

Semestrální práce z KIV/NET

# Implementace webové aplikace pro tvorbu kvízů

Přemysl Kouba  
A21N0056P  
kouba.prem@gmail.com

25. 05. 2022

# Obsah

<b>1</b>	<b>Zadání</b>	<b>1</b>
1.1	Popis a cíl projektu . . . . .	1
1.2	Požadavky . . . . .	1
1.3	Možná rozšíření . . . . .	1
<b>2</b>	<b>Analýza</b>	<b>3</b>
<b>3</b>	<b>Implementace</b>	<b>4</b>
3.1	Kontrolery . . . . .	4
3.1.1	Správa uživatelů . . . . .	4
3.1.2	Správa kvízů . . . . .	4
3.1.3	Veřejný přístup . . . . .	5
3.2	Data . . . . .	5
3.3	Manažer . . . . .	5
3.4	Models . . . . .	5
3.5	Tests . . . . .	6
3.6	Views . . . . .	6
3.7	Stavy kvízu . . . . .	7
<b>4</b>	<b>Programátorský deník</b>	<b>8</b>
<b>5</b>	<b>Uživatelská příručka</b>	<b>9</b>
5.1	Přihlášení . . . . .	10
5.2	Funkce kvízů . . . . .	11
5.2.1	Zobrazení stavu a výsledků kvízu . . . . .	11
5.2.2	Práce s uživateli kvízu . . . . .	12
5.2.3	Vyplňování kvízu . . . . .	13
5.2.4	Vytvoření kvízu . . . . .	15
5.3	Veřejný přístup . . . . .	16
<b>6</b>	<b>Závěr</b>	<b>17</b>
	<b>Literatura</b>	<b>18</b>

# 1 Zadání

Vytvořte webovou aplikaci, která umožní uživatelům efektivněji nakládat se svým volným časem za pomoci využití kvízů, které mezi sebou uživatelé sdílí.

## 1.1 Popis a cíl projektu

Aplikace umožní uživateli vytvářet/vyplňovat a sdílet kvízy (dotazníky). Po vyplnění kvízu proběhne vyhodnocení, které umožní uživatelům rozhodnout která z odpovědí je jejich volba

## 1.2 Požadavky

Požadavky na aplikaci se dělí na povinné, které jsou od aplikace očekávány a na volitelné, které by bylo vhodné implementovat. Následující kapitola popisuje povinné požadavky. Vytvořit webovou aplikaci za použití programovacího jazyka .NET s využitím perzistentního úložiště. Dodržení MVC architektury, která bude řádně otestovaná a zdokumentovaná. Uživatel:

- správa kvízů
- vyplňování kvízů
- vytváření kvízů
- historie kvízů + vyhodnocení

Kvíz:

- umožnění přidání dalších otázek v průběhu vyplňování kvízu
- sdílení kvízů s uživateli (nebo emailem)

Odpovědi

- možnost odpovědět ANO/NE/Když ostatní chtějí

## 1.3 Možná rozšíření

Aplikace by v ideálním případě měla být dostupná v Cloudu. Uživatel:

- správa uživatelů (registrace/autorizace)

Otázky:

- Automatické napovídání otázek podle tagů
- Automatické napovídání otázek podle historie

Kvíz:

- public / protected kvízy

- volba vhodné odpovědi pomocí náhody (např. aplikace „random wheel“)

## 2 Analýza

První reference<sup>1</sup> je internetový kvíz, který umožňuje uživatelům odpovídat na otázky ohledně činností ANO/NE/ <když někdo další> chce. Po vyplnění kvízu oběma zúčastněnými, přišlo vyhodnocení, kde se ukázalo kde se oba nejvíce shodují a kde mají největší shodu. Bohužel tato aplikace neběžela úplně podle potřeb. Občas se uživatelům s odkazem vůbec nenačetla. Další nevýhodou bylo, že po vyplnění dat občas nedošlo ke správnému odeslání a uživatel tak ztratil svoje odpovědi. To je poměrně krutá vlastnost pro dotazník s 50 a více otázkami.

Další referencí je aplikace Doodle[2], kterou myslím všichni znají. Ta je používána pro synchronizaci činností a umožňuje odpovídat ANO/NE/Možná. Aplikace Doodle představuje dobrý model, ze kterého lze vyjít.

Aplikace je primárně určena k domluvě termínů a ne toho, co dělat ve volném čase. Navíc neumožňuje znovuvyužití otázek pro další použití. Další vlastností, která u Doodle chybí je možnost automatického vyhodnocení odpovědí. Zákazník (já) by využil možnosti náhodné volby z množiny vhodných odpovědí.

Další aplikací, kterou v současné době používám k podobné funkci je aplikace „Picker wheel“[1], která umožňuje náhodný výběr ze zadaných aktivit. Problémem s touto aplikací je, že nedrží historii možností a navíc možnosti může vkládat jen jeden a nejsou priorizovány.

---

<sup>1</sup>Bohužel jsem odkaz na tuto aplikaci ztratil, takže nemohu referencovat

## 3 Implementace

Implementace je volně dostupná na [gitu\[3\]](#)<sup>1</sup>, kde je vidět i moje aktivita v průběhu semestru. Při implementaci aplikace jsem se snažil dodržet architekturu MVC. Občas jsou viditelné chyby v architektuře kterých jsem se dopustil. Hustota chyb stoupá s časem který jsem na aplikaci strávil. A to hlavně vzhledem k tomu, že jsem si vědom toho, že jsem na vývoji strávil spoustu hodin a výsledek není takový jaký bych si představoval. Aplikace je rozdělena do následujících složek:

- Controllers – kontrolery pro aplikaci (Quiz, Question, User)
- Data - Obsahuje část logiky aplikace. Služby (Services) a databázový kontext pro práci s databází
- Doc - dokumentace
- Drivers - složka s drivery pro testování Seleniem
- Managers - řízení logiky
- Models - obsahuje modelové třídy a objekty používané napříč aplikací
- Tests – testování aplikace
- Views - příslušné pohledy nad datovými modely

### 3.1 Kontrolery

Aplikace obsahuje tři kontrolery - pro správu uživatelů, správu kvízů a veřejný přístup do aplikace.

#### 3.1.1 Správa uživatelů

Je definována v souboru **AccountController**. V souboru je logika pro správu uživatelů - jejich přihlašování, registrování a odhlašování

#### 3.1.2 Správa kvízů

Soubor **QuizController** obsahuje logiku pro práci s kvízy. Kvíz v současné verzi umožňuje následující funkčnost:

- tvorbu kvízu
- vyplňování
- sdílení s dalšími uživateli
- přidávání otázek do již vytvořeného kvízu

---

<sup>1</sup><https://github.com/Sedmikras/DayOfFun>

- zobrazení informací o uživateli kteří se účastní kvízu
- smazání kvízu
- zobrazení kvízů pro daného uživatele

### 3.1.3 Veřejný přístup

Uživatelé mají přístup do aplikace přes veřejný přístup, kdy není nutné být přihlášený. Jsem si zcela vědom toho, že zde je velká bezpečnostní trhlina a není jediná (je jich skutečně mnoho). Uživatelé ve veřejném přístupu jsou identifikováni pomocí emailu.

Kontroler pro veřejný přístup je v souboru **PublicController** a obsahuje následující funkčnost:

- vyplňování
- přidávání otázek do již vytvořeného kvízu
- zobrazení informací o uživateli kteří se účastní kvízu
- zobrazení kvízů pro daného uživatele

## 3.2 Data

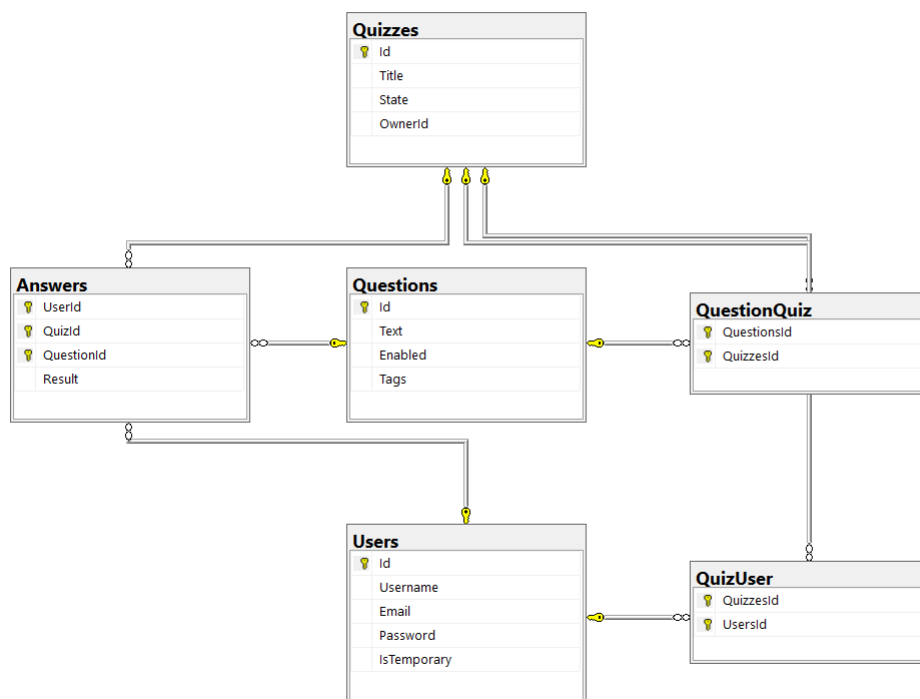
Datová vrstva aplikace, která zprostředkovává přístup do databáze. Přístup není rovnou přes kontext, ale přes jednotlivé služby, které uvnitř využívají kontext (snad bezpečně).

## 3.3 Manažer

Soubor **ApplicationManager** obsahuje hlavní logiku aplikace. Každý kontroler má referenci na tuto třídu a volá některou z metod.

## 3.4 Models

Obsahuje třídy použité pro komunikaci s databází a ukládání dat. Využil jsem přístupu Code-first a tak je databáze odrazem obsahu této složky. Na obrázku 3.1 je vidět relační schéma použité databáze.



Obrázek 3.1: Relační schéma databáze

## 3.5 Tests

Pro otestování aplikace jsem se rozhodl k využití frameworku Selenium společně s NUnit. Musím říct, že použití frameworku v .NET je poměrně složité a „ukecané“. Psaní scénářů mi zabralo poměrně hodně času.

I z tohoto důvodu testování pokrývá jen základní scénáře použití + zkoušení krajních hodnot a chybových stavů, které je potřeba si ohlídat.

Cílem bylo, aby bylo možné testy znovuspouštět mnohokrát po sobě a aby na sobě testy datové nezávisely.

## 3.6 Views

Tato složka obsahuje pohledy, které vytvářejí frontend pro uživatele. Vycházím z výchozí šablony pro vývoj webových aplikací a tudíž frontend využívá



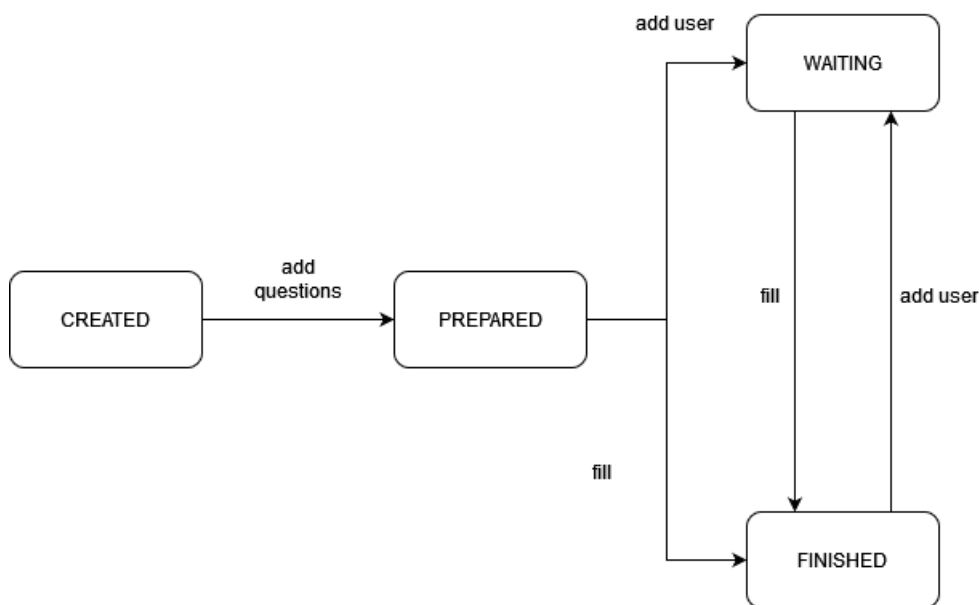
Javascript, JQuery a pro vzhled je využíván Bootstrap.

V aplikaci používám nejen základní „layout“, ale také částečné pohledy pro modální okna. Použití je například při přidávání otázek do kvízu, nebo také při přidávání uživatelů.

### 3.7 Stavy kvízu

Kvíz má definováno několik stavů a akcí, které může v těchto stavech nabízet:

- **CREATED** - nenabízí žádné akce, je potřeba doplnit otázky
- **PREPARED** - kvíz je připraven k vyplnění, akce jsou **delete**, **view users**, **fill**
- **WAITING** - kvíz čeká na některého z účastníků, akce jsou **view users**, **fill**
- **FINISHED** - kvíz byl úspěšně vyplněn všemi. Akce jsou **view users**, **view results**, **delete**
- **INVALID** - kvízy v tomto stavu se uživateli neukazují. Tento stav může nastat uvnitř aplikace (ale nemělo by k tomu dojít) a jedná se o nekonzistenci dat.



Obrázek 3.2: Schéma stavy kvízu

## 4 Programátorský deník

Programátorský deník se snaží zachytit čas který jsem strávil na jednotlivých aktivitách. Vedený byl v textové podobě v souboru **sp\_doc.odt** ve složce Doc.

Datum	Aktivita	Délka [hod]
12/03/22	Příprava modelu (v Excelu) a vyzkoušení zda-li je to funkční princip	3
18/03/22	Vytvoření GIT repozitáře	1
18/03/22	Nástřel designu	2
30/03/22	Příprava modelu	4
14/04/22	Přechod na Rider	2
15/04/22	Test driven vývoj prototypu na zpracování kvízu	4
15/04/22	Praní se s vazbami + entity frameworkem	6
17/04/22	Prototyp MVC architektury (vytvoření prvního náhledu) – studium + vývoj	9
17/04/22	Dokumentace	1
21/04/22	Vývoj	4
22/04/22	Vývoj – Create View návrh	1
23/04/22	Vývoj – dopracování Create View návrhu + registrace	10
24/04/22	Vývoj – vyplňování kvízu	8
25/04/22	Vývoj – velký refaktor – změna modelu, protože EF nefungoval úplně tak jak měl. Větší oddělení Servis a Controllerů	6
27/04/22	Detail kvízu příprava na share a logika pro temporary účet	3
28/04/22	Autocomplete + prefill quiz with answers	2
04/05/22	Velký refaktor	6
06/05/22	Refaktor + login / registrace a úprava indexu (ve vlaku)	2
21/05/22	Testování + bugfixing	6
22/05/22	Testování, bugfixing, doplnění fýčur, finišování	7
23/05/22	Dokumentace, validace, odevzdávka, testování, doplnění otázek	3
25/05/22	Doplnění komentářů, dokumentace, odevzdání	5
	<b>CELKEM</b>	<b>95</b>

Obrázek 4.1: Programátorský deník

## 5 Uživatelská příručka

Před spuštěním aplikace je nutné nechat si vygenerovat databázi. Aplikace využívá jako databázový systém Microsoft SQL server. Vygenerování je možné ve složce projektu aplikace pomocí příkazu

```
dotnet ef migrations add <název>
```

Ten nejprve vygeneruje migraci a následně je nutné zavolat příkaz

```
dotnet ef database update
```

Nebo příslušné alternativy pro Visual Studio<sup>1</sup>

Aplikaci je možné spustit buď pomocí souboru DayOffun.exe, nebo přes IDE.

Přes prozkoumání doporučuji spustit testování aplikace které je v souboru **SeleniumTests.cs**. Testování aplikace založí uživatele pro přihlášení a první kvíz. Navíc je možné pochytit funkčnost aplikace.

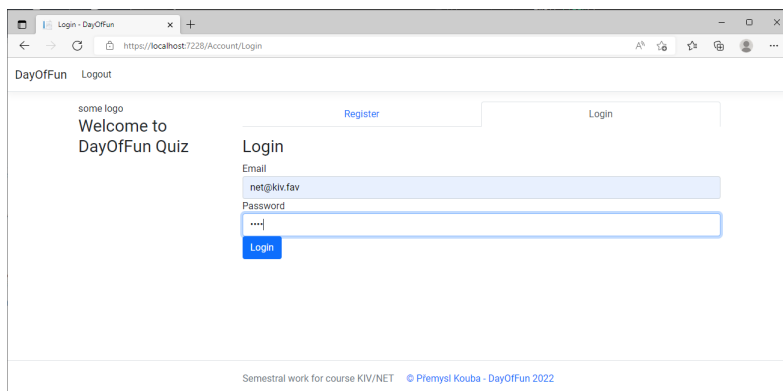
Na následujících stránkách bude znázorněn postup jak používat tuto aplikaci.

---

<sup>1</sup><https://docs.microsoft.com/cs-cz/ef/core/managing-schemas/migrations/?tabs=dotnet-core-cli>

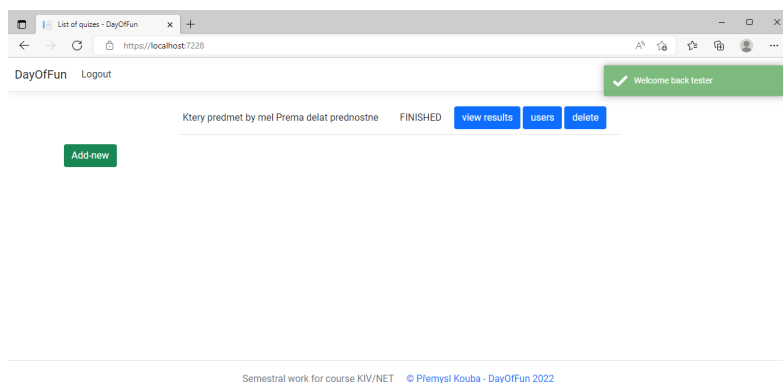
## 5.1 Přihlášení

Po proběhnutí testů je možné se přihlásit do aplikace pomocí účtu **net@kiv.fav** s heslem **test**



Obrázek 5.1: Přihlášení uživatele

Po přihlášení je uživatel přesměrován na seznam kvízů, kterých se účastní (obrázek 5.2)

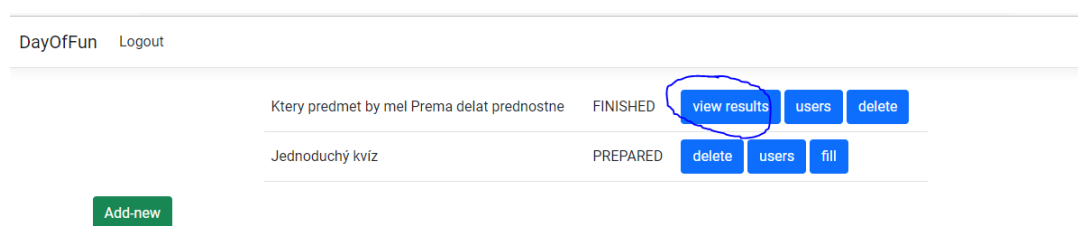


Obrázek 5.2: Seznam kvízů

## 5.2 Funkce kvízů

Tato obrazovka umožňuje uživateli několik funkcí. Pokud je kvíz ve stavu FINISHED, je možné se podívat na výsledku kvízu viz obrázek 5.3 a ??.

### 5.2.1 Zobrazení stavu a výsledků kvízu



Obrázek 5.3: Akce pro zobrazení výsledků kvízu

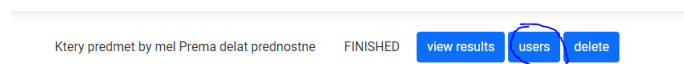
The screenshot shows the results of a quiz. At the top, there is a header with 'DayOfFun' and a 'Logout' link. Below the header, there is a table with the following data:

Question text	Score	Number of respondents
KIV/DP <a href="#">Back to List</a>	2	3

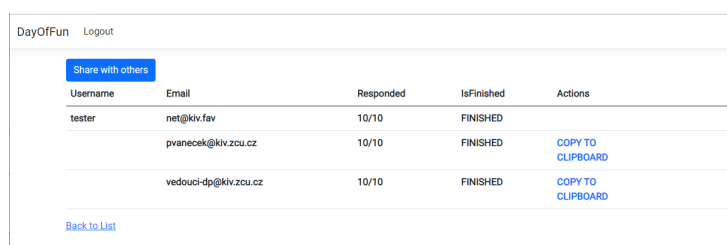
Obrázek 5.4: Výsledek kvízu

### 5.2.2 Práce s uživateli kvízu

Na obrazovce seznamu kvízů je možné jako další akci použít zobrazení uživatelů kvízu (obrázek 5.5). Po kliknutí na akci je uživatel přesměrován na seznam uživatelů, kteří se účastní kvízu (obrázek 5.6).



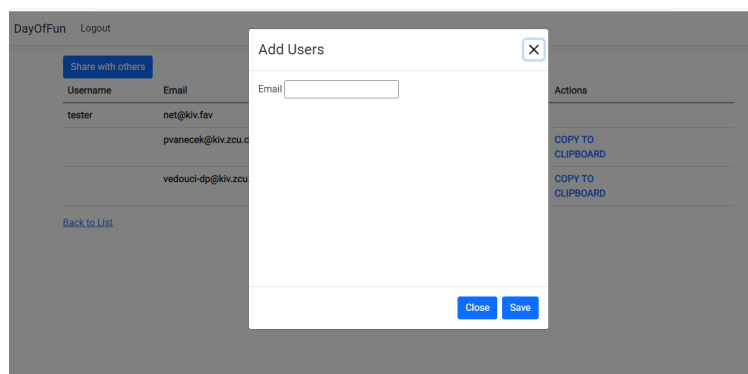
Obrázek 5.5: Akce pro zobrazení uživatelů kvízu



Username	Email	Responded	IsFinished	Actions
tester	net@kiv.fav	10/10	FINISHED	
	pvaneczek@kiv.zcu.cz	10/10	FINISHED	<a href="#">COPY TO CLIPBOARD</a>
	vedouci-dp@kiv.zcu.cz	10/10	FINISHED	<a href="#">COPY TO CLIPBOARD</a>

Obrázek 5.6: Seznam uživatelů kvízu

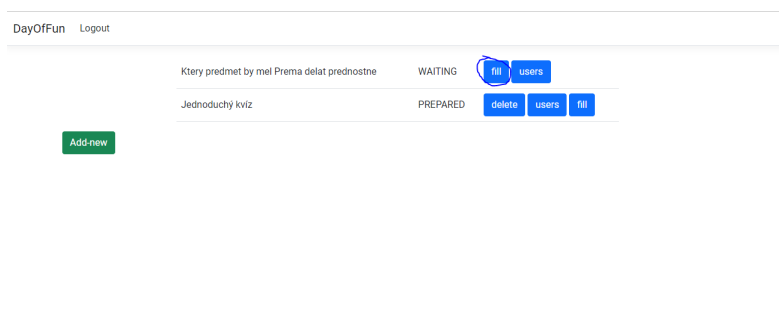
Tato obrazovka umožňuje přidat další uživatele do kvízu pomocí tlačítka „Share with others“ (obrázek 5.15). Tím se otevře dialogové modální okno, ve kterém lze přidat dalšího uživatele (obrázek 5.7).



Obrázek 5.7: Dialog pro přidání uživatele

### 5.2.3 Vyplňování kvízu

Po kliknutí na akci „Fill“ (obrázek 5.8) je uživatel přesměrován na obrazovku vyplňování kvízu (obrázek 5.9). Zde jsou vypsané jednotlivé otázky a je možné vybrat z možností ANO/NE/Když ostatní chtějí.



Obrázek 5.8: Akce pro vyplňování kvízu

DayOfFun Logout

Ktery predmet by mel Prema delat prednostne

KIV/NET

YES

KIV/ISW

NO

KIV/VSS

NO

KIV/LA

NO

KPV/MNT

NO

KIV/ACS1

NO

KIV/FJP

NO

KIV/PIA

NO

KIV/PPR

NO

KIV/DP

IF OTHER WANTS

Add QuestionSubmit Form

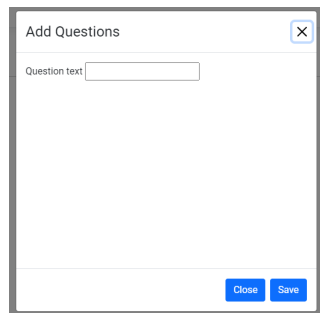
[Back to List](#)

Semestral work for course KIV/NET © Přemysl Kouba - DayOfFun 2022

Obrázek 5.9: Vyplňování kvízu s akcí přidání otázky

Během vyplňování je možné také přidat další otázku, na kterou tvůrce kvízu mohl zapomenout. K tomu slouží tlačítko „Add question“. Po kliknutí je uživateli zobrazeno dialogové okno, ve kterém může doplnit svojí otázku (obrázek 5.10).

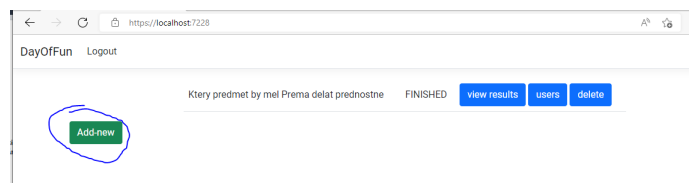




Obrázek 5.10: Dialog přidání otázky

## 5.2.4 Vytvoření kvízu

Pro vytvoření nového kvízu slouží tlačítko „Add-new“ (obrázek 5.11). Po kliknutí je uživatel přesměrován na formulář vytváření kvízu (obrázek 5.12) kde je možné pomocí tlačítka add přidávat další otázky kvízu. Pro vyplňování textu otázky funguje autocomplete funkce, která napovídá texty otázek na základě vloženého textu do pole vstupu.



Obrázek 5.11: Akce vytvoření kvízu

### Create new quiz

Title


  



Obrázek 5.12: Přidání otázky do kvízu

## 5.3 Veřejný přístup

Neregistrovaným uživatelům je umožněn přístup na „routy“ které mají část cesty „public“. Přístup je potřeba uživatelům poslat tak, aby se mohli na tuto adresu dostat. K tomu slouží tlačítko na zkopírování odkazu (obrázek 5.13 na obrazovce uživatele (obrázek 5.6 podkapitola 5.2.2).

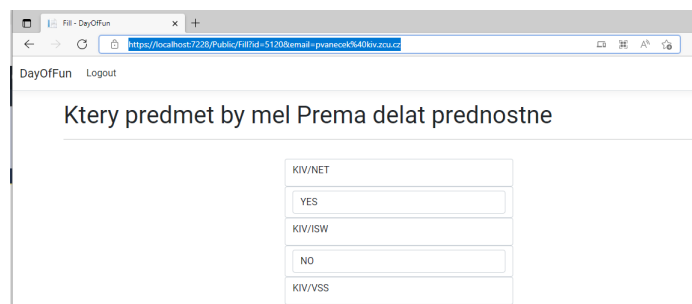


Share with others		Responded	IsFinished	Actions
Username	Email			
tester	net@kiv.fav	10/10	FINISHED	
	pvanec@kiv.zcu.cz	10/10	FINISHED	COPY TO CLIPBOARD
	vedouci-dp@kiv.zcu.cz	10/10	FINISHED	COPY TO CLIPBOARD

[Back to List](#)

Obrázek 5.13: Zkopírování odkazu

Po kliknutí je do schránky zkopírován text odkazu, který pak stačí vložit do navigační řádky prohlížeče (obrázek ??).



DayOfFun Logout

Ktery predmet by mel Prema delat prednostne

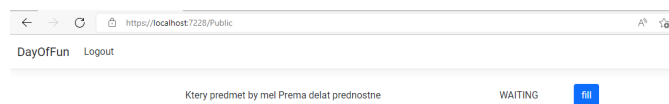
KIV/NET  
☐ YES

KIV/ISW  
☐ NO

KIV/VSS  
☐

Obrázek 5.14: Vyplňování kvízu z veřejného přístupu

Uživatelé ve veřejném přístupu mají taktéž k dispozici zobrazení detailu o kvízech, kterých se účastní (obrázek ??).



DayOfFun Logout

Ktery predmet by mel Prema delat prednostne

WAITING

Obrázek 5.15: Seznam kvízů neregistrovaného uživatele

## 6 Závěr

Odevzdaná práce je proof of concept, který poskytuje jen část funkcionality, která byla původně uvažována. Poskytovaná funkcionalita je testovaná a myslím, že funguje jak má.

Při práci na aplikaci mi celkem rychle došla časová dotace. V současnosti již „přetékám“ o 15 hodin a to velká část funkčnosti ještě nefunguje. Je to dáno tím, že jsem se při vývoji potýkal s mnoha problémy - například jsem špatně použil Entity framework pro vazby M:N.

Práce má velké množství bezpečnostních hrozeb a rozhodně jí nehodlám používat. Asi nejzásadnější je práce s autorizací a autentikací uživatelů - část logiky jsem si implementoval sám (kontrolu přihlášení přes session) a myslím, že to není správná volba.

Čím více jsem se blížil 80 hodinám tím více jsem se odkláněl od architektonického návrhu a v současné verzi je to tak trochu „Ball of mud“, ale snaha byla.

Zcela upřímně by si práce zasloužila více času, ale nemůžu si dovolit tento čas věnovat. Navíc časová dotace (kreditová) pro semestrální práci je 80 hodin podle sylabu.

# Literatura

- [1] Spin the wheel to decide a random choice. Dostupné z:  
<https://pickerwheel.com/>.
- [2] DOODLE. Free online meeting scheduling tool. Dostupné z:  
<https://doodle.com/en/>.
- [3] SEDMIKRAS. Sedmikras/DayOfFun: Semestral work for KIV/Net Course on West Bohemian University. Dostupné z:  
<https://github.com/Sedmikras/DayOfFun>.