Home Together Canada - Group B

Requirements Report

Prepared By:
Jaden Balogh
Alvin Putra
Jared Wright
Athena An

Table of Contents

High-level Project Description

Project Summary

HomeTogether is the non-profit hub for all things home sharing. Differing from other for-profit enterprises in the same space, HomeTogether aims to compete with nobody, by offering fair promotion of everyone.

Users can browse local services, search for potential housemates, connect with local classes and seminars, seek out legal advice, and more. Members will be able to communicate by username via a secure on site messaging system with the option to receive email notifications. Homesharers can filter potential housemates by factors like: working status, location, pets, budget, and more. Members will be able to review businesses and leave star-ratings on businesses and members.

Businesses will be able to list via the classified system for a small fee taken via PayPal which helps cover site costs. Organizations and Members directly involved in home sharing (e.g Government Shared Housing Project) will be able to list for free. Business ads aka "Classifieds" and home-sharing ads aka "Service Listings" will be separated and then further organized into sub-categories such as "Home and Yard Services" for ease of browsing.

Tech Stack

MongoDB + Express.js + React.js + Node.js (MERN Stack).

General Requirements

The client has requested that we make an effort to choose popular languages and frameworks to allow them to have an easier time finding developers in the future.

Server / Hosting

Client has requested that the server is located in Canada.

As of the October 20th, 2020 client meeting it has been determined that hosting is out of scope for this project.

Front-end Languages

HTML - HyperText Markup Language

HTML 5 is the standard modern web markup language. It is a defacto web standard. There are no alternatives.

Pros:

- Easy to use and understand, human readable by design.
- All browsers support HTML.
- Most development tools support HTML.
- HTML is a markup language easily modified and analyzed by other programs.
- Simple to edit, does not require any sort of compilation or virtual machine to run. Changes can be viewed in near real-time.

Cons:

- Takes time to learn the syntax.
- HTML is very fault tolerant and so writing "proper" HTML can be tricky for people unfamiliar with it as it will work even if improper.

CSS - Cascading Style Sheets

CSS is used to transform the raw markup in HTML into a dynamic stylization adjusted based on the user's device and the designer's intentions. It is a defacto web standard. There are no alternatives.

Pros:

- Reusable, the same CSS can be applied to multiple HTML files to ensure design consistency.
- Easy maintenance, an external CSS file can be referenced by multiple HTML documents allowing consistent styling throughout a design
- Search engine friendly.
- Fast webpage loading.
- Printer Friendly.

Cons:

- CSS can behave differently across multiple implementations and so multiple browsers and clients need to be tested for compatibility.
- Need to pay attention to different level of CSS（e.g CSS2， CSS3, etc.）

JavaScript

JavaScript is a scripting language. We briefly considered alternatives like Typescript & Dart however the ubiquity of JS and the consistency throughout our tech-stack makes it our choice.

Pros:

- Comparatively fast for the end user.
- Platform independent.
- Easy to debug and test.
- Extended functionality to web pages.
- Rich interfaces.

Cons:

- Single Inheritance.

Front-end Framework

React.js

React is a library that is mainly used for creating user interfaces.

Pros:

- Good choice in determining how the user will see the site.
- Smaller size, loads faster .
- Fully backwards compatible, making it easier to pick up in the future
- Highly popular with developers, so more likely to find someone to pick the project up in the future.

Back-end Framework

Express.js

Express is a backend framework that is highly used, used by companies such as PayPal, Uber, and IBM. We also considered ThinkPHP but given its lower popularity and that most of the documentation is in Chinese we went with Express.js.

Pros:

- Flexible and highly extensible, as features can easily be imported via plugins.
- The Virtual Domain Object Model (Virtual DOM) allows for selective DOM updates, greatly improving performance as changes don't require reloading the entire DOM.
- React has a large community which is important to our client.

Runtime Library

Node.js

Node.js is a runtime environment that's based on the V8 Javascript engine.

Pros:

- Uses JavaScript like the rest of our tech-stack, allows us to work in one language.
- Supports Asynchronous processing and event handling.
- Development is directly supported by IBM, Microsoft, PayPal, and more.
- Excellent JSON Support which allows it to communicate easily with our MongoDB Database.

Cons:

- Node can performance bottleneck on CPU heavy tasks as it's primarily single threaded.
- Code can rapidly become messy with Callbacks, requiring solid documentation to keep code maintainable.

Database

MongoDB

MongoDB is a Document-Oriented NoSQL database program. We also considered MySQL and

MariaDB but decided to choose MongoDB for its flexibility, scalability, and for the ease of integration with Node.js.

Pros:

- Uses user-readable rich JSON documents.
- Highly scalable, supports seamless horizontal scalability.
- Provides high level APIs to work with data.
- Does not require as much up-front design work, database can be adjusted more "on the fly" than a traditional RDBMS database.

Cons:

- Less "reliability" than an Atomic RDBMS implementation.
- Less performant for complex queries.
- Less popular than MySQL databases.

Map API

The client has requested that we use Google Maps.

Payment API

The client has requested that we use PayPal.

User Groups

Listed are the main user groups for HomeTogether.ca, users will come from all ages and all backgrounds and there will be no specific target demographic. A site for everyone from the freshly graduated to the long retired.

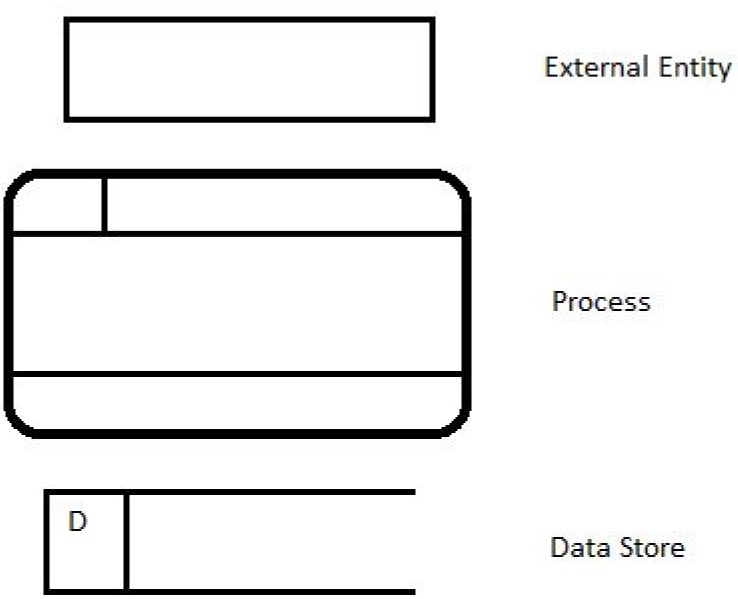
1. Users who are looking for housemates. These users will enter their personal information and register on site thereby becoming Members. They will search the site for other members who match their criteria (Budget, Pets, Location, etc). Members will be able to message each other via the on-site messaging system and receive email notifications. Members will also be able to share their personal information with another member if they wish to do so. It is not shared by default.
2. Users who are looking for services. These users may not register on the site. They will be able to search the site for services, view reviews, leave reviews (if registered). Signup will be the same as users who are looking for housemates. As listings will simply host contact information (E-Mail, Phone Number) these users will not use the on-site messaging system.
3. Businesses who are purchasing classified listings. These businesses will verify with the site before they are able to post any advertisements. They will pay a small fee via PayPal to list their ads which helps cover site costs. They will have limited moderation powers over reviews on their own business (ability to report as fraudulent, etc). Some examples include: Lawn Care Services, for-profit legal assistance, rentals, and classes.
4. Individuals and Organizations who are posting service listings. These include things like: Co-ops, non-profits, members with homes to share, and Government services. They will

signup via a similar method as the business users and will also have to be verified however service listings do not require payment. Once approved they will be able to place service listings for free.

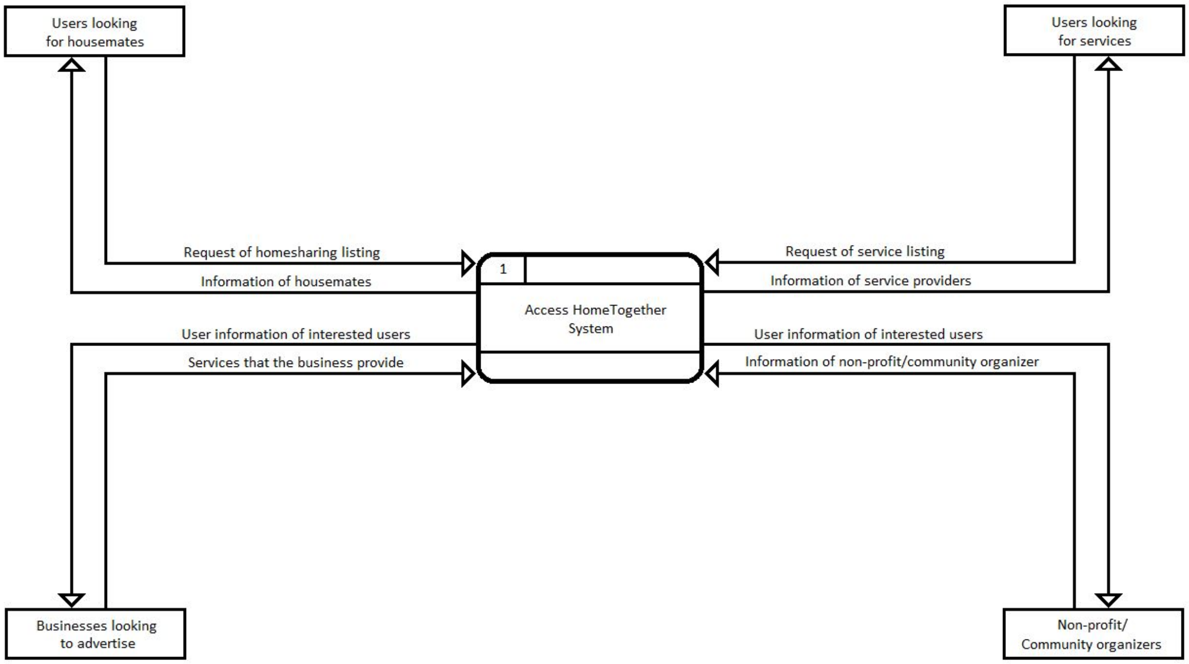
There may be overlap between groups: Eg. User x may be looking for a housemate and to hire house cleaning services.

System Architecture in Data Flow Diagrams

DFD Notations



Level 0 DFD


Users looking for housemates
Users looking for services
Request of homesharing listing
Information of housemates
Request of service listing
Information of service providers
1
Access HomeTogether System
User information of interested users
Services that the business provide
User information of interested users
Information of non-profit/community organizer
Businesses looking to advertise
Non-profit/ Community organizers

The system:
Access HomeTogether System

The external entities:
Users looking for housemates
Users looking for services
Businesses looking to advertise
Non-profit/community organizers

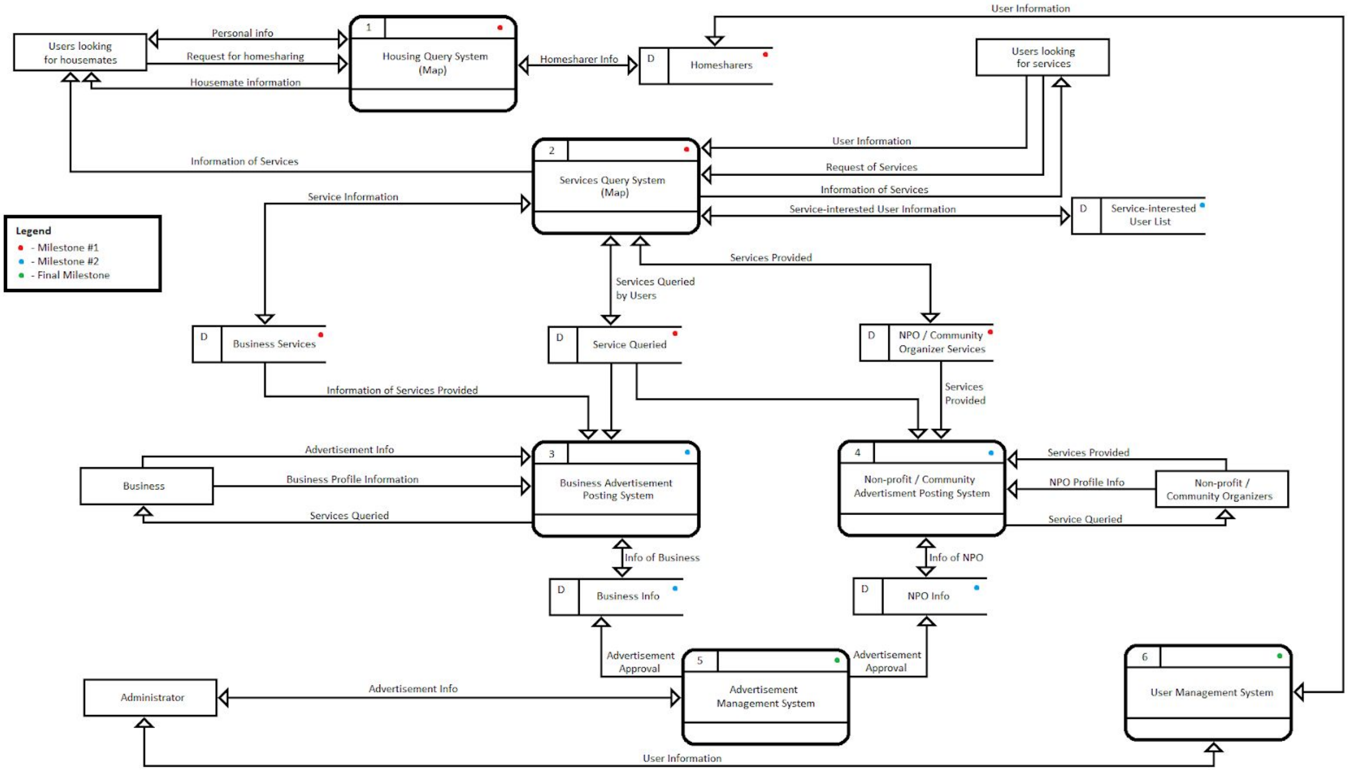
The external entity of users looking for housemates sends the request of homesharing listing to the Access HomeTogether System. It receives information of housemates from the system.

The external entity of users looking for services sends the request of service listing to the Access HomeTogether System. It receives information of services providers from the system.

The external entity of businesses looking to advertise sends the services that the business provides to the Access HomeTogether System. It receives users' information (i.e. the users' requests) of interested users from the system.

The external entity of non-profit/community organizers sends the services that the business provides to the Access HomeTogether System. It receives users' information (i.e. the users' requests) of interested users from the system.

Level 1 DFD



The Systems:

- Housing Query System.
- Services Query System.
- Business Advertisement System.
- Non-profit/Community Advertisement Posting System.
- Advertisement Management System.
- User Management System.

The External Entities:

- Users looking for housemates.
- Users looking for services.
- Businesses looking to advertise.
- Non-profit/community organizers.
- Administrator.

The Data Stores:

- Homesharers.
- Service-interested user list.
- Business services.
- Services queried.
- Non-profit/community organizers service.
- Business info.
- Non-profit/community organizers info.

Functional Requirements

This section describes the functional requirements split out by milestone, and then by user group. Refer also to our user stories described after this section for a better understanding of the functional flow of the website.

Project Terminology

Before we dive into the details of the project's functionality, let's outline some important terms relating to HomeTogether. This glossary was created in collaboration with all 3 project teams.

Profile: Restricted to use for personal accounts
Listing: Any posting made by a business or advertiser in any category

- **Classified:** Related non-home sharing listings (paid)
- **Service Listing:** Home sharing listings (free)

User: Anyone who uses the website

- **Non-registered User:** Someone who uses the website but who is not registered
- **Registered User:** Someone who uses the website and has made an account
    - **Member:** A personal account with a profile.
    - **Business & Service:** A business account that can post listings
        - **Business:** An organization that posts Classifieds
        - **Service:** An organization that posts Service Listings
    - **Admin:** A HomeTogether volunteer who is managing the site.

Milestones

This section covers the four milestones of the project, with each section including an overview of features and the functional requirements for each user group for that milestone.

Requirements - October 21

This milestone involves building documentation, requirements and initial design.

Features Overview

- High level description of the site
- Description of user groups for the system
- Data Flow Diagram of the system architecture, in level 0 and 1
- List of functional requirements at each milestone
- Set of user stories for requirements
- List of non-functional requirements
- Options between 3 tech stacks
- Testing approach and associated frameworks

Milestone 1 - November 25

This milestone involves the core feature set for the primary user group, members creating a personal account (home sharers). This includes map integration for searching, and the classified and user search pages. In this milestone, we focus on implementing a working backend first with only the most important front-end features.

Features Overview

- **About page**
    - Site features
    - HomeTogether description
    - Home sharing definition
    - Website purpose
    - Contact info
- **Classified listing page**
    - Selectable category hierarchy for filtering listings
    - Google Maps integration (filter and display listings by location)
    - Sortable results list
    - Listing database backend (manually entered data in this stage)
- **User search page**
    - Google Maps integration (filter and display users by location)
    - Panel for filtering search results
    - Sortable results list
    - User database backend (manually entered data in this stage)

Features by User Group

User Group	Feature Description
Members	● View the about page including website purpose, background of Home Together, the definition of home sharing, and contact info. ● View the listings section of the website. This must have a way to select the category of listing you want to filter by, as well as using a map to search by location. Categories include: ○ Classifieds ■ Rentals ■ House & Yard Services ■ Legal & Sales ■ Classes, Clubs & Events ○ Service Listings ■ Cohousing, Co-ops, Intergenerational, planned neighborhoods ■ Home Share Facilitation & Matching Services ■ Government & Non-Profit / Shared Living Supports and Services ■ Members with Homes to Share ● View the user search section of the website. This must have a way to filter users by some basic properties, as well as using a map

	interface to filter by location. Results should also be sortable. In this phase, only a basic results list will be shown, as individual profile pages are not yet implemented. At this phase, search filters only include a small subset of the searchable properties described in Milestone 2: ○ Gender (M/F/Other) ○ Age Group ○ Family Status (single/couple/couple with children/single parent/other group) ○ Maximum monthly budget (to nearest hundred)
Businesses	N/A
Admins	N/A

Milestone 2 - February 24

This milestone is focused on the account creation process for all user groups, in addition to listing creation and user messaging. This is also when the front-end flow is refined and remaining pages are added, such as member profiles and individual listing pages.

Features Overview

- **Account creation flow**
  - Log-in and log-out options
  - Sign up screen including display info
  - Account type options (member or business)
  - Edit account info
  - View personal profile and set self as visible/invisible to others
- **Classified listings**
  - Business accounts can view their active listing(s)
  - View individual listing pages
  - Post/edit listings as business account
  - Post/edit homes to share as personal account
  - Rating/review system
- **User interaction**
  - View personal profiles of other users
  - Message other users (with email notifications)
  - Star rating system for other users

Features by User Group

User Group	**Feature Description**
Members	● Create a member account, which includes a sign up screen asking for the user’s personal information (NO identifying information is displayed to other users): ○ First name, last name

	○ Birth Year ○ Username (display name) ○ Password ○ Email address (verified) ○ Home address (use a friend’s, relative’s or verifiable organization’s address if you don’t have one) ○ Mailing address (if different than home; use a friend’s, relative’s or verifiable organization’s address if you don’t have one) ○ Cell or Home phone number (use a friend’s, relative’s or verifiable organization’s address if you don’t have one) ● As part of the signup process, members must submit their searchable profile information: ○ Gender (M/F/Other) ○ Age Group ○ Family Status (single/couple/couple with children/single parent/other group) ○ If couple: second person’s username ○ Maximum monthly rental budget (to nearest hundred) ○ Areas you are interested in living ○ Are you willing to purchase a home with others? (Y/N) ○ Do you have a home to share? (Y/N) ○ Do you have pets (Y/N); if so, elaborate ○ Do you smoke (Y/N); if so, elaborate ○ Do you have health/mobility issues (Y/N); if so, elaborate ○ Do you have allergies (Y/N); if so, elaborate ○ Work Status (full time, part time, retired, semi-retired, full time student, part time student, other - elaborate) ● View their own profile page and edit their personal and display information. ● Toggle their profile as active/inactive which determines whether they show up in search results. ● Click on and view other users’ profiles from the user search results list. ● Message other users through their profiles, with email notifications implemented. ● Apply a rating (out of 5 stars) for other users. ● View individual listing pages in the listings search results. ● Post and edit listings in the Members with Homes to Share section of the listings page. ● Apply a rating (out of 5 stars) for listings. ● On user profiles and listings, have a bookmark option to access quickly later in the bookmarked section of your profile.
Businesses	● Make a business account which will require a set of business information: ○ Full name of business/group/organization/person ○ Incorporated (Y/N); if so, give full incorporation name and owner name ○ Contact person first name/last name ○ Contact person phone number ○ Sign-in username ○ Password

	○ Business phone number ○ Business cell ○ Business email address ○ Business address ○ Business mailing address (if different) ○ Address shown in map search (use postal code for general area if customers do not come to your place of business) ○ Website link (only if they have one) ● Post and edit listings in every category except Members with Homes to Share. ● View list of currently active listings on your business account.
Admins	N/A

Final Milestone - April 8

This stage of the project is for finalizing the user interface, and adding safety and administrator features.

Features Overview

- Safety
  - Report other users
  - Block other users
  - Verification process for business account and classifieds
  - Payment process for classified listings
- Administration
  - Ban feature with reason field
  - View reports/complaints
  - Export lists of users/listings/businesses
  - Verify businesses and classified listings with reason field
  - View complete profiles of all users and businesses
- User interface/user experience update (based on heuristic evaluations)
  - Standardize all pages
  - Ensure basic user experience standards are met
  - Update visuals and graphics
  - Add additional ways to navigate the website where appropriate

Features by User Group

User Group	**Feature Description**
Members	● Report other users through a form accessed through the other person's profile. ● Block other users through an option on the other person's profile.
Businesses	● Pay for classifieds through a Paypal payment process. ● Wait for their listings to be approved before becoming public.

Admins	● Admins are given an admin account with a desired username and password, which they can log in with through the usual form. ● Access the admin page through the site home page, on which they can access a variety of admin features: ○ View active user reports, open and close reports. ○ Option to export desired site usage information to file (members/businesses/listings). ○ Review new listings and submit a verification form with approval or denial options with a written reason that gets sent to the business via email. ● View the profiles of all users. ● Ban any user through their profile with a form stating a reason which gets emailed informing the banned user.

User Stories

User stories are an essential part of communicating and understanding the requirements of our project. These have been created as a collaborative effort between all 3 groups for the Home Together project, and have been reviewed and approved by the client.

Example Member: Michael Scott

Michael is an experienced homesharer. He has shared homes with 3 different people over the last 20 years. Now, he wants to take advantage of online services to facilitate his home sharing needs.

Michael logs onto hometogether.ca for the first time and wants to learn more about the service. He goes directly to the about page and learns about HomeTogether’s purpose and background. Then, he navigates to the account creation tab and signs up for a Personal account with his personal information including address, phone number, email and such information which is then used to uniquely identify his account. He also gives a username which he will use on the site, allowing him to browse the site anonymously. He then enters his display information including fields like monthly budget, working status, desire to have pets, and other such information which should be visible only to other registered users.

Michael wants to find 2 people who do not currently have a permanent residence, who are wanting to live within a 10km radius of Nelson, BC. To do this, he navigates to the user search screen, and enters his desired location and search radius. In addition, he adds a list of search filters including specifying ages 15-30, male, and must be ok with dogs. Upon entering this information, he receives a list of results and finds 4 people who fit his requirements. He goes to each of these users’ profiles and decides that 3 of the 4 would be a good fit from looking at their profile information. Michael then messages each of these 3 users independently and anonymously to connect with them about home sharing opportunities. He then logs off for the day.

The next day, Michael receives 3 notifications via email that he has received replies on HomeTogether.ca. He logs into the site, and finds that 2 of the people he messaged were interested. He coordinates with them through private messaging in HomeTogether.ca to use a specific service that he’s familiar with for home sharing, and chooses to share his real name and email with these 2 people.

Example Business: BestSharing.com

Miranda Platt is the owner of BestSharing.com, a top website for users with homes to list those homes for potential homesharers. She wants to display her website on the HomeTogether listing. Miranda goes to the HomeTogether website and navigates to the account creation page and creates a Business account. As part of this signup process she needs to upload identifying company information including address, company name, mailing information, logo, etc.

Miranda then navigates to the classifieds section of the website and under Homesharing Services, selects the Homesharing Facilitation and Matching Services section. In that section, she creates a new service listing including a listing title, description, pictures, rates, etc as well as choosing a listing sub-category from the available options. Miranda then waits for HomeTogether administrators to accept her new listing.

After having her listing accepted, she goes to her business account details and views her active listings. She sees that some users have already rated her listing, and she wants to change the description slightly. She clicks on the listing from her active listings which takes her to the listing page and edits it accordingly.

Example Business: LawnDoers

Scot Redrick is the social manager of the massive corporation LawnDoers (a lawn-mowing service). He knows that many people who are interested in home sharing are also looking to have their lawns mowed. He creates a Business account on HomeTogether.ca and fills out the company information for LawnDoers.

After signing up, Scot navigates to the classifieds section and then the Non-Homesharing Services section. Being a lawn mowing company, LawnDoers fits perfectly under the Home and Yard Services category, so he chooses that one. He creates a new listing under this section and provides the necessary listing information, along with the selected sub-category(s) from the list of available options. After adding the details, he then pays for the listing using PayPal. This payment is on hold until the listing is accepted by an administrator, at which point it goes through.

Later, Scot comes back to HomeTogether.ca and sees his listing is now publicly visible and receiving reviews. Scot is very pleased.

Example Admin: Sasha Mora

Sasha is on the HomeTogether volunteer team and she has been tasked with managing activity on the website. The first thing Sasha does is log into her admin account. She is then able to see the website as a regular user would be able to, in addition to having access to every user’s and business' profile including normally hidden identifying information. She also has access to a special admin page which allows her to manage activity on the site.

Sasha navigates to the admin page and checks the list of verification requests. These are the businesses and listings that require verification by an administrator before they become visible to the public. There are currently 2 listings that require verification and 1 business, which are shown in a list on the admin page. Sasha clicks on the listings from this list and is taken to the respective listing pages on the main website. She then reviews their information and accepts both listings. She also views the business that signed up, and finds that their website does exist and the information they provided seems suspect. She denies the verification request for this business, with a note that their business information could not be verified. This note is sent as an email to the rejected business. Later, the denied business can return and edit their account information and submit another account creation request.

After reviewing the verification requests, Sasha navigates to the reports and complaints section of the admin page. This shows one new report about a user named Michael Scott. Sasha opens the report and reads the description, along with an attached screenshot of harassing messages from the user. Seeing this, she decides to ban the offending user, Michael. When she does this, she includes a brief note about how inappropriate messages are not acceptable on HomeTogether, which Michael receives as an email telling him he's been banned and why.

Now that Sasha has reviewed all the most recent website activity, she remembers that her boss told her to export a list of all the current users on the site. She goes to the admin panel and uses the export feature and selects All Users. This exported list is downloaded on her computer as a file, which she can then use to filter or send around as desired.

Non-Functional Requirements

The non-functional requirements lays out certain constraints that will ensure the system will be usable and effective to a certain degree, and helps as a guide. Non-functional requirements must be measurable.

Requirement 1: Simplicity

Due to the target market for the users being people aged 50 and up, the simplicity of the system is very important to ensure that the system is usable for the users. The goal is to have all tasks be completed by users in only 1 step. The constraint will be four steps, and all tasks with 4 steps or more will need to be re-evaluated and redesigned.
This constraint is to be placed in:

- Home page
- Listing page
- Account/User Page
- Advertisement/Listing pages

Administrator pages will not have this constraint in place, as functionality will be more important than simplicity.

Requirement 2: Security

The client firmly believes in the importance of how the site appears to the users, with an emphasis in looking trustworthy. This will be maintained by having verification methods put in place during user signup. The goal is to have 2 verification methods, which would be email and phone number, with the constraint being 1 verification method. In addition to looking more trustworthy, verification methods prevent people from easily making multiple accounts to spam or commit malicious attacks. Captchas would be another option that can be put in place to further improve security.
This constraint will be placed in:

- Sign-up page
- Listing page

Environmental Constraints

These are constraints brought up from the surrounding situation, whether it is technological limits or a client request.

Requirement 1: Word Choice

The client requires certain terms to be used to mean specific things to avoid confusion. These terms can be found in the Project Terminology section above.

Requirement 2: Payment Methods

The client has required that the payment method used for advertising be PayPal, with the business account having been set up.

Requirement 3: Budgetary Restrictions

Due to the client being a non-profit, the client does not have a budget. As such, we do not have the funds available for 3rd party software support, expensive cloud-based web-hosting, etc.

Testing Strategy

React.js is a highly component-based framework, and as such, our testing approach will be focused strongly around testing the functionality of these individual frameworks. Each component will be associated with a corresponding integration test that provides a high-level insight into the overall stability of the website. Furthermore, heavily reused components will implement unit testing as well to ensure that the correct output will always be received for a large range of use-cases. This approach along with a focus on continual client feedback gives us confidence in both the functionality, and the dependability of the product we are creating.

Frameworks

We will use the Jest and Enzyme frameworks for testing our React code. Jest provides bountiful options for both unit and integration testing through functions such as it(), test() and expect(). Enzyme is designed specifically for component testing in React, and complements Jest beautifully in this regard. The functions mount() and shallow() will be used to interface directly with the HTML of the webpage in our tests, allowing us to closely mimic and test user behaviours. As mentioned above, each React component will have a corresponding integration test, in addition to unit tests for the most crucial components.

Continuous Integration

We will make heavy use of GitHub Actions in our repo, and one of these core uses is for automatically running the tests mentioned above. This can be done easily by creating a workflow file in our repository and enforcing a status check on every pull request into the dev or master branches. This status check would then run both our unit and integration tests on every update, in addition to other checks such as JavaScript linting. The continuous integration process ensures we keep a consistent standard across our codebase, and that live branches remain in a stable state.

Testing References

We've compiled some useful references that were used to research the above options as well as provide direct implementation examples and details:
[1] GitHub Continuous Integration with ReactJS
[2] Jest and Enzyme for testing React components
[3] Implementing Jest and Enzyme in React
[4] Integration testing in React

Feedback Q&A

General

What devices are compatible with HomeTogether.ca?
The site will be designed to be used on computers, with support for all common browsers (Firefox, Chrome, Edge, Safari). Legacy browsers such as Internet Explorer 7 and legacy devices such as the Nokia 6233 will not be supported.

Are you integrating your development into work already done by the client or building from the ground up?

The project is being built from the ground up. However, the design decisions made are influenced by past website development efforts by the HomeTogether organization.

Did your group come up with its own milestones or were they given?
Our group decided our own milestones.

What are some of the challenges your group is expecting to face and how are you planning to get through them?
Given that the tech stack will be largely new to us, we expect a learning curve on the development side of things. We plan to get through this using collaborative efforts to learn and teach each other new material.

What keeps a user from creating a fake phone number? Or fake emails?
In the end, there’s nothing stopping an incredibly dedicated user from bypassing the unique account restrictions. However, this would require an address, email and phone number, so we are not concerned about this possibility.

How is this different from other listing sites like roomster?
Roomster charges for full access to the app. They state that they are the “best way for people to monetize their extra space” while HomeTogether.ca is aiming to offer a centralized *non-profit* platform for people to find shared accommodations and local services. HomeTogether aims to compete with nobody, by offering fair promotion of everyone.

Functional

How are bans implemented within the system?
Bans will disable and delete the user’s account while maintaining their email and phone number to ensure that they aren’t used to sign up with the same ones again. We considered implementing “Shadow Banning” where the member isn’t notified of a ban but instead their messages stop being sent to other users. However we decided this feature was out-of-scope for the project.

Who are the user groups?
The target user groups are people of all ages looking to homeshare, home-sharing services, and businesses that have interests in advertising to homeowners such as lawn-mowing or plumbing services.

How much will a business have to pay to list?
This is something for our client to decide what is required to meet their needs, we aim to provide a flexible payment API so that it can be dynamically adjusted.

Are the admins allowed to access users' accounts?
Yes, site admins will be able to access users accounts. This includes all entered information, and message history as an extra feature if we have remaining time before the deadline.

What is the classifieds page in Website areas?

This will be the page where services such as lawn mowing and plumbing advertisements will be placed. The page will show the services that can be found within an area that the user selects.

How will each user interact with the system and each other?
Users interact with the system through a simple web interface which includes two primary sections: users and ad listings. Both of these can be searched using the map UI as well as a set of filters. To interact with other users directly, HomeTogether offers a private and anonymous messaging system between registered members.

Will the users need to give access to the map to share their locations for house searches?
Users will have the option to share their location for more convenient use of the map interface, but it is not required.

Will there be advertisements and how do businesses interact with the users?
There will be advertisements, but they will be static. Users will have to personally contact the service on their own (such as calling them by phone) and the messaging system placed in the site will only be to message other users looking to homeshare.

Will there be a verification process for the account sign up for security reasons?
Yes, the user will need to use an email, a phone number and a valid address.

Can users who aren't admins report or block potentially problematic users from messaging them or what not?
Yes, users will be able to report and block other users.

What does "account creation flow" mean?
The account creation flow describes the signup process: members sign up with personal/identifying information for verification as well as entering their profile display information; businesses provide their information to be verified by the admins. All of this also includes logging in and out, as well as authentication for features that are specific to certain types of users (e.g. messaging for members).

Non-functional

How do you define a step (for the "simplicity" requirement)? Is it the number of pages you go through, the number of buttons you must press?
The steps will be measured in the number of pages.

Why are simplicity and security the only categories for the non-functional requirements?
The client believes we are more qualified in designing the system, and has set very few non-functional requirements. These two constraints are the primary focus of the client, especially since accessibility for non-technical users is critical to the success of the website.

How did you arrive on the constraint of 4 steps per task?
The sign-up page will need 3 steps to satisfy the security constraint (sign-up, email verification, and phone verification) and cannot be reduced.

Tech stack

Is the Google Maps API to show users pins of available housing? Will this be filtered based on user preferences?
Ideally yes, the map will display a dynamic map and show available houses and update via filters however we are currently reviewing the Google Maps API to get an understanding of the pricing. Google provides $200 of service credit each month however since we're designing HomeTogether to be highly scalable we don't want to implement a design with ballooning costs.

Why are you using Google Maps API over other open source maps?
The client has requested that the Google Maps API be used for this project.

Which tech stack was ultimately chosen and why?
We will be following the popular modern tech stack known as MERN: MongoDB, Express.js, Node.js and React.js. The reason for this choice is that the client has requested that the project be easy to pick up by future developers, and therefore use a well-known stack.

How are you going to handle the payment?
The client has requested that all payments be handled by PayPal.

Will the security be created by the team, or already made using a technology from the tech stack?
We will be using open-source libraries whenever possible for security. If we have the time we hope to implement optional 3rd party 2-factor authentication via Authy & Google Authenticator.

Why would you consider using a noSQL database like Mongo for such a simple website?
Our client has requested that the website be flexible for further development for future developers and for the site to be highly scalable. Using a NoSQL DB may seem like overkill currently but given its looser database schema compared to an SQL based database and the ease of horizontal scaling it provides we believe it to be the optimal choice.

Do you plan on running Express with Node?
Yes, we will be implementing the MERN stack (Mongo, Express, React, Node).

Are there any major security concerns with implementing PayPal into the program?
No more so than any other payment processor. Using a secure 3rd party is far more secure compared to rolling our own payment API given our limited knowledge and work in the financial/cryptographical domain.

Testing

Will any time be set aside for acceptance testing or integration testing?
Our primary testing approach will be through unit and integration testing, as described in the Testing Strategy. However, at natural points throughout the project we expect to involve the client to do some basic evaluations of the website, as well as testing functionality ourselves throughout development.

How often are you guys going to check the status on the dev and master branch?
The master branch will be used exclusively for complete features and stable releases. Dev will be used for any completed feature branches and for testing integration of independent features prior to release. We expect these to be updated every week or two depending on the amount of updates.

What DB will you be using for the testing period?
We will use MongoDB for all phases of the project. In the first milestone, test data will be manually added to the DB to test reading data, while in later phases create, update and delete functionality will be integrated into the site itself.

How will you make sure that unit tests are implemented consistently by the team members?
By using well-known and well-documented frameworks such as using Jest and Enzyme for React testing, we can create trustworthy tests that are standardized across our codebase.