

# Progress Report: React - Pair'd

Guan (Raymond) Chen, Brent Condon, Katie Ho,  
Ella Holmes, Helena Kleinschmidt, Wesley Liao

## Outstanding bugs

- ☐ Intro page displays with navbar at top; this will be removed
- ☐ Cannot delete information from the database when a post date is expired
- ☐ Need to add database to post upon creation
- ☐ The functionality for the glyphicon pencil (edit) buttons on the Shipping & Payment section of the user profile page and the larger Edit button on the top of the profile page is not finalized. These buttons currently redirect to the EditProfilePage, but we would like to integrate EditProfilePage and ProfilePage into one component.
  - ☐ We intend to implement two different views to be rendered (based on the KanBan to do list notes example from class) whether the input fields are being edited or not. So when a user clicks on any edit button, a new view will be rendered within the same one page.
- ☐ The search results page does not yet implement the waterfall jQuery plugin that we are still interested in using; we still have just one flat image collage displayed on this page. We intend for the terms that the user types into the search bar to filter results and only display those deal images in a waterfall format.
  - ☐ The dropdown menu for recent, price: high to low, and price: low to high, is not currently functional.
- ☐ The six images on the homepage will eventually be pulled at random from all the deal images in the database.
- ☐ Still need to design/create our actual Pair'd logo! The leaf glyphicon and Pair'd wordmark are currently serving as our mock logo.
- ☐ Tagbar doesn't load correctly on the first search.
- ☐ Currently tags cannot be deleted.
- ☐ Tags to not split tags words by blank spaces
- ☐ Fake page that is linked to from the search results page does not display the search text
- ☐ Introduction image of pears is visible in post Item and edit user page.
- ☐ When clicking on the profile image in Edit User Page, you are redirected to the introduction page.

## Key React Components

- ☐ *Intro* renders the landing page for Pair'd, which includes a background image of pears. Currently, both the guest & login buttons navigate to the homepage.
- ☐ *Home* renders six featured deals (images) in a tile format.
- ☐ *Item* renders the information for a single deal. Users can be navigated to these single deal pages by clicking on a deal from the homepage, from another user's profile, or by selecting it from displayed search results.
- ☐ *Navbar* navigates to search results, user profile, post new deal, and intro page.

- ❑ *Tagbar* renders the search terms inputted by the user in the search bar and makes them into individual Pinterest-like pills that sit below the search bar.
- ❑ *PostItem* renders the page that has a form/input format for users to fill out and post a new deal on Pair'd. It includes an upload feature for the image and other deal details.
- ❑ *ProfilePage* renders the profile page of the user currently logged into Pair'd. This page renders the *ActiveDealsWidget* by default, as this is the section (of the three pills/panels in the bottom half of the page) that should be seen when this page first renders, along with the *ProfileDetails* that sits in the top half of the page.
- ❑ *ProfileDetails* renders the top section of the user's profile page, which includes their name and contact info, as well as an Edit button.
- ❑ *ProfileWidget* renders the three different pills/panels (Active Deals, Deals History, and Shipping & Payment) of this widget in the bottom half of the profile page, but enables only the 'Active Deals' tab to be active.
- ❑ *ActiveDealsWidget* renders the 'Active Deals' tab of the three different panel area on the user profile page. This section features a 'Post New Deal' button, and two deals posted to Pair'd by our mock user.
- ❑ *DealsHistoryWidget* renders the 'Deals History' tab of the three different panel area on the user profile page. This section features a 'Post New Deal' button, and two deals that our mock user has previously participated in (now expired).
- ❑ *ShippingPaymentWidget* renders the 'Shipping & Payment' tab of the three different panel area on the user profile page. This section features our mock user's current shipping and billing addresses, as well as payment information (all are not currently editable).
- ❑ *ProfilePageGuest* renders the guest view of a user's profile page (as if our mock user was looking at someone else's profile), which includes some basic info and their current (Active Deals) and past (Deals History) deals.
- ❑ *EditProfilePage* renders the edit view of a user's profile page, so that they can edit their basic contact info, billing/shipping info, and payment info.
- ❑ *SearchResultPage* renders the search results (soon to be the waterfall of deal images, but currently one image) based on the terms entered into the search bar by the user.

## Founders' Contributions

1. Katie Ho:
  - a. Combined our separate css files (homepage, item, navbar, paired\_user\_profile, searchresults, widget) into index.css
  - b. Navbar
    - i. Used the Link tag to direct '+' glyphicon in navbar (top right) to postItem page
    - ii. Used the Link tag to direct the leaf glyphicon and Pair'd wordmark to the homepage
  - c. Home page
    - i. Used the Link tag to direct the six deals (images) to our mock item page
  - d. ProfilePage

- i. Added ProfileDetails.contextTypes and router code to app.js to enable the 'Edit' button to direct to EditProfilePage with onClick
    - ii. Added ActiveDeals.contextTypes and router code to app.js to enable 'Post New Deal +' button to direct to PostItem with onClick
  - e. ProfilePageGuest
    - i. Moved profilePageGuest html into render() method for profilepageguest React component
  - f. PostItem page
    - i. Moved postItem html into render() method for postItem React component
    - ii. Added PostItem.contextTypes and router code to app.js to enable the 'Post Deal' button (using the Link tag) to direct to mock item page with onClick
  - g. Item page
    - i. Used the Link tag to direct 'John Hopkins' to ProfilePageGuest
    - ii. Added [www.ae.com](http://www.ae.com) to the deal title's href
    - iii. 'Pair me!' button on item page opens up into a simple modal to congratulate user for participating in a deal and check their email for further details
- 2. Ella Holmes:
  - a. Intro:
    - i. Added intro html into its own component
    - ii. Linked two buttons to navigate to the home page
    - iii. added intro.css into index.css
  - b. Edit Profile Page:
    - i. Added The ability to upload profile picture.
    - ii. Linked Save Edits button to navigate back to the user profile.
  - c. Navbar
    - i. Connected the search button to navigate to the result page
    - ii. Linked the logout button to the intro page
  - d. Tagbar
    - i. Created the tag bar components
    - ii. Changed css buffer so tag bars are visible
  - e. Profile Page
    - i. Connected links from your active deals and you deal history widgets to an deal post.
    - ii. Added working links into the active and pas deals components.
  - f. App.js
    - i. Routed links into apps file into respected functions.
  - g. Server.js
    - i. Contributed to the hours spent of finding a way to write and access the database due to our bug
- 3. Helena Kleinschmidt:
  - a. PostItem.js

- i. Figured out how to add upload file button using simple Bootstrap; this interacts with the client's filesystem where they can ideally upload pictures of their deals
  - b. App.js
    - i. Made homepage display at root
    - ii. Used John's method for making buttons reroute page on click; this was to remove the Link we had previously used on the search bar button (bad)
    - iii. Made all Components operate under user={1} (corresponds to our dummy user with id 1 named John Hopkins)
    - iv. Added :id parameter to ProfilePageGuest route so we can view a particular user's page as a guest
  - c. Navbar
    - i. See b.ii.
  - d. Database
    - i. Reconstructed initial data (reformatted and added new documents)
    - ii. Added reset DB and debugged the whole database fetch mess (as of yet, we are still only fetching one user, but it should get easier from here)
- 4. Wesley Liao:
  - a. Navbar
    - i. Linked the user button to the user profile page
  - b. Profile Page:
    - i. Created subcomponents for profile page, split between each of the profile widgets and profile details.
    - ii. Edit button properly links to the edit page.
    - iii. Readjusted profile details css and html using tables for neater formatting.
  - c. Profile Page Widgets:
    - i. Added links, where each profile widget properly links another.
    - ii. Shipping and Payment widget pencil links to edit page
  - d. Profile Guest Page:
    - i. Properly displays information only available to guests
    - ii. Deals History and Active Deals widget properly link to one another on click.
    - iii. Deals in widgets properly link to a temporary item page.
  - e. Edit Profile Page:
    - i. Profile privacy drop down menus change properly
    - ii. Put HTML elements into tables for much neater formatting.
- 5. Brent Condon:
  - a. Navbar
    - i. Stores input inside of the search bar in a state inside of navbar.
  - b. Tagbar
    - i. Fixed a problem where tagbar would always render, now it only renders when on the search page.

- ii. Creates new tag within the tagbar when the search button is pressed (or enter key is press).
  - iii. Fixed problem where the tags stay the same when leaving the search page and then returning.
  - iv. Prevented user from entering multiple tags with the same name.
- 6. Guan Raymond Chen:
  - a. Database
    - i. Added mock initial data to database according to our embeddings and reference relationships illustrated in our ER diagram
  - b. Profile Widgets
    - i. Added the initial components
  - c. Apps.js
    - i. Added route to a fake deal page
    - ii. Added fake deal page class
  - d. Search Results Page
    - i. Linked the image of the search result to a fake deal page

### **Cut Product Features**

We have so far not decided to cut features from what we have presented in the past. Our app is designed to be 1.0 from the beginning; we can think about 2.0 later if we have the opportunity.

### **Updated ER Diagram**

Our previous ER Diagram is still up to date.