

Software Requirements Specification

Project for NordVisa

2DV603

Leif Karlsson

Johan Gudmundsson

Francis Menkes

Feiyu Xiong

Axel Nilsson

Table of Contents

1. Introduction	4
1.1. Project Overview	
1.2. Purpose and Scope of Specification	
1.3. Definitions and Glossary	
2. Product description	6
2.1. Product Context	
2.2. Product Functions	
2.3. User Characteristics	
2.4. Constraints	
2.5. Assumptions and dependencies	
3. Requirements	9
3.1. Events - Create, Read, Update, Delete	
3.2. Authorization and Authentication	
3.3. Geographical information	
3.4. Other Functional Requirements	
3.5. Quality Requirements	
3.6. Platform Requirements	
3.7. Process Requirements	
4. Scenarios	16
4.1. Visitor	
4.2. Administrator	
4.3. Super Administrator	
4.4. Registered User	
4.5. Organization	
5. Use Cases	20
UC1. Login	
UC2. Log out	
UC3. Register Account	
UC4. Recover lost password	
UC5. Update User Account	
UC6. Add administrative privileges	
UC7. Remove administrative privileges	
UC8. Add super administrative privileges	
UC9. Create Event	
UC10. Update Event	
UC11. Delete Event	
UC12. View Event in Dashboard	
UC13. List Events	
UC14. Generate Widget Code	

- UC15. Change region
- UC16. View Single Event in Widget
- UC17. View Calendar
- UC18. Widget language selection
- UC19. Dashboard language selection
- UC20. Widget switch to Maps View

Appendix I: Domain Model	46
Appendix II: Use Case Diagram	47
Appendix III: Interaction Diagrams	48
Appendix IV: Requirement/Use Case Traceability Matrix	68

1. Introduction

1.1 Project Overview

In this project we have been tasked to create an event calendar for NordVisa. NordVisa is an organization around the Nordic tradition of folk ballads. There exist many organizations and artists who host events and perform on local stages around the Nordic countries. There is however an information problem since there is no one place for events for all organizations.

NordVisa wants us to develop an event calendar where anyone can create events. Each organization can then embed this calendar in their organization website and filter the events to a certain geographical region.

1.2 Purpose and Scope of Specification

The purpose of this specification is to create a common understanding between all stakeholders what is to be created. There is an overview of how the system will be structured and how the interactions between the user and the system will look. There will however be very limited technical details on how the system functions.

A detailed list of all requirements of the software solutions is presented in the document. All requirements have a priority rank which shows how essential they are to the end product. There is also a couple of use cases to illustrate how the functional requirements will work in practice which are derived from scenarios, which are also included in the document.

Diagrams for illustrating use cases and the domain model can be found in the appendixes of the document together with a traceability matrix mapping use cases to requirements.

1.3 Definitions and Glossary

Dashboard	Personal user page for administering events and user settings.
Widget	The widget is the calendar which unregistered users can interact with to view events.
Registered User	A user with an account registered in the system.
Visitor	A user without a registered account who only use the calendar widget.
Administrator	A registered user with administrative privileges for a specified group. They could be administrator for a specific organization or over all members in the system.

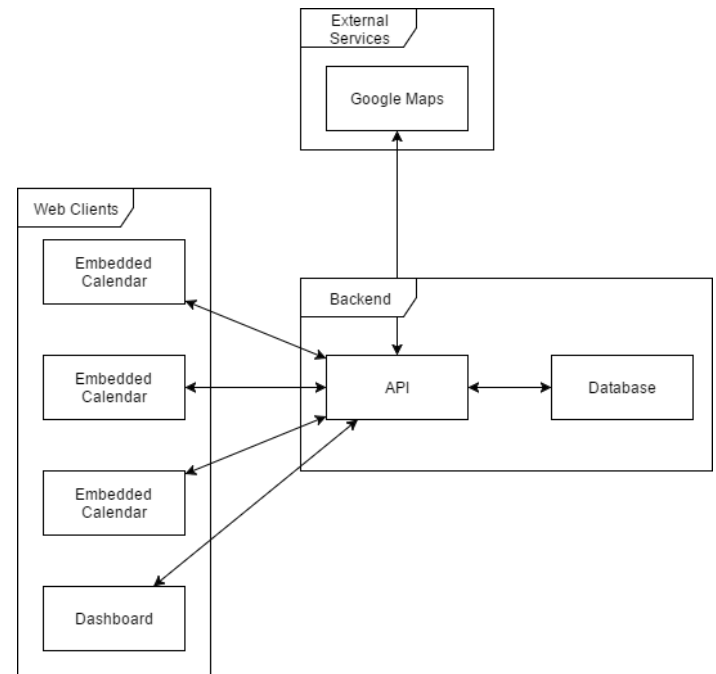
Super Administrator	A registered user with super administrative privileges to manage even administrators.
Issue command	A generalization of an action performed by a user in the system.
Organization	An organization which embed the calendar in their website or wish to do so.

2. Product Description

2.1 Product Context

The overall structure of the application will be a server-client architecture. The reasoning is that the application will be used on the web which lends itself to a clear separation of client and server.

Also the fact that we want to have a centralized system for storing all events in one place and at the same time have many varying clients. This is where the API comes in. The API will be used to communicate between the clients and the server. When a user wants to add an event to the calendar an HTTP request is sent to the API which then works with the database to fulfill this request.



The backend will also utilize the Google Maps API for managing locations for events.

2.2 Product Functions

A widget can be generated to be embedded into a website by their website administrator. When the widget is generated the default settings for region can be set and then code is generated based on the requested preferences.

When the widget is embedded into a website anyone can visit this website and check for upcoming events. The widget should fit and be usable on desktop, tablet and mobile web browsers.

For managing events the user will have to be registered and logged in to the dashboard. Anyone can register and the registration is kept simple and is only there to identify who created what event and to keep bots away. Once logged in the user can create events and manage events already created by the same account.

Any account can be turned into an administrative account which can edit and remove any events added to the system. This is so someone can remove any unwanted events from the calendar. There is also the super administrator which is there to manage and remove any administrator privileges.

2.3 User Characteristics

Each type of user has tasks they can perform. Each type of user also inherits all functionality of every type above it. So Organization also has the functionality of the Visitor type. An Administrator also has the functionality of Registered User, Organization, and Visitor.

Here it is illustrated with \leftarrow . In the case ($a \leftarrow b$), a has only functionality of a , and b has the functionality of a and b . Visitor \leftarrow Organization \leftarrow Registered User \leftarrow Administrator \leftarrow Super Administrator.

Visitor

A visitor is a user who visits a website where the client widget is embedded. This widget is the only way the visitor can interact with the system. The visitor will be able to view and read events from the calendar through the widget.

Organization

The organization's embed the calendar into their websites. When embedded, the calendar can be configured to show events in a certain region. There are three levels of organizations, Nordic, national, and local. For Nordic organizations, the region will be all Nordic countries, for national organizations this will be their respective country, and for local organizations their nearby area.

Registered User

An registered user is a user able to create and manage events. The registered user is able to authenticate themselves and gain authorization to the dashboard, and from there they are able to create events and manage events created by themselves.

Administrator

The administrator is more or less a Registered User but with some extra privileges to manage all events created by a specified group. This group could be an organization or it could be for the entire system. The administrator authenticates in the same way as a registered user and gain authorization to the dashboard.

Super Administrator

The super administrator work in a similar way to the administrator but can also remove administrative privileges from administrator and grant other users with super administrative privileges. Once gained the super administrative privileges can't be removed from an account.

2.4 Constraints

A constraint is the performance of the server that the application resides on. The throughput, availability, and response time of the application are all dependent on these qualities of the server.

Some parts of the software require location data of the user, but these are only a very small portion of the application.

2.5 Assumptions and dependencies

We are assuming that the user is running a modern web browser and is allowing JavaScript to be executed in the browser. Since the widget will use JavaScript in order to render itself, this is necessary for the application to function correctly.

A dependency for the application is the Google Maps API since the application uses it to retrieve location-based data for the correct storage of events, and also to input such data.

3. Requirements

Priority is a scale 1 - 5 where 1 is critical to the project and 5 is a good extra feature if there is time.

3.1 Events - Create, Read, Update, Delete

ID	Name	Priority	Description
FR-1	Create event	1	The registered user should be able to create a new event.
FR-1.1	Create event : Event details	1	The user should be able to specify the following event details: name, location, description, date and time, duration, if/when event recurs, link to website.
FR-1.1.1	Create event : Upload images	1	The user should be able to upload and save one or more images.
FR-1.1.2	Create event : System event details	1	The system should add date and time of creation to the event details.
FR-1.2	Create event : Administrator	1	The administrator should have the same privileges as the registered user.
FR-1.3	Create event : Super administrator	1	The super administrator should have the same privileges as the registered user.
FR-2	Read event	1	The visitor should be able to retrieve a single event.
FR-2.1	Read event : Event details	1	The following event details should be present in the widget: name, location, description, date and time, duration, if/when event recurs, date and time when the event was created, date and time when the event was updated.
FR-2.1.1	Read event : Display images	1	The widget should display the image(s) associated with the event.
FR-2.1.2	Read event: Map	1	The visitor should be able to switch from calendar view to maps view, to only view the location of the events.
FR-2.2	Read event : Registered user	1	The registered user should have the same privileges as the visitor, but can view the

			event in the dashboard.
FR-2.3	Read event : Administrator	1	The administrator should have the same privileges as the registered user, but can see which user created the event.
FR-2.4	Read event : Super administrator	1	The super administrator should have the same privileges as the administrator.
FR-3	Update event	1	The registered user shall be able to update a single event.
FR-3.1	Update event : Event details	1	The registered user should be able to update the following event details: name, location, description, date and time, duration, if/when event recurs, link to website.
FR-3.1.1	Update event : Update images	1	The registered user should be able to upload one or more images and delete existing images.
FR-3.1.2	Update event : System details	1	The system should set the date and time of the update.
FR-3.2	Update event : Administrator	1	The administrator has the same abilities as the registered user, but the administrator can update any event created by any user.
FR-3.3	Update event : Super administrator	1	The super administrator has the same abilities as the administrator.
FR-4.1	Delete event : Registered User	1	The registered user can delete any event they have created.
FR-4.2	Delete event : Administrator	1	The administrator can delete any event created by any registered user.
FR-4.3	Delete event : Super administrator	1	The super administrator can delete any event created by any registered user.

3.2 Authorization and Authentication

ID	Name	Priority	Description
FR-5	Register	1	Anyone should be able to register an account in the system.
FR-5.1	Register : Email	1	An email address is required to register. It has to match the email format and not already belong to a registered account. A confirmation email is also sent to the address for confirmation.
FR-5.2	Register : Password	1	A password is required to register. It has to be between 10 and 255 characters and has to be written twice to avoid mistakes.
FR-5.3	Register : Captcha	2	User has to complete a captcha to prove he/she is human
FR-5.4	Register : Personal Data	3	The account should not contain any personal data
FR-5.5	Register : Organization	2	When registering the user can pick an already existing organization to join or create a new one by giving the name and the registered user will then become administrator for that organization. If the organization already exists the user is included into the already existing organization.
FR-5.6	Register : Validation	2	When a new account is registered it has to be validated by an administrator of the organization the user has registered for. If the user has not chosen an organization then the administrator for the entire system has to accept them.
FR-6	Authentication	1	A user with a registered account should be able to authenticate with the credentials he/she provided during the registration
FR-6.1	Authentication : Email	1	To log in an email account has to be provided. It has to match an account

			already registered in the system
FR-6.2	Authentication : Password	1	To log in a password has to be provided. It has to be validated with the password of the account with matching email.
FR-6.3	Authentication : Logout	1	The user should be able to log out of the application.
FR-7	Add Administrative Privileges	1	Administrative privileges can be added to any account which does not already have it. It can be added from another administrator account, unless the receiving account is a super administrator. A system wide administrator can give administrative privileges for both the whole system or just a specific organization.
FR-8	Remove Administrative Privileges	2	Administrative privileges for the whole system can only be removed by a super administrator. There is an exception for organizations where this can be done by any other administrators.
FR-9	Add Super Administrative Privileges	2	Super administrative privileges can be added to any account which is not already a super administrator. It can only be added by another super administrator
FR-10	Remove Super Administrative Privileges	2	Super administrative privileges can not be removed
FR-11	Authorization : Dashboard	1	User has to be authorized to access the dashboard
FR-11.1	Authorization : Events	1	Has to be authorized to manage events
FR-11.2	Authorization : Account	1	Has to be authorized to manage the account

FR-11.3	Authorization : Calendar	1	No authorization is needed for accessing the calendar widget
FR-11.4	Authorization : Generate widget code	3	No authorization is needed for generating widget code
FR-12	Update user account	3	A registered user should be able to update their user account to set new email, password, or change organization
FR-12.1	Update user account : Password confirmation	3	When a registered user want to change their password they have to validate it by providing their old password
FR-12.2	Update user account : Email confirmation	3	When a registered user wants to change their email address this has to be confirmed with an email containing a link to the new email address. The user has to click this link to validate the email address.
FR-13	Recover lost password	2	A registered user should be able to get a new password without knowing their previous password
FR-13.1	Recover lost password : Security	3	Will use email to avoid untrusted third parties use the recovery system for gaining access to accounts

3.3 Geographical information

ID	Name	Priority	Description
FR-14	Refinement	2	The location should be refinable by different levels: none, country, region
FR-15	Google Maps Integration	2	The application should be integrated with the Google Maps API.
FR-15.1	Google Maps	2	A registered user should be able to choose

	Integration : Event Location		event location with Google Maps when creating events.
FR-15.2	Google Maps Integration : Region Grouping	4	Events should be grouped together by region using distance calculation.
FR-16	Filtering	1	The widget should have a filter for the appropriate location level.
FR-16.1	Filtering : On creation	2	The filter should be set when creating the widget code.
FR-16.2	Filtering : After creation	3	The filter should be modifiable after creating the widget.
FR-17	Location Storage	1	Location data should be saved in plaintext and coordinate format in the database.
FR-18	Localization	3	The calendar widget should be available in English, Swedish, Danish, Norwegian, and Icelandic.

3.4 Other Functional Requirements

ID	Name	Priority	Description
FR-19	Generate Widget Code	3	Any Visitor should be able to generate the client side code for embedding the calendar on their website.

3.5 Quality Requirements

ID	Name	Priority	Description
QR-1	Response Time	4	The system should always respond within 2 seconds from the request being sent.
QR-2	Recovery from failure	3	If a failure occurs resulting in a crash of the system it should restart to a working state.
QR-3	Reusability	3	The software should be able to be implemented into any domain dealing with location based events in a calendar setting.
QR-4	Availability	4	The service should never be down because of our system, but might be down due to hosting or other changes in the environment

3.6 Platform Requirements

ID	Name	Priority	Description
PR-1	Open Source Software	2	All third party software should be licensed under an open and free software license.
PR-2	Responsiveness	5	The application should be responsive and fit for use on desktop, tablet, and mobile.

3.7 Process Requirements

ID	Name	Priority	Description
ProcR-1	Open Source	1	The source code for this project should be available and under an open license.

4. Scenarios

4.1 Visitor

4.1.1 Register account

Mary navigates to the account registration page of the dashboard. She enters her email, mary@xyzmail.com and her password, "helloworld123" in both the password and password confirmation boxes. And since Mary is a member of the organization "Folkmusik in Vaxjo" she selects that organization. She then clicks on the "Register" button. She is then redirected to a page and sees a message saying "A confirmation email has been sent" on the page. Mary opens the email and clicks on the confirmation link which redirects her to a page with a message saying "Registration has been sent to the administrators". A little while later she receives a new email saying that the registration was successful. She is now able to log in to the dashboard.

4.1.2 View calendar

Mary navigates to a page with the calendar widget embedded, www.folkmusicinvaxjo.com. The calendar is enabled to show events in Vaxjo. She is happy with what she sees.

4.1.3 Change region

Mary navigates to a page with the calendar widget embedded, www.folkmusicinvaxjo.com. The calendar is enabled to show events in Vaxjo. She clicks the country selection drop down and selects "All". The calendar widget then changes from showing events in Vaxjo to showing all events.

4.1.4 View single event in widget

Mary navigates to a page with the calendar widget embedded, www.folkmusicinvaxjo.com. She clicks on a specific event in the calendar widget taking place on March 12th. She then clicks on the "Event details" button of that event. She is shown the details of that event.

4.1.5 Widget language selection

Mary navigates to a page with the calendar widget embedded, www.folkmusicinvaxjo.com. The calendar is automatically set to be in Swedish. She clicks on the language selection dropdown and selects "English". The calendar then changes from being in Swedish to being in English.

4.1.6 Widget switch to Maps view

Mary navigates to a page with the calendar widget embedded, www.folkmusicvaxjo.com. As default Mary is presented with a calendar view which list upcoming events. Mary switches to the maps view which only shows the locations of the events by clicking the maps icon in the

widget interface. The system then rerenders the view to be presented as a map with locations of events.

4.2 Administrator

4.2.1 Add administrative privileges

John navigates to the login page for the dashboard. He inputs his credentials for his administrator account and logs in. In the dashboard he navigates to the members page. He selects the user with the email bob@xyz.com and change the users access level to “Admin”. When the system asks him if he is sure he answers “yes”. He then logs out from the system.

4.2.2 Remove any active event

John navigates to a page with the calendar widget and see an event which he deems should not be on this calendar. John navigates to the login page for the dashboard, inputs his credentials and logs in. On the dashboard he finds the specific event in the list and clicks the “Remove event” button. The system asks him if he is sure he want to complete this action, and he answers “yes”. He then logs out and navigates back to the page with the calendar widget and checks that the post is removed.

4.2.3 Accept Pending Registration

John navigates to the login page for the dashboard, inputs his credentials and logs in. He notices that there is a pending registration. He looks at the registration and determines that this is someone from his organization so he accepts it.

4.2.4 Remove administrative privileges within organization

John navigates to the login page for the dashboard. He inputs his credentials for his administrator account for organization “Folkmusik in Växjö” and logs in. In the dashboard he navigates to the members page. He selects the user with the email jill@xyz.com and clicks on the “Remove administrative privileges”. When the system asks him if he is sure about this action he answers “yes”. He then logs out from the system.

4.3 Super Administrator

4.3.1 Remove administrative privileges

Simon navigates to the login page for the dashboard. He inputs his credentials for his super administrator account and logs in. In the dashboard he navigates to the members page. He selects the user with the email jill@xyz.com and clicks on the “Remove administrative privileges”. When the system asks him if he is sure about this action he answers “yes”. He then logs out from the system.

4.3.2 Add super administrative privileges

Simon navigates to the login page for the dashboard. He inputs his credentials for his super administrator account and logs in. In the dashboard he navigates to the members page. He selects the user with the email jill@xyz.com and change “Access Level” to “Super Administrator”. When the system displays a warning and asks him if he is sure about this action he answers “yes”. He then logs out from the system.

4.4 Registered User

4.4.1 Create event

Peter logs in to the system and navigates to the page where he can create events. He clicks on “Create event” and starts to fill in the event details. He enters a name for the event and writes a description. He chooses the location of the event and when it will be held. It’s a recurring event, so Peter chooses when the event recurs. He also uploads an image to go with the event description. When he is done, he clicks on “Save event” and is presented with a preview of the entered details. As he is content with what he sees, he clicks on “Confirm event” to save the newly created event in the system.

4.4.2 Delete event

Peter logs in to the system and navigates to the page where he can see a list of the events he has created. He clicks on “Delete” and answers “Yes” when the system asks him if he is sure that he wants to delete the event.

4.4.3 Update event

Peter logs in to the system and navigates to the page where he can see a list of the events he has created. He clicks on the event he wants to update and is taken to the page where he can edit the event’s details. He updates the description of the event and changes the date and time it takes place. The location has changed too, so he chooses a new location. When he is done with updating the event, he clicks on “Preview” and is presented with a preview of the updated event details. He clicks on “Save” and the updated details are saved in the system.

4.4.4 Login

Peter navigates to the dashboard and get asked by the system to login. Peter provides email and password and then clicks login. The system stores a session on the client and redirects Peter to the main dashboard page.

4.4.5 Logout

Peter is viewing the dashboard and clicks logout. The system removes the session from the client and redirects Peter to the login page.

4.4.6 Update Account

Peter logs in to the dashboard and navigates to the update account page by clicking update account. Peter then provide a new email and new password. He also provides his old password for validation and clicks save. The system checks that the email is valid, and that the password matches. He receives a confirmation email on the new email address and he clicks the link.

4.4.7 Recover password

Peter navigates to the login page for the dashboard. He clicks the Forgot password button and is redirected to the recovery page. He writes his email address he used to register his account and requests a password reset.

Peter then opens his email client and opens the email he received from the system and clicks on the link in the email. He inputs his new password “Very5trongPa22w0rd” in the box and the same in the comparison box, and clicks change password. He is redirected to the login page and tries to log in with his new password which grants him access to the dashboard.

4.5 Organization

4.5.1 Generate Widget

The organization's website administrator Bob navigates to the login page to the dashboard. He clicks on the “Generate Widget Code” link and is redirected to the widget generator page. Bob selects the default region and clicks on the “Generate” button. Code snippets are then generated and displayed on the page together with instructions on how to use these code snippets. Bob then copies the code and inputs in into the code for the calendar page on the organization website.

5. Use Cases

UC1: Login

Actors

- Registered User

Goals

- The registered user is authenticated

Preconditions

- UC2: Logout
- UC3: Register

Preconditions

- The user has an account in the system
- The user is not already logged in to the system
- The user has navigated to the login page

Postconditions

- The user is authenticated and redirected to the main page of the dashboard

Summary

The user enters their details in the login form. If the details are correct, the user is authenticated and sent to the main page of the dashboard. If not, they are presented with an error message.

Steps

Main Flow

1. The user enters email and a password in the appropriate inputs
2. The user issues the 'login' command
3. The system checks if a user with this email address exists
4. The system checks if the password is correct
5. The system stores an authentication token in the user's client
6. The authenticated user is redirected to the "My events"-page

Alternative Flow 1 - Invalid username

1. Main flow 1 - 3

2. The system fails to find a user with this address
3. The system sends an error message to the user's client
4. The error message is displayed to the user

Alternative Flow 2 - Invalid password

1. Main flow 1 - 4
2. The system determines that the password is incorrect
3. The system sends an error message to the user's client
4. The error message is displayed to the user

UC2: Log out

Actors

- Registered User

Goals

- The user is logged out (no longer authenticated)

Related Use Cases

- UC1: Login
- UC3: Register

Preconditions

- The user is logged in to the system

Postconditions

- The user is redirected to the login page for the dashboard

Summary

The user issues the 'log out' command and is logged out of the system. The authentication token stored in their browser is cleared.

Steps

Main Flow

1. The user issues the 'log out' command
2. The system clears the authentication token in the user's client
3. The user is redirected to the login page for the dashboard

UC3: Register Account

Actors

- Visitor
- Administrator

Goals

- The visitor has an account in the system

Related Use Cases

- UC1: Login
- UC2: Logout

Preconditions

- The visitor is not already a registered user
- The user has navigated to the create an account page

Postconditions

- An account is created for the visitor (now registered user)

Summary

The visitor enters their details in the account registration form. The system checks that the details are all valid. If they are not, the visitor is presented with information concerning why these details are not valid. If they are, the system stores their information in the database, authenticates the visitor, and finally redirects the visitor to the login page of the dashboard.

Related Use Cases

- UC1: Login
- UC5: Update user account
- UC4: Recover lost password

Steps

Main Flow

1. The visitor enters an email, password, and password confirmation in the provided inputs.
2. The visitor selects no organization.
3. The visitor checks the reCaptcha checkbox
4. The visitor issues the 'Register' command
5. The system checks that the email format is valid
6. The system checks that the email is not already in use
7. The system checks that the password matches the password confirmation
8. The system checks that the password is valid
9. The system sends a confirmation email to the email address.
10. The system displays message "Registration is being verified"
11. The visitor opens the email and clicks on the link.
12. The system adds the registration to "pending registration" of the administrators for NordVisa.
13. An Administrator accepts the requested registration
14. The system stores the user information
15. The system sends an email to visitor saying that the registration was successful

Alternative Flow 1 - reCaptcha fails validation

1. Main flow 1 - 4
2. The system determines that the visitor has failed reCaptcha validation
3. The system displays an error message to the visitor

Alternative Flow 2 - Invalid email format

1. Main flow 1 - 5
2. The system determines that the email format is invalid
3. The system displays an error message to the visitor

Alternative Flow 3 - Email already exists

1. Main flow 1 - 6
2. The system determines that the email is already in use in the system
3. The system displays an error message to the visitor

Alternative Flow 4 - Password does not match password confirmation

1. Main flow 1 - 7
2. The system determines that the password and password confirmation do not match
3. The system displays an error message to the visitor

Alternative Flow 5 - Password is invalid

1. Main flow 1 - 8
2. The system determines that the password is invalid
3. The system displays an error message to the visitor

Alternative Flow 6 - Selects organization

1. The visitor enters an email, password, and password confirmation in the provided inputs.
2. The visitor selects an existing organization
3. The visitor checks the reCaptcha checkbox
4. The visitor issues the 'Register' command
5. The system checks that the email format is valid
6. The system checks that the email is not already in use
7. The system checks that the password matches the password confirmation
8. The system checks that the password is valid
9. The system sends a confirmation email to the email address.
10. The system displays a message saying an email has been sent
11. The visitor opens the email and clicks on the link.
12. The system displays message "Registration has been sent to the administrators"
13. The system adds the registration to "pending registration" of the administrators for the selected organization, and also for global and super administrators.
14. An Administrator accepts the requested registration
15. The system stores the user information
16. The system sends an email to visitor saying that the registration was successful

Alternative Flow 7 - Creates organization

1. The visitor enters an email, password, and password confirmation in the provided inputs.
2. The visitor creates new organization by writing the name
3. The visitor checks the reCaptcha checkbox
4. The visitor issues the 'Register' command
5. The system checks that the email format is valid
6. The system checks that the email is not already in use
7. The system checks that the password matches the password confirmation
8. The system checks that the password is valid
9. The system sends a confirmation email to the email address.
10. The system displays a message saying an email has been sent
11. The visitor opens the email and clicks on the link.
12. The system displays message "Registration has been sent to the administrators"
13. The system adds the registration to "pending registration" of the administrators for NordVisa.
14. An Administrator accepts the requested registration
15. The system stores the user information
16. The system sends an email to visitor saying that the registration was successful

Alternative Flow 8 - Organization already exists

1. Main flow
2. Visitor (now registered user) is added to the existing organization

Alternative Flow 9 - Confirmation link expires

1. Main flow 1 - 10
2. After 24 hours the system removes the registration and link is unusable
3. The visitor opens the email and visits the link
4. The system displays a "Link expired" message

Alternative Flow 10 - Administrator denies registration

1. Main flow 1 - 13
2. Administrator denies registration
3. The systems sends an email the visitor saying that the registration was denied.

UC4: Recover lost password

Actors

- Registered User

Goals

- Change the password of a specific user without having the password

Preconditions

- Registered user has access to the email address registered with the account
- Registered user is not logged in.
- Registered user is on the login screen to the dashboard

Postconditions

- Password has been changed

Summary

When a user wants to reset their password but don't know their current password this can be done by email. The user issue the "Forgot password" command and gives an email address which generates a unique link to a "Change password" page for the account with a matching email address. The link is then sent to the email address so it can only be opened by someone with access to the email account. The link is active for 24 hours after being generated and after that it is expired and can not be used. The link can only be used once. Once on the change password page the user can change their password. The password is given twice to avoid mistakes, and the same rules for passwords as during registration.

Steps

Main Flow

1. User issues "Forgot password" command
2. User inputs an email address
3. User issues "Request password reset" command
4. System validates email address format
5. System checks if the email address is registered in the system
6. A unique link to a "Change password" page is generated
7. An email with the link is sent to the provided email address
8. System tells the user that the email is sent
9. User receives an email to the mail address provided.
10. User click on the link
11. User fills in new password twice
12. User issues "Change password" command
13. Check if password is valid
14. Check if passwords match each other
15. System display a success message
16. System redirects to the login screen for the dashboard

Alternative Flow 1 - Invalid email address format

1. Main flow 1 - 4
2. System tells user that the email is not valid

Alternative Flow 2 - Email is not registered

3. Main flow 1 - 5
4. System tells user that the email is not registered

Alternative Flow 3 - Expired link

1. Main flow 1 - 9
2. The user clicks on the link after 24 hours
3. A message is displayed to the user saying the link has expired

Alternative Flow 4 - Used link

1. User goes through all steps of Main Flow
2. User clicks the link in the email again
3. A message is displayed to the user saying the link has expired

Alternative Flow 5 - Invalid password

1. Main flow 1 - 10
2. User fills in an invalid password twice
3. User issues "Change password" command
4. Check if password is valid

5. System displays message saying the passwords is invalid and lets the user fill in a new password

Alternative Flow 6 - Passwords not matching

1. Main flow 1 - 10
2. User fills in new password twice, but without matching
3. User issues "Change password" command
4. Check if password is valid
5. Check if passwords match each other
6. System displays message saying the passwords did not match and lets the user fill in the information again

UC5: Update User Account

Actors

- Registered User

Goals

- For a Registered User to be able to change their password or email or both.

Related Use Cases

- UC3: Register

Preconditions

- Registered User is logged into his/her account.
- Registered User is on the Update/Edit section of the dashboard.

Postconditions

- Account is Updated

Summary

The Registered User edits his or her account. Update password, email or both.

Main Flow - Update both email and organization.

1. Registered User inputs new email.
2. Registered User inputs existing organization.
3. Registered User issues "Save" command
4. System confirms email is of valid format.
5. System send confirmation email to the email address

6. System informs user that a confirmation email has been sent to the new email address
7. User opens the email and clicks on the link.
8. Systems displays message saying that the email has been changed.
9. Administrator approves organization change
10. System updates data entries with the new information.

Alternative Flow 1 - Update Email

1. Registered User inputs new email.
2. Registered User issues "Save" command.
3. System confirms email is correct
4. System send confirmation email to the email address
5. System informs user that a confirmation email has been sent to the new email address
6. User opens the email and clicks on the link.
7. Systems displays message saying that the email has been changed.
8. System updates data entries with the new information in the database

Alternative Flow 2 - Update Password

1. Registered User inputs old password.
2. Registered User inputs new password.
3. Registered User confirms new password by writes it again.
4. Registered User issues "Save" command
5. System hash and salt password.
6. System updates data entries with the new in the database.

Alternative Flow 3 - Invalid Email

1. Alternative flow - Update email 1-2 or Main flow 1-5
2. System fails to confirm email. Email is invalid.
3. Error message is outputted to the Registered Users.

Alternative Flow 4 - Email already exists

1. Alternative flow - Update email 1-2 or Main flow 1-5
2. System fails to confirm email. Email already exists.
3. Error message is outputted to the Registered Users.

Alternative Flow 5 - Confirmation link expires

1. Main flow 1 - 9
2. 24 hours passes by and the link expires and can't be used

Alternative Flow 6 - New password does not match or invalid

1. Alternative Flow - Update Password 1 - 4 or Main flow 1-5
2. System is not able to match passwords.
3. Error message is outputted to the Registered User.

Alternative Flow 7 - Old password is not correct

4. Alternative Flow - Update Password 1 - 4 or Main flow 1-5
5. Old password does not what the registered user put as their old password
6. Error message is outputted to the Registered User.

Alternative Flow 8 - Change to another existing organization

1. Registered user selects an existing organization
2. Registered User issues "Save" command
3. An administrator accepts the organization change and creation
4. System saves changes

Alternative Flow 9 - Change to new organization

1. Registered User inputs the name of the new organization
2. Registered User issues "Save" command
3. System check if organization already exists. If so join that organization
4. An administrator of the organization approves the change
5. Changes are saved.

UC6: Add administrative privileges

Actors

- Administrator

Goals

- Give a specific account the privileges of an administrator

Related Use Cases

- UC7: Remove administrative privileges
- UC8: Add super administrative privileges

Preconditions

- The administrator has to be logged in to the dashboard
- The account the administrator is logged into has to have administrator privileges.
- The target account has to already exist.
- The target account is not already an administrator

Postconditions

- The administrator is redirected to the main page of the dashboard.

Summary

The administrator select the specific account to grant the administrative privileges to and issues the make administrator command. The system the asks the administrator if he/she is sure about this action. If yes the target account is granted administrative privileges, if no then no change is done the the target account.

Steps

Main Flow

1. The administrator selects the target account
2. The administrator issues the “Administrator” command
3. The system asks the administrator if he/she is sure to grant administrator privileges to the target account
4. The administrator issues the “Yes” command
5. The system grants administrator privileges to the target account
6. The administrator is redirected to the main page of the dashboard.

Alternative Flow 1 - Administrator undoes changes

1. The administrator select the target account, the account to be altered
2. The administrator issues the “Administrator” command
3. The system asks the administrator if he/she is sure to grant administrator privileges to the target account
4. The administrator issues the “No” command
5. The administrator is redirected to the main page of the dashboard without granting privileges to the target account.

UC7: Remove administrative privileges

Actors

- Super administrator
- Administrator

Goals

- Remove any user privileges from an administrator account, making it a normal account which can only edit events created by itself.

Related Use Cases

- UC6: Add administrative privileges
- UC8: Add super administrative privileges

Preconditions

- The target account has to already exist.
- The target account has administrator privileges, but not super administrator privileges
- The account issuing the removal of privileges has to have super administrator privileges
- The super administrator or administrator has to be logged in to the dashboard.

Postconditions

- The super administrator or administrator is redirected to the main page of the dashboard

Summary

Only a super administrator can remove administrator privileges from an administrator. The super administrator select the target account which will have its administrative privileges remove, and issues the “Remove administrator privileges”. The system will then ask the super administrator if he/she is sure about the decision. If yes then the system removes administrator privileges from the target account, and if no then nothing is changed on the target account. There is an exception where within an organization any administrator can remove any other administrators privileges.

Steps

Main Flow

1. The super administrator selects the target account
2. The super administrator issues the “Remove administrator privileges” command
3. The system asks the super administrator if he/she is sure to remove administrator privileges from the target account
4. The super administrator issues the “Yes” command
5. The system removes administrator privileges from the target account
6. The super administrator is redirected to the dashboard

Alternative Flow 1 - Super administrator undoes changes

1. The super administrator selects the target account
2. The super administrator issues the “Remove administrator privileges” command
3. The system asks the super administrator if he/she is sure to remove administrator privileges from the target account
4. The super administrator issues the “No” command
5. The super administrator is redirected to the dashboard without removing any privileges from the target account.

Alternative Flow 2 - Remove privileges within organization

1. The administrator selects the target account

2. The administrator issues the "Remove administrator privileges" command
3. The system asks the administrator if he/she is sure to remove administrator privileges from the target account
4. The administrator issues the "Yes" command
5. The system removes administrator privileges from the target account
6. The administrator is redirected to the dashboard

UC8: Add super administrative privileges

Actors

- Super administrators

Goals

- Give a registered account super administrator privileges

Preconditions

- The target account has to already exist.
- The target account is not already a super administrator
- The account granting privileges has to have super administrator privileges
- The super administrator has to be logged in to the dashboard

Postconditions

- The super administrator is redirected to the main page of the dashboard.

Summary

A super administrator can add super administrator privileges to an account. The super administrator does this by selecting a target account and issuing the "Make super administrator command" on that account. The super administrator is then prompted with a warning because this can't be reversed, even by a super administrator. The super administrator is also asked if he/she wants to proceed. If yes the target account is granted super administrator privileges, if not then no change is done to the target account.

Steps

Main Flow

1. The super administrator selects the target account
2. The super administrator issues the "Make super administrator" command
3. The system asks the super administrator if he/she is sure to grant administrator privileges to the target account. This includes a warning saying this can not be reversed.
4. The super administrator issues the "Yes" command
5. The system grant super administrator privileges to the target account

Alternative Flow 1 - Super administrator undoes changes

1. The super administrator selects the target account
2. The super administrator issues the “Make super administrator” command
3. The system asks the super administrator if he/she is sure to grant administrator privileges to the target account. This includes a warning saying this can not be reversed.
4. The super administrator issues the “No” command
5. The system redirects to the dashboard without granting any privileges to the target account.

UC9: Create event

Actors

- Registered user

Goals

- The event is created and stored in the system

Related Use Cases

- UC10: Update Event
- UC11: Delete Event
- UC12: View Event Dashboard
- UC13: List Events

Preconditions

- The user needs to be logged in to the system
- The user is logged in to the dashboard

Postconditions

- The user is redirected to the dashboard

Summary

The user enters the event details and chooses to store the information in the database. The user specifies information such as event name, description, location, date and time, duration, whether the event is a recurring event, and website information. The user uploads images to the system.

Steps

Main flow - Create event

1. The user enters a name for the event
2. The user chooses a location for the event
3. The user enters a description for the event
4. The user chooses a date and time for the event
5. The user enters the duration of the event
6. The user specifies if the event is recurring
 - 6.1. The user chooses when the event recurs
7. The user issues the "Choose image" command to upload an image
 - 7.1. The open file dialog appears
 - 7.2. The user chooses an image to upload
 - 7.3. The system saves the image
8. The user issues the "Preview" command
9. The system presents a preview of the event
10. The user issues the "Save" command
11. The system adds the current date and time to the event details
12. The system stores the event

Alternative Flow 1 - Continue edit event

1. The user follows the Main flow 1–9
2. The system presents a preview of the event
3. The user issues the "Edit" command
4. The system redirects the user back to the edit event page

Alternative Flow 2 - Required details not entered

1. The user fails to enter any of the following details:
 - Name
 - Location
 - Description
 - Date and time for the event
2. The user issues the "Preview" command
3. The system informs the user about the required details
4. The system does not store the event

Alternative Flow 3 - System cannot process image

1. The user issues the "Choose image" command to upload an image
2. The open file dialog appears
3. The user chooses an image to upload
4. The system fails to process the image
5. The system presents an error message

UC10: Update Event

Actors

- Registered user

Goals

- Update the event and the new event is stored in the system.

Related Use Cases

- UC9: Create Event
- UC11: Delete Event
- UC12: View Event Dashboard
- UC13: List Events

Preconditions

- The user need login to the system
- The user is viewing “My Events”-page

Postcondition

- The user is already save the changes in the new web page.

Summary

The user updates the event details and store the information in the system.

Related use cases

- UC9: Create Event
- UC11: Delete Event
- UC12: View Single Event in dashboard

Steps

Main flow

1. The user issues “Edit”-command.
2. The user update the event with the new information.
3. The user issues “Preview”-event.
4. A preview is shown to the user.
5. The user issues “Save”-command.
6. The system stores the new event in the system.

UC11: Delete event

Actors

- Registered user

Goals

- The event is removed from the system

Related Use Cases

- UC9: Create Event
- UC10: Update Event
- UC12: View Event in Dashboard
- UC13: List Events

Preconditions

- The user needs to be logged in to the system
- The user is viewing “My Events”-page

Postconditions

- The Event is deleted

Summary

The user is viewing a single event and chooses to delete it. The system removes the event.

Steps

Main flow

1. The user issues the “Delete event” command
2. The system asks the user to confirm the deletion
3. The user confirms the deletion
4. The system removes the event

Alternative flow 1 - User undoes changes

1. The user issues the “Delete event” command
2. The system asks the user to confirm the deletion
3. The user cancels the deletion
4. The system doesn’t remove the event

UC12: View Event in Dashboard

Actor

- Registered User

Goals

- The user is able view single event information.

Related Use Cases

- UC9: Create Event
- UC10: Update Event
- UC11: Delete Event
- UC13: List Events

Preconditions

- The user is signed into the dashboard.

Postcondition

- The event information is visible on the screen.

Summary

The user selects a single event, issues the command “view event details”, and the event’s details are presented on the screen.

Steps

Main flow

1. The user is viewing “My Events”-page
2. The user issues view command
3. System outputs event details.

UC13: List Events

Actors

- Registered User
- Administrator

Goals

- Show all events which the currently logged in user can access and manage.

Related Use Cases

- UC9: Create Event
- UC10: Update Event
- UC11: Delete Event
- UC12: View Event Dashboard

Preconditions

- User is logged in to the system

Postconditions

- All events accessible to the signed in user is displayed

Summary

When an registered user navigates to the main page of the dashboard, the system retrieves a list of all events created by the user and presents them in a list.

Related use cases

- UC12: View single event in dashboard

Steps

Main flow

1. The Registered User navigates to the main page of the dashboard (also know as “My Events”-page)
2. System check if you have administrator privileges
3. System determined that the user do not have administrator privileges
4. Gets all active events created by the user
5. A list of the events are presented

Alternative Flow 1 - Administrator flow

1. The Administrator navigates to the main page of the dashboard (also know as “My Events”-page)
2. System check if you have administrator privileges
3. Gets all active events in the system
4. A list of the events are presented

UC14: Generate Widget Code

Actors

- Organization

Goals

- Generate code snippets to be embedded into websites.

Preconditions

- Organization is on the page for generating Widget Code.

Postconditions

- Code snippet for Widget is outputted on the screen for the Organization to copy and paste into their own website.

Summary

An Organization inputs different criteria that it has for its widget. Based on the input code for the Widget, it is generated and outputted onto the screen for the Organization to copy.

Main Flow

1. Organization selects default region
2. Organization issues "generate widget" command
3. System generates the code snippets.
4. System outputs the code
5. Organization copies the code
6. Organization paste the code into their website.

UC15: Change region

Actors

- Visitor

Goals

- Display all events in the specified region to the user.

Preconditions

- The visitor is on a page with the client widget embedded

- The default settings set on implementation of the widget is not the region the visitor is looking for.

Postconditions

- The visitor is presented with a calendar containing all events in the selected region

Related use cases

- UC13: List Events

Summary

When a user visits the widget client they can change which events are shown in the calendar based on the where you want to look and where the events are happening. A user can pick “Show all” option to show all events in all Nordic countries, or he/she can select a specific Nordic country and only events in that specific country. There are also ways for the user to view local events based on smaller regions which is selected in a similar way to countries, or by their current location and their surrounding area.

Steps

Main Flow

1. Visitor asks to select country
2. System presents a list of all available countries, including an options for all Nordic countries
3. Visitor selects “All” option
4. System presents all events in all Nordic countries

Alternative Flow 1 - Show specific country

1. Visitor asks to select country
2. System presents a list of all available counties, including an options for all Nordic countries
3. Visitor select a country to display
4. System presents all events within the specified country.

Alternative Flow 2 - Select province/region

1. Visitor asks to select country
2. System presents a list of all available counties, including an options for all Nordic countries
3. Visitor select a country to display
4. System presents a list of regions
5. Visitor selects a region
6. System presents all events within the local area.

Alternative Flow 3 - Select region near visitor

1. Visitor issues "Events near me" command
2. System asks the visitor for maximum distance from current location
3. Visitor selects a maximum distance.
4. System presents all events within the area.

UC16: View Single Event in Widget

Actor

- Visitor

Goals

- The user is able view single event information.

Preconditions

- The visitor views the widget

Postcondition

- The event information is visible on the screen.

Summary

The visitors selects what event to view and issues command to view event details. The event is then presented to the visitor.

Steps

Main flow

1. The visitor selects the event
2. The visitor issues command view event details.
3. System outputs event details.

UC17: View Calendar

Actors

- Visitor

Goals

- For the visitor to be able to view upcoming events.

Preconditions

- None

Postconditions

- A calendar containing events in the specified region

Related use cases

- UC8: Change Region

Summary

A visitor views all events in a specific region by navigating to a page that contains the widget.

Steps

Main Flow

1. Visitor navigates to a page with widget embedded.
2. System gets all events within the default region.
3. System outputs the events to the widget.

UC18: Widget language selection

Actors

- Visitor

Goals

- Change all text in the user interface and buttons to the preferred language of the registered user in the widget.

Preconditions

- Visitor has access to page with calendar embedded

Postconditions

- All text for the user interface and menus are changes to the selected language

Related use cases

- UC19: Dashboard language selection

Summary

When the visitor navigates to the widget the widget will try to locate the visitor to select a language preferred for that country. If anything goes wrong, the person is outside the Nordic countries or does not want to share location details then the application will be set to English.

Steps

Main Flow

1. Visitor navigates to a page with the calendar widget
2. System check what country the user is in
3. System set the language to match the language of the user's location

Alternative flow 1 - Manual language change

1. Main flow 1-3
2. Visitor asks to list available languages
3. System presents a list of available languages
4. Visitor selects one alternative
5. Visitor issues "Change language" command
6. System updates the user interface to show the new language.

Alternative flow 2 - Country outside of Nordics

1. Visitor navigates to a page with the calendar widget
2. System check what country the user is in
3. System set the language to English

Alternative flow 3 - Visitor does not allow location tracking

1. Visitor navigates to a page with the calendar widget
2. System check what country the user is in, but can't because it's not allowed to.
3. System set the language to English

UC19: Dashboard language selection

Actors

- Registered User

Goals

- Change all text in the user interface and buttons to the preferred language of the registered user in the dashboard.

Preconditions

- Registered user is logged in to the dashboard

Postconditions

- All text for the user interface and menus are changes to the selected language

Related use cases

- UC18: Widget language selection

Summary

When the registered user navigates to the widget the widget will try to locate the user to select a language preferred for that country. If anything goes wrong, the person is outside the Nordic countries or does not want to share location details then the application will be set to English.

Steps

Main Flow

1. Registered User navigates to the dashboard
2. System check what country the user is in
3. System set the language to match the language of the user's location

Alternative flow 1 - Manual language change

1. Main flow 1-3
2. Registered User asks to list available languages
3. System presents a list of available languages
4. Registered User selects one alternative
5. Registered User issues "Change language" command
6. System updates the user interface to show the new language.

Alternative flow 2 - Country outside of Nordics

1. Registered User navigates to the dashboard
2. System check what country the user is in
3. System set the language to English

Alternative flow 3 - Visitor does not allow location tracking

1. Registered User navigates to the dashboard

2. System check what country the user is in, but can't because it's not allowed to.
3. System set the language to English

UC20: Widget switch to Maps View

Actors

- Visitor

Goals

- Change the view of the widget to only show a map with the locations of the upcoming events.

Preconditions

- Visitor is viewing a page with the widget embedded.

Postconditions

- The maps view is visible on the screen.

Summary

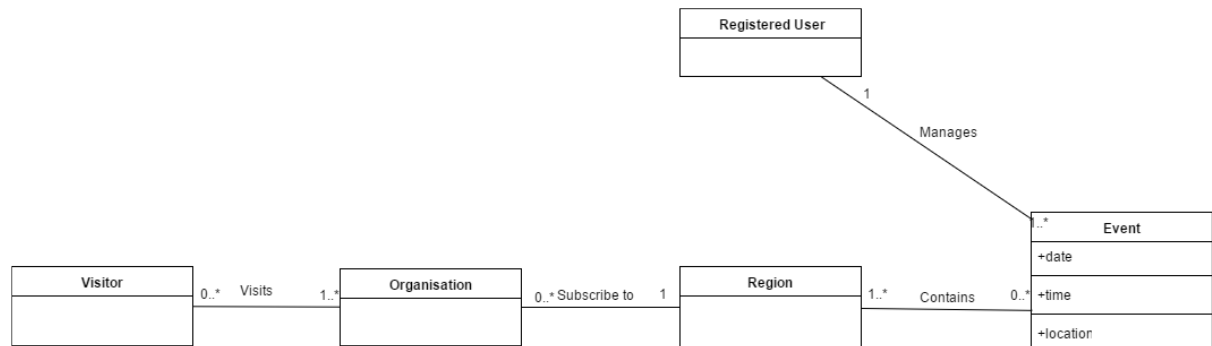
When a visitor views the widget they should be able to switch to maps view to only view the location of the upcoming events.

Steps

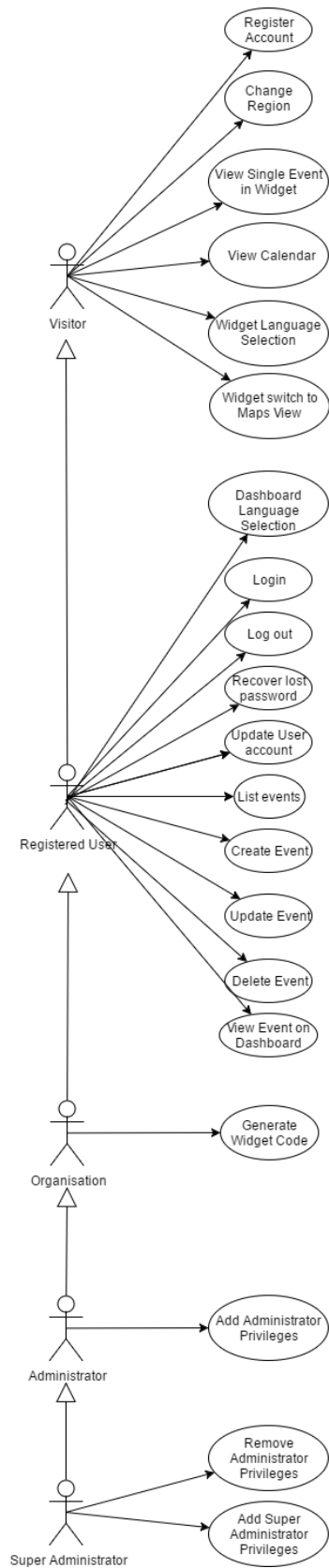
Main Flow

1. The visitor issues "Switch to maps view"- command.
2. The system re renders the widget interface to be presented as a map.

Appendix I: Domain Model

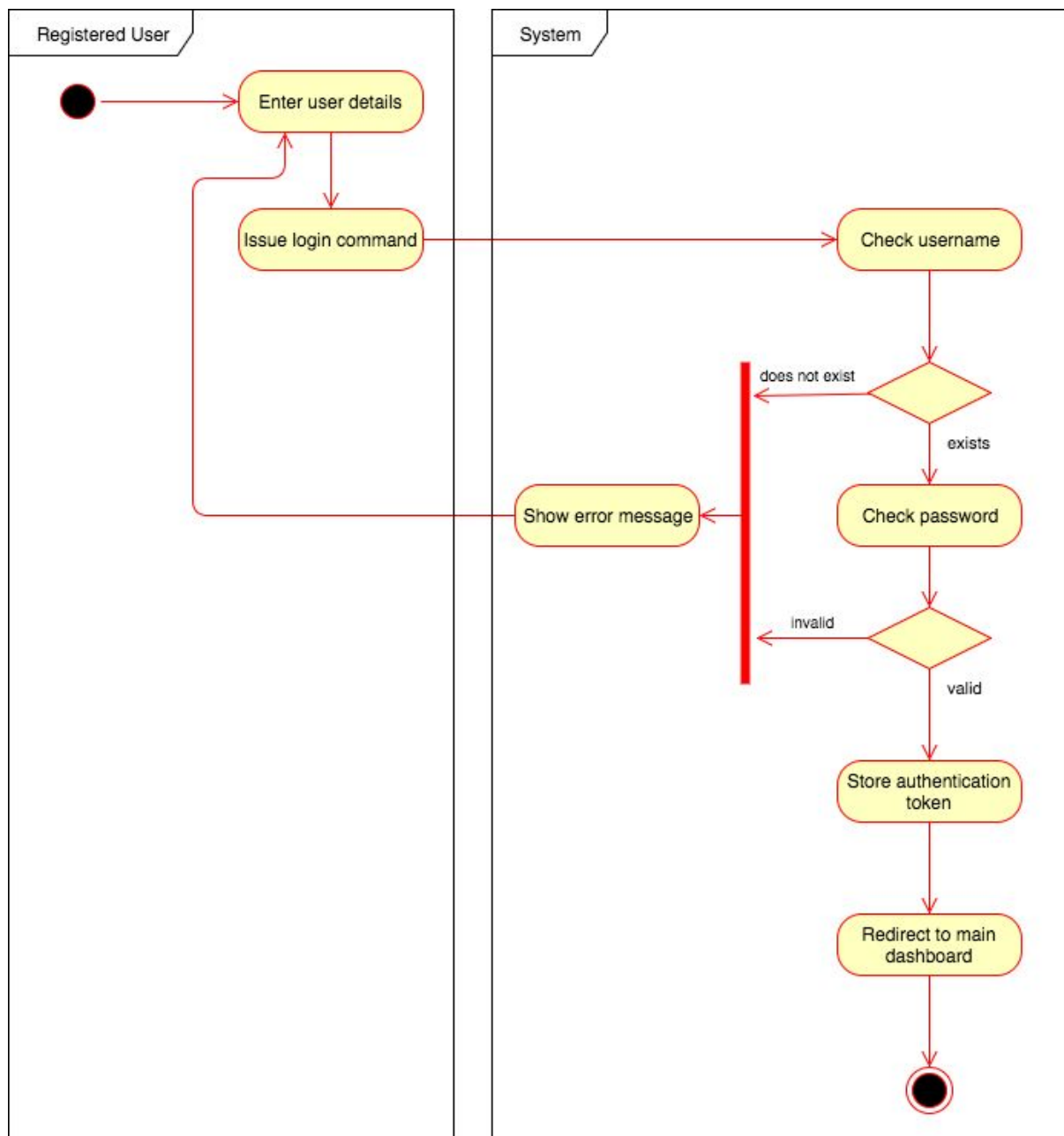


Appendix II: Use Case Diagram

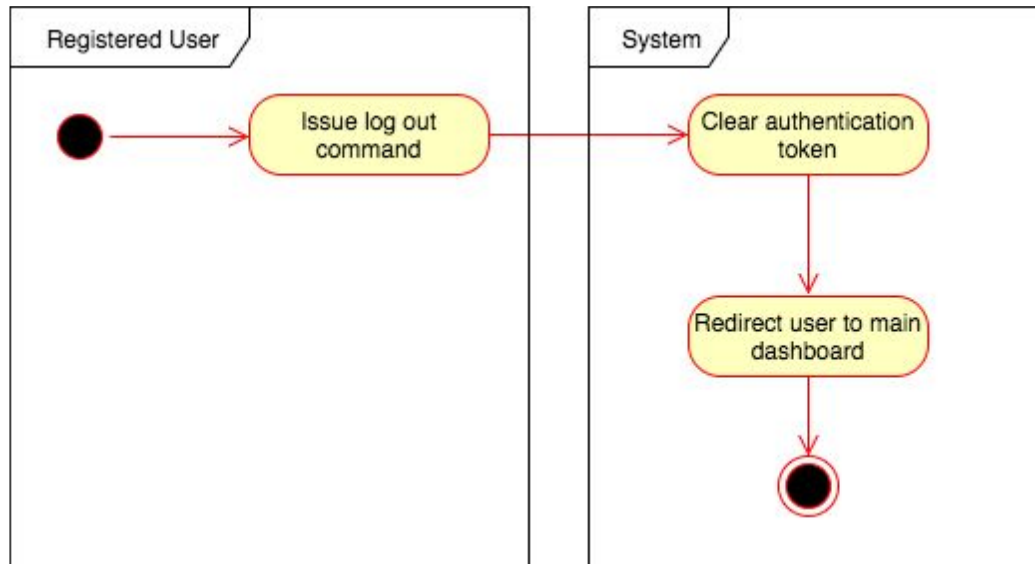


Appendix III: Interaction diagrams

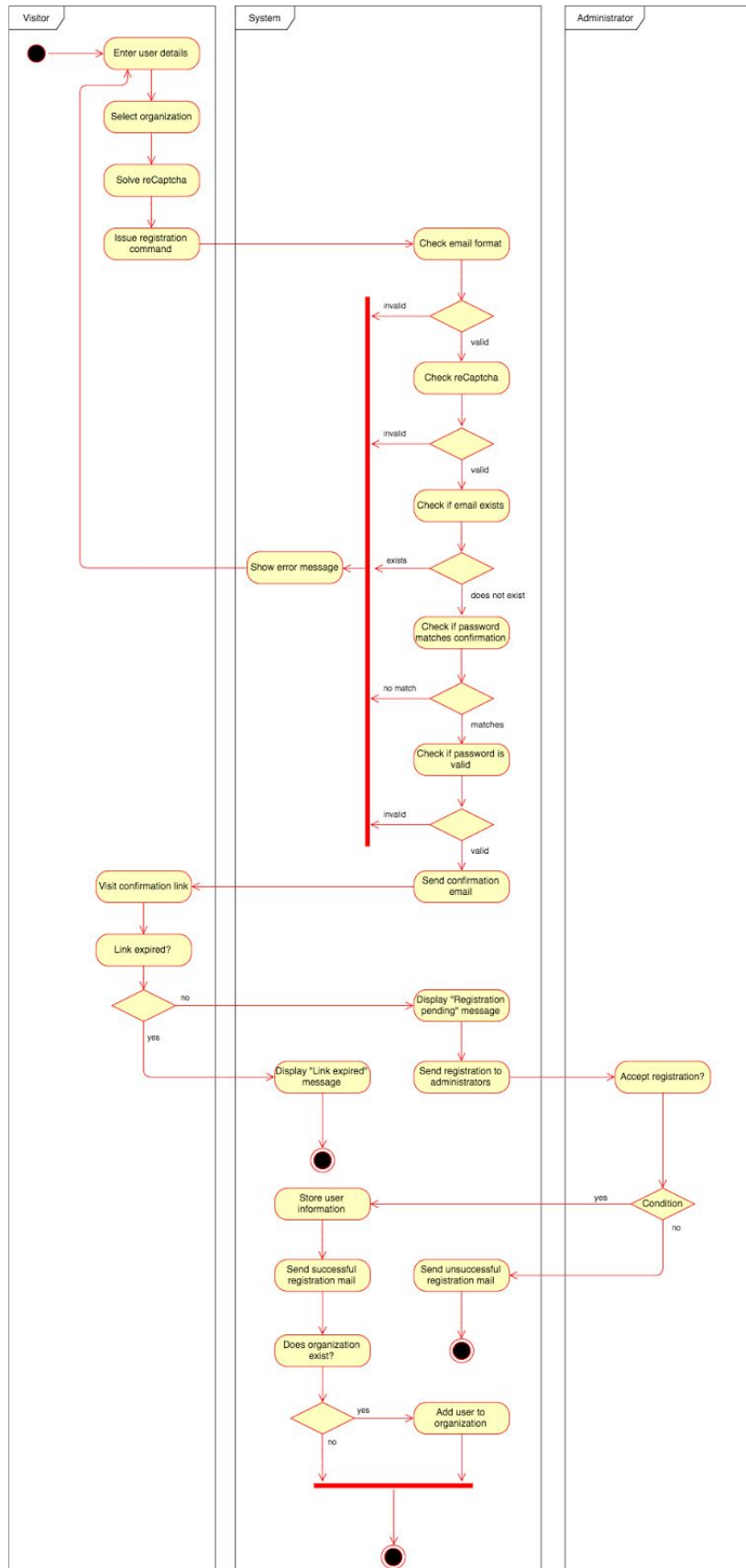
UC1: Login



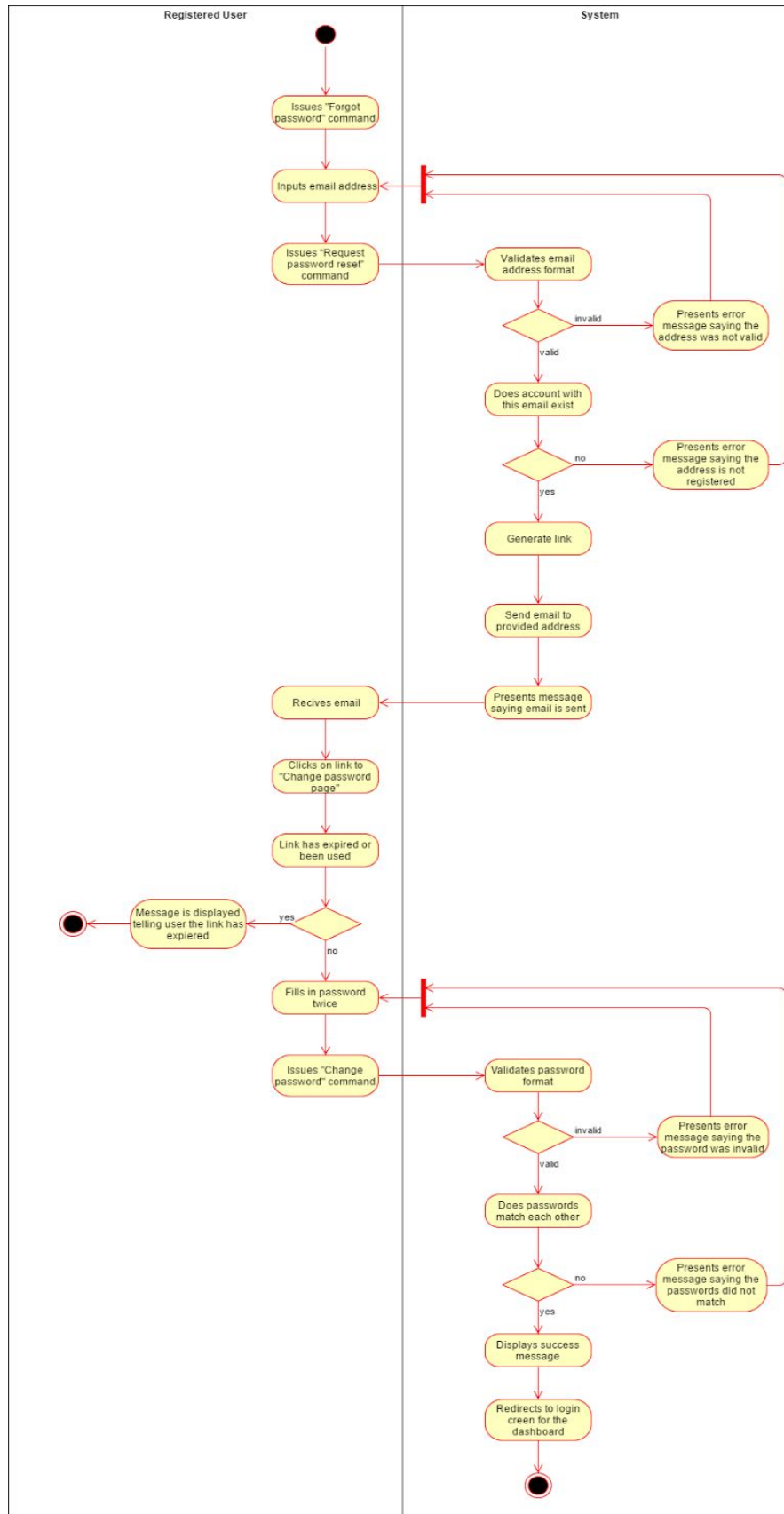
UC2: Log out



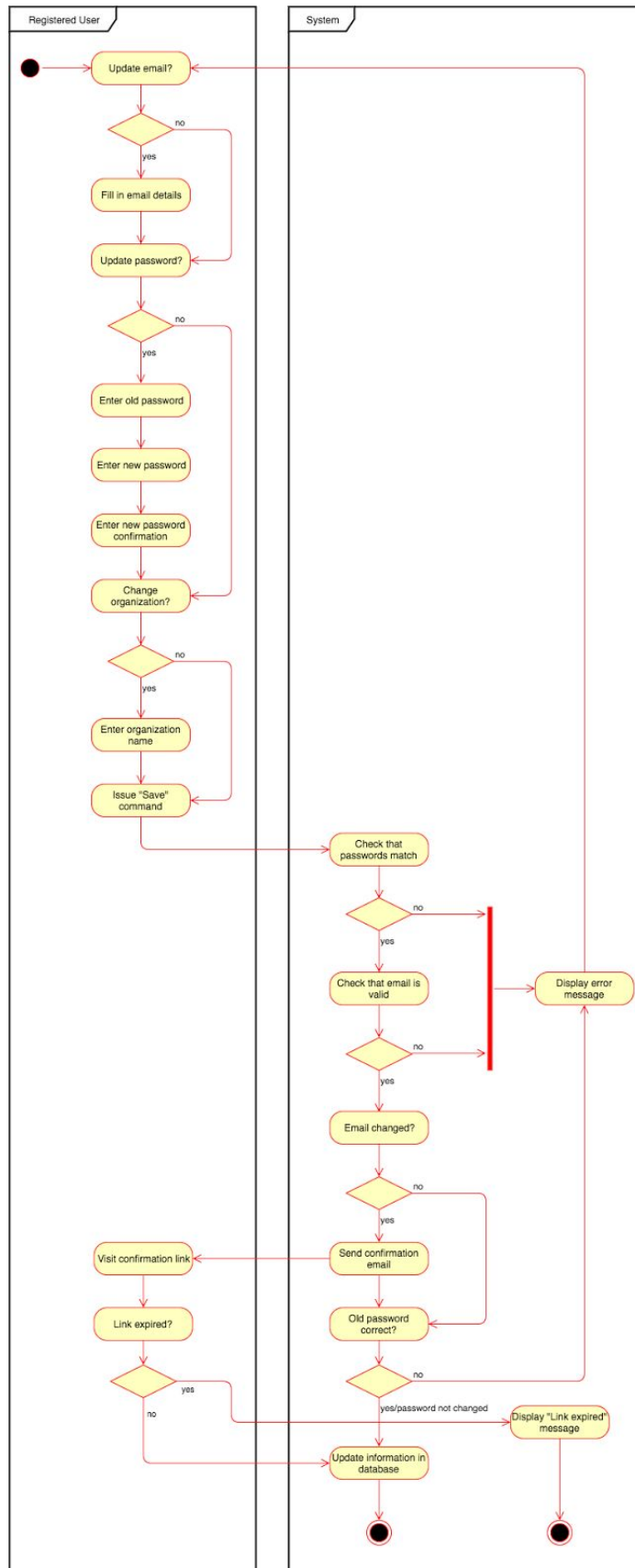
UC3: Register Account



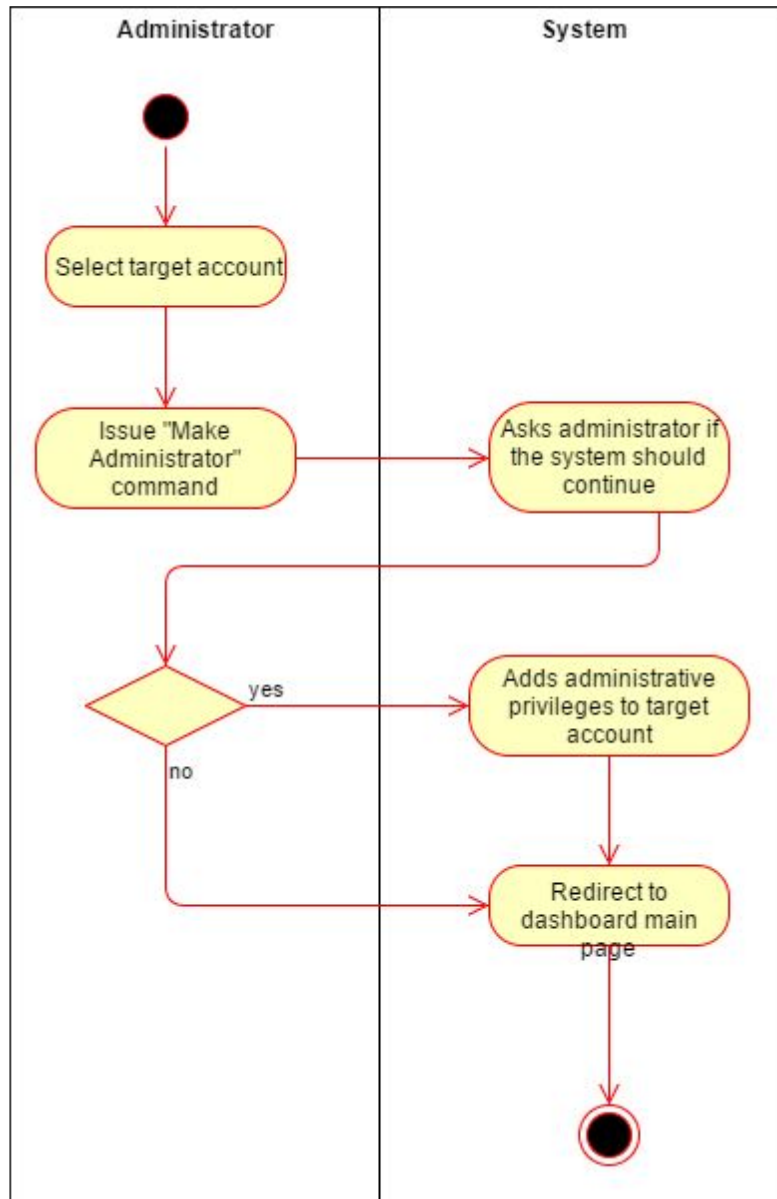
UC4: Recover lost password



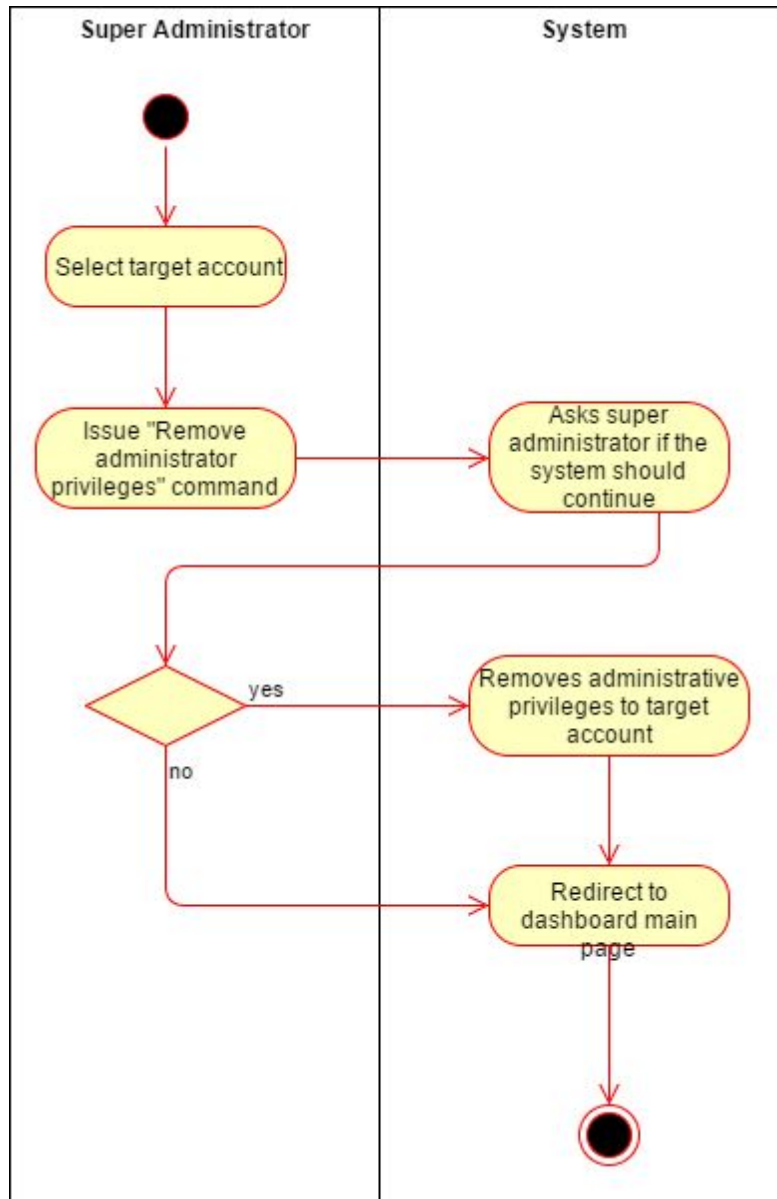
UC5: Update user account



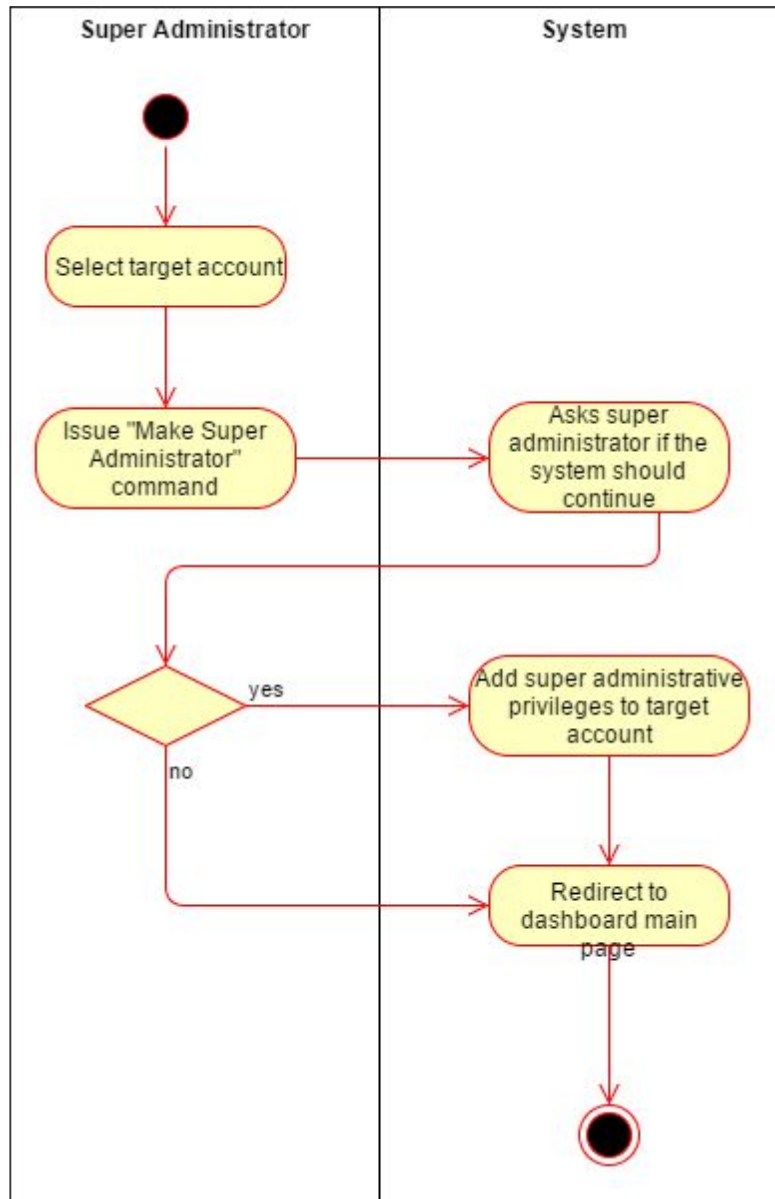
UC6: Add administrative privileges



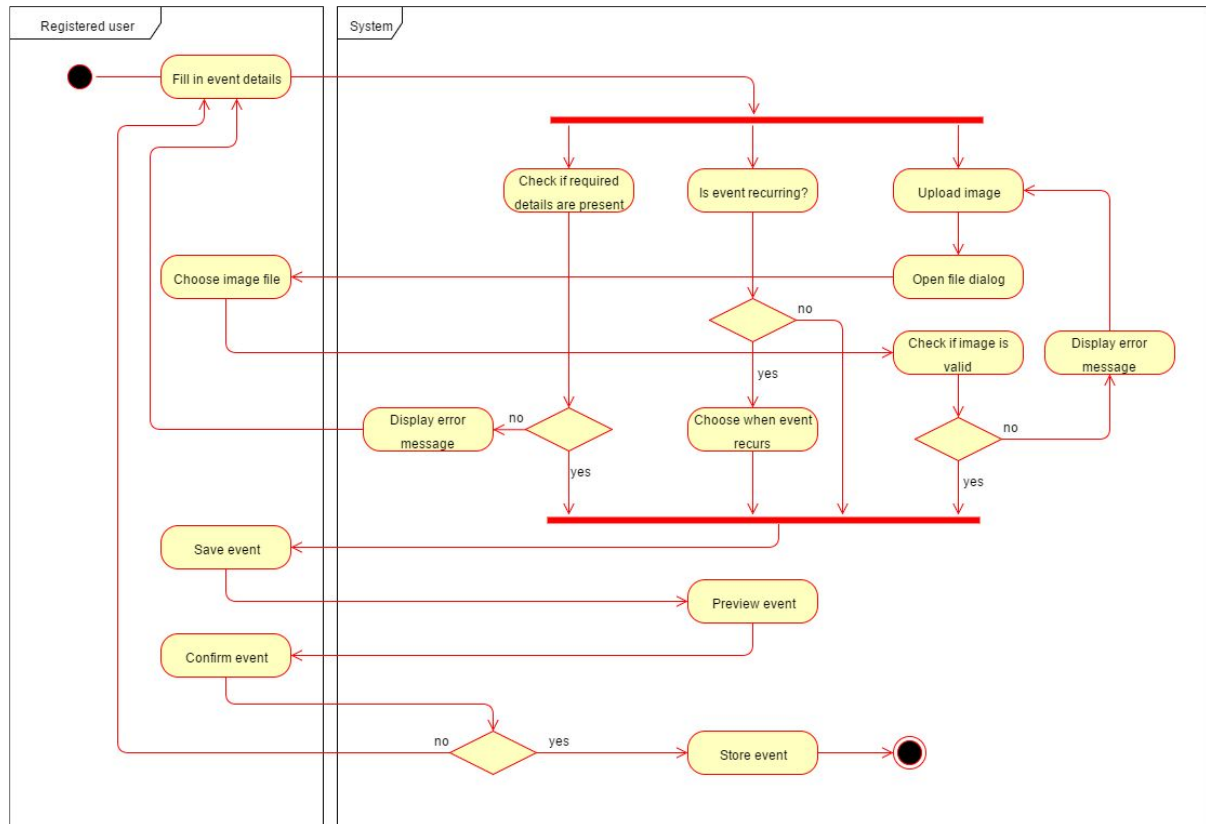
UC7: Remove administrative privileges



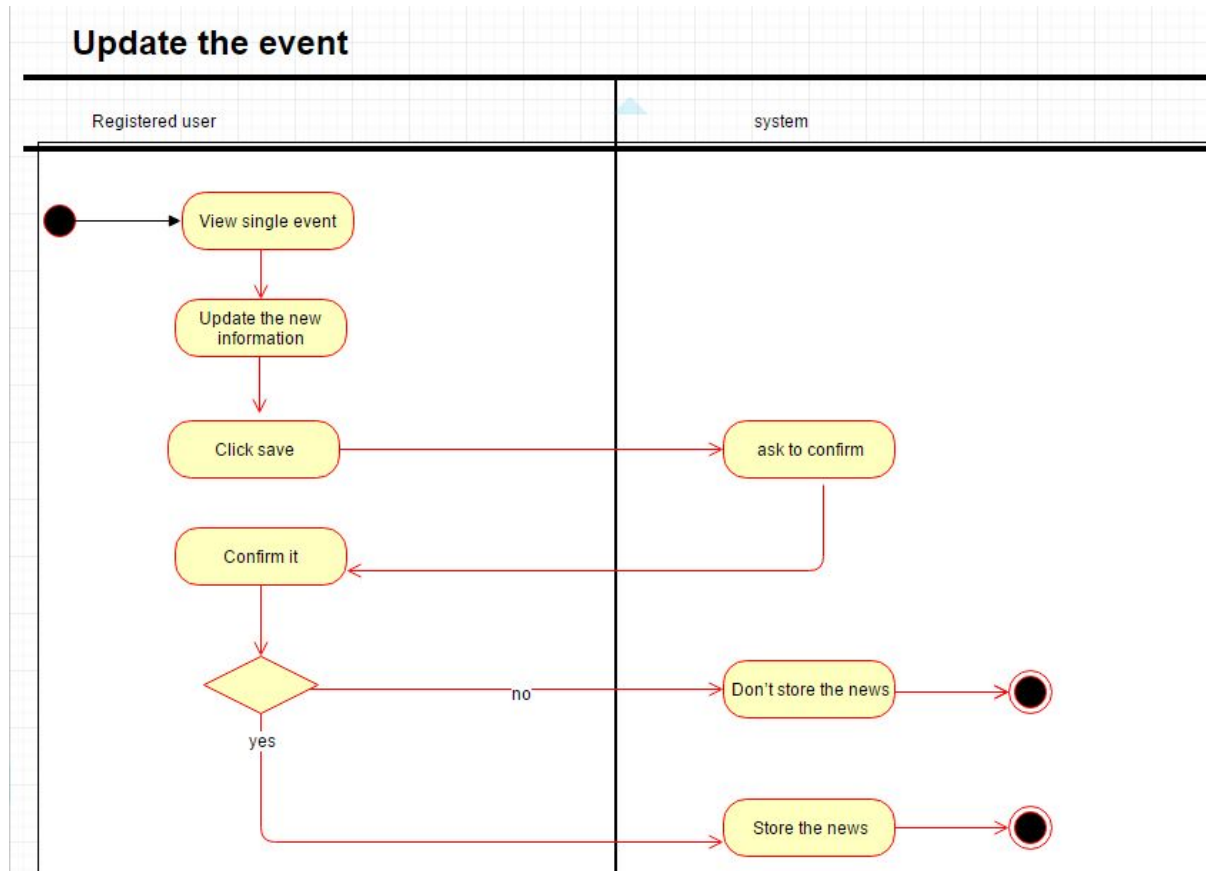
UC8: Add super administrative privileges



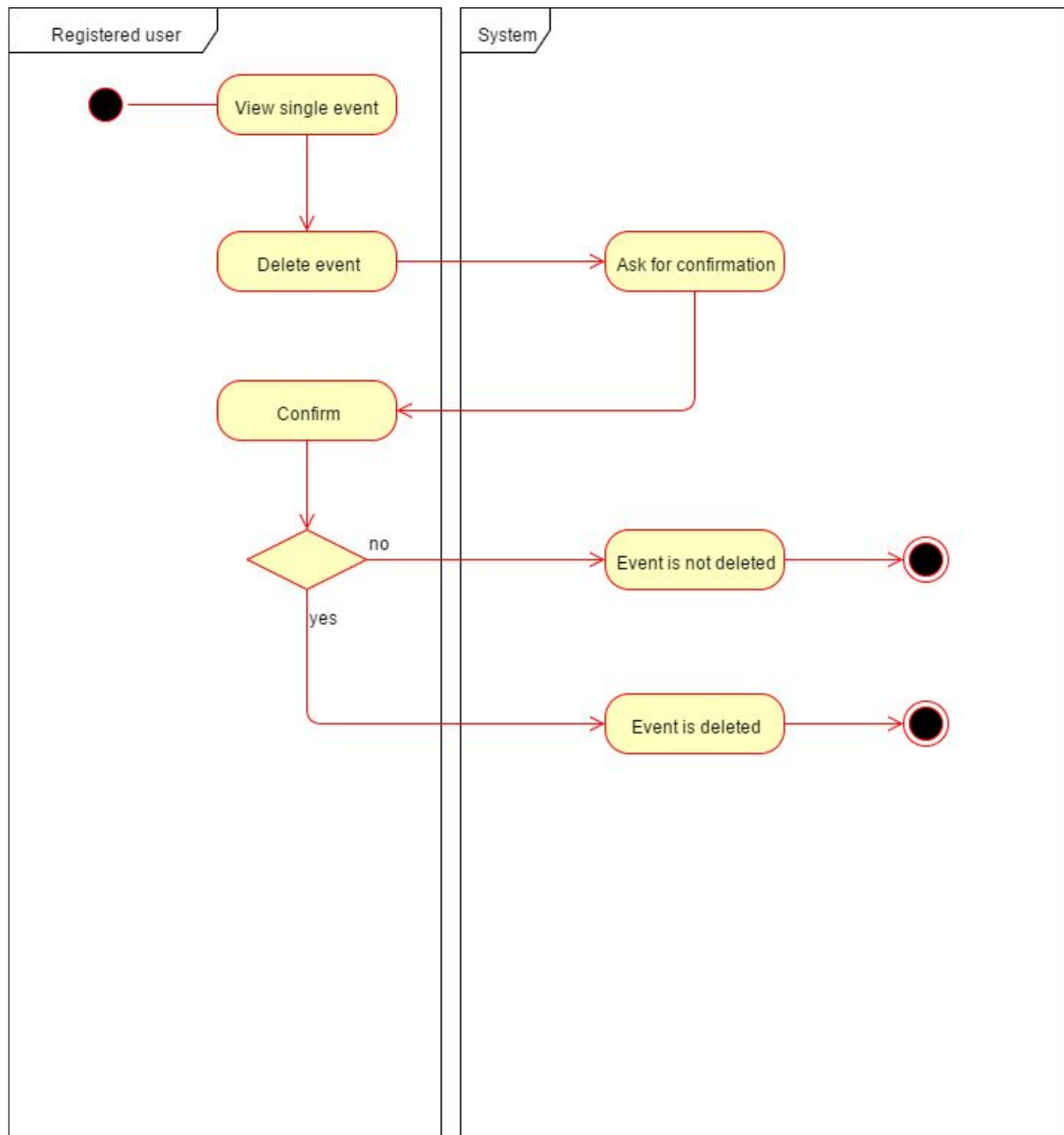
UC9: Create event



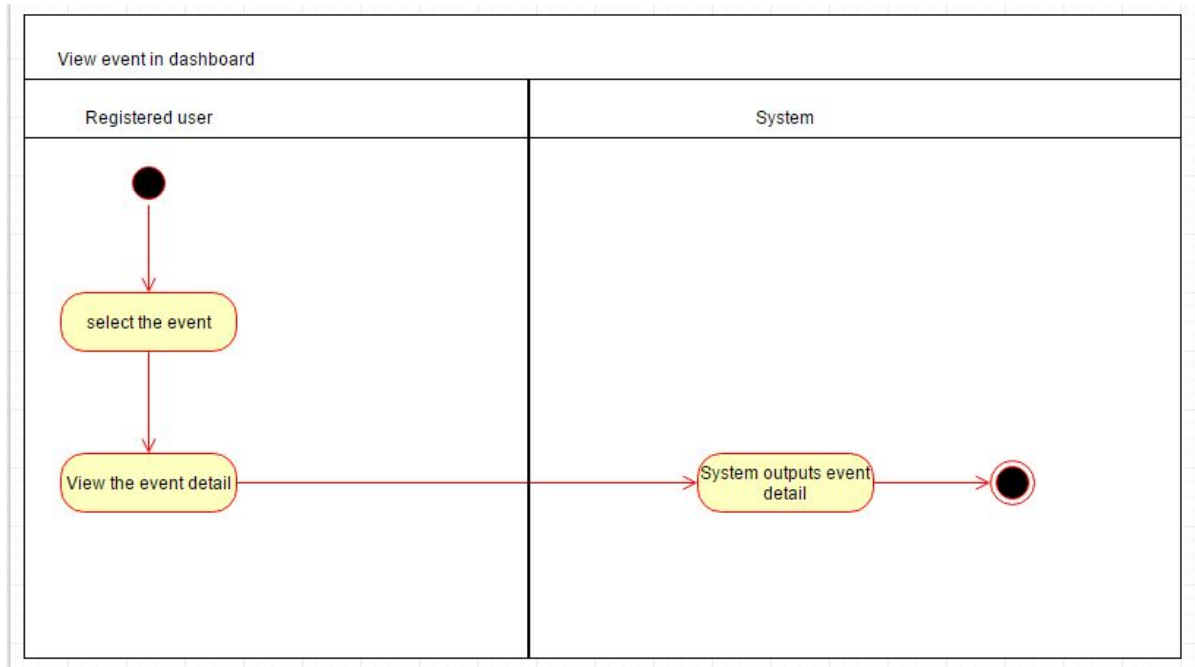
UC10: Update event



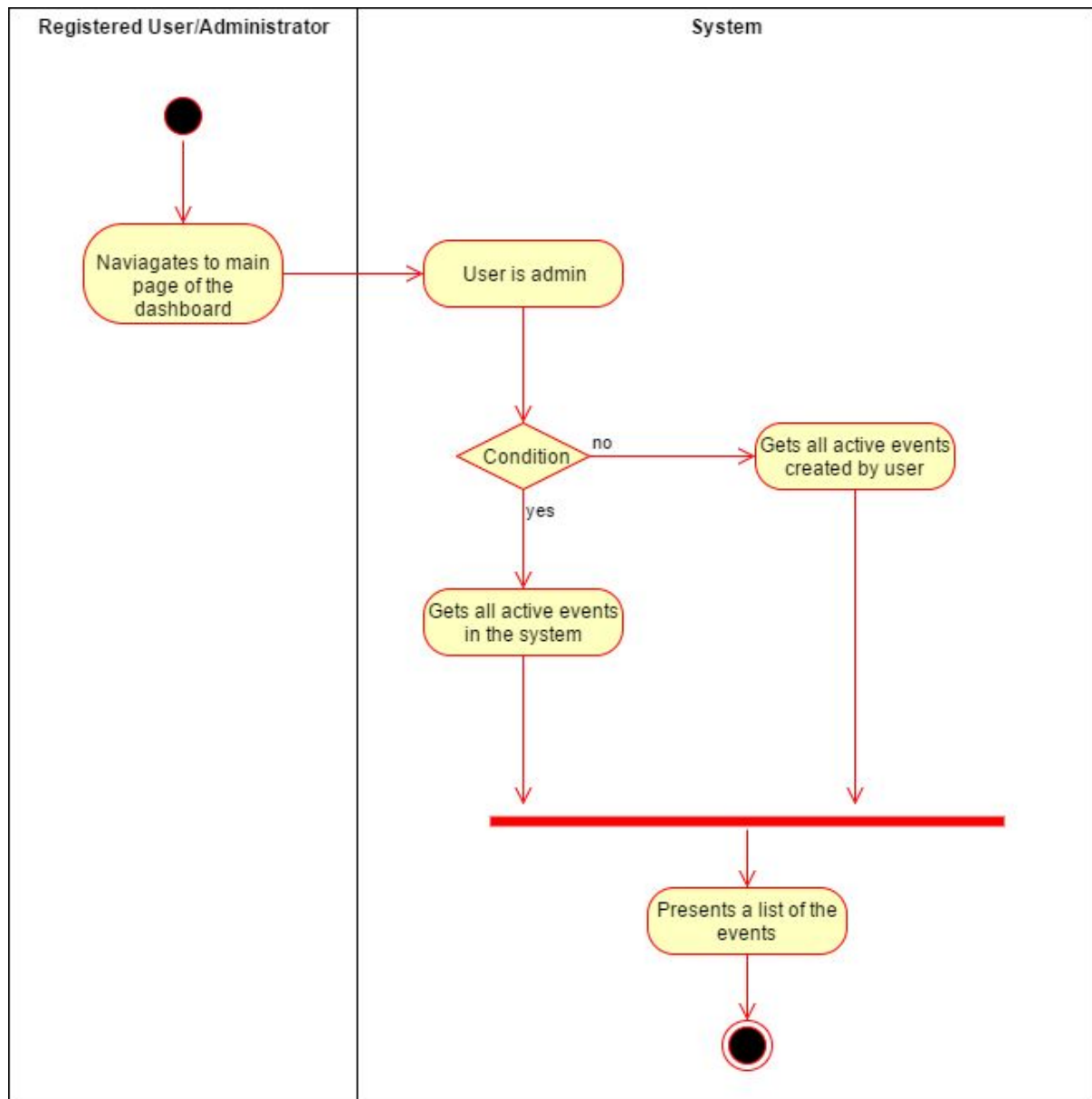
UC11: Delete event



UC12: View event in dashboard

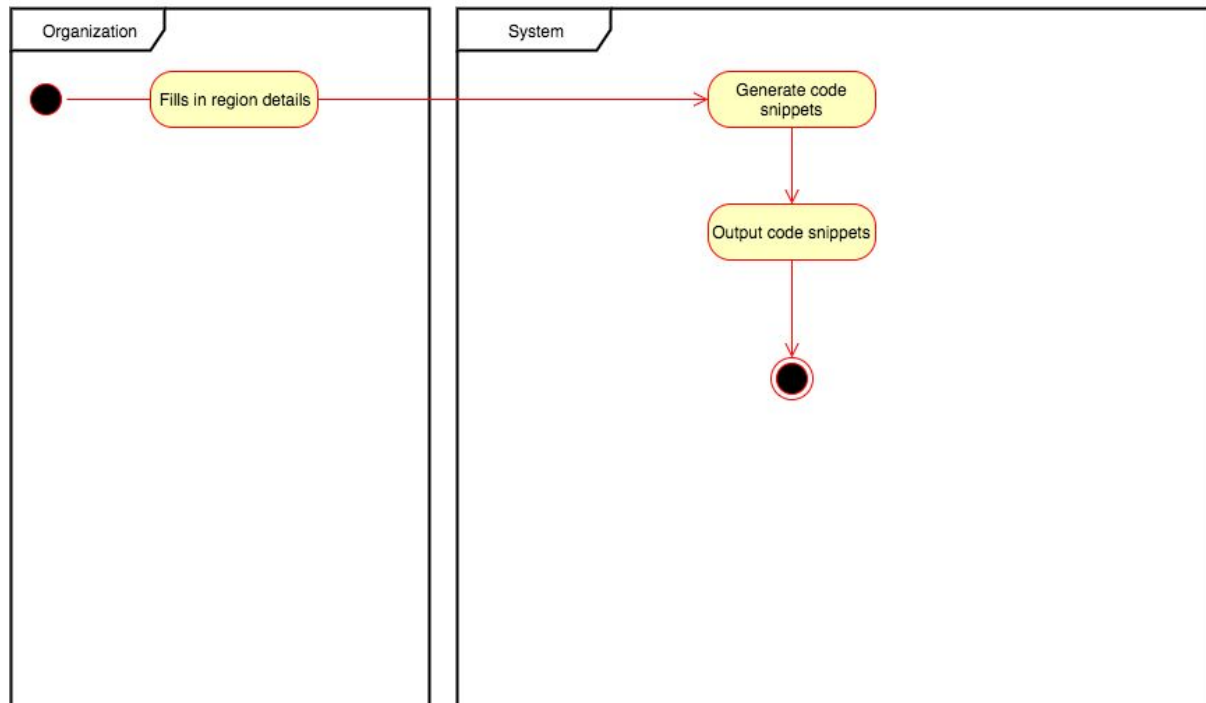


UC13: List events

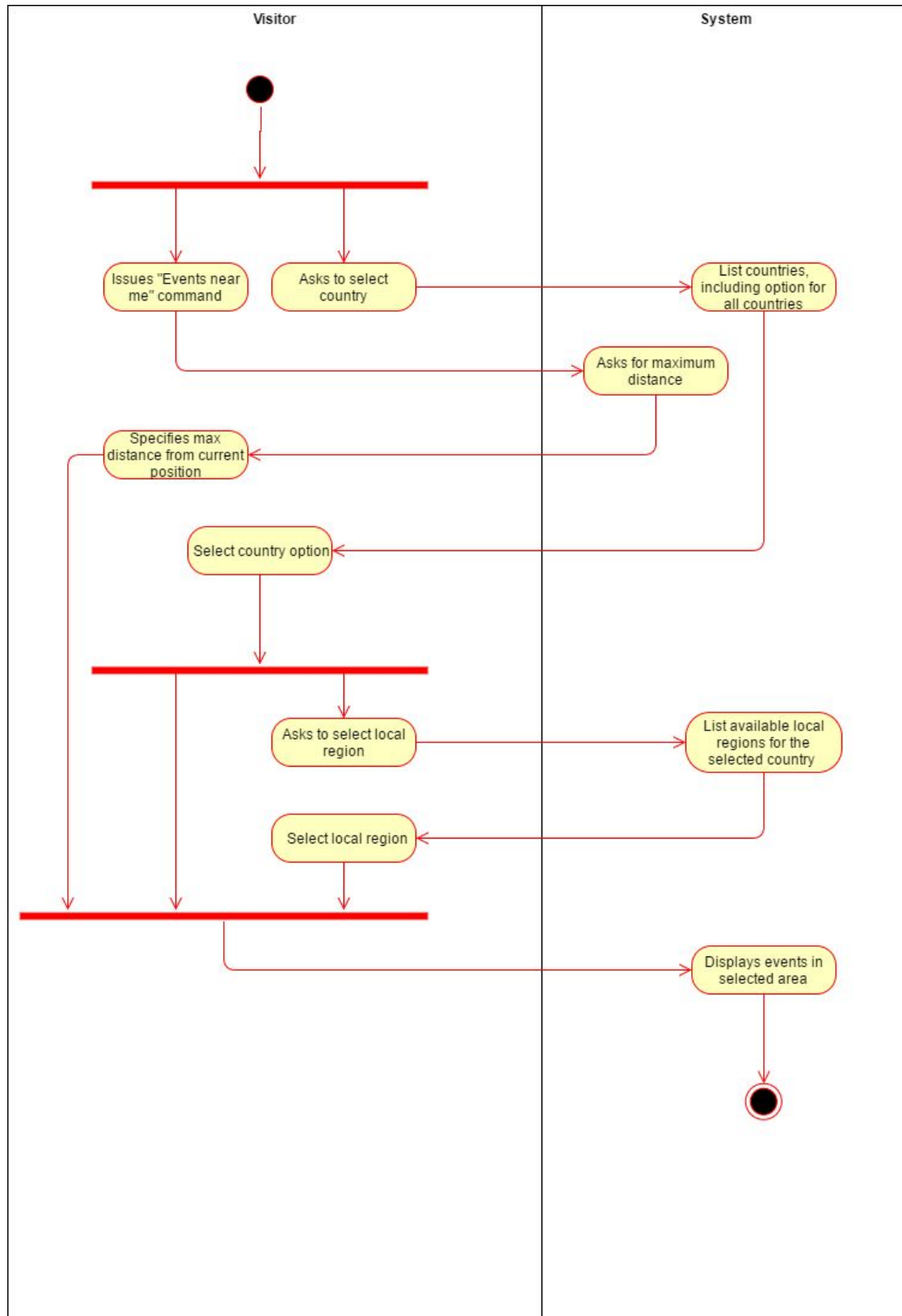


UC14: Generate widget code

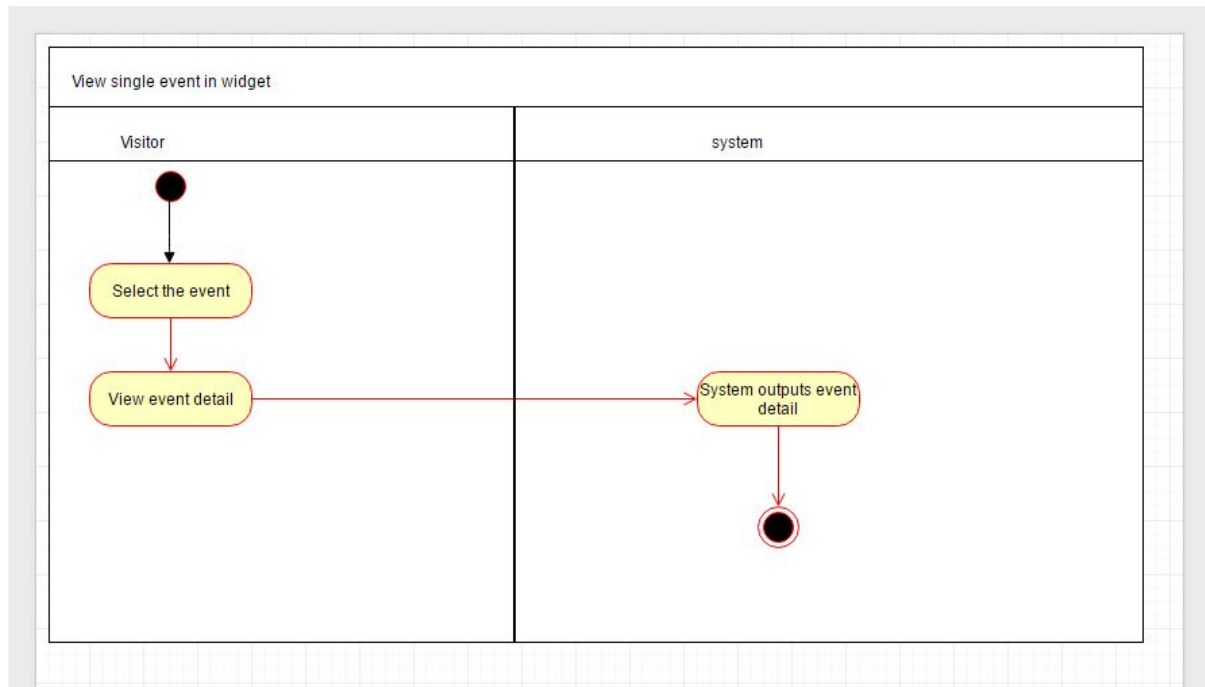
Generate Widget Code



UC15: Change region

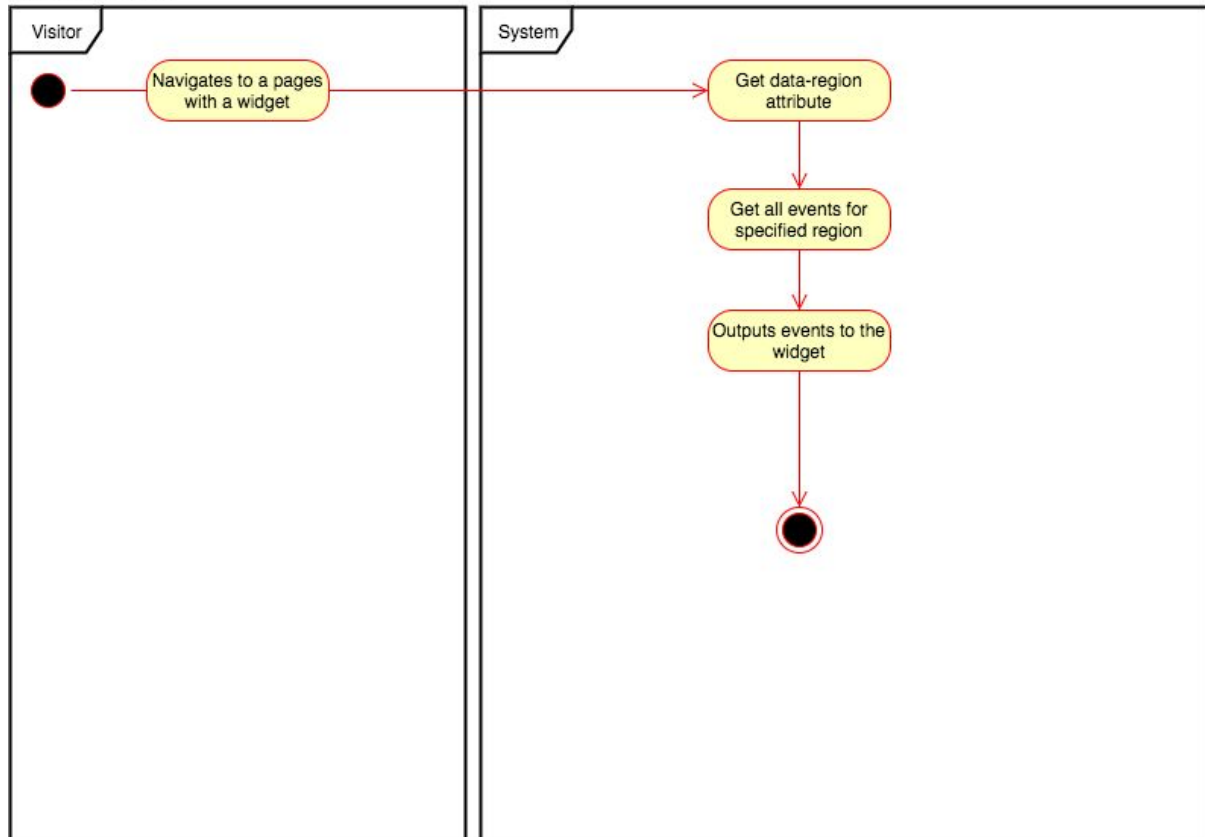


UC16: View single event in widget

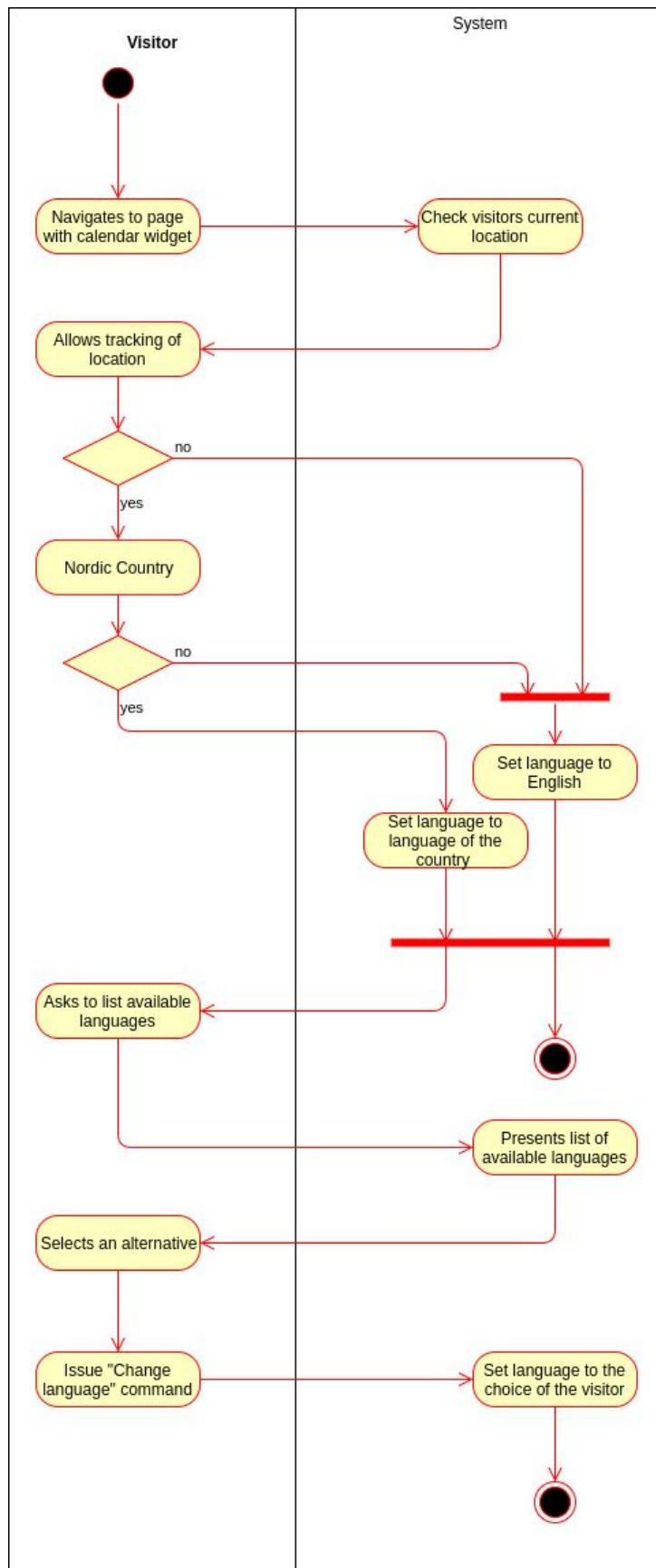


UC17: View calendar

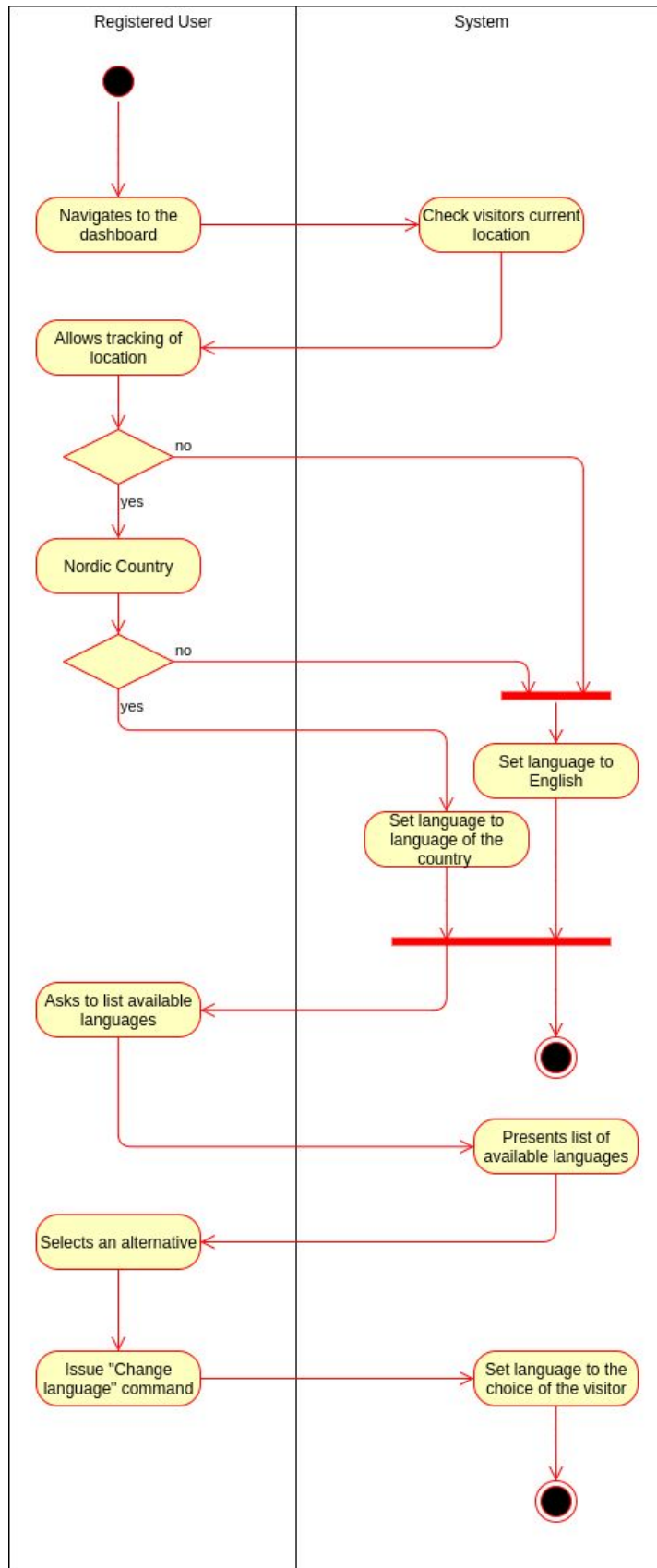
View Calendar



UC18: Widget language selection



UC19: Dashboard language selection



UC20: Widget switch to Maps View

