Iteration 3 Report

Project - Child Care Enrollment Management System

About

Organization/Stakeholder - Child Care Group (https://childcaregroup.org/)
Client - Yan Li (yli@ccgroup.org)

Team Roles

Tanmai Harish - Scrum Master
Daehee Han - Product Owner
Chidambaram Ganesan
Bharath Kumar Ravichandran
Atharva Phand
Noah Pang

Client Demo Meeting

- We changed our client meeting from biweekly to weekly on 27th October 2023.
- The demo meeting for iteration 3 was on 27th October, 2023 1:30-2:35 PM.
- Client Feedback
    - Need a single page for listing all cases, and users should be able to filter out cases depending on the needs.
        - Cases not ready to be assigned (package not received)
        - Cases ready to be assigned (package received and not assigned)
        - Already assigned cases
    - New cases are ready to be assigned only after the package is received from the customer, so not all unassigned cases are ready to be assigned.
    - Want column to show outreach date, the date when package was sent, return status, and user assigned to case on cases page.
    - Only Manager can change the user assigned to the case.
    - For Non-manager staff, they should only see cases assigned to them.
    - Use the term “Cases” instead of “Requests”.
    - Regarding app design, keep Noah’s sidebar style, and keep Daehee’s style for page content and logo.

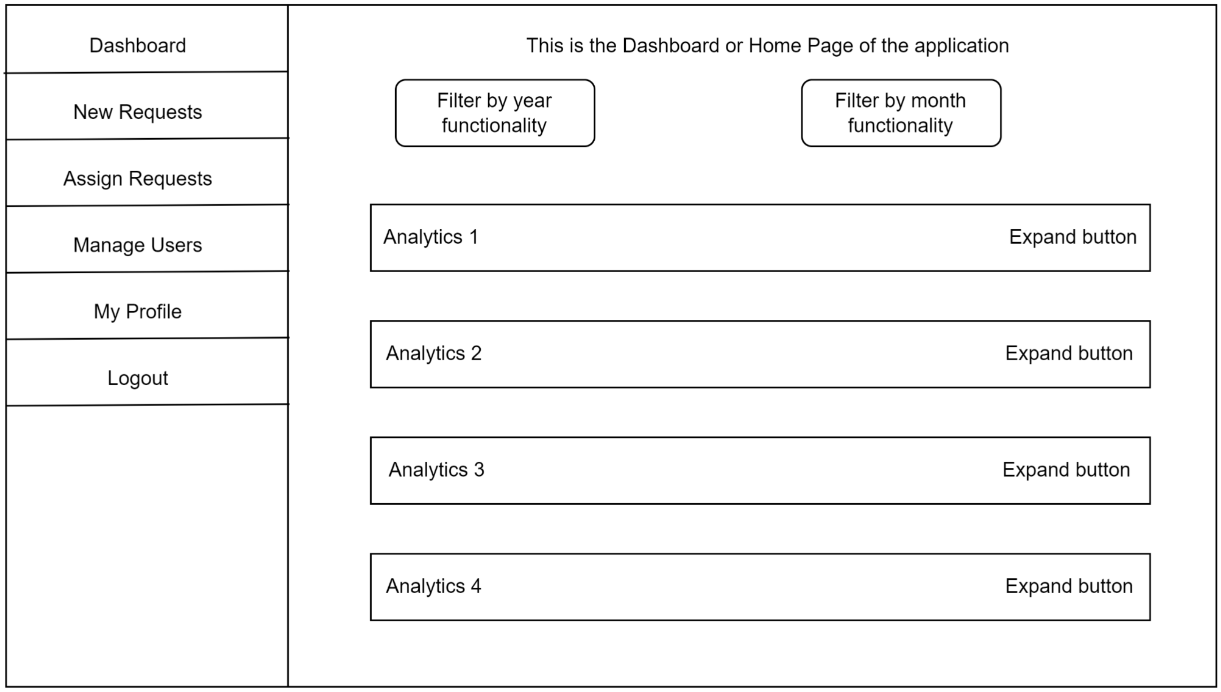
- New Feature Request:
    - Analytics Page:
        - Critical part of app
        - Filter following data by year
            - Outreach data
                - Number of families outreached
                - Chart showed proportion of waiting for response and not returned
            - Placement data
                - Number of families and children currently enrolled
                - Statistics on not enrolled reason

User Stories implemented in Iteration 3

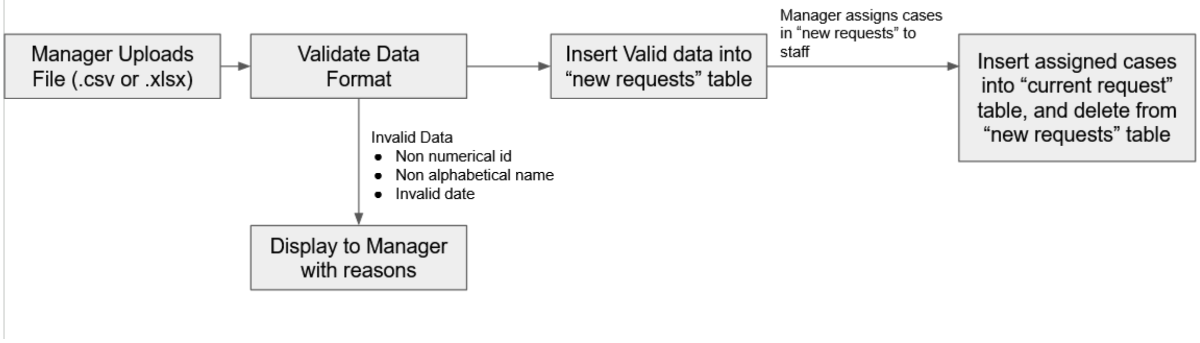
1. Setup My Requests Page for Staff, Minor DB changes
    a. On the 'My Requests' page the staff can see the requests which are assigned only to them.
    b. They can update the case as well and save the changes.
2. Integrate dashboard UI with user features
    a. The dashboard UI used to link to different pages and was updated to integrate the dashboard and features together following the same format.
3. Add UI for users and cases
    a. Added UI for the users pages and cases pages including users list, manage users, upload new cases, and assign new cases
4. Fix Behave Tests for Upload Feature
    a. Since the upload file feature was enhanced and change, the behave steps had to be fixed to accommodate the change
5. Add Unit test cases for login addition
    a. Added unit test cases for the login controller and helper
    b. Added unit test cases for the Set password feature
6. Fix behave cases for Login and set password
    a. Added behave steps for the newly developed Set password feature.
7. Assign New Requests Popop & Rest of logic
    a. The modal popup for assigning a request to a user, populating the users in the dropdown, and sending the post request for the assignment and handling the response. Implement Unit Tests, behave tests, and the UI

Design Diagrams

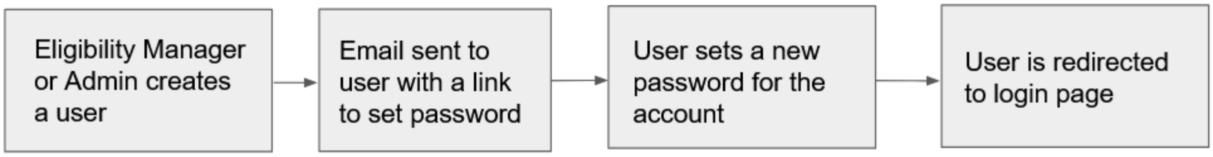
View of the Homepage for the role ‘Eligibility Manager’
Eligibility Manager has access to all the services as shown below. The homepage is nothing but the dashboard page where various statistics such as number of requests completed in a month/year, number of children enrolled in a time period, etc. can be seen in an efficient manner.



New Cases Flowchart


Manager Uploads File (.csv or .xlsx)
Validate Data Format
Insert Valid data into “new requests” table
Manager assigns cases in “new requests” to staff
Insert assigned cases into “current request” table, and delete from “new requests” table
Invalid Data
Non numerical id
Non alphabetical name
Invalid date
Display to Manager with reasons

Set new user password flow:


Eligibility Manager or Admin creates a user
Email sent to user with a link to set password
User sets a new password for the account
User is redirected to login page

Documentation of Changes in Iteration 3

Implement user sessions and protect auth routes

a. Added @login_required and @admin_required decorators to prevent users from accessing protected routes without logging in.
b. Added unit test cases and behave tests to validate this.
c. Earlier all routes were unprotected, now we have decorators to protect routes with authentication.

Unfinished Stories

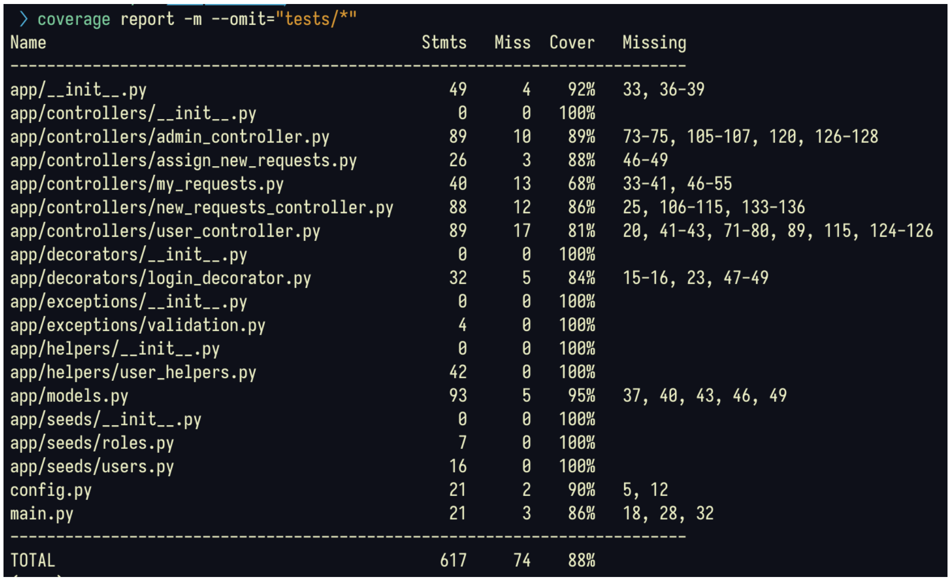
We have implemented and tested all the user stories we planned for iteration 3.

Evaluations of Code and Test Quality

We have Pylint for static code analysis and autopep8 for automatically formatting the code following the pep8 standards.

Pylint Code Rating: 7.23/10

Coverage Report

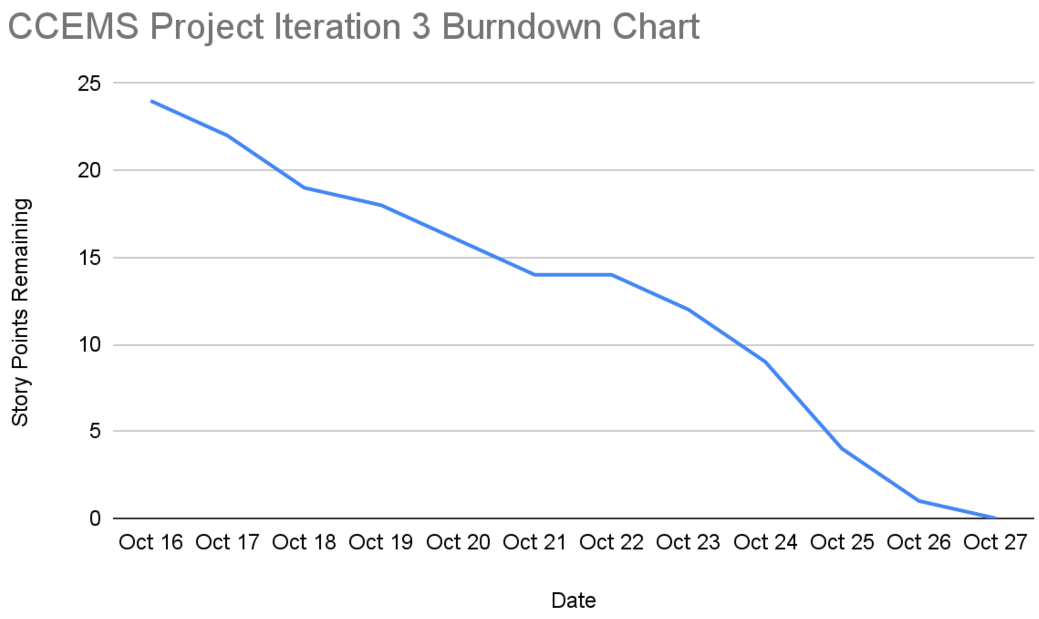


```
❯ coverage report -m --omit="tests/*"
Name                                          Stmts   Miss  Cover   Missing
--------------------------------------------------------------------------
app/__init__.py                                  49      4    92%   33, 36-39
app/controllers/__init__.py                       0      0   100%
app/controllers/admin_controller.py              89     10    89%   73-75, 105-107, 120, 126-128
app/controllers/assign_new_requests.py           26      3    88%   46-49
app/controllers/my_requests.py                   40     13    68%   33-41, 46-55
app/controllers/new_requests_controller.py       88     12    86%   25, 106-115, 133-136
app/controllers/user_controller.py               89     17    81%   20, 41-43, 71-80, 89, 115, 124-126
app/decorators/__init__.py                        0      0   100%
app/decorators/login_decorator.py                32      5    84%   15-16, 23, 47-49
app/exceptions/__init__.py                        0      0   100%
app/exceptions/validation.py                      4      0   100%
app/helpers/__init__.py                           0      0   100%
app/helpers/user_helpers.py                      42      0   100%
app/models.py                                    93      5    95%   37, 40, 43, 46, 49
app/seeds/__init__.py                             0      0   100%
app/seeds/roles.py                                7      0   100%
app/seeds/users.py                               16      0   100%
config.py                                        21      2    90%   5, 12
main.py                                          21      3    86%   18, 28, 32
--------------------------------------------------------------------------
TOTAL                                           617     74    88%
```

Links

Project Management Tool:
https://github.com/orgs/https-github-com-ahphand-tamu/projects/3
Repo Link:
https://github.com/https-github-com-ahphand-tamu/CCEEMS
Deployed App:
https://childcare-d71b0285d615.herokuapp.com/

Burndown Chart



Effort Score and Table

	Atharva Hemant Phand	Bharath Kumar Ravichandran	Chidambaram Ganesan	Daehee Han	Noah Pang	Tanmai Harish
Story Points	4	4	4	3	4	5
Score	92.30769231	92.30769231	92.30769231	69.23076923	92.30769231	115.3846154

BDD & TDD

Behave Tests Output for all features -

```
 ❯ behave
[19:03:15]
[2023-10-27 19:03:15,491] INFO in _init_: Application environment: test
[2023-10-27 19:03:15,960] INFO in environment: ───── Feature: Admin Controller ─────
[2023-10-27 19:03:15,960] INFO in admin_controller_service: Seeding database in admin controller
[2023-10-27 19:03:16,514] INFO in user_controller: User with email: test1@tamu.edu and role:
Admin is successfully logged in!
[2023-10-27 19:03:16,516] INFO in admin_controller_service: Response: 302
@admin-controller
Feature: Admin Controller # features/admin_controller.feature:2

  Scenario: Get a list of users              # features/admin_controller.feature:4
    Given the application is running         # features/steps/admin_controller_steps.py:14
0.000s
    When I access the "/admin/users" endpoint # features/steps/admin_controller_steps.py:19
0.077s
    Then I should see user details           # features/steps/admin_controller_steps.py:25
0.001s

  Scenario: Get a user by ID                   # features/admin_controller.feature:9
    Given the application is running           # features/steps/admin_controller_steps.py:14
0.000s
    When I access the "/admin/users/2" endpoint # features/steps/admin_controller_steps.py:19
0.003s
    Then I should see that user's details      # features/steps/admin_controller_steps.py:58
0.000s

  Scenario: Add a new user                                  #
features/admin_controller.feature:14
    Given the application is running                        #
features/steps/admin_controller_steps.py:14 0.000s
    When I send a POST request to "/admin/users" with JSON    #
features/steps/admin_controller_steps.py:67 0.620s
      """
      {
        "name": "John Doe",
        "email": "johndoe@random.com",
        "role": "Admin"
      }
      """
    Then the response status code should be 201              #
features/steps/admin_controller_steps.py:78 0.000s
    And the response should contain "User added successfully" #
features/steps/admin_controller_steps.py:83 0.000s

  Scenario: Update an existing user                           #
features/admin_controller.feature:27
    Given the application is running                          #
features/steps/admin_controller_steps.py:14 0.000s
```

```
    When I send a PUT request to "/admin/users/2" with JSON      #
features/steps/admin_controller_steps.py:89 0.020s
      """
      {
        "name": "Updated John Doe",
        "email": "updatedjohndoe@example.com",
        "role": "Senior Leader"
      }
      """
    Then the response status code should be 200                  #
features/steps/admin_controller_steps.py:78 0.000s
    And the response should contain "User updated successfully" #
features/steps/admin_controller_steps.py:83 0.000s

  Scenario: Delete an user that does not exist         # features/admin_controller.feature:40
    Given the application is running                   #
features/steps/admin_controller_steps.py:14 0.000s
    When I send a DELETE request to "/admin/users/100" #
features/steps/admin_controller_steps.py:97 0.008s
    Then the response status code should be 404        #
features/steps/admin_controller_steps.py:78 0.000s
    Then the response should contain "User not found"  #
features/steps/admin_controller_steps.py:83 0.000s
[2023-10-27 19:03:17,256] INFO in environment: ───── Feature executed: Admin Controller ─────
[2023-10-27 19:03:17,257] INFO in admin_controller_service: Dropping all tables in admin
controller
[2023-10-27 19:03:17,271] INFO in environment: Dropping all tables

[2023-10-27 19:03:17,276] INFO in environment: ───── Feature: Assign Requests Controller ─────
Feature: Assign Requests Controller # features/assign_requests_controller.feature:1

  Scenario: Setup DB  # features/assign_requests_controller.feature:2
    Given create db   # features/steps/assign_requests_controller_steps.py:53 0.303s

  Scenario: View List of New Requests                                                       #
features/assign_requests_controller.feature:4
    Given I am on the "new-requests" page                                                   #
features/steps/assign_requests_controller_steps.py:14 0.000s
    When I view the list of new requests                                                    #
features/steps/assign_requests_controller_steps.py:18 0.005s
    Then I should see a list of new requests with customer IDs, first names, and last names #
features/steps/assign_requests_controller_steps.py:22 0.000s

  Scenario: Assign a New Request                       #
features/assign_requests_controller.feature:9
    Given I am on the "new-requests" page              #
features/steps/assign_requests_controller_steps.py:14 0.000s
    When I click the "Assign" button for a new request #
features/steps/assign_requests_controller_steps.py:32 0.016s
    And I select a user from the dropdown              #
features/steps/assign_requests_controller_steps.py:40 0.000s
    And I submit the assignment form                   #
features/steps/assign_requests_controller_steps.py:44 0.019s
```

```
    Then I should see a success message                 #
features/steps/assign_requests_controller_steps.py:48 0.000s

  Scenario: Purge DB  # features/assign_requests_controller.feature:16
    Given purge db    # features/steps/assign_requests_controller_steps.py:77 0.015s
[2023-10-27 19:03:17,637] INFO in environment: ----- Feature executed: Assign Requests Controller
-----
[2023-10-27 19:03:17,637] INFO in environment: Dropping all tables

[2023-10-27 19:03:17,640] INFO in environment: ----- Feature: New Requests Controller -----
Feature: New Requests Controller # features/new_requests_controller.feature:1

  Scenario: Setup DB  # features/new_requests_controller.feature:2
    Given create db   # features/steps/assign_requests_controller_steps.py:53 0.277s

  Scenario: Upload new requests in CSV format
# features/new_requests_controller.feature:4
    Given the application is running
# features/steps/admin_controller_steps.py:14 0.000s
    When I access the "/upload-new-requests" endpoint
# features/steps/admin_controller_steps.py:19 0.007s
    And I attach a file named "testfiles_invalid_csv.csv"
# features/steps/new_requests_controller_steps.py:19 0.000s
    And I submit the form
# features/steps/new_requests_controller_steps.py:25 0.034s
    Then the rendered html should contain "Emily" in the valid table and "As76" in invalid table
# features/steps/new_requests_controller_steps.py:34 0.000s

  Scenario: Upload new requests in Excel format
# features/new_requests_controller.feature:11
    Given the application is running
# features/steps/admin_controller_steps.py:14 0.000s
    When I access the "/upload-new-requests" endpoint
# features/steps/admin_controller_steps.py:19 0.015s
    And I attach a file named "sample_new_cases.xlsx"
# features/steps/new_requests_controller_steps.py:19 0.000s
    And I submit the form
# features/steps/new_requests_controller_steps.py:25 0.089s
    Then the rendered html should contain "Grace" in the valid table and "NONE" in invalid table
# features/steps/new_requests_controller_steps.py:34 0.000s

  Scenario: Purge DB  # features/new_requests_controller.feature:17
    Given purge db    # features/steps/assign_requests_controller_steps.py:77 0.011s
[2023-10-27 19:03:18,075] INFO in environment: ----- Feature executed: New Requests Controller
-----
[2023-10-27 19:03:18,075] INFO in environment: Dropping all tables

[2023-10-27 19:03:18,079] INFO in environment: ----- Feature: Set Password -----
[2023-10-27 19:03:18,079] INFO in update_password_service: Seeding database for update password
INFO:root:ADMIN ROLE: <Role Admin>
[2023-10-27 19:03:18,484] INFO in user_controller: User with email: test1@tamu.edu and role:
Admin is successfully logged in!
INFO:app:User with email: test1@tamu.edu and role: Admin is successfully logged in!
```

```
[2023-10-27 19:03:18,485] INFO in update_password_service: Login Response: 302
INFO:app:Login Response: 302
[2023-10-27 19:03:19,204] INFO in update_password_service: Create User Response: 201
INFO:app:Create User Response: 201
@update-password
Feature: Set Password # features/update_password.feature:2

  Scenario: Update password for a new User
# features/update_password.feature:3
    Given the application is running
# features/steps/admin_controller_steps.py:14 0.000s
    When I access endpoint "/users/updatePassword?email=test3@tamu.edu" and enter a valid email
"test3@tamu.edu"("Test User3"), password "abcd@1234567" and re-password "abcd@1234567" #
features/steps/admin_controller_steps.py:103 0.001s
    Then the Login Page should be displayed
# features/steps/admin_controller_steps.py:117 0.000s

  Scenario: Email or Password not present in POST request  # features/update_password.feature:7
    Given the application is running                       #
features/steps/admin_controller_steps.py:14 0.000s
    When I access the endpoint "/user/login" with JSON     #
features/steps/admin_controller_steps.py:129 0.001s
      """
        {
          "name": "John Doe"
        }
      """
    Then the login response status code should be 400      #
features/steps/admin_controller_steps.py:134 0.000s
[2023-10-27 19:03:19,209] INFO in environment: ----- Feature executed: Set Password -----
INFO:app:----- Feature executed: Set Password -----
[2023-10-27 19:03:19,209] INFO in update_password_service: Dropping all tables in admin
controller
INFO:app:Dropping all tables in admin controller
[2023-10-27 19:03:19,219] INFO in environment: Dropping all tables
INFO:app:Dropping all tables

4 features passed, 0 failed, 0 skipped
15 scenarios passed, 0 failed, 0 skipped
46 steps passed, 0 failed, 0 skipped, 0 undefined
Took 0m1.525s
```