System Design Document

Description:

BizReach is an online web platform which serves the purpose of connecting local mobile businesses of all sizes to provide services for customers in search of their services. BizReach's UI is primarily built using React framework and Chakra UI library. The backend is built using Express JS framework, Node.js and Firebase (Firestore DB and authentication). This document provides a high-level description of the components used to create the web app using CRC cards. It will also outline the system and environments interaction, brief overview of the architecture, and error handling strategies.

CRC Cards for Components:

Discover	
• Render the Discover component. • Include the GoogleMap component.	• GoogleMap (component)

GoogleMap	
• Render a map using the Google Maps API. • Fetch posts from customers using Firebase Firestore. • Handle marker clicks and show info windows with post details. • Handle the transfer of a selected post from the original poster to the current user.	• React Component • GoogleApiWrapper • Firebase (Firestore DB) • UserContext

Home	
• Rendering the Home component. • Display header section with the title, description and logo picture. • Provide a button to navigate around the website.	• React Component

Gig	
• Rendering a gig with its price and date.	• React Component

Login	
• Rendering the Login component. • Handle the user authentication using Firebase Auth. • Fetching user's data from Firebase (Firestore DB). • Handling user creation and user account type selection. • Rendering different UI based on the user's authentication and user account type (business or customer).	• React Component • UserContext • Firebase (Auth and Firestore DB) • Dashboard (component)

Navbar	
• Rendering the Navbar component. • Displaying navigation links to different pages. • Highlight the active link based on the current route.	• React Component, Router

Profile	
• Displaying user's profile information (name, number, and rating). • Allow user to edit, update and save their profile information. • Allow user to sign out of the web app.	• React Component • UserContext (Component) • Firebase libraries

Routing	
• Rendering Navbar at the top of the page. • Rendering routed components based on the current URL path. • Setting up routes using the Route and Routes components from react-router-dom. • Map specific paths to corresponding components (Home, Profile, Discover, Post).	• Navbar (component) • Route and Routes components from react-router-dom • Home (component) • Profile (component) • Discover (component) • Post (component)

UserContext	
• Create a UserContext object to provide a context for sharing user data across components.	• React

Post	
• Rendering the Post form and handle post submission. • Retrieving user information and validate user type. • Geocode the location entered by the user. • Add the post data to the user's profile in Firestore DB.	• React Component, Router • UserContext (component) • Firebase

System Interaction With Environment:

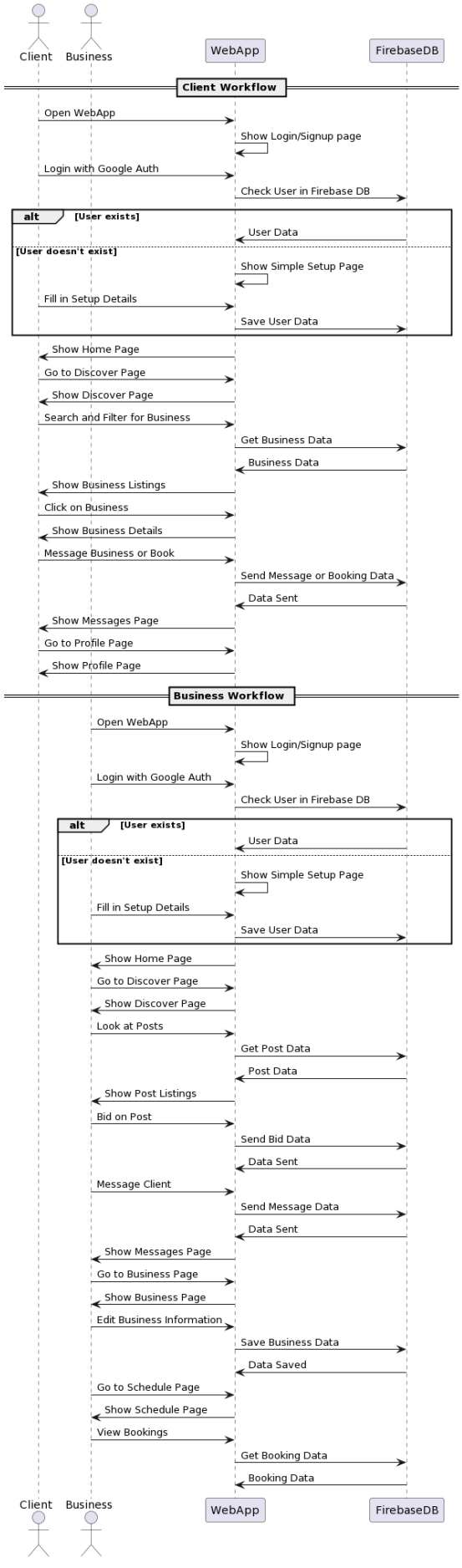
As an online web application, BizReach can be accessed by anyone using any operating system on a web browser such as Chrome, Mozilla Firefox, Edge, etc. The web app is built solely using Javascript by leveraging popular modern frameworks including React, and utilizing Node.js with Express JS in the back-end. It also incorporates various libraries such as Chakra UI, React libraries, and much more. BizReach employs Firestore DB, the noSQL database to store all data related to customers, businesses, jobs, etc. An assumption made is having a stable internet connection such that it will be sufficient to stay concurrent with the database. There is no required language compiler or virtual machine.

System Architecture:

BizReach follows the traditional monolithic architecture. The application is built with React on the front end with the assistance of Chakra UI for an aesthetic appearance. On the backend BizReach utilizes Node.js with Express framework. The frontend and backend communicate with each other through a REST api built with Express. Regarding where the data is stored, Firebase cloud services are utilized. The Firestore database is where all customer, business and job related information are stored along with any relevant data needed to be stored for the web app. Below is a workflow

This application is used as a service booking platform where the business reaches out to the user. In this application there are 2 types of users. A business user and client user. When you run the application it gives you a log in/ sign up page. Users can login through google auth. If a user doesn't exist in the firebase db they are prompted with a simple set up of their account such as name, client or business etc.
If user is a client:
There are 5 pages. Home page, Discover page, Post page, Messages Page and Profile Page
The home page is a simple homepage with our company's goal and information.

The Post Page is where a client can create a posting for a service they require. They must include specifics such as what service, price they are willing to pay, address of where service is required etc. Once they click post these posts can be seen by business (explained in business workflow).
The Discover page is where clients should be able to discover various businesses around. They can search and filter for Business they want to see there is also a map with locations of business on it. Clients should be able to click on a business from this page, view what they have to offer and details, and should be able to message the business or book with them.
The messages page is where users can see a list of all their conversations with business and can write new messages.
Profile page is a simple page with profile specifics of the client.
If user is Business:
There are 5 pages. Home page, Discover page, Messages Page, Business Page with a personal profile page inside it, and Schedule Page.
The home page is a simple homepage with the company's goal and information
The Discover page is where Business users should be able to look at all the postings of users who require their service, Also should be a map with postings on it. They should be able to bid on these posts against other businesses and message the user. Once a user accepts the bid, this post is booked and removed from the discover page.
The messages page is where Business users can see a list of all their conversations with clients and can write new messages.
Business page is where businesses can display what services they have to offer with prices, location, details etc. They can edit this page and customize it. Profile page is a simple page with personal profile specifics of the business.
Schedule page is a page with the business schedule with a calendar and times on it of all the bookings they have, booking specifics. Bookings are added here once the client makes a booking and the business can manually add and cancel bookings from here.


Client
Business
WebApp
FirebaseDB
Client Workflow
Open WebApp
Show Login/Signup page
Login with Google Auth
Check User in Firebase DB
alt
[User exists]
User Data
[User doesn't exist]
Show Simple Setup Page
Fill in Setup Details
Save User Data
Show Home Page
Go to Discover Page
Show Discover Page
Search and Filter for Business
Get Business Data
Business Data
Show Business Listings
Click on Business
Show Business Details
Message Business or Book
Send Message or Booking Data
Data Sent
Show Messages Page
Go to Profile Page
Show Profile Page
Business Workflow
Open WebApp
Show Login/Signup page
Login with Google Auth
Check User in Firebase DB
alt
[User exists]
User Data
[User doesn't exist]
Show Simple Setup Page
Fill in Setup Details
Save User Data
Show Home Page
Go to Discover Page
Show Discover Page
Look at Posts
Get Post Data
Post Data
Show Post Listings
Bid on Post
Send Bid Data
Data Sent
Message Client
Send Message Data
Data Sent
Show Messages Page
Go to Business Page
Show Business Page
Edit Business Information
Save Business Data
Data Saved
Go to Schedule Page
Show Schedule Page
View Bookings
Get Booking Data
Booking Data
Client
Business
WebApp
FirebaseDB

System Decomposition:

BizReach values users' privacy and makes sure to keep your account and data safe. We have employed Firebase's Auth service during the login process. The Auth service ensures valid users to login and handles authentication errors while giving the individual a clear message regarding their login credentials. Below is a list that identifies the role of each component in the high-level architectural overview.

1. Authentication Component:
    - Responsible for user authentication and authorization.
    - Integrates with Google Auth to handle user login and signup.
    - Verifies user credentials and grants access to the system.
    - Deals with errors such as invalid credentials, authentication failures, or unauthorized access.
2. User Management Component:
    - Manages user accounts and profiles.
    - Handles user registration, updates, and deletion.
    - Stores user-specific information such as name, email, role, and profile details.
    - Deals with errors such as duplicate accounts, missing information, or invalid user inputs.
3. Posting Component:
    - Enables clients to create service posts with specific details.
    - Stores post information such as service type, price, address, etc.
    - Allows clients to edit or delete their posts.
    - Deals with errors such as missing information, invalid inputs, or post deletion failures.
4. Discovery Component:
    - Facilitates the discovery of businesses and service posts.
    - Provides search and filtering functionality based on user preferences.
    - Displays a map with business locations and post markers.
    - Handles errors such as no search results found or map rendering failures.
5. Messaging Component:
    - Manages user conversations between clients and businesses.
    - Stores messages and tracks message history.
    - Enables users to send and receive messages within a conversation.
    - Deals with errors such as message sending failures or message retrieval issues.
6. Profile Component:

- Displays user profiles for both clients and businesses.
- Shows profile-specific information such as name, address, services offered, etc.
- Allows users to update their profile details.
- Handles errors such as profile updates failures or missing information.

7. Business Management Component:
    - Enables businesses to manage their services and business-specific details.
    - Allows businesses to add, edit, or delete services they offer.
    - Provides functionality to customize the business page and profile.
    - Handles errors such as service addition failures, profile customization issues, or service deletion errors.
8. Scheduling Component:
    - Manages the scheduling and booking of services.
    - Displays a calendar with available time slots.
    - Enables clients to book appointments with businesses.
    - Allows businesses to view and manage their bookings.
    - Deals with errors such as overlapping bookings, scheduling conflicts, or booking cancellation failures.