

# Home-ME Sf

**By:**

**Copyright team 7 CSC648-848 Spring 2019**

(Team Lead-Bakrola Divyaraj, Front-Henok Welday & Back-end- Nour)

(Tech Lead-Nour, Database-Bakrola Divyaraj & Back-end-Preyansh) (Front-end UI-Akash, Database-

Bakrola Divyaraj) (Front-Bing & Back-end-Nour)

(Front-end UI-Henok) (Front-end UI-Bing/ Akash )

## **1. Executive Summary**

Team 7 is Designing a Web Service Client Application named “Home-Me” that will be Real-Estate Website which is an all-in-one real estate site that gives you the local scope about homes for sale, apartments for rent, neighborhood insights, markets and trends to help you figure out exactly what, where, and when to buy, sell or rent. The service will allow property owners to market themselves on Home-Me by providing general information about the property. The Home-Me users will have to be registered, for inquiring about any listings, if they want to post their own property. The Property host will be able to view every Inquiry and will have an option if they want to respond to it or not. The site will also allow the user to see the base price and negotiate the amount in a chat box. Properties owners will be able to keep track inquiring customers and will have data available for tracking customer browsing habits at their listings. As more properties register with Home-Me, customers will get an expanded option of properties and will be able to search by availability, price, or property type.

Team 7 is designing a web client application called “Home-Me ” that will be a real estate website template which will satisfies some common real estate website requirements including dynamic pages, listing management, blog articles, author management, Formik for forms, Google Analytics, and website search. Built using [Node.js](#), express framework for the server, React library for the frontend.

## **2. Use cases**

### **1. General User:**

1. Host: Prior to the service, the host logs in on his station and shall see the reservations for the upcoming service. He sees that one of the earlier reservations is Michael's party of 24. He and the staff set up everything necessary for the party of 24. Michael arrives at the restaurant first and approaches the host. The host asks Michael how big his party is, Michael responds saying he has a reservation under Michael for a party of 24. The host immediately directs Michael to his reserved seating.
2. Property Owner: Marco is a new restaurateur, opening his first restaurant, he's looking for a service to handle his reservations and a hub which he can use as his restaurant's homepage. He finds HandyTable, and registers, he then uploads a few pictures, writes a description of his restaurant, uploads his menu and submits, after a short period his restaurant is listed on the site and his pictures, menu and description are being used on his Hub.
3. The admin is doing his daily due diligences, he checks his notifications for new restaurant registrations, and any flagged reviews. He sees that he has a few flagged reviews. He clicks the notification and is directed to a page with the review. He reviews the post and chooses to remove the review from the reviews page.

## **3. Data Definition**

## User

-**Casual User** – User that has not registered with the site, can browse the majority of the site, but cannot make reservations.

-**Registered User** – User with the same privileges as the casual user, but can make reservations

**Admin** – A trained user in the employment of <us> who maintains user information, restaurant information and reviews

## Properties

-**Owner** – The client using our service that owns and operates the restaurant

-**Manager** – Person in the employment of the restaurant owner responsible for the operation of the restaurant

**Host/Hostess** – Person in the employment of the restaurant owner responsible for greeting

**Privacy policy** – Policies dealing with what data we collect, why we collect it, and what we do with the data

**Profile** – a central place where a user can save their personal information and preferences to be used later for convenience

**Map view** - A list of all restaurants in the vicinity of the city viewed in a map

**Capacity** - The maximum seating available in the restaurant mandated by state laws

**Property availability** – a table that has not been reserved and is open to seating immediately

**Current Online waiting list** – a call ahead list where customers put their name in online and when they expect to come in

**Size of Property** – how many Sq. ft

**Reward points** – points accumulated by Inquiring about the property from our website gets you discount

**User history** - A view of the user's past

**Help** – a document briefly explaining how the application works

**Targeted marketing emails** – emails sent to members based on their reservations and cuisine preferences

2. **Personae** and main **Use Cases**: Summarize key *personas* (categories of users) for your application – their general characteristics, goals, skills, pain points related to the application you are developing. About 1/3 of a page per persona – see class notes. (Note: in personas you stay general, in use cases you say how personas will use your app (at high level)). Then provide 4-5 main *use cases* (one paragraphs for each use case) - see class notes on more detailed format for requirements. Focus only on main use cases. Simple text format is OK and preferable – tell us a story about who and how the application is used. Focus on WHAT users do, their skill level, not on HOW is the SW implemented. NOTE: avoid specific on HOW functions will be done and text resembling user manual: this is supposed to guide the design of the future product and is NOT a description of how the product will work (you don't

know that yet) – see class slides for details. Please assign a descriptive title to each use case so it can be tracked.

3. List of main data items and entities - define main terms, data structures and “items” or “*entities*” *at high or logical (not implementation) level* (e.g. name, meaning, usage, and NOT how the data is stored in memory) so it is easier to refer to them in the document. Focus on key terms (main data elements/records used in your app, types of users and their privileges etc. These terms and their names *must be used consistently* from then on in all documents, user interface, in naming SE components and database elements etc. In cases where you attach behavior and privileges to data items (e.g. user types) that also drives the design of the SW. In later milestones you will add more implementation details for each item. You will later expand this section with more details. This will help define planning and design for the DB for example.

4. Initial list of **functional** requirements – see class notes. This refers to high level functions you plan to develop to the best of your knowledge at this point. Focus on WHAT and not HOW. Keep the user in mind. Develop these functions to be

consistent with use cases and requirements above. Number each requirement with *unique numeric value* and use these numbers consistently from then on. For each functional requirement use 1-3 line description. At this stage no need to prioritize the requirements. We are looking for 20 or so requirements.

#### **USERS:**

- 1- Once on the page, casual users shall be able to browse and search for content on the page without logging in.
- 2- Casual users shall be able to create free accounts.
- 3- Newly registering users shall be able to input personal information and upload a profile picture.
- 4- Registered users shall be able to login using their accounts.

- 5- All users shall be able to search for content on search bar based on location such as address, city or zip code.
- 6- All users shall be able to filter their search results based on price range, size of property, levels, garage, time of listing ....
- 7- Registered users shall be able to setup preferences for location, neighborhood, price range, preferred agents, availability dates, levels, garage and more...
- 8- All users shall receive search results (cards) showing basic information about properties, as a list view and a map based view.
- 9- All users viewing search results, upon choosing and clicking on a property link, shall be shown details containing more information about property.
- 9- Registered users shall be able to buy, sell, rent and place bids on properties while casual users shall be asked to register prior to the process.
- 10- Registered users shall be asked to submit credit report or bank statement prior to transaction process.
- 11- Registered users shall be asked to submit a fully refundable down payment prior to placing bids.
- 12- Down payment of user with highest bid then becomes partially refundable after closing bids.
- 13- Registered users that did bid shall be able to modify their bid.
- 14- All users shall be able to see a list of the bids placed, and time left before bidding ends.
- 15- Registered users cancelling their bids shall receive a full refund.
- 16- Registered users bidding, cancelling bids, or modifying their bids shall receive a confirmation email with information about each bid process.
- 17- Registered users shall receive an email about any updates or modifications made to the property bid.
- 18- Properties shall be initially listed based on the latest listing.
- 19- Registered users shall be able to request services from agents.
- 20- Agents shall be listed based performance over time, popularity, and area.
- 21- Registered users shall have the option to request more information about the property from the Agent.
- 22- Registered users shall have the option to subscribe for notifications based on latest listings, price drops and open houses.
- 23- Registered users shall have a favorites page
- 24- Registered users shall be able to add properties to their favorites page
- 25- All users shall be able to share information about properties and refer properties to others using email.
- 26- Registered users shall be able to write reviews about properties.
- 27- All users shall have access to a help section that show them how the web page can be used.

**Property Owners:**

- 1- Registered users shall be able to post properties as Property Owners.

- 2- Property owners shall be able to add properties information, including photos, details, address, open house hours, price, levels, room sizes.
- 3- Property owners shall be able to specify which photo shows as their main listing photo.
- 4- Property owners shall be able to rent, sell and open bidding for their properties.
- 5- Property owners shall be able to request services from agents.
- 6- Property owners shall be able to modify information about their properties.
- 7- Property owners shall be able to cancel bidding process.
- 8- Property owners shall be required to provide legal documented proof of ownership of their properties or other legal documents prior to moving to bidding process.
- 9- Property owners shall receive inquiries and requests about their properties via email.

#### **Agents:**

- 1- Registered users shall be able to become real estate agents upon submitting an application.
- 2- The real estate agent's application shall ask the requesting registered users for more personal information including, Resume, Broker's license and submit a written statement about their experience as brokers. ( does not have to be real )
- 3- Registered users shall receive a confirmation about their application processes.
- 4- Once applicants become agents, they shall have access to information about other users requesting their services.
- 5- If services were requested, agents shall connect all users to property owners.

#### **Admins:**

- 1- Admins shall be able to approve property listings.
- 2- Admins shall be able to approve real estate agent applications.
- 3- Admins shall be able to approve bidding users informations.
- 4- Admins shall be able to approve bidding property owners informations.
- 4- Admins shall be able to edit and delete users accounts.
- 5- Admins shall be able to delete and edit property listings.
- 6- Admins shall be able to delete users comments.
- 7- Admins shall be receiving processes, requests, confirmations over email.

5. List of **non-functional** requirements (performance, expected load, security requirements, storage, availability, fault tolerance...). Note that mandatory high level non-functional specs are given in high level document, so for M1 we recommend you simply copy them from high level document from iLearn. Please observe and adhere to these non-functional requirements in your design and development from now on – you are not allow to change them unless you get permission.

#### **Security:**

1. No limit casual user
2. Primary User and Administration User are be able to log in.
3. Password / username be able to save on browser.

4. User name would be real email.
5. Server will send a confirmation email when user register.
6. Uploading and Context must be approved by administrator.
7. This site shall not accept third party cookies.
8. Primary User should accept the background check documentation.
9. User information and user information should be viewed by administrator and accept by administrator.

Audit:

1. New registrations shall be audited by the administrator
2. New registrations shall be approved by the administrator
3. Causal user will not authorized to administrator and register page.
4. Administrator will approves all the register user posts.
5. Only administrators are able to make changes for websites.

Performance:

1. There will be less than 1 sec time for loading/opening all the web pages
2. Search Engine should be running on background improving the web performance.
3. Uploading should be running background with server

Captivity :

1. Total data stores should not exceed 80% than the server capacity.
2. The site is able to handle 50 users at same time.
3. The site is able to add more features.

Reliability

1. Downtime maintenance should be announced with email to all users.
2. Downtime maintenance should be shown on the main page with time and approximately time.
3. Downtime maintenance should be less than 1 hour.
4. After Maintenance, announcement email should be send to all user.

Data:

1. Data should be collected and saved by administrator once a week.
2. Administrator able to backup and recover the data.
3. Image /videos should be be uploaded on right format.
4. Image / videos should be compressed and posted.
5. Image should be limited to 1 megabyte.

Compatibility:

1. The site shall be compatible with the last version of Safari browser
2. The site shall be compatible with the last version of IE explorer
3. The site shall be compatible with the last version of Firefox browser
4. The site shall be compatible with the last version of Chrome browser

5. The site shall be compatible with at least an old version of all the browsers listed above
6. Third party applications shall not be able to modify any content that may affect the site compatibility
7. The site shall be ready to support with any or minimal changes any other compatibility that may be added in future versions
8. The site should be compatible to escalate to new databases

#### Conformance with Coding Standards:

1. Only working code should be submitted and upload to the repository.
2. Coding should meet all requirement that comes up on this milestones.
3. The site will be debugged and tested before uploading.
4. Any error will not be affect the functionality of the site should announced to user.
5. Error should be stored in the log.
6. The site will not be launched if 1 feature is not meet.

#### Looks and Design

1. Font should be in the correct format, Languages should not be garbled
2. Every element on site should be in correct size, and font.
3. Every element on site should have the correct density.
4. Every picture and text should be in correct size.
5. Website has acceptable colors and layout.

#### Internationalization / Localization Requirements:

1. Default language should be English
2. The site will insert with map location in U.S.

#### Web Policies

1. No share Data with 3rd parties.
2. A link to the policies of this site shall be always visible in all its pages to be accessible by all parties

**6.Competitive analysis:** Home me SF have a different future than the popular real state websites.

	Home me SF	Zillow.com	Craglist.com	Trulia.com
Quality	+	+	-	+
Auction	+	-	-	-
Credit score	+	+	+	+
Service fee	+	-	-	-



Search Algorithm	+	+	+	+
------------------	---	---	---	---

Home me SF is an affordable and effective service giver in the real state business when it compared with others similar websites in the real sate industry. Beside our affordable price and we have an auction options for our customers we seal houses in auction and the winner have 24 hours to pay and own the place is the winner does not pay or signed the contract we will give the chance for the next higher payer of the house this option will put us in different position in the real state industry. Our search algorithm in very effective and will helps you to save your time and energy. You can search the home by zip code or Arial map of the city of san Francisco.

7.High-level **system architecture and technologies used:** Briefly provide itemized list of all main SW components such as frameworks, APIs, tools and systems to be used, supported browsers and deployment platform (SW and server) to be used. This list is to be the list of approved tools and systems from M0 (which may be the list you have modified during or after M0 – but get it approved). Any other external code/ API/tool must be approved by instructors and you have to justify it.

Framework:Javascript environment , Express Framework

API's: Google login API ,

Tools:

Supported browsers: Chrome , Internet Explorer , mozilla

Deployment platform: AWS

8.Team: list student names, mark their roles (team leader, front and back team lead and github master, document master, team member front end, team member back end etc.). Note that leads and github/document master also should have other roles.

(Team Lead-Bakrola Divyaraj, Front-Henok Welday & Back-end- Nour)

(Tech Lead-Nour, Database-Bakrola Divyaraj & Back-end-Preyansh) (Front-end UI-Akash, Database-Bakrola Divyaraj) (Front-Bing & Back-end-Nour)

(Front-end UI-Henok) (Front-end UI-Bing/ Akash )

9. Checklist: for each item below you must answer with only one of the following: **DONE**; or **ON TRACK** (meaning it will be done on time, and no issues perceived); or

**ISSUE** (you have some problems, and then define what is the problem with 1-3 lines)

Team found a time slot to meet outside of the class

- 

Github master chosen

Team decided and agreed together on using the listed SW tools and deployment server

Team ready and able to use the chosen back and front end frameworks and those who need to learn are working on learning and practicing

Team lead ensured that all team members read the final M1 and agree/ understand it before submission

Github organized as discussed in class (e.g. master branch, development branch, folder for milestone documents etc.)

### **Background reading:**

Document we posted about high-level vision of our application.

Class material on requirements and specs

Relevant existing applications and products.

Info about allowed frameworks – class notes and posted on iLearn

M0 document and documentation on SW tools and frameworks you plan to use Git and Github tutorials

## **Submission for Milestone 1 document for review – you must follow the instructions**

### **below PRECISELY:**

Teams must collaborate in creating M1 document by having working M1 document on

their team Github private repository (similar to managing code) so all team members and instructors can access it. We recommend having a folder named “Team N class milestones” (and also a folder for general documentation).

We strongly suggest the following collaborative approach for creation and completion of M1 document (NOTE: creating a team document is similar to creating a code by the team of programmers):

- Team leads assigns M1 editor
- Team lead/M1 editor assign individual chapters to team members
- M1 editor collect chapters, edits/corrects then integrates them into a well formatted document (with same font and formats)

• • • • •

• •

- 
- 
- 

- M1 editor posts final candidate full document on team repo so that all team members read full document for one more review and any feedback

- M1 editor completes the final version as per feedback

- Team lead submits M1 info for review as per submission instructions.

Submission instructions (below) must be followed precisely and completely or grade penalty will be imposed

**The whole student team submits one milestone document for M1, as follows:** Team leads will send e-mail with a link (NOT the attached file) pointing directly to M1 Document (in PDF) to jortizco@sfsu.edu

**Submission e-mail subject line:** MUST be "CSC648-848 Spring 2019 Milestone1 Team N" in the subject line (N is a team number 01...15).

**File name of the M1 document in PDF (to which the link is pointing) to MUST be:** CSC648-848 Spring 2019 Milestone1 Team *N*.PDF (*N* is your team number) (We use only PDF so I can send you feedback as yellow sticky notes).

Submission must be done by the deadline specified, any extension has to be approved at least 24 h ahead of the deadline.

### **M1 document format and structure**

**Title page** MUST include

–“SW Engineering CSC648/848 Spring 2019”

Project/application title and name (you can use the name you chose for your application) –Team number

–Names of students (team lead first) with e-mail of team lead. Please mark those

who are team lead, front end, back end leads and github master –“Milestone 1”

–Date

–History table (revisions) (Note: you will update this document based on instructors’ feedback so this is important)

**The rest of the document** has to contain ALL sections as described above under “Content and structure for Milestone 1 document for review”

Team leads and M1 editors: make sure document is well formatted, reads well, is complete, and looks professional. This will be part of your portfolio and will influence

•

•

the grade. Make sure all team members read final version and give comments before submission.

### **Instructor's feedback and creating final Milestone document for Final Project delivery**

In the course of developing M1 you can ask instructor questions via e-mail and during team session in the last hour of the class. Upon submission of M you will get feedback from instructors by any of the following: e-mail, markings on your document and in class during team meetings. This feedback must be analyzed and taken into account by your team in order to revise your M1 and this must be used subsequently for the rest of the project. Please enter the revision summary in history table.

Instructor will comment from the standpoint of CEO, VP of Marketing (who translates customer and marketing requirements) and CTO (Architecture etc.). You may choose not to agree with the comments. This is OK as long as you justify this and are prepared to live with that design and deliver it. In some cases, instructor may insist on some features or decisions.

Upon getting instructor' feedback on your questions and submitted document, you need to revise your first draft, freeze it (meaning no more changes on this document even if future design changes) and use it as a basis for developing Milestone 2 (M2). The frozen document M1 will be submitted as part of final project delivery in Milestone 5. Do not start working on M2 before you get feedback on M1 and make sure all team members read frozen M1 document.

Future M2 functions and actual SW app may differ from what you proposed in M1, that is normal and in fact expected to happen in the spirit of iterative SW development. In that case there is no need to modify frozen M1 document.

### **Evaluation and grading**

I will grade each "frozen" milestone only when it is submitted with final project at the end of the class (Milestone 5 folder), after it has been modified for instructors' feedback. Note that instructors' feedback is NOT graded in order to encourage interaction.

Milestones improperly submitted will first be returned, and if problems persist 10% penalty will be applied to the grading of that milestone. Only one error in submission in M1 will be “forgiven”, any subsequent problem submission in any milestone documents will be recorded with negative points under the rubric *SE Process grade: submissions*.