TALLINN UNIVERSITY OF TECHNOLOGY

SCHOOL OF INFORMATION TECHNOLOGIES

IT College

Aleksandr Kezerev 185742IADB

WEB APPLICATION FOR THESES PROJECTS CREATION

Bachelor's thesis

Supervisor: Nadežda Furs

MBA

Tallinn 2021

TALLINA TEHNIKAÜLIKOOL
INFOTEHNOLOOGIA TEADUSKOND

IT Kolledž

Aleksandr Kezerev 185742IADB

VEEBILAHENDUS LÕPUTÖÖDE KAVANDITE LOOMISEKS

Bakalaureusetöö

Juhendaja: Nadežda Furs

MBA

Tallinn 2021

Author's declaration

Author's declaration

I confirm that I have prepared this dissertation independently and it has not been submitted for defense by anyone else before. All the works of other authors used in compiling the work, important views, data from literature sources and elsewhere are referred to in the work.

Author: Aleksandr Kezerev

20.04.2021

Abstract

The purpose of this thesis is to create a prototype of web application within the framework of the TalTech center of IT-didactics project to facilitate writing thesis projects and testing students' ideas for a thesis. The solution is designed for TalTech students and teachers. Moreover, it should improve the quality of theses proposed by students and reduce the amount of time that teachers spend on routine work related to thesis idea or thesis draft refinement?

The solution is a prototype ready for deployment on a server and testing with a target group. The web application allows to test an idea and get a result that can be shown to the teacher as well as to export a docx document that complies with TalTech standards.

The app will be demonstrated to the TalTech center of IT-didactics commission on May 20th 2021 in the form of a video demo.

This thesis is written in English and is 35 pages long, including 5 chapters, 30 figures and

6 tables.

Annotatsioon

Veebilahendus lõputööde kavandite loomiseks

Antud lõputöö eesmärk on luua veebirakenduse prototüüp projekti TalTech IT-didaktika keskuse raames, et hõlbustada lõputöö kavandite kirjutamist ja õpilaste lõputöö lõputöödee ideede testimist. Lahendus on mõeldud TalTechi üliõpilastele ja õpetajatele. Lõputöö tulemina valmiv lahendus peaks parandama üliõpilaste poolt pakutud tööde kvaliteeti ja vähendama õpetajate korduvale tööle kuluvat aega.

Lahendus on prototüüp, mis on valmis serveris juurutamiseks ja laiema sihtgrupiga testimiseks. Veebirakendus võimaldab katsetada ideed ja saada tulemus, mida saab õpetajale näidata, samuti eksportida TalTechi standarditele vastav docx-dokument.

Rakendust demonstreeritakse 20. mail 2021 TalTechi IT-didaktikakeskuse seminaril videosalvestuse kujul.

Käesolev lõputöö on kirjutatud inglise keeles ja on 35 lehekülge pikk, sealhulgas 5 peatükki, 30 joonist ja 6 tabelit.

List of abbreviations and terms

Smoke testing	Is preliminary testing to reveal simple and failures severe enough before release
CRUD	Create, read, update, delete
ORM	Object-Relational Mapping
ODM	Object-Document Mapping
SQL	structured query language
NoSQL	non-relationa SQL
Middleware	Software that provides services to software applications beyond those available from the operating system
Asgi	Asynchronous Server Gateway Interface
Wsgi	Web Server Gateway Interface
URL	Uniform Resource Locator,
Cascade delete	A foreign key with cascade deletes means that if a record in the parent table is deleted, the child table will automatically be deleted.
MVT	Model View Template
MVC	Model View Controller
DRY	Don’t repeat yourself
IDE	Integrated development environment
Npm	Node Package Manager
Happy path (happy flow) testing	A happy path is a default scenario featuring no exception or error conditions.

Table of content

List of figures

List of tables

1 Introduction

“Research is one of the central missions of Higher Education Institutions” [1]. In order to write a thesis, it is very important for student to objectively assess skills and, be able to clearly present idea to both the expected supervisor of the thesis and the commission. While at the stage of searching for the supervisor of a thesis, a student may not yet have any idea for thesis, or he may have sketches of idea that are far from a meaning of scientific work.

Teachers at the university have to conduct interviews with the student or conduct long dialogues by e-mail in order to understand what he wants, what he knows about research problem and whether it makes sense to continue working with this idea or not. Often teachers do this outside of working hours, as this is a time-consuming process.

This situation in the process of research and writing a thesis could be solved by a platform which can help a student to independently review the problem and to find out his interest in this problem. As a result, a student will offer idea to a teacher only after conducting personal research and going deeper into the topic.

1.1 Problem statement

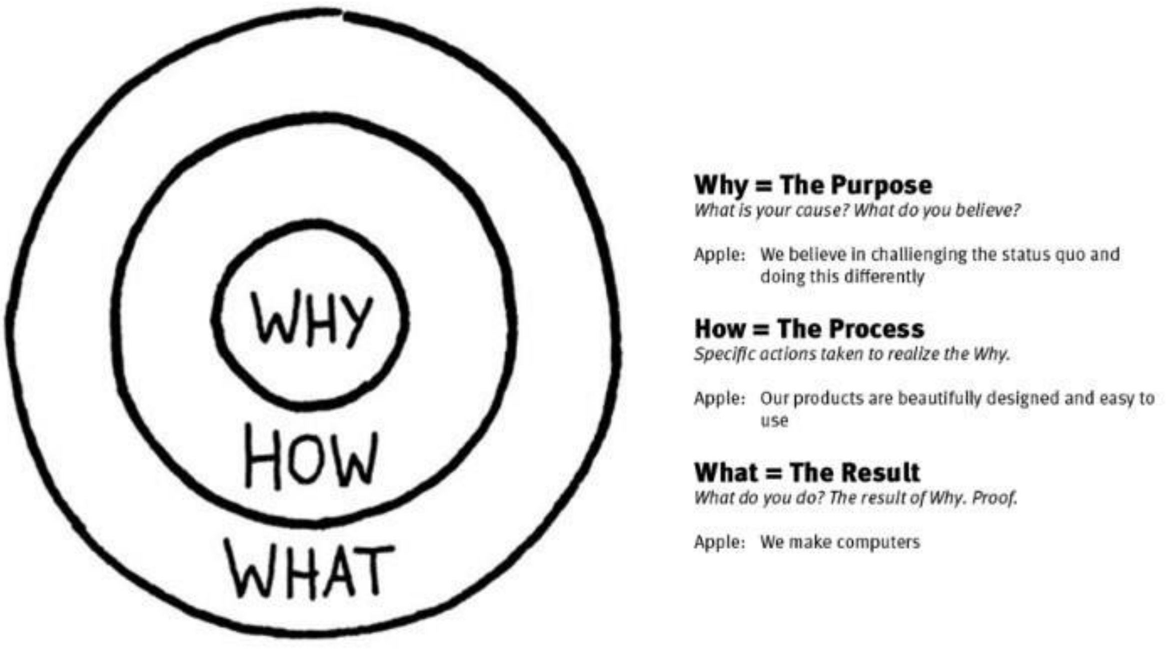
TalTech does not have a solution to speed up and facilitate the determination of an idea for a thesis and finding a supervisor for it. Also, many students do not know how to correctly assess their capabilities, ideas and skills. Students usually come to a teacher without thinking in advance about the relevance and complexity of the work. This is especially true for the theses that students write according to their own idea, and not on the orders of a company. However, even having all the necessary requirements, tools or even sketch received from the thesis topic’s provider the student may have difficulty drawing up a hesis project.

Many students do not do basic market research before proposing an idea. It is good if a supervisor can say about the irrelevance of the idea at the first interview with student. However, it is worse when the student is told by the commission at the first colloquium that his idea is not needed, since there are better analogues. In this case, a student loses motivation, because he has to change the idea, or invent the pseudo-importance of his work.

All these troubles can be avoided if the student himself, even better in advance, makes a research of the problem and available solutions. If a student has a prepared list of ideas, where it will immediately be seen how he evaluates himself in every aspect necessary for writing a thesis on this idea, with which he can come to the teacher, would be the best scenario. It could save a lot of time for student, supervisor and committees, and the student will already receive the necessary feedback and allso some skills for research.

1.1.1 Start with WHY

“Start with why” is a principle put forward by Simon Sinek in his book “Start with why” and in his TEDx speech. [2] This principle defines a “golden circle” that contains three keywords. WHAT, HOW and WHY (Figure 1). The principle shows that everyone knows what they are doing, but only some people know how they do it and even less people know why they do it. And “why” there cannot be a monetary profit or writing a thesis for just writing a thesis, but personal benefit or common good.



Why = The Purpose
What is your cause? What do you believe?

Apple: We believe in challenging the status quo and doing this differently

How = The Process
Specific actions taken to realize the Why.

Apple: Our products are beautifully designed and easy to use

What = The Result
What do you do? The result of Why. Proof.

Apple: We make computers

Figure 1 Concept of Golden Circle -By Simon Sinek [3]

This is so-called "golden circle” above, it is purposely similar to a dartboard, because it visually makes it clear that the most valuable thing is in the centre.

Based on the experience of IT college and feedback from teachers, most students approach writing a thesis from the most obvious, outer circle “what”. They start thinking about what they will do – the end result. Sometimes they already know how they would prefer to do a thesis, but they do not think about what exactly they will do, and what is the real problem that they are going to resolve. As a result, everything leads to the fact that their works are not relevant, and a student himself may lose the motivation to do it. But only few students start the work with the question "why?". In fact, this is the problem of the thesis itself. Problem statement, one of the most important parts of the thesis.

1.1.2 How to: Write a Problem Statement

Each thesis requires a clear problem statement. The main three questions in writing a thesis problem are also indicated in the problem statement guide from the University of Sheffield. [4] It also contains the questions “Who?” and “When?”. However, the main focus remains on three key questions:

- Why does someone need to focus on the problem?
- What is this problem and why the current solution does not solve it?
- And, how to solve this problem?

1.2 Relevance

The value of this thesis is that it saves a lot of time and money for the university. For example, a subscription to ideapuzzle.com described in Chapter 2 costs 19 euros per year for a student and 1000 euros for a university with an unlimited number of users. [5]

Saving time is just as important as saving money. This software will save a lot of time and reduces struggles for both teachers and student, and probably pre-defence committee. Moreover, it will help to avoid the routine work of asking typical questions that may cause confusion and drop motivation of students in an interview / colloquium.

However, all the existing solutions intended to help in writing doctoral (PhD) papers. They are too complex for writing master's or bachelor's works and have many science-related aspects. The application prepared in the framework of this thesis is aimed specifically at helping bachelors and masters.

1.3 Purpose of the thesis

Main purpose of the thesis is to make a primary working prototype of the application for a project of a thesis. The application should help students in formulating their idea, as well as visualize level of their confidence in this idea, which will improve visibility and communication with potential supervisors. Furthermore, one of the aims is to conduct smoke testing and testing with the customer. An important part of this thesis is proposals for the development of the project in the future and launch on an external server.

1.4 Author’s role

The role of the author of the work is to create and test the first prototype of the application. The author also makes an analysis of available solutions and proposes ways for the future development of the application.

In addition to author, supervisor of the thesis is also included in the project. Supervisor represents interests of TalTech center of IT-didactics in given project, performs as a customer, also provides concept, high-level requirements and participates in demos and testing, where she provides feedback.

2 Analysis

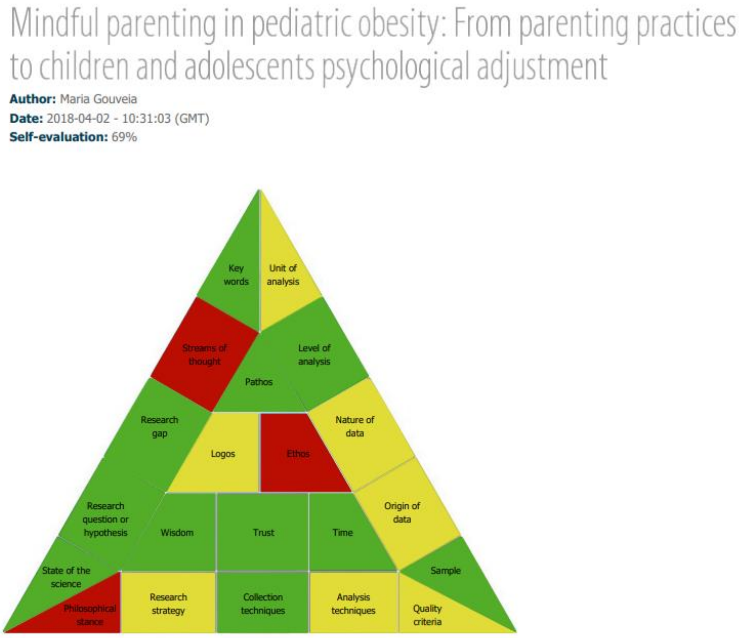
All the available solutions are analysed in this chapter, moreover, the analysis of the selected technologies for the implementation of the project and development methods is presented.

2.1 Existing solution

There are few existing solutions on this issue covering the entire problem. The main one is IdeaPuzzle, the rest are tools with a different set of capabilities and functionality, but they are all relatively the same. IdeaPuzzle is considered separately and more closely as the most similar solution. All the others are analysed in one chapter, because they are from the exact solution proposed by this thesis.

2.1.1 IdeaPuzzle

The closest in its idea and implementation is the IdeaPuzzle web application created by Ricardo Morais in 2007. [6] It is a supportive tool to assist researchers in the research design process by focusing on three main studies (ontology, epistemology, and methodology), which together represent a triangle (Figure 2).



Mindful parenting in pediatric obesity: From parenting practices to children and adolescents psychological adjustment

Author: Maria Gouveia
Date: 2018-04-02 - 10:31:03 (GMT)
Self-evaluation: 69%

Figure 2 An example of a triangle from the IdeaPuzzle project site [7]

However, the implementation of this project is rather difficult to use. The researcher cannot see all the questions at once, and navigation through the triangle is very unintuitive. Moreover, this project was created for people receiving a doctoral degree. As a result, the questions there are appropriate for PhD, at the same time, in order to answer this “puzzle”, bachelors and master’s students need to spend a lot of time for understanding the meaning of the questions. However, in the result they still will not get the main answer for the bachelor's and master’s work - “why”, because they are more research-oriented.

This application can convert PDF format to Word format. However, it has its own structure, which is not suitable for generating a thesis plan.

In addition to the fact that the application requires a paid subscription, after the end of the subscription for this application, the user loses the ability to view already completed puzzles (what happened to supervisor of this thesis).

This solution is not suitable for TalTech, since it has limited capabilities, because it does not offer tools to help bachelors and masters in writing thesis, it has a paid subscription and the vocabulary that is rarely in use of undergraduate students.

2.1.2 Other applications for visualizing and structuring knowledge

Other applications for visualizing and structuring student knowledge such as miro.com, mural.com, lucidchart.com or xmind.net offer a wide variety of tools. However, they are rather suitable as supporting programs for planning the thesis already during the process, than for the full preparation of an idea from the beginning.

Basically, these applications are graphic editors, kanban boards, mind maps, sketch tools, which is very convenient for any work. But there are several disadvantages here.

First, they are free for personal use with a limited number of whiteboards, while for a large organization like TalTech the price will be substantial. For example, use of one of the most popular similar platforms, Miro, costs $16 per month per person for a team of 20-49 employees. [8] The price is negotiable for teams consisting of more than 50 participants, but savings should not be expected given the number of students at the university.

Secondly, they do not have a common standard. Each user can do what is convenient for him, this complicates the initial quick assessment of the idea, since both the scale and visualization of an idea may differ from idea to idea. One student makes a mind map, the second a table, the third a diagram. These are very convenient tools to work with, but they are not suitable for the solution of general and massive problem.

2.2 Application requirements

The system proposed in the thesis should visualize the student's knowledge by golden circle diagram (Figure 1), which will help to quickly evaluate the idea itself. The prototype should have a working admin panel with CRUD functions. The application must be fast and comfortable in use.

Users can create an idea, answer questions about their idea, mark their "self-evaluation" on each question and get a result that can be sent to the teacher by email. Another useful feature of the prototype is to generate a thesis plan in .docx format for key issues to upload to Moodle as a final plan of thesis. The user can also view public works (if they are set to this status).

Ideas can be created in two languages: English and Estonian. The user can also choose the type of thesis: bachelor's or master’s. Questions are also duplicated in both languages and vary depending on the type of idea.

The app must follow TalTech's style guidelines [9]. Such as the color palette, fonts, and logo.

2.3 Technologies analysis

The following tools are used to develop the application:

1. Pycharm professional 2021.1.1 - Integrated development environment for the Python programming language;
2. Open Server Panel - A software environment for web development;
3. PhpMyAdmin - Open source web application for MySQL server administration;
4. Python 3.9.2 - Python is an interpreted, interactive, object-oriented programming language; [10]
5. Django 3.1.7 - Free framework for web applications in Python;
6. Django template - text document or a Python string marked-up using the Django template language;

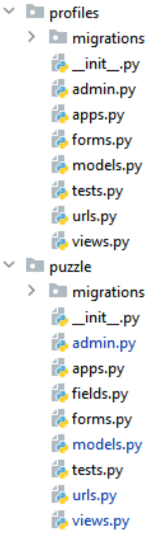
7. Bootstrap 5 - free set of tools for creating websites and web applications;
8. Git - version control system;
9. MySQL and SQLite as databases – Open sourse databases;
10. Trello – kanban table application;

2.3.1 Django framework

Django is a hight-level open source python web framework. Django is a very popular framework because of its security, stress tolerance, and speed. According to a survey made by stackoverflow.com in 2020, django is ranked 7th in the “Most loved web frameworks” ranking with 55.3% (% of developers who are developing with the language or technology and have expressed interest in continuing to develop with it). [11]

Django is used by huge companies such as Reddit, NASA, Bitbucket, Instagram, National Geographic, Spotify and Mozilla.

Django is protected from SQL injection and CSRF by default [12]. This framework is very quickly scalable. It has a modular structure (apps) that allows to build an application in parts (Figure 3). This in turn allows to change and integrate these applications into the project almost independently of the rest of the project. As a result, this framework was chosen largely because of this feature. Since this allows to organize the workflow more competently, concentrating on smaller tasks.



```
profiles
  migrations
  __init__.py
  admin.py
  apps.py
  forms.py
  models.py
  tests.py
  urls.py
  views.py
puzzle
  migrations
  __init__.py
  admin.py
  apps.py
  fields.py
  forms.py
  models.py
  tests.py
  urls.py
  views.py
```

Figure 3 Django Apps

This framework also has a convenient ORM [13] that allows make a query using Python syntax and easily switch between databases without rewriting a query code. Another big advantage of using django ORM is automatic primary key generation, auto tracking and version control of migrations.

2.3.2 Pycharm professional

Based on JetBrains surveys conducted from 2016 to 2020, it is the most popular IDE for Python development. In 2020, its share among web developers was 42%. [14] This IDE was chosen because it is the most convenient in the opinion of the author.

2.3.3 Open Server Panel

Open Server Panel is a portable software environment inspired by WampServer and Denwer, created in 2010. [15] This software was chosen, because it combines all the basic tools for developing a web application, such as: Apache, MySQL, PhpMyAdmin, PHP adminer and other useful utilities like color picker.

2.3.4 Git

Git is a distributed version control system. It is used in the development of an application for tracking versions, as well as for storing a project in a repository on GitHub. The advantage of this approach to development is that it will allow, after the completion of the prototype development, to transfer all source files to the university with all code versions, which will promote further development. The author uses only the main branch of the project for work since he makes the application alone.

2.4 Methodology analysis

2.4.1 Agile

Methods and principles from the agile methodology are used for developing the application. Kanban is mainly used, since it allows to visualize the work and concentrate on completing one task by limiting the number of simultaneous tasks, which in turn allows increasing the quality of the project. Author and supervisor decided not to use Scrum and sprints – development team size is small and there is no need to prioritize requirements each time. Atlassian's Trello app is used to implement the Kanban board.

2.5 Database selection and analysis

Chosen framework officially supports 5 databases (PostgreSQL, MariaDB, MySQL, Oracle, and SQLite) and they are all relational databases, there is no question of choosing a SQL or NoSQL database. Using third-party Object-Relational Mappings (ORM) and Object Document Mappers (ODM) for other databases can lead to unexpected errors and slow down development. Therefore, the most logical choice is to use a database from 5 officially supported databases.

The free version of Oracle (Oracle XE) can only work with 2 processor threads, 2 gigabytes of RAM and 12 gigabytes of data. [16] This database is not suitable for university use, as it expected a heavy load on the application seasonally, usually at the beginning of semesters.

SQLite database is purely for development, it is fast, stored in a single binary file, and is installed in Django by default. However, SQLite lacks user control: advanced databases provide users with the ability to manage relationships in tables according to privileges, but SQLite does not. Moreover, the impossibility of additional configuration plays crucial role: SQLite cannot be made more productive by digging into the settings – that is how it works.

The remaining 3 databases are open source. They all have their own advantages and disadvantages. However, MySQL was chosen for development, due to the fact that the author is more familiar with it, and significant difference is only noticeable on very large amounts of data. Moving from one database to another does not take long, because django's ORM supports these databases out of the box,

2.6 Project structure

The project has the following structure:

```
ProjectLibrary/
      Manage.py
ProjectLibrary/
      Config files…
      App/
      App files…
            Migrations/
      Templates/
      Static/
```

Figure 4 Django project structure

This project structure allows to keep all apps independently of each other, as well as separate apps migrations.

2.6.1 Main config files

By default, when a project is starts up in the django framework, configuration files are created that are needed to set up the project.

Table 1 Main config files

settings.py	This file is used to register all available applications (both internal and external) and add middleware. It also configures information about templates, databases, time, localization language and static files. Basically, this is the main file for the Django web application.
wsgi.py	Web Server Gateway Interface is used for deploying application on to servers like Apache.
asgi.py	Asynchronous Server Gateway interface works similarly to WSGI but provides more flexibility for development
urls.py	This file processes all the main url requests of the site including “/admin”

This table lists the main configuration files of the project on which the operation of the entire application depends.

2.6.2 Manage.py

This file is used as a command line utility to run, debug, and deploy the application. [17]

The main commands are:

1. python manage.py runserver – to start the application
2. python manage.py makemigrations – to make a database migrations via ORM
3. python manage.py migrate – to apply migrations
4. python manage.py createsuperuser – to create an admin

2.6.3 App basic structure

There are files in the project that are generated when the app is created, however, some files should be created manually to simplify the work with the app.

Table 2 App structure

admin.py	This file is needed to register entities in the admin panel.
apps.py	This file is needed to configure the application.
models.py	This file contains application entities as classes and auxiliary functions to them.
urls.py	This file is needed for routing inside the app
view.py	This file contains functions or classes for rendering templates.

The table contains only the most necessary files for the application. Also, it possible to create other files as needed in the app, for example forms.py to create custom forms that can be implemented in the template or fealds.py to create custom fields inside the database such as a phone number or ranged int.

2.7 Routing

Initially, the **urls.py** file is not automatically generated, but the author creates it in each application separately so as not to overload the main route file. Also, creating this file improves app reuse, since all internal addresses are generated by the app, and the main file only redirects the request to the application.

```
urlpatterns = [
    path('account/',
include('allauth.urls')),
    path('signup/',
profiles_views.signup,
name='signup'),
    url('admin/',
admin.site.urls,
name='admin'),
    path('profile/',
include('profiles.urls')),
    path('',
include('puzzle.urls'),
name='home'),
]
```

Figure 5 Main urls.py

The main routing file contains links to **urls.py** apps files using the **include()** method, as well as a link to the admin panel and a custom registration method.

```
urlpatterns = [
    path('', home, name="home"),
    path('myPuzzles/<slug:username>/', my_puzzles, name='my_puzzles'),
    path('create_puzzle', create_puzzle, name='create_puzzle'),
    path('detail_puzzle/<int:puz_id>', detail_puzzle,
name='detail_puzzle'),
    path('fill_puzzle/<int:puz_id>', puzzle_filling,
name='fill_puzzle'),
    path('get_docx_idea/<int:puz_id>', generate_docx,
name="get_docx_idea")
]
```

Figure 6 App urls.py file

At the same time, this file makes routing by app functions. The **path()** function receives a part of the request url, connecting it to the part of the link obtained before, then it gets the name of the method to be executed from the **view.py** file, and, as a third parameter, it takes a unique name for this route.

```
<a class="btn btn-talteh" href="{% url 'get_docx_idea' puzzle.id %}">DOCX</a>
```

Figure 7 Link example

Example: User clicks on a button on the page. Inside html a url template tag is inserted as a link (Figure 7), inside which a unique route name and a parameter are placed if necessary. Next, the framework builds a link to the specified route and executes a function from the view.py file into which it passes requset and all parameters from the link that are specified in triangle brackets that can be seen in Figure 6 in last **path()** method.

2.8 MVT architecture

The application is based on MVT (Model View Template) architecture. Template is a presentation layer which handles User Interface part completely. View is used to work with business logic and communicate with the model level, making queries to the database. There is no separate controller and complete application is based on Model View and Template. [18]

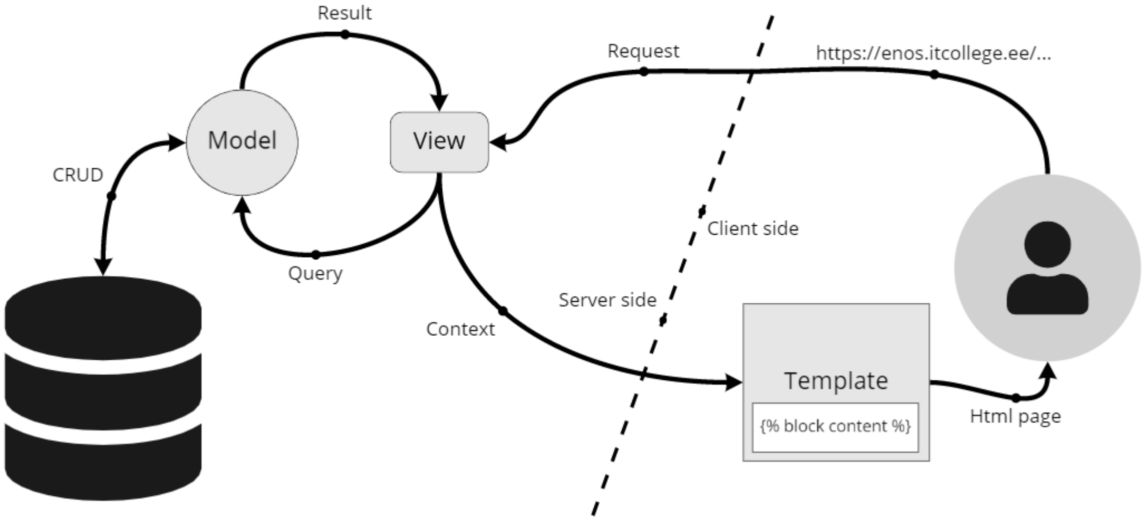


Figure 8 MVT scheme[1]

[1] The scheme made by the author of the work.

As it can be seen from the MVT diagram, it is very similar to the classic MVC (Model View Controller) architecture. But there are a number of differences between them [19]:

1. In MVC, it is necessary to write all the control specific code, while in MVT this task is taken over by the framework itself.
2. MVT is easier to modify
3. MVT allows to work with both small projects and significantly expand them over time, but MVC is not very suitable for small applications.

The use of MVT is considered a classic among python developers when working with django.

2.9 Flowchart

A flowchart is a type of diagram that represents a workflow or process. Based on this diagram, all the main functionalities of the application are developed. (Figure 9)

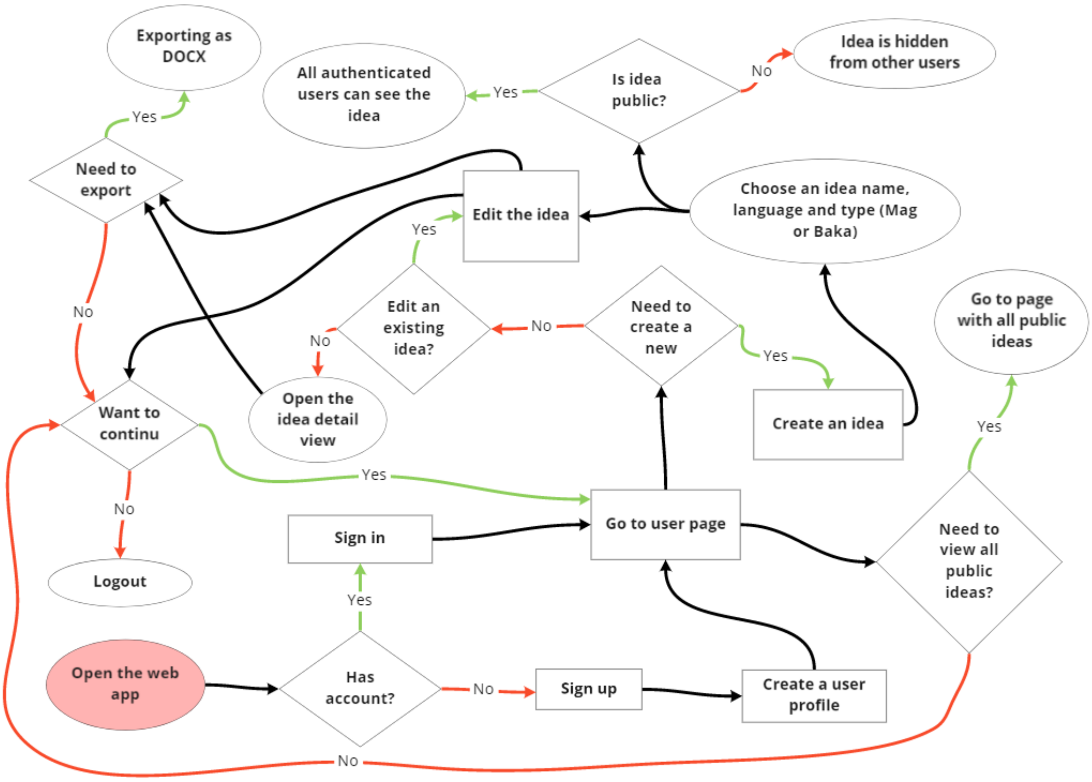


Figure 9 Flowchart diagram

The diagram shows possible scenarios for using the application. Development is underway and tests are carried out based on this diagram.

2.10 Analysis summary

The analysis covered the requirements for a web application, reviewed the available solutions, introduced the technologies and tools that were used to create the application. Furthermore, the key features of the framework were considered and the database for the application was selected in the analysis. This application can be developed and improved, since modern technologies and methods were used in its creation.

3 Implementation of the solution

The implementation of the solution is divided into three parts: development of the back-end, creation of the front-end using the template language built into the framework, as well as testing the finished prototype.

3.1 Back-end development

The following apps were used to create the backend:

1. Puzzle - This is the working title of the app, inspired by the existing solution, which is described in the chapter 2.1.1. Application needed to create and work with user ideas.
2. Profiles - This app is used to create a user profile.
3. Allauth - External app for working with registration and user logging. Further development of the project implies a complete replacement of this application with a custom one for use by the university service (such as logging in via Uni-ID)
4. Django pre-installed apps – This apps django generate automatically with initial project creation. In this work, the author does not consider these apps.

```
INSTALLED_APPS = [
'django.contrib.admin',
'django.contrib.auth',
'django.contrib.sites',
'django.contrib.contenttypes',
'django.contrib.sessions',
'django.contrib.messages',
'django.contrib.staticfiles',
'profiles',
'puzzle',
'allauth',
'allauth.account'
]
```

Figure 10 Settings.py instaled apps

To use the application, both custom and external, need to install it using the package installer, register it in the **settings.py** file like in Figure 10 and configure the routs in the main **urls.py** file.

3.1.1 Profiles app

App profiles has only one entity (In Django documentation, an entity is called a model [20]) – Profile. This model is connected by a one-to-one relationship with the predefined User model using the **@receiver** method, which creates and assigns a profile authomatically to user at the time of its registration by a user.

```
@receiver(post_save, sender=User)

def create_user_profile(sender, instance, created, **kwargs):
    """Create user profile when user is registered"""
    if created:
        Profile.objects.create(user=instance,
                               # user_id=instance.id,
                               first_name=instance.first_name,
                               last_name=instance.last_name,
                               email=instance.email
                               )
        instance.profile.save()
```

Figure 11 Function to create profile automatically

The profile has six fields besides the ID field, which is generated when the model is migrated to the database. In the future, the profile can be combined with the university's database in order to get all data about a student and a teacher from there.

Table 3 Profile model structure

user	One-to-one field to User, cascade delete
first_name	CharField, max_length=50
last_name	CharField, max_length=50
avatar	ImageField, blank=True

emai	EmailField
phone	CharField, max_length=25

It is worth doing a new migration every time when changing these fields or adding new ones. Moreover, it possible to return the old version of the migration, since Django keeps track of the versions of migrations in the database.

```
    def __str__(self):
        return f"{self.first_name}
{self.last_name}"

    class Meta:
        verbose_name = "User profile"
        verbose_name_plural = "User profiles"
```

Figure 12 Additional method and meta class for the model

Also, the __str __ () method is defined to represent the object as a string for the model and the meta class (Figure 12) to get the verbose name and plural verbose name.

3.1.2 Puzzle app

There are 9 models in this app. The relationship of entities in the application can be seen in the picture (Figure 13).The author considers in detail only the main models, because many models are auxiliary and will be considered within the main models.

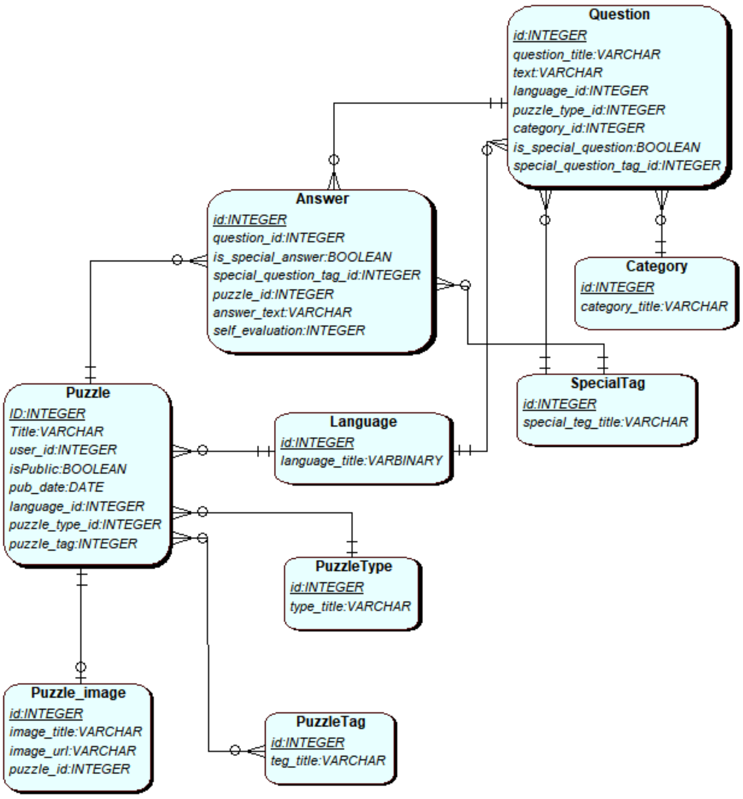


Figure 13 ERD of puzzle app

In the ERD, many-to-many and one-to-one relationships can be seen, they are depicted this way because the django ORM allows to avoid writing intermediate models in the code, but to use such relationships. (Figure 14)

```
puzzle_tag = models.ManyToManyField(PuzzleTag, blank=True)
```

Figure 14 ManyToMany relationship

However, all required in-between tables are created in database during migration. It can be seen in Figure 15.

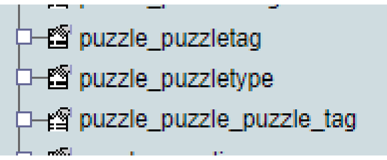

puzzle_puzzletag
puzzle_puzzletype
puzzle_puzzle_puzzle_tag

Figure 15 Example from phpMyAdmin

When working with queries, they are made through these tables without directly addressing to in-between tables.

Puzzle is a model describing an idea that a student creates in order to test it. This model has the following structure:

```
class Puzzle(models.Model):
    title = models.CharField("Title", max_length=150, blank=False,
null=False)
    user = models.ForeignKey(User, verbose_name="User",
on_delete=models.CASCADE)
    isPublic = models.BooleanField("Public")
    pub_date = models.DateTimeField("Publication date", null=True)
    language = models.ForeignKey(Language, on_delete=models.SET_NULL,
null=True, blank=False)
    puzzle_type = models.ForeignKey(PuzzleType,
on_delete=models.SET_NULL, null=True, blank=False)
    puzzle_tag = models.ManyToManyField(PuzzleTag, blank=True,
null=True)
```

Figure 16 Puzzle model

Some fields require clarification to better understand how this model works:

1. isPublic - is responsible for making this idea available to other students.
2. language - Foreign key to language. It depends on the language in which a student will be asked questions, as well as in which the project of the bachelor's or master’s work will be generated.
3. puzzle_type - Foreign key to PuzzleType. Determines the type of work, the student tests the idea for the bachelor's work or for the master's. It depends on which questions will be shown and which project will be exported as a result.
4. puzzle_tag - Foreign key to puzzleTag. Allows to assign tags to ideas in order to use them for search. It was decided to abandon this function for the first prototype, with the customer but it remains for future work.

The question model is responsible for those questions that a student will answer to test an idea.

```
class Question(models.Model):
    question_title = models.CharField("Question Title", max_length=300)
    text = models.CharField("Question description", max_length=250,
blank=True)
    language = models.ForeignKey(Language, on_delete=models.SET_NULL,
null=True)
    puzzle_type = models.ForeignKey(PuzzleType,
on_delete=models.CASCADE, null=True)
    category = models.ForeignKey(Category, on_delete=models.CASCADE,
blank=True)
    is_special_question = models.BooleanField("Question to final work",
blank=True, null=True)
    special_question_tag = models.ForeignKey(SpecialTag,
on_delete=models.CASCADE, blank=True, null=True)
```

Figure 17 Question model

This model (Figure 17) is independent of the puzzle model. They have only two common fields to receive questions from the database related to the type of this idea: puzzle_type and language

Example: a student creates an idea and indicates that he is going to write a bachelor's work in English. Thus, when testing an idea, he will answer questions in English and questions will be related to the bachelor's degree (for masters, the questions may be more difficult).

Explanation for special fields:

1. The is_special_question and special_question_tag fields are needed to generate the project of thesis.
2. The category field is responsible for the category of the question (Why? How? And What?) For the subsequent compilation of the test result.

The answer model refers to both the question and the puzzle.

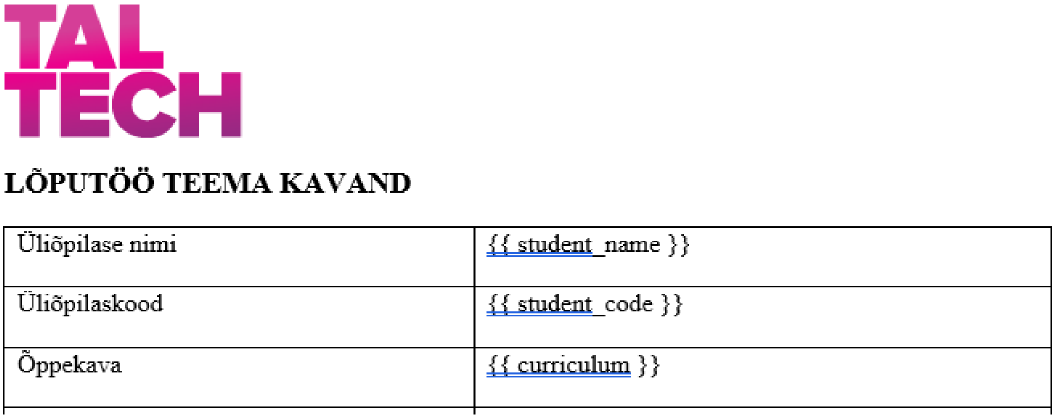
```
class Answer(models.Model):
    question = models.ForeignKey(Question, verbose_name="Question",
on_delete=models.CASCADE)
    is_special_answer = models.BooleanField("Answer to final work",
blank=True, null=True)
    special_question_teg = models.ForeignKey(SpecialTeg,
on_delete=models.CASCADE, blank=True, null=True)
    puzzle = models.ForeignKey(Puzzle, on_delete=models.CASCADE)
    answer_text = models.TextField("Answer text", max_length=500)
    self_evaluation = IntegerRangeField(min_value=1, max_value=100,
default=1)
```

Figure 18 Answer model

Also, this model has fields responsible for consideration of this answer when creating a project of thesis. These fields are inherited from the question when the answer is created. Only answer_text and self_evaluation fields change when editing an answer, and the rest remain the same, thus, when a student changes his answer or self-evaluation, a new answer is not created, but an old one is changed, this allows more economical use of database resources. However, if the answer did not exist before, then a new one is created.

3.1.3 Docx export and the golden circle generating

The **docxtpl** package is used to create a docx, this package combines the **python-docx** packages for creating and reading a docx document and **jinja2** for document templating. It makes no sense to create a document from scratch, as it complicates the management of export in the future. The document is generated in a buffer and is not stored on the server, but it is immediately downloaded from the user.



TAL
TECH

LÕPUTÖÖ TEEMA KAVAND

Üliõpilase nimi	{{ student_name }}
Üliõpilaskood	{{ student_code }}
Õppekava	{{ curriculum }}

Figure 19 Docx template example

Document templating looks like in Figure 19. Those fields that can be generated based on the student's answer have a tag inside them enclosed in curly braces. These tags correspond to those in the question model. This allows to easily modify the document template in the future, as well as moderate the generation of the project by a person who does not understand the program code.

There are 4 templates in total: for bachelors in English; for bachelors in Estonian; for masters in Estonian; for masters in English. The template in use depends on the language and the type of idea a student has chosen when creating it. Tags in templates are used the same. However, it is impossible to create a fully completed project, since there are fields that a student must enter manually, for example, the name of the supervisor. In order to avoid the problem, when a student will send the work that he has not viewed, the message "This is automatically generated project" is inserted into the notes field by default.

For golden circle generating the graphic library **Pillow** is used. After a student confirms the completion of testing an idea, a picture (Figure 20) that reflects his average self-assessment in 3 categories of questions (Why? How? What?) is generated and assigned to the Puzzle model.

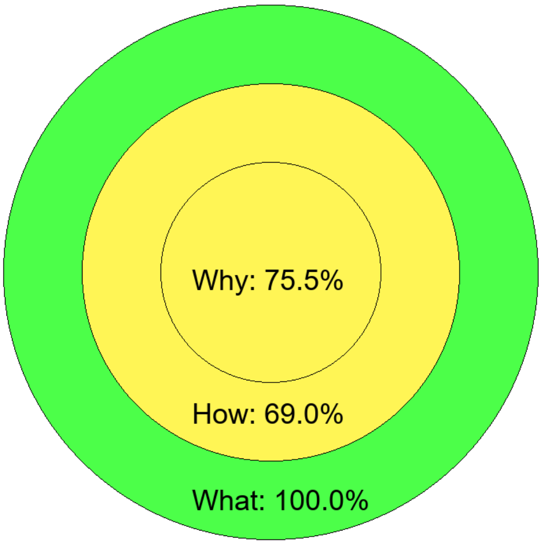


Figure 20 Generated golden circle

If a student changes his answers in the test, then a new picture is generated, and it replaces the previous one. This allows to use server resources more economically, similar to the answer model. The name of the diagram is generated uniquely using idea title, users' username, and idea ID. Colors depend on the average student's grade in each category: green is 82 points inclusive and up to 100; yellow is from 62 to 81 points inclusive; red is from 1 to 61 points inclusive.

3.2 Front-end

For the frontend, the application uses Django template language, bootstrap 5 and custom CSS for website design in TalTech style. All files used to create the user interface by analogy with apps must be registered in the settings.py file.

```
TEMPLATES = [
    {
        'BACKEND':
      'django.template.backends.django.DjangoTemplates',
        'DIRS': [BASE_DIR / 'templates',
                 'templates/profiles',
                 'templates/puzzle',
                 'templates/account'
                 ]
        ,
        'APP_DIRS': True,
        'OPTIONS': {
            'context_processors': [
                'django.template.context_processors.debug',
                'django.template.context_processors.request',
                'django.contrib.auth.context_processors.auth',
              'django.contrib.messages.context_processors.messages',
            ],
        },
    },
]
```

Figure 21 Templates registration in settings.py

There is no need to register each template separately, but only to register the path to the folder where the html templates are located.

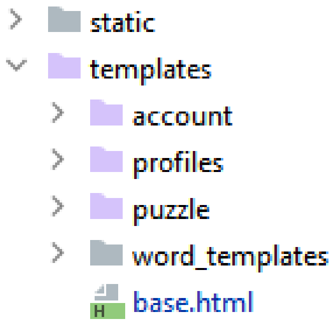
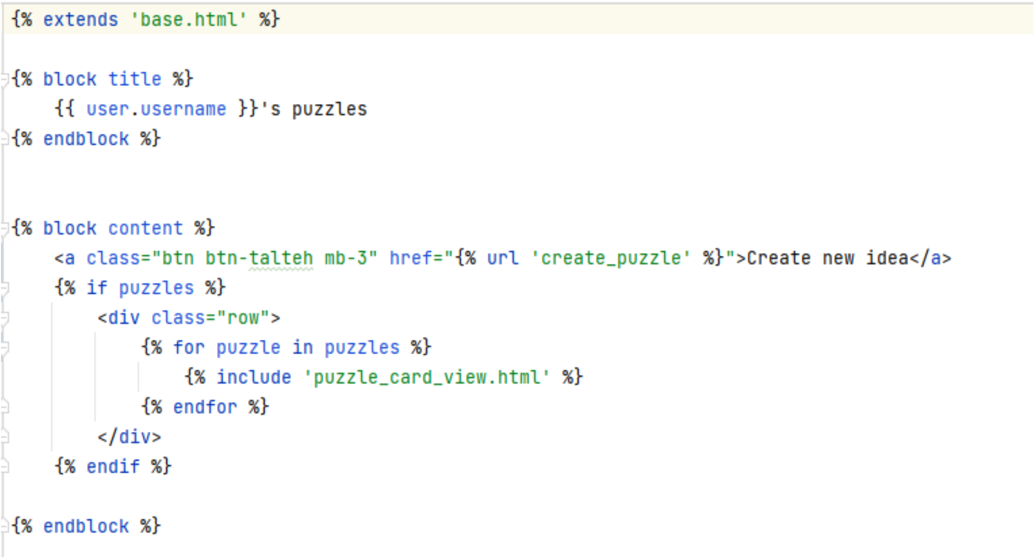


Figure 22 Front-end files structure

Also, from a programming point of view, it is not necessary to divide templates into separate folders, it is quite enough to store them all together. However, considering that this is a modular framework, it makes more sense to store templates for each application separately. All templates extend the base template, which contains the menu, as well as the declared CSS and bootstrap files. All files related to visual design, such as bootstrap and CSS files as well as the university logo are stored in the static folder and are also registered in the settings.py file.

3.2.1 Django template language

Any text document can be used as a template, for example html csv or xml [21]. Html-based templates were used during the creation of the web application.



```
{% extends 'base.html' %}

{% block title %}
    {{ user.username }}'s puzzles
{% endblock %}


{% block content %}
    <a class="btn btn-talteh mb-3" href="{% url 'create_puzzle' %}">Create new idea</a>
    {% if puzzles %}
        <div class="row">
            {% for puzzle in puzzles %}
                {% include 'puzzle_card_view.html' %}
            {% endfor %}
        </div>
    {% endif %}

{% endblock %}
```

Figure 23 Template example

As it can be seen in Figure 23, the template is an html document in which, in addition to html tags, there are django template language tags. These tags are used for variable insertion, simple iteration, and if methods. Also, the template language allows to include and extend other templates, this also allows to speed up the development process and comply with the DRY principle.

```
{% csrf_token %}
```

Figure 24 CSRF token

All forms are generated along with the CSRF token [22] and sent to the server along with the form submission; without this token, the template simply will not compile. Also, if somebody changes or intercepts this token, then the server will not execute the function that processes this form.

Also, the template language wraps all characters entered by the user, which may affect the html template, what makes the code protected from attacks both on the client side and on the server side.

3.2.2 Bootstrap and TalTech.css

When developing the application, the open source library Bootstrap 5 is used. The library is located directly in the static folder without using jsDelivr, which allows for faster file uploads. The use of version 5 can be explained by the fact that it is the newest and meets all the requests of the primary prototype. Also, many of the methods in version 4 are deprecated, and newer versions of npm do not compile it. Using old technologies to use old technologies diminishes the relevance of the application.

As it is bad practice to change bootstrap files directly in the source code, and it is not possible to recompile it with custom files as it was done in bootstarp 4, it was decided to use custom CSS files, which is loaded later than bootstrap, thereby rewriting some parameters, for example: color.

The colors used in the application correspond to the color palette specified in the TalTech design guidelines.

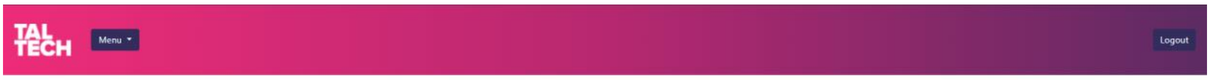


Figure 25 Topbar menu

However, the gradient used in the main menu follows the colors of the TalTech main site. The colors were matched using the "color picker" application from the open server panel toolkit.

3.3 Testing

This chapter covers testing the application. Tests are presented in the form of test story tables and they check all the main functions from flowchart. All tests are performed on a local system and a local database. Tests compiled in accordance with ISO 9126 quality standards.

3.3.1 Smoke testing

This type of testing is necessary for the initial prototype, as it will make it clear whether more detailed tests, such as tests with a customer, can be done. It will also help to identify critical errors that are undesirable during a demo with a customer.

Table 4 Test story 1

<table>
<tr><td><b>Test story id:</b></td><td>User_can_signup</td><td colspan="2"><b>Testable requirement:</b></td><td>Usability</td></tr>
<tr><td><b>Purpose of the test story:</b></td><td colspan="4">To check password and email validation; uniqueness of username. When registering a new user.</td></tr>
<tr><td><b>Prerequisites:</b></td><td colspan="4">• The user is not logged into the site.
• The user has an internet connection.</td></tr>
<tr><td><b>Activities:</b></td><td colspan="2"><b>Expected result:</b></td><td colspan="2"><b>Actual result:</b></td></tr>
<tr><td>The user clicks on the “sign up” button.</td><td colspan="2">The registration page opens.</td><td colspan="2">Result matches expectation.</td></tr>
<tr><td>The user enters his data correctly and click “Sign Up” button.</td><td colspan="2">The user is redirected to his profile.</td><td colspan="2">Result matches expectation.</td></tr>
<tr><td><b>Alternative action:</b></td><td colspan="2"><b>Expected result:</b></td><td colspan="2"><b>Actual result:</b></td></tr>
<tr><td>User enters an already registered username</td><td colspan="2">The user is notified that such a name already exists, the passwords are reset, and all the rest of the data remains.</td><td colspan="2">Result matches expectation.
Appendix 2-1</td></tr>
<tr><td>The user enters a light password, for example his username + 123</td><td colspan="2">Passwords are reset, the user is shown a notification that the password is too light or too similar to his other information</td><td colspan="2">Result matches expectation.
Appendix 2-2</td></tr>
</table>

User enters invalid email	The user is shown a message, the passwords have been reset.	Result matches expectation. Appendix 2-3
Environmental requirements:	Java script is running in browser	

In this test, the registration system was tested. It fully met expectations, the tests did not show any unexpected errors.

Table 5 Idea Creation

Test story id:	User_can_make_an_idea	**Testable requirement:**	Funcionality
Purpose of the test story:	The user can create and test an idea, and then change the answers.		
Prerequisites:	- The user is logged into the application. - The user has an internet connection.		
Activities:	**Expected result:**	**Actual result:**	
The user goes to the "my ideas" page and clicks the create a new idea button.	The idea creation page opens.	Result matches expectation. Appendix 2-4	
The user enters the name of the idea, selects the language and type of idea.	The user is redirected to idea testing page.	Result matches expectation. Appendix 2-5	
The user answered all the questions and	The user is redirected to the result page, where a diagram with his	Result matches expectation. Appendix 2-6	

clicked on the confirm button	average self-evaluation is generated	
The user has changed the answers and self-evaluation.	The user is redirected to a page with new data, and the diagram is re-created.	Result matches expectation.
Alternative action:	**Expected result:**	**Actual result:**
The user does not fill in all the fields and goes to another page / closes the application by pressing the confirm button.	After the discovery of testing the idea, the previous answers have been saved.	Result matches expectation.
The user left the answer blank, while editing the test.	The answer has not changed, the previous answer remains when opening the test.	Result matches expectation.
User only changed self-concept by leaving the same answer.	Self-esteem has changed, the diagram has been re-generated	Result matches expectation.
Environmental requirements:	Java script is running in browser	

This test verified the functionality of the idea testing system. The system acted as intended during the testing.

Table 6 Teacher test

Test story id:	Teacher_permisions_test	**Testable requirement:**	Funcionality
Purpose of the test story:	To check the permissions of the user who is in the teacher group.		
Prerequisites:	• The user is logged into the application. • The user is in the group teacher. • The user has an internet connection.		
Activities:	**Expected result:**	**Actual result:**	
The user visits the main page of the application	A teacher sees all ideas (including private ones) and has access to buttons for the detailed view of the idea and the export of the idea in docx	Result matches expectation. Appendix 2-7	
Alternative action:	**Expected result:**	**Actual result:**	
The user gave the teacher the right to edit his idea.	A teacher additionally sees the edit button	Result matches expectation.	
Environmental requirements:	Java script is running in browser		

This test proved that the prototype allows a teacher to view all ideas, as well as edit them if the student gave the teacher permission.

3.3.2 Customer testing

Tests with the customer were carried out in the form of an online demo about 2 times a month. The demo showed the progress over the past 2 weeks, as well as discussed further development and customer comments. A demo video was also recorded together with the customer, which will be demonstrated to TalTech center of IT-didactics before the initial thesis defence on May 20th 2021. A happy path test was carried out with the customer, and the customer was satisfied with the work of the application and the result in general.

4 Further developments

The current version of the application provides all necessary functions for creating and testing an idea by a student, as well as the ability to export a project for future development work. However, there are still several options for improving the application in the future.

One of the possible ways for development mentioned earlier is integration with the university database to log in via Uni-ID and automatically create a student account by analogy with Moodle and ÕIS. Also, it would allow to create a profile for teachers automatically and add the necessary permissions to them from the university database.

It is possible to create a separate user interface using this project only as a server part if the university wishes to proceed with the project. As a result of the research and practical work conducted in the framework of this thesis, it is recommended to use either React.js or Vue.js.

4.1 Deployment

It is necessary to prepare the application itself and the server for it to deploy an application to an external server. The instructions for deploying the application in the future can be provided in the result of this thesis.

4.1.1 Application side

There is a security key parameter (Figure 26) in the settings.py file. This key is used to generate a CSRF token, so it is critical that this parameter is not in the source code. It should either be requested from another server or stored in a separate read-only file.

```
# SECURITY WARNING: keep the secret key used in production secret!
SECRET_KEY = 'en$=ftbt5c7@e8i280h1&4)$@u*wl81%2vp$xicwqrdgnb0(*w'
```

Figure 26 Security key

This key should be regenerated, since the one that is indicated in the work was used only at the prototype stage.

Another parameter that needs to be changed is the DEBUG. This is a Boolean value that is responsible for the fact that if a critical error occurs during the application running, then the user sees all the logs of this error, including all keys, and parts of the source code where the error was.

```
# SECURITY WARNING: don't run with debug turned on in production!
DEBUG = True
```

Figure 27 Debug oprion

The value of this parameter should be **False**.

“When DEBUG = False, Django does not work at all without a suitable value for ALLOWED_HOSTS. This parameter is also needed to protect against CSRF attacks. For example, on nginx might setup a default server to return “444 No Response” on an unrecognized host” it can be seen in Figure 28.

```
server {
listen 80 default_server;
return 444;
}
```

Figure 28 Example from deployment documentation

It is also possible to get a list of all the settings that need to be done using the command (Figure 29).

```
python3 manage.py check –deploy
```

Figure 29 Command to get checklist for deploy

4.1.2 Server side

The django documentation [23] specifies the following web servers:

1. Gunicorn
2. uWSGI
3. Apache
4. Dephne
5. Hypercorn
6. Uvicorn
7. Nginx

The choice of a web server remains with the university. The operating system can be either Linux or Windows.

According to the author's estimates, stable operation requires a 2-core processor with a frequency of 2.4 GHz, 2-4 gigabytes of RAM and an SSD with a volume of 20 gigabytes or more. These parameters should be enough for about 600-700 unique users per day.

4.2 Roadmap

A roadmap is needed to roughly represent plans for future development. As it can be seen in the Figure 30, there are 4 phases of application development.

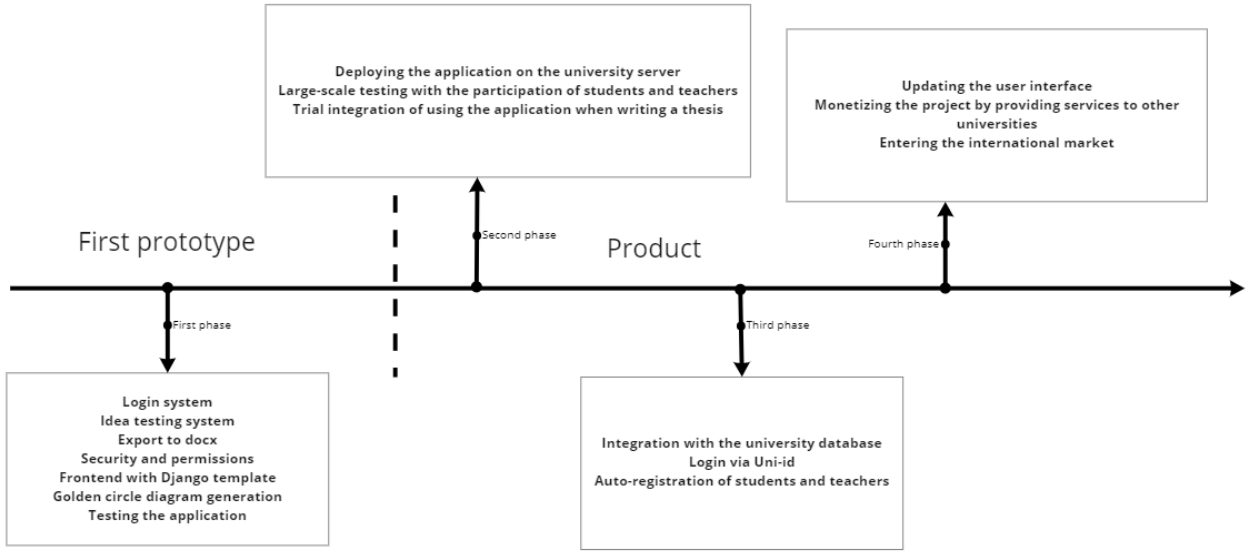

Deploying the application on the university server
Large-scale testing with the participation of students and teachers
Trial integration of using the application when writing a thesis
Updating the user interface
Monetizing the project by providing services to other universities
Entering the international market
First prototype
Second phase
Product
Fourth phase
First phase
Third phase
Login system
Idea testing system
Export to docx
Security and permissions
Frontend with Django template
Golden circle diagram generation
Testing the application
Integration with the university database
Login via Uni-id
Auto-registration of students and teachers

Figure 30 Roadmap

Roadmap can also be seen in Appendix 3 in more details.

In this work, the development process for the first phase has been described. This phase is the creation of a prototype with all the necessary functionality at the initial stage.

The second phase is to deploy the application on the university server and conduct large-scale testing among students and teachers. As well as trial use of the application in writing real theses.

In the third phase, it is planned to complete the application and integrate it with the university database. And it is already possible to use the application in full in the process of writing theses. Furthermore, suitable access for people with disabilities is provided on the third phase. [24]

In the fourth phase, it is suggested to update the user interface. Moreover, the university can monetize the application by providing services to other universities, or even gymnasiums for writing research work. It can grant licenses to universities in other countries or sell private licenses for students.

The application can be easily extended and any new functions and services that are required in the future can be easily added due to the modularity of the application.

New name for the application has to be suggested by TalTech center of IT-didactics.

5 Summary

As a result of the research and practical work conducted in this thesis, weak sides of existing solutions were found, and the prototype was improved in accordance with revealed problems of other similar solutions. Moreover, the prototype was tested, and its ability to effectively solve the problem posed in the beginning was proven.

The problem of the student's presentation of ideas for the thesis was considered in this thesis. The problem of wrong approach of students to the problem statement for a thesis appeared to be relevant, because teachers had to receive a large number of thoughtless ideas, spend more time interviewing a student to clarify issues that a student should prepare in advance.

Proposed solution allows a student firstly to conduct a research on the relevance of his idea, the availability of materials and other important issues that many students do not think about in advance. In turn, the teacher will receive a ready-made result of testing the student’s idea, where a student's self-assessment for each question will also be seen in addition to answering questions about the idea. This will allow teacher to give an answer regarding supervising quickly, give advice on improving the idea and avoid wasting time on unnecessary conversations. The student is more serious about writing a thesis by testing ideas, and in order to compose a successful project of the thesis, he does his own personal research of the issue. Furthermore, a student can export project of a thesis and do not double work by answering similar questions.

The presented solution is suitable for TalTech as it will be completely under the control of the university and does not require any payment of an annual license. Third-party solutions lack important functions, do not guarantee data protection, and cannot be customized by the university. Besides solving a problem, the project also can be profitable for the university in general both monetary and reputationally. This solution is expected to improve the overall quality of TalTech theses. As a result, the research contribution of the university to the scientific community and state of the art will grow and its status in the world ranking [25] will also be increased.

References

[1] R. Morais, “Digital research design,” *International Association of Universities Horizons*, vol. 23, no. 1, pp. 31-32, May 2018.

[2] S. Sinek, *How great leaders inspire action | Simon Sinek*, TED, 2010.

[3] R. Babak, “Concept of Golden Circle -By Simon Sinek,” 1 February 2016. [Online]. Available: https://www.linkedin.com/pulse/concept-golden-circle-by-simon-sinek-babak-mohammadi/. [Accessed May 2021].

[4] The University of Sheffield, “How to: Write a Problem Statement,” [Online]. Available: https://www.sheffield.ac.uk/polopoly_fs/1.440722!/file/HowtoWriteaProblemStatement.pdf. [Accessed May 2021].

[5] R. Morais, “Introduction to the software,” 1 May 2021. [Online]. Available: https://www.ideapuzzle.com/index.php?id=213.

[6] R. Morais, 24 May 2016. [Online]. Available: https://www.ideapuzzle.com/downloads/file138_pt.pdf. [Accessed May 2021].

[7] M. Gouvela, “Inroduction to the software,” 2 April 2018. [Online]. Available: https://www.ideapuzzle.com/downloads/file155_pt.pdf. [Accessed May 2021].

[8] Miro, “Pricing,” Miro, [Online]. Available: https://miro.com/pricing/. [Accessed May 2021].

[9] Tallinna Tehnikaülikool, “LOGOD, SLAIDID,” [Online]. Available: https://www.taltech.ee/logod. [Accessed May 2021].

[10] Python Software Foundation, “General Python FAQ,” 12 May 2021. [Online]. Available: https://docs.python.org/3/faq/general.html#what-is-python. [Accessed 12 May 2021].

[11] Stack overflow, “2020 Developer Survay,” Stack Overflow, February 2020. [Online]. Available: https://insights.stackoverflow.com/survey/2020#developer-profile--survey-respondents. [Accessed May 2021].

[12] Django Software Foundation, “Security in Django,” [Online]. Available: https://docs.djangoproject.com/en/3.2/topics/security/. [Accessed May 2021].

[13] Django Software Foundation, “Making queries,” [Online]. Available: https://docs.djangoproject.com/en/3.2/topics/db/queries/. [Accessed May 2021].

[14] JetBrains s.r.o., “Python Developers Survey 2020 Results,” October 2020. [Online]. Available: https://www.jetbrains.com/lp/python-developers-survey-2020/. [Accessed May 2021].

[15] OSPanel.io, [Online]. Available: https://ospanel.io/. [Accessed May 2021].

[16] Oracle, “Free Oracle Database for Everyone,” [Online]. Available: https://www.oracle.com/database/technologies/appdev/xe.html. [Accessed May 2021].

[17] Django Software Foundation, “django-admin and manage.py,” [Online]. Available: https://docs.djangoproject.com/en/3.2/ref/django-admin/. [Accessed May 2021].

[18] JavaTpoint, “Django MVT,” [Online]. Available: https://www.javatpoint.com/django-mvt. [Accessed May 2021].

[19] nishkarsh146, “Difference between MVC and MVT design patterns,” 20 May 2020. [Online]. Available: https://www.geeksforgeeks.org/difference-between-mvc-and-mvt-design-patterns/. [Accessed May 2021].

[20] Django Software Foundation, “Models,” [Online]. Available: https://docs.djangoproject.com/en/3.2/topics/db/models/. [Accessed May 2021].

[21] Django Software Foundation, “The Django template language,” [Online]. [Accessed May 2021].

[22] Django Software Foundation, “Cross Site Request Forgery protection,” [Online]. Available: https://docs.djangoproject.com/en/3.2/ref/csrf/. [Accessed May 2021].

[23] Django Software Foundation, “Deploying Django,” [Online]. Available: https://docs.djangoproject.com/en/3.2/howto/deployment/. [Accessed May 2021].

[24] M. C. L. G. R. G. V. Ben Caldwell, “Web Content Accessibility Guidelines (WCAG) 2.0,” [Online]. Available: https://www.w3.org/TR/WCAG20/. [Accessed May 2021].

[25] QS Quacquarelli Symonds Limited, “QS World University Rankings,” [Online]. Available: https://www.topuniversities.com/universities/tallinn-university-technology-taltech/undergrad. [Accessed May 2021].

[26] Django Software Foundation, “Deployment checklist,” [Online]. Available: https://docs.djangoproject.com/en/3.2/howto/deployment/checklist/. [Accessed May 2021].

Appendix 1 – Non-exclusive licence for reproduction and publication of a graduation thesis

I Aleksandr Kezerev

1. Grant Tallinn University of Technology free licence (non-exclusive licence) for my thesis " Web Application for Theses Projects Creation", supervised by Nadežda Furs

   1.1. to be reproduced for the purposes of preservation and electronic publication of the graduation thesis, incl. to be entered in the digital collection of the library of Tallinn University of Technology until expiry of the term of copyright;

   1.2. to be published via the web of Tallinn University of Technology, incl. to be entered in the digital collection of the library of Tallinn University of Technology until expiry of the term of copyright.

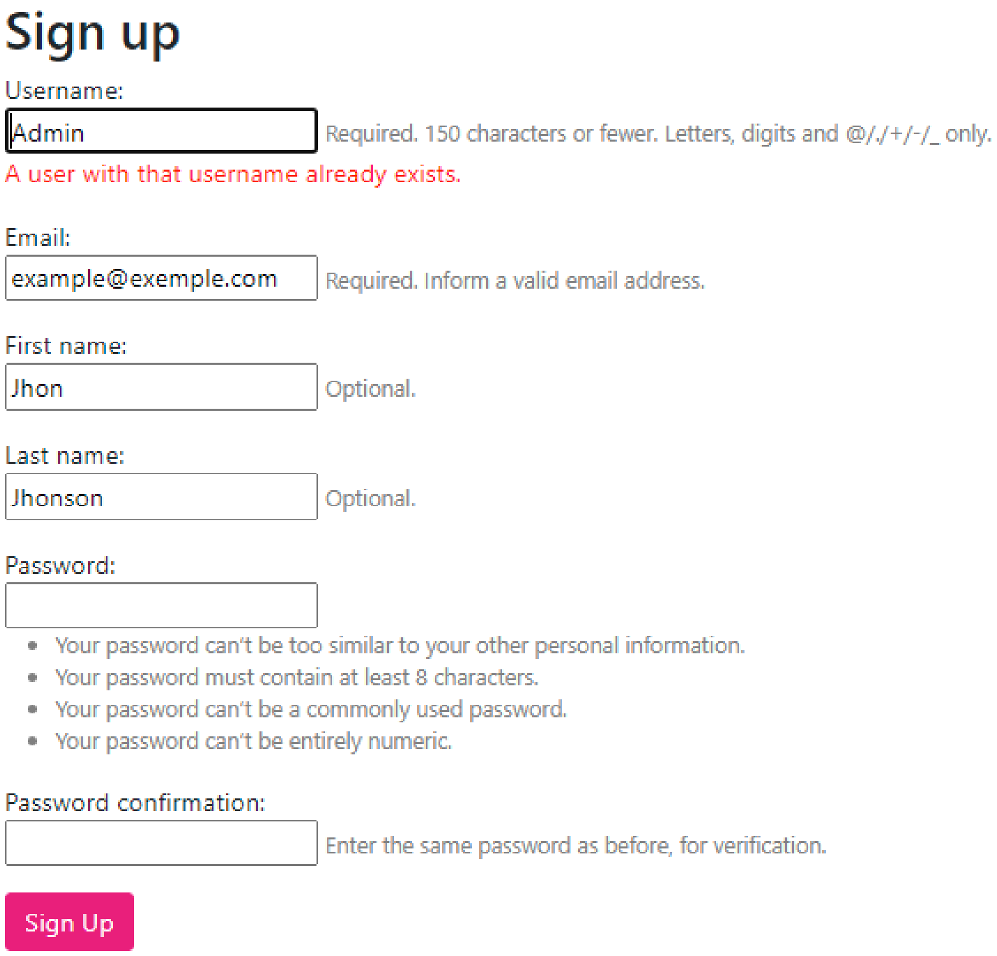
2. I am aware that the author also retains the rights specified in clause 1 of the nonexclusive licence.

3. I confirm that granting the non-exclusive licence does not infringe other persons' intellectual property rights, the rights arising from the Personal Data Protection Act or rights arising from other legislation.

15.05.2021

Appendix 2 – Test results

1) Same username error



Sign up

Username: Admin Required. 150 characters or fewer. Letters, digits and @/./+/-/_ only.

A user with that username already exists.

Email: example@exemple.com Required. Inform a valid email address.

First name: Jhon Optional.

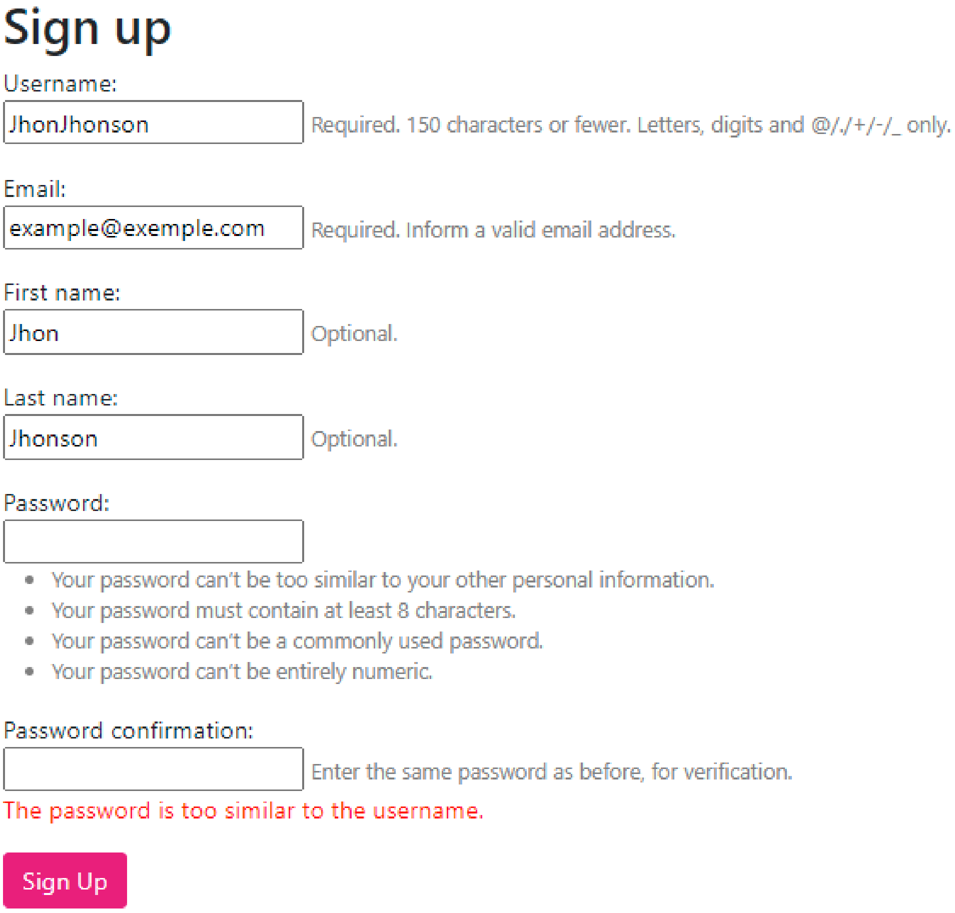
Last name: Jhonson Optional.

Password:

- Your password can't be too similar to your other personal information.
- Your password must contain at least 8 characters.
- Your password can't be a commonly used password.
- Your password can't be entirely numeric.

Password confirmation: Enter the same password as before, for verification.

2) Too easy password



Sign up

Username:

JhonJhonson

Required. 150 characters or fewer. Letters, digits and @/./+/-/_ only.

Email:

example@exemple.com

Required. Inform a valid email address.

First name:

Jhon

Optional.

Last name:

Jhonson

Optional.

Password:

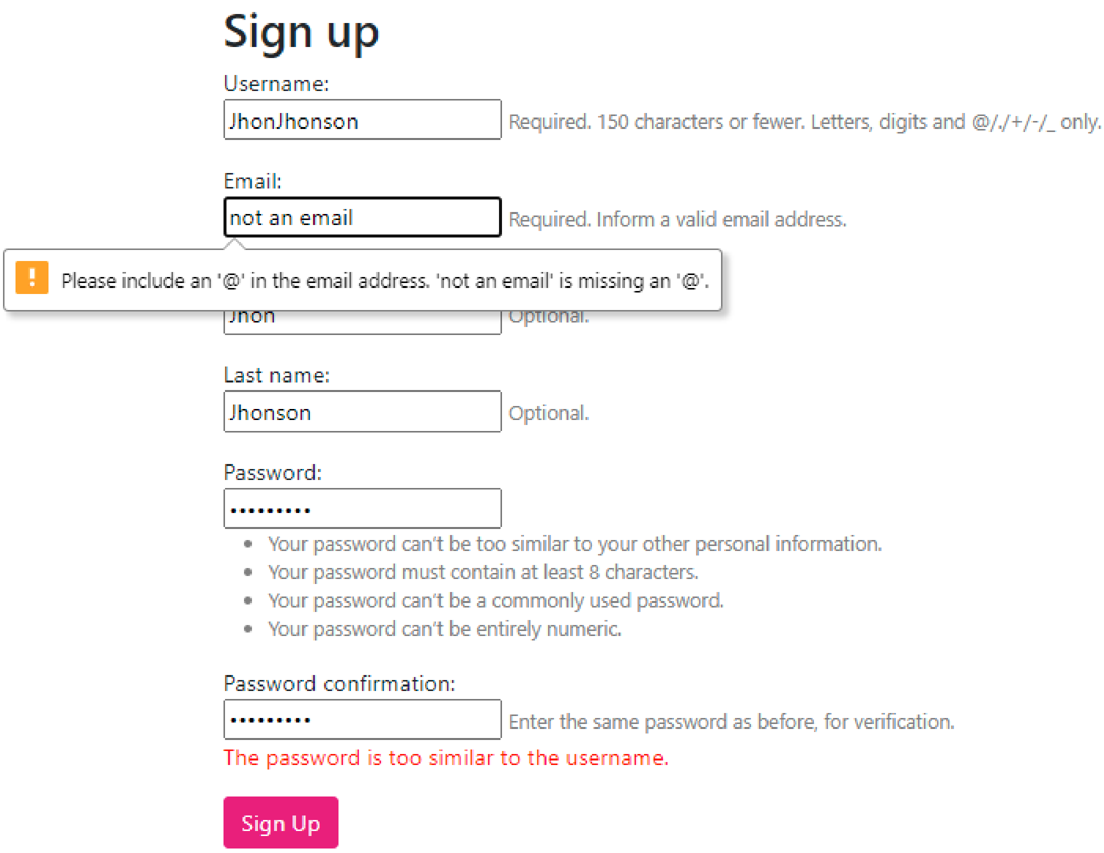
- Your password can't be too similar to your other personal information.
- Your password must contain at least 8 characters.
- Your password can't be a commonly used password.
- Your password can't be entirely numeric.

Password confirmation:

Enter the same password as before, for verification.

The password is too similar to the username.

3) Non valid email



Sign up

Username:

JhonJhonson

Required. 150 characters or fewer. Letters, digits and @/./+/-/_ only.

Email:

not an email

Required. Inform a valid email address.

Please include an '@' in the email address. 'not an email' is missing an '@'.

Jhon

Optional.

Last name:

Jhonson

Optional.

Password:

•••••••••

- Your password can't be too similar to your other personal information.
- Your password must contain at least 8 characters.
- Your password can't be a commonly used password.
- Your password can't be entirely numeric.

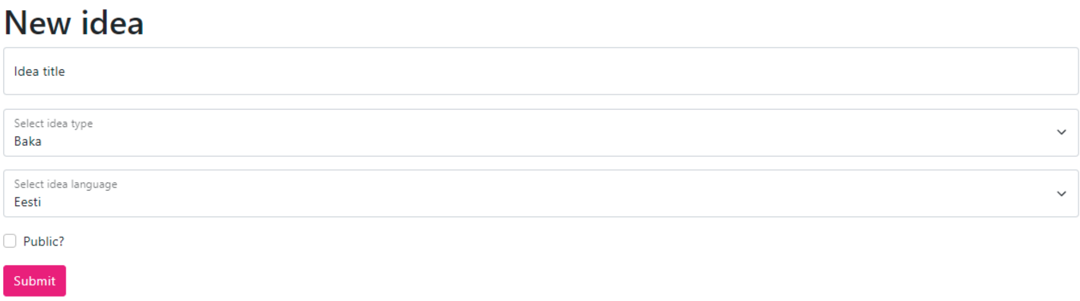
Password confirmation:

•••••••••

Enter the same password as before, for verification.

The password is too similar to the username.

4) New idea creation



New idea

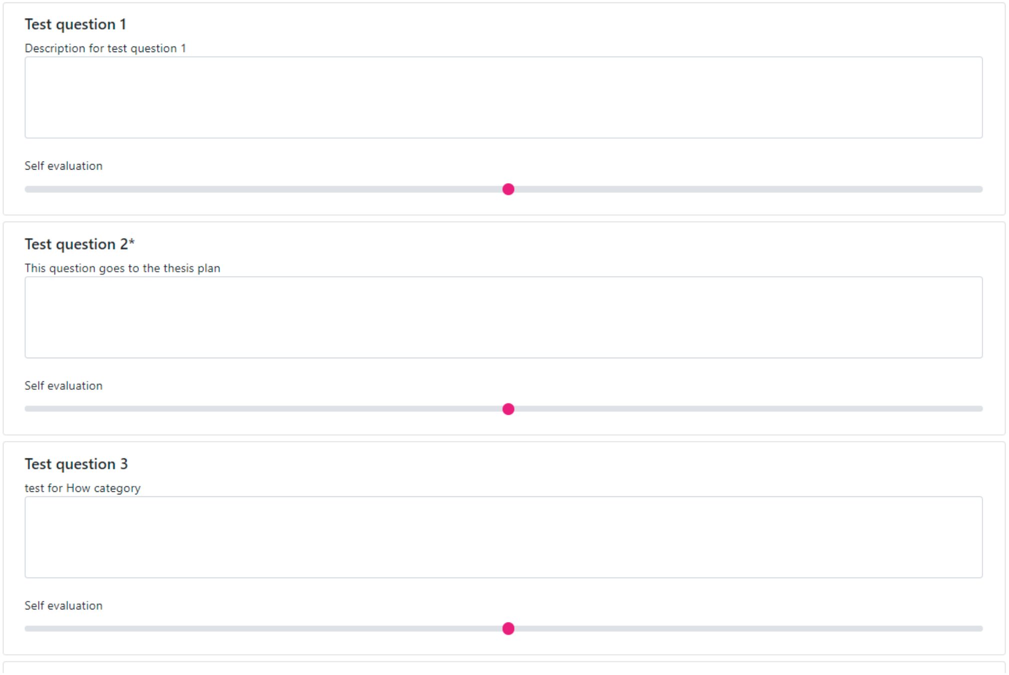
Idea title

Select idea type
Baka

Select idea language
Eesti

□ Public?

5) Testing process



Test question 1

Description for test question 1

Self evaluation

Test question 2*

This question goes to the thesis plan

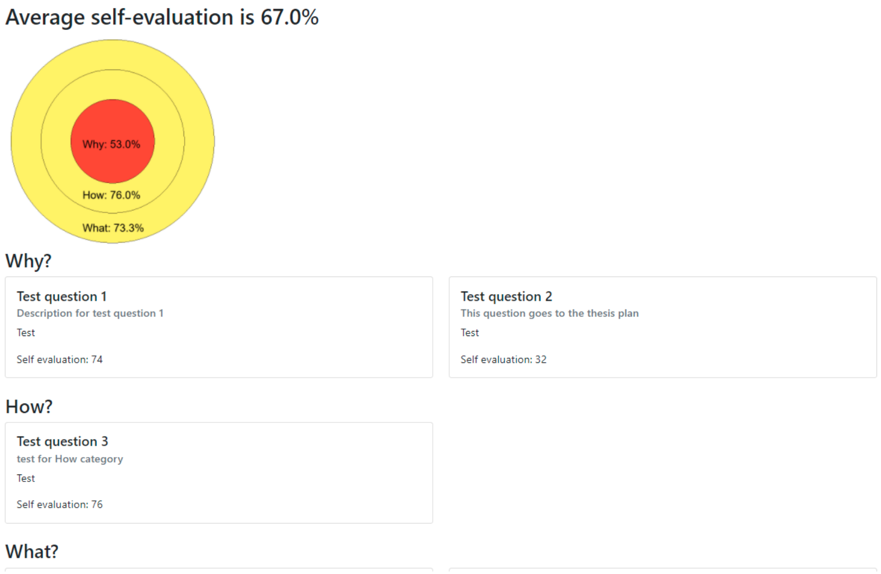
Self evaluation

Test question 3

test for How category

Self evaluation

6) Result of idea testing



Average self-evaluation is 67.0%

Why: 53.0%

How: 76.0%

What: 73.3%

Why?

Test question 1

Description for test question 1

Test

Self evaluation: 74

Test question 2

This question goes to the thesis plan

Test

Self evaluation: 32

How?

Test question 3

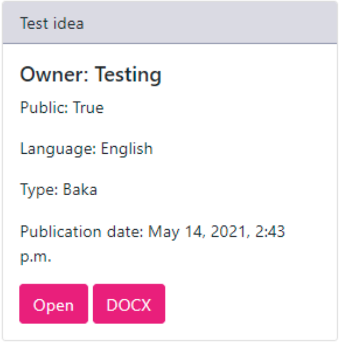
test for How category

Test

Self evaluation: 76

What?

7) Teacher can see privat works


Test idea

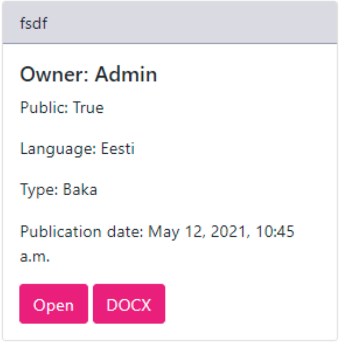
Owner: Testing

Public: True

Language: English

Type: Baka

Publication date: May 14, 2021, 2:43 p.m.


fsdf

Owner: Admin

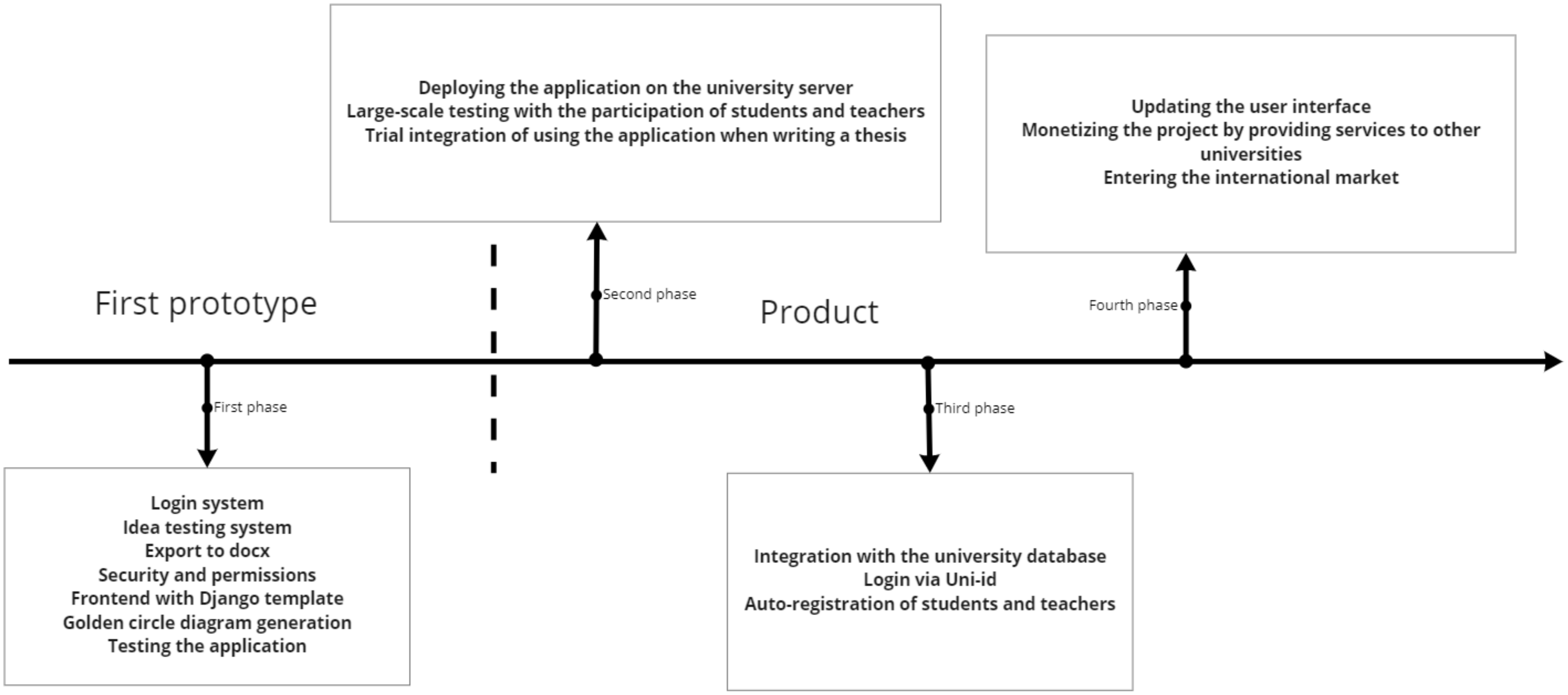
Public: True

Language: Eesti

Type: Baka

Publication date: May 12, 2021, 10:45 a.m.

Appendix 3 – Solution roadmap



Login system
Idea testing system
Export to docx
Security and permissions
Frontend with Django template
Golden circle diagram generation
Testing the application

Deploying the application on the university server
Large-scale testing with the participation of students and teachers
Trial integration of using the application when writing a thesis

Integration with the university database
Login via Uni-id
Auto-registration of students and teachers

Updating the user interface
Monetizing the project by providing services to other universities
Entering the international market