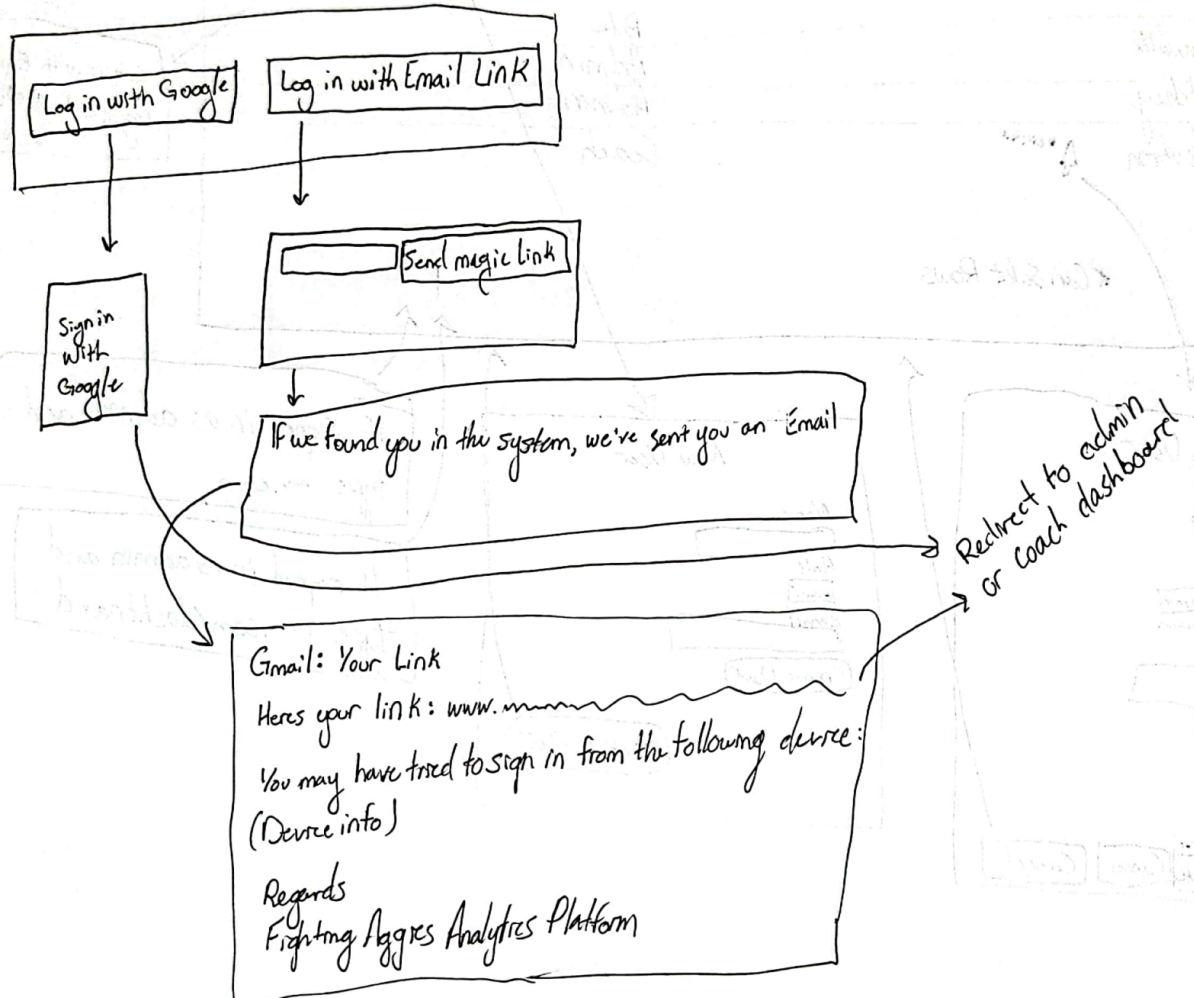


Lo-fi UIs

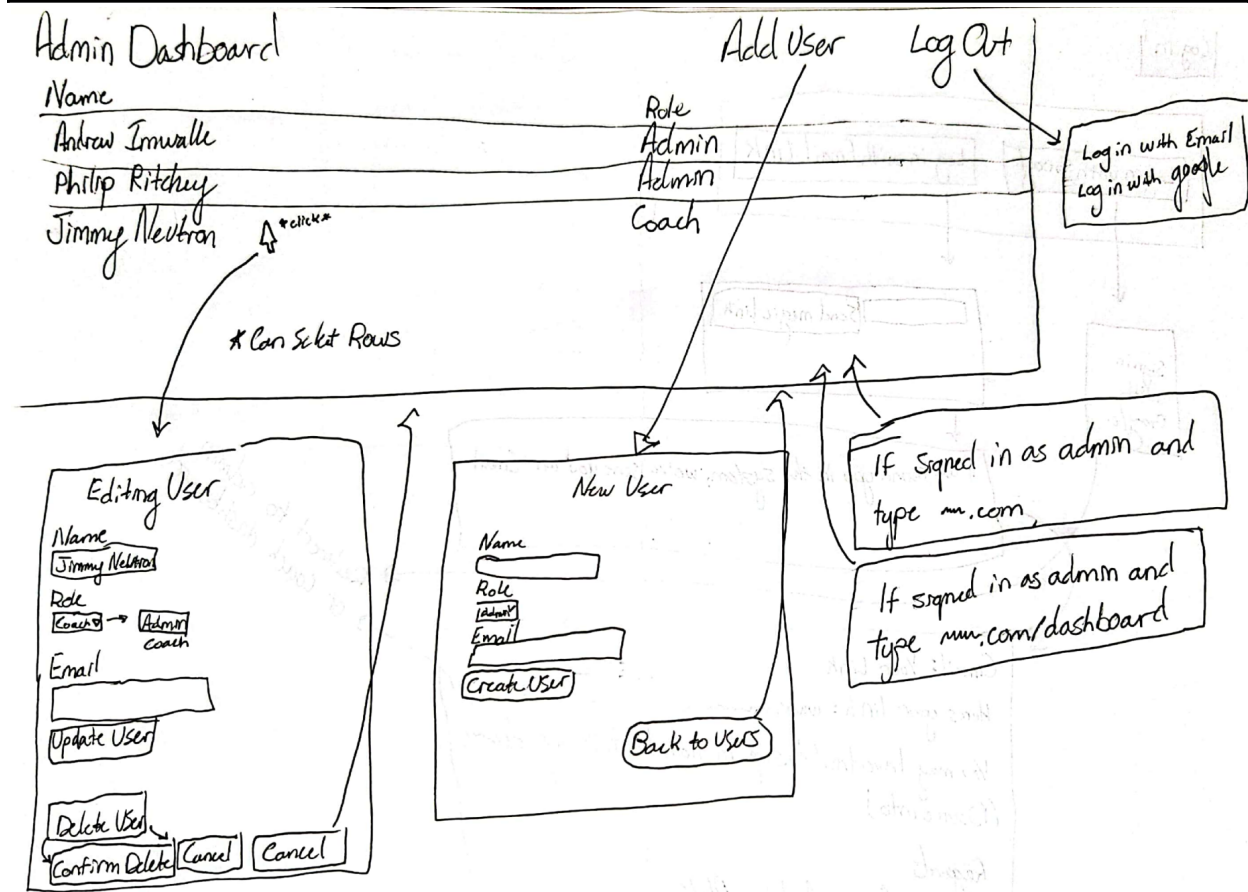
User Stories related to this User Interface

Feature: User should be able to login with a one time passcode

Log in



User Stories related to this User Interface
Feature: Stylized Admin Dashboard
Feature: Confirmation before deleting a user
Feature: Drop down role assignment
Feature: Log out Admin



User Stories related to this User Interface

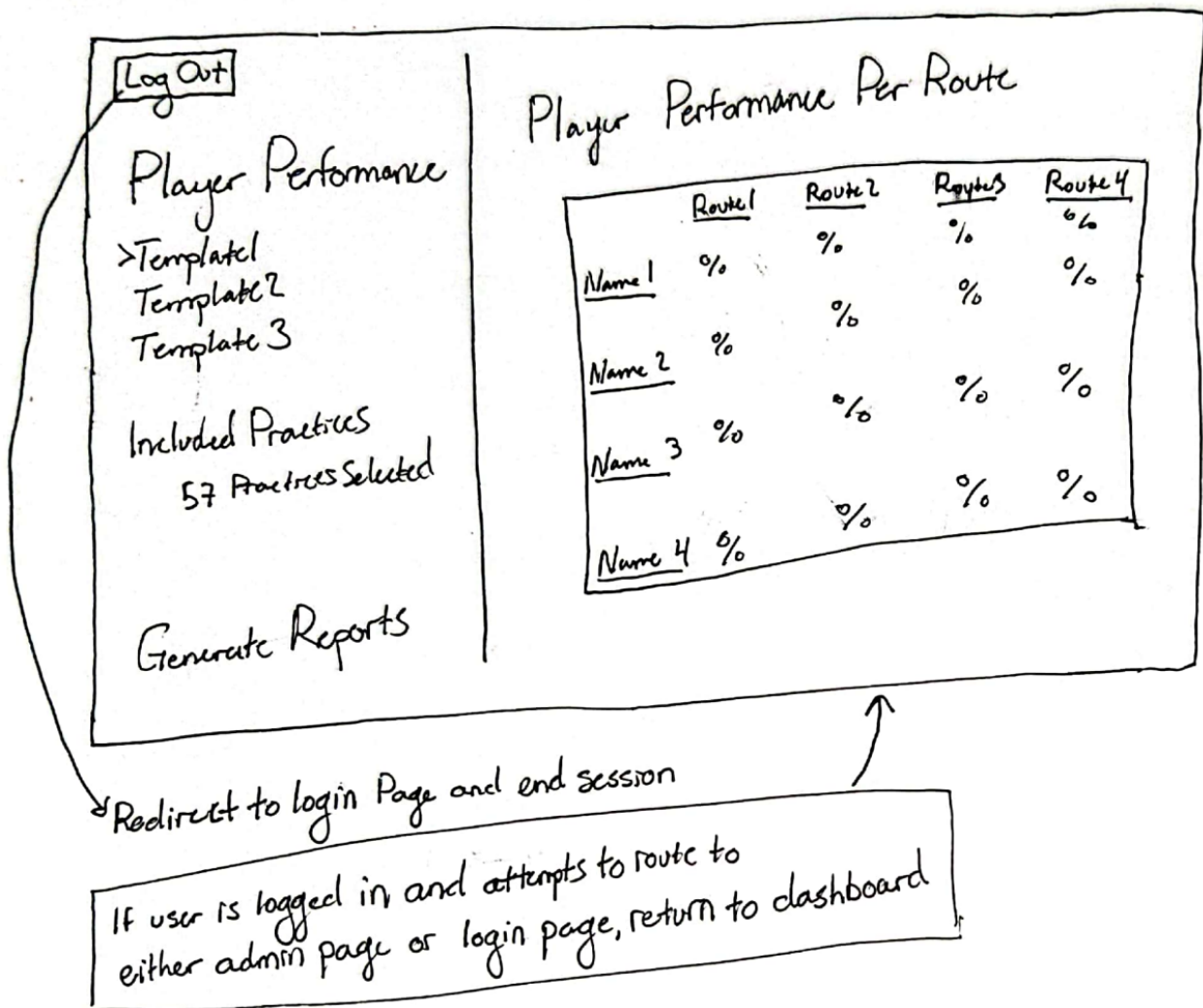
Feature: Restrict Admin Page Access

Feature: User does not have to sign in repeatedly / Restrict Page Access

Feature: Log out Dashboard

Feature: View Analytics Dashboard

Coach Dashboard



Coach Dashboard

when user selects name 1 from Template 1:

Log out

Player Performance

> Template 1

Template 2

Template 3

Included Practices

57 Practices Selected

Generate Reports

Name 1

Route Name	Play Name	Timestamp	Success
~~~~~	~~~~~	00:00:00	1
~~~~~	~~~~~	00:00:10	0
~~~~~	~~~~~	00:00:20	1
~~~~~	~~~~~	00:00:30	1

If the user is on template 1, and they select a player's name, they will view data relevant to that specific player during the included practices.

Coach Dashboard

When user selects route 1 from Name 1

Log out

Player Performance

> Template 1

Template 2

Template 3

Included Practices

57 Practices Selected

Generate Reports

Name 1 > Route 1

Play Name	Timestamp	Success
~~~~~	~~~~~	1
~~~~~	~~~~~	0
~~~~~	~~~~~	1
~~~~~	~~~~~	0

If the user is on the player's data and they select a specific route, they will view data relevant to that specific player during the included practices, on the specific route.

Updated User Stories

Feature: User does not have to sign in repeatedly / Restrict Page Access
--

This feature was rewritten and refactored from the previous code. The pages that the users are able to access need to align with their respective roles. Therefore any coach logged in should not be able to view the admin dashboard or the page to edit users, add users etc. If any admin is logged in, they should not be able to view the dashboard or any pages that relate to it. Additionally, when a user is logged in and they navigate back to the homepage, they would need to login originally. This has been changed to immediately navigate the user back to their respective dashboards.

Incomplete User Stories

N/A

Legacy Report:

Legacy User Stories	Points	Assigned To
Update Rakefile	N/A	Eli

Due to the various external dependencies of the legacy code, there are a number of installations that need to be performed on a new instance of a machine in order to run the code. A goal of this sprint is to be able to run a single rake task setup in order to complete all of the installations that a user may need. Unfortunately it is not yet possible to get rid of external dependencies due to the fact that the legacy code was written in react, which needs yarn and npm to be installed in order to function properly.

It is an ongoing goal for the team to refactor and eventually replace the legacy code that exists as it does not align to the client's vision for this product. Thus far, the team has successfully replaced the login system and created an admin dashboard for the client so that they may manage the users. As the sprints continue, the team will be replacing the dashboard that was written in react. This is due to the fact that the dashboard does not deliver the information that the client desires and the react interface requires additional dependencies that cannot be placed in the gemfile. As soon as the code is refactored out of the product, the external dependencies will no longer be necessary.

In the gemfile, the gems have been relocated to be in their respective locations. This ensures that the user does not need to install pg as this will only be necessary in the production version of the product. Additionally, the 'cucumber' and 'rspec' gems have been relocated to be strictly in the testing version of this product.

Code Evaluation:

Cucumber:

- 28 scenarios (1 undefined, 6 pending, 21 passed)
- 143 steps (9 skipped, 2 undefined, 6 pending, 126 passed)
- 100 / 130 LOC (76.92%) covered. (Note that we are not testing legacy code that has not been refactored yet.)

Rspec:

- Passing 29 examples

SimpleCov:

All Files (94.87% covered at 2.58 hits/line)

35 files in total.

390 relevant lines, 370 lines covered and 20 lines missed. (94.87%)

File	% covered	Lines Relevant	Lines	Lines covered	Lines missed	Avg. Hits / Line
app/channels/application_cable/channel.rb	0.00 %	6	4	0	4	0.00
app/channels/application_cable/connection.rb	0.00 %	6	4	0	4	0.00
app/controllers/admin_controller.rb	100.00 %	8	4	4	0	5.50
app/controllers/application_controller.rb	95.65 %	38	23	22	1	7.04
app/controllers/login_controller.rb	100.00 %	50	29	29	0	13.14
app/controllers/pages_controller.rb	100.00 %	7	3	3	0	3.33
app/controllers/users_controller.rb	90.24 %	82	41	37	4	2.32
app/helpers/admin_helper.rb	100.00 %	4	1	1	0	2.00
app/helpers/api/v1/analyses_helper.rb	100.00 %	8	3	3	0	2.00
app/helpers/api/v1/visions_helper.rb	100.00 %	8	3	3	0	2.00
app/helpers/application_helper.rb	100.00 %	4	1	1	0	2.00
app/helpers/login_helper.rb	100.00 %	4	1	1	0	2.00
app/helpers/pages_helper.rb	100.00 %	4	1	1	0	2.00
app/helpers/users_helper.rb	100.00 %	4	1	1	0	2.00
app/jobs/application_job.rb	0.00 %	9	2	0	2	0.00
app/mailers/application_mailer.rb	0.00 %	6	4	0	4	0.00
app/models/application_record.rb	100.00 %	5	2	2	0	2.00
app/models/user.rb	100.00 %	10	3	3	0	2.00
config/application.rb	100.00 %	24	6	6	0	1.00
config/boot.rb	100.00 %	6	3	3	0	1.00
config/environment.rb	100.00 %	7	2	2	0	1.00
config/environments/test.rb	100.00 %	67	19	19	0	1.00
config/initializers/assets.rb	100.00 %	14	1	1	0	1.00
config/initializers/content_security_policy.rb	100.00 %	26	0	0	0	0.00
config/initializers/filter_parameter_logging.rb	100.00 %	10	1	1	0	1.00
config/initializers/inflections.rb	100.00 %	17	0	0	0	0.00
config/initializers/omniauth.rb	100.00 %	5	2	2	0	1.00
config/initializers/passwordless.rb	100.00 %	15	10	10	0	1.00
config/initializers/permissions_policy.rb	100.00 %	12	0	0	0	0.00
config/routes.rb	100.00 %	43	35	35	0	1.00
spec/controllers/application_controller_spec.rb	100.00 %	62	34	34	0	1.47
spec/controllers/login_controller_spec.rb	100.00 %	88	53	53	0	1.38
spec/controllers/user_spec.rb	100.00 %	72	46	46	0	1.17
spec/models/user_spec.rb	100.00 %	57	34	34	0	1.00
spec/rails_helper.rb	92.86 %	67	14	13	1	0.93

Generated by [simplecov](#) v0.22.0 and [simplecov-html](#) v0.12.3
using Cucumber Features, RSpec

Rubycritic:

Overview:

