

## **Bike Part Picker Dynamic Website Written Report**

**Tony Gao**  
**Matthew Briden**  
**Andrew Wang**  
**Peter Nguyen**  
**Rajesh Susai**

### **List of undesirable behaviors:**

1. On the SavedBuilds page, it lists 3 builds that belong to a user. When you click on a build, it takes you to the SelectBikeType page rather than the SelectBikeParts page with the parts filled in
  - a. This problem is caused by the state logic used in Build.js, there was no time to implement the SelectBikeParts state loading from a saved build.
  - b. In app.js, the prop for current\_state must be passed and handled properly depending on if it is a “new build” use case or a “saved build” use case.
2. When clicking on the navbar to navigate to another page, the part of the url displays [object Object]
  - a. This problem is caused by some code in navbar.js where the router links the button clicks to different pages. It tries to link the page that’s tagged with the userId object and because of that, it’s displaying an object of type Object in the URL.
  - b. There are no plans to solve this, as there are no functional differences as of now.
3. When you review a build and save, it doesn’t show that build in SavedBuilds
  - a. This is an unimplemented feature, there was not enough time to create “build instances” and relate them to a “user” in the database, and any of the dependent classes/components.
  - b. The plan to resolve this is to implement the user->build one-to-many instance relationship properly in the next Sprint.
4. My Account save button doesn’t work
  - a. This could be a server function or an improperly implemented save button.
  - b. We plan on reconstructing some server functions to combine updating the field into a single function.
5. Logout does not actually log off the user from the website, it just takes the user to the main page.
  - a. Support for multiple users of the same type is not supported yet.
  - b. We plan on resolving this once servers/database are introduced to the project
    - i. Blocked by current curriculum.
6. When you select a bike type, enter SelectBikeParts, and then try to click New Build on the navbar, the app does not re-enter SelectBikeType.
  - a. This is a problem with state logic when you enter SelectBikeParts
  - b. There are no plans to solve this yet, if we have free time, we will implement this, but refreshing the index to create a new build is a feasible workaround.

7. No matter what bike type is selected, all possible parts are populated in the dropdown. For example, bike type "Road" has options in the dropdowns for both shock and fork, even though there shouldn't be those options available for type "Road"
  - a. Issues with Bike Type differentiation after a Bike Type is selected
  - b. The plan to fix this is to properly filter the list by bike\_type
8. A "new build" saves the old build from any previous build session, because builds, users, and all data are currently just being mocked to only one build id, user id, and so on.
  - a. Once parts are added, we will constantly read that build with those parts added no matter where we are in the website. The build isn't being written to the database and a new build with a generated buildId isn't being assigned to it. There was no time to implement these features.
  - b. The plan to resolve this is to properly implement the "instances" of the one-to-many relationships (same as issue 3) between accounts -> build and build -> parts. The intended meaning of "instance" is that builds currently are not associated to instances of users, merely a hard-coded user that are assumed to be logged in. Regarding multiple users, there are no "instances" of users yet.

### **Key React Components:**

- Navbar
  - This component renders on every page in the web application.
  - The navbar contains buttons that route to different pages in the application
- Build.js
  - This component renders both SelectBikeType and SelectBikeParts based on state switching logic, along with ReviewBuild with SelectBikeParts
  - Step 1: SelectBikeType
  - Step 2: SelectBikeParts
  - Step 3: ReviewBuild
- SelectBikeType
  - This component renders buttons of different bike types, which are currently "mountain", "city", "trail", "winter"
  - Clicking on any of those buttons will take you to SelectBikeParts
- SelectBikeParts
  - This component shows the list of parts in a single build with a review button to save and review the current built bike.
  - Clicking the review button will add the ReviewBuild component to the bottom of the page.
- ReviewBuild
  - This component will show the the total price and let you name your build and when you press save, it'll display a message congratulating you on making your build.
- Account

- This component shows the account information of the user who's logged into the website and by clicking the save button the user can update their account information.

### **Member Contributions:**

#### Tony:

- Managed tasking
  - Lead project planning, distribution of tasks
- Home.js page wrapper
- HomeItem.js
- Contributed to main app.js page wrapper
- Styled and consistency check for transparency/styling consistency
- SASS inline css loading through webpack
- SelectBikeType component
- Created state switching logic and main code wireframe for build.js, pair programmed with Andrew for the rest of the work
- Assisted/made hotfixes on router development
- Fixed all console warnings and errors caused by bugs in SelectBikeParts.js, app.js, Account.js, Build.js, SeletBikeType.js, ReviewBuild.js
- Created ReviewBuild.js
- Assisted with server functions with Matt and Raj

#### Matt (pair programmed with Raj for majority of work):

- Made SavedBuilds.js component and logic
- Worked on adding dynamic page name and current user name on NavBar
- Wrote all mock database data with Raj
- Helped Raj's write server functions, and modify them to work for the particular components we used them for
- Designed logic to populate dropdowns in SelectBikePart.js component
- Designed logic to add name and price of each part selected in SelectBikePart.js (Raj helped with these two SelectBikePart.js logic implementations)
- Helped pass props to select components

#### Rajesh (pair programmed with Matt for majority of work):

- NavBar component
  - Passed props for getting name on NavBar
- Developed Server functions
- Helped with the integration of server functions in React components
- Translated data model into JSON
- Assisted with SavedBuilds development with Matt
- Worked on development of SelectBikeParts with Matt

#### Andrew:

- Build.js component+state logic(with Tony)
- Provided skeletal/basic JSON mock data format
- Linked Build.js state logic to selectBikeType, SelectBikeParts, ReviewBuild(with Tony)
- React Router
- Paired programming with Tony for Build.js, selectBikeType

Peter:

- Footer component
- Developed Account page

**Features we have decided to cut:**

- There will be no algorithmic searching of compatible bike parts. This was found to be too large a time commitment to be feasible. Instead, bike parts are offered by bike\_type, which can currently be “mountain”, “city”, “trail”, “winter”. Any bike part offered under the certain bike\_type categories will be pre-screened to be compatible, and when a user wishes to build a certain bike\_type, only compatible parts will be listed, no mutually incompatible parts will be provided.