MANİSA
CELAL BAYAR
UNIVERSITY

CSE 3105/ CSE 3137

OBJECT ORIENTED ANALYSIS AND DESIGN

FALL 2020

COURSE PROJECT: *Media Browser Application*

System Design Document

Group 27

ÖMER BENEK – 200315006

KARDELEN ÇETİN – 170316036

MUZAFFER ÖZEN – 170316044

AYSELİ EREM BATI – 170316006

DENİZ ALTUN – 170316023

HAKAN BEKDEMİR – 170316031

16 January 2021

İçindekiler

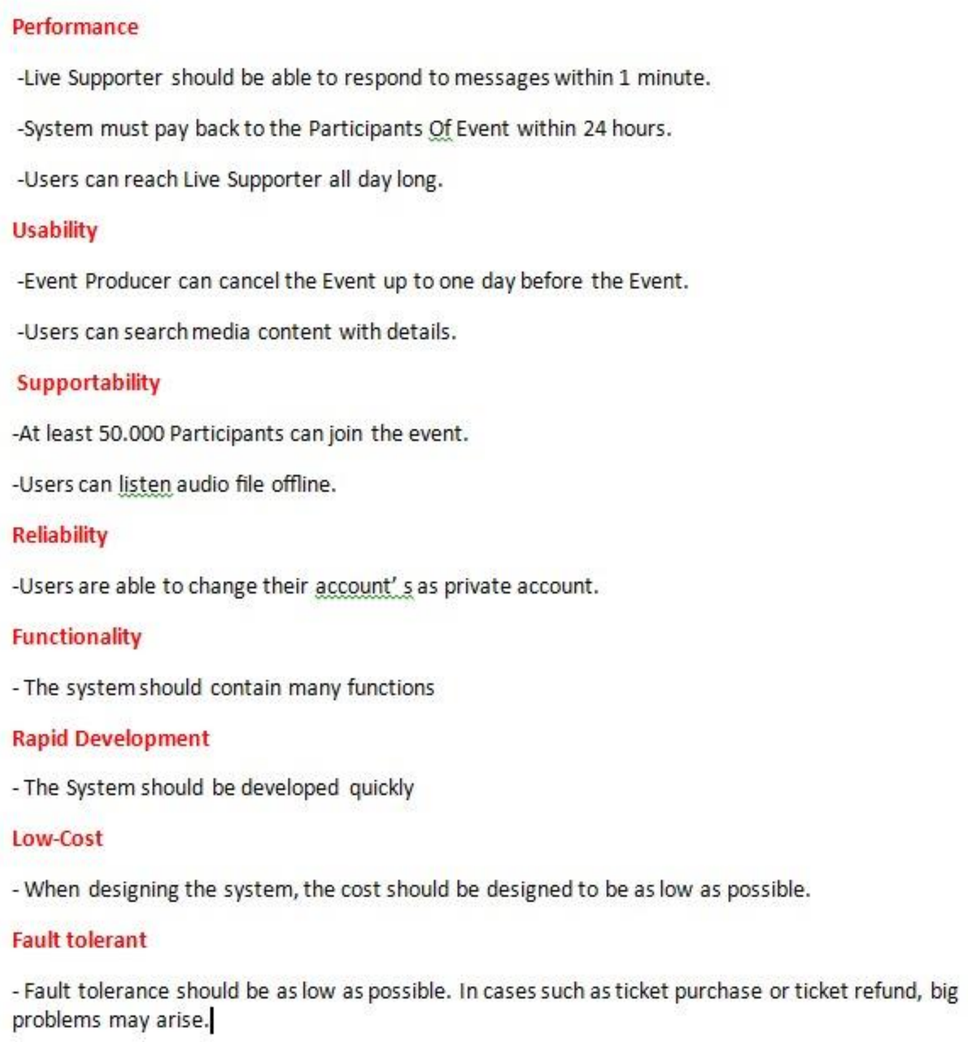
1 Introduction

The name of our application is M&E. Although the main purpose of this application is to view video / audio / images, in addition to this feature, we are creating in-app activities and interacting between users (following, geting likes, messaging). For this purpose, although we are in our homes due to covid-19, we have tried to socialize people with such events (e.g.,online concerts, live broadcasts, competitions).

1.1 Purpose of the System

The purpose of this system is to provide features such as organizing online events, viewing common media, creating and sharing playlists that other media browsers do not provide to the user.

1.2 Design goals



Performance

-Live Supporter should be able to respond to messages within 1 minute.

-System must pay back to the Participants Of Event within 24 hours.

-Users can reach Live Supporter all day long.

Usability

-Event Producer can cancel the Event up to one day before the Event.

-Users can search media content with details.

Supportability

-At least 50.000 Participants can join the event.

-Users can listen audio file offline.

Reliability

-Users are able to change their account' s as private account.

Functionality

- The system should contain many functions

Rapid Development

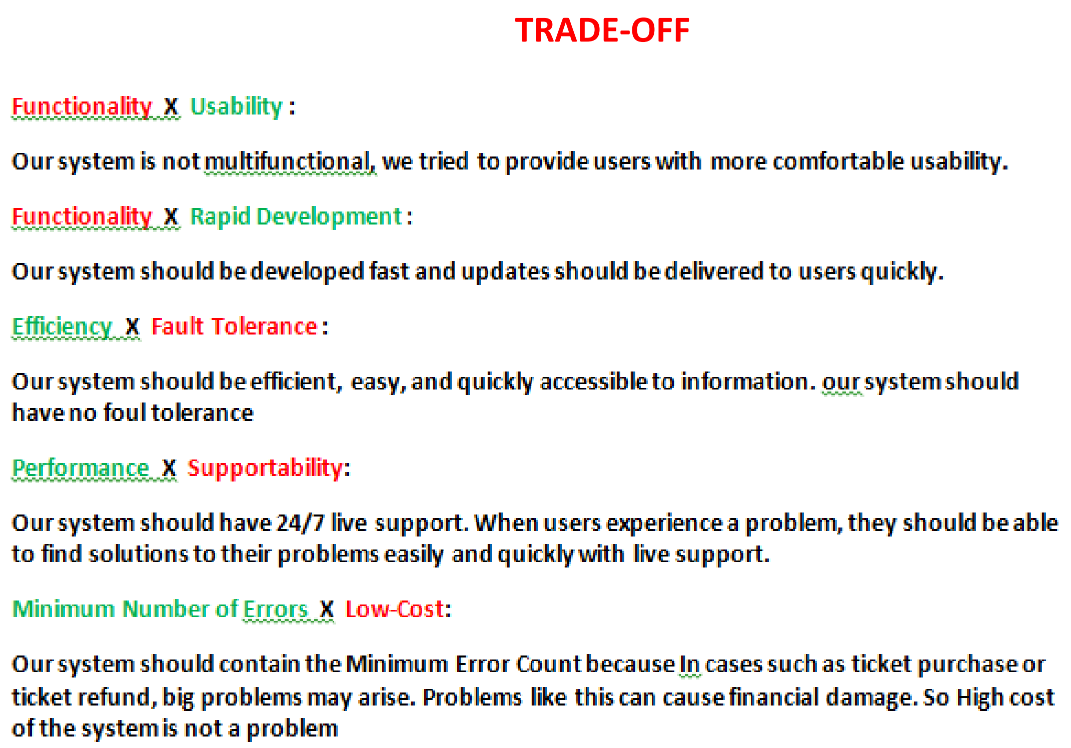
- The System should be developed quickly

Low-Cost

- When designing the system, the cost should be designed to be as low as possible.

Fault tolerant

- Fault tolerance should be as low as possible. In cases such as ticket purchase or ticket refund, big problems may arise.



TRADE-OFF

Functionality X Usability :

Our system is not multifunctional, we tried to provide users with more comfortable usability.

Functionality X Rapid Development :

Our system should be developed fast and updates should be delivered to users quickly.

Efficiency X Fault Tolerance :

Our system should be efficient, easy, and quickly accessible to information. our system should have no foul tolerance

Performance X Supportability:

Our system should have 24/7 live support. When users experience a problem, they should be able to find solutions to their problems easily and quickly with live support.

Minimum Number of Errors X Low-Cost:

Our system should contain the Minimum Error Count because In cases such as ticket purchase or ticket refund, big problems may arise. Problems like this can cause financial damage. So High cost of the system is not a problem

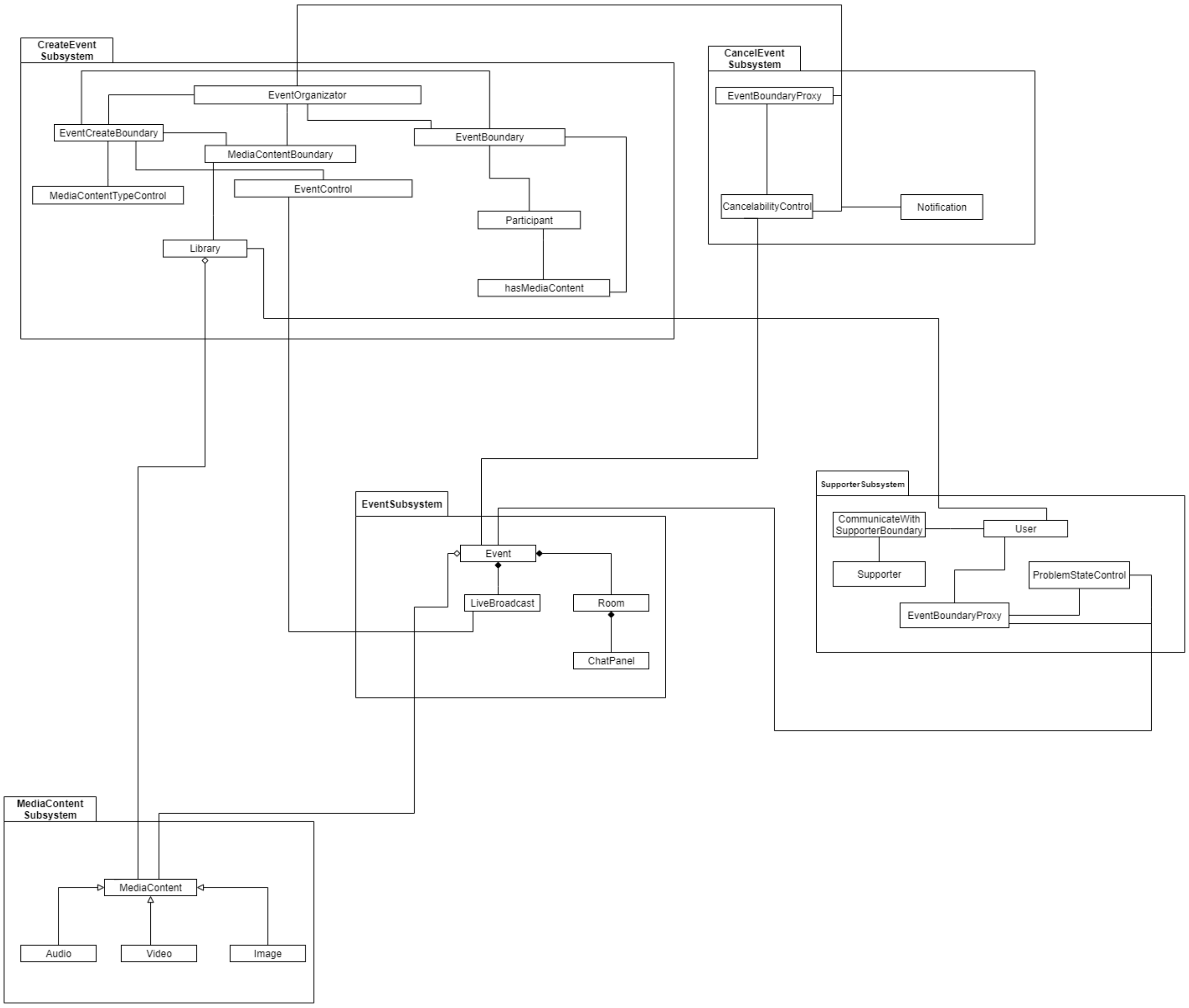
2 Current Software Architecture

While making this application, we decided to make such an application, inspired by the missing aspects of various platforms such as instagram, icloud, spotify and youtube. For example; Simultaneous performance of some activities is not supported in all of these applications. (The two friends cannot listen to the same song simultaneously.) To give another example; Most of these platforms do not organize an online event. Finally, other apps except Instagram do not prioritize account privacy.

3 Proposed Software Architecture

The feature that distinguishes the system we propose from other systems is that it can create events. For example, an artist can hold an online concert, a producer can hold an online script contest, or a teacher can lecture live. In addition, two or more people can simultaneously view media in our system. People will be able to stay in touch with the messaging feature that is not available in most media browsers. Finally, we aim to solve the problems of our users as soon as possible by adding a live support line to our system (via messaging), even if we see it on a few platforms.

3.1 Subsystem decomposition



CreateEventSubsystem: CreateEventSubsystem is responsible for the EventOrganizator to create events using other classes in the subsystem.It is also responsible for the connection of participant to the created event.

CancelEventSubsystem: CancelEventSubsystem is the subsystem responsible for canceling the Event that EventOrganizator created.It is also responsible for sending notifications to users after this Event is canceled.

EventSubsystem: EventSubsystem is responsible for the live broadcast of the event and the creation of a live chat panel to be used by the participants during the broadcast.

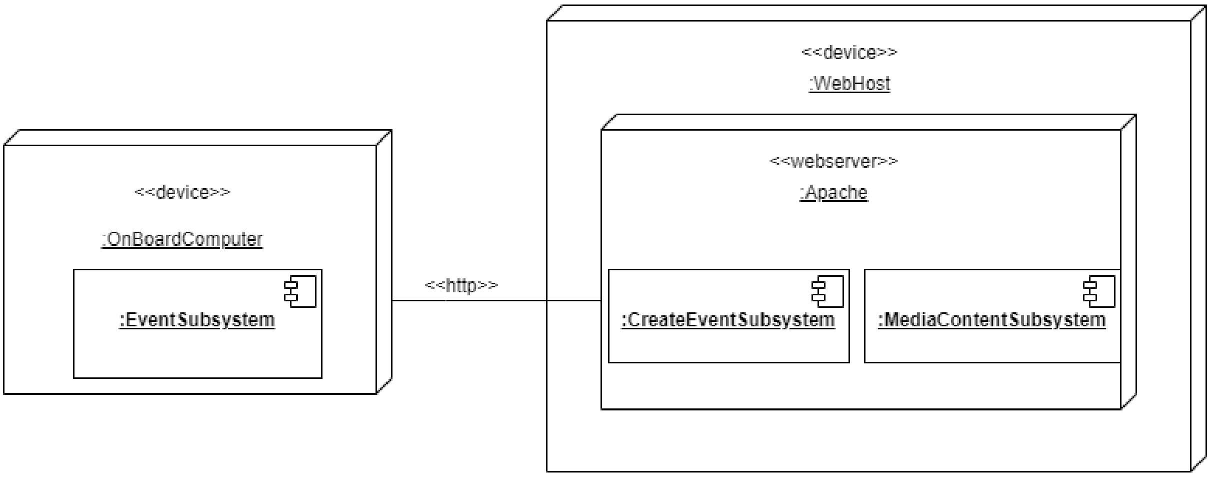
SupporterSubsystem : SupporterSubsystem is responsible for users contacting Live Supporter when they encounter a problem.

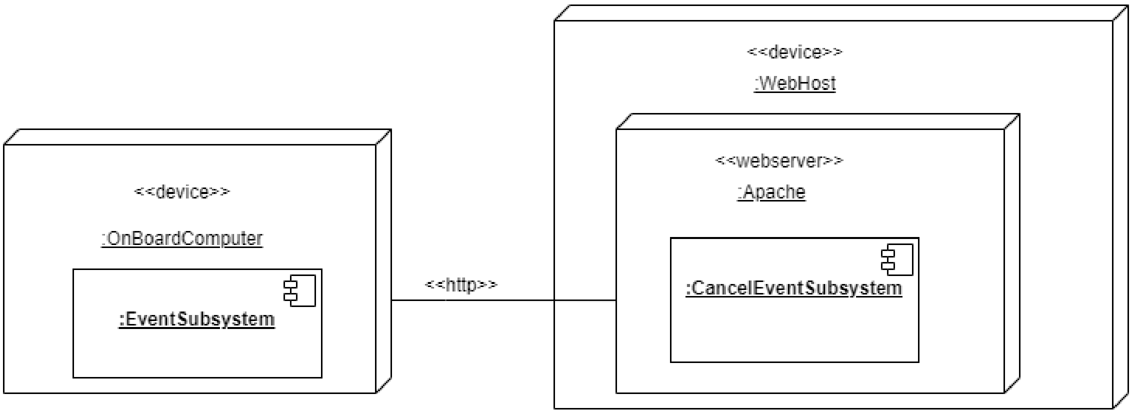
MediaContentSubsystem: Media content subsystem is associated with the library class. It is responsible for the content in the media content class.

Subsystem Desing Explanation: While designing the packages, we examined the structures used in general. The design in mind was the Open Layer structure, but we could not adapt the systems of our project to the Open Layer structure because the coupling is a bit high. For example, each subsystem has a relationship with the Event subsystem so we designed a design according to ourselves.

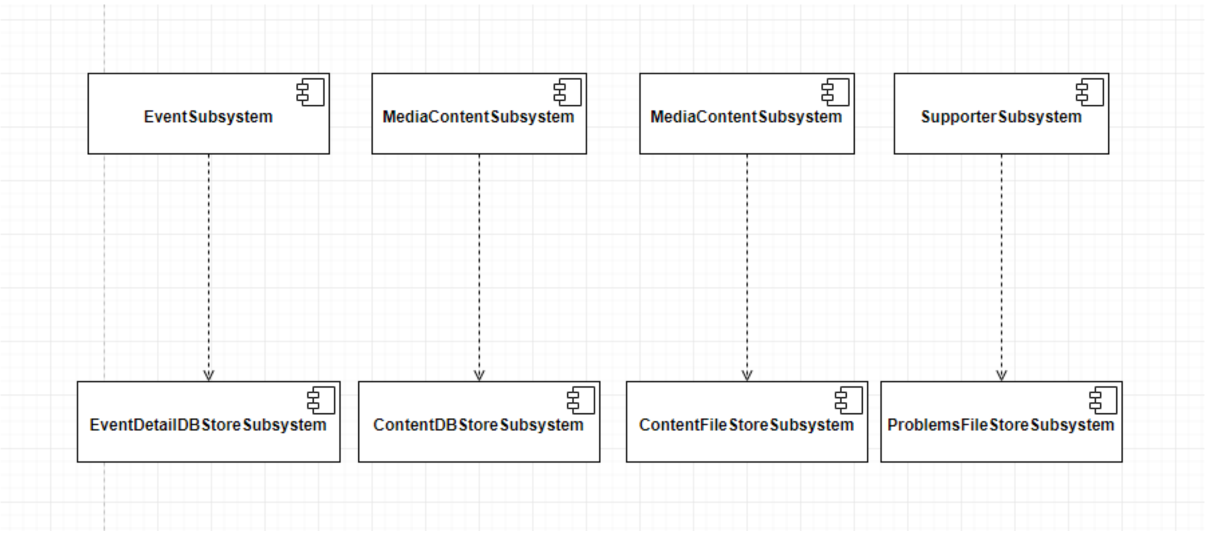
3.2 Hardware/software mapping

:





3.3 Persistent data management



EventDetailDBStoreSubsytem :

The EventDetailDBStoreSubsystem is responsible for storing event details which are:

When will be the event , how will it be published(live broadcast or media content) in the database for EventSubsystem..Because this system supports multiple participants to reach the event.

ContentFileStoreSubsystem :

The ContentFileStoreSubsystem is responsible for storing media contents in files on the onboard computer.Because our users can reach their downloaded contents when they are offline..

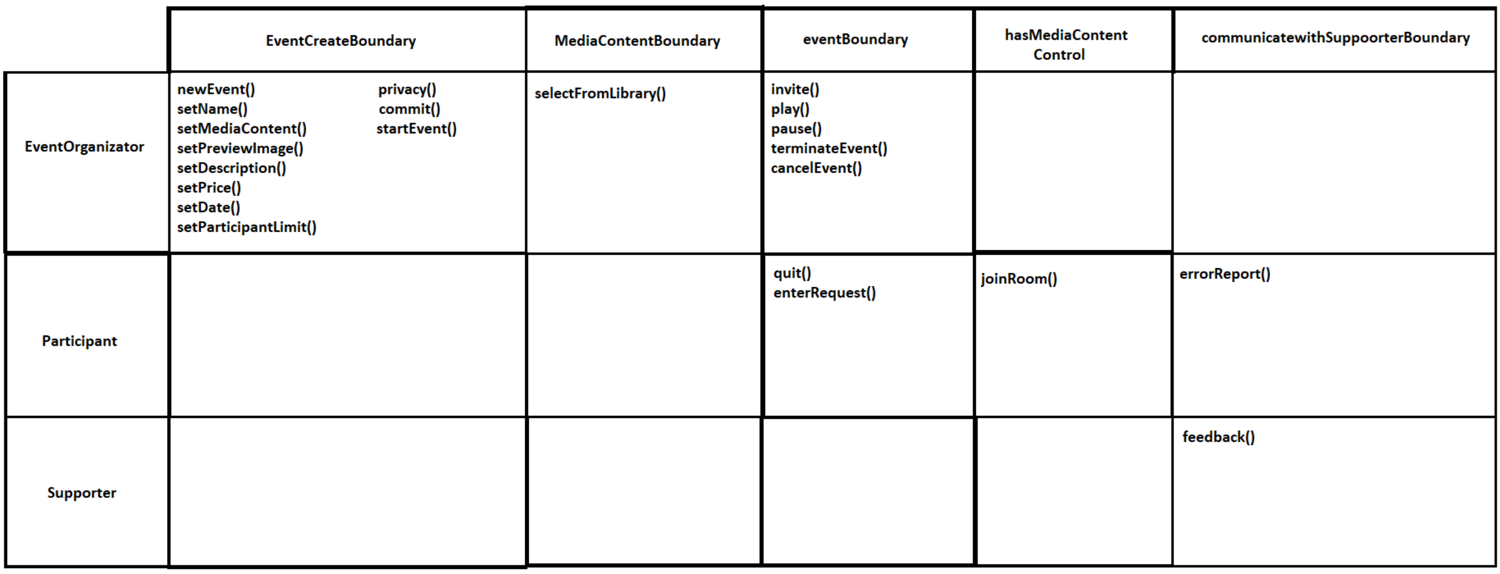
ContentDBStoreSubsystem:

The ContentDBStoreSubsystem is responsible for storing media contents in files on the onboard computer.Because our users can reach contents when they are online.

ProblemsFileStoreSubsystem :

The ContentFileStoreSubsystem is responsible for storing problems in files on the onboard computer for Supporter. Because only the Supporter must reach problems.

3.4 Access control and security



	EventCreateBoundary	MediaContentBoundary	eventBoundary	hasMediaContent Control	communicatewithSuppoorterBoundary

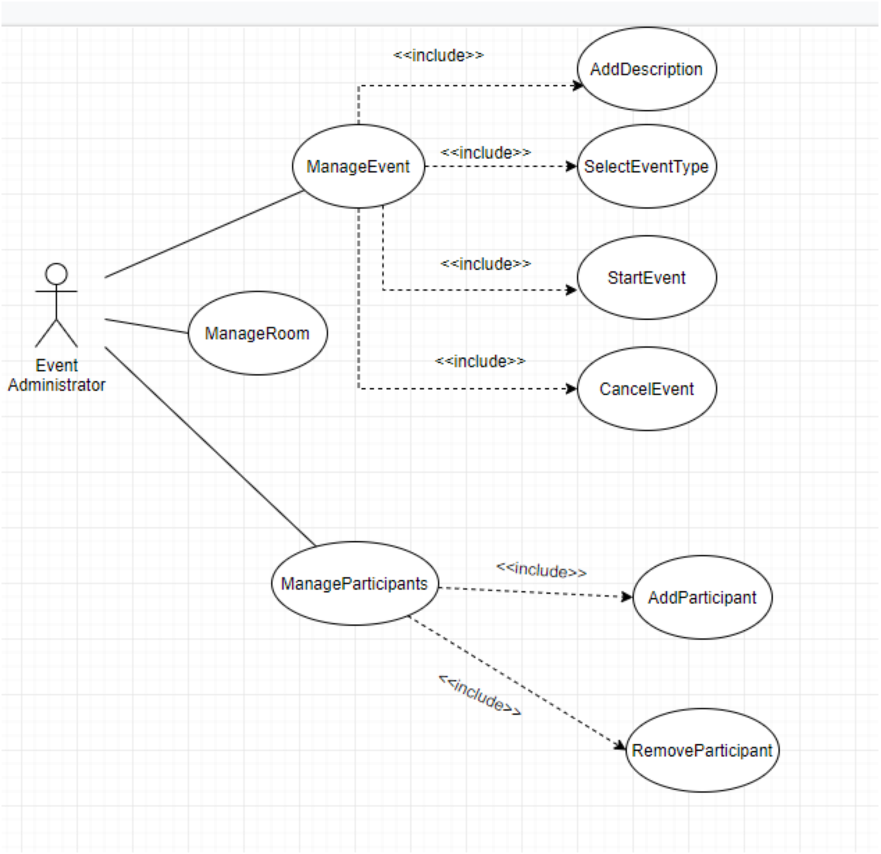
EventOrganizator	newEvent()
	setName()
	setMediaContent()
	setPreviewImage()
	setDescription()
	setPrice()
	setDate()
	setParticipantLimit()
	privacy()
	commit()
	startEvent()
		selectFromLibrary()	invite()
			play()
			pause()
			terminateEvent()
			cancelEvent()		

Participant			quit()
			enterRequest()
				joinRoom()	errorReport()

Supporter					feedback()

SECURITY: Each user registered in this system can only access their own account. Access to accounts is provided by entering the username and password set by the user.

3.5 Boundary conditions



ManageRoom:

The Event Administrator decides the content to be played in the room. He/She can manage participants, content and chat room in the room with this function.

ManageParticipants:

The Event Administrator can manage the participants with this function.

AddParticipant:

The Event Administrator can add the participants who request to participate in the event according to their own criteria and send an invitation.

RemoveParticipant:

The Event Administrator can use this function to remove a participant who is participating in the event.

ManageEvent:

The Event Administrator can add description, select the starting type, start and end the event by using this function.

AddDescription:

The Event Administrator can add information about the event (date, type, subject, privacy) by using this function.

SelectEventType:

The Event Administrator decides what the content of the event will be by using this function.(Live broadcast or video / audio / image)

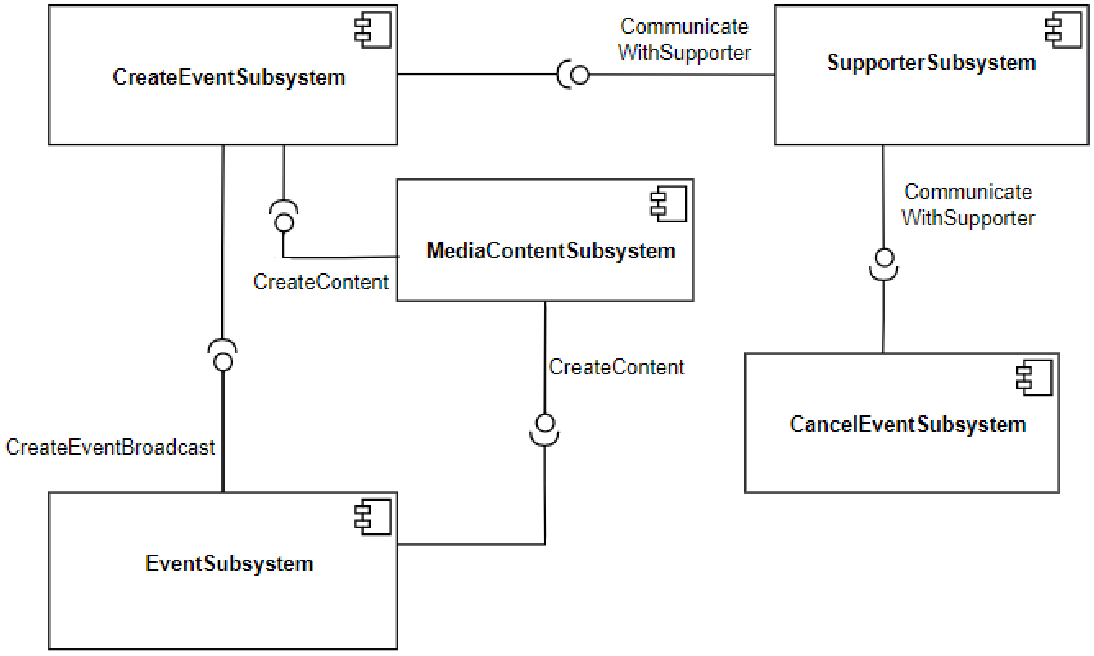
StartEvent:

The Event Administrator can start an event with this function.

CancelEvent:

The Event Administrator can cancel an event with this function.

4 Subsystem Services



5 References

We referenced from Requirements Analysis Documents .