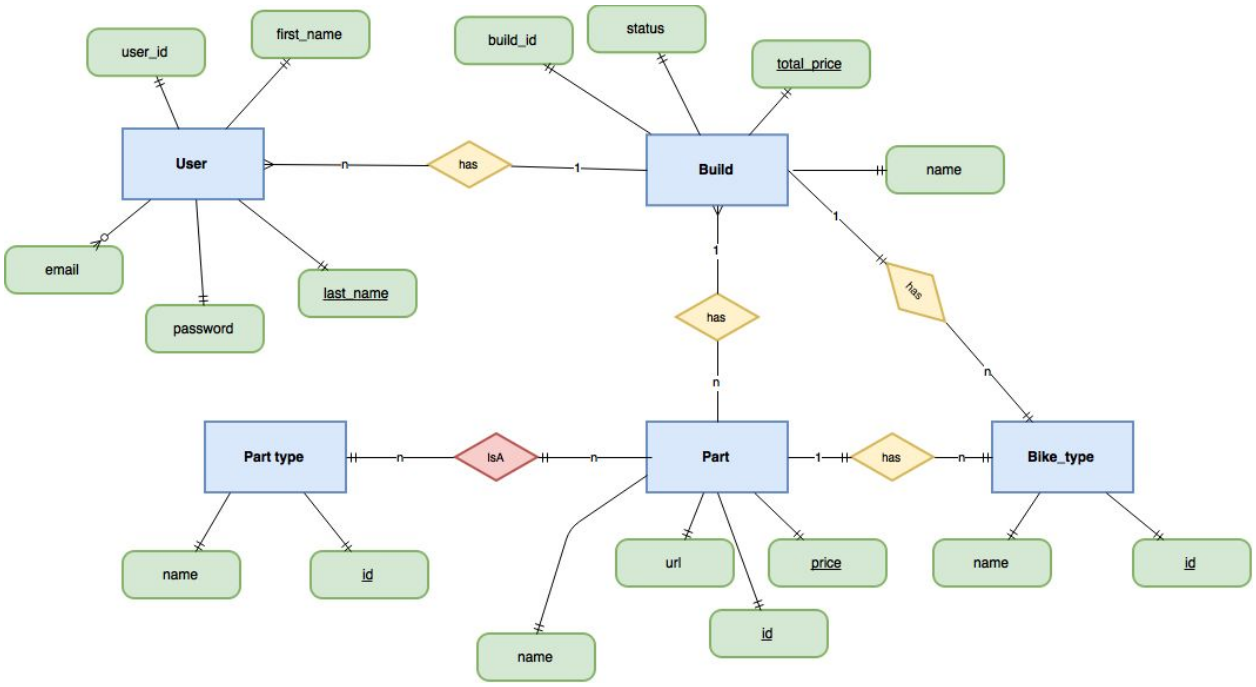
Team Apple:

- Rajesh Susai
- Matt Briden
- Peter Nguyen
- Tony Gao
- Cong Wang

Entity Relationship Diagram



Notation:

- Embedded relationship: red
- Reference relationship: yellow
- Entities: Blue
- Attributes: Green

Entity Descriptions

User: The representation of a user, which will be an account on the webpage. This will be used by all pages to load user information (name) into the dropdown at the top right corner. This will also be used by the account page to modify name, email, and password attributes. One user can be related to zero or more builds. The SavedBuilds page will allow each User to have a list of builds, accessible by the build_id. The MyAccount page will show the majority of the User

attributes (user_id and password being omitted). Excluding the user_id, the User attributes can be modified through an EditAccount page. This entity has the following attributes:

user_id : the unique id key for a user, will be used internally to differentiate users
first_name : the first name of the user
last_name : the last name of the user
password : the user-set password (hash will be used for security)
email : the email address of the user, will be used to send a registration confirmation

Build: This is a representation for a build. One user can be the owner of many builds. Builds will be stored by an id reference in a user. The SavedBuilds page will load all builds created by the current user. A link will be added in the table for each build with the type of bike, name of the build, total price, author (current user), and completion status. When the link is clicked, then MyBuild page will be loaded with the chosen parts associated with that build. This entity has the following attributes:

build_id : the unique id key for a build, will be used to relate users with builds
status : the completion status of the build (completed/ not completed)
total_price : the total price of the build
name : the user-defined name of the build

Bike_type: This is a stored enumeration of types of bikes, which will currently hold the values ‘mountain’, ‘city’, ‘trail’, ‘snow’. This enumeration will be used internally to classify builds and parts (most parts are only compatible with one type of bike) and also as a first-step compatibility filter (if a part does not have a certain type listed in its bike_type list, then it will not be displayed on the my build page). The image buttons on the build bike page will correspond with the set of stored bike_type values. The main page in our website that will be using the Bike_type entity is the MyBuild page. The MyBuild page is where the user actually selects the Build_type for the bike they are building. Thus this is the page where the Build will become connected to the selected Bike_type. This entity has the following attributes:

name : the name of the bike type
id : the unique id key for the bike type, will be used to reference bike types (ex. id value of 1 correlates to mountain bike, so if ‘1’ is in the list of Bike_type values for a part, it is compatible with mountain bikes)

Part: The Part entity represents each part that we will store in our database. Each Part will have a corresponding id, url of the third party website where the part can be bought, the price of the part, and the name of the part. Along with this the Part entity is connected to both Part_type and Bike_type. The connection to Part_type is that of a “is a” relationship via the Part_type id. The connection to Bike_type, however, is that of a “has a” relationship likewise using the Bike_type id. This is necessary due to the extensive amount of different parts we will offer our users and the filtration of parts unique to each of the different bike types. The pages of our website that will be using the Part entity are My Build and Bike Build. Bike Build, the page where the user decides which type of bike that they would like to build, will filter what Part objects we will

provide the user with when they are building that bike. However, the My Build page is where the user will actually select what Part objects will be in their Build. This is where the Build will be associated with the corresponding Parts. This entity has the following attributes:

name: name of the part
url: url of the third party website that hosts the part
price: price of the part
id: unique id of the part

List of part type NAMES:

WHEELS – (FULL WHEELS) (including hub, rims, spoke, all pre-built and ready to go)

HANDLEBARS - (Divided into specific styles for each – make sure it matches frame spec. Dropbars for road, and standard bars for mountain.)

SEATPOST - (UNIVERSAL PART FOR ALL BIKE TYPES. 27.2mm only for us for simplicity, attaches to saddle, fits in all frames)

SADDLE - (Universal-somewhat. Part of the bike the person sits on(139-149mm is the common width range))

FRAME - (TYPE SPECIFIC BIKE FRAME - TRAIL/MTN/ROAD/WINTER(FAT))

Shock - (TRAIL/WINT/MTN ONLY. Rear suspension shock, self explanatory.)

DERAILLEUR - (Front) (Divided into two categories – mtn for trail/winter, road for everything

Else. Shifts chain onto gears at middle of bike)

DERAILLEUR - (Rear) (Shifts gears at rear of bike, has mountain and road versions)

CHAIN - (Match to derailleur speed count. KMC brand only for easy choice)

BRAKE - (Hydraulic only for all – Shimano has lines covering all of these. Stops the bike, comes with lever, caliper, and rotor)

FORK - (Included on road frames, required for trail/winter. Suspension fork.)

SHIFTER - (Divided into mtn/road, again. Attaches to derailleurs to shift gears)

TIRE - (Tire types. 29x2.x for trail/mtn, FAT for winter, 700c for road/city)

Above, we have a list of part type ‘names’

Part type: This entity has attributes that allows the app to filter results that return when a user queries for available parts for a build under some bike type, in the myBuild page. They have the following attributes:

name (from the list of types above),
ID assigned to each type to identify in database.

These attributes will be used to filter what appears in a search query to the database and cannot be modified.

The MyBuild page provides the user with all the Part_types that need to be selected to build their bike for each specific Bike_type.This entity is linked to part.