Design Document

02.02.2018

Team 26
Craig Ruble
William O'Malley
Tanner Ward
Ashwin Gokhale
Jialu Gu

Index

Purpose

A common issue in society today is that people require a good or service but may not have the money to pay for it. Although money is the driving factor of most sales, we believe that bartering is an effective means of payment. For example, many people possess assets of value that are not necessarily monetary, but do not have the means to easily liquidate them. These assets include goods such as computers, ladders, and suitcases, or services such as haircuts, car washes, and computer repair. With Bartr, users will be able to easily exchange these items for other goods or services in their area. This will allow people to get what they need by providing other means than just money.

We will do this by giving users an organized and manageable platform that will allow them to post listings, barter, and communicate with other users easily and efficiently. Bartr will be a profile oriented website that will give users the ability to manage their profile, and view others profiles to see what they have to offer and what their service rating is. This centralized marketplace will allow users to obtain what they need fast, effectively, and most importantly, without having to use money.

Functional Requirements

1. As a user, I would like to register an account on Bartr using Facebook authentication to be able to create a profile.
2. As a developer, I want to ensure that no two users register with the same email, so that the website doesn't have duplicate users.
3. As a user, I would like to login in to Bartr to access my profile, manage my listings, and search for goods in my area.
4. As a user, I would like to manage my account information, so that I can have the ability to change my information.
5. As a user, I would like to view other's profiles to view their listings and services they offer.
6. As a user, I would like to view my own profile, so that I can access my own listings and account information.
7. As a user, I would like to post a listing on Bartr, so others can see what I have to offer.
8. As a user, I would like to view a feed of listings in my area to browse services and goods I can exchange for.

9. As a user, I would like to rate my experience with other users, so I can tell if other users offer good services.
10. As a developer, I want to ensure users can't rate another user more than once, so that they don't inflate or deflate another user's rating.
11. As a user, I would like to search for specific goods in my area, so I can easily find what I am looking for.
12. As a user, I would like to set my viewing radius of other offers, so I can limit how far I am searching from me.
13. As a user, I would like to send another user a direct message, so I can easily communicate with the other person involved in the barter.
14. As a user, I would like to view my messages, so that I can read and manage conversation with other users.
15. As a user, I would like the ability to delete my listings, so that its show the barter has been completed or the offer exists anymore.
16. As a user, I would like the ability to edit any of my listings, so that I can correct any information that is wrong or edit any information on the listing.
17. As a user, I shouldn't be able to edit anyone elses listings, so that no one else can edit my listings and vice versa.
18. As a user, I shouldn't be able to remove other's listings, because that is beyond a general users scope on a social media site.
19. As a user, I should be able to report another user so that I can alert of any users or listings that aren't appropriate.
20. As a user, I want to be able to delete my account, so it doesn't exist on the website anymore.
21. As a user, I want the ability to choose what contact information I want to be public so I can protect myself from malicious calls/texts/emails not relevant to my listing.
22. As a user, I want the ability to tag the item/service I list and what I am looking for, so others know briefly what the listing entails to at a quick glance.
23. As a user, I would like to filter items in my listings feed by tag, so I can reduce the clutter of the listings to what I'm looking for specifically.
24. As a user, I would like to be able to send an email for support assistance, so I can obtain help with issues beyond a user's control.
25. As a user, I would like to view a support/troubleshooting page, so I can attempt to solve my problem with the common issues and solutions provided or find contact information for additional help.
26. As a user, I would like to request my password with the paired email when I forget my login, so I can log in to my account again.

27. As a developer, I would like to query user posts based on geolocation of each post, so I can display all listings around by a specific radius.
28. As a user, I would like to see another user's public contact information, so I that I know how to contact them for exchanges (unless hidden by user, must ask via direct message for public contact information).
29. As a user, I would like to apply to be a verified barterer if my trades and rating meet the requirement, so that others can visibly see, as a barterer, I can be highly trusted.
30. As a developer, I want to create a system for managing, tracking, and executing transactions between users so that I can automate the handling of transactions.

Non Functional Requirements

Design

1. As a user, I would like the web interface to be easy to navigate.
2. As a developer, I want to create a serverless application hosting using Firebase, which scales automatically based on network activity.

Performance

1. As a developer, I would want the client to receive the html within 1000 milliseconds from requesting it from the server.

Appearance

1. If time allows, I would like to ensure that the web pages are formatted for all browsers and mobile devices.

Security

1. As a developer, I would like to use the Facebook API to allow for secure logins.
2. As a developer, I would like to create Firebase functions that do not allow for reading and writing to the database unwarranted.
3. As a developer, I want to use a serverless application hosting, which doesn't have any security concerns that an actual server would.

Design Outline

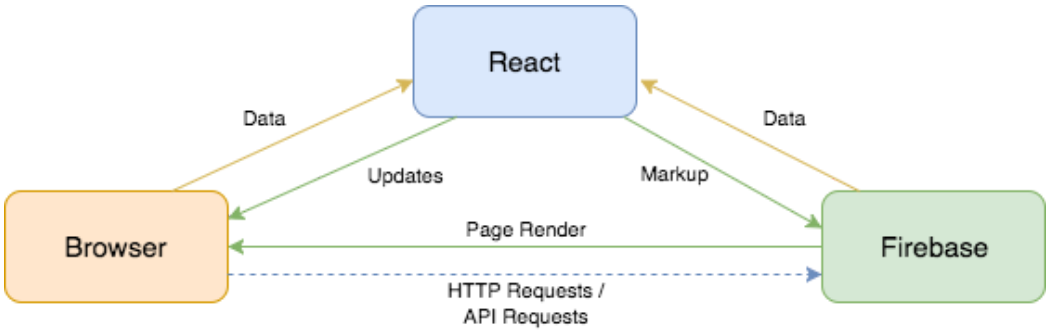
Our project is a service that provides an interface for users to barter their items and services. Our implementation will be an isomorphic web application; the front-end will access the server via two different methods. The first call involves a React component, returned as a JSX, which is then used to create real HTML output that is rendered in the browser. The second call communicates with the api and routes to appropriate Firebase functions.

1. React Web Application
    a. The website client will be preloaded HTML using a React component.
    b. The browser will request the component by events and retrieve the HTML to render.

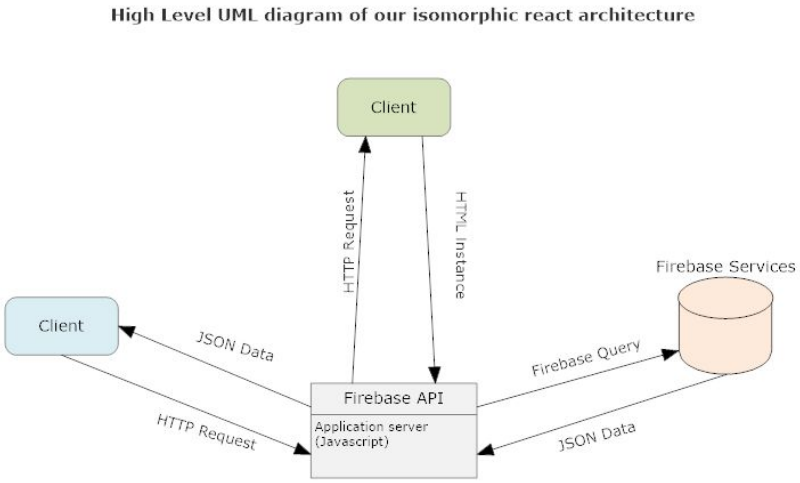
2. Firebase
    a. Firebase provides a NoSQL database that stores data in nested key-value pairs. Firebase provides a very helpful and versatile API that makes it unnecessary to have to implement your own database or API. Firebase handles all the components that usually come along with creating a backend for applications.
    b. Firebase Authentication allows you to confirm clients by using backend services, SDK's, and instant UI libraries. Allows for a variety of sign-in techniques that we can implement in our web application.
    c. Firebase messaging uses cloud functions and a centralized platform to allow users to be notified when they receive a message, and handle the exchange of messages between users.
    d. Firebase hosting allows you to rapidly and effectively send web applications and static content to a CDN with a single command.

3. Facebook API
    a. Users will be able to register their own personal account on Bartr through the use of the Facebook user authentication.
    b. Users can log back in to their Bartr account using the same valid Facebook info used when registering.

High Level Overview and Diagram



Our project will follow an isomorphic react model, which basically means that we get to use the same code on the client and on the server. We use firebase to handle things like database management, authentication, messaging, hosting, and routing. In our isomorphic react app, react will make markups to our node.js server which will send requests using the firebase api to firebase. React is also used to update and manage the browser and current state, as well as handle client requests. The figure above demonstrates our model.



For an isomorphic app the first time the user visits our site, the content is rendered and served to the browser by the server. This is done using server-rendering through Node.js and javascript. Once the data is retrieved for the first time from the firebase hosting services, the server will build the api and react will generate the markup using it renderToString function. The browser then parses the react DOM and receives an HTML element. This state will be stored using software like redux. After this page render is stored all updates and future renders are done through the browser.

Design Issues

Functional Issues

1. How should users register and login to Bartr?

   Option 1: Users can register and login using Facebook authentication

   Option 2: Users can register and login using the Google authentication

   Option 3: Users can register and login using the Firebase authentication only

   Decision: Have decided to use Facebook as a means of register and logging into Bartr. The reasoning for choosing Facebook as a means of registering and logging in is because Facebook, by a quick survey, is more common than having a Google account. Many people use other services for email.

2. How should we tag a listing's location for other users to filter as a distance from them?

   Option 1: Use the location stamp on a listing

   Option 2: Use a user's personal address from their profile

   Option 3: Use a provided zip code

   Decision: We decided to use GeoFire as a means of tagging a listing's location. It has the capability to store a user's location on their post through Firebase, which we have decided to use for this project.

3. How should the listings be presented on the main page?

   Option 1: Single news feed style that combines services and item listings

   Option 2: Pair of news feeds where one has items and the other has services

   Decision: We decided that a single news feed would reduce the clutter that two would bring. This means we will need to implement a toggle feature for how to filter the listings by item, service, or both.

4. How would you navigate the website?

   Option 1: Unified navigation bar on the side

   Option 2: Central location for all routes that go out from the home page

   Decision: We chose to work with a unified navigation bar on the side. This will allow users to access any other section of the website on any other page (profile, home page, about us, and help). Using the other option would require a user to go back to the main page and then click on another link rather than having all pages available on the side.

5. How should a user set their account information?

   Option 1: Pull contact information from their Facebook profile

   Option 2: Enter in their contact information when registering

   Decision: We decided to allow users to enter their preferred information while registering an account and also edit it from their profile page. This is so they can set their preferred contact information for other users on Bartr.

Non Functional Issues

1. How we want want to structure our backend?

   Option 1: Use Firebase functions and other functionality

   Option 2: Use a custom backend implementation

   Decision: We decided on using Firebase functions for increased security, as they will return simple information back to the front end to use when requested. This guarantees the user won't be able to communicate with the database to alter data.

2. What type of database fits our project the best?

   Option 1: MySQL relational database

   Option 2: Firebase NoSQL database

   Decision: Firebase provides a variety of services, one of them being a NoSQL database. We chose firebase because it provides an easy api that allows for easy deployment and manipulation of our database. Also firebase is a nested key-value pair database rather than a SQL relational database. The benefit of this is that all database entries are stored as JSON objects in key-value pairs. This allows for easy queries, faster retrieval of data, and an overall simplistic data management layout.

3. How do we want to load the front end?

Option 1: Static HTML pages with filled in information from the database

Option 2: Preloaded HTML, using React

Decision: Rendering preloaded HTML from React components on the server will decrease web page loading times compared to loading static HTML with variables filled in from database queries. We want users to experience smooth page loading.

4. How will the barter post feed be updated with new posts?

Option 1: Post request to server

Option 2: Query from the database

Decision: We chose to do a post request to the server because we delegate all database queries and mutations to the server, which reduces load on the client and protects against network failures while modifying the database.

5. How should a user be determined as a verified barterer?

Option 1: Automatic algorithm that checks all users at a frequency for a minimum overall rating and minimum total number of received reviews

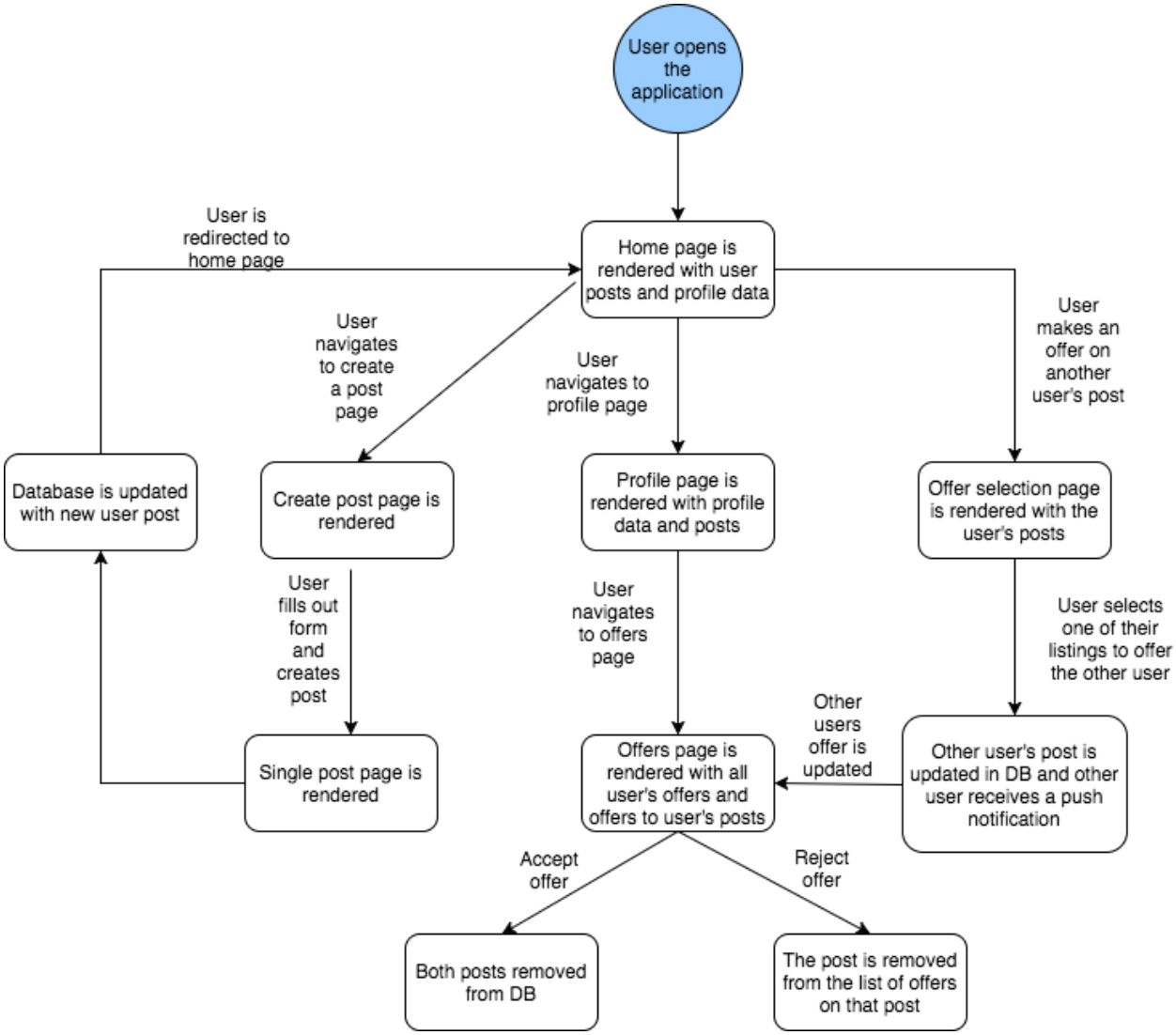
Option 2: Button to submit a request for an admin to review and accept/deny the request

Option 3: Button to submit a request that is followed up by an algorithm that will check for a minimum rating and minimum total number of received reviews

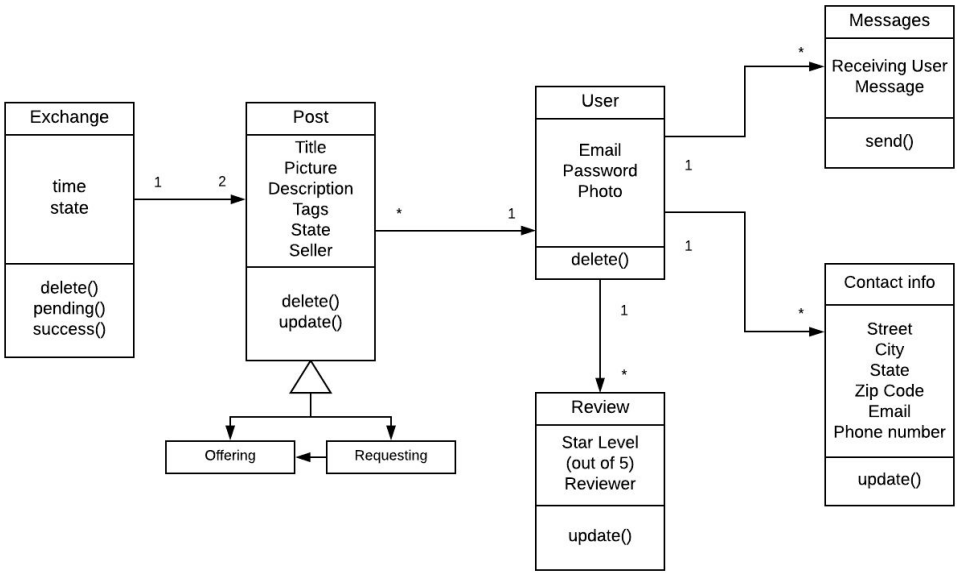
Decision: By means of ease and simplicity, we chose to have a request button where an algorithm will choose to accept or deny based on the required minimum rating and number of ratings.

Design Details

Activity/State Diagram



Class Diagram



Description of Data Classes and their Interactions

User:

- Represents each personal profile
- Created when a person registers an account on Bartr
- Contains contact info, has messages, has reviews, and has posts

Contact Info:

- Identifying information provided by the users
- Each user has an address, email, and phone number

Review:

- Each review represents how a user rates the receiving user
- Each review is out of five stars

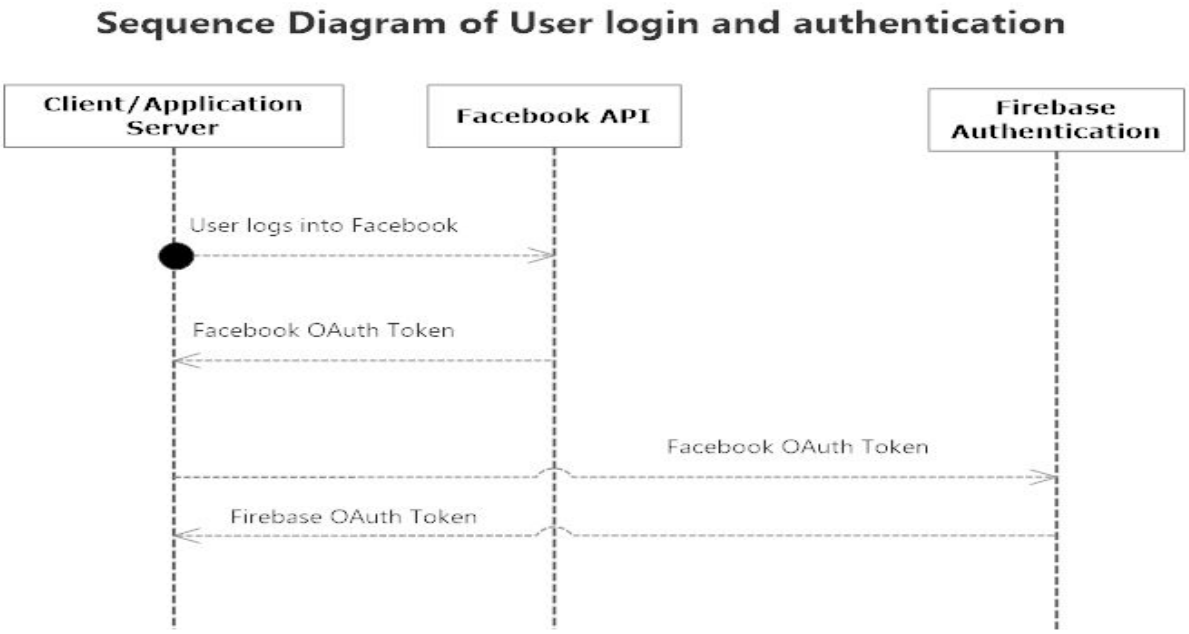
Post:

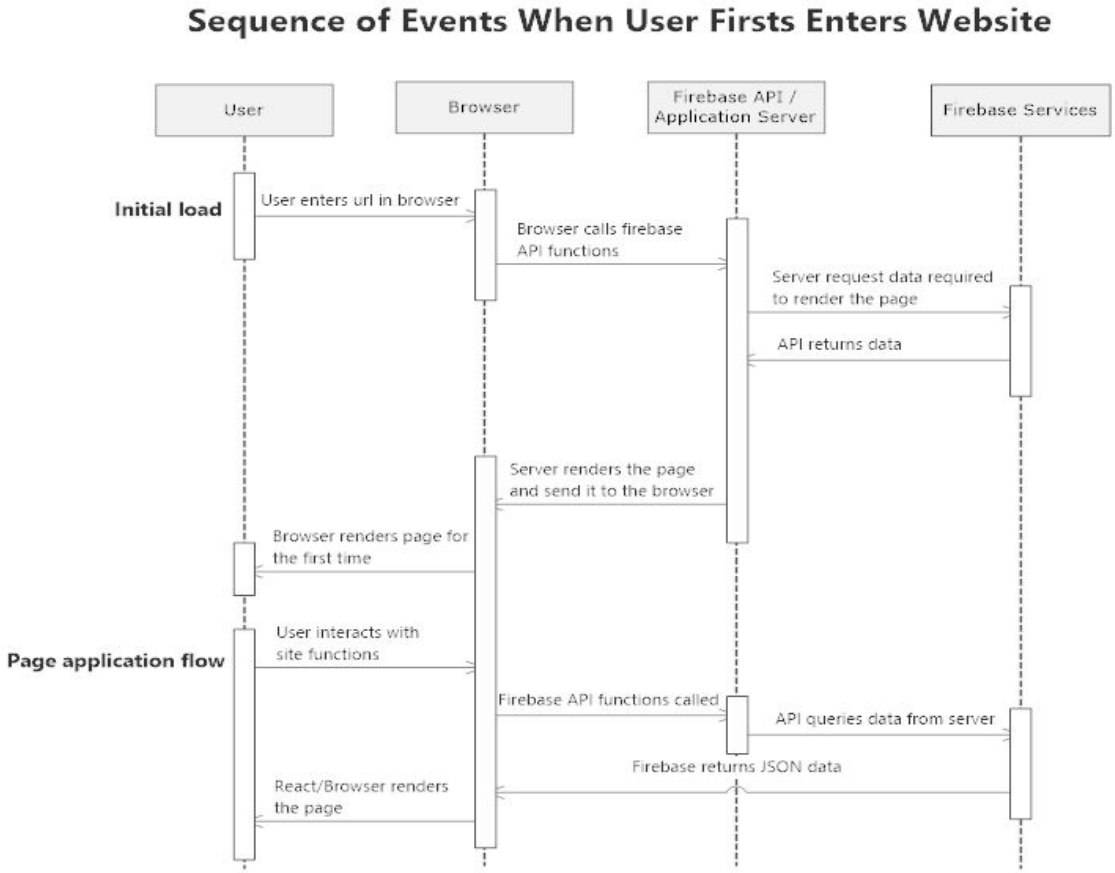
- Information about the products the users want to exchange.
- Each post has a title, description, tags, and state of post

Exchange:

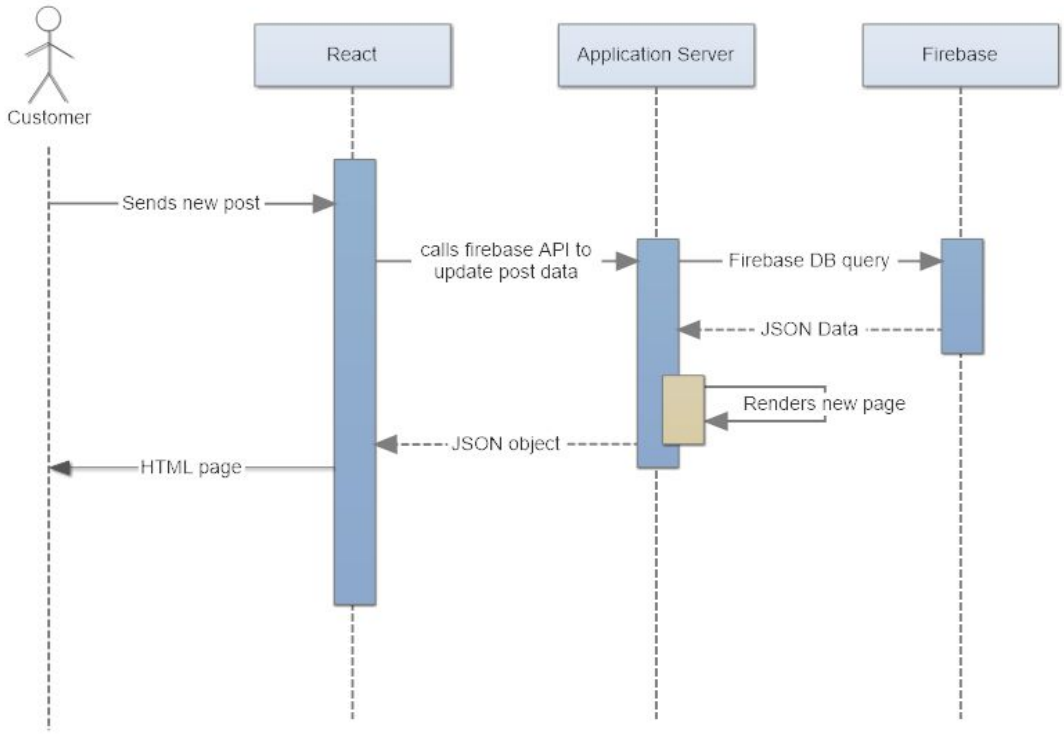
- Contents two posts being exchanged.
- Each post has two fields, a timestamp and state

Sequence Diagrams



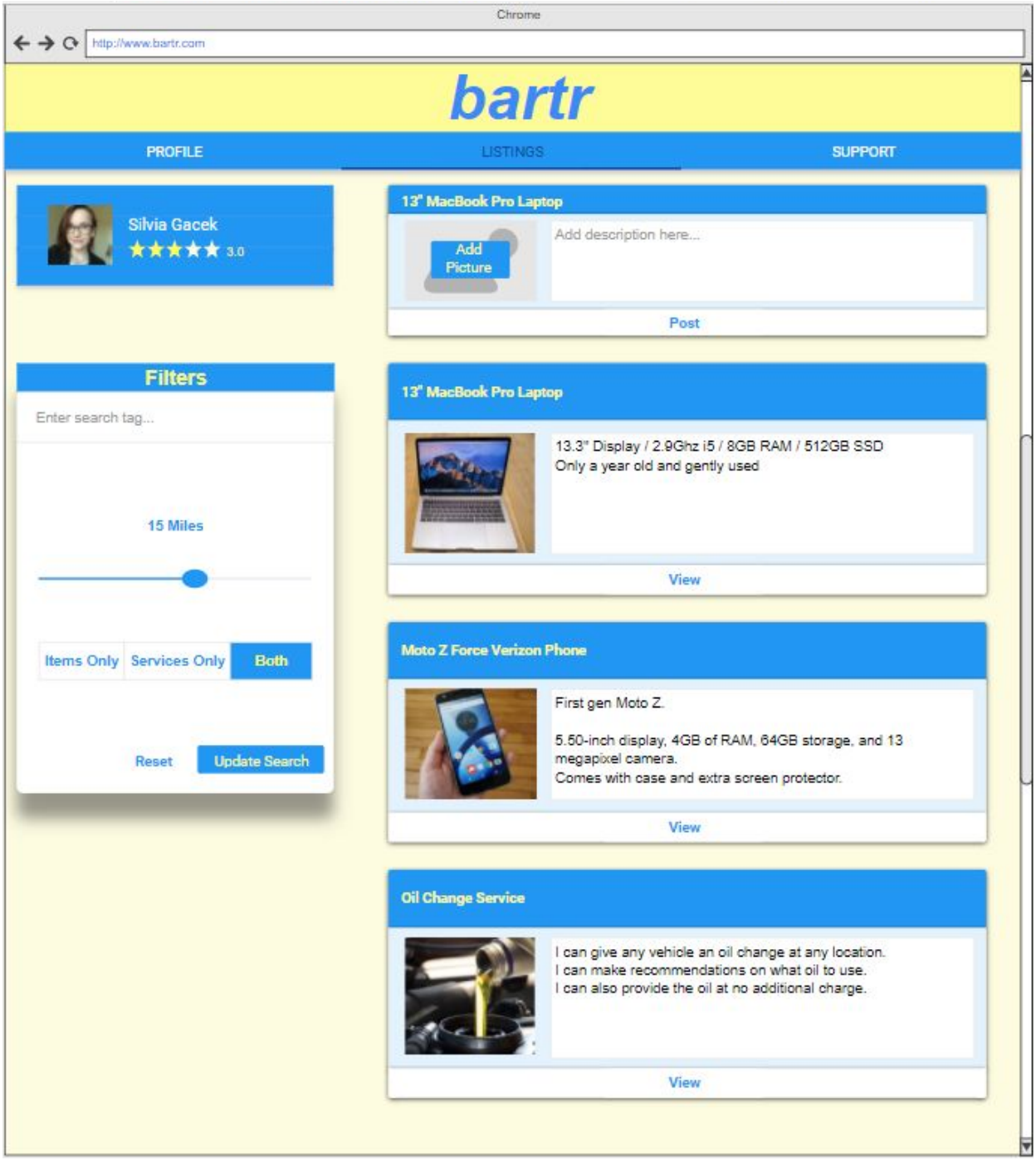


Sequence Diagram of User Making A New Post

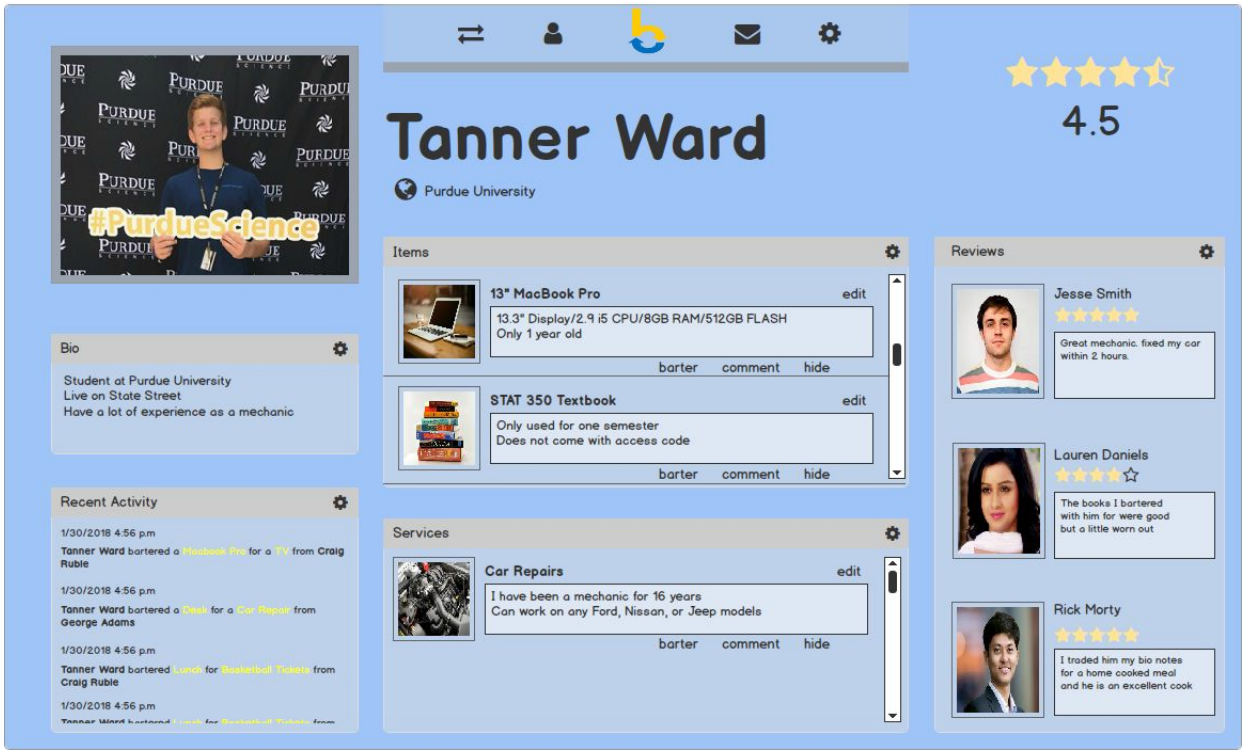


UI Mockups

Website Main Page

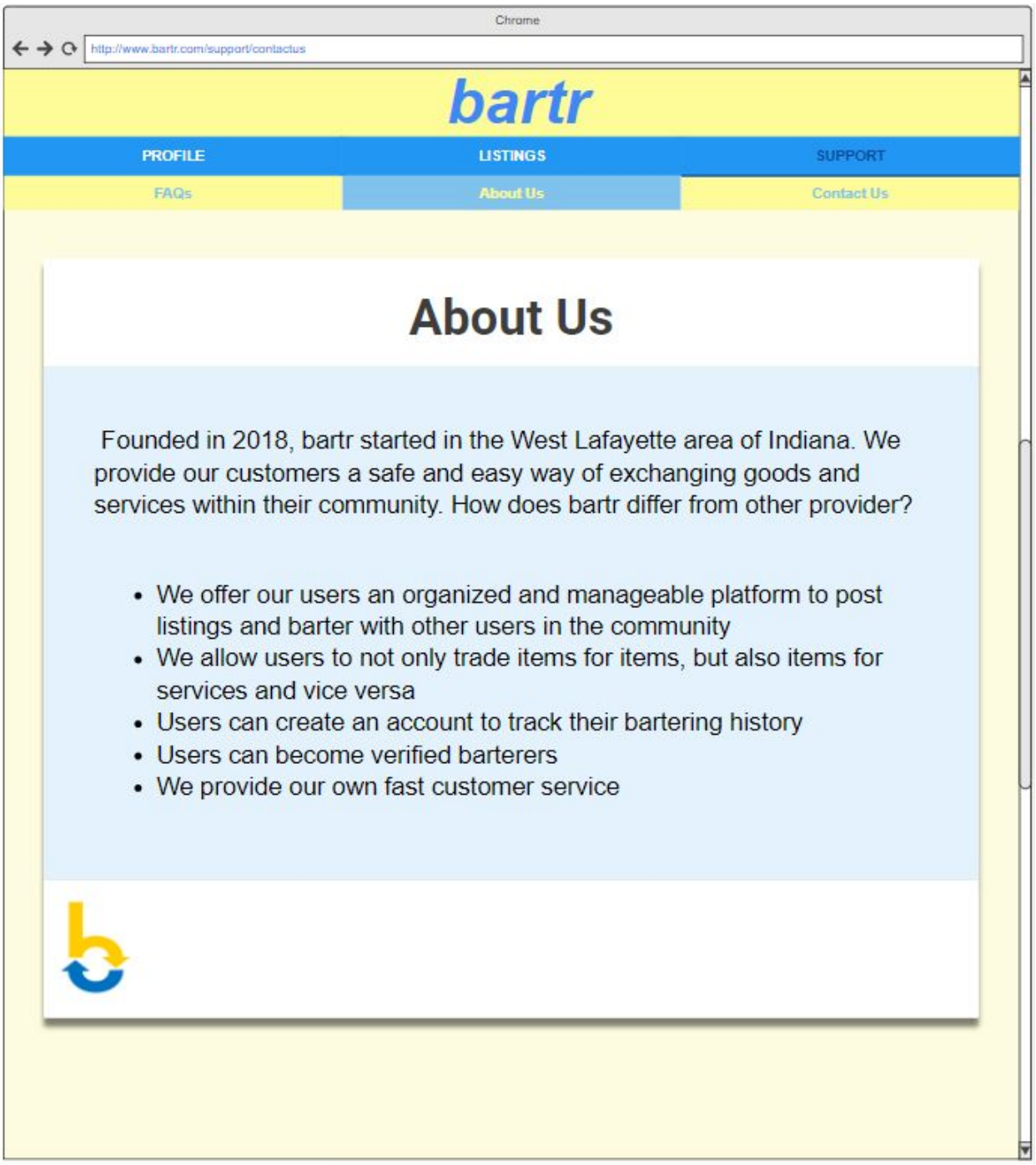


Website User Profile



Website Support and Contact Us

Website Support and About Us Page



Website Support and FAQs Page