

Project Milestone 4

Gr8 Team

Revised List of Features

- **This is an updated list of your FEATURES inventory (from Milestone 2).**
- **It is normal for feature lists to change during the course of a project.**
- **Some features may have been dropped. Some features may have been added.**
- **This revised features list should reflect these changes.**
- **This revised features list should identify the PRIORITY order of how the features will be developed.**

Features:

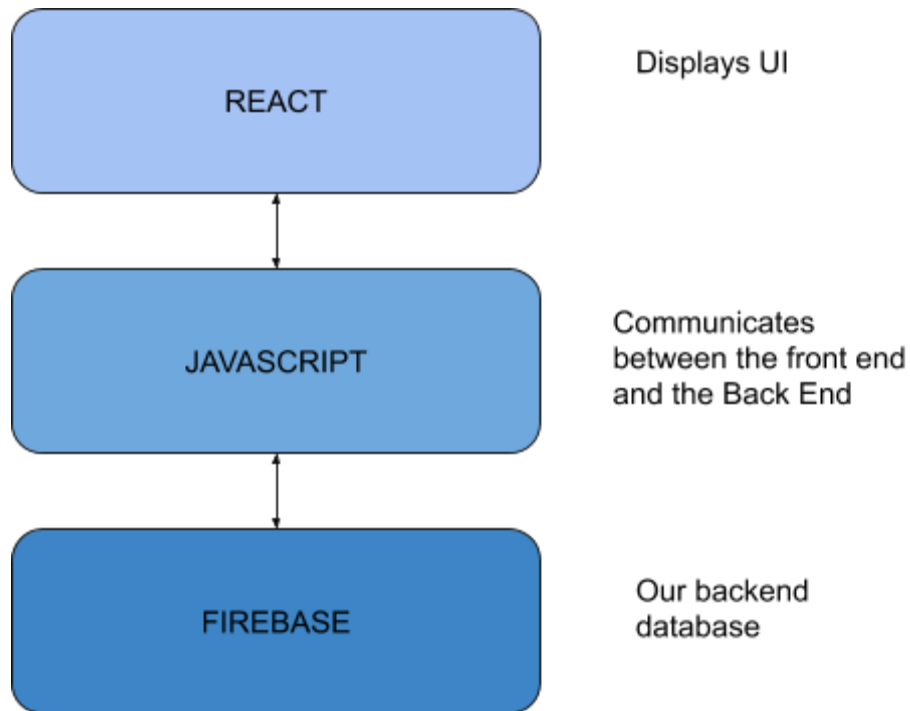
- Search Feature
 - Functional:
 - The user can search and filter books by class, price, ISBN Number, etc.
 - Non-Functional:
 - The client will request all the listing and filter results based on the search filters
- Product Page
 - Functional:
 - Page with information about the content, price, and reviews of the book. Product page will also show if there is availability
 - Non-Functional:
 - Linked from the search page
 - Request data from the listing on Firebase
 - User can choose one of the listings and get in contact with the seller, as well as go to the payment page if they want to purchase
- Account Page/History
 - Functional:
 - Only accessible to logged in users, will show purchased history and sold history
 - Non-Functional:
 - Request User's info from Firebase and display that data
- Seller Page - New Listing
 - Functional:
 - The sellers of the book should be able to put up their books for sale, setting their desired price, giving a description of the quality of the

- book, and verifying the ISBN in order to make sure that it is a real book which will be useful to the marketplace.
 - Non-Functional:
 - Users will be able to enter their book into the database of books being sold.
 - ISBN will likely be verified using an API of some sort
- Home Page
 - Page to inform the user what our product is, as well as providing an aesthetically pleasing landing page for our marketplace.
- Messaging Feature
 - Functional:
 - Buyer and Seller should be able to send each other messages to confirm delivery of the book, or do other things such as negotiate price
 - Non-Functional:
 - Client will request from that user's chat history and push new messages/conversations when they occur
- Payment Page (will implement if we have time at the end of the project)
 - Functional:
 - The user is able to enter payment information and complete a purchase
 - Non-Functional:
 - Will reroute users to a third-party payment system so that our website doesn't have to deal with transactions and storing credit card info, etc.
 - Paypal would be an example of a payment system option
- Login/User Authentication (going to implement if we have time at the end of project)
 - Functional:
 - The user shall be able to Login using @colorado.edu Google account or Identikey
 - Non-Functional:
 - Authentication using Firebase Authentication (OAuth2)

Architecture Diagram

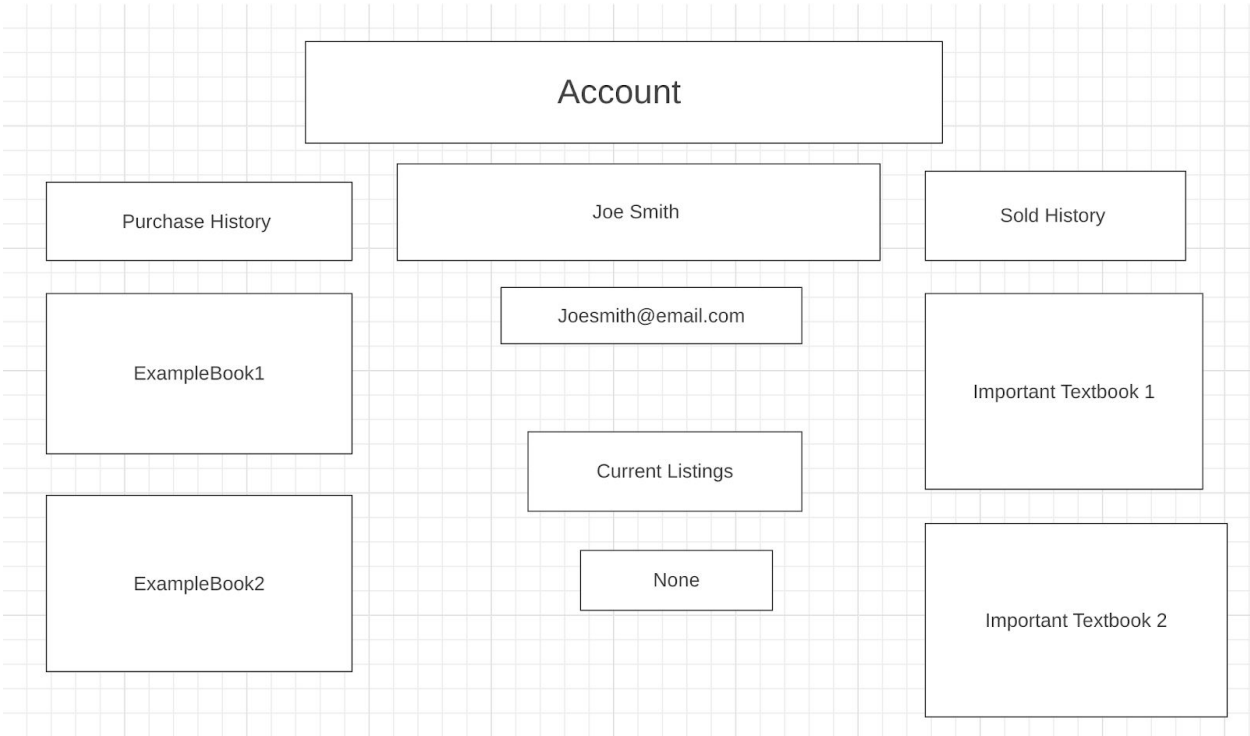
- **This deliverable is a picture or diagram that shows each architectural component of your application.**

- The diagram should identify how your application's front-end, integration layer, and backend processes will be hosted.
- This diagram should identify the flow of data from one layer to another.
- This diagram should identify the protocols being used to/from each component layer.



Front End design

- This deliverable is a series of diagrams that show the basic design of your application's front end.
- Typically, this design is most easily presented in terms of a WIREFRAME.
- This may be hand drawn, or created using a web page wireframe drawing tool.
- The front end design should identify each major feature of the application's front end



Chat

New Chat

Bob Smith : I want to purchase your calc textbook

Bobby Joe : Can you do a better price for your english textbook

Billy Bob : Was this textbook helpful for that class

Web Service Design

- **If your application is using Web Services via APIs, this deliverable should list the Web Service being used along with a description of the Web Service's API including the data being passed to and received from the API. If your application is using Web Services via APIs, this deliverable should list the Web Service being used along with a description of the Web Service's API including the data being passed to and received from the API.**

Currently we are looking into an API which will allow us to verify ISBN numbers, and determine if they exist, so that the seller is unable to list non existent books, and we can use ISBN as a search parameter..

Database design

- **This deliverable provides a summary design of your application's database.**
- **The design document should identify each type of data being stored in your database.**
- **This may be documented in terms of a schema definition, showing data entities ("files") and attributes ("fields")**
- **This may be documented via an Entity Relationship Diagram showing database tables and columns.**
- **The document should identify the specific DBMS technology being used to store your application data (PostgreSQL, MySQL, Firebase, etc.)**

We will be using firebase as a back-end database. Everything in the firebase database is stored in a json file which allows for readability and easy compatibility through a combination of dictionaries and arrays. There are no tables or records unlike a SQL database. When data is added to the JSON tree, it becomes a node in the existing structure with an associated key. SCHEMA pictured below. Typically, it is best to avoid creating a deep hierarchical structure in your database as this can lead to security issues, so our structure attempts to be as flat as possible to deal with this.

