RASD

MeteoCal

Software Engineering 2 Project 2014-2015

Nico Mkhatvari

Jia Sheng Xia

Table of Contents

1 Introduction

1.1 Problem description

The main scope of the project is to create a web application using JavaEE platform which informs the registered users about the incoming events, helping them to avoid bad weather conditions in case of outdoors activities.

This web application will permit to the registered users to create, delete, update and invite people to the personal events. The calendar permits to the creator of the event to invite other registered users at any time, where the latter can only accept or decline the invitation.

In fact the software will allow to organizer to specify more details about the future event like: place, date and whether it will be indoor or outdoor appointment. Furthermore, each event has also private or public visibility. Public events details can be consulted by any registered users whereas private events are visualized with busy status where all details are disclosed only to the private sphere.

Once the event has been committed every participating user can retrieve information about forecast condition and other important details.

The system will notify the organizer one day before about incoming bad weather, recommending him another appropriate day for the outdoor event

1.2 Glossary

In order to avoid some lexicon's ambiguities, throughout the documentation will use the following definitions:

User:
a person already registered in the system who is able view notifications, create new events, reply to invitations, view details of own and public events.

Organizer:
is a registered user who arranges an activity, which is the only one who can update and delete an event. Also is known as owner.

Event:
is a planned activity present in the system with a specific date and place and other details.

Public:
all events which information are accessible to every registered user.

Private:
information is available only to selected members.

Indoor:
all activities happening inside a building or under a covered structure.

Outdoor:
all activities existing, happening, or done outside, rather than inside a building.

Invitation:
a notification with a specific request to join an event that can be either accepted or declined.

Notification:
is an official message sent by the system to the registered user.

Registration:
is the procedure that permit to an user to be acknowledged by the system.

1.3 General specifications of the project

In order to to guarantee the main functions here below are listed the most ones:

- The system permits only to the owner of the event the creation, deletion, update of the schedule;
- Each registered user can create an event;
- The system is able to send invitations and receive responses from the users;
- The organizer has the complete list of the users that has committed to the event;
- In case of incoming bad weather conditions the system has to notify the forecast situation for outdoor events one day before;
- In case of incoming bad weather conditions the system has to suggest to the creator the closest sunny day (if any) three days before;
- For private events only invited registers can view the details of the event;
- The system provides the forecast to user's upcoming events;
- The system has to guarantee access only to the registered users;
- Only when users are logged in can view own notifications and personal schedule;
- Registered users can see other users' schedule if the calendar is set as public, check when they are busy and, in case of public event, to see the details of the event;

1.4 Assumptions

The initial document of the project presents some not well defined points. In order to avoid ambiguities and misunderstandings in the project we will take some assumptions. The crucial points from our point of view are:

Bad weather condition:
in nature there is a large set of bad weather conditions, therefore for humans it is a subjective point of view, i.e. something related to their planned activities, mood, or other sort of factors. Considering these premises the application shall provide to user when creating an event the opportunity to define what bad weather conditions are.

One day before:
usually for people this concept is perceived as 24 hours before a given event, this concept will be used in this project as well.

Three days before:
we consider this elapse of time as 72 hours to the event's beginning.

Closest sunny day:
we intend such a day that is the closest date with the sunshine and comes after the event's day.

Notification :
are sent immediately after the event creation.

Place:
to avoid ambiguities in this case the owner will provide unique identification of the event's location, e.g. zip code.

Modification of the event:
in case of one of important fields have been modified (i.e. location or date), the notification shall be sent to all originally participants.

Overlapped events:
an user can create an event only if he isn't participating to any other scheduled event at that lapse of time, i.e. creation only of synchronously disjointed events are permitted.

Acceptance of the event:
once an invitation has been either accepted or declined. If a user doesn't reply to the invitation, it will be automatically set as declined.

Calendar:
Gregorian.

Visibility of the calendar:
the user beside own calendar can view other users' calendars only when they are set as public. Calendar's public visibility let to other users to view upcoming events and event their details if the last set to public as well. Contrarily if the calendar is set as private, is inaccessible to other users and furthermore its events can be consulted only by the owner.

2 Requirements

2.1 Stakeholders

The main stakeholders of the project is Politecnico of Milan which is represented by internal committee of professors. The main needs of this project is to fulfill all requested requirements respecting all deadlines. Furthermore we have to exhibit all stages of the project throughout of the development process.

The other stakeholder is our development team which is focused on (is really into/ be concerned) delivering a high-quality web application. In addition we are concerned with acquiring main soft engineering techniques. Besides what has been stated before, our team is interested especially to get a good mark for the university career.

Finally the main intention is to provide an ease-to-use software, that will accommodate user experience and fulfill all requested needs.

2.2 Actors related to the project

In this system under consideration, there are basically three main actors:

Registered user:
a person acknowledged by the system, has access to all application's features such as creation, deletion, update events, check up other users' schedules and signed up arrangements with their relative weather forecast.

Weather forecast:
an external service which provides an analysis of the state of the weather in certain area and date.

The web application:
is the core of the whole project which makes possible the interaction with calendar.

2.3 Functional requirements

User	System
○ Each registered user can create an event and view personal notifications, invitations and schedule. ○ Each registered user can access to other users' calendar, see details of events if they're public. ○ Each user can see detail of events on other public calendars; ○ Each user can access to event's summary on public calendars; ○ Registered user can decline or accept the invitation before its deadline, once expired no possible actions can be performed; ○ Registered user can change his mind about participation to an event before the deadline. ○ During the creation of an event the user has to fill in mandatory information fields, otherwise it won't be possible to proceed with the creation; ○ User can choose privacy settings for own calendar and events. ○ Organizer defines the meaning of “bad weather”; ○ The organizer can invite registered users for own event. ○ Updates and cancellation of event can be performed by the organizer at any time. ○ Only the organizer is allowed to add more users to the event. ○ The member of the event can view	○ Any interaction with the calendar, is possible only when user is logged into the system; ○ All calendar events are based on Gregorian calendar and on local time. ○ Any created event has to be acknowledged as “committed event” by the system otherwise the user has to repeat the creation procedure or fulfill missing requirements. ○ Private calendar is hidden or inaccessible to public except its owner. ○ The system doesn't allow any creation of a new event if it is overlapped with an event previously created by the same organizer. ○ Each event has a summary log at least with essential details: weather forecast, date, place, name of activity, and its type (i.e. outdoor or indoor); ○ Once the event has been committed the invitations are automatically sent to all involved members. ○ Invitations contain all details about the event; ○ If any modifications of any location field or any date field are submitted, new invitations are sent to users who have previously declined the participation; ○ If any modifications of any location field or any date field are submitted, notifications are sent to users who have previously accepted the participation;

User	System
event summary; ○ The user can accept the invitation to if the current event which overlaps with another one accepted before; ○ The organizer can shift only forward already planned event by him.	○ Each event has a deadline to reply to an invitations; ○ All expired invitations are considered as declined. ○ All expired events are eliminated from the database; ○ The system is synchronized to a weather forecast system; ○ Weather forecast for next 14 days is considered trustworthy otherwise is not available; ○ In case of upcoming bad weather conditions a reminder is sent to all participants in outdoor activities one day before. ○ In case of upcoming bad weather conditions the system suggest the closest sunny day (if any) for the outdoor events.

2.4 Non functional requirement

2.4.1 Product Functions

As has been stated before our web application is conceived to be able to allow people to see both personal and others users' calendar.

Each new user shall to register through a very simple registration form, in order to be acknowledged by the system. Every user, after logging into the application, will be able to:

- In first place to interact with the personal calendar creating, updating or deleting their personal event through a simple user interface;
- Retrieve information about forecast conditions in the period of incoming event;
- See users' schedules;
- Receive and create invitations;
- View weather forecast;

- Link easily other people to the programmed events.

2.4.2 Usability and user interfaces

- minimal
- user-friendly
- universal accepted symbols for each function
- hints on the icons
- guided procedures
- login page, calendar main (only preview), notifications, upcoming events (shows details)

To improve the usability of the software we will implement a very simple and clear user interface that allows the user to easily manage all the functionalities.

The structure of the application will be at the same time both complete and essential.

In the initial page will appear a log in interface. After log in the user will be able to see the notification area in a lateral bar where will appear a section with invitation to events, accepted invitation and user's event and a section where will appear system messages. In the main section a big calendar where the user can interact with all the events.

2.4.3 Reliability and Security

Since we are dealing with an academic project and our customer hasn't specified any particular request about reliability, we will assume that:

- The project runs on reliable hardware therefore any hardware faults can’t happen;
- The project deals in harmless environment so no cyber attacks can occur;
- There aren't any catastrophic scenarios that may damage the environment with irreversible consequences.

To ensure a better reliability of personal information each user has to authenticate himself with an username and password to access to the application functionalities.

Furthermore, every information it's stored in a secure server, so personal sensitive data are safe from external attack to user's personal computer.

2.4.4 Performance

No specifications has been given to this point as well, but still some assumptions with common sense can’t be ignored:

- The response time of functions should not annoy the user;
- Should support most of common browsers;
- Can be accessible from low-medium latency connections.

3 Possible scenarios

Title	*Registration of a new user*
Goal	*Create shared events and/or program personal schedule.*
Assumptions	*Person is not yet registered in the system.*

Scenario

- Pinco has to organize a cocktail party with some old friends coming from Spain;
- Pinco is told that there is a really good application called MeteoCal that allows him to invite his friends to events;
- Enters into the site;
- sees the welcome banner with a short description;
- Under the banner there are two tabs: Sign up and Log in;
- Clicks on Sign up;
- Redirected to registration page;
- Fills 3 fields:
    - email;
    - password: system requires at least 6 chars and no blank chars;
    - reenter password;
- Clicks on submit;
- System prompts about succeeded registration.

Title	*Password recovery*
Goal	*The user has forgotten own password therefore wants to restore it.*
Assumptions	*[A1] User is already registered in the system;* *The registered user's email is his default one and is still active;* *Password is stored in DB.*

Scenario

- Enters into the site;
- Sees the welcome banner with a short description;
- Under the banner there are two tabs: Sign up and Log in;
- Clicks on Log in;
- Redirected to login page;
- Enters own credentials, but the system doesn't accept;
- System prompts “not existing user with this email”;
- Reinsert again the credentials, but now system prompts “Password is wrong”;
- After several attempts, decides to click on “Forgot your password?”
- Fills email field;
- Clicks on submit;
- System prompts “The email with your password has been sent”.

Title	*Create a new event.*
Goal	*The organizer can easily create a new event and link other users to it through an invitation.*
Assumptions	*[A1];* *[A2] Invitation can be only sent to registered members;* *Organizer is a member of the event by default;* *The organizer hasn't created any event yet.*

Scenario

- Once entered into the site user desires to create an event for a cocktail party;
- Click on add;
- Fill in ALL fields;
- Adds emails to the invitation list;
- Last entered email is not a registered in database;
- The unknown email will receive a simple message from the organizer with simple detail about party. This mail suggests to sign up on the site to get access all the features;
- The organizer successfully submits the new event.

Title	*Update of already existing event.*
Goal	*The organizer is willing to add new users and modify some data about the event.*
Assumptions	*[A1] [A2];* *[A3] The event is already present in the system and is consistent;* *Main data about event are present in the system;* *Some invitations have already been sent.*

Scenario

- The user realized a missing person in the invitation list;
- Desires to add the missing friend to the event;
- Logged into the system;
- Select the created event from the “Organized events” box;
- Once the event is selected, the user clicks on gear icon in order to update some info;
- In the update page, which is already filled with existing event's information, user add a new email to the invitation list;
- Afterwards he changes his mind about the location of the event and makes some modifications;
- Submits the modifications;
- To new members and members who declined the invitation will be sent an invitation;
- For members who already accepted the previous invitation a notification will be sent.

Title	*Browsing in the web application*
Goal	*Each user is able to decline or accept incoming invitations, view details of programmed schedules.*
Assumptions	*[A1] [A3];* *There are present some notifications and invitations in the inbox;* *The user has organized at least one event;* *The user has accepted the participation to some events;* *The user knows the emails of other registered users.*

Scenario

- Decide to take a look at the incoming events by scrolling the bar on the right side of the main page;
- Afterwards rechecks the preview of the organized events by scrolling the cursor up and down;
- The user wonders about the incoming arrangements one of his friends;
- The user is redirected to friends calendar by typing his email in the search bar;
- When the user has accomplished all wanted operations, he logs out the system.

Title	*Creation of a consistent event.*
Goal	*The user is willing to create a new event.*
Assumptions	*[A1]* .

Scenario

- After the registration, the user wants to create a new event in order to fill the calendar with his holidays;
- Logs into the site with own credentials;
- Clicks on add button and enters in creation of a new event page;
- Fills all the fields of the page, without inviting any person;
- Click on submit, immediately after the system prompts the message: “Invalid Event – all fields are required”;
- Notices that is missing the zip code in location fields, so adds it as required;
- Clicks again on submit, the system prompts another message: “Invalid end date – the end date has to be after the begin date”;
- Notices that the start date is set to “24/12/14” and the end date is set to “07/01/14”, so the user corrects the date field;
- This time, finally, he submits to the system the event successfully.

Title	*Overlapped events.*
Goal	*The user is willing to create a new event. Participating in different events.*
Assumptions	*[A1]*; *The user has already created at least an event*; *The user accepted some incoming programmed events.*

Scenario

- The user has been invited from one of his best friends to an event where he can't be missing;
- The user checks out own schedule and finds out that one event overlaps with the party of his friend;
- Luckily, seems that both events are organized near to each other so he accepts to come by his friend's party after all;
- The week after the same friend invites the user to accompany to a certain event;
- The user tries to accept the invitation but the system prompts him that in that lapse of time there is another upcoming event where he is the organizer;
- The user recalls about the organized event and decided to decline the invitation friend;

Title	*Cancel a participation to an event*
Goal	*User can change own mind at any time about participation.*
Assumptions	*[A1];* *User has already accepted at least one invitation;* *The weather doesn't have any influence on the user's decision.*

Scenario

- Realizes to have the influence, so he changes his mind about the participation to the upcoming event;
- Logs into the site, with own credentials;
- Selects the event in the upcoming event box;
- After clicks on the highlighted info icon;
- The user has been redirected on the page of the event, with all the details;
- Clicks on “Abort the participation” bottom;
- The System prompts the message: “Are you sure that you want to withdraw your participation to the event?”;
- Clicks on “Yes” and returns on own calendar page;
- Observes that the event is disappeared both from the calendar and the upcoming event section;

Title	*Upgrade of visibility.*
Goal	*The organizer is desiring to upgrade an event from private to public.*
Assumptions	*[A1];The user has already created an event;* *The event is set as private;* *The event has multiple date.*

Scenario

- The user has organized a conference;
- Invited some people;
- This conference aroused the interest of people and some of them passed the word to other registered users;
- The users interested in the event can't view its details, because such has been set as private;
- The only available data to the user are the organizer's author and the date of the event;
- Some users decide to send a message to the organizer for further details;
- The organizer wonders about receiving all that amount of email asking about details of the event;
- Takes a close look at event's details;
- The organizer notices that such event is set as private;
- Modifies the visibility property of the event and immediately after submits it;
- The following conferences have registered the increase of public attendance.

Title	*Interaction with notifications and invitations*
Goal	*The organizer is desiring to view the presence of weather notifications and invitations to new events.*
Assumptions	*[A1];* *The user has already created an event.*

Scenario

- Decide to review some details of the incoming events.
- Notices that the inbox icon is highlighted so decidesi to check it out;
- One notification informs about some updates to the “Cocktail Party by pinco@email.com on 29/02/16” event;
- Due to the modifications of the event, the user is unable to rich the new location therefore declines the invitation;
- The user considers the details of the invitations and decides to accept some of them;

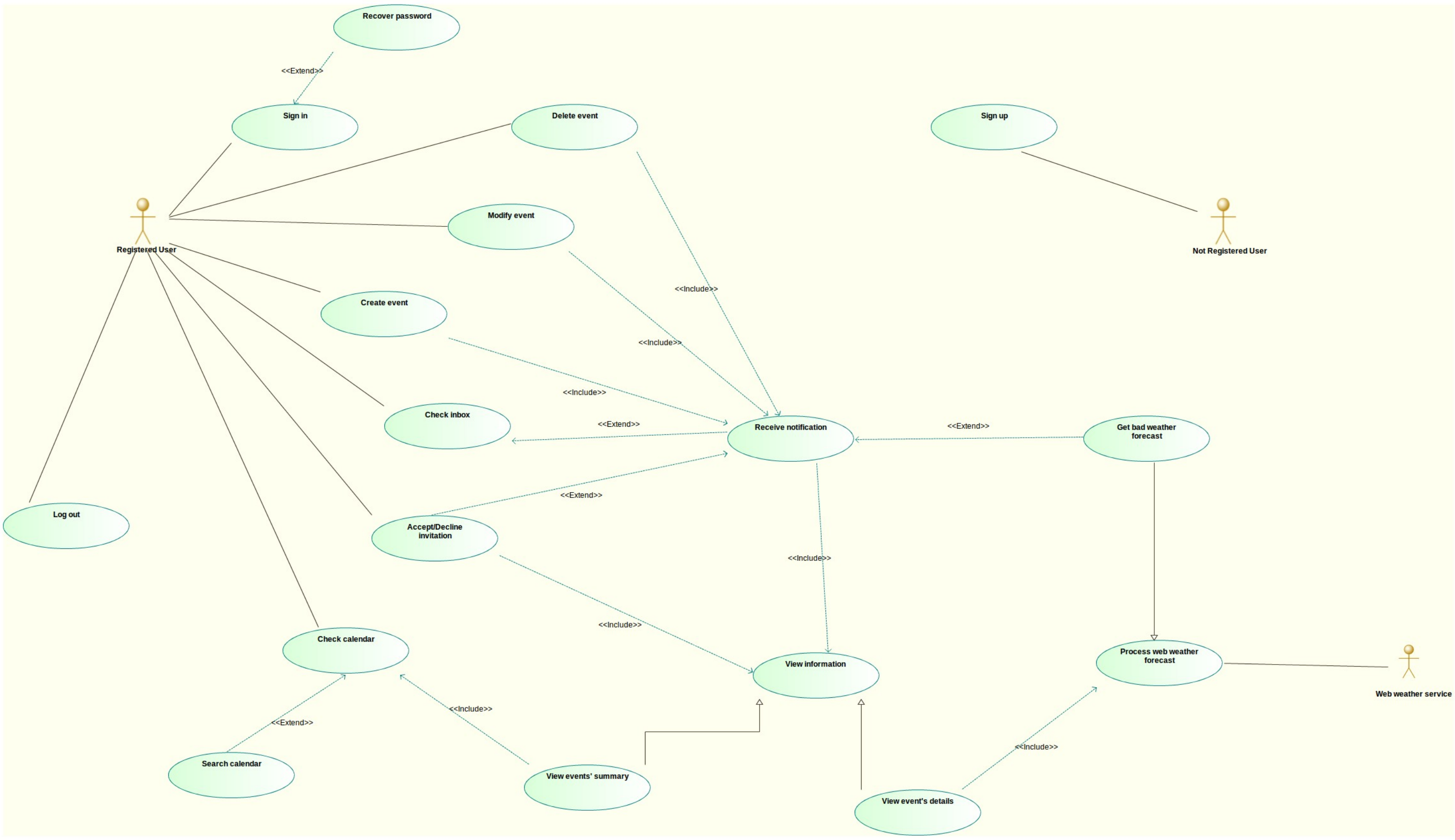
Title	*Incoming bad weather conditions.*
Goal	*The organizer can shift the date when there are forecasted unsuitable weather conditions.*
Assumptions	*[A1][A2][A3];* *User has already created an event;* *Some users have accepted the invitation;* *Bad weather has been defined by the organizer;* *The event will take place in outdoor.*

Scenario

- The user has created an event called “Basket match” with his friends;
- Pinco and Pallino declined the invitation because of overlapping with a birthday party of a common friend, but all the other friends accepted the invitation;
- The days go by and the organizer wants to check the event's forecast on the application one week before;
- Therefore, enters in the site, selects the event on the organized event section and clicks on the info icon;
- Enters in event page and notices that it will rain;
- User is a bit worried but he knows that the weather forecasts aren't so accurate so many day before, so decides to do nothing;
- Other four days pass, and when the organizer logs in the site again, the user notices that there is a new notification in the inbox, so decides to see what it is;
- The message says: “Alert! The day of the event “Basket match” will be rainy!”;
- At this point, the organizer decides to shift the date of this event to the next week;
- So, returns on the main page, selects the event and clicks on update icon;
- In the update page, modifies the date, shifting it to the next week, and finally submit the update;
- As Pinco and Pallino have declined the previous invitation, they receive a new one with up to date data.

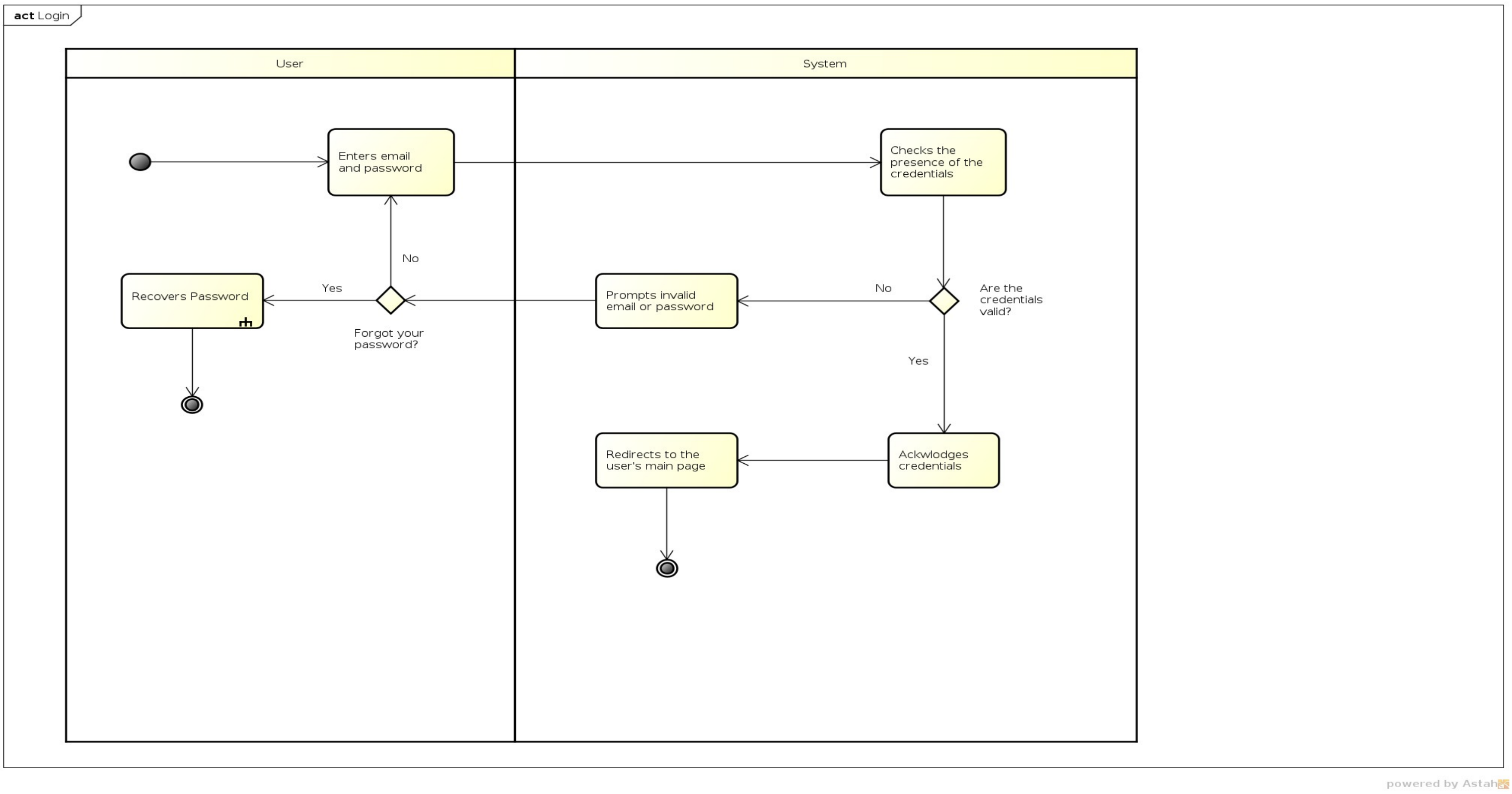
4 UML model

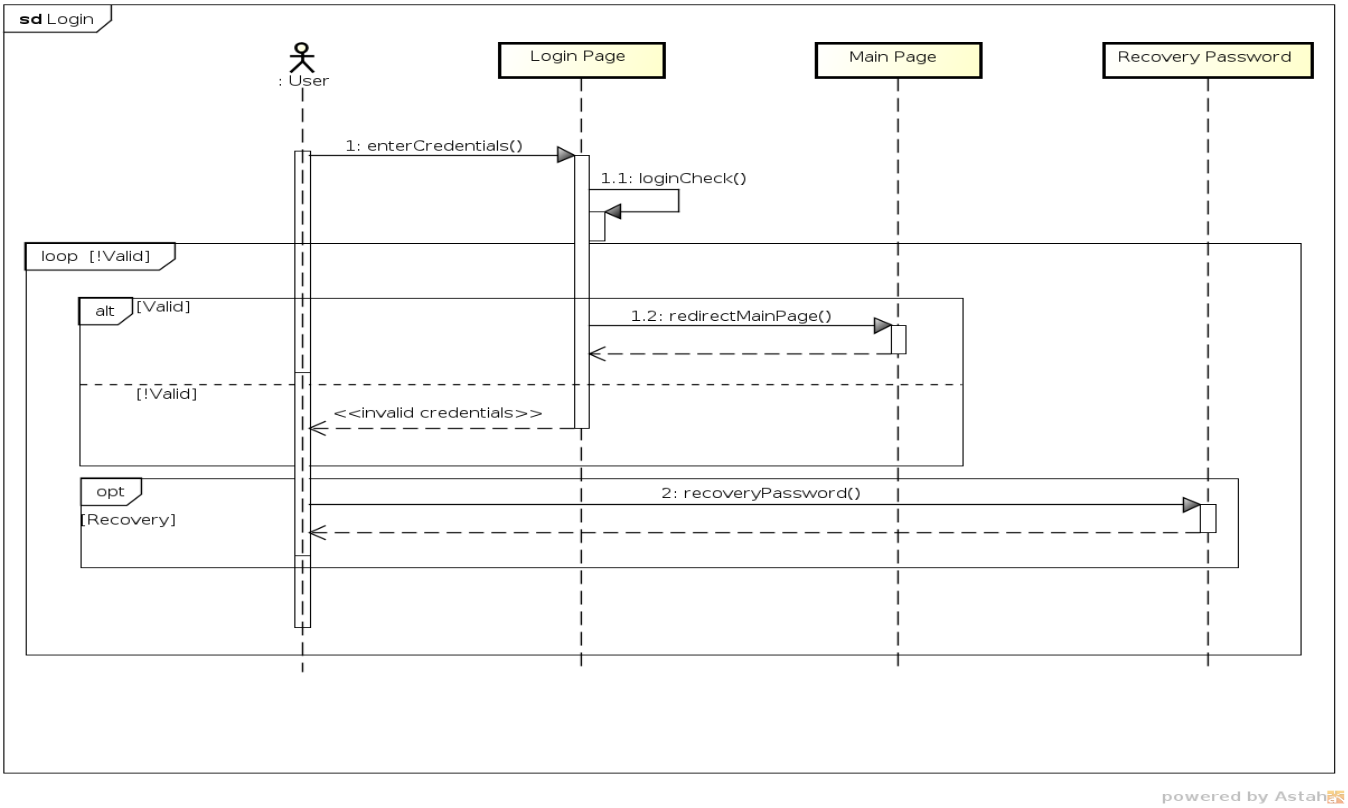
4.1 Use case diagram



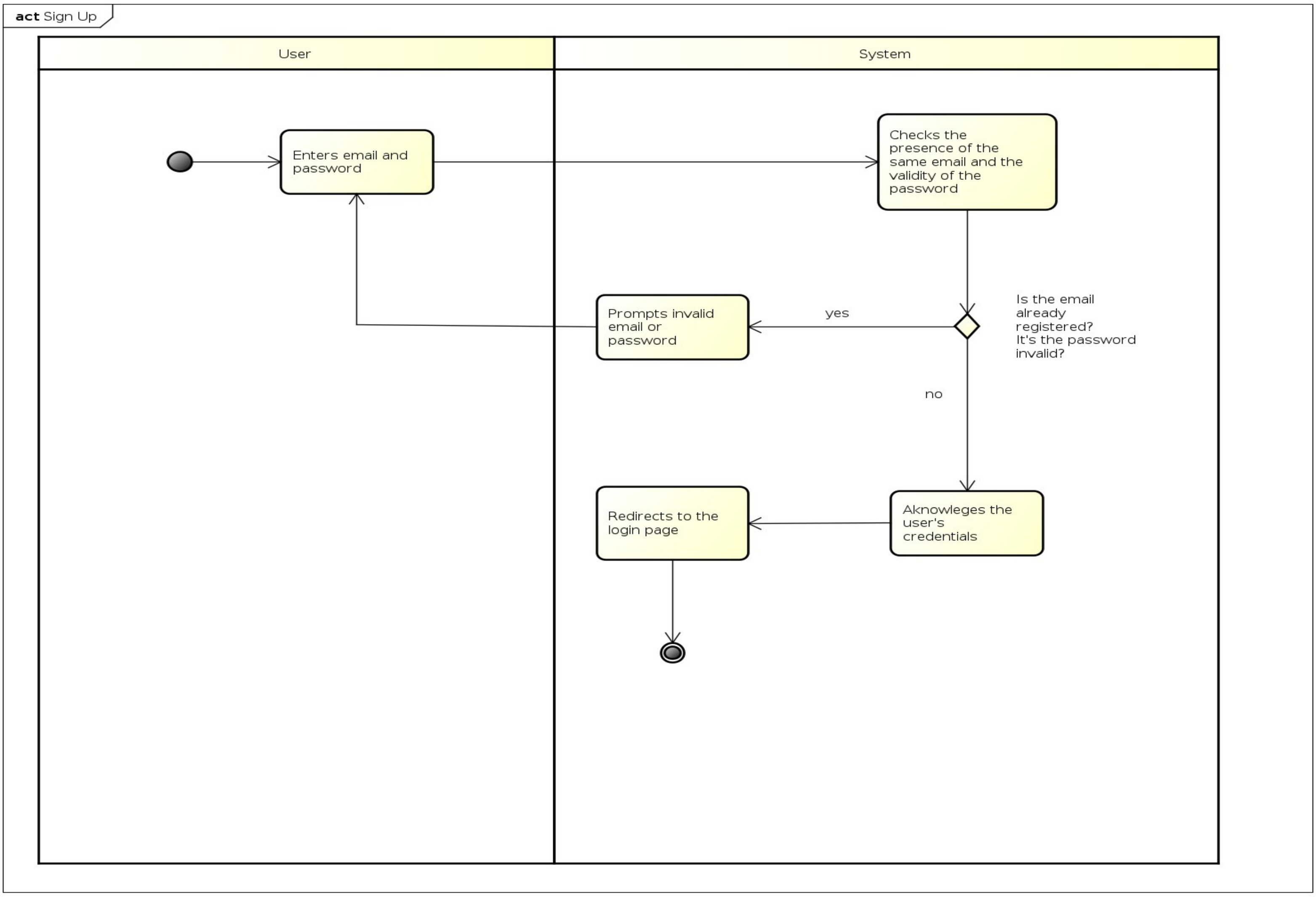
4.2 Use cases

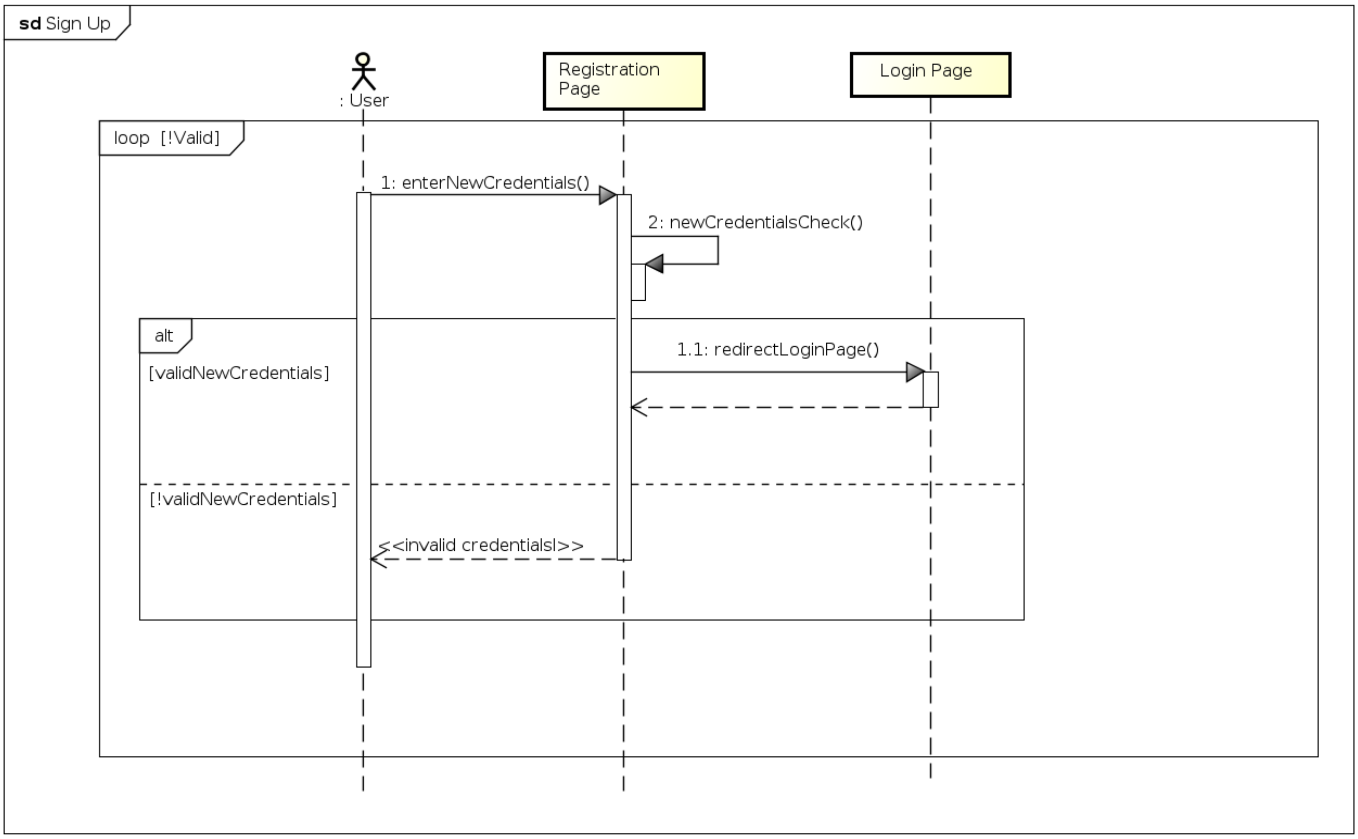
Use case name	**Sign in**
Participating actors	Registered user.
Flow of events	
Exceptions	• Not registered email; • Entered wrong password;
Entry condition	• The user has successfully signed up to the system. • The user is on Log in page.
Exit condition	• The user is redirected to the personal web calendar; • Password recovery.
Quality requirements	• The system doesn't permit simultaneous login for the same account.


act Login
User
System
Enters email and password
Checks the presence of the credentials
No
Recovers Password
Yes
Forgot your password?
Prompts invalid email or password
No
Are the credentials valid?
Yes
Redirects to the user's main page
Ackwlodges credentials
powered by Astah

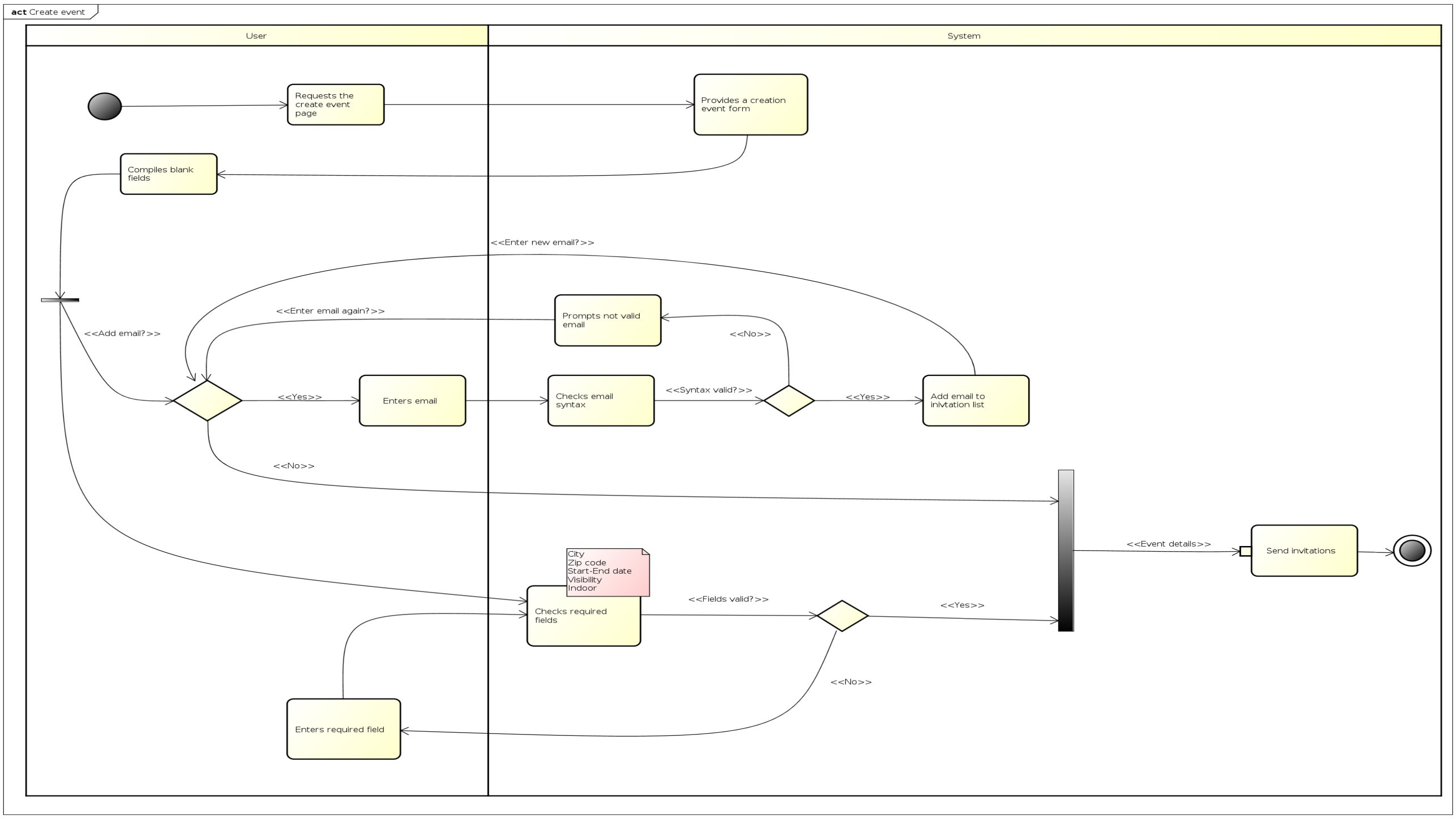

sd Login
: User
Login Page
Main Page
Recovery Password
1: enterCredentials()
1.1: loginCheck()
loop [!Valid]
alt [Valid]
1.2: redirectMainPage()
[!Valid]
<<invalid credentials>>
opt
[Recovery]
2: recoveryPassword()
powered by Astah

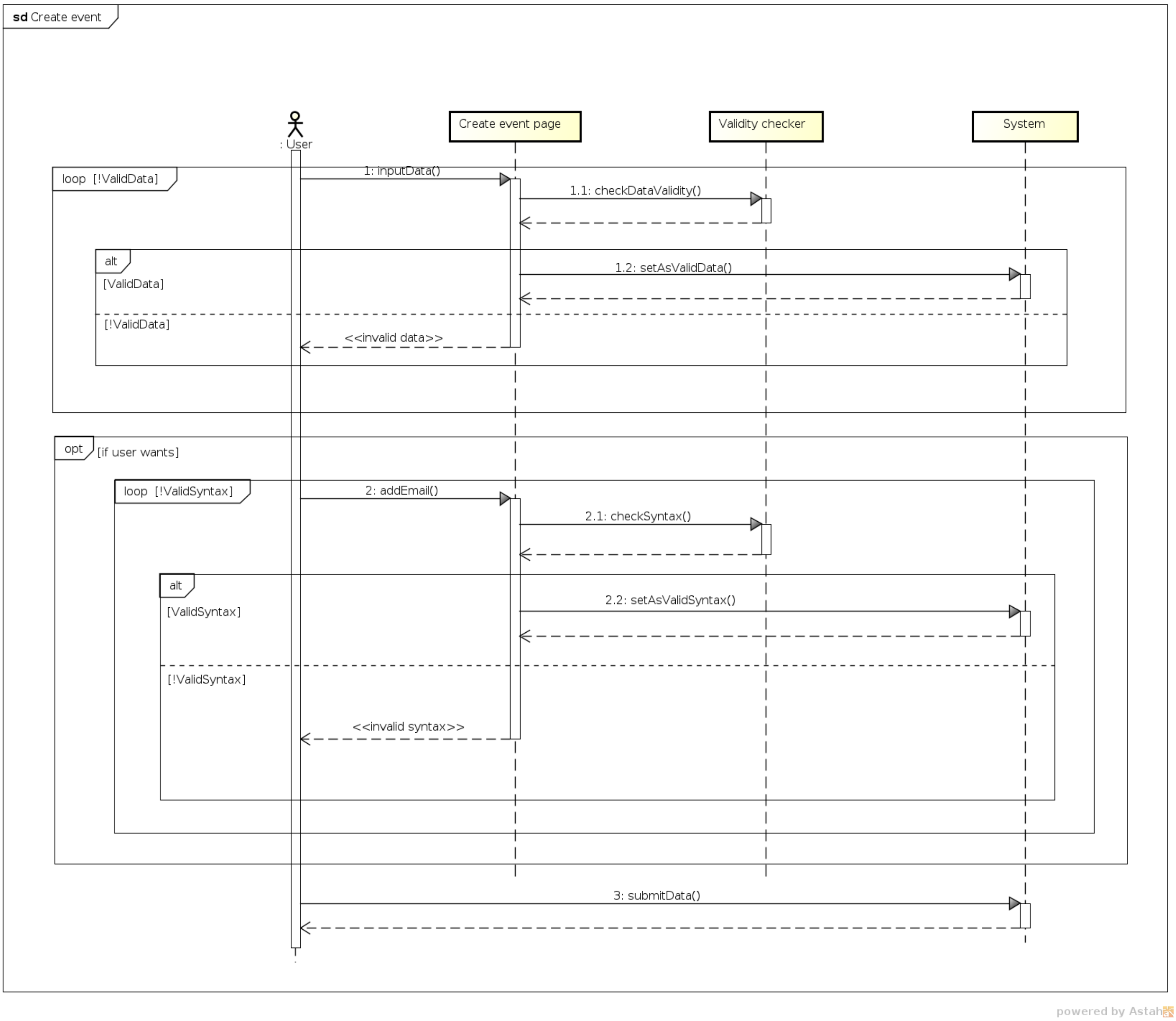
Use case name	**Sign up**
Participating actors	Not registered user.
Flow of events	
Exceptions	• Wrong email syntax; • Email already registered; • Entered less than 6 characters for password field.
Entry condition	• The user has been successfully redirected to sign up page; • User not registered yet.
Exit condition	• The system acknowledges registration.
Quality requirements	• Password length is greater than 5 characters; • Email syntax check; • System confirms the registration via mail.


act Sign Up
User
System
Enters email and password
Checks the presence of the same email and the validity of the password
Is the email already registered? It's the password invalid?
yes
Prompts invalid email or password
no
Aknowleges the user's credentials
Redirects to the login page

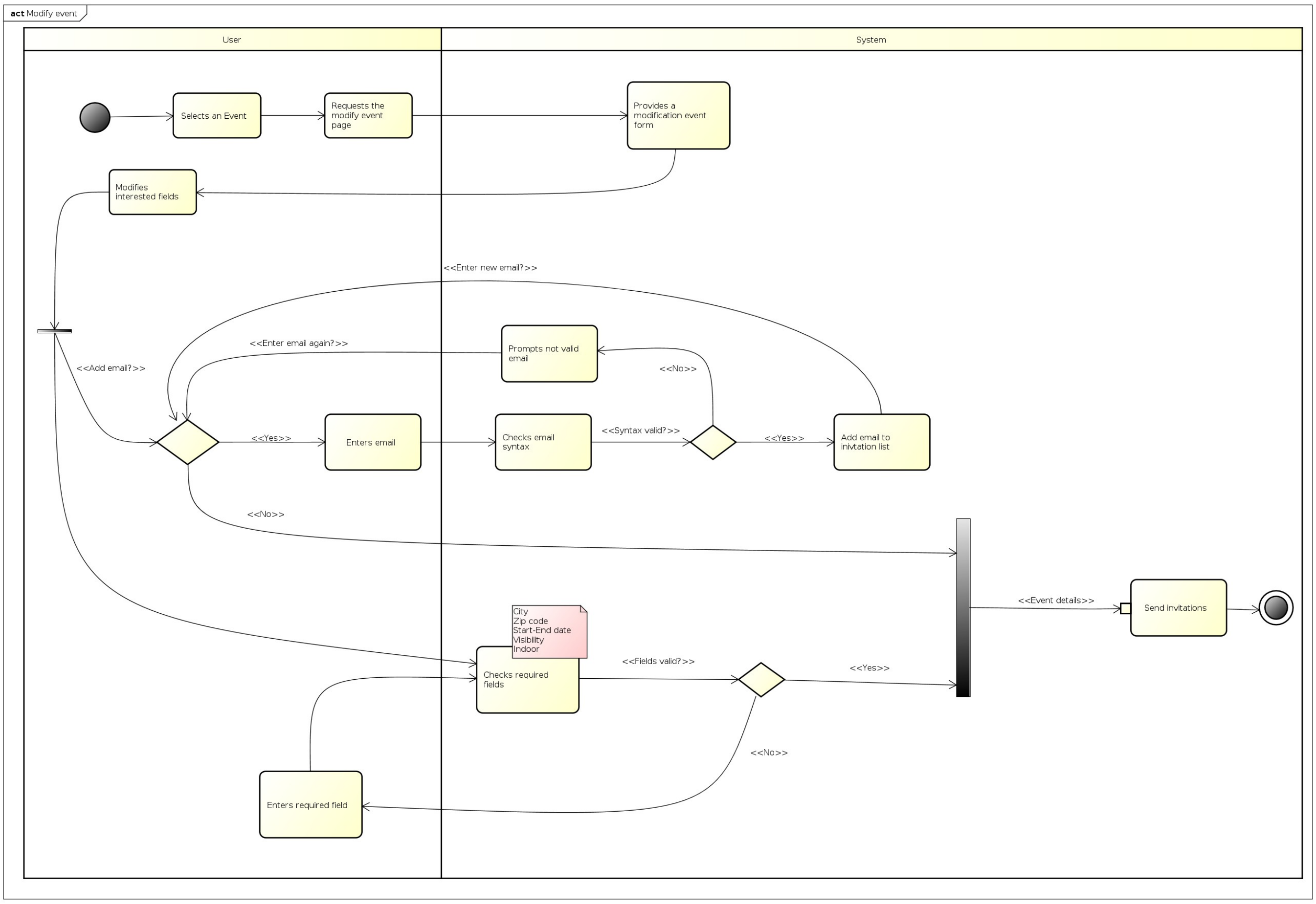

sd Sign Up
: User
Registration Page
Login Page
loop [!Valid]
1: enterNewCredentials()
2: newCredentialsCheck()
alt
[validNewCredentials]
1.1: redirectLoginPage()
[!validNewCredentials]
<<invalid credentialsl>>

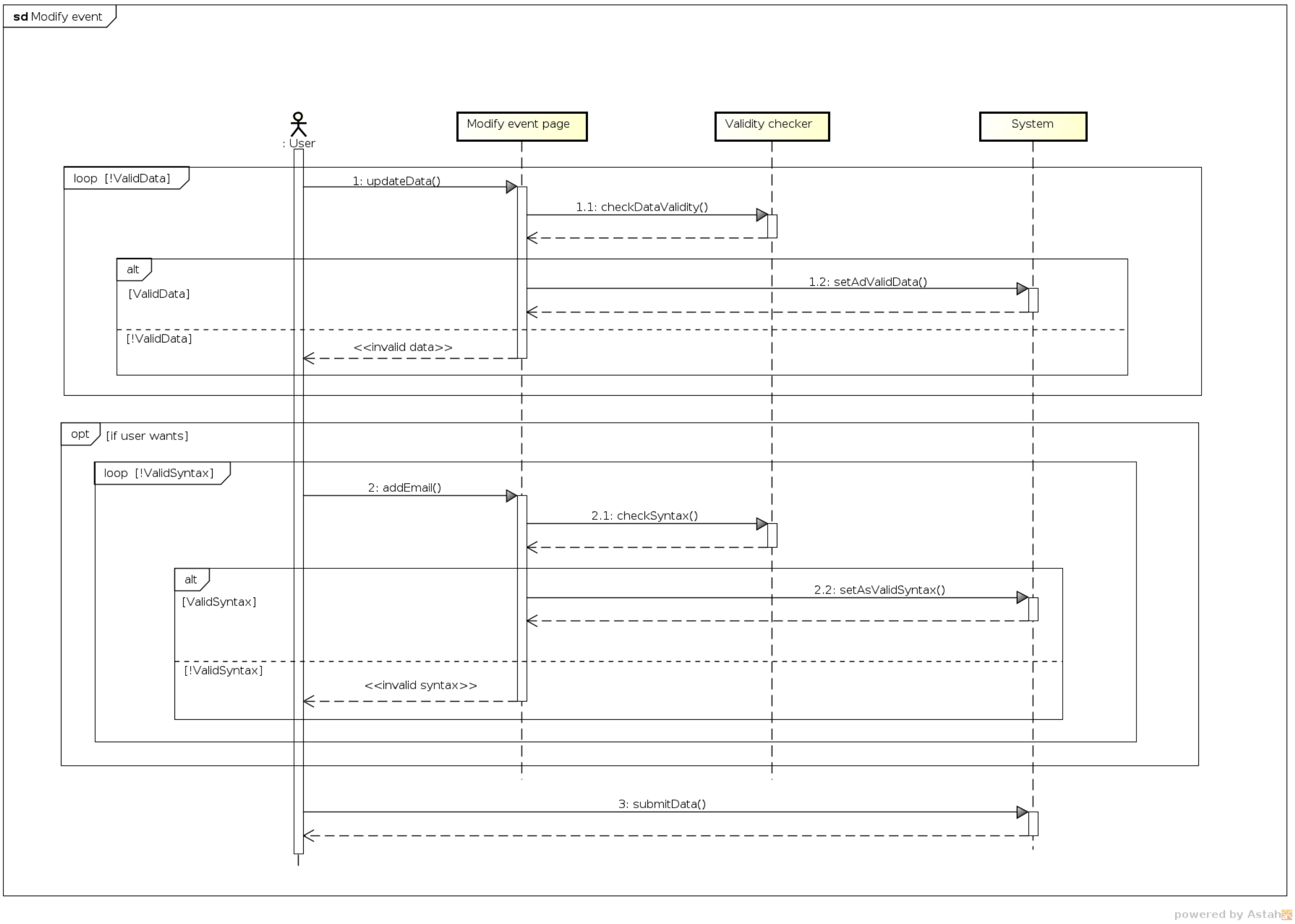
Use case name	**Create event**
Participating actors	Registered user.
Flow of events	
Exceptions	• New event overlaps with already organized event; • One of the required data is invalid; • Entered city not present in database; • Wrong email syntax on invitation list.
Entry condition	• User logged in; • User is on create event page.
Exit condition	• User fills at least the required blanks in compilation form and submits it to the system.
Quality requirements	• User defines bad weather conditions; • Email syntax check; • The end date is dynamically set to be after the start date; • The notification about created event is sent immediately.


act Create event
User
System
Requests the create event page
Provides a creation event form
Compiles blank fields
<<Enter new email?>>
<<Enter email again?>>
Prompts not valid email
<<Add email?>>
<<No>>
<<Yes>>
Enters email
Checks email syntax
<<Syntax valid?>>
<<Yes>>
Add email to inivtation list
<<No>>
<<Event details>>
Send invitations
City
Zip code
Start-End date
Visibility
Indoor
Checks required fields
<<Fields valid?>>
<<Yes>>
<<No>>
Enters required field

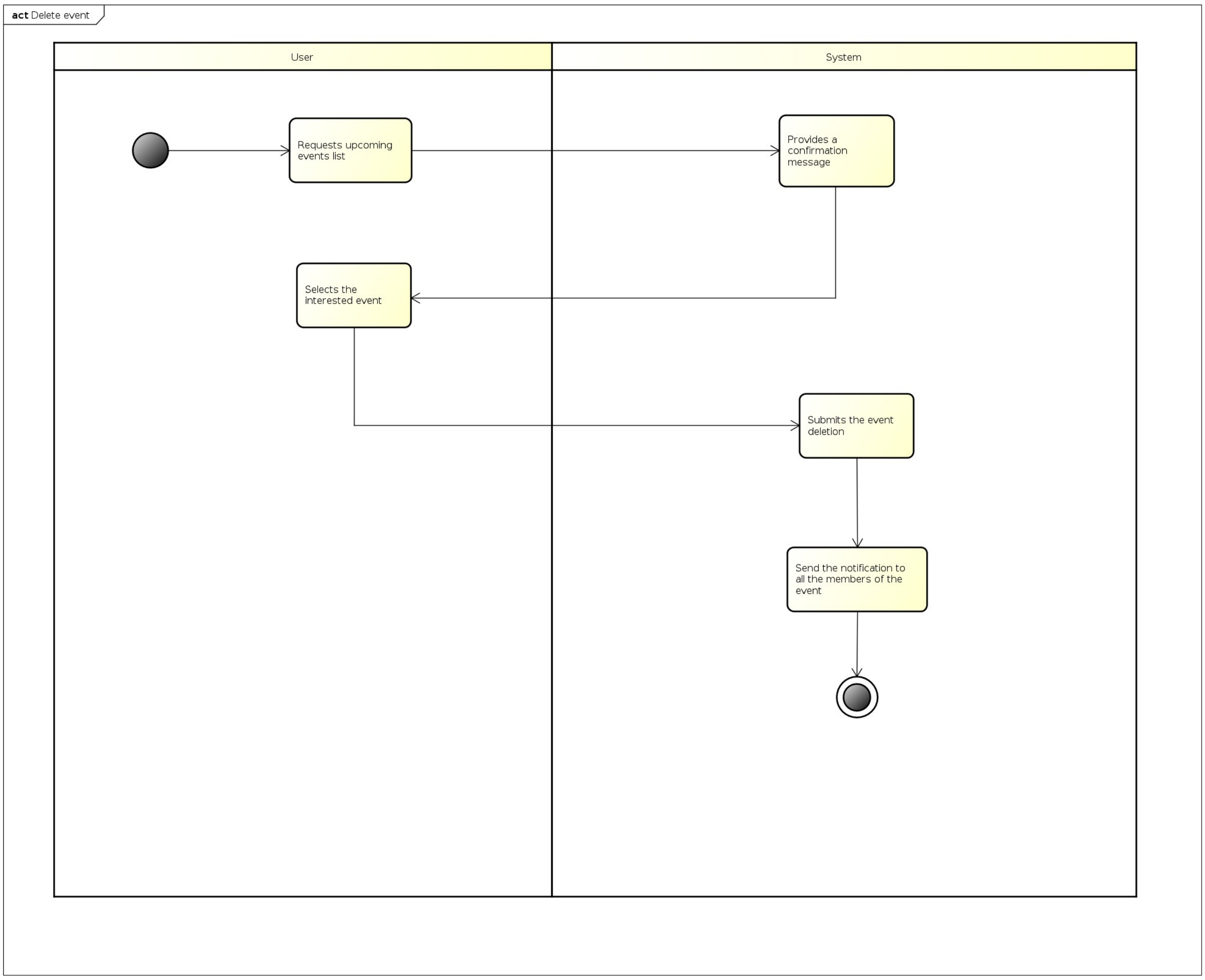

sd Create event
: User
Create event page
Validity checker
System
loop [!ValidData]
1: inputData()
1.1: checkDataValidity()
alt
[ValidData]
1.2: setAsValidData()
[!ValidData]
<<invalid data>>
opt [if user wants]
loop [!ValidSyntax]
2: addEmail()
2.1: checkSyntax()
alt
[ValidSyntax]
2.2: setAsValidSyntax()
[!ValidSyntax]
<<invalid syntax>>
3: submitData()
powered by Astah

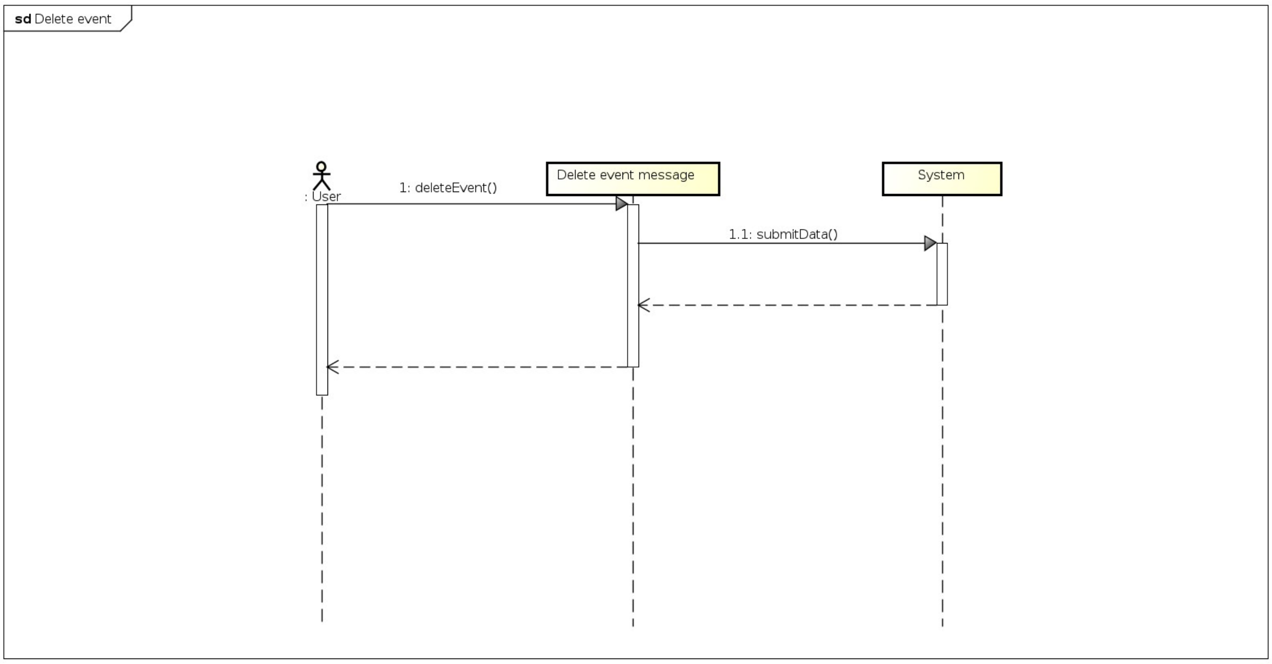
Use case name	**Modify event**
Participating actors	Registered user.
Flow of events	
Exceptions	• Updated event overlaps with already organized event; • One of the required data is invalid; • Wrong email syntax on invitation list; • Entered city not present in database.
Entry condition	• User logged in; • User has at least one created event.
Exit condition	• Update is submitted.
Quality requirements	• User defines bad weather conditions; • Email syntax check;The end date is dynamically set to be after the start date; • The notification about updated event is sent immediately.


act Modify event
User
System
Selects an Event
Requests the modify event page
Provides a modification event form
Modifies interested fields
<<Enter new email?>>
<<Enter email again?>>
Prompts not valid email
<<No>>
<<Add email?>>
<<Yes>>
Enters email
Checks email syntax
<<Syntax valid?>>
<<Yes>>
Add email to inivtation list
<<No>>
<<Event details>>
Send invitations
City
Zip code
Start-End date
Visibility
Indoor
Checks required fields
<<Fields valid?>>
<<Yes>>
<<No>>
Enters required field

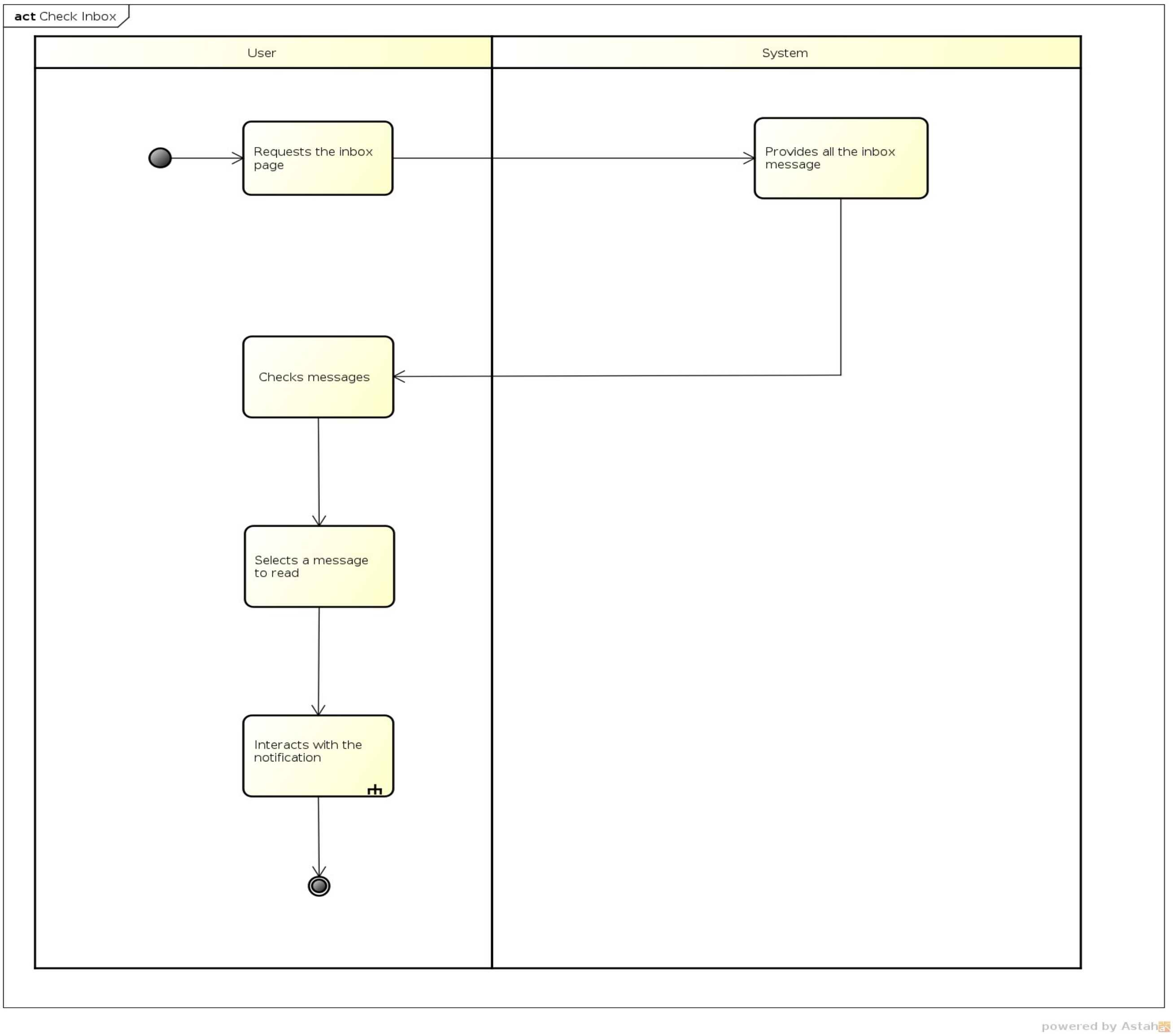

sd Modify event
: User
Modify event page
Validity checker
System
loop [!ValidData]
1: updateData()
1.1: checkDataValidity()
alt
[ValidData]
1.2: setAdValidData()
[!ValidData]
<<invalid data>>
opt [if user wants]
loop [!ValidSyntax]
2: addEmail()
2.1: checkSyntax()
alt
[ValidSyntax]
2.2: setAsValidSyntax()
[!ValidSyntax]
<<invalid syntax>>
3: submitData()
powered by Astah

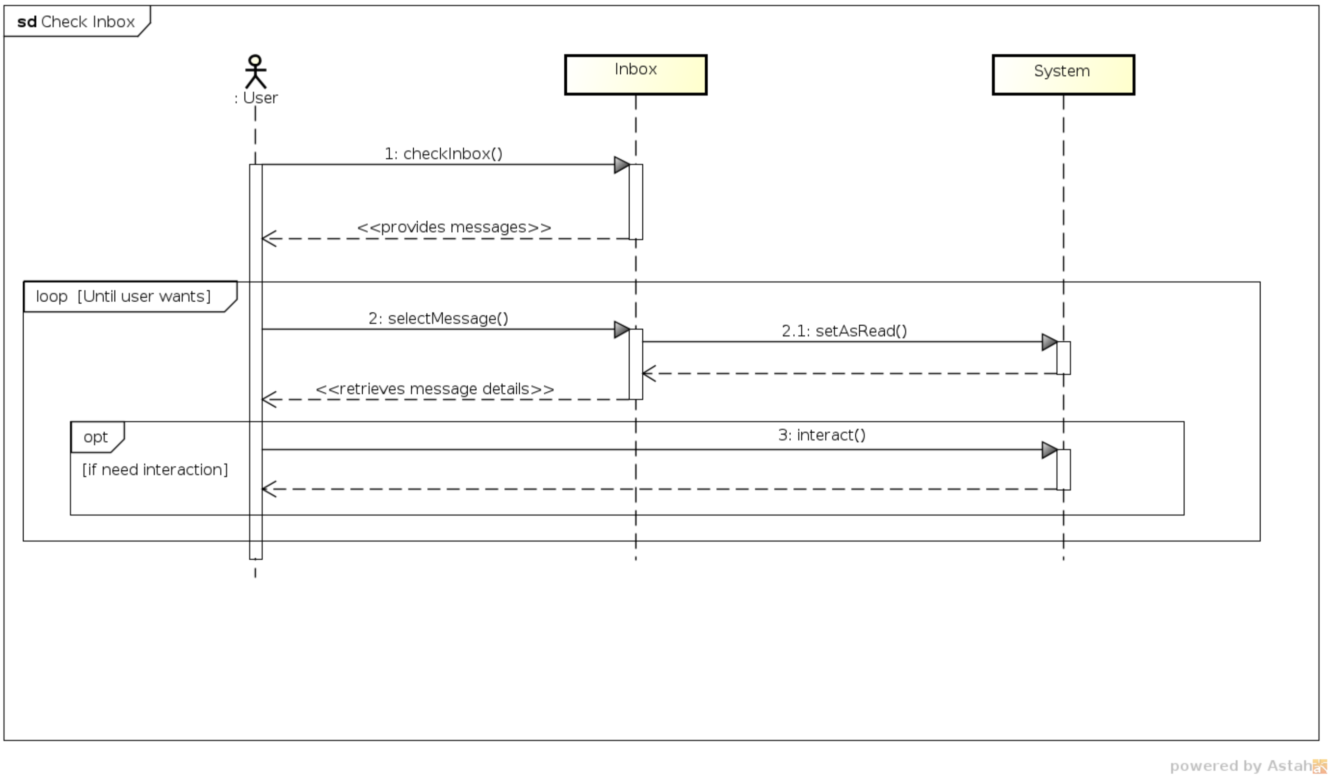
Use case name	**Delete event**
Participating actors	Registered user.
Flow of events	
Exceptions	
Entry condition	• User logged in; • User has at least one created event.
Exit condition	• Deletion is submitted.
Quality requirements	• The notification about updated event is sent immediately.


act Delete event
User
System
Requests upcoming events list
Provides a confirmation message
Selects the interested event
Submits the event deletion
Send the notification to all the members of the event

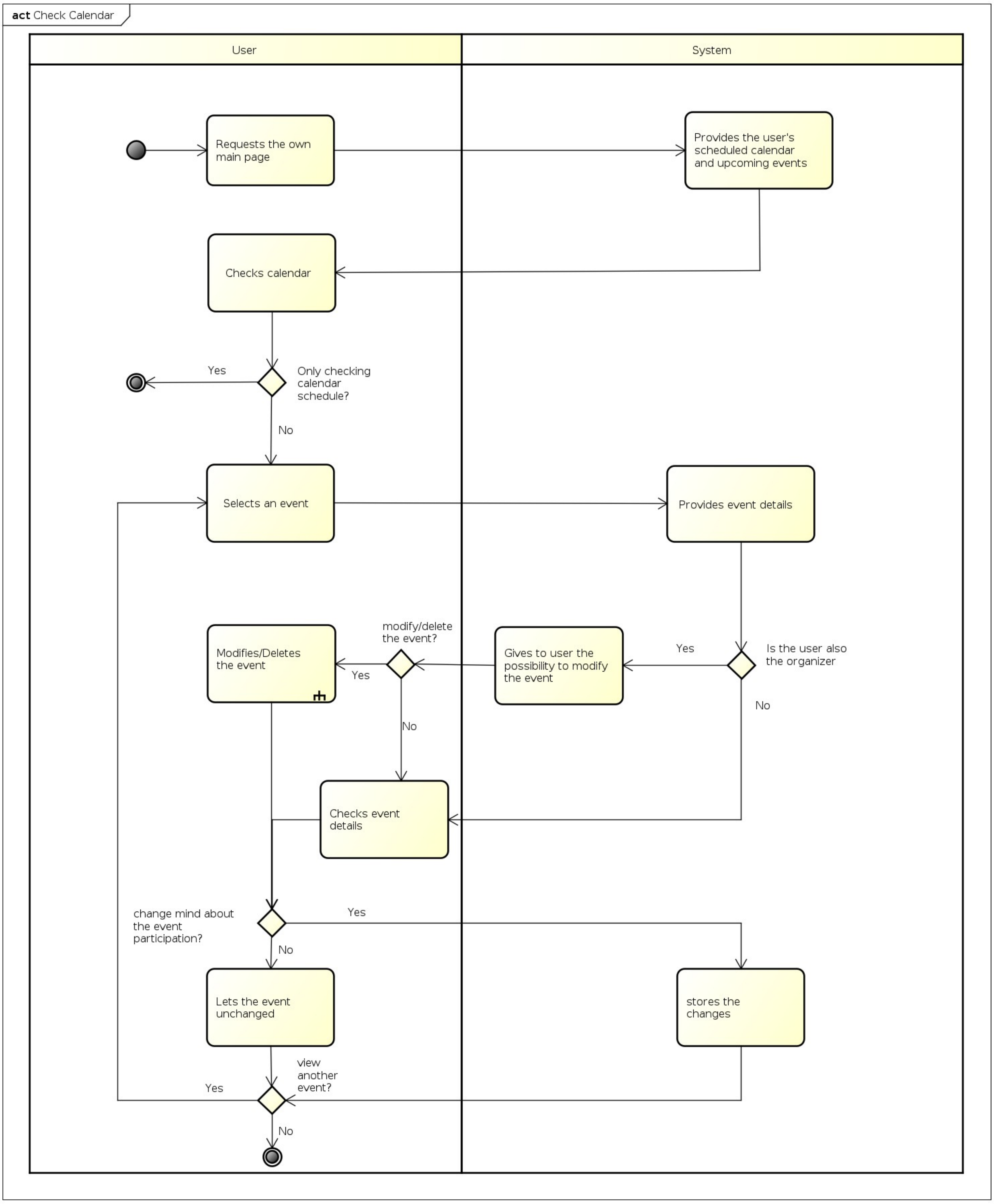

sd Delete event
: User
Delete event message
System
1: deleteEvent()
1.1: submitData()

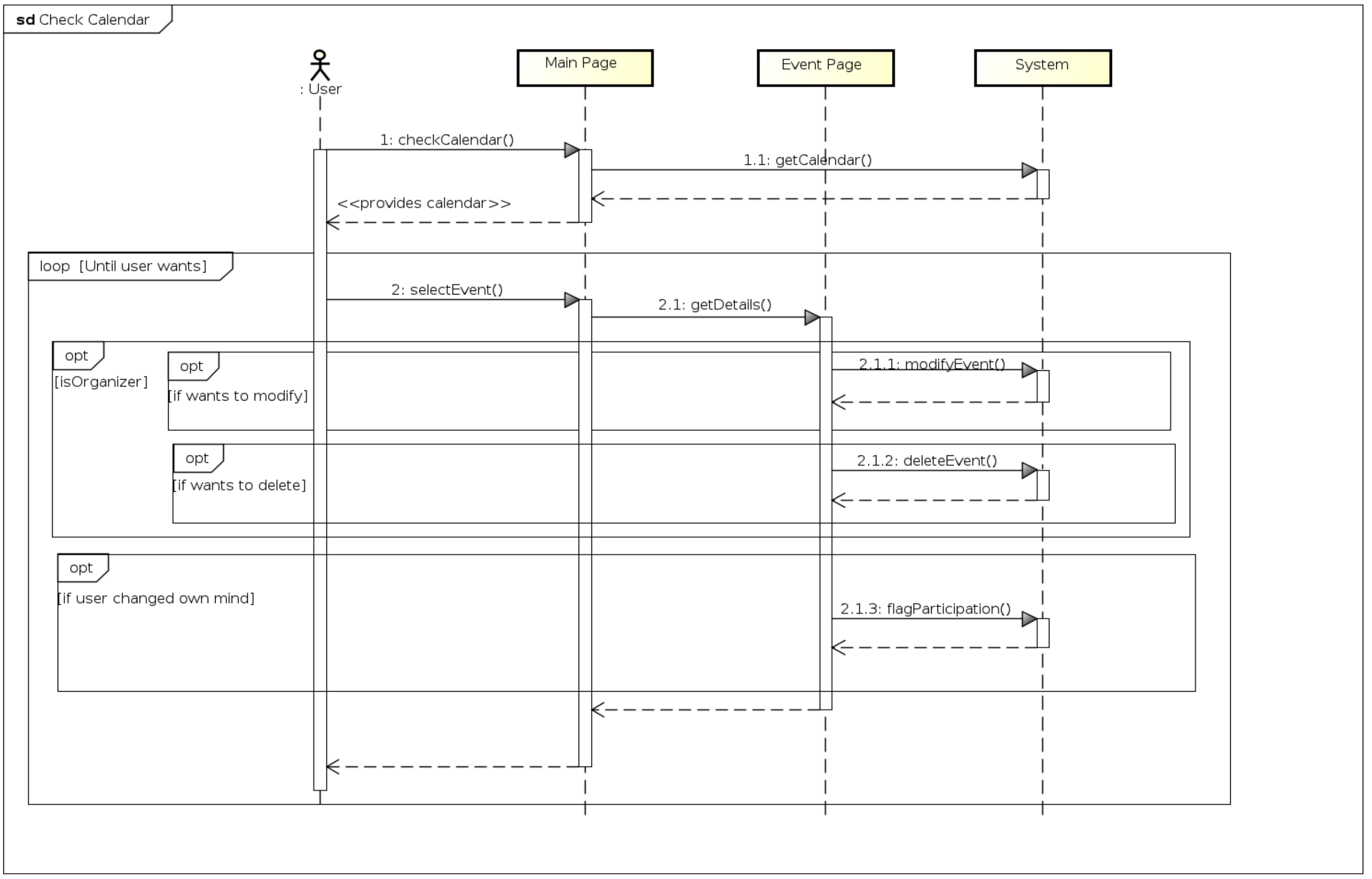
Use case name	**Check inbox**
Participating actors	Registered user.
Flow of events	
Exceptions	
Entry condition	• User logged in; • User clicks on inbox icon.
Exit condition	• User is redirected to inbox page.
Quality requirements	• A copy of the notifications are redirected to the user's email.


act Check Inbox
User
System
Requests the inbox page
Provides all the inbox message
Checks messages
Selects a message to read
Interacts with the notification
powered by Astah


sd Check Inbox
: User
Inbox
System
1: checkInbox()
<<provides messages>>
loop [Until user wants]
2: selectMessage()
2.1: setAsRead()
<<retrieves message details>>
opt
[if need interaction]
3: interact()
powered by Astah

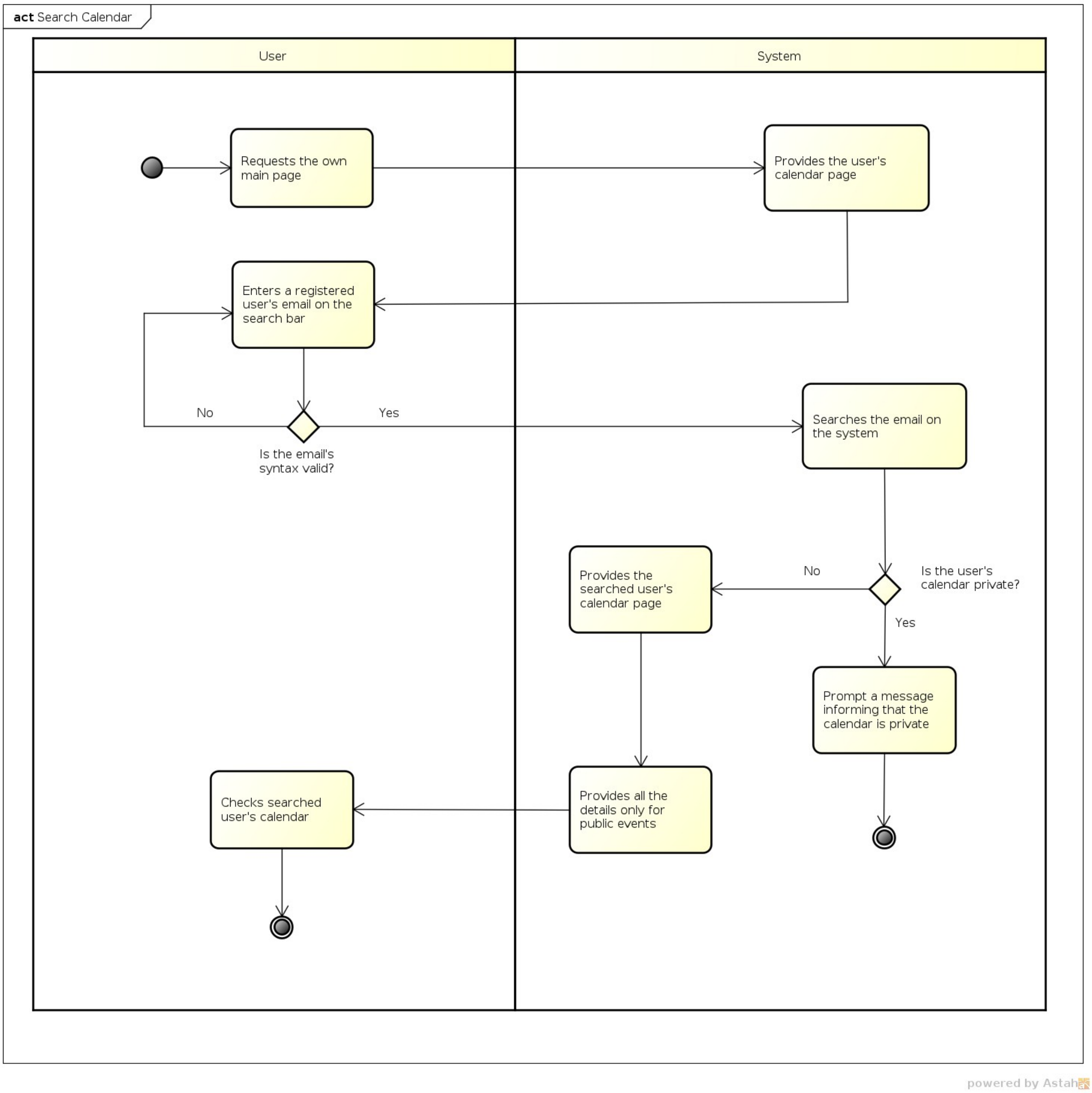
Use case name	**Check calendar**
Participating actors	Registered user.
Flow of events	
Exceptions	• Wrong email syntax entered in the search bar.
Entry condition	• User logged in.
Exit condition	• User is redirected to the user's main page.
Quality requirements	• Preview of the user's upcoming events; • Search user's calendar by email.

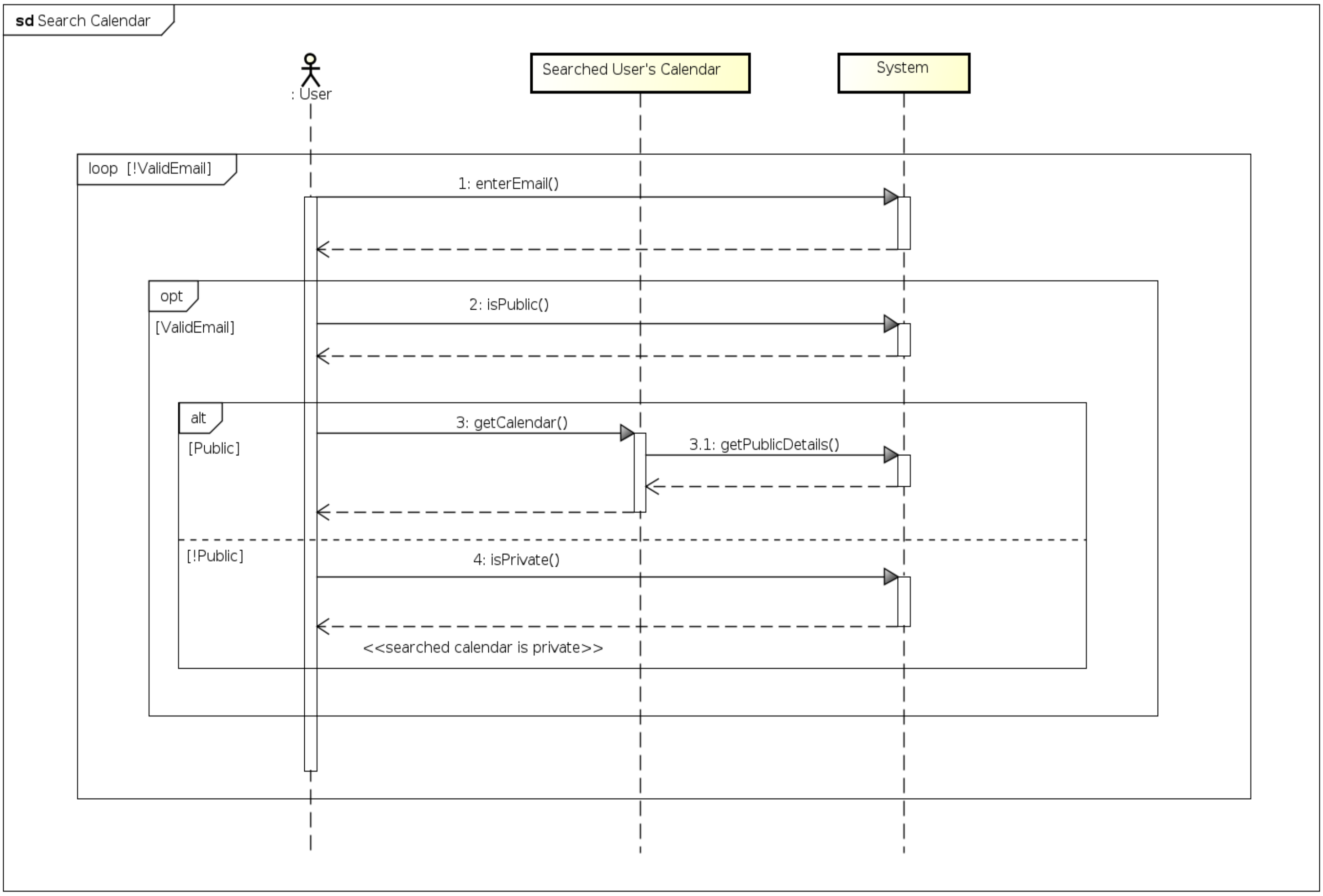

act Check Calendar
User
System
Requests the own main page
Provides the user's scheduled calendar and upcoming events
Checks calendar
Yes
Only checking calendar schedule?
No
Selects an event
Provides event details
modify/delete the event?
Modifies/Deletes the event
Yes
Gives to user the possibility to modify the event
Yes
Is the user also the organizer
No
No
Checks event details
change mind about the event participation?
Yes
No
Lets the event unchanged
stores the changes
view another event?
Yes
No



powered by Astah

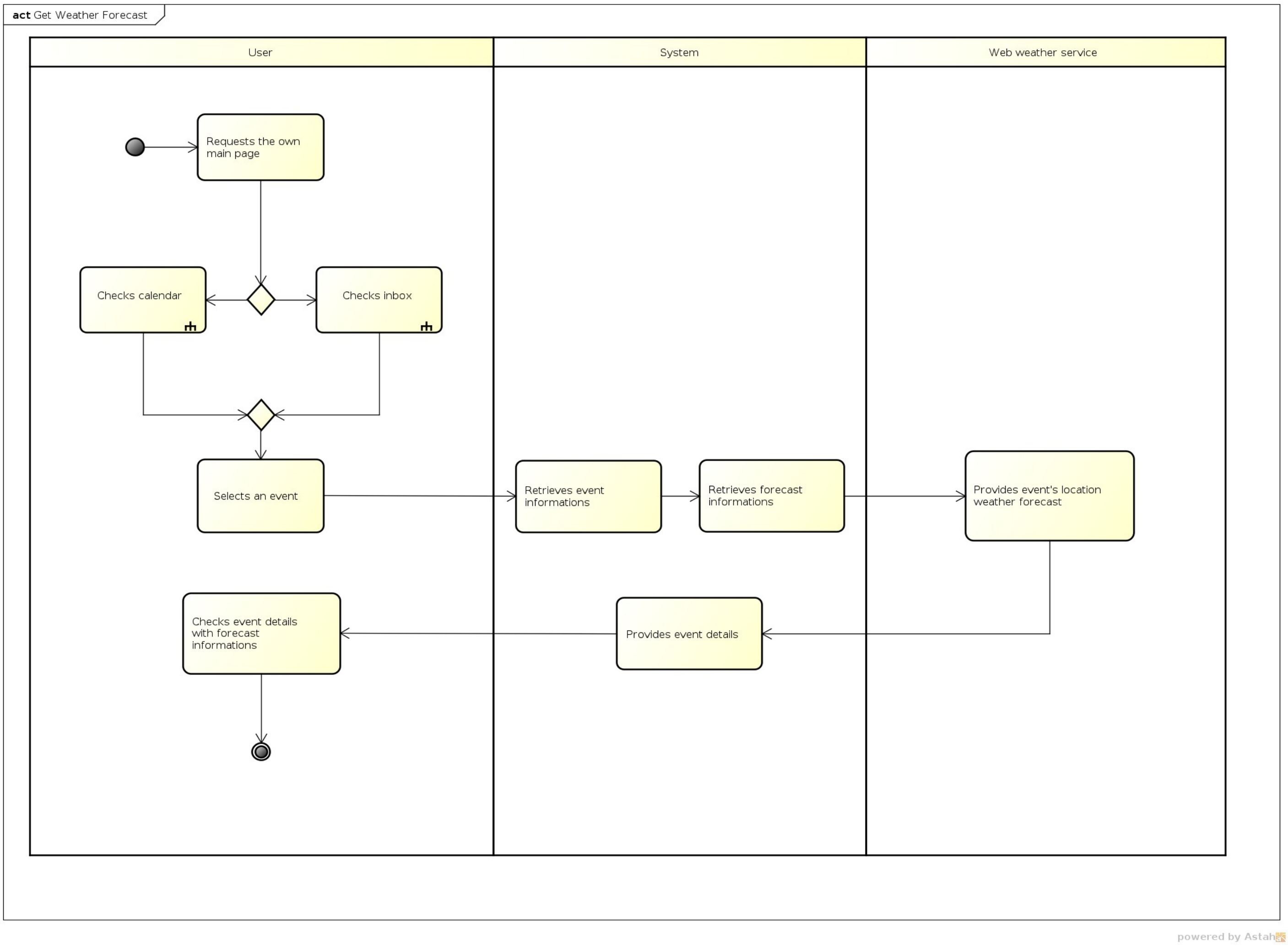
Use case name	**Search calendar**
Participating actors	Registered user.
Flow of events	
Exceptions	• Wrong email syntax entered in the search bar; • Searched unregistered email.
Entry condition	• User logged in.
Exit condition	• User is redirected to the searched user's calendar page; OR • User has been informed that searched user's calendar is private.
Quality requirements	• Email syntax check.

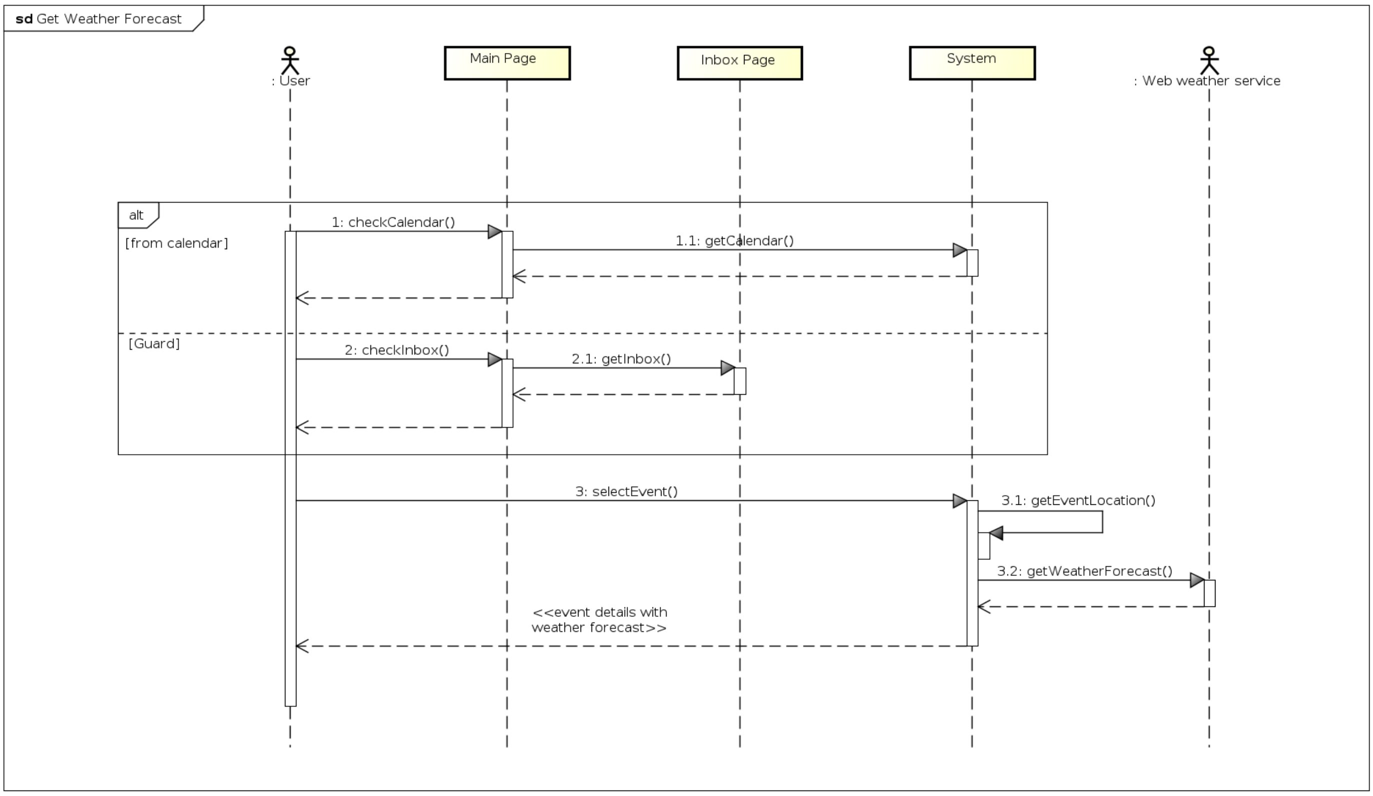

act Search Calendar
User
System
Requests the own main page
Provides the user's calendar page
Enters a registered user's email on the search bar
No
Yes
Is the email's syntax valid?
Searches the email on the system
Provides the searched user's calendar page
No
Is the user's calendar private?
Yes
Prompt a message informing that the calendar is private
Checks searched user's calendar
Provides all the details only for public events
powered by Astah



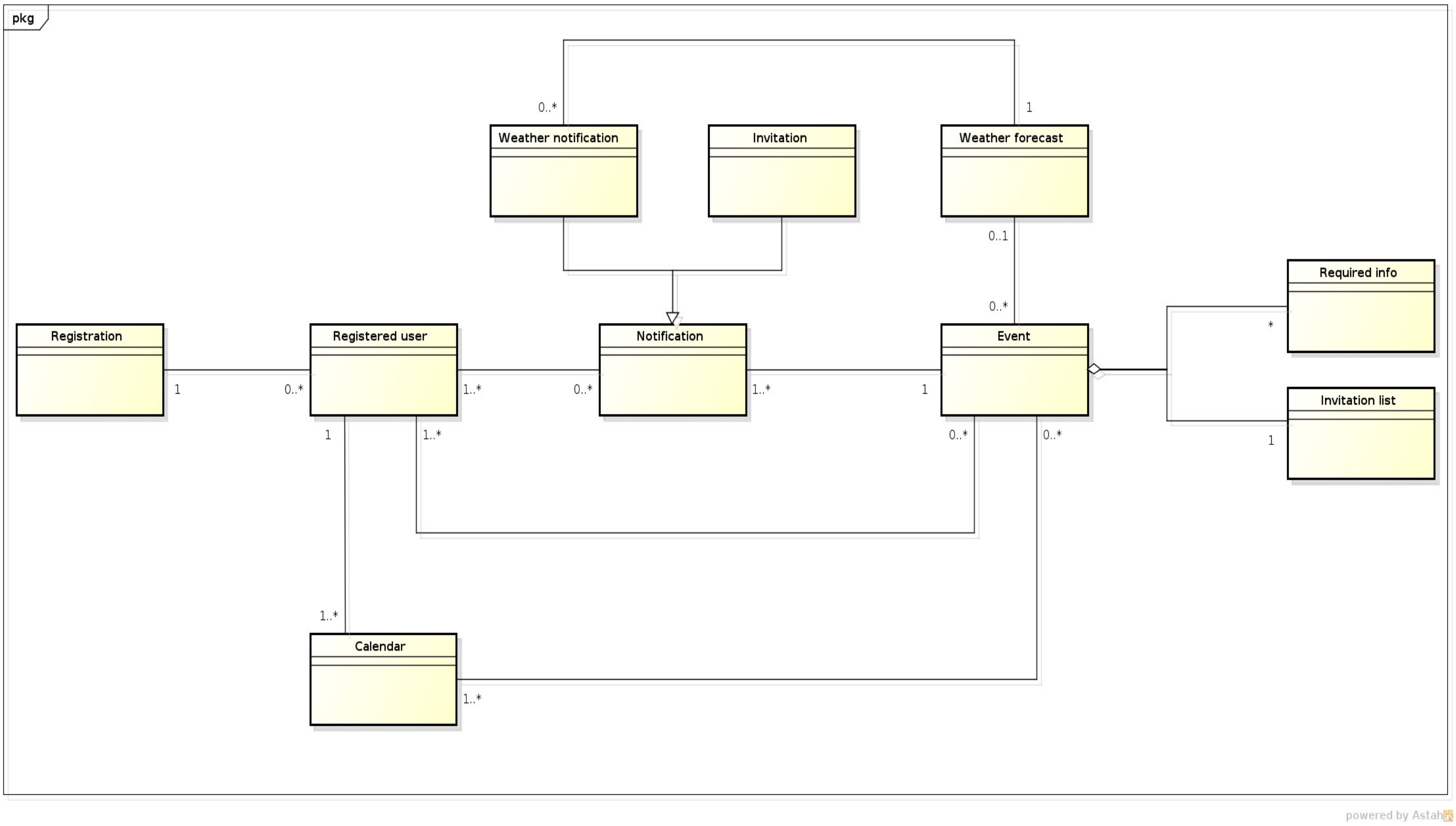
powered by Astah

Use case name	**Get weather forecast**
Participating actors	Web weather service.
Flow of events	
Exceptions	• Such city not present in database; • The web weather service is offline and the forecast is not retrievable.
Entry condition	• City and date are ready for request.
Exit condition	• Send to the system weather forecast information.
Quality requirements	• Filter weather details: ○ Only Celsius degrees are considered; ○ Fog is when the visibility less than one km; ○ Wind in km/h; ○ Precipitations in mm.

act Get Weather Forecast
User
System
Web weather service
Requests the own main page
Checks calendar
Checks inbox
Selects an event
Retrieves event informations
Retrieves forecast informations
Provides event's location weather forecast
Provides event details
Checks event details with forecast informations
powered by Astah


sd Get Weather Forecast
: User
Main Page
Inbox Page
System
: Web weather service
alt
[from calendar]
1: checkCalendar()
1.1: getCalendar()
[Guard]
2: checkInbox()
2.1: getInbox()
3: selectEvent()
3.1: getEventLocation()
3.2: getWeatherForecast()
<<event details with
weather forecast>>

4.3 Class diagram



5 Alloy model