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 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**Table of Contents**

[**I. REGISTER/ LOG IN Use-cases:**](#_heading=h.xh29hp8bzs82) **4**

[**II. NAVIGATOR/ ACCOMMODATION DETAIL VIEW/ EXPLORE Use-cases:**](#_heading=h.n90c7c1cp1er) **5**

[**III. BOOKING/ CHECK AVAILABILITY/ VERIFY PAYMENT Use-cases:**](#_heading=h.ugvk295dondm) **7**

[**IV. POST ROOM INFORMATION/ HOST INSIGHT/SORT FILTER Use-cases:**](#_heading=h.uznjr9z7hwly) **9**

[**V. CHECK IN/ CHECK OUT/ CANCEL BOOKING Use-cases:**](#_heading=h.e5kwu0q16x3q) **12**

[**VI. SEARCH ROOM/ RESERVATION LIST VIEW/ FAVORITE LISTS Use-cases:**](#_heading=h.fe9o1dutzq7w) **13**

[**VII. RATING/ CUSTOMER SERVICE/ CHATBOX Use-cases:**](#_heading=h.y71k9vkgx44b) **15**

[**VIII. SETTING/ NOTIFICATION Use-cases:**](#_heading=h.2j1mr5rzbqk) **17**

[**IX. CONFIGURE PAYMENT/ LANGUAGE SETTING/ UPDATE PROFILE Use-cases:**](#_heading=h.8fa80gklbbfy) **20**

**Use-Case Specification**: KOOLB

# REGISTER Use-cases:

| **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. |

# LOG IN Use-cases:

| **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. |

# NAVIGATOR/ ACCOMMODATION DETAIL VIEW/ EXPLORE Use-cases:

| **Use case: Navigator** | **Use case: Accommodation Detail View** | **Use case: Explore** |
| --- | --- | --- |
| **Brief Description:** This use-case is the navigator to every other use-cases. It is under the Navigation bar in UI design. | **Brief Description:** This use-case describes how the system acts when the user views the details of accommodation. | **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:** User | **Actors:** Renters and Hosts | **Actors:** Renters and Hosts |
| **Preconditions:**  Users have to first log in or register to be granted permission to enter the homepage section with a navigation bar. | **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 | **Preconditions:**  Users have to first log in or register to be granted permission to enter the homepage section. |
| **Flow of event:**  The navigation bar in cluded of many button that navigate to the specific use-cases. | **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. | **Flow of event:**  The page can be scrolling up and down to view all the Feature Accommodation, Category, Search bar… |
| **Postconditions:**  Be able to travel user to the expected use-cases. | **Postconditions:**  It should be able to navigate Renters to Booking or back to Explore use-case. | **Postconditions:**  Be able to present the feature and category that the user demand. |
| **Alternative flow:**  - | **Alternative flow:**  - | **Alternative flow:**  **-** |
| **Extension point:**  - | **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. | **Extension point:**  **-** |

# BOOKING/ CHECK AVAILABILITY/ VERIFY PAYMENT Use-cases:

| **Use case: Booking** | **Use case: Check Availability** | **Use case: Verify payment** |
| --- | --- | --- |
| **Brief Description:** This use case describes how a user books a room in the system.. | **Brief Description:** This use case describes the availability checking. | **Brief Description:** This use-case describes the verify payment process of the app KOOLB. |
| **Actors:** Renters | **Actors:** Renters | **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. | **Preconditions:**  **Booking**  Users must booked this accommodation first | **Preconditions:**  **Booking**  Users must booked this accommodation first |
| **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 | **Flow of event:**  The system check if the duration is available | **Flow of event:**  The system check if the payment method is valid or not. |
| **Postconditions:**  **The system wait for the host to accept**  The system will notify the host and wait for he/she accepts | **Postconditions:**  **-** | **Postconditions:**  - |
| **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. | **Alternative flow:**  - | **Alternative flow:**  - |
| **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. | **Extension point:**  **-** | **Extension point:**  **-** |

# POST ROOM INFORMATION/ HOST INSIGHT/SORT FILTER Use-cases:

| **Use case: Post Room Information** | **Use case: Host Insight** | **Use case: Sort and Filter** |
| --- | --- | --- |
| **Brief Description:** The KoolB App posting accommodation process is described in this use case. | **Brief Description:** This use-case describes the tools for hosts to manage every information and data about their service. | **Brief Description:** This use-case describes the process of filtering and sorting in the search bar in app KOOLB. |
| **Actors:** Host | **Actors:** Host | **Actors:** Renters |
| **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 | **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. | **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. 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 | **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**  - | **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:**  - | **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 | **Postconditions:**  **Display wanted result**  The result users wanted to have after applying filtering and sorting should be displayed accurately and successfully. |
| **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 | **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 | **Alternative 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 | **Extension point:**  **-** | **Extension point:**  **-** |

# 

| Use case: Cancel Booking |
| --- |
| Brief Description: This use case describes how a user cancel their bookings in the system. |
| Actors: Renters |
| Preconditions:The renter must login or sign up firstThe renter is required to login first or create an account if he/she is a new user.The renter must have a booking that need to be canceledThere is no point in canceling a non-existing booking, so the app must be sure to have the booking. |
| Flow of event:1. The system update the database2. The system notify to the Host |
| Postconditions:The system wait for the host to acceptThe system will notify the host and wait for he/she accepts |
| Alternative flow:The Renter can add a reasonThe Renter can give a reason why they want to cancel the booking. |
| Extension point:- |

# 

| 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 updatedStatus such as ‘vacant’ or ‘booked’ should be updated. Renters can now book this accommodation |
| Alternative flow:- |
| Extension point:RatingAfter 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: Favorite Lists |
| --- |
| Brief Description: This use case shows a list of saved accommodations |
| Actors: Renters |
| Preconditions:Actor must log in or sign up firstThe renter is required to login first to save places.Actor must press the correct icon to navigate to the list viewThe 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 accommodations3. Renter click to another screen, the flow ends here. |
| Postconditions:List of saved item is updatedActors can now see their newly saved accommodation in the list of saved places. |
| Alternative flow:Actor unsave the locations1. When an accommodation is saved and actors press the ‘heart’ icon again, this flow is triggered.2. Heart icon transform back to the unsaved icon3. The accommodation is removed from saved list4. The flow end hereHost deletes the saved accommodation1. When an item in saved list is deleted by host, this flow is triggered2. The list is updated; i.e. the saved place that is deleted will also deleted from saved listRenter has no saved accommodationThe screen shows a text “Wow, so empty” |
| Extension point:Renter add new favorite accommodationWhen user press the ‘heart’ icon of an accommodation, the list of favorite will be updated by adding that accommodation |

# 

| Use case: Reservation List View |
| --- |
| Brief Description: This use-case describes how the user views their reservations. |
| Actors: Hosts |
| Preconditions:The user must log in or sign up firstThe 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 reservations4. 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 reservation1. The user delete a reservation2. The system remove that from database3. The application remove from the list4. The application updates the list. |
| Extension point:- |

# 

| 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 applicationThe 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 informationAfter 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 roomBook roomsThe renter can do the room booking after navigating the detail page. |
| Alternative flow:Not filling in choices1. 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 choices1. 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 list1. 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 searchingFour pieces information for searching are set default with:Destination: EverywhereTime: One weekGuest: Add guests |

# 

| Use case: Chatbox |
| --- |
| 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 applicationActors must log in to the application before they can chat with each other.Press the right iconThe 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 accommodationAny 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 messages1. 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 pageActors 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: Customer Service |
| --- |
| Brief Description: The KoolB App Customer service process is described in this use case. |
| Actors: Admins |
| Preconditions:Actor must be adminActors 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 renters2. Actors answer questions of the customer3. 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 blank1. 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 accommodationAt 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: Rating |
| --- |
| 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 dataThe 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 updatedRating of accommodation will be recalculated and updated. New reviews will also be displayed. |
| Alternative flow:No StarsThe renter only inputs the comments of the accommodation and is asked to input the stars before submitting.The renters forget to submit feedbacksThe renters may forget to rate the service after use. The notification will be sent to the renter in case this happens. |
| Extension point:The renter login or create an accountThe system will let the user login or create a new account in order to rate.The renter lost connectionAt step 1, 2 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. |

# 

| Use case: Notification |
| --- |
| 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 firstThe 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 notifications4. 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 readIn 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: Host and Renter |
| Preconditions:Sign into the applicationActors must log in to the application before they can chat with each other.Press the right iconThe actor must press the right icon in order to be navigated to the chat page. |
| Flow of event:Setting of Personal information and security1. Users click on personal information and security setting section2. The current username and password are displayed (password is displayed in encrypted format)3. Users click on either username or password to change the information4. Users submit their personal information changed, the system will validate and update the informationSetting of General information1. Users click on the general information setting sectionThe current Country, Currency, and Language2. Users click on either username or password to change the information3. 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 notification1. Users click on the personal notification setting section2. 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 section1. Users click on About Us section2. 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 section1. 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 Password1. At step 3 of “Setting of Personal information and security” basic flow, if password of actor is invalid, then trigger the first alternative flow2. Display the message about invalid password3. Require actors to correct password4. Repeat this step until both username and password are valid5. End use caseInvalid Username1. At step 3 of “Setting of Personal information and security” basic flow, if username of actor is invalid, then trigger the second alternative flow2. Display the message about invalid username3. Require actors to correct email or register username as new account4. Repeat this step until both username and password are valid5. End use case |
| Extension point:Update profileThe application changes the UI so that the user can update profile pages and the system allows users to change their information.Language settingThe application allows users to change their language and the system will update into the database.Configure paymentThe application allows users to change the payment method and default one. |

# 

| 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 caseAccount availableThe user need to have the account available in the app to perform setting use-caseStable Internet ConnectionIn 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: 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 caseAccount availableThe user need to have the account available in the app to perform setting use-caseStable Internet ConnectionIn 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 |
| --- |
| 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 caseAccount availableThe user need to have the account available in the app to perform setting use-caseStable Internet ConnectionIn 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. |