Iteration 4 - Students Requirements Proposal

Team ID/Name: S20-ESN-RW1

Student Name: Jean Baptiste Tuyizere

TA Name: Peter Yefi

Within your team, each use case needs to be different and independent:
There should be **NO overlaps or dependencies between use cases**.

Step 0: Theme Identification

Team Application Theme	**Approved**
Search to Rescue	Y

Student N	**Proposal**	**Approved**
Jean Baptiste Tuyizere	**Step 1: High-Level Requirement** User Story Name: Sharing video with the community Use Case Name: Share video chat Use Case Brief Description: The use case is for helping citizens share visual context of the situation they are in with the community by sharing a video	Y/N
	Faculty/TA Comments, including value points (min required is 3): 4	
	Step 2: Clickable UI Mockups Link:https://www.figma.com/proto/2gc56Up2DF1lcHCtEoChvH/S20-ESN-RW1?node-id=13%3A5&scaling=scale-down	Y/N
	Faculty/TA Comments:	
	Step 3: Elaboration - Use Case Specification See below (min 6 steps in basic flows and 6 alternative flows or rules)	Y/N
	Faculty/TA Comments:	
	Step 4: Use-Case Analysis Model (OOA) See below	Y/N
	Faculty/TA Comments:	

User Story Specification

User Story

As a **Citizen**, I want to **share a video with a caption** as a chat so that the community can see what the situation I am in is like and the place where I am

Acceptance Criteria

Use Case Specification

Participating Actors

Citizen

Brief Description

A citizen can capture a video to show the situation they are in and share it with the community

Assumption

The citizen has a device to capture video and can get that video onto the device by which they chat on ESN

Flow of Events

Basic Flow

1. The use case starts when a citizen clicks a button to select a video
2. The system opens a file explorer

3. The Citizen selects a video
4. The system loads the video and presents to the user a button to send the video

5. The Citizen selects to send the video
6. The system stores the video and displays it in chats with a button to play it.
7. The citizen elects to play the video
8. The system pops up a video player with an a button to close the video player and the video player automatically starts playing the video
9. The citizen elects to close the video player by clicking the closing button
10. The system closes the video player and returns the user to the chats screen.

The use case ends.

Alternative Flows

- A1. Unexpected file type: In step 4, if the citizen selected a file that is not a video, the system displays “Unexpected video file type.” message. The use case ends.

- A2. Can’t open file explorer: In step 2, if the system can not open file explorer, the citizen can not select video file. The use case ends.
- A3. Too large video file. In step 6, if the video is larger than 30mb, the system does save the video file and displays “Video should not be larger than 30 mb” message. The use case ends.
- A4.
- A5.
- A6.

Use Case Analysis Model (OOA)

Entity classes

Chat
ChatController
ChatFormUI
ChatRepository

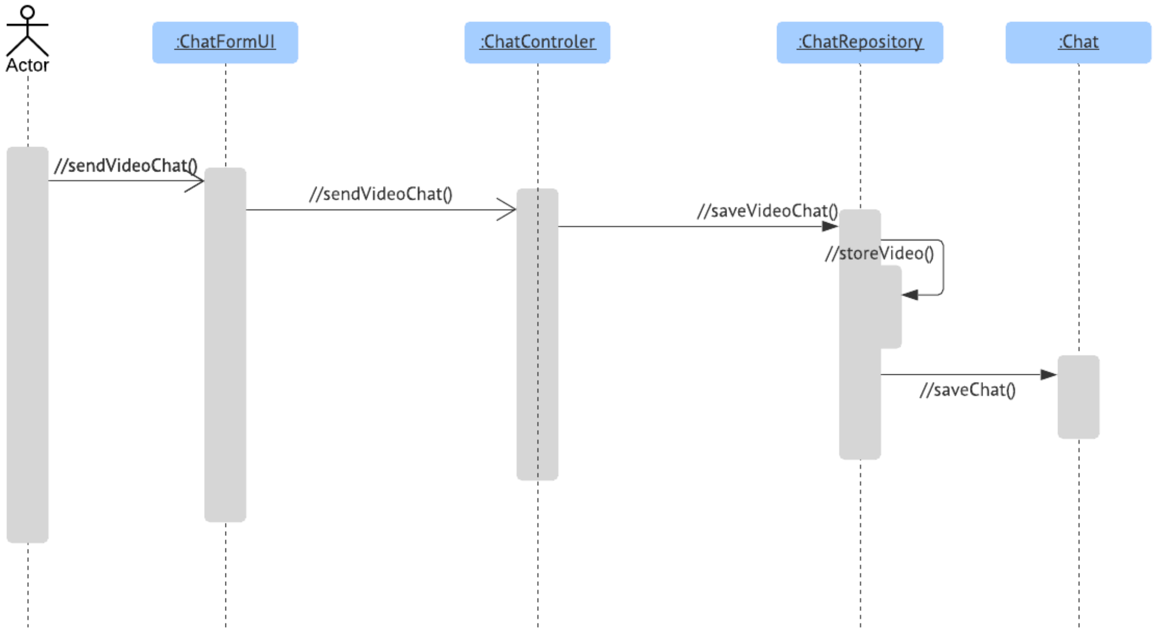
Boundary classes

ChatFormUI
ChatRepository

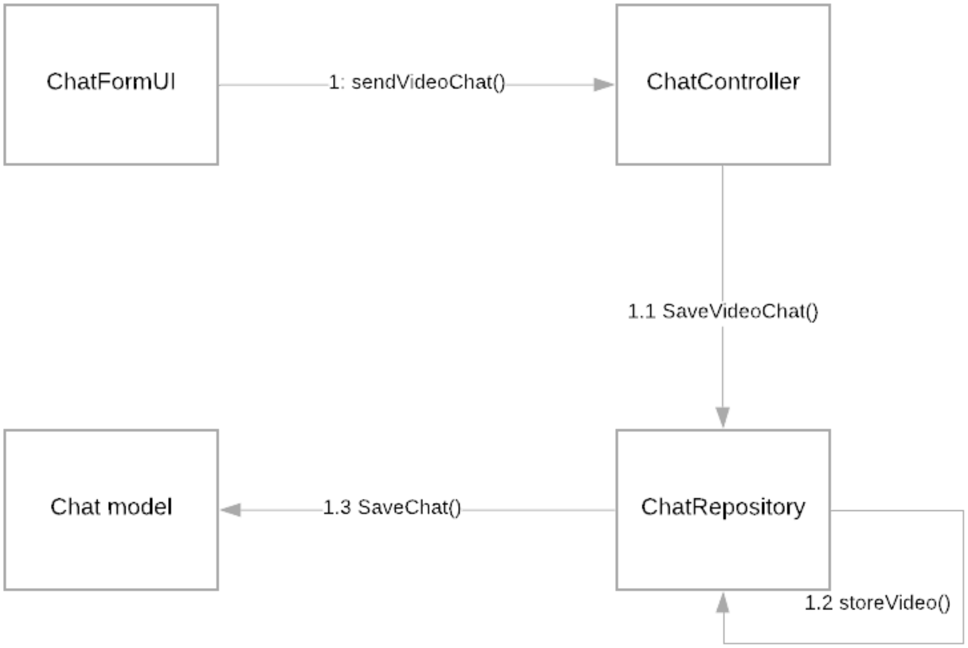
Control class

ChatController

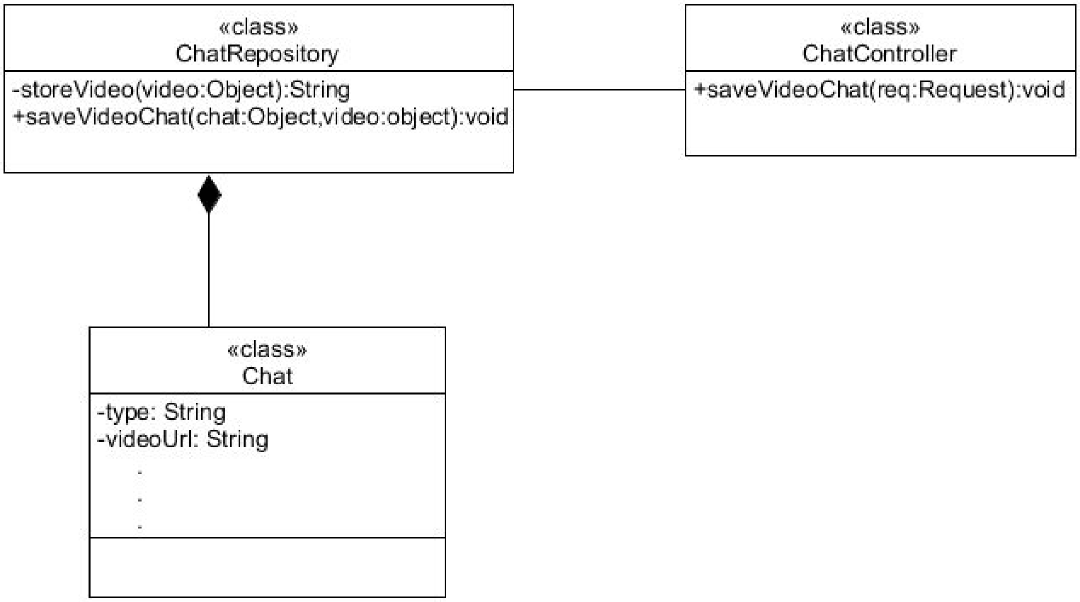
Sequence diagram modeling



Communication Diagram



Class diagram / VOPC


«class»
ChatRepository
-storeVideo(video:Object):String
+saveVideoChat(chat:Object,video:object):void
«class»
ChatController
+saveVideoChat(req:Request):void
«class»
Chat
-type: String
-videoUrl: String
.
.
.

Mapping Between Analysis Classes and Code:

Analysis Classes	**Implementation Elements** (e.g. modules, files, components, databases)
Chat	models/Chat.js
ChatController	controllers/ChatController.js
ChatRepository	repositories/ChatRepository
ChatFormUI	ui/src/components/ChatRoom.vue