

Slovenská technická univerzita v Bratislave
Fakulta informatiky a informačných technológií
Ilkovičova 2, 842 16 Bratislava 4

Tímový projekt



Story Teller

Projektová dokumentácia – inžinierske dielo

Vedúci projektu: Ing. Karol Rástočný, PhD.
Názov tímu: CoolStoryBro
Členovia tímu: Bc. Jakub Ondík
Bc. Patrik Januška
Bc. Adam Neupauer
Bc. Martin Olejár
Bc. Miroslav Hurajt
Kontakt: storyteller-04@googlegroups.com
Akademický rok: 2016/2017

Obsah

1	Úvod	1-1
2	Globálne ciele projektu	2-1
3	Celkový pohľad na systém	3-1
3.1	Backend	3-2
3.1.1	UserManagement	3-2
3.1.2	ProjectManagement.....	3-2
3.1.3	SketchManagement	3-2
3.1.4	ScenarioManagement	3-2
3.1.5	TestManagement	3-2
3.1.6	DrawableComponents	3-3
3.1.7	DrawableTemplateProvider	3-3
3.1.8	HtmlRenderer	3-3
3.1.9	Notifications	3-3
3.1.10	WebsocketHandler	3-3
3.1.11	Authentication	3-3
3.1.12	AccessControlHandler	3-4
3.1.13	wkhtmltopdf	3-4
3.1.14	PostgreSQL	3-4
3.2	Client	3-4
3.2.1	UserService	3-4
3.2.2	ProjectService.....	3-4
3.2.3	SketchService	3-5
3.2.4	ScenarioService	3-5
3.2.5	TestService	3-5
3.2.6	DrawableComponents	3-6
3.2.7	Navigation	3-6
3.2.8	WebsocketHandler	3-6
3.2.9	EventHandler.....	3-6
3.2.10	Notifications	3-7
3.3	Dátový model	3-7

1 Úvod

Tento dokument predstavuje dokumentáciu inžinierskeho diela, ktoré je vytvárané v rámci predmetu Tímový projekt v akademickom roku 2016/2017. Dokument zahŕňa globálne ciele projektu pre jednotlivé semestre a celkový pohľad na štruktúru a funkcionality systému.

Názov témy nášho projektu je *Story Teller – Zber a vyhodnocovanie požiadaviek* a jeho cieľom je vytvoriť informačný systém, ktorý umožní pohodlné zadávanie a sledovanie splnenia používateľských scenárov (funkcionálnych požiadaviek). Zber požiadaviek bude realizovaný prostredníctvom vizuálneho skicovania obrazoviek scenáru a ich komentovania, na základe čoho môžu byť vygenerované akceptačné testy. Ďalšou funkcionality systému bude vykonávanie akceptačných testov a komentovanie ich výsledkov.

Prvá kapitola dokumentu obsahuje globálne ciele projektu. V druhej kapitole dokumentu je bližšie popísaná architektúra systému, jej jednotlivé komponenty a vzťahy medzi nimi. Pri každom komponente je uvedený odkaz na priložené elektronické dokumenty (technická dokumentácia a používateľská príručka) a ich kapitoly. Kapitola obsahuje taktiež dátový model aplikácie.

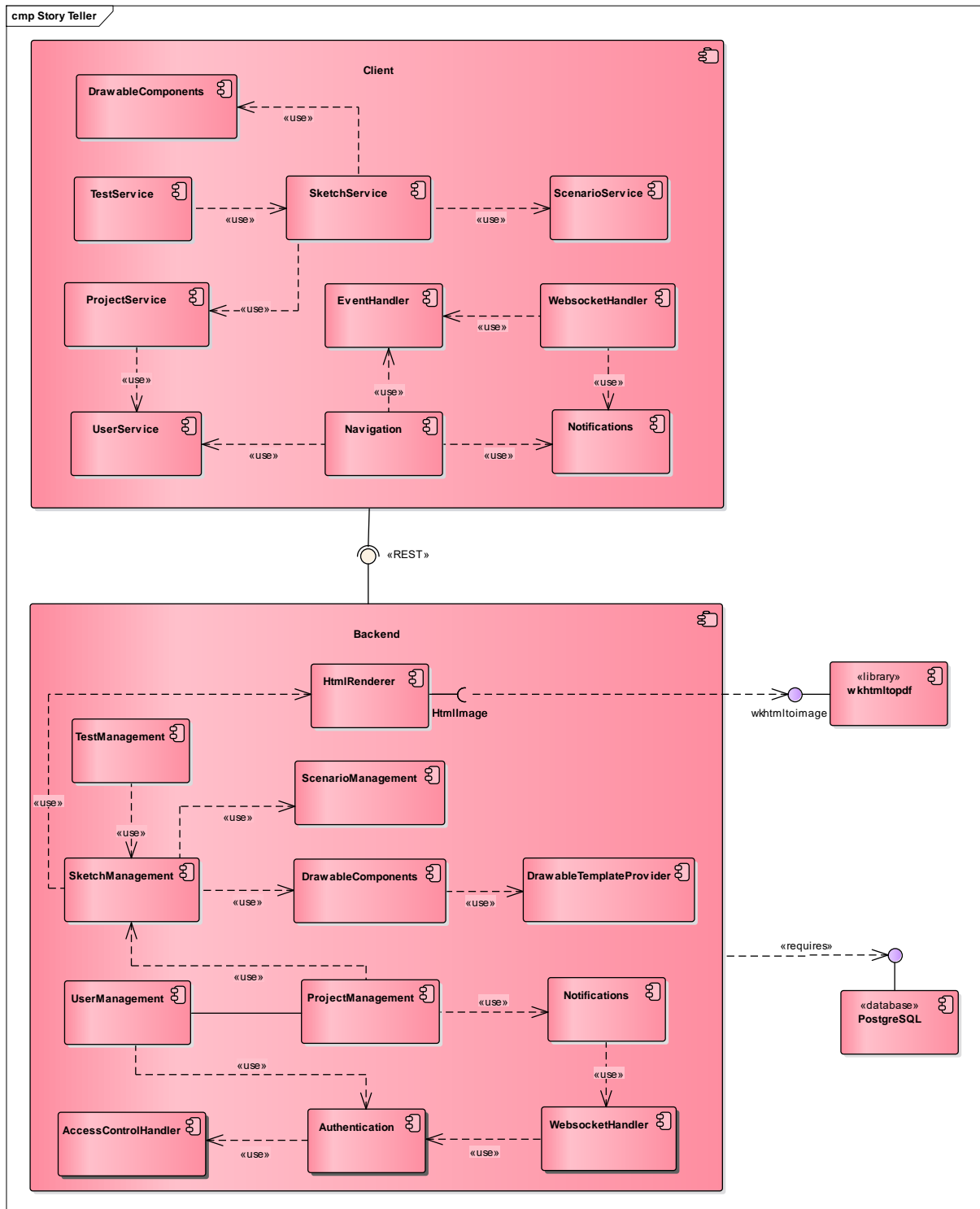
2 Globálne ciele projektu

Globálne ciele projektu sú nasledovné:

1. Autentifikácia používateľa vrátane mechanizmov na obnovenie prístupu
2. Riadenie prístupu k projektom
3. Vytvorenie a správa projektov
4. Tvorba a modifikácia skíc
5. Zdieľanie projektov a skíc
6. Generovanie akceptačných testov
7. Centrálné vykonávanie akceptačných testov aspoň pre jednu platformu
8. Možnosť sledovania vykonávania (výsledkov) akceptačných testov

3 Celkový pohľad na systém

Aplikácia je realizovaná klient – server architektúrou, pričom klienti sú tenkí – webové prehliadače a aplikácie pre tablety. Títo klienti komunikujú so serverom pomocou HTTP protokolu cez REST rozhranie. Táto architektúra vrátane závislosti komponentov je zobrazená na obrázku 1.



Obrázok 1: Architektúra systému

3.1 Backend

3.1.1 UserManagement

Komponent UserManagement rieši celkovú správu používateľov, vrátane aktivácie a deaktivácie používateľského účtu. Taktiež rieši zmenu a obnovu hesla, obnovu aktivačných, resp. deaktivovaných tokenov a úpravu informácií o používateľoch.

Technická dokumentácia

Kapitola 2: https://1drv.ms/b/s!AnORgXjFACoEn0AdHgigu8NL_DrG

3.1.2 ProjectManagement

Komponent ProjectManagement zabezpečuje vytváranie a správu projektov, pre ktoré sa budú vytvárať skice a generovať testy.

Technická dokumentácia

Kapitola 5: https://1drv.ms/b/s!AnORgXjFACoEn0AdHgigu8NL_DrG

3.1.3 SketchManagement

Komponent SketchManagement zabezpečuje správu skíc. Taktiež zabezpečuje vygenerovanie ich náhľadu ako obrázku.

Technická dokumentácia

Kapitola 8: https://1drv.ms/b/s!AnORgXjFACoEn0AdHgigu8NL_DrG

3.1.4 ScenarioManagement

Komponent ScenarioManagement zabezpečuje správu scenárov, pre ktoré sa budú generovať testy.

Technická dokumentácia

Kapitola 9: https://1drv.ms/b/s!AnORgXjFACoEn0AdHgigu8NL_DrG

3.1.5 TestManagement

Komponent TestManagement zabezpečuje vygenerovanie testu pre konkrétny scenár skice a správu testov a ich výsledkov.

Technická dokumentácia

Kapitola 9: https://1drv.ms/b/s!AnORgXjFACoEn0AdHgigu8NL_DrG

3.1.6 DrawableComponents

Komponent DrawableComponents predstavuje komponenty tvoriace skicu. Príkladom takéhoto komponentu je textové pole. Tento komponent taktiež zabezpečuje nastavenie CSS vlastností (štýlov) a konverziu na html kód pred renderovaním skice ako obrázka.

Technická dokumentácia

Kapitola 8: https://1drv.ms/b/s!AnORgXjFACoEn0AdHgigu8NL_DrG

3.1.7 DrawableTemplateProvider

Komponent DrawableTemplateProvider predstavuje poskytovateľa šablón dostupných komponentov, ktoré ponúka komponent DrawableComponents.

Technická dokumentácia

Kapitola 8: https://1drv.ms/b/s!AnORgXjFACoEn0AdHgigu8NL_DrG

3.1.8 HtmlRenderer

Komponent HtmlRenderer predstavuje wrapper nad knižnicou wkhtmltopdf. Zabezpečuje vytvorenie procesu, v ktorom je táto knižnica spúšťaná, a presmerovanie deskriptora stdout do bajtového poľa v Jave.

Technická dokumentácia

Kapitola 8: https://1drv.ms/b/s!AnORgXjFACoEn0AdHgigu8NL_DrG

3.1.9 Notifications

Komponent Notifications zabezpečuje odosielanie notifikácií konkrétnym používateľom, resp. používateľovi. Odosielanie prebieha prostredníctvom websocketov.

Technická dokumentácia

Kapitola 7: https://1drv.ms/b/s!AnORgXjFACoEn0AdHgigu8NL_DrG

3.1.10 WebSocketHandler

Komponent WebSocketHandler zabezpečuje vytvorenie STOMP websocketového servera. Taktiež zabezpečuje registráciu kanálov a samotné publish / subscribe funkcie.

Technická dokumentácia

Kapitola 7: https://1drv.ms/b/s!AnORgXjFACoEn0AdHgigu8NL_DrG

3.1.11 Authentication

Komponent Authentication zabezpečuje overenie používateľa. Taktiež zabezpečuje vygenerovanie autentifikačných tokenov v prípade správneho zadania prihlasovacích údajov a

overenie identity používateľa pre websockety. Tokeny boli zvolené z dôvodu poskytovania možnosti stateless autentifikácie pri prípadnom škálovaní aplikácie.

Technická dokumentácia

Kapitola 2: https://1drv.ms/b/s!AnORgXjFACoEn0AdHgigu8NL_DrG

3.1.12 AccessControlHandler

Komponent AccessControlHandler zabezpečuje riadenie prístupu na základe rolí. Roly nie sú globálne, ale pre konkrétny projekt.

Technická dokumentácia

Kapitola 5: https://1drv.ms/b/s!AnORgXjFACoEn0AdHgigu8NL_DrG

3.1.13 wkhtmltopdf

Komponent wkhtmltopdf predstavuje externú knižnicu zodpovednú za renderovanie html súborov do pdf formátu. Taktiež umožňuje renderovanie do jpg a png formátu, ktoré je použité komponentom HtmlRenderer.

3.1.14 PostgreSQL

Komponent PostgreSQL predstavuje databázový server, ktorý je využitý na perzistenciu údajov. Komunikácia je zabezpečená prostredníctvom JDBC pripojenia, pričom je použitý rámec Spring Data, ktorý používa entitno-relačný mapovač Hibernate.

3.2 Client

3.2.1 UserService

Komponent UserService zabezpečuje registráciu a prihlásenie, zaslanie požiadavky na obnovu hesla a aktivačných, resp. deaktivovaných tokenov, zobrazenie a modifikáciu používateľského profilu vrátane zmeny hesla a nastavenia lokalizácie. Tiež zabezpečuje uloženie autentifikačného tokenu a jeho pripojenie k požiadavkám zasielaných serverovej aplikácií.

Technická dokumentácia

Kapitola 1: https://1drv.ms/b/s!AnORgXjFACoEn0AdHgigu8NL_DrG

Používateľská príručka

Kapitoly 2 a 3: <https://1drv.ms/b/s!AnORgXjFACoEhQ7nvJNqzZfLQxQM>

3.2.2 ProjectService

Komponent ProjectService zabezpečuje vytvorenie projektu a modifikáciu jeho informácií, vrátane správy používateľov, ktorý k projektu majú prístup. Taktiež zabezpečuje rozhranie pre

projekt a projektovú navigačnú lištu a zobrazenie projektov pre aktuálne prihláseného používateľa.

Technická dokumentácia

Kapitola 5: https://1drv.ms/b/s!AnORgXjFACoEn0AdHgigu8NL_DrG

Používateľská príručka

Kapitoly 4: <https://1drv.ms/b/s!AnORgXjFACoEhQ7nvJNqzZfLQxQM>

3.2.3 SketchService

Komponent SketchService zabezpečuje vytvorenie skice a v budúcnosti aj ich modifikácie a zdieľanie vrátane zobrazovania. Taktiež zabezpečuje zloženie skice do JSON objektu, ktorý je následne odoslaný na serverovú časť aplikácie na ďalšie spracovanie.

Technická dokumentácia

Kapitola 8: https://1drv.ms/b/s!AnORgXjFACoEn0AdHgigu8NL_DrG

Používateľská príručka

Kapitoly 5: <https://1drv.ms/b/s!AnORgXjFACoEhQ7nvJNqzZfLQxQM>

3.2.4 ScenarioService

Komponent ScenarioService zabezpečuje vytvorenie scenárov a v budúcnosti aj ich modifikácie.

Technická dokumentácia

Kapitola 9: https://1drv.ms/b/s!AnORgXjFACoEn0AdHgigu8NL_DrG

Používateľská príručka

Kapitoly 6: <https://1drv.ms/b/s!AnORgXjFACoEhQ7nvJNqzZfLQxQM>

3.2.5 TestService

Komponent TestService zabezpečuje spustenie a prehranie testu prostredníctvom videa.

Technická dokumentácia

Kapitola 9: https://1drv.ms/b/s!AnORgXjFACoEn0AdHgigu8NL_DrG

Používateľská príručka

Kapitoly 7: <https://1drv.ms/b/s!AnORgXjFACoEhQ7nvJNqzZfLQxQM>

3.2.6 DrawableComponents

Komponent DrawableComponents zabezpečuje inicializáciu dostupných komponentov na základe ich šablón získaných zo serverovej časti aplikácie. Taktiež zabezpečuje nastavenie ich základných vlastností, medzi ktoré patrí napríklad farba.

Technická dokumentácia

Kapitola 8: https://1drv.ms/b/s!AnORgXjFAcOEn0AdHgigu8NL_DrG

Používateľská príručka

Kapitoly 5: <https://1drv.ms/b/s!AnORgXjFAcOEhQ7nvJNqzZfLQxQM>

3.2.7 Navigation

Komponent Navigation zabezpečuje zobrazenie navigačnej lišty v hornej časti aplikácie. Taktiež zabezpečuje zobrazenie počtu aktuálne neprečítaných notifikácií a poskytuje zobrazenie okna s notifikáciami. Tiež zabezpečuje zobrazenie mena aktuálne prihláseného používateľa a zobrazenie používateľskej ponuky.

Technická dokumentácia

Kapitola 7: https://1drv.ms/b/s!AnORgXjFAcOEn0AdHgigu8NL_DrG

Používateľská príručka

Kapitoly 2: <https://1drv.ms/b/s!AnORgXjFAcOEhQ7nvJNqzZfLQxQM>

3.2.8 WebSocketHandler

Komponent WebSocketHandler zabezpečuje pripojenie na websocketový server a autentifikáciu aktuálne prihláseného používateľa pripojením autentifikačného tokenu do správy. Použitý je protokol STOMP s upraveným SockJS klientom. Tento komponent taktiež zabezpečuje subscribe na konkrétny kanál, odoslanie správ a spracovanie publishnutých dát, napr. zavolaním notifikačného komponentu.

Technická dokumentácia

Kapitola 7: https://1drv.ms/b/s!AnORgXjFAcOEn0AdHgigu8NL_DrG

Používateľská príručka

Nie je, keďže ide o iba o spracovanie dát.

3.2.9 EventHandler

Komponent EventHandler zabezpečuje broadcast a zachytenie udalostí. Tieto udalosti sú v aplikácii použité na aktualizáciu počítadla neprečítaných notifikácií a na úpravu zoznamu notifikácií.

Technická dokumentácia

Kapitola 7: https://1drv.ms/b/s!AnORgXjFACoEn0AdHgigu8NL_DrG

Používateľská príručka

Nie je, keďže ide o iba o spracovanie dát.

3.2.10 Notifications

Komponent Notifications slúži na zobrazenie konkrétnych notifikácií s využitím knižnice ngToaster. Taktiež zabezpečuje načítanie počtu neprečítaných notifikácií, načítanie stránkovaných notifikácií a načítanie prehľadových notifikácií.

Technická dokumentácia

Kapitola 7: https://1drv.ms/b/s!AnORgXjFACoEn0AdHgigu8NL_DrG

Používateľská príručka

Kapitoly 3: <https://1drv.ms/b/s!AnORgXjFACoEhQ7nvJNqzZfLQxQM>

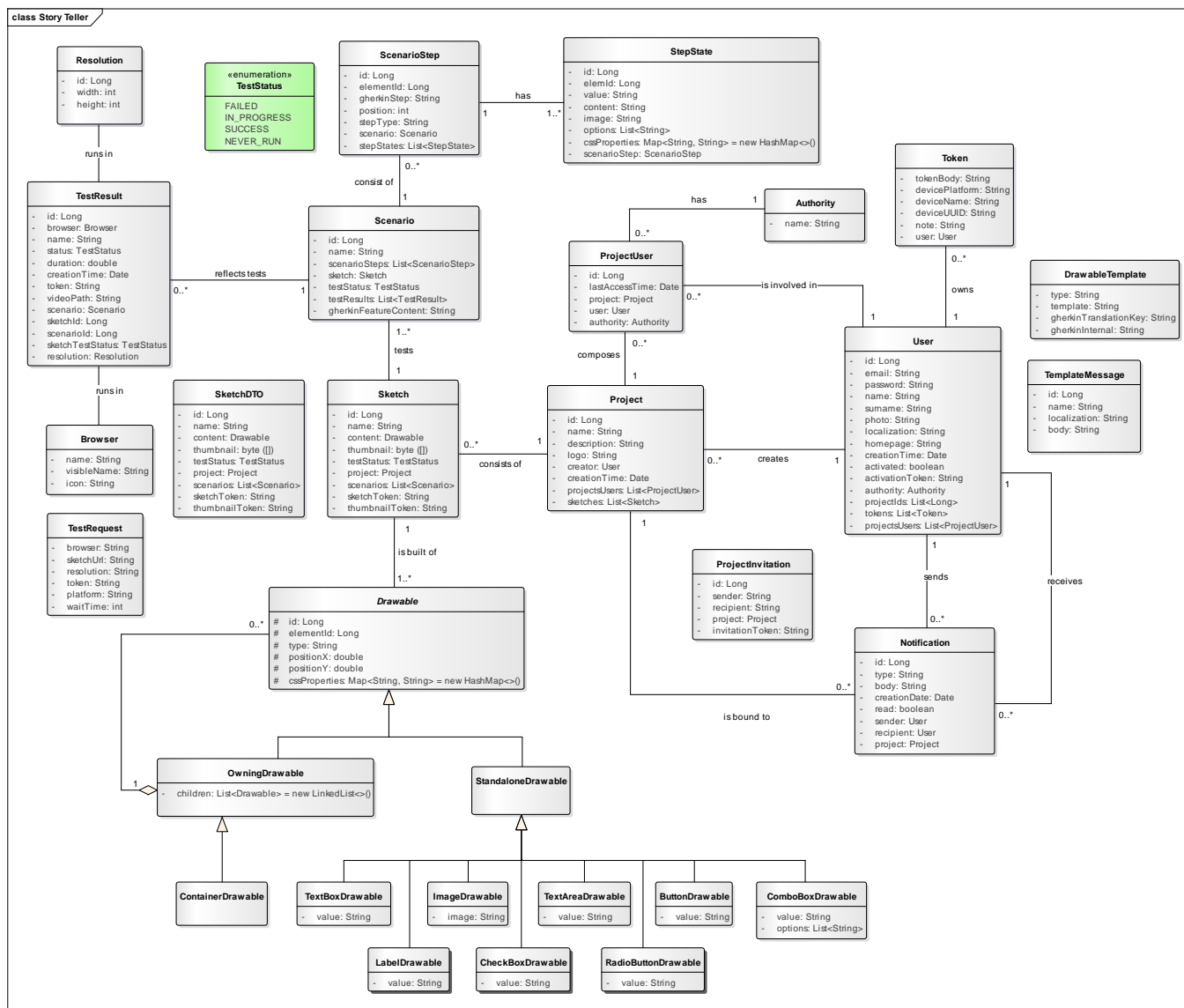
3.3 Dátový model

Dátový model aplikácie je znázornený na obrázku 2.

Opis jednotlivých tried:

- **Trieda User** predstavuje entitu používateľa vystupujúceho v systéme. Okrem základných samoopisných atribútov obsahuje zoznam autentifikačných tokenov pre všetky zariadenia, z ktorých je používateľ prihlásený.
- **Trieda Token** predstavuje entitu pre autentifikačné tokeny.
- **Trieda Authority** predstavuje konkrétnu rolu používateľa v danom projekte.
- **Trieda Project** reprezentuje projekt vytvorený v systéme.
- **Trieda ProjectUser** predstavuje spájaciu entitu obohatenú o konkrétnu rolu používateľa a čas posledného prístupu k projektu.
- **Trieda ProjectInvitation** predstavuje entitu pre pozvánky používateľov do projektov.
- **Trieda Notification** predstavuje notifikáciu zaslanú používateľovi systémom alebo iným používateľom. Preto môže byť naviazaná vzťahom 'is bound to' s projektom a vzťahom 'sends' s používateľom.
- **Trieda TemplateMessage** predstavuje emailovú šablónu, ktorá je staticky vytvorená a uložená v databáze a pri odoslaní sa vyplní reálnymi dátami.
- **Trieda Sketch** predstavuje entitu pre skice konkrétneho projektu.
- **Trieda SketchDTO** predstavuje pomocnú entitu pre skice.
- **Trieda Scenario** predstavuje entitu pre scenáre skice.
- **Trieda ScenarioStep** predstavuje entitu pre kroky scenáru.
- **Trieda StepState** predstavuje entitu pre stav kroku v scenári skice.
- **Trieda TestResult** predstavuje entitu pre výsledky testov jednotlivých scenárov.
- **Vymenovaný typ TestStatus** predstavuje entitu pre stav jednotlivých testov.
- **Trieda Browser** predstavuje entitu pre dostupné prehliadače.
- **Trieda Resolution** predstavuje entitu pre dostupné rozlíšenia.
- **Trieda DrawableTemplate** predstavuje entitu pre šablóny prvkov skíc.

- **Trieda Drawable** predstavuje abstraktnú reprezentáciu vykresliteľného komponentu skice. Táto časť je implementovaná pomocou návrhového vzoru Composite.
- **Trieda OwningDrawable** predstavuje komponent, ktorý môže obsahovať ďalšie komponenty. Príkladom takéhoto komponentu je kontajner.
- **Trieda StandaloneDrawable** predstavuje samostatný komponent, t. j. komponent, ktorý nemôže obsahovať ďalšie komponenty. Príkladom takéhoto komponentu je textové pole.
- **Trieda TextBoxDrawable** predstavuje konkrétny komponent – textové pole.
- **Trieda LabelDrawable** predstavuje konkrétny komponent – popisné pole.
- **Trieda ContainerDrawable** predstavuje konkrétny komponent – kontajner.
- **Trieda ImageDrawable** predstavuje konkrétny komponent – obrázok.
- **Trieda TextAreaDrawable** predstavuje konkrétny komponent – textová oblasť.
- **Trieda ButtonDrawable** predstavuje konkrétny komponent – tlačidlo.
- **Trieda CheckBoxDrawable** predstavuje konkrétny komponent – checkbox.
- **Trieda ComboBoxDrawable** predstavuje konkrétny komponent – combobox.
- **Trieda RadioButtonDrawable** predstavuje konkrétny komponent – radio button.



Obrázok 2: Dátový model aplikácie

Slovenská technická univerzita v Bratislave
Fakulta informatiky a informačných technológií
Ilkovičova 2, 842 16 Bratislava 4

Tímový projekt



Story Teller

Projektová dokumentácia – moduly systému

Vedúci projektu: Ing. Karol Rástočný, PhD.
Názov tímu: CoolStoryBro
Členovia tímu: Bc. Jakub Ondík
Bc. Patrik Januška
Bc. Adam Neupauer
Bc. Martin Olejár
Bc. Miroslav Hurajt
Kontakt: storyteller-04@googlegroups.com
Akademický rok: 2016/2017

Obsah

1	Úvod.....	1-1
2	Modul – Používateľ.....	2-1
2.1	Analýza.....	2-1
2.2	Návrh a implementácia.....	2-1
2.2.1	Registrácia.....	2-3
2.2.2	Prihlásenie.....	2-3
2.2.3	Úprava údajov v profile.....	2-4
2.2.4	Obnova hesla.....	2-4
2.2.5	Obnova registračného emailu.....	2-4
2.2.6	Správa prihlásených zariadení.....	2-4
2.2.7	Riadenie prístupu.....	2-5
2.2.8	Načítanie zoznamu používateľov v projekte.....	2-5
2.2.9	Pridanie používateľskej roly v rámci projektu.....	2-5
3	Modul – Lokalizácia.....	3-1
3.1	Analýza.....	3-1
3.2	Návrh a implementácia.....	3-1
4	Konfigurácia - Server a nasadzovanie.....	4-1
4.1	Konfigurácia.....	4-1
4.2	Backend.....	4-1
4.2.1	Závislosti.....	4-1
4.3	Client.....	4-2
5	Modul – Projekt.....	5-1
5.1	Analýza.....	5-1
5.2	Návrh a implementácia.....	5-1
5.2.1	Vytvorenie projektu.....	5-2
5.2.2	Načítanie zoznamu projektov prihláseného používateľa.....	5-2
5.2.3	Načítanie informácií o konkrétnom projekte.....	5-2
5.2.4	Úprava údajov o projekte.....	5-3
5.2.5	Priradenie roly používateľovi projektu.....	5-3
5.2.6	Pozvanie používateľa do projektu.....	5-3
6	Používateľské rozhranie.....	6-1

7	Modul – Notifikácie	7-1
7.1	Analýza	7-1
7.2	Návrh a implementácia	7-1
7.2.1	Odosielanie notifikácií	7-1
7.2.2	Prijímanie notifikácií	7-1
7.2.3	Zobrazenie notifikácií	7-2
8	Modul – Skice	8-1
8.1	Analýza	8-1
8.2	Návrh a implementácia	8-1
8.2.1	Perzistencia	8-1
8.2.2	Vytvorenie skice	8-2
8.2.3	Získanie skice	8-2
8.2.4	Úprava skice	8-3
8.2.5	Odstránenie skice	8-3
8.2.6	Načítanie skíc projektu	8-3
8.2.7	Export layoutu skice	8-3
8.2.8	Načítanie náhľadu skice	8-3
8.2.9	Zdieľanie skice	8-3
8.2.10	Načítanie zdieľaného náhľadu skice	8-4
8.2.11	Načítanie zdieľanej skice	8-4
9	Modul – Scenáre a testy	9-1
9.1	Analýza	9-1
9.2	Návrh a implementácia	9-1
9.2.1	Perzistencia	9-1
9.2.2	Vytvorenie, odstránenie a získanie scenáru	9-1
9.2.3	Vykonávanie testov	9-2

1 Úvod

Tento dokument obsahuje popis všetkých modulov vystupujúcich v systéme. Každý modul obsahuje analýzu, návrh a implementáciu vrátane diagramu tried tam, kde je to vhodné.

2 Modul – Používateľ

Úlohou modulu používateľ má byť riadenie prístupu k jednotlivým častiam a funkciám systému. Zároveň má ponúkať používateľovi možnosť prezentovať sa verejným profilom, obnoviť prístup do systému v prípade zabudnutia alebo straty identifikačných údajov.

2.1 Analýza

Registrácia bude prebiehať zadaním prihlasovacieho mena – emailu a prihlasovacieho hesla, podobne ako prihlásenie. Po registrácii bude používateľovi odoslaný aktivačný a deaktivovaný odkaz prostredníctvom emailovej správy. Pomocou týchto odkazov si bude môcť svoj účet aktivovať, čím sa mu umožní prihlásenie, ale aj deaktivovať, čím sa jeho účet zo systému odstráni.

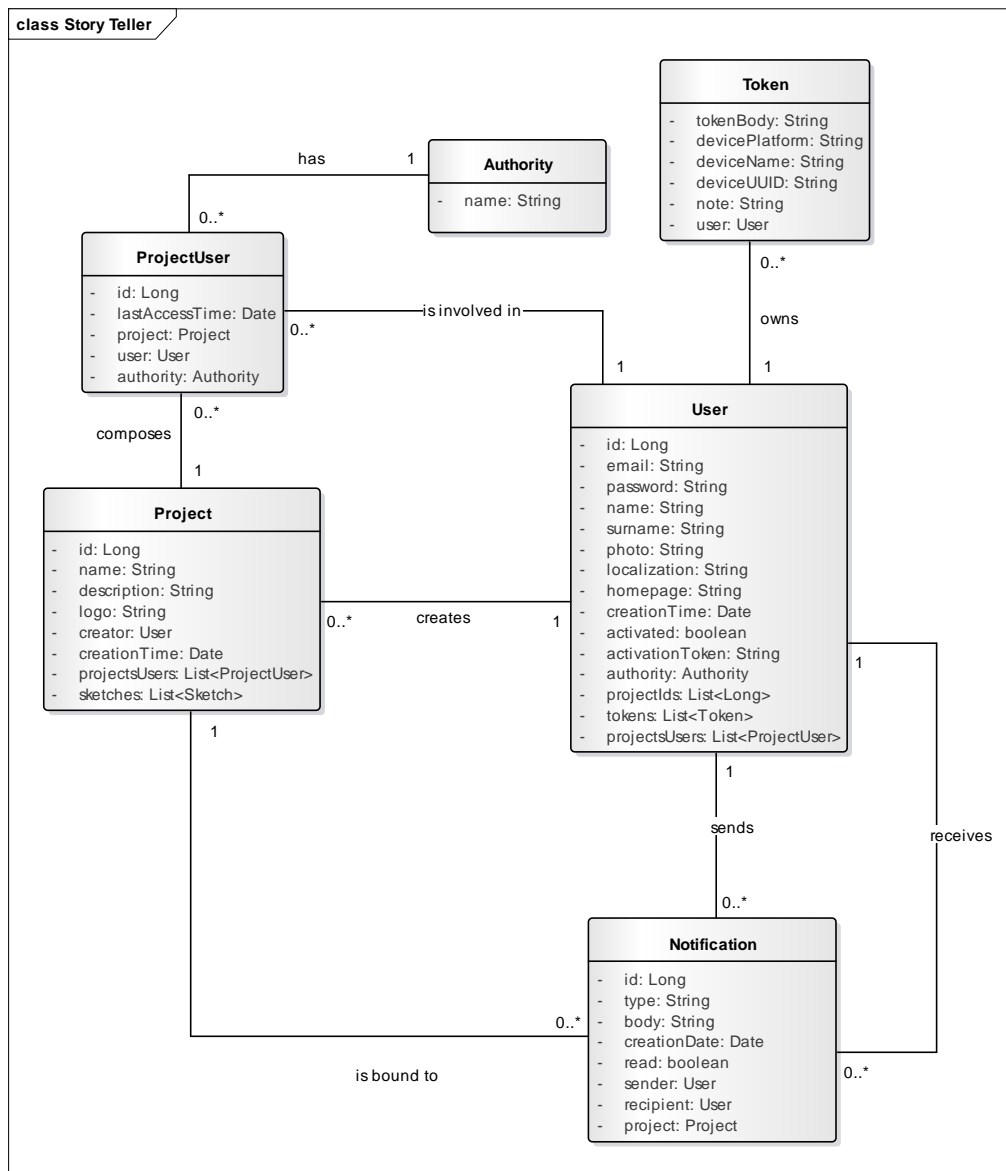
Používateľ bude mať možnosť vyplniť základné informácie o sebe vo svojom profile – meno, priezvisko, homepage a fotografiu a bude mať možnosť zmeniť si heslo. V prípade straty alebo zabudnutia hesla bude mať možnosť odoslania nového hesla na emailovú adresu, pomocou ktorej sa zaregistroval.

Riadenie prístupu bude prebiehať pomocou rolí. Používateľ si bude mať možnosť vybrať hlavnú rolu – rolu analytika alebo zákazníka, v ktorej bude v systéme vystupovať. Používateľ bude mať možnosť túto rolu kedykoľvek upraviť vo svojom profile.

2.2 Návrh a implementácia

Na perzistenciu objektu používateľa reprezentovaného triedou **User** bol použitý OR mapovač s použitím modulu Spring Data rovnako ako na perzistenciu zvyšných tried zobrazených v dátovom modeli na obrázku 1 – **Token** a **Authority**. Trieda **Authority** je určená na riadenie prístupu prostredníctvom rolí, v tomto prípade rolí analytika a zákazníka. Trieda **Token** reprezentuje autentifikačný token pre používateľa – teda nepoužívame HTTP relácie (z angl. sessions). Použitie tokenov zjednodušuje škálovanie aplikácie v budúcnosti, keďže nám umožňuje prechod na bezstavovú (z angl. stateless) autentifikáciu.

K základnému prístupu k údajom v databáze boli použité repozitáre (balík **com.storyteller.repository**) rozhrania, ktoré rozširujú rozhrania modulu Spring Data a umožňujú tvorbu tzv. dopytových metód (z angl. query methods), pomocou ktorých modul Spring Data dokáže vygenerovať dopyty do databázy na základe názvu metódy. Repozitáre taktiež podporujú definíciu dopytov pomocou **@Query** anotácií a JPA syntaxe.



Obrázok 1: Dátový model používateľa

Taktiež boli použité servery (z angl. services) obsahujúce biznis logiku a riadenie databázových transakcií. K dátam sa teda neprístupuje priamo cez repozitáre, ale cez servery, ktoré následne v transakciách pristupujú k dátam. Týmto je dodržaný princíp viacvrstvovej architektúry.

V rámci projektu bolo rozpoznaných viacero typov rolí. Ide o tieto roly:

- Zákazník – rola zákazníka umožňuje prezeranie akceptačných testov a skíc a pridávanie ľudí do projektu, resp. pridelovanie ich rolí
- Analytik – rola analytika umožňuje modifikovanie, prezeranie alebo odstránenie akceptačných testov a skíc a vytváranie nového projektu.
- Používateľ – rola charakteristická pre všeobecného používateľa a opisuje spoločné vlastnosti nadtriedy
- Administrátor – rola administrátora určuje špecifické vlastnosti, ktoré môže vykonávať jedine administrátor

2.2.1 Registrácia

Registrácia používateľa je možná prostredníctvom REST endpointu metódou **POST**, ktorý je zdokumentovaný na adrese <https://api.story-teller.xyz/swagger-ui.html#!/authentication-controller/createNewUserUsingPOST>. Tento endpoint vykonáva kontrolu na duplicitu emailových adries. V prípade správne vyplnených registračných údajov sa vytvorí a uloží do databázy nový objekt triedy **User** s nastavenými atribútmi **email**, **password**, **activated**, **activationToken** a **creationTime**. Na validovanie emailu a hesla boli použité JPA anotácie **@Email** a **@NotNull** s kombináciou anotácie **@Size(min, max)** ošetrojúcou minimálnu a maximálnu dĺžku reťazcov.

Po úspešnom vytvorení nového používateľa je používateľovi zaslaný email prostredníctvom **zoho.com** z adresy **noreply@story-teller.xyz** obsahujúci aktivačný a deaktivovaný odkaz. Túto konfiguráciu je možné zmeniť upravením konfiguračného súboru **application.properties** v **/src/main/resources/**. Používateľ si svoj účet môže aktivovať do 12 hodín od odoslania odkazu, inak mu bude účet pri nasledujúcom pokuse o aktiváciu, resp. v pravidelnom nastavenom intervale, zrušený. Deaktivácia účtu môže prebehnúť, pokiaľ účet nie je aktivovaný.

Aktivácia účtu je možná prostredníctvom REST endpointu metódou **GET**, ktorý je zdokumentovaný na adrese <https://api.story-teller.xyz/swagger-ui.html#!/authentication-controller/activateUserUsingGET>. Na základe atribútu **activationToken** sa vyhľadá používateľ v databáze a zmení sa mu atribút **activated** na **true**. V prípade neskorej aktivácie účtu je používateľ odstránený z databázy.

Deaktivácia účtu je možná prostredníctvom REST endpointu metódou **GET**, ktorý je zdokumentovaný na adrese <https://api.story-teller.xyz/swagger-ui.html#!/authentication-controller/deactivateUserUsingGET>. Na základe atribútu **activationToken** sa vyhľadá používateľ v databáze a odstráni sa z databázy.

2.2.2 Prihlásenie

Prihlásenie bolo implementované s využitím konfigurácie modulu Spring Security a servletových filtrov. Tieto filtre zabezpečujú overenie autentifikačného tokenu, ktorý používateľ pri autentifikácii musí priložiť do HTTP hlavičky **X-AUTH-TOKEN** v HTTP požiadavke. Samotná konfigurácia sa nachádza v triede **SecurityConfiguration** v balíku **com.storyteller.config** spolu s triedou reprezentujúcou Spring Security servis používateľa **UserDetailsService**. Po úspešnom prihlásení odoslaním požiadavky metódou **POST** na endpoint **/login** je používateľovi vygenerovaný autentifikačný token reprezentovaný triedou **Token**, ktorý je uložený v databáze. Tento token je v klientskej časti uložený do **local storage** prehliadača, resp. zariadenia a následne pripájaný do HTTP hlavičky **X-AUTH-TOKEN** v HTTP požiadavke pomocou AngularJS interceptora **authInterceptor.js** v **/components/interceptor/**. Pri odhlásení používateľa (REST endpoint **/logout** metódou **GET**) je tento token odstránený z **local storage** ako aj z databázy.

2.2.3 Úprava údajov v profile

Úprava základných informácií prebieha odoslaním celého objektu používateľa **User** do REST endpointu metódou **PUT**, ktorý je zdokumentovaný na adrese <https://api.story-teller.xyz/swagger-ui.html#!/user-controller/updateUserProfileUsingPUT>. Pre aktuálne prihláseného používateľa sú uložené nové údaje o jeho profile do databázy, konkrétne atribúty **name**, **surname**, **photo**, **localization** a **homepage**.

Získanie aktuálnych dát o prihlásenom používateľovi prebieha odoslaním požiadavky metódou **GET** na REST endpoint, ktorý je zdokumentovaný na adrese <https://api.story-teller.xyz/swagger-ui.html#!/user-controller/getUserProfileUsingGET>. Pre aktuálne prihláseného používateľa sú vybraté všetky údaje o používateľovi z databázy.

Zmena hesla prebieha odoslaním iba starého a nového hesla metódou **POST** na REST endpoint, ktorý prostredníctvom metód v servisných triedach overuje minimálnu dĺžku nového hesla a zhodu starého hesla s aktuálnym heslom (atribút **password** v triede **User**). V prípade úspechu je heslo zmenené na nové heslo, ktoré je zašifrované. Tento REST endpoint je zdokumentovaný na adrese https://api.story-teller.xyz/swagger-ui.html#!/user-controller/changePasswordUsingPOST_1.

2.2.4 Obnova hesla

Obnova hesla je možná prostredníctvom REST endpointu metódou **POST**, ktorý je zdokumentovaný na adrese <https://api.story-teller.xyz/swagger-ui.html#!/authentication-controller/changePasswordUsingPOST>. V prípade, že zadaný email je platný, vygeneruje sa nové 10-miestne heslo použitím java **UUID.randomUUID()** utility a uloží sa do atribútu **password** triedy **User**. Potom je nové heslo poslané používateľovi emailom z adresy **noreply@story-teller.xyz**.

2.2.5 Obnova registračného emailu

Systém poskytuje taktiež možnosť opätovného zaslania emailu s aktivačným a deaktiváčným odkazom. Táto funkcionality je možná prostredníctvom REST endpointu metódou **POST**, ktorý je zdokumentovaný na adrese <https://api.story-teller.xyz/swagger-ui.html#!/authentication-controller/recoverTokenUsingPOST>. Táto metóda najprv skontroluje, či existuje používateľ s daným emailom. Potom nasleduje kontrola aktivácie účtu na základe atribútu **activated**. Ak bol účet používateľa s daným emailom aktivovaný, email nie je nutné poslať. Inak sa generuje nový aktivačný token, ktorý sa uloží do atribútu **activationToken**. Nasleduje poslanie emailu na emailovú adresu používateľa tak ako pri registrácii.

2.2.6 Správa prihlásených zariadení

Správa prihlásených zariadení, ktoré sú reprezentované entitou **Token**, bola implementovaná prostredníctvom viacerých REST endpointov. Výber všetkých zariadení, cez ktoré je používateľ prihlásený v aplikácii, je možný prostredníctvom REST endpoint metódou **GET**, ktorý je zdokumentovaný na adrese: <https://api.story-teller.xyz/swagger-ui.html#!/user-controller/getAllUserTokensUsingGET>. Pre aktuálne prihláseného používateľa sú z databázy vybraté všetky jeho zariadenia.

Pri každom zariadení je možné modifikovať poznámku prostredníctvom REST endpointu metódou **PUT**, ktorý je zdokumentovaný na adrese: <https://api.story-teller.xyz/swagger-ui.html#!/token-controller/changeTokenNoteUsingPUT>. Poznámka je vyhľadaná na základe atribútu **tokenBody** triedy **Token** a následne je zmenená.

Používateľ môže tiež odstrániť zariadenie zo zoznamu prihlásených zariadení prostredníctvom REST endpointu metódou **DELETE**, ktorý je zdokumentovaný na adrese: <https://api.story-teller.xyz/swagger-ui.html#!/token-controller/deleteTokenUsingDELETE>. Zariadenie je vyhľadané na základe atribútu **tokenBody** triedy **Token** a atribútu **email** triedy **User** a následne je odstránené.

2.2.7 Riadenie prístupu

Riadenie prístupu k údajom v databáze je realizované pomocou prístupových anotácií **@PreAuthorize**. Tie sú použité pri každom REST endpointe, ktorý prístupuje ku kritickým dátam, špecifickým pre konkrétnych používateľov. Autorizačné anotácie majú tvar: **@PreAuthorize('@roleSecurityService.authorizationMethod(#arg1, 'constant1'))**. **@roleSecurityService** reprezentuje triedu, v ktorej sú umiestnené autorizačné metódy. **AuthorizationMethod** reprezentuje metódu, na základe ktorej je rozhodnuté, či používateľ má prístup k daným údajom. Autorizačné metódy sú definované v triede **RoleSecurityService**, v balíku **com.storyteller.security**. Autorizačná metóda môže prijať argumenty ako **#arg1**, čo je premenná ako napríklad id projektu, id používateľa. Konštanty ako **'constant1'** sú využívané najmä pre roly, keď priradíme REST endpointu konkrétnu rolu, pre ktorú je prístupný.

2.2.8 Načítanie zoznamu používateľov v projekte

Získanie zoradených používateľov prislúchajúcich aktuálne prehliadanému projektu je možné prostredníctvom REST endpointu metódou **GET**, ktorý je zdokumentovaný na adrese: <https://api.story-teller.xyz/swagger-ui.html#!/user-controller/getUsersInProjectUsingGET>. Na základe identifikátora triedy **Project** sa vyhľadá projekt a pomocou triedy **ProjectUser** sa získa zoznam používateľov v projekte.

2.2.9 Pridanie používateľskej roly v rámci projektu

Pridanie používateľskej roly v rámci projektu je možná prostredníctvom REST endpointu metódou **POST**, ktorý je zdokumentovaný na adrese: <https://api.story-teller.xyz/swagger-ui.html#!/authority-controller/changeUserAuthoritiesUsingPOST>. Na základe identifikátorov tried **User** a **Project** sa vyhľadá objekt typu **ProjectUser** a zmení sa jeho rola. Následne je poslaná notifikácia s nastavenými atribútmi **type**, **recipient**, **project** a **body** používateľovi, ktorého rola v projekte bola zmenená.

3 Modul – Lokalizácia

Úlohou modulu lokalizácia je umožnenie nastavenia a vytvorenie prekladu celého systému bez zásahu do funkcionality systému

3.1 Analýza

Na lokalizáciu je možné použiť funkcie jazyka Java a prekladový modul ng-translate pre AngularJS. Samotné preklady je možné ukladať v properties súboroch s názvom v tvare **messages_LOKALIZÁCIA.properties**, ktorý obsahuje preklady v tvare HODNOTA=KLÚČ.

3.2 Návrh a implementácia

Na implementáciu lokalizácií sme použili prístup popísaný v návrhu. Súborné properties s prekladmi sú umiestnené v **/src/main/resources** a ich obsah si klient preberá cez REST endpoint metódou **GET**, ktorý je zdokumentovaný na adrese <https://api.story-teller.xyz/swagger-ui.html#!/resource-bundle-controller/listUsingGET>. Táto metóda na základe vyžiadaného kódu lokalizácie, napr. sk alebo en, načíta obsah daného properties súboru (v prípade en lokalizácie sa načíta súbor **messages_en.properties**, viď ukážka 1) do objektu typu **Properties**, ktorý je následne serializovaný na JSON a spracovaný modulom ng-translate v klientskej AngularJS aplikácii.

```
myprofile.success.login=You were logged in successfully
myprofile.password.old=Old password
myprofile.password.new=New password
myprofile.password.new.confirm=Confirm new password
myprofile.update=Update!
myprofile.password.update=Change password!
myprofile.password.old.required=Old password is required
myprofile.password.new.required=New password is required
```

Ukážka 1: Časť súboru messages_en.properties

4 Konfigurácia - Server a nasadzovanie

4.1 Konfigurácia

Náš projekt je založený na architektonickom vzore klient-server, a z toho dôvodu je rozdelený na dva podprojekty. Prvým je projekt **backend**, ktorý sa stará o aplikačnú logiku a perzistenciu. Druhým je projekt **client**, ktorý sa stará o vykresľovanie obsahu a poskytuje používateľovi grafické rozhranie. Oba projekty sú nasadené na servery **Ubuntu Server 16.04**.

Náš server poskytuje tieto domény :

- story-teller.xyz – **tímová stránka o projekte**
- api.story-teller.xyz – **backend aplikácia**
- app.story-teller.xyz – **client aplikácia**
- app.story-teller.xyz – **staging client aplikácia**

4.2 Backend

Projekt backend je založený na **Spring framework-u**, a teda je nasadený na aplikačnom serveri **Apache Tomcat**. Tomcat umožňuje nasadzovanie priamo pomocou URL, to znamená, že priamo po vykonaní buildu je nová verzia nasadená.

4.2.1 Závislosti

Na spúšťanie akceptačných testov sú potrebné nasledujúce nástroje a závislosti:

- Python
 - Python vo verzii > 3.5
 - python3-pip
 - behave
 - selenium
- Linux
 - jdk-8
 - postgresql
 - wkhtmltoimage
 - nodejs
 - npm
 - cordova
 - ionic
 - maven
 - storyteller
 - xvfb
 - tmux
 - ffmpeg
 - firefox
 - geckodriver

- chromium
- chromedriver
- feh
- unclutter

Všetky vyššie uvedené závislosti musia byť umiestnené v ceste PATH. Okrem týchto závislostí je potrebné mať umiestnené skripty **test_definitions** v ceste uvedenej v súbore **pom.xml** a súbor **storyteller-wallpaper.jpg** v ceste **/opt/storyteller-wallpaper.jpg** .

4.3 Client

Client je založený na frameworku **Ionic**, ktorý umožňuje zostaviť aplikácie pre mobilné zariadenia, rovnako ako umiestnenie priamo na webe. Client je nasadzovaný pomocou **SCP** kopírovania a následného spustenia skriptu. Skript sa postará o rozbalenie a nahradenie už existujúcich súborov. Ionic poskytuje službu, ktorá spustí aplikáciu ako webovú stránku, na ktorú sa dá následne pristupovať na základe zvoleného mena.

5 Modul – Projekt

Úlohou modulu projekt má byť možnosť vytvárania nových projektov a spravovania existujúcich projektov prihláseným používateľom.

5.1 Analýza

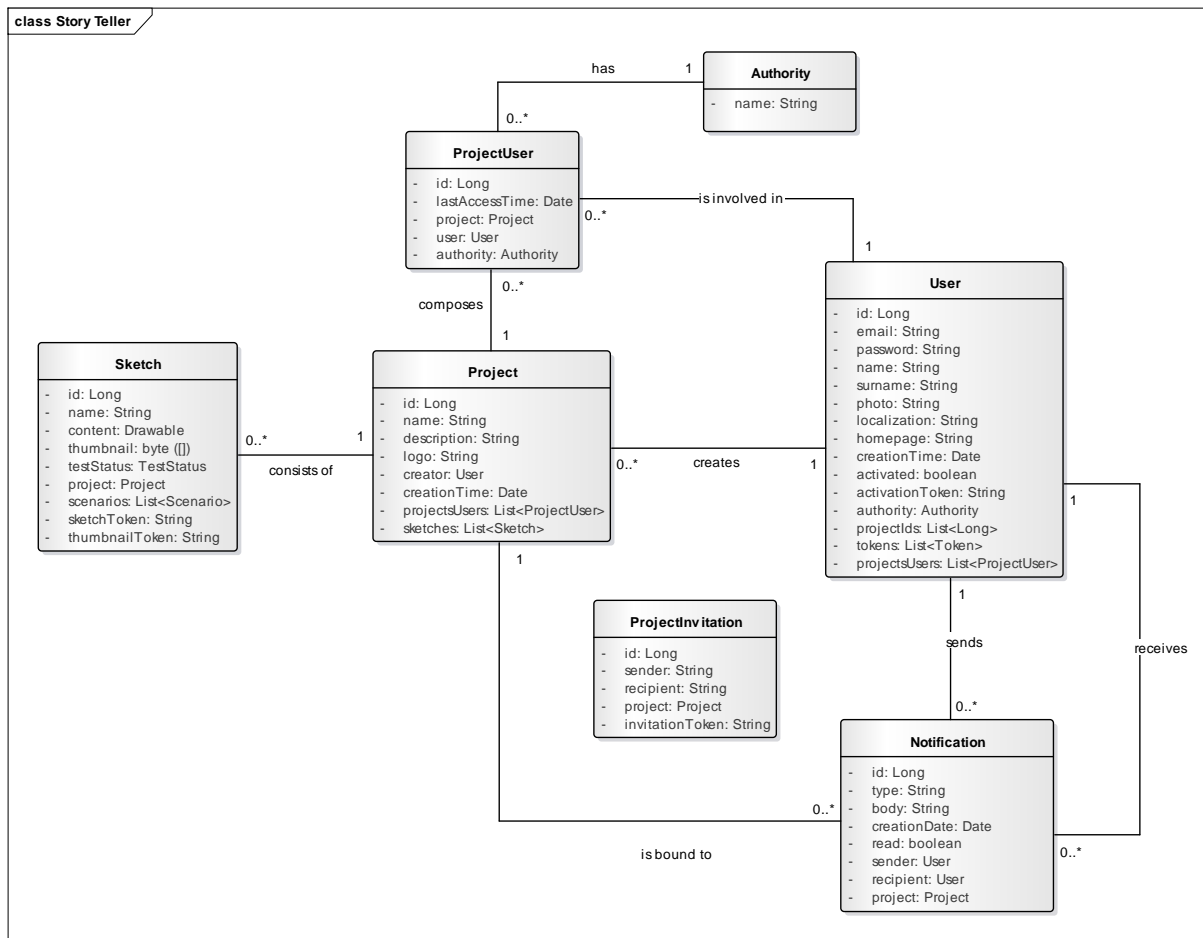
Možnosť vytvárať projekty bude sprístupnená prihláseným používateľom. Pri vytváraní projektu bude používateľ vyplňať príslušný formulár. Vo formulári bude potrebné zadať názov projektu, opis projektu a prípadne vlastné logo projektu. Následne po odoslaní formulára na spracovanie sa k projektu priradí jeho tvorca – čiže automaticky je tomuto projektu priradený ďalší atribút vyjadrujúci údaje o používateľovi, ktorý projekt vytvoril.

5.2 Návrh a implementácia

Na perzistenciu objektu projektu reprezentovaného triedou **Project** bol použitý OR mapovač s použitím modulu Spring Data.

K základnému prístupu k údajom v databáze boli použité repozitáre rozhrania, ktoré rozširujú rozhrania modulu Spring Data a umožňujú tvorbu tzv. dopytových metód (z angl. query methods), pomocou ktorých modul Spring Data dokáže vygenerovať dopyty do databázy na základe názvu metódy. Repozitáre taktiež podporujú definíciu dopytov pomocou **@Query** anotácií a JPA syntaxe.

Taktiež boli použité servisy (z angl. services) obsahujúce biznis logiku a riadenie databázových transakcií. K dátam sa teda neprístupuje priamo cez repozitáre, ale cez servisy, ktoré následne v transakciách pristupujú k dátam. Týmto je dodržaný princíp viacvrstvovej architektúry.



Obrázok 2: Dátový model projektu

5.2.1 Vytvorenie projektu

Vytvorenie projektu je možné prostredníctvom REST endpointu metódou **POST**, ktorý je zdokumentovaný na adrese <https://api.story-teller.xyz/swagger-ui.html#!/project-controller/createProjectUsingPOST>. Nový projekt s nastavenými atribútmi **name**, **description**, **logo**, **creator** a **creationTime** je uložený do databázy. Na validovanie názvu a opisu projektu bola použitá JPA anotácia **@NotNull**.

5.2.2 Načítanie zoznamu projektov prihláseného používateľa

Získanie zoradených projektov prislúchajúcich aktuálne prihlásenému používateľovi je možné prostredníctvom REST endpointu metódou **GET**, ktorý je zdokumentovaný na adrese <https://api.story-teller.xyz/swagger-ui.html#!/project-controller/getUserProjectsUsingGET>. Projekty pre aktuálne prihláseného používateľa sú vyhľadane na základe atribútu **email** triedy **User**, pričom je využité stránkovanie.

5.2.3 Načítanie informácií o konkrétnom projekte

Načítanie konkrétneho projektu je realizované načítaním jeho objektu podľa identifikátora. To je možné prostredníctvom REST endpointu metódou **GET**, ktorý je zdokumentovaný na adrese <https://api.story-teller.xyz/swagger-ui.html#!/project-controller/getProjectByIdUsingGET>.

Projekt je vyhľadovaný na základe identifikátora triedy **Project**. Na strane klienta je tento endpoint volaný už pri zmene obrazovky (stavu) na obrazovku s prehľadom informácií o konkrétnom projekte.

5.2.4 Úprava údajov o projekte

Načítanie konkrétneho projektu je realizované prostredníctvom REST endpointu metódou **PUT**, ktorý je zdokumentovaný na adrese <https://api.story-teller.xyz/swagger-ui.html#!/project-controller/updateProjectUsingPUT>. Projekt je vyhľadovaný na základe jeho identifikátora a sú upravené jeho atribúty **name**, **description** a **logo**.

5.2.5 Priradenie roly používateľovi projektu

Riadenie prístupu ku projektu je založené na rolách. Tie sú používateľovi priradené pomocou REST endpointov, ktoré najprv načítajú všetky dostupné roly, a následne vyznačia aktuálne priradenú. Načítanie všetkých rolí je možné prostredníctvom **REST** endpointu metódou **GET**, ktorý je zdokumentovaný na adrese <https://api.story-teller.xyz/swagger-ui.html#!/authority-controller/getAllRolesUsingGET>. Načítanie aktuálnej roly používateľa je možné prostredníctvom **REST** endpointu metódou **GET**, ktorý je zdokumentovaný na adrese <https://api.story-teller.xyz/swagger-ui.html#!/authority-controller/getUserProjectAuthorityUsingGET>. Najprv sa vyhľadajú objekty triedy **ProjectUser** na základe identifikátorov tried **User** a **Project** a nová rola sa vyhľadá na základe atribútu **name** triedy **Authority**.

5.2.6 Pozvanie používateľa do projektu

6 Používateľské rozhranie

Používateľské rozhranie je realizované v projekte **client**. Ten je založený na technológii **Ionic**, ktorá umožňuje zostavovanie mobilných aplikácií. Framework Ionic ponúka vlastné elementy a štýly na použitie, tie však nie sú príliš vhodné pre webovú stránku. Z toho dôvodu je použitý mobile-first framework **Bootstrap**. V tomto projekte je použitá knižnica **Angular-UI**, nakoľko Ionic využíva AngularJS, ktorý by mohol spôsobovať konflikty s klasickým jQuery.

Bootstrap je responzívny framework, umožňujúci efektívne zobrazenie webovej stránky na rôznych zariadeniach, s rôznym rozlíšením. Dokáže automaticky prispôsobiť rozloženie grafických elementov vzhľadom na aktuálnu veľkosť obrazovky. Bootstrap využíva triedy na špecifikáciu štýlu pre príslušný element. V projekte sú využité len štandardné triedy, ktoré využíva bootstrap. Všetky ostatné štýly ako odsadenie, posunutie, a iné, ktoré sú špecifické pre konkrétny element sú definované priamo v HTML. Bootstrap štýly sú definované v súbore **custom.css** a následne minifikované v súbore **custom.min.css**. Iba minifikovaný súbor je importovaný do projektu, a teda pri každej zmene je nutné súbor znovu minifikovať.

Bootstrap dokumentácia : <http://getbootstrap.com/getting-started/http://getbootstrap.com/getting-started/>

7 Modul – Notifikácie

Úlohou modulu notifikácie je poskytnutie dôležitých informácií používateľovi v reálnom čase, bez nutnosti manuálne vyžiadania.

7.1 Analýza

Na prenos notifikácií je možné použiť websockety s modelom publish – subscribe. Z dôvodu viacerých typov notifikácií je nutné navrhnúť šablónový podmodul, ktorý umožní dynamické nastavenie šablóny konkrétnej notifikácie v závislosti od jej typu, napr. či daná notifikácia bude poskytovať možnosť potvrdenia. Použitie websocketov taktiež umožňuje budúcu implementáciu chatovacieho modulu a modulu paralelného upravovania skíc.

7.2 Návrh a implementácia

Samotná notifikácia je reprezentovaná pomocou triedy **Notification** z balíka **com.storyteller.model**. Databázové operácie nad ňou sú realizované prostredníctvom repozitárov a servisov, podobne ako je to v prípade modulu používateľa. Prístup k notifikáciám z klientskej aplikácie je realizovaný prostredníctvom REST endpointov definovaných v triede **NotificationController** z balíka **com.storyteller.controller**.

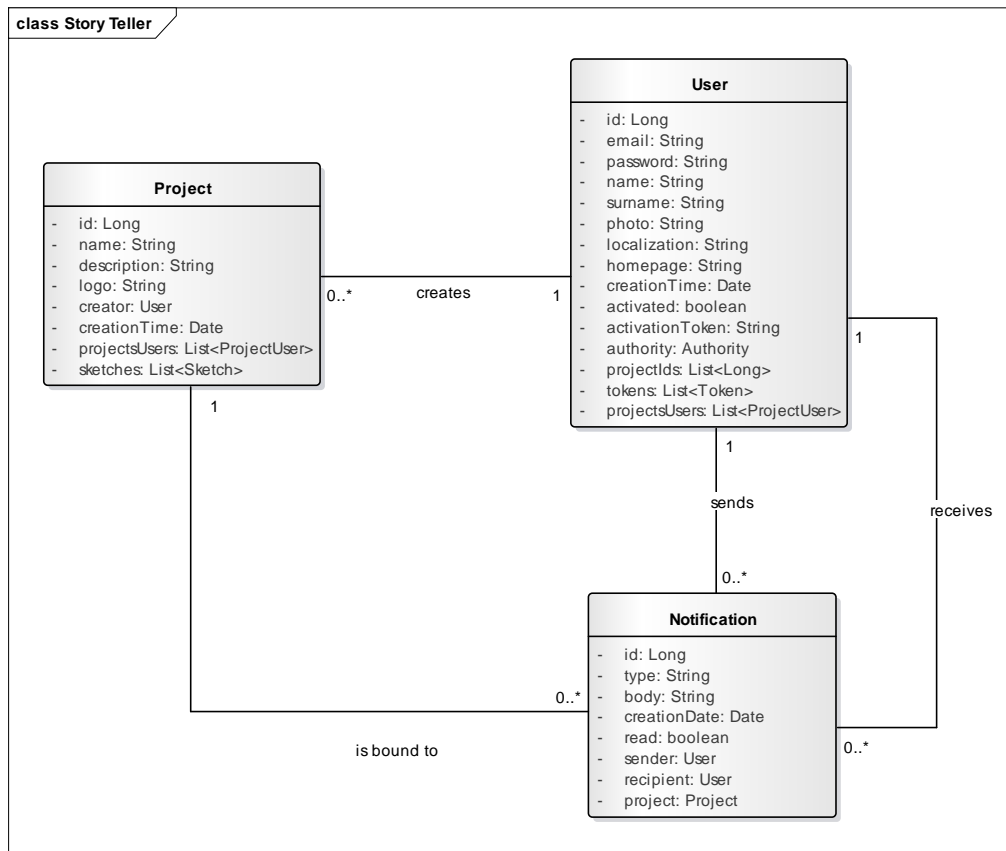
7.2.1 Odosielanie notifikácií

Notifikácie sú odosielané z backendovej strany projektu pomocou protokolu **STOMP**, ktorý je podporovaný rámcom Spring, ktorý používame. Notifikácie sú odosielané prostredníctvom volania metódy **sendNotification(Notification notification)**, do ktorej vstupuje už vyplnený objekt notifikácie vrátane príjemcu. Táto metóda volá Spring metódu **convertAndSendToUser(String user, String destination, Object payload)**. Argument **user** predstavuje identifikátor používateľa, v našom prípade jeho email. Argument **destination** predstavuje cestu websocketového publish kanálu, do ktorého sa má notifikácia zverejniť. Táto cesta musí obsahovať prefix **'/topic'** z dôvodu odlíšenia publish ciest od subscribe. Argument **payload** predstavuje samotný obsah správy, t. j. notifikáciu. Táto metóda je volaná prostredníctvom objektu **SimpMessageSendingOperations**, ktorý je súčasťou Spring modulu pre websockety. Autentifikácia používateľa pre websockety je realizovaná pomocou objektu **AuthenticationPrincipal**, ktorý je vytvorený pred samotným odoslaním správy do konkrétneho kanála, t. j. po prijatí dispatcherom.

7.2.2 Prijímanie notifikácií

Notifikácie sú prijímané s využitím knižnice **SockJS** a javascriptového modulu pre protokol **STOMP**. Knižnicu **SockJS** bolo potrebné upraviť z dôvodu **CORS** mechanizmu – knižnica neposkytuje možnosť vypnúť atribút **withCredentials** pre **XMLHttpRequest** volania a rovnako sa neriadi globálnymi nastaveniami rámca AngularJS. Bolo preto nutné nastaviť atribút **withCredentials** na **false**. Prijímanie notifikácií zabezpečuje subscribe na kanál **'/user/topic/notifications'** v metóde **onConnect(frame)** komponentu **socketService**. Tento

komponent predstavuje wrapper nad knižnicou **SockJS**, ktorý bolo nutné vytvoriť z dôvodu chýbajúcej podpory autentifikácie prostredníctvom tokenu.



Obrázok 3: Dátový model notifikácií

Po prijatí správy je notifikácia pretransformovaná z JSON reťazca na objekt, následne je zobrazená a je broadcastnutý event '\$newNotification', čo umožňuje spracovanie notifikácie aj ďalším modulom, ako je napríklad navigačný modul. Tento modul inkrementuje počítadlo zatiaľ nevidených notifikácií v navigačnom paneli nachádzajúcom sa na vrchu aplikácie a indikuje novú notifikáciu animáciou zatrasenia ikony notifikácií.

7.2.3 Zobrazenie notifikácií

Notifikácie sú zobrazované ako **toast popup** prostredníctvom modulu **ngToaster**. Na zobrazenie sa používa metóda **showNotification(notification)** komponentu **notificationService**, do ktorej ako argument prichádza vyplnená notifikácia určená na zobrazenie.

Taktiež je možné prezerat' notifikácie cez otváracie popup okno, ktoré sa zobrazí kliknutím na ikonu notifikácií v navigačnom paneli – volanie metódy **showPreviewNotifications()** z komponentu **navigationController**. Táto ikona taktiež zobrazuje počet neprečítaných notifikácií, čo je realizované volaním metódy **getUnreadNotificationsCount()** z komponentu **navigationController**. Po kliknutí na ikonu sa zobrazí päť najnovších notifikácií, ktoré sú načítané volaním metódy **getBriefNotifications()** z komponentu **navigationController**.

K starším notifikáciám je možné pristupovať cez tlačidlá pre stránkovanie. Stránkovanie je realizované volaním metódy **getNotificationPage(page)** z komponentu **navigationController** do ktorej ako argument vstupuje číslo strany, ktorá sa má zobrazit'.

Označovanie notifikácií za videné prebieha tak, že notifikácie, ktoré boli zobrazené cez otváracie popup okno sú odoslané backendovej aplikácii, ktorá im zmení príznak **read** na true. Toto je realizované metódou **setNotificationsAreRead()** z komponentu **navigationController**. Po úspešnom odoslaní je tento príznak nastavený aj daným notifikáciám v klientskej časti a počítadlo neprečítaných notifikácií je aktualizované.

8 Modul – Skice

Možnosť vytvárať skice je jednou z najdôležitejších funkcionalít v tomto projekte. Pomocou skíc je možné prototypovať používateľské rozhranie, na základe ktorého budú neskôr generované akceptačné testy.

Používateľské rozhranie na webových stránkach býva zložené z viacerých komponentov, ktoré slúžia ako ovládacie prvky, alebo ako tzv. kontajnery, do ktorých sa potom vkladajú iné prvky.

Dôležitým aspektom pri skladaní rozhrania pomocou komponentov je určovanie pozícií jednotlivých prvkov. Prvky môžu mať absolútnu pozíciu, kde ich pozícia je počítaná od kraja plátna, na ktorom sú umiestnené, alebo môžu mať relatívne pozície, pri ktorých sa pozície počítajú vzhľadom na 'rodiča'.

8.1 Analýza

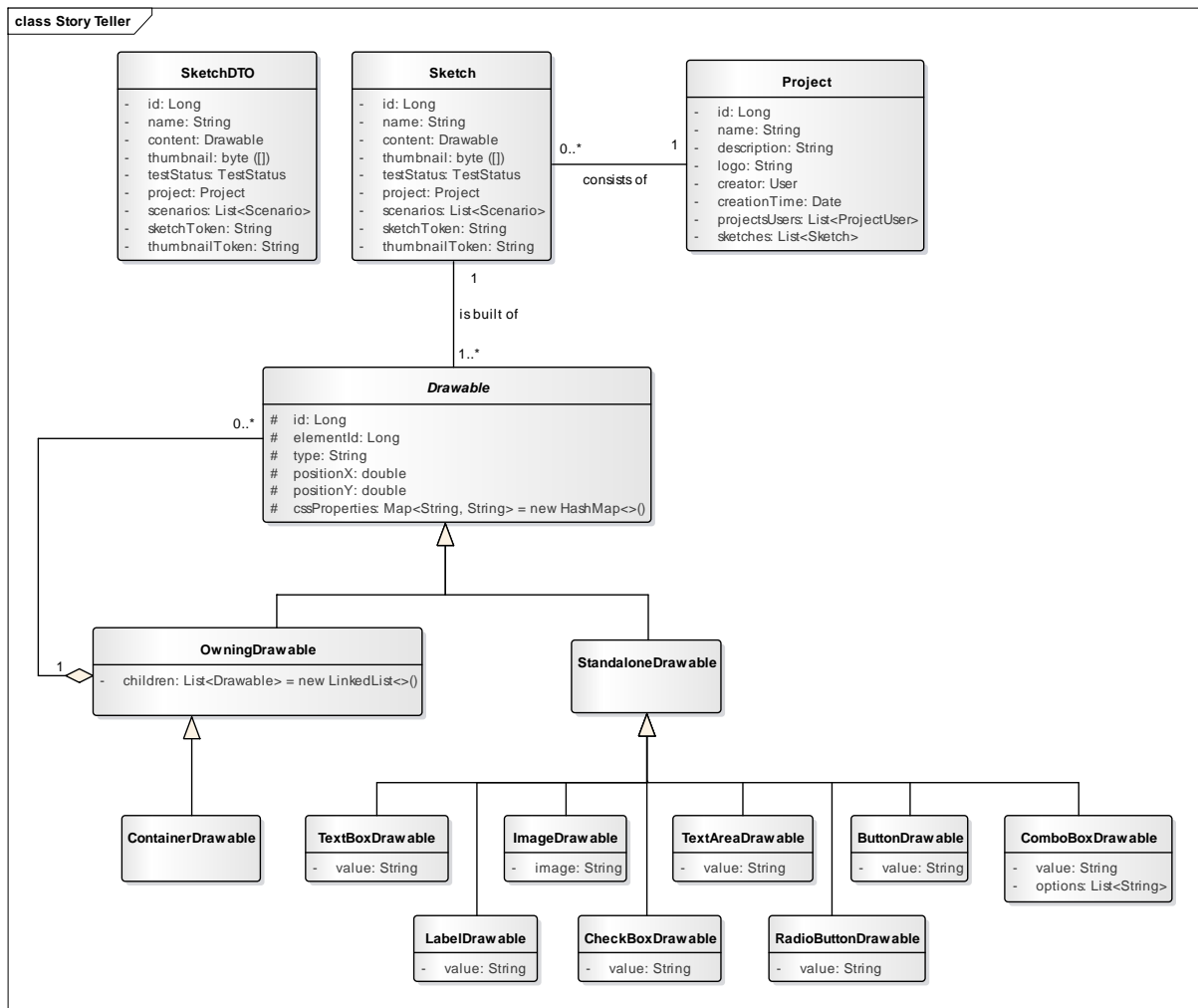
Pre možnosť tvorby skice je nutné vytvoriť používateľské rozhranie, ktoré bude poskytovať plátno a panel nástrojov, z ktorého sa budú môcť pridávať prvky na plátno.

Prvky používateľského rozhrania majú vo všeobecnosti veľkú časť vlastností spoločných. Ide napríklad o ich pozíciu na plátne, farbu, veľkosť, a mnoho iných. Najefektívnejším spôsobom je vytvoriť všeobecnú abstraktnú triedu, ktorá bude tieto vlastnosti obsahovať, a konkrétne prvky používateľského rozhrania budú od tejto abstraktnej triedy dediť, pričom ich vlastné atribúty budú len tie, ktoré sú špecifické pre konkrétny prvok.

8.2 Návrh a implementácia

8.2.1 Perzistencia

Dátový model skice a jej komponentov je zobrazený na obrázku 4. Trieda **Sketch** predstavuje samotnú skicu, pričom atribút **thumbnail** predstavuje generovaný náhľad a atribút **content** jej obsah, t. j. zloženie z komponentov. Komponenty skice sú navrhnuté ako vzor composite, pričom komponent predstavuje abstraktná trieda **Drawable**. Od tejto triedy dedia triedy **OwningDrawable**, ktorá predstavuje komponent, ktorý môže mať ďalšie vnorené prvky, čo v kontexte vzoru composite predstavuje časť composite. Príkladom komponentu, ktorý dedí od tejto triedy, je komponent kontajner. Ďalšou dediacou triedou je trieda **StandaloneDrawable**, ktorá predstavuje samostatný komponent, t. j. komponent, ktorý nemôže obsahovať ďalšie vnorené komponenty. Príkladom je komponent textové pole, ktorý je reprezentovaný triedou **TextBoxDrawable** so špecifickým atribútom **value**, čiže jeho hodnotou. Všetky tieto triedy obsahujú metódu na vygenerovanie inline CSS s názvom **getInlineCss()**, ktorá vracia inline CSS ako reťazec na základe obsahu poľa **cssProperties**. Taktiež obsahujú metódu **toHtml()**, ktorá vracia html reprezentáciu jednotlivého prvku ako reťazec, čo znamená, že v prípade zavolania tejto metódy na inštanciu triedy **Drawable** s typom (pole type) **ROOT_CONTAINER** sa vytvorí html reprezentácia celej skice. Takéto volanie je použité napríklad pri generovaní náhľadu skice. Trieda **SketchDTO** predstavuje pomocnú triedu, ktorá má rovnaké atribúty ako trieda **Sketch**.



Obrázok 4: Dátový model skice a komponentov

8.2.2 Vytvorenie skice

Vytvorenie skice je možné prostredníctvom REST endpointu metódou **POST**, ktorý je zdokumentovaný na adrese <https://api.story-teller.xyz/swagger-ui.html#!/sketch-controller/saveSketchUsingPOST>. Skica a jej komponenty sú uložené do atribútov **name**, **content**, **thumbnail**, **scenarios**, **testStatus** triedy **Sketch** a projektu, ktorý je vyhľadaný na základe jeho identifikátora, je priradená vytvorená skica. Súčasťou vytvorenia skice je automatické generovanie náhľadu vo forme obrázku. Na to je použitá knižnica **wkhtmltopdf**, resp. jej časť **wkhtmltoimage**. Keďže táto knižnica nie je priamo kompatibilná s Javou, bol vytvorený (upravený) wrapper, ktorý túto knižnicu volá prostredníctvom vytvorenia nového procesu a dočasného html súboru so skicou, ktorý je špecifikovaný ako prvý argument volania príkazu **wkhtmltoimage**. Výstup (vyrenderovaný obrázok) je presmerovaný cez deskriptor do bajtového poľa v Jave.

8.2.3 Získanie skice

Získanie skice podľa jej identifikátora je možné prostredníctvom REST endpointu metódou **GET**, ktorý je zdokumentovaný na adrese <https://api.story-teller.xyz/swagger->

[ui.html#!/sketch-controller/getSketchUsingGET](#). Táto metóda vyberie skicu z databázy na základe identifikátora aktuálneho projektu.

8.2.4 Úprava skice

Úprava skice prebieha odoslaním celého objektu skice a identifikátora skice do REST endpointu metódou **PUT**, ktorý je zdokumentovaný na adrese <https://api.story-teller.xyz/swagger-ui.html#!/sketch-controller/updateSketchUsingPUT>. Získa sa skica na základe jej identifikátora a nové informácie o skici sú uložené do databázy, ide o atribúty **name**, **content**, **thumbnail**, **scenarios**, **testStatus**.

8.2.5 Odstránenie skice

Systém poskytuje tiež možnosť odstránenia skice, čo bolo implementované prostredníctvom REST endpointu metódou **DELETE**, ktorý je zdokumentovaný na adrese <https://api.story-teller.xyz/swagger-ui.html#!/sketch-controller/deleteSketchUsingDELETE>. Najprv sa získa skica na základe jej identifikátora a nájdená skica je odstránená z databázy.

8.2.6 Načítanie skíc projektu

Načítanie skíc projektu bolo implementované prostredníctvom REST endpointu metódou **GET**, ktorý je zdokumentovaný na adrese <https://api.story-teller.xyz/swagger-ui.html#!/sketch-controller/getSketchesByProjectIdUsingGET>. Získa sa zoznam skíc na základe identifikátora projektu.

8.2.7 Export layoutu skice

Export layoutu skice bolo implementované prostredníctvom REST endpointu metódou **POST**, ktorý je zdokumentovaný na adrese <https://api.story-teller.xyz/swagger-ui.html#!/sketch-controller/getSketchLayoutForTypeUsingPOST>. Na základe vstupného parametra, ktorý označuje typ layoutu, sa vygeneruje HTML alebo Android layout pre skicu vyhladanú na základe jej identifikátora. Vygenerovaný layout je automaticky stiahnutý pre klienta.

8.2.8 Načítanie náhľadu skice

Načítanie náhľadu skice je možné prostredníctvom REST endpointu metódou **GET**, ktorý je zdokumentovaný na adrese <https://api.story-teller.xyz/swagger-ui.html#!/sketch-controller/getSketchThumbnailUsingGET>. Na základe identifikátora je vyhladaná skica a vráti sa objekt typu **SketchDTO** s nastavenými atribútmi **id** a **thumbnail**.

8.2.9 Zdieľanie skice

Zdieľanie skice je možné pomocou vygenerovaných odkazov na zdieľaný náhľad skice alebo zdieľanú skicu. Táto funkcionálna bola implementovaná prostredníctvom REST endpointu metódou **POST**, ktorý je zdokumentovaný na adrese <https://api.story-teller.xyz/swagger-ui.html#!/sketch-controller/generateRouteUsingPOST>. Pre zdieľanie skice sa získa skica na základe jej identifikátora a je vrátená objekt typu **SketchDTO** s nastavenými vygenerovanými

atribútmi **sketchToken** alebo **thumbnailToken**, podľa toho, či ide o zdieľanú skicu alebo zdieľaný náhľad skice.

8.2.10 Načítanie zdieľaného náhľadu skice

System poskytuje tiež možnosť načítania zdieľaného náhľadu skice, čo bolo implementované prostredníctvom REST endpointu metódou **GET**, ktorý je zdokumentovaný na adrese <https://api.story-teller.xyz/swagger-ui.html#!/sketch-controller/getSharedThumbnailUsingGET>. Skica je načítaná na základe atribútu **thumbnailToken** a vráti sa objekt typu **SketchDTO** s nastavenými atribútmi **name** a **thumbnail**.

8.2.11 Načítanie zdieľanej skice

System poskytuje tiež možnosť načítania zdieľanej skice, čo bolo implementované prostredníctvom REST endpointu metódou **GET**, ktorý je zdokumentovaný na adrese <https://api.story-teller.xyz/swagger-ui.html#!/sketch-controller/getSharedSketchUsingGET>. Skica je načítaná na základe atribútu **thumbnailToken** a vráti sa objekt typu **SketchDTO** s nastavenými atribútmi **name**, **content**, **scenarios**, **thumbnailToken** a **sketchToken**.

9 Modul – Scenáre a testy

Na vykonávanie akceptačných testov je nutné mať k dispozícii okrem skice aj scenár testov. Scenáre testov pozostávajú z individuálnych krokov, ktoré na seba často nadväzujú a v prípade akceptačných testov týkajúcich sa softvérových systémov tieto kroky predstavujú akcie používateľa. Úlohou scenárov je teda obohatiť vytvorenú skicu o akcie a transformovať statickú skicu do vykonateľnej podoby.

9.1 Analýza

Scenáre pozostávajú z jednotlivých krokov – akcií. Tieto akcie sú naviazané na vizuálne elementy skice a teda mali by byť pridávané súčasne s vizuálnymi prvkami. Vykonávanie scenárov by malo používateľovi poskytovať informáciu o stave vykonávania a upozorniť ho v prípade, že scenár nemohol byť úspešne vykonaný. Toto je možné zabezpečiť buď jednoduchým stavovým indikátorom, alebo videozáznamom z testovania.

9.2 Návrh a implementácia

9.2.1 Perzistencia

Dátový model skice a jej komponentov je zobrazený na obrázku 05. Trieda **Scenario** predstavuje samotný scenár, pričom atribút **gherkinFeatureContent** predstavuje vyjadrenie scenáru v notácii Gherkin. Tento atribút je tranzitívny a teda nie je perzistovaný do databázy – je využitý len na prenos tejto informácie na stranu klienta. Scenár pozostáva z krokov, ktoré sú reprezentované triedou **ScenarioStep**. Táto trieda pozostáva z vyjadrenia kroku a typu kroku – akcie vhodnej pre notáciu Gherkin. Taktiež obsahuje informáciu o poradí v scenári. Trieda **StepState** sa vzťahuje na triedu **ScenarioStep** a vyjadruje stav pre krok scenára. Trieda **TestResult** predstavuje informáciu o vykonávaní akceptačných testov. Okrem iného obsahuje stav testu, reprezentovaný pomocou vymenovaného typu **TestStatus**, dĺžku trvania a cestu k súboru s videozáznamom testu. Taktiež obsahuje informáciu o webovom prehliadači a rozlíšení, v ktorom bol daný test vykonávaný. Dostupné webové prehliadače sú reprezentované triedou **Browser** a dostupné rozlíšenia sú reprezentované triedou **Resolution**.

9.2.2 Vytvorenie, odstránenie a získanie scenáru

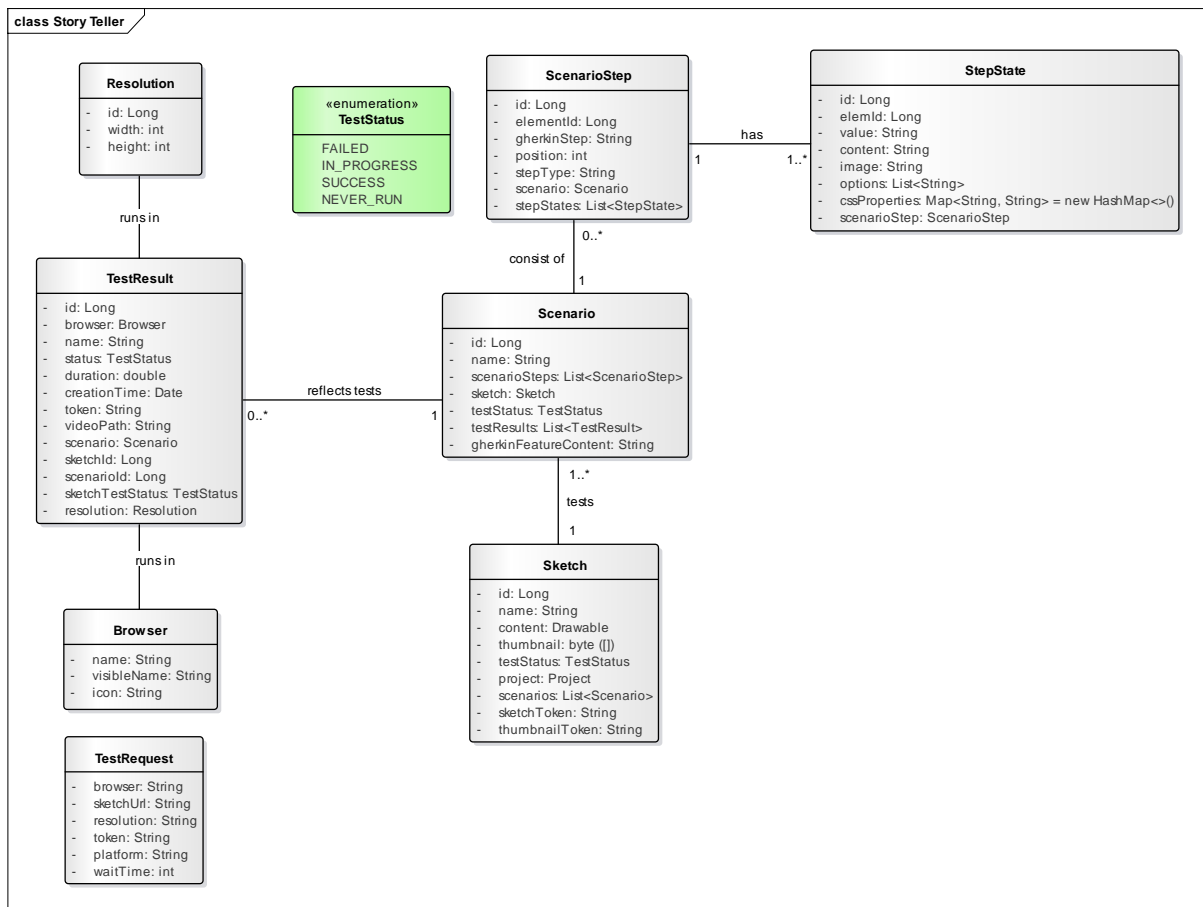
Vytvorenie scenáru, a teda jeho krokov, prebieha súčasne s vytváraním skice. Kroky sú pridávané v závislosti od pridaného vizuálneho prvku skice. Každému prvku je možné nastaviť meno a definovať obmedzenie na jeho vlastnosť. Obmedzenie sa vždy začína znakom „<“ a končí znakom „>“ a musí byť v presnom formáte. Implementované sú tieto obmedzenia:

- <not-empty> - vlastnosť prvku nesmie byť prázdna,
- <length>N> - dĺžka vlastnosti prvku musí byť väčšia ako N, kde N je kladné číslo, namiesto znaku „>“ môžu byť použité znaky „<“ a „=“,
- <email> - vlastnosť prvku musí mať tvar emailu.

Odstránenie krokov prebieha taktiež súčasne s odstránením vizuálnych prvkov skice – ich presunutím z plátna na ikonu zobrazujúcu kôš. Následná perzistencia scenáru je

zabezpečená spolu s perzistenciou skice a teda prostredníctvom REST endpointu metódou **POST**, ktorý je zdokumentovaný na adrese <https://api.story-teller.xyz/swagger-ui.html#!/sketch-controller/saveSketchUsingPOST>, a pri aktualizácii prostredníctvom REST endpointu metódou **PUT**, ktorý je zdokumentovaný na adrese <https://api.story-teller.xyz/swagger-ui.html#!/sketch-controller/updateSketchUsingPUT>. Ich funkcionálnosť je popísaná v časti 8.2.

Získanie scenáru je možné pri získaní skice a prostredníctvom REST endpointu metódou **GET**, ktorý je zdokumentovaný na adrese <https://api.story-teller.xyz/swagger-ui.html#!/scenario-controller/getScenarioByIdUsingGET>. Na základe identifikátora scenára je scenár vyhľadaný a vrátený.



Obrázok 5: Dátový model scenárov a testov

Získanie Gherkin notácie scenára je možné prostredníctvom REST endpointu metódou **GET**, ktorý je zdokumentovaný na adrese <https://api.story-teller.xyz/swagger-ui.html#!/scenario-controller/getGherkinFeatureByScenarioIdUsingGET>. Skica je určená svojím identifikátorom a sú vygenerované a vrátené Gherkin notácie.

9.2.3 Vykonávanie testov

Spustenie testu pre konkrétny scenár je možné prostredníctvom REST endpointu metódou **POST**, pričom takýto test je spustený v prednastavenej webovej prehliadači PhantomJS a prebieha na webovej adrese zadanej pomocou parametra **url**. Tento endpoint je

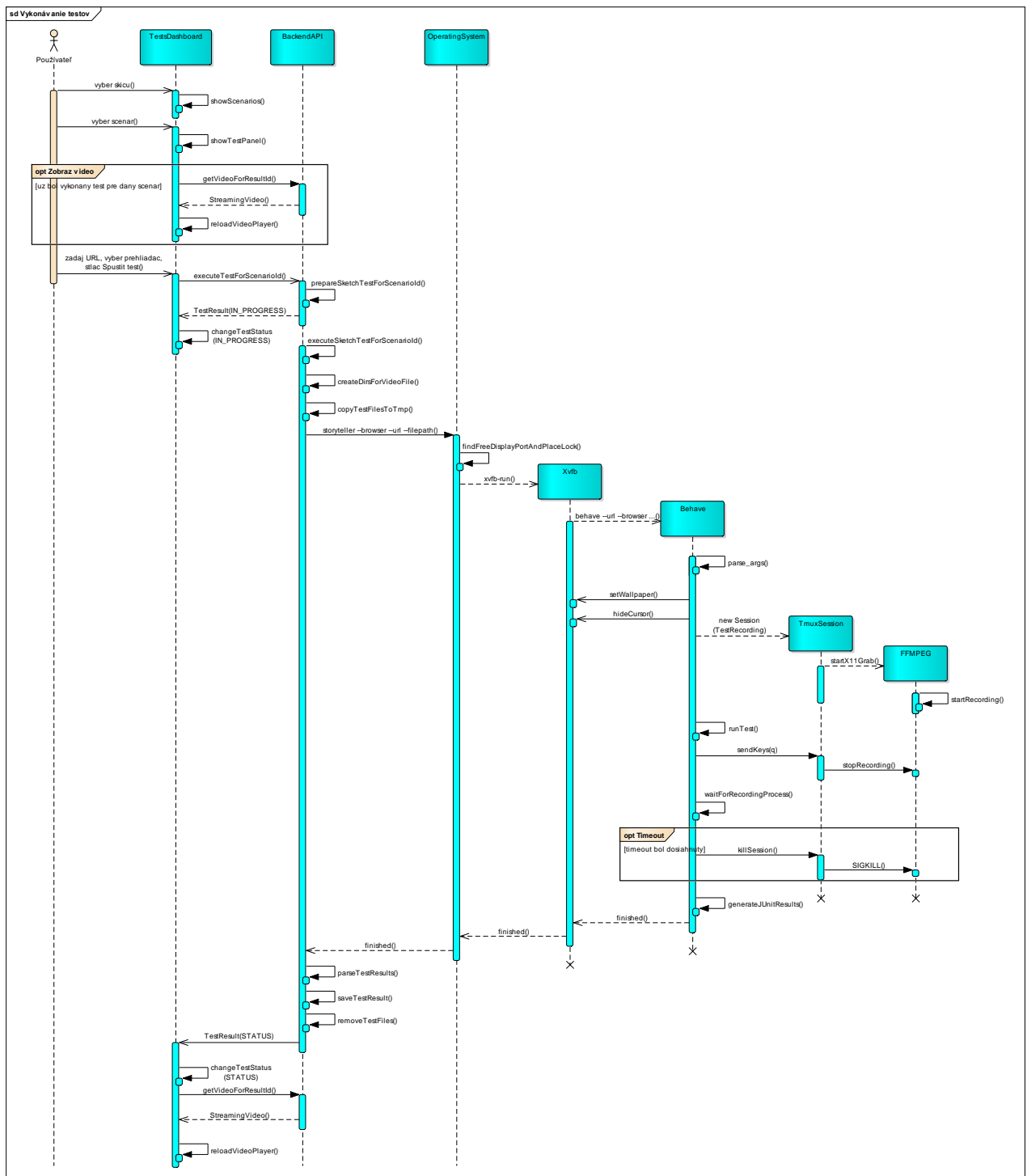
zdokumentovaný na adrese <https://api.story-teller.xyz/swagger-ui.html#!/scenario-controller/executeTestForScenarioIdWithDefaultBrowserUsingPOST>.

Spustenie testu v inom ako prednastavenom prehliadači je možné prostredníctvom REST endpointu metódou **POST**, pričom parametre požiadavky – **browser**, **url** a **resolution** špecifikujú použitý webový prehliadač, webovú adresu a rozlíšenie. Tento endpoint je zdokumentovaný na adrese <https://api.story-teller.xyz/swagger-ui.html#!/scenario-controller/executeTestForScenarioIdWithDefaultBrowserUsingPOST>. Dostupné webové prehliadače sú načítané prostredníctvom REST endpointu metódou **GET**, ktorý je zdokumentovaný na adrese <https://api.story-teller.xyz/swagger-ui.html#!/test-controller/getAvailableBrowsersUsingGET>. Dostupné rozlíšenia sú načítané prostredníctvom REST endpointu metódou **GET**, ktorý je zdokumentovaný na adrese <https://api.story-teller.xyz/swagger-ui.html#!/test-controller/getAllResolutionsUsingGET>.

Priebeh vykonávania testov je opísaný na obrázku 6. Pred samotným spustením testu sa predpokladá, že používateľ je prihlásený, má vybraný projekt a nachádza sa v časti **akceptačné testy**.

Po vybratí webového prehliadača, zvolení webovej adresy stránky na testovanie a stlačení tlačidla na spustenie testu sa na stranu servera odošle požiadavka na začatie testu. Následne sa vytvorí inštancia objektu **TestRequest**, ktorá sa odošle na jeden z uzlov zodpovedných za vykonávanie modulárnych testov. Tento uzol na základe tokenu stiahne testovacie súbory, ktoré mu pripraví server v zip archíve a spustí test v docker kontajneri. Na strane servera pripraví nová inštancia objektu **TestResult**, vyplní sa jej polia, uloží sa do databázy a nastaví sa stav daného testu na **IN PROGRESS**. Samotný test prebieha v izolovanom docker kontajneri, ktorý zavolá skript **storyteller** s parametrami a otestuje stiahnuté testovacie súbory. Tento skript následne vytvorí inštanciu virtuálnej **Xvfb** obrazovky, v ktorej spustí testovací nástroj **behave**. Ten následne spustí nahrávanie testu pomocou nástroja **ffmpeg** v samostatnej **tmux** relácii. Po vykonaní testu **behave** odošle relácii **tmux** pokyn na zastavenie nahrávania videa. Pokiaľ toto nahrávanie neskončí do určitého času po odoslaní pokynu na zastavenie, nástroj **behave** odošle pokyn relácii **tmux** na vynútenie zastavenia nahrávania – relácia **tmux** vtedy odošle nástroju **ffmpeg** signál **KILL**.

Po ukončení nahrávania nástroj **behave** vygeneruje testovú správu v podobe JUnit správy, ktorý odošle hlavný uzol spolu s videozáznamom testu na server, ktorý tieto súbory spracuje a uloží v podobe aktualizovaného objektu **TestResult**. Tento objekt následne odošle na klientsku časť. Klientska časť tento objekt spracuje, aktualizuje stav vykonávaného testu a načíta videozáznam z testovania.



Obrázok 6: Priebeh testu

Slovenská technická univerzita v Bratislave
Fakulta informatiky a informačných technológií
Ilkovičova 2, 842 16 Bratislava 4

Tímový projekt



Story Teller

Projektová dokumentácia – používateľská príručka

Vedúci projektu: Ing. Karol Rástočný, PhD.
Názov tímu: CoolStoryBro
Členovia tímu: Bc. Jakub Ondik
Bc. Patrik Januška
Bc. Adam Neupauer
Bc. Martin Olejár
Bc. Miroslav Hurajt
Kontakt: storyteller@googlegroups.com
Akademický rok: 2016/2017

Obsah

1	Registrácia a prihlásenie do aplikácie	1-1
1.1	Scenár – Spustenie aplikácie	1-1
1.2	Scenár – Registrácia do aplikácie	1-1
1.3	Scenár – Prihlásenie do aplikácie	1-2
1.4	Scenár – Obnovenie prihlasovacieho hesla	1-3
1.5	Scenár – Obnovenie registračného emailu	1-4
1.6	Scenár – Odhlásenie z aplikácie	1-6
2	Úprava profilových informácií a zariadení	2-1
2.1	Scenár – Úprava profilových informácií	2-1
2.2	Scenár – Zmena hesla	2-2
2.3	Scenár – Zobrazenie prihlásených zariadení	2-2
2.4	Scenár – Zmena poznámky pri zariadení	2-3
2.5	Scenár – Odstránenie zariadenia.....	2-4
3	Notifikácie	3-1
3.1	Scenár – Zobrazenie toast popup notifikácie.....	3-1
3.2	Scenár – Zobrazenie notifikácií v navigačnom paneli.....	3-1
4	Vytvorenie a správa projektov	4-1
4.1	Scenár – Vytvorenie nového projektu	4-1
4.2	Scenár – Zobrazenie zoznamu projektov.....	4-2
4.3	Scenár – Výber projektu	4-3
4.4	Scenár – Úprava nastavení projektu	4-4
4.5	Scenár – Zobrazenie zoznamu používateľov v projekte.....	4-5
4.6	Scenár – Pridanie používateľa do projektu.....	4-7
4.7	Scenár – Zmena roly používateľa v projekte.....	4-9
5	Vytvorenie a správa skíc	5-1
5.1	Scenár – Vytvorenie novej skice	5-1
5.2	Scenár – Pridanie nových prvkov do skice.....	5-3
5.2.1	Scenár – Pridanie prvku TextBox do skice	5-3
5.2.2	Scenár – Pridanie prvku Label do skice	5-3
5.2.3	Scenár – Pridanie prvku Button do skice	5-4
5.2.4	Scenár – Pridanie prvku TextArea do skice	5-5

5.2.5	Scenár – Pridanie prvku Image do skice	5-5
5.2.6	Scenár – Pridanie prvku Combobox do skice	5-6
5.3	Scenár – Úprava vlastností prvkov skice.....	5-6
5.4	Scenár – Posúvanie prvkov v skici	5-7
5.5	Scenár – Odstránenie prvkov zo skice.....	5-8
5.6	Scenár – Zobrazenie zoznamu skíc v projekte	5-9
5.7	Scenár – Zdieľanie skice.....	5-10
5.8	Scenár – Stiahnutie šablóny skice	5-11
6	Správa scenárov.....	6-1
6.1	Scenár – Duplikácia scenára.....	6-1
6.2	Scenár – Presúvanie krokov v scenári	6-1
6.3	Scenár – Odstránenie scenára	6-2
7	Správa akceptačných testov	7-1
7.1	Scenár – Spustenie akceptačného testu	7-1

1 Registrácia a prihlásenie do aplikácie

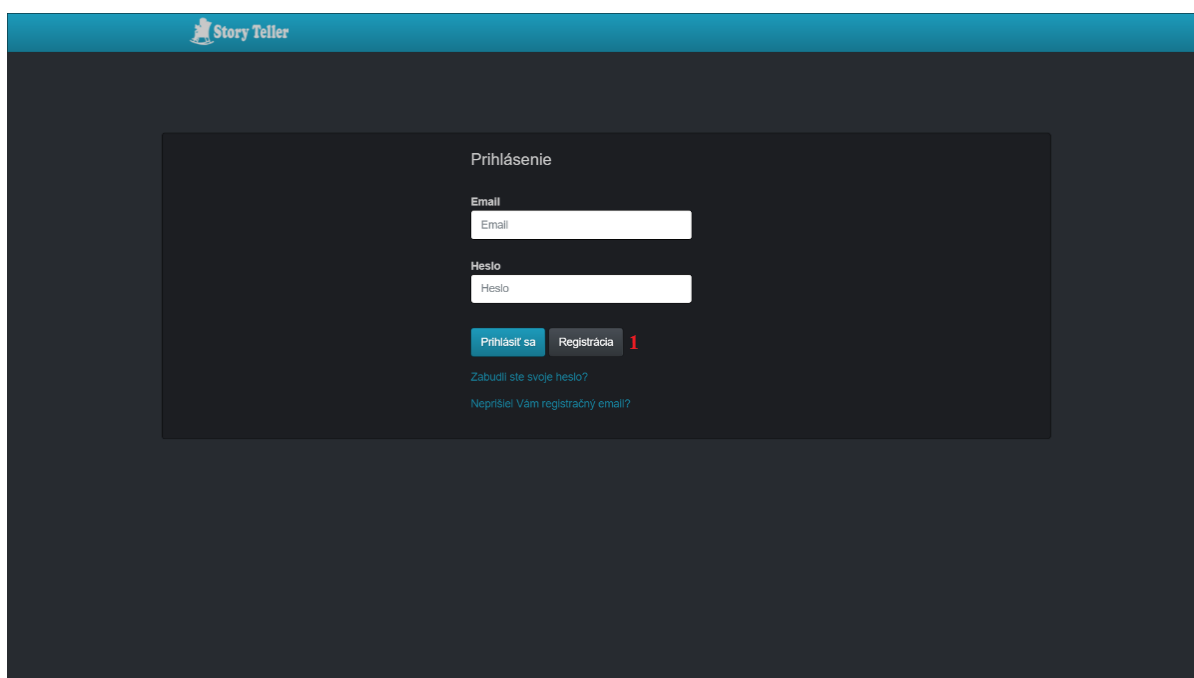
1.1 Scenár – Spustenie aplikácie

Pre spustenie našej aplikácie zadajte do prehliadača adresu <https://app.story-teller.xyz/#/login>.

1.2 Scenár – Registrácia do aplikácie

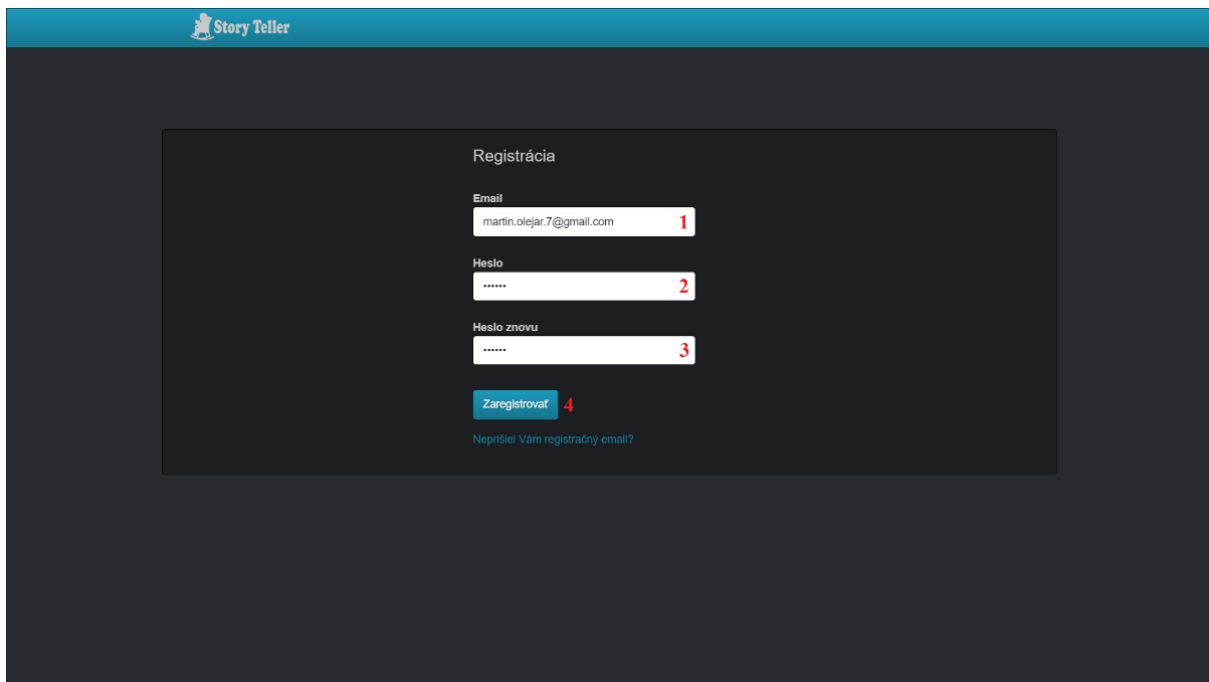
Pre registráciu do našej aplikácie postupujte podľa nasledujúcich krokov:

1. Spustíte aplikáciu (Scenár – Spustenie aplikácie). Zobrazí sa prihlasovací formulár na obr. 1, v ktorom kliknite na tlačidlo Registrácia označené č. 1.



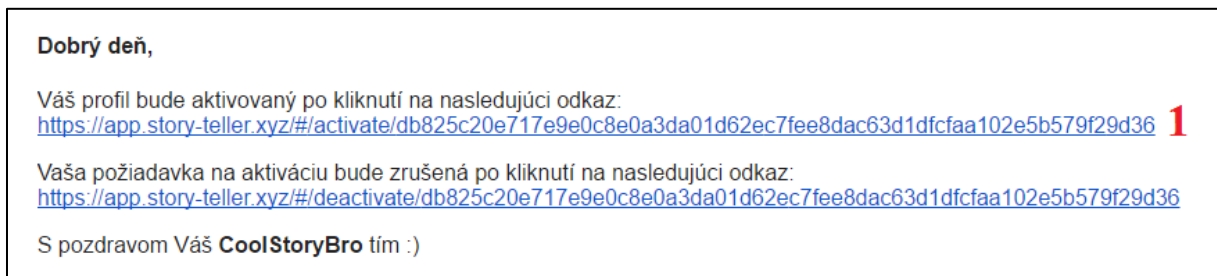
Obrázok 1: Prihlasovací formulár.

2. Následne sa zobrazí registračný formulár na obr. 2, v ktorom:
 - a. vyplňte vašu platnú emailovú adresu do poľa č. 1,
 - b. vyplňte heslo do poľa č. 2, ktoré musí mať aspoň 6 znakov,
 - c. vyplňte ešte raz heslo do poľa č. 3 vyplnené v predchádzajúcom kroku,
 - d. kliknite na tlačidlo Zaregistrovať označené č. 4.



Obrázok 2: Registračný formulár.

3. Prihláste sa do vašej emailovej adresy, ktorú ste zadali pri registrácii. Príde vám registračný email (skontrolujte si aj Spam) na obr. 3, v ktorom kliknite na odkaz označený č. 1.

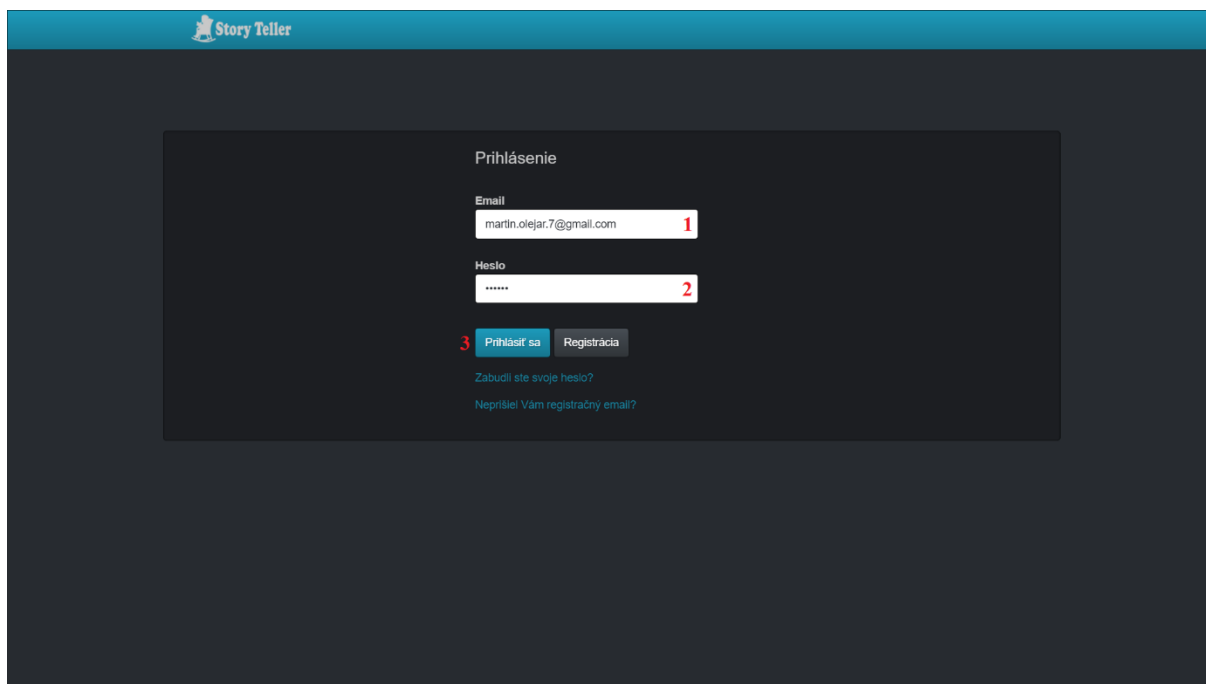


Obrázok 3: Registračný email.

1.3 Scenár – Prihlásenie do aplikácie

Pre prihlásenie sa do našej aplikácie postupujte podľa nasledujúcich krokov:

1. Spustíte aplikáciu (Scenár – Spustenie aplikácie). Zobrazí sa prihlasovací formulár na obr. 4, v ktorom:
 - a. vyplňte email do poľa č. 1, ktorý ste zadali pri registrácii,
 - b. vyplňte heslo do poľa č. 2, ktoré ste zadali pri registrácii,
 - c. kliknite na tlačidlo Prihlásiť sa označené č. 3.

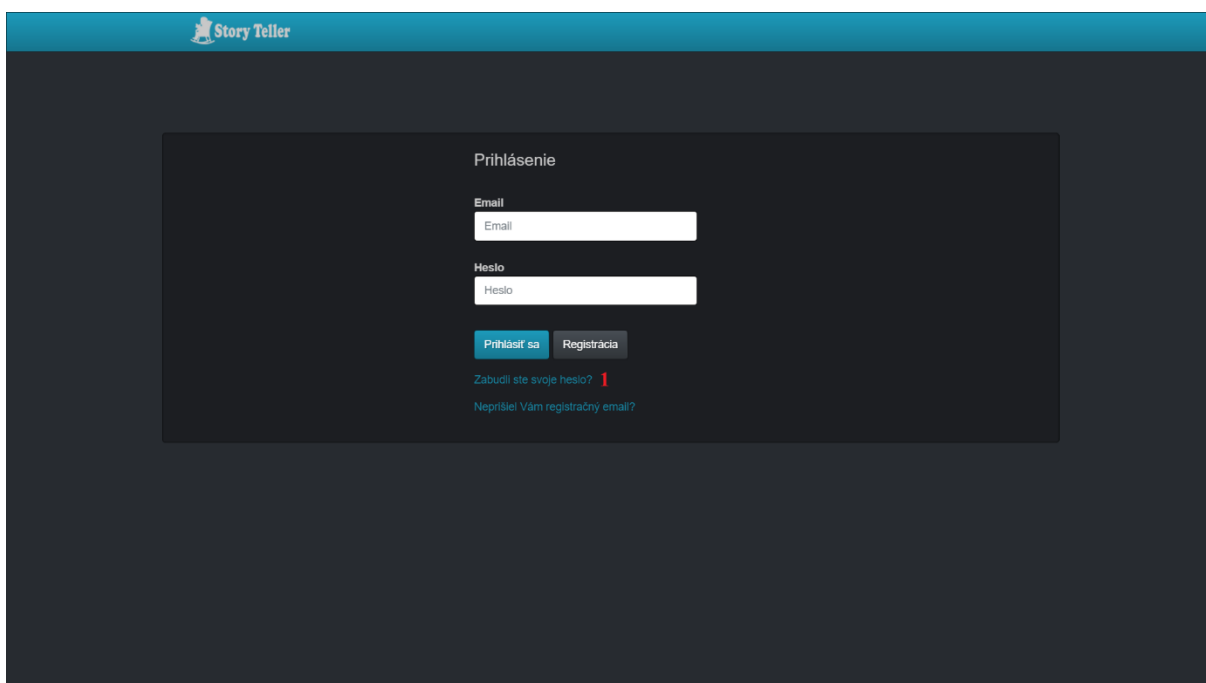


Obrázok 4: Prihlasovací formulár.

1.4 Scenár – Obnovenie prihlasovacieho hesla

Pre obnovenie prihlasovacieho hesla do aplikácie postupujte podľa nasledujúcich krokov:

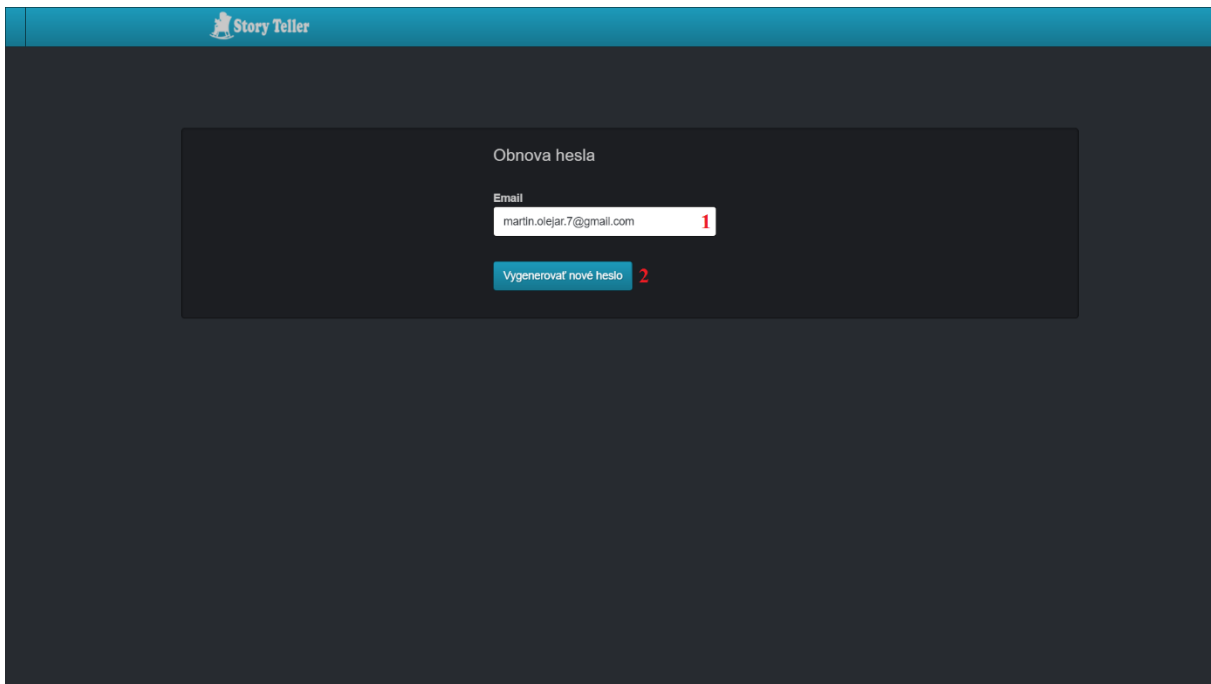
1. Spustíte aplikáciu (Scenár – Spustenie aplikácie). Zobrazí sa prihlasovací formulár na obr. 5, v ktorom kliknite na odkaz označený č. 1.



Obrázok 5: Prihlasovací formulár.

2. Následne sa zobrazí formulár pre obnovenie prihlasovacieho hesla na obr. 6, v ktorom:
 - a. vyplňte emailovú adresu do poľa č. 1, ktorú používate na prihlásenie,

- b. kliknite na tlačidlo Vygenerovať nové heslo označené č. 2.



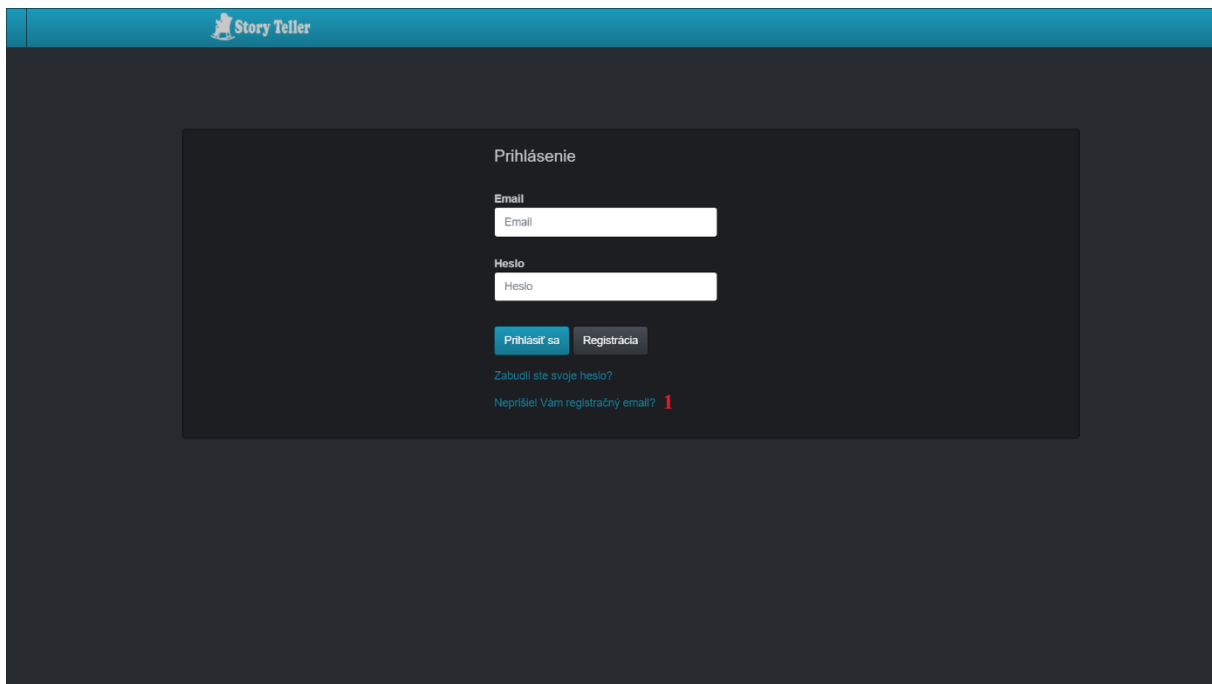
Obrázok 6: Formulár pre obnovu hesla.

3. Prihláste sa do vašej emailovej adresy, na ktorú vám príde email obsahujúci nové prihlasovacieho heslo.

1.5 Scenár – Obnovenie registračného emailu

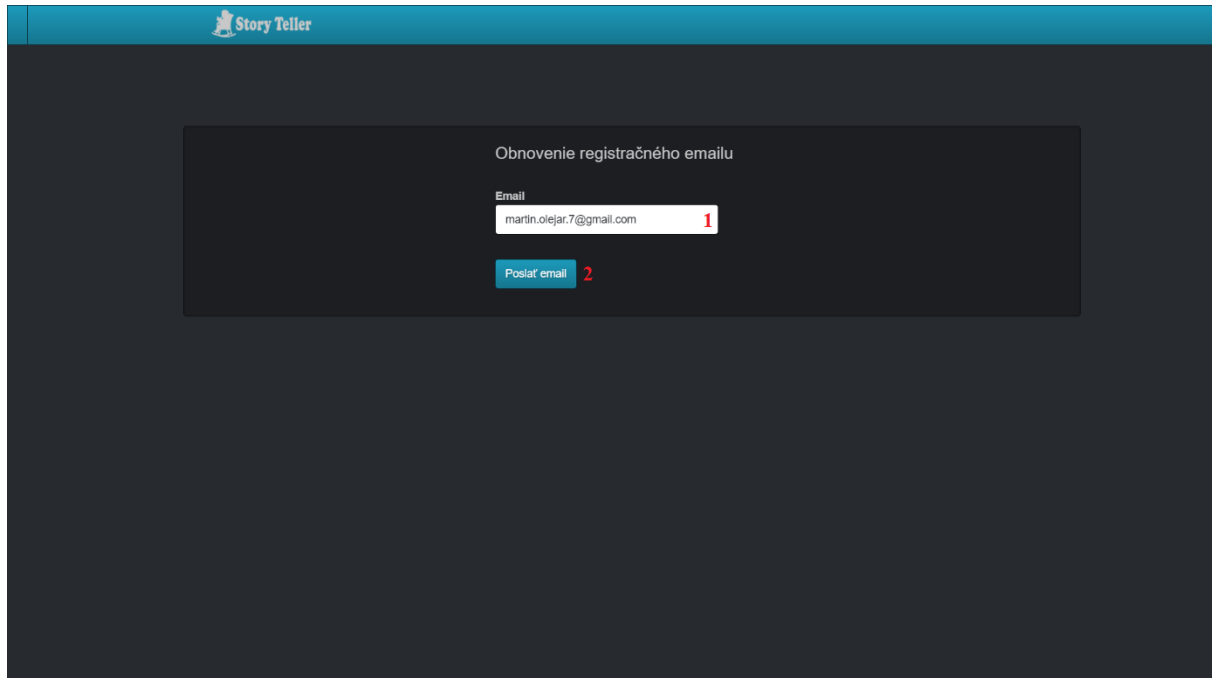
Pre obnovenie registračného emailu postupujte podľa nasledujúcich krokov:

1. Spustíte aplikáciu (Scenár – Spustenie aplikácie). Zobrazí sa prihlasovací formulár na obr. 7, v ktorom kliknite na odkaz označený č. 1.



Obrázok 7: Prihlasovací formulár.

2. Následne sa zobrazí formulár pre obnovenie registračného emailu na obr. 8, v ktorom:
 - a. vyplňte emailovú adresu do poľa č. 1, ktorú ste použili pri registrácii,
 - b. kliknite na tlačidlo Poslať email označené č. 2.

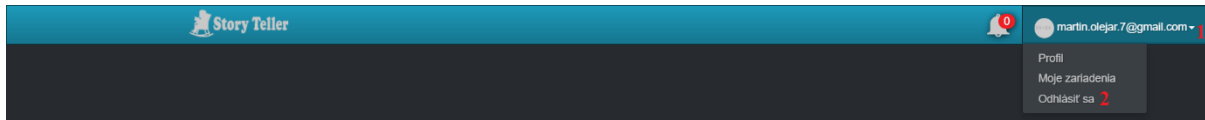


Obrázok 8: Formulár pre obnovenie registračného emailu.

1.6 Scenár – Odhlásenie z aplikácie

Pre odhlásenie sa z aplikácie postupujte podľa nasledujúcich krokov:

1. Po prihlásení (Scenár – Spustenie aplikácie) v navigačnom paneli aplikácie na obr. 9:
 - a. kliknite na vašu emailovú adresu vpravo hore označenú č. 1,
 - b. kliknite na možnosť Odhlásiť sa označenú č. 2.



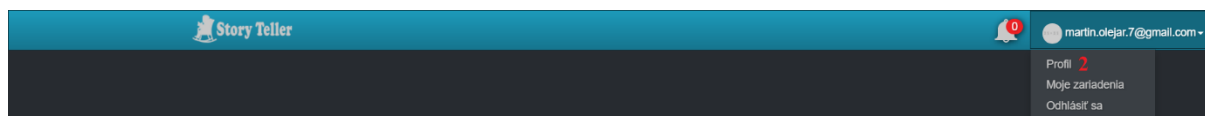
Obrázok 9: Navigačný panel aplikácie.

2 Úprava profilových informácií a zariadení

2.1 Scenár – Úprava profilových informácií

Pre úpravu profilových informácií postupujte podľa nasledujúcich krokov:

1. Ak nie ste prihlásený v aplikácii, prihláste sa do aplikácie (Scenár – Prihlásenie do aplikácie).
2. V navigačnom paneli aplikácie na obr. 10:
 - a. kliknite na vašu emailovú adresu vpravo hore označenú č. 1,
 - b. kliknite na možnosť Profil označenú č. 2.



Obrázok 10: Navigačný panel aplikácie.

3. Následne sa zobrazí formulár pre úpravu profilových informácií na obr. 11, v ktorom:
 - a. vyberte vašu profilovú fotku kliknutím na tlačidlo Vyberte fotku označené č. 1,
 - b. vyplňte vaše meno do poľa č. 2,
 - c. vyplňte vaše priezvisko do poľa č. 3,
 - d. vyplňte vašu domovskú stránku do poľa č. 4,
 - e. vyberte lokalizáciu aplikácie v poli č. 5,
 - f. kliknite na tlačidlo Aktualizovať označené č. 6.

Obrázok 11: Formulár pre úpravu profilových informácií.

2.2 Scenár – Zmena hesla

Pre zmenu hesla postupujte podľa nasledujúcich krokov:

1. Ak nie ste prihlásený v aplikácii, prihláste sa do aplikácie (Scenár – Prihlásenie do aplikácie).
2. V navigačnom paneli aplikácie na obr. 12:
 1. kliknite na vašu emailovú adresu vpravo hore označenú č. 1,
 2. kliknite na možnosť Profil označenú č. 2.



Obrázok 12: Navigačný panel aplikácie.

3. Následne sa zobrazí formulár pre zmenu hesla na obr. 13, v ktorom:
 - a. vyplňte vaše staré prihlasovacie heslo do poľa č. 1,
 - b. vyplňte vaše nové prihlasovacie heslo do poľa č. 2,
 - c. vyplňte ešte raz vaše nové prihlasovacie heslo do poľa č. 3,
 - d. kliknite na tlačidlo Zmeniť heslo označené č. 4.

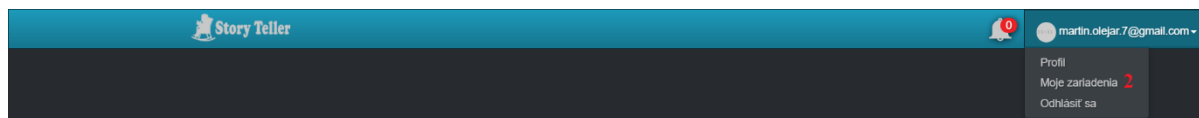
Obrázok 13: Formulár pre zmenu hesla.

2.3 Scenár – Zobrazenie prihlásených zariadení

Pre zobrazenie prihlásených zariadení postupujte podľa nasledujúcich krokov:

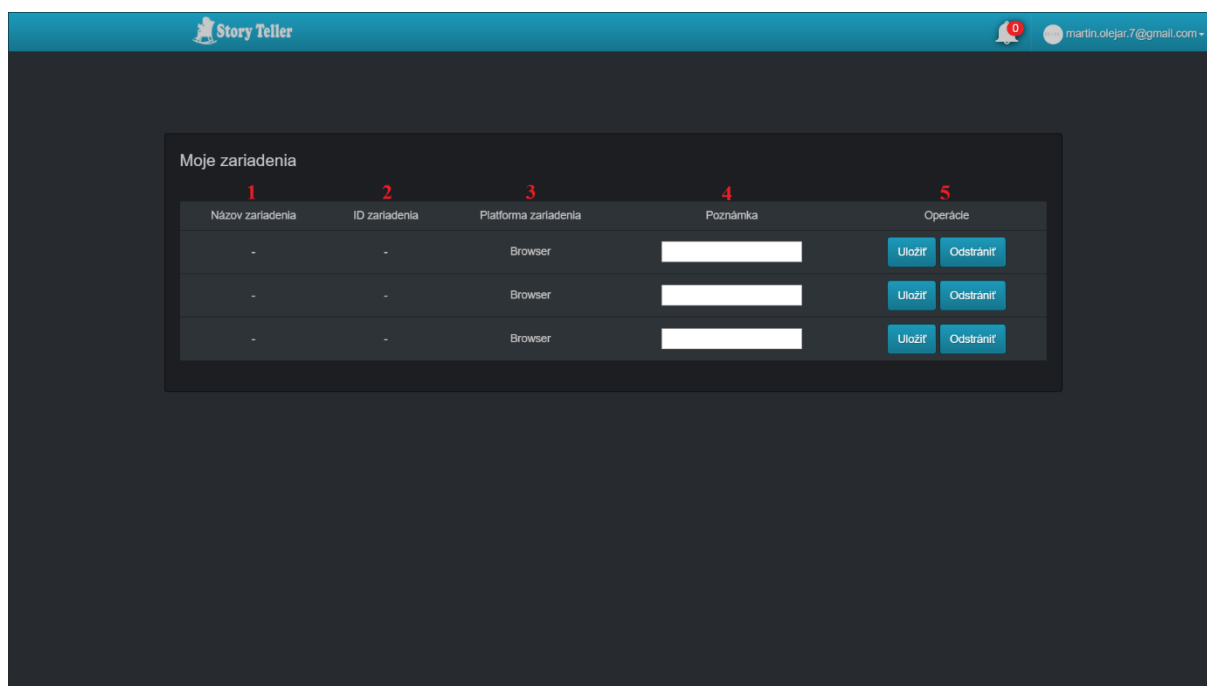
1. Ak nie ste prihlásený v aplikácii, prihláste sa do aplikácie (Scenár – Prihlásenie do aplikácie).
2. V navigačnom paneli aplikácie na obr. 14:

- kliknite na vašu emailovú adresu vpravo hore označenú č. 1,
- kliknite na možnosť Moje zariadenia označenú č. 2.



Obrázok 14: Navigačný panel aplikácie.

- Následne sa zobrazí zoznam prihlásených zariadení na obr. 15. Tento zoznam reprezentovaný tabuľkou obsahuje pre každé prihlásené zariadenie:
 - jeho názov v stĺpci s názvom Názov zariadenia označenom č. 1,
 - jeho identifikátor v stĺpci s názvom ID zariadenia označenom č. 2,
 - jeho platforma v stĺpci s názvom Platforma zariadenia označenom č. 3,
 - poznámku, ktorá sa vzťahuje k zariadeniu, v stĺpci s názvom Poznámka označenom č. 4 a
 - operácie, ktoré je možné vykonať, v stĺpci s názvom Operácie označenom č. 5.

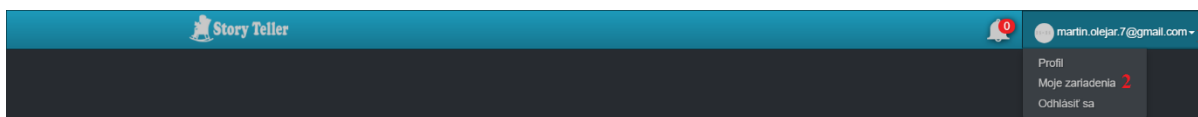


Obrázok 15: Tabuľka pre zoznam prihlásených zariadení.

2.4 Scenár – Zmena poznámky pri zariadení

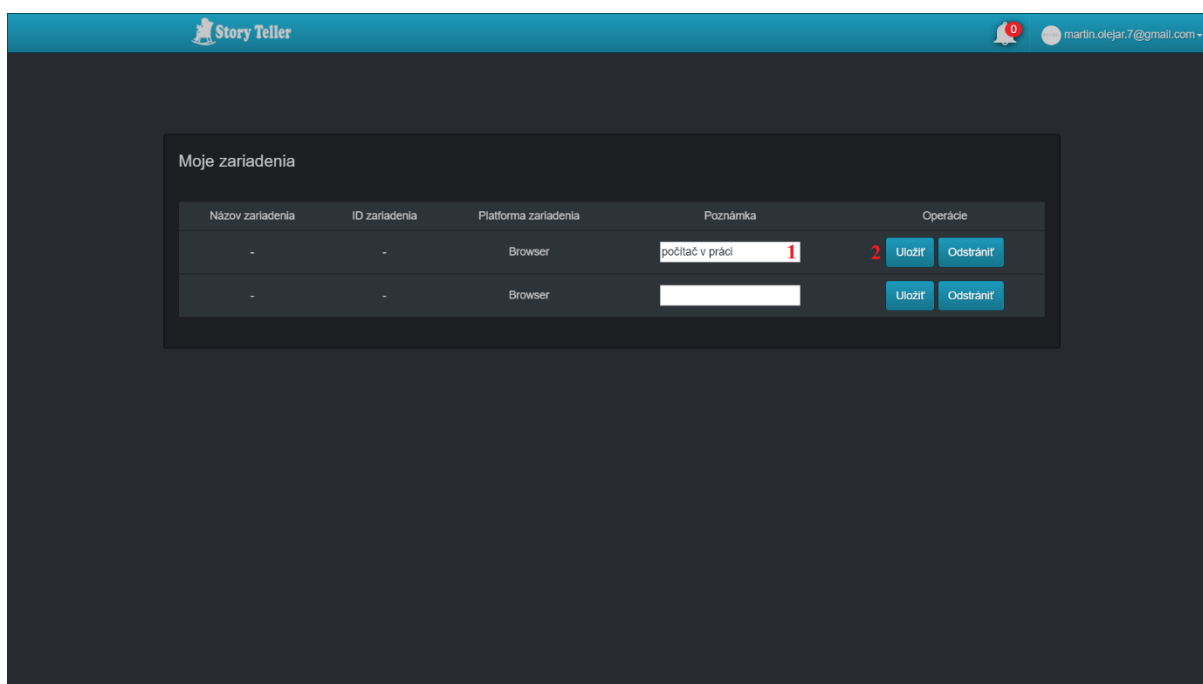
Pre zmenu poznámky pri prihlásenom zariadení postupujte podľa nasledujúcich krokov:

1. Ak nie ste prihlásený v aplikácii, prihláste sa do aplikácie (Scenár – Prihlásenie do aplikácie).
2. V navigačnom paneli aplikácie na obr. 16:
 - a. kliknite na vašu emailovú adresu vpravo hore označenú č. 1,
 - b. kliknite na možnosť Moje zariadenia označenú č. 2.



Obrázok 16: Navigačný panel aplikácie.

3. Následne sa zobrazí formulár pre zoznam prihlásených zariadení na obr. 17, v ktorom:
 - a. vyplňte poznámku pri konkrétnom zariadení do poľa č. 1,
 - b. kliknite na tlačidlo Uložiť označené č. 2 v tom istom riadku.

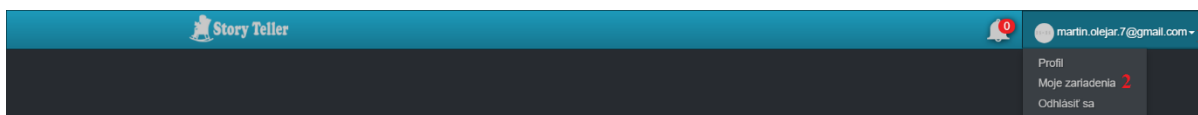


Obrázok 17: Tabuľka obsahujúca zoznam prihlásených zariadení.

2.5 Scenár – Odstránenie zariadenia

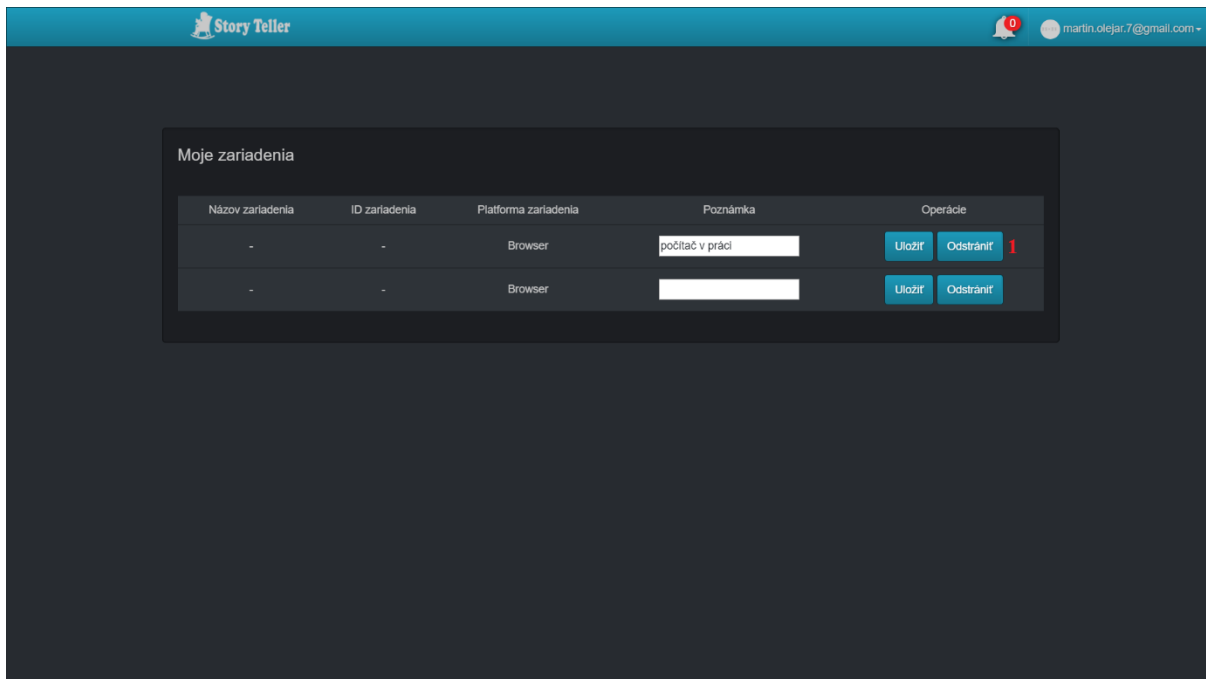
Pre zmenu poznámky pri prihlásenom zariadení postupujte podľa nasledujúcich krokov:

1. Ak nie ste prihlásený v aplikácii, prihláste sa do aplikácie (Scenár – Prihlásenie do aplikácie).
2. V navigačnom paneli aplikácie na obr. 18:
 - a. kliknite na vašu emailovú adresu vpravo hore označenú č. 1,
 - b. kliknite na možnosť Moje zariadenia označenú č. 2.



Obrázok 18: Navigačný panel aplikácie.

3. Následne sa zobrazí formulár pre zoznam prihlásených zariadení na obr. 19, v ktorom:
 - a. kliknite na tlačidlo Odstrániť označené č. 1 v tom riadku, v ktorom sa nachádza dané zariadenie, ktoré chcete odstrániť.



Obrázok 19: Formulár pre zoznam prihlásených zariadení.

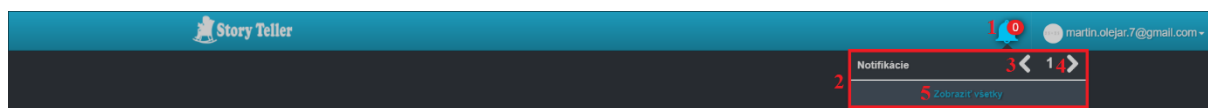
3 Notifikácie

3.1 Scenár – Zobrazenie toast popup notifikácie

3.2 Scenár – Zobrazenie notifikácií v navigačnom paneli

Pre zobrazenie notifikácií v navigačnom paneli postupujte podľa nasledujúcich krokov:

1. Ak nie ste prihlásený v aplikácii, prihláste sa do aplikácie (Scenár – Prihlásenie do aplikácie).
2. V navigačnom paneli aplikácie na obr. 20:
 - a. kliknite na zvonček označený č. 1,
 - b. zobrazí sa okno obsahujúce prvých 5 notifikácií označené č. 2,
 - c. pre zobrazenie predchádzajúcich 5 notifikácií kliknite na šípku označenú č. 3,
 - d. pre zobrazenie nasledujúcich 5 notifikácií kliknite na šípku označenú č. 4,
 - e. pre zobrazenie všetkých notifikácií kliknite na odkaz Zobraziť všetky označený č. 5.



Obrázok 20: Navigačný panel aplikácie.

4 Vytvorenie a správa projektov

4.1 Scenár – Vytvorenie nového projektu

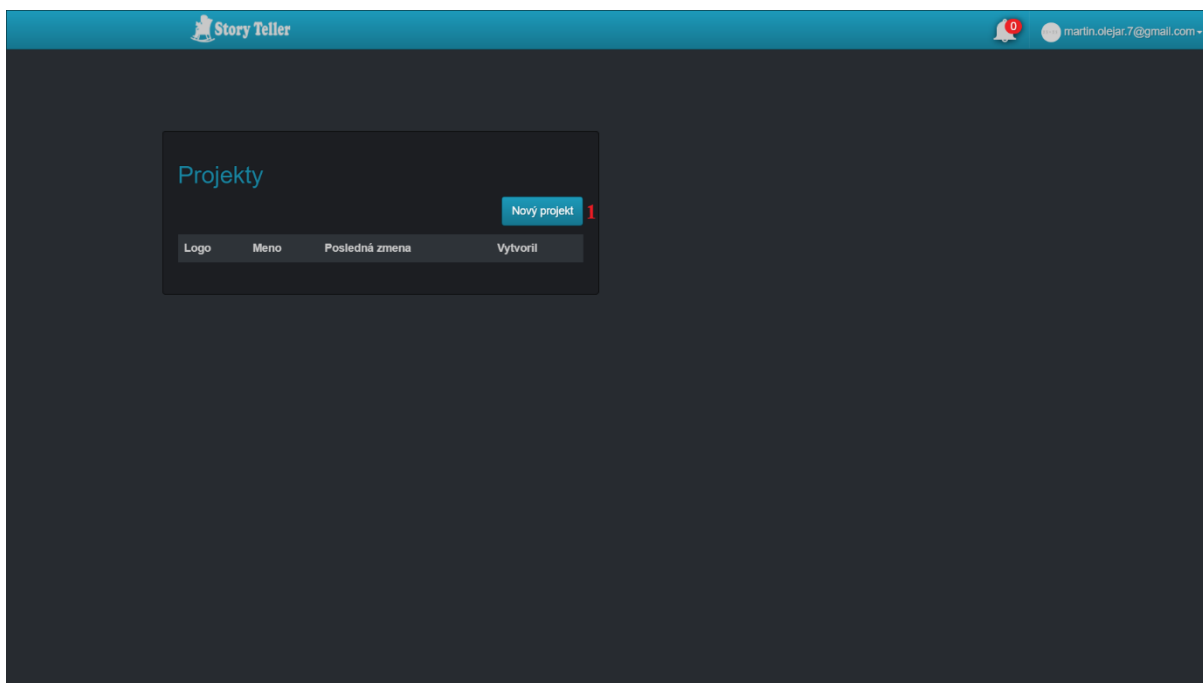
Pre vytvorenie nového projektu postupujte podľa nasledujúcich krokov:

1. Ak nie ste prihlásený v aplikácii, prihláste sa do aplikácie (Scenár – Prihlásenie do aplikácie). Ak ste prihlásený v aplikácii, v navigačnom paneli aplikácie na obr. 21 kliknite na logo aplikácie označené č. 1.



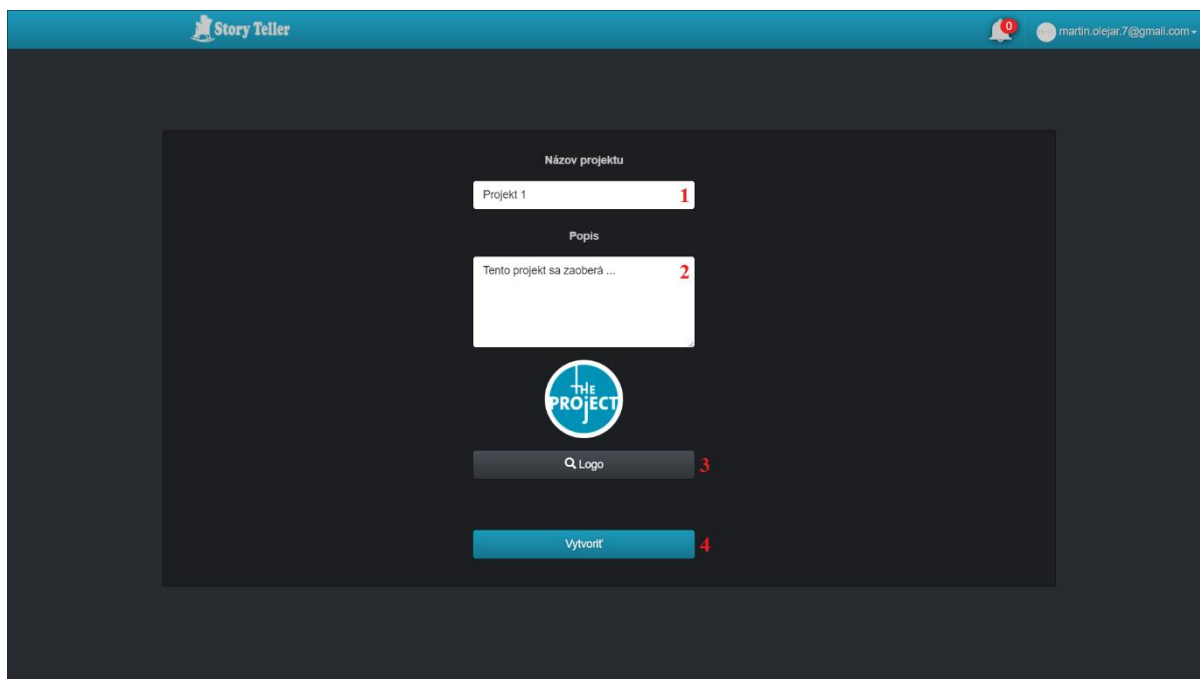
Obrázok 21: Navigačný panel aplikácie.

2. Zobrazí sa úvodná obrazovka aplikácie na obr. 22, v ktorej kliknite na tlačidlo Nový projekt označené č. 1.



Obrázok 22: Úvodná obrazovka aplikácie.

3. Zobrazí sa formulár pre vytvorenie nového projektu na obr. 23, v ktorom:
 - a. vyplňte názov projektu do poľa č. 1,
 - b. vyplňte popis projektu do poľa č. 2,
 - c. vyberte logo projektu kliknutím na tlačidlo Logo označené č. 3 (nepovinné),
 - d. kliknite na tlačidlo Vytvoriť označené č. 4.



Obrázok 23: Formulár pre vytvorenie nového projektu.

4.2 Scenár – Zobrazenie zoznamu projektov

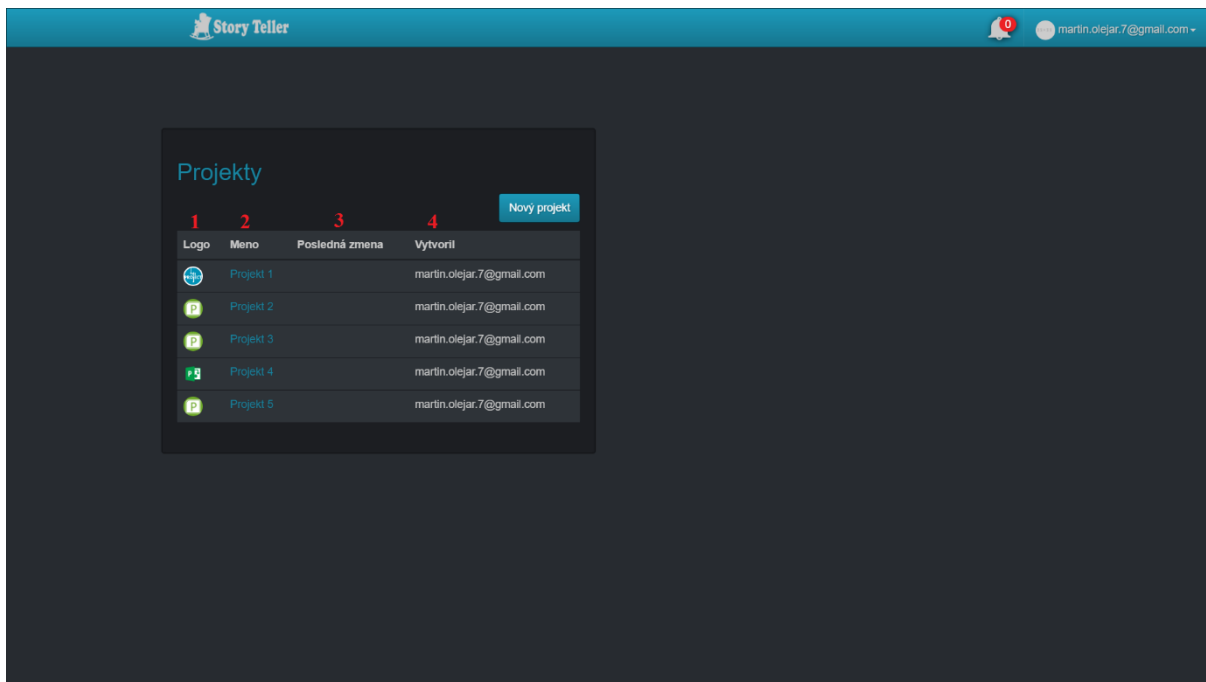
Pre zobrazenie zoznamu vytvorených projektov postupujte podľa nasledujúcich krokov:

1. Ak nie ste prihlásený v aplikácii, prihláste sa do aplikácie (Scenár – Prihlásenie do aplikácie). Ak ste prihlásený v aplikácii, v navigačnom paneli aplikácie na obr. **Obrázok 24** kliknite na logo aplikácie označené č. 1.



Obrázok 24: Navigačný panel aplikácie.

2. Zobrazí sa úvodná obrazovka aplikácie na obr. 25, v ktorej je zoznam vytvorených projektov. Tento zoznam reprezentovaný tabuľkou obsahuje pre každý projekt:
 - jeho logo v stĺpci s názvom Logo označenom č. 1,
 - jeho meno v stĺpci s názvom Meno označenom č. 2,
 - dátum poslednej zmeny v stĺpci s názvom Posledná zmena označenom č. 3 a
 - email používateľa, ktorý ho vytvoril, v stĺpci s názvom Vytvoril označenom č. 4.



Obrázok 25: Úvodná obrazovka aplikácie.

4.3 Scenár – Výber projektu

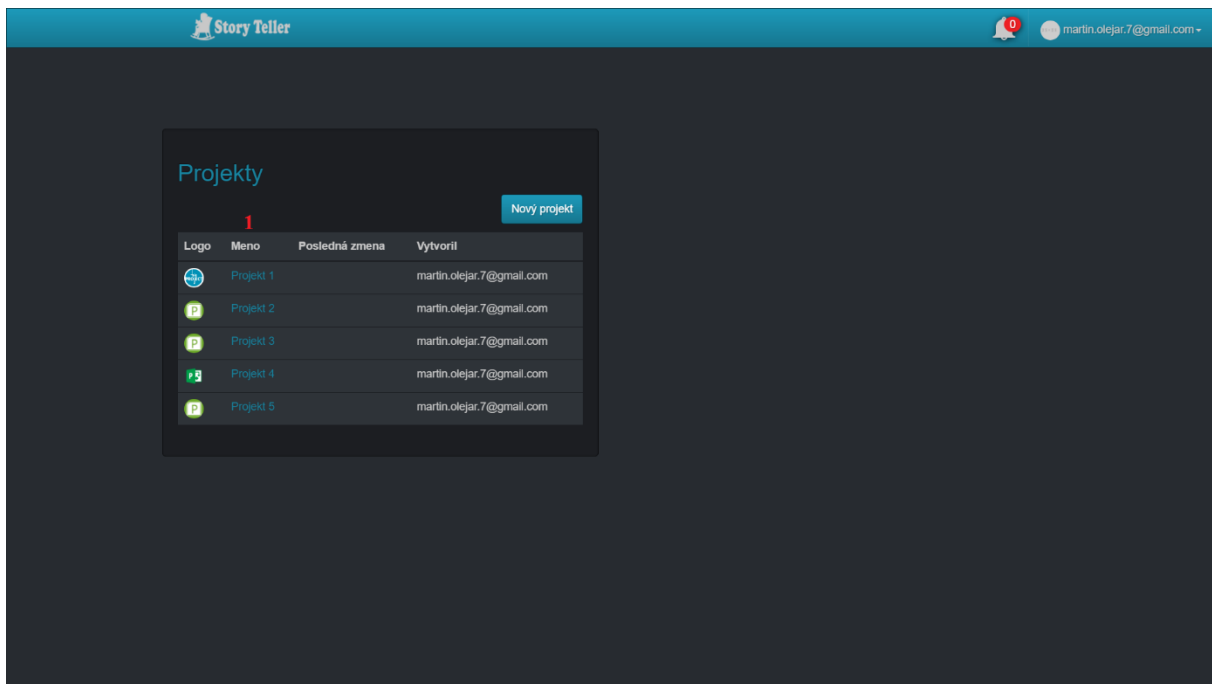
Pre výber existujúceho projektu postupujte podľa nasledujúcich krokov:

1. Ak nie ste prihlásený v aplikácii, prihláste sa do aplikácie (Scenár – Prihlásenie do aplikácie). Ak ste prihlásený v aplikácii, v navigačnom paneli aplikácie na obr. 26 kliknite na logo aplikácie označené č. 1.



Obrázok 26: Navigačný panel aplikácie.

2. Zobrazí sa úvodná obrazovka aplikácie na obr. 27, v ktorej je zoznam vytvorených projektov. Vyberte si projekt kliknutím na jeho meno v stĺpci s názvom Meno označenom č. 1.

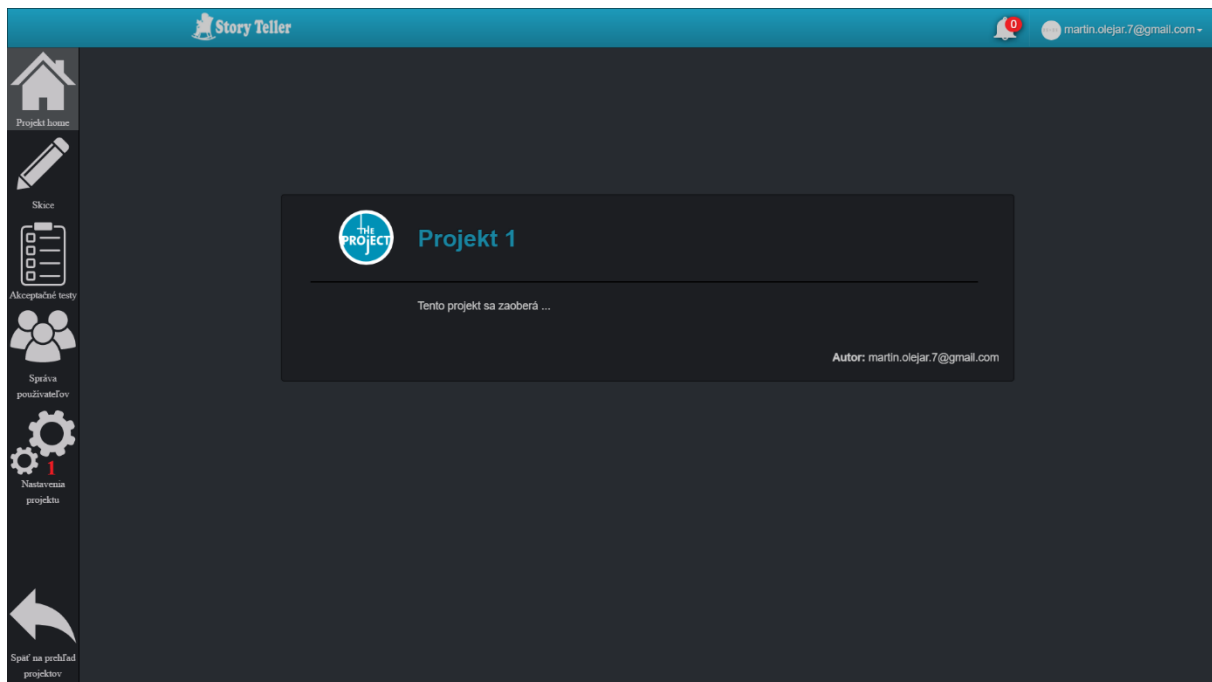


Obrázok 27: Úvodná obrazovka aplikácie.

4.4 Scenár – Úprava nastavení projektu

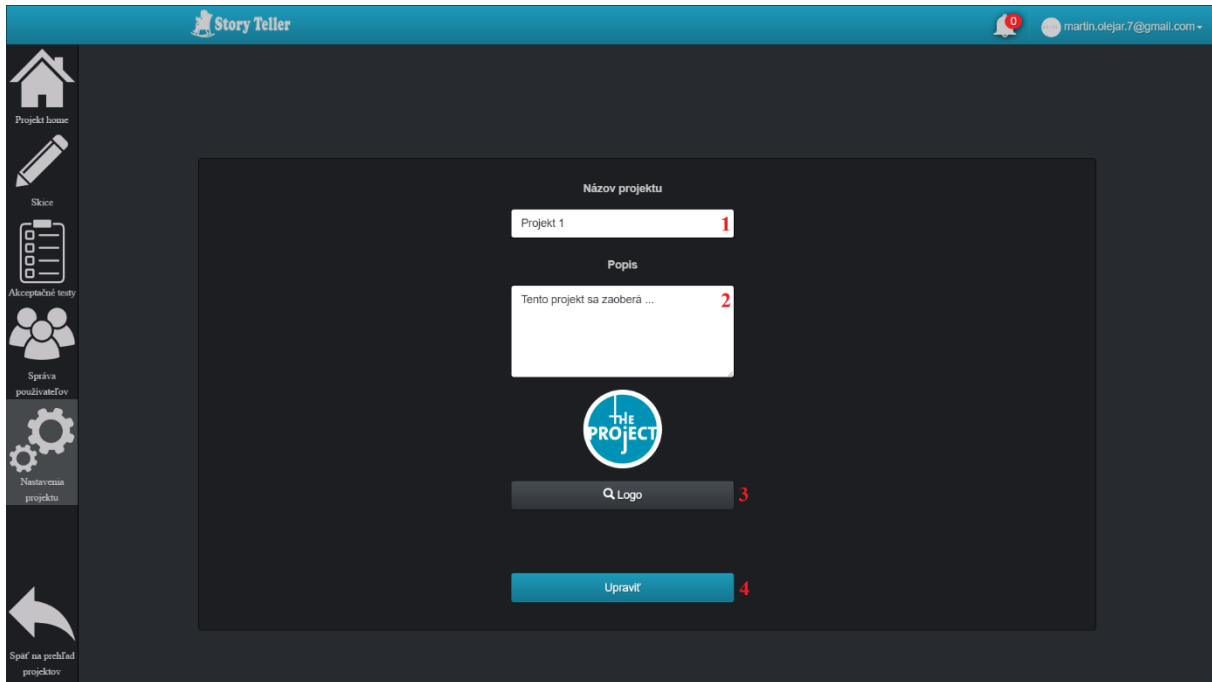
Pre úpravu nastavení projektu postupujte podľa nasledujúcich krokov:

1. Vyberte si projekt, ktorého nastavenia chcete upraviť (Scenár – Výber projektu).
2. Zobrazí sa domovská obrazovka projektu na obr. 28, v ktorej kliknite na časť Nastavenia projektu označenú č. 1.



Obrázok 28: Domovská obrazovka projektu.

3. Následne sa zobrazí formulár na úpravu nastavení projektu na obr. 29, v ktorom môžete upravovať:
 - a. názov projektu v poli označenom č. 1,
 - b. popis projektu v poli označenom č. 2,
 - c. logo projektu kliknutím na tlačidlo Logo označené č. 3 a výberom nového loga.
4. Upravené nastavenia projektu potvrdíte kliknutím na tlačidlo Upraviť označené č. 4.

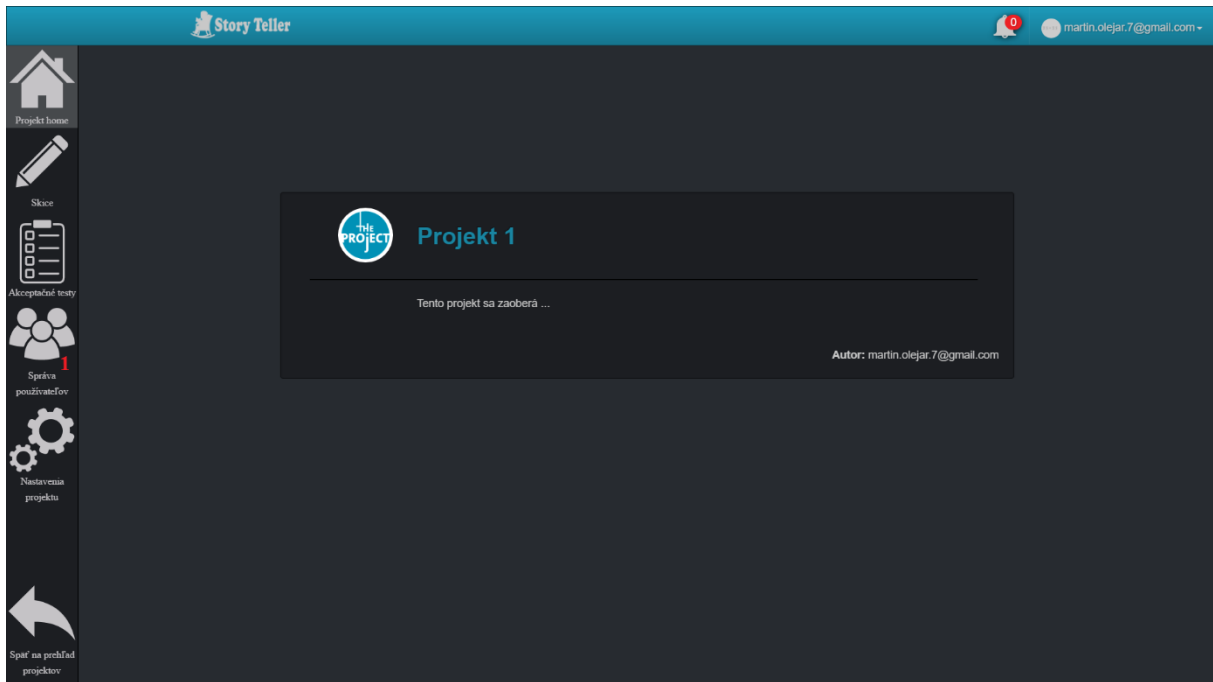


Obrázok 29: Formulár na úpravu nastavení projektu.

4.5 Scenár – Zobrazenie zoznamu používateľov v projekte

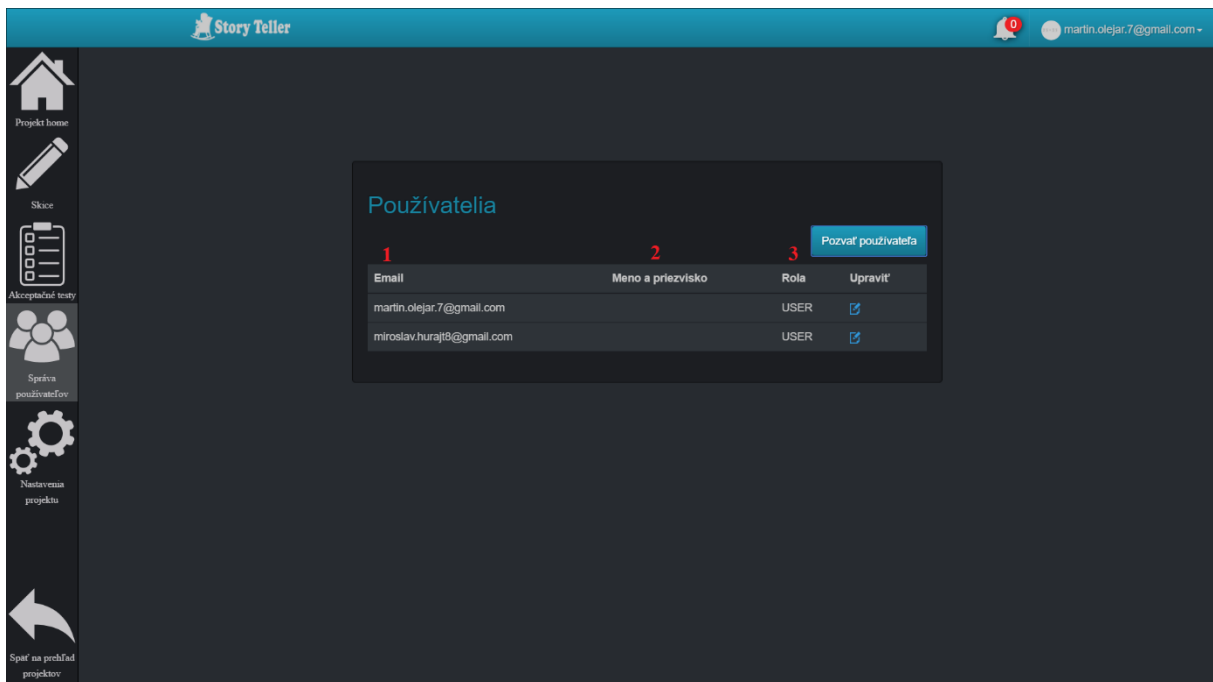
Pre zobrazenie zoznamu používateľov v projekte postupujte podľa nasledujúcich krokov:

1. Vyberte si projekt, ktorého nastavenia chcete upraviť (Scenár – Výber projektu).
2. Zobrazí sa domovská obrazovka projektu na obr. 30, v ktorej kliknite na časť Správa používateľov označenú č. 1.



Obrázok 30: Domovská obrazovka projektu.

3. Následne sa zobrazí zoznam používateľov projektu na obr. 31. Tento zoznam reprezentovaný tabuľkou obsahuje pre každého používateľa projektu:
 - a. jeho email v stĺpci s názvom Email označenom č. 1,
 - b. jeho meno a priezvisko v stĺpci s názvom Meno a priezvisko označenom č. 2,
 - c. jeho rolu v rámci projektu v stĺpci s názvom Rola označenom č. 3.

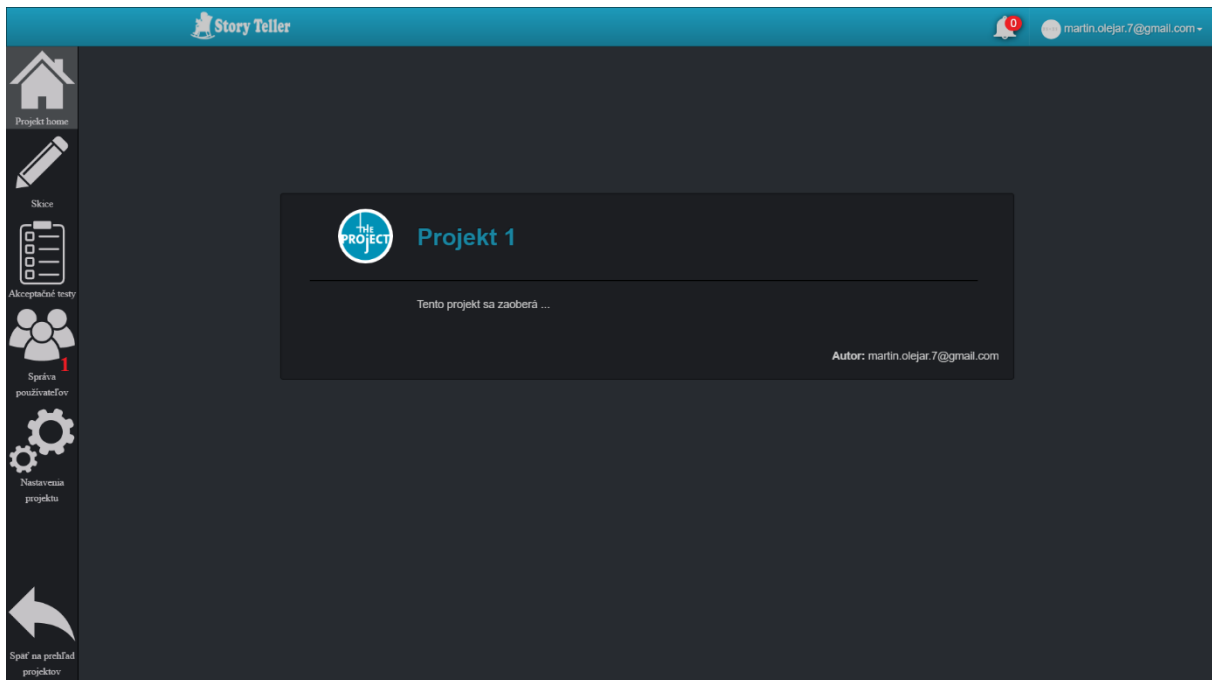


Obrázok 31: Tabuľka pre zoznam používateľov projektu.

4.6 Scenár – Pozvanie používateľa do projektu

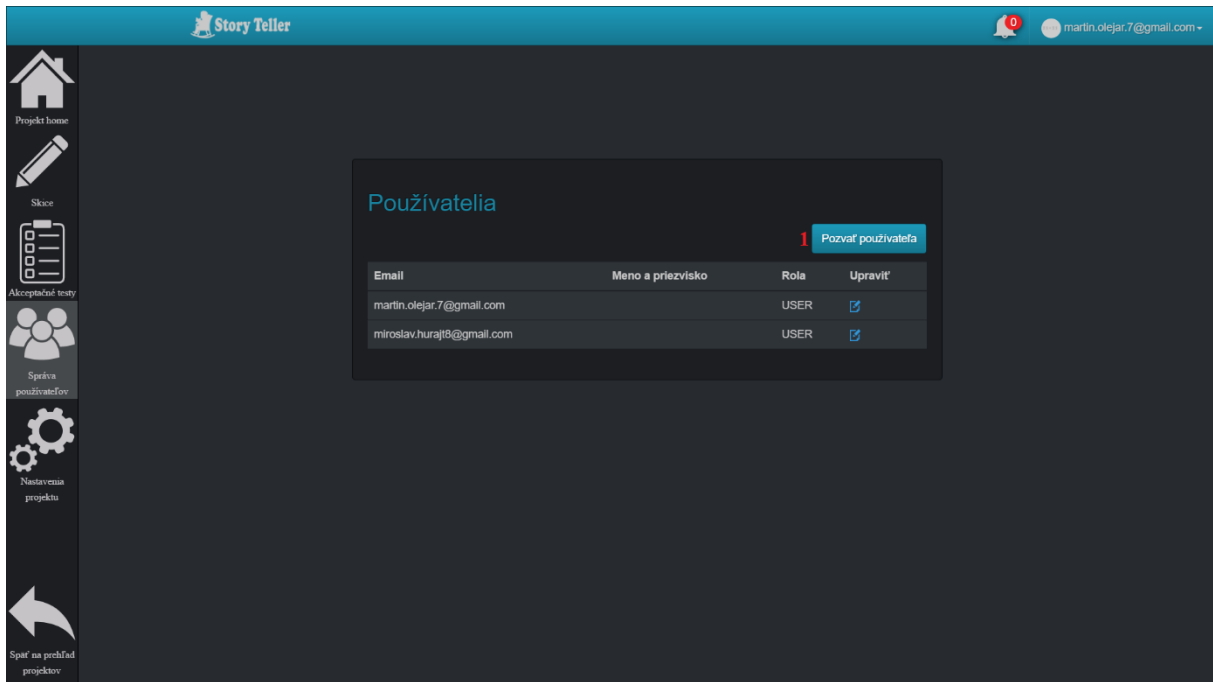
Pre pozvanie používateľa do projektu postupujte podľa nasledujúcich krokov:

1. Vyberte si projekt, ktorého nastavenia chcete upraviť (Scenár – Výber projektu).
2. Zobrazí sa domovská obrazovka projektu na obr. 32, v ktorej kliknite na časť Správa používateľov označenú č. 1.



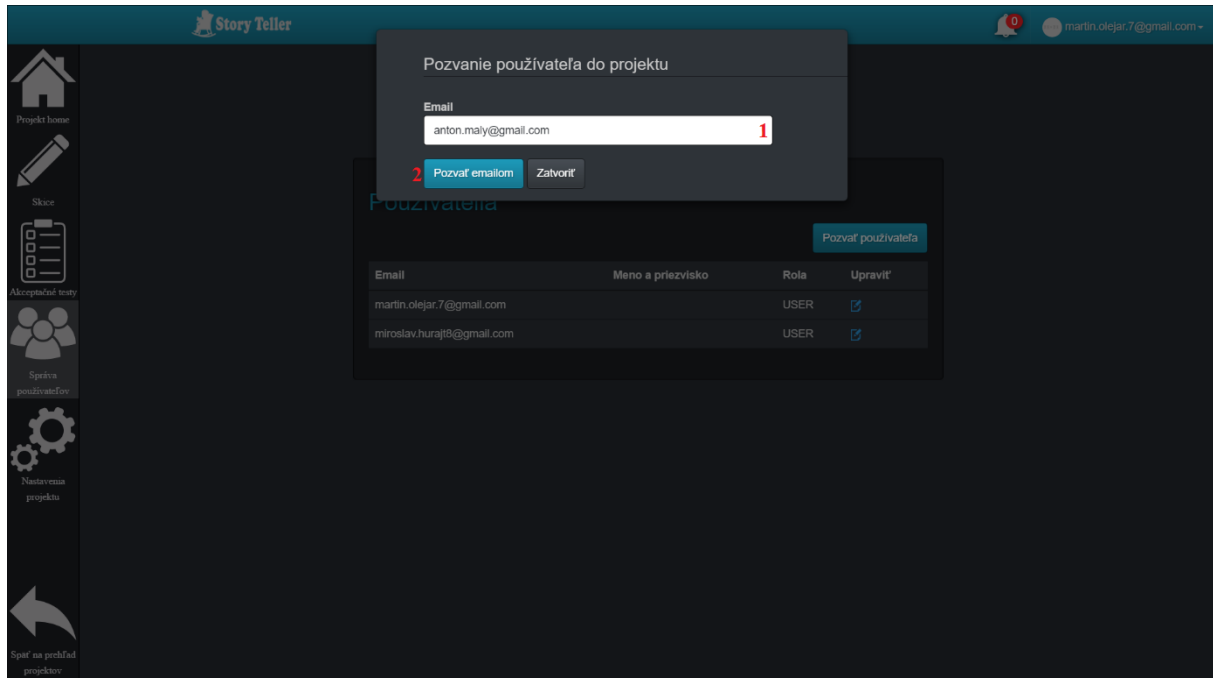
Obrázok 32: Domovská obrazovka projektu.

3. Následne sa zobrazí obrazovka pre správu používateľov projektu na obr. 33, v ktorej kliknite na tlačidlo Pozvať používateľa označené č. 1.



Obrázok 33: Správa používateľov projektu.

4. V zobrazenom modálnom okne na obr. 34:
 - a. vyplňte email používateľa, ktorého chcete pozvať do projektu, do poľa č. 1,
 - b. kliknite na tlačidlo Pozvať emailom označené č. 2.

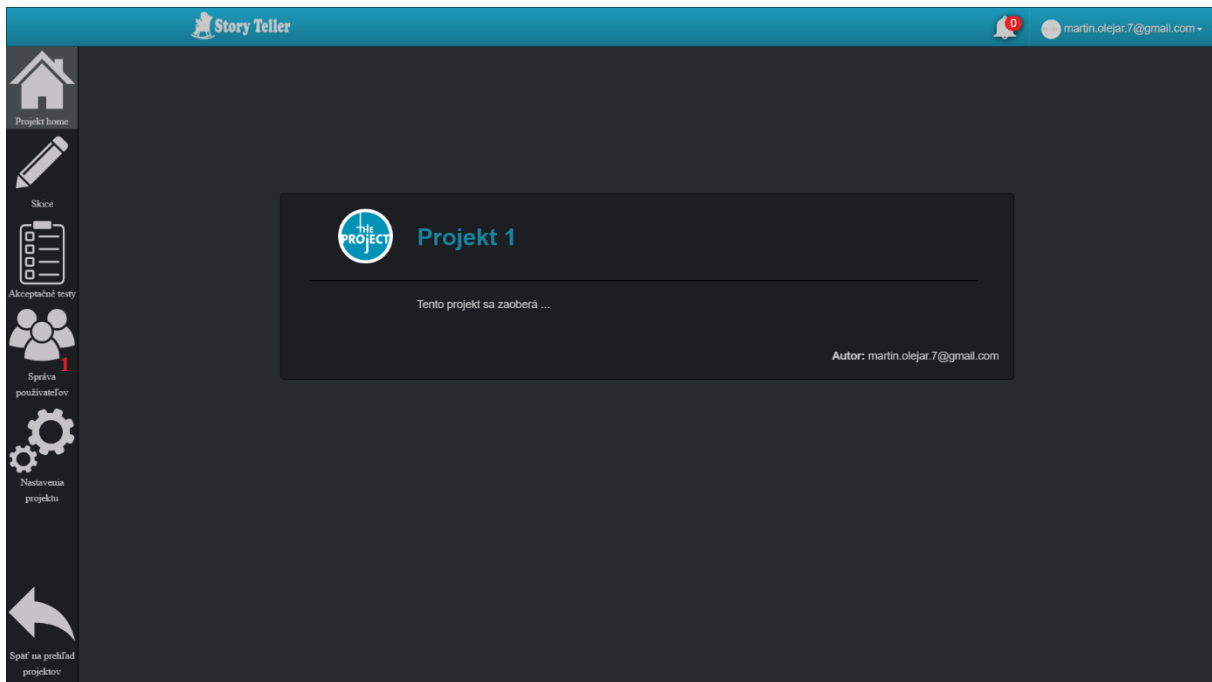


Obrázok 34: Modálne okno pre pozvanie používateľa do projektu.

4.7 Scenár – Zmena roly používateľa v projekte

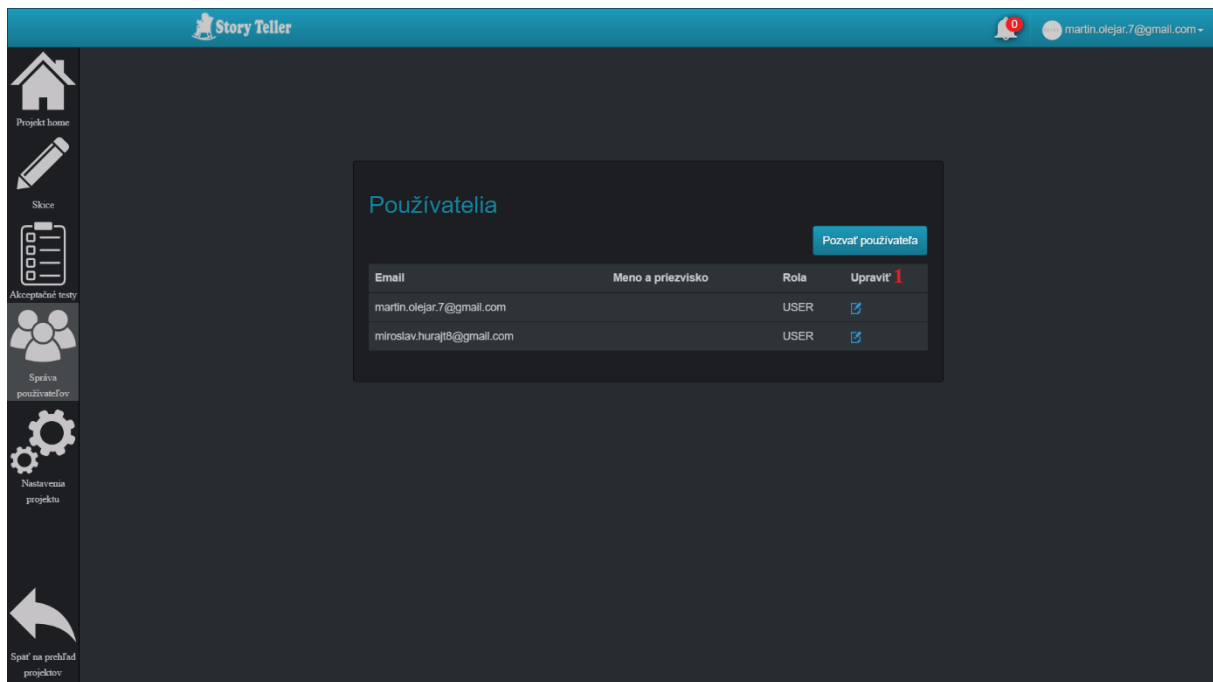
Pre zobrazenie zoznamu používateľov v projekte postupujte podľa nasledujúcich krokov:

1. Vyberte si projekt, ktorého nastavenia chcete upraviť (Scenár – Výber projektu).
2. Zobrazí sa domovská obrazovka projektu na obr. 35, v ktorej kliknite na časť Správa používateľov označenú č. 1.



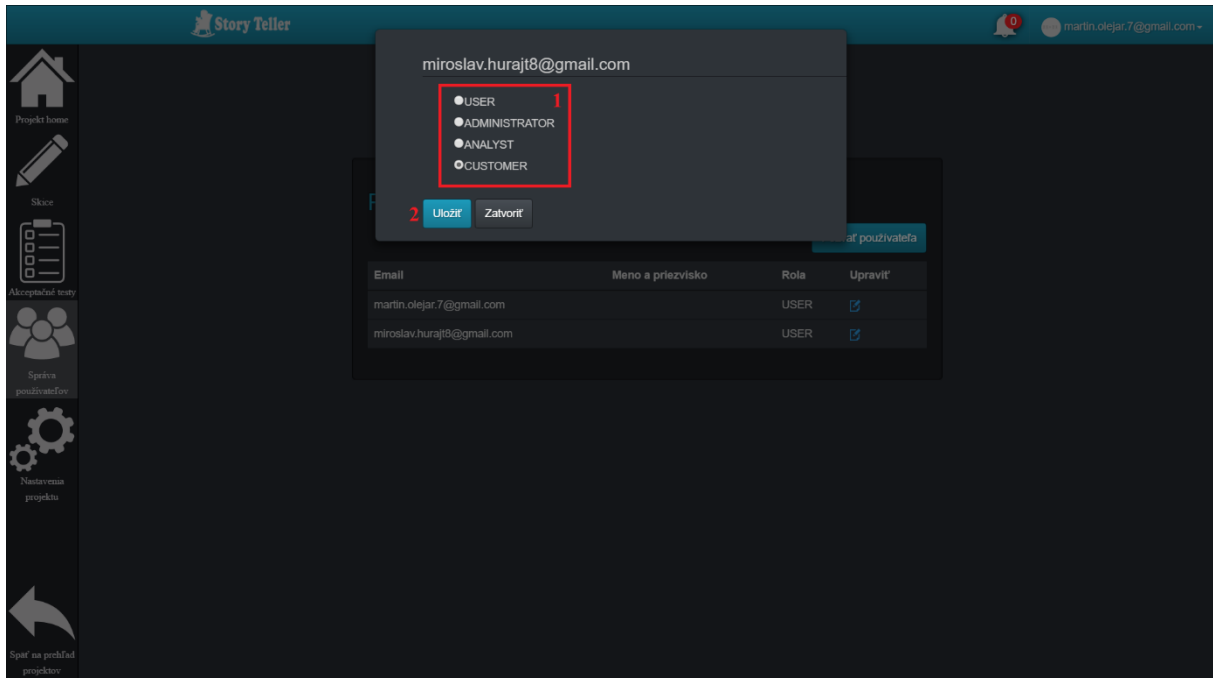
Obrázok 35: Domovská obrazovka projektu.

3. Následne sa zobrazí obrazovka pre správu používateľov projektu na obr. 36, v ktorej kliknite na ikonu v stĺpci s názvom Upraviť označenom č. 1 pre používateľa, ktorého rolu v projekte chcete zmeniť.



Obrázok 36: Správa používateľov projektu.

4. V zobrazenom modálnom okne na obr. 37:
 - a. vyberte rolu z ponúkaných možností v časti označenej č. 1,
 - b. kliknite na tlačidlo Uložiť označené č. 2.



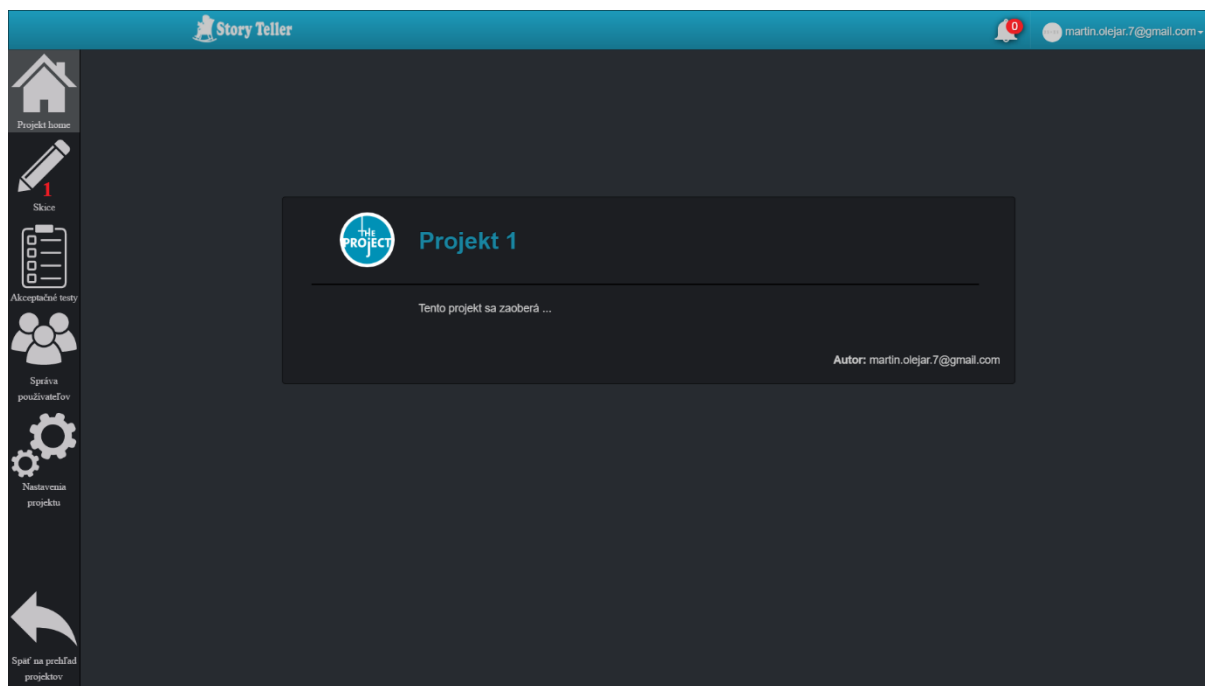
Obrázok 37: Modálne okno pre zmenu roly používateľa v projekte.

5 Vytvorenie a správa skíc

5.1 Scenár – Vytvorenie novej skice

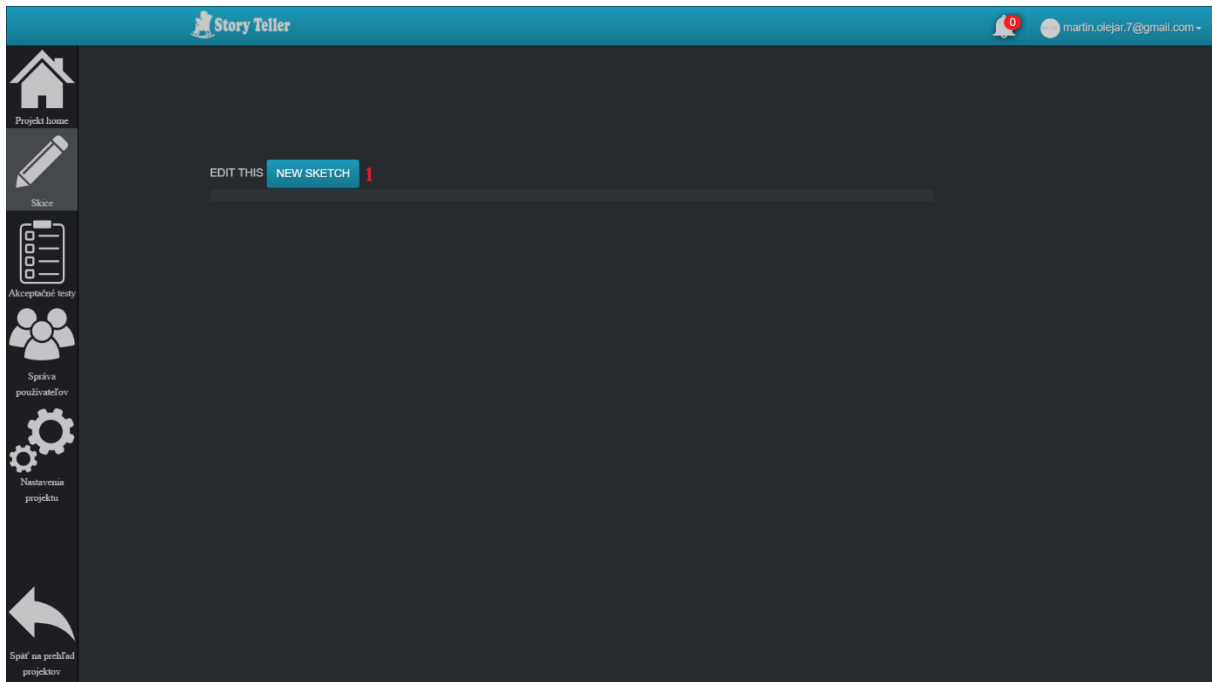
Pre pridanie používateľa do projektu postupujte podľa nasledujúcich krokov:

1. Vyberte si projekt, v ktorom chcete vytvoriť novú skicu (Scenár – Výber projektu).
2. Zobrazí sa domovská obrazovka projektu na obr. 38, v ktorej kliknite na časť Skice označenú č. 1.



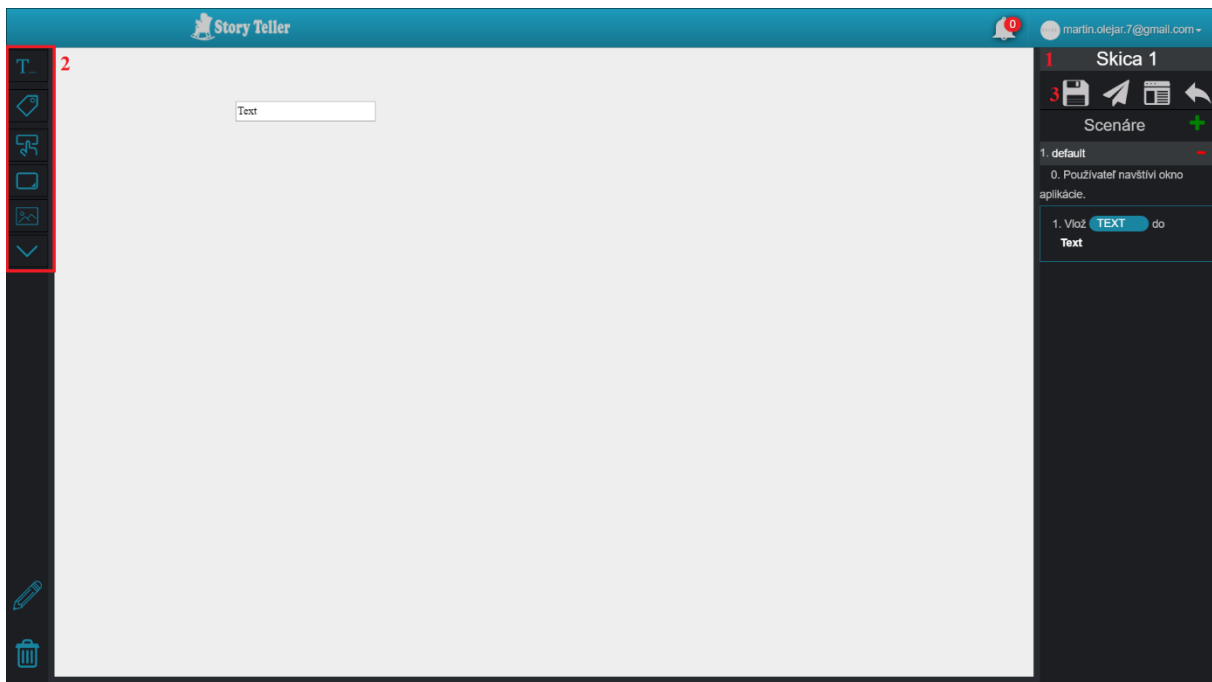
Obrázok 38: Domovská obrazovka projektu.

3. Následne sa zobrazí hlavná obrazovka skíc na obr. 39, v ktorej kliknite na tlačidlo New Sketch označené č. 1.



Obrázok 39: Hlavná obrazovka skíc.

4. Zobrazí sa obrazovka obsahujúca plátno pre kreslenie skice na obr. 40, v ktorom:
 - a. vyplňte názov skice v poli označenom č. 1,
 - b. pridajte aspoň 1 prvok do skice z časti označenej č. 2 (pozri Scenár – Pridanie nových prvkov do skice)
 - c. uložte vytvorenú skicu kliknutím na ikonu uloženia označenú č. 3.



Obrázok 40: Obrazovka obsahujúca plátno pre kreslenie skice, scenáre skice a operácie.

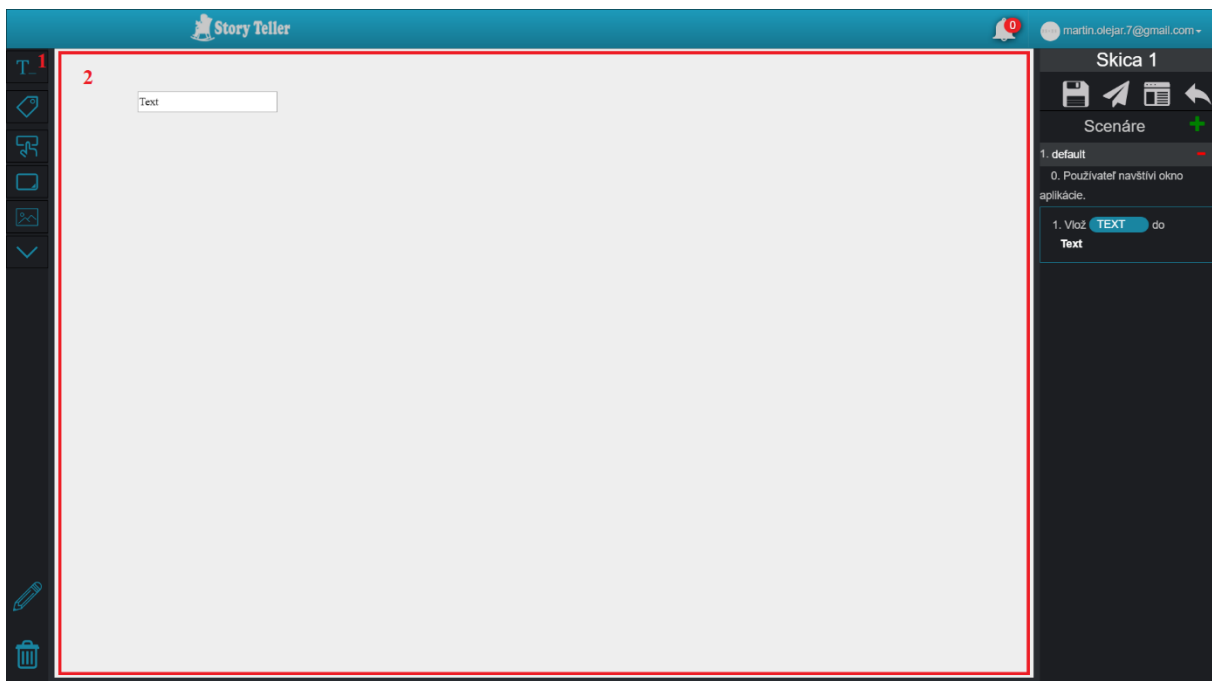
5.2 Scenár – Pridanie nových prvkov do skice

Do skice je možné pridať nasledujúce prvky:

- TextBox,
- Label,
- Button,
- TextArea,
- Image,
- Combobox.

5.2.1 Scenár – Pridanie prvku TextBox do skice

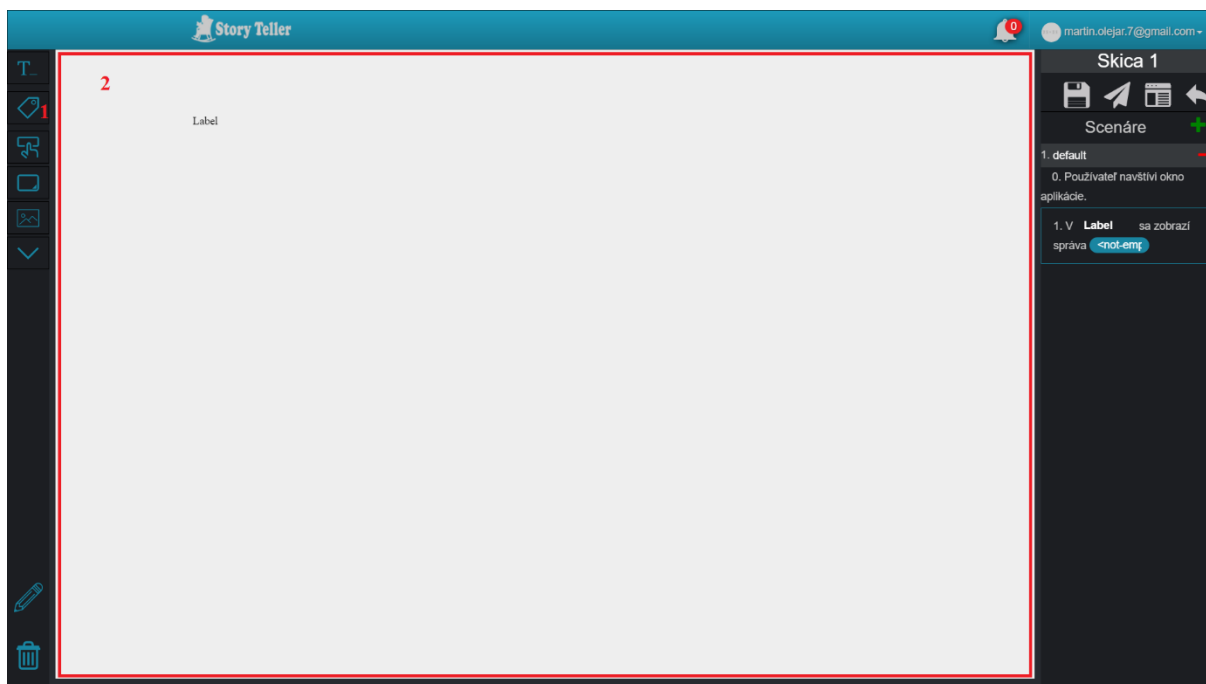
Pre pridanie prvku TextBox do skice kliknite v obrazovke na obr. 41 na ikonu prvku TextBox označenú č. 1 a ťahajte ju do oblasti plátna označenej č. 2.



Obrázok 41: Obrazovka obsahujúca plátno pre kreslenie skice, scenáre skice a operácie.

5.2.2 Scenár – Pridanie prvku Label do skice

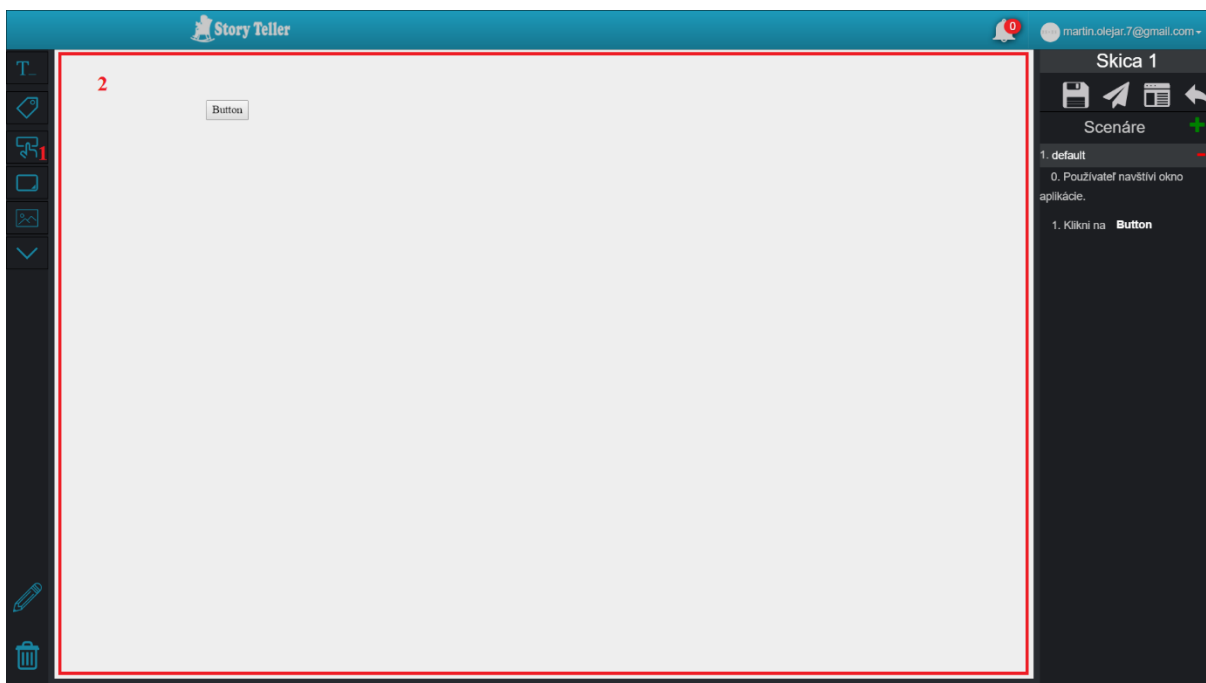
Pre pridanie prvku Label do skice kliknite v obrazovke na obr. 42 na ikonu prvku Label označenú č. 1 a ťahajte ju do oblasti plátna označenej č. 2.



Obrázok 42: Obrazovka obsahujúca plátno pre kreslenie skice, scenáre skice a operácie.

5.2.3 Scenár – Pridanie prvku Button do skice

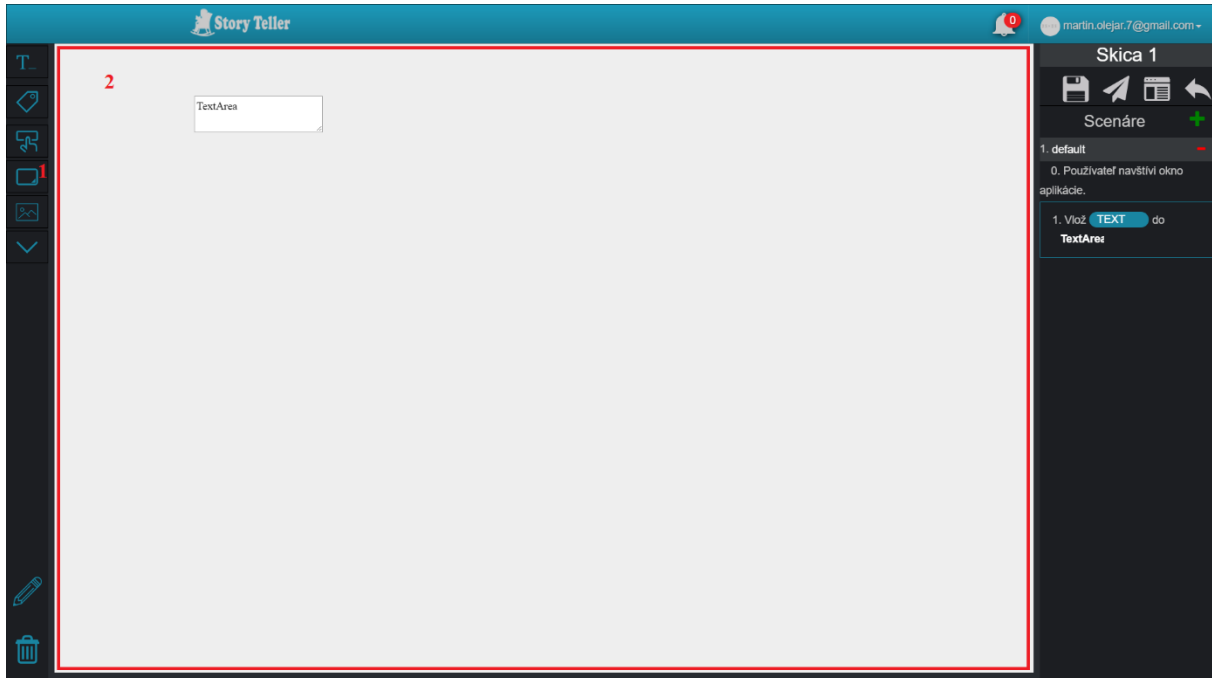
Pre pridanie prvku Button do skice kliknite v obrazovke na obr. 43 na ikonu prvku Button označenú č. 1 a ťahajte ju do oblasti plátna označenej č. 2.



Obrázok 43: Obrazovka obsahujúca plátno pre kreslenie skice, scenáre skice a operácie.

5.2.4 Scenár – Pridanie prvku TextArea do skice

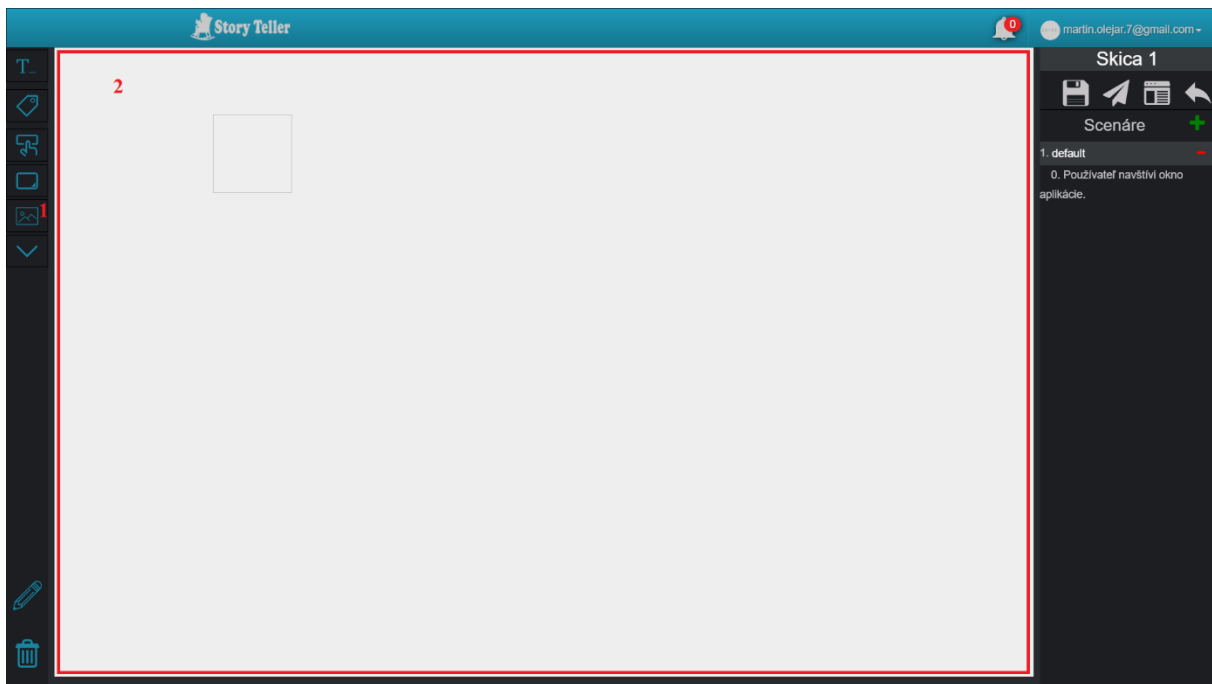
Pre pridanie prvku TextArea do skice kliknite v obrazovke na obr. 44 na ikonu prvku TextArea označenú č. 1 a ťahajte ju do oblasti plátna označenej č. 2.



Obrázok 44: Obrazovka obsahujúca plátno pre kreslenie skice, scenáre skice a operácie.

5.2.5 Scenár – Pridanie prvku Image do skice

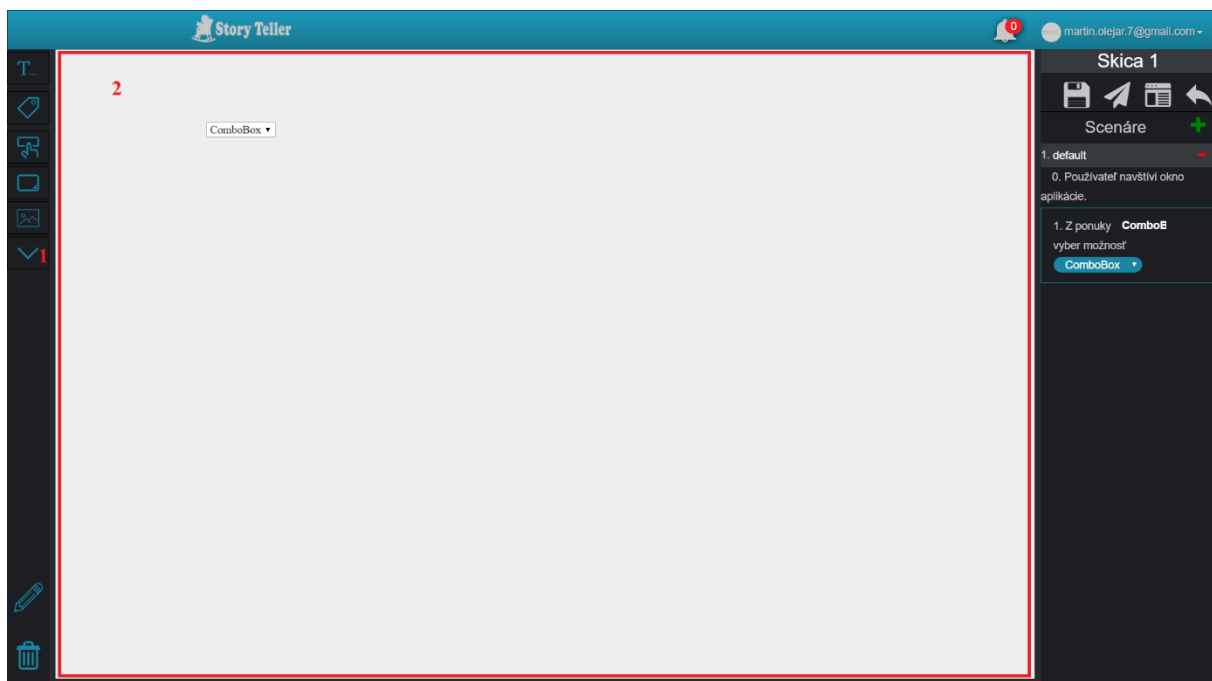
Pre pridanie prvku Image do skice kliknite v obrazovke na obr. 45 na ikonu prvku Image označenú č. 1 a ťahajte ju do oblasti plátna označenej č. 2.



Obrázok 45: Obrazovka obsahujúca plátno pre kreslenie skice, scenáre skice a operácie.

5.2.6 Scenár – Pridanie prvku ComboBox do skice

Pre pridanie prvku ComboBox do skice kliknite v obrazovke na obr. 46 na ikonu prvku ComboBox označenú č. 1 a ťahajte ju do oblasti plátna označenej č. 2.

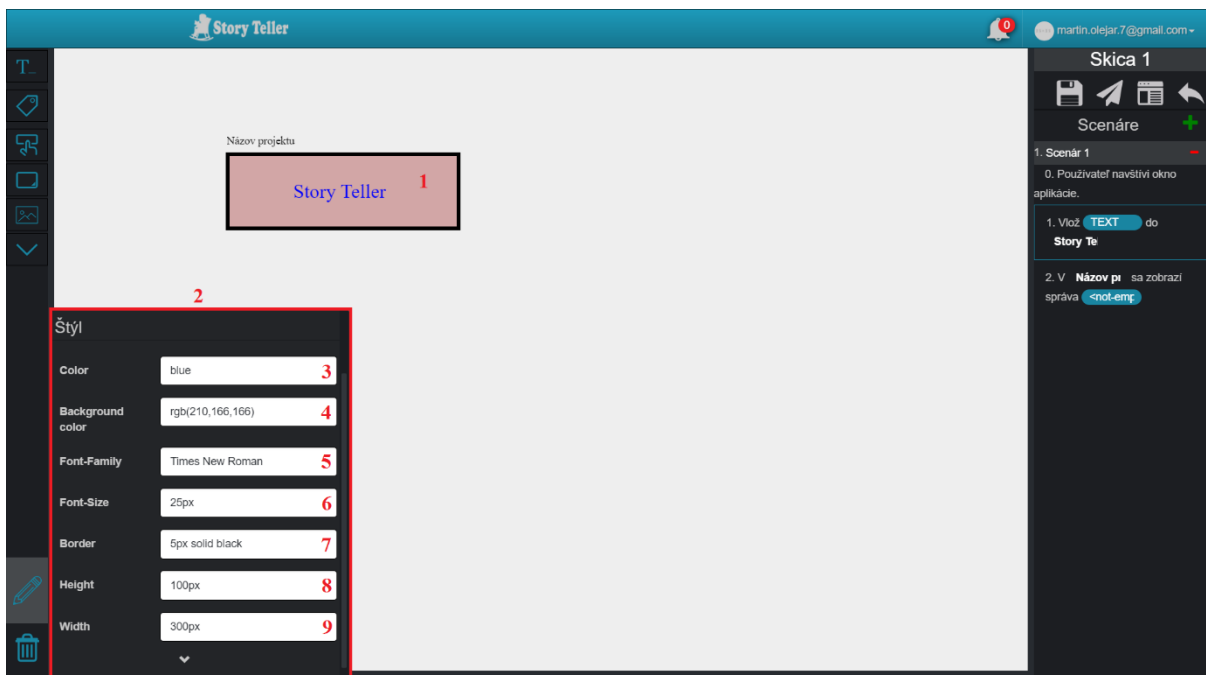


Obrázok 46: Obrazovka obsahujúca plátno pre kreslenie skice, scenáre skice a operácie.

5.3 Scenár – Úprava vlastností prvkov skice

Pre úpravu vlastností ľubovoľného prvku v skici, v obrazovke na obr. 47:

1. kliknite do prvku (označené č. 1), v ktorom môžete upravovať jeho názov,
2. zobrazí sa okno vlastností prvku označené č. 2, v ktorom môžete upravovať:
 - a. farbu textu prvku v poli č. 3,
 - b. farbu pozadia prvku v poli č. 4,
 - c. typ písma v prvku v poli č. 5,
 - d. veľkosť písma v prvku v poli č. 6,
 - e. hrúbku ohraničenia prvku v poli č. 7,
 - f. výšku prvku v poli č. 8,
 - g. šírku prvku v poli č. 9.

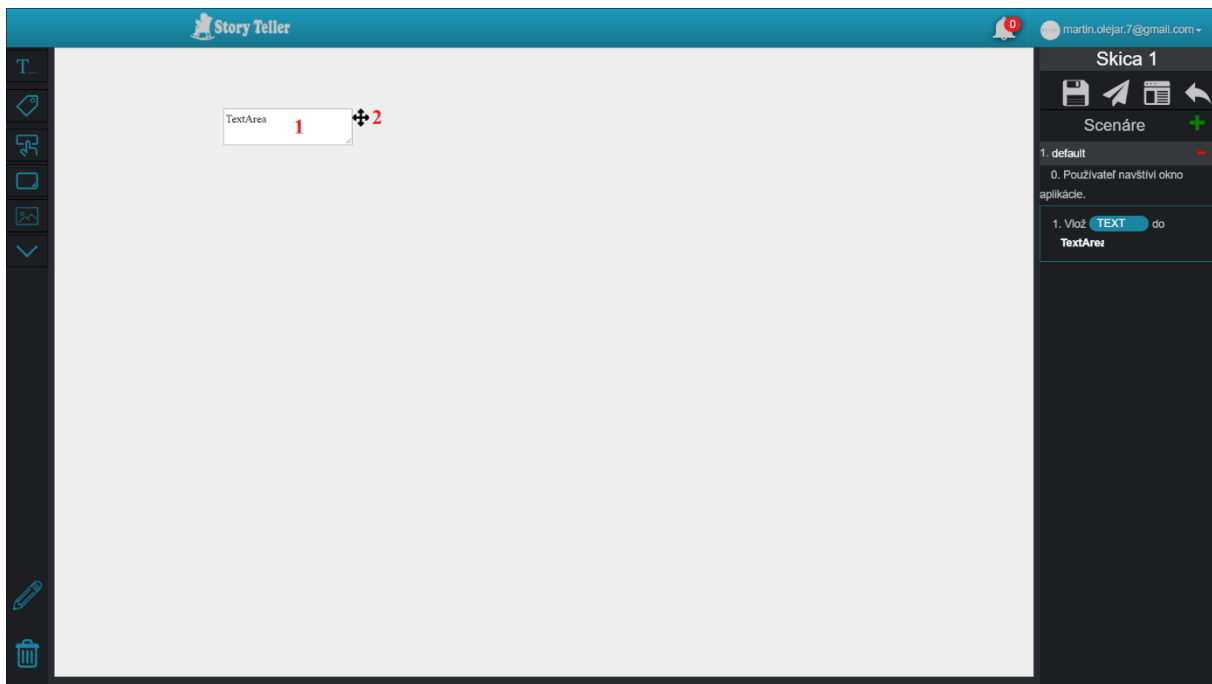


Obrázok 47: Obrazovka obsahujúca plátno pre kreslenie skice, scenáre skice a operácie.

5.4 Scenár – Posúvanie prvkov v skici

Pre posúvanie ľubovoľného prvku v skici, v obrazovke na obr. 48:

1. choďte myšou na prvok, ktorý chcete posunúť (označené č. 1).
2. zobrazí sa ikona posúvania označená č. 2, na ktorú kliknite a ťahajte na inú pozíciu v skici.

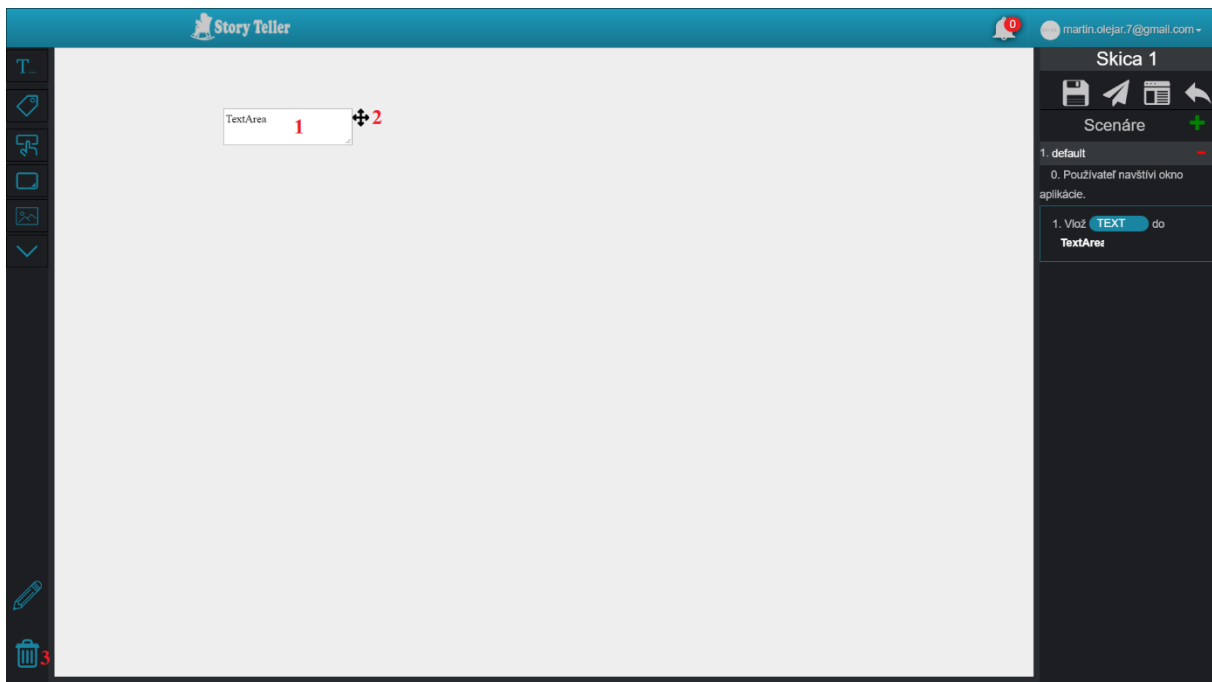


Obrázok 48: Obrazovka obsahujúca plátno pre kreslenie skice, scenáre skice a operácie.

5.5 Scenár – Odstránenie prvkov zo skice

Pre odstránenie ľubovoľného prvku zo skice, v obrazovke na obr. 49:

1. choďte myšou na prvok, ktorý chcete odstrániť (označené č. 1)
2. zobrazí sa ikona posúvania označená č. 2, na ktorú kliknite a ťahajte ju na ikonu koša označenú č. 3.

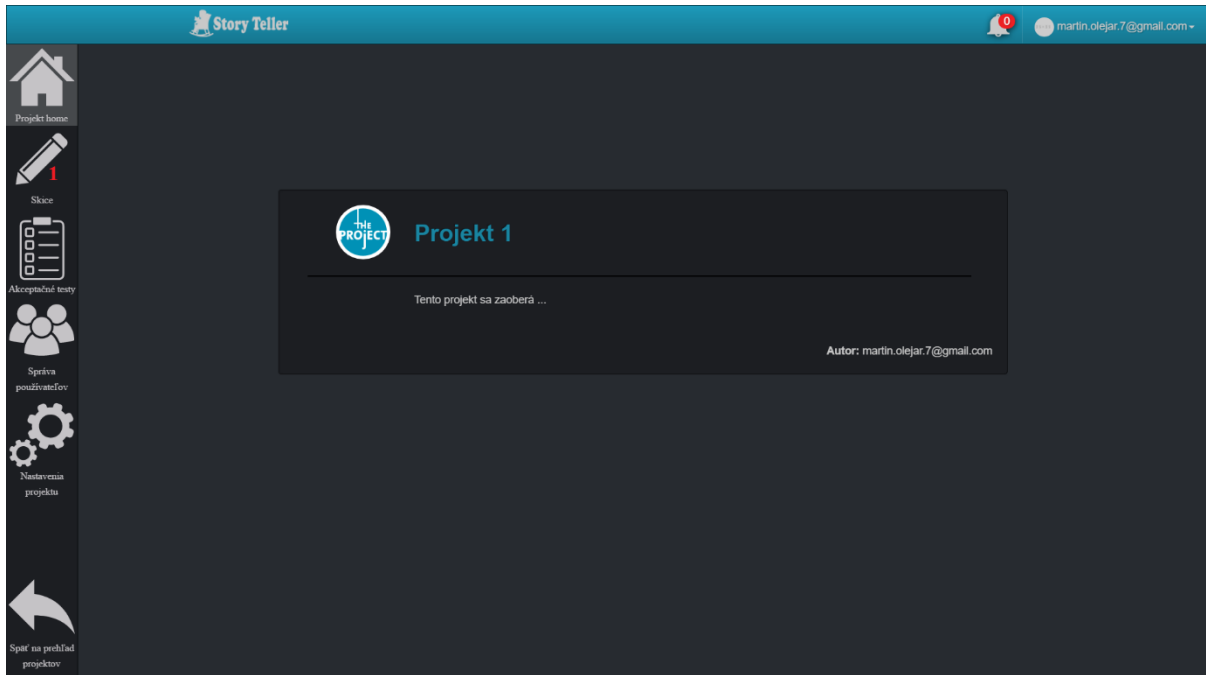


Obrázok 49: Obrazovka obsahujúca plátno pre kreslenie skice, scenáre skice a operácie.

5.6 Scenár – Zobrazenie zoznamu skíc v projekte

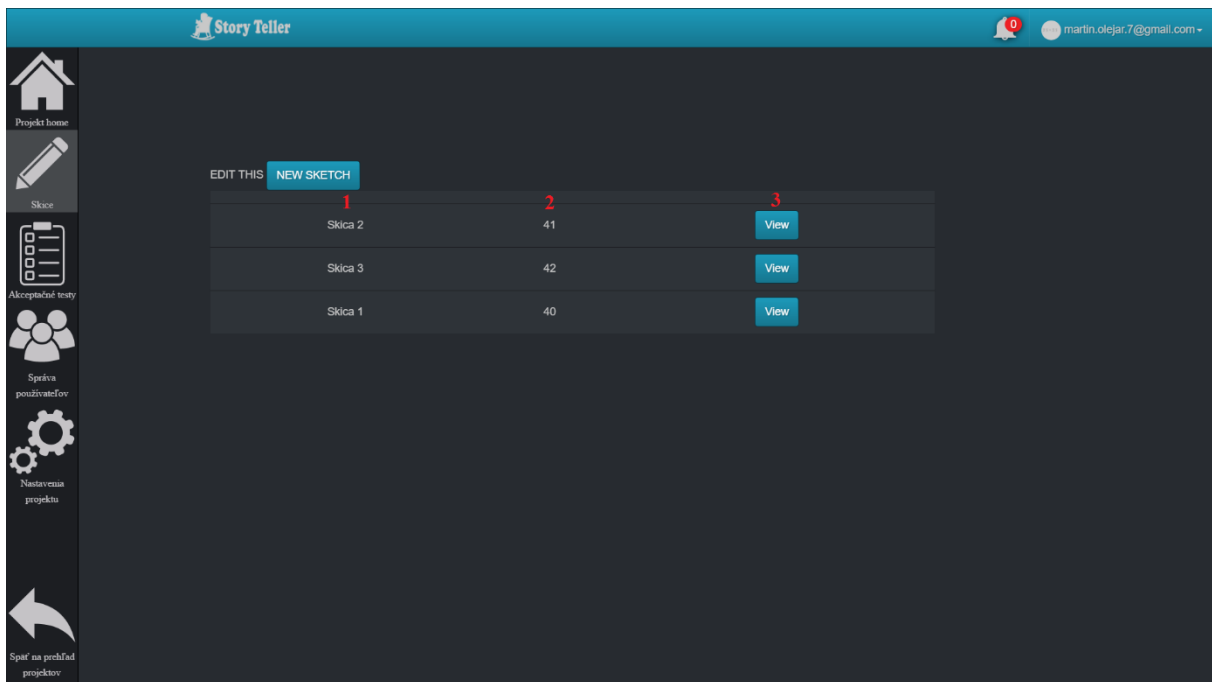
Pre zobrazenie zoznamu skíc v projekte postupujte podľa nasledujúcich krokov:

1. Vyberte si projekt, ktorého nastavenia chcete upraviť (Scenár – Výber projektu).
2. Zobrazí sa domovská obrazovka projektu na obr. 50, v ktorej kliknite na časť Skice označenú č. 1.



Obrázok 50: Domovská obrazovka projektu.

3. Následne sa zobrazí zoznam skíc v projekte na obr. 51. Tento zoznam reprezentovaný tabuľkou obsahuje pre každú skicu projektu:
 - a. jej názov v stĺpci označenom č. 1,
 - b. jej identifikátor v stĺpci označenom č. 2,
 - c. možné operácie v stĺpci označenom č. 3.

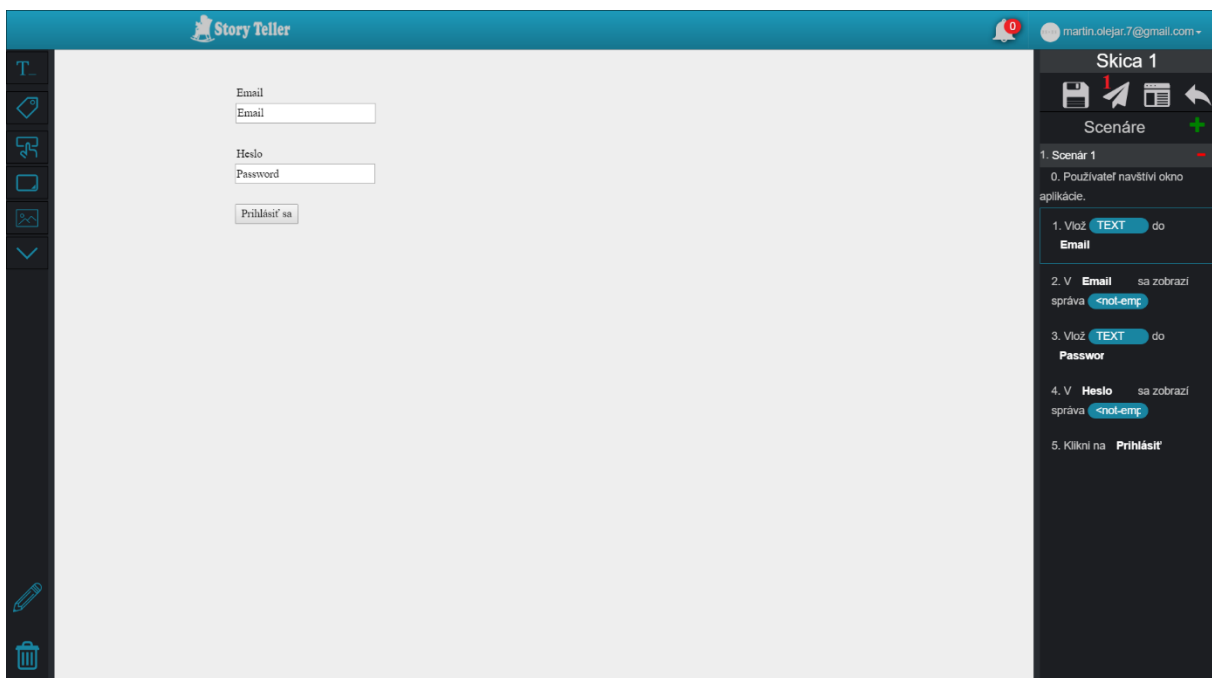


Obrázok 51: Tabuľka pre zoznam skíc v projekte.

5.7 Scenár – Zdieľanie skice

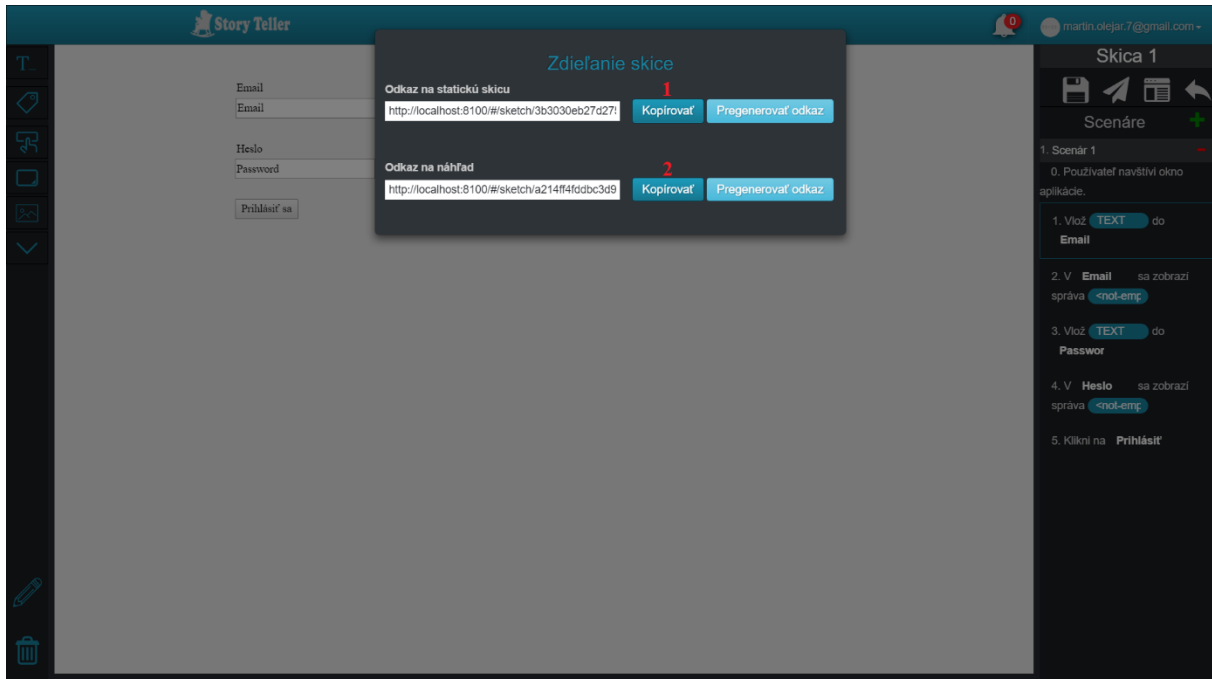
Pre stiahnutie šablóny skice postupujte podľa nasledujúcich krokov:

1. V obrazovke na obr. 52 kliknite na ikonu stiahnutia šablóny označenú č. 1.



Obrázok 52: Obrazovka obsahujúca plátno pre kreslenie skice, scenáre skice a operácie.

2. Zobrazí sa modálne okno na obr. 53, v ktorom kliknite:
 - a. na tlačidlo Kopírovať označené č. 1, ak chcete odkaz na statickú skicu,
 - b. na tlačidlo Kopírovať označené č. 2, ak chcete odkaz na náhľad.

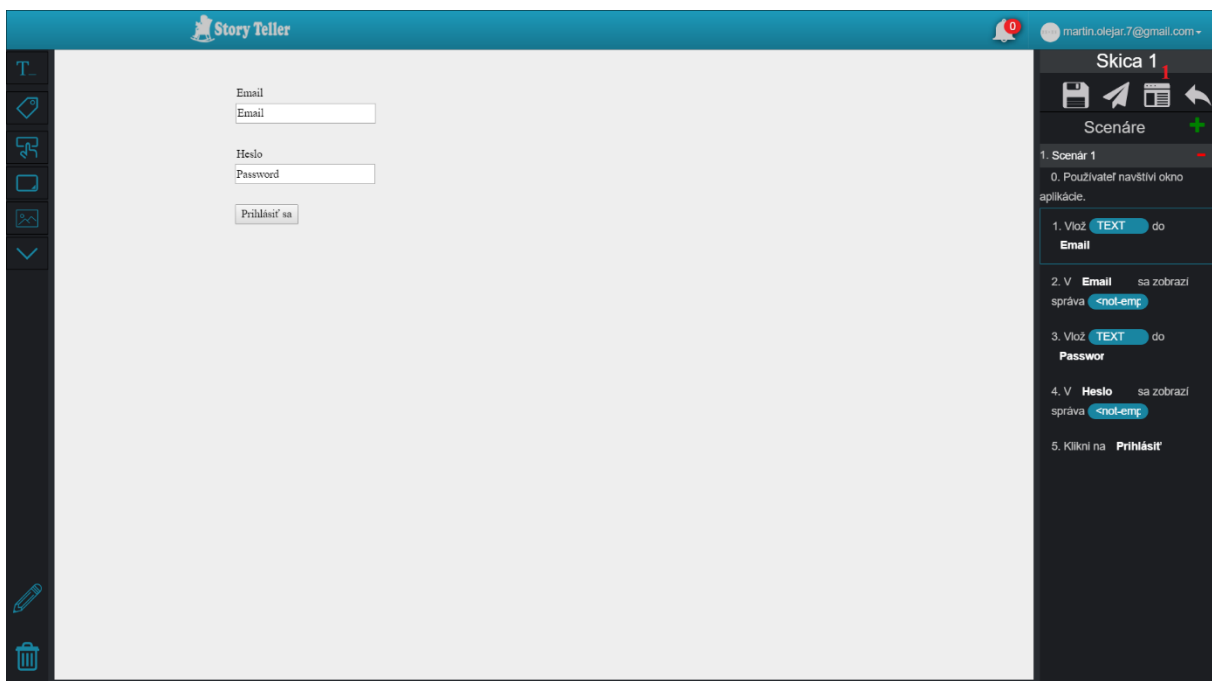


Obrázok 53: Modálne okno pre zdieľanie skice.

5.8 Scenár – Stiahnutie šablóny skice

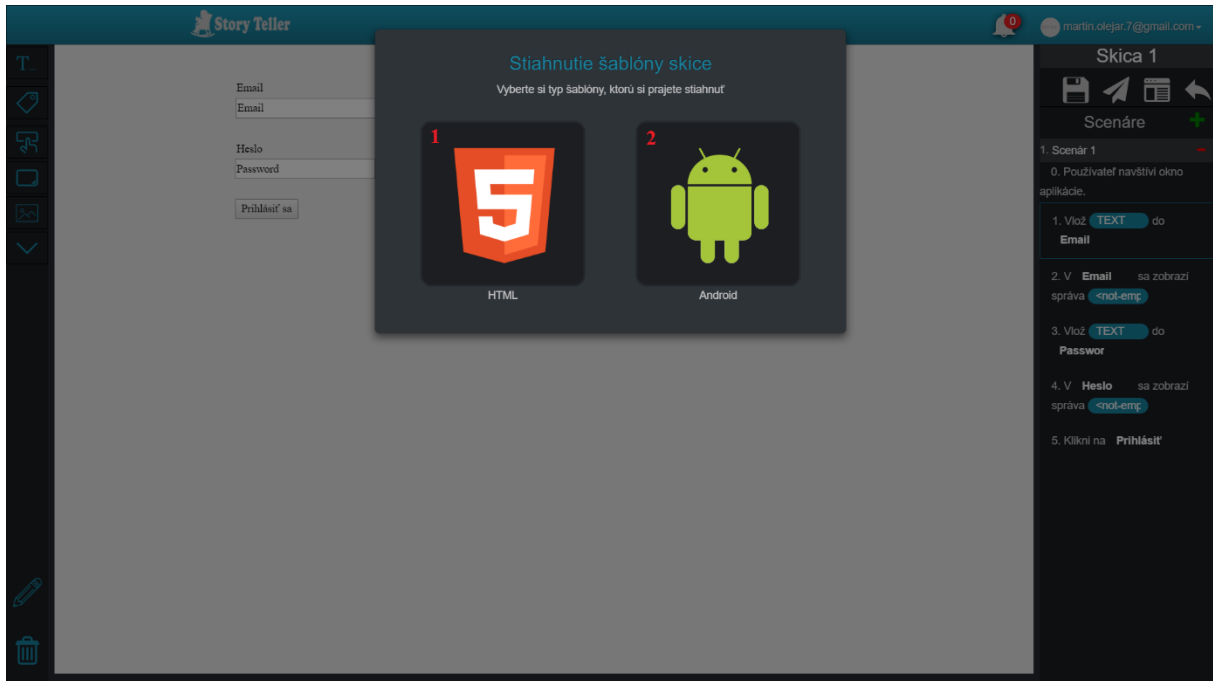
Pre stiahnutie šablóny skice postupujte podľa nasledujúcich krokov:

1. V obrazovke na obr. 54 kliknite na ikonu stiahnutia šablóny označenú č. 1.



Obrázok 54: Obrazovka obsahujúca plátno pre kreslenie skice, scenáre skice a operácie.

2. Zobrazí sa modálne okno na obr. 55, v ktorom kliknite na ikonu HTML označenú č. 1 alebo na ikonu Android označenú č. 2. Sťahovanie šablóny začne o chvíľu.

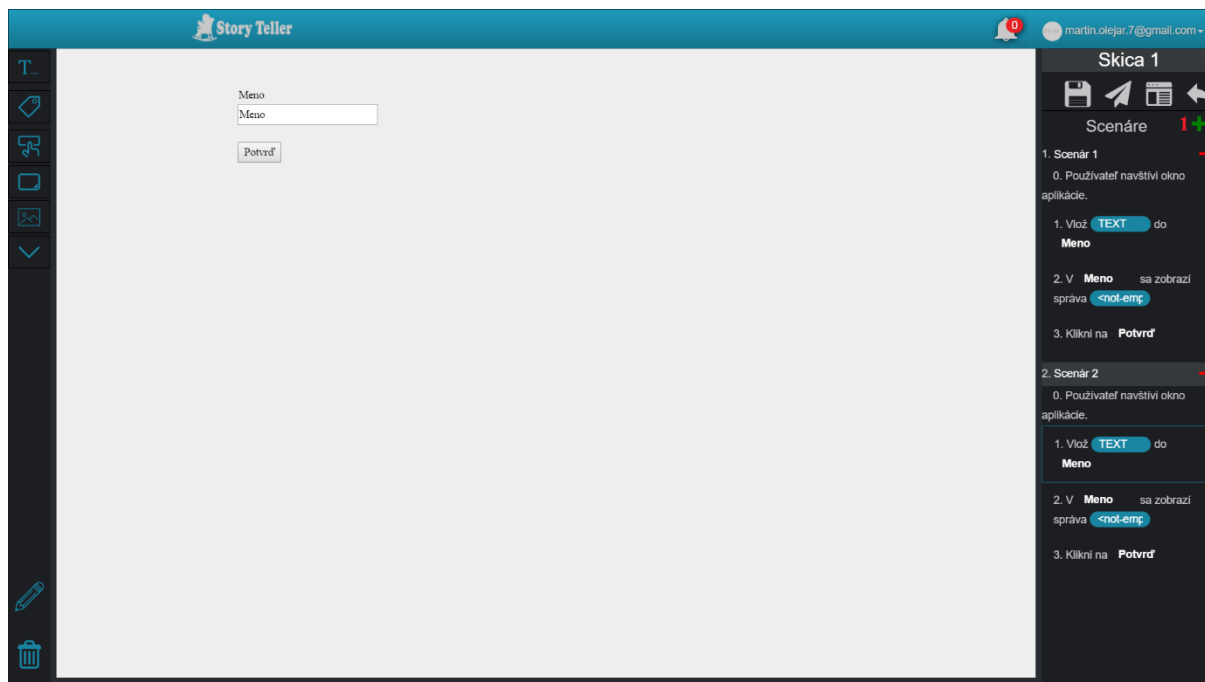


Obrázok 55: Modálne okno pre stiahnutie šablóny skice.

6 Správa scenárov

6.1 Scenár – Duplikácia scenára

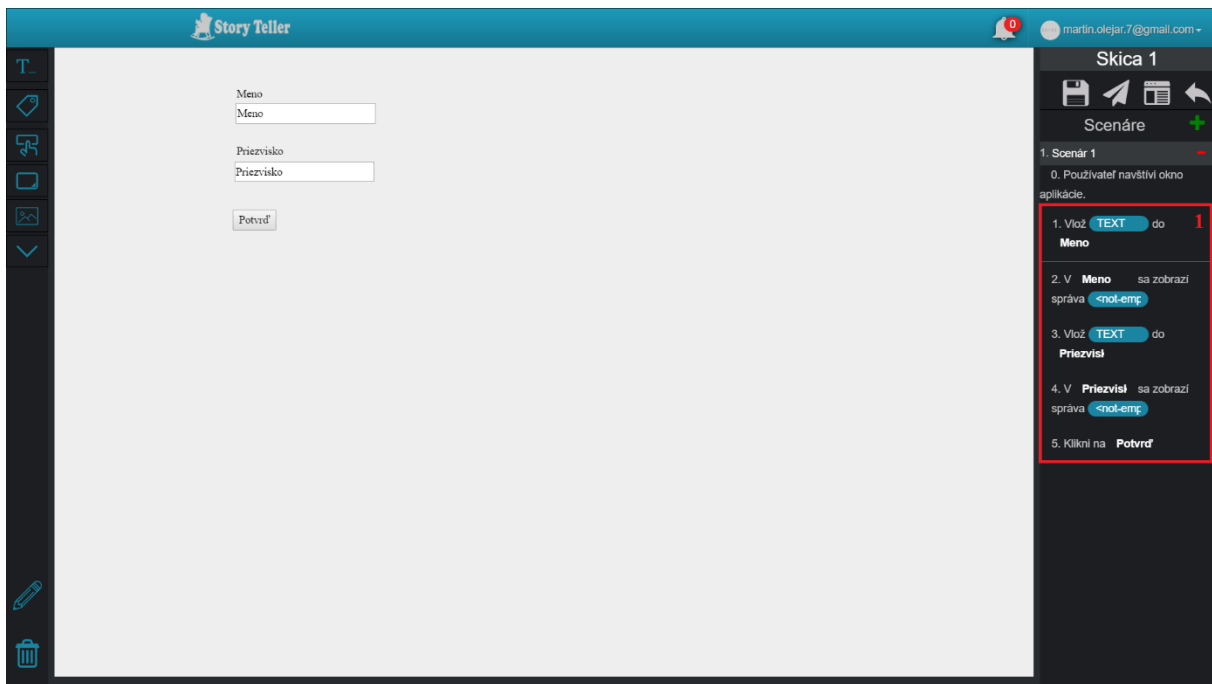
Pre duplikáciu scenára skice kliknite v obrazovke na obr. 56 na ikonu plus označenú č. 1.



Obrázok 56: Obrazovka obsahujúca plátno pre kreslenie skice, scenáre skice a operácie.

6.2 Scenár – Presúvanie krokov v scenári

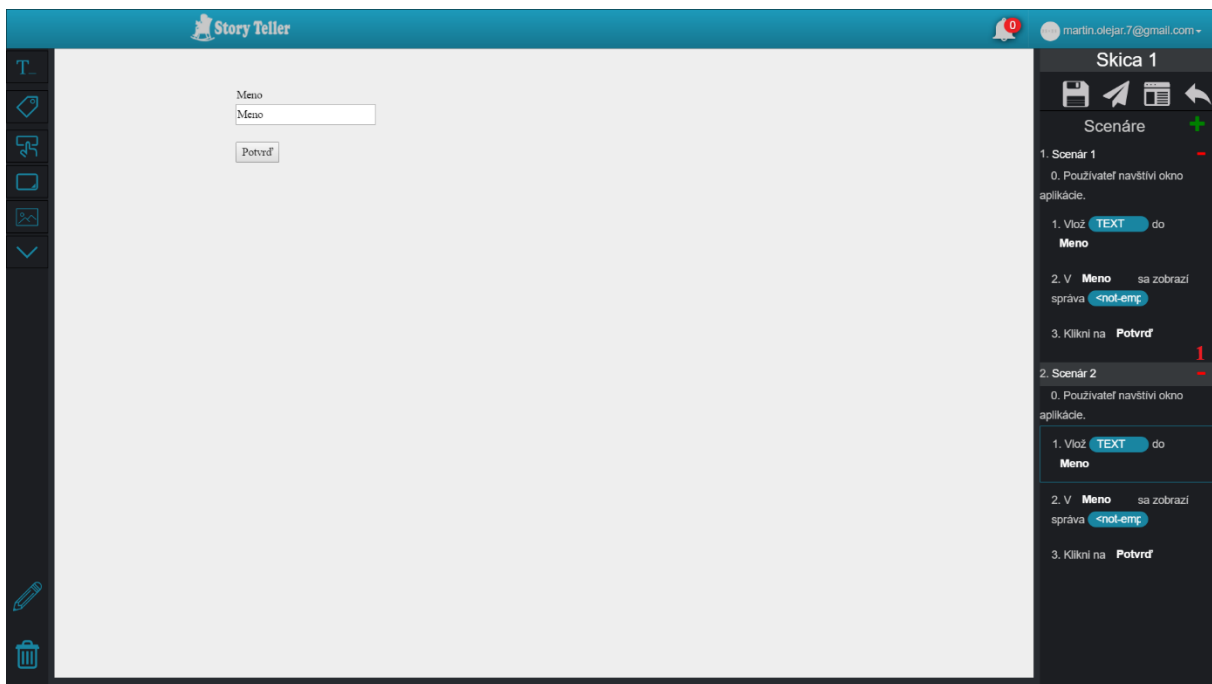
Pre presúvanie krokov v scenári skice kliknite v obrazovke na obr. 57 v časti označenej č. 1 na krok a ťahajte ho na inú pozíciu v scenári.



Obrázok 57: Obrazovka obsahujúca plátno pre kreslenie skice, scenáre skice a operácie.

6.3 Scenár – Odstránenie scenára

Pre odstránenia scenára skice kliknite v obrazovke na obr. 58 na ikonu mínus označenú č. 1.



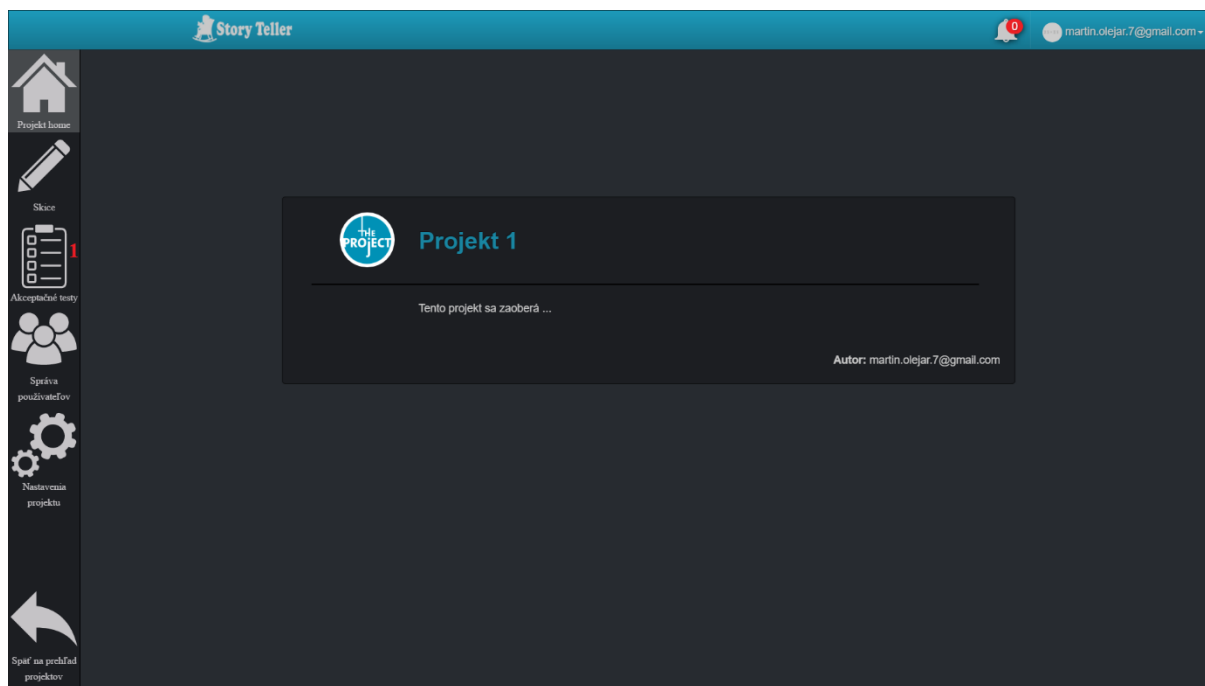
Obrázok 58: Obrazovka obsahujúca plátno pre kreslenie skice, scenáre skice a operácie.

7 Správa akceptačných testov

7.1 Scenár – Spustenie akceptačného testu

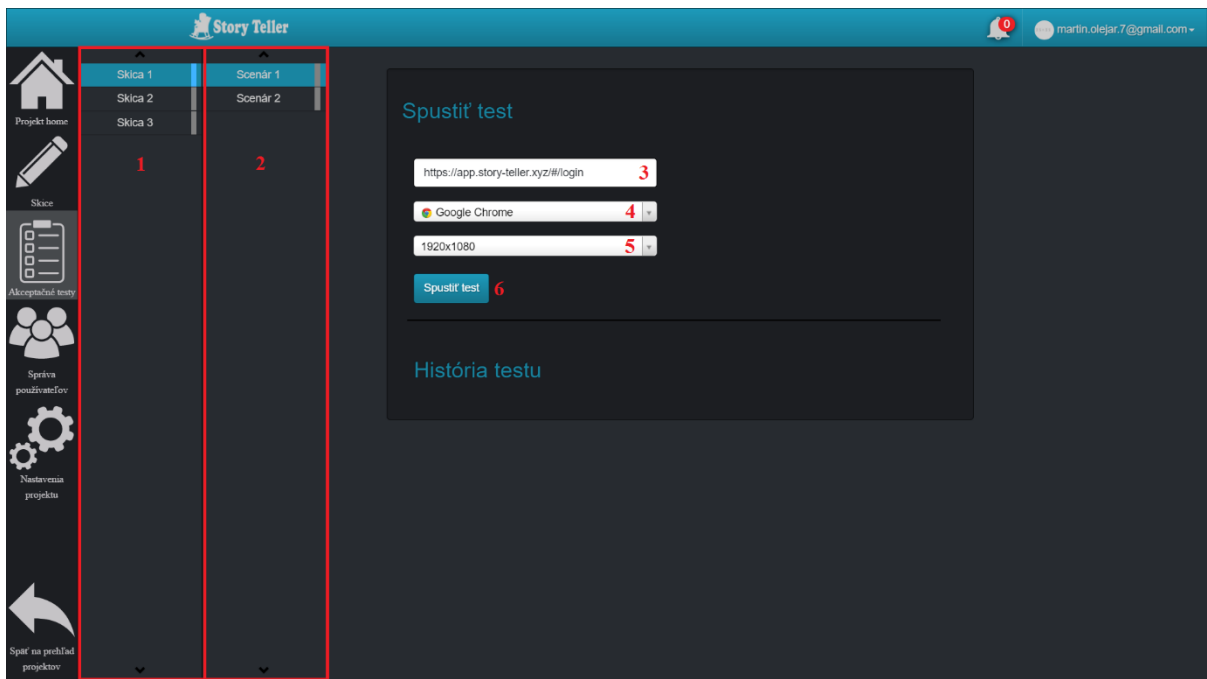
Pre spustenie akceptačného testu postupujte podľa nasledujúcich krokov:

1. Vyberte si projekt, v ktorom je testovaná skica (Scenár – Výber projektu).
2. Zobrazí sa domovská obrazovka projektu na obr. 59, v ktorej kliknite na časť Akceptačné testy označenú č. 1.



Obrázok 59: Domovská obrazovka projektu.

3. Následne sa zobrazí hlavná obrazovka akceptačných testov na obr. 60, v ktorej:
 - a. vyberte skicu v časti č. 1,
 - b. vyberte scenár patriaci zvolenej skici v časti č. 2,
 - c. vyplňte adresu testovanej stránky v poli č. 3,
 - d. vyberte prehliadač v poli č. 4,
 - e. vyberte rozlíšenie v poli č. 5,
 - f. kliknite na tlačidlo Spustiť test označené č. 6.



Obrázok 60: Hlavná obrazovka pre akceptačné testy.

Slovenská technická univerzita v Bratislave
Fakulta informatiky a informačných technológií
Ilkovičova 2, 842 16 Bratislava 4

Tímový projekt



Story Teller

Protokol o priebehu testovania

Vedúci projektu: Ing. Karol Rástočný, PhD.
Názov tímu: CoolStoryBro
Členovia tímu: Bc. Jakub Ondik
Bc. Patrik Januška
Bc. Adam Neupauer
Bc. Martin Olejár
Bc. Miroslav Hurajt
Kontakt: storyteller-04@googlegroups.com
Akademický rok: 2016/2017
Vypracoval: Bc. Martin Olejár

1 Testované scenáre

Pred testovaním sme si pripravili scenáre, ktorých otestovaním sme chceli čo najviac pokryť funkcionálnosť nášho systému. V nasledujúcich podkapitolách sú tieto scenáre opísané v takom poradí, v akom sa vykonávali.

1.1 Scenár č. 1 – Registrácia a prihlásenie

Pre respondentov sme pripravili webový prehliadač s otvorenou aplikáciou (<https://app.story-teller.xyz/#/login>). Úlohou respondenta bola registrácia a následné prihlásenie do systému. Registrácia zahŕňa presmerovanie na registračný formulár, vyplnenie emailu, hesla a potvrdzovacieho hesla a kliknutie na aktivačný odkaz v registračnom emaile. Pre prihlásenie je nutné zadať email a heslo.

1.2 Scenár č. 2 – Vyplnenie údajov o sebe

Po prihlásení sme požiadali respondentov o vyplnenie údajov o sebe vo svojom profile. Pre realizáciu tohto scenára bolo nutné vyhľadať v navigačnom paneli tlačidlo na úpravu profilu a následne vybrať profilovú fotku, vyplniť meno, priezvisko a domovskú stránku a vybrať lokalizáciu.

1.3 Scenár č. 3 – Zmena hesla

Ďalším scenárom bola zmena hesla, čo zahŕňalo vyhľadanie tlačidla na úpravu profilu v navigačnom paneli a vyplnenie starého hesla, nového hesla a potvrdzovacieho hesla. Úlohou respondenta bolo odskúšať zmenené heslo odhlásením sa a následným prihlásením sa.

1.4 Scenár č. 4 – Zabudnutie hesla

Potom sme požiadali respondenta, aby sa odhlásil zo systému a predstavil si situáciu, že zabudol svoje heslo. Tento scenár zahŕňa presmerovanie na formulár pre zabudnutie hesla a vyplnenie emailu používateľa. Následne je poslaný email používateľovi, ktorý obsahuje nové heslo. Respondent musel odskúšať nové vygenerované heslo.

1.5 Scenár č. 5 – Vytvorenie projektu

Ďalšou úlohou respondenta bolo vytvorenie projektu, čo zahŕňalo presmerovanie na formulár pre vytvorenie projektu z hlavnej stránky, vyplnenie názvu a popisu projektu a výber loga.

1.6 Scenár č. 6 – Úprava informácií o projekte

Tento scenár zahŕňal presmerovanie na časť Nastavenia projektu a úpravu názvu a popisu projektu a výber iného loga.

1.7 Scenár č. 7 – Prezeranie notifikácií

Ďalšou úlohou respondenta bolo prezeranie notifikácií, ktoré je možné vykonať kliknutím na zvonček v navigačnom paneli.

1.8 Scenár č. 8 – Zmena vlastnej roly v projekte

Pre vytvorený projekt bolo nutné zmeniť rolu autora projektu. To zahŕňa presmerovanie na časť Správa používateľov, kliknutie na konkrétneho používateľa a kliknutie na inú rolu v zozname.

1.9 Scenár č. 9 – Vytvorenie skice

Tento najväčší scenár zahŕňal vykonanie nasledujúcich akcií:

1. vytvorenie skice s 2 popisnými poľami a 1 textovým poľom,
2. zmena grafických vlastností prvkov v skici,
3. presúvanie prvkov skice,
4. odstránenie prvkov skice,
5. uloženie skice,
6. zdieľanie skice,
7. zmena poradia krokov v scenári,
8. pridanie a odstránenie scenára v skici,
9. zmena názvu scenára.

1.10 Scenár č. 10 – Pridanie používateľa do projektu

Cieľom tohto scenára bolo len nájsť rozhranie, prostredníctvom ktorého je možné pridať používateľa do projektu.

1.11 Scenár č. 11 – Spustenie testu

Úlohou respondenta bolo spustenie testu pre už vytvorený scenár. To je možné realizovať presmerovaním na časť Akceptačné testy, kliknutím na skicu, jej scenár, vyplnenie URL adresy, výber prehliadača a rozlíšenia a kliknutím na tlačidlo.

1.12 Scenár č. 12 – Vytvorenie formulára pre prihlásenie

Tento scenár zahŕňal vytvorenie novej skice, ktorá mala obsahovať typický prihlasovací formulár, konkrétne 2 popisné polia, 2 textové polia a tlačidlo. Všetky tieto komponenty mali byť v kontajneri.

2 Výsledky testovania

Na testovanie sme pozvali 2 respondentov, ktorí mali štandardné znalosti v oblasti IT. Pre každý scenár uvádzame problémy, ktoré sa vyskytli pri ich riešení. Na konci sú spomenuté pripomienky, ktoré nám dali respondenti počas testovania a po testovaní.

2.1 Problémy pri riešení scenárov

2.1.1 Scenár č. 1 – Registrácia a prihlásenie

1. respondent:

- Po kliku na aktivačný odkaz v registračnom emaille sa zobrazil registračný formulár s chybovou hláškou o vypršaní času na aktiváciu účtu. Mal sa zobrazit' prihlasovací formulár s hláškou o úspešnej registrácii.

2. respondent:

- Webový prehliadač bol v angličtine a v registračnom formulári boli slová Confirm password napísané pod sebou, nie vedľa seba.
- Potom, ako vyplnil prihlasovacie heslo bez aktivácie účtu, boli zobrazené 2 hlášky (nesprávne heslo aj informácia o poslaní registračného emailu).

2.1.2 Scenár č. 2 – Vyplnenie údajov o sebe

1. respondent:

- Respondent nevedel, čo sa myslí pod domovskou stránkou. Nemal žiadnu domovskú stránku a pritom nevedel, či ide o povinný údaj.

2.1.3 Scenár č. 3 – Zmena hesla

2. respondent:

- Zmenu hesla potvrdil kliknutím na Enter, nič sa nestalo.

2.1.4 Scenár č. 4 – Zabudnutie hesla

Nevyskytli sa žiadne problémy u oboch respondentov.

2.1.5 Scenár č. 5 – Vytvorenie projektu

Nevyskytli sa žiadne problémy u oboch respondentov.

2.1.6 Scenár č. 6 – Úprava informácií o projekte

1. respondent:

- Časti Nastavenia projektu a Spät' na zoznam projektov sa prekrývajú.

2.1.7 Scenár č. 7 – Prezeranie notifikácií

1. respondent:

- Hľadal notifikácie v ľavom paneli, zmiatla ho ikona pre akceptačné testy, podľa respondenta pripomína zoznam notifikácií, poznámkovač.

2.1.8 Scenár č. 8 – Zmena vlastnej roly v projekte

1. a 2. respondent:

- Po zmene roly bola vypísaná hláška Nepodarilo sa zmeniť rolu – implementačná chyba.

2.1.9 Scenár č. 9 – Vytvorenie skice

1. respondent:

- Klikol na prvok skice, nedržal tlačidlo myši a klikol do plátna skice.
- Vlastnosti v dolnej časti okna pre vlastnosti prvku skice neboli dobre graficky znázornené.
- Hľadal tlačidlo na potvrdenie zmeny vlastností prvku skice.
- Nevedel, ktorý prvok je popisné pole.
- Realizoval odstránenie prvku skice potiahnutím kroku scenára do koša.
- Mal problém s odstránením kontajnera, potiahnutie prvku na kôš nefungovalo.
- Nevedel, ako zatvoriť okno pre vlastnosti prvku skice.
- Pri zmene názvu scenára dvakrát klikol na názov scenára, čím sa scenár zavrel.
- Chcel meniť názov prvkov v krokoch scenára.

2. respondent:

- Klikol na prvok skice, nedržal tlačidlo myši a klikol do plátna skice.
- Nevedel, ktorý prvok je popisné pole. Obrázky prvkov skice nie sú jasné.
- Chcel meniť názov prvkov skice v okne vlastností prvku skice.

2.1.10 Scenár č. 10 – Pridanie používateľa do projektu

Nevyskytli sa žiadne problémy u oboch respondentov.

2.1.11 Scenár č. 11 – Spustenie testu

2. respondent:

- Po kliknutí na spustenie testu sa zobrazila hláška, ktorá neobsahovala počet sekúnd do začatia testu.

2.1.12 Scenár č. 12 – Vytvorenie formulára pre prihlásenie

1. respondent:

- Nefungovalo vkladanie už vytvorených prvkov do kontajnera.

2.2 Pripomienky od respondentov

1. respondent:

- Doplňenie tlačidla na presmerovanie na projekt pri vytváraní skice.
- Zmena farby na sivú farbu pre názov prvku v kroku skice.
- Zobrazenie okna pre vlastnosti prvkov skice po kliknutí na prvok skice.
- Nahradenie hviezdčky ceruzkou pri okne pre vlastnosti prvkov skice.

2. respondent:

- Automatické vyplnenie mena, priezviska, emailu podľa uložených údajov v prehliadači.
- Presmerovanie na projekt po potvrdení zmeny informácií o projekte.
- Nahradenie písania farby prvku skice za color picker v okne pre vlastnosti prvkov skice.
- Priame vyplnenie odkazov pri zdieľaní skice bez zbytočného klikania.
- Pridanie upozornenia pri odstránení scenára skice.
- Pridanie upozornenia na uloženie skice po kliknutí na šípku späť.
- Nedostatočná spätná väzba po vykonaní akcií.
- Presun šípky späť vo vytváranej skici do ľavej hornej strany.
- Oprava zobrazenia názvu prvku pri jeho ťahaní do skice.
- Zväčšenie oblasti pre ťahanie prvku skice.