Milestone 1

IS specification

Group number:	42

Team member 1	
Name:	Péter Ferenc Gyarmati
Student ID:	11913446
E-mail address:	a11913446@unet.univie.ac.at

Team member 2	
Name:	Simon Eckerstorfer
Student ID:	11911424
E-mail address:	simon.eckerstorfer@unet.univie.ac.at

Conceptual Modeling **2**
- Project Outline 2
- Entity-Relationship Model 3

Use-Case Design **4**
- Use-Case 1: User Sign-up 4
- Use-Case 2: User Sign-in 6
- Use-Case 3: Create a topic (main) 8
- Use-Case 4: Add a tag to a topic 10
- Use-Case 5: Add a new snippet to a topic (main) 12

Data Analytics **14**
- Report 1: Find users active in specific languages 14
  - Report Elements 14
- Report 2: Find the most dominant languages by tags 14
  - Report Elements 14

Conceptual Modeling

Project Outline

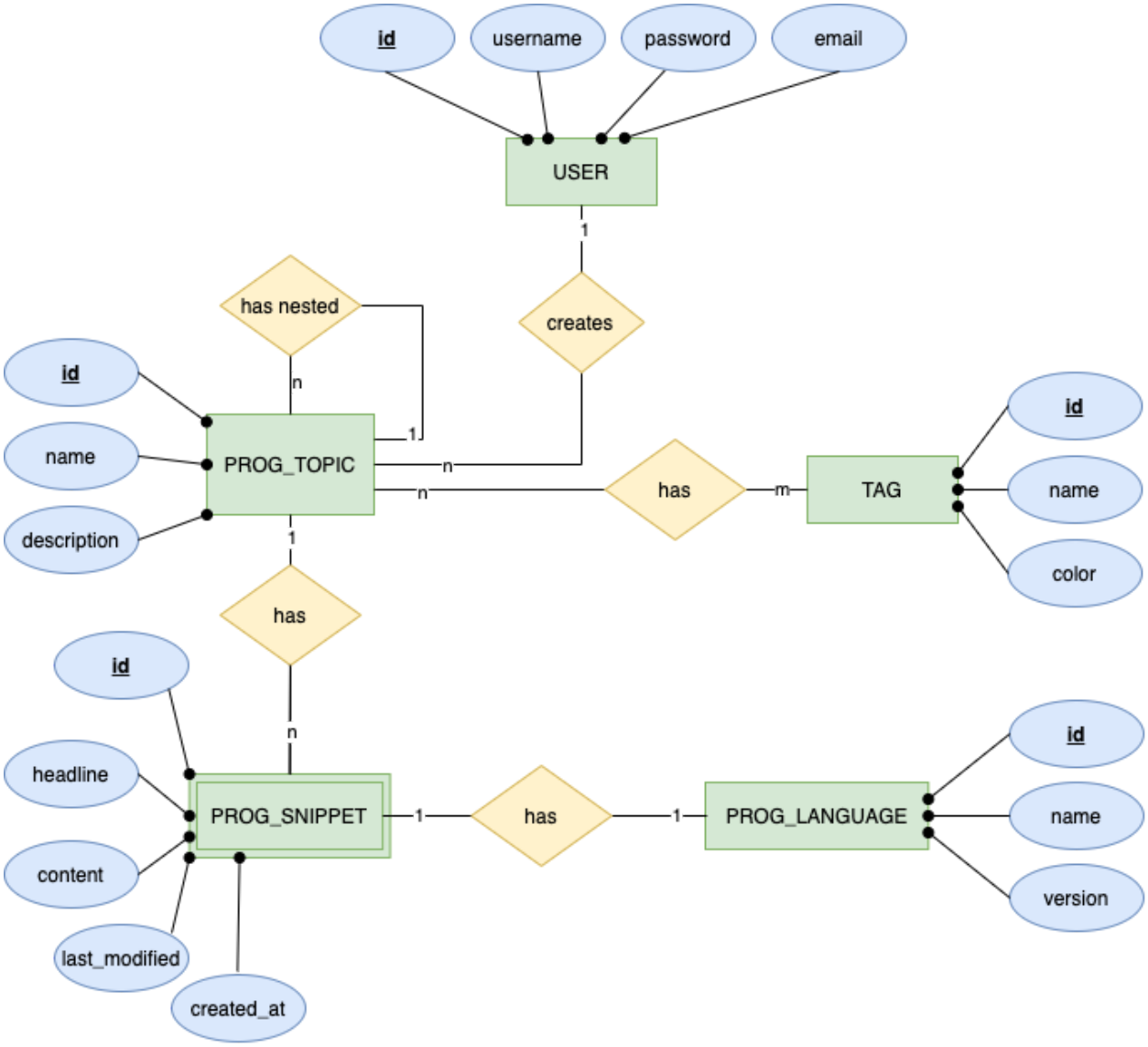
The project's main purpose is to implement a system that allows its users to manage reusable code snippets of different programming languages. They are organized into topics, which can be enriched by tags to facilitate search and filtering.

A user has a username, a password and an email address. Users have the possibility to create new programming topics as the main unit of organization for the code snippets.

A programming topic has a name and a description. Topics can have nested topics (subtopics) of an arbitrary depth level. Tags can be assigned to each topic in order to facilitate lookup for the user. A tag has a name and a colour in hexadecimal representation to enhance the look & feel of the app. Subtopics inherit the tags of their parent topics automatically.

A topic can include multiple code snippets. A code snippet has a headline, the actual code content as well as the timestamp of creation and last modification. A snippet has a single programming language, which in turn has a name (e.g Java) and optionally a version (e.g 11).

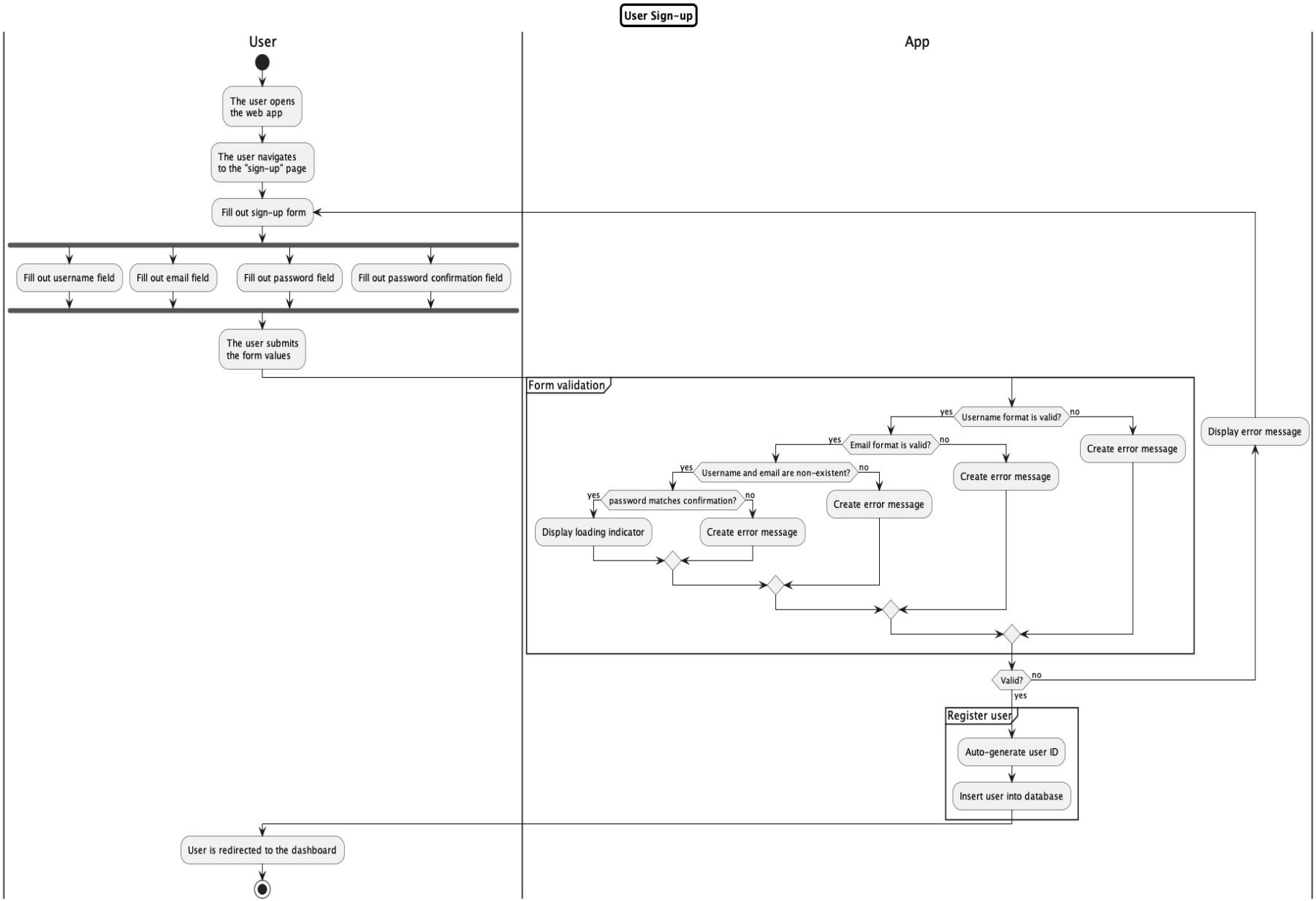
Entity-Relationship Model



Use-Case Design

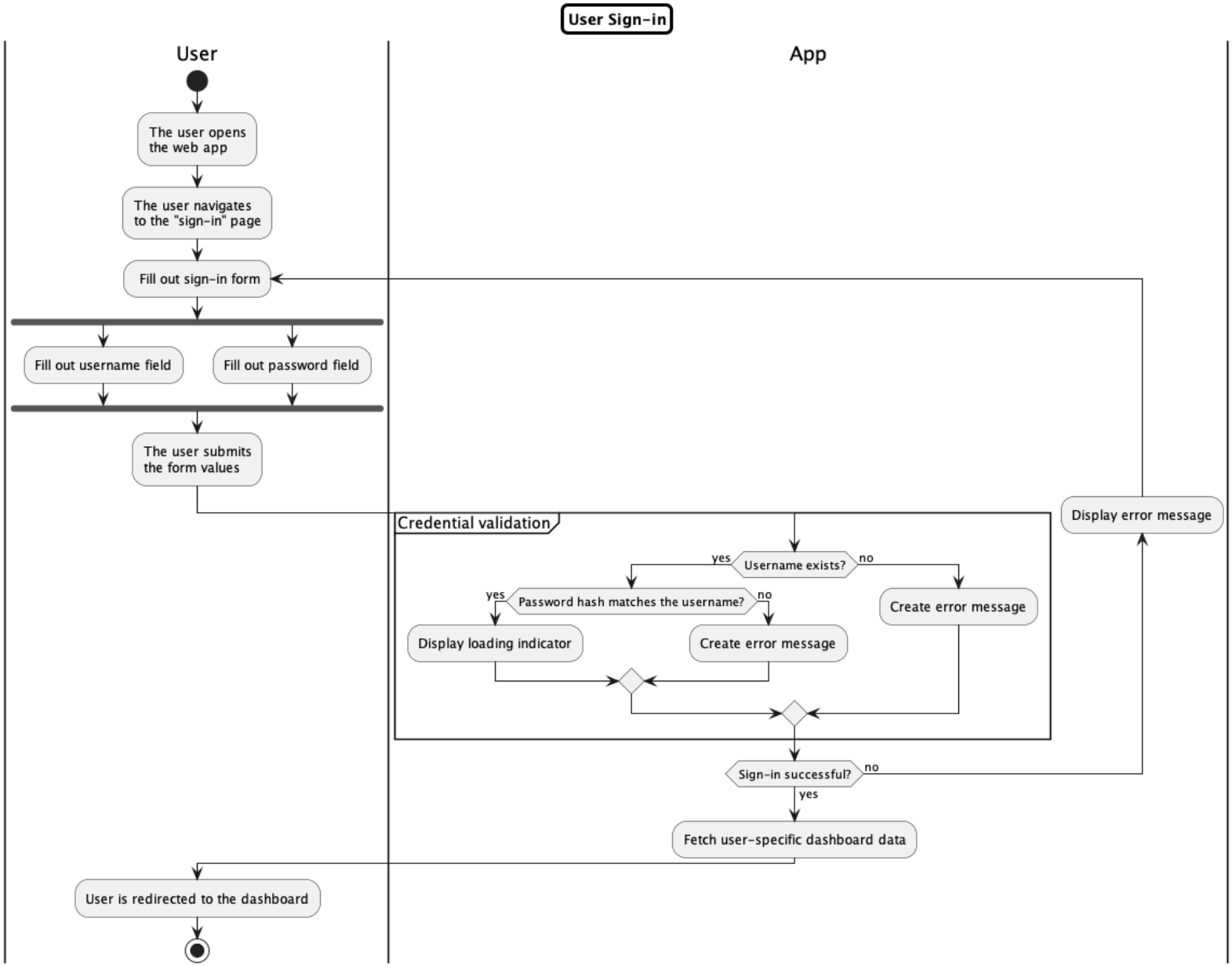
Use-Case 1: User Sign-up

Created By	Péter Ferenc Gyarmati
Objective	Add a new user to the system.
Description	A site visitor signs up to the snippet manager application.
Pre-condition	The site-visitor accesses the application with no pre-existing account (identified by an email address and a username).
Activity	The user visits the sign-up page and submits data to the sign-up form. In the background the form gets validated and based on the validation result error prompts appear or a success message gets displayed. Upon successful sign-up, the new user gets redirected to the app dashboard.
Post-condition	*Success* A new user account gets added to the database and the user is redirected to the app dashboard where further activities are available. *Error* In case of mal-formatted form input data or in case of an already existing record with the supplied email-username combination, an informative error message gets displayed.


User Sign-up
User
App
The user opens the web app
The user navigates to the "sign-up" page
Fill out sign-up form
Fill out username field
Fill out email field
Fill out password field
Fill out password confirmation field
The user submits the form values
Form validation
Username format is valid?
yes
no
Create error message
Email format is valid?
Create error message
Username and email are non-existent?
Create error message
password matches confirmation?
Display loading indicator
Create error message
Display error message
Valid?
Register user
Auto-generate user ID
Insert user into database
User is redirected to the dashboard

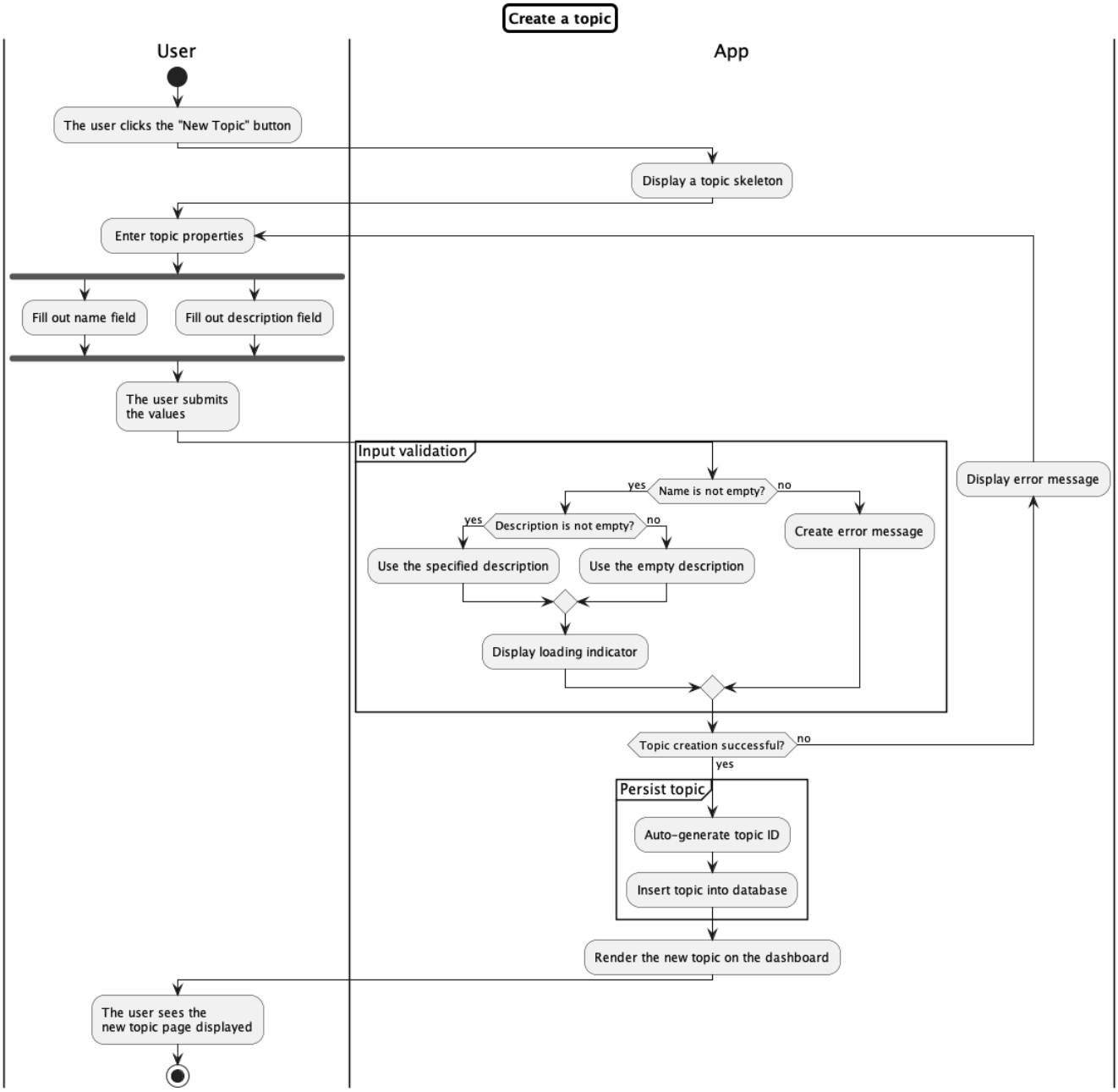
Use-Case 2: User Sign-in

Created By	Péter Ferenc Gyarmati
Objective	Sign in to the application to access its auth-guarded features
Description	A user signs in using their username and password combination
Pre-condition	The user has already created an account and knows its credentials
Activity	The user visits the sign-in page and submits their username and password. In the background, it is verified that such an input combination exists in the database. If the verification succeeds, the user is redirected to the app dashboard.
Post-condition	*Success* The user signs in is successful and the user gets redirected to the app dashboard. *Error* If the entered username does not exist in the database, the user will be prompted to create an account first. If the entered username exists, but the password is incorrect, then an error message is displayed.


User Sign-in
User
App
The user opens the web app
The user navigates to the "sign-in" page
Fill out sign-in form
Fill out username field
Fill out password field
The user submits the form values
Credential validation
Username exists?
yes
no
Password hash matches the username?
yes
no
Display loading indicator
Create error message
Create error message
Display error message
Sign-in successful?
no
yes
Fetch user-specific dashboard data
User is redirected to the dashboard

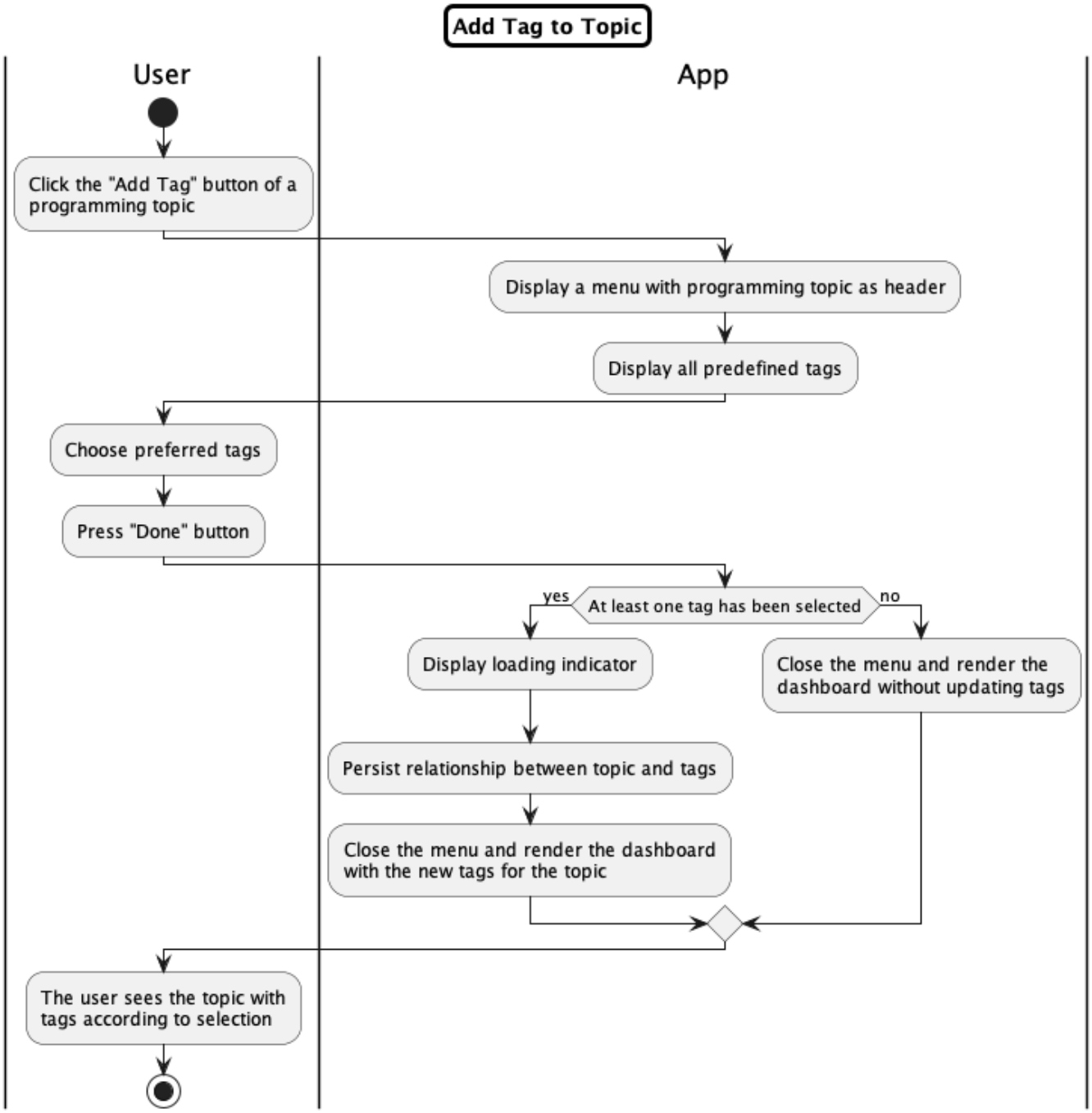
Use-Case 3: Create a topic **(main)**

Created By	Péter Ferenc Gyarmati
Objective	Create a new topic to serve as a unit of categorization for code snippets
Description	A user can create a new topic with a name and description in order to better categorize code snippets. A topic can be created as a root topic as well as a subtopic inside existing topics.
Pre-condition	The user already has an account and is currently authenticated. The user has the dashboard page of the web app opened.
Activity	*Create root topic* The user clicks on the “New Topic” button. *Create subtopic* The user right-clicks on an existing topic from the list of existing topics. And clicks the “New Topic” button. An empty skeleton of a topic appears in which the topic name and description can be specified. The topic can be saved by clicking the “Save Topic” button. The system checks whether the topic’s name is non-empty but it allows an empty description.
Post-condition	*Success* The new topic gets added to the database and the user has the possibility to either add new subtopics to the newly created topic and/or to create code snippets inside it. *Error* In case of receiving an empty topic name an error message gets displayed. The user gets the chance to complete the topic name input field.


Create a topic
User
App
The user clicks the "New Topic" button
Display a topic skeleton
Enter topic properties
Fill out name field
Fill out description field
The user submits the values
Input validation
Name is not empty?
yes
no
Description is not empty?
yes
no
Use the specified description
Use the empty description
Create error message
Display loading indicator
Display error message
Topic creation successful?
no
yes
Persist topic
Auto-generate topic ID
Insert topic into database
Render the new topic on the dashboard
The user sees the new topic page displayed

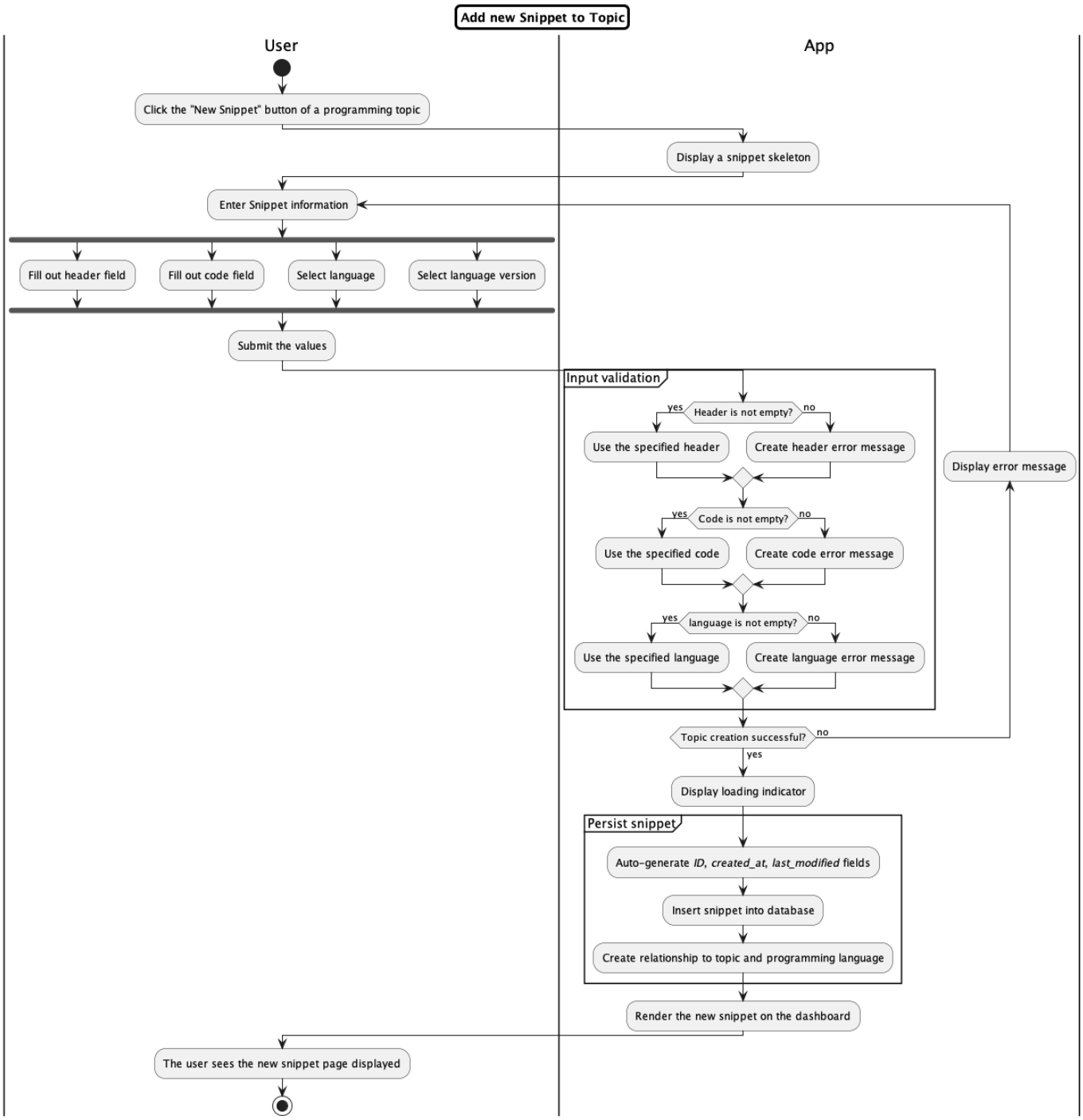
Use-Case 4: Add a tag to a topic

Created By	Simon Eckerstorfer
Objective	Categorize a programming topic by assigning a tag to it
Description	A user can select one or many predefined tags to describe and categorize the domain of a programming topic in a quick to comprehend way.
Pre-condition	The user has already created an account and knows its credentials. The user has already created at least one programming topic. The programming topic does not have associated tags yet.
Activity	The user clicks the “add tags” button of a programming topic. A menu with all possible tags appears, where the user can select zero to many tags by clicking on them. The selection is confirmed by pressing the “Done” button of the menu. The route is popped and all selected tags are shown for the programming topic.
Post-condition	*Success* Selected tags are shown for the correct programming topic and a relation between them is created in the database. *Alternative* In case zero tags are chosen, no change to the tags of the programming topic is done in the frontend or the database. This is not considered an error, but a user choice.


Add Tag to Topic
User
App
Click the "Add Tag" button of a programming topic
Display a menu with programming topic as header
Display all predefined tags
Choose preferred tags
Press "Done" button
At least one tag has been selected
yes
no
Display loading indicator
Close the menu and render the dashboard without updating tags
Persist relationship between topic and tags
Close the menu and render the dashboard with the new tags for the topic
The user sees the topic with tags according to selection

Use-Case 5: Add a new snippet to a topic **(main)**

Created by	Simon Eckerstorfer
Objective	Create a new reusable code snippet for a specific programming topic
Description	A user can create a new code snippet with a headline and content. The associated programming language has to be selected from a list of possibilities. Additionally, the version of the programming language can optionally be added by the user in case it is essential for the snippet.
Pre-condition	The user has already created an account and knows its credentials. The user has already created at least one programming topic.
Activity	The user clicks the “add snippet” button of a programming topic. A menu appears where the user has to enter a headline of the snippet and the actual code. A programming language has to be selected from a dropdown field. The user can also optionally add the version of the used language in case it affects the usage of the snippet. By pressing the “done” button, the route is popped and the snippet is visible under the selected programming topic.
Post-condition	*Success* The dashboard is shown with the new snippet and all its information. *Error* If required fields are empty, an error message gets displayed and the user gets the chance to fill out said fields.


Add new Snippet to Topic
User
App
Click the "New Snippet" button of a programming topic
Display a snippet skeleton
Enter Snippet information
Fill out header field
Fill out code field
Select language
Select language version
Submit the values
Input validation
Header is not empty?
yes
no
Use the specified header
Create header error message
Code is not empty?
Use the specified code
Create code error message
language is not empty?
Use the specified language
Create language error message
Display error message
Topic creation successful?
Display loading indicator
Persist snippet
Auto-generate ID, created_at, last_modified fields
Insert snippet into database
Create relationship to topic and programming language
Render the new snippet on the dashboard
The user sees the new snippet page displayed

Data Analytics

Report 1: Find users active in specific languages

This report enumerates the email addresses of all users who have created at least 3 code snippets in the last month written in a specific programming language.
Example: The language TypeScript, for which the email addresses of users with at least 3 new snippets in the last month get listed.

Report Elements

Created By	Péter Ferenc Gyarmati

Involved Entities	`USER`
	`PROG_SNIPPET`
	`PROG_LANGUAGE`

Involved Columns	`USER.email`
	`PROG_SNIPPET.created_at`
	`PROG_LANGUAGE.name`

Sorted by	`PROG_SNIPPET.created_at`
Filtered by	`PROG_LANGUAGE.name`

Report 2: Find the most dominant languages by tags

This report enumerates the programming languages in which the most lines of code snippets have been written under a specific tag.
Example: The tag “data science”, under which languages like Python or Julia get ranked.

Report Elements

Created By	Simon Eckerstorfer

Involved Entities	`TAG`
	`PROG_LANGUAGE`
	`PROG_SNIPPET`

Involved Columns	`TAG.name`
	`PROG_LANGUAGE.name`
	`PROG_SNIPPET.content`

Sorted by	`len(PROG_SNIPPET.content) //derived`
Filtered by	`TAG.name`