CAN THO UNIVERSITY

COLLEGE OF INFORMATION AND COMMUNICATION TECHNOLOGY

GRADUATION THESIS

BACHELOR OF ENGINEERING IN INFORMATION TECHNOLOGY

(HIGH-QUALITY PROGRAM)

VSTAY - A SYSTEM FOR SEARCHING AND RENTING ACCOMMODATIONS USING REACT AND NODEJS

Student: Tong Minh Duc

Student ID: B1608292

Class: 2016-2020 (K42)

Advisor: Dr. Thai Minh Tuan

Can Tho, 01/2021

CAN THO UNIVERSITY

COLLEGE OF INFORMATION AND COMMUNICATION TECHNOLOGY

DEPARTMENT OF INFORMATION TECHNOLOGY

GRADUATION THESIS

BACHELOR OF ENGINEERING IN INFORMATION TECHNOLOGY

(HIGH-QUALITY PROGRAM)

VSTAY - A SYSTEM FOR SEARCHING AND RENTING ACCOMMODATIONS USING REACT AND NODEJS

Student: Tong Minh Duc

Student ID: B1608292

Class: 2016-2020 (K42)

Advisor: Dr. Thai Minh Tuan

Can Tho, 01/2021

EVALUATION OF ADVISOR

.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....

Advisor

Thai Minh Tuan

ACKNOWLEDGEMENTS

I wish to express my deep gratitude and sincere thanks to my professor Thai Minh Tuan – A lecturer of College Of Information and Communication Technology who gave me the golden opportunity to do this wonderful thesis on the topic “Vstay - a system for searching and renting accommodation using React and Nodejs”, which also helped me in doing a lot of Research and I came to know about so many new things I am really thankful to them. Then I would like to thank the lecturers of Can Tho University, specifically, lecturers of College Of Information and Communication Technology who taught me invaluable knowledge when I studied.

I am extremely grateful to my family for their loves, prayers and caring for completion of this thesis. I am very much thankful to my friends for their support when I was doing research in Can Tho university.

Sincerely,

Can Tho, 06/01/2020

Tong Minh Duc

TABLE OF CONTENT

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

Abbreviations	Description
API	Application Programming Interface
Usecase	Use Case Diagram
ERD	Entity Relationship Diagram
UI/UX	User interface, user experience

ABSTRACT

Nowadays, finding accommodations in a big city is really a challenge, there is a lot of unreliable and scam information on social networks or newspapers. For example, students from rural areas start their study in cities, or businessmen who are on travel usually need to find accommodation. The task usually consumes a large amount of time on searching and they have to go to that place to check and sometimes it is not what they expected and does not fit with their plan. As a result, with an aim of supporting users to search and rent accommodation more easily, this thesis develops VSTAY which is a system for searching and renting accommodation using React and Nodejs.

The system is implemented with the main functions such as uploading posts of accommodation sharing or seeking; accommodation searching; Map function; Online chat. The frontend and backend of VSTAY are developed by exploiting React and NodeJS, respectively. Besides, Google Firebase for online chat, HTML, CSS, Javascript, MongoDB (NoSQL database). Hopefully, with provided features, the system can help users in accommodation searching and renting more easily. It also helps the owners of accommodations, boarding houses or motels to approach ideal customers.

PART 1: INTRODUCTION

Vstay - a system for searching and renting accommodation using react and nodejs

Nowadays, the need for searching and renting accommodation is high, everyone always wants to find a good place to live for a long time. However, there is a lot of unreliable information about renting accommodation on the internet, people always confuse whether accommodation is good enough or they are going to be lied to or not, a bunch of questions appear when they search by those information. For example, students who study far away from home or people who are going to have business travel always want to search for a good accommodation with efficient cost, comfortable place and high security. Owners of boarding houses and motels or people who just have an empty room need a methodology to connect to ideal customers. However, people usually rent accommodation by recommendation or following the advertisements, after that they have to go to a lot of places and review, it wastes time and low security.

VSTAY was created to provide a useful tool that supports both people who want to search and share accommodation in the most convenient way. The application will allow users uploading seeking or renting posts, users have to provide details and image of accommodation, it will help others users to have an overview of that place. Users will provide contacts on their posts for other users to communicate with, the type of contacts will be social networks: Facebook, Twitter and Instagram. On the main page, other users can see and search for their ideal targets. The system will allow users to edit their profile such as: social network, jobs and description to make their profile more attractive. Vstay will help users to find ideal accommodation and save a lot of time compared with traditional methods.

Related work

On the world, there are a lot of websites or applications that provide searching for accommodation such as: Roomgo in Australia [1], Flatmate [2] or Real Estate [3]. In Viet Nam, searching and renting accommodation online is not popular, there are just a few applications such as: Ohana Living [4], timphongtrovn [5] or batdongsan [6]. Almost all applications just allow users to upload posts with restricted areas, search in an easy way and contain un-useful functions.

The system helps users to search and rent accommodation easier, support owners of boarding houses, motels or people who want to find their ideal flatmates and customers. The application has some improvements, for example, it allows users to search for their accommodation on the map with friendly UI or users can chat directly with owners of properties or flatmates to discuss cost or information of the property.

The objectives and scope

The application provides main functions to help users find, search and explore wonderful accommodation and increase customers for owners of accommodation. This thesis will be focused on the problem of searching and renting accommodation, the need of finding ideal customers from businesses such as boarding houses, hotels or homestays in Viet Nam. The scope of the study is: researching the problem of searching and renting accommodation in Viet Nam, solving the problem by implementing applications with friendly UI/UX.

Research content

Methods:

- Analyzing requirements: research problems on the internet, analyzing functionals, describing requirements to build systems.
- Design: Designing UI/UX; Analyzing and designing models: ERD.
- Implementing: Javascript - Programming language (NodeJS for backend server, ReactJS for frontend); Database: MongoDB.

Solutions:

- Using knowledge of analyzing and designing information systems, building web applications: HTML, Javascript, CSS, Algorithms,etc
- Softwares: Google Chrome - web browser; Framework: ReactJS, Bootstrap, NextJS; Library: Firebase.

Thesis outline

Part 1 - Introduction:

This part introduces research problems, related works, the purpose of study, the scope of the study and research content. At the end of this chapter, there is the content introduction of each chapter in this thesis.

Part 2 - Thesis content:

Including three main parts:

Chapter 1: Background and related work:

General information about the study and main functions of the system.

Chapter 2: Design and implementation

Introduction of UI/UX designs, models and implementation, describing technologies which will be used in the study, use case diagrams, database and details of function in the system.

Chapter 3: Presentation of the test goals, scenario of testing and the results.

Part 3 - Conclusion and future work:

This part summarizes the result and recommends the improvements in future.

PART 2: THESIS CONTENT

CHAPTER 1: BACKGROUND AND RELATED WORK

2.1. BACKGROUND

2.1.1. ReactJS

ReactJS [7] is an open-source, component-based front end library responsible for the view layer of the application. ReactJS is a declarative, efficient, and flexible JavaScript library for implementing user interfaces. It is maintained by Facebook.

React uses a declarative paradigm that makes it easier to reason about your application and aims to be both efficient and flexible. It designs simple views for each state in your application, and React will efficiently update and render just the right component when your data changes. The declarative view makes your code more predictable and easier to debug.

A react application is made of multiple components, each responsible for rendering a small, reusable piece of HTML. Components can be nested within other components to allow complex applications to be built out of simple building blocks. A component may also maintain an internal state – for example, a TabList component may store a variable corresponding to the currently open tab. In this project, we will use this framework to implement the user interface of the application.

2.1.2. NodeJS

NodeJS [8] is an open-source and cross-platform JavaScript runtime environment. NodeJS runs the V8 JavaScript engine, the core of Google Chrome, outside of the browser, so NodeJS is very performant.

A NodeJS app is run in a single process, without creating a new thread for every request. NodeJS provides a set of asynchronous I/O primitives in its standard library that prevent JavaScript code from blocking and generally, libraries in NodeJS are written using non-blocking paradigms, making blocking behavior the exception rather than the norm. In this project, we will use NodeJS for building the backend side (APIs) of the application.

2.1.3. MAPBOX API

Mapbox [9] is a developer platform used across industries to create custom applications that solve problems with maps, data, and spatial analysis. Mapbox's tools are building blocks that support every part of the web and mobile map-making process. Mapbox provides methods for embedding maps easily into the application.

2.1.4. GOOGLE FIREBASE

Google Firebase [10] is Google-backed application development software which allows developers to develop Android, IOS, and Web apps. For reporting and fixing app crashes, tracking analytics, creating marketing and product experiments, firebase provides several tools.

Google Firebase offering:

- Real-time database: Store and sync data with our NoSQL cloud database. Data is synced across all clients in real time, and remains available when your app goes offline.
- Authentication: Firebase Authentication provides backend services, easy-to-use SDKs, and ready-made UI libraries to authenticate users to your app. It supports authentication using passwords, phone numbers, popular federated identity providers like Google, Facebook and Twitter, and more.
- Cloud storage: Cloud Storage is built for app developers who need to store and serve user-generated content, such as photos or videos.

2.1.5. OTHER TECHNOLOGIES

Bootstrap: Bootstrap is the most popular CSS Framework for developing responsive and mobile-first websites. Bootstrap 4 is the newest version of Bootstrap.

React feather: is a collection of simply beautiful open source icons for React. Each icon is designed on a 24x24 grid with an emphasis on simplicity, consistency and readability.

MongoDB: is a document database designed for ease of development and scaling. The Manual introduces key concepts in MongoDB, presents the query language, and provides operational and administrative considerations and procedures as well as a comprehensive reference section.

Elasticsearch: is a distributed, open source search and analytics engine for all types of data, including textual, numerical, geospatial, structured, and unstructured.

Draw.io: is proprietary software for making diagrams and charts.

Github: is a website and cloud-based service that helps store and manage code, as well as track and control changes of code.

Github Action: are a new feature of Github that allow set up CI/CD workflows using a configuration file right in Github repo. We will use this feature to automatically update production server.

RestfulAPI (Representational State Transfer): is architectural style for distributed hypermedia systems and was first presented by Roy Fielding in 2000 in his famous dissertation. Principles of REST:

- Client–server: By separating the user interface concerns from the data storage concerns, we improve the portability of the user interface across multiple platforms and improve scalability by simplifying the server components.
- Stateless: Each request from client to server must contain all of the information necessary to understand the request, and cannot take advantage of any stored context on the server. Session state is therefore kept entirely on the client.
- Cacheable: Cache constraints require that the data within a response to a request be implicitly or explicitly labeled as cacheable or non-cacheable. If a response is cacheable, then a client cache is given the right to reuse that response data for later, equivalent requests.
- Uniform interface: By applying the software engineering principle of generality to the component interface, the overall system architecture is simplified and the visibility of interactions is improved. In order to obtain a uniform interface, multiple architectural constraints are needed to guide the behavior of components. REST is defined by four interface constraints: identification of resources; manipulation of resources through representations; self-descriptive messages; and, hypermedia as the engine of application state.
- Layered system: The layered system style allows an architecture to be composed of hierarchical layers by constraining component behavior such

that each component cannot "see" beyond the immediate layer with which they are interacting.

- Code on demand (optional): REST allows client functionality to be extended by downloading and executing code in the form of applets or scripts. This simplifies clients by reducing the number of features required to be pre-implemented.

2.2. RELATED WORK

No	Name	Search	Map Function	Online chat	Post interaction
1	Real estate [3]	Yes	No	No	Yes
2	Ohanaliving [4]	Yes	No	No	No
3	timphongtrovn [5]	Yes	No	No	No
4	Vstay	Yes	Yes	Yes	Yes

Table 1. Comparing between VSTAY and other applications

VSTAY is a very helpful application when comparing with others. We use new technologies to implement this application to make it work properly and has high performance. VSTAY has a friendly UI/UX which makes it easier to use. Moreover, online chat function will connect users and authors of posts, this functions make VSTAY become an useful application, it saves time and secure because users can directly communicate with the authors to ask for the accommodation. Post interaction is a wonderful function of VSTAY, it helps users to know that post is reliable or not by seeing the amount of likes, dislikes and views of a post. It supports users perfectly to not only find accommodation but also help them to find customers, owners of accommodation such as: boarding houses, houses or hotels can upload posts to describe their accommodations, others will see the posts and contact them via social networks or chat function.

CHAPTER 2: DESIGN AND IMPLEMENTATION OF VSTAY SYSTEM

2.1. OVERVIEW OF VSTAY

VSTAY is a website application that supports users for searching and renting accommodation, it also provides functions that allows owners of motels or boarding houses to find their ideal customer. The system will make the private information of users secure by encoded. It also provides functionalities: uploading post, online chat, like or dislike post, search post, explore and search on map. The application will be implemented based on "Guest", "Member" and "Moderator".

When "Guest" access the website, they can watch uploaded posts about renting or needing accommodation, uploaded posts will be shown by title, figures and details of information (number of rooms, width, height,etc). Moreover, all of the sharing and renting posts will be shown detaily on map using the open-source MapBox API in exploring map function, "Guest" can watch and select them easily.

"Guest" has to sign up for an account to work on advanced functions. When "Guest" is signing up, they have to provide a real email address, unique nickname and password or they just need to authenticate their gmail address and enter password, if email or nickname is duplicated with other users or password is invalid, the system will show an error notification. "Guest" can update private information after signing up except email and nickname because these are unique, they can also update their password. After signing up, "Guest" will become "Member", they can watch posts with detailed information and interact with them such as: like and dislike. They can also chat with authors of the post to ask about accommodation information or bargain. "Member" can upload a post by describing details of information and that post will be a pending post.

Pending posts will be previewed by the "Moderator" before it is public for everyone on the website. Posts could be removed by authors. This step will make uploaded posts from users more reliable because "Moderator" will preview the detail to make it clear before it is public.

"Moderator" will be granted full functionalities on the application and interact with the database.

2.2. MAIN FUNCTIONS

No	Actor	Functions
1	Guest	- Sign up - Sign in - Watch uploaded posts - Search
2	Member	- Sign in - Online chat - Watch detail of posts - Like/dislike posts - Update private user's information - Upload posts - Update or remove owned posts - Watch/Search uploaded posts on map
3	Moderator	- Manage posts - Interact directly with database

Table 2. Main functions of system

2.3. USE CASES DIAGRAMS

2.3.1. GUEST

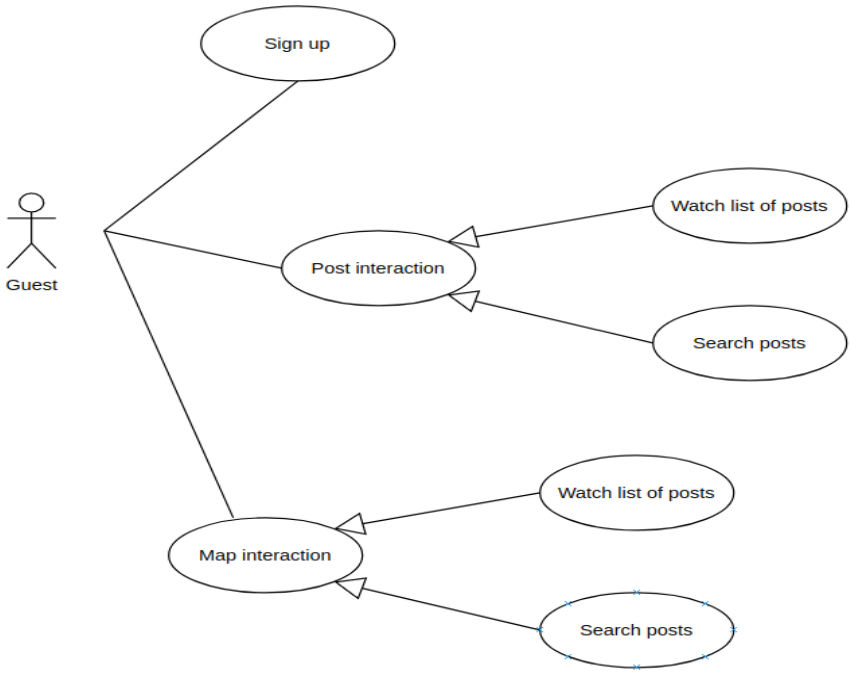


Figure 1. Use case diagram of guest

"Guest" is a guy who accesses the application and uses functions without authentication such as: sign up, watch list of posts, search and map interaction. Users can watch the list of posts on the main page, but if they click to see the detail of that post, the system will redirect them to the login page. Map interaction allows users to see all of the seeking and renting posts of the system on the map, users can search by location (city, district or address), the map will display the result directly. Moreover, users can select accommodations and go to the post page to see the details. Post page will show the information of accommodation for users to make choices, the system also allows users to interact with posts to evaluate to like or dislike posts.

2.3.2. MEMBER

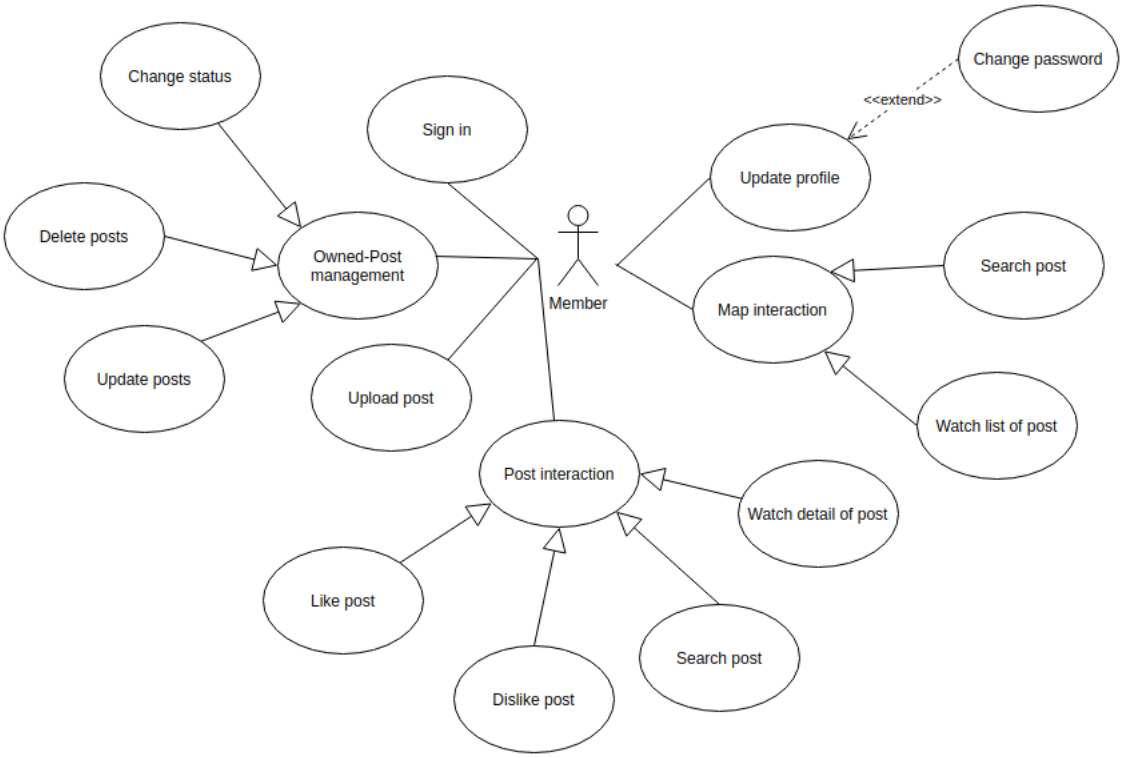


Figure 2. Use case diagram of guest

After signing up, users will become “Member”, they can not only use functions of “Guest” users but also advanced functions: watch detail of posts, like or dislike posts, search posts, interact with map, send message to author of post, upload post, update profile, change password and manage owned-post (update or delete).

2.3.3. ADMIN

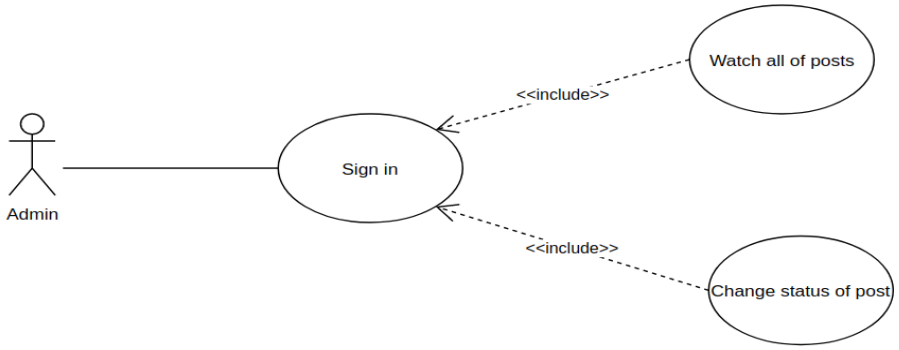


Figure 3. Use case diagram of guest

"Admin" who has full rights on the system and interacts directly with the database. "Admin" can login and manage uploaded posts from users, they can watch a list of uploaded posts from "Member", preview and decide to make it public or not.

2.4. DATABASE DESIGN

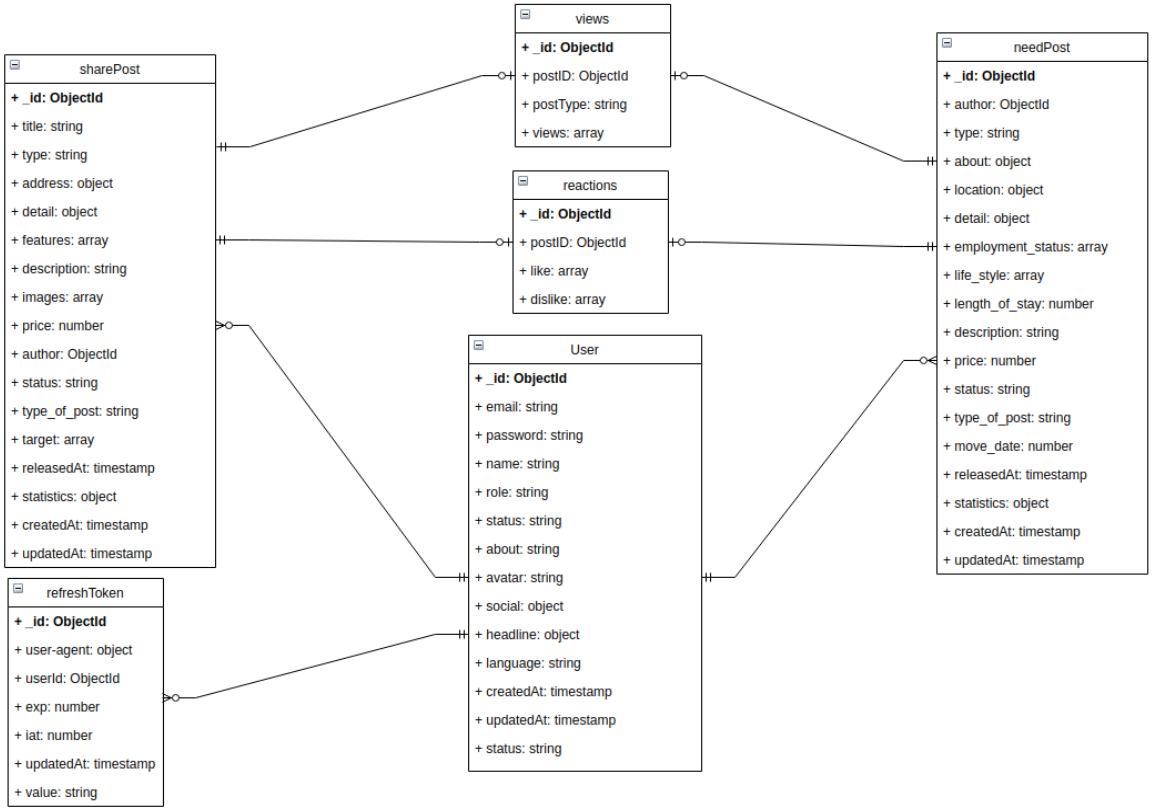


Figure 4. Entity relationship diagram

We used MongoDB to store data from VSTAY. There are 6 collections in the database.

2.5. UI/UX DESIGN

UI/UX design includes the way that the application interacts with users, the overall design and how information is presented. It is the ever-evolving relationship between users and the system. UI/UX design helps developers know and understand what they are going to implement and how the application displays. This is the basic building blocks of how the application is set up and functions when visited by the audience.

In this application, we use Figma - a vector-based tool that lives in the cloud, allowing users to work anywhere from a browser. It's a zippy tool that is made for design, prototyping, collaboration, and organizational design systems.. It's made so that users can collaborate on projects and work pretty much anywhere. Figma is used by some big brands including Slack, Twitter, Zoom, Dropbox, and Walgreens. Those names alone show that this tool is solid enough to power almost any project. With Figma, we can easily create a sketch UI/UX of a page in the application, preview it and make a decision to implement it or not.

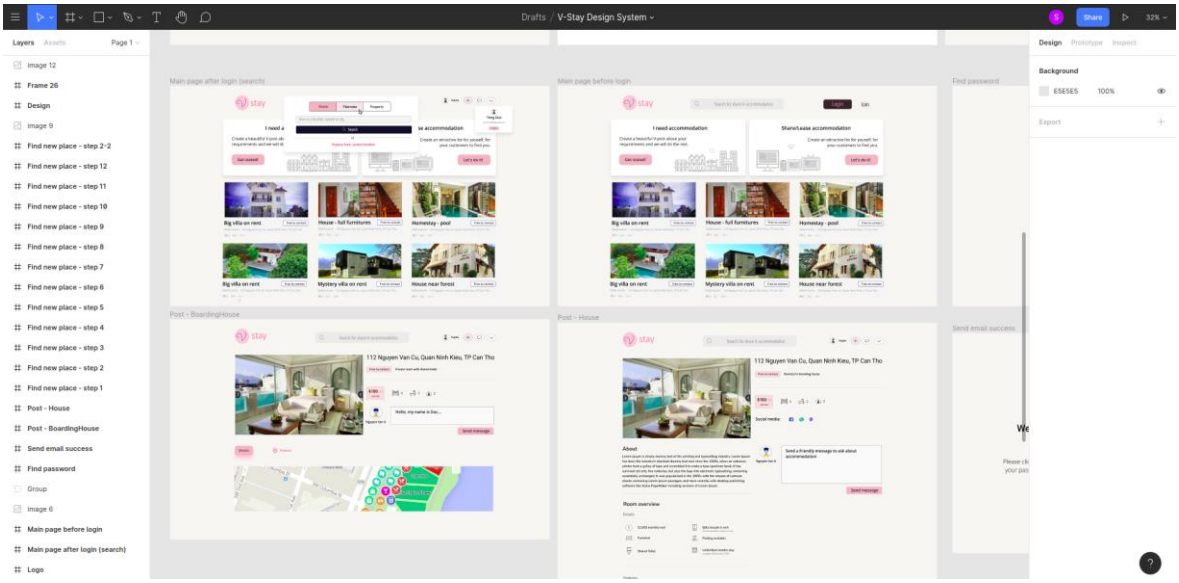


Figure 5. Figma design page

Before implementing the system, we will design all of the pages of VSTAY by Figma.

2.6. DETAIL OF FUNCTIONS IN THE SYSTEM

2.6.1. Authentication

Sign up function

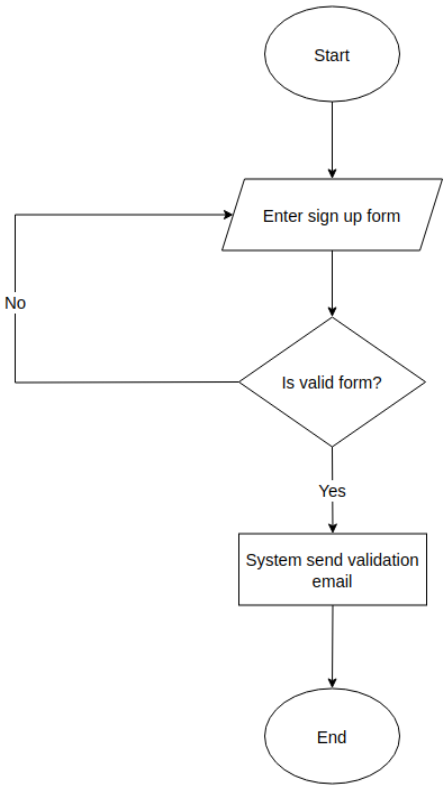


Figure 6. Flowchart of sign up function

When users sign up, they have to enter their real email address, nickname and password. To verify a real email address, the system will send a verification email to their signed up email address, users have to click a button in that email to create their account. If there are some errors in the signing up process, the system will show notifications.

Sign in function

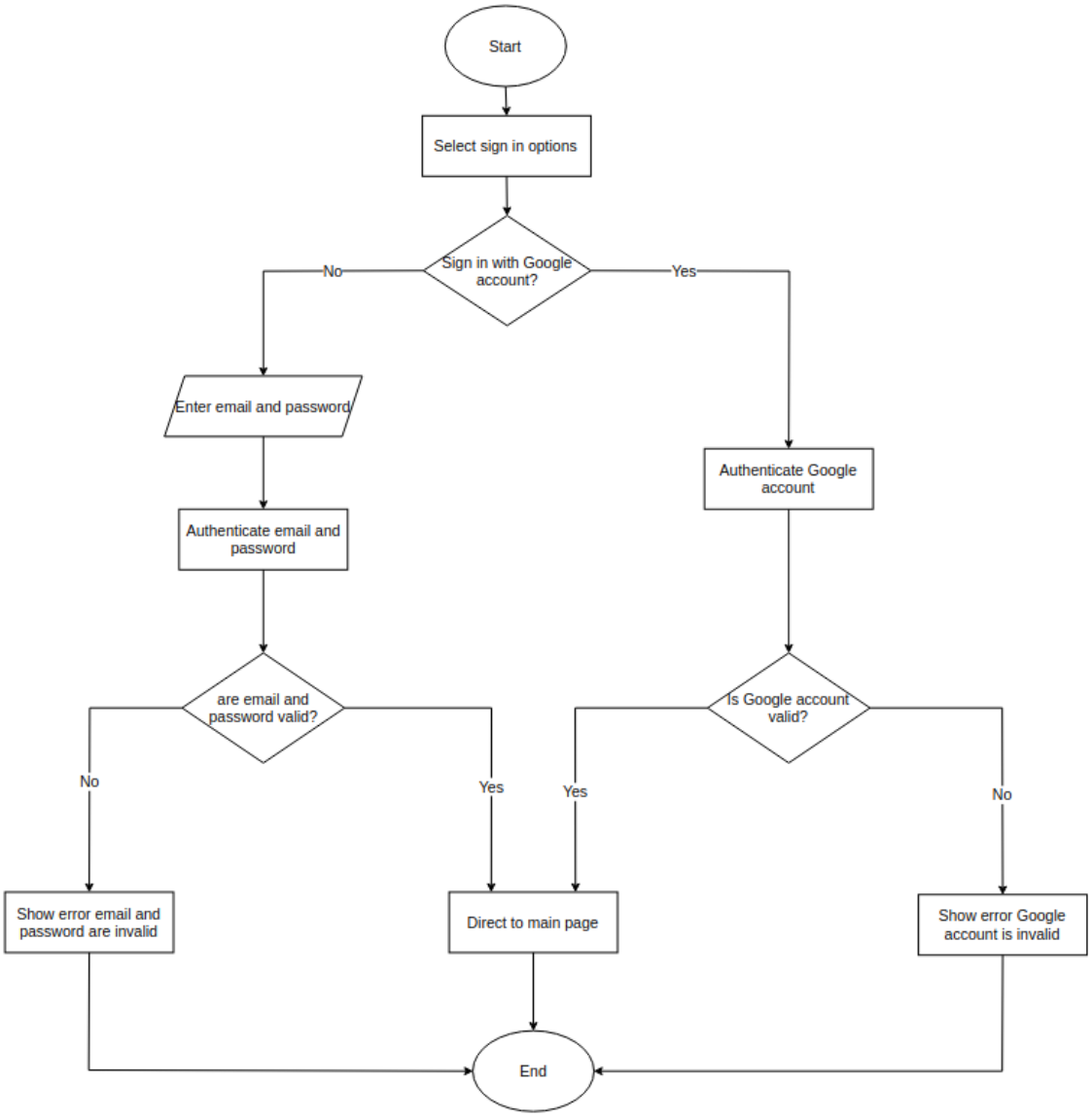


Figure 7. Flowchart of sign in function

Users will select options to sign in. If users select sign in with signed up email and password, the system will authenticate, its email and password are both valid and in the database, the system will direct users to the main page. Otherwise, the system will show an error message. If users select sign in with Google Account, the system will use Firebase authentication to authenticate Google account, if their account is valid, it will direct users to the main page, if not, it will show an error message.

2.6.2. Post function

Post is a type of information that contains details of property and is uploaded by members. There are two types of Post in Vstay: Share post and Need post, Share post will be uploaded by users who need to share or rent accommodation such as: room in boarding house, room in house or full property, Need post will be uploaded by users who need accommodation. Post makes users have an overview about the property and helps them to make the best decision.

Post will be published after the admin previews the details to make sure it does not contain unclear information, after that, the admin will decide to approve or reject it.

Posts searching functionality

To search posts, users need to enter keywords in the input box, after that, pressing "Enter" and clicking search button. The system will compare with name, address and description in the database with the keywords. If it returns empty, the system will show a notification about no result found.

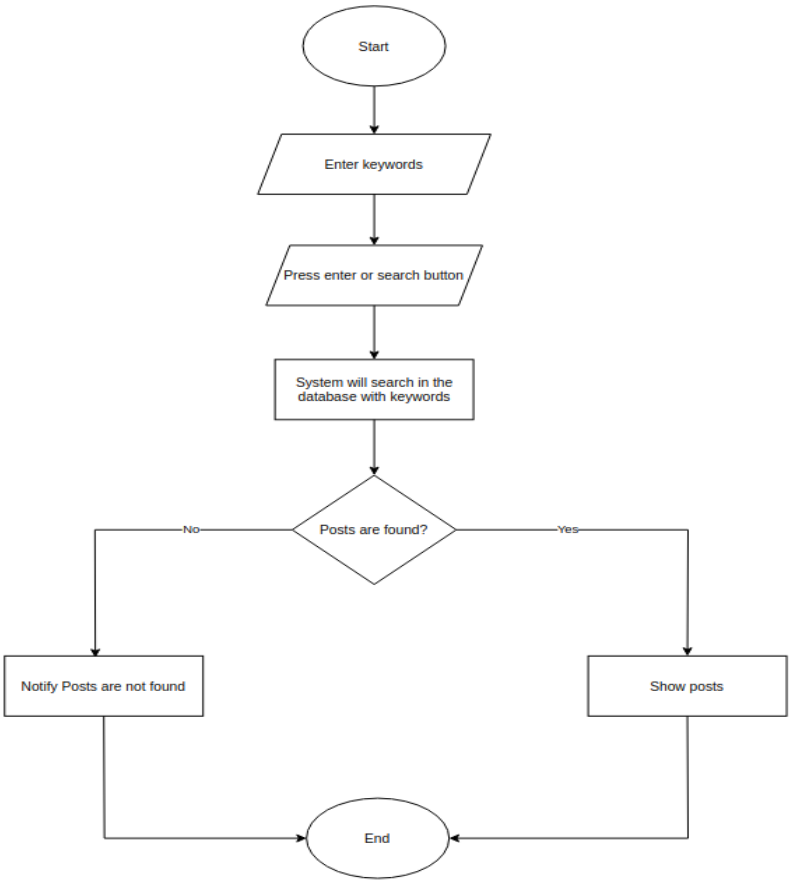


Figure 8. Flowchart of searching function

2.6.3. Online chat

Online chat is a function that allows users to communicate with others and talk about the information of accommodations. When users access the post page, they can click the button "Send message" to chat with the author of that Post and discuss the information.

Google Firebase will be used to serve online chat. It is a high performance, free framework and easy to intergrate with the application. We will use Firestore - A part of the Google Firebase application development platform which is a NoSQL document database and real time updates to store data.

Structure of data in Firestore

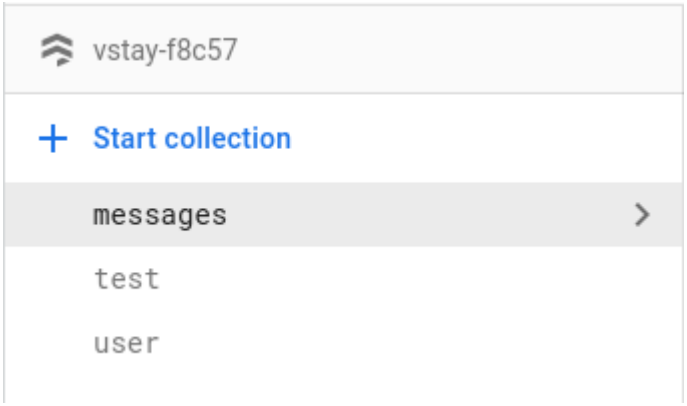


Figure 9. Structure of data in Firestore

"messages" collection will be used to store messages and information between two users. When a user starts to send a message, the system will automatically create a document in this collection, the structure of the document will be described below.

"user" collection will be used to store information of users. When a user starts to send messages, the system will check, if that user is not in this collection, it will automatically create a document about this user. Otherwise, the system will update the database.

"test" collection will be used for testing in development mode.

Figure 10. Document name in "message" collection

In figure 7, the name of each document is combined from two user's id, it represents a conversation between two users. When user A starts to chat with user B, the system will hash each user's id by Rolling Hash - a hash function that transforms a string of characters into a 32 bit integer known as a hash of that string, after that, two hashed integers will be compared and generated to id of conversation's id by this method: if user A's hashed id is greater than user B's hashed id, the conversation's id will be "A's id - B's id", otherwise it will be "B's id - A's id". When a user sends a message to another user, the system will automatically compare two ids and create a conversation in Firebase with that id.

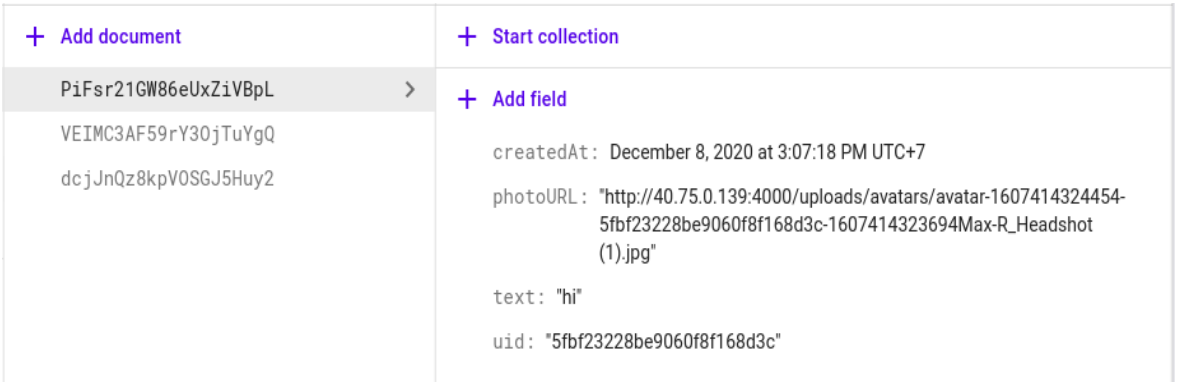


Figure 11. Information of each document in Firebase

Each message will be saved in each document. When a user sends a message, the system will create a document that represents a message and contains: id of user, content and timestamp that message is created.

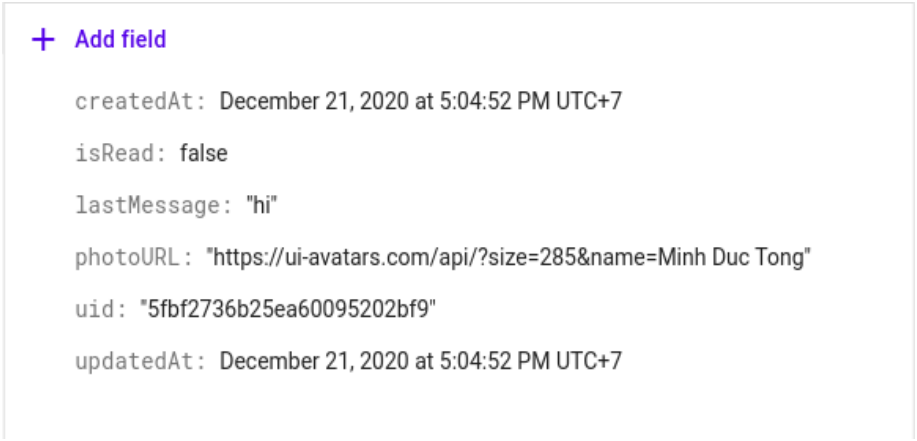


Figure 12. Detail of each user in Firebase

Each user will be saved some important information to display in the chat page. When users start a message, the system will automatically create a document. If this user exists, it will update "lastMessage", "isRead" and "updatedAt".

2.6.4. Map

Map of Vstay will show locations of all accommodations of both type need and share post. There is an input on the map for users to search by address, users have to enter keywords and the map will move to that address. If users click to a marker, the application will direct the user to the post page.

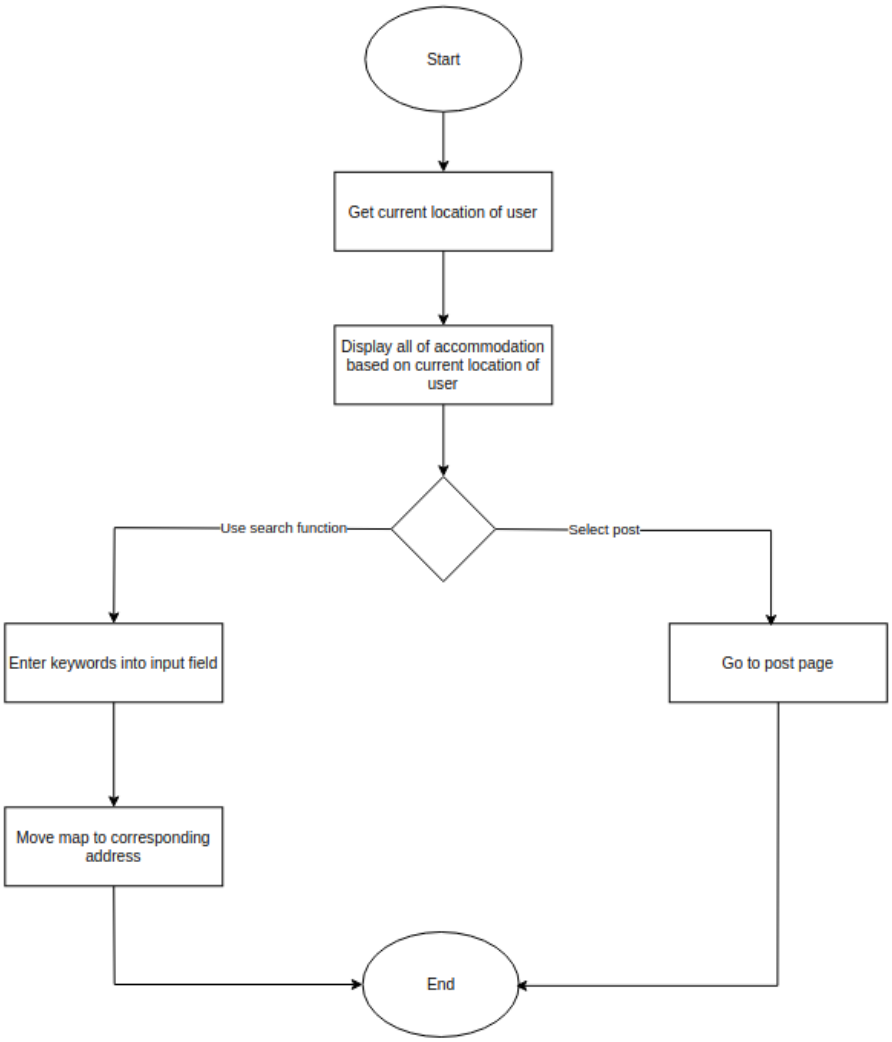


Figure 13. Flowchart of map function

2.6.5. User interface

a. Navigation bar

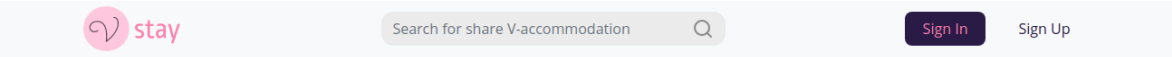


Figure 14. Navigation bar before signing in

Search bar, sign in button and sign up button will be displayed on the navigation bar before signing in.

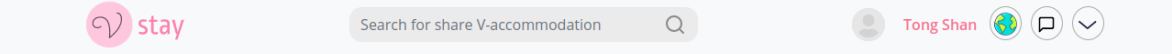


Figure 15. Navigation bar after signing in with normal account

After signed in, users will see the profile, map button and chat button.

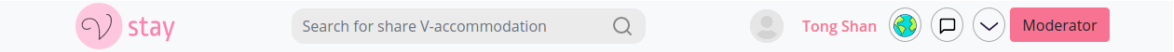


Figure 16. Navigation bar after signing in with moderator account

After signed in with moderator account, the navigation bar will display a button to direct users to moderator page

b. Main page

Main page will be shown when users access the application. The navigation bar shows: name and logo of the application, search bar, sign in and sign up. Under the navigation bar, there are two big buttons that show two main functions of Vstay - uploading posts. The button with title "I need accommodation" is for users who need accommodation and they will upload posts to show their need with the others. The button with title "Share/Rent accommodation" is for users who want to share or rent accommodation, they will upload posts and show the details of their property to attract users.

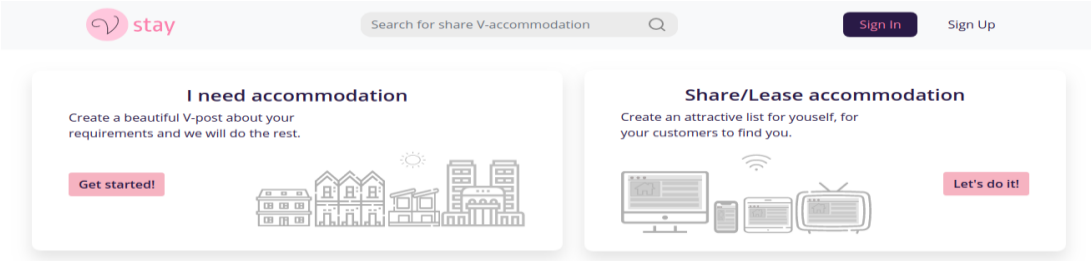


Figure 17. Navigation of main page

Sharing accommodation section of the main page will show all of the approved share posts, each post will display the basic information of each post: title, price, features,etc It will give an overview of accommodation for users to choose. To watch details of accommodation, users have to sign in.

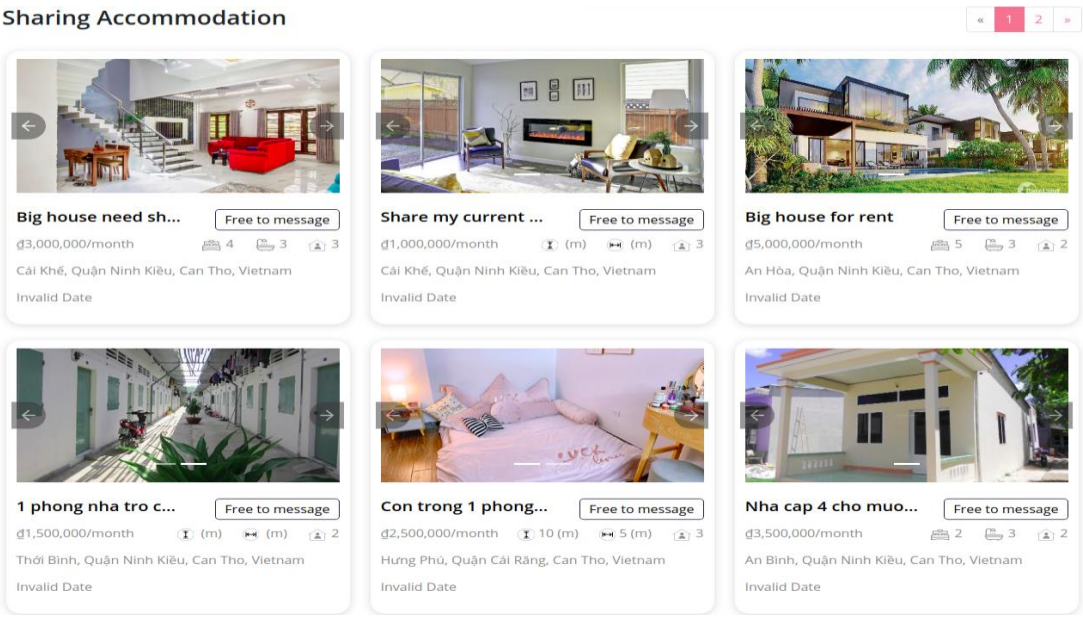


Figure 18. List of posts on main page

c. Sign in page

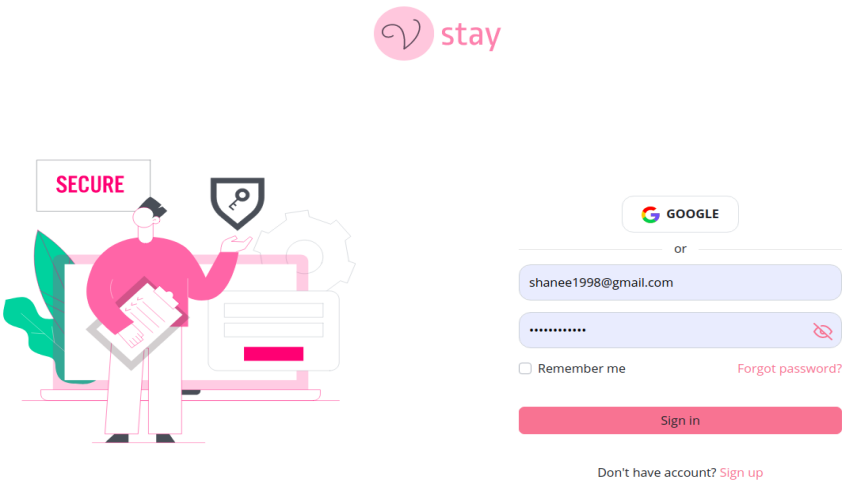


Figure 19. Sign in page

Users have two options to sign in. First, users can sign in with their email and password that they signed up before, if email or password is invalid, the system will show the error. Second, they can sign in with Google account, if their Google account is not in the database, the system will redirect them to sign up page.

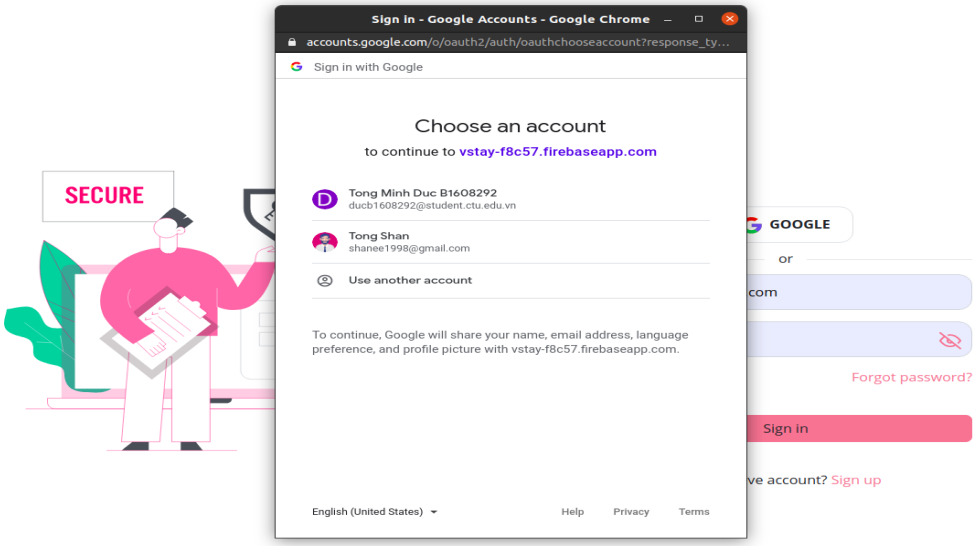


Figure 20. Sign in with Google account

When users select an option to sign in with a Google account, the system will use Authentication of Google Firebase to check their email, if the account is valid, VSTAY will direct users to the main page.

d. Sign up page

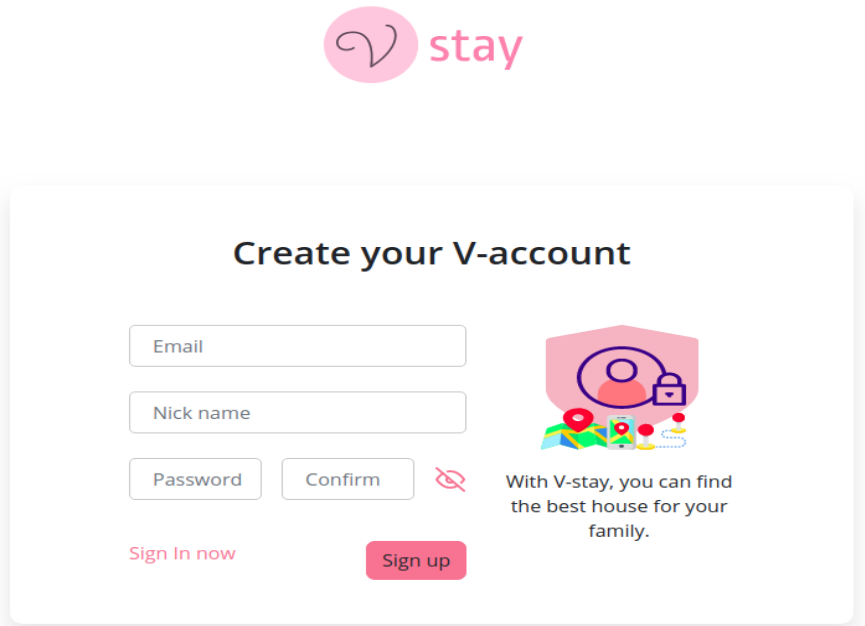


Figure 21. Sign up page

To become a "Member", users have to sign up with email, nickname and password. Email and nickname of the user must be unique for identification, if users enter duplicated email and nickname, the error message will be shown.

e. Upload sharing post

When users want to find customers, they can use this function to upload posts and describe their accommodation in detail. The process of uploading will be splitted to multiple steps.

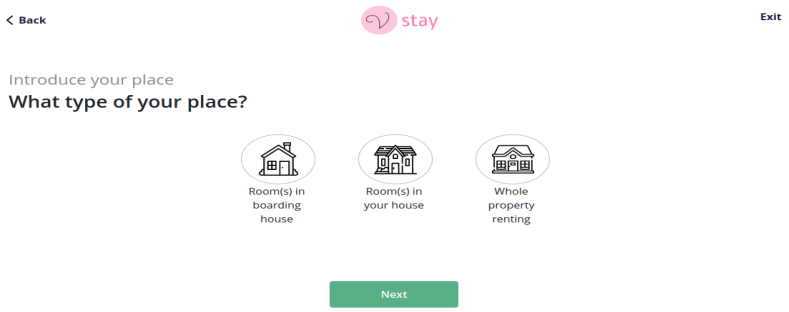


Figure 22. Select type of share post

First of all, users have to choose the type of their accommodation, there are three types: Room(s) in the boarding house, Room(s) in your house and the whole property for renting.

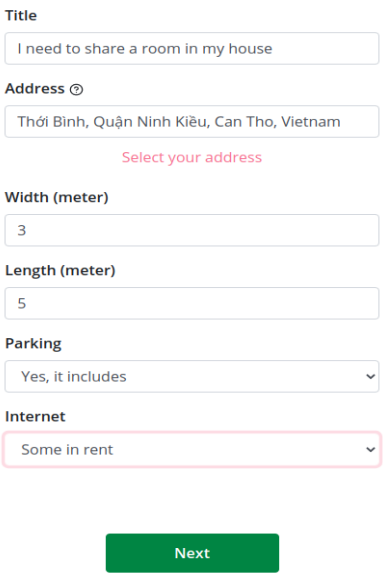


Figure 23. Enter information of post

Secondly, users have to enter the details of the post: title, address, width and length of accommodation, parking and internet.

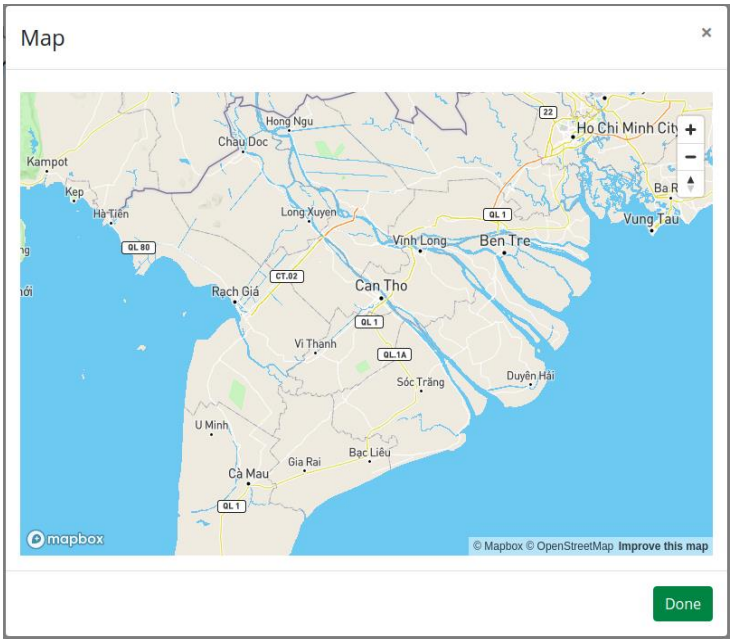


Figure 24. Select location of post

When users select an address, a map popup (Figure 22) will be displayed for them to select the location on the map. The address will be automatically added to the address field using Mapbox API.

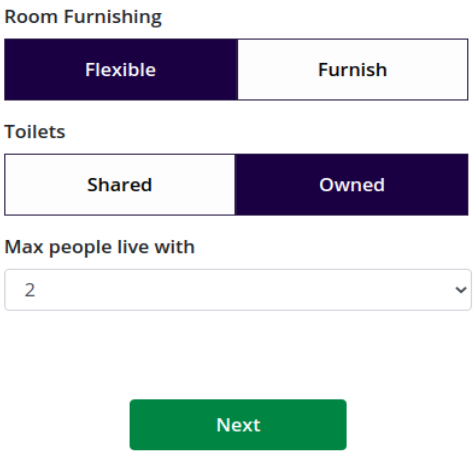


Figure 25. Select room furnishing of post

Thirdly, users have to select whether the accommodation is allowed to furnish or not, the toilet is shared or owned and the number of people living in the accommodation.

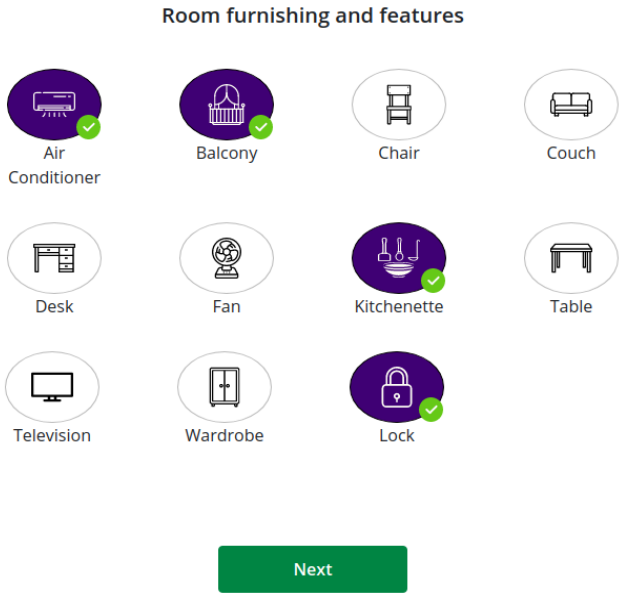


Figure 26. Select features of accommodation

Fourthly, users will choose the features that are available in the accommodation. If there are no features in accommodation, users can click "Next" button without selecting any features.

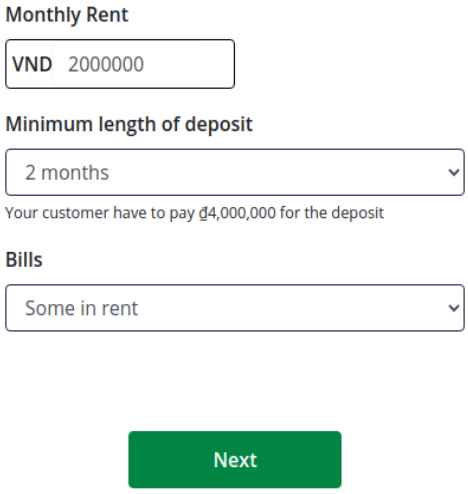


Figure 27. Select rent and bills

Thirdly, users will enter the cost of renting in VND and the length of deposit that others have to pay when they accept to rent. After users enter the monthly rent and deposit, the system will show the cost that users have to pay. Then, users will define whether the bills of water and electricity are included in the bills or not.

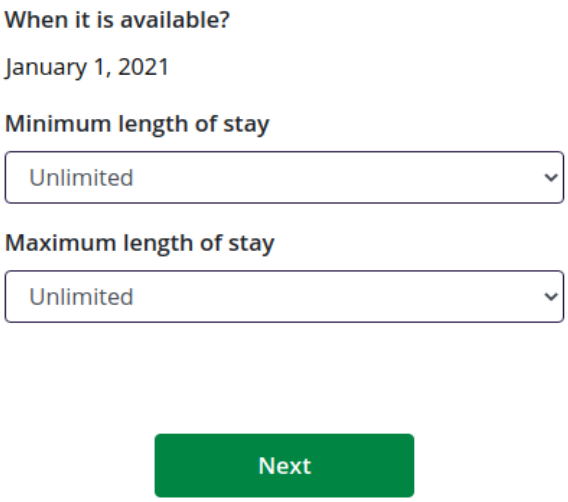


Figure 28. Select room availability

Sixthly, users will define the availability of accommodation. When the accommodation is available for others to rent, what minimum length of stay and maximum length of stay that they allow.

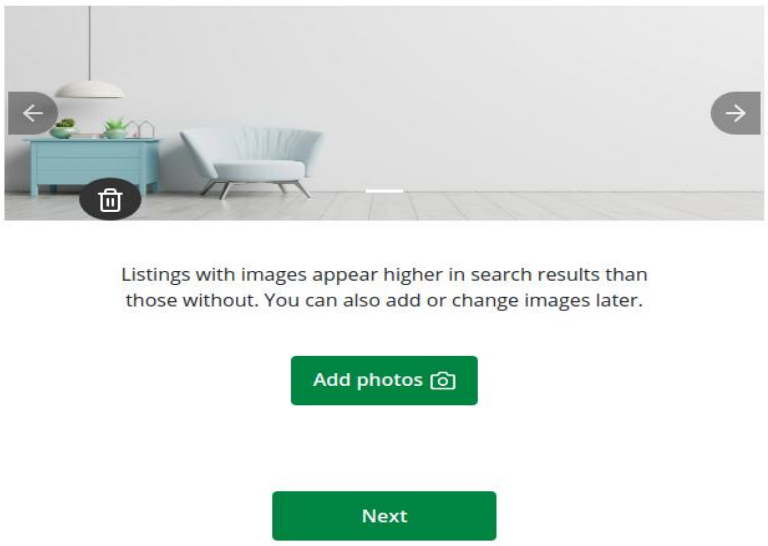


Figure 29. Select images for accommodation

Seventhly, users will add images of accommodation which show the detail of accommodation, images will make the post more attractive because other users can preview accommodation.

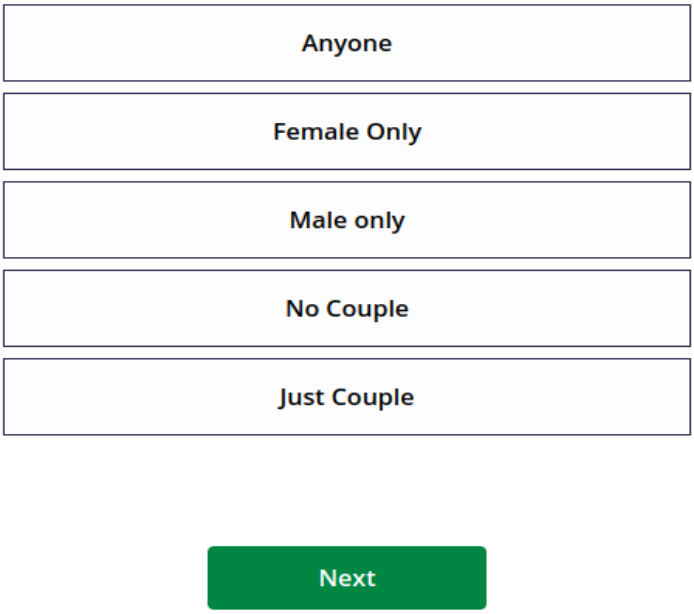


Figure 30. Select customers preference

Eighthly, users will choose the customer preference that they expect. Users can select "Anyone" if they do not expect any preference from their customers, otherwise, they can select other options to tell their customers about what they expect.

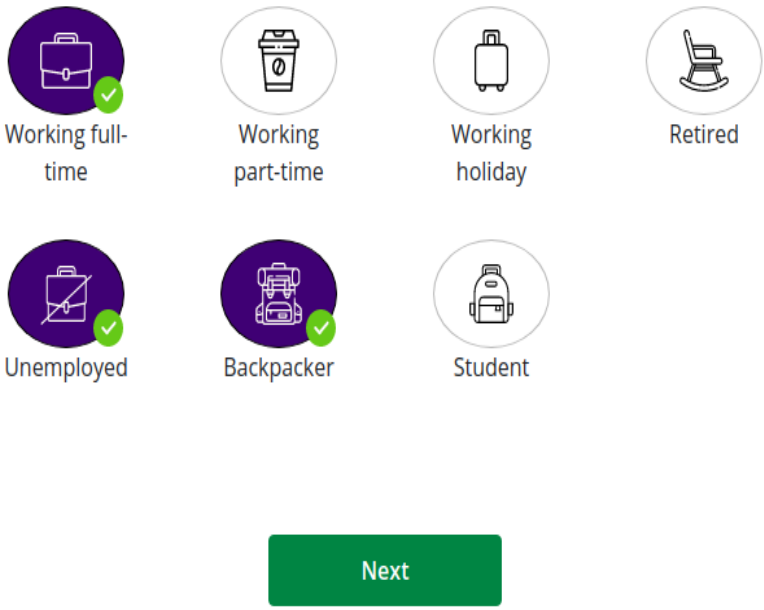


Figure 31. Select employment status of customer

Ninethly, users will select what employment status of customers that they expect.

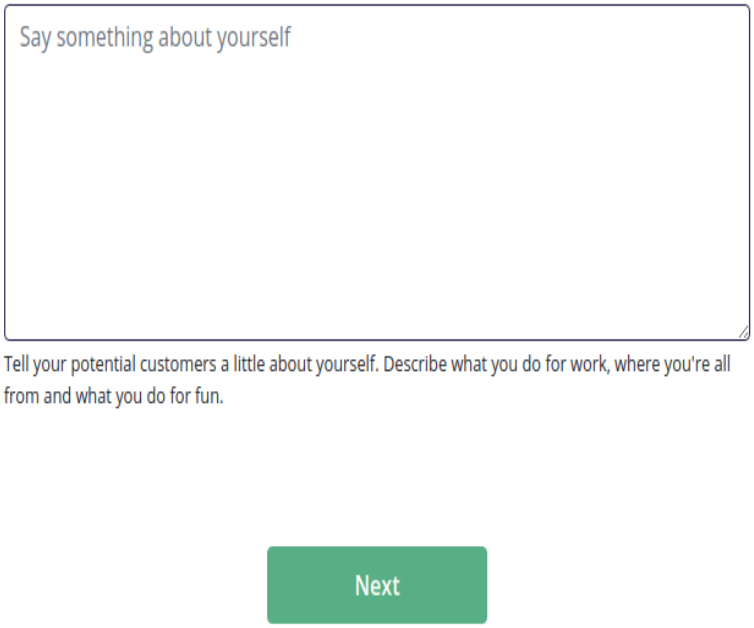


Figure 32. Enter description of share post

Finally, users have to enter a short description about this accommodation, tell what they expect or what is included.

f. Upload needing post

When users want to find accommodation, they can use this function to upload posts and describe what they expect about accommodation in detail and the others will contact them if their accommodation. The process of uploading will be splitted to multiple steps.

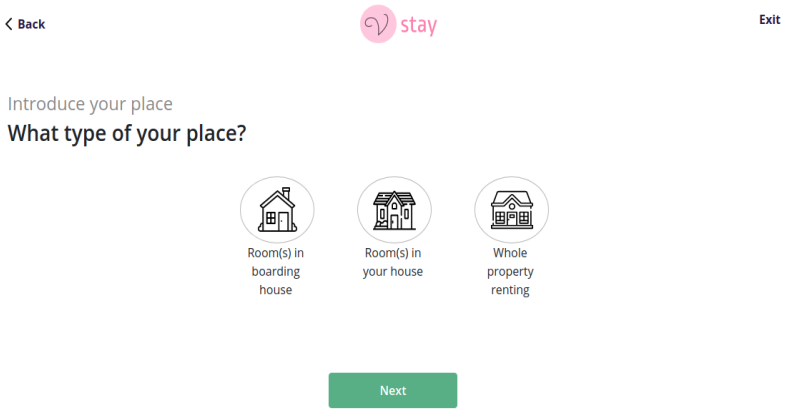


Figure 33. Select type of place for need post

First of all, users have to choose the type of accommodation that they want, there are three types: Room(s) in the boarding house, Room(s) in your house and the whole property for renting.

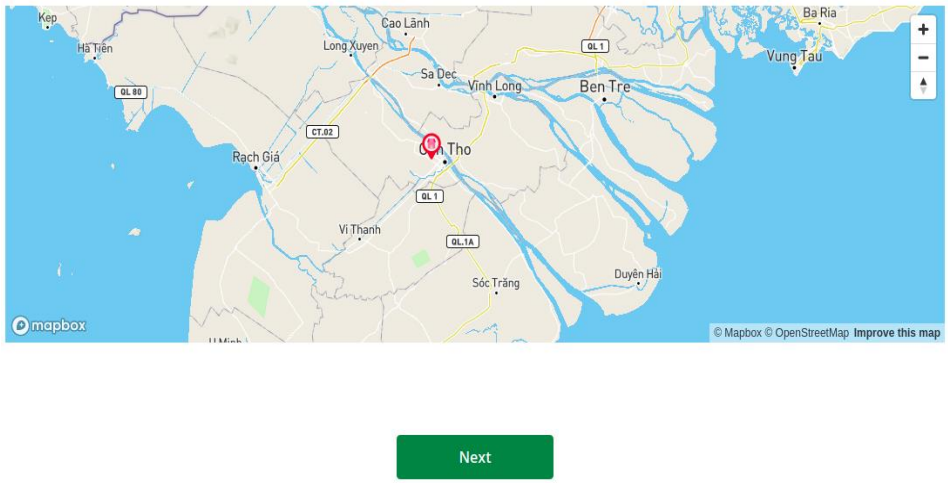


Figure 34. Select location for needing accommodation post

Secondly, users will select the location that they want to find accommodation.

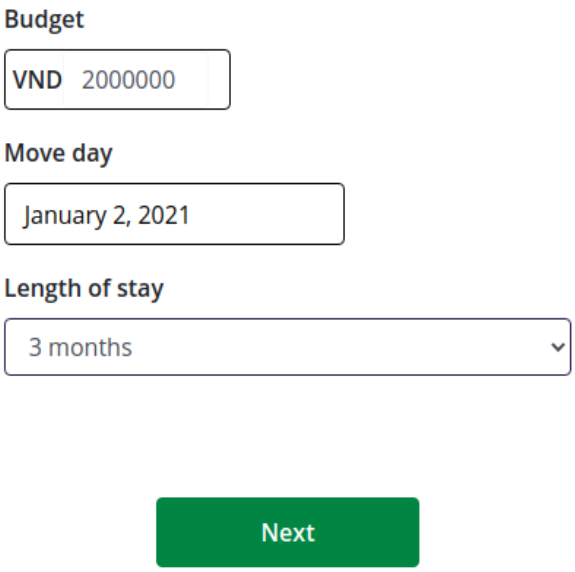


Figure 35. Select rent and timing for needing accommodation post

Thirdly, users will define their budget, day of moving and length of stay.

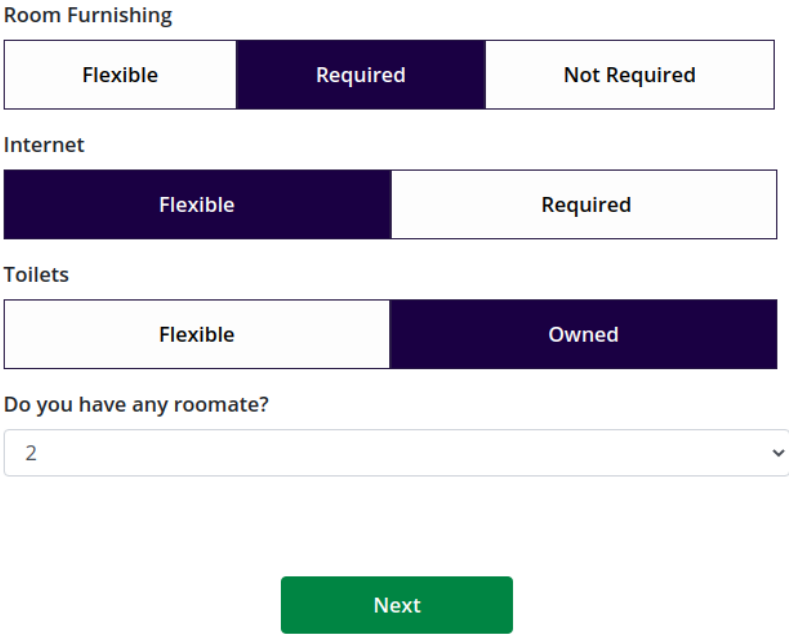


Figure 36. Select property reference for needing accommodation post

Fourthly, users will select property preference that they want to rent: room furnishing, internet and toilets.

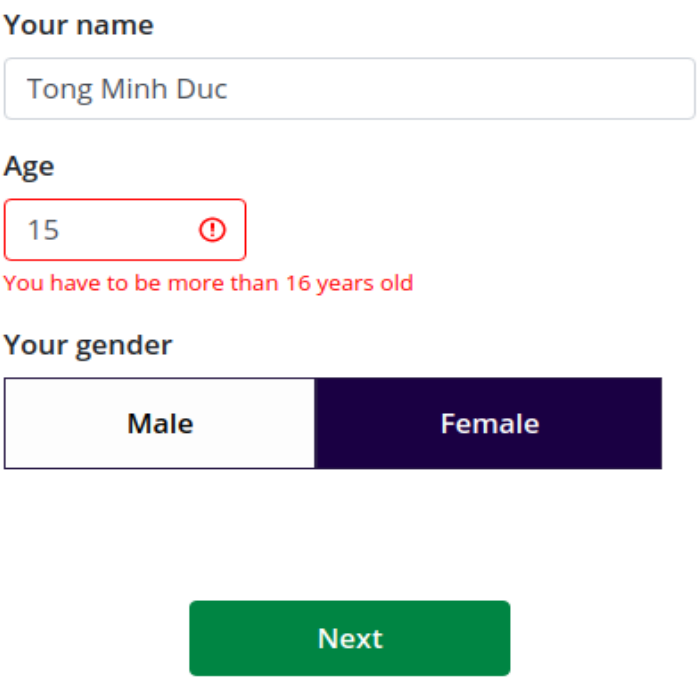


Figure 37. Enter information for author of post

Fifthly, users will enter private information. When users enter age, they have to make sure that field is greater than 16 or it will show the error notification.

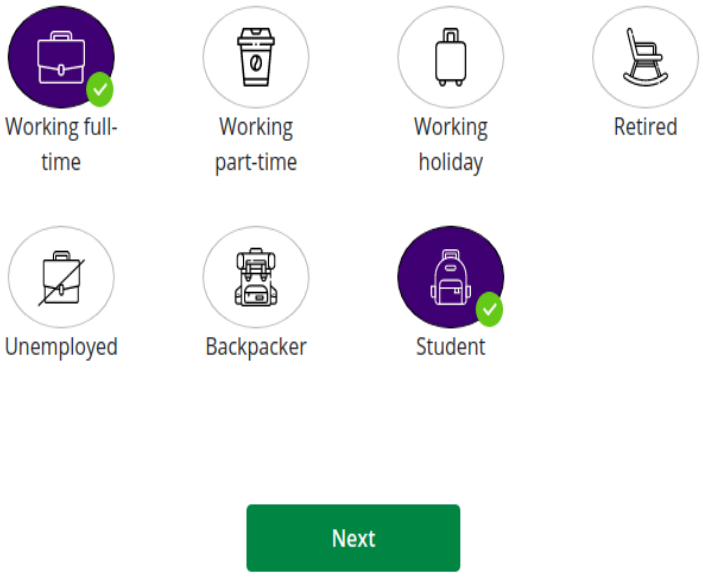


Figure 38. Select employment status

Sixthly, users will select their employment status. The employment status will make other users know what employment status of this users, this information will make others know more about the author.

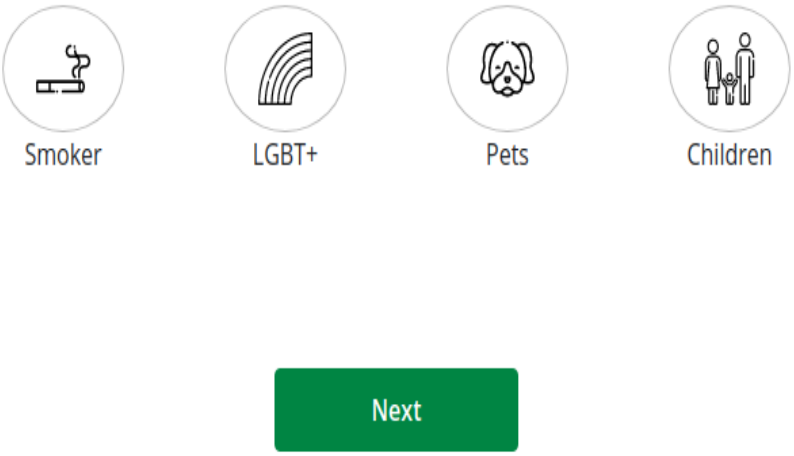


Figure 39. Select lifestyle of author

Seventhly, users will select their lifestyle.

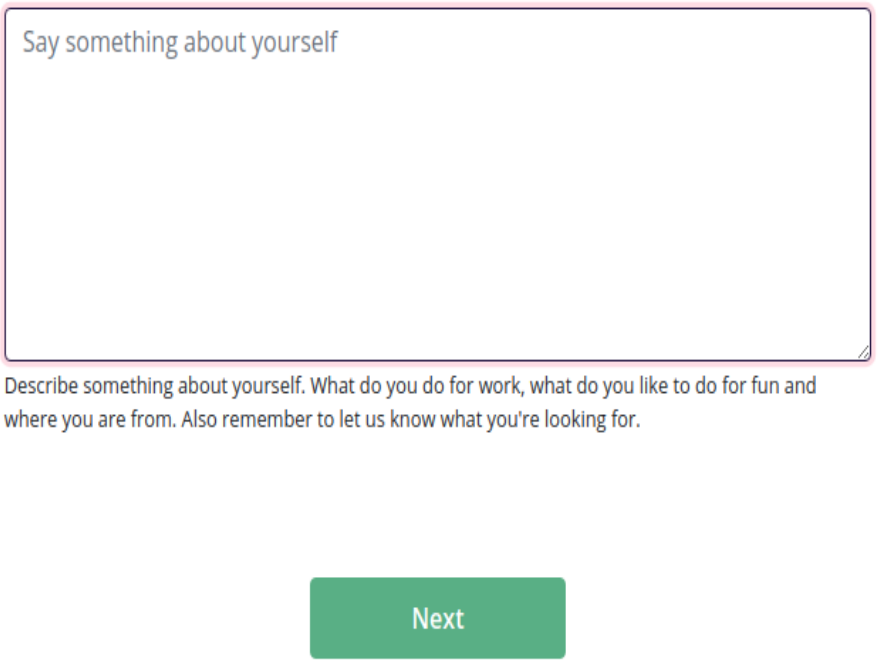


Figure 40. Describe information in detail

Finally, users will describe their information. They can describe shortly about who they are, what their hobbies are or how long they live.

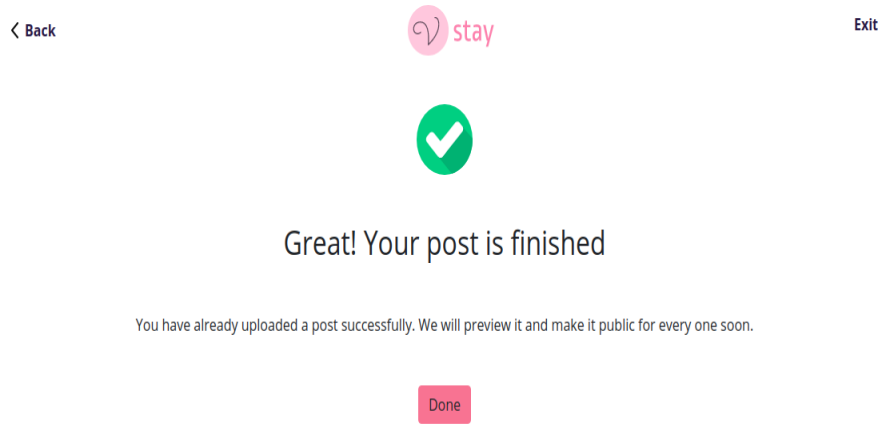


Figure 41. Describe information in detail

After finishing everything, the system will show that the post is uploaded successfully.

g. Renting accommodation post page

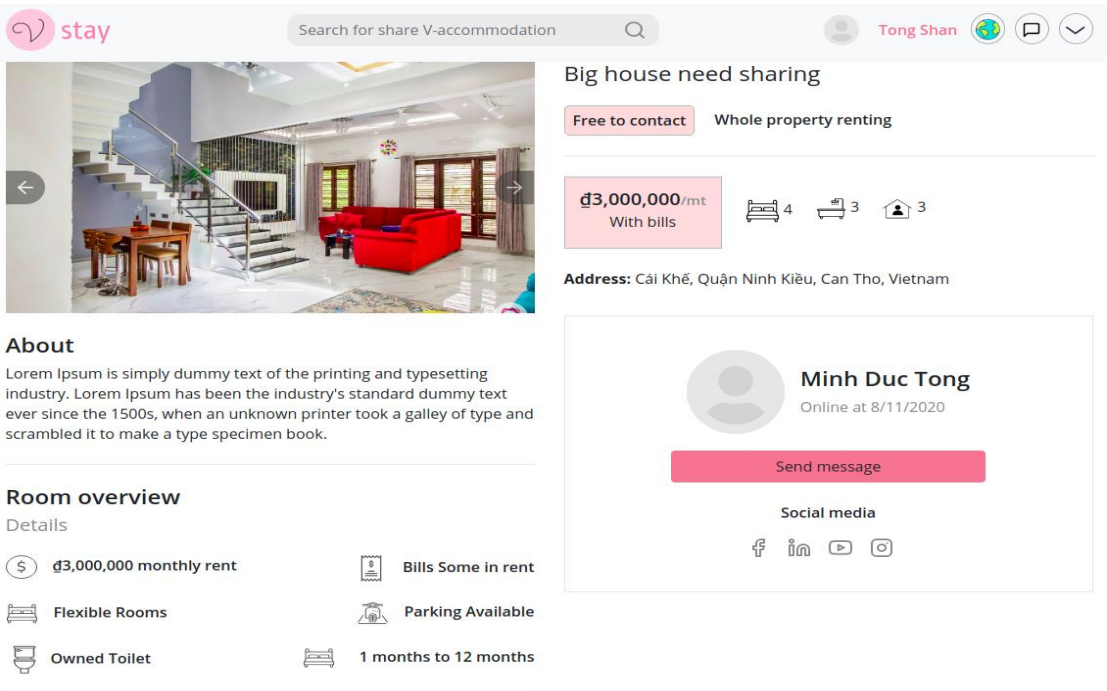


Figure 42. Renting accommodation post page

Renting accommodation post page will display the details of accommodation. Moreover, when accessing this page, they can send messages directly to the author to discuss the information.

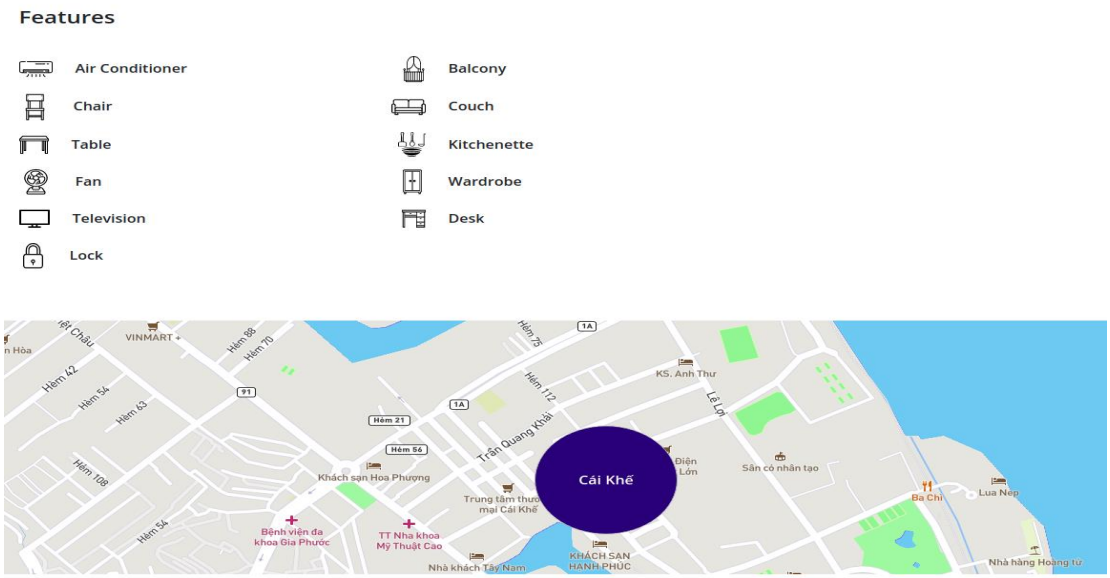


Figure 43. Renting accommodation post page (2)

h. Seeking accommodation post page

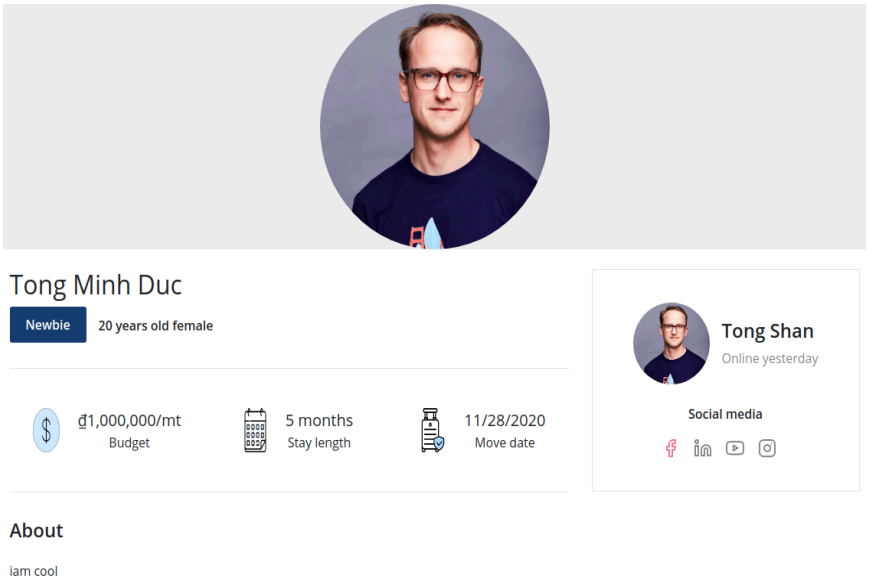


Figure 44. Seeking accommodation post page

Seeking accommodation post page will show the details of accommodation that users want to find.

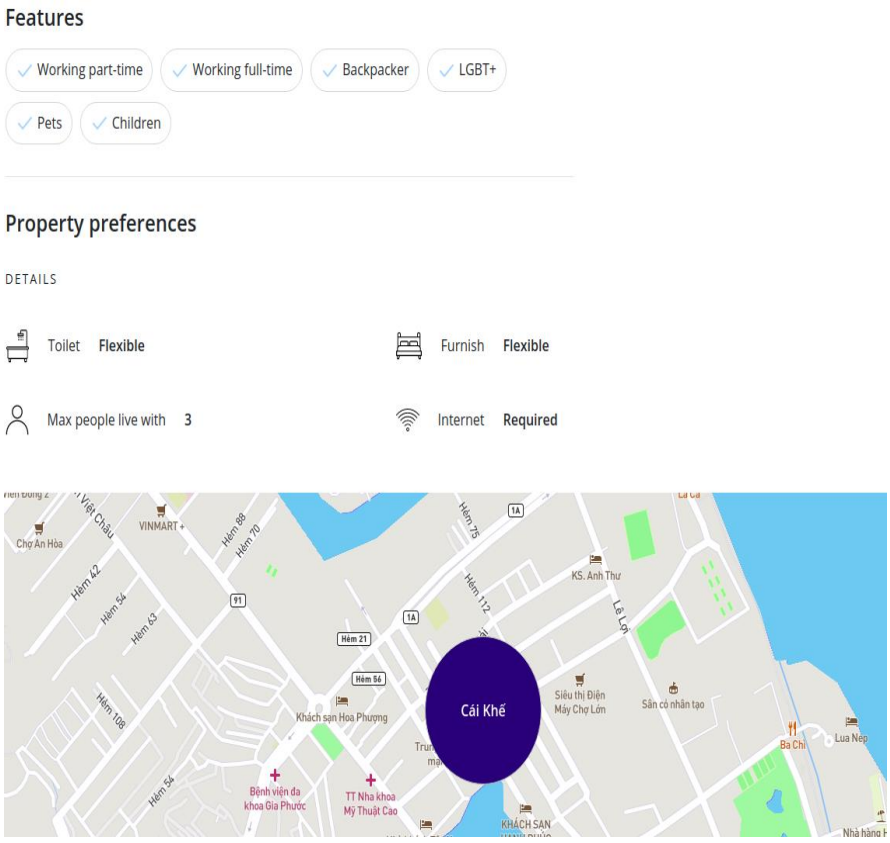


Figure 45. Seeking accommodation post page (2)

i. Chat page

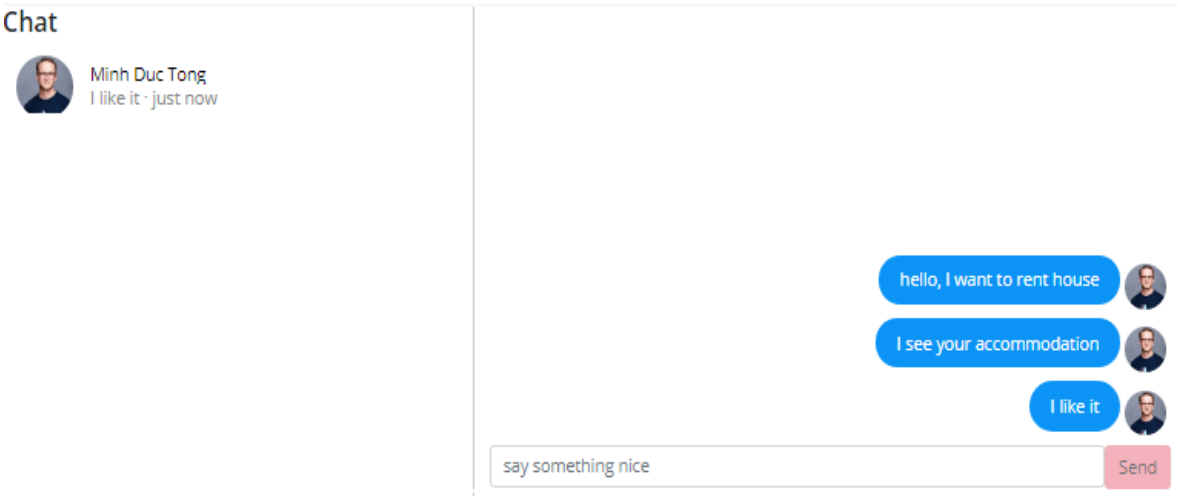


Figure 46. Chat page

On chat page, users can see their history of chat and continue to communicate with other users.

j. Map page

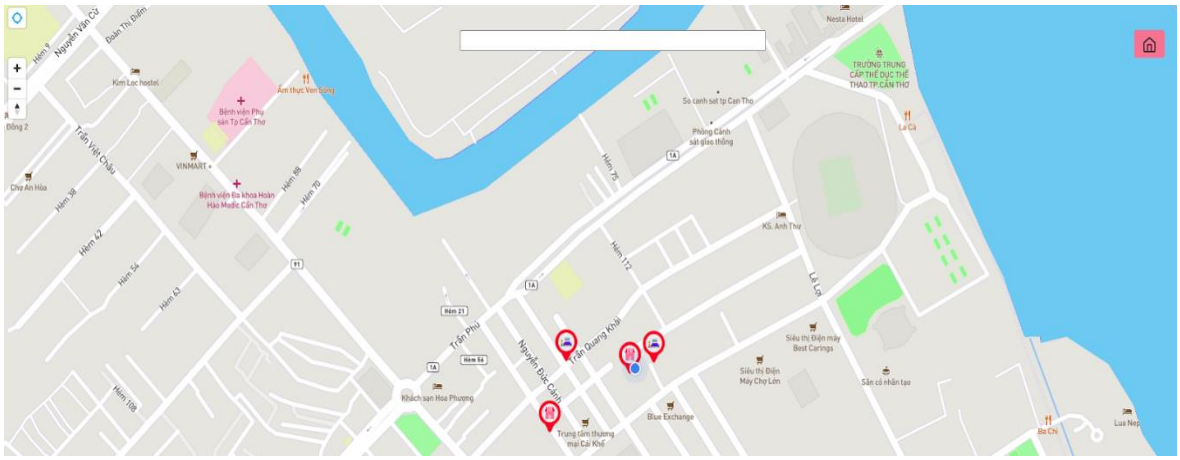


Figure 47. Map page

On the map page, users will see all of the posts on map. They can easily select locations and choose for their ideal accommodation. Users can search by location in the search bar and select to move the map to that location. Home button on the right will redirect users to the home page.

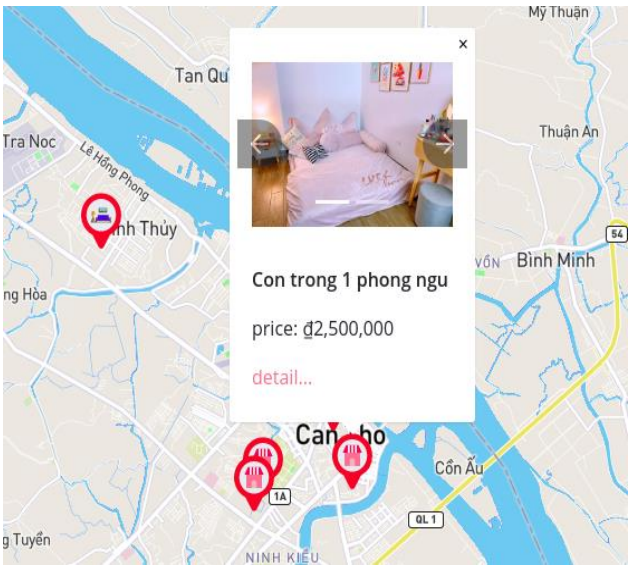


Figure 48. Map page (2)

Users can select the accommodation by clicking on the icon on the map. After that, a brief information about the post will be displayed (title and price), if they want to watch the detail of the post, they will click the button “detail”, the system will redirect users to the post page.

k. Search

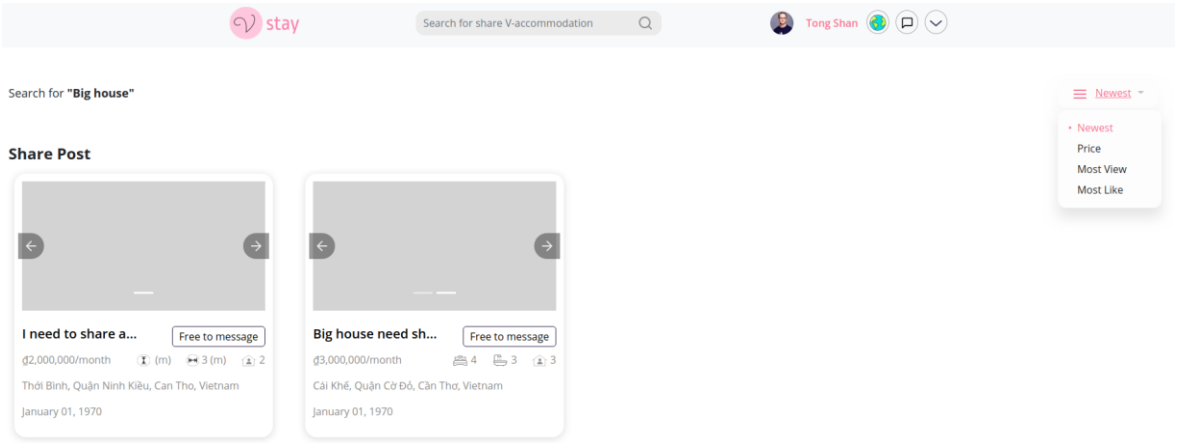
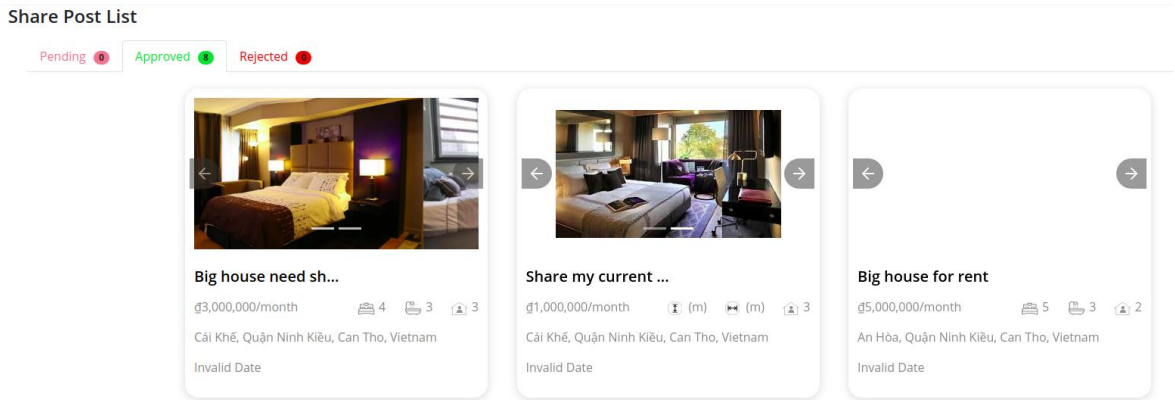


Figure 49. Search page

When users search by keywords, VSTAY will direct users to search page with the result. Moreover, they can filter the result by "Newest", "Price", "Most View" and "Most Like".

l. Moderator page



Moderator page will show all of posts in the system for moderators to preview and make it public. There will be 3 tabs to show posts by status: Pending – system will show all pending posts, Approved – system will show all approved posts and Rejected – system will show all rejected posts.

m. Profile page

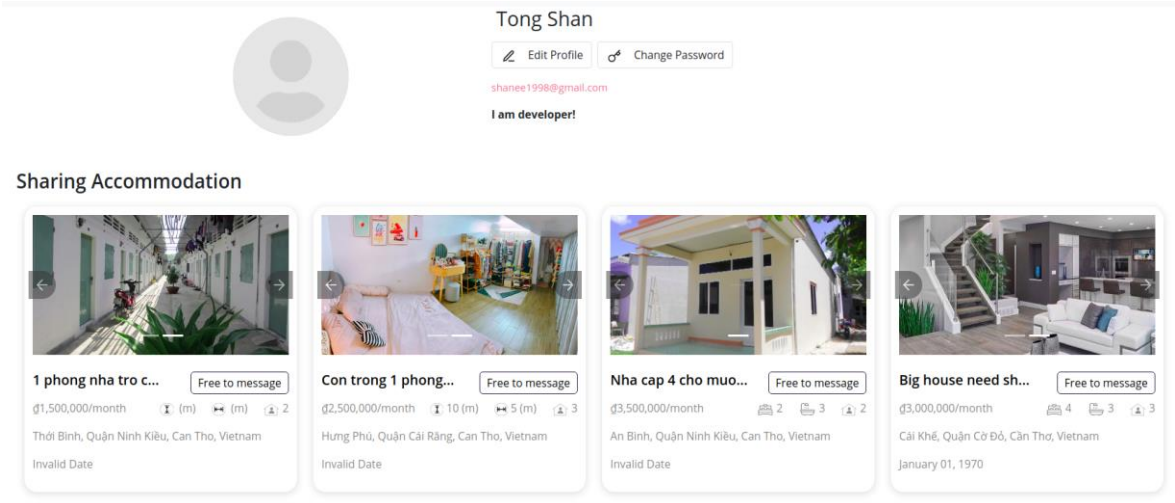


Figure 50. Profile page

Update profile

Profile page will display information of user and "Edit Profile" and "Change password" button. Moreover, the system will show all of the shared and needed posts of this user.

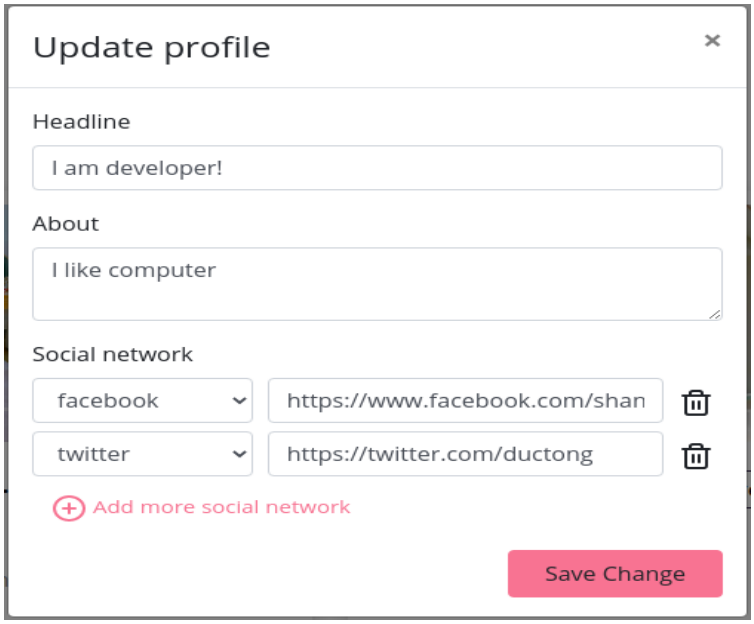


Figure 51. Update profile dialog

Users can update their profile to make it more attractive and the contacts that the others can communicate with them.

Update password

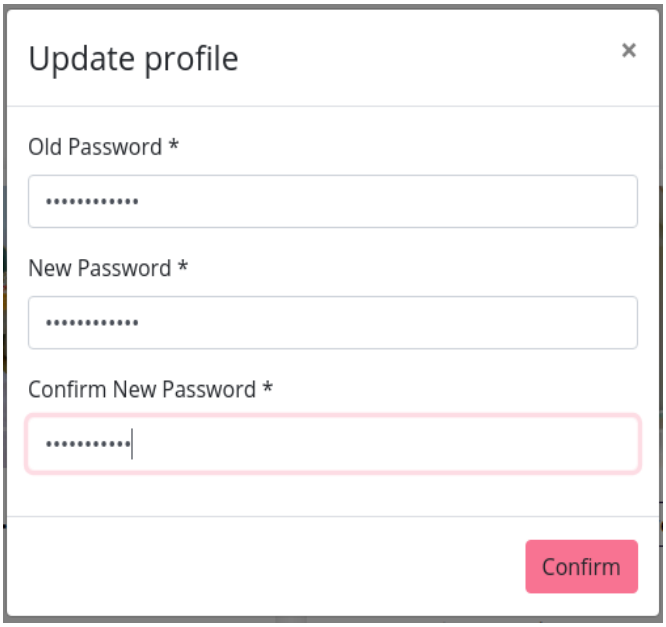


Figure 52. Update user's password

Users can update their password to make their account more secure.

Update avatar

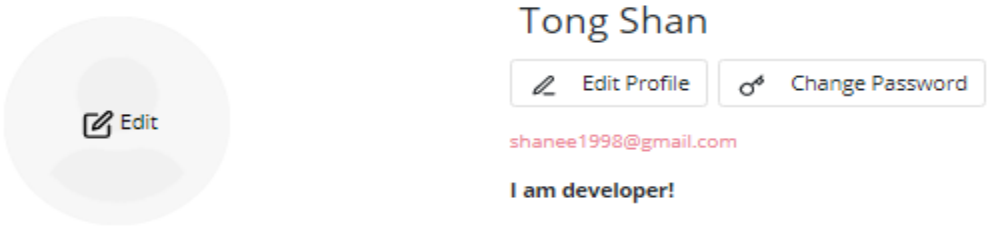


Figure 53. Update avatar (1)

First, users can update their avatar by clicking Edit button on avatar.

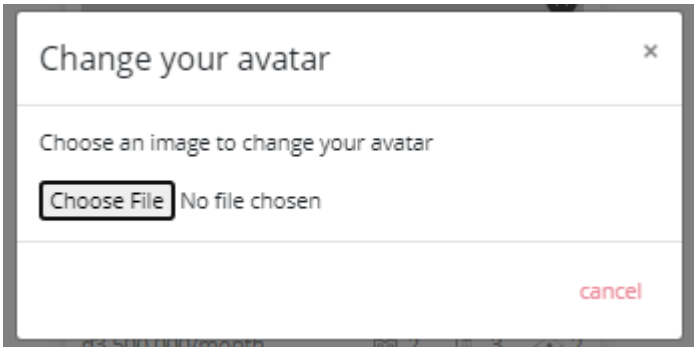


Figure 54. Update avatar (2)

Second, users will select image from their device.

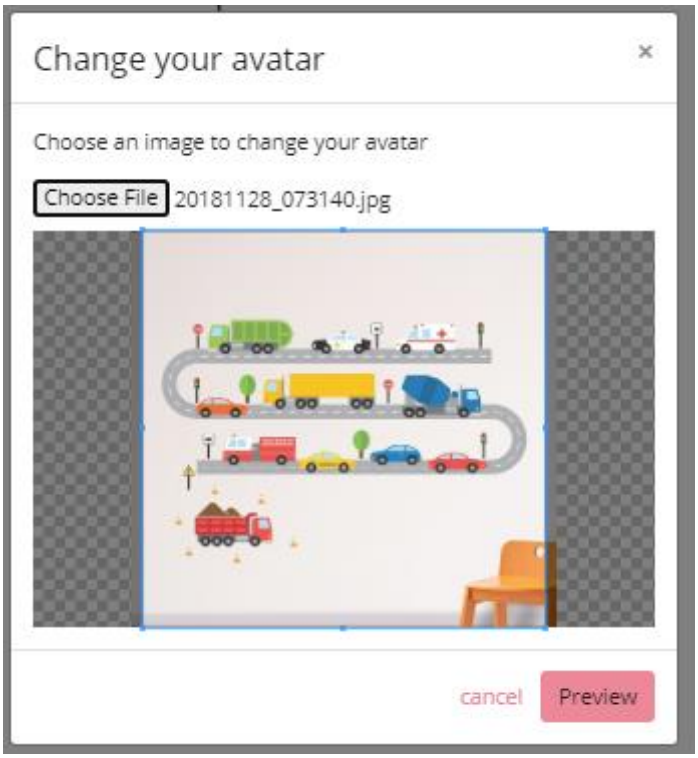


Figure 55. Update avatar (3)

Selected image will be displayed on the dialog. Users can crop and cut the selected image.

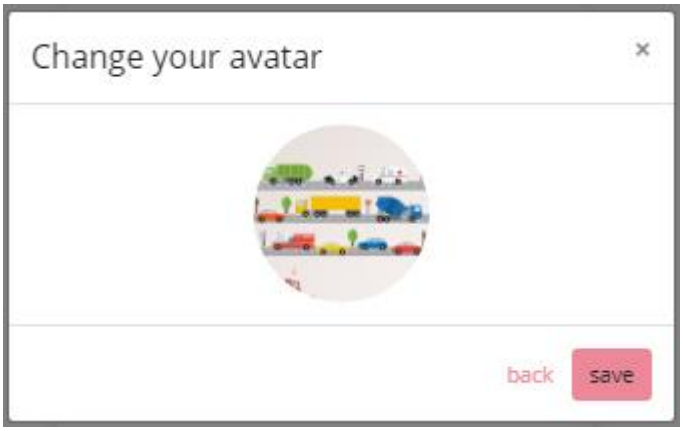


Figure 56. Update avatar (4)

Users will preview their avatar before making decision to update it.

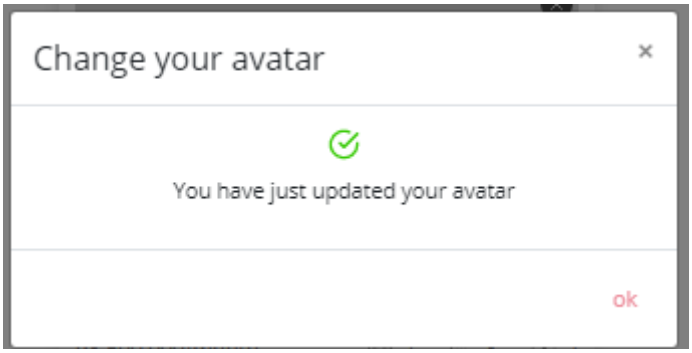


Figure 57. Update avatar (5)

The system shows notification that users updated their avatar successfully.

CHAPTER 3: TESTING AND EVALUATION

3.1. GOAL OF TESTING

The goal of testing is to find errors of the application, to make sure that the application is working properly. The process of testing includes availability testing, functional testing, compatibility testing, database testing and security testing.

Availability testing is checking the UI/UX of the application, does it have a friendly user interface or not? Are users happy with the UI/UX or not?

- The content of application is accurate without any grammatical errors.
- Error notifications are displayed properly without any grammatical errors.
- Link to the main page is displayed on every page.
- The confirm notification is displayed with all of the update actions.

Functional testing is verifying that the application has functionalities which are mentioned in the documentation or not.

- Testing the email when signing up (users provide duplicated email, wrong format).
- Testing static paths on the application is valid or not.
- Testing all of the links that are working properly or not.
- Testing all of the forms (sign in, sign up, upload posts, update profile).
- Testing error notifications that displayed when users enter invalid data

Compatibility testing is to evaluate the application is working properly on a web browser.

- Testing the application on web browsers (Firefox, Google Chrome and Opera).
- Testing the images and fonts that are displayed truly on different browsers.

Database testing is checking if data that is displayed on the application matches the data is stored in the database or not? The data that is manipulated on the application is inserted into the database or not.

- The data that is displayed for users match with data in the database.
- Testing the data is inserted into the database successfully.

Security testing is checking if there are any security holes in the application or not.

- The password is encoded.

3.2. SCENARIO OF TESTING

Scenario of availability testing

No	Description	Date
1	Testing the content of application	02/01/2020
2	Link to the main page is displayed on every page	02/01/2020
3	The notifications is displayed with update actions	02/01/2020

Table 3. Scenario of availability testing

Scenario of functional testing

No	Description	Date
1	Sign in function	02/01/2020
2	Sign up function	02/01/2020
3	Search function	02/01/2020
4	Map function	02/01/2020
5	Upload posts function	02/01/2020
6	Update profile function	02/01/2020
7	Chat function	02/01/2020
8	Update password function	02/01/2020

9	Like/dislike posts function	02/01/2020
10	Log out function	02/01/2020

Table 4. Scenario of functional testing

Scenario of compatibility testing

No	Description	Date
1	The application is displayed on web browsers	02/01/2020
2	Content and images on web browsers	02/01/2020

Table 5. Scenario of compatibility testing

Scenario of database testing

No	Description	Date
1	Data display	02/01/2020
2	Data is inserted into the database	02/01/2020

Table 6. Scenario of database testing

Scenario of security testing

No	Description	Date
1	Sign in function	02/01/2020

Table 7. Scenario of security testing

Testing Environment

- Hardware:
  - Asus FX503
  - Intel Core i7
- Software:
  - Database management system: MongoDB Compass
  - Web browsers: Google Chrome, Firefox, Opera

3.3. RESULT OF TESTING

3.3.1. Authentication

No	Description	Step	Expected result	Result	Date
1	Sign up	- Step 1: Accessing http://vstay.codes/sign-up - Step 2: Sign up with email, nickname and password tmduc0908@gmail.com/ Tong Duc/ TongDuc1909@ - Step 3: Open email inbox and verify	- Step 1: Accessing link success - Step 2: System send verification email success - Step 3: Verify email and create account success	Success	02/01/2020
2	Sign in	- Step 1: Accessing http://vstay.codes/sign-in - Step 2: Sign in with email and password tmduc0908@gmail.com/ TongDuc1909@	- Step 1: Accessing link success - Step 2: Sign in success	Success	02/01/2020
3	Log out	- Step 1: Click logout button on navigation bar - Step 2: Page redirect and display navigation bar without authentication	- Users logouted success	Success	02/01/2020

Table 8. Test case authentication function

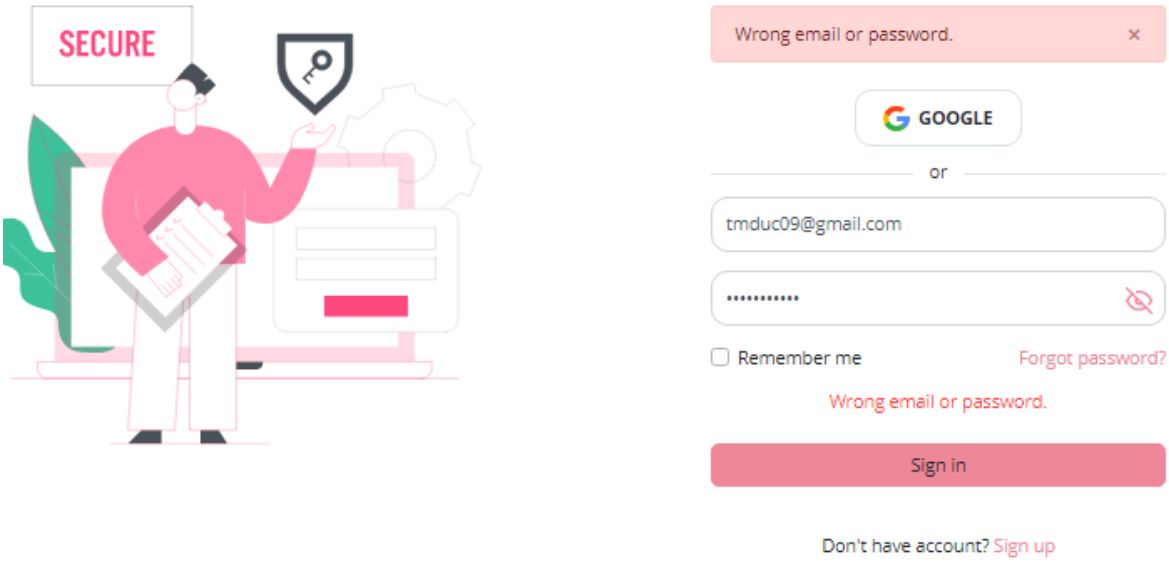


Figure 58. Error notification in sign in form

3.3.2. Post function

No	Description	Step	Expected result	Result	Date
1	Upload seeking accommodation post (signed in)	- Step 1: Accessing http://www.vstay.codes/upload-post?t=share - Step 2: Enter information of post - Step 3: System validates and create posts	- Step 1: Accessing link success - Step 2: Validate data success - Step 3: Create posts success	Success	02/01/2020
2	Upload searching accommodation post (signed in)	- Step 1: Accessing http://www.vstay.codes/upload-post?t=need - Step 2: Enter information of post - Step 3: System validates and create posts	- Step 1: Accessing link success - Step 2: Enter and validate data success - Step 3: Create posts success	Success	02/01/2020

3	Searching posts	- Step 1: Accessing http://www.vstay.codes - Step 2: Enter keyword - Step 3: Press “Enter” or click search icon	- Step 1: Accessing link success - Step 2: Enter keywords and show errors if keywords are empty success - Step 3: System shows search result success	Success	02/01/2020
4	Like/ dislike posts (signed in)	- Step 1: Accessing post page - Step 2: Click like/ dislike button	- Step 1: Accessing link success - Step 2: System shows like and dislike value success	Success	02/01/2020
5	Watch post (signed in)	- Step 1: Accessing main page->Select a post - Step 2: System directs to post page - Step 3: System shows detail of post page	- Step 1: System shows list of posts success - Step 2: Direct to true post page - Step 3: Show post page success	Success	02/01/2020

Table 9. Test case of post function

3.3.3. Map function

No	Description	Step	Expected result	Result	Date
1	Show posts on map (signed in)	- Step 1: Accessing main page-> Select map function - Step 2: Click on a post - Step 3: Click to watch detail of post	- Step 1: Accessing link success - Step 2: System shows posts success - Step 3: System directs users to watch detail of post	Success	02/01/2020

Table 10. Test case of map function

3.3.4. Update information

No	Description	Step	Expected result	Result	Date
1	Update profile (signed in)	- Step 1: Accessing main page-> Select profile function - Step 2: Click on update profile - Step 3: Enter form update profile->click “update profile”	- Step 1: Show profile success - Step 2: Show form update profile - Step 3: System updates profile success	Success	02/01/2020
2	Update password (signed in)	- Step 1: Accessing main page-> Select profile function - Step 2: Click on update password - Step 3: Enter form update password>click “update password”	- Step 1: Accessing link success - Step 2: Show form update password - Step 3: Update password success, system notifies notificatiom	Success	

Table 11. Test case of testing sign up function

3.3.5. Chat function

No	Description	Step	Expected result	Result	Date
1	Chat function	- Step 1: Accessing post page-> Click on “Send message” - Step 2: Enter content to chat - Step 3: Press enter and click send button	- Step 1: Show dialog chat - Step 2: Show form chat and history - Step 3: New content of chat updated	Success	02/01/2020

Table 12. Test case of chat function

PART 3: CONCLUSION AND FUTURE WORK

I. CONCLUSION

VSTAY has reached the goal. We successfully created an application that provided a friendly UI/UX with workable functionalities: supporting users to upload posts to search for accommodation or find their ideal customers, a post shows the detail of accommodation that helps users to have an overview of that accommodation, we also provide online chat to help users communicate with each other to discuss about the accommodation, map function is very helpful for users to search for their accommodation by location, they can easily move to the location that they need and watch the details. Moreover, the system also allows users to update their profile to make their info more attractive and beautiful, changing password will make users's accounts more secure. "Moderator" will make sure the posts of the system will be clear, they will preview all of pending posts and preview information of posts, they will make decision to change the status of posts to make it public or not.

Although VSTAY provides a lot of helpful functionalities, there are some limits. At this present, VSTAY does not contain admin functions to manage posts and users directly. The total steps of the uploading process are too many, maybe it makes users to be confused.

II. FUTURE WORK

In the future, we plan to improve the upload posts function to help users describing their posts more details to make a post more attractive. We will add more fields for users to add to make users describe their accommodations detaily. We will design more functions for post interactions, for example, comment and reply on posts will make users comment and evaluate accommodations after renting. We will improve search function to allow users to search with multiple options, so they can find more accommodations with their needs. Moreover, we will integrate machine learning into recommendation systems based on user's behaviors and posts to show more effective posts and implement more functions for moderator.

REFERENCES

[1] "Roomgo," [Online]. Available: https://au.roomgo.net/.

[2] "flatmate," [Online]. Available: https://flatmate.com/.

[3] "Realestate Australia," [Online]. Available: https://www.realestate.com.au/buy.

[4] "Ohanaliving," [Online]. Available: https://www.ohanaliving.vn/#/.

[5] "timphongtrovn," [Online]. Available: http://www.timphongtro.vn/.

[6] "batdongsan," [Online]. Available: https://batdongsan.com.vn/.

[7] React. [Online]. Available: https://reactjs.org/tutorial/tutorial.html.

[8] NodeJS. [Online]. Available: https://nodejs.dev/learn/introduction-to-nodejs.

[9] Mapbox. [Online]. Available: https://docs.mapbox.com/api/.

[10] G. Firebase. [Online]. Available: https://firebase.google.com/.

APPENDIX 1

1. Collection user

No	Field name	Data type	Description
1	_id	text	ID of user
2	email	text	Email of user
3	name	text	Nickname of user
4	password	text	Password of user (encoded)
5	role	text	Role of user
6	status	text	Status of user
7	about	text	Description of user
8	avatar	text	Link of avatar
9	social	object	Social network of user
10	headline	text	Headline of user
11	lastActivity	timestamp	Last time that user logged in
12	createdAt	timestamp	Date of created
13	updatedAt	timestamp	Date of updated

Collection “user” will store all of users in the database. Each document contains details of users.

2. Collection sharePost

No	Field name	Data type	Description
1	_id	text	ID of post
2	title	text	Title of post

3	type	text	Type of post
4	address	object	Object address of accommodation
5	address.name	text	Address of accommodation
6	address.geocode.longitude	text	Longitude of address
7	address.geocode.latitude	text	Latitude of address
8	detail	object	Detail of accommodation
9	detail.parking	text	Parking feature
10	detail.internet	text	Internet feature
11	detail.total_bedrooms	number	Total bedrooms
12	detail.total_bathrooms	number	Total bathrooms
13	detail.max_people_live_with	number	Max people living
14	detail.toilets	number	Number of toilets
15	detail.furnishing	number	Status of furnishing
16	detail.bills	text	Status of bills
17	detail.depositLength	number	Deposit length
18	detail.room_availability	object	Room availability
19	detail.room_availability.min_length_of_stay	number	Min length of stay
20	detail.room_availability.max_length_of_stay	number	Max length of stay
21	detail.date_availability	timestamp	Date availability
22	detail.except	text	Accommodation excepting

23	features	array	Features of accommodation
24	description	text	Description of post
25	images	array	Accommodation' s images
26	price	number	Price of renting
27	poster	objectId	Id of author
28	status	text	Status of post
29	statistics	object	Statistic of post
30	statistics.viewCount	number	Number of view
31	statistics.likeCount	number	Number of like
32	statistics.dislikeCount	number	Number of dislike

Collection "sharePost" will store all of the posts that users uploaded for renting purposes.

3. Collection needPost

No	Field name	Data type	Description
1	_id	text	ID of user
2	about	object	Information of author
3	type	text	Type of post
4	location	object	Object location of accommodation
5	location.name	text	Address of accommodation
6	location.geocode.longitude	text	Longitude of address
7	location.geocode.latitude	text	Latitude of address
8	detail	object	Detail of accommodation
9	detail.toilets	text	Parking feature

10	detail.internet	text	Internet feature
13	detail.max_people_live_with	number	Max people living
14	detail.toilets	number	Number of toilets
15	detail.furnishing	number	Status of furnishing
16	employment_status	array	Employment status of author
17	length_of_stay	number	Length of stay
18	life_style	array	Life style of users
19	move_date	timestamp	Move date of users
24	description	text	Description of post
26	budget	number	Budget of author
27	poster	objectId	Id of author
28	status	text	Status of post

Collection "needPost" will store all of the posts that users uploaded for seeking purposes.

4. Collection reactions

No	Field name	Data type	Description
1	_id	text	ID of reaction
2	postID	objectId	ID of post
3	like	Array	Array of user ID
4	dislike	Array	Array of user ID

Collection "reactions" will store the reactions of posts, it contains the number of likes and dislikes of posts.

5. Collection views

No	Field name	Data type	Description
1	_id	text	ID of document
2	postID	objectId	ID of post
3	postType	text	Type of post
4	views	array	Array of user ID

Collections “views” will store all of the views of posts.

6. Collection of refreshToken

Collection “refreshToken” will store refresh tokens of users. Refresh tokens will be used for authentication.

No	Field name	Data type	Description
1	_id	text	ID of document
2	userId	objectId	ID of user
3	exp	number	Unix timestamp of expire
4	iat	number	Unix timestamp of created
5	value	text	Value of refresh token