

Iteration 4 - Students Requirements Proposal

Team ID/Name: ___Team 1_____

Student Name: ___Bukombi brian_____

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.**

TA: CLONE THIS DOCUMENT FOR EACH STUDENT IN THE [STUDENT DELIVERABLES FOLDER](#) UNDER THE TEAM NAME DIRECTORY).

GIVE EACH TEAM READ-ACCESS TO THEIR DIRECTORY AND GIVE EACH TEAM MEMBER EDIT-ACCESS TO THEIR FILE

Check the following pages for the template for each use case and student...

Step 0: Theme Identification

As a team, agree on a main theme for your application. Your theme must be relevant to emergency situations due to natural disasters (e.g. Provide Medical Assistance). Make sure that all your proposed individual requirements relate to the team theme.

Remember that during the final project presentation, your team will be asked to demonstrate the implementation of those requirements as an integrated scenario (versus a bunch of unrelated features). Identifying the scenario during iteration 4 might help your team identify an interesting theme and the necessary requirements to support the theme.

Team Application Theme	Approved
Search and rescue	Y

Student N	Proposal	Approved
Name ???	Step 1: High-Level Requirement User Story Name: Uploading an image to the community Use Case Name: share an image Use Case Brief Description: A citizen should be able to upload a picture of an emergency situation. This will help give more detail to how critical an emergency situation is. Functionality should add value to Citizens (or other actors) in case of emergency. Implementation will demonstrate your coding ability, exercising both frontend & backend, supporting a REST API, and hitting the database.	Y
	Faculty/TA Comments, including value points (min required is 3):	
	Step 2: Clickable UI Mockups Link: https://balsamiq.cloud/sfjp81g/pisfnty	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

A citizen shares a photo of an emergency situation to help the community get a better understanding of a particular emergency situation. This could be a photo of a natural disaster like flooding

Acceptance Criteria

Use Case Specification

Participating Actors

- citizen
- administrator

Brief Description

This feature allows a citizen to upload a picture taken either with a smartphone or another device like a camera

...

Assumption

A citizen is using a smartphone to take the picture

Flow of Events

Basic Flow

1. The use case starts when the user clicks a button to upload an image
 - a. The system will open the file explorer in a dialogbox
2. The user selects the image to upload.
3. The image is uploaded
4. The user enters a message and selects to save the image together with the message
5. The image is uploaded to the server and message committed to the database

Alternative Flows

- Send a message without an image. The user should not be allowed to send an image without a message since the images are meant to augment the messages
- Check file size before uploading. If the file is too large, display a pop up message warning the user
- Check that the user is uploading an image. If not, display a pop up notifying the user

Use Case Analysis Model (OOA)

Entity classes: ...

Chat

ChatRepository

ChatFormUI

ChatController

Boundary classes: ...

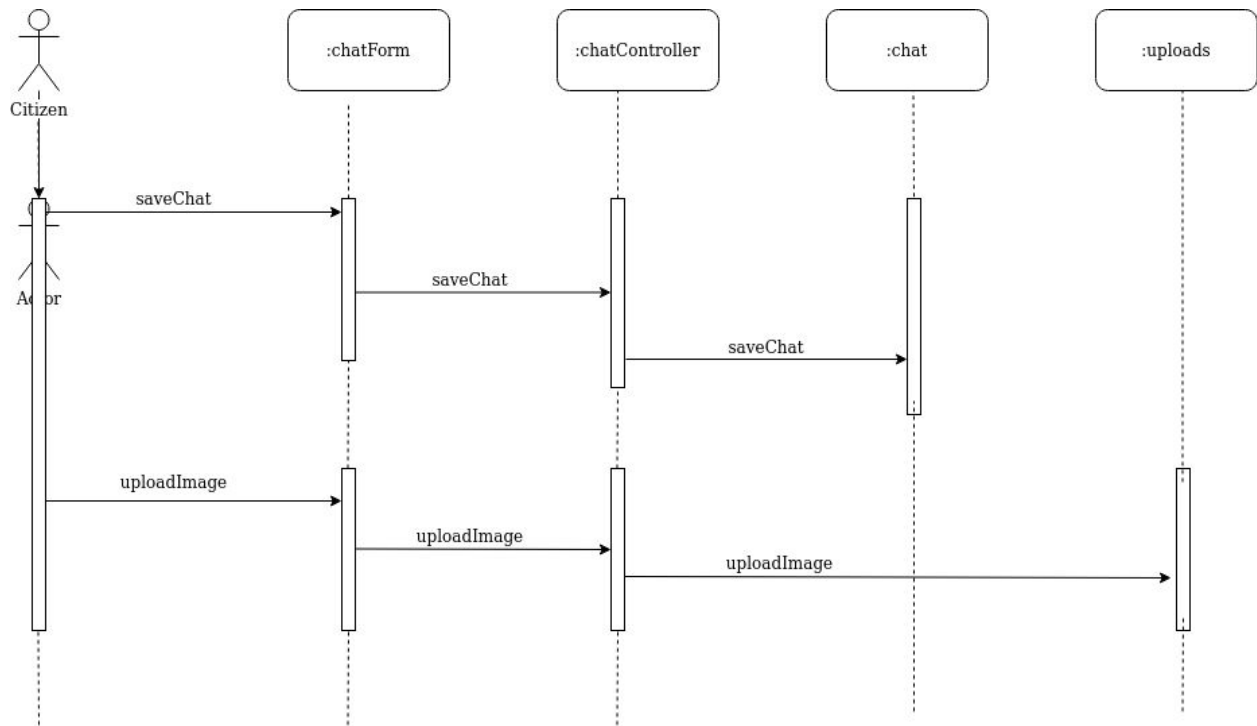
ChatRepository

ChatFormUI

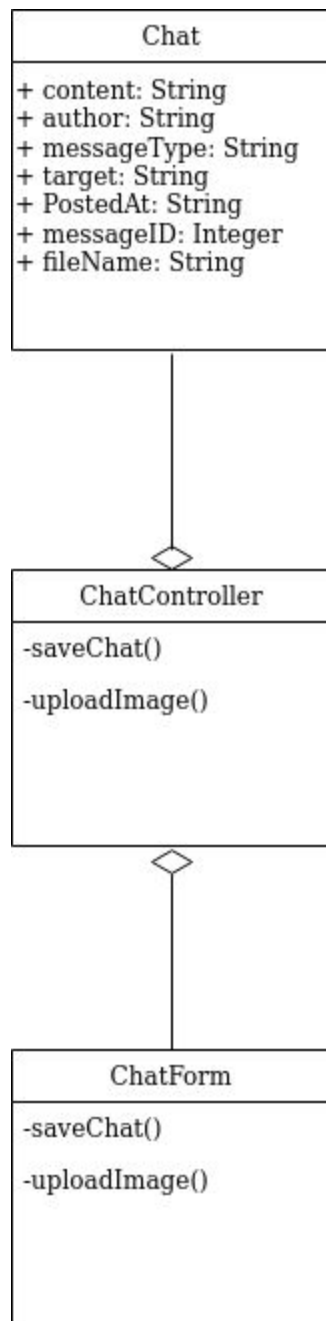
Control classes: ...

ChatController

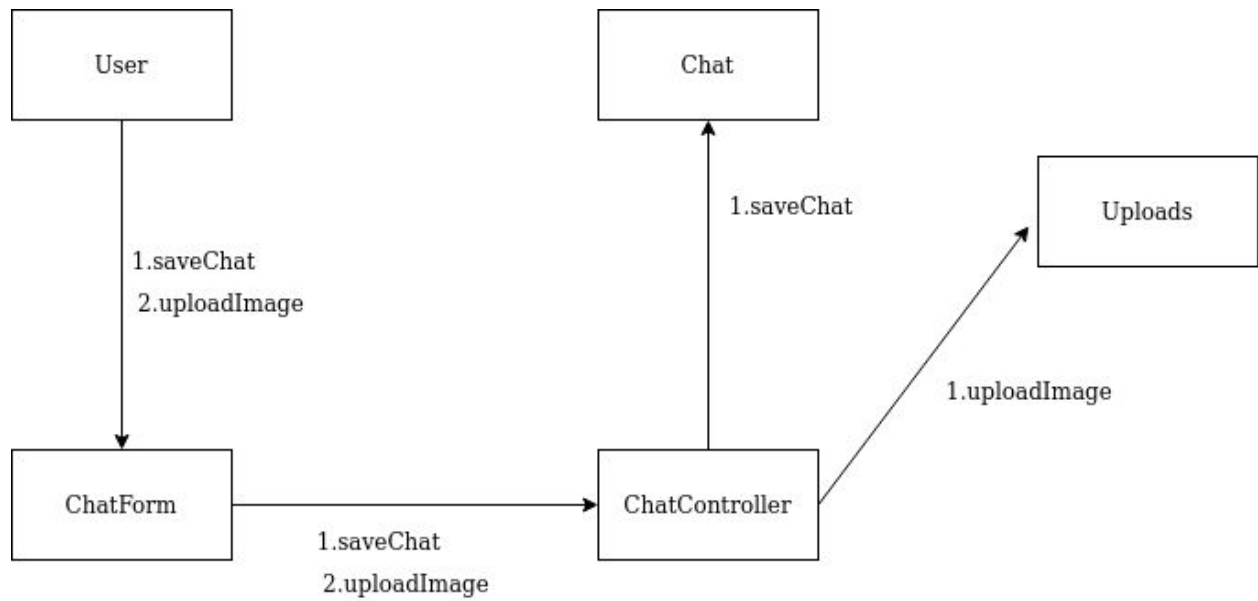
Sequence diagram modeling the use-case behavior: ...



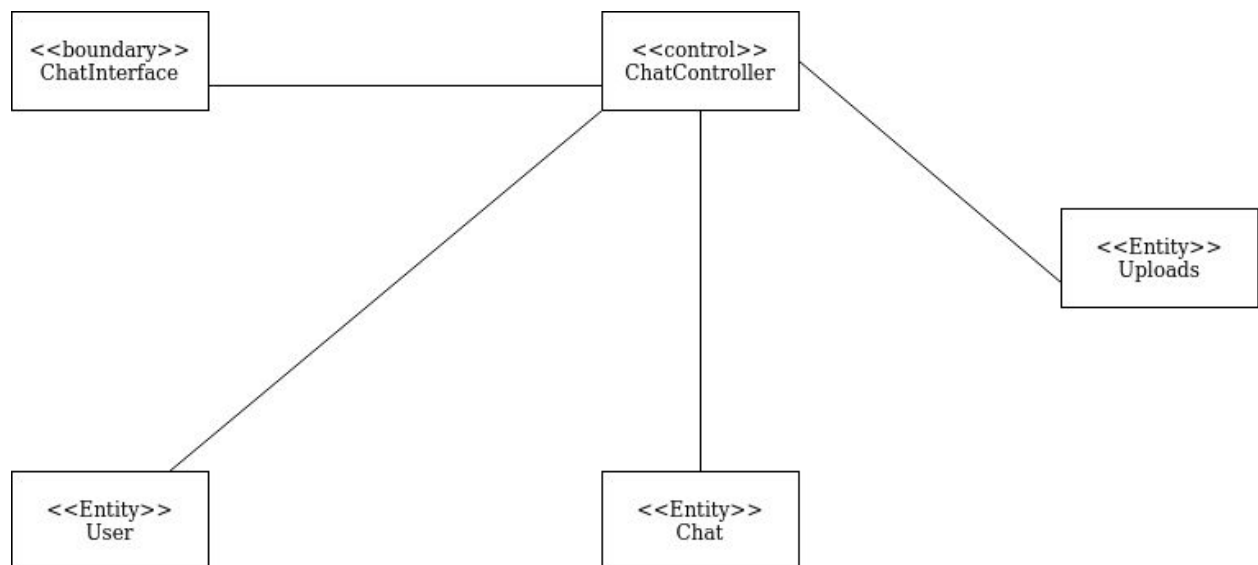
Class diagram (based on the sequence diagram) modeling the use-case structure: ...



Communication diagram



VOIP



Mapping Between Analysis Classes and Code:

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

Chat	models/Chat.js

M