whatsRoaring

Product Evaluation

In most cases, our product behaves as intended. We describe the intended and actual behavior of our product in more detail below.

Testing

Throughout the development of our application, we tested the various features. We first tested the filtering functions. In particular, we confirmed that users are able to select multiple filters at once. We would hope that filters within the same dropdown are applied using an OR relation and that filters across diffferent dropdowns or checkboxes are applied using an AND relation. For instance, if a user filters for “Academic” and “Cultural” events concurrently, our calendar should output events that fall under ≥ 1 of those categories. In contrast, if a user filters for both organization “PUFSC” and his/her favorite events, then the output should include events that are both associated with “PUFSC” and in his/her favorites. The same relationship extends to more complicated filters. Testing our program with several filtering combinations against our desired output revealed that it behaves as expected.

Next, we tested the different states of our program. In particular, we ensured that the desired buttons appeared in specific situations. First, we verified that the “+ Event” and “+ Organization” buttons only appear for users who are admins. Similarly, we verified that the “Request Admin Access” button only appears for users who are not admins. Next, we confirmed that the edit and delete buttons do not appear on most of the details pages. Specifically, they only appear on the details pages for events created by the user. Our program behaves correctly.

We also performed tests on the fields of our forms. In particular, we tested the behavior of our program with missing fields. We ensured that users cannot submit forms with missing fields by displaying a dialog box requesting that they input the required fields. Additionally, we forbid users from submitting forms with fields whose length is > *X* (where *X* varies depending on the form and field) characters by displaying a dialog box requesting the user to shorten the name. We also do not allow users to submit dates with years < 1000 and > 9999 to ensure that all years have four digits.

Additionally, we performed stress tests on our application. First, we wrote a script that added 1000 events with the same start date/time (February 21, 2019 at 10am) and different names to our calendar in immediate succession. While the day and week views of our calendar for February 21, 2019 were not particularly readable afterwards, our application did not crash and successfully displayed all of the events. We then wrote a script to delete all of these events. We also tested the behavior of the program with a user who presses “Submit” many times in succession on the “Create Event,” “Edit Event,” or “Create Organization” forms or “Register” on the “Register” form. The user does not receive duplicate event/organization/user notifications because our program redirects the user immediately after the buttons are originally pressed. Furthermore, we tested the behavior of our application upon for submissions with odd fields, such as “if old_name != name or old_starttime != start_datetime: !@#$%^&*()_åß∂ƒ©˙∆˚¬…Ω≈ç√∫˜µ≤≥¡™£¢∞§¶•ªº”. If all of the fields met the rest of our requirements, the forms successfully submitted.

We tested our program in different timezones by changing the timezone settings on our computers. In particular, we ensured that events are displayed on our calendar and in details pages in the user’s current timezone. We also ensured that events export to Google Calendar in the user’s current timezone and that the “Edit Event” form pre-populates with information reflecting the user’s current timezone. Furthermore, we ensured that when users create events, they are created in the user’s current timezone. We also verified that our program detects duplicate events and that the “Today” button works as expected regardless of the user’s current timezone.

Furthermore, we tested the behavior of our application upon navigating to different pages by changing the url. Specifically, we ensured that if a user is not logged in, entering any valid url redirects him/her to the landing page. We also confirmed that if a user who is not an admin enters the /createEvent or /addOrg urls, he/she is redirected to the calendar page. Additionally, if a user enters the /editEvent or /details url without an event name, he/she is also redirected to the calendar page.

We tested our application in Firefox, Google Chrome, Safari, and Microsoft Edge.

Robust Features

Our most robust features include (1) filtering, (2) log in, (3) creating and editing an event, (4) creating an organization, and (5) creating a new user. In particular, we thoroughly tested our program's filtering capabilities (as described in the "Testing" section) to ensure that our application supports any combination of filters. If a user attempts to log in with an incorrect email/password pair, then our program displays an "Incorrect email/password" dialog box. We thoroughly validate events during creation/editing. Specifically, we display a dialog box with an appropriate message if the form is missing ≥ 1 field (with a list of the missing fields), if any field is too long (with a list of the relevant fields), if the end date/time is after the start date/time, if the start date/time is in the past, if the year does not have four digits, or if a duplicate event already exists in the calendar. Additionally, we highlight any text field containing input that is too long in red. We also validate organizations when they are created by displaying a dialog box with an appropriate message if the form is missing ≥ 1 field (with a list of the missing fields), if any field is too long (with a list of the relevant fields), or if a duplicate organization already exists. Last, we validate the registration form. We display an appropriate message in a dialog box if the form is missing ≥ 1 field (with a list of the missing fields), or if any field is too long (with a list of the relevant fields), or if a duplicate user exists. We also inform users of whether or not their password is sufficiently strong (≥ 10 characters).

Fragile Features

Our most fragile features include (1) logging into two accounts concurrently, (2) refreshing the details and edit event pages, and (3) originally visiting the site. Specifically, our application does not always support multiple users concurrently on the same browser. If a student is logged in as User 1 and then attempts to login as User 2, our application should log the student out of User 1. However, after logging into User 2, if the student opens a new tab, it is sometimes logged into User 1. Additionally, because React erases the current state when a user refreshes the page, we automatically redirect users to the calendar page if he/she refreshes the "Details" or "Edit Event" page because the event name is lost. Last, if we have not used WhatsRoaring for a few hours, we might need to refresh the page a few times before we can load the landing page. We believe that this is because Heroku depriotizes our application because it is free and low traffic.

Known Bugs & Limitations

We have a few known bugs and limitations. First, a user who changes the timezone on his/her computer must refresh the page to display the calendar events or details in the new timezone. Additionally, users must wait for the calendar to populate with events before applying filters. Otherwise, the calendar will not display the output correctly.

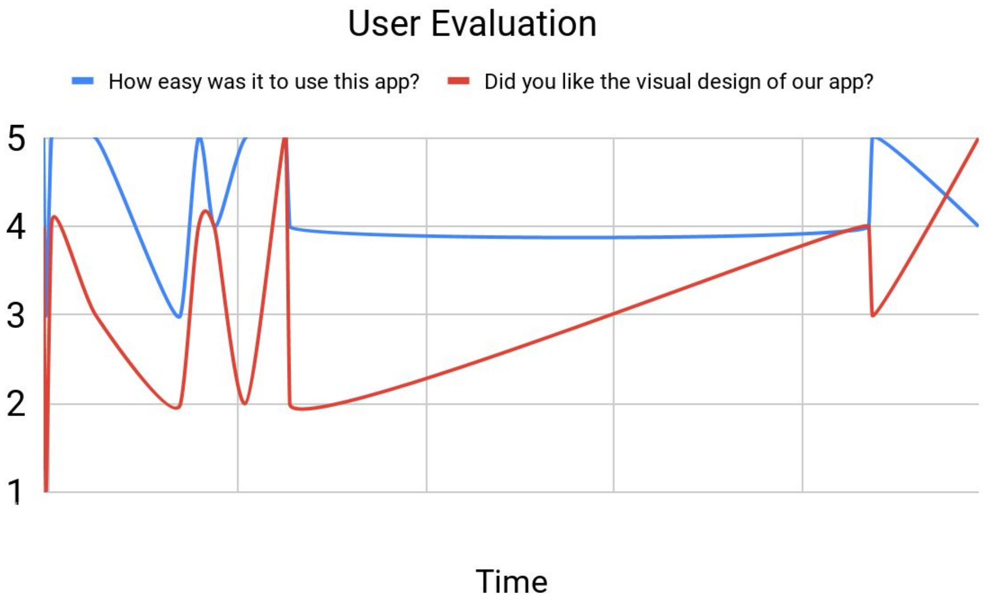
Formal User Evaluation

We walked three users, Helena, Kayli, and Ivy, through a set of tasks detailed in the Appendix. Our notes from the observations are also detailed in the Appendix. In general, our users thought that the login page was intuitive; the only request was to allow users to register by pressing “Enter.” Our users all noted that they found filtering useful and that the toolbar was easy to find. Though all of them easily navigated to the details page upon request, Ivy noted that we could make it more obvious that clicking on an event leads to a new page. They all commented on the usefulness and ease of favoriting, and Ivy was especially excited about Google Calendar export. Each user found the Create Event form fairly intuitive, though Helena noted that it would be nice for the calendar to automatically pop up for the start date/time and end date/time fields. Ivy also mentioned that she thought it would be helpful to add a feature that allows for recurring events. While Helena said that she liked the pop up confirming deletion, Ivy said that she thought it could get annoying. All of our users were glad that it was only possible to edit/delete events created by the specified user.

We intend to incorporate many of these suggested features into our product in the future, such as allowing users to use the “Enter” key to submit forms, and opening the calendar selection as soon as the user navigates to the start date/time and end date/time fields. However, we plan to keep the dialog box confirming deletion because it is used to prevent errors, as described in User Control and Freedom.

Additionally, in an effort to collect a large assortment of user feedback, each of us publicized our application with a Google feedback form on various platforms including email listservs and Facebook.

The overwhelming response was positive. In particular, everyone who filled out our feedback form said that they liked our application and that they would recommend it to a friend. All of our users said that they would use our app at some point, and more than 60% of them said that they would use it at least weekly. We also asked our users how easy it was to use our app on a scale of 1 to 5 (with 5 being the easiest); 54% of users selected 5, 31% of users selected 4, 15% of users selected 3, and 0% of users selected 2 or 1. Additionally, we asked our users if they liked the visual design of our application on a scale of 1 to 5 (with 5 indicating that they liked it the most); about 70% of users selected 3 or above. It is important to note that as illustrated in the figure below, user feedback became more positive with time. We suspect that this is because we implemented many of our users’ suggestions, as detailed below.



Our users had a lot of feedback that we implemented upon their requests. Specifically, one user complained that he/she did not get a page confirming the submission of his/her event, so we added a dialog box indicating success. Another user from Sweden pointed out that his events were added at the incorrect times due to the time difference, so we worked on standardizing timing in different locations. Someone else pointed out that there was a bug with our “see more events” button on the monthly calendar view for crowded days, so we fixed that. Another user successfully added a duplicate event and suggested that we prevent this from happening, so we started checking for duplicate events and organizations. A few users said that they wanted to see weekly and daily views, so we incorporated them into our display. Someone else complained that he/she received no confirmation after favoriting an event, so we added a dialog box indicating success and changed the logo from an open star to a filled star upon favoriting an event. Last, many users expressed how useful it would be for them to export our events directly to their Google Calendars, so we added this feature.

Our users suggested a few features that we hope to implement in the future. For example, one user suggested that we allow users to RSVP to events and that we notify admins of their response. Someone else recommended that we enable users to search for events by name. Another user indicated that he/she would appreciate a feature to sign up for regular email updates about events from a particular organization. We appreciate all of the feedback we have received, and because we intend for this application to live beyond the scope of this course, we plan to incorporate their suggestions after this project is due.

Heuristic Evaluation

(1) Visibility of System Status

Our application successfully keeps users informed about what is going on with appropriate feedback in reasonable time. First, we use a number of dialog boxes to communicate with the user. In particular, we display dialog boxes upon successful registration, event creation, event editing, organization creation, favoriting, and unfavoriting. We also display dialog boxes with descriptive errors if any of these actions or user login fails. For example, if the user attempts to create an event in the past, we display a dialog box saying “Invalid Times: Start date/time should be after the current time.” Additionally, when a user favorites an event, the star icon on the details page changes from open to filled, and when a user unfavorites an event, it changes from filled to open. Furthermore, when the user initially logs in and is directed to the calendar page, if it takes a while to load, it displays a “Loading…” message.

(2) Match Between System and the Real World

In general, our system speaks the user’s language and follows real-world conventions. Based on our evaluation and user feedback, we do not use any words that are unfamiliar to users. For example, our details page, which displays event fields, uses very common terms like “Time,” “Location,” “Organization,” and “Event Type.” Our application also utilizes icons that are familiar to most users, such as a star for favoriting, a calendar for Google Calendar export, a pencil for edits, and a trash can for deletions.

Our calendar is displayed in a format similar to Google Calendar to make it more intuitive for users. In particular, the way users toggle between month, week, and day views and jump to “Today” is similar to Google Calendar. We also followed Google Calendar conventions regarding timezones. Specifically, our calendar’s events are displayed at times that reflect the user’s current timezone, and when users create or edit events, the events are added at times that reflect the user’s current timezone. While this can be confusing, Google Calendar uses the same conventions, so it is our hope that users are more familiar with the way our system operates.

Furthermore, our informations appears in a logical order. Users are first shown a landing page to register or log in, then directed to the main calendar page. The calendar page contains the functionality of our application that does not pertain to specific events, such as viewing events in different formats and filtering. Users can also click on events for details on the main calendar page. The details page houses functionality for specific events, such as favoriting and exporting, which seems fairly intuitive.

(3) User Control and Freedom

Our application has many reverse actions to support users who choose system functions by mistake. For example, any user who creates an event is able to both edit and delete the event afterwards, in case he/she added it by mistake or entered a detail incorrectly. Additionally, users can easily undo favoriting and unfavoriting by performing the reverse function. Furthermore, users must consciously save events to their Google Calendars when they press the export button to prevent them from accidentally adding events they are not interested in to their personal calendars. Last, if a user tries to remove an event, he/she is shown a dialog box confirming his/her action to prevent users from accidentally deleting events.

(4) Consistency and Standards

Our applications makes use of many conventions so users do not have to wonder whether different words, situations, or actions share the same meanings. For instance, each of our event fields (except category and event type, which are naturally correlated) is consistent across our different pages — the main calendar page, the details page, the create event page, the edit event page, and the create organization page. Additionally, all of the details pages look identical in that the information is displayed in the same format and the favoriting, exporting, and back functions look the same and are in the same locations. Furthermore, the create event and edit event forms have the same format for consistency; the only difference between the two is that the edit event form is pre-populated with the event’s current details. Last, we use the same logo, font, and colors throughout our site for consistency.

(5) Error Prevention

We implemented many functions to avoid errors. To evade database errors, we forbid users from entering events and organizations with missing or invalid fields by displaying appropriate dialog boxes. For example, if a user attempts to create an event without a name, we display a dialog box requesting that he/she add one. We also do not allow users to add duplicate events or organizations by displaying an appropriate dialog box because this would cause problems retrieving information later. To prevent server errors, if a user manually enters a valid url in his/her browser that he/she does not have access to (such as the calendar page if he/she is not logged in), then we redirect the user to a page that he/she does have access to (as explained in Testing). Similarly, refreshing the details page redirects users to the calendar page because doing so erases the current state, and we want to avoid a details page with any unknown fields. (Erasing the current state is a product of using React, as described in Fragile Features.)

(6) Recognition Rather Than Recall

We designed our application to minimize the user's memory load. For instance, it is easy for users to discern the current calendar state. The dropdown that allows users to select between month, week, and day views displays the state that the calendar is currently in. Similarly, the current month (in month view), week (in week view), or day (in day view) is visible. With regards to filtering, each dropdown menu displays how many of its options have been selected (if applicable), and users can click on each dropdown to view which options have been selected. Also, the favorites and free events checkboxes illustrates whether or not their respective filters are currently applied.

Our details page displays the name of the event of interest at the top of the page so users can easily recognize the relevant event. Additionally, each of our forms — registration, create event, edit event, and create organization – display their respective names at the top so users can easily identify which page they are viewing. All through our application, all of our functions are easily visible to users so they do not have to recall the actions that they are able to perform.

(7) Flexibility and Efficiency of Use

Our application does not support accelerators; both inexperienced and experienced users experience the same views and functionality. The only difference is that admin users (who are presumably more experienced) have the ability to create and edit events and organizations, while non-admin users do not. Based on user feedback, we are under the impression that our application is intuitive enough for new users, so we do not have to alter it to fit their needs, and that it comfortably supports the needs of more experienced users.

(8) Aesthetic and Minimalist Design

Each of our dialogues contains information which is relevant and often needed. We do not have any components without core functionality, and there is no extra text on our site. Each of our communicative dialog boxes were either designed to prevent errors or were requested by users. Additionally, we do not overcrowd the calendar in the month view because we include an option to see more events. Based on feedback from users, it seems as though we successfully designed an interface that is both minimalist and intuitive.

(9) Help Users Recognize, Diagnose, and Recover From Errors

We designed our application to precisely indicate errors. Specifically, if a user enters an invalid field on any of our forms, we display a dialog box indicating the specific nature of the error. For instance, if a user tries to submit a blank create event form, then the dialog box says “The following fields are missing: event name, description, location, start time, end time, organization, category, is free.” Furthermore, if a user enters a form field that is too long, then the text box changes from orange to red to indicate that he/she should shorten the input. We also display an appropriate dialog box saying “Incorrect Login: Incorrect email/password” if a user attempts to log in with an invalid username/password pair. With these communicative error messages, users can appropriately alter their problematic requests. The users that have tested our system have not indicated that they were unable to recognize or recover from an error.

(10) Help and Documentation

We have attached our documentation to our landing page on our website to assist any interested user. We believe that the user’s guide will be the most informative for users.

Cognitive Walkthrough

(1) Scenario: A New User is Interested in Learning About and Exporting the Media Arts Show on 1/21/2019

Once on http://whatsroaring.herokuapp.com/, the user must register.

1. **Will the correct action be sufficiently evident to the user? Will the user know what to do to achieve the task?**
   There is a “Register” button underneath text that reads “Need an account?” on the landing page, so the user knows what to do to achieve this task.

2. **Will the user notice that the correct action is available? Can users see the button or menu item that they should use for the next action?**
   Yes, the button is readily available to the user.

3. **Will the user associate and interpret the response from the action correctly? Will users know from the feedback that they have made the correct or incorrect choice of action?**

   The user is taken to a “Register” form, so he/she knows that he/she made the correct choice of action. If the user made an incorrect choice of action, such as attempting to log in, he/she will not be able to, and will therefore realize that he/she made a mistake.

The user must fill out and submit the Register form.

1. **Will the correct action be sufficiently evident to the user? Will the user know what to do to achieve the task?**

   The user must fill out the form and press the submit button, which is evident to the user.

2. **Will the user notice that the correct action is available? Can users see the button or menu item that they should use for the next action?**

   Yes, all of the text fields and buttons are available.

3. **Will the user associate and interpret the response from the action correctly? Will users know from the feedback that they have made the correct or incorrect choice of action?**

   Yes. If a user makes an error on the form, then pressing submit will display a dialog box indicating the error. If the user successfully fills out the form and presses submit, a dialog box indicating success will appear.

The user must log in.

1. **Will the correct action be sufficiently evident to the user? Will the user know what to do to achieve the task?**

   The user must enter his/her email and password to log in, which is sufficiently evident.

2. **Will the user notice that the correct action is available? Can users see the button or menu item that they should use for the next action?**

   Yes, both of the text fields and the login button are available.

3. **Will the user associate and interpret the response from the action correctly? Will users know from the feedback that they have made the correct or incorrect choice of action?**

Yes. If a user enters his/her email or password incorrectly, a dialog box will appear indicating the error. Otherwise, the user will be directed to the main calendar page.

The user must navigate to the details page for the Media Arts Show.

1. **Will the correct action be sufficiently evident to the user? Will the user know what to do to achieve the task?**
   To achieve this task, the user must click on the event. While this action is somewhat less intuitive than the prior tasks, it is sufficiently evident that clicking an event takes users to a details page.

2. **Will the user notice that the correct action is available? Can users see the button or menu item that they should use for the next action?**
   Yes, users can see the event, and thus the button to view the event's details.

3. **Will the user associate and interpret the response from the action correctly? Will users know from the feedback that they have made the correct or incorrect choice of action?**
   Yes. if the user successfully navigates to the details page, he/she will see a page with a box titled "Media Arts Show." If the user makes an incorrect choice of action, then he/she will not see the details of this event.

The user must export the Media Arts Show to his/her Google Calendar.

1. **Will the correct action be sufficiently evident to the user? Will the user know what to do to achieve the task?**
   To achieve this task, the user must click on the calendar icon. It is sufficiently evident that clicking on this icon enables calendar export.

2. **Will the user notice that the correct action is available? Can users see the button or menu item that they should use for the next action?**
   Yes, users can see the icon button.

3. **Will the user associate and interpret the response from the action correctly? Will users know from the feedback that they have made the correct or incorrect choice of action?**
   Yes. if the user successfully presses the button to export the event, he/she will be redirected to the Google Calendar page that allows users to create events with all of Media Arts Show details prefilled. If the user makes an incorrect choice of action, then he/she will not be directed to Google Calendar.

(2) Scenario: An Admin User Creates an Event

Once on http://whatsroaring.herokuapp.com/, the user must log in.

1. **Will the correct action be sufficiently evident to the user? Will the user know what to do to achieve the task?**
   The user must enter his/her email and password to log in, which is sufficiently evident.

2. **Will the user notice that the correct action is available? Can users see the button or menu item that they should use for the next action?**
   Yes, both of the text fields and the login button are available.

3. **Will the user associate and interpret the response from the action correctly? Will users know from the feedback that they have made the correct or incorrect choice of action?**
   Yes. If a user enters his/her email or password incorrectly, a dialog box will appear indicating the error. Otherwise, the user will be directed to the main calendar page.

The user must navigate to the create event form.

1. **Will the correct action be sufficiently evident to the user? Will the user know what to do to achieve the task?**
   There is a button that reads "+ Event" on the main calendar page, so the correct action to achieve this task is sufficiently evident to the user.

2. **Will the user notice that the correct action is available? Can users see the button or menu item that they should use for the next action?**
   Yes, the button is readily available.

3. **Will the user associate and interpret the response from the action correctly? Will users know from the feedback that they have made the correct or incorrect choice of action?**
   Yes. If a user makes the correct choice of action, he/she is taken to a "Create Event form." If the user made an incorrect choice of action, he/she will not be able to add an event, and will thus be aware of his/her mistake.

The user must fill out and submit the create event form.

1. **Will the correct action be sufficiently evident to the user? Will the user know what to do to achieve the task?**
   The user must fill out the form and press the submit button, which is evident to him/her.

2. **Will the user notice that the correct action is available? Can users see the button or menu item that they should use for the next action?**
   Yes, all of the text fields and buttons are available.

3. **Will the user associate and interpret the response from the action correctly? Will users know from the feedback that they have made the correct or incorrect choice of action?**
   Yes. If a user makes an error on the form, then pressing submit will display a dialog box indicating the error. If the user successfully fills out the form and presses submit, a dialog box indicating success will appear.

Appendix

We composed and presented the following tasks to users. We observed them perform the tasks and took notes on their behavior. The results are detailed in the Formal User Evaluation section.

1. Register as a new user.
2. Log in.
3. View tomorrow in the day view.
4. View only “Arts” and “Athletics” events.
5. Find the location for the PU Band Open Music Rehearsal on January 22, 2019.
6. Favorite Jummah Prayer in Murray-Dodge on January 25, 2019.
7. View your favorites.
8. Unfavorite the event.
9. Click the button you would press if you wanted to add an event. (The new user is not an admin, so he/she cannot add an event. He/she must request admin access.)
10. Log out. (We then logged into an admin account so our user could add an event.)
11. Create a new event.
12. Edit the location of the event.
13. Delete the event.

Our notes from the observations are copied below:

Helena

- “The register button page is intuitive.”
- “It’s easy to navigate between month/week/day view.”
- “I like filtering! I think that will be useful.”
- “I like the dialog boxes confirming that I favorited/unfavorited the event.”
- “I don’t think it is very intuitive that I need to request admin access to create an event.”
- “It would be nice if the calendar opened directly when adding an event.”

Kayli

- “The 10 character password is excessive.”
- It took her a little while to find tomorrow in day view.
- She clearly figured out which fields were required in the Create Event form based on the *.
- “Maybe add a search bar to easily find information about a particular event?”
- “I like that you can filter to see only free events.”
- “I think favoriting is useful.”

Ivy

- “What if I could just push enter to register instead of clicking the button after filling out the form? That would be nice.”
- “The toolbar is obvious because of the different color.”
- “I like the filtering. I guess there are a lot of options, which can be good or bad. It could be overwhelming for the user.”
- “Okay, so I just click the event for details? Oh look, a new page! Is there a way you can tell the user to click the event for details?”
- “The favoriting button is obvious, which is good. I like the popup message.”
- “My event is favorited on the calendar now! Cool, I can see why this would be useful.”
- “OMG the times and stuff are already in the Google Calendar fields!”
- “What about for recurring events? Do I just have to make multiple? You could streamline that.”
- “I like the icons, it’s obvious what they do. I think they should have labels or something though, just in case people don’t know.”
- “I think the popup “Are you sure?” box might get annoying if you want to delete more than one event.”
- “Ah nice, I can only edit and delete my own events.”