Stakeholder Review #2

Testing Documents:

Our website will be Alpha tested using simulated server environment running Python 2.7.11 and Django 1.9 (Beta testing will be done using a proper Apache server). Our testing approach utilizes human testing of the website by members of the project team. Multiple people will go through each test case and check if the functionality is up to expectations and no unforeseen errors occur. Any errors found during testing will be added to the Github Bug tracker for prioritization and fixing. Ian will be testing on Google Chrome (version 50.0.2661.87 m) and Nick will be testing on Internet Explorer 11 (version 11.0.9600.18282). Both testers will be testing on Windows 7.

Testing Schedule:

Day	Task
4/24	Define test cases
4/25	Alpha test 1: Ian Gross, Chrome
4/26	Alpha test 2: Nicholas Smith, Internet Explorer
4/27	Compile alpha test data, update bug tracker

The test cases are defined below:

Test Case 1: Making a new account successfully

Preconditions:

- User is on the internal landing page and is logged in with admin credentials

Sequence of actions:

- The User enters their information into each

Postconditions:

- The User successfully creates a new account
- Text on the bottom of the page indicating that a new user has been successfully created
- The User is taken back to the login page

Test Case 2: Logging in

Preconditions:

- User has an account already and is on the login page

Sequence of actions:

- The User enters their username in the “username” field
- The User enters their password in the “password” field
- The User clicks on the “log in” button

Postconditions:

- The User successfully logs in and displays the main page

Test Case 3: Attempting incorrect user login

Preconditions:

- User has an account already (or doesn’t have an account) and is on the login page

Sequence of actions:

- The User enters an incorrect username in the “username” field
- The User enters an incorrect password in the “password” field
- The User clicks on the “log in” button

Postconditions:

- The User returns back to the login page
- The bottom of the of the page says: “Invalid username or password. Please try again.”

Test Case 4: Logging out of account

Preconditions:

- User has already logged in to his/her account and is on the landing page, or any main page

Sequence of actions:

- User clicks on the ‘Logout’ button

Postconditions:

- User is taken to the login page

Test Case 5: Adding an Event to the Calendar

Preconditions:

- User has already logged in to his/her account and is on the landing page

Sequence of actions:

- User clicks on the “Calendar/Bulletin Board” button in the navigation bar

- User clicks the “+” button in the bottom right hand corner of the calendar
    - User should be redirected to the website’s google calendar page
    - User adds an event to the google calendar
    - User returns to and reloads the frat app calendar page

Postconditions:

- The new event should be displayed on the calendar upon page reload

Test Case 6: Removing an Event from the Calendar

Preconditions:

- User has already logged in to his/her account and is on the landing page
- Calendar currently has >0 events

Sequence of actions:

- User clicks on the “Calendar/Bulletin Board” button in the navigation bar
- User clicks the “+” button in the bottom right hand corner of the calendar
    - User should be redirected to the website’s google calendar page
    - User removes an event from the google calendar
    - User returns to and reloads the frat app calendar page

Postconditions:

- The event should no longer be displayed on the calendar upon page reload

Test Case 7: Adding an Announcement to the Bulletin

Preconditions:

- User has already logged in to his/her Admin account and is on the landing page

Sequence of actions:

- User clicks on the “Calendar/Bulletin Board” button in the navigation bar
- User fills out the announcement form, including: “Announcement”, “Description”, and “Expiration Date”
- User clicks on the “Add Announcement” button

Postconditions:

- The page should refresh
- The new announcement should display at the top of the page, including: the announcement, the username that posted it, the date and time posted, the description, and the expiration date.

Test Case 8: Creating a task

Preconditions:

- User has already logged in to his/her account and is on the landing page

Sequence of actions:

- The user clicks on the Tasks page
- The user enters the name of the task
- The user clicks on whatever groups or individuals they want to assign the task to
- The user clicks the 'Submit' button

Postconditions:

- Page is refreshed
- A task is added under the "General Tasks" list, "Personal Task" list, or "Tasks You Created" list, depending on whether it was assigned to a group or an individual and who assigned the task to who
- An email will be sent to the user that was assigned a task

Test Case 9: Deleting a Task the User Created For A Group or the User

Preconditions:

- User has logged in and is on the landing page
- User has assigned a task to a group that they are included in or has a task assigned to them by another individual

Sequence of actions:

- The user clicks on the "Tasks/Supplies" button in the navigation bar
- The user clicks on "Mark Complete" button on the task desired to be deleted

Postconditions:

- Task has been removed

Test Case 10: Deleting a Task the User Created For Another User

Preconditions:

- User has logged in and is on the landing page
- User has assigned a task to another user (that is not himself/herself)

Sequence of actions:

- The user clicks on the "Tasks/Supplies" button in the navigation bar
- The user clicks on "Cancel Task" button on the task desired to be deleted

Postconditions:

- Task has been removed

Test Case 11: Adding to the Supply List

Preconditions:

- User has already logged in to his/her account and is on the landing page

Sequence of actions:

- User clicks on the "Tasks" button in the navigation bar
- User navigates to and fills out the "Add to supply list" form with

"Bananas" in the item field and "4" in the quantity field

Postconditions:

- Page should refresh
- The "Supplies Needed" list at the top of the page should be updated with "Bananas: 4" included

Test Case 12: Removing a Quantity of an Item that less than the requested number, from the supply list

Preconditions:

- User has already logged in to his/her account and is on the landing page
- There is at least one item already added to the "Supplies Needed" list that is greater than 2 (example: "Bananas: 4")

Sequence of actions:

- User clicks on the "Tasks" button in the navigation bar
- User navigates to the "Remove from the supply list" form
- User clicks on the supply item in the list box they would like to remove. In this case, click on "Bananas".
- User enters the quantity of the item they would like to remove in the number box titled "Quantity to Remove". In this case, enter "1"

Postconditions:

- Page should refresh
- The Supplies List should be updated with "Bananas: 3" included

Test Case 13: Removing a full item from the Supplies Needed List

Preconditions:

- User has already logged in to his/her account and is on the landing page
- There is at least one item already added to the "Supplies Needed" list (example: "Bananas: 4")

Sequence of actions:

- User clicks on the "Tasks" button in the navigation bar
- User navigates to the "Remove from the supply list" form
- User clicks on the supply item in the list box they would like to remove. In this case, click on "Bananas".
- User enters the quantity of the item they would like to remove in the number box titled "Quantity to Remove". This number should be equal to the quantity needed of the iterm. In this case, enter "4"

Postconditions:

- Page should refresh
- The Supplies List should be updated by removing Bananas from the supplies list.

Test Case 14: Viewing User's Data

Preconditions:

- User has already logged in to his/her account and is on the landing page

Sequence of actions:

- User clicks on the "User Directory" button on the navigation bar
- User clicks on his/her name in the list displayed

Postconditions:

- User is redirected to their info page

Test Case 15: Creating an Attendance List

Preconditions:

- User has already logged in to his/her account and is on the landing page

Sequence of actions:

- User clicks on the "Attendance Lists" button on the navigation bar
- User fills out"Banana Party" in the name field, a description for the event, a location (optional) and an event date (optional).
- User clicks on the "Add Event" button

Postconditions:

- A new attendance list is created named "Banana Party". The attendance list will show the name of the event, the user that created it, the location (if provided) and the event date (if provided).

Test Case 16: Adding an Attendee to the Attendance List

Preconditions:

- User has already logged in to his/her account and is on the "Attendance Lists" page
- User has already created an Attendance List titled: "Banana Party"

Sequence of actions:

- User clicks on "Banana Party" in the list
- User fills in "Bruno" in the "Attendee Name" field
- User clicks the "Add Attendee" button

Postconditions:

- "Bruno" is added to the bottom of the "Banana Party" Attendance List

Case 17: Deleting an Attendee from an Attendance List

Preconditions:

- User has already logged in to his/her account and is on the "Attendance Lists" page
- User has already created an Attendance List titled: "Banana Party"
- "Banana Party" already has an attendee. In this case, we have an Attendee named "Bruno".

Sequence of actions:

- User clicks on "Banana Party" in the list

- User clicks the “Delete” button underneath where “Bruno” is listed in the “Registered Attendees” list

Postconditions:

- The page is refreshed
- “Bruno” is removed from the “Banana Party” Attendance List

Test Case 18: Deleting an Attendance List

Preconditions:

- User has already logged in to his/her account and is on the “Attendance Lists” page
- User has already created an Attendance List titled: “Banana Party”

Sequence of actions:

- User clicks on the “Delete” button under “Banana Party” list

Postconditions:

- “Banana Party” is removed from the Attendance List page

Test Case #	Tester	Passed	Comments
1	Ian	YES	Able to enter some incorrect values
2	Ian	YES	
3	Ian	YES	
4	Ian	YES	
5	Ian	YES	The google calendar display on the Calendar/Bulletin Board page slightly interferes with the side bar interface
6	Ian	YES	
7	Ian	YES	Able to enter an expiration date that has already passed, seems to just never appear once created
8	Ian	YES	
9	Ian	YES	
10	Ian	YES	

11	Ian	YES	Able to enter numbers less than 1 for the "Quantity Needed". This causes the item to not appear on the "Supplies Needed" list Adding an item that already exists on the Supply List causes the number to increase by the quantity requested. Does work with numbers less than 1.
12	Ian	YES	
13	Ian	YES	Entering a number larger than the quantity requested has no impact, but is allowed. If attempting to remove from multiple supply items at the same time by ctrl+clicking on the items in the listbox, it will only remove from the bottom-most item.
14	Ian	YES	
15	Ian	YES	
16	Ian	YES	
17	Ian	YES	
18	Ian	YES	

Test Case #	Tester	Passed	Comments
1	Nick	YES	Format of "Expected Graduation Date" is just a text box, should be a date field. Frequent errors, must be in: YYYY-MM-DD format to work.
2	Nick	YES	
3	Nick	YES	
4	Nick	YES	
5	Nick	YES	
6	Nick	YES	

7	Nick	YES	
8	Nick	YES	
9	Nick	YES	
10	Nick	YES	
11	Nick	YES	
12	Nick	YES	
13	Nick	YES	
14	Nick	YES	
15	Nick	YES	
16	Nick	YES	
17	Nick	YES	
18	Nick	YES	

Code Review:

Team Harmony Review of Due2mr:

Due2mr showed us a beta release and the source code of their application, which is an Android app that allows users to make events and receive notifications on their phone. Their source code was well-organized and followed a constant format. Methods and variables all had clear breaks, intuitive names, and a good level of commenting. Even with little to no Android development experience, it was easy to tell their flow and functionality of their program despite the limited time we had to view it. There was little in the way of code smells and basic Java best practices were being followed. We suggested to the group that they add in functionality for recurring notifications, which would be useful for remembering weekly or monthly times that the user had constantly. We also suggested that they consider checking their comments for consistency in styling. While almost every function and large block of code was commented, each programmer seemed to have a different idea of what constituted necessary information for the comment. A bit of consistency there would go a long way to making it easier to follow. Overall, Due2mr surpassed expectations and had a robust, well-documented codebase and a functioning beta.

DAPS Reviews Team Harmony:

We walked the team through sections of our codebase and showed them a prototype of our website. The only major issues they presented were styling and consistency in our coding between files. They pointed out that a linter would help solve a lot of these problems. They first mentioned we had yet to have a consistent coding style, with things like tab spacing and blocking being inconsistent across files. They also suggested we try to add more comments in our larger code sections to explain functionality and break up the code more. Readability was the main concern, and we went through and edited the majority of our codebase to meet a single standard. The application as a whole was good, and our class structures accurately depicted our code base. They also suggested we try our application on a web server, which we plan to do for beta testing. Overall we got a thumbs up, on the condition we work on the readability of our code base.

Contribution summary:

Nicholas Smith: Testing strategy, Test cases, Code Review, Contribution Summary, Status Report
Nevin Jacob: Source code cleanup, finishing/fixing last features
Bruno Harvey: Source code cleanup and styling, finishing/fixing last features
Ian Gross: Test cases, formatting test data

Status Report:

Our almost full-feature website is now almost completely feature-full. Notifications and the supply lists were finished, and the new user schema was integrated into the existing database. Emails are now sent to people upon being assigned a task. We also continued work on styling the website and cleaning up the codebase with greater consistency in format and more commenting. Our code now uses the PEP 8 python standard. Testing was also a key part of this iteration: Our alpha testing consisted of two separate trials to check for full functionality and no major issues. Any issues found during testing were logged in the Github Bug Tracker. With this iteration closed, our construction is complete. All that remains is testing and cleanup.