

Software Requirements Specification



**Prepared by KeepMeAware
for the project {KeepMeAware}**

METU - Department of Computer Engineering
CENG 491 Senior Design Project I
Fall 2015-2016

1. Introduction

This document is a software requirement specification for Keepmeaware senior project. The purpose is to develop application where people can come for free information, which will help them to advance in their ordinary, professional and academic lives and help them to save time and enhance their focus on more important things. In this SRS we are going to give information about purpose and scope of this document. In addition to this use cases, data models, and behavioral models will be specified.

1.1 Problem Definition

It's hard to be on right foot with pace of life, does not matter what is your occupation, we may face difficulties with arranging and planning. The problem that we have noticed as students of Middle East Technical University is that most of students in order to be aware of our academic schedule, homeworks, quizzes, exam and lecture times we have to permanently check and recheck our emails and so on. By checking and rechecking we waste lot of time nonetheless miss important dates. Moreover, a lot of interesting events are held in our university but we cannot attend them because we do not usually see all the advertisements or sometimes forget about events we wanted to attend. Problems we aim to solve are do not miss important dates and activities, be aware of activities and social events going around, explore and even create them, save time of students from redundant checking of their mail and schedule, help event organizers reach their audience more effectively in university or anywhere else.

1.2 System Overview

Purpose

The purpose of this document is to give a detailed description of the project. By giving details of the architecture, functionalities and specifications, it will explain the purpose and features of the system that will be developed. This document is going to serve as a guideline for both the development team and the users. Since we use diverse frameworks and also develop our app both in mobile side and desktop side it will be release more than one version of this document as there may be modifications to adapt changes of requirements and specifications of the project and even may include functionality changes in the future.

Briefly, in our application we will have several user interfaces, which will be connected between each other to create user-friendly and functionally responsive application, with help of which users will be able to create and share their schedules, add publicly available schedules of others users to own scheduler and be on track and notified based on settings and Smart Notification System tool. Users also will be able to view events around given location and be on track with events users are interested in. For our Event Organizers(EO) we will provide

statistics varieties with help of which we believe EO will be able to boost their event participation.

Scope

The project that is stated in this document, is aims to describe mainly working principle of schedule and notification mechanism of our project. A user can register both from mobile and from website of our application. After registration the user should pass verification process with email. After starting usage of application the user can create his/her own schedule, add someone's schedule to his own schedule, view events from events page, add preferable event to his schedule in order to be notified time to time. All data like events, schedules, users will be kept on database of system.

1.3 Definitions, acronyms, and abbreviations

Terms	Definitions
SRS NodeJS AngularJS	Software requirements specification
MongoDB	NoSQL database for document storing, retrieval
Ionic DigitalOcean HTML	
CSS	
Use Case Diagram	Use case diagram shows users' interaction with system
Class Diagram	A special diagram that presents classes of a system, attributes of the classes, methods of the classes and the relationships between objects.
KMA	KeepMeAware
EO KMA Admin Panel	Event Organizer

1.4 Assumptions and dependencies

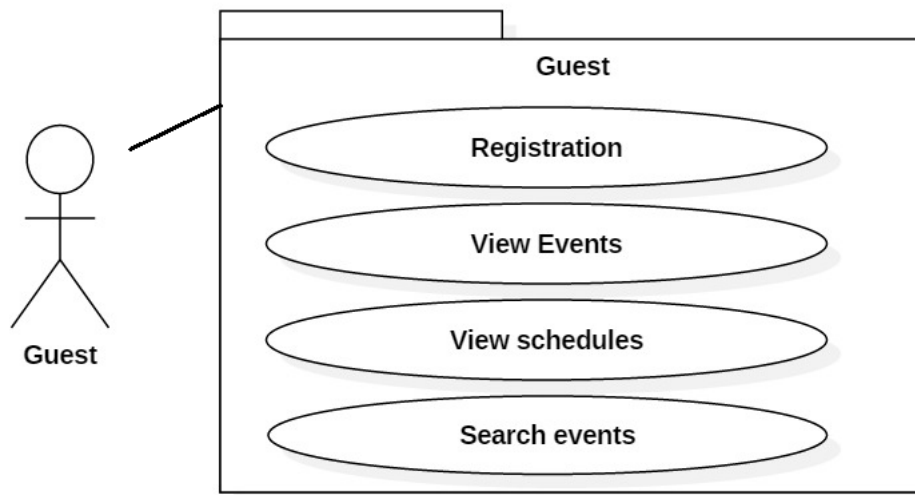
Basic computer and telephone usage skills are required. A user should be able to use a web browser or use mobile phone with Android or IOS operation system. In addition, notions of login, register, and account verification with email are also required.

2. Overall description

The purpose of this part of the document is to provide the general description of the KeepMeAware system and its requirements.

2.1 Product functions

2.1.1 Use-case model survey



1. Registration use case

Description: Registering a user to be able to use the system.

Detailed Description:

For registration a user should fill the boxes in registration page like:

- Email
- Login
- Name
- Password

Actors: Guest

Trigger: This use case is initiated when user clicks on Register button on the website

Basic Flow

Step 1: User opens the KeepMeAware homepage on the web browser.

Step 2: User clicks on button mentioned in Trigger tag and a popup menu is shown to the user.

Step 3: User fills the 'Login', 'Email', 'Name', and 'Password' fields.

Step 4: If a user with specified Login exists, then a 'user exists' notification will pop up.

2. View events use case

Description: Listing all public events in the system.

Detailed Description:

There are three types of events in the system: external, campus, and location.

In listing events by using these filters one can list specific events from the event pool.

There are two ways for listing events in the system. One is by clicking Events button in bottom right side of the screen of a mobile device.

Other one is by using browser in Desktop device. One just typing our system's webpage can reach all events in the home page of the website.

Basic Flow for Mobile

Step 1: User opens application on mobile device.

Step 2: User clicks Events button which in the bottom right side of the screen

Basic Flow for Desktop

Step 1: User opens browser in Desktop

Step 2: User types kma.com in URL field. All events will be listed after loading of webpage.

Actors: Guest, User, Event Organizer, Admin

3. View schedules use case

Description: Listing all public schedules in the system.

Detailed Description:

There is user created schedules those are pushed to the system's schedule pool. If schedule is public, then all guests of the website will be able to view it. If it is private only the creator of an event can see it.

Basic Flow

Step 1: User opens browser in Desktop

Step 2: User types kma.com in URL field.

Step 3: User clicks on 'Schedules' button, then he/she can view all of schedules.

Actors: Guest, User, Event Organizer, Admin

4. Search events use case

Description: Searching events both in desktop side and mobile side

Detailed Description:

Basic Flow for Desktop

Step 1: User opens browser in Desktop

Step 2: User types kma.com in URL field.

Step 3: User clicks on Search field, then he/she can search events with specific words.

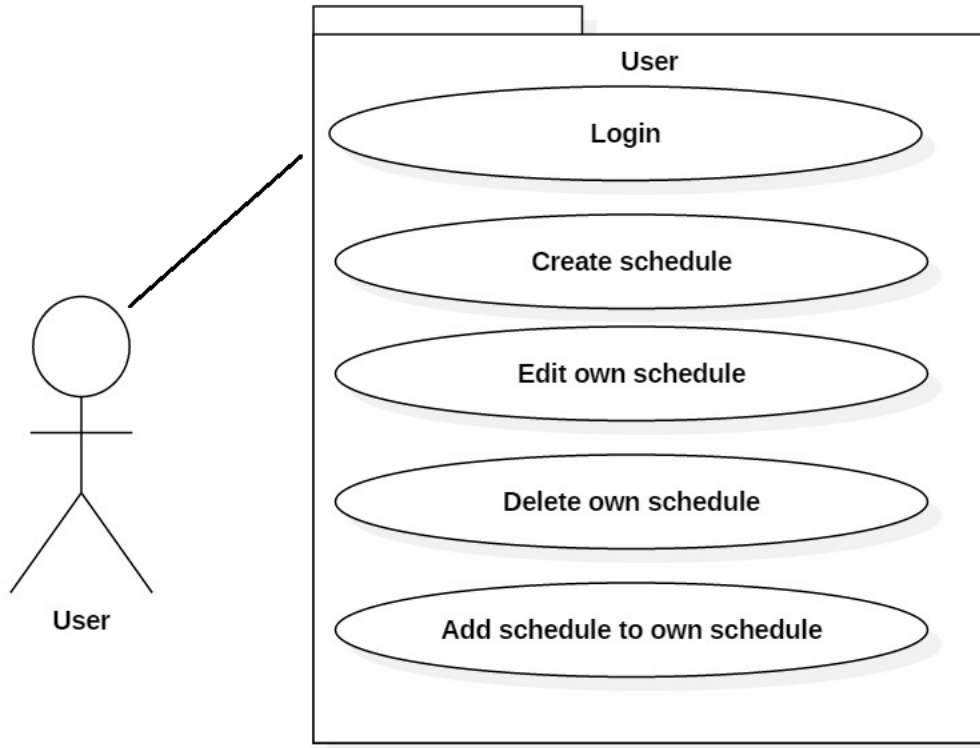
Basic Flow for Mobile

Step 1: User opens app in mobile device

Step 2: User clicks on Events button in bottom right part of the screen.

Step 3: User clicks on Search field, then he/she can search events with specific words.

Actors: Guest, User, Event Organizer, Admin



5. Login use case

Description: Log in a user to the system.

Detailed Description:

A guest after registration will become a user of the system. By logging in a user can use more functionalities than a general guest of the website.

Basic Flow

Step 1: User opens browser in Desktop

Step 2: User types kma.com in URL field.

Step 3: User clicks on 'Sign in button, then popup window will be opened.

Step 4: User enters login and password and clicks to submit button, after that the user will be logged in to the system.

6. Create schedule use case

Description: Creation of schedule

Detailed Description:

In order to create a schedule one should be signed in to the system. By clicking create new schedule in admin panel one should fill schedule name and description fields appropriately. By selecting Public checkbox, it can be visible to all other users of the system, otherwise it is private to the user himself/herself.

After creation of a new schedule the user will have a calendar view to it. By clicking to a specific day the user can fill it with data like title, date, color, description.

There are two types of schedules in the system. One is private other one is public. Private ones are only private of author himself, i.e. only author can view them. Public ones are open to everyone. Anyone can add public schedules to his schedule.

Basic Flow

Step 1: User signs in to the system

Step 2: User opens admin panel of his account.

Step 3: User clicks on 'Create new schedule' button, then popup window will be opened.

Step 4: User enters schedule name and description and clicks to create button, after that new schedule will be added to his/her account.

Actors: User, Event Organizer, Admin

7. Edit own schedule use case

Description: Edition of schedule

Detailed Description:

After creation of a schedule after some time may need to edit and make some changes on a schedule. After clicking 'Edit schedule' in admin panel a user will be redirected to page where fields filled with schedule's previous data. By doing some modifications on schedule any user can edit own schedule and clicking on submit the user sent it to database in order to save it.

Actors: User, Event Organizer, Admin

8. Delete own schedule use case

Description: Deletion of own schedule

Detailed Description:

Anyone can delete own schedule any time.

Actors: User, Event Organizer, Admin

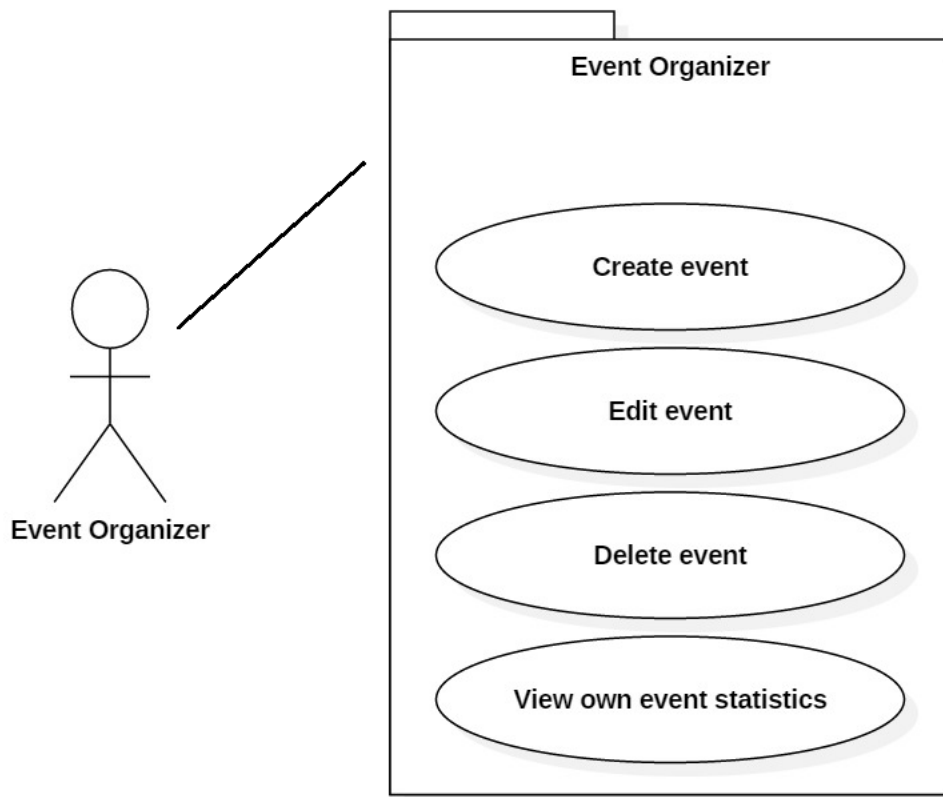
9. Add schedule to own schedule use case

Description: Adding specific public schedule to own schedule

Detailed Description:

Any user can add any schedule from the schedules pool. By clicking on specific schedules add button a user can add time units defined in that specific schedule to his/her own schedule.

Actors: User, Event Organizer, Admin



10. Create event use case

Description: Creation of event and send it to approval of KMAadmin

Detailed Description:

For event creation a user should fill the boxes in create event page like:

- Title for event
- Choose Date from date picker
- Choose Color from color picker
- Upload a photo for event cover
- Write a description for an event

After creation of event it will be stored in system's database with additional information like author of event and setting published field with false. After admin approval this event will be shown in events page.

After being registered and signed in to the system a user can click Create Event button for event creation and sending it for admin approval. By creating a single event generic

user will become an Event Organizer who can edit and delete his events from Admin Panel.

Actors: Event Organizer, Admin

11. Edit event

Description: Edition of event from admin panel

Detailed Description:

If an event organizer wants to edit any field of his/her created event, event organizer should edit it and send it to admin for approval. Since it may include any inappropriate content to publish.

After clicking edit button of an event, a user will be redirected to create event like page with filled fields with the event data. After changing needed fields, the event organizer will send the event for approval to admin.

13. Delete event

Description: Deletion of event from admin panel

Detailed Description:

If an event organizer wants to delete his/her created event, event organizer should delete the event from admin panel where his/her events are listed. In this case there is no need of admin approval. Event organizer can delete his/her events any time without permission from admin.

Actors: Event Organizer, Admin

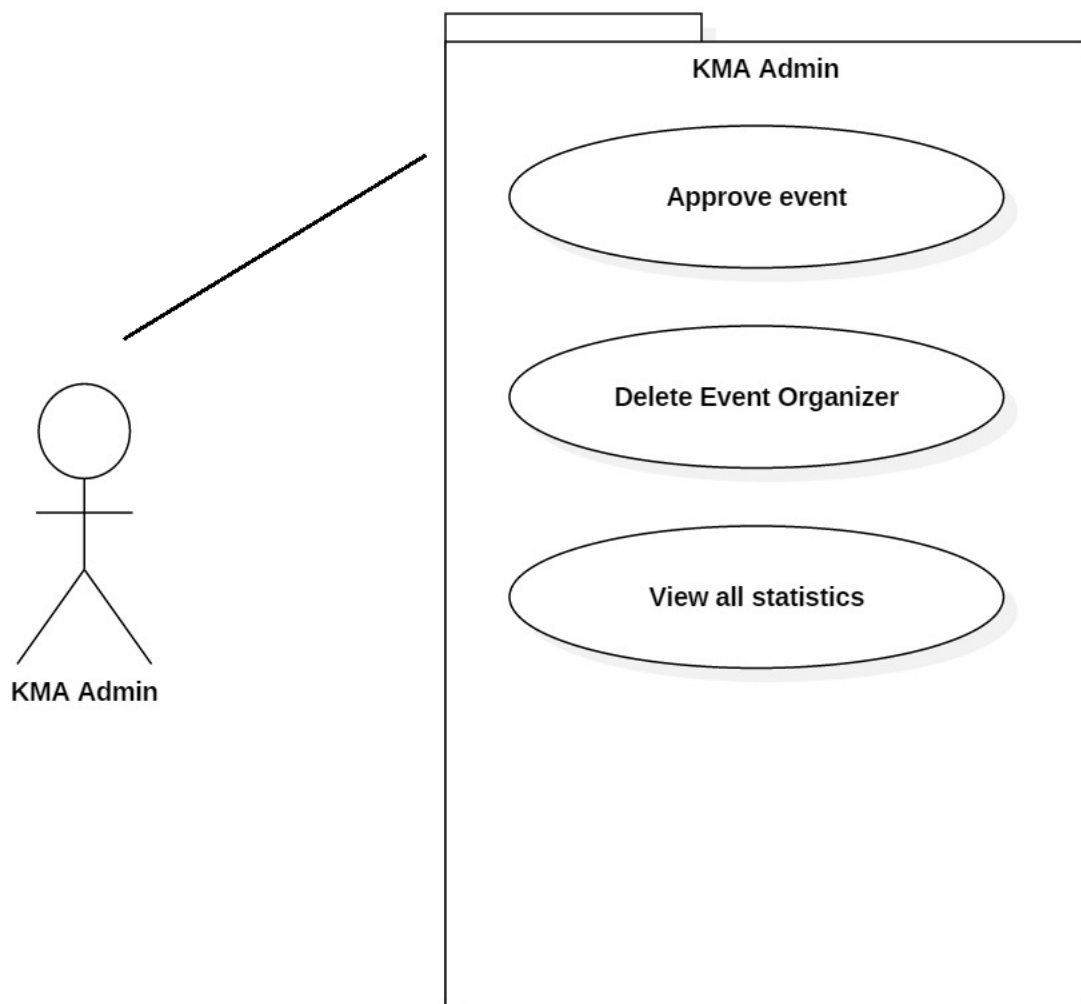
14. View own event statistics

Description: Viewing statistics related data of specific event

Detailed Description:

After creation of an event owner of event or KMAadmin can view statistics of that event. For example: how many users have subscribed to it, how many people viewed it and etc.

Actors: Event Organizer, Admin



15. Approve event use case

Description: Approving event organizer created event to publish it on events page.

Detailed Description:

After creation of event by event organizer it will be stored in database but it will be not shown on events page. Because of filtering inappropriate content, all new created events will be sent for approval to KMAAdmin. KMAAdmin analyses content data then approves for it for showing it in events page.

Actor: KMAAdmin

16. Delete Event Organizer use case

Description: Deleting an Event Organizer user from users in database.

Detailed Description:

KMAAdmin any time can delete any user from database any time.

Basic Flow

Step 1: KMAAdmin clicks on 'Event organizers' button in admin panel. After all Event Organizers will be listed in page

Step 2: KMAAdmin will click on specific Event Organizer's profile

Step 3: After clicking, Delete button will be shown. After popup window will be open asking 'Are you sure?' question.

Step 4: By clicking Yes Event Organizer will be deleted from the database.

Actor: KMAAdmin

17. View all statistics use case

Description: Viewing all statistics related data

Detailed Description:

Actor: KMAadmin

2.1.2 Actor survey

- **Guest**
Description: Guest is a user that only runs application on mobile device and views it or entering the website that is kma.com. He/she can only view events and view schedules, search events, and do registration in system both in mobile device and in desktop browser.
- **User**
Description: A user being guest has more functionalities than guest actor. A user has functionalities like login to the system, create/delete/edit schedules those are he/she owns, and also adding specific schedule from schedule pool to his/her own schedule in addition to guest privileges.
- **Event Organizer**
Description: An event organizer has both guest and user privileges. A user after creating his/her first event in the system, he/she will be automatically turned in an Event Organizer of the system. Extra functionalities of Event Organizer are create/delete/edit events and view own event statistics.
- **KMAadmin**
Description: The KMAadmin is the controller of the system. He/she can use all use cases which are defined in the system.

2.2 Interfaces

KMA application's interface structure divided in four different interfaces to provide clean, understandable and user-friendly UI. Initially at the start, user positioned with our KeepMeAware main page, where he/she can switch between different environments based on user's needs.

2.2.1 User Interfaces

In sake of easy usage and understanding, the UI will be intuitive and friendly, which means that all the buttons, panels and fields will be organized in similar way the most famous webpages (Facebook, for example) have. Therefore, user will be provided with the layout he/she has already seen and used. If this is not the case, the purpose of every part of the UI is clearly stated and understandable. There will be five different pages. These include: search page, profile page, edit page and a particular news page. In addition, the administrator's page is provided.

The characteristics of UIs:

- The UI will be in English.
- In case of an error, the relevant message will be displayed.
- All the interfaces are shown by the browser.
- Going to the previous page is implemented by the back button (browser's button).

However, all the previous results will be saved and displayed when going back.

- To define various search features dropdown lists containing check and text boxes will be used.

- In the search page, user will be able to select the different search criteria (most popular, latest, specific, etc.) and get the desired results.

As it was mentioned above, our application interfaces divided in four different interface structures:

2.2.1.1 User Main Activity View Interface

UI where user will be able to add/modify/delete/update schedule, also will be able to view/add events. With help of Smart Notification System users will be notified and will be on track with their schedule and events.

2.2.1.2 Create Schedule Module Interface

On the particular UI, user will be able to create schedules with varieties of options. User will be able to set schedule sharing settings as private, public or with invite id settings. Also user will be able to view other publicly accessible schedule modules, and also will be able to add it to his own.

2.2.1.3 Event Activity Interface

In this particular UI users will be able to view and search all publicly available events and add it to own event list, other than that Event Organizers will be able to create events and view statistics based on data collected.

2.2.1.4 KMA Admin Panel

This UI created for controlling whole system. Publically shared data firstly must be approved by KMA Admins before it will be able for other users to see. KMA Admin will also able to access different variation of statistics. Those are: user, event and schedule module statistics.

2.2.2 Hardware Interfaces

The hardware interface is host platforms and for the mobile interfaces: Android and iOS.

2.2.3 Software Interfaces

	Software Product & Version	Source
Client using Internet from PC	Web Browser, OS	Any
Database Server	MongoDB	http://www.mongodb.org/
Development	HTML5, CSS3, JS, JSON, Bootstrap	Latest & Stable Version
End (RAD)		
Server side programming	NodeJS (version 5.4)	http://www.nodejs.org/
Client side programming	AngularJS (version 2.0)	http://www.angularjs.org/
real-time, bi-directional communication between web server and client	Socket.io	

2.2.4 Communications Interfaces

The communication between the KeepMeAware and users will be provided by TCP/IP protocol by using internet with mobile or desktop devices.

2.3 Constraints

3. Specific requirements

3.1 Nonfunctional Requirements

The only nonfunctional requirement of our system can be the Manual Event/ Schedule Validation/Approval. This operation requires a team of people and is usually done by a KMA Admins, who check all the event's/schedule's fields in order not to publish the inappropriate ones.

3.2.1 Usability

Easy use to all users who uses this product and understandable with basic icon buttons.

Product is real time and system is not complex.

3.2.2 Reliability

Availability: system will be available for 24 x 7 at 100%.

Mean time to repair: it depends how big is problem. At most in 3-4 hours will required to solve problem.

The system provides storage of all databases on redundant computers with automatic switchover.

The reliability of the overall program depends on the reliability of the separate components. The main pillar of reliability of the system is the backup of the database which is continuously maintained and updated to reflect the most recent changes. Thus the overall stability of the system depends on the stability of container and its underlying operating system.

3.2.3 Performance

Response time for a transaction: Any interface between a user and the system shall have a maximum response time of 2 seconds.

Throughput: it depends on end-user internet connection.

Capacity: >1000,000

Degradation modes: there will be reference pages when system degraded.

Resource utilization: main utilization is getting millions of data from database. There will be used indexing to utilize performance.

3.2.4 Supportability

A commercial database is used for maintaining the database and the application server takes care of the site. In case of a failure, a re-initialization of the program will be done. Also the software design is being done with modularity in mind so that maintainability can be done efficiently.

4 Data Model and Description

4.1 ER Diagram

Let us start this section with the ER Diagram of the KMA database. This is the essential part of the whole system.

ER diagram consists of 4 main parts: **User, Event, TimeUnit, Schedule.**

1. **User:** stores all the data required to freely manipulate user objects.

Note: one additional field can be added - **role [String]**, which represents user's possible role according to figure below.

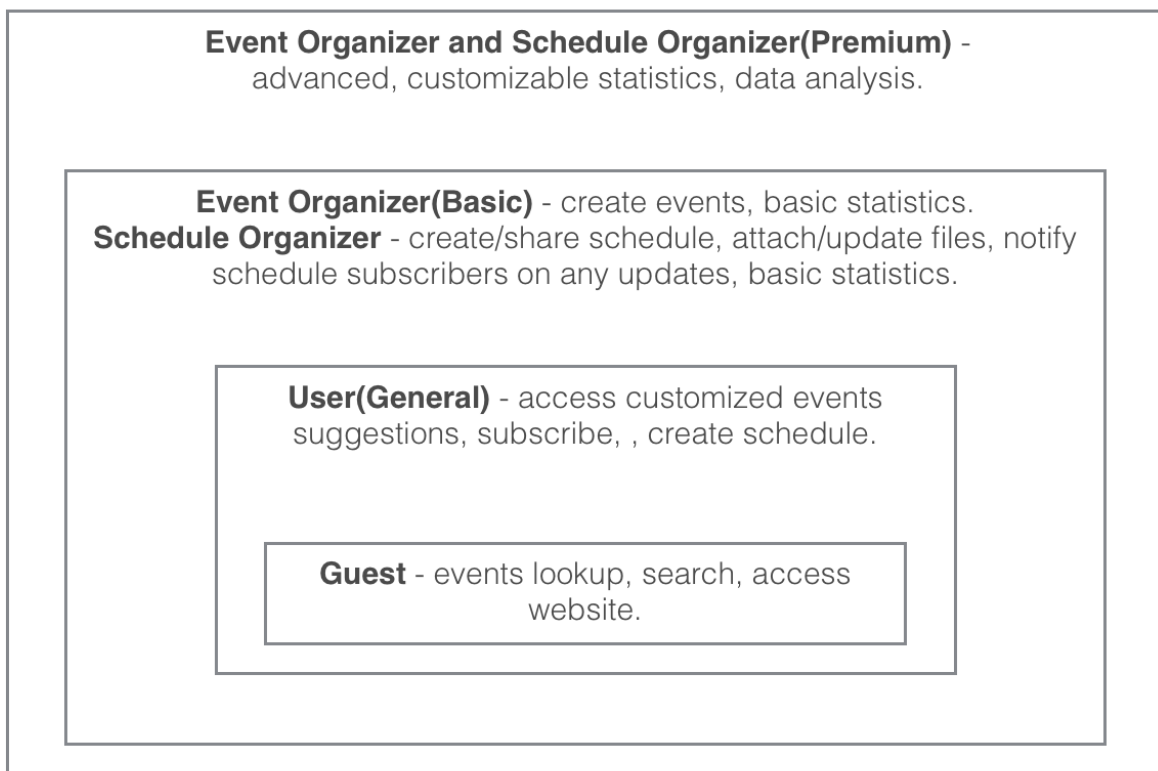


Figure 4.1 - KMA User hierarchy

Each box represents separate functionality granted to every **User level**.

For example, on page entry, if user is **Guest** (not yet registered in the system) he/she is capable of events search, schedule lookup, etc. If user shifts from **Guest** to **User(General)**, i.e registers in the system, then he/she gains all the functionality provided by this level. The discussed logic applies to all different user levels up to **Premium**, which opens up the whole system functionality.

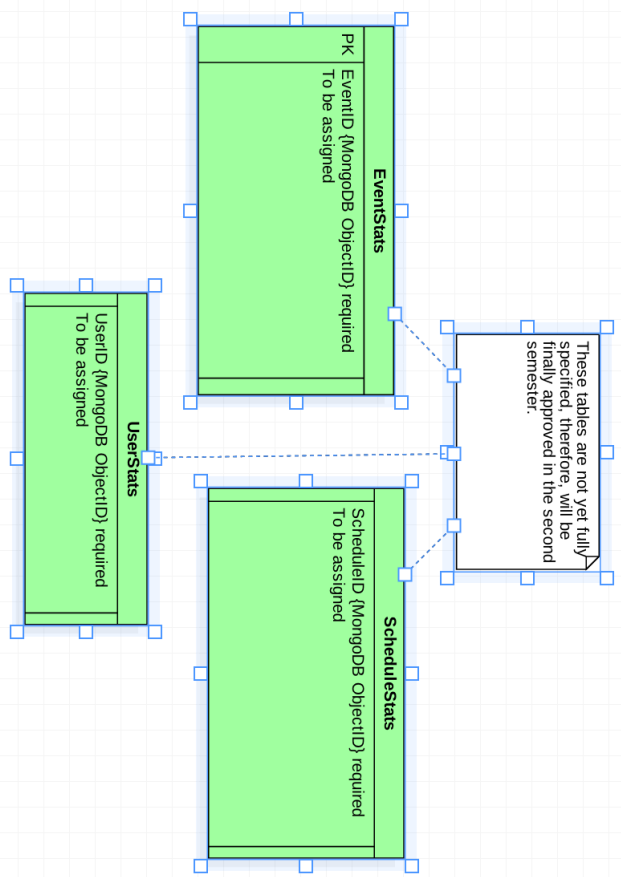
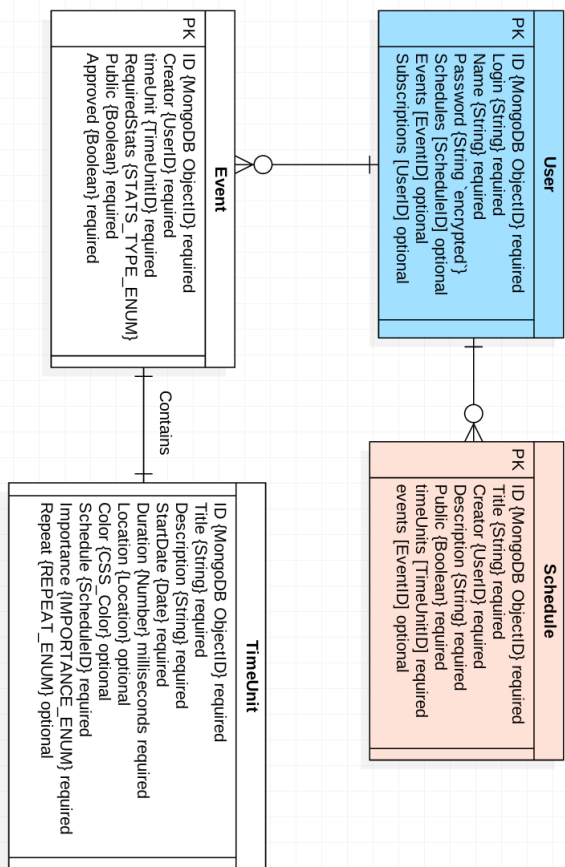


Figure 4.2 - ER diagram

This strategy eases the validation and permissions check implementation, since follows very simple rules.

Important note here is that the final User Level, which is **KMA Admin**, is not shown in the figure 4.1 simple because this user level cannot be transitioned to by a regular user. All **KMA Admins** are unique users responsible for the sustainability and reliability of the system. Therefore, each new **KMA Admin** can be created only by another existing admin.

2. Event: is a fundamental part of the whole project. Events are basically moments in time that different users can save to his/her schedule. After doing so, the system will send notification in advance according to user-defined settings.

3. TimeUnit: is another crucial part of the application. This is the base for both Schedule and Event objects.

4. Schedule: a collection of Events and TimeUnits that are organized in meaningful and representable way. Users can share/update schedule, invite other users to schedule and define authentication types, so that only the limited amount of people can view and save it to their accounts.

4.2 Objects

This section specifies all the major classes used in the system and their relationships.

Brief explanation:

1. User Not Registered: users that are not yet saved to our system.

2. User Registered: user already registered in the system with all the necessary permissions, fields and functionality set.

3. KMA Admin: projects main Admin Object, responsible for events/schedules approval and overall system maintenance, a child of General User Class.

4. Event Organizer/Schedule Organizer: two identical objects that inherit basic user functionality plus the ability to create/remove/update both Events and Schedules.

5. Schedule: object representation of the systems fundamental entity. Already discussed in ER Diagram section.

6. Schedule Organizer/Event Controller: classes responsible for observing and monitoring both **Events** and **Schedules** created by users. All the **Event** and **Schedule** functionalities are actually delegated to this objects.

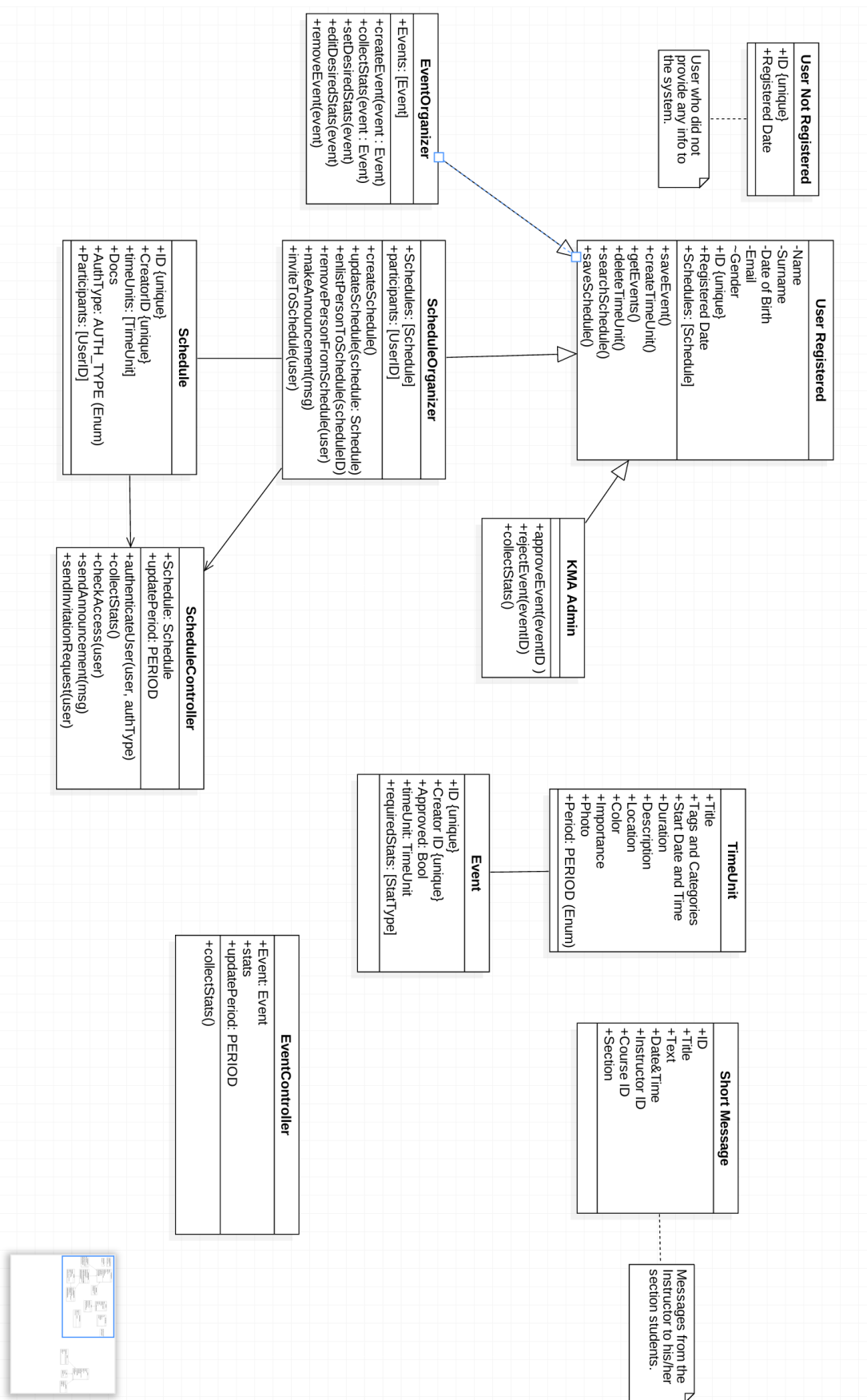


Figure 4.3 - Class Diagram

7. TimeUnit & Event: class representations of entities discussed in **ER Diagrams** section.

8. Short Message: basic communication unit for users.

4.3 Event/ Public Schedule Validation flow

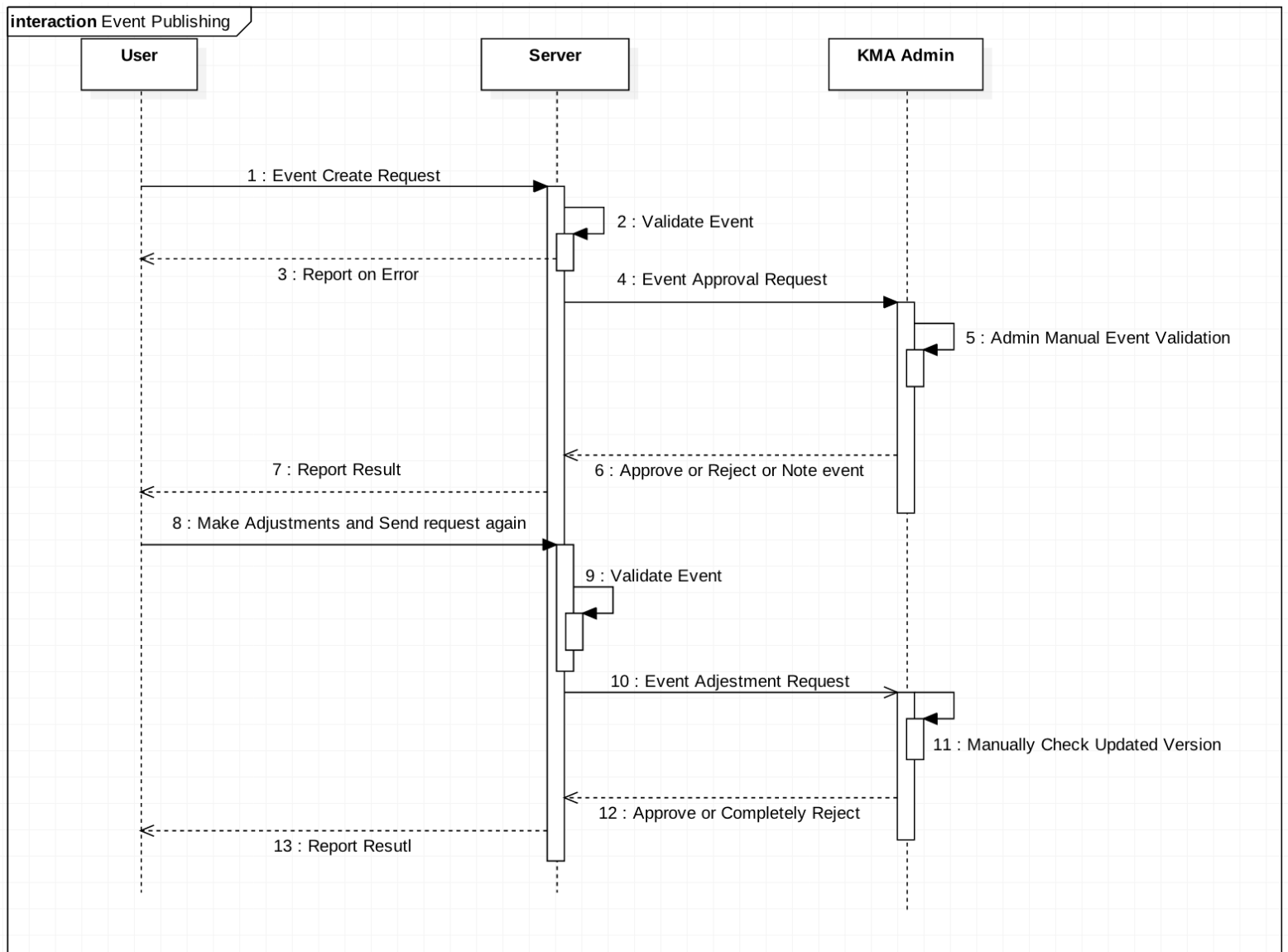


Figure 4.3 - Event/Schedule Validation Flow

This sequence diagram shows the basic validation of every event or public schedule created by the users of our system.

Firstly, the request goes to our server, where automatic validation is performed. Further, the server saves the **Event/Schedule** object to the database so that KMA Admin will see it as '**new**' and be informed that this event/schedule needs manual validation.

After above steps complete, KMA Admin can send note to the user indicating that some of the parts need update, otherwise, it approves the Event/Schedule and it becomes globally visible. If the user correctly updates his event/schedule according to Admins feedback, it is saved to the database and the process ends. If even after the update this Event/Schedule is considered to be inappropriate, Admin removes it completely and this user will have to create a new one instead.

5 References

- [1] IEEE Guide for Software Requirements Specifications," in IEEE Std 830-1984 , vol., no., pp.1-26, Feb. 10 1984, doi: 10.1109/IEEESTD.1984.119205,
URL: <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=278253&isnumber=6883>
- [2] Appendix C of Don Widrig, Dean Leffingwell, "Managing Software Requirements: A Unified Approach," Addison-Wesley Professional, Release Date: October 1999, ISBN: 0201615932.