System Specification - wastun

1 Initial Situation and Goal

1.1 Initial Situation

During a teenagers life, time is prescious, they want to make the best out of it. So spending a lot of time, just to plan your weekly partying or activities with friends is not desirable.
Googling for activities, just to find ones you do not like or ones that are just too expensive for your small budget is really not the yellow of the egg.

Another example: Did you ever spontaneously find an event for tonight or just the weekend.? Several minutes of Facebook event searching will do the trick, but again, it would be much easier and way more comfortable if you could see a list of all ongoing events at one sight.

1.1.1 Application Domain

The website helps with finding events nearby including other filters like price, if there is public transport accessible, sporty or not and shows routes and prices for them.
Also included is a öbb scotty support for public transport and nearby cab companies.

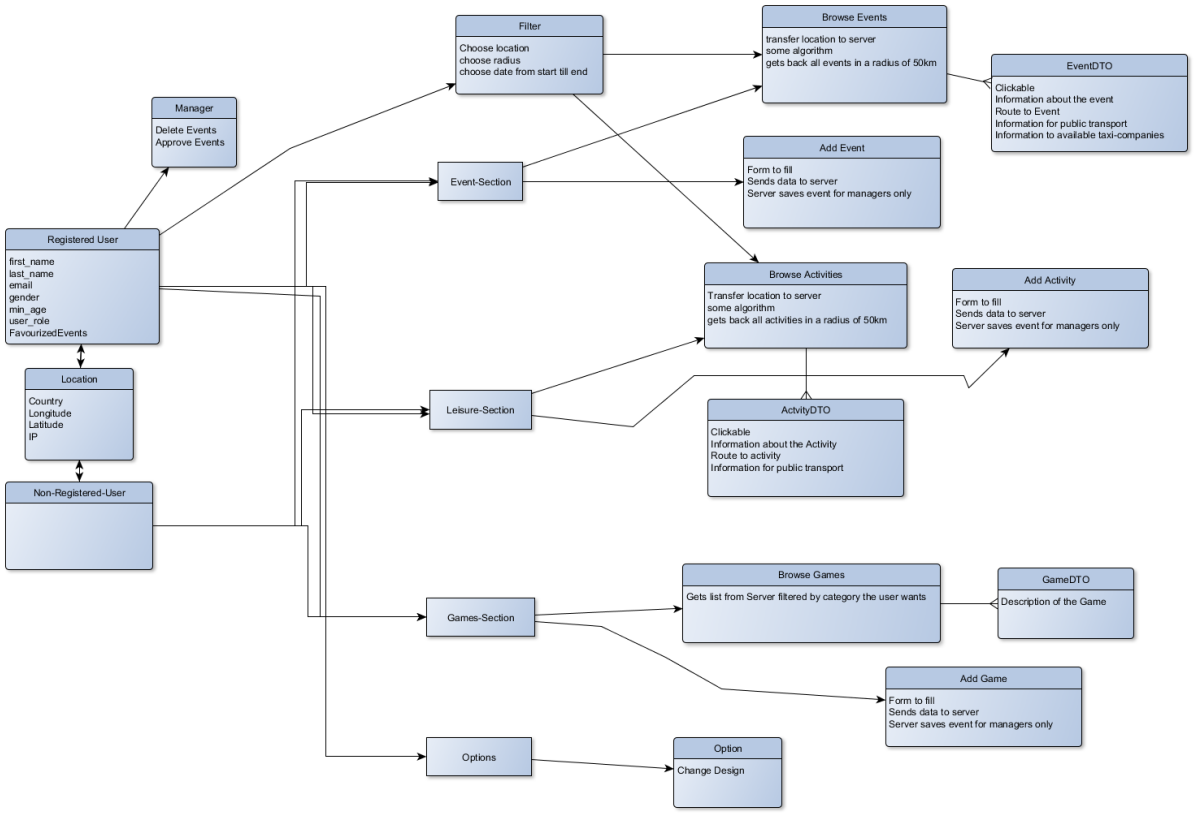
Our website shows several event types including:

- Parties
- Club-Events
- Bars
- Spare time activities
    - Sporty Activities(swimming,climbing, lasertag, billiard)
    - Non-Sporty Activities(games for home, cinema, points of interest)

1.1.2 Glossary

<API> <Application Programming Interface>

1.1.3 Model of the Application Domain



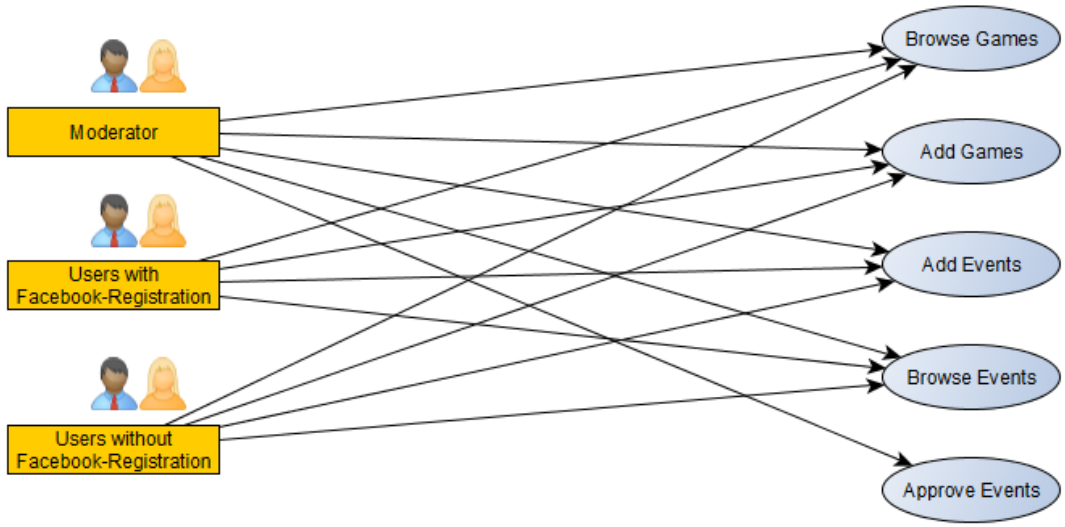
1.2 Goal Definition

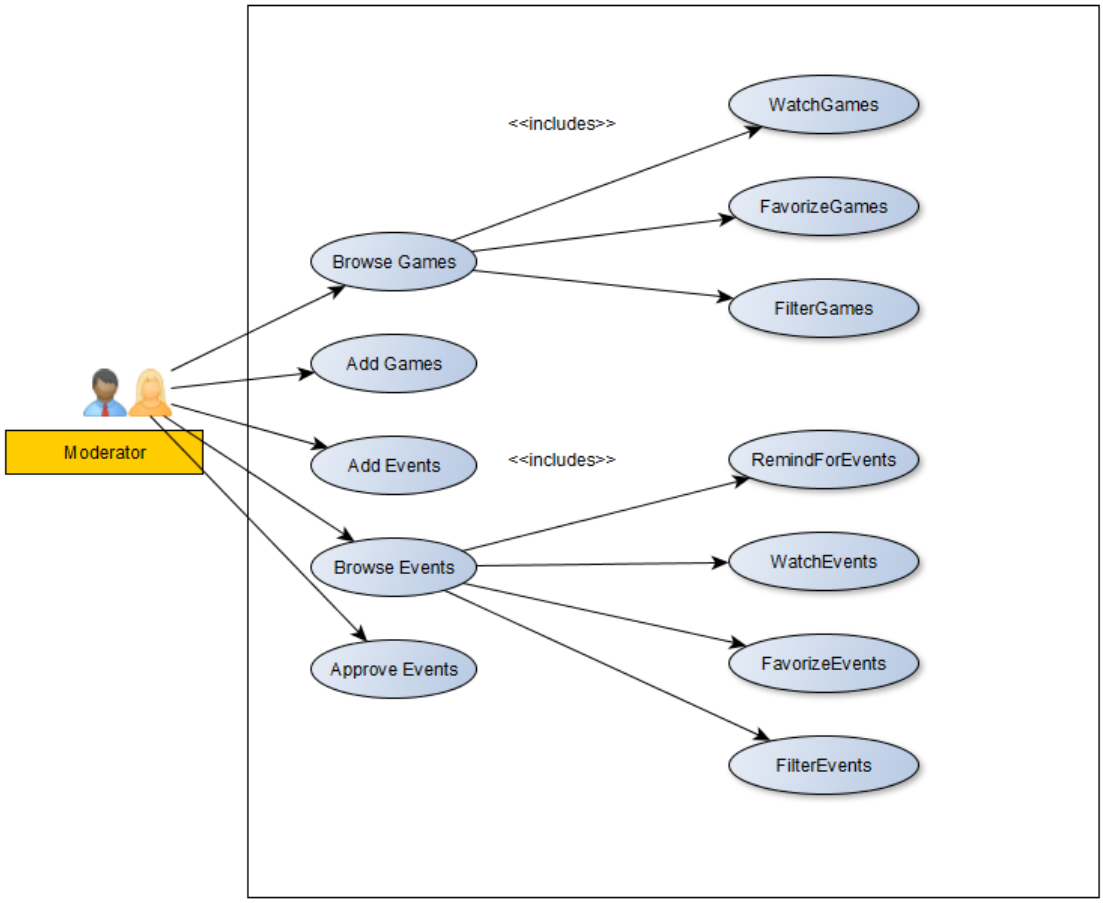
The main goal of our idea is to ease finding the perfect event for everyone. To make this happening we are working on our website called wastun.at which searches all events available nearby your location plus additional information to the event and routes. There will be also many more filters added.

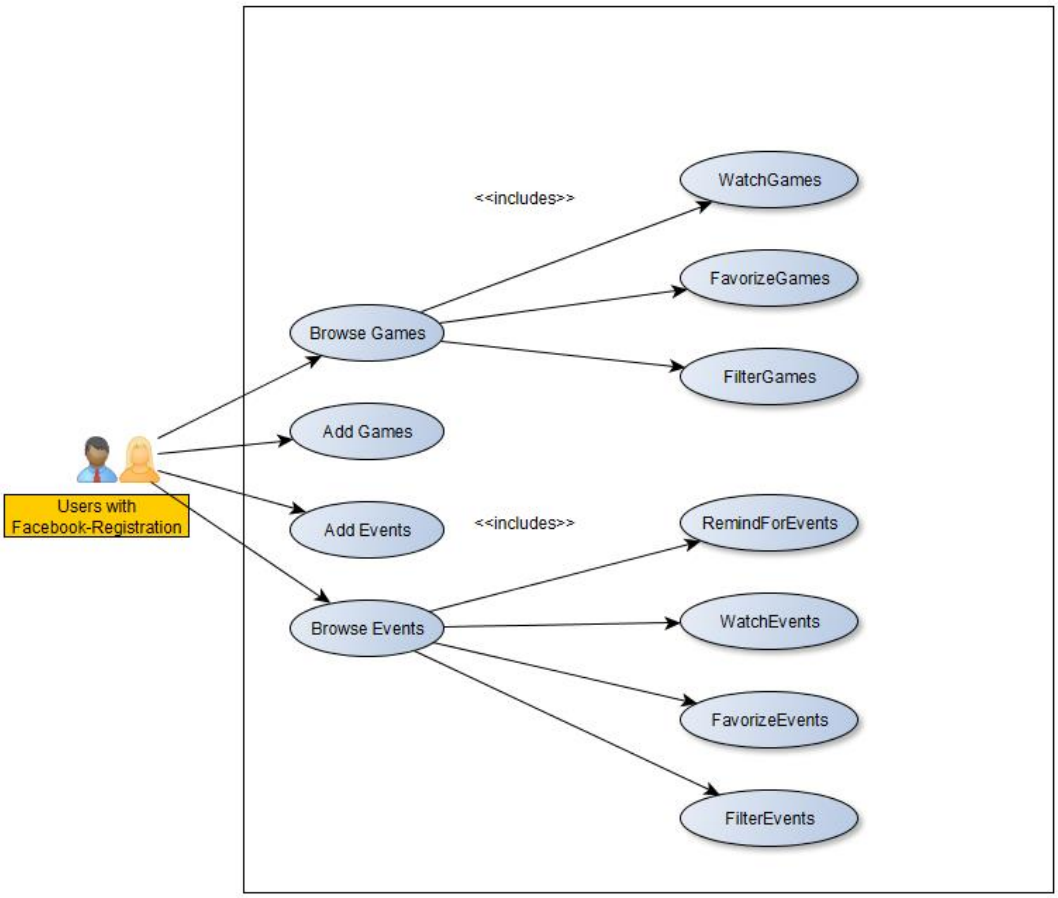
The main target group for our project are teenagers, but also adults or families, more precisely: Everybody that wants to get the best events available in a short time. wastun.at will be easy to handle and nearly everybody has an internet access today, more is not needed to run our service. We will also release several apps for android, iOS, Windows and Mac OS X.

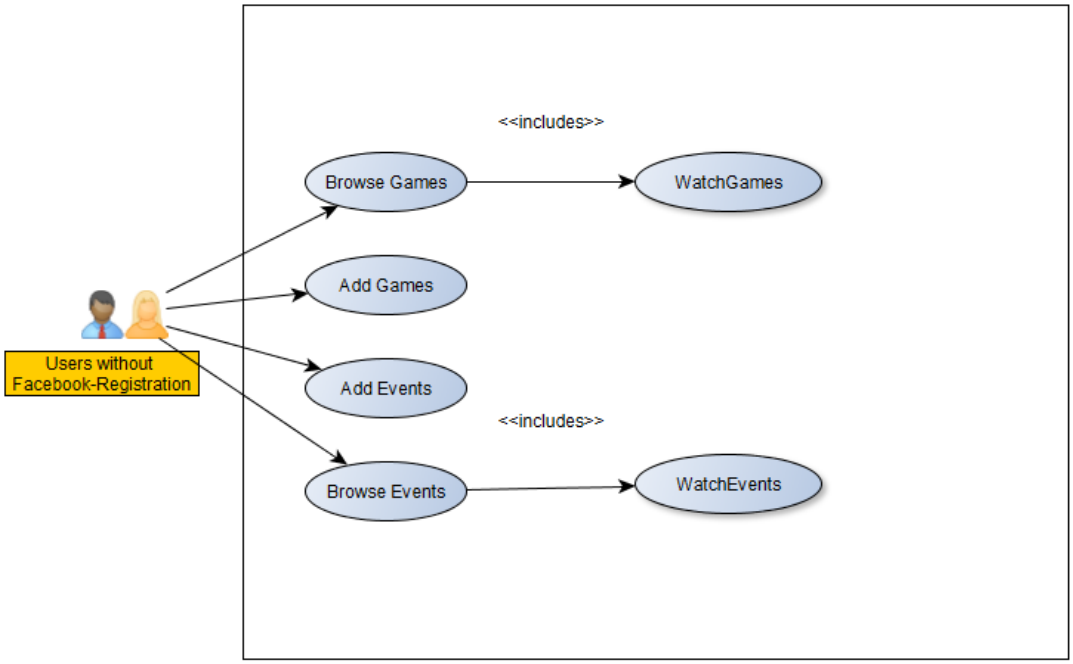
2 Functional Requirements

2.1 Use Case Diagrams




Moderator
Browse Games
Add Games
Add Events
Browse Events
Approve Events
<<includes>>
<<includes>>
WatchGames
FavorizeGames
FilterGames
RemindForEvents
WatchEvents
FavorizeEvents
FilterEvents


Users with
Facebook-Registration
Browse Games
Add Games
Add Events
Browse Events
<<includes>>
<<includes>>
WatchGames
FavorizeGames
FilterGames
RemindForEvents
WatchEvents
FavorizeEvents
FilterEvents


Users without
Facebook-Registration
Browse Games
Add Games
Add Events
Browse Events
<<includes>>
<<includes>>
WatchGames
WatchEvents

2.2 Description of use cases

<1 BrowseGames>

In the game section of the website, the user is able to watch, favourize or add games.

<1.1 WatchGames>
The user can watch games created by other users, including how to play them, the minimum or maximum of people that are able to play them and things that are required, like dices for example.

<1.2 FavourizeGames>
If the user likes a game, he is able to favorize it. After that, the game is added to the Section MyFavourites, next to his favourite events.

<2 AddGames>
If one clicks at the plus button in the right bottom corner, a form opens where he is able to add his own game. After an algorithm has checked that the game is not spam or junk, it gets added to the games section.

<3 AddEvents>
If one clicks at the plus button in the right bottom corner, a form opens where he is able to add his own party- or leisure-time-event. After an algorithm has checked that the event is not spam or junk, it gets added to the event or leisure time section.

<4 BrowseEvents>
The user is able to filter, favorize, watch or add events on the website. He can also set a reminder for an event.

<4.1 WatchEvents>
The user can watch events in the leisure-time or party section, created by other users or directly imported from Facebook.

<4.2 FavorizeEvents>
If the user likes a event, he is able to favorize it. After that, the event is added to the Section MyFavourites, next to his favourite games. In this section are also the events the user is interested on in Facebook.

<4.3 FilterEvents>
The website has a filter menu to search for events more specific. Planned options for filtering are: location(current position of the user by default), radius around the location(25km by default), timespan between day x and y(a timespan of two weeks starting with the current day by default), categories. If there is any way to reach it with public transport or not and price range.

<4.4 RemindForEvents>
The user is able to set a reminder for an event. He then gets a push notification at the day of the event, to make him not forget about it.

<5 ApproveEvents>

Users with moderator permissions, are also able to approve Events. In the moderator section of the website, all “form fillings” from our users can be found. Before we have the algorithm finished to check the fillings automatically, our moderators have to read over them before they get posted on the site.

2.2.1 Characteristic Information

Superior business process:	1: BrowseGames
Goal:	The user finds a game he likes and is able to favorize it if he wants to
Precondition:	The user switches to the Game section and wants to find a game He wants to add a game to his favourites
Postcondition:	The user has found his game and is able to play it The game is added to the favourites of the user
Involved User:	User: Searches for a game and favorizes it Server: Hands the game informations for the games. After the user favorizes a game, the server adds it to the Favourites section.
Triggering Event:	Pressing the favorize button

Superior business process:	1.1: WatchGames
Goal:	The user finds a game he likes
Precondition:	The user switches to the Game section and wants to find a game
Postcondition:	The user has found his game and is able to play it
Involved User:	User: Searches for a game Server: Hands the game informations for the games.
Triggering Event:	Pressing the Games button

Superior business process:	1.2: FavorizeGames
Goal:	The user finds a game he likes and is able to favorize it if he wants to
Precondition:	The user wants to add a game to his favourites
Postcondition:	The game is added to the favourites of the user
Involved User:	User: Favorize a game it Server: After the user favorizes a game, the server adds it to the Favourites section.
Triggering Event:	Pressing the favorize button

Superior business process:	2: AddGames
Goal:	Successfully add a game
Precondition:	The user has a game idea and wants to share it via our website
Postcondition:	After filling the fields of the registration formular with right information, the game gets handed to the moderator-section
Involved User:	User: Needs to fill out the formular properly Server: hands the filled-out formular to the moderator-section Moderator: Checks the formular and adds it to the page if it is complete and serious (no junk or spam)
Triggering Event:	After filling in correct information, pressing the Send Form-Button will save the informations and handle it to the moderator section

Superior business process:	3: AddEvents
Goal:	Successfully add a event
Precondition:	The user has an event and wants to share it via our website
Postcondition:	After filling the fields of the registration formular with right information, the event gets handed to the moderator-section

Involved User:	User: Needs to fill out the formular properly Server: hands the filled-out formular to the moderator-section Moderator: Checks the formular and adds the event to the page if it is complete and serious (no junk or spam)
Triggering Event:	After filling in correct information, pressing the Send Form-Button will save the informations and handle it to the moderator section

Superior business process:	4: Browse Events
Goal:	The user is able to watch and favorize an Event, he is able to filter the informations and set a reminder for it.
Precondition:	The user switches to the Events section and wants to find an event He wants to filter the events He wants to set an Reminder for an event He wants to add a event to his favourites
Postcondition:	The user found his event and has enough information to attend to it easily He finds now only events which apply to the filters he chose He now gets an alert on the day of the event The event is added to the favourites of the user
Involved User:	User: Searches for an Event Chooses filters for searching Chooses an events he wants to be reminded of Favorizes an event Server: Hands the event informations to the users Applies filters Sets an reminder for the event Sets an event as a favourite
Triggering Event:	Pressing the Events-Button Pressing the Apply Filters-Button Pressing the Remind me-Button Pressing the favorize button

Superior business process:	4.1: Watch Events
Goal:	The user is able to watch an Event
Precondition:	The user switches to the Events section and wants to find an event
Postcondition:	The user found his event and has enough information to attend to it easily
Involved User:	User: Searches for an Event Server: Hands the event informations to the users
Triggering Event:	Pressing the Events-Button

Superior business process:	4.2: Filter Events
Goal:	The user is able to filter the required informations
Precondition:	The user wants to filter the events
Postcondition:	He finds now only events which apply to the filters he chose
Involved User:	User: Chooses filters for searching Server:Applies filters
Triggering Event:	Pressing the Apply Filters-Button

Superior business process:	4.3: RemindForEvents
Goal:	The user is able to set a reminder for an Event.
Precondition:	The user wants to set an Reminder for an event
Postcondition:	He now gets an alert on the day of the event
Involved User:	Chooses an events he wants to be reminded of Server: Sets an reminder for the event
Triggering Event:	Pressing the Remind me-Button

Superior business process:	4.4: Favorize Events
Goal:	The user is able to favorize an Event
Precondition:	He wants to add a event to his favourites
Postcondition:	The event is added to the favourites of the user
Involved User:	User: Favorizes an event Server: Sets an event as a favourite
Triggering Event:	Pressing the favorize button

2.2.2 GUI to call the use case

AddEvent

Step	User	Activity
1	user	Presses + Button in the right Corner of the Event section
2	user	Fills out the form and presses submit
3	server	hands form to moderator section
4	moderator	checks if form is not junk or spam
5	server	if form is not spam, adds event to website
6	user	gets a notification if the event is successfully added

AddGame

Step	User	Activity
1	user	Presses + Button in the right Corner of the Game section
2	user	Fills out the form and presses submit
3	server	hands form to moderator section
4	moderator	checks if form is not junk or spam
5	server	if form is not spam, adds game to website
6	user	gets a notification if the game is successfully added

Favorize Games

Step	User	Activity
1	user	Clicks on the “star”-Icon from a game to favorize it
2	server	Server saves the game to the user’s favourite page

WatchGames

Step	User	Activity
1	user	Clicks on the games section
2	server	Gets the games posted by users
3	server	Lists the games

WatchEvents

Step	User	Activity
1	user	Clicks on the events section
2	server	Gets location of the user and sends it to the server
3	server	Retrieves information and sends back events near the user

FilterEvents

Step	User	Activity
1	user	Clicks on the filter button
2	user	can set options to filter for (Location,Date,Radius)
3	user	clicks on the “filtern” button
4	server	responds with a new list of events filtered by the user input

RemindForEvents

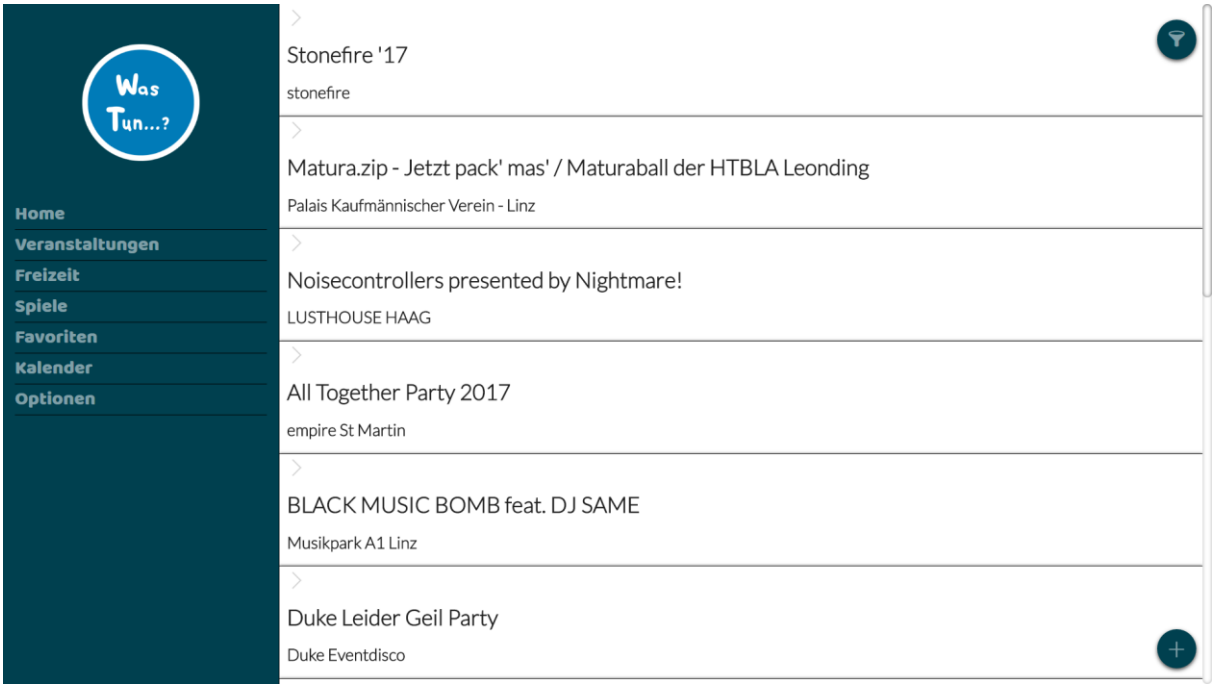
Step	User	Activity
1	user	Clicks on the remind button of an event
2	server	saves a reminder for the user the user

FavorizeEvents

Step	User	Activity
1	user	Clicks on the “star”-Icon from an event to favorize it
2	server	Saves the event to the user’s favourite page

2.2.3 GUIs for the standard use

2.2.3.1 Screenshots



2.2.4 Scenarios for non-standard uses (bad cases or work around cases)

AddEvent

Step	User	Activity
1	user	Presses + Button in the right Corner of the Event section
2	user	Fills out the form and presses submit
3	server	invalid form data

Step	User	Activity
1	user	Presses + Button in the right Corner of the Event section
2	user	Fills out the form and presses submit
3	server	hands form to moderator section
4	moderator	checks if form is not junk or spam
5	server	if form is not spam, adds event to website
6	client	connection interrupt

AddGame

Step	User	Activity
1	user	Presses + Button in the right Corner of the Game section
2	user	Fills out the form and presses submit
3	server	invalid form data

Step	User	Activity
1	user	Presses + Button in the right Corner of the Game section
2	user	Fills out the form and presses submit
3	server	hands form to moderator section
4	moderator	checks if form is not junk or spam
5	server	if form is not spam, adds event to website
6	client	connection interrupt

Filter

Step	User	Activity
1	user	Clicks on the filter button
2	user	can set options to filter for (Location,Date,Radius)
3	user	clicks on the “filtern” button
4	client	connection interrupt

Step	User	Activity
1	user	Clicks on the filter button
2	user	can set options to filter for (Location,Date,Radius)
3	user	clicks on the “filtern” button
4	server	invalid form data

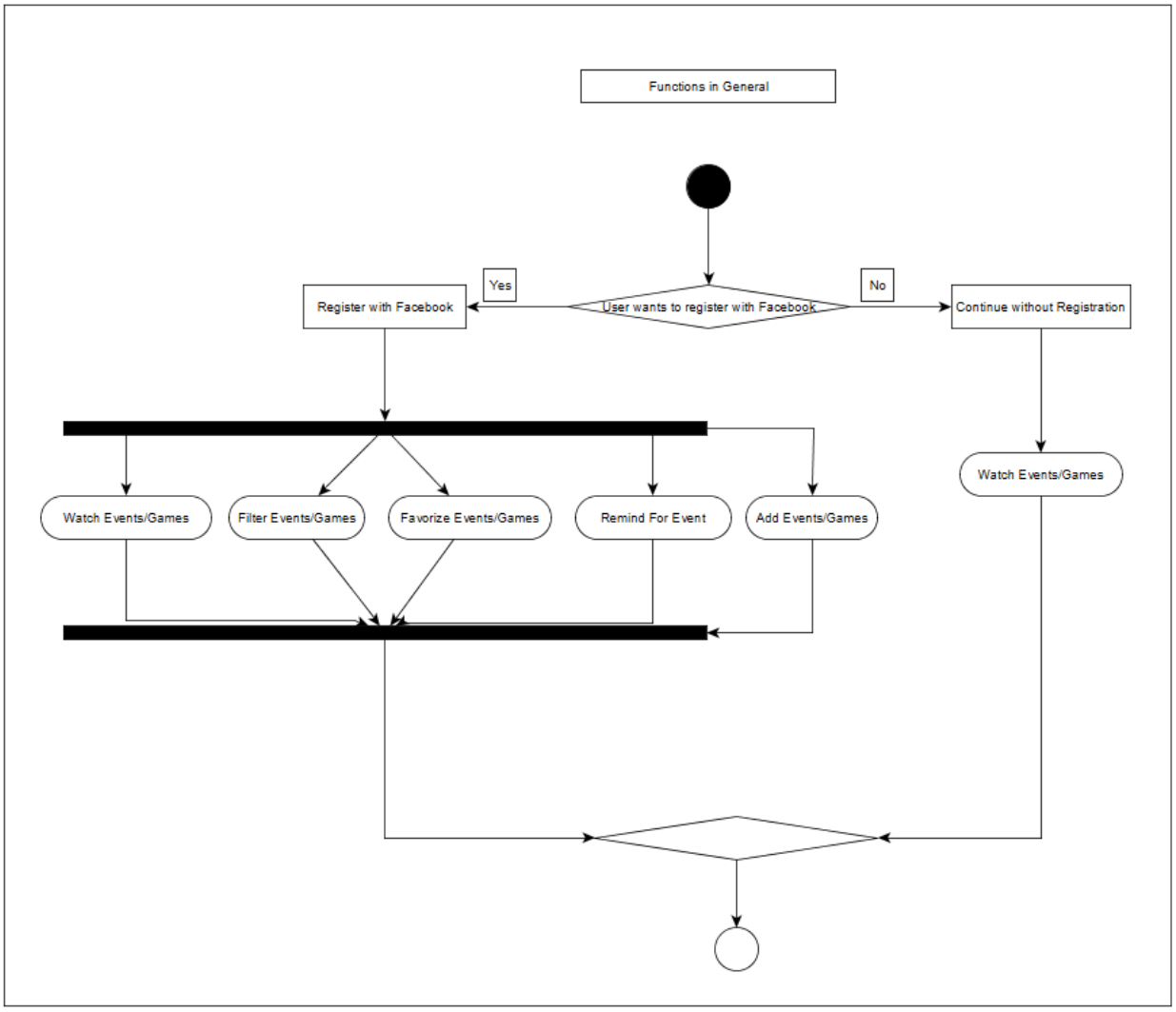
Approve Event/Game

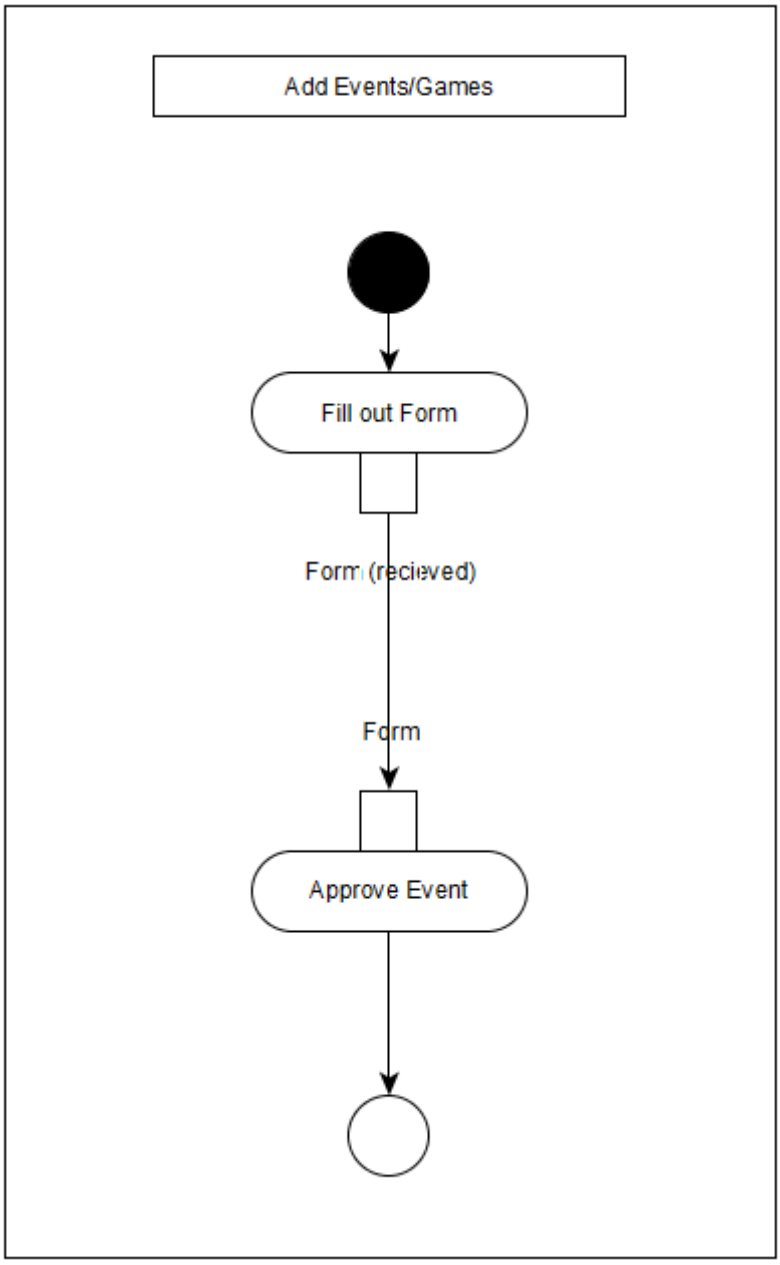
Step	User	Activity
1	admin	accepts event/game
2	client	connection interrupt

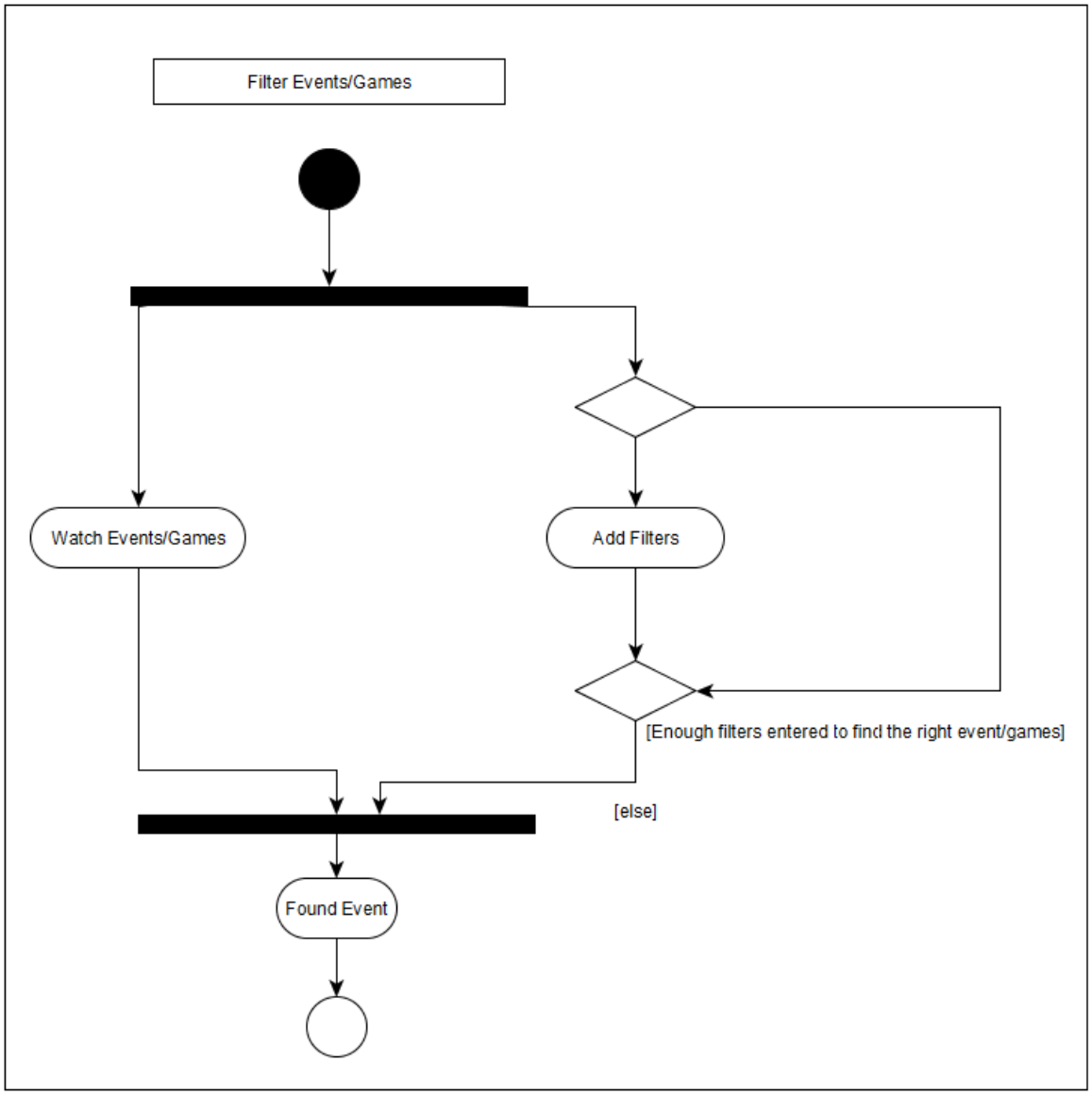
Decline Event/Game

Step	User	Activity
1	admin	declines event/game
2	client	connection interrupt

2.2.6 Workflow




Add Events/Games
Fill out Form
Form (recieved)
Form
Approve Event



2.2.7 Open Points

·Plans for the Diploma thesis

- Big Data Analysis: The user gets event suggested, which could be relevant for him, based on the events he already attended and on the events his friends are attending. He also gets alerts if he is in the vicinity of an event, and if he is leaving, gets tips for cabs he could call or public transport to take him home
- Android and iOS native apps: To make the best performance possible, and making additional features which are only available if the app is natively programmed, an android and an ios app are planned.

3 Non-functional Requirements

ID:	NFR_001
Name:	Design
Type:	USE
Description:	Design Conditions for Android apps, iOS-Apps and JavaFX applications have to be kept to create a familiar environment for the user
Assigned use cases:	----------

ID:	NFR_002
Name:	Layout
Type:	USE
Description:	Our target is to create a layout which is simple to understand and easy to work with for all generations.
Assigned use cases:	----------

ID:	NFR_003
Name:	Language
Type:	USE
Description:	We want to implement our software in the language english
Assigned use cases:	----------

ID:	NFR_005
Name:	Correct Data Insertion
Type:	SEC
Description:	In the Event-Form, inserts have to be checked if they are not spam or junk
Assigned use cases:	RegisterEvent(2)

ID:	NFR_006
Name:	Avoidance of redundancy
Type:	EFFIC
Description:	The website/app has to ensure that there is not redundancy in the database(for example an event is twice registered)
Assigned use cases:	RegisterEvent(2)

ID:	NFR_007
Name:	Implementing of another language
Type:	MAINT
Description:	In a later process we also want to publish our site in german, because it is our mother language and is one of the biggest languages in the world
Assigned use cases:	----------

Types of non-functional requirements

Type	Name	Description
USE	Usability requirement	This requirement is to make the target group as described in section 1 is liking to work with that system.
EFFIC	Efficiency requirement	Run-time and memory efficiency. What are the constraints under which the system has to run.
MAINT	Maintenance and portability requirement	Which maintenance or porting effort is expected in the future? Internationalization expected? Porting to different hardware platform?...
SEC	Security requirement	Security requirements comprise confidentiality, data integrity, and availability. How much do we have to consider that data is not accessible to unauthorized persons? Is the correctness and/or consistency of data to be guaranteed? How severe are total system faults?
LEGAL	Legal requirement	Are there any standards or legal constraints to be considered?

4 Quantity Structure

The following informations about the user get saved in our database:

- Name, E-Mail
- Favorized Events, Reminders for Events as well as for Games
- All Attended Events(see Open Points, big data analysis)

The database will be a plain, Oracle MySQL Database.

As our Webhost we chose world4you.at

2 System Architecture and Interfaces

Our website works with the following APIs. To find the events, the Facebook-API is used. If a user uses our site with Facebook, his whole public events get recognized and put into our event finder, the private events remain private, of course. To have a static number of events, we also use several “crawlers” which are Facebook users, just made to get as many events as possible on the website.
To get travel routes, taxi-drivers nearby, events including their prices, we use the Google Maps API. For the public transport information, ÖBB Scotty API is used.
Connected with the user’s location, we can provide information about the best events in range, public transport, travelling routes and taxis nearby.

To list the event on our website we take the data provided by facebook. If this is not sufficient, we send an automatically generated message via Facebook to the event owner. If he wants to provide better information on our website, he is able to submit extra data with our included form. The same form is also directly available on our website, for adding a new event or spare time activity. After submission, an algorithm checks if the form is not spam or junk, if the program can not certainly detect any of these an email gets sent to us.

For the Frontend we use HTML 5, Sass and Typescript, the Backend is written in Java 8 with additional Frameworks like Spring. To make our web application accessible for Smartphones, we use Electron to convert it into a mobile application for Android, iOS and Windows, instead of developing a native application each.

3 Acceptance Criteria

Test Step	Expected Behaviour	Test fails if...

Favourize Game:

User with Facebook connect clicks the favourize button attached to a certain game	chosen game gets added to users favourites	conection interrupt, user not connected with Facebook

Add Game

User fills the form	user enters correct Data	User enters incorrect data
User submits the form	add game to list of games to approve	invalid form data, connection interrupt

Add Event

User fills the form	user enters correct Data	User enters incorrect data
User submits the form	add event to list of events to approve	invalid form data, connection interrupt

Favourize Event:

User with Facebook connect clicks the favourize button attached to a certain event	chosen event gets added to users favourites	conection interrupt, user not connected with Facebook

Approve Event:

Admin approves an event added by an user	event gets added to event-list	connection interrupt

Decline Event

Admin declines an event added by an user	event gets deleted	connection interrupt