System and Unit Test Report

Njoy
03/06/21

System Test Scenarios

Sprint 1)

A) As a user, I want to be able to sign up and login so that I can have access to my own personalized schedule

B) As a user, I want to be able see the main page when I login so that I know I have successfully logged in.

Scenario:

1) start Njoy App; Select ‘Don’t have an account? Sign Up”
    - Full Name = <Test>
    - Email Address = <test@test.com>
    - Password = <test123>
    - Confirm Password = <test123>
    - press “Sign Up” button
    - User should see alert that sign up was successful

2) In Sign In page
    - Email Address = <test@test.com>
    - Password <test123>
    - press “Sign In” button
    - User should be routed to activity form page

Sprint 2)

A) As a user, I want to be able to submit an activity text input form and time block (duration) so that my activity can be scheduled

B) As a user, I want to make sure that the duration input is valid so that I will always receive an available time slot upon input.

C) As a user, I want to be able add and delete more activities so that I can update my schedule if needed.

Scenario:

1) Sign in and input activities into fields
    - Activity = “Test Web Application”
    - Duration = “30”
    - Press “Generate Schedule”

2) Press + button to add one more activity and type invalid input to Duration input
    - Activity = “anything”
    - Duration = “two”
    - Duration input field should show red indicating invalid input
    - “Generate Schedule” button should be greyed out
    - Press “Generate Schedule” and nothing happens

3) Click add button or delete button in generate Schedule tab
    - Enter: Activity = “Testing”, Duration = “40
    - Click + button
    - A new text field should appear under the previous text fields
    - Click trash can button next to First test fields
    - The first test field and the input should disappear and the new empty text fields remain

Sprint 3)

A) As a user, I want the text input that is wrong to correctly show red so that I can tell which input is incorrect.

B) As a user, when I submit the activities, I want the data of the generated schedule so that I can see my schedule.

C) As a user, I want to make sure that the activity name and time block I inputted is persistent so that I don't have to resubmit the information.

Scenario:

1) Sign in and input invalid activities into fields
    - Activity = "anything"
    - Duration = "two"
    - The duration field should show red
    - Click + button
    - Enter in new fields
    - Activity = "Something"
    - Duration = "Three"
    - The duration field should show red
    - Change Duration "three" to 3
    - The duration should not be red anymore

2) Input multiple activities and generate schedule
    - Activity = "Testing Web App", Duration = "40"
    - Click + button
    - Activity = "Demo Web App", Duration = "15"
    - Click + button
    - Activity = "Evaluate Web App" Duration = "10"
    - Click Generate Schedule
    - It should route to schedule page and show a randomly generated schedule with time slots and activity names as inputted by user

3) Click add button or delete button in generate Schedule tab
    - Activity = "Testing Web App", Duration = "40"
    - Click + button
    - Activity = "Demo Web App", Duration = "15"
    - Click + button
    - Activity = "Evaluate Web App" Duration = "10"
    - Click Generate Schedule
    - It should route to schedule page; record times and activities
    - Logout and log back in

  - Click on the view schedule button and it should show the same times and activities.

Sprint 4)

A) As a user, I want to be able to drag and drop a generated activity into a different time slot in case something comes up and I need to change the time of an activity.

B) As a user, I want to be able to add more activities to my schedule after I’ve seen it so that my schedule is flexible

C) As a user, I would like a visually pleasing interface.

Scenario:

1) Create a schedule, go to view schedule, drag and drop an activity
   - Activity = “Testing Web App”, Duration = “40”
   - Click + button
   - Activity = “Demo Web App”, Duration = “15”
   - Click + button
   - Activity = “Evaluate Web App” Duration = “10”
   - Click Generate Schedule
   - Click View schedule
   - Drag drop “Testing Web App” activity 30 min later than before
   - The element in schedule should reflect changes
   - Refreshing the page should let the element be persistent
2) After creating a schedule add on more activity
   - Activity = “Testing Web App”, Duration = “40”
   - Click + button
   - Activity = “Demo Web App”, Duration = “15”
   - Click Generate Schedule
   - Click Add more Button
   - It should route to generate schedule tab and have previous activities inputted
   - Click + button
   - Activity = “Evaluate Web App” Duration = “10”
   - Click Generate Schedule button
   - It should route to schedule with the newly added activity “Evaluate Web App”
3) Open App and sign up login and view generate schedule tab and view schedule tab
   - The interface should look nice and styled in a pleasant way

Unit Tests

Patrick Phuong

For user authentication APIs, I used Postman to test API endpoints and check the intended output and did manual testing

User authentication APIs

POST /api/users/register
// Endpoint to register user

```
body: {
        email: test@test.com,
        password: test123,
        password: test123
}
```

```
body response: {
        success: true,
        token: Bearer ...
}
```

Manual testing

- Start web app
- Press sign up button
- fill out new account information
- Receive success alert
- Sign in with new account
- Route to homepage

POST /api/users/login
// Endpoint to login user

```
body: {
        name: Patrick,
        email: test3@test.com,
        password: test123,
        password2: test123
}
```

```
body response: {
        _id: autogenerated uid
        name: Patrick,
        email: test3@test.com.
        password: hashed password
        date: date of creation
}
```

Manual testing

- Start web app
- Press sign up button
- fill out new account information
- Receive success alert
- Sign in with new account
- Route to homepage
- Logout
- Sign in and route to homepage

Schedule APIs

POST /api/schedule/generate
// Endpoint to generate a schedule

manual testing

- Sign in
- Fill out activities and durations
- Press generate schedule
- Should route to view schedule page with auto generated schedule

POST /api/schedule/reset
// Endpoint to clear and reset a user’s schedule

manual testing

- Sign in
- Fill out activities and durations
- Press generate schedule
- Route to view schedule with auto generated schedule
- Press ‘Reset Schedule’
- Should see activities in view schedule disappear

- Can go to any other tab or log out and go back to view schedule to see clear/reset schedule

GET /api/schedule/getSchedule
// Endpoint to retrieve a user’s schedule if it exists

manual testing

- Sign in
- Fill out activities and durations
- Press generate schedule
- Route to view schedule with auto generated schedule
- Go to any other tab or log out and go back to view schedule to see the saved schedule

POST /api/schedule/setSchedule
// Endpoint to set updated schedule with new activity interval from drag and drop

manual testing

- Sign in
- Fill out activities and durations
- Press generate schedule
- Route to view schedule with auto generated schedule
- Drag and drop any activity
- Activity should drop where user wants to drop it
- Go to any other tab or log out and go back to view schedule to see the the new updated schedule with the new time

Shingo Nakagaki

SignUp/SignIn UI

User Input Sign-up

Check to see if the fields actually save state when user enters input

manual testing

- Add div component with values of each input states
- Open Web App
- Fill out inputs of Name, email, password and confirm password
- As filling out inputs the div component should reflect these changes
- Input should match each saved state

User input Sign-in

Check to see if the fields actually save state when user enters input

manual testing

- Add div component with values of each input states
- Open Web App
- Fill out inputs of Email, and password
- As filling out inputs the div component should reflect these changes
- Input should match each saved state

Activity Form UI

User Input Activity Form

Check to see if the fields actually save state when user enters input

manual testing

- Add div component with values of each input states
- Open Web App
- Fill out inputs of Activity, Duration.
- As filling out inputs the div component should reflect these changes
- Input should match each saved state

Add button for Activity form

Check to see if the add button creates a new component with two input fields: Activity and Duration.

manual testing

- Open Web App
- Click Sign in and click on Add button on bellow the inputs
- If the new component appears under the first one it works.
- Try it again a few times to see if it continues to work.

delete button for Activity form

Check to see if the delete button deletes a component with two input fields: Activity and Duration.

manual testing

- Open Web App
- Click Sign in and click on Add button on bellow the inputs
- Then click on one of the delete button to see if it deletes the component
- Try it again a few times to see if it continues to work.

Schedule UI

Displaying schedule

Check to see if inputs will appear in scheduler

manual testing

- Manually input schedule appointments into scheduler component
- Open Web App
- Sign in and open scheduler view
- If the manual imputed schedule appears it works

Drag and Drop

Check to see if you can drag and drop appointments in scheduler

manual testing

- Open Web App
- Sign in and input a few activities and generate a schedule
- Drag and drop one of the appointments to a different location
- If it successfully is picked up and moved to a different location it worked

Matthew Ngor

Redux Action Handler/Dispatching to Reducer

- Print statements to ensure the program calls flow as intended
- E.g. ensuring the api gets called from the redux action handler → dispatch to reducer → reducer correspondingly updates the global store
- Ensure that the data object is handled correctly

User Authentication

- Create account (email + password)
- Sign in with just created credentials to ensure it works
- Invalid credentials would prevent signing into the application

Schedule Algorithm/UI Testing

- Test for valid duration fields
- Test for occurrences of the trash icon UI bug by adjusting browser size in between routing of pages like view schedule, generate schedule, etc.

Manual testing for generating a schedule

- Sign in to account
- Fill out activities and durations
- Click generate schedule
- Delete 1-2 activities and re-add them to ensure that functionality works as intended
- Ensure that the page routes to the View Schedule page with the auto generated schedule
- Ensure that the activity names and duration are accurate according to the inputs
- Repeat 2-3 times to ensure that the generated schedule is indeed random

Manual testing for resetting a schedule

- Sign in to account
- Fill out activities and durations
- Click generate schedule
- Ensure that the page routes to the View Schedule page with the auto generated schedule
- Click ‘Reset Schedule’ button
- Ensure that the activities in the View Schedule page have disappeared
- Logged out of the application and logged back in to make sure that the schedule data has been cleared and no longer visually appears

Manual testing for adding more activities

- Sign in to account
- Fill out activities and durations
- Click generate schedule
- Ensure that the page routes to the View Schedule page with the auto generated schedule
- Click ‘Add More’ button
- Ensure that the filled in activity and duration information from step 2 is still there
- Ensure that we’re able to delete activities and add more
- Click generate schedule
- The page routes back to the View Schedule page with another auto generated schedule

Manual testing for checking persistent schedule data

- Sign in to account
- Fill out activities and durations
- Click generate schedule
- Ensure that the page routes to the View Schedule page with the auto generated schedule
- Click on the user profile icon and log out
- Sign back into the application
- Click ‘View Schedule’
- Ensure that the page routes to the View Schedule page AND the aforementioned generated schedule still has the same originally inputted information

Manual testing for drag and drop functionality

- Sign in to account
- Fill out activities and durations
- Click generate schedule
- Ensure that the page routes to the View Schedule page with the auto generated schedule
- Hover over an activity and click+drag it to another position in the schedule list
- The time period on the activity block updates accordingly
- Click on the user profile icon and log out
- Sign back into the application
- Click ‘View Schedule’
- Ensure that the page routes to the View Schedule page
- The activity that was dragged and dropped displays the new time period that it was placed under, AND the other activities that were not dragged and dropped are still displayed in their respective positions in the schedule

Sooyoung Kim

SignUp/SignIn UI

1) User Input Sign-up

Check to see if the fields actually save state when user enters input

manual testing

- Add div component with values of each input states
- Open Web App
- Fill out inputs of Name, email, password and confirm password
- As filling out inputs the div component should reflect these changes
- Input should match each saved state

2) User input Sign-in

Check to see if the fields actually save state when user enters input

manual testing

- Add div component with values of each input states
- Open Web App
- Fill out inputs of Email, and password
- As filling out inputs the div component should reflect these changes
- Try typing email with wrong password to see it allows to sign in or not
- Input should match each saved state

Activity Form UI

1) User Input Activity Form

Check to see if the fields actually save state when user enters input

manual testing

- Add div component with values of each input states
- Open Web App
- Fill out inputs of Activity, Duration.
- As filling out inputs the div component should reflect these changes
- Input should match each saved state

2) Add button for Activity form

Check to see if the add button creates a new component with two input fields: Activity and Duration.

manual testing

- Open Web App
- Click Sign in and click on Add button on bellow the inputs
- If the new component appears under the first one it works.
- Try it again a few times to see if it continues to work.

3) Delete button for Activity form

Check to see if the delete button deletes a component with two input fields: Activity and Duration.

manual testing

- Open Web App
- Click Sign in and click on Add button on bellow the inputs
- Then click on one of the delete button to see if it deletes the component
- When there is only one component check if the delete icon disappear
- Try it again a few times to see if it continues to work.

4) Text field for Activity form

The text fields for duration turn red when the invalid inputs are given.

Manual testing

- Open Web App
- Click Sign in and click on Add button on bellow the inputs
- Type input other than number for duration to check if it turns into error state(red outline)
- Check if generating schedule is denied when there's any error state for text field
- Check if the error state goes back to normal when the input changes to number
- Try with multiple text field components and see if they all work independently.
- Try it again a few times to see if it continues to work.

UI/UX Design

1) Navigation Bar

Check if the Navigation bar is responsive and correctly adjusts the text size or location according to the size of the browser

Manual testing

- Open Web App
- Click Sign in and click on Add button on bellow the inputs

- Resize the browser and see if the Njoy logo is still centered, user icon is aligned right, and View schedule/Generate Schedule stays on the left side of the navigation bar.
- When the size of the browser changes from large to small, “View schedule” and “Generate Schedule” change to “View” and “Generate”
- Try if it still looks the same when the user signs out/ signs in/clicks view schedule / clicks the generate schedule button.

2) Styling Generate schedule / Add more / Reset Buttons

When the user clicks these buttons, they should work as we expected.

Manual testing

- Open Web App
- Click Sign in and click on Add button on bellow the inputs
- Check if all the buttons work as we expected.
- Try if it still works the same when the user signs out/ signs in