Use Cases

for

Meet-Now

A Facetime Application

Version 1.0 approved

Prepared by :

Aman Seervi
Harshit Gupta
Naman Mulani
Junaid

17-10-2023

Revision History

Name	Date	Reason For Changes	Version
Aman , Naman , Harshit , Junaid	17-10-23	First implementation	1.0

Use Cases

Use Case ID:	1
Use Case Name:	Peer-to-Peer chat
Created By:	Harshit Gupta	Last Updated By:	Harshit Gupta
Date Created:	17-10-2023	Date Last Updated:	17-10-2023
Actors:	Patrons
Description:	A patron can send messages to his friends and friends can also reply back.
Preconditions:	Sender and Receiver should be friends.
Postconditions:	Text messages will be visible in the chat box.
Normal Flow:	**1.0 Adding a Friend** 1 . A patron will register his account in the application. 2 . Patrons will click on add a friend option. 3 . A dialog box will appear to enter the email id of the receiver. 4 . After entering the email id of the receiver, a friend request will be sent to the receiver. 5 . Now the receiver will either accept or reject the friend request. 6 . After accepting the request, the receiver will be added to the dashboard of the patron and the request will be deleted in case of rejection. **1.1 Sending a Message** 1 . Sender will select the friend to whom he wants to send the message. 2 . Sender will write a message for his friend. 3 . Sender will send a message. 4 . Sender can receive messages asynchronously.
Alternative Flows:	None
Exceptions:	1 . Patron is unable to register successfully. 2 . Receiver email id is not a valid email address. 3 . Chat databases may blow up.
Includes:	1 . Adding a friend.
Priority:	high
Frequency of Use:	Approximately 10 users per patron (depends on the patron)
Business Rules:	None
Special Requirements:	Email id of the receiver , Internet connectivity
Assumptions:	Email address of the receiver should exist.
Notes and Issues:	Expect a high frequency of executing this use case within the first 2 weeks after the system is released.

Use Case ID:	2
Use Case Name:	Adding a friend
Created By:	Harshit Gupta	Last Updated By:	Harshit Gupta
Date Created:	17-10-23	Date Last Updated:	17-10-23
Actors:	Patrons
Description:	A patron can add a friend request using the receiver's email id. Receiver can accept or reject the request accordingly.
Preconditions:	Sender should know the email id of the receiver.
Postconditions:	If the receiver accepts the friend request then he will be added to the sender's dashboard. If the receiver rejects the friend request then the request will be deleted.
Normal Flow:	**1.0 Adding a friend** 1 . A patron will register his account in the application. 2 . Patron will click on add a friend option. 3 . A dialog box will appear to enter the email id of the receiver. 4 . After entering the email id of receiver, a friend request will be send to the receiver. 5 . Now the receiver will either accept or reject the friend request. 6 . After accepting the request, the receiver will be added to the dashboard of the patron and the request will be deleted in case of rejection.
Alternative Flows:	None
Exceptions:	1 . Patron is unable to register successfully. 2 . Receiver email id is not a valid email address.
Includes:	1 . Chat messages 2 . Communication via Audio only 3 . Video Call
Priority:	Very high
Frequency of Use:	Approximately 10 users per patron (depends on the patron)
Business Rules:	None
Special Requirements:	Email id of the receiver , Internet connectivity
Assumptions:	Email address of the receiver should exist.
Notes and Issues:	Expect a high frequency of executing this use case within the first 2 weeks after the system is released.

Use Case ID:	3
Use Case Name:	Sharing screen for presentations

Created By:	Junaid Anwar	Last Updated By:	Junaid Anwar
Date Created:	17-10-2023	Date Last Updated:	

Actors:	Patrons
Description:	A Patron accesses the MeetNow application from the link to the website, registers if not already registered, or else logs in and sends a friend request to his friends using the onboard facility, and either joins a meeting already started by his friends or creates a meeting for his friends to join.
Preconditions:	1. Patron logs onto the application. 2. Patron's identity has been authenticated. 3. Patrons send friend requests to their friends. 4. Patron joins an already existing meeting or creates a new meeting. 5. Patron starts sharing the screen using the onboard facility.
Postconditions:	1. The Patrons data is stored in the existing database, along with his friends data 2. Patterns logs off
Normal Flow:	**1.0 The Patron starts the meeting** 1. Patrons send friend requests to their friends. 2. Patron creates a new meeting. 3. Patron starts sharing the screen using the onboard facility. 4. After the patron joins the meeting, they have the facility to share their screen 5. After the presentation is done, the patron can stop the screen share and leave the meeting. 6. then the patron logs off the web application
Alternative Flows:	**1.1 The Patron's friend starts the meeting** (after step 1) 1. The patron joins a new meeting created by his friends. 2. return to step 4
Exceptions:	
Includes:	None
Priority:	Medium
Frequency of Use:	Approximately 100 users, average of 10 per hour
Business Rules:	not currently set up
Special Requirements:	1. Patron shall be able to leave the meeting anytime 2. Patron can choose which window to share 3. Patron can mute/unmute their mic
Assumptions:	Assume that only one Patron can share a presentation at a time for at most 4 people.
Notes and Issues:	Expect a high frequency of executing this use case within the first 2 weeks after the system is released.

Use Case ID:	4
Use Case Name:	Communication via Audio only
Created By:	Naman Mulani	Last Updated By:	Naman Mulani
Date Created:	October 12,2023	Date Last Updated:	October 17,2023
Actors:	Patron
Description:	A Patron can interact with web-clients through communication via Audio-mode . Optionally can screen-share and provide video facility.
Preconditions:	1. Patron is logged-in. 2. Patron sends friend requests to other Users/web-clients. 3. Patron joins an already existing meeting or creates a new meeting. 4. Patron generates Event-based functionality by clicking on the audio button.
Postconditions:	1. Patrons can interact now through audio , which is also bidirectional in nature .
Normal Flow:	**1.0 Patron joins an existing Meeting for audio communication** 1. Patron logs-into the system through his credentials . 2. Patron joins on an on-going meeting . 3 . Patron then clicks on the audio button which toggles between on/off.
Alternative Flows:	**1.1 Patron first creates a Meeting for communication.** 1. Patron logs-into the system through his credentials . 2. Patron creates a new meeting which can be joined by other Users. 3. Partition starts the chat-button which creates a relation between users. 4. Return to step 3.
Exceptions:	1.0.E.1 Patron could not login into the web-application , for whatsoever reason . 1. In the above case , user should re-attempt to re- login .
Includes:	None
Priority:	High
Frequency of Use:	Once/Twice per user on average
Business Rules:	BR-20 , BR-21 , BR-33 , BR-38
Special Requirements:	1. User is logged into the server . 2. Users should join/create a meeting in order to communicate via audio , thereafter.
Assumptions:	1. Users must have an Audio console in their system . 2. Also , Users would have to click on the audio-enable button to get access to communication via audio .
Notes and Issues:	Expect a high frequency of executing this use case within the first 2 weeks after the system is released.

Use Case ID:	5		
Use Case Name:	Video Call		
Created By:	Aman Seervi	Last Updated By:	Aman Seervi
Date Created:	October 12,2023	Date Last Updated:	October
Actors:	Patron		
Description:	A patron accesses the MeetNow website and then starts a video call using the appropriate user interface provided. Other friends can then join the call.		
Preconditions :	1. Patron needs to be logged into his/her account 2. Any person who wants to join the video call needs to be an existing friend. If not then an invitation needs to be sent to include them as a friend.		
Postconditions:	1. Patron ends the call once the meeting is complete 2. Patron logs out after the work is done		
Normal Flow :	1. Patron texts his/her friends if they would like to join a video call 2. If they agree, patron starts a video call 3. The friends of patron can then join the video call 4. They can then communicate on the platform 5. A person can leave the call if they want to 6. Patron can end the call when everyone has left or when he/she feels fit 7. Patron logs out from the platform once competed with his/her work		
Alternative Flows:	1. Patron directly starts a video call 2. Other friends of the patron see that he/she is online and currently engaged in a video call. 3. They can then join the video call if they want to 4. Everyone can communicate using the platform 5. Anyone can leave the call when they feel fit 6. Patron logs out once done with his/her task		
Exceptions:	1. Patron is not logged in 2. Patron is not connected to the internet 3. Patron has a weak internet connection 4. Firewall blocks the WebRTC traffic		
Includes:	Adding a friend		
Priority:	High		
Frequency of Use:	Multiple times per user in a day		
Business Rules:	BR-0		
Special Requirements:	1. User should be logged into his/her account 2. User should a friend of the person he/she intends to chat with		
Assumptions:	1. Users have a webcam in thor system 2. Users have given all the permissions to the application		
Notes and Issues:	Expect high frequency throughout the lifetime of the application		