Rework endpoints

Double categorize

* One list for what requests will be made on each page
* One list for what requests will be handled by each controller
  + This is where we will format actual enpoints
    - Method
    - Path
    - Function Name
      * Maybe a quick description

Question for Brady

Should I sperate functionality or group it together, one request with a lot of function or multiple requests with simple functions?

By File

* Auth Controller
  + Look at previous project for examples
  + POST – login
    - /login
    - Body: credentials
  + POST – register
    - /register
    - Body: new user information
* Ticket Controller
  + GET – get upcoming tickets
    - /tickets
    - getTickets
      * Get a list of tickets with the closest due dates
  + GET – get users tickets
    - /tickets/:userId
    - getUserTickets
      * Get a list of tickets for a given userId sorted by date
  + GET – ticket by ID
    - /ticket/:ticketId
    - getTicketDetails
      * Get ticket, user, and bike information on a specific ticket
  + GET – get tickets according to search params
    - /tickets/search
    - searchTickets
      * Get tickets according to search params
  + POST – ticket
    - /tickets
    - newTicket
  + PUT – edit ticket
    - /tickets/:ticketId
    - editTicket
    - update notes/status/due\_date
  + DELETE – ticket by ID
    - /tickets/:ticketId
    - deleteTicket
* Ticket Items Controller
  + GET – all items by ticket ID
    - /tickets/items/:ticketId
    - getTicketItems
  + GET – ticket total by ticket ID
    - /tickets/total/:ticketId
    - getTicketTotal
  + POST – add ticketLabor item
    - /tickets/labor
    - Body: ticketId, laborId
    - addTicketLabor
  + PUT – Update ticketLabor quantity
    - /tickets/labor/:ticketLaborId
    - Body: new quantity
    - updateTicketLabor
  + DELETE – ticketLabor
    - /tickets/labor/:ticketLaborId
    - deleteTicketLabor
  + POST – add ticketProduct Item
    - /tickets/products
    - Body: tickedId, productid
    - addTicketProduct
  + PUT – Update ticketProduct quantity
    - /tickets/products/:ticketProductId
    - Body: new quantity
    - updateTicketProduct
  + DELETE – ticketProduct
    - /tickets/products/:ticketProductId
    - deleteTicketProduct
* Users controller
  + GET – customer list
    - /users
    - Query: field input
    - searchCustomers
  + GET – bike list
    - /users/:userId
    - getBikes
  + PUT – update customer / bike
    - /users/:ticketId
    - updateUserInfo
  + POST – create new user
    - /users
    - Body: user information
    - createUser
  + POST – create new bike
    - /users/bike
    - Body: new bike information
    - createBike
* Todo list controller
  + GET – TO DO List
    - /todoList
  + POST – new TO DO List item
    - /todoList
    - Body: status, description
  + PUT – edit TO DO List item status
    - /todoList/:todoId
    - Body: status, description
  + DELETE – TO DO List item
    - /todoList/:todoId
* Items Controller
  + GET – search products / labor
    - /tech/catalogue
    - Query: search input
  + There will be more customer related endpoints

Tech Requests

* Dashboard
  + GET – Ticket List + user + bike + ticket total
    - Sorted by due date
  + GET – Ticket Details + user + bike
    - Params: ticket\_ID
    - Redirect to ticket details page
  + GET – Ticket search + user + bike + ticket total
    - Query: search input
  + GET – TO DO List
    - Get all and sort by created date
  + POST – TO DO List
    - Create new list item
  + PUT – TO DO List
    - Change list item status
  + DELETE – TO DO List
    - Delete list item
* Ticket Search Page
  + GET – Ticket search + user + bike + ticketLabor + ticketProducts
    - Query: search input + sidebar refinement
  + GET – Ticket Details + user + bike + ticketLabor + ticketProducts
    - Params: ticket\_ID
    - Redirect to ticket details page
* New Ticket Page
  + POST – Ticket (potentially create a new user and/or new bicycle)
    - Body: all ticket information
    - Redirects to ticket details page
  + GET – Customers list
    - Query: match inputs value
  + GET – Bikes list
    - Param/query: user\_id
* Ticket Details Page
  + GET – Ticket Details + user + bike
    - Param: ticket\_id
  + PUT – Edit customer + bike
    - Param: ticket\_ID
    - Update on submit
  + PUT – Update ticket notes / status
    - Param: ticket\_ID
    - Update on change
  + GET – Ticket Items
    - Params: ticket\_ID
    - Get all items related to a ticket
  + PUT – Update ticketLabor quantity
    - Param: ticketLabor\_ID
  + DELETE – tikcetLabor
    - param: ticketLabor\_ID
  + PUT – Update ticketProduct quantity
    - Param: ticketProduct\_ID
  + DELETE – ticketProduct
    - Param: ticketProduct\_ID
  + DELETE – Ticket (and ticket items)
    - Param: ticket\_ID
  + GET – search products / labor
    - Query: search input
    - Redirect to search page
* Product/Labor Search Page
  + GET – search products / labor
    - Query: search input + sidebar refinement
    - Redirect to search page
  + POST – add ticketLabor to Ticket
  + POST – add ticketProduct to Ticket

Login Requests

* POST – register
* POST – login

Customer Requests

* Dashboard
  + GET – user ticket list + bike + total
    - Params: user\_id
  + GET – Product List
    - Preset initial list
  + GET – refined product list
    - Refine return based on search input
  + GET – product details
    - Params: product\_ID
* Ticket Details Page
  + GET – ticket details + user + bike
    - Params: ticket\_id
  + GET – ticket items
    - Params: ticket\_id
* Shopping Page
  + GET – refined product list
    - Return products based off search input
    - Refine return by settings in side bar
    - Page switching functionality
  + GET – product details
    - Params: product\_ID
* Product Details Page
  + POST – add product to cart
    - body: product\_ID, user\_ID
    - redirect back to search after success message
* Manage Cart Page
  + GET – get cart
  + PUT – edit item quantity
  + DELETE – remove item from cart