# 

# EventMate

**Plan-Driven and Agile Programming**

**TEI Crete, Winter semester 2018/2019**

Petra Cendelínová

Petr Kalas

[EventMate 1](#_Toc528662693)

[1. System specification 3](#_Toc528662694)

[1.1. Introduction 3](#_Toc528662695)

[1.2. Description 3](#_Toc528662696)

[1.3. Requirements 3](#_Toc528662697)

[1.3.1. Requirements list 4](#_Toc528662698)

[1.3.2. Requirements diagram 4](#_Toc528662699)

[1.4. Use Case 4](#_Toc528662700)

[1.4.1. Use case diagram 4](#_Toc528662701)

[1.4.2. Use case details 4](#_Toc528662702)

[1.5. Mockups 5](#_Toc528662703)

[1.5.1. Event owner (Android) 5](#_Toc528662704)

[1.5.2. Event owner (web) 5](#_Toc528662705)

[1.5.3. Event assignee 5](#_Toc528662706)

[2. Architecture design 5](#_Toc528662707)

[3. Attachments 8](#_Toc528662708)

[3.1. Attachment 1 – Use case diagram 1](#_Toc528662709)

## System specification

## Introduction

For human beings, it is natural to socialize and interact with each other. Technology has become more and more important in the social aspects of life. Nowadays people prefer to spend more time on portable devices. Social networking has moved to mobile platforms, which are accessible anywhere and anytime.

The objective of our work is to develop a social application for organizing events any type. As events can be considered birthday parties, new year eve, weddings, baby showers etc.

As we want to make this application more general for any kind of events, we have decided to call it by name “EventMate”. The main goal of the application is to provide tasks management (create, assign, close) and also to create a communication channel among event owners and guests. The application is going to content gamification elements such as a scoreboard and badges.

## Description

The following chapter explains major features. To use the application it is necessary to create a user's account or log in via existing social accounts. In our application, there are various user roles such as owners, contributors, guests. As we mentioned earlier, the main goal is to provide task management for an event. For all practical purposes, it means that user can create a task with corresponding attributes such as name, deadline, and persons to be assigned to it. All these fields are saved and continuously maintained. During the whole event, gamification principles are applied which help users to feel that they are a part of the game. When the event has finished, event summary is provided to particular users.

## Requirements

This section describes requirements for EventMate application.

## Requirements list

In this section there are listed all functional and also non-functional requirements.

* + - 1. Functional requirements
* User registration
* User registration via Facebook / Google account
* User login
* Creating events
* Creating events from template
* Listing events
* Filtering events
* Modifying events
* Deleting events
* Changing event state
* Creating tasks
* Listing tasks
* Modifying tasks
* Changing task state
* Submitting task results
* Editing task photo
* Assigning points
* Creating reports
* Sharing reports
* Sending private messages
* Sending group messages
* Showing user profile
* Changing own settings
* Notification system
  + - 1. Non-functional requirements
* Online / Offline Android mode
* Supported Android version 6 – 8
* Supported Web version by all browsers
* Responsible Android frontend
* Responsible Web frontend
* Android app available in portrait mode
* General usable REST API
* Secured REST API
* Ownership permission policy
* Maximum response time of 2 seconds
* Multiplatform backend support
* Log rotation ability
* Android permission neco
* Account password encryption
* Multilanguage support

## Requirements diagram

asaddas

## Mindmap

Mindmap for EventMate apliccation is located at attachment 2.

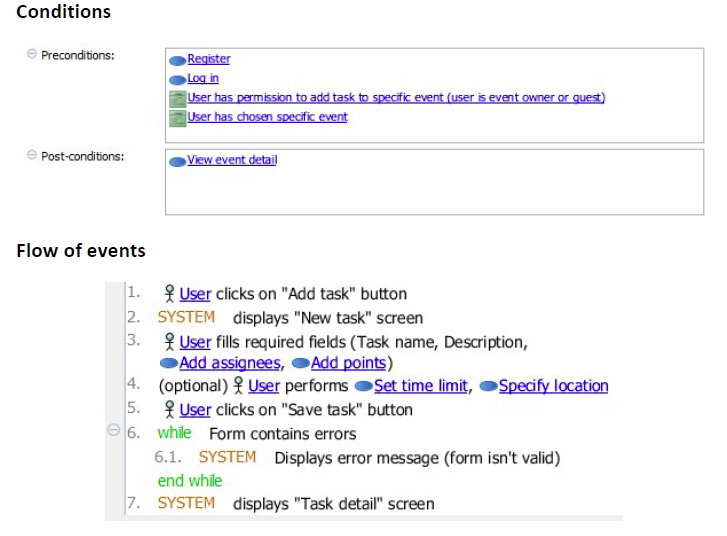
## Use Case

asdas

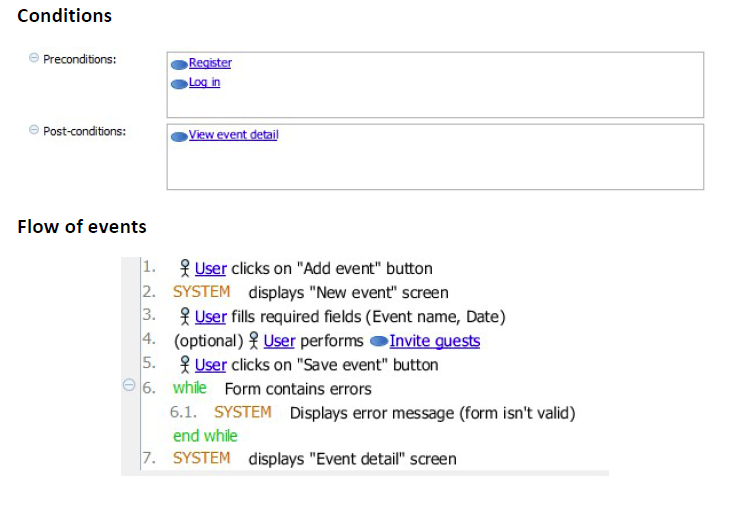
## Use case diagram

Use case diagram is located at attachment 1.

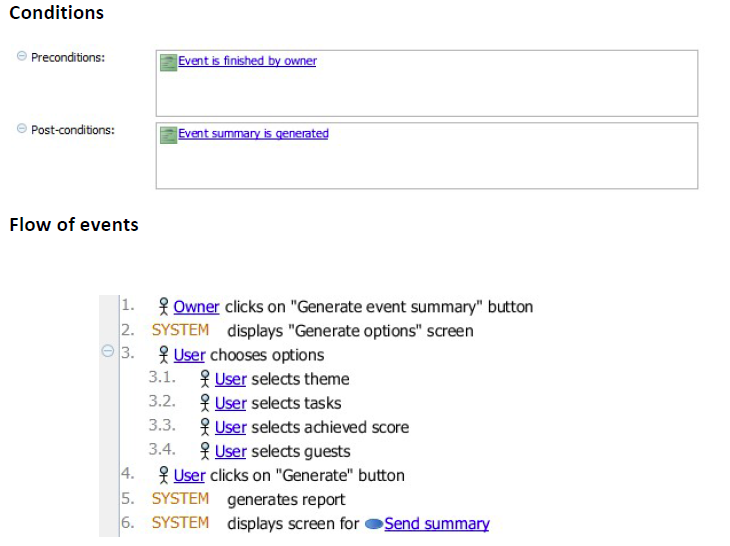
## Use case details



Caption 1 - Add task use case detail

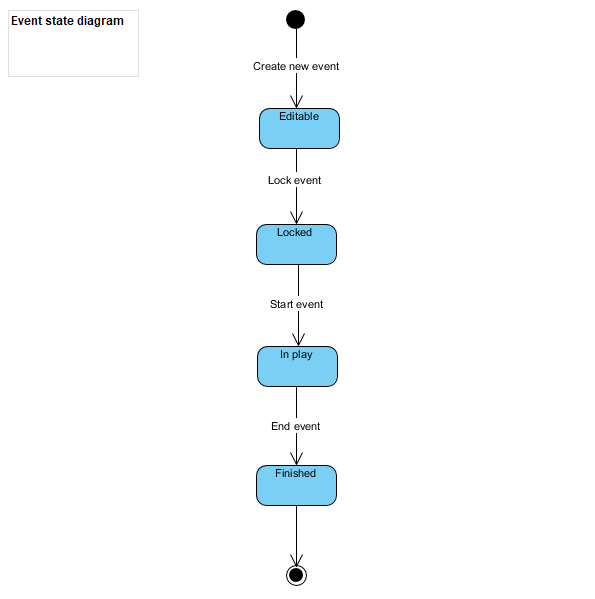


Caption 2 - Create event use case detail

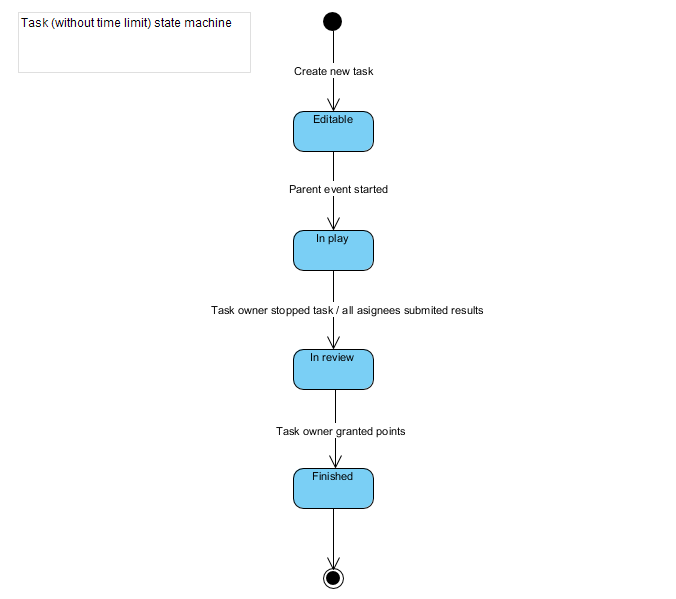


Caption 3 - Generate event summary use case detail

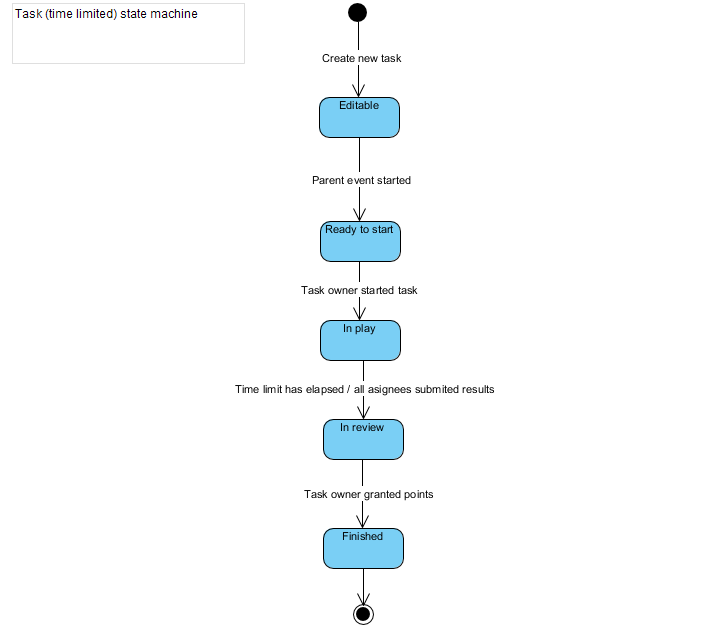
## State machine diagram



Caption 4 - Event state diagram



Caption 5 - Task state diagram

s

Caption 6 - Task (time limited) state diagram

## Mockups

asda

## Event owner (Android)

asd

## Event owner (web)

asd

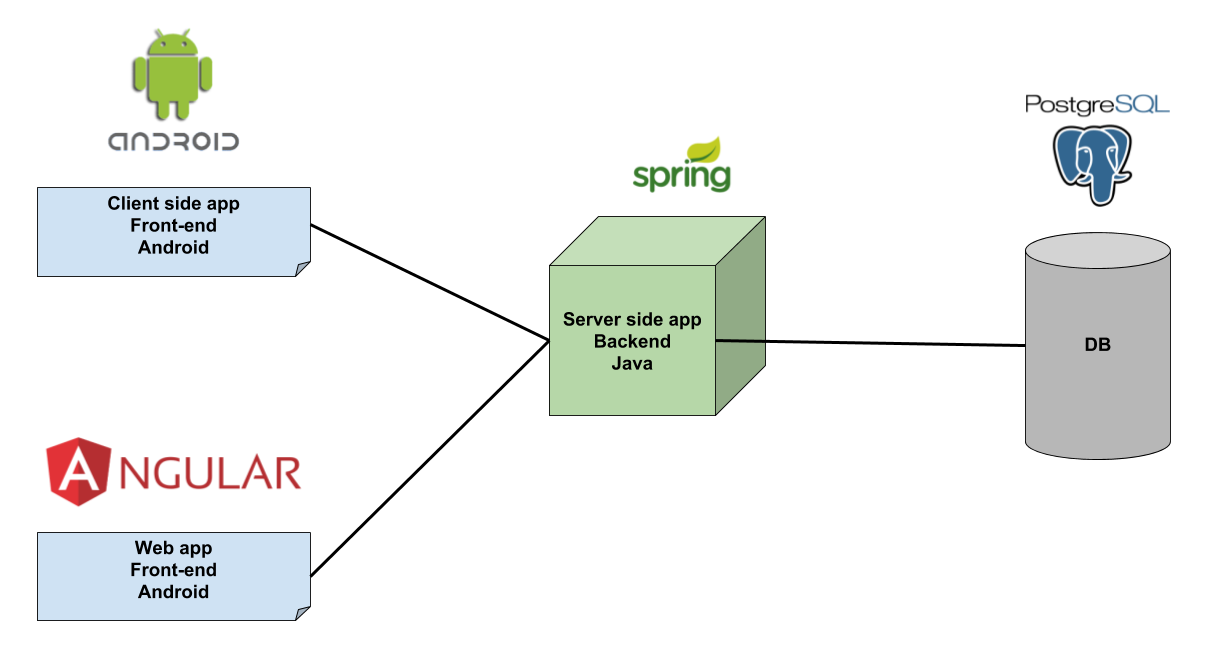
## Event assignee

asd

## Architecture design

Architecture style is based on **client-server.** The project consists of two client side parts and server side with a database.

* 1. Component diagram
  2. Architecture diagram



**Server side**

Purpose of server-side application is to provide API, user authentication and persistence to the client side. The asynchronous notification system will be also provided by the server side.

Technology stack (early version)

* Java
* Spring Framework
* PostgreSQL

**Client side**

**Client side - Android**

First type of client-side is going to be implemented as a native mobile application for platform Android There will be a huge emphasis on UX (User experience) and also on gamification techniques. Android application is going to adhere MVVM architecture that allows separating the user interface logic and the business logic.

Technology stack

* Kotlin
* LiveData
* Library Retrofit for API calls
* Dagger for dependency injection

**Client side – Web**

Another type of client-side is going to by implemented as a web application based on Angular framework.

Technology stack

* Angular 6 framework

TODO:

Chapter 1 – system specification (diagrams..)

Description, requeiremnts list + diagram, textual description, use case diagram, mockups, short description,

Chapter 2 (Architecture design (title))

- Component diagram

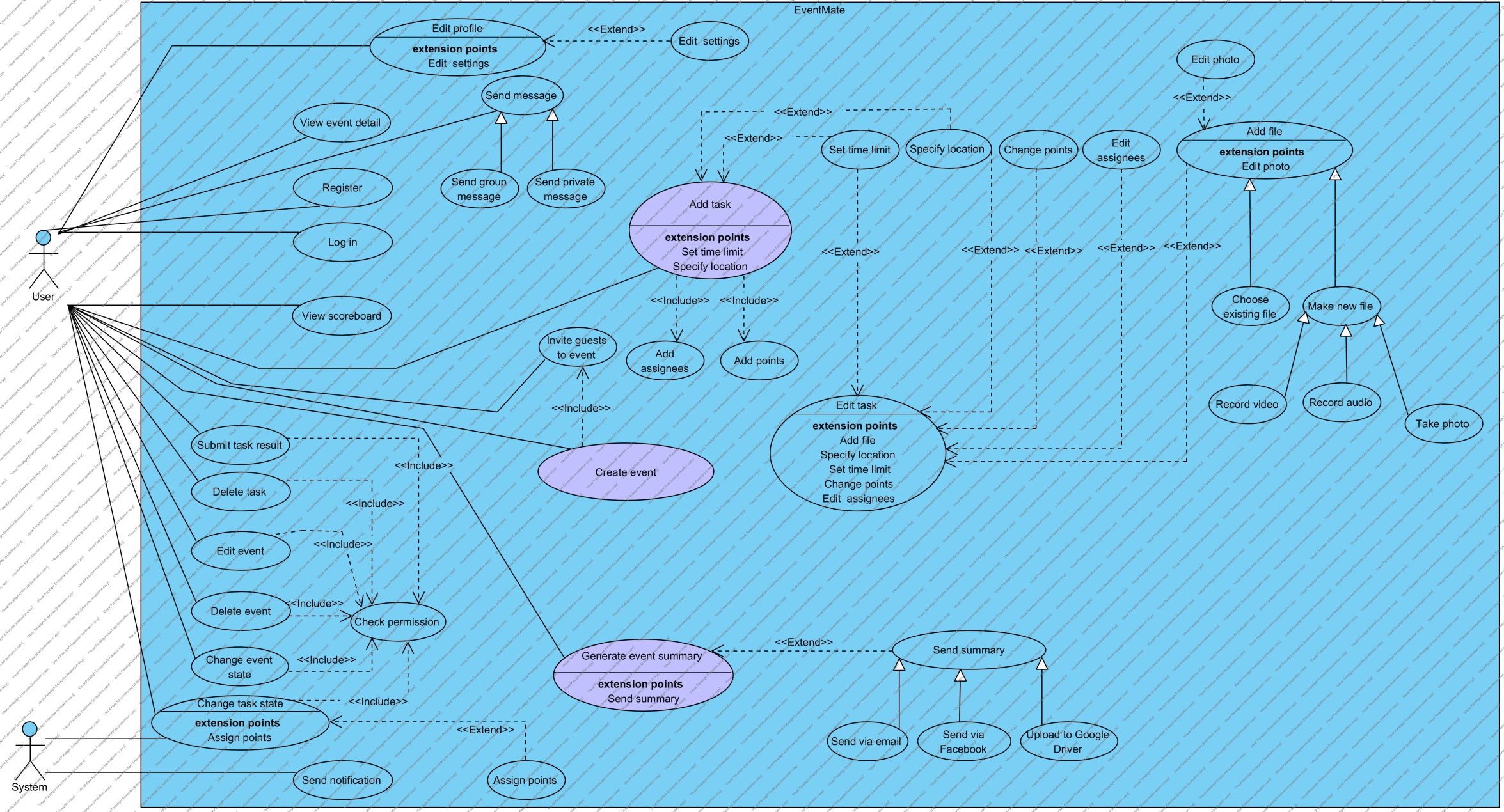
* Architecture diagram

Prezentace

* + - 1. Slide – mockup
      2. Slide – req. diagram
      3. Slide – use case diagram

## Attachments

## Attachment 1 – Use case diagram



## Attachment 2 – Mindmap

