Startup Assignment 6
Team Florida Strangler

Team Members:
Michael Schmittlein
John O'Brien
Minxin Gao
Elvis Chen
Andy Liu

HTTP Requests / Routes:

- GET /user/:userid
  - It replaces the mock server method getUserById. It will return a JSON object of corresponding /user's data by providing the user ID. A user with the specified :userid can issue this request.
- GET /username/:name
  - It replaces the mock server method getUserDataByName. It will return a JSON object of corresponding /user's data by providing the user name. Any user can invoke this method.
- GET /user/:userid/mailbox
  - It replaces the mock server method getRequestData. It will return a JSON object that contains all requestItems that are associated with the user. A user with the specified :userid can issue this request.
- GET /friend/:userid
  - Replaces the mock server method getFriendDataById.
  - Returns a collection of JSON objects, one for each friend of the user.
  - A user with the specified :userid can issue the request.
- POST /requestitem [requestData]
  - It replaces the mock server method writeRequest. It creates a new /requestitem and put it in database. The author of requestData must be the requester to process the request.
- PUT /group/:groupname/user/:username/requestitem/:requestid
  - It replaces the mock server method joinGroup. It marks the /requestitem as read, adds the user as a member of the group, and updates user's groupList. It will return the updated /requestItem. Either the author of the /requestitem or the receiver of the /requestitem need to be the requester.
- GET /user/:userid/feeditem
  - It replaces the mock server method getForumData. It returns the JSON object by passing in userid, the user does not need to get authorization.
- POST /thread [thread Data]

    - The thread data is a JSON Object describing a thread. This creates the JSON object and puts the thread in the forum. Replacing postThread.
    - The current user is authorized to send this request.
- GET /feeditem/:feeditemid
    - It replaces the mocker server method getPostDataById. It returns the JSON object of the feed item. The user does not need to get authorization.
- POST /message [message]
    - Where a message is a JSON object describing a message.
    - Creates a request JSON object and puts it in the mailbox of a user, replaces onMessage mock server method.
    - The current user is authorized to send this request.
- POST /friendRequest [request]
    - Where request is a JSON object describing a friend request.
    - Creates a request JSON object and puts it in the mailbox of a user, replaces onRequest mock server method.
    - The current user is authorized to send this request.
- GET /userData/:userid
    - Replaces mock server method "getUserDataById". The user receives a JSON object of the data for the user given the ID.
- POST /userData/:userid
    - If the user is authorized, a JSON object is sent containing name, grade, email, description and major.
    - Anything updated is sent and anything not updated is just resent.
- POST /resetdb []
    - Replaces mock server method resetDatabase.
    - The developers could would issue the request whenever need to reset the database.
- POST /search [search_key]
    - Replaces mock server method search.
    - Any user including guests could issue the request to search group according to input keys.
- GET /postItem
    - Replaces mock server method getRecommendPost.
    - The request is issued automatically by anyone who load the index page including both authorized users and unauthorized users.
- GET /unReadReq/:userId
    - Replaces mock server method getUnreadMsgs.
    - The request is issued automatically by an authorized users who load the index page.
    - User would be able to receive his/her unread message notification.
- PUT /readRequest/:requestItemId/:userId
    - Replaces mock server method readRequest.
    - The request is issued when an authorized user click the unread messages.
    - That unread messages would be marked as read and removed from user's unread List.
- PUT /user/:userid/friend/friendName

- Replace POST /friendRequest [request], since it doesn't work. It adds friends directly instead of sending a request. It can be invoked if :userid matches the requester id.

Individual Contributions:

Andy Liu:

- Wrote and tested following request
  - GET /user/:userid
  - GET /username/:name
  - GET /user/:userid/mailbox
  - POST /requestitem [requestData]
  - PUT /group/:groupname/user/:username/requestitem/:requestid
  - PUT /user/:userid/friend/friendName
- Wrote requestitem.json for POST /requestitem [requestData]
- Fixed bugs caused by migration of structure (in REACT components)

Michael Schmittlein:

- Migrated to client / server file structure
- Moved reset database to server
- Implemented error banner
- Post message and post friendRequest for friend pages added to server
- Implemented reading friend data from server (with help from Elvis)

John O'Brien:

- Created userSchema JSON for updating user info
- Worked on account page HTTP routes (GET /userData/:userid and POST /userData/:userid)
- Fixed previous bugs with account page, user can now update multiple fields at a time and an alert box asks user for extra confirmation upon submitting new info.

Minxin Gao:

- Forum page feature with HTTP route: GET /user/:userid/feeditem, POST /thread, GET /feeditem/:feeditemid
- Thread.json - json schema for thread.

Elvis Chen:

- POST /search [search_key]
- GET /postItem
- GET /unReadReq/:userId

- PUT /readRequest/:requestItemId/:userId
- Friend Page bugs fixed
- Error-banner bugs fixed
- Generally styling and organizing

Lingering Bugs / Issues / Dropped Features:

Lingering Bugs:

- After posting new thread - original thread title + content still there.

Issues:

- Friend request still having issues
- Thread - Reply to Thread not working
- Thread - Not showing comments
- Direct Messaging between friends is not working

Dropped Features:

- Page numbers