

# Sprint 2 Incremental Testing and Regression Testing

Team 14: CoconutWatr

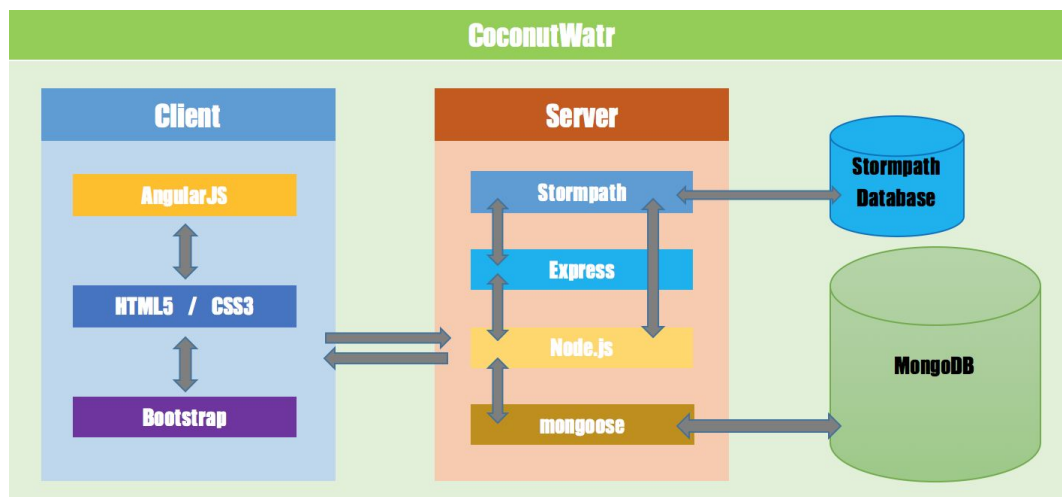
February 14, 2017

## Team Members

Kate Wang, Dominic Valentine, Wilson Husin, Jiho Choi, Kyle Leaf

## Classification of Components

### 1. Define all components



- a. Web Server (Node.js, Express, mongoose)
  - i. Input: HTTP methods (GET/POST/etc.) from client's web browser
  - ii. Output: Web application on client's web browser
  - iii. Dependencies: MEAN stack and Stormpath framework
  - iv. Dependents: Clients (users)
- b. Database (MongoDB)
  - i. Input: Data from users, classes, groups, events and assignments
  - ii. Output: Data from users, classes, groups, events and assignments
  - iii. Dependencies: N/A
  - iv. Dependents: Server
- c. Database (Stormpath Database)
  - i. Input: Data from users
  - ii. Output: Authentication
  - iii. Dependencies: N/A
  - iv. Dependents: Server
- d. User Management (Server-side Stormpath)
  - i. Input: Login credentials (email and password)

- ii. Output: Success/failure logging in
  - iii. Dependencies: Google OAuth, Node.js, user credentials
  - iv. Dependents: Application (determine which data to present)
- e. Clients (basically web browser with JS engine)
  - i. Input: User input (Button / Textbook)
  - ii. Output: content updated webpage
  - iii. Dependencies: JavaScript engine
  - iv. Dependents: User

## **2. Which form of incremental testing did you follow**

We followed bottom up incremental testing. Our product is data oriented, where users will get what they have stored or not. Therefore, we started by testing if we can retrieve what has been (manually) stored in the database. Once that stage has passed, we will then test from another level: would it still work if the data is being inserted the way users do it? If it works, the result will be seen directly.

## **Incremental and Regression Testing**

### **1. Automation**

We would populate the sample of expected pages and then we run a client application which compares the result that we get from the server versus the expected pages.

On the detailed part (checking if data is stored to database, etc) we utilize the retrieve and store functions to check if they are actually stored or not.

We automated createClass by running a script to add multiple class with (00001 to 00100).

We automated createSubgroup by running a script to add multiple combinations of classmates (classmate 01 to classmate 10: 10C2, 10C3, 10C4, 10C5, ..., 10C10)

## 2. Defect log

### Incremental Testing

#### 1. Subgroup

| Defect No. | Description   | Severity | How To Correct   |
|------------|---|----------|--|
| 1          | Subgroups are often "lost" immediately after creation.  | 1        | Ensure that the creator is a member as soon as the group has been created. (Since non-members do not know about any group) |
| 2          | Add classmate to the subgroup not properly showing all the classmates that I can invite to the subgroup.      | 1        | Fix the function to properly show all the user of the current class.   |
| 3          | “Send Invitation” button for sending an invitation to selected classmates not properly sending the invitation | 1        | Fix the server function to properly send invitation to the selected user.  |

#### 2. Sub group chat

| Defect No. | Description   | Severity | How To Correct  |
|------------|---|----------|---|
| 1          | Old chat history can still be seen at the beginning of a new semester                     | 3        | Delete all subgroups and chat history at the end of each semester |
| 2          | Messages from other subgroups were displayed on summary section of all users in the class | 2        | Only retrieve messages from subgroups that the users joined       |

#### 3. MongoDB, Mongoose

| Defect No. | Description  | Severity | How To Correct   |
|------------|--|----------|--|
| 1          | Classes from previous semesters can still be accessed  | 2        | Create a function to clear classes after the semester ends |
| 2          | getUserProperties functions return output in the form of an array of object id's rather than an array of names | 1        | Create parsing functions for each                          |

#### 4. Subgroup Popup Window

| Defect No. | Description   | Severity | How To Correct  |
|------------|---|----------|---|
| 1          | Subgroup popup window is not properly showing the member information of the current subgroup.                   | 2        | Fix the <code>getSubgroupMemberInfo</code> to properly retrieve the member information of the current subgroup. |
| 2          | After clicking "Post" button of the Subgroup popup window, the posted information is not showing in the window. | 1        | Fix the Subgroup window to retrieve and refresh the posted information after user post something.               |

#### 5. Class page - Front-end (HTML5/CSS3/Bootstrap)

| Defect No. | Description   | Severity | How To Correct  |
|------------|---|----------|---|
| 1          | Mobile view (small-size screen of the Bootstrap) of the Class page is not aligned properly to show all the required information in the mobile view. | 3        | Fix the id and class of the frame to proper element to support Bootstrap functions. |
| 2          | Chat window overlaps with class segment   | 2        | Fix alignment of frame  |

## Regression Testing

### 1. Subgroup

| Defect No. | Description   | Severity | How To Correct  |
|------------|---|----------|---|
| 1          | After properly inviting classmate-A to the subgroup, inviting classmate-B not working.  | 1        | Fix mongoose / database to support inviting multiple classmates in to the subgroup.     |
| 2          | User can create / add classes infinitely without any alert or prompt to user which can be potentially harmful for the system. | 1        | Add a feature to track and regulate the number of created and registered class of user. |

### 2. Database

| Defect No. | Description  | Severity | How To Correct   |
|------------|--|----------|--|
| 1          | Database can query unlimited user input of create subgroup (Tested: 50 subgroups add by a user) without any alert. | 1        | Add functionality to alert the admin when size of the user input (create subgroup) reaches certain amount. |
| 2          | Database can query unlimited user input of create class (Tested: 50 class add by a user) without any alert.        | 1        | Add functionality to alert the admin when size of the user input (create class) reaches certain amount.    |

## Updated Product Backlog

| ID | Functional Requirement  | Hours | Status                |
|----|---|-------|-----------------------|
| 1  | As a user, I would like to login with Google Account.   | 7     | Completed in sprint 1 |
| 2  | As a user, I would like to be able to see a summary of what has been going on in my classes (something like news feed). | 15    | Completed in sprint 2 |
| 3  | As a user, I would like to be able to add my classes.   | 8     | Completed in sprint 1 |
| 4  | As a user, I would like to be able to create a class if it does not exist.  | 6     | Completed in sprint 1 |
| 5  | As a user, I would not want to see history of previous semester of a class.   | 8     | Completed in sprint 2 |
| 6  | As a user, I would like to be able to remove class from my feed (in case I drop the class).                             | 6     | Completed in sprint 1 |
| 7  | As a user, I would like to be able to see the classmates in the same class.   | 6     | Completed in sprint 1 |
| 8  | As a user, I would like to be able to communicate with classmates in the same class (public channel of the class).      | 10    | Completed in sprint 2 |
| 9  | As a user, I would like to be able to create a subgroup to chat with certain users in the same class.                   | 11    | Completed in sprint 2 |
| 10 | As a user, I would like to be able to leave a subgroup.   | 9     | Completed in sprint 2 |
| 11 | As a user, I would like to be able to invite classmates to the subgroup.  | 10    | Completed in sprint 2 |
| 12 | As a user, I would like to be able to accept the invitation of a subgroup.  | 7     | Completed in sprint 2 |
| 13 | As a user, I would like to be able to post a message in a specific conversation.  | 6     | Completed in sprint 2 |
| 14 | As a user, I would like to be able to post reminder of an assignment that will be due.                                  | 16    | Completed in sprint 2 |
| 15 | As a user, I would like to be able to create public events with important information.                                  | 10    | Completed in sprint 1 |

|    |   |    |                       |
|----|---|----|-----------------------|
| 16 | As a user, I would like to be able to edit the public events that I made.   | 5  | Completed in sprint 2 |
| 17 | As a user, I would like to be able to delete the public events that I made.   | 7  | Completed in sprint 2 |
| 18 | As a user, I would like to see the public event of others had created.  | 6  | Completed in sprint 1 |
| 19 | As a user, I would like to be able to RSVP to events.   | 8  | Completed in sprint 2 |
| 20 | As a user, I would like to be able to see who is attending an event.  | 6  | Completed in sprint 1 |
| 21 | As a user, I would like to be able to see who created an event.   | 4  | Completed in sprint 2 |
| 22 | As a user, I would like to be able to log in using my Facebook account. (If time permits)   | 9  | Completed in sprint 2 |
| 23 | As a user, I would like to have a help section in the application to provide more detailed explanations of functionality. (If time permits) | 15 | Completed in sprint 2 |