CS Software Project
A dynamic referral search engine for Realtors and their clients

Jordan and Reece

2026-01-21

1 Client Information

By sharing this client information and the rest of this document, you are stating that this client has provided this project as something they want (not something you created and asked if they wanted), and that they are interested in having you complete this project for your capstone.

- Client name: Rodney Fontil
- Client title: Real Estate Agent
- Client email address:
- Client employer: Hearth Realty
- How you know the client: Cold outreach

2 Project Description

2.1 Overview

Real estate agents struggle to stay connected with clients after closing. Rodney reports that 80 percent clients don't return to their realtors when selling their next home or to refer friends. It matters to him because this is a missed opportunity to nurture that client relationship and for repeat business. The root cause is that agents fall out of clients' minds once the transaction ends.

2.2 Key Features

[At this point you should have a basic understanding of your client's needs. List out the key features of the software system the client wants you to build.] These are from most key to least important.

- User types: There will be 4 types of users, agents, clients, admins, guests. User relationships
- Clients, Agents and Organizations: Clients will be invited and associated with specific agents on the platform for more exclusive referrals, and even access to the agent's broader organization.
- Guests: Guests can view search engines agents and their organizations share to the public.
- Admins: Admins can manage referral databases for their branches. This is for the organizations who want all their agents to use the same database.
- Client interfaces: These interfaces are where users like clients can query their agents' referral engine, find public referral engines, and manage chats with their agent.
- Agent and Admin interfaces: These interfaces will house all the functionality for agents to invite clients, add and update their referral databases and chat with their clients.

- Natural language search: Clients type questions conversationally ("good pediatrician near downtown?") rather than filtering through categories. Here we can use natural language processing and AI to make the user search experience more intuitive.
- Analytics dashboard: Shows agents what types of referrals clients request most often, helping them build stronger local networks
- Branch collaboration mode: Multiple agents in the same real estate office can pool their recommendations into one shared database, expanding the value for all their clients
- Lead tracking: When clients use the platform, agents see which services are being searched, helping them identify clients who might be ready to sell (e.g., someone searching for major contractors might be preparing their home for market)
- Agent / client chats: When questions can't be supported by a database query, clients can send their Agent a message on the platform.

2.3 Why this Project is Interesting

Because we get a chance to work on a real problem. We also get to tackle it using AI technology as it makes what would just be a simple search engine into a more dynamic intuitive experience for the end user — clients.

2.4 Areas of CS required

[What subfields of computer science seem most likely to be relevant to your project? A capstone must involve multiple.]

- Databases: Store agents' referral catalogs with business details, categories, and metadata. Support efficient queries across potentially thousands of entries as the platform scales
- Machine Learning: Power the natural language search using text embeddings that match client questions to relevant referrals, even when exact keywords don't match
- Object-Oriented Programming: Structure the codebase with classes for Users (Agents/Clients), Referrals, Searches, and Analytics to maintain clean, extensible code
- Software Architecture: Design the system to handle the branch mode feature where multiple agents share one database while maintaining separate analytics, requiring careful data separation
- User Experience (UX): Create an interface simple enough that clients will use it spontaneously when they need a referral, not just when the agent asks them to
- Cybersecurity: Protect sensitive search data (e.g., if a client searches for divorce lawyers or therapists, that information must remain private)
- Ethics: Give clients control over which searches are visible to their agent, balancing privacy with the agent's business need to understand client needs

2.5 Potential Concerns and Questions

[Is there any aspect of this project that makes you unsure if it will work, either due to your own interests/background, or that you aren't sure if it fits the requirements? Are there questions you have about this project that you want instructor feedback about?]

Data privacy is the primary concern. Clients might search for sensitive services (therapists, family lawyers, medical specialists). The platform needs to let clients mark certain searches as private—hidden from their agent's analytics dashboard—while still returning helpful recommendations. This creates a technical challenge: how do we provide value to agents through search insights while protecting client privacy when

they need it? We'll need to determine the right default setting: Are all searches visible to agents unless clients opt out? Or is everything private unless clients opt in? The answer affects both the platform's value proposition to agents and clients' trust in using it.

3 Requirements

3.1 Non-Functional Requirements

[Non-functional requirements are just as important as functional requirements. Dont forget to specify them.]

ID	NFR Title	Category	Description
01	mini RAG databases	Usability	For the natural language search we have to be able to let the users modify, update and set up these retrieval databases so that in the platform as clients ask questions we can simply make calls to the agents personal API endpoint and get results
02	private messages	Security	all messages clients, or quests share (even if saved) should be kept visible only to the sender since some questions could be personal. The only attribute that should be visible is the category of the query for analytical purposes.
03	RAG database management	Usability	Our system should be able to dynamically set up these rag system via a API. For this we can use AWS.
04	Branding customization	User-experience	Branding should be flexible. an agent may switch brokerages overtime, update their logo etc. Those small changes should be able to be done in the platform.
05	Search back-ups	User-experience	A agent's data base may not have all the answers to a users question. In this event we should have a back up to pass the users question off to google API, or to send a inquiry to their Agent.

Table 1: Non-Functional requirements

3.2 Functional Requirements (User Stories)

[In CS482, all functional requirements are written as User Stories. In CS496, some projects may use a different template to write the requirements. The table below is an example of writing the Stories. Adapt accordingly to different templates or if you want to display more info.]

ID	Story Title	Points	Description
01	Request Personalized Referral	4	As a Client, I want to message my agent when no referrals matc my search so that I can request a personalized recommendation.
02	Search Referrals with Natural Language	8	As a Client, I want to search for referrals using natural languag questions (e.g., "Where can I get my roof fixed?") so that I ca find relevant recommendations without needing to know exact cat egories or keywords.
03	Basic referral search	2	As a Client, I want to search my agent's referral database fo specific entries so that I can quickly find the recommendations need.
04	Save search history	5	As a Client, I want to save searches I have with my agents databas so that I can quickly refer back to old questions I had.

05	Invite clients	2	As an Agent, I want to send platform invites to my clients so tha they can access my exclusive referral list and communicate wit me directly.
06	client search analytics	5	As an Agent, I want to view and filter through analytics on m clients' searches so that I can understand their needs and identif gaps in my referral database.
07	Referral management	5	As an Agent, I want to create, and delete referrals so that I ca keep my recommendations current and relevant for my clients.
08	Referral management — edits	3	As an Agent, I want to edit referrals so that I can keep my rec ommendations current and relevant for my clients.
09	Guest searches	3	As a Guest, I want to access public referral lists from agents an branches so that I can find local recommendations without creat ing an account.
10	Exclusive access to referral lists	2	As an Agent, I want to restrict my referral list to only my client so that I can provide exclusive value to the people I serve.
11	Log in and out	2	As a User, I want to log in and out of my account so that I ca securely access my agent's personalized recommendations.
12	Client account management	2	As a Client, I want to update my profile information and prefer ences so that my agent can provide more relevant recommenda tions.
13	Admin — create a branch	5	As an Admin, I want to create a branch profile with our brandin and office information so that clients can identify and connec with our team.
14	Add new agents to a branch	2	As an Admin, I want to add agents to my branch so that they ca manage their own client relationships and referral lists.
15	Branch database management	8	As an Admin, I want to create, update and delete a master referra database for my branch so that all agents have access to our office' recommended vendors and services.
16	Google Search back ups	3	As an Agent, I want the system to provide Google search result when my database has no matches so that my clients always re ceive helpful information even for referrals I haven't added yet.
17	Message inbox	5	As a User, I want to access an inbox so that I can receive an manage invites, messages, and notifications in one place.
18	Search filters	3	As a Client, I want to filter my searches by category, location, o service type so that I can find the most relevant referrals for m specific need.
19	Custom Referral tags	5	As an Agent/Admin, I want to add, update, and delete tags o my referrals so that clients can filter results and find what the need more easily.
20	Search suggestions	3	As a User, I want to select from suggested search questions so tha I can quickly find common referrals without typing.

Table 2: Functional requirements as User Stories.

4 System Design

4.1 Architecture

We are following the MVC (Model View Controller) architecture.

4.2 Diagrams

[CS482, on sprints/iterations 2-3, you need to create and update a diagram (check the assignment for which type of diagram). On CS496, since before sprint/iteration 1 you should have a class diagram and keep it up-to-date.]

4.3 Technology

- Bubble.io — rapid front-end development and database functionality
- JavaScript / react and python for coding
- Docker to package our code for deployment
- Render or AWS for deployment

4.4 Coding Standards

[Are your team going to follow any coding standards? For example, using a naming convention for Database tables (like only singular lowercase names). Another example, only allowing code with unit tests and above 60% coverage to be committed (good convention since testing is going to be evaluated). If you need inspiration to define your coding standards, the Extreme Programming approach has a set of coding, design, and test rules.]

Our team will only use a naming convention for database tables consisting of lowercase and underscores. Our code will test with Jest and require above 90% coverage to be committed.

4.5 Data

[What is the main structure of your data? In SQL-like databases, this would be the planning of the main tables, their attributes, and interactions with other tables (basically an ER diagram). In NoSQL databases, this would be the main collections and general attributes of the JSON you will store in each collection.]

5 UI Mocks

5.1

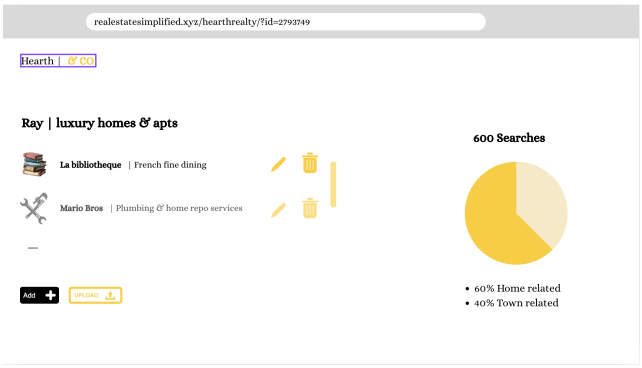

Figure 1: Mock up for the Agents view, all their referrals and a view to see analytics

5.2

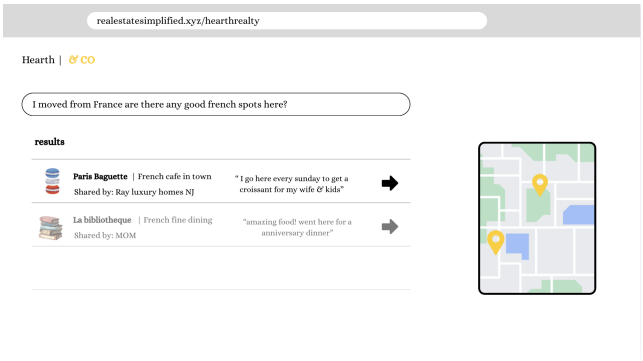

Figure 2: Mock up for how the search interface could look like

5.3

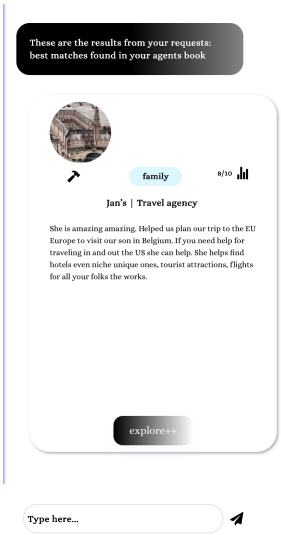

Figure 3: Another Mock up for how the search interface could look like, more like a AI chat user interface

6 Iterations

6.1 Iteration Planning

[In CS496, you plan all iterations beforehand. In CS482, you update the planning here at each iteration.]

Iteration	Dates	Stories	Points
1	02/10/2026	07 Referral Management, 08 Referral Management - Edits, 04 Basic Referral Search, 11 Log In and Out ,04 Save search history	14
2	02/24/2026	03 Natural Language Search, Guest searches, 10 Exclusive Access to Referral Lists, 12 Client Account Management, 16 Google Search Backups, 18 Search Filters	21
3	03/17/2026	13 Admin — Create a branch, 14 add new agent to a branch, 15 Branch database management, 17 Message inbox, 05 invite clients	22
4	03/31/2026	01 Ask your agent for more information, 02 Request Personalized referrals, 06 Client Search analytics	18
5	04/14/2026	19 custom referral tags, 20 Search suggestions, 01 Personalized Referrals	11
total 86			

Table 3: Iteration Planning for Incremental Deliveries

6.2 Iteration Planning

6.3 Iteration/Sprint 1

6.3.1 Planning

6.3.2 Work Done

6.3.3 Testing Coverage

6.3.4 Retroespective & Reflection

6.4 Iteration/Sprint 2

6.4.1 Planning

6.4.2 Work Done

6.4.3 Testing Coverage

6.4.4 Retroespective & Reflection

6.5 Iteration/Sprint 3

6.5.1 Planning

6.5.2 Work Done

6.5.3 Testing Coverage

6.5.4 Retroespective & Reflection

6.6 Iteration/Sprint 4

6.6.1 Planning

6.6.2 Work Done

6.6.3 Testing Coverage

6.6.4 Retroespective & Reflection

6.7 Iteration/Sprint 5

6.7.1 Planning

[Which stories did you plan for this iteration/sprint. Add the total points for this plan. You can also explain the reason behind your planning, and what major feature(s) your team is focusing on delivering by

completing these stories. You may use a table for a summary display of the planning, but elaborate in text more detail in your focus and feature plan.]

6.7.2 Work Done

[Which stories did you complete in this iteration/sprint. Which ones did you partially complete? Who worked on which story? You may elaborate in paragraph(s) to add more detail about the work done.]

6.7.3 Testing Coverage

[Testing is very important. Show your coverage here. Is this coverage good enough? Explain why you think so. Is it not good enough? Explain a plan to increase the coverage. You may also elaborate on why some artifacts do not undergo much testing. If the testing changed from the last iteration, explain the reasons.]

6.7.4 Retroespective & Reflection

[What were the pitfalls, challenges, and issues you had in this iteration? How can you address them to improve the process in the next iteration? Did anything not go according to plan? Why so and how to avoid the same mistake? Write a personal reflection on what you learned in this iteration (even if a small technical thing like Database storage).]

7 Final Remarks

7.1 Overall Progress

[Have you completed everything? If so, present evidence on how you brought value to your client, and the overall client satisfaction. Otherwise, estimate how much progress you done and how long it would take to finish this project.]

7.2 Project Reflection

[Your personal reflection on the project. What lessons did you learned. What would you have done differently. How can you do better work in future projects? You may write this as a team or per person (or both)]

Appendix

[Appendix section if needed]