



**KAUNO TECHNOLOGIJOS UNIVERSITETAS**  
**INFORMATIKOS FAKULTETAS**

**Today I Learned (TIL)**

Projektas

**Atliko:**

IFF-7/13 grup. stud.

Tautvydas Dikšas

**Dėstytojas**

Doc. dr. Sigitas Drąsutis

## Turinio lentelė

Terminų ir santraukų žodynas .....	3
Įvadas .....	4
1. Analizė .....	5
1.1. Esamų sprendimų analizė (pavyzdys).....	5
1.1.1. Reddit, Today I Learned (TIL) .....	5
1.1.2. Today I Found Out.....	6
1.1.3. Esamų sistemų palyginimas (pavyzdys) .....	7
1.1. Technologijų analizė.....	7
2. Projektas .....	8
2.1. Funkciniai reikalavimai sistemai .....	8
2.2. Nefunkciniai reikalavimai sistemai .....	8
2.3. Duomenų bazės projektas .....	9
2.4. UML diagramos.....	10
2.4.1. Panaudos atvejų diagrama .....	10
2.4.2. Klasių diagrama (veiklos diagramos).....	11
3. Vartotojo vadovas .....	17
3.1. Diegimas .....	17
3.2. Sistemos veikimas .....	17
4. Išvados .....	21
5. Literatūros sąrašas.....	22
6. Priedas .....	23
6.1. Semestro darbų suvestinė.....	23

## Terminų ir santraukų žodynas

<b>HTML</b>	Hiperteksto žymėjimo kalba – kompiuterinė žymėjimo kalba, naudojama pateikti turinį internete (angl. <i>Hypertext Markup Language</i> ).
<b>Symfony</b>	Tai tinklapių kūrimo sistema, kuri naudoja PHP bibliotekas.
<b>CSS</b>	Puslapių stilių rašymų kalba, kuri skirta HTML suteikti estetinį vaizdą
<b>PHP</b>	Plačiai naudojama programavimo kalba, kuri interpretuojama ir galima įtraukti į HTML. Pavadinimas yra rekursyvus akronimas (angl. <i>PHP: Hypertext Preprocessor</i> ).
<b>Doctrine</b>	Tai PHP bibliotekų visuma, kuri priskiria objektų ryšius ir sukuria duombazę. Užklauskos rašomos DQL, t.y. objektyvi SQL versija.
<b>Twig</b>	Lankstus, greitas ir saugus šablono variklis PHP programavimo kalbai.
<b>Composer</b>	PHP programavimo kalbos projekto paketų valdymo sistema. Naudojama valdant projekto priklausomybes ir bibliotekas.
<b>git</b>	Projektų versijų kontroliavimo sistema.
<b>Github</b>	Sistema naudojama projektams saugoti debesyse, glaudžiai veikia su git.
<b>MagicDraw</b>	Programa naudojama UML diagramų atvaizdavimui.
<b>UML</b>	(angl. <i>Unified Modeling Language</i> ) Tai kalba, kurios tikslas leisti klientui ir programuotojui aiškiau komunikuoti. Taip užtikrinamas darbo efektyvumas, nes viskas atliekama pagal užsakovo reikalavimus.
<b>UI</b>	(angl. <i>User Interface</i> ) Naudotojo sąsaja. Tai yra viskas, ką mato sistemoje žmogus ir gali sąveikauti.
<b>JavaScript</b>	Programavimo kalba, plačiai naudojama internete su HTML ir CSS. Ji leidžia lengviau valdyti naudotojo atliekamus veiksmus.
<b>HTML</b>	(angl. <i>HyperText Markup Language</i> ) Tai tinklapiui suteikianti struktūrą kalba, artimai naudojama su CSS.

## **Įvadas**

Praleisdamas daug laiko internete pastebėjau, kad tinklapiuose, kur ateina didelis vartotojų srautas, straipsniai neturi jokios struktūros, įtraukiančios ir išlaikančios dėmesį. Taip pat tekstai parašyti buitiskai. Šiuo darbu noriu pasiekti, kad žmonės galėtų rašyti mokslinius straipsnius, bet laikydamiesi reikalavimų, t.y. turėti santykį tarp informacijos vizualizavimo, nuoseklaus teksto, praktinio panaudojimo ir pan. Sistemoje apsilankantys žmonės galės kokybiškai skleisti ir gauti informaciją.

**Darbo tikslas** – pasirinkta tema realizuoti sistemą, kuri suteiktų žmonėms naudingos edukacinės informacijos ir būtų nemokama.

### **Darbo uždaviniai:**

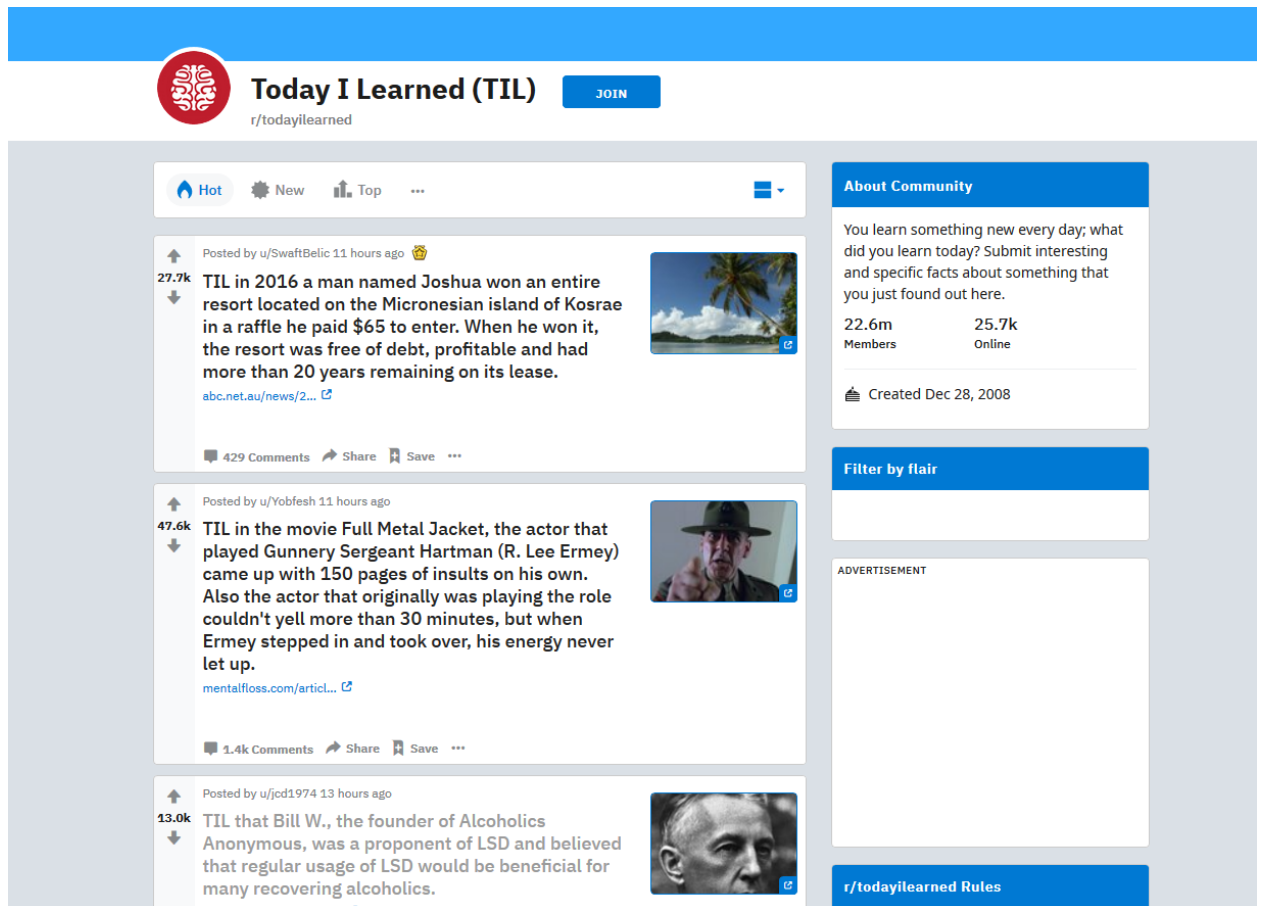
1. Išanalizuoti Reddit Today I Learned (TIL) ir Today I Found Out sprendimus.
2. Išanalizuoti Symfony ir Doctrine technologijas.
3. Suprojektuoti sistemą, iškelti funkcinius ir nefunkcinius reikalavimus, sukurti duomenų bazių naudojant Doctrine.
4. Realizuoti sistemą, ištestuoti ją, patikrinti, kaip tiksliai atliktas darbas pagal reikalavimus.

# 1. Analizė

## 1.1. Esamų sprendimų analizė (pavyzdys)

### 1.1.1. Reddit, Today I Learned (TIL)

Tinklapis Reddit žinomas kaip interneto veidas, kur žmonės generuoja tinklapių turinį. Kaip kuriamo tinklapių pavyzdį pateiksiu subredditą TIL (žr. Pav. 1.1)



Pav. 1.1 Žmonių sugeneruota informacijos platforma Reddit, subreddit TIL

#### Sistemos privalumai:

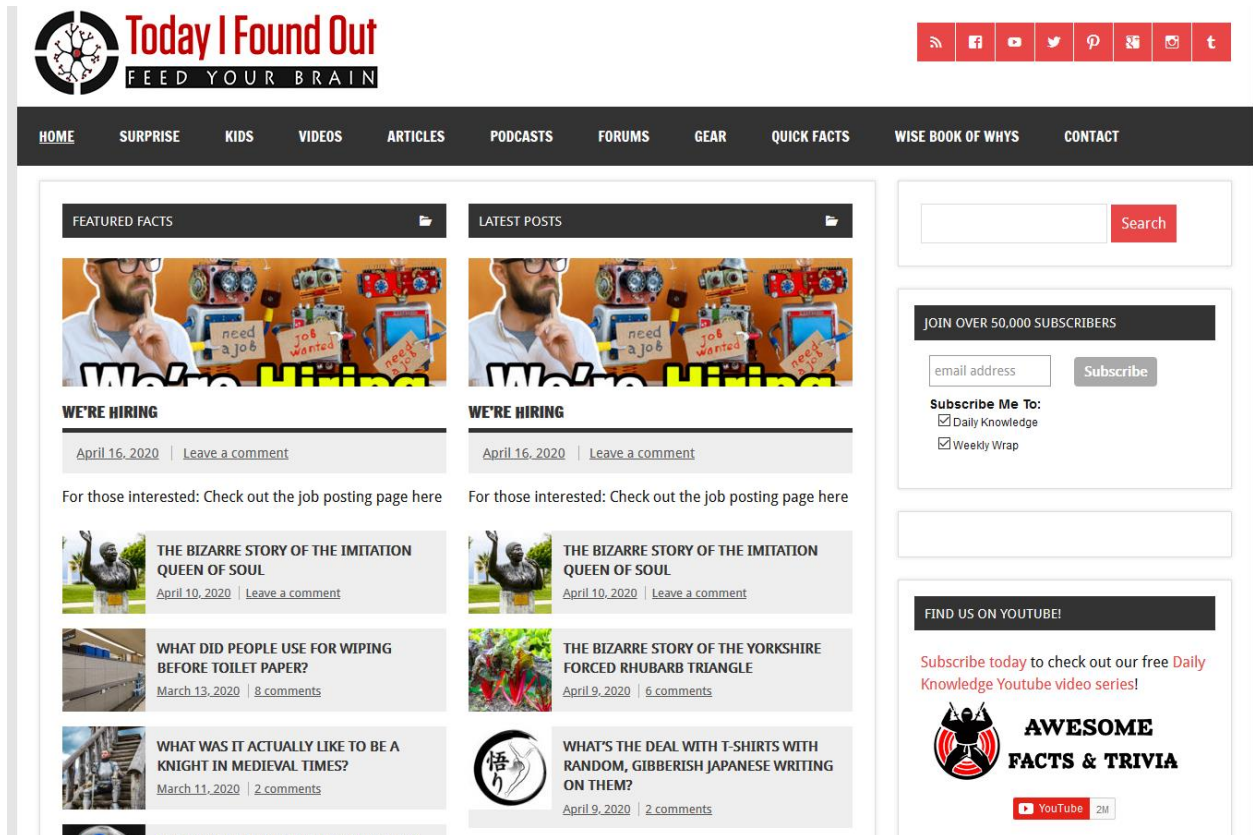
- Įdomi informacija;
- Įrašai neišmėtyti per visą puslapį, t.y. sudėlioti sąrašo būdu;
- Lengva naviguoti;
- Turinys nemokamas.

#### Sistemos trūkumai:

- Nėra turinio unikalios struktūros;
- Visą informaciją galima perskaityti tik nuėjus į šaltinį, t.y. nuorodą;
- Nėra faktų patikrinimo.

### 1.1.2. Today I Found Out

Tinklapyje rašomi unikalūs straipsniai, kurie turi naudotojui atskleisti naujų nežinomų faktų (žr. Pav. 1.1).



Pav. 1.2 Today I Found Out informacijos saugojimo sistema

#### Sistemos privalumai:

- Ne perteklinis dizainas;
- Turi forumą diskusijoms.

#### Sistemos trūkumai:

- Nenaudinga informacija;
- Turinys neatkreipia dėmesio;
- Nuobodus dizainas;
- Per daug išmėtyta informacija;
- Negalima vertinti turinio;
- Negali bet koks žmogus parašyti teksto.

### 1.1.3. Esamų sistemų palyginimas (pavyzdys)

Kriterijus	Today I Learned (TIL)	Today I Found Out	Kuriama sistema
Informacijos įvedimas ir redagavimas	Informacija skirstoma įrašais arba nuoroda į šaltinio puslapį	Informacija skirstoma įrašais arba puslapiais	Bus galima informaciją skirstyti į pastraipas, puslapius ir paveiksliukų galerijas
Kategorijų priskyrimas	Tik pagal populiarumą ir datą	Tik pagal datą	Pagal datą, populiarumą, specialybę, aktualumą
Informacijos pateikimas	Didelis nuoseklus tekstas	Nuoroda į šaltinį	Palaikomas santykis tarp nuoseklaus teksto, vizualizacijos ir praktinių uždavinių
Turinio priežiūra	Priežiūri subreddito moderatoriai	Priežiūri tinklapio kūrėjai	Viešą turinį priežiūri moderatoriai, asmeninį – teksto kūrėjas
Turinį įveda	Bet koks prisijungęs žmogus	Tik tinklapio administracija	Bet koks prisijungęs žmogus

### 1.1. Technologijų analizė

Projektas kaip pagrindą naudoja Symfony. Kūrimui bus naudojami tokie griaučių įrankiai kaip doctrine, twig.

#### Composer:

Tai įrankis, kuriuo galima sukurti PHP projektą ir nurodyti jo priklausomybes bei bibliotekas.

#### Symfony:

Tai griaučiai, kuriais kuriami tinklapiai. Jis nemokamas, greitas ir patogus, turi didelį bibliotekų bei įrankių pasirinkimą profesionaliai atrodančio interneto portalo kūrimui. Symfony skatina jo naudotojus programuoti objektyviau (angl. *Object-oriented*). Kuriant duomenų bazę naudojamas įrankis doctrine. Taip pat šie griaučiai turi funkciją paleisti gyvą serverį neįsirašant papildomų programų.

#### Doctrine:

Pasirinkimas gali būti SQL, bet kadangi Doctrine įrašomas kartu su Symfony, privaloma šiuo įrankiu sukurti duombazę. Užklauskos imamos naudojant DQL, t.y. objektams orientuotas SQL.

#### Twig:

Rašant tinklapio kodui su HTML naudojamas twig, t.y. šablono sistema, kuri padeda įvesti turinį universaliai ir programiškai suteikia skaitomumo. Iš karto įrašomas su Symfony projektu.

## **2. Projektas**

### **2.1. Funkciniai reikalavimai sistemai**

- Registruotis;
- Prisijungti;
- Atsijungti;
- Redaguoti profilio informaciją;
- Atkurti profilio slaptažodį;
- Peržiūrėti straipsnių sąrašą;
- Kurti straipsnio įrašą;
- Trinti straipsnio įrašą;
- Redaguoti straipsnio įrašą;
- Peržiūrėti straipsnio įrašą;
- Komentuoti po straipsniu;
- Peržiūrėti komentarus;
- Vertinti straipsnį;
- Peržiūrėti paskyros informaciją.

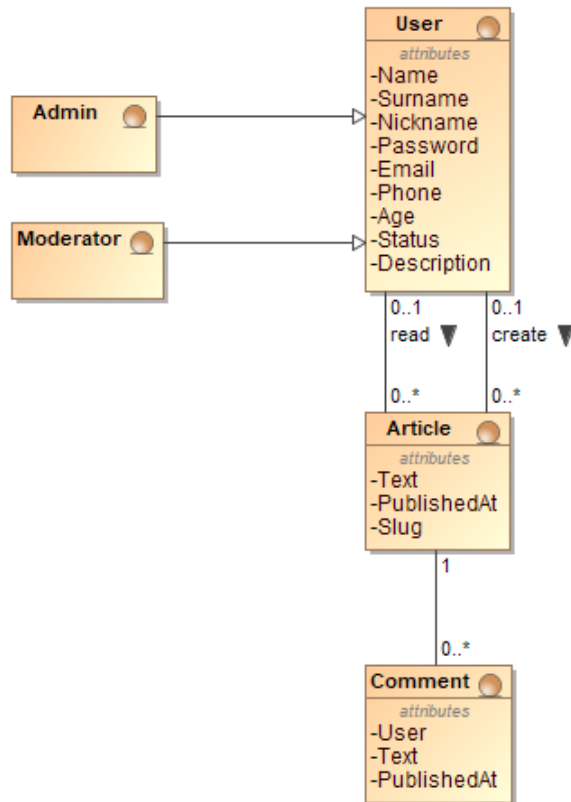
### **2.2. Nefunkciniai reikalavimai sistemai**

- Paskyrų slaptažodžiai ir slapti klausimų atsakymai šifruojami;
- Paskyros patvirtinimui naudojamas autentifikatorius;
- Sistemoje naudojama anglų kalba;
- Sistemoje rodys laiką pagal nustatytą kompiuterio lokalę;
- Sistema veikia ant visų OS;
- Naudotojo sąsaja dinamiškai keičiasi kintant ekrano dydžiui.



### 2.3. Duomenų bazės projektas

Šiam projektui duomenų bazė yra mažesnė, nes pagrindinė funkcija žiūrėti straipsnius (žr. Pav. 2.1). Naudotojas gali kurti daug straipsnių, jei nenori, gali ir nekurti. Žinoma, straipsnis gali būti perskaitomas naudotojų, o naudotojas gali perskaityti daug straipsnių. Kiekvienas komentaras privalo priklausyti tik vienam straipsniui. Straipsnis gali turėti daug arba neturėti komentarų.

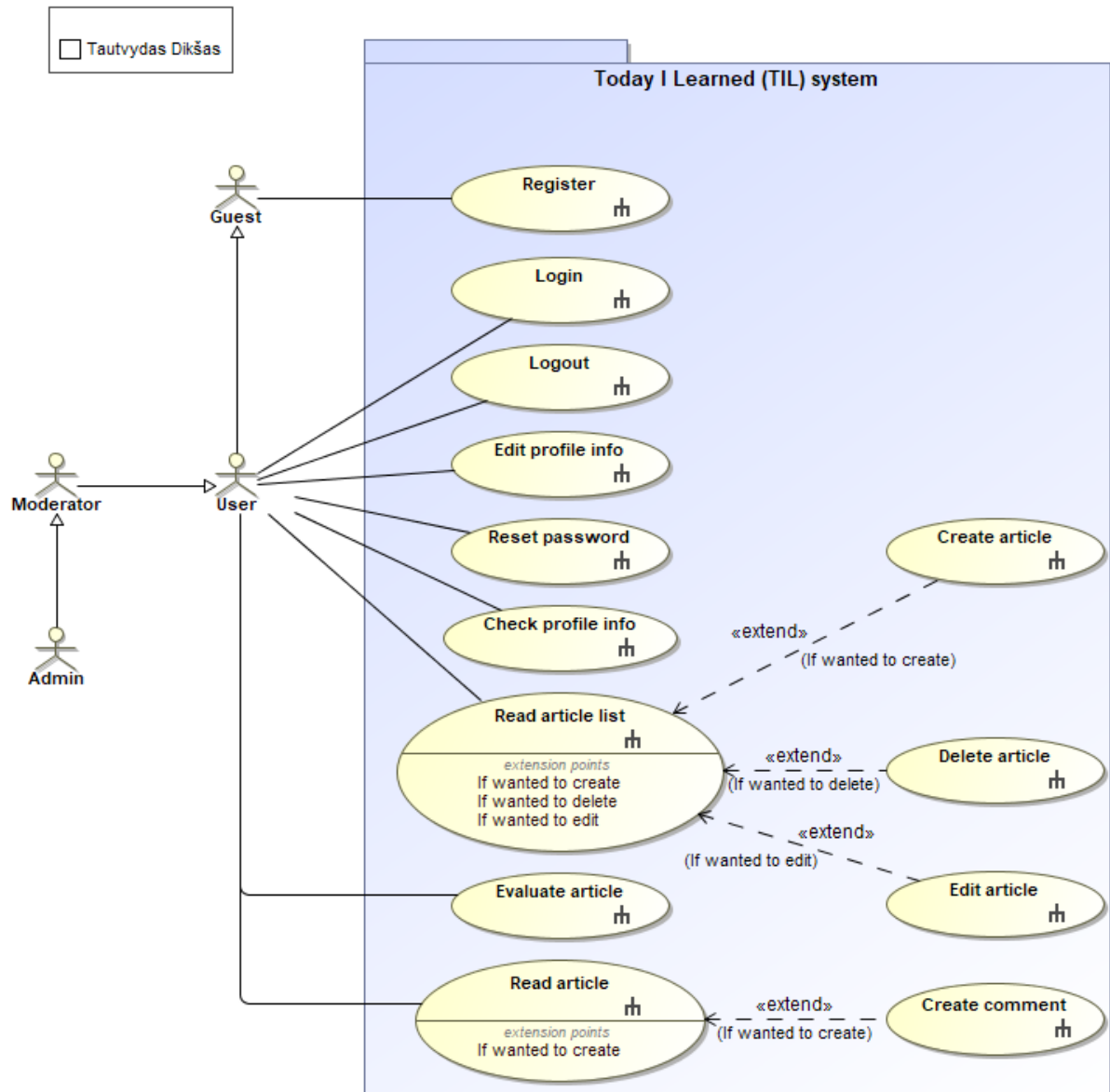


Pav. 2.1 Klasių diagrama

## 2.4. UML diagramos

### 2.4.1. Panaudos atvejų diagrama

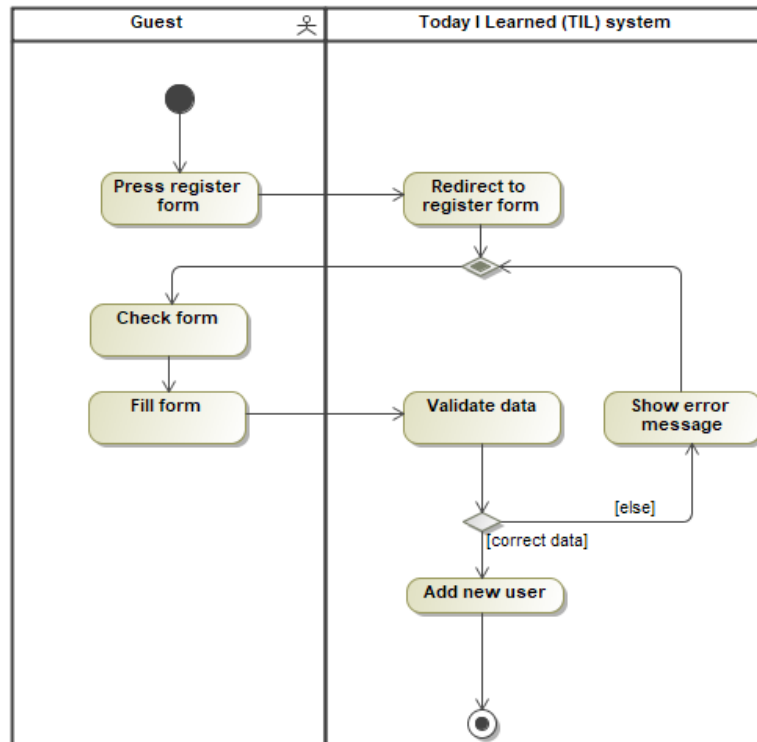
Darbo metu buvo sugalvoti funkciniai reikalavimai. Tada galima pradėti sistemos projektavimą. Pirmoji diagrama realizuota yra panaudojimo atvejų (PA) (žr. Pav. 2.2). Atlikta naudojant „MagicDraw“ įrankį.



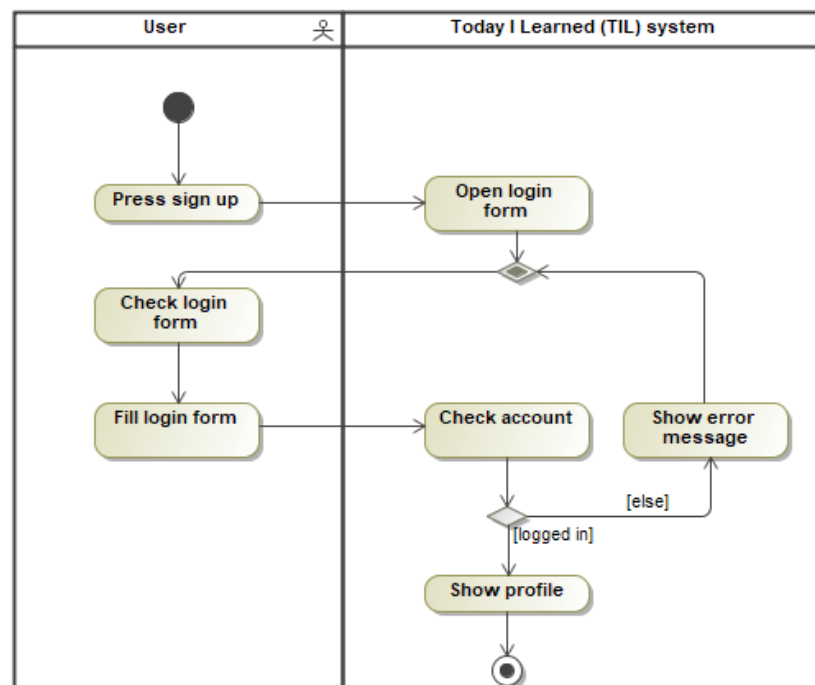
Pav. 2.2 Funkcinių reikalavimų atvaizdavimas panaudojimo atvejų diagrama

### 2.4.2. Klasių diagrama (veiklos diagramos)

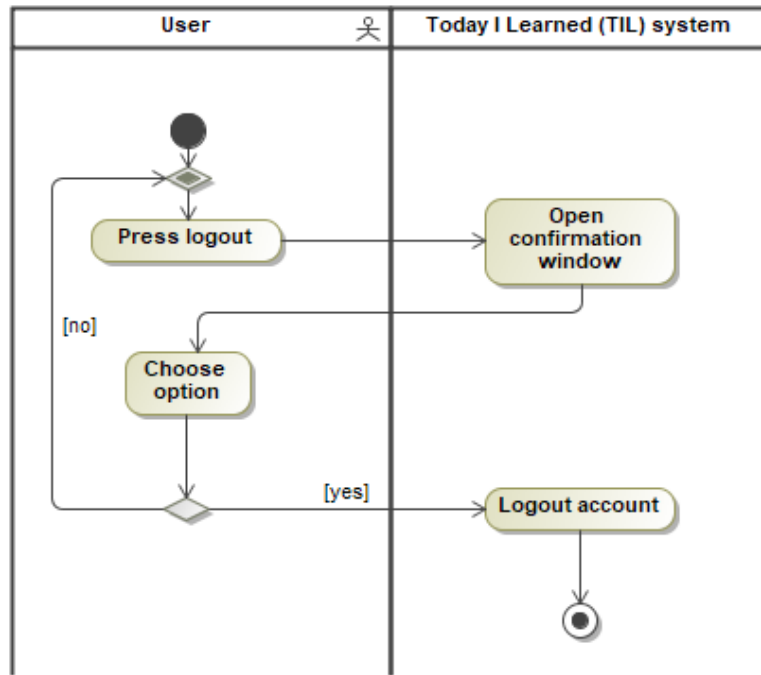
Kiekvienos funkcijos kaip turėtų veikti atvaizdavimui naudojamos veiklos diagramos.



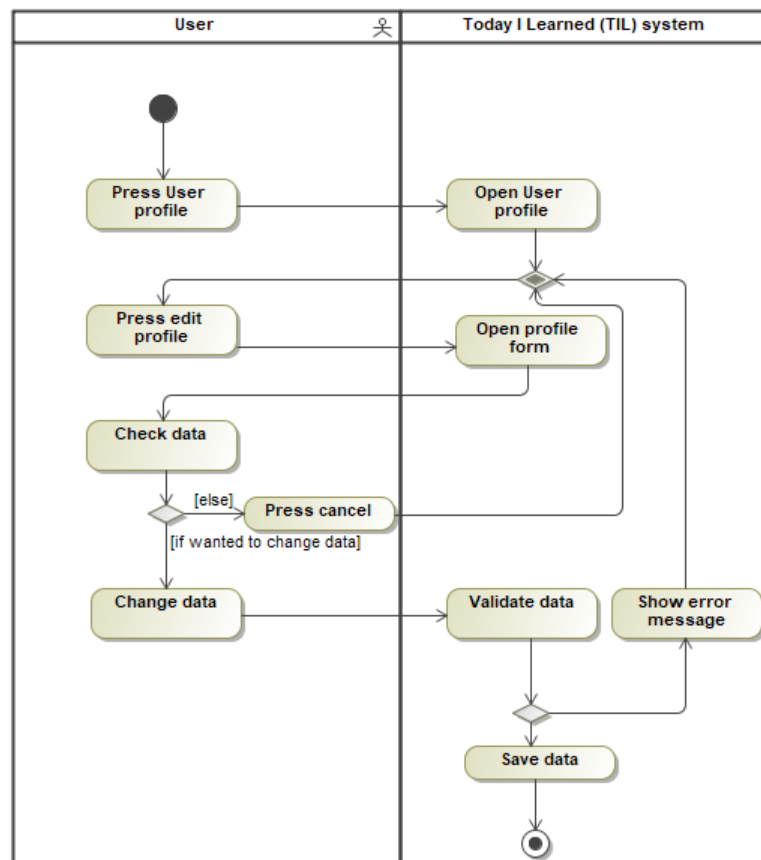
Pav. 2.3 Registracijos veiklos diagrama



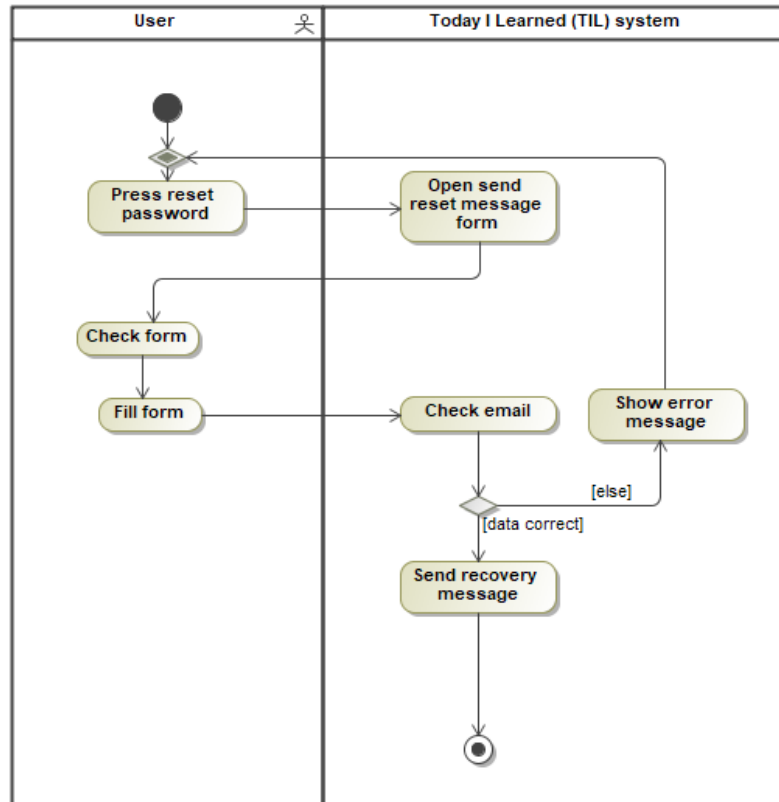
Pav. 2.4 Prisijungimo veiklos diagrama



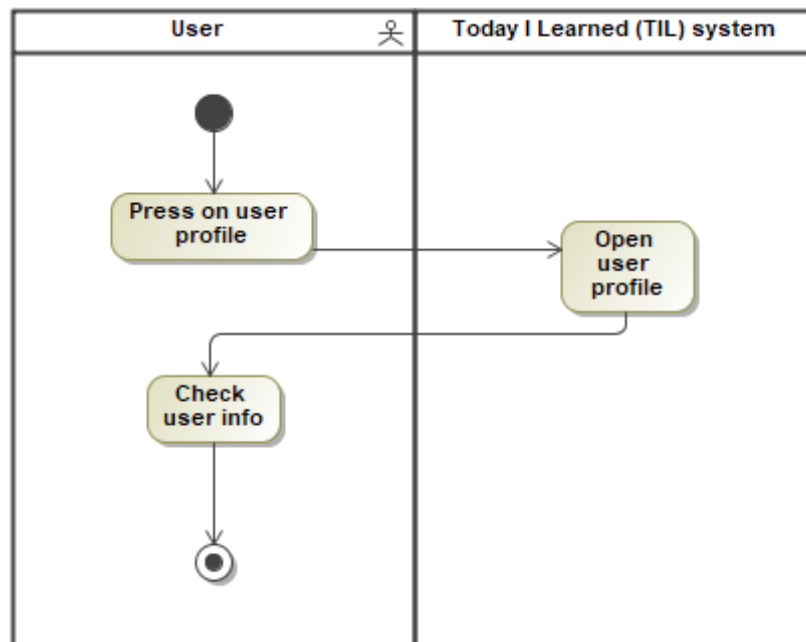
Pav. 2.5 Atsijungimo veiklos diagrama



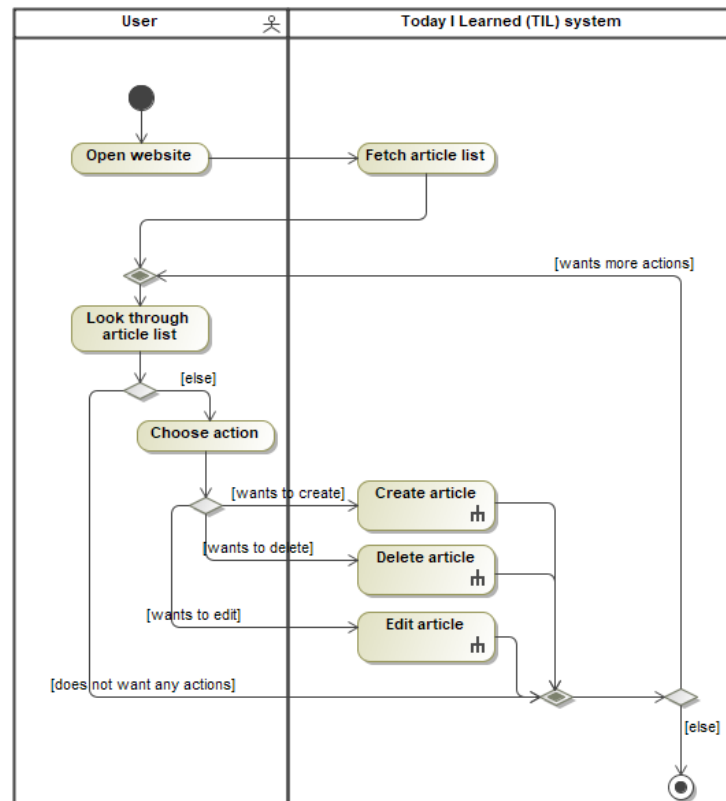
Pav. 2.6 Pakoreguoti profilį veiklos diagrama



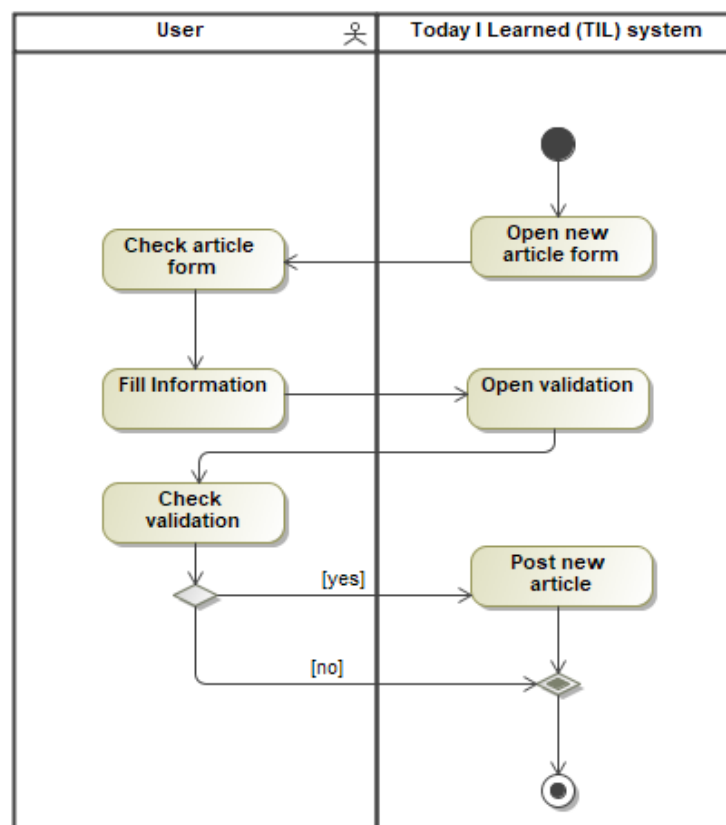
Pav. 2.7 Susigražinti slaptažodį veiklos diagrama



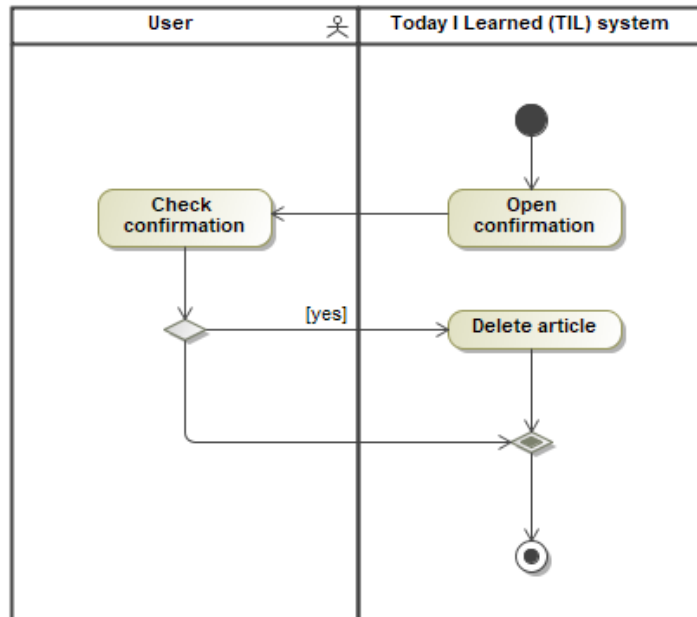
Pav. 2.8 Peržiūrėti profilį panaudojimo atvejis



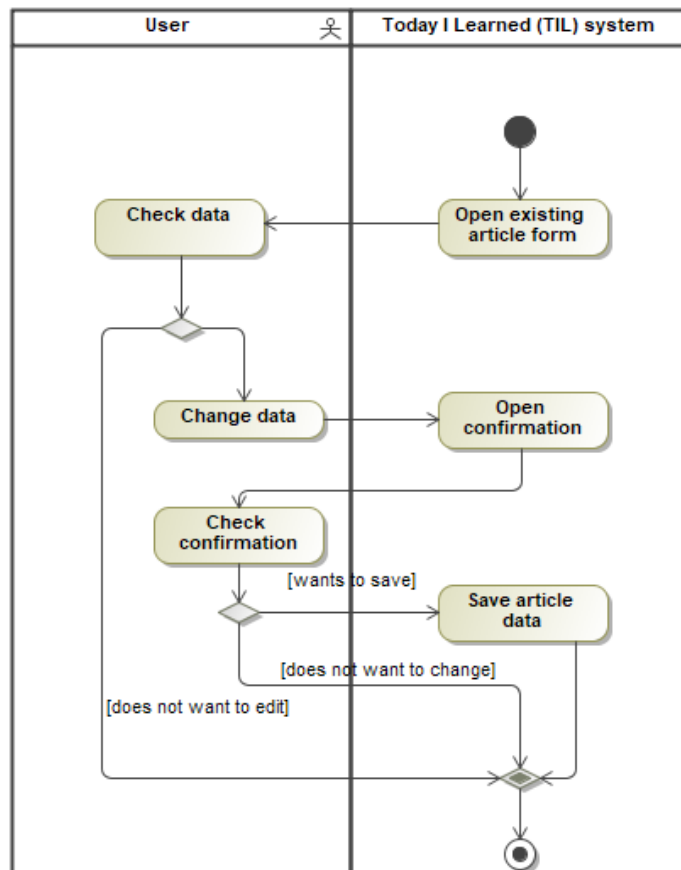
Pav. 2.9 Peržiūrėti straipsnių sąrašą veiklos diagrama



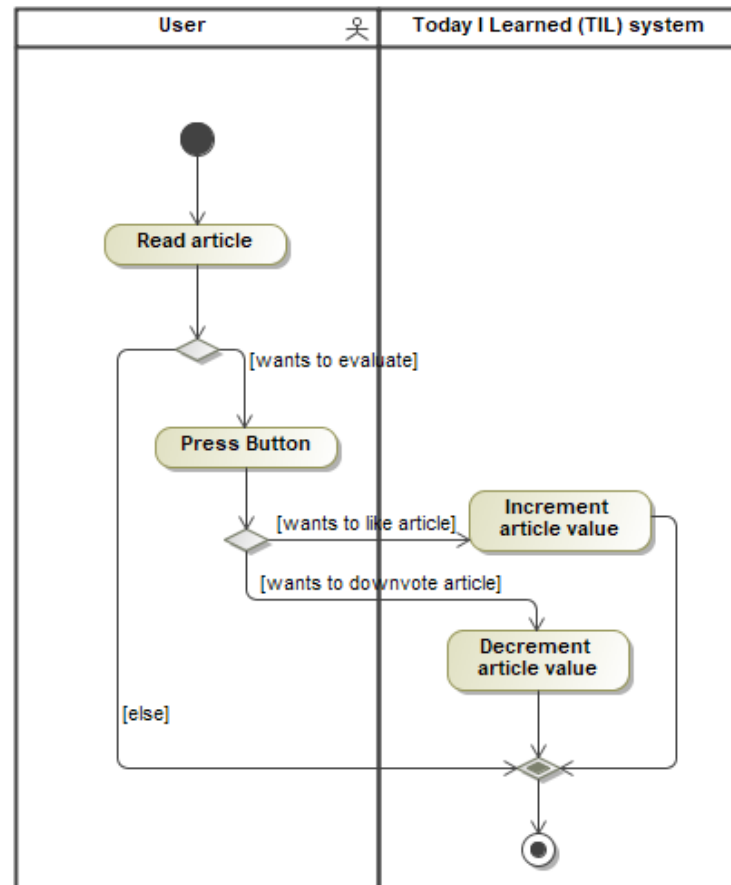
Pav. 2.10 Kurti straipsnį veiklos diagrama



Pav. 2.11 Trinti straipsnį veiklos diagrama



Pav. 2.12 Redaguoti straipsnį veiklos diagrama



Pav. 2.13 Vertinti straipsnį veiklos diagrama



### 3. Vartotojo vadovas

#### 3.1.Diegimas

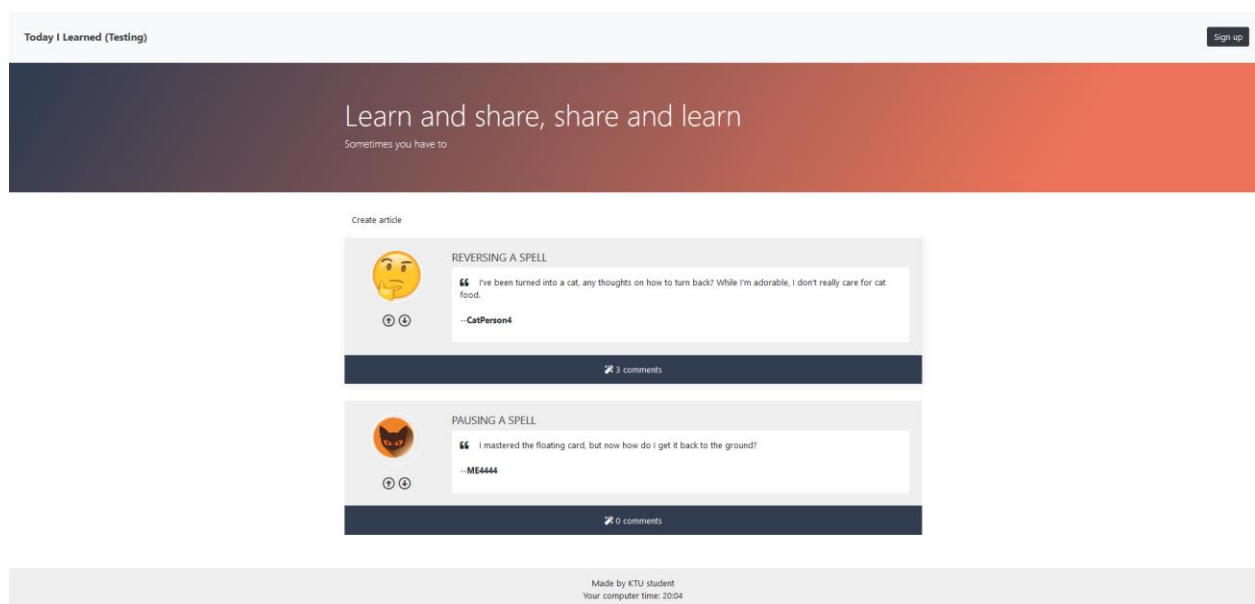
Norint pasileisti patį projektą reikia įsirašyti:

- XAMPP;
- Composer;
- PHP;
- Nodejs;
- Yarn.

Tada reikia atsidaryti projekto aplanką ir parašyti komandą *composer install*, tai įrašys visus reikalingus Symfony failus.

#### 3.2.Sistemos veikimas

Naudotojas atidaręs tinklapį mato tokį vaizdą. Jis gali naršyti straipsnių sąrašą, skaityti juos, jų komentarus, vertinti juos (žr. Pav. 3.2.1).



Pav. 3.2.1 Tinklapių pagrindinis langas

**Įvedimo laukų ir mygtukų paaiškinimai:**

- Mygtukas „# comments“ – atidaro straipsnio komentarų sekciją.
- Mygtukas „Sign up“ – atidaro prisijungimo formą.
- Mygtukas „Today I Learned(Testing)“ – atidaro puslapio pagrindį puslapį.
- Mygtukas „Create article“ – atidaro naujo straipsnio kūrimo formą.
- Mygtukas „Up vote“ – prideda +1 prie straipsnio įvertinimo.
- Mygtukas „Down vote“ – atima -1 nuo straipsnio vertinimo.

Paspausdus ant „Create article“ atidaromas straipsnio kūrimo langas. Naudotojas gali uždėti pavadinimą ir parašyti straipsnį, kurį matys kiti naudotojai.

Today | Learned (Testing) Sign up

Create new article

Title

Text

Create new Cancel

Made by KTU student  
Your computer time: 20:07

Pav. 3.2.2 Naujo straipsnio kūrimo langas

### Įvedimo laukų ir mygtukų paaiškinimai:

- „Title“ – straipsnio pavadinimo laukas
- „Text“ – straipsnio teksto laukas
- Mygtukas „Create New“ – sukuria duomenų bazėje naują straipsnį
- Mygtukas „Cancel“ – atšaukia veiksmą ir grąžina į pagrindinį langą

Žmogui nesisiekant prisijungti, jis spaudžia paryškintą žodžių junginį „Forgot Password“. Atidaromas laukas, kur jis gali įrašyti savo paštą ir jei egzistuoja paštas su paskyra – siunčiama susigrąžinimo nuoroda.

Today | Learned (Testing) Sign up

Reset Password

Email

Send to email

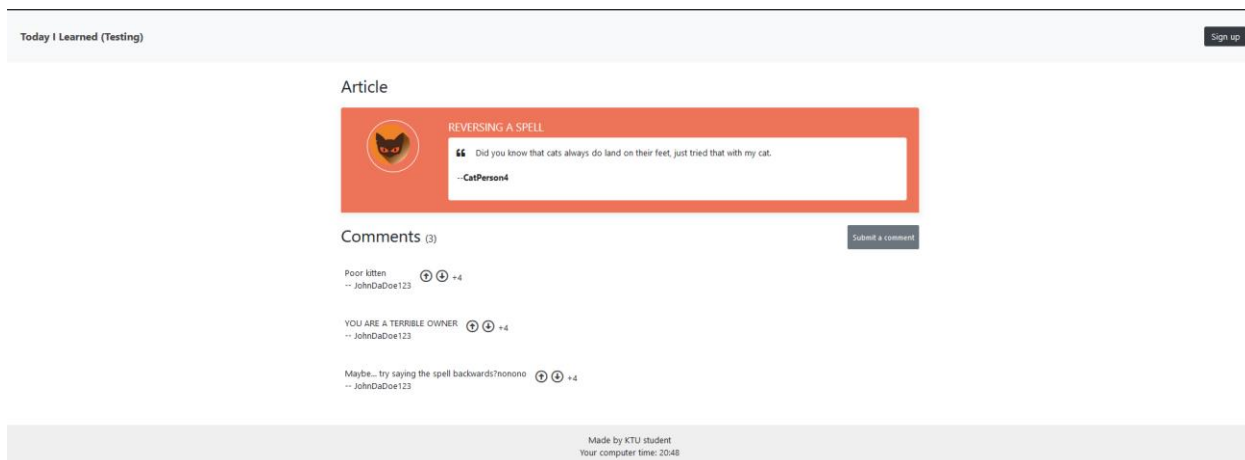
Made by KTU student  
Your computer time: 20:08

Pav. 3.2.3 Slaptažodžio atgavimo langas

### Įvedimo laukų ir mygtukų paaiškinimai:

- „Email“ – reikia įrašyti egzistuojantį paštą
- Mygtukas „Send to email“ – siunčia susigrąžinimo nuorodą į nurodytą paštą

Pasirinkus straipnį ir ant jo paspaudus atidaromas jo turinio langas. Jame matomi komentarai, kuriuos galima įvertinti gerai arba blogai.



Pav. 3.2.4 Straipsnio turinio langas

#### Įvedimo laukų ir mygtukų paaiškinimai:

- Mygtukas „Upvote“ - paspaudus pridedamas komentarui taškas
- Mygtukas „Submit a comment“ – sukuriamas komentaras
- Mygtukas „Downvote“ – paspaudus atimamas komentarui taškas

Prisijungimo lange paspaudus mygtuką „Register“ įjungiamas registracijos langas. Čia naudotojas suveda duomenis ir sukuriamą paskyrą.

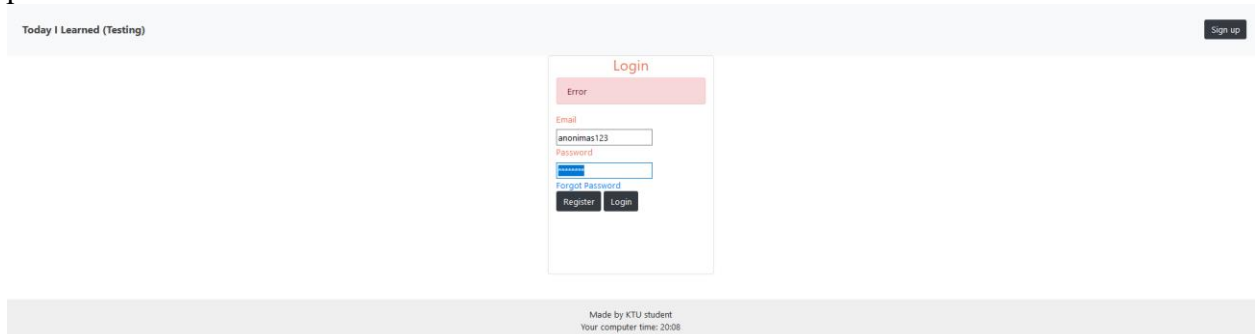
Pav. 3.2.5 Registracijos langas

#### Įvedimo laukų ir mygtukų paaiškinimai:

- „Name“ – naudotojo vardas
- „Surname“ – naudotojo pavardė
- „Username“ – naudotojo slapyvardis
- „Password“ – naudotojo slaptažodis
- „Email“ – naudotojo paštas
- „Phone number“ naudotojo telefono numeris
- „Age“ – naudotojo amžius

- Mygtukas „Register“ – jį paspaudus tikrinami duomenys, jei teisingi – sukuriama paskyra

Naudotojui paspaudus „Sign Up“ mygtuką įjungiama prisijungimo forma. Joje gali suvesti savo paskyros duomenis, jei jis turi tokią. Jei paskyrą neegzistuojanti – metamas klaidos pranešimas.



The screenshot shows a web application interface. At the top left, it says "Today I Learned (Testing)". At the top right, there is a "Sign Up" button. In the center, there is a "Login" form. The form has a red "Error" message at the top. Below the error message, there are two input fields: "Email" with the value "anonas123" and "Password" with the value "123456". Below these fields, there are two buttons: "Register" and "Login". At the bottom of the page, there is a footer that says "Made by KTU student" and "Your computer time: 2008".

Pav. 3.2.6 Prisijungimo langas

### Įvedimo laukų ir mygtukų paaiškinimai:

- Email – Naudotojo egzistuojantis paštas.
- Password – Naudotojo prisijungimui sukurtas slaptažodis.
- Mygtukas „Register“ – Nuoroda į registracijos formą.
- Mygtukas „Login“ – Naudotojo duomenų siuntimas į sistemą ir patvirtinti

## 4. Išvados

1. Atlikus esamų sprendimų analizę galima teigti, kad sunku konkuruoti su didelio masto, įsitvirtinusiems ir žmonėms labai patinkančiais projektais. Reddit'as populiarus platforma, kurioje galima sukurti savo norimos temos mini puslapį, kuris vadinamas subreddit'u. Visą informacijos struktūrą, tipą kontroliuoja moderatoriai ir žmonės. Intuityvus UI leidžia žmonėms patogiai kurti prie straipsnių diskusijas, apdovanojimų funkcija skatina žmones įsiterpti arba sukurti savo komentarą. Today I Found Out tinklapis yra ryškus kontrastas. Jo turinio kūrėjus sudaro maža grupelė žmonių, kurie rašo visus tinklapio straipsnius, todėl pasikartojantis rašymo stilius gali naudotojui atsibosti. Dizainas nėra patogus ir nėra nuosekliai išdėstytas, todėl blaško dėmesį ir paskui nepastebimi įdomesni straipsniai. Atliekant darbą atsižvelgta į Reddit'o stiprias puses ir stengtasi jas išnaudoti.
2. Atlikus technologijų analizę sužinojau išsamiau apie Symfony. Įrankiai skirti padėti programuotojui realizuoti sistemą išlaikant Model View Controller (MVC) architektūrą. Jų esmė suteikti kodui universalumo, kad būtų galima daug kartų naudoti. Aprašant Symfony aplikaciją PHP failai rašomi klasėmis ir taip propaguojamas objektinis programavimas (OOP angl. *Object-Oriented Programming*). Todėl sudarant duomenų bazę labai gerai sąveikauja Doctrine, kuris suteikia kodui lankstumo ir paprastumo. Dėl Symfony skatinamo kodo rašymo stiliaus, aprašant kontrolierius su PHP, procesas buvo panašus kaip programuojant su C# kalba. Visą darbą padėjo realizuoti SymfonyCast šaltiniai. Ypač gerai sąveikaujantis su Symfony IDE (angl. *Integrated Development Environment*) buvo PHPStorm, kuris automatiškai saugo kodą, glaudžiai integruotą įrankių sąsają su nustatytais veikiančiais nustatymais.
3. Projektavimo metu buvo naudotas MagicDraw įrankis, kurio funkcionalumas atitinka UML standartus. Jis leido sukurti panaudojimo atvejų, veiklų ir klasių diagramas. Turint sugalvotus funkcinius reikalavimus buvo lengva realizuoti panaudojimo atvejų diagramą. Po to reikėjo realizuoti kiekvienos funkcijos veikimą. Duomenų bazė buvo atvaizduota dviem būdais: ER diagrama ir Doctrine būdu.
4. Realizuojant sistemą buvo susidurta su daug problemų. PHP yra pritaikytas programuoti ne Windows aplinkoje, todėl dažnai kildavo sunkumų veikimo greičiu. Naudojant Symfony tai išryškėja, nes tuščią projektą atidaryti Windows'ų operacinėje sistemoje prireikia 16 s, kas mažina darbo produktyvumą. Išsaugoti naujas reikšmes reikėdavo 5 s. Realizuoti visi sistemos funkciniai reikalavimai.
5. Atliekant sistemos testavimą buvo pastebėta, kad trūksta tam tikrose vietose funkcijų ir naudotojui aiškių nurodymų. Atsiliepimai nurodė, kad galėtų būti gražesnis dizainas.

## 5. Literatūros sąrašas

- [1] „SymfonyCasts,“ KnpUniversity, [Tinkle]. Available: <https://symfonycasts.com/courses>. [Kreiptasi 20 05 2020].
- [2] „Symfony,“ Symfony SAS, [Tinkle]. Available: <https://symfony.com/>. [Kreiptasi 20 05 2020].
- [3] SymfonyCasts, „Github,“ KnpUniversity, [Tinkle]. Available: <https://github.com/SymfonyCasts/symfony5>. [Kreiptasi 20 05 2020].
- [4] SymfonyCasts, „Symfony 4 Github,“ KnpUniversity, [Tinkle]. Available: <https://github.com/knpuniversity/symfony4>. [Kreiptasi 20 05 2020].
- [5] J. B. Nils Adermann, „Composer,“ Community Contribution, [Tinkle]. Available: <https://getcomposer.org/>. [Kreiptasi 20 05 2020].
- [6] „StackOverFlow,“ Stack Exchange Inc, [Tinkle]. Available: <https://stackoverflow.com/>. [Kreiptasi 20 05 2020].
- [7] „W3Schools,“ Refsnes Data, [Tinkle]. Available: <https://www.w3schools.com/>. [Kreiptasi 20 05 2020].
- [8] Bootstrap team, „Bootstrap,“ Bootstrap team, [Tinkle]. Available: <https://getbootstrap.com/>. [Kreiptasi 20 05 2020].
- [9] Arcanis, „YarnPkg,“ [Tinkle]. Available: <https://yarnpkg.com/>. [Kreiptasi 20 05 2020].
- [10] Subreddit community, „Reddit r/todayilearned,“ Advance Publications, 28 12 2008. [Tinkle]. Available: <https://www.reddit.com/r/todayilearned>. [Kreiptasi 20 05 2020].
- [11] Todayifoundout team, „Today I Found Out,“ 01 01 2010. [Tinkle]. Available: <https://www.todayifoundout.com/>. [Kreiptasi 20 05 2020].
- [12] „Twig,“ Symfony, 2010. [Tinkle]. Available: <https://twig.symfony.com/>. [Kreiptasi 20 05 2020].
- [13] „NodeJS,“ Joyent Inc., [Tinkle]. Available: <https://nodejs.org/en/>. [Kreiptasi 20 05 2020].

## 6. Priedas

### 6.1. Semestro darbų suvestinė

Dikšas Tautvydas, IFF-7/13

Veikla	Sugaištas laikas valandomis
El. paštui patvirtinimo siuntimas	7
Pradinės ataskaitos parengimas	4
Panaudojimo atvejų kūrimas	0.5
Symfony sistemos paruošimas	1.5
Prisijungimo langas	1
Slaptažodžio susigrąžinimo langas	1
Registracijos langas	1.5
Doctrine paskaitų peržvalga ir taikymas	3
Veiklos diagramų braižymas	3
Galutinės ataskaitos užpildymas	4
Įrankių greičio tvarkymas	7
Projekto dydžio tvarkymas	6
Objektų ryšių realizavimas	4
Nereikalingų įrankių naikinimas	4
Perteklinių paketų trinimas	6
Kodo tvarkymas	8
Cache problemų sprendimas	4
Klausimų ir slaptažodžių šifravimas	7
Straipsnių „upvote“ ir „downvote“	1.5
Komentarų „upvote“ ir „downvote“	1
Profilų nuotraukų apdorojimas	3
Profilio redagavimas	3
Vartotojo vadovo rašymas	2
Paskyros autentifikavimas	2
Straipsnio trynimas	2
Kodo optimizavimas	2
Komentarų trynimas	3
Klaidų puslapių kūrimas	3
Duomenų bazės perdarymas	1
Google Cloud Platform paruošimas	2
Projekto paleidimas ant hostingo	7
Duomenų bazės tvarkymas hostinge	5
Galutinės ataskaitos ruošimas	1.5
Puslapiavimas	8
Doctrine migracijų tvarkymas	2
Viso:	121.5