<Team Name>

<Project Name>
Use-Case Specification

Version <1.0>

[Note: The following template is provided for use with the Rational Unified Process. Text enclosed in square brackets and displayed in blue italics (style=InfoBlue) is included to provide guidance to the author and should be deleted before publishing the document. A paragraph entered following this style will automatically be set to normal (style=Body Text).]

[To customize auto

matic fields in Microsoft Word (which display a gray background when selected), select File>Properties and replace the Title, Subject and Company fields with the appropriate information for this document. After closing the dialog, automatic fields may be updated throughout the document by selecting Edit>Select All (or Ctrl-A) and pressing F9, or simply click on the field and press F9. This must be done separately for Headers and Footers. Alt-F9 will toggle between displaying the field names and the field contents. See Word help for more information on working with fields.]

Revision History

Date	Version	Description	Author
20/11/2022	1.0	Initial use-case specification	Nguyen Hoang Khang, Ta Quang Khoi
22/11/2022	1.1	Draw use-case UML diagram	Bui Duy Bao, Nguyen Huynh Sang
25/11/2022	1.2	Full use-case revision	Nguyen Hoang Khang, Vo Huynh

Table of Contents

1. Use-case Model 4

2. Use-case Specifications 4

2.1 Use-case: Sign up 4

2.2 Use-case: Sign in 5

2.3 Use-case: Create a channel 6

2.4 Use-case: Subscribe to a channel 6

2.5 Use-case: Turn on stream 6

2.6 Use-case: Turn off stream 7

2.7 Use-case: Join the stream’s room 7

2.8 Use-case: Leave the stream’s room 8

2.9 Use-case: Chat to stream room 9

2.10 Use-case: Adjust video preferences 9

2.11 Use-case: Viewer donate to streamer 9

2.12 Use-case: Viewer deposit money 10

2.13 Use-case: Link to third party 11

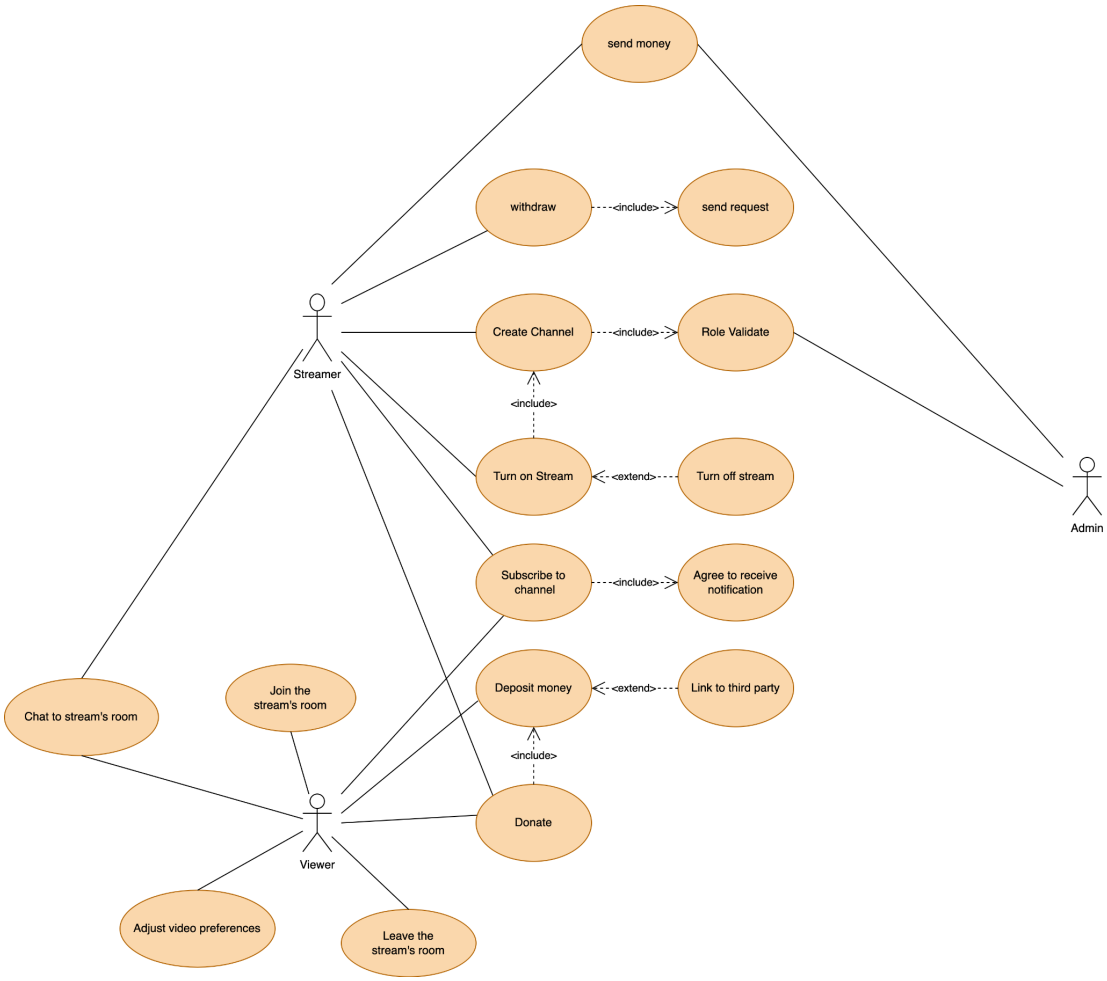
2.14 Use-case: Withdraw money 11

2.15 Use-case: Send request withdraw money 11

2.16 Use-case: Admin send money 12

1. Use-case Model

[*Put an image of the use-case diagram modeling all use-cases in this section*]



2. Use-case Specifications

2.1 Use-case: Sign up

Use case Name	Sign up a new account
Brief description	This use case allows the viewer and streamer to create their own account.
Actors	Viewer, streamer.
Basic Flow	1. At the homepage, the user will hover on the circle avatar and click on the "sign up" button. 2. System redirects to sign-up page. 3. User connects to Google, or Facebook account or manually enters the username, and password and accepts the muly's agreement 4. The user submits a form to finish signing up.

	5. The system will show a success message and redirect to the homepage after the user submits successfully.
Alternative Flows	**Alternative flow 1: The user does not enter the username.** From #2 of the basic flow, username text input will be focused and highlighted. Continue step #3 in the basic flow **Alternative flow 2: The user does not enter the password.** 1. From #2 of the basic flow, password text input will be focused and highlighted. 2. Continue step #3 in the basic flow **Alternative flow 3: User enters the wrong password format.** 3. From #2 of the basic flow, password text input will be focused and highlighted. 4. Continue step #3 in the basic flow **Alternative flow 4: User does not accept muly's agreement.** 1. From #2 of the basic flow, muly's checkbox will be highlighted. 2. Continue step #3 in the basic flow
Pre-conditions	User goes to the homepage at muly.com
Post-conditions	The user successfully signs up an account.

a.

2.2 Use-case: Sign in

Use case Name	Sign in
Brief description	This use case allows the viewer and streamer log in to muly website.
Actors	Viewer, streamer.
Basic Flow	1. At the homepage, the user will hover on the circle avatar and click on the "sign in" button. 2. System redirects to the sign-in page. 3. User connects to Google, or Facebook account or manually enters the username, and password. 4. User submits username and password. 5. System will show a success message and redirect to the homepage after the user submits successfully.
Alternative Flows	**Alternative flow 1: The user does not enter the username.** 1. From #2 of the basic flow, username text input will be focused and highlighted. 2. Continue step #3 in the basic flow **Alternative flow 2: The user does not enter the password.** 1. From #2 of the basic flow, password text input will be focused and highlighted. System will show warning messages below the focused input. 2. Continue step #3 in the basic flow **Alternative flow 3: User enters the wrong password format.** 1. From #2 of the basic flow, password text input will be focused and highlighted. The system will show warning messages below the focused input.

	2. Continue step #3 in the basic flow
Pre-conditions	User goes to the homepage at muly.com
Post-conditions	The user signs in successfully.

b.

2.3 Use-case: Create a channel

Use case Name	Create a channel.
Brief description	This use case describes how a user can create a channel for themselves.
Actors	Streamer
Basic Flow	1. At the profile screen, streamers click on the 'Create Channel' button. 2. User types in the name of the channel they want to create or by default will be their username. 3. Users click on the 'Create' button. 4. System checks whether the channel name is already taken or not. 5. System informs that the channel has been created.
Alternative Flows	**Alternative flow 1: The channel name has been taken or is not appropriate** 1. From #2 of the basic flow, the user enters another name. 2. Continue step #3 in the basic flow... **Alternative flow 2: User press on the 'cancel' button:** 1. In step #2, the user does not want to create a channel anymore and click on the 'cancel' button. 2. System close creates a channel panel.
Pre-conditions	● User has a Streamer role, validated by the Admin. ● User has not created a channel.
Post-conditions	Streamer created a channel successfully.

c.

2.4 Use-case: Subscribe to a channel

Use case Name	Subscribe to a channel.
Brief description	This use case describes how a viewer or a streamer can subscribe to a streamer channel.
Actors	Viewer, Streamer.
Basic Flow	1. On the homepage, the user clicks on a profile of a streamer. 2. User click on the 'Subscribe' button next to the avatar of the streamer. 3. System adds users to the subscribers' list. 4. System changes the 'Subscribe' button to 'Unsubscribe' button.
Alternative Flows	
Pre-conditions	User has agreed to receive a notification mail from the system.
Post-conditions	The viewer, and streamer successfully subscribed to a channel and will receive a notification when that streamer goes live.

d.

2.5 Use-case: Turn on stream

Use case Name	Turn on stream

Brief Description	This use case describes how a streamer can start a stream session.
Actors	Streamer
Basic Flow	1. On the home page, a user can click on the stream button. 2. A streaming panel pops up. 3. Streamer type in a name for the streaming session. 4. Streamer clicks on the “Stream” button. 5. System sends notification mail to subscribers. 6. System starts a streaming thread for the user.
Alternative Flows	**Alternative flow 1: Streamer does not type in any name for streaming session** 1. In #4, the user does not type in any name for the streaming session. 2. System uses the default streaming name with a placeholder of the channel name or the previous streaming session name if exists. **Alternative flow 2: Streamer chooses to stream after a timeframe** 1. In #2, the user chooses not to stream right now. 2. The system holds for a chosen timeframe. 3. Continue step #3 in the basic flow.
Pre-conditions	● User has a Streamer role validated by the Admin and has created a channel. ● The Streamer does not have any current active streaming sessions.
Post-conditions	The user starts a streaming session successfully.

e.

2.6 Use-case: Turn off stream

Use case Name	Turn off stream
Brief Description	This use case describes how the Streamers can turn off the current streaming session.
Actors	Streamer
Basic Flow	1. On the streaming page, the user clicks on the “Stop Streaming” button. 2. The system asks if the user really wants to close the streaming session. 3. The user hit the “YES” button. 4. The system closes the streaming session and forces the viewers currently in the stream room to leave.
Alternative Flows	**Alternative flow 1: Streamer decides to keep streaming.** 1. In #3, the streamer clicked on the “NO” button. 2. System exit the basic flow.
Pre-conditions	● The user has a Streamer role. ● The user is currently running a streaming session.
Post-conditions	The user successfully turns off their streaming session.

2.7 Use-case: Join the stream’s room

Use case Name	Join the stream’s room.
Brief description	This use case describes how the Viewers can join the stream’s room.

Actors	Viewer
Basic Flow	1. At the homepage, the viewer enters the stream title's keywords (or Stream ID) on the 'Search' field or taps on an arbitrary featured stream thumbnail. 2. If the viewer uses the 'Search' field, then clicks on the 'Search' button to start searching a stream 3. The system displays the stream found 4. The viewer enters the stream's room password (if required) 5. Viewers tap on the 'Join' button to join the stream 6. The system displays the stream, and channel information… with the correct ID
Alternative Flows	**Alternative flow 1: Viewer cannot find the stream searched** 1. The system announces the stream ID does not exist 2. From #1 of the basic flow, the user enters another term 3. Continue step #2 in the basic flow **Alternative flow 2: Viewers can not join the stream due to being banned** 1. From #6 of the basic flow, display the notification: "You are banned from the channel" 2. Viewers can contact the admin's system or the official streamer for permissions 3. Continue step #4 in the basic flow
Pre-conditions	Viewers go to the homepage's link web.
Post-conditions	The user successfully joins the desired streams.

2.8 Use-case: Leave the stream's room

Use-case Name	Leave the stream's room.
Brief description	This use case describes how the Viewers can turn off the current stream.
Actors	Viewer
Basic Flow	1. At the stream interface, viewers tap on the 'Leave' button 2. The system removes participant from the room 3. The system displays the home display 4. The system can recommend other related videos
Alternative Flows	**Alternative flow 1: Streamer turn off the stream before** 1. The system displays a 'Thank you' interface and exit option. 2. Continue step #2 in the basic flow **Alternative flow 2: The stream is interrupted before leaving** 1. From #2 of the basic flow, display the notification: "The current stream is interrupted!" 2. Continue step #2 in the basic flow
Pre-conditions	Viewers go to the valid live room and join successfully.
Post-conditions	The viewers leave the stream's room.

f.

2.9 Use-case: Chat to stream room

Use case Name	Chat to a streaming room
Brief Description	This use case describes how the Viewer text to a chatbox of a live stream video
Actors	Viewer
Basic Flow	1. At the homepage, the user enters text on the chatbox 2. Users click on the 'Send' button to start typing comments. 3. The system displays the text sent and the name of the user on the chatbox
Alternative Flows	**Alternative flow 1: User is banned from current Livestream** 1. From #2 of the basic flow, display the notification: "You are banned from the channel" 2. The user taps on the "Close" button and the use case ends **Alternative flow 2: text has inappropriate words, hate speeches, toxicities,...** 1. From #2 of the basic flow, the system displays the name of the sender and the text "[This chat has inappropriate content and has been deleted]"
Pre-conditions	User is currently at a live streams page and not in the fullscreen mode
Post-conditions	The text is displayed on the chatbox and the chatbox keeps floating new texts

g.

2.10 Use-case: Adjust video preferences

Use case Name	Adjust video preferences
Brief Description	This use case describes how the viewer change the video resolution
Actors	Viewer
Basic Flow	1. On the streaming video's page, the user clicks on the "Settings" button 2. Users then choose "Video Quality" 3. The system displays the list of available resolutions, with the current selection highlighted 4. The user chooses a specific resolution from the dropdown list displayed 5. The system changes the resolution of the current live-streaming video to the selected one. 6. System closes the "Settings" window
Alternative Flows	
Pre-conditions	User is currently at a live streams page
Post-conditions	The user successfully changes the video's resolution

h.

2.11 Use-case: Viewer donates to streamer

Use case Name	Donate to streamer
Brief Description	This use case describes how the Viewer donate to Streamer
Actors	Viewer, Streamer
Basic Flow	1. On the streaming page, the viewer clicks to donate button to start the donation 2. The viewer enters the amount of token to send to the streamer

	3. Viewer click button sends to send token 4. A dialog will pop up then the viewer will click on to button confirm to confirm sending token 5. Increase the token balance of streamer 6. Display amount of tokens that the viewer sent
Alternative Flows	**Alternative flow 1: Token balance of the user is not enough** 1. From #3 of the basic flow, the token balance of the viewer is not enough 2. Pop up modal with deposit button 3. The viewer clicks the button deposit to buy token 4. The viewer moves to deposit money using case 5. Continue step #2 **Alternative flow 2: The user does not click the button confirm** 1. From #4 of the basic flow, the user does not click the button confirm 2. Move back the user to step #2
Pre-conditions	Viewer logged in to platform Viewer move to streaming view of streamer
Post-conditions	The user successfully send the token to streamer and streamer successfully receive a token from viewer

i.

2.12 Use-case: Viewer deposit money

Use case Name	Deposit money
Brief description	This use case describes how viewer deposit to the platform
Actors	Viewer
Basic Flow	1. At the pop-up of send token on the streaming page or deposit page, the viewer enters the deposit button 2. The viewer enters the amount of token 3. Pop-up dialog for the viewer to choose the third party 4. Link viewer to the chosen third party page to pay for that amount token (momo, ZaloPay) 5. The viewer clicks the button confirm of the third party to confirm paying 6. Move back the user to the current screen 7. Increase the token balance of viewer
Alternative Flows	**Alternative flow 1: Viewer moves back to current screen** 1. From #3 or #4 of the basic flow, the viewer exit page 2. Move to step #6 in the basic flow **Alternative flow 2: Viewer is not connected to the third party** 1. From #4 of the basic flow, the viewer is not connected to third-party 2. Move the user to the connect page of the party 3. Move to step #4 of the basic flow **Alternative flow 3: Viewer does not have enough money in the third party** 1. From #4 of the basic flow, the viewer does have enough money in the third

	party 2. The third-party pop-up dialog with the message NOT ENOUGH MONEY 3. Viewer clicks on the button close 4. Move to step #6 in the basic flow
Pre-conditions	Viewer in deposit page or in streaming page
Post-conditions	Viewer successfully or fail to deposit to the platform

2.13 Use-case: Link to third party

Use case Name	Link to third party
Brief description	This use case describes how to link to third party wallet (momo, zalo pay)
Actors	Viewer
Basic Flow	1. At the deposit page, the viewer clicks to connect button to connect to wallet 2. Confirm connecting the wallet with platform
Alternative Flows	**Alternative flow 1: Viewer does not confirm to connect with wallet** 1. From #2 of the basic flow, the viewer does not confirm to connect with wallet 2. Move back user to the current page
Pre-conditions	Viewer logged in to platform Viewer stay at deposit page
Post-conditions	The user successfully connect with third party

2.14 Use-case: Withdraw money

Use case Name	Withdraw money
Brief description	This use case describes how a user can withdraw money they deposit in the system.
Actors	Streamer
Basic Flow	1. At withdraw page, the streamer clicks button withdraw 2. Pop-up modal and streamer will enter the amount token and the card information 3. Check the amount of streamers and whether it is valid or not 4. Streamer moves to send request use case 5. Admin confirms the request and send the money to the streamer in 2- 4 days
Alternative Flows	**Alternative flow 1: Streamer does not connect to the platform** ● From #2 of the basic flow, streamer clicks button add to add card information ● Fill in the information of the card and click button save to save the card ● Move to step #2 of the basic flow
Pre-conditions	Streamer logged in to the platform Streamer move to the withdraw page
Post-conditions	Streamer successfully send the request to admin

2.15 Use-case: Send request to withdraw money

Use case Name	Send withdraw request to admin
Brief description	This use case describes how a streamer can send withdraw request

Actors	Streamer
Basic Flow	6. At withdraw button in the pop-up of use case withdraw, streamer clicks button withdraw 7. pop up message successful to streamer
Alternative Flows	
Pre-conditions	Streamer logged in to the platform Streamer stays at the pop-up of withdraw use case
Post-conditions	Streamer successfully sends the request to admin

2.16 Use-case: Admin sends money

Use case Name	Admin sends money
Brief description	This use case describes how an admin can send money to streamer
Actors	Admin
Basic Flow	1. At confirm page of the admin, then the admin clicks the button send to send money to the streamer
Alternative Flows	**Alternative flow 1: Streamer rejects request** 1. From #1 of the basic flow, the streamer click rejects request and send mail back to streamer 2. Move to the current stage of streamer **Alternative flow 2: Streamer pending request** 2. From #1 of the basic flow, the streamer clicks the pending request and sends mail back to streamer 3. Move to the current stage of streamer
Pre-conditions	Admin logged in to the platform Admin in confirm request page
Post-conditions	Admin successfully reject, accept or pending the request

a.