

# HofSwap Architecture Document

**Produced by:**

Trey Jean-Baptiste

Frank Martin

Parhum Ebrahimian

Aishik Mallick

Joseph Gentile

# ***Table of contents***

**1. ARCHITECTURAL OVERVIEW**

**2. CLIENT SERVER PATTERN VISUALIZED**

**3. AUTHENTICATION PAGES**

**4. PROFILE PAGES**

**5. TEXTBOOK UPLOAD PAGE**

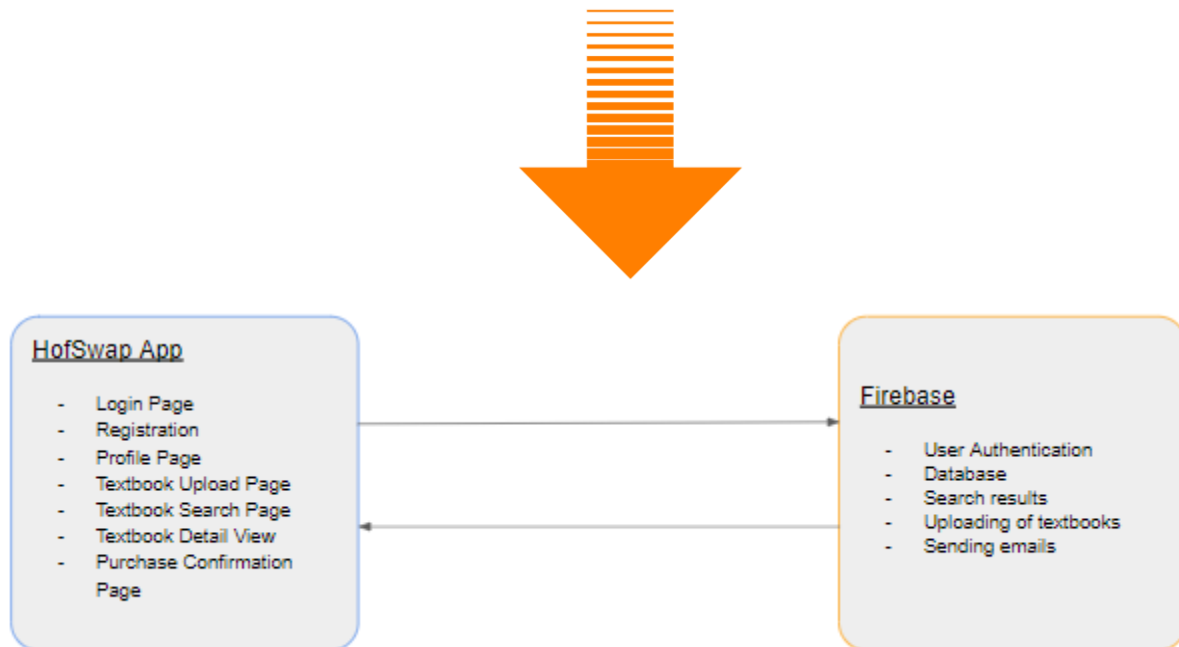
**6. TEXTBOOK SEARCH**

**7. DRAWBACKS**

# Architectural overview

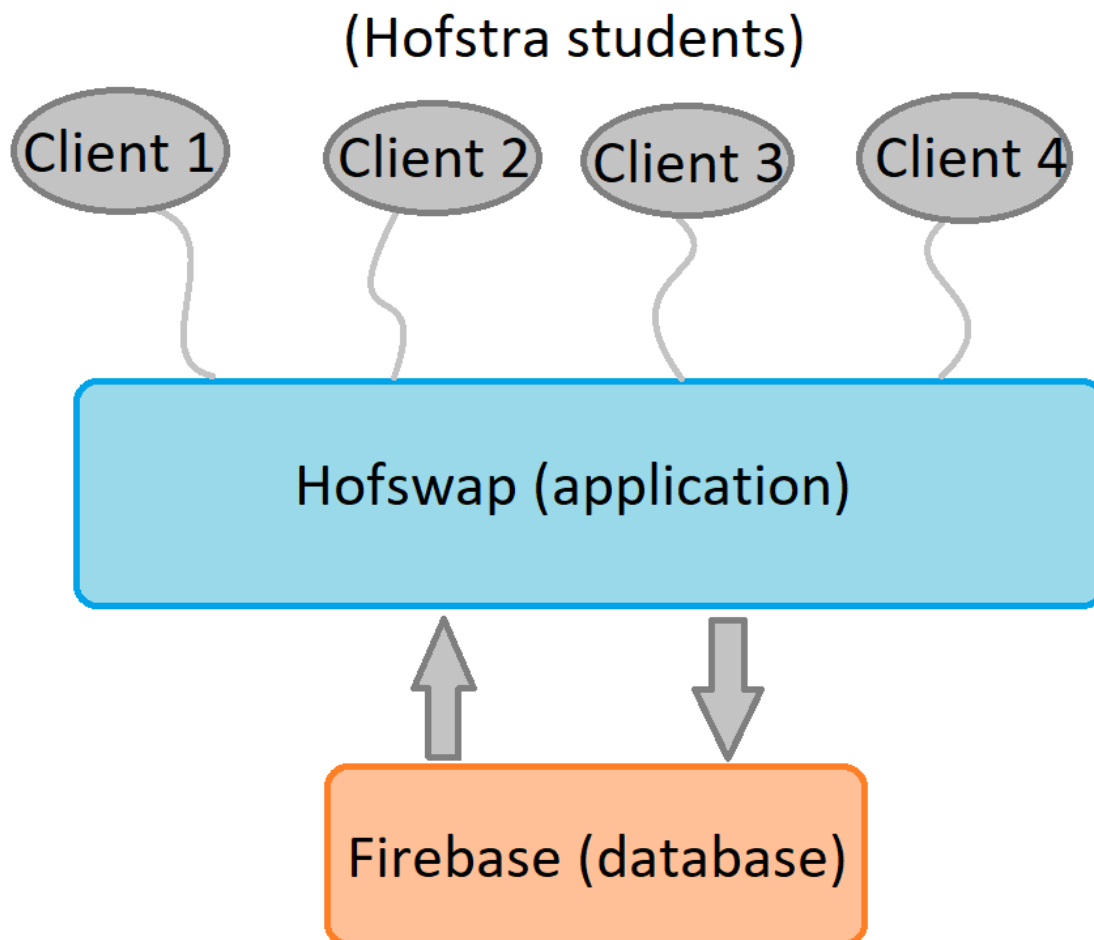
For the HofSwap application we will be communicating with the amazon firebase. We are not building our own server. Instead, HofSwap uses a database built on Google's Firebase technology. We have little to no experience with the development of databases. Firebase should allow us to develop substantially faster and reduce our risk. All backend processing of textbooks and user information is handled by Firebase.

Demonstration of profile page visualized:

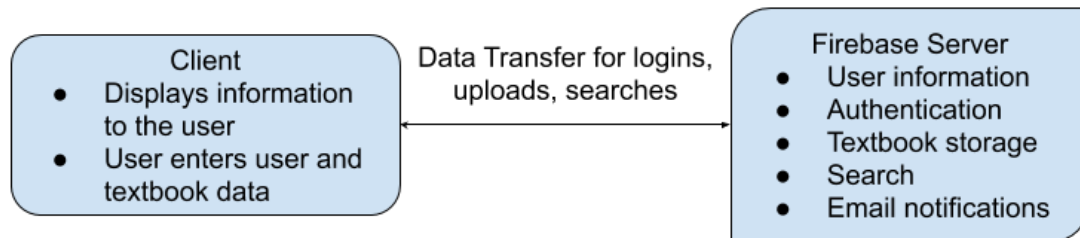


# Client-server pattern visualized

Architecture pattern: The architecture design pattern that is being used is the client server model. The clients (Hofstra students exchanging textbooks) are using the HofSwap application to access data from the shared database (Firebase).



# Architectural Model

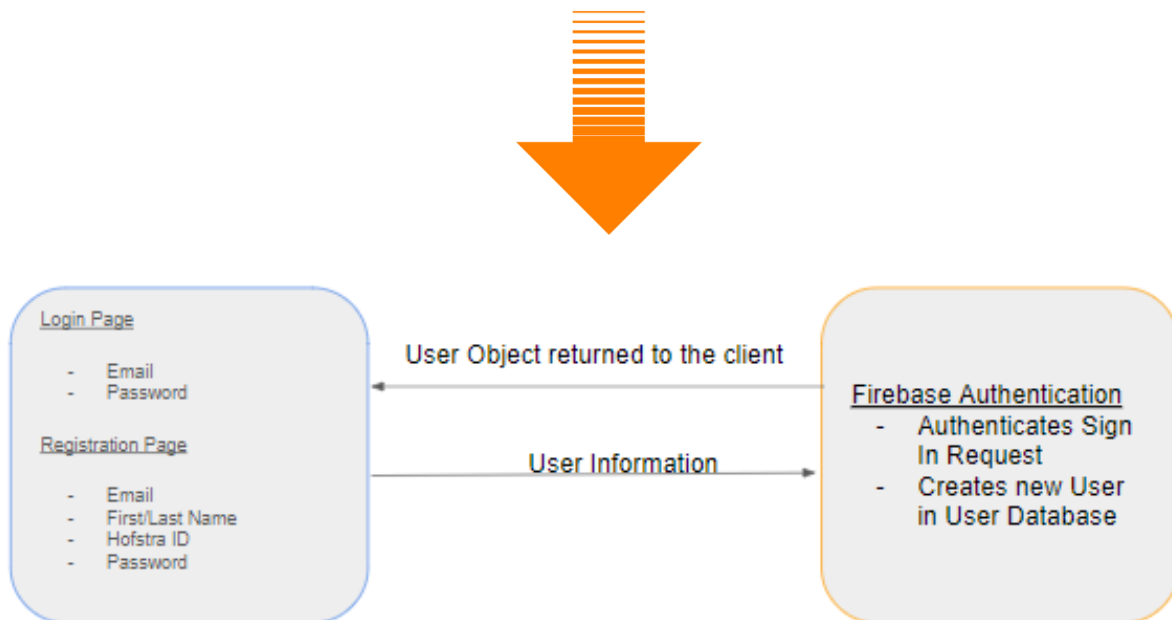


The HofSwap application uses a thin-client architectural model. All of the data for the application is stored in the Firebase server. Additionally, authentication and email notifications are also handled by Firebase. The client itself is where the user enters information like textbook data and personal information. This data is sent to the database. Processing, sorting, and other functions are not handled locally. Displaying textbooks requires pulling them from the database.

## Authentication Pages

Authentication pages consist of the login page and the registration page which will send the user information input by the user to the firebase authentication.

## Demonstration of authentication pages visualized:

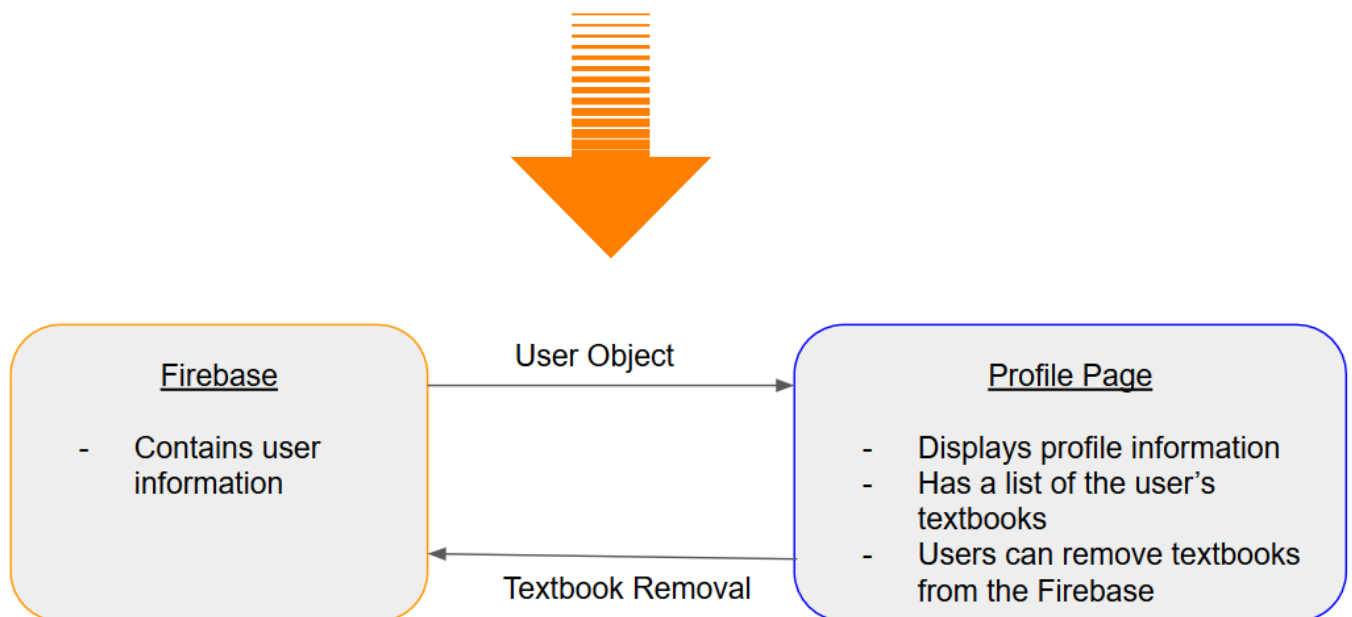


- The Login page is where the users can input their hofstra email and password and sign in to the Hofswap application.
- The Registration page is where the users can sign up for Hofswap using their Hofstra email, ID, and personal credentials.
- Data is sent to the Firebase upon the user clicking “Sign In” or “Sign Up”
- Firebase Authentication will authenticate the users sign in request from the login page. If they had not already made an account on the Hofswap application, the Firebase Authentication will take in the user’s information from the Registration page and create a new user in the user database. This is all processed through the Firebase API.

# Profile page

The profile page will show the user's basic information. The Profile page will interact with Firebase when a user needs to remove textbooks or a user object needs to be sent to the profile page.

Demonstration of profile page visualized:



## User Object

- Contains the users information (email, id number, textbooks)
- Created when the user signs in and destroyed when the user signs out

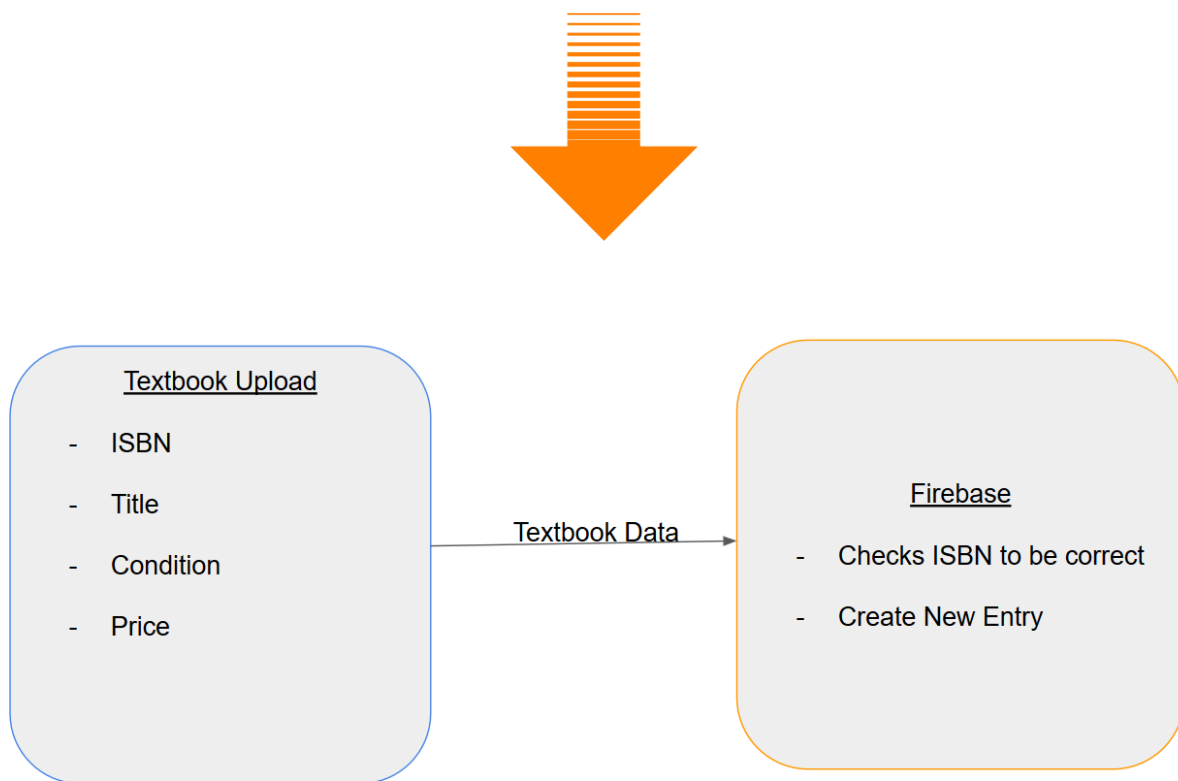
Inputs: The User Object is created in the Firebase when a user first creates their account. This information is passed from Firebase to the Profile Page when they successfully log in.

Outputs: Information is sent back to Firebase when the user removes a textbook from their profile. On the profile page, users can press a button next to each textbook to remove the listing. This is the only way a textbook can be removed.

# Textbook Upload Page

The user will input the fields that require indicated fields which consist of: the ISBN number, title, condition, price. This data will then be sent to the Firebase, creating a new listing. Here, it is associated with the user's ID in the Firebase. Data transfer to Firebase through the API is triggered when the user presses the upload button. The database of textbooks are not stored locally and instead are stored in the Firebase cloud.

Demonstration of textbook upload visualized:





# Textbook search

Users will input information regarding the textbook title, or ISBN number. The database will then be sent this information. After processing the input indicated by the user, the database will send back a list of textbooks correlated with the input. This is done through an API call to Firebase. Synchronous communication is not necessary for managing this list as textbook listings do not disappear when someone indicates their interest in a book.

Demonstration of textbook search visualized:

