FAKULTA INFORMAČNÍCH TECHNOLOGIÍ ČVUT V PRAZE

Documentation

AUXBox_1st_iteration

Documentation of a project for the purpose of the course BIE-SI1.

Authors: Aykut, Basel, Egemen, Lucas, Mark, Wally

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ ČVUT V PRAZE

Contents

1. Business process Model

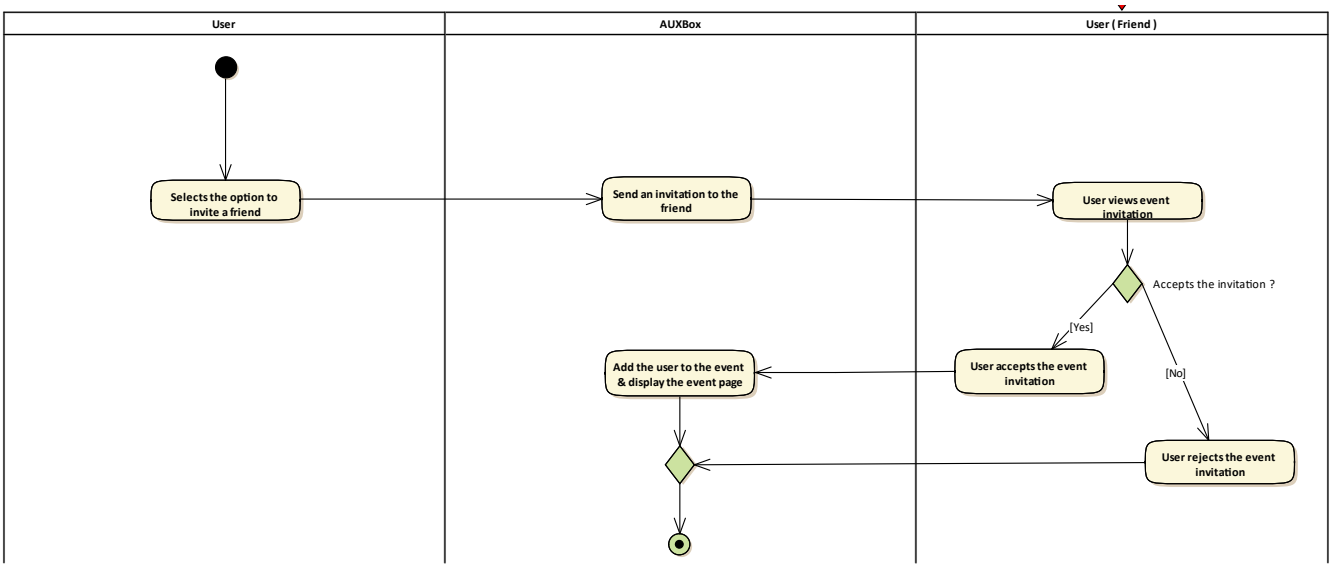


Figure 1 - User Invitation

Business process model outlines the User & DJ interaction when a User is posting a song request in an event

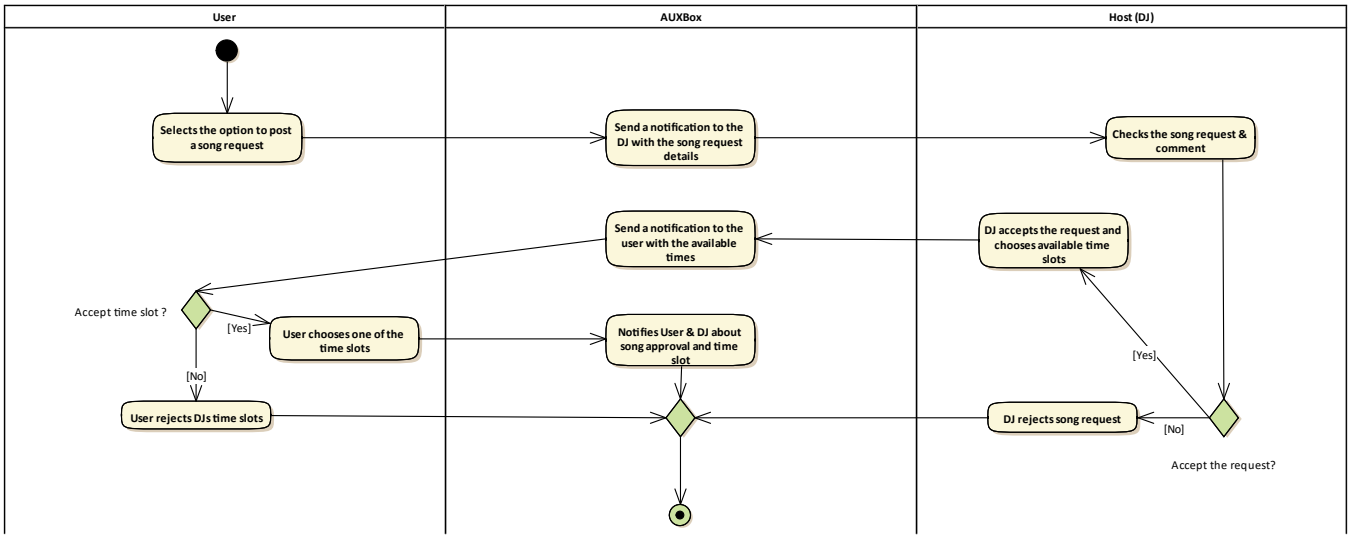


Figure 2 - User Requests

Business process model outlines the User invitation option that allows a User to invite a friend to an event

2. Domain Model

2.1 Class Diagram

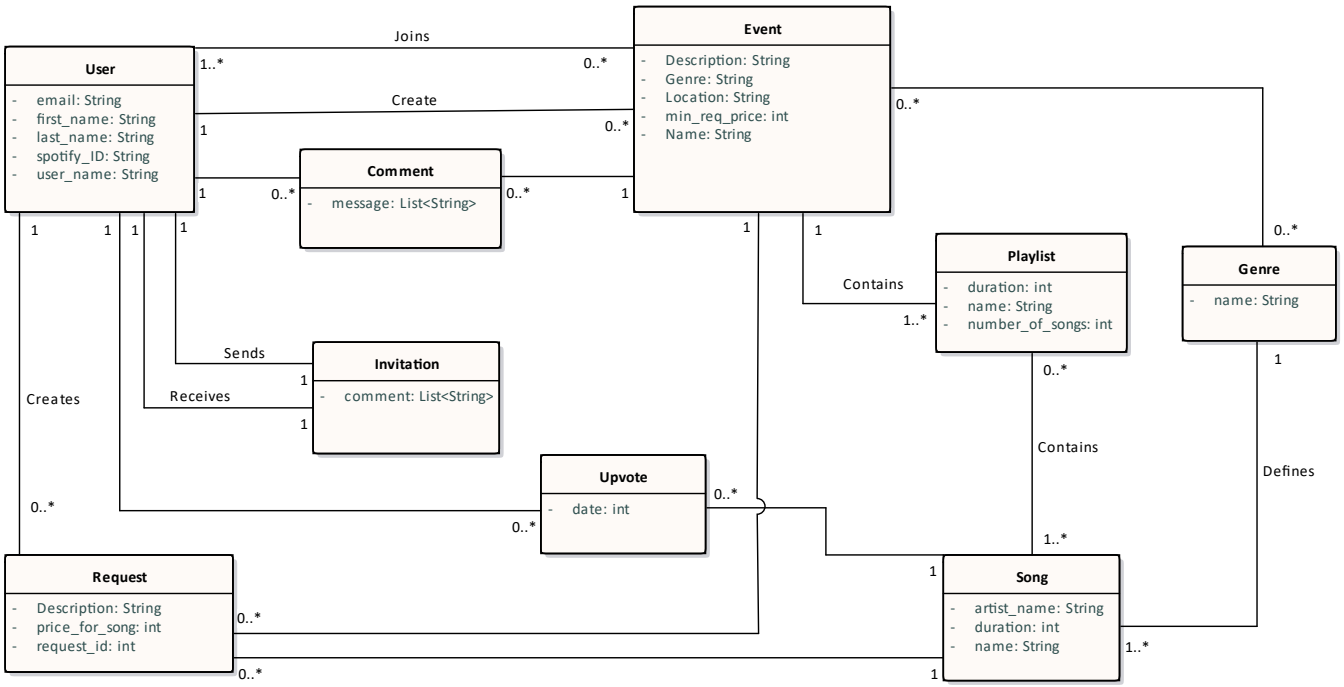


Figure 3 - Class diagram

2.1.1 Comment

Text that the user leaves at the event page

Attribute title	Description
message	

2.1.2 Event

Class that represents a single event.

Attribute title	Description
Description	

Attribute title	Description
Genre	
Location	
min_req_price	
Name	

2.1.3 Genre

Specific type of a song given in an event.

Attribute title	Description
name	

2.1.4 Invitation

A invite created by an user top invite another user for an event.

Attribute title	Description
comment	

2.1.5 Playlist

List of songs to be played in the event

Attribute title	Description
duration	
name	
number_of_songs	

2.1.6 Request

A user created request for a song in a given event

Attribute title	Description
Description	
price_for_song	
request_id	

2.1.7 Upvote

Representation of a user "liking" a given song.

Attribute title	Description

Attribute title	Description
date	

2.1.8 User

Class used to represent all users. Hosts are created as a inherited class.

Attribute title	Description
email	
first_name	
last_name	
spotify_ID	
user_name	

2.2 State Diagrams

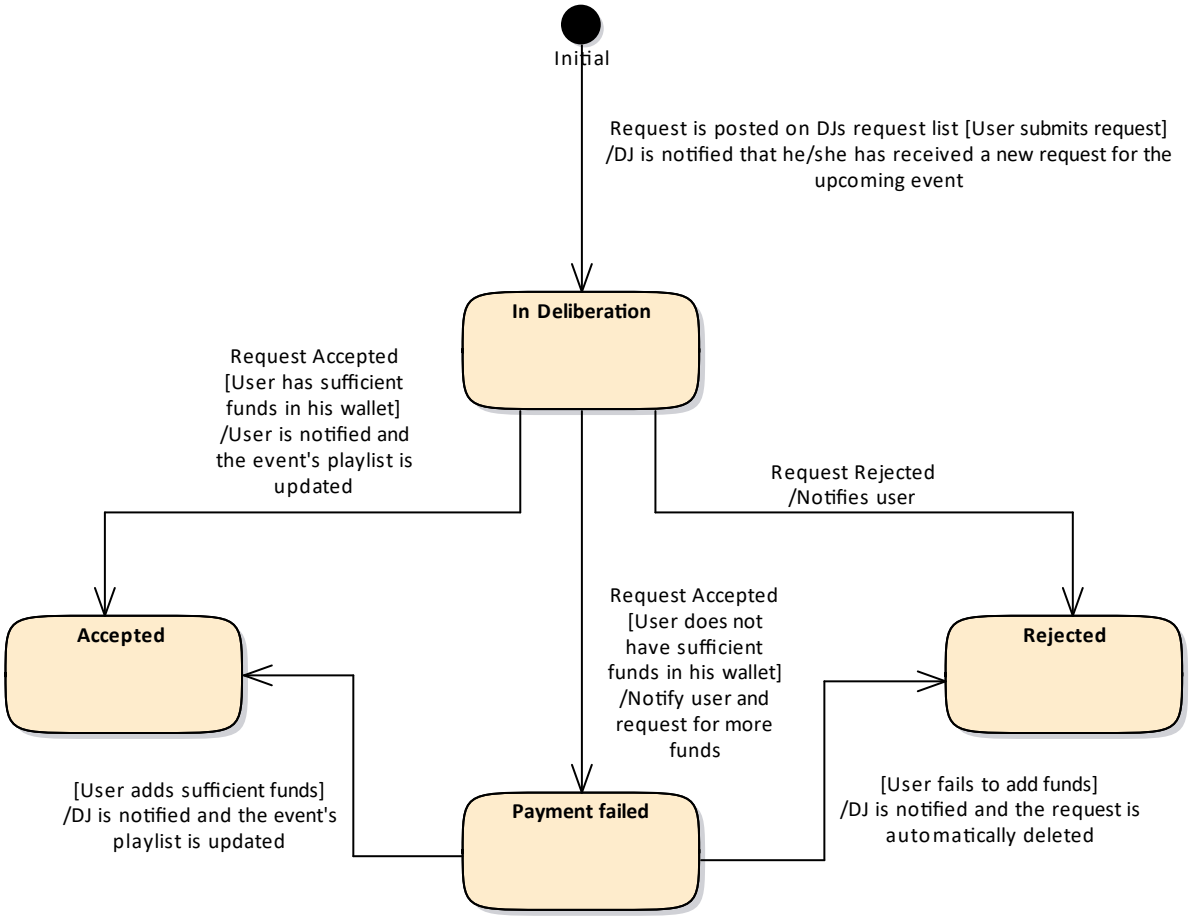


Figure 4 - Invitations

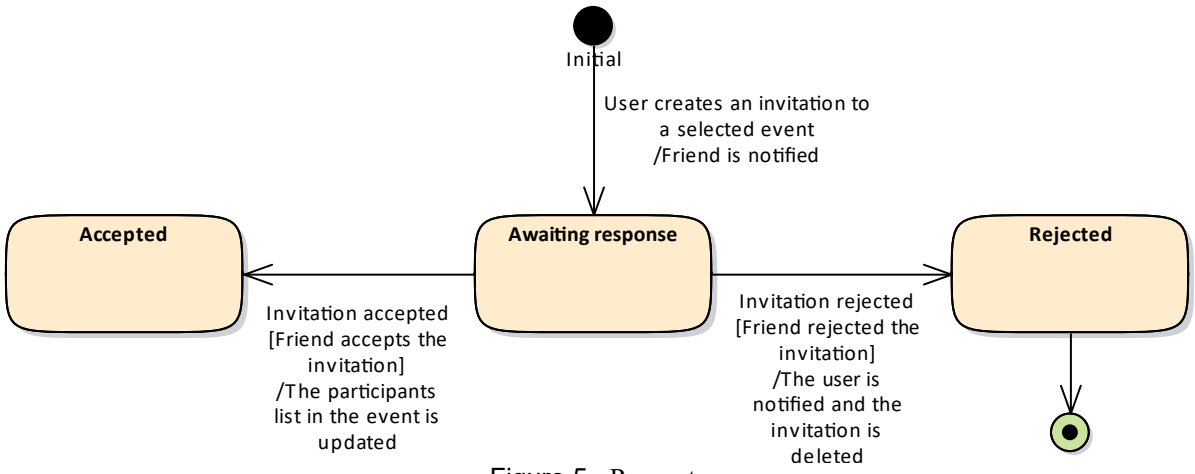


Figure 5 - Requests

2.2.1 Accepted

If the user accepted the request, he is then added to the event participants list.

2.2.2 Accepted

DJ accepted the request and added the given song to the event's list. The system saves the record of the request acceptance.

2.2.3 Awaiting response

After the user sends the invitation, the system awaits the other users response.

2.2.4 In Deliberation

State where the request is being considered by the DJ.

2.2.5 Payment failed

Temporary state where, after the user's account was unsuccessful to provide the payment for the request, the user is notified of the problem and is given some time to provide the funds for the payment.

2.2.6 Rejected

DJ rejected the request, so the user is notified and the rejection is saved by the system.

2.2.7 Rejected

If the user rejected the invitation, the invitation is deleted.

3. Requirements

Figure 6 - Functional requirements

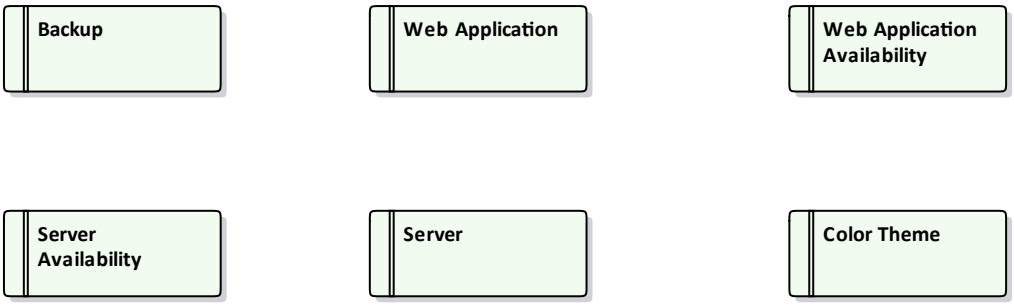
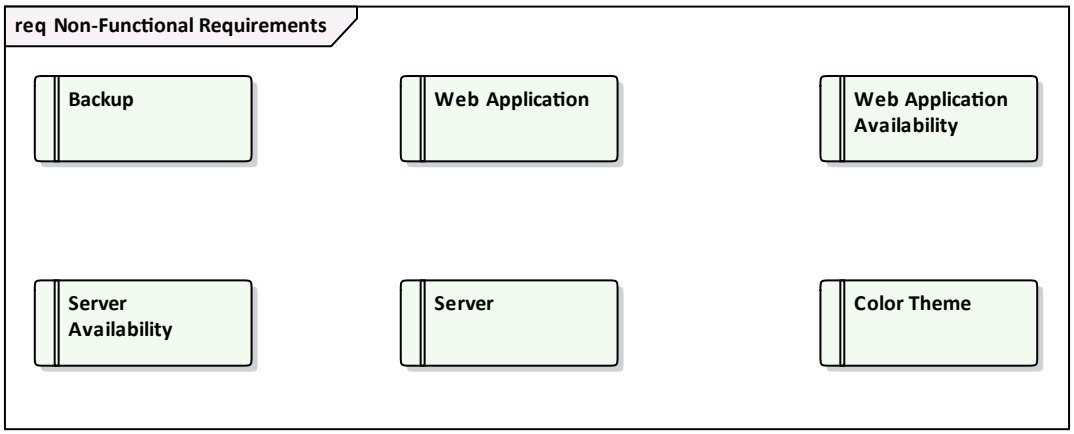


Figure 7 - Non-Functional Requirements



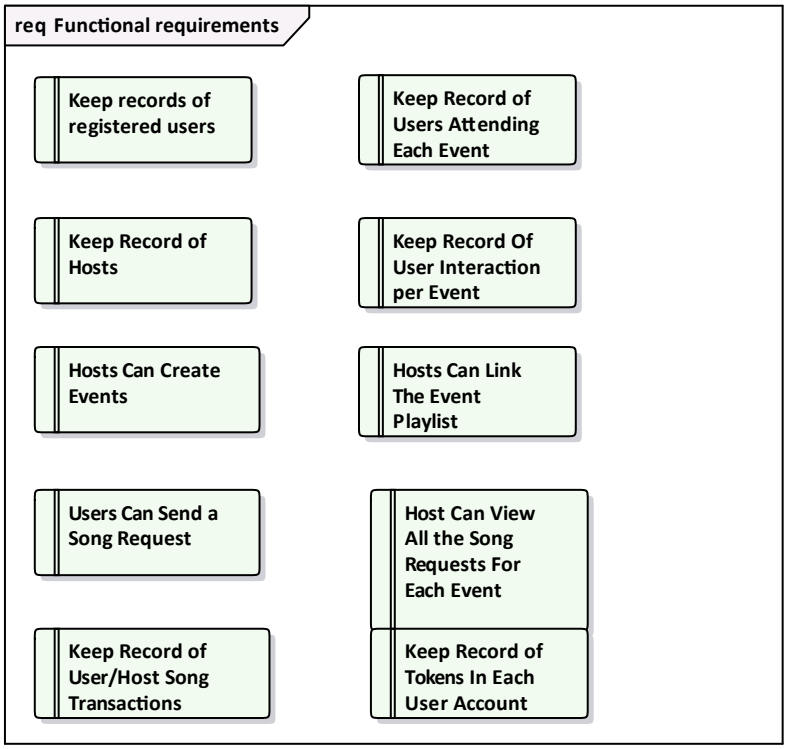


Figure 8 - Requirements

3.1 Backup

* There needs to be a daily backup of the data stored in a separate database on FIT servers. The backup is to be executed at 6 am where the website generates the least amount of traffic.

3.2 Color Theme

* The application interface will be dark themed with themes of sea-foam green and grey.

3.3 Event creation by hosts

Each host has the option to create an event on a certain date and time, where they can specify the location and set restrictions for the genres being played. Hosts mention a short description of the event. Host has the option to limit the number of up votes per user. Host sets the minimum price for a song request from the user.

3.4 Host Can View All the Song Requests For Each Event

* Host has the option to accept a song request, by responding with possible time periods where the song can be played. The user then can choose a time slot or decline the request within 48 hours of Host's response. If the user accepts the time slot, the Host needs to provide proof by recording a video of the song being played and using Shazam API we can verify if the song was played or not. If the song was not played, the user gets his money back.

3.5 Hosts Can Create Events

* Each user has the option to host an event on a certain date and time, where they can specify the location and set restrictions for the genres being played. Hosts mention a short description of the event. Hosts have the option to limit the number of upvotes per user. Hosts set the minimum token price for a song request from the user.

3.6 Hosts Can Link The Event Playlist

* Hosts can link the playlist being played at the event by linking their pre-existing Spotify playlist using the Spotify API.

3.7 Keep Record of Hosts

* The system keeps a record of how many tokens are present in each user account.
* The user can top-up tokens by making payments.
* The user has the option to convert the tokens back into real money in their bank account, after reaching a minimum of 1000 tokens.

3.8 Keep Record of Tokens In Each User Account

* The system keeps a record of how many tokens are present in each user account.
* The user can top-up tokens by making payments.
* The user has the option to convert the tokens back into real money in their bank account, after reaching a minimum of 1000 tokens.

3.9 Keep Record Of User Interaction per Event

* The system will keep track of the user's upvotes and comments for each event.
The system keeps track of the limitations of the upvotes (if any) and will not allow the user to upvote more if they have exceeded their upvote count.

3.10 Keep Record of User/Host Song Transactions

* Keep records of all of the user/host transactions and their status which specifies whether the transactions were declined, accepted or in process.

3.11 Keep Record of Users Attending Each Event

* Keep records of all of the user/host transactions and their status which specifies whether the transactions were declined, accepted or in process.

3.12 Keep records of registered users

* The system keeps record of the registered users by authorising their Spotify accounts through the Spotify API. Each User has a unique username. The users also have the option of linking their payment card and billing address for making or receiving payments. Each user has access to their personal data through their profile where they can modify their details.

3.13 Requesting a song

User send song requests to host within event genre restrictions. A user has to set up their payment details in order for them to request a song. They can pay more than the minimum amount specified in order to increases their chances of the host accepting their request. They have the option to add a short message to the host.

3.14 Server

* The backend of the web application will be stored on a third party server running Linux.

3.15 Server Availability

* The server needs to be available at least 90% of the time from 12 pm till 6 am. The rest of the time can be used for maintenance if need be.

3.16 Users Can Send a Song Request

* User send song requests to host within event genre restrictions. A user can use the token balance available. They can pay more tokens than the minimum amount specified in order to increases their chances of the host accepting their request. They have the option to add a short message to the host.

* Host can view all the song requests for each event:
* Host has the option to accept a song request, by responding with possible time periods where the song can be played. The user then can choose a time slot or decline the request within 48 hours of Host's response. If the user accepts the time slot, the Host needs to provide proof by recording a video of the song being played and using Shazam API we can verify if the song was played or not. If the song was not played, the user gets his money back.

3.17 Web Application

* The client side (hosts and users) will have a web application interface accessible on a web browser.
* The web application will be running on a third party server.
* The application must be running in Microsoft Edge, Safari, FF 12+ and Chrome.

3.18 Web Application Availability

* The web application needs to be available at least 90% of the time from 12 pm till 6 am. The rest of the time can be used for maintenance if need be.

4. Use Cases

4.1 Account Management

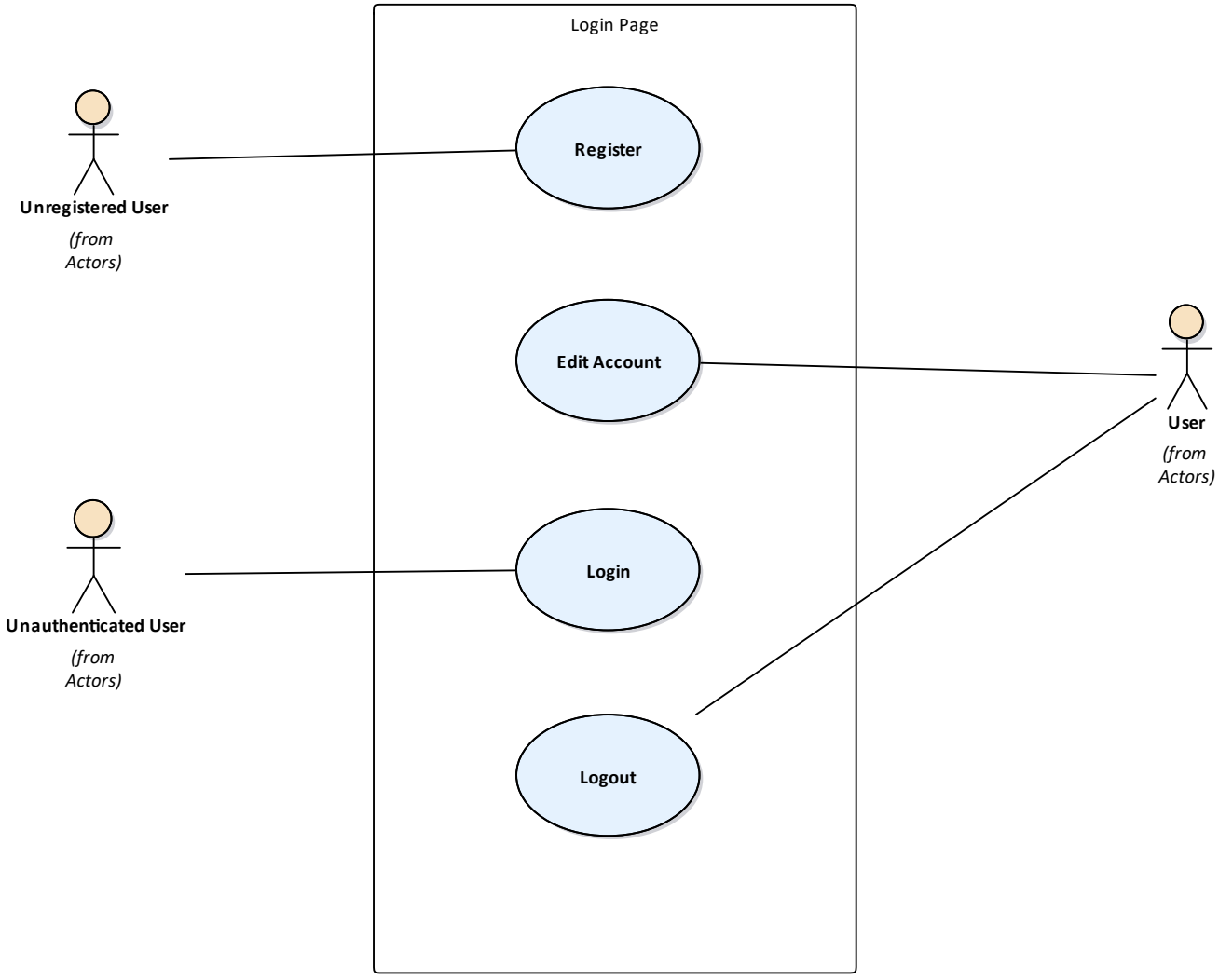


Figure 9 - Account Management

4.2 Actors

Contains Actors that represent the roles that users play with respect to the system.

4.3 Backup Management

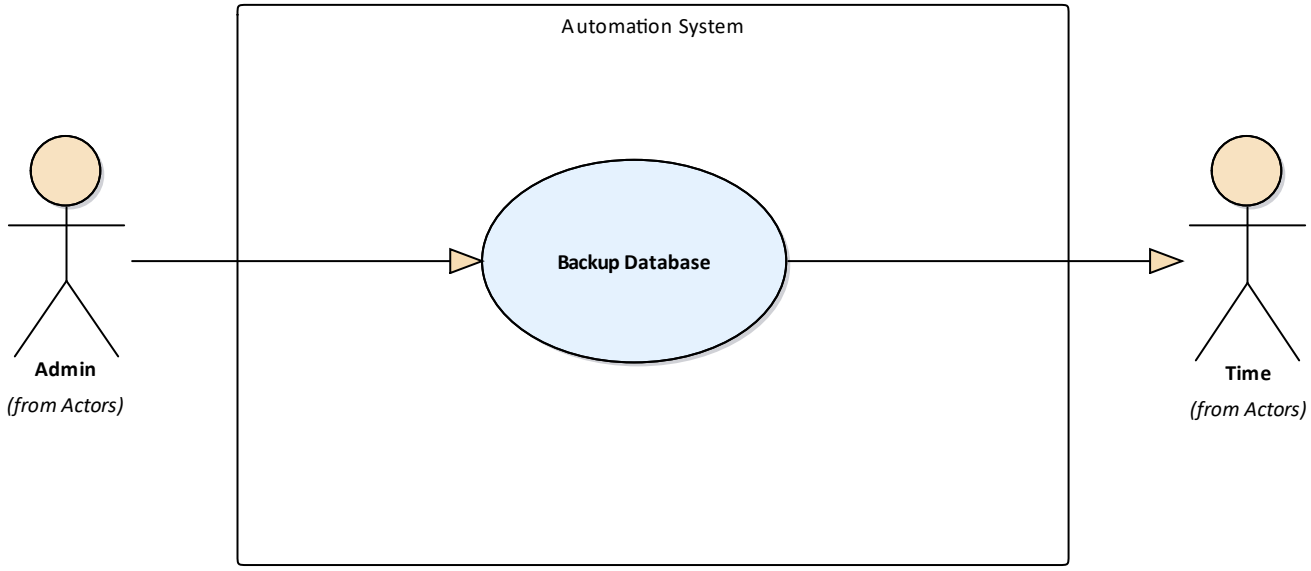


Figure 10 - Backup Management

4.4 Event Management

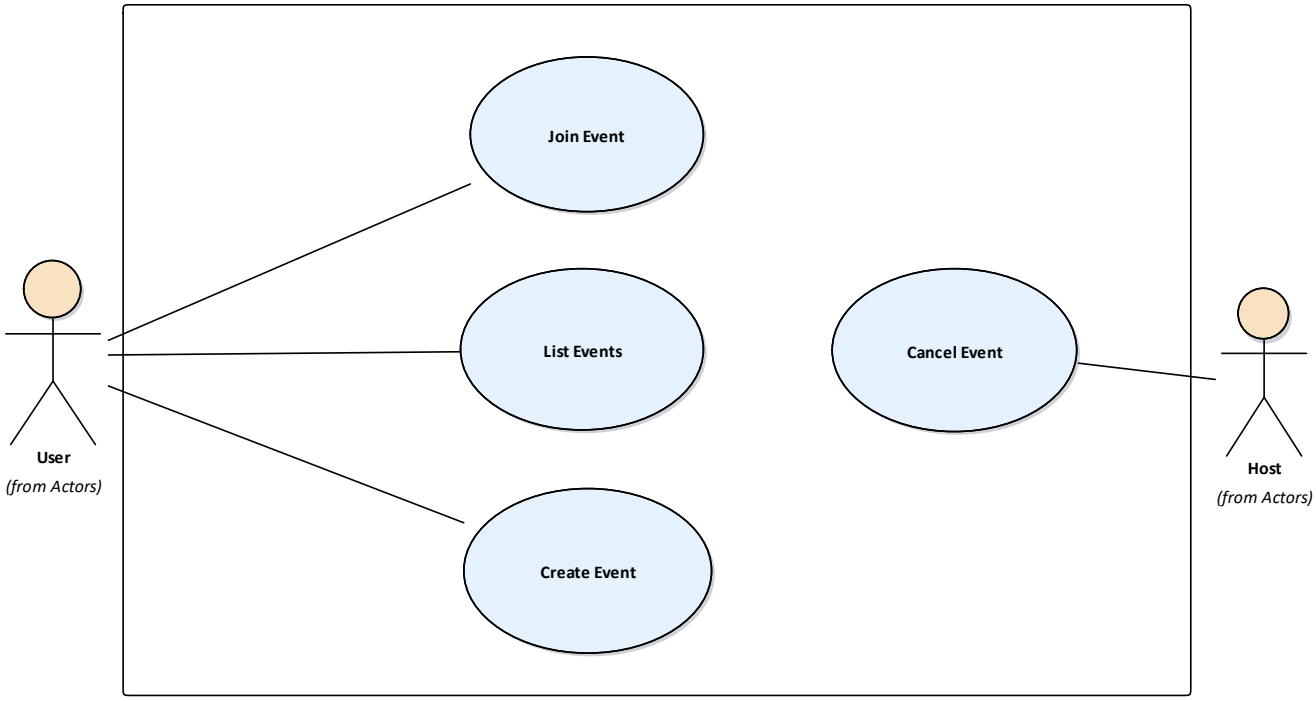


Figure 11 - Event Managment

4.4.1 Cancel Event

Host can cancel the event he/she created at any given time.

4.4.2 Create Event

Host creates event and make it public for all app users.

Basic Path: Create event

1. A user decides they would like to host an event and clicks on 'Create Event' button on the homepage.
2. The system redirects the user to a form to fill the event name, location, time period, date, description and minimum request price for a song.
3. The user fills in the information and clicks submit.
4. The system validates the data and redirects the user to their new events page.

Exception: Mandatory Fields not filled out

1. System prompts User to fill out required fields before continuing.

Exception: Duplicate event

1. System detects that event at this location, time, and with this name is already created and notifies the user

with this error

4.4.3 Join Event

User joins the events and has the ability to upvote songs.

Basic Path: Join Event

1. The user is on an Event's page that they are interested in and clicks on 'Join Event' button
2. The system then adds the user to the list of participants to the current event's page and gives them access to the event's contents

4.4.4 List Events

User sees list of events created by Hosts.

Basic Path: List Event

1. In the home page the user finds 'View nearby events' view and clicks it
2. The system then shows the user the list of events happening nearby by prioritizing events by the date and then by amount of friends
3. The user can filter the view by selecting a specific genre, date or location
4. The system renders the view accordingly

4.5 Friends Management

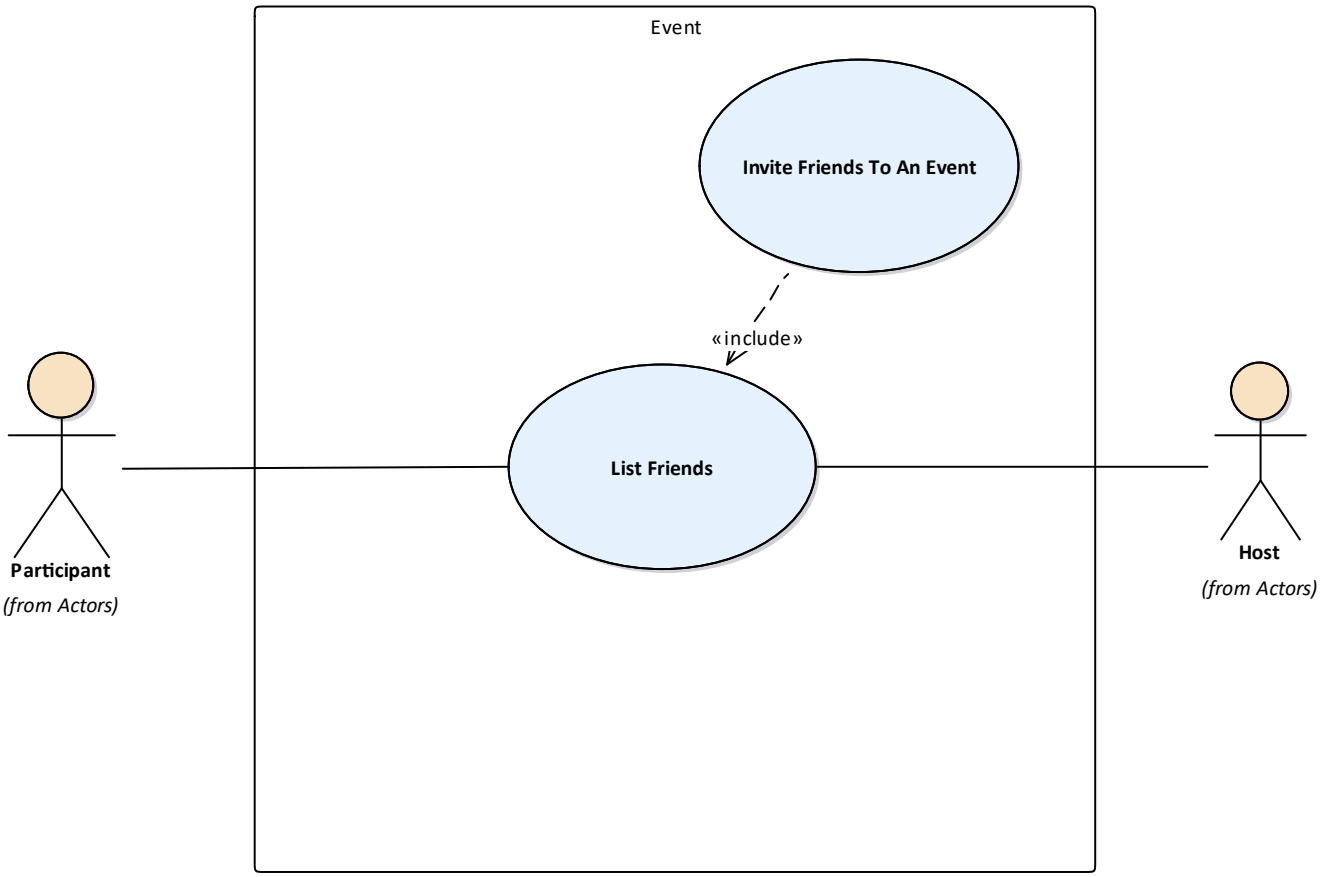


Figure 12 - Friends Management

4.5.1 Invite Friends To An Event

Participant can list his/her friends and invite any of them to an event if they are not already joined.

4.5.2 List Friends

Participants can see their list of friends possibly for inviting them to an event.

4.6 Request Management

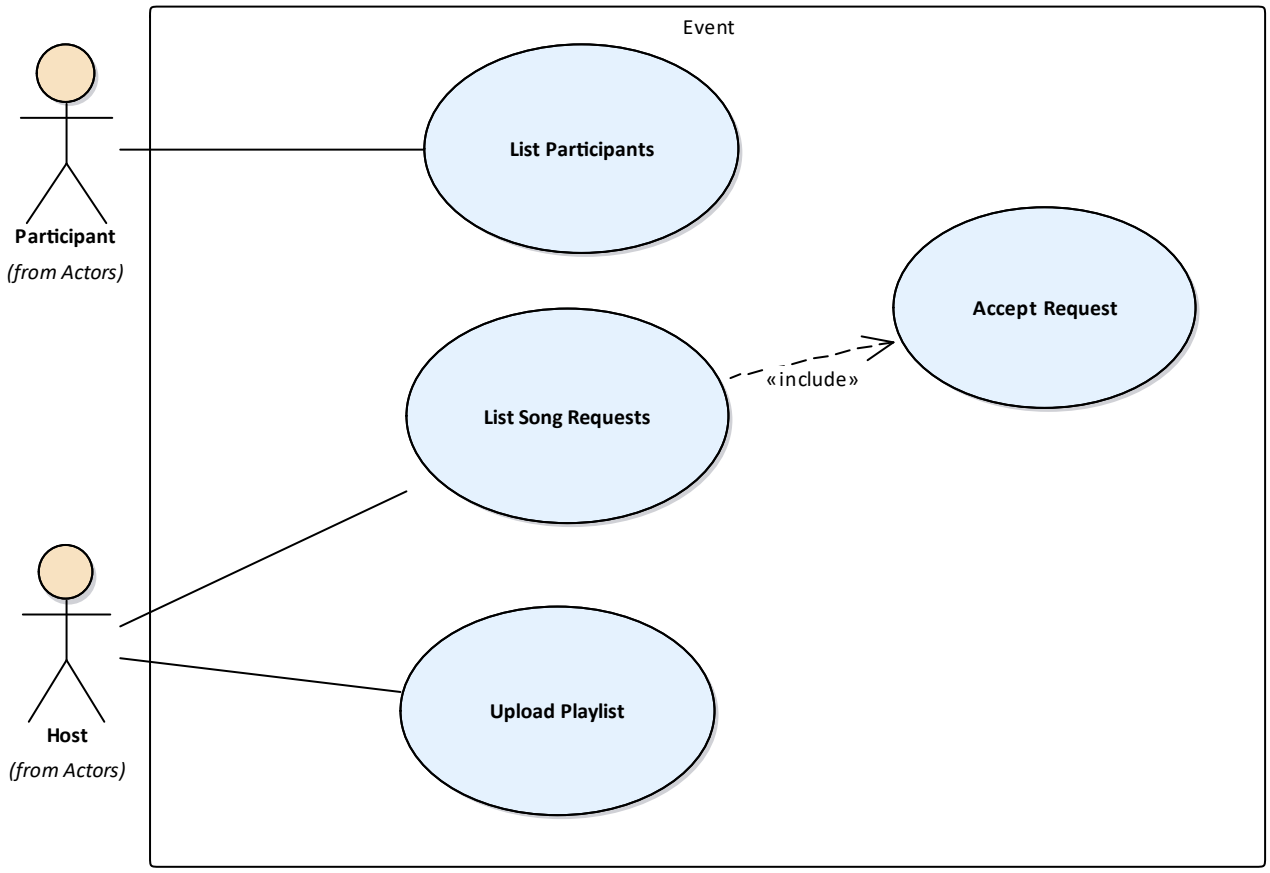


Figure 13 - Request Management

4.6.1 Accept Request

Host has the ability to reject or accept song requests in exchange for money.

4.6.2 List Participants

Host is able to see the list of participants who joined his/her event.

4.6.3 List Song Requests

Host lists the song requests made by the users.

4.6.4 Upload Playlist

Host forms and uploads playlists to the public app platform.

4.7 Voting Management

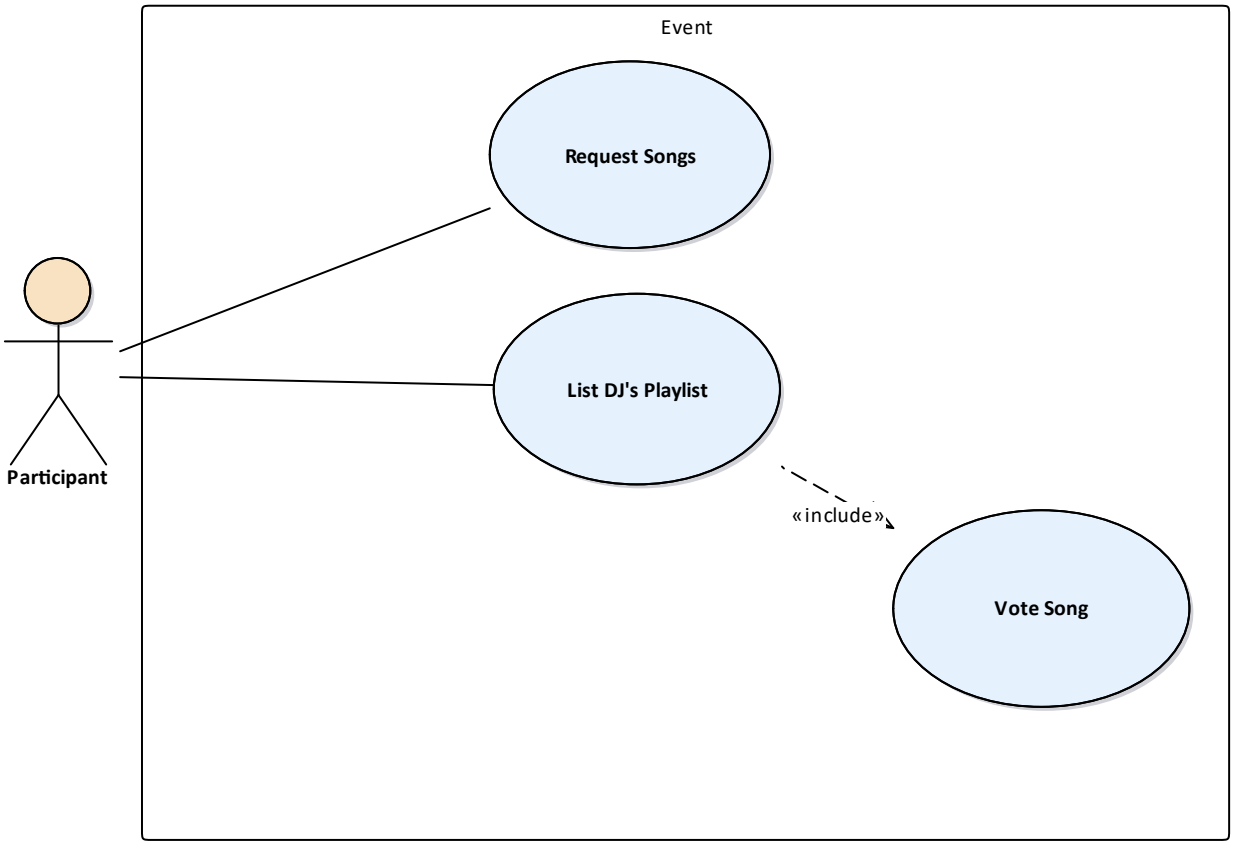


Figure 14 - Voting Management

FAKULTA
INFORMAČNÍCH
TECHNOLOGIÍ
ČVUT V PRAZE