VILNIUS UNIVERSITY
FACULTY OF MATHEMATICS AND INFORMATICS
INSTITUTE OF COMPUTER SCIENCE
INFORMATION TECHNOLOGIES STUDY PROGRAM

Problem-Based Project

Content Sharing System for Teachers

Done by:
Andrej Jegorov
Vidmantas Moisevičius
Aras Urbonas

Supervisor:
dr. Linas Bukauskas

Vilnius
2022

Contents

Abstract

The project which is going to be described in this report is a teaching material sharing website, "STorm", made specifically for teachers. It is based on a modern full-stack web framework and is accessible via the most common browsers in use. There is also a companion Android app available for certain features. The system was made with a design-first approach, with strong emphasis on competitive analysis and preliminary design. Human-computer interaction principles were also taken into account in order to increase the usability and accessibility of the system.

Santrauka

Tinklapis STorm

Projekto tikslas – sukurti mokymo medžiagos dalinimosi sveitainę mokytojams. Tai svarbu, nes šiuo metu Lietuvoje nėra visuotinai paplitusios, mokytojams pritaikyto, medijos dalinimosi tinkapio. Tai buvo išsiaiškinta analizės metu, vykdant mokytojų apklausas. Taigi, "STorm" sistemą siekia išspresti šią problemą, užpildyti švietimo sistemoje esančią nišą.

Pagrindinės sistemos funkcijos yra sutelktos į tai, kad mokytojui būtų suteikta galimybė įkelti ir dalintis tekstu, skaidrėmis, nuotraukomis ar vaizdo, garso įrašais, kitais failais. Šios medžiagos įkeliamos publikacijos forma tam, kad sistemos naudotojai galėtų ieškoti ir filtruoti medžiagą pagal mokymo dalyką, mokinių klasę bei kalbą. Taip pat, tai įgalina paiešką pagal raktažodžius. Viena iš sistemos dalių - Android programėlę leidžia greitai ir patogiai įkelti nuotraukas iš mobiliojo įrenginio neperkeliant jų į asmeninį kompiuterį.

Tinklapio integracija su kitais socialiniais tinklais taip pat vienas iš svarbių sukurtos sistemos aspektų. Mokytojas gali prisegti savo socialinių tinklų nuorodas savo profilyje ir tapti prieinamas kitiems vartotojams, norintiems su juo susisiekti ar sužinoti apie jį daugiau. Taip pat, mokytojas gali prisegti nuorodą į šią svetainę kitose socialinėse bendravimo platformose. Tai suteikia galimybę sistemai tapti ne tik mokymo medžiagos dalinimosi vieta, bet ir jaunų mokytojų kompetencijos įrodymų pateikimo įrankiu.

Introduction

The purpose of this system arises from the needs of teachers. Currently there is no content sharing system commonly in use by the teachers in Lithuania and most probably other countries as well. A need for young educators to build up their portfolio also exists, which is not particularly easy, especially for the people who are not technologically inclined. The "STorm" content sharing system seeks to solve these challenges by providing an intuitive, easy to use website and mobile app in order to achieve the required functionality.

This includes: uploading of text and media via the website, uploading one or more images via the STorm Android app, searching and filtering content, creating your educator profile, creating teacher groups and more.

Social network integration is also a significant part of the "STorm" system's purpose, as a teacher can add links to their social media into their profile, thus enabling themselves to be found and contacted, if they so desire. Additionally, teacher profile and material viewing is enabled for anonymous users, thus it is possible for the educator to add a send a link to their "STorm" profile to anyone interested in their content or portfolio.

All of this is accomplished via the use of a web framework that is used to process and save media. It is connected to a database, which stores the relevant information in a well structured, easily retrievable way. The web framework is not only used for providing the website functionality, but it also runs an application programming interface (API) which is used by the "STorm" Android application.

This report seeks to explain the main components, major functionality and requirements of the created system. One of the sections is preliminary design, which served as the basis for the system's later implementation. It is explained via wireframes and showcases the expected behavior of the system from the user's perspective. Further on, system architecture section utilizes diagrams and descriptions to show the implementation and specifications of the major system components. The later requirements sections explain the goals of "STorm" via the use of user stories. Testing section shows examples of actual functionality and expected results. The setup part serves as the tutorial for anyone interested in hosting the "STorm" system, for more in-depth analysis and testing. Competitive analysis reviews related work and provides justification for the ideas and decisions made during the development process. Finally, the human computer interaction section explains some of the principles which were used when designing and implementing the system.

1 Preliminary Design

The following section describes individual webpages with emphasis to the user perspective with the intent of revealing concrete information regarding the functionality provided. Some of the descriptions of the pages are accompanied with image extracts from wireframes. The design shown in such figures is by no means a direct recreation of the User Interface of the application but rather a simple representation of the interactivity that is supposed to be provided by the system to the user in those specific occasions.

1.1 Homepage

Homepage follows the pattern typical to homepages of most contemporary websites. This page features the base description of the features provided to the user as well as information regarding the top three contributors and the currently top three most popular materials uploaded. This is done in an attempt to provide more insight into the functionality as well as to increase the curiosity of the potential user to try the application.

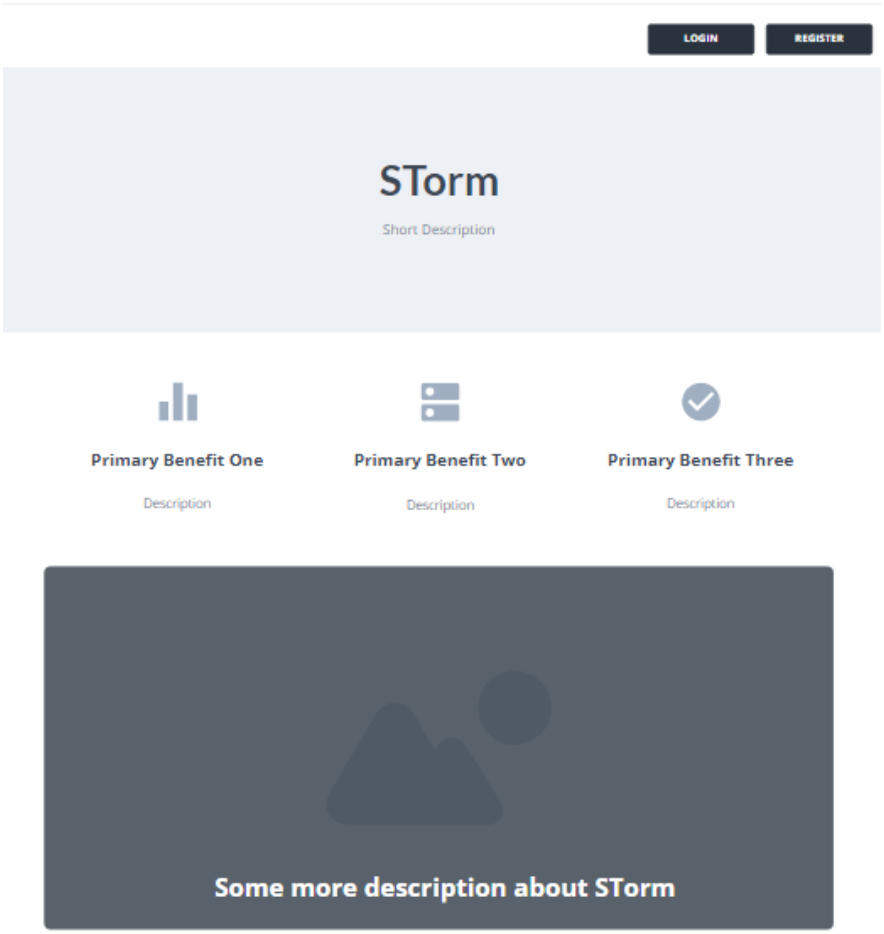

Figure 1. Homepage

1.2 Registration/Login

Both, the login and the registration, screen are designed in a similar fashion to each other. Registration requires the user to input the following fields: email address, password and name. Password confirmation for registration is also required and checks for password length, strength, dissimilarity to username and other such measures are provided by the system. It is done in order to ensure that the user's account has better security and protection. The users account also has additional customization options after registration that will be discussed in (*section 1.3*). Login authentication requires the user to input their registered email and password. After successful authentication the user is able to access further functionality provided by the website.

1.3 User Profile

Following the login the user is directed to their profile. Each registered user has a profile page associated with them. The profile page contains relevant information in regard to the user. The information regarding a user is provided in order: teacher's title (subject which the teacher is associated with), name, description (customizable by the user, (*see section 1.5*)), achievement numbers (follower, download, upload and total like counts) and buttons that allow the user to be followed or invited to groups (the functionality of these is described in (*section 1.14*) and (*section 1.4*)). User other social media links are also optionally displayed on the profile. Additionally, the user's profile also holds their top three best uploads and showcases recommendations that other users have written on user's profile. Important to note that if the user is on their own profile page a button to edit the profile Is also visible and functional.

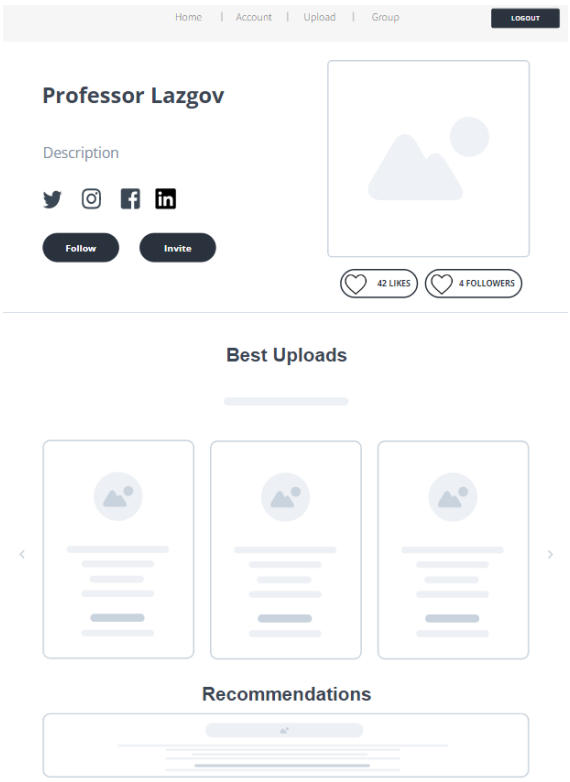

Figure 2. User Profile

1.4 Group Profile

Groups provide the ability for users of the same interests and/or organizations to band together and share content between each other. Each group has its own profile with very similar functionality compared to user profiles. It displays the member count as well as total number of group member likes. Option button or request to join the group is also present. Similarly to user profile, group profiles also feature top materials uploaded, but it is accumulated from uploads of group members specifically.

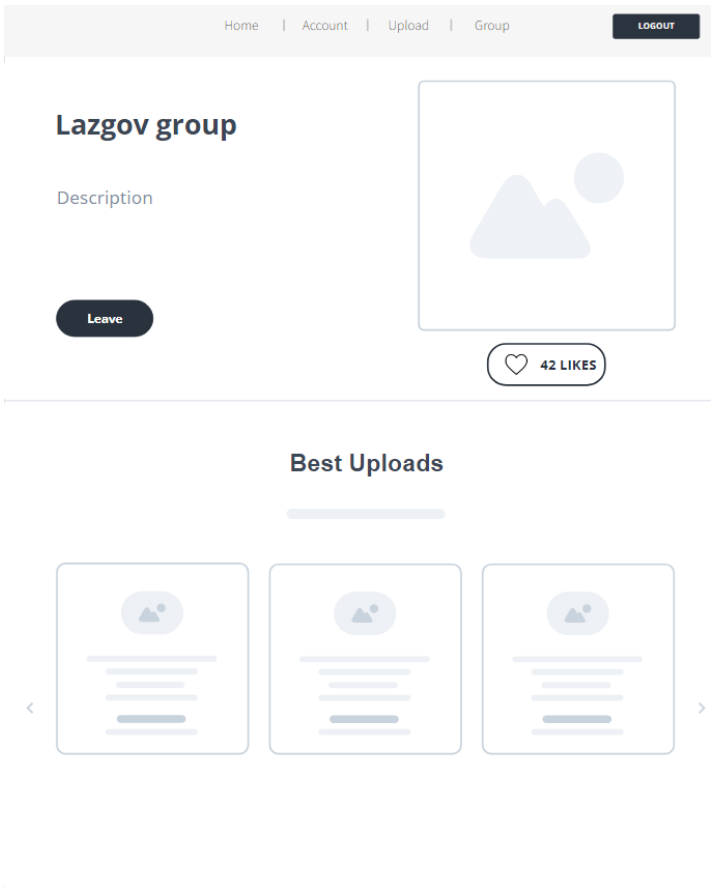

Figure 3. Group Profile

1.5 User Edit

Users can edit their respective profiles. The User profile allows customizing the description and subject. The page also features the option to add a profile picture. Ability to add social media links for the user is also available (social media links supported include “Facebook”, “Instagram”, “LinkedIn” and “Twitter”.

1.6 My Groups

Users are additionally able to view any groups they are a member of. This page provides the names of the users groups as well as the ability to leave any groups shown.

1.7 Group Creation

Groups can be created by any registered user in the group creation page. For that to pass the following information needs to be provided: group name, short group description and full group description. Also optionally a group image maybe provided that will appear in group search.

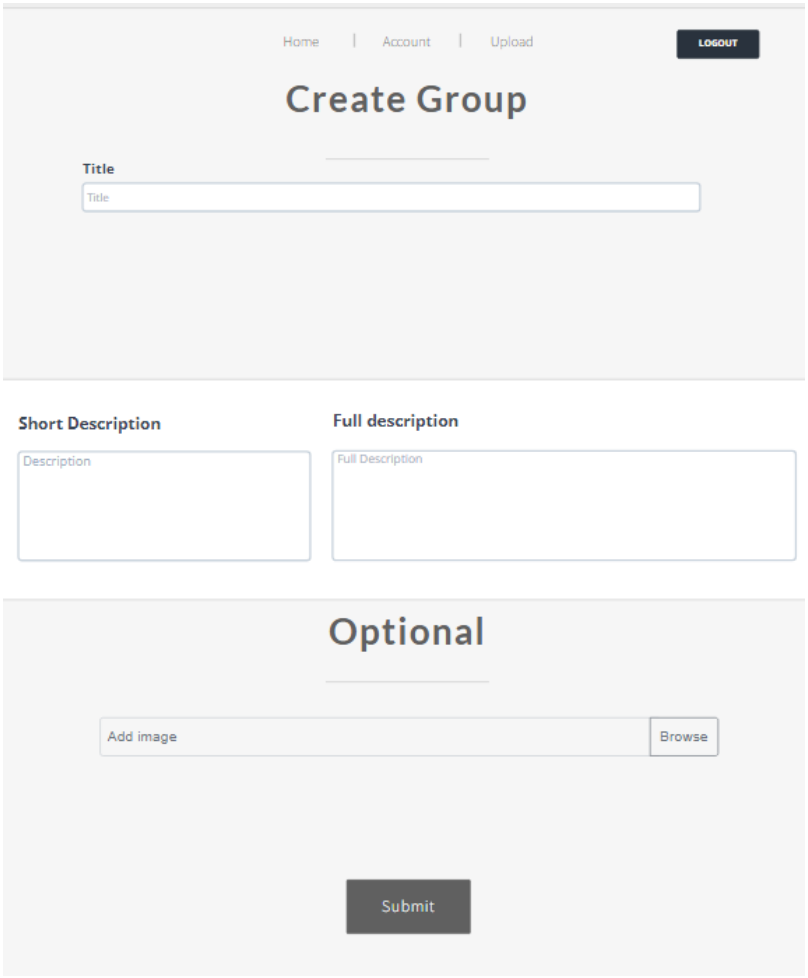

Figure 4. Group Creation

1.8 Material Search

As for the primary functionality of the system, the user is able to search for subject related materials. The search for the material is implemented with several different options for searching, filtering and sorting the results to best allow the showcase of most relevant information in conformation of the search. The search allows the user to type the name of the material that is needed, the search for this is done in regards to keywords in the title of the material. The found material may further be filtered by age (which school class the material is directed for), material subject and language. Furthermore, these results may further be sorted by ratings, upload date and alphabet. All these options are not interdependent, meaning that the user may use any combination of sorting, filtering, searching or none of these options at all. Groups also ease the difficulty of finding specific types of materials as the user could select to only display uploads that were private to the group that the user belongs to.

After finishing the search the user is able to see the results from the aforementioned action that are displayed below the search options in cards that further describe the material by providing short descriptions, tags and a picture associated with that material. The options to view the material as well as see the numbers of likes are also present. The author's name and link to the author's profile are displayed at the top of each card as well.
Important to note that the system for ratings is based on likes with no ability to dislike content. This was a coherent decision due to the reason of platforms growth, since the users profile (see (*section 1.3*)) has a total of likes accumulated, a dislike system would discourage users from uploading files. That would not be preferred by the users as a dislike system might result in a large amount of negative ratings to be displayed on their profile. A like only system mitigates such a situation and allows users to comfortably upload as many materials as they would wish to with little to no fear of getting large swathes of negative responses.

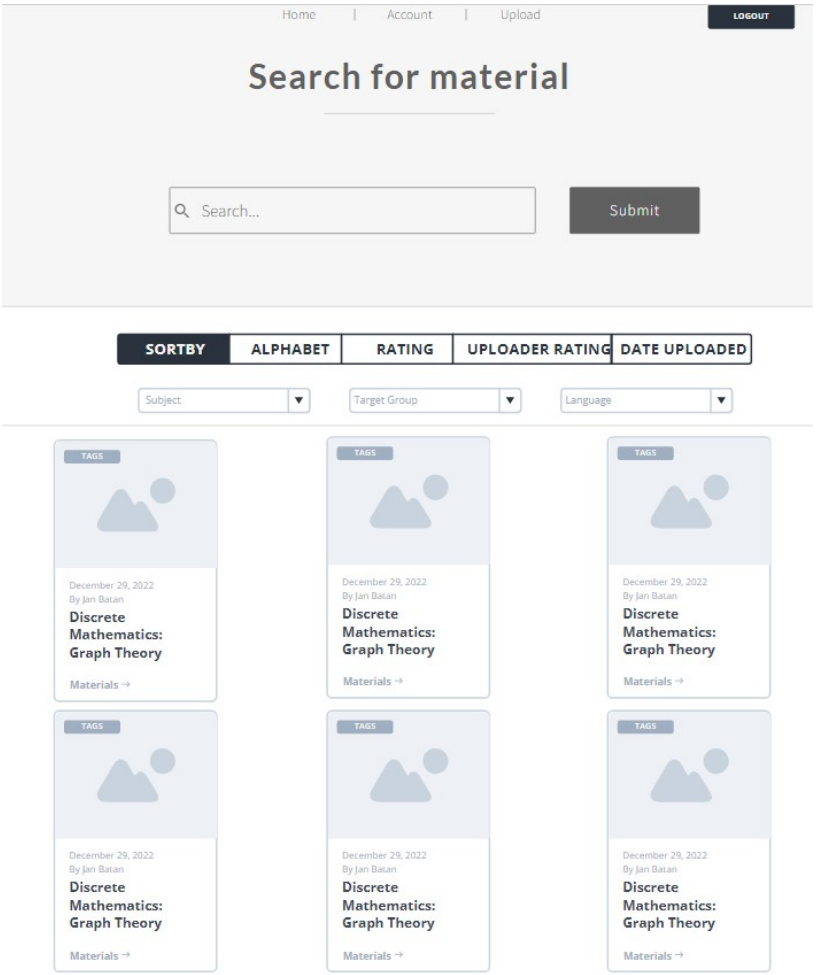

Figure 5. Material Search

1.9 Material Upload

Naturally, users of the platform may publish their own materials inside the system. The material upload page allows to do that after filling a form containing relevant information regarding the material to-be-uploaded. The information required includes: material title, subject, target grade, language, media contents (files that are uploaded as part of the material) and publication photo (optional image that reflects what material is about and is shown in material search card (*see section 1.8*). Two separate descriptions for the material upload are also essential. These include a short and a full description. Short description is to be displayed in the material search card, full description is displayed once the user selects to view the material.
Please note as of the current implementation of the system in order to upload multiple files the user has to zip them by themselves, the system does not provide any additional facilities to do that automatically at the time of writing of this section.

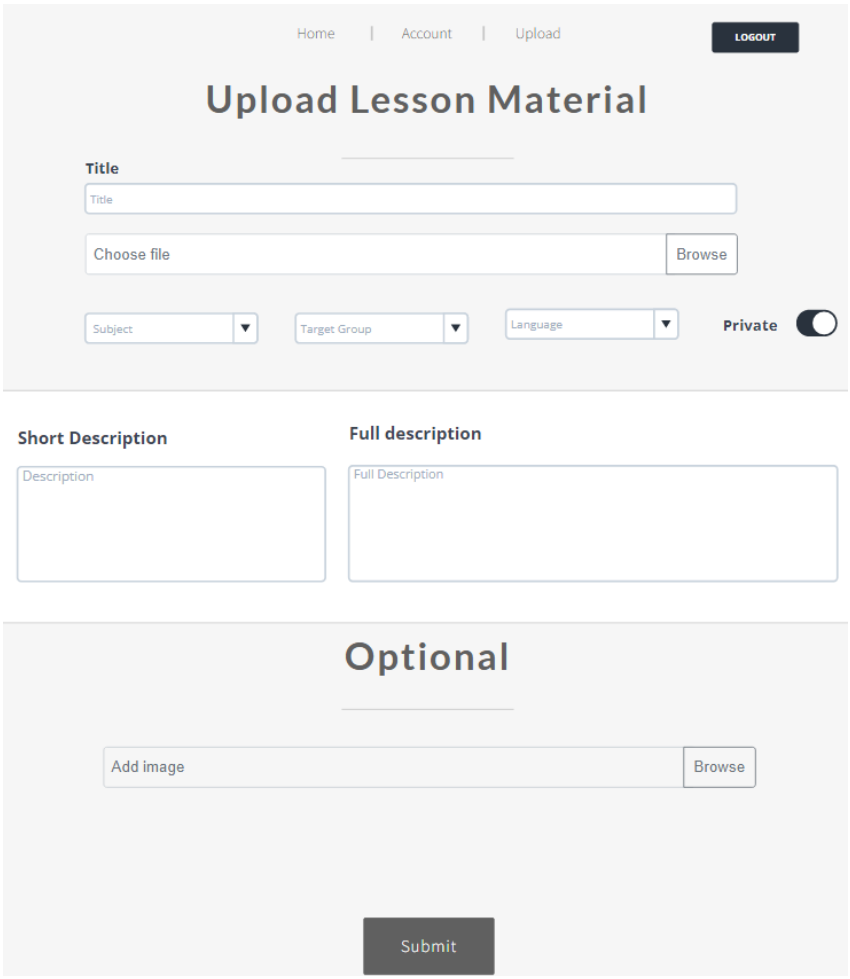

Figure 6. Material Upload

1.10 Material Info

Once the user selects to view a specific material they are redirected to a page dedicated to that specific publication. The page includes material title, relevant material information mentioned the previous sections of the document (subject, grade, language, author, number of likes), the ability to add a like to the material is also present in this page. Full description of the publication is presented to the viewer as well as the obvious option to download the files associated with the material selected. The button to download only appears if the publication contains any files (as the files could be uploaded by the user to just contain some useful links in the description without any files associated with the upload)

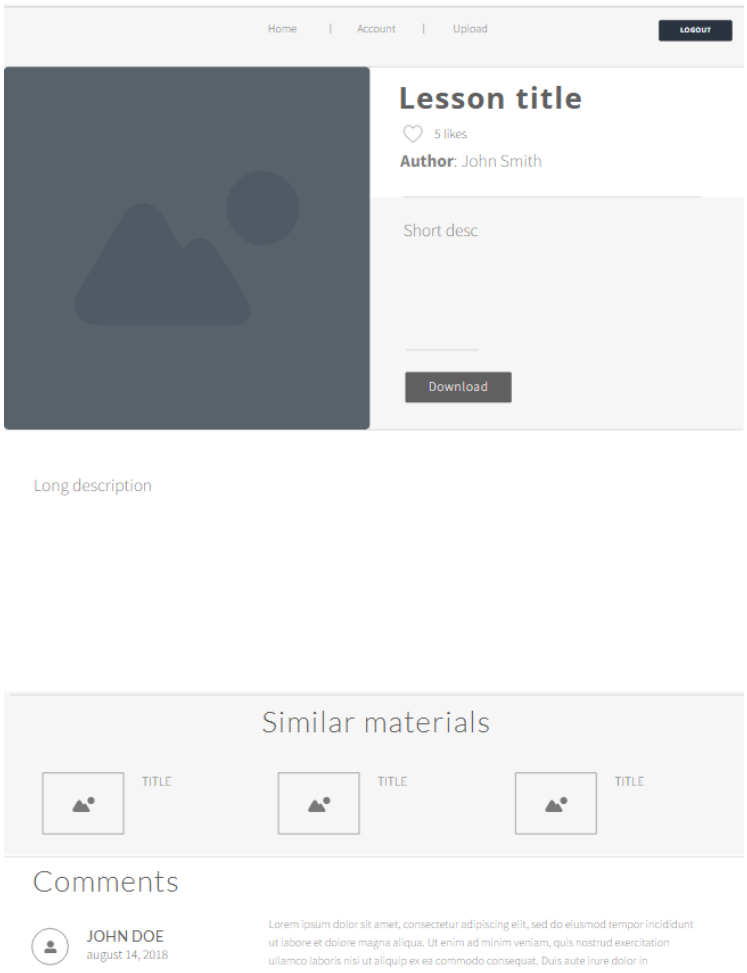

Figure 7. Material Info

1.11 User/Group Search

Additionally, two additionally pages to search for specific users and groups are also provided. Their design and functionality follows the search page described in (*section 1.8*) in a slightly more restrained form. Only the name search option is provided, the users and groups found are displayed in cards with their names and the beginning of descriptions provided the option to “view profile” redirects to user to the respective profile of user or group.

1.12 Leaderboard

Lastly, the system also provides the ability to find the best rated users easily through a leaderboard page. The leaderboard ratings are based on total number of user likes as they not only indicate that this user has provided well received materials, but a large number of likes could potentially indicate a large amount of user uploads with fewer likes, thus both activity and quality are rewarded through this method. Through the leaderboard page it is possible to view user profiles by clicking the "view profile" button near user's leaderboard entry in the table.

1.13 Page Design Similarities

Since the system in discussion is developed with the utilization of Django web framework, template inheritance (see section 2.4.1 that describes template inheritance) is used to help with maintainability and coherency of UI design and mutual functionality between different pages. Mutual design and functionality of pages primarily include the navbar and the footer which offer simple navigation between different pages on the site. Additionally, the navbar's functionality slightly differs depending on whether the user is logged in or not, with access to the majority of the pages (apart from the ones related to home and authentication) being greatly restricted if the user does not have an account logged into the system.
Mutual UI design in all pages follows a simple design pattern dominated by blue, navy and white colors. All UI elements are based off designs that are provided by the utilized Boomerang UI Kit and Bootstrap 4 framework (*section 2.3*). Public bases (accessible to not registered users) include login/registration, homepage, material search, user profile and material info. Other pages require user authentication to access. Without proper authentication, attempts of accessing pages not mentioned in the prior list redirects to the login page.

1.14 Android Application

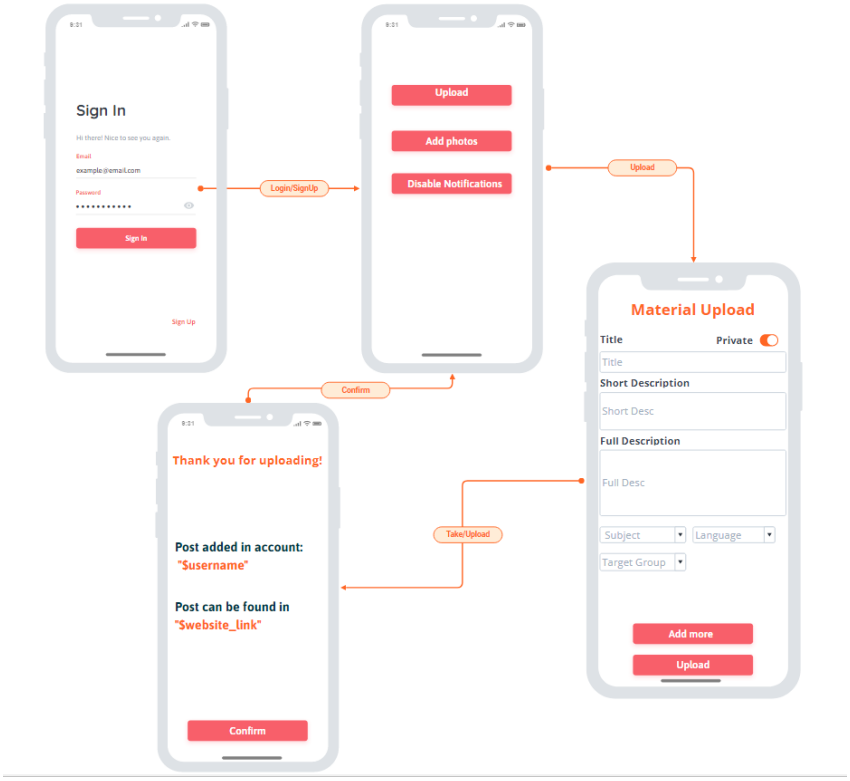

Figure 8. Android Application Wireframe

The system is also set to have an Android application. The main purpose of which is to be able to easily upload materials to the site that are on user's mobile devices and are time consuming to move to the computer for uploading. For instance, something that has recently been written on the board could be photographed and uploaded in a similar way to the previously discussed websites upload form. In addition, the Android application would serve as a means of delivering notifications about newly uploaded materials from followed users. The Android application has very limited functionality in comparison to the website and should not be used a substitute for it, but rather as a supplement. The following subsections describe the functionality of different Android activities in a short manner.

1.14.1 Login Activity

Upon opening the application, the user shall be presented with a login screen. Following a successful login to user is moved to the menu.

1.14.2 Menu Activity

Menu activity is set to contain 2 general options, upload and notification toggle. The push notifications are given by the application to the user once their followed user uploads new materials and can be manually switched on or off here.

1.14.3 Upload Activity

The upload activity works in the same manner as material upload page described in (*section 1.9*). The only notable difference between these is the fact that custom image upload for search display is not planned for the Android application, as this would most certainly create confusion for a large amount of the users what is the image and what is the material since most of the materials uploaded through the mobile device would also be of image format.

1.14.4 Results Activity

After an upload the application would display relevant application in regards to the upload. This would include a confirmation that the upload was successful, the account name of the user that was added and the link to the upload. After the exit the user goes back to the menu activity.

2 System Architecture

2.1 High-level Overview

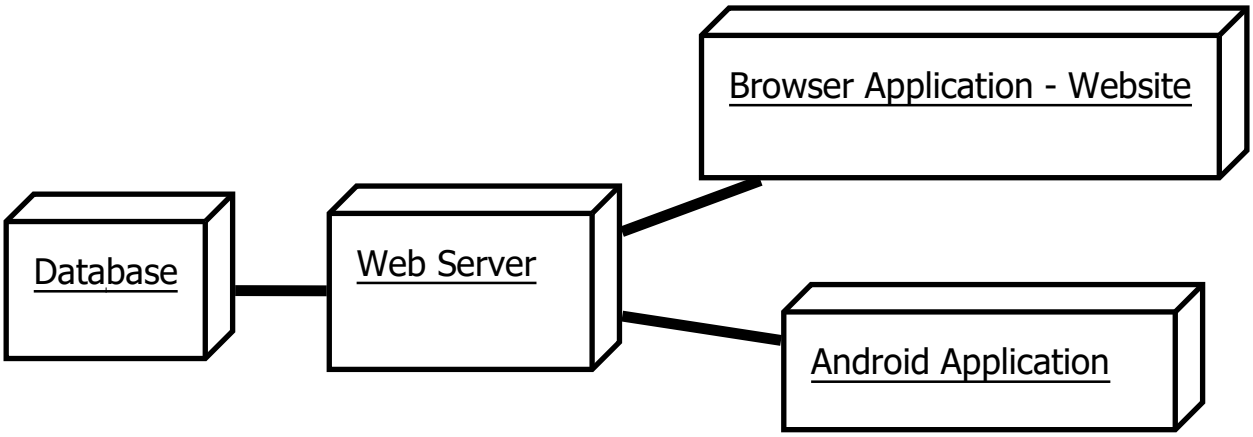

Figure 9. High-level overview of the STorm system

The Android App is being made for Android mobile devices using the Kotlin programming language.

The Browser Application - Website design is being made using Python Django and Bootstrap CSS frameworks.

The Database is needed to store teacher data, publications and other content. For this MySQL relational database management system is being used.

The Web Server is used to connect the database and the client side. It also stores media files such as publication materials that were uploaded by the users. (the "Web Server" node pictured in the diagram represents both the web server itself and the web framework for simplicity.).

2.2 Deployment

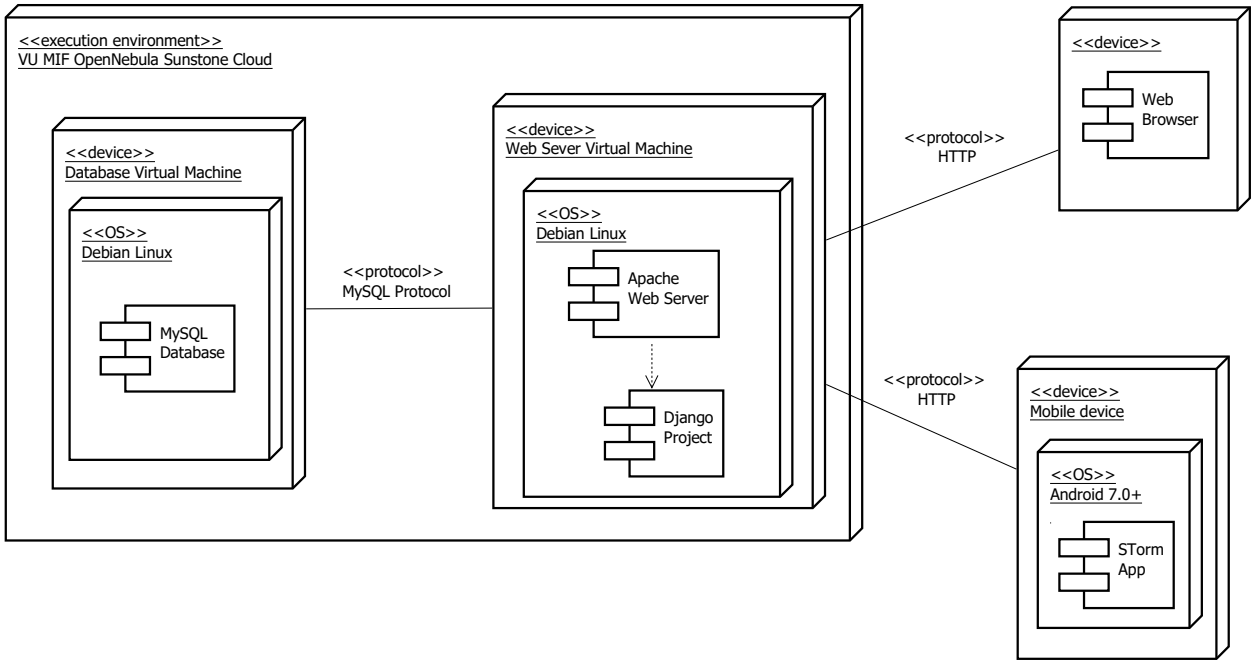

Figure 10. STorm UML Deployment Diagram

UML deployment diagram - overview of the whole system showing the deployment environment and high level interactions between the main components.

2.3 Technologies and Tools

The front-end of the browser application is being made using the Bootstrap CSS framework along with the Boomerang UI kit.
The Android app is made in Android Studio IDE using the Kotlin programming language.
The back-end of the system is programmed using the Python-based web framework Django. The framework is connected to an Apache web server using the Web Server Gateway Interface (WSGI). They are both running inside of a Linux virtual machine.
For the database part, MySQL relational database management system is used. It is running inside of a separate Linux virtual machine from the web server and web framework.

2.4 Front-end

2.4.1 Template inheritance

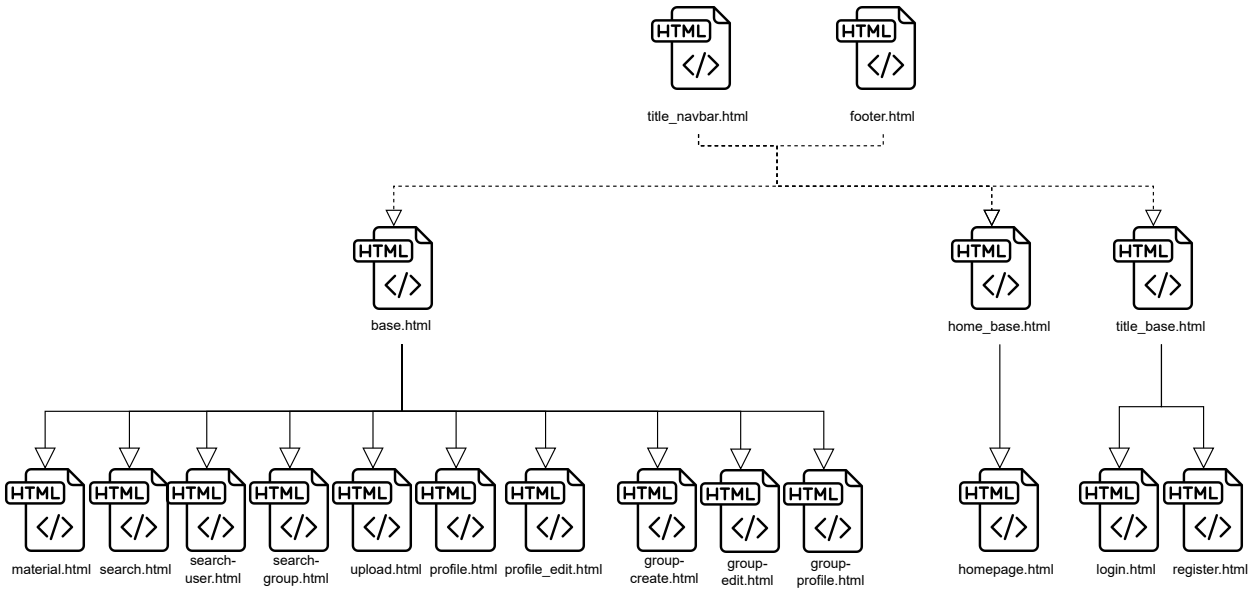

Figure 11. Template Inheritance Diagram

Part of Django's template system is template inheritance. Template inheritance allows building a base "skeleton" template that contains all the common elements of the site and defines blocks that child templates can override. The templates, which are called base.html, title_base.html, home_base.html define HTML skeleton documents. It's the job of "child" templates to fill the space with content. Inside on the templates the block tag defines blocks that child templates can fill in. All the block tag does is to tell the template engine that a child template may override those portions of the template. The diagram above (*see figure 11*) showcases the template inheritance model of the system being described.

As visible in (*figure 11*) all templates in the system in discussion uses the same custom footer and navbar. It is of major importance to note that this does not directly mean that the navigation options stay the same in the whole system. Logic in the navbar template allows for it to display different options depending on user authentication status. Footer, on the contrary, behaves the same every time and has no special logic in place.

Due to a major structural differences in design between most of the pages and title based pages (homepage and authentication pages) there is need to have different templates for main base and title/homepage base templates. Title base is used for authentication-based pages and home base is used for the homepage only. The rest of the pages follow the same design layout, thus they use the same base. All base templates in the current system include necessary script inclusions, imports and div tags that define structure and spacing that is common among numerous templates.

2.4.2 Android Application

Android application is developed as a companion to the website with the primary purpose of allowing easy upload of materials containing images, that are not easily transferable to or contained in a computer (but instead are in the users phone). For the purpose of achieving the ability to upload materials from the mobile devices, a custom REST api was built with the utilization of django-api. The android application also contains a custom client built with the utilization of retrofit 2 library and kotlin programming language.

As mentioned in section 1.14 the functionality closely resembles the upload form of the web app, the key difference is that only acceptable upload media are images of .jpeg format (standard format of pictures taken from android). The android application also contains a custom zip method in the case of user choosing more than one image for upload, the zipping is done automatically upon multiple image selection and does not require any additional action from the user. The choice of zipping files in the front-end instead of from the back-end was made due to performance limitations with the provided resources via OpenNebula, which it is a requirement to use.

The minimum target android api version for the application is set to 26 (version 8.0), security permissions required also differ due to changes at api level 29. Below api level 29 the application requires access to read and write into external storage (to create the zip file and read (and upload) from gallery)), if the device is running a device with android api level 29 or higher the application requires file management permissions for the aforementioned reasons. If the user is running the application on a device below target minimum stability is not guaranteed due to differences of how android stores files.

2.5 Back-end

The back-end mainly consists of 2 main parts: the web server and database. Both of them are deployed in VU MIF OpenNebula Sunstone cloud infrastructure on separate virtual machines. Both of them are running natively on Debian Linux.

2.5.1 Database

For this project MySQL is used as a relational database management system. Its schema was derived from ER model. (*figure 12*). Object–relational mapping (ORM) is used in order to create and manage database tables, relations and entries.

2.5.2 Web Framework

Full-stack Python based web framework Django of version 4.0.3 is being used. It consists of 4 Django 'applications':

- 'mysite' - the default 'core' of the Django project, containing the main configuration files and main Uniform Resource Locator (URL) mappings.
- 'news' - contains the code related to publications.
- 'groups' - contains the code related to methodic groups.
- 'users' - contains the code related to user (teacher) accounts.
- 'restapi' - contains the code for running the representational state transfer application programming interface (REST API), which serves the requests from the STorm Android application.

It is connected to the Apache web server using the Web Server Gateway Interface (WSGI).

2.5.3 Web Server

For serving HTTP requests, Apache web server, version 2.4.53 is used. It is running inside of the same virtual machine as the web framework

2.5.4 Application Programming Interface

For the running the representational state transfer application programming interface (REST API), an additional framework is used- Django rest framework version 3.13.1. It is being used on top of the base Django framework, integrated in the aforementioned 'restapi' Django application.

The specifications for the application programming interface end-points are described below.

1. **Login and get JWT**

   HTTP POST
   /restapi/login/

   Header: none

   Body:

```
{
    "email": "admin@storm.com",
    "password": "myfavoritepassword"
}
```

   Expected response:

```
{
    "msg": "Login Success",
    "access": "eyJ....something something",
    "full_name": "admin"
}
```

2. **Test to see if JWT is valid and working**

   HTTP GET

   /restapi/hello/

   Header:

   "Authorization: Bearer eyJ5345753fg...something something...jhdfjghdfj.hsdvjfh596849nbgf"

   Body: none

   Expected response:

```
{
    "message": "user email is admin@storm.com"
}
```

3. **Post a publication (without media/images)**

   HTTP POST

   /restapi/upload/

   Header:

   "Authorization: Bearer eyJ5345753fg...something something...jhdfjghdfj.hsdvjfh596849nbgf"

   Body:

```
{
    "title": "Another Publication from API",
    "language": "English",
    "student_year": "12",
    "is_public": "True",
    "short_description": "short description from API",
    "description": "description from api",
    "subject": "1"
}
```

   Expected response:

```
{
    "id": 16,
    "title": "Another Publication from API",
    "language": "English",
    "student_year": 12,
    "is_public": true,
    "short_description": "short description from API,
    "description": "description from api",
    "subject": 1,
    "media": null
}
```

4. **Media Upload for Existing Publication**

   HTTP PUT

   /restapi/publication/PUBLICATION_PK/FILE.name/

   Example:

   /restapi/publication/16/file.pdf/

   Header:

   "Authorization: Bearer eyJ5345753fg...something something...jhdfjghdfj.hsdvjfh596849nbgf"

   Body: binary file

   Expected response:

```
{
    "msg": "File Uploaded Sucessfully"
}
```

5. **Get all subjects from database.**

   HTTP GET

   /restapi/subjects/

   Takes no parameters, does not require authentication.
   Dumps all the subjects that are in the database.

   Expected response:

```
{"id": 3, "name": "Business Fundamentals"},
{"id": 1, "name": "Informatics"},
{"id": 4, "name": "Maths"},
{"id": 2, "name": "Psychology"}
```

2.6 Conceptual Modelling

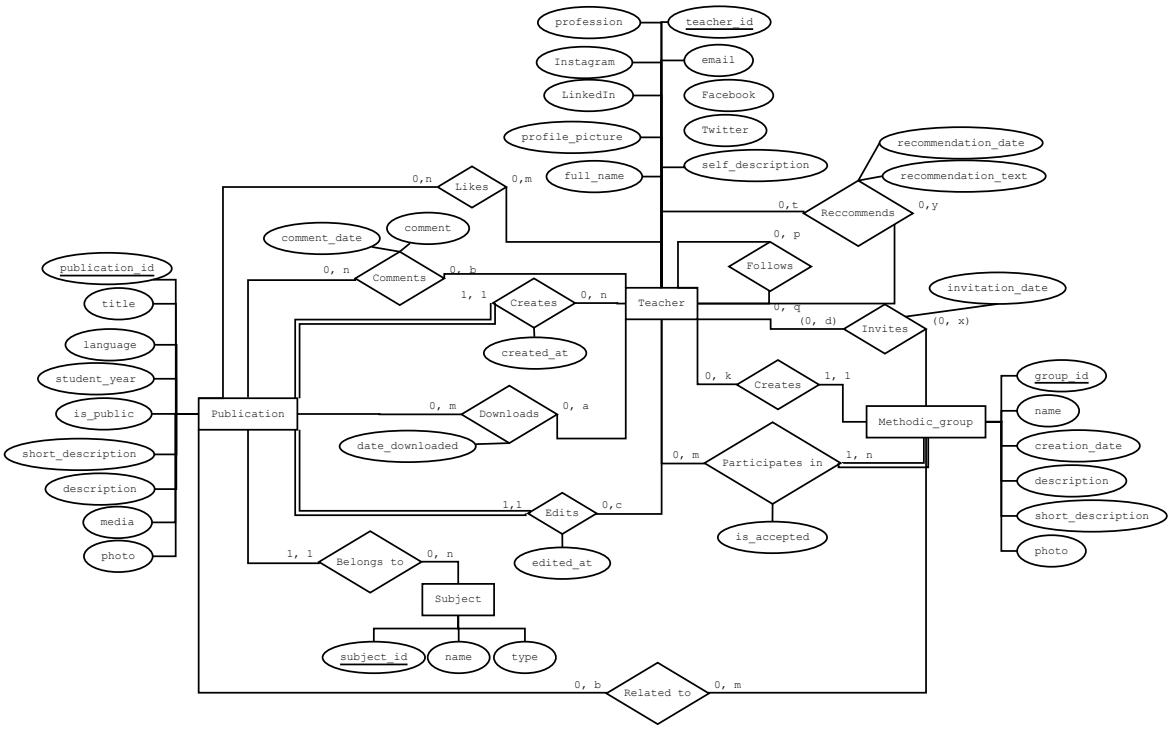

Figure 12. Entity-Relationship diagram

This diagram (*see figure 12*) represents all of the entities and relationships required for the system. The database structure is created according to this diagram.

There are 4 entities: **teacher**, **publication**, **subject** and **methodic group**. In addition to them, there are also relationships between:

1. Each subject may have multiple publications associated with it, however, each publication should belong only to one subject.
2. Any teacher can create many publications, however, every single publication cannot be created by more than one teacher. In addition, creation date of publication is saved.
3. Any publication might be downloaded by multiple teachers, who also may download many different publications. In addition, download date of publication is saved.
4. Any publication might be commented by multiple teachers, who also may comment many different publications. In addition, comment date and text of publication are saved.
5. Each publication can be edited only by the teacher who created it, but teacher can edit many publications that were created by him.
6. Any publication can be liked by multiple teachers, who also may like many different publications.
7. Any publication can be related to a methodic group if publication author (teacher) decides in such way.

8. Any teacher can follow any teacher.
9. Any teacher can recommend any teacher. In addition, recommendation text is saved.
10. Any teacher can create his/her own methodic group, however, every single methodic group can be created by only one teacher.

Entities:

The **'teacher'** entity represents a user that can create publications, like them, follow and/or recommend other teachers and be part of methodic group. Each teacher has full name and unique email address as required fields. Optionally teacher can fill in more information about him/her: add social media links (*LinkedIn, Facebook, Twitter and Instagram are supported*), exact profession or specialization (*for instance it can be Chemistry or Physical Education teacher*), write short description of himself/herself and add a profile picture.

The **'publication'** entity is the way for teachers share their work. By creating a publication a registered teacher can write any relevant information in the description section, while also having the possibility to attach media. To make use of our service more convenient, each publication has attributes to specify the year of students and the language of the publication. Moreover, as was mentioned above, each publication must be associated with one and only one subject. Furthermore, each publication has a Boolean attribute to specify whether the publication is accessible by anyone or only by other methodical groups member the author currently in. If teacher is accepted group member, he/she can make publication be related to specific methodic group.

The **'subject'** entity represents a topic of scientific field. One of its attributes is dedicated for a topic *it can be "equations" or "phonetics"* in scientific field (*it can be "Algebra" or "Lithuanian Language" accordingly*).

The **'methodic group'** entity is for teachers who are looking to find other teachers interested in the some of the same subjects and topics. It can be anything starting from *"Fundamentals of mathematics for schoolchildren"* and ending with more specific topics as *"The best ways of introducing unknown variables to mathematical equations"*. Methodic groups can have publications that are related to them. Finally, methodic groups optionally have profile photo.

3 Functional Requirements

The functional requirements in this section are prioritized and will be described in the form of user stories.

3.1 High Priority

1. As a teacher, I would like to have a way to search for publications based on key words.
2. As a registered teacher, I would like to upload a publication that is related to a specific subject or course.
3. As a registered teacher, I would like to attach files (images, video, presentations, etc.) to the publication being uploaded.
4. As a registered teacher, I would like to be able to add self description in my profile in order to better represent myself.
5. As a registered teacher, I would like to add links of my social media accounts to my profile, so that I could gain a following on other platforms.
6. As a registered teacher, I would like to add a profile picture, so that I could represent myself in a more personalized way.

3.2 Medium Priority

1. As a teacher, when I am searching for material I would like to filter the publication search results based on student year, language and subject to find most relevant publications.
2. As a registered teacher, I would like to be able to upvote ("like") a publication.
3. As a registered teacher, I would like to be able to see a total count of likes on all publications.
4. As a registered teacher, I would like to be able to receive recommendations from other teachers in the form of comments on my profile.
5. As a registered teacher, who wants to keep my identity secret, I would like to be able to post lesson material anonymously.
6. As a registered teacher, I would like to be able to quickly and easily upload a photo from my phone.
7. As a registered teacher, I would like to create a group that other teachers will be able to join.
8. As a registered teacher, I would like to leave the group I belong to.
9. As a teacher, who is a creator of a group, I would like to remove teachers from organization members.
10. As a teacher, I would like to sort search results based on upload date, publication name and rating, so I can find publication I need faster.
11. As a teacher, I would like to search for my colleagues who are registered in the system, so I can find publications posted by them.

3.3 Low Priority

1. As a teacher, I would like to switch between multiple supported languages: English, Russian and Lithuanian.
2. As a registered teacher that favors another educator's content I would like to receive notifications when said person uploads material.
3. As a registered teacher, I would like to be able to see the leaderboard of all the users based on reception of published content.
4. As a registered teacher, I would like to leave comments and suggestions on publication pages.
5. As a teacher, when looking at another teacher's profile, I would like to see top 3 publications, so I can quickly evaluate the general quality of their work.
6. As a teacher, who is considering to join the learning content sharing system "STorm", I would like to read more about the system in the homepage, so I can make my decision.

4 Non-functional Requirements

4.1 Compatibility

1. The website should run properly on the latest versions of Chrome and Mozilla Firefox and Edge browsers.
2. The Android application should run on Android 7.0 and above.
3. The system must run in Vilnius University infrastructure.

4.2 Usability

1. Teachers should be able to intuitively navigate through the website.

4.3 Reliability

1. The system should provide a non-frustrating experience free from major bugs and crashes.

4.4 Maintainability

1. The code should be written according to the common practices of the languages used, as to facilitate maintainability.

4.5 Security

1. The database and the server should be in separate virtual machines.

5 Testing

5.1 Test Outline

The following system tests will be described:

SYS001 - Registration.

SYS002 - Login.

SYS003 - Material Search by keywords.

SYS004 - Material Filtering.

SYS005 - Material Sorting.

SYS006 - Material Upload.

SYS007 - Material View.

SYS008 - Material Picture View.

SYS009 - Material Download.

SYS010 - Viewing Own Profile.

SYS011 - Viewing Other User’s Profile.

SYS012 - User Search.

SYS013 - Profile Edit.

SYS014 - Profile Recommendations (comments).

SYS015 - "Like" functionality.

SYS016 - Dynamic Homepage.

5.2 System Tests

SYS001

Title: System test for registration of new users.

Description: User should be able to enter new account data and create an account in the STorm website.

Precondition: Email used for registration should not be already taken.

Assumption: Database VM, Web Server VM and client-side device all have access to the internet.

Test Steps:

1. Open the STorm website
2. Click on the login button on the top right of the window.
3. Click on “Register Now!” text.
4. Enter email, full name, password and repeat password for the new account that is to be created.
5. Click the white "Sign Up" button at the bottom of the form.

Expected Result: Login screen should be shown with 2 field: email and password.

SYS002

Title: System test for login (authentication) of users.

Description: User should be able to login to the STorm website using the credentials of an existing account.

Precondition: An account with valid credentials is already exists and is known to the tester.

Assumption: Database VM, Web Server VM and client-side device all have access to the internet.

Test Steps:

1. Open the STorm website.
2. Click on the login button on the top right of the window.
3. Enter email and password for the new account that was already created.
4. Click the white "Sign In" button at the bottom of the form.

Expected Result: User profile should be shown containing their full name.

SYS003

Title: Material Search by keywords.

Description: A user should be able to search for publications using keywords.

Precondition: There must be at least two publications with unique names already in the STorm system.

Assumption: Database VM, Web Server VM and client-side device all have access to the internet.

Test Steps:

1. Open the STorm website.
2. Click on the "Pages" dropdown menu.
3. Click "Material Search".
4. From the "Results" section, pick a publication with a unique name.
5. Enter one or more words from the selected publication title into the search bar at the top of the page.
6. Click the "Search" button.

Expected Result: Only the publications that have the entered keywords in their title are displayed.

SYS004

Title: Material Filtering

Description: A user should be able to filter materials based on subject, grade and language.

Precondition: There must be publications with unique subjects, grades and languages already in the STorm system.

Assumption: Database VM, Web Server VM and client-side device all have access to the internet.

Test Steps:

1. Open the STorm website.
2. Click on the "Pages" dropdown menu.
3. Click "Material Search".
4. From the "Subject" selector near the search bar, pick a subject.
5. From the "Grade" selector near the search bar, pick a grade.
6. From the "Language" selector near the search bar, pick a language.
7. Click the "Search" button.

Expected Result: Only the publications that belong to the selected subject, grade and language are shown.

SYS005

Title: Material Sorting

Description: A user should be able to sort materials alphabetically (A-Z or Z-A) and by upload date (newest or oldest).

Precondition: There must be several publications with unique titles.

Assumption: Database VM, Web Server VM and client-side device all have access to the internet.

Test Steps:

1. Open the STorm website.
2. Click on the "Pages" dropdown menu.
3. Click "Material Search".
4. From the "Sort" selector near the search bar, pick a sorting option.
5. Click the "Search" button.
6. Repeat steps 4-5 using different sorting options until all of the sorting types have been tested.

Expected Result: Publications are sorted according to the option that was selected.

SYS006

Title: Material Upload

Description: A user should be able to upload a publication with attached pircture and media files.

Precondition: A user must be authorized (logged in) in order to perform this test.

Assumption: Database VM, Web Server VM and client-side device all have access to the internet.

Test Steps:

1. Open the STorm website.
2. Click on the "Pages" dropdown menu.
3. Click "Material Upload".
4. Enter some title in the "Material Title" field.
5. Select subject, grade and language using the selectors below the title field.
6. Click on the "Choose File" field for the media contents and select a file from your computer. (note: the file name must not contain any non-latin characters.)
7. Click on the "Choose File" field for the publication photo and select a photo from your computer. (note: the file name must not contain any non-latin characters.)
8. Input some text in the short description text field.
9. Click the "Submit" button at the bottom of the page.

Expected Result: User is redirected to the material search page, uploaded publication is visible.

SYS007

Title: Material View

Description: A user should be able to view the contents of a publication.

Precondition: There must be at least one publication already in the STorm system.

Assumption: Database VM, Web Server VM and client-side device all have access to the internet.

Test Steps:

1. Open the STorm website.
2. Click on the "Pages" dropdown menu.
3. Click "Material Search".
4. From the "Results" section, pick any publication.
5. Click the "View Material" button for the desired publication.

Expected Result: The user is redirected to the page where the following information about the publication is displayed: Subject, Grade, Language, Author, number of likes and description.

SYS008

Title: Publication Picture View

Description: A user should be able to see the publication photo if it was provided during creation.

Precondition: Completed SYS006 - There must be an uploaded publication with a publication picture attached.

Assumption: Database VM, Web Server VM and client-side device all have access to the internet.

Test Steps:

1. Open the STorm website.
2. Click on the "Pages" dropdown menu.
3. Click "Material Search".
4. Find the publication that is known to have the "publication picture" attribute populated.

Expected Result: The picture that was attached to the publication is visible in the material search.

SYS009

Title: Material Download

Description: A user should be able to download the publication media if it was provided during creation.

Precondition: Completed SYS006 - There must be an uploaded publication with media attached.

Assumption: Database VM, Web Server VM and client-side device all have access to the internet.

Test Steps:

1. Open the STorm website.
2. Click on the "Pages" dropdown menu.
3. Click "Material Search".
4. Find the publication that is known to have the "publication media" attribute populated.
5. On the publication tile, click "View Material"
6. Click the "Download" button at the bottom of the page.

Expected Result: The file that is attached to the publication is downloaded to the user's computer.

SYS010

Title: Viewing own profile

Description: A user should be able to view their own profile.

Precondition: A user must be authorized (logged in) in order to perform this test.

Assumption: Database VM, Web Server VM and client-side device all have access to the internet.

Test Steps:

1. Open the STorm website.
2. Click on the "Pages" dropdown menu.
3. Click "Profile".

Expected Result: The self profile page of the user that is logged in is shown, along with their full name. "Edit profile" button is visible.

SYS011

Title: Viewing Another User's Profile

Description: A user should be able to view another teacher's profile.

Precondition: There must be at least one publication in the STorm system.

Assumption: Database VM, Web Server VM and client-side device all have access to the internet.

Test Steps:

1. Open the STorm website.
2. Click on the "Pages" dropdown menu.
3. Click "Material Search".
4. Find any publication.
5. Click on the uploader's name above the publicaiton.

Expected Result: The profile page of the user that is uploaded said material is visible.

SYS012

Title: User Search

Description: A user should be able to search for other users by name.

Precondition: A user must be authorized (logged in) in order to perform this test. There must be at least two users with unique names already in the STorm system.

Assumption: Database VM, Web Server VM and client-side device all have access to the internet.

Test Steps:

1. Open the STorm website.
2. Click on the "Pages" dropdown menu.
3. Click "User Search".
4. From the "Results" section, pick any user.
5. Enter the user's name into the search bar.
6. Click the "Search" button.

Expected Result: Only the users that have the given name are shown.

SYS013

Title: Profile Edit

Description: A user should be able to edit their own profile.

Precondition: A user must be authorized (logged in) in order to perform this test.

Assumption: Database VM, Web Server VM and client-side device all have access to the internet.

Test Steps:

1. Open the STorm website.
2. Click on the "Pages" dropdown menu.
3. Click "Profile".
4. In the profile page, click "Edit Profile"
5. Input profession text.
6. Attach profile picture file from your computer.
7. Input profile description.
8. Attach all possible social media links (facebook, twitter, instagram, linkedin).
9. Click the "Save" button.

Expected Result: The user is redirected to their profile page. Profession, picture, profile description and buttons for social media links are visible and working.

SYS014

Title: Profile Recommendations (comments)

Description: A user should be post recommendations (in the form of comments) on other user's profiles.

Precondition: A user must be authorized (logged in) in order to perform this test.

Assumption: Database VM, Web Server VM and client-side device all have access to the internet.

Test Steps:

1. Open the STorm website.
2. Click on the "Pages" dropdown menu.
3. Click "User Search".
4. From the "Results" section, pick any user.
5. Find the text field that is below the banner "Write your recommendation".
6. Input some encouraging text into the field.
7. Click on the "Recommend this teacher!" button.

Expected Result: The comment immediately appears on the selected user's profile.

SYS015

Title: "Like" functionality

Description: A user should be able to "like" a publication.

Precondition: There must be at least one publication already in the STorm system.

Assumption: Database VM, Web Server VM and client-side device all have access to the internet.

Test Steps:

1. Open the STorm website.
2. Click on the "Pages" dropdown menu.
3. Click "Material Search".
4. From the "Results" section, pick any publication.
5. Click the "View Material" button for the desired publication.
6. Click on the button "Click to like!" next to the red heart.

Expected Result: The like counter is increased by one. If this publication is already liked by the user, nothing changes.

SYS016

Title: Dynamic Homepage

Description: The homepage should show 3 users and 3 publications.

Precondition: There must be at least 3 publications and at least 3 users in the STorm system.

Assumption: Database VM, Web Server VM and client-side device all have access to the internet.

Test Steps:

1. Open the STorm website.
2. Click on the "Pages" dropdown menu.
3. Click "Homepage".
4. Navigate to the "Active Community of Teachers" section at the middle-bottom part of the page.
5. Navigate to the "Featured Materials" section at the bottom part of the page.

Expected Result: The "Active Community of Teachers" section should showcase the profiles of 3 teachers and the "Featured Materials" should show 3 publications.

6 Setup Tutorial for Website Hosting

6.1 Web Server Setup

step 1. Create an Ubuntu/Debian based virtual machine (or install OS on dedicated server.)
step 2. Run this script with root privileges:

```
#!/bin/bash
apt update
apt -y install apache2
apt -y install python3-pip
apt -y install python3-venv
apt -y install libapache2-mod-wsgi-py3
apt -y install python3-dev default-libmysqlclient-dev build-essential
apt -y install git
pip3 install --upgrade pip
```

step 3. Pull the project from GitLab (or any other source) and place your desired directory, for example: /home/linuxuser/storm
step 4. Navigate to /home/linuxuser/storm/src/Back-end/
step 5. Create and activate a Python virtual environment by running these two commands:

```
python3 -m venv venv
source venv/bin/activate
```

step 6. Run this script after activating venv:

```
#!/bin/bash
pip3 install Django
pip3 install mysqlclient
pip3 install Pillow
pip3 install django-crispy-forms
pip3 install djangorestframework
pip3 install djangorestframework-simplejwt
```

Or use one of the following commands:

```
pip install -r requirements.txt (Python 2)
pip3 install -r requirements.txt (Python 3)
```

step 7. Open the Apache configuration file located in /etc/apache2/apache2.conf and add this at the end:

```
Alias /static /home/linuxuser/storm/src/Back-end/mysite/static

<Directory /home/linuxuser/storm/src/Back-end/mysite/static>
    Options Indexes FollowSymLinks
    AllowOverride None
    Require all granted
</Directory>
```

step 8. Open the Apache configuration file located in /etc/apache2/sites-avaliable/000-default.conf and add this after deleting existing contents:

```
<VirtualHost *:80>
        ErrorLog  /home/linuxuser/error.log
        CustomLog /home/linuxuser/access.log combine

      <Directory  /home/linuxuser/storm/src/Back-end/mysite/mysite>
                Require all granted
        </Directory>

      WSGIDaemonProcess WSGId python-home=/home/linuxuser/storm/src/Back
end/venv python-path= /home/linuxuser/storm/src/Back-end/mysite
        WSGIProcessGroup WSGId
      WSGIScriptAlias / /home/linuxuser/storm/src/Back-end/mysite/mysite

</VirtualHost>
```

step 9. Create a file named mysql.conf in storm/src/Back-end/mysite with the following contents: (replace DATABASE_IP with the IP address of your database VM or server)

```
[client]
host = DATABASE_IP
database = devdb
user = web_server
password = mysecretdatabasepassword
default-character-set = utf8
```

step 10. Open the file settings.py located in /home/linuxuser/storm/src/Back-end/mysite/mysite/settings. and add your hosts IP to ALLOWED_HOSTS variable on line 28 step 9. Make sure port 80 of the web server is forwarded and try the site in browser.

step 11. Set the permissions of the directory /home/linuxuser/storm/src/Back-end/mysite/media and set directory owner to Apache.

```
chmod -R 775 /home/linuxuser/storm/src/Back-end/mysite/media
chown -R www-data:www-data
/home/linuxuser/storm/src/Back-end/mysite/media
```

6.2 Database Setup

step 1. Create a virtual machine (or install desired OS on dedicated server.)
step 2. Install MySQL
step 3. Enter mysql DBMS commandline and run these commands (replace 'PRIVATE_IP' with your web server's IP address and mysecretdatabasepassword' with your desired password):

```
CREATE USER 'web_server'@'PRIVATE_IP' IDENTIFIED BY
'mysecretdatabasepassword';
GRANT ALL PRIVILEGES ON devdb.* TO 'web_server'@'PRIVATE_IP';
FLUSH PRIVILEGES;
```

step 4. Connect to the web server VM that was created in section 5.1 and run these commands:

```
cd storm/src/Back-end/mysite
python3 manage.py makemigrations
python3 manage.py migrate
```

7 Competitive Analysis

7.1 Analysis of Similar Systems

7.1.1 "PhET" by University of Colorado Boulder [2]

Advantages:

- **Filter based on subject, grade level, language**
- **Grid and list views** - users can chose whether look at search results
- **Every lesson has a description** - topics that are covered, learning goals and teaching resources.
- **Lessons can be translated by volounteers** - any lesson can be translated to any language by people who are familiar with the language lesson going to be translate to.
- **The website seems intuitive, easy to navigate** - only necessary buttons and options can be accessed by users.
- **Built-in games** - built-in games are extremely helpful to understand a material.
- **Interactive demonstrations** - in some way similar to built-games, however without gamefication elements, demonstrations are extremely helpful for students to understand topic through experimenting.

Disadvantages:

- **Only registered users can download learning materials** - some teachers might want to download learning materials without going through registration and authentication processes.
- **No rating system** - rating system might be helpful as it would allow to sort poorly-made materials and get the top-rated ones.
- **No filter based on a topic** - some teachers might want to quickly search for materials that are related more specific topic.
- **Material only for STEM subjects** - as the website has already well-developed system, it is a big loss not to allow teachers of subjects that are not considered as STEM subjects load their materials.

General thoughts:

The website can be used as a great addition for teachers to provide additional illustrative and/or interactive materials during their lessons. Filtering is intuitive and well-done, but if there would be more lessons, the filtering functionality would not be enough.

The main advantage of the website is interactive demonstrations and support for translations to multiple languages. Disadvantage is that content creators are hidden in separate tab or even missing - this can discourage people from creating quality material.

7.1.2 "ReadWriteThink" by NCTE[3]

Advantages:

- **The website seems intuitive, easy to navigate.**
- **There is a way to filter material based on 'Grade', 'Type' and 'Topics'.** This functionality is available immediately on the homepage.

Disadvantages:

- **Large amount of content on screen at once** - this can sometimes be overwhelming.
- **Purpose of "collections" tab is not entirely clear.** A first time user may have some trouble understanding the functionality

General thoughts:

This site seems well polished, contains lots of high quality content for teaching, along with a way to filter it. It can be taken as an example of a good teacher focused website.

7.2 Teacher Interviews

Since the target demographic for our system is already known and straightforward it was in our best interest to expand the scope of our competitive analysis by not only looking at other applications with similar goals. Because of that, we have also conducted numerous interviews with teachers who our system is directed toward. Seven separate approximately 30 minute one-to-one interviews of qualitative research approach were conducted with the current teachers of Trakai Vytautas Magnus Gymnasium. They consisted of showcasing the preliminary design of the system, asking for feedback and giving some predetermined questions which were based on the problems that our system wishes to solve during this project. Results are as follows.

Firstly, most teachers have said that they utilize the google search engine as their primary search for material, but filtering and finding good quality material is difficult and time consuming this way. Our system is trying to solve this issue by streamlining the process. The fact that all teachers said that google is their primary search tool shows that there are no prominent local services that have the same functionality as we plan to have in our system, thus giving a slight competitive edge. We have also gotten generally favorable feedback regarding the design and functionality showcased in the mockups, especially regarding the usefulness of the android application which prompts to reaffirm the idea that this project topic is worth pursuing.

Though there were some concerns in the interviews regarding getting people to start using the system initially, attracting the user base. Thus our focus primarily falls into putting a lot more effort in regards to new teachers and making the website another place which could enhance the possibilities for new teachers that are building their careers to represent themselves. That includes the features of profile building: ability to receive recommendations from other teachers, ability to add social media links. showing likes on the uploaded material, as well as other metrics. Essentially, the systems goal is to be another way for new teachers to represent their skills and gain recommendations from their peers and thus enhance their careers as educators by uploading created materials. This focus on the system is especially relevant today as the Lithuanian government is very actively pushing for more new teachers to enter the field.

8 Human Computer Interaction Principle Implementation

The following section describes parts of the functionality of the system in accordance to Human Computer Interaction principals published by Jakob Nielsen in 1994 [1] to help individual evaluators make their assessments and improve evaluation effectiveness. This section is by no means a direct representation of all the functionality of the system but rather an evaluation of features of the system that would provide a proper user experience in based on the principals described by Nielsen.

1. **Visibility of system status** - the system keeps users up to date on the current state and actions by providing appropriate visual cues and feedback. This is achieved by clear, large labels (titles) on each page where the user is as well as the tab window in the browser made to showcase the page name. Additionally, if the user is not logged in but wants to perform certain tasks a direct would ask them to log in.

2. **User control and freedom** - the system arranges information in a simple yet orderly manner. The features such as buttons that represent a certain action do fulfill that specific action, unless otherwise indicated to the user with greying. Users access to certain pages is controlled through authentication.

3. **Match between system and the real world** - search functionality is made in order to match the real world with its search filtering options, as the representations of "class", "subjects" and "languages" as well as their provided correspond to the regular local school curriculum in Lithuania.

4. **Consistency and standards** - all pages in STorm system follow the same design layout and color scheme with the utilization of the same UI kit. Page names and labels are simple and easily recognizable. Navigation implemented in the navbar is done in a logical pattern with each spinner's name indicating of the options it provides. Page translation is performed from a special file that was translated manually to allow consistent language change that would not be possible with auto-translation.

5. **Recognition rather than recall** - most user interface elements contain labels to allow the user to easily recognize what action/purpose they perform. All pages have labels on the top of them, all search, sort and filter options contain hints that tell the user of what is supposed to be the input there. Buttons also contain either easily recognizable images on them to indicate their purpose, or word labels that tell the purpose to the user.

6. **Error prevention** - STorm system features both front-end (Bootstrap) constraints as well as back-end constraints to prevent users from inputting incorrect information in form-based pages. Any attempt at accessing pages that require user authentication redirects to login pages.

7. **Aesthetic/Minimalist design** - the website does not contain major unnecessary or clutter-some details/objects in the layout. Each button and reference has a purpose in the overall implementation and is labeled to convey that purpose to the user. The UI is made to be pleasant on the eye, with matching color scheme and slight hover effects that add a slight amount of life to the application.

8. **Error recognition and recovery** - STorm system is designed to allow the user to recognize the mistakes they make and adjust them. The login and registration page forms all contain constraints that, if not fulfilled or left empty, will indicate the user of the problem and specify the exact mistake. This type of implementation is also prevalent in other form-based pages in the system.

9. **Flexibility and efficiency** - the system provides tools for users to quickly find what they are looking for in an efficient manner. The material search provided by the system contains several options such as sorting by rating, uploader rating, upload date and alphabet, filtering by different tags including target class, subject and language as well as the primary name keyword search. Additionally, the functionality for finding groups or users based on their name is also provided in a simple keyword search. The navigation bar is also structured in a way for each spinner's title to contain indication what the possible navigation options it would contain.

10. **Help and documentation** - the footer in STorm website contains links to git repository and this document to allow the users to better understand the application they are using.

9 Conclusions

To conclude this report it is imperative to reiterate the importance of the design phase in a software project. Starting work on the project by drafting a preliminary design is far more productive than immediately attempting to write code. This is due to the fact that the design phase makes it possible to foresee upcoming problems and to start coming up with solutions well in advance. Moreover, when working in this manner, it is far easier to come up with a modular structure, as the main parts that are going to be needed can be identified fairly quickly. Deciding on the ways in which the parts are going to communicate is also easier when starting from the conceptual design. All of this enables the team to divide the workload and increase efficiency, just like the multiple threads on a modern CPU.

Moreover, the use of a modern web framework, Django, with all of its features and libraries greatly increased the efficiency of the work. This is because we have not had to "reinvent" the wheel, but rather just use well proven technologies and features written by more qualified professionals of our field.

Even though the system that was built satisfies most of the functional requirements, there are still parts that could be improved. Those being mainly: the group related features as well as further advances in internationalization.

References

[1] Jakob Nielsen. Enhancing the explanatory power of usability heuristics. https://dl.acm.org/doi/pdf/10.1145/191666.191729, April 1994.

[2] University of Colorado Boulder. Phet interactive simulations. https://phet.colorado.edu, March 2022.

[3] National Council of Teachers of English. Read write think. https://www.readwritethink.org/, March 2022.