



# Static UI Mockup (With React)

March 2017

## **Team Mango Members**

John Drakos

Yixiang Xu

Michael Feuerstein

Yi Fung

Bryce Bodley-Gomes

Ryan Semple

## Components:

**Navbar.js:** This component contains the navbar that was previously hard coded into each page navbar. By reactifying this it means that one change to the navbar.js will be instantly applied to all pages rather than having it only apply to one page.

**Homepage.js:** This component is the container for all the react components on the homepage (navbar and homepagebody).

**Homepagebody.js:** This component contains the two carousels, the advertizing carousel and the recommended items carousel.

**HomepageCarouselItem.js:** This component is an easily modifiable and reusable container for an item in the recommended items carousel (contained in homepagebody). It takes three props: the **name** of item, the **price** of item, and a **reference** to the image of the item.

**profilepage.js:** This component contains navbar (using navbar.js) and all the react components on the profilepage (profilepagebody).

**profilepagebody.js:** This component contains the profileinformation react component, the selling history react component, and save container.

**profileinformation.js:** This component is what allows users to change their avatar, nickname, password, email address and telephone number. It takes 4 props: **nickname** (the user's nickname), **password** (the user's password field), **email** (the email associated with the account), **tel** (the telephone for contacting the user).

**profilesellinghistory.js:** This component contains a list of profile items which the user has sold or is currently selling.

**profileitem.js:** This component represents an individual item the user is selling or has sold. The props it takes are: **img** (a reference to the item's image), **name** (the item's name), and **color** (a hex value representing the current selling status. Red = sold, Green = selling), **status** (the text value indicating whether an item has sold, or is selling).

**classpage.js:** This component contains the navbar (using navbar.js) and all the react components on the classpage (classpagebody).

**classpagebody.js:** This component contains the classinfo(using classinfo.js) and the class tabs. Currently only the class materials tab is in use. It populates rows of items(classitem.js) with course material.

**classinfo.js:** This component contains the data related to course: Course title, description, professor, credits and the term.

**classitem.js:** This component contains information for the individual items being sold, including image, brief description, and price. The props are: **reference** (the reference to the image of the

item), **description** (a description for the item), and **price** (the price for the item).

**searchpage.js**: This component is the base level container for the search page. It contains the navbar (using navbar.js) and all the react components on the body of the search page.

**searchresults.js**: This component is primary content composing the search page body. It contains a grid of the selling items, reactified further by using searchitem (searchitem.js) subcomponents. It also contains the pagination for the search page (searchpagination.js) at the bottom. Takes prop named **query** (what the user searched on).

**searchitem.js**: This component contains information for an individual item. including image, brief description, and price. It also contains a heart glyphicon for users to add the item to the wish list. Prop names are: **des** (the item description), **src** (the reference to the image for the item), and **price** (the price for the item).

**searchpagination.js**: This component contains a horizontal listing of the pages of the search results, identified by the page number 1, 2, 3, etc. The page number is reactified further by the searchpaginationid.js subcomponent.

**searchpaginationid.js**: This component renders the specific page number used in the pagination for the search page results.

**submissionform.js**: This component is the container for all the react components on the submission form (navbar and submissionformBody).

**submissionformBody.js**: This component contains the header for the page, and then separates the left and right side of the document into separate components for easy editing.

**submissionformLeft.js**: The left side contains the components for the picture of the item, the main criteria descriptions of the item, and the submit button.

**submissionformPicture.js**: This component allows a user to upload a picture and displays it.

**submissionformDescription.js**: This contains the main criteria required for submitting an item. It includes details such as title, price, and condition.

**submissionformRight.js**: The right side contains other useful information fields for describing an item. This includes whether or not an item is class-related and the category that the item belongs to.

**submissionformClass.js**: This component contains the fields used to express how the item may or may not be class related.

**submissionformCategory.js**: This component contains all the different categories and the information fields related to each category.

**item.js**: This component is the container for all the components on the item page (navbar,

itemhero, itemdescription, moreitems, and subitem).

**itemhero.js:** This component contains the image and description of the item, as well as the buttons for contacting the seller and watching the item.

**itemdescription.js:** This component contains the description of the item and buttons for action by the user.

**moreitems.js:** This component is used to display the “Related Items” and “More by Seller” groups of items. It contains several SUBITEMS.

**subitem.js:** This component represents one item in either the “Related Products” or “More by Seller” groups. It contains an image, a title, and a condition.

### Bugs

**Homepage:** No known bugs. Fixed “bug” relating to image sizes in the carousel, the images are now scaled to a max size in case the user attempts to upload an exceptionally large image it will no longer distort the page layout.

**Submission Form:** No known bugs. Fixed a ton of bugs related to styling issues caused by using in-line styling as opposed to using css. Converting in-line to css fixed issues.

**Profilepage:** Bugs are all about styling, and they are fixed using css to do all the styling part.

**Searchpage:** No known bugs. Fixed styling issues caused by using in-line styling as opposed to using css.

**Classpage:** No known bugs. Fixed issues with image scaling incorrectly by setting a max height and width.

**Itempage:** No known bugs, some minor changes were made with content to simplify the React props used.

### Responsibilities

**Bryce Bodley-Gomes:** Rewrote navbar as a react component to be reused across all pages. Reactified homepage into the subcomponents:(Homepage.js, homepagebody.js and homepageCarouselItem.js), helped compose Static React UI Report.

**John Drakos:** Reactified Item page into the subcomponents: (itemhero.js, itemdescription.js, moreitems.js, and subitem.js). Helped compose Static React UI Report.

**Yixiang Xu:** Reactified Profile page into the subcomponents: (profilepage.js, profilepagebody.js, profileinformation.js, profilesellinghistory.js, profileitem.js, profilesave.js), helped compose Static React UI Report.

**Michael Feuerstein:** Reactified Classpage into subcomponents: (classpage.js, classpagebody.js, classinfo.js, classitem.js); helped compose Static React UI Report.

**Yi Fung:** Reactified searchpage into the subcomponents: (searchpage.js, searchresults.js, searchitem.js, searchpagination.js, searchpaginationid.js); helped compose Static React UI Report.

**Ryan Semple:** Reactified Submission Form into the subcomponents. Components include: submissionform.js, submissionformBody.js, submissionformLeft.js, submissionformPicture.js,

submissionformDescription.js, submissionformRight.js, submissionformClass.js, and submissionformCategory.js, helped compose Static React UI Report.