Group 01

KOOLB
Use-Case Specification
Version 1.0

Revision History

Date	Version	Description	Author
10/12/22	1.0	The first version	Nguyễn Hà Ngọc Linh
10/12/22	1.1	The final version	Trương Thúy Tường Vy
			Bùi Lê Gia Cát

Table of Contents

Use-Case Specification: KOOLB

-

- **Use case: Register**
Brief Description: This use case describes how a new user registers into the KOOLB.
Actors: Host and Renter
Preconditions: The username is not found in the login The user tries to log in but the username is not found in the system. The user uses the app for the first time The user downloads and uses the app for the first time
Flow of events: 1. The system validates the username to check if it has been taken and the password to check if it is strong enough. If he/she uses other alternative ways, the system checks if those accounts have already been registered. Then, the system creates a new user based on the information. 2. The system displays the registration forms (including retype password) and icons of alternative ways below the form.
Post Condition: **The system allows the user to log in immediately after registration** After satisfying all the requirements, the system allows the user to login immediately.
Alternative flow **Invalid username** Issue: The field to type username is empty or the username may have been already taken. He/She needs to change this. **Invalid password** Issue: The password may not meet the condition and need to be changed **The password is not strong enough** Issue: The user enters a password that does not meet the requirements of a strong password. The user is required to change the password. **The string in the retype-password field is different from the one in the password** Issue: The user typing the string in the retype-password field does not match the one in the password. The user is required to enter the same password or change it.
Extension Point **If the user have their account registered (email found)** If the user's account has been found on the database, announce the user and ask to be led to the login

page.

-

- **Use case: Log in**
Brief description: The KoolB App login process is described in this use case.
Actors: Host and Renter
Preconditions: **The user has valid email** The username must exist as registered email in the app's database; meaning that actors must registered account before they log in -
Flow of events: 1. Users open app KoolB 2. The Login Page is displayed 3. Actors fill in username and password 4. The system validates username and password of actors, then logs him/her into the app. 5. The system displays Home Page corresponding to each role and end the use case
Postconditions: **Navigating immediately to Home Page** After logging in, users are navigated to Home Page according to his/her role Initial setup The system set up user's booking history, chat log, saved accommodation, etc and all other personal information
Alternative flow 1: **Invalid Passwords** At step 4, if password of actor is invalid, then trigger the first alternative flow Display the message about invalid password Require actors to correct password Repeat this step until both username and password are valid End use case **Invalid Username** At step 4, if username of actor is invalid, then trigger the second alternative flow Display the message about invalid username Require actors to correct email or register username as new account Repeat this step until both username and password are valid End use case **Forget Password** At step 4, if actors click the 'Forget Password' button, then trigger this flow Display Change Password Screen Ask actors to fill in new passwords

Change passwords for user

Navigate back to Login Page for users

Extension Point
Register
If the account does not exist, the system asks the user to create an account.
If the user accepts to create an account, the application moves the user to the registration page.

-

- **Use case: Explore**
Brief Description: This use-case is the first destination of users after successfully logging in or signing up. It represents under Homepage in UI design
Actors: Hosts
Preconditions: Users have to first log in or register to be granted permission to enter the homepage section.
Flow of event: The page can be scrolling up and down to view all the Feature Accommodation, Category, Search bar…
Postconditions: Be able to present the feature and category that the user demand.
Alternative flow: -
Extension point: -

-

- **Use case: View Accommodation Detail**
Brief Description: This use-case describes how the system acts when the user views the details of accommodation.
Actors: Hosts
Preconditions: **The renter must login or sign up first** The renter is required to login first or create an account if he/she is a new user. **Accommodation must be posted by hosts** Accommodation must be posted publicly by hosts first
Flow of event: **Add accommodation to favorite list** At step 1 of Basic flow, Renter clicks on the 'heart' icon. The heart icon transformed to a 'heart' filled with color to indicate the accommodation is saved. The accommodation is added to saved list **Update accommodation** At step 1 of Basic flow, Host clicks on update accommodations options The application allows Host to change any section of detail

The application waits for Host done updating and accepts the changes

The system creates a temporary changes on database

After Admin accepts the changes, the system updates the information of the accommodation

Change the status of accommodation

At step 1 of Basic flow, Host click on changing status options (closing, temporarily closing, open)

The system database updates the status accommodation.

Postconditions:

It should be able to navigate Renters to Booking or back to Explore use-case.

Alternative flow:

-

Extension point:

Add accommodation to favorite list

The system will update the favorite list

Add accommodation to booking list

When Renter clicks on booking options, this use-case will move to Booking use-case.

Chat with Host

The Renter can chat with the Host to confirm all information about accommodation.

-

- Use case: Navigate

Brief Description: This use-case is the navigator to every other use-cases. It is under the Navigation bar in UI design.

Actors: Hosts

Preconditions:

Users have to first log in or register to be granted permission to enter the homepage section with a navigation bar.

Flow of event:

The navigation bar in cluded of many button that navigate to the specific use-cases.

Postconditions:

Be able to travel user to the expected use-cases.

Alternative flow:

-

Extension point:

-

-

- Use case: View Reservation List

Brief Description: This use-case describes how the user views their reservations.

Actors: Hosts

Preconditions:

The user must log in or sign up first

The user is required to log in or sign up first to see their reservation.

Flow of event:

1. The application shows a list of reservations.

2. Each notification contains the title and brief description.

3. The system shows the most recent reservations

4. The user scrolls to view more reservations.

5. The user clicks a notification to view more detail.

6. The use-case ends when the user ends the application or moves to other pages.

Postconditions:
-

Alternative flow:
The user delete a reservation
1. The user delete a reservation
2. The system remove that from database
3. The application remove from the list
4. The application updates the list.

Extension point:
-

-

- **Use case: Analyze Host Insight**
Brief Description: This use-case describes the tools for hosts to manage every information and data about their service.
Actors: Host
Preconditions: **Host should finish posting the accommodation on the public site** In order to view the statistics and checklist of the accommodation, the host should have that accommodation publicized so that the app will generate real-time data based on the number of guests visiting the accommodation.
Flow of event: **Cleaning and supply checklist** 1. Click on the Cleaning and Supply checklist bar 2. Manually checking the "cleaning and supply" task in real situations of the accommodation and either adding or ticking on the completed (incompleted) tasks. 3. Clicking submit to complete the process of checking. Any completed tasks will be hidden and the uncompleted tasks will be visible to the host insight. Statistic of the Accommodation -
Postconditions: **The checklist should be temporarily automatically saved** In case the user fails to save the checklist due to networking lost, the process will be automatically saved and appear once the user reconnects the server. **The statistic automatically updated** Every time the application receives information of new guests in the accommodation, the statistics will change and the history of their information will be added to the database
Alternative flow: **The users don't define the name for the new tasks** 1. In step 2 of basic flow, if the host fails to define the name of the new task added, it will trigger the first alternative flow. 2. Display red of the new checkbox that has not had a specific name, stop users from submitting the checklist. 3. Require the users to either input a name for the check box or delete that checkbox. 4. Repeat this step until the checkbox has been filled or deleted 5. End use case
Extension point: -

-

- **Use case: Post Room Information**
Brief Description: The KoolB App posting accommodation process is described in this use case.
Actors: Host
Preconditions: **Actor must be host** Actor have to be logged in and be a host so that they can create new accommodation under their names **Admin Approval** The accommodation must be waited for approval from admin before being published and displayed on app
Flow of event: 1. The system displays a list of information about the accommodation to fill in 2. Actors fill in information 3. Actors complete the form and post new accommodation 4. Displays a complete form for double checking 5. Actors confirm all information is correct and submit final form
Postconditions: -
Alternative flow: **Lack of required information** 1. If fundamental information such as name, location, price, image is left black, then this alternative flow is triggered 2. Mark where the information is lacked of 3. Ask users to fill in those empty information 4. Wait until all informations are all filled in completely 5. End flow
Extension point: **User disconnected from app when posting an accommodation** At step 5 in Basic Flow, if actor is disconnected from app because of many reasons, then all the progress is stopped and information is saved until the next login of actor

-

- **Use case: Notify**
Brief Description: This use case describes the notifications acting in Renter and Host.
Actors: Host and Renter
Preconditions: The user must log in or sign up first The user is required to log in or sign up first to view the notifications
Flow of event: 1. The application shows a list of notifications. 2. Each notification contains the title and brief description. 3. The system shows the most recent notifications 4. The user scrolls to view more notifications. 5. The user clicks a notification to view more detail. 6. The use-case ends when the user ends the application or moves to other pages.
Postconditions: -

Alternative flow:
The notification change to have read
In step 4, when the user clicks a notification, the background of notifications will turn darker (gray) when the user has finished reading.

Extension point:
-

-

- **Use case: Access Setting**
Brief Description: This use-case describes the setting process of the app KOOLB.
Actors: Hosts
Preconditions:
Sign into the application
Actors must log in to the application before they can chat with each other.
Press the right icon
The actor must press the right icon in order to be navigated to the chat page.
Flow of event:
Setting of Personal information and security
1. Users click on personal information and security setting section
2. The current username and password are displayed (password is displayed in encrypted format)
3. Users click on either username or password to change the information
4. Users submit their personal information changed, the system will validate and update the information
Setting of General information
1. Users click on the general information setting section
The current Country, Currency, and Language
2. Users click on either username or password to change the information
3. This information will be changed according to existing data. Every country(currency, language) available in this app will appear in scrolling lists.
4. Users click on the corresponding option. At the time a new option is chosen, the app will be updated.
Setting of notification
1. Users click on the personal notification setting section
2. The current notification state will be displayed (Banners, sounds, badges, announcements…)
3. Users click on the corresponding option to change the way the app will notice new information changed.
4. At the time the new option is chosen, the app will be updated.
About Us section
1. Users click on About Us section
2. The app displays all information about this app's basic information about the store and its founders, explains the company's purpose and how it differs from the competition, and encourages discussion and interaction.
Logout section
1. Users click on the Logout long button at the end of the setting page.
2. The app logs out the current account. Any process in progress will either temporarily be saved or dismissed.
Postconditions:
-
Alternative flow:
Invalid Password
1. At step 3 of "Setting of Personal information and security" basic flow, if password of actor is invalid, then trigger the first alternative flow
2. Display the message about invalid password
3. Require actors to correct password

4. Repeat this step until both username and password are valid
5. End use case
Invalid Username
1. At step 3 of "Setting of Personal information and security" basic flow, if username of actor is invalid, then trigger the second alternative flow
2. Display the message about invalid username
3. Require actors to correct email or register username as new account
4. Repeat this step until both username and password are valid
5. End use case

Extension point:
Update profile
The application changes the UI so that the user can update profile pages and the system allows users to change their information.
Language setting
The application allows users to change their language and the system will update into the database.
Configure payment
The application allows users to change the payment method and default one.

-

- **Use case: Access Chatbox**
Brief Description: This use case allows a renter and a host to discuss more information about the room.
Actors: Hosts
Preconditions: **Sign into the application** Actors must log in to the application before they can chat with each other. **Press the right icon** The actor must press the right icon in order to be navigated to the chat page.
Flow of event: 1. Actors choose the chat icon in the navigation bar to go to the chat page. 2. The actor chooses the icon of the other actor that they want to chat with. 3. Actors answer questions if they receive any. Actors can send questions or messages to the other actor. 4. The use case ends when actors go to another application page.
Postconditions: **The chat is saved in the chat list** The conversations between actors are saved in the chat list of each actor. **Customers have enough information about the accommodation** Any questions about accommodation have been answered by hosts so customers can now decide whether or not they will book this accommodation.
Alternative flow: **Unable to send messages** 1. In step 3, if one actor cannot send messages, then trigger the first alternative flow. 2. Display the message about messages that cannot be sent. Additionally, an option to resend messages is shown. 3. The actor press "Resend" 4. The system resends their messages. If the actor still cannot send messages, this flow is repeated.
Extension point: **The Chat box is accessed from the detail page** Actors access the chat box by pressing the button "Contact host" on the detail page for each room. Then, a new conversation is displayed and the two actors can chat with each other now.

-

- **Use case: Update Profile**
Brief Description: This use-case describes the update profile process of the app KOOLB.
Actors: Host and Renter
Preconditions: **User must be in the setting use case** Account available The user need to have the account available in the app to perform setting use-case Stable Internet Connection In order to save setting successfully, users need to have a stable internet connection
Flow of event: The user clicks on the profile setting option in setting page
Postconditions: -
Alternative flow: -
Extension point: Return back to the setting page.

-

- **Use case: Configure Language Setting**
Brief Description: This use-case describes the language setting process of the app KOOLB.
Actors: Host and Renter
Preconditions: **User must be in the setting use case** **Account available** The user need to have the account available in the app to perform setting use-case **Stable Internet Connection** In order to save setting successfully, users need to have a stable internet connection
Flow of event: Change language of the application
Postconditions: -
Alternative flow: -
Extension point: Return back to the setting page.

-

- **Use case: Configure Payment processing**
Brief Description: This use-case describes the configured payment process of the app KOOLB
Actors: Host and Renter
Preconditions: **User must be in the setting use case** **Account available** The user need to have the account available in the app to perform setting use-case

Stable Internet Connection
In order to save setting successfully, users need to have a stable internet connection
Flow of event:
The user configure the payment of the account
Postconditions:
-
Alternative flow:
-
Extension point:
Return back to the setting page.

- **Use case: Display Wishlist**
Brief Description: This use case shows a list of saved accommodations
Actors: Renter
Preconditions:
Actor must log in or sign up first
The renter is required to login first to save places.
Actor must press the correct icon to navigate to the list view
The screen will move to the Favorite list view.
Flow of event:
1. Renter click at the icon.
2. Renter is navigated to the screen displaying a list of accommodations
3. Renter click to another screen, the flow ends here.
Postconditions:
List of saved item is updated
Actors can now see their newly saved accommodation in the list of saved places.
Alternative flow:
Actor unsave the locations
1. When an accommodation is saved and actors press the 'heart' icon again, this flow is triggered.
2. Heart icon transform back to the unsaved icon
3. The accommodation is removed from saved list
4. The flow end here
Host deletes the saved accommodation
1. When an item in saved list is deleted by host, this flow is triggered
2. The list is updated; i.e. the saved place that is deleted will also deleted from saved list
Renter has no saved accommodation
The screen shows a text "Wow, so empty"
Extension point:
Renter add new favorite accommodation
When user press the 'heart' icon of an accommodation, the list of favorite will be updated by adding that accommodation

- **Use case: Log in**
Brief description: The KoolB App login process is described in this use case.
Actors: Renter
Preconditions:

The user has valid email

The username must exist as registered email in the app's database; meaning that actors must registered account before they log in

Flow of events:

1. Users open app KoolB
2. The Login Page is displayed
3. Actors fill in username and password
4. The system validates username and password of actors, then logs him/her into the app.
5. The system displays Home Page corresponding to each role and end the use case

Postconditions:

Navigating immediately to Home Page

After logging in, users are navigated to Home Page according to his/her role

Initial setup

The system set up user's booking history, chat log, saved accommodation, etc and all other personal information

Alternative flow 1:

Invalid Passwords

At step 4, if password of actor is invalid, then trigger the first alternative flow

Display the message about invalid password

Require actors to correct password

Repeat this step until both username and password are valid

End use case

Invalid Username

At step 4, if username of actor is invalid, then trigger the second alternative flow

Display the message about invalid username

Require actors to correct email or register username as new account

Repeat this step until both username and password are valid

End use case

Forget Password

At step 4, if actors click the 'Forget Password' button, then trigger this flow

Display Change Password Screen

Ask actors to fill in new passwords

Change passwords for user

Navigate back to Login Page for users

Extension Point

Register

If the account does not exist, the system asks the user to create an account.

If the user accepts to create an account, the application moves the user to the registration page.

- **Use case: Register**
Brief Description: This use case describes how a new user registers into the KOOLB.
Actors: Renter
Preconditions:
The username is not found in the login
The user tries to log in but the username is not found in the system.
The user uses the app for the first time
The user downloads and uses the app for the first time
Flow of events:
1. The system validates the username to check if it has been taken and the password to check if it is strong

enough. If he/she uses other alternative ways, the system checks if those accounts have already been registered. Then, the system creates a new user based on the information.

2. The system displays the registration forms (including retype password) and icons of alternative ways below the form.

-

Post Condition:
The system allows the user to log in immediately after registration
After satisfying all the requirements, the system allows the user to login immediately.

Alternative flow
Invalid username
Issue: The field to type username is empty or the username may have been already taken. He/She needs to change this.

Invalid password
Issue: The password may not meet the condition and need to be changed

The password is not strong enough
Issue: The user enters a password that does not meet the requirements of a strong password. The user is required to change the password.
The string in the retype-password field is different from the one in the password
Issue: The user typing the string in the retype-password field does not match the one in the password. The user is required to enter the same password or change it.

Extension Point
If the user have their account registered (email found)
If the user's account has been found on the database, announce the user and ask to be led to the login page.

-

Use case: Explore

Brief Description: This use-case is the first destination of users after successfully logging in or signing up. It represents under Homepage in UI design

Actors: Hosts

Preconditions:
Users have to first log in or register to be granted permission to enter the homepage section.

Flow of event:
The page can be scrolling up and down to view all the Feature Accommodation, Category, Search bar…

Postconditions:
Be able to present the feature and category that the user demand.

Alternative flow:
-

Extension point:
-

-

-	**Use case: View Accommodation Detail**

Brief Description: This use-case describes how the system acts when the user views the details of accommodation.

Actors: Hosts

Preconditions:
The renter must login or sign up first
The renter is required to login first or create an account if he/she is a new user.
Accommodation must be posted by hosts
Accommodation must be posted publicly by hosts first

Flow of event:
Add accommodation to favorite list
At step 1 of Basic flow, Renter clicks on the 'heart' icon.
The heart icon transformed to a 'heart' filled with color to indicate the accommodation is saved.
The accommodation is added to saved list
Update accommodation
At step 1 of Basic flow, Host clicks on update accommodations options
The application allows Host to change any section of detail
The application waits for Host done updating and accepts the changes
The system creates a temporary changes on database
After Admin accepts the changes, the system updates the information of the accommodation
Change the status of accommodation
At step 1 of Basic flow, Host click on changing status options (closing, temporarily closing, open)
The system database updates the status accommodation.

Postconditions:
It should be able to navigate Renters to Booking or back to Explore use-case.

Alternative flow:
-

Extension point:
Add accommodation to favorite list
The system will update the favorite list
Add accommodation to booking list
When Renter clicks on booking options, this use-case will move to Booking use-case.
Chat with Host
The Renter can chat with the Host to confirm all information about accommodation.

-

-	**Use case: Navigate**

Brief Description: This use-case is the navigator to every other use-cases. It is under the Navigation bar in UI design.

Actors: Hosts

Preconditions:
Users have to first log in or register to be granted permission to enter the homepage section with a navigation bar.

Flow of event:
The navigation bar in cluded of many button that navigate to the specific use-cases.

Postconditions:
Be able to travel user to the expected use-cases.

Alternative flow:
-

Extension point:
-

-

Use case: Verify payment

Brief Description: This use-case describes the verify payment process of the app KOOLB.

Actors: Renters

Preconditions:
Booking
Users must booked this accommodation first

Flow of event:
The system checks if the payment method is valid or not.

Postconditions:
-

Alternative flow:
-

Extension point:
-

-

- Use case: Check Availability

Brief Description: This use case describes the availability checking.

Actors: Renters

Preconditions:
Booking
Users must booked this accommodation first

Flow of event:
The system check if the duration is available

Postconditions:
-

Alternative flow:
-

Extension point:
-

-

- Use case: Book Accommodation

Brief Description: This use case describes how a user books a room in the system..

Actors: Renters

Preconditions:

The renter must login or sign up first

The renter is required to login first or create an account if he/she is a new user.

The renter must verify the payment

Before booking, the system asks the Renter to choose the payment method.

The Renter must check in

For more convenience, the Renter verifies check in to let the Host know.

The Renter must check out

The Renter must click check out to let the Host know

The Renter can rating

After booking and checking out, the Renter can rate and leave the comment of the accommodation.

Flow of event:

1. The renter choose the duration
2. The renter enters number of guests
3. The renter choose payment method
4. The renter accepts the options
5. The renter wait for the host to accept

Postconditions:

The system wait for the host to accept

The system will notify the host and wait for he/she accepts

Alternative flow:

Invalid duration

The renting duration is not valid and the renter is asked to choose again.

Invalid duration due to the logic

The renter may choose the end day before the start day. The renter is only required to choose again.

Invalid duration due to the accommodation is not available in these days

The renter is required to choose another day that accommodation is available.

The renter does not accept the options

The system will go back to the options form and wait for the retenter to accept

The host does not accept

The renter will get a notification about the rejection of the host.

Extension point:

The renter login or create an account

The system will let the user login or create a new account.

The renter lost connection

At step 1, 2, 3, 4 in Basic Flow, if the renter is disconnected from the app because of many reasons, the system will save the option and wait for the renter to go back online.

Chatbox

Renters can chat with hosts about the accommodation they consider to book in order to assure the quality before deciding whether or not they will book this accommodation.

Cancel booking

The system updates the database and notify the Host to let them know of the cancellation. The system also allows Renter to write a message about why they canceled the booking.

-

Use case: Checkout

Brief Description: This use-case describes the checkout process of the app KOOLB.

Actors: Renters
Preconditions: The renters have the booking on progress and wanted to end their stay.
Flow of event: Renter comes to the accommodation. The check out process will be based on each individual stayover that requires anything.   The user clicks on the checkout icon in order to have a room checkout.
Postconditions: **Status of accommodation is updated**   Status such as 'vacant' or 'booked' should be updated. Renters can now book this accommodation
Alternative flow: **-**
Extension point: **Rating**   After checkout, if users want to rate accommodation and leave opinion, they can click on 'Rating' and trigger Rating use case

-

- **Use case: Check in**
Brief Description: The KoolB App check in process is described in this use case.
Actors: Renters
Preconditions: The renter needs to successfully book the accommodation first before check in.
Flow of event: Renter comes to the accommodation. The check in process will be based on each individual stayover that requires anything.
Postconditions: The renters need to check out at the right time that has been booked before. Overstaying will be penalized (Extra charges…)
Alternative flow: -
Extension point: Checkout to the accommodations

-

- **Use case: Sort and Filter**
Brief Description: This use-case describes the process of filtering and sorting in the search bar in app KOOLB.
Actors: Renters

Preconditions:
Account available
The user need to have the account available in the app to perform setting use-case
Search section
In order to filter and sort search results, users need to be in the search section.

Flow of event:
1. Renter can press the "filter and sort" icon on the search bar to filter and sort the result.
2. A dialog with choices for sorting, and filter is displayed. Choices for sorting include prices, ratings and distance from center. Choices for filtering include rating, breakfast included, number of rooms,...
3. The renter fills in choices.
4. The renters the button "Show".
5. The home page shows a list of rooms equivalent to choices of renter.

Postconditions:
Display wanted result
The result users wanted to have after applying filtering and sorting should be displayed accurately and successfully.

Alternative flow:
-

Extension point:
-

-

Use case: Chat

Brief Description: This use case allows a renter and a host to discuss more information about the room.

Actors: Hosts and Renters

Preconditions:
Sign into the application
Actors must log in to the application before they can chat with each other.
Press the right icon
The actor must press the right icon in order to be navigated to the chat page.

Flow of event:
1. Actors choose the chat icon in the navigation bar to go to the chat page.
2. The actor chooses the icon of the other actor that they want to chat with.
3. Actors answer questions if they receive any. Actors can send questions or messages to the other actor.
4. The use case ends when actors go to another application page.

Postconditions:
The chat is saved in the chat list
The conversations between actors are saved in the chat list of each actor.
Customers have enough information about the accommodation
Any questions about accommodation have been answered by hosts so customers can now decide whether or not they will book this accommodation.

Alternative flow:
Unable to send messages
1. In step 3, if one actor cannot send messages, then trigger the first alternative flow.

2. Display the message about messages that cannot be sent. Additionally, an option to resend messages is shown.
3. The actor press "Resend"
4. The system resends their messages. If the actor still cannot send messages, this flow is repeated.

Extension point:
The Chat box is accessed from the detail page
Actors access the chat box by pressing the button "Contact host" on the detail page for each room. Then, a new conversation is displayed and the two actors can chat with each other now.

-

-   Use case: Rate

Brief Description: This use-case describes the rating of a specific accommodation.

Actors: Renters

Preconditions:
The renter's information have to be in the accommodation history data
The renter has to use the service first and their information has to be added on the history of the accommodation data in order to rate the service.

Flow of event:
1. The renter rates the accommodation by ticking the stars from 1 to 5 corresponding to their experiences during staying at the accommodation.
2. The renter writes comments about the accommodation (Facility, Service, Attitude,...)
3. The renter click "Submit"

Postconditions:
Accommodation's rating is updated
Rating of accommodation will be recalculated and updated. New reviews will also be displayed.

Alternative flow:
Answer box for question is leaving blank
1. Announce users that the question is still not answered.
2. Make sure the users answer the question (box filled) before sending it back to customers.

Extension point:
User disconnected from app when posting an accommodation
At step 5 in Basic Flow, if the user is disconnected from the app because of many reasons, then all the progress is stopped and information is saved until the next login of the user.

-

-   Use case: Search Room

Brief Description: The use case allows a renter to search rooms with their preferences which includes destination, time and the number of people staying.

Actors: Renters

Preconditions:
Access the application
The renter must access the KOOLB application on their smartphone.

Flow of event:
1. Home page shows a search bar along with recommendations.
2. The renter touches the search bar.
3. A dialog appears with a list of text boxes for information including: destinations, check-in date, check-out date and the number of people staying.
4. The renter fills in choices.
5. The renter presses the button "Find",
6. The application will collapse the dialog, then retrieve and display a list of rooms available and with a map corresponding to renter's choices in the homepage. Prices of each room are pinned on the map equivalent to its location.
7. The use case ends when the renter moves to another application page.

Postconditions:
View detail information
After a list of rooms is shown, the renter can click on each room in order to be navigated to the detail page with detailed information for each room
Book rooms
The renter can do the room booking after navigating the detail page.

Alternative flow:
Not filling in choices
1. At step 4, if the renter does not fill in information in the dialog, then triggers the not filling choices flow.
2. The dialog is set with default choices.
3. Now, if the renter can press the button "Find", and step 6 of Basic flow occurs. If not, the dialog will wait for the renter to fill in the information.
Clear all choices
1. At step 5, if the renter presses the button "Clear all", then triggers the "Clear all choices" flow.
2. This dialog will be back to its initial state.
3. Now, "Not filling in choices" flow occurs.
Add a room to favorite list
1. There is a heart icon shown in every room displayed in the result of searching.
2. The renter presses the heart icon.
3. The room is added to the renter's favorite list.

Extension point:
Default searching
Four pieces information for searching are set default with:
-      Destination: Everywhere
-      Time: One week
-      Guest: Add guests

-

Use case: Notify
Brief Description: This use case describes the notifications acting in Renter and Host.
Actors: Renter
Preconditions: The user must log in or sign up first The user is required to log in or sign up first to view the notifications

Flow of event:
1. The application shows a list of notifications.
2. Each notification contains the title and brief description.
3. The system shows the most recent notifications
4. The user scrolls to view more notifications.
5. The user clicks a notification to view more detail.
6. The use-case ends when the user ends the application or moves to other pages.

Postconditions:

Alternative flow:
The notification change to have read
In step 4, when the user clicks a notification, the background of notifications will turn darker (gray) when the user has finished reading.

Extension point:

-

-	**Use case: Setting**

Brief Description: This use-case describes the setting process of the app KOOLB.

Actors: Renter

Preconditions:
Sign into the application
Actors must log in to the application before they can chat with each other.
Press the right icon
The actor must press the right icon in order to be navigated to the chat page.

Flow of event:
Setting of Personal information and security
1. Users click on personal information and security setting section
2. The current username and password are displayed (password is displayed in encrypted format)
3. Users click on either username or password to change the information
4. Users submit their personal information changed, the system will validate and update the information
Setting of General information
1. Users click on the general information setting section
The current Country, Currency, and Language
2. Users click on either username or password to change the information
3. This information will be changed according to existing data. Every country(currency, language) available in this app will appear in scrolling lists.
4. Users click on the corresponding option. At the time a new option is chosen, the app will be updated.
Setting of notification
1. Users click on the personal notification setting section
2. The current notification state will be displayed (Banners, sounds, badges, announcements…)
3. Users click on the corresponding option to change the way the app will notice new information changed.
4. At the time the new option is chosen, the app will be updated.
About Us section
1. Users click on About Us section
2. The app displays all information about this app's basic information about the store and its founders, explains the

company's purpose and how it differs from the competition, and encourages discussion and interaction.

Logout section

1. Users click on the Logout long button at the end of the setting page.

2. The app logs out the current account. Any process in progress will either temporarily be saved or dismissed.

Postconditions:

Alternative flow:

Invalid Password

1. At step 3 of "Setting of Personal information and security" basic flow, if password of actor is invalid, then trigger the first alternative flow

2. Display the message about invalid password

3. Require actors to correct password

4. Repeat this step until both username and password are valid

5. End use case

Invalid Username

1. At step 3 of "Setting of Personal information and security" basic flow, if username of actor is invalid, then trigger the second alternative flow

2. Display the message about invalid username

3. Require actors to correct email or register username as new account

4. Repeat this step until both username and password are valid

5. End use case

Extension point:

Update profile

The application changes the UI so that the user can update profile pages and the system allows users to change their information.

Language setting

The application allows users to change their language and the system will update into the database.

Configure payment

The application allows users to change the payment method and default one.

-

-	**Use case: Configure Payment**

Brief Description: This use-case describes the configured payment process of the app KOOLB

Actors: Renter

Preconditions:

User must be in the setting use case

Account available

The user need to have the account available in the app to perform setting use-case

Stable Internet Connection

In order to save setting successfully, users need to have a stable internet connection

Flow of event:

The user configure the payment of the account

Postconditions:

Alternative flow:

Extension point:
Return back to the setting page.

-

- **Use case: Configure Language Setting**
Brief Description: This use-case describes the language setting process of the app KOOLB.
Actors: Renter
Preconditions: **User must be in the setting use case** Account available The user need to have the account available in the app to perform setting use-case Stable Internet Connection In order to save setting successfully, users need to have a stable internet connection
Flow of event: Change language of the application
Postconditions:
Alternative flow:
Extension point: Return back to the setting page.

-

- **Use case: Update Profile**
Brief Description: This use-case describes the update profile process of the app KOOLB.
Actors: Renter
Preconditions: **User must be in the setting use case** Account available The user need to have the account available in the app to perform setting use-case Stable Internet Connection In order to save setting successfully, users need to have a stable internet connection
Flow of event: The user clicks on the profile setting option in setting page
Postconditions:
Alternative flow:

Extension point:
Return back to the setting page.

-

-	**Use case: Analyze Host Insight**

Brief Description: This use-case describes the tools for hosts to manage every information and data about their service.

Actors: Host

Preconditions:
Host should finish posting the accommodation on the public site
In order to view the statistics and checklist of the accommodation, the host should have that accommodation publicized so that the app will generate real-time data based on the number of guests visiting the accommodation.

Flow of event:
Cleaning and supply checklist
1. Click on the Cleaning and Supply checklist bar
2. Manually checking the "cleaning and supply" task in real situations of the accommodation and either adding or ticking on the completed (incompleted) tasks.
3. Clicking submit to complete the process of checking. Any completed tasks will be hidden and the uncompleted tasks will be visible to the host insight.
Statistic of the Accommodation

Postconditions:
The checklist should be temporarily automatically saved
In case the user fails to save the checklist due to networking lost, the process will be automatically saved and appear once the user reconnects the server.
The statistic automatically updated
Every time the application receives information of new guests in the accommodation, the statistics will change and the history of their information will be added to the database

Alternative flow:
The users don't define the name for the new tasks
1. In step 2 of basic flow, if the host fails to define the name of the new task added, it will trigger the first alternative flow.
2. Display red of the new checkbox that has not had a specific name, stop users from submitting the checklist.
3. Require the users to either input a name for the check box or delete that checkbox.
4. Repeat this step until the checkbox has been filled or deleted
5. End use case

Extension point:

-

-	**Use case: Assist Customer Service**

Brief Description: The KoolB App Customer service process is described in this use case.

Actors: Admins

Preconditions:
Actor must be admin
Actors have to be logged in and be an admin so that they can manage FAQ of other roles.

Flow of event:
1. The system displays a list of information about FAQ of either host or renters
2. Actors answer questions of the customer
3. Actors submit information

Postconditions:
The answer to each question must be delivered to the right customers asking.

Alternative flow:
Answer box for question is leaving blank
1. Announce users that the question is still not answered.
2. Make sure the users answer the question (box filled) before sending it back to customers.

Extension point:
User disconnected from app when posting an accommodation
At step 5 in Basic Flow, if the user is disconnected from the app because of many reasons, then all the progress is stopped and information is saved until the next login of the user.

-