rentnoww

Team 08

Software Design

CSCI-P465/565 (Software Engineering I)

Project Team

Zachary Zirin

Leo Chen

Joshua Baker

Devansh Jain

1. Introduction

The team will work in an agile environment following the component-oriented design approach to ensure that all the requirements from the customers are met and the website functions as expected.

1.1 System Description

The team plans to create a website that will narrow down the gap between the people who are looking to rent out entities and people who are looking for those entities. Essentially, the goal is to build a marketplace where people can rent entities from anywhere at any time. To meet these requirements, the team will be working in the direction to give users a seamless experience on the website through a responsive and user-friendly user interface. The team is set to make major features in the successive sprints over the period of next three months. After discussion within the team, the tech stack of the application will primarily be React JS for the front end, and Django for the backend.

1.2 Design Evolution

To ensure no customer is lost due poor website experience, and to give them the best rental experience, the team is extremely focused on easing the user management and rental management on the website. If a customer can login, register, add items to listing and rent them without facing any technical difficulties, the goal of this project will primarily be met. The focus on these functionalities has helped the team to divide the project into components and thus enhanced the understanding with respect to the users.

1.2.1 Design Issues

Since this project is a web-based project, the core requirement on every user end, may it be the developing team or the users, is a stable internet connection. Apart from this, there are components of the project which depend on the APIs by Google and other third-party vendors, which are expected to buy up 99.4% of the time. The team has implemented the authorization and authentication services offered by Auth0, which also enables people to login through their social media accounts like Google and Facebook.

1.2.2 Candidate Design Solutions

Stable Connection and Session: This web-based application will ensure that people are connected to the server by constantly checking the alive status of the session and the authorization session will still be active if the user has lost the internet connection.

Auth0: The team will be using authorization services by Auth0 and to enable login through social media accounts, the application will also be registered with social media like Facebook and Google.

Google APIs: Since at some point of time during this project, there will be map components which enable users to navigate through the renter's location, the application will make use of the API offered by Google.

Search and Recommendation System: To give our users a seamless search experience based on their interests, the search and recommendation system will make use of a third-party library called 'Algolia'.

Instant Messaging System - To deliver users with an end-to-end encrypted instant messaging system, the team aims to make use of a Firebase server.

1.2.3 Design Solution Rationale

The rationale behind using component-based design is that it helps the developers and the customer to have a result-oriented approach. Each sprint during development is then viewed as a release. It also helps the developers to manage and maintain the code quality by reusing the components, which does not lead to a lot of bugs with every release.

1.3 Design Approach

1.3.1 Methods

Describe the methods that are employed to capture your design. What techniques did you use (prototypes, object-oriented design, design patterns) and why?

The team will work in an agile environment and will schedule and plan tasks with respect to a sprint. The sprint is set to be two weeks long and at the end of each sprint, there will be a small demo which will be like a release prototype.

The sprint-wise goals have been defined in the project plan document and the plan for the first week is to release user authentication, login, and registration. The team also aims to deliver the feature to enable the owners of the products to add to the renting listing.

1.3.2 Standards

Describe any design standards to which your design adhered. These may include safety standards, operator interface standards, naming conventions, structure, and hierarchy of components, etc.

The team is determined to give users the best rental experience, by ensuring a seamless and responsive web layout, which is also user-friendly, and this is accomplished by implementing the front end in Material UI.

1.3.3 Tools

Describe any tools that you plan to use to assist you in developing the design and specify exactly what products will be generated by the tools.

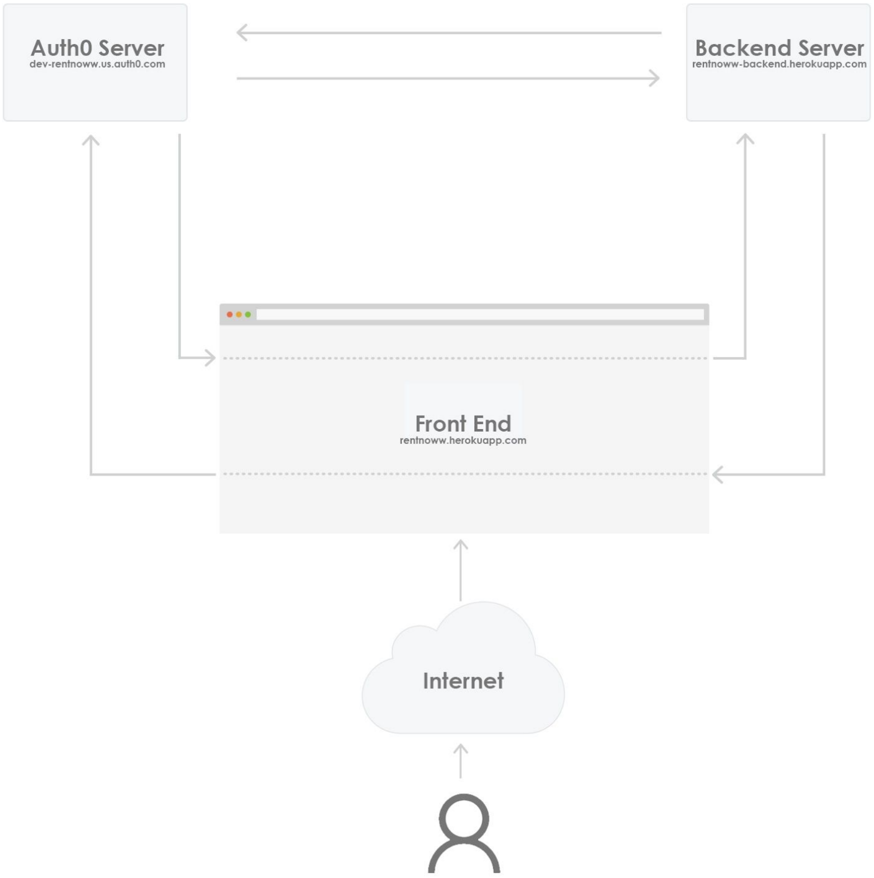
Front End: React JS and Material UI
Back End: Python (Django Framework)
Database: Postgres
Instant Messaging Chat: XMPP
Search and Recommendation: Algolia
Authorization: Auth0, Google and Facebook
Maps: Google Places API and Maps JavaScript API
Image Uploader Library

2. System Architecture

2.1 System Design

Provide a description of the high-level design of your system. Include a system diagram that captures the major components and the external interfaces. Describe the operation of the system (high-level) and the interaction between components.

All the users interact with the web application using the front end built in React. The react components are rendered using Material UI and, front end is also integrated with a handful of APIs like Google Maps, Google Places to give users a seamless experience.
During the login, registration step, the user interacts with the front end and then the front end sends http requests to the Auth0 server, and that responds with an access token which is valid for 24 hours. After that, all the user interactions with the front end are sent to the backend to update/delete/add data. All the requests are also authenticated by the backend and Auth0 server collectively.



2.2 External Interfaces

Describe in detail all external interfaces that interact with your system. Be sure to include a description of the interface and a summary of its relationship with your system. Be sure to include specifics about the communication between your system and the external system including protocols, data structures exchanged, and timing/handshaking patterns.

Google Maps for JavaScript API and Google Places API are used for the purpose of integrating maps to the web application. For authentication, the project uses the Google OAuth API and Facebook OAuth API.

Firebase standards are used to implement the feature of instant messaging where users will be able to interact with each other using a chat.

To effectively implement the search and recommendation feature according to the user's interest, the project will make use of Algolia, which helps create relevant, scalable, and lightning fast Search and Discovery experiences.

3. Component Design

The Component Design section details the proposed design of each system component. A system component is a functional partition of the system. Components may be organized as you see fit - a component may be a collection of objects, or a single object. However, a system must be composed of multiple components (that is, a system cannot be one component). The layout of this section is at your discretion, but please include the following (at a minimum) information for each component:

- **Component Name**
  Authentication Component (Login and Registration)

  Profile Component(Creating and modifying the Profile)

  Listing Component(Adding request to the list for admin to confirm)

  Renting Item Component(Items for renting)

  Cart (Adding Items to the Cart or so called a "wishlist")

Chat Bot (Answer users questions and help them navigate through)

Chat Feature (Users can interact with each other)

- **Component Description**

  Authentication component primarily aims to verify the identity of an individual who is trying to access the product. A user can interact with a web application using multiple actions. Access to certain actions or pages can be restricted using user levels. This component helps the project in controlling user access via assigned roles & privileges. On login/registration, the http request is sent to the Auth0 server and if verified server responds with an access token, which is then used to authenticate all other requests that the user sends to the backend.

  Profile Component basically aims to store the information of every user of our product. It includes basic information such as address, date of birth etc. Both user and admin are able to modify it and the information will be stored in our database.

  Listing Component is for storing the requests which are sent by users. While users are trying to post items or services they want to rent out or sell, the items have to be confirmed by the admin, and while they are waiting for the confirmation, they will be stored temporarily in the list.

  Renting Item Component is for storing the information of the items that is posted on the website. It includes every details of items, including name, id, address, category, reviews, rating and etc. After the admin confirms the request that renter sent, the information of the items will be removed from the list and will be stored in our database.

- **Responsible Development Team Member**

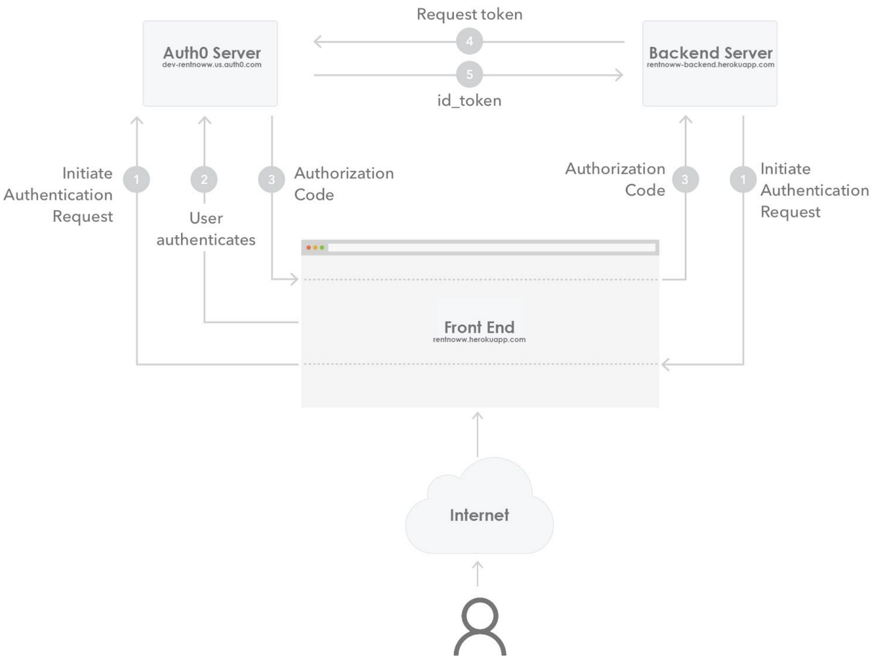
  Devansh Jain (Authentication)

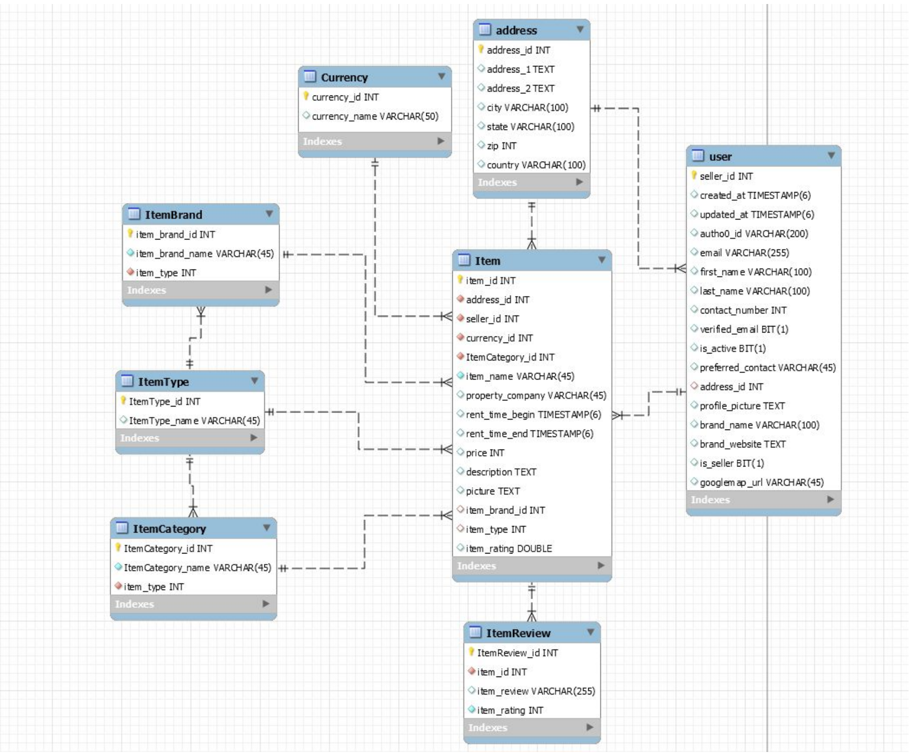
  Zachary Zirin(Profile, listing)

  Josh Baker(Profile, Renting Item)

  Leo Chen(Profile, Renting Item)

- **Component Diagram**





- **Component User Interface**
  The UI for authentication component an HTML page where the users enter the details.

  The UI for profile and renting items components are also HTML pages where users can just enter details of themselves and items.

- **Component Objects**
  Describe the objects/classes that comprise this component. Provide a listing of expected data members and methods for each class. Note in the description if a given object/method does any of the following (Note: Some of this information may be captured in class diagrams or other parts of the design - simply be sure it is included somewhere):
    - Authentication component is called whenever the user tries to access the private pages on the website or whenever users want to sign into the application.
    - Authentication component creates a form object which is sent to the Auth0 server along with a code and state which is generated by the application on the front end.
    - The Auth0 server responds with an access token which is stored in the cookies for 24 hours.
    - Profile component is called whenever the user signs up the application, and the information about the profile will be sent to our database and stored.
    - Listing component is called whenever the user tries to post any items, the system will temporarily store the information of it in the list until it is confirmed by the admin.
    - Item renting component is called after the request in the list is confirmed by the admin, the item will be posted on the webpage and the information of the item will be properly stored in the database. Later it will be used for searching as well.
- **Component Interfaces (internal and external)**
  Communicates with the Auth0 server using HTTP requests.
- **Component Error Handling**
  In case a user enters incorrect login credentials, the page responds with a prompt that the login details are incorrect and asks the user to re-enter them. If the users enter incorrect information of themselves and items, or there is a missing information that needs to be filled, the page will respond and ask them to fill it out correctly.

Revision History

Revision	**Date**	**Change Description**
1	October 18, 2020	Updated Database Settings
2	November 01, 2020	Added Cart Component
3	November 15, 2020	Added Chat Bot and Chat Components Chat will now be implemented using Firebase and not XMPP

Page Author/Creator: Zachary Zirin, Leo Chen, Joshua Baker, Devansh Jain

Last Modified: 11/15/2020