CSCC01 eBeauty
System Design Document

Lim Lei,
Allen Chien-Luan Lo,
Jason Chen,
Jin Ting Zhou,
Junxun Liu,
Hyun Woo Shin,
Yuehai Zhou

June 10, 2022

Table of Contents

CRC Cards

Frontend

Class: /src/components/SignUp.js	
Responsibilities	**Collaborators**
• Allows users to sign up/create new accounts for the app.	• /css/SignUp.css

Class: /src/components/Login.js	
Responsibilities	**Collaborators**
• Allows users to login to their accounts. • Receives user credential inputs for verification.	• /css/Login.css

Class: /src/components/ViewProfileProvider.js	
Responsibilities	**Collaborators**
• Allows users to view the profiles of service providers.	• N/A

Class: /src/components/ProfileProvider.js	
Responsibilities	**Collaborators**
• Allows service providers to edit their profile data. • Allows service providers to edit their account information.	• N/A

Class: /src/components/ViewProfileProvider.js	
Responsibilities	**Collaborators**
• Allows users to view the profiles of service providers.	• N/A

Class: /src/components/SearchPage.js

Responsibilities	Collaborators
• Displays a search result page after users provide search input • Allows users to search again and use filters with different attributes	• /components/ServiceDropdown.js • /components/ViewProfileProvider.js

Class: /src/components/ServiceDropdown.js

Responsibilities	Collaborators
• Allows users to specify the service they are searching for.	• /components/SearchPage.js

Class: /src/App.js

Responsibilities	Collaborators
• Manages the components of the app as a root component. • Used to render the app.	• /src/index.js

Class: /src/components/Home.js

Responsibilities	Collaborators
• Displays general information about the app. • Provides a quick search option. • Attracts potential users to sign up to the app.	• N/A

Class: /src/component/FindJob.js

Responsibilities	Collaborators
• Allows service providers to view the job page and access to their register page.	• /components/Animation.js

Class: /src/component/SignUpProvider.js

Responsibilities	Collaborators
• Allows service providers to fill in their information and register to the app as a first step. • Collect basic information about service providers.	• N/A

Class: /src/component/SignUpProviderTwo.js

Responsibilities	Collaborators
• Allows service providers to view the job page and access their registration page as a second step. • Collects additional information about service providers.	• N/A

Class: /src/component/ScrollToTop.js

Responsibilities	Collaborators
• A helper class which resets the position of client by setting the position of viewer's location of page to top after switching pages.	• N/A

Class: /src/component/Navbar.js

Responsibilities	Collaborators
• Allows users to have access to homepage, register the different purpose's accounts and login	• N/A

Class: /src/component/NotFound.js

Responsibilities	Collaborators
• Used to redirect users back to homepage when they encounter invalid URLs.	• N/A

Class: /src/component/AddressPage.js	
Responsibilities	**Collaborators**
• Allows users to enter a new address or change address.	• MapContainer.js • Footer.js • AddressForm.js

Class: /src/component/MapContainer.js	
Responsibilities	**Collaborators**
• Used to connect to google map API and display a map.	• N/A

Class: /src/component/AddressForm.js	
Responsibilities	**Collaborators**
• Used to connect to create a form to handle user input for addresses.	• N/A

Class: /src/component/index.js	
Responsibilities	**Collaborators**
• Used to render the app by combining all the components inside App.js into one final web app.	• App.js

Backend

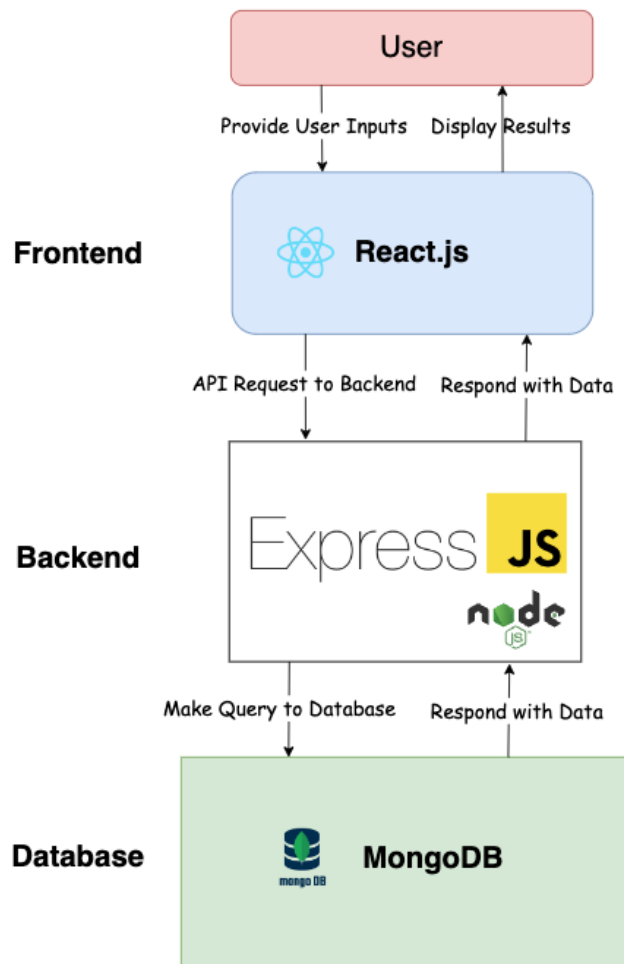| Class: User.js	
Responsibilities	**Collaborators**
• Define User data objects depending on the types of user and convert user inputs into appropriate JSON objects.	• Server.js • /components/

Class: Database.js	
Responsibilities	**Collaborators**
• Initializes a database and executes the initial setup for proper operations. • Performs CRUD (Create, Read, Update, Delete) operations. • Interacts with the server-side pointer to receive user requests and send responses.	• Server.js

Class: Server.js	
Responsibilities	**Collaborators**
• Initializes and runs Express.js on Node.js. • Manages REST API methods such as POST, PUT, DELETE and GET to communicate with DB and send the data to the frontend. • Communicates with both the frontend and the backend to establish a data flow and a stable connection.	• Database.js • /components/

System Architecture Design

MERN Stack

Our web application follows a 3-tier architecture which is composed of frontend, backend, and database using JavaScript and JSON. React.js is responsible for the frontend operations of the app while Express.js and Node.js perform server-side operations. To store the data from the app, we use MongoDB, a document-oriented NoSQL database. This 3-tier architecture is also known as MERN stack which is an abbreviated form of **M**ongoDB, **E**xpress, **R**eact, and **N**ode.

System Flow Decomposition

Users first interact with the UI components of React.js in the web browser. They can browse the different pages of the app and provide inputs for different operations such as login/register, search, comments, or reviews. Since the web page is served by Express.js running on Node.js, whenever users provide inputs, the server-side applications can collect them and send them to the MongoDB database. Also, whenever users request data from the database, Express.js is responsible for retrieving the data from the database through querying.