

# Aplikácia pre tvorbu testov

Technická zpráva k projektu do předmětu ITU  
FIT VUT v Brně, 2020

Název týmu

Itučkáři

Autoři

Eva Moresová, xmores01

Jakub Richtarik, xricht29

Jiří Žák, xzakji02

# Obsah

## [1. Zadání a organizace týmu](#)

### [1.1 Cíl](#)

### [1.2 Tým](#)

### [1.3 Roadmapa](#)

### [1.4 Rizika a opatření](#)

## [2. Průzkum a zkušenosti](#)

### [2.1 Existující řešení](#)

[Survio \(Jiří Žák\)](#)

[Testedit \(Jiří Žák\)](#)

[Daypo \(Eva Moresová\)](#)

[E-kurz \(Eva Moresová\)](#)

[WebTesty.sk \(Jakub Richtarik\)](#)

[ClassMarker \(Jakub Richtarik\)](#)

### [2.2 Uživatelské potřeby](#)

### [2.3 Shrnutí](#)

## [3. Architektura řešení](#)

### [3.1 Architektura systému](#)

### [3.2 Architektura aplikace/í](#)

### [3.3 Datový model](#)

### [3.4 Vybrané technologie a implementace](#)

## [4. Návrh GUI - webová aplikace \[xzakji02\]](#)

### [4.1 Požadavky na GUI](#)

### [4.2 Makety](#)

### [4.3 Pilotní test](#)

### [4.4 Vyhodnocení testu a revize návrhu](#)

## [5. Návrh GUI - desktopová aplikácia \[xmores01\]](#)

### [5.1 Požadavky na GUI](#)

### [5.2 Makety](#)

### [5.3 Pilotní test](#)

### [5.4 Vyhodnocení testu a revize návrhu](#)

## [6. Návrh GUI - webová aplikácia \(panel\) \[xricht29\]](#)

### [6.1 Požadavky na GUI](#)

### [6.2 Makety](#)

### [6.3 Pilotní test](#)

### [6.4 Vyhodnocení testu a revize návrhu](#)

## [Reference](#)

# 1. Zadání a organizace týmu

## 1.1 Cíl

Cieľom projektu je vytvoriť aplikáciu pre tvorbu testov. Aplikácia bude umožňovať intuitívne vytváranie testových otázok rôzneho typu, ich kategorizovanie a úpravu po uložení, s možnosťou pridania ukážky (obrázok, alebo textová ukážka). Vytvorený test bude možné uložiť a exportovať do PDF pre prípadnú tlač. Taktiež bude možné vygenerovať pre vytvorený test viacero testovacích skupín, vytvoriť a vytlačiť šablónu so správnymi odpoveďami. Užívateľ si bude môcť zobraziť predošlé vytvorené testy a otázky. Aplikáciu bude možné v budúcnosti napojiť na väčší systém pre online testovanie študentov s automatickou opravou testov.

Aplikácia je primárne určená pre učiteľov základných a stredných škôl. Bude slúžiť na urýchlenie a uľahčenie procesu tvorby testov a tiež ich prehľadnú správu.

## 1.2 Tým

Eva Moresová	- dizajn a tvorba desktopovej aplikácie, práca s api
Jakub Richtarik	- návrh a tvorba backendu (MVC), bočný panel a responzivita webovky
Jiří Žák	- dizajn a tvorba webovej aplikácie, práca s api

## 1.3 Roadmapa

1. Rozdelenie zodpovedností v rámci tímu a výber technológií, ktoré budú použité. (Kapitoly: 1.2, 3.4, Termín: 6.10.)
2. Prieskum užívateľských potrieb (na starosti Jakub Richtarik) a spoločný návrh dátového modelu aplikácie (štruktúry zobrazovaných a ukladaných dát). (Kapitoly: 2.2, 3.3, Termín: 10.10.)
3. Každý člen si pripraví návrh vzhľadu (štruktúry) GUI (zahŕňa vytvorenie makety a uskutočnenie pilotných testov), prezentuje návrh ostatným členom tímu, spoločne sa vyberie a prípadne ešte upraví ten najlepší. (Kapitoly: 4.\*, Termín: 14.10.)
4. Návrh architektúry aplikácie, prepojenie jednotlivých častí (backend s webovým a desktopovým GUI) a návrh backendového API. (Kapitoly: 3.1, 3.4, Termín: 16.10.)
5. Implementácia jednotlivých častí, počas ktorej sa členovia tímu synchronizujú minimálne v týždňových intervaloch. (Kapitoly: 5.1, 5.2, Termín: 6.12.)
6. Finálne testovanie vytvorených aplikácií. (Kapitoly: 5.3, 5.4, Termín: 13.12.)

## 1.4 Rizika a opatrení

Medzi najväčšie riziká, ktoré môžu nastať pri realizácii projektu patria nižšie uvedené. Všetky z nich sme sa snažili čo najvia eliminovať.

1. neznalosť vlastného tímu - je lepšie pracovať s ľuďmi, ktorých poznáme, aby sme vedeli, čo od nich môžeme očakávať a či sa na nich môžeme spoľahnúť
2. výpadok člena tímu - riešením je snažiť sa vyhotoviť produkt v predstihu (pri výpadku je čas na dorábku zvyšnými členmi) a rozvrhnúť prácu rovnomerne, aby nestál projekt na 1 osobe
3. nesúhlas v tíme - rozhodovanie hlasovaním (demokracia v praxi), robenie kompromisov, stratégia win-win
4. problémy v komunikácii - z dôvodu COVID-u nemožnosť osobných mítingov, preto pravidelné videohovory s ich nahrávaním, aby sa členovia tímu mohli neskôr vrátiť k tomu, čo zaznelo
5. neznalosť technológií - dá sa vyriešiť len naštudovaním

## 2. Průzkum a zkušenosti

### 2.1 Existující řešení

#### Survio (Jiří Žák)

Rozsáhlejší webová aplikace na tvorbu testů či dotazníků. Aplikace nabízí opravdu hodně věcí, které už nemají s tvorbou testů moc společného. Ovládání je překvapivě jednoduché a intuitivní. Bohužel pokud si uživatel neplatí premium účet, tak je omezen pouze na 100 odpovědí na měsíc.

#### Testedit (Jiří Žák)

Webová stránka nabízející implementaci přímo do vlastních stránek. Bohužel za tuto funkci se platí. Demo aplikace, kterou zde mají k vyzkoušení, pak pouze ukáže jak vypadají hotové testy. Je to opět velice jednoduché a přehledné, takže uživatel by neměl mít problém se v tom vyznat a případně si i vyzkoušet předpřipravený test.

#### Daypo (Eva Moresová)

Služba je dostupná ako webová stránka aj desktopová aplikácia. Poskytuje veľa druhov otázok, nielen otázky s možnosťami alebo s otvorenou otázkou a možnosť rôzneho bodovania otázok. Prostredie je jednoduché a poskytuje iba potrebné funkcie. Avšak kvôli zastaralému vzhľadu je aplikácia neprehľadná. Niektoré tlačidlá neobsahujú text, iba ikonu, z ktorej novému užívateľovi nie je hneď jasná ich funkcia.

#### E-kurz (Eva Moresová)

Aplikáciu je možné využívať cez webovú stránku alebo stiahnuť do mobilu. Vytváranie testov je jednoduché a prehľadné, testy je možné triediť do priečinkov. Dizajn stránky je moderný a

prívetivý. Dajú sa ale vytvárať iba základné typy otázok, čo nemusí byť pre niektorých užívateľov postačujúce. Ďalšou nevýhodou je, že aplikácia je dostupná len po anglicky.

### WebTesty.sk (Jakub Richtarik)

Ide o webovú aplikáciu s dobre prepracovaným dizajnom. Ďalšou výhodou je, že neobsahuje príliš veľa tlačidiel, kvôli ktorým by došlo k informačnému smogu pre používateľa.

Nevýhodou je určite máť oveľa ovládanie. Občas je totiž z veľkého tlačidla klikateľnou časťou len text. Ďalším príkladom je tlačidlo dole pod otázkou, pri kt. by človek očakával, že prejde k vytvoreniu ďalšej otázky, miesto toho slúži len na presuny medzi už vytvorenými otázkami. Navyše je užívateľ schopný vytvárať ďalšie otázky v teste, bez toho aby predchádzajúce uložil (príznak nikde nesignalizuje), a tak môže dôjsť k tomu, že keď je celý test hotový, používateľ neskôr zistí, že má vlastne 20 prázdnych otázok a k vyplneným dátam sa už nedostane.

### ClassMarker (Jakub Richtarik)

Webstránka kladie dôraz na funkčnosť pred dizajnom. Obrovskou výhodou je, že dokáže vytvoriť v podstate akýkoľvek test. Nevýhodou je ale angličtina pre český a slovenský trh (nie každý učiteľ ZŠ a SŠ vie po anglicky). A ešte väčším negatívom je zahltenosť obsahom - jednoduchí učitelia, kt. nie sú až taký zdatní s počítačmi, by sa strácali vo veľkom počte podstránok a nastavení. Chýba preto nejaká možnosť voľby medzi jednoduchým a rozšírenejším rozhraním.

## 2.2 Užívateľské potreby

Primárnymi užívateľmi budú učitelia základných a stredných škôl, no neskôr by sa systém mohol rozširovať do viacerých sfér (testy pre autoškoly, psychologické testy, atď...).

Z rozhovorov vyplynulo, že učitelia by boli vďační najmä za prehľadnú správu testov. Keďže niektorí si toľko neveria s aplikáciami (online testovanie), ocenia ich možnosť tlače, s odpovedovým hárkom pre kontrolu. Aplikácia by nemala byť veľmi náročná s veľkým množstvom tlačidiel v ktorých sa stratia.

Asi najväčším prínosom by pre nich bola práve možnosť vkladania už použitých úloh do nových testov, kt. vo väčšine existujúcich riešení chýba.

#### **Súhrn vlastností systému (čo by mal systém vedieť)**

- vytvorenie kategórií, podkategórií, ich premenovanie a mazanie
- vyššie spomenuté pri testoch
- vytváranie otázok s písaním odpovedí, ale aj s výberom možností
- možnosť pridania ukážky textovej/obrázkovej, no v budúcnosti kvôli angličtinárom aj zvukovej, prípadne nejaký nástroj pre matematikárov na vkladanie vzorcov
- vkladanie už existujúcich úloh, prípadne informácia, v kt. testoch sa už nachádzajú
- generovanie skupín, odpovedového hárku
- test by sa mal ukladať priebežne počas jeho tvorby, aby sa nestalo, že pri 20. úlohe stratíme internetové spojenie

## 2.3 Shrnutí

Existujúce riešenia väčšinou poskytujú buď všetky funkcie potrebné pri tvorbe testov a sú zložité na ovládanie alebo sú dostatočne jednoduché aby ich vedeli ovládať aj učitelia, ktorí sú "počítačovými laikmi" avšak nemajú potrebnú funkcionálnu. Tento nedostatok by sme sa pokúsili v našej aplikácii odstrániť. Dáme dôraz na vytvorenie aplikácie, ktorá bude jednoduchá a funkčná. Tak ako pri väčšine riešení, dizajn našej aplikácie by mal byť moderný a užívateľsky prívetivý. Existujúce aplikácie sú vo veľkej miere v angličtine, keďže však tvoríme aplikáciu pre slovenský/český trh, zvolíme jeden z týchto jazykov ako jazyk aplikácie.

## 3. Architektura řešení

### 3.1 Architektura systému

Systém bude mať nasledujúce časti:

1. databáza - (lokálne phpMyAdmin, na serveri PostgreSQL) uložené dáta, komunikácia cez kontroléry v Laravel<sup>1</sup> backende (api)
2. Laravel kontroléry - ukladanie dát do databázy, ich načítavanie a vytváranie rôznych pohľadov, komunikácia s aplikáciami v JSON formáte
3. prostredie pre tvorbu testov - ako webová, tak aj desktopová aplikácia budú nahrávať a upravovať informácie v databáze prostredníctvom api

### 3.2 Architektura aplikace/í

#### Desktopová aplikácia

Vzhľadom na zvolené technológie (WPF) sa aplikácia riadi návrhovým vzorom Model, View, ViewModel (MVVM), ktorý je pre WPF typický.

1. Model - Popisuje dáta, s ktorými aplikácia pracuje. O stave ovládacích prvkov nesmie nič vedieť.
2. View - Reprezentuje užívateľské rozhranie v jazyku XAML. Môže ísť o okno, stránku, alebo ovládací prvok.
3. ViewModel - Podobná funkcia ako Controller v MVC. Spája View s Modelom a drží stav aplikácie. S View je prepojený pomocou bindingu dát. Svoje dáta poskytuje v dátových štruktúrach, ktoré vyvolávajú pri ich zmene udalosti. Vďaka tomu sa nové dáta môžu zobraziť v užívateľskom rozhraní hneď, ako sa vo ViewModeli zmenia. Na základe stavu aplikácie filtruje dáta.

#### Webová aplikace

Při implementaci webové aplikace byl použitý návrhový vzor Model, View, Controller.

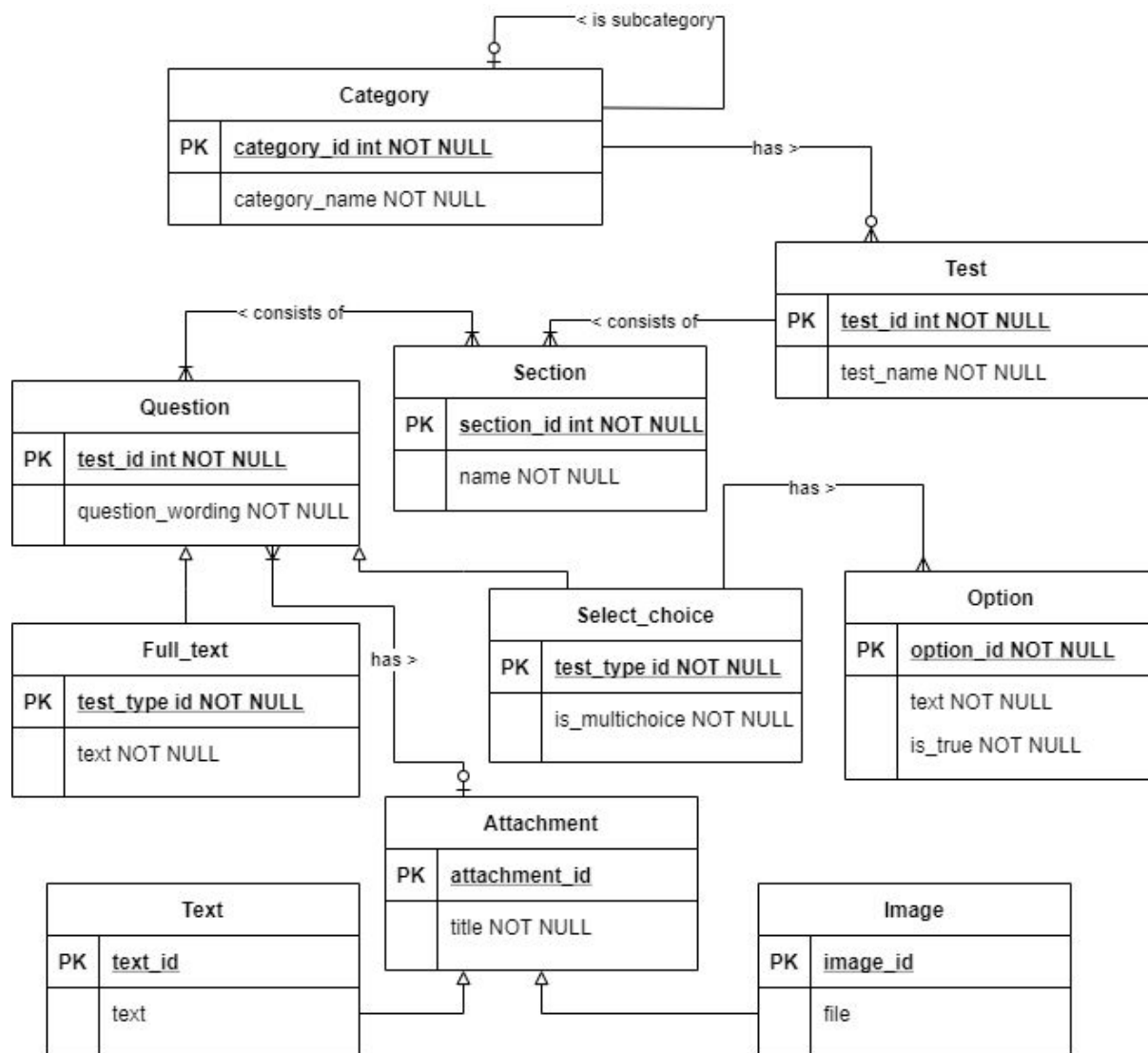
1. Model - popisuje data se kterými bude aplikace pracovat. O model se stará aplikační rozhraní.
2. View - zobrazuje aktuální stav aplikace a jeho data. Bude implementováno pomocí HTML, JSX, JSON a SASS.

---

<sup>1</sup> <https://laravel.com/docs/8.x/>

3. Controller - práci s datovým modelem a co zobrazí View zajistí Controller, který bude implementován pomocí moderního Javascriptu. Aplikace se bude měnit podle toho co udělá uživatel a pošle tento stav na View, který nové změny ihned zobrazí.

### 3.3 Datový model



Testy budú uložené v hierarchii kategórií a podkategórií, s nejakým názvom. Test pozostáva z 1, alebo viacerých sekcií (teória, dopĺňanie... alebo podľa celkov učiva). Pri každej budú uložené otázky (ich znenie) s plnou odpoveďou, alebo s jednotlivými možnosťami (1 správna, alebo viacero). Ku každej otázke bude možné pridať textovú prílohu (ukážku), alebo obrázok.

### 3.4 Vybrané technológie a implementace

Backend je realizovaný pomocou php frameworku Laravel, ktorý jednoducho umožňuje vytvárať modely, kontrolery, taktiež migráciu tabuliek (ich vytvorenie) do databázy. Ďalšou z

funkcionalít je naplnenie tabuliek vzorovými dátami cez seeder. Predpokladá sa implementácia autentizácie pomocou mena a hesla a autorizácia pomocou cookies. Aplikácie komunikujú s modelom pomocou api, využívajúc čistú url ("Clean URLs", "RESTful URLs"), čiže url, kt. je user-friendly.

Method	URI
POST	attachments
GET HEAD	attachments
DELETE	attachments/{attachment}
PUT PATCH	attachments/{attachment}
GET HEAD	attachments/{attachment}
POST	categories
GET HEAD	categories
DELETE	categories/{category}
GET HEAD	categories/{category}
PUT PATCH	categories/{category}
POST	options
GET HEAD	options
DELETE	options/{option}
PUT PATCH	options/{option}
GET HEAD	options/{option}
GET HEAD	questions
DELETE	questions/{question}
GET HEAD	questions/{question}
GET HEAD	tests
POST	tests
DELETE	tests/{t}
PUT PATCH	tests/{t}
GET HEAD	tests/{t}
GET HEAD	tests/{t}/sections
POST	tests/{t}/sections
PUT	tests/{t}/sections/{s}
DELETE	tests/{t}/sections/{s}
POST	tests/{t}/sections/{s}/questions
GET HEAD	tests/{t}/sections/{s}/questions
PUT	tests/{t}/sections/{s}/questions/{question}
DELETE	tests/{t}/sections/{s}/questions/{question}
GET HEAD	tests/{t}/sections/{s}/questions/{question}/attach

Desktopová aplikácia bude realizovaná pomocou Windows Presentation Foundation. Vývojová platforma WPF podporuje veľkú škálu funkcií pre vývoj aplikácií (aplikačný model, ovládacie prvky, dátové väzby, grafika a iné). Využíva jazyky XAML a C#.

Webová aplikácia bude vytvorená predovšetkým moderným Javascriptom, ktorý sa bude starať o hladký chod aplikácie. K snadnejšej práci bude využívať JSX a JSON. Moderným Javascriptom ďalej zaisťujeme prístup k api a práci s ním. Na stylovanie aplikácie sme zvolili SASS preprocessor.



## 4. Návrh GUI - webová aplikace [xzakji02]

### 4.1 Požadavky na GUI

Webová aplikace zobrazí menu s kategoriemi, podkategoriemi a testy. Uživatel bude moci vytvořit kategorii, podkategorii či test. Poté co klikne na vybraný test se mu daný test zobrazí a bude možné ho libovolně upravovat. Po uložení se vše přepośle do databáze a uživatel si ho může kdykoliv znovu zobrazit. Dizajn aplikace by měl být intuitivní a jednoduchý pro rychlé pochopení funkčnosti a snadné ovladatelnosti. Základem by mělo být použití jen pár barev. Aplikace by měla být single page.

### 4.2 Makety

The wireframe illustrates a single-page web application layout. At the top, a light blue header bar contains a 'Logo' on the left and three 'Menu' items on the right. The main content area has a white background. At the top of this area is the label 'Název testu'. Below it is a large rectangular container with a thin black border. Inside this container, there is a light blue rectangular box labeled 'Znění otázky'. Below that is a larger light blue rectangular box labeled 'Odpověď'. At the bottom of the container, there are two light blue buttons: 'Uložit' on the left and 'Přidat otázku' on the right.

### 4.3 Pilotní test

Tento návrh testovala studentka pedagogické fakulty ze které bude učitelka prvního stupně. Dostala za úkol vytvořit krátký test a jednu podkategorii. Bude sledováno jak rychle se uživatelka zorientuje v aplikaci a jakou dobu ji zabere zadané věci splnit. Dále vyhodnotíme, jestli je spokojená s návrhem a barevným dizajnem aplikace.

## 4.4 Vyhodnocení testu a revize návrhu

Uživatelka se překvapivě rychle zorientovala a vytvořit pár podkategorií ji zabralo jen několik minutek. Hlavní a stěžejní úkol, vytvořit krátký test, zabral více času, ale i tak to byl splněn rychleji než se původně očekávalo, takže to hodnotím kladně. Zato s dizajnem moc spokojená nebyla. Nelíbilo se jí rozvržení stránky a barevný návrh aplikace. Na tyto věci ohledně rozvržení a barevné škály bych se chtěl nejvíce zaměřit a co nejvíce je vyladit. Dále by chtěla mít možnost si test vyexportovat do pdf, což mě zaujalo a chtěl bych tuto funkci implementovat. Nejvíce byla uživatelka překvapená z nápadu single page aplikace, což ji velmi zaujalo a zhodnotila tento nápad kladně za což jsem velmi rád.

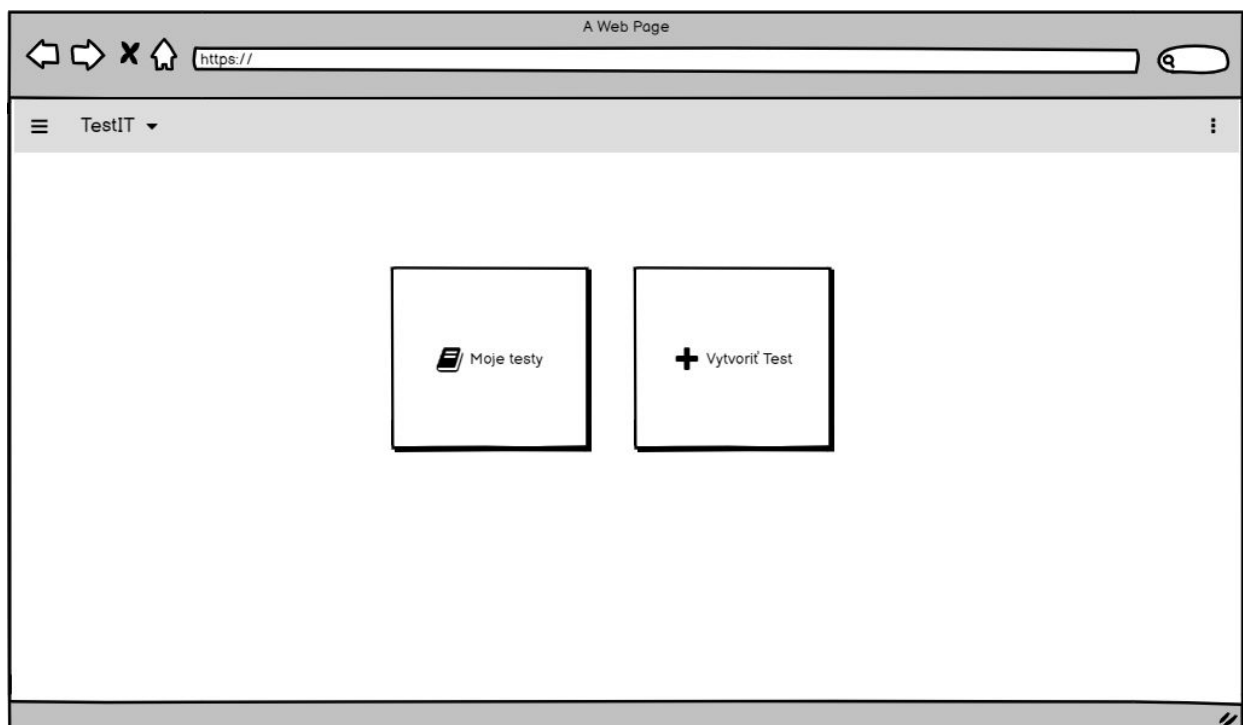
## 5. Návrh GUI - desktopová aplikácia [xmores01]

### 5.1 Požadavky na GUI

Uživateľ musí byť schopný jednoducho, rýchlo a bez školenia vytvoriť test. Dôležité je zobrazenie kategórií, prípadne podkategórií, do ktorých sú testy roztriedené - malo by byť prehľadné, jednotlivé kategórie jasne oddelené. Dizajn GUI by mal byť jednoduchý, prívetivý a moderný, farby uspôsobené tak, aby aj pri dlhšom používaní neunavovali oči užívateľa. Keďže ide o desktopovú aplikáciu, mala by mať jednoduchú inštaláciu a responzívny obsah.

### 5.2 Makety

Úvodná obrazovka:



Prvá obrazovka pri vytváraní testu:

A Web Page

https://

TestIT

< Späť

Názov testu

Kategória

Vybrať kategóriu

Matematika

Fyzika

Angličtina

Iné

Vytvoriť

Pridanie otázky:

A Web Page

https://

TestIT

< Späť

Názov testu

Pridať otázku

Zrušiť

Uložiť

A Web Page

https://

TestIT

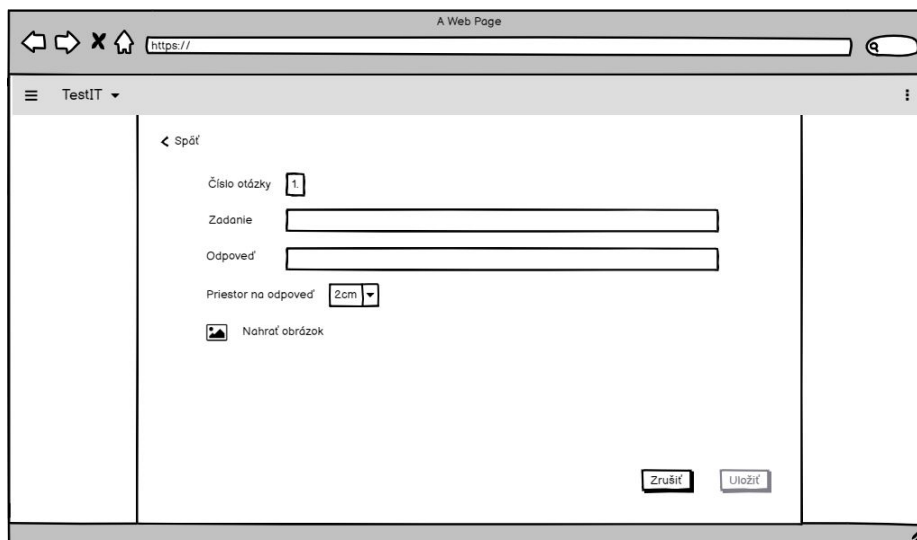
< Späť

Pridať otázku

Otázka s otvorenou odpoveďou

Otázka s možnosťami

Vybrať otázku z histórie



## 5.3 Pilotní test

Pilotný test bol realizovaný na stredoškolskej učiteľke angličtiny. Vytvorené makety mali definované akcie po kliknutí na určité prvky, pomocou čoho sa simulovala interakcia s aplikáciou. Úlohou bolo vytvoriť jednoduchý test s tromi otázkami, pričom užívateľ videl aplikáciu prvýkrát až počas testu.

Sledovaná bola schopnosť užívateľa zorientovať sa v aplikácii bez predošlého vysvetlenia, prípadne čokoľvek, čo bolo preňho máťúce. Ďalej čas a počet krokov (kliknutí), ktorý bol potrebný na základné úkony, ako napríklad vytvorenie otázky. Tiež sa pozorovalo či užívateľ postupuje očakávaným spôsobom alebo od aplikácie očakáva iný workflow.

## 5.4 Vyhodnocení testu a revize návrhu

Test odhalil, že hlavnou vadou v návrhu je moc veľká komplikovanosť. Vytvorenie testu bolo síce priamočiare, ale časovo náročné - užívateľ sa musel preklikať veľkým počtom obrazoviek, kým sa mu podarilo dosiahnuť to, čo chcel. Taktiež oddelenie časti aplikácie určenej pre vytváranie testov a časti na prehľad uložených testov nie je praktický. Návrh bol teda zmenený tak, aby kategórie a vytvorené testy boli viditeľné hneď po otvorení aplikácie a vytváranie testu a jeho otázok bolo na jednej stránke.

# 6. Návrh GUI - webová aplikácia (panel) [xricht29]

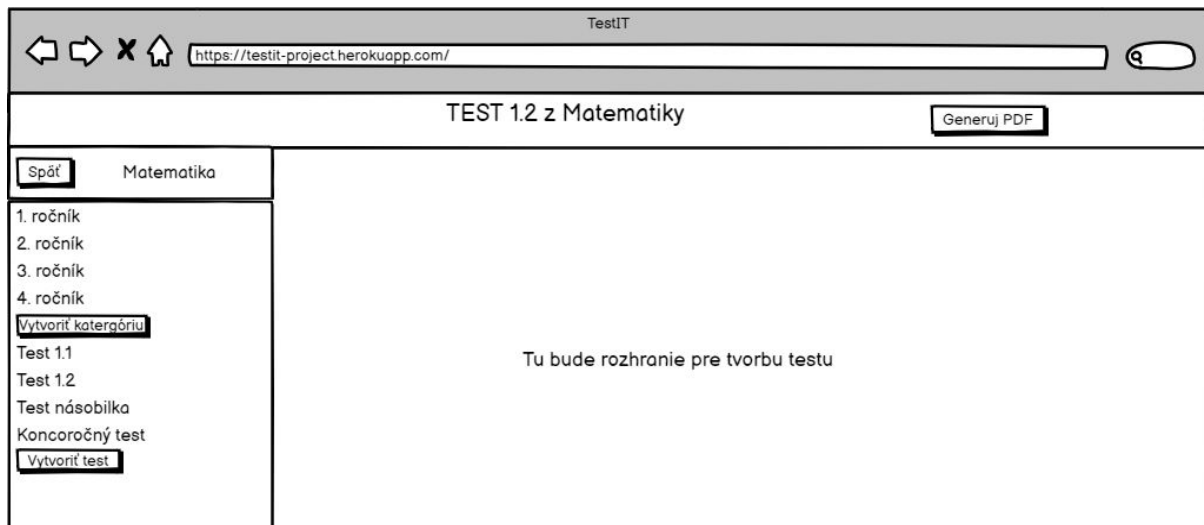
## 6.1 Požiadavky na GUI

V aplikácii je potrebné vytvoriť akéhosi správcu testov. Realizovať by sa mohol pomocou fixného bočného panelu, kde by sa dalo jednoducho prechádzať jednotlivými kategóriami a podkategóriami, alebo aj pomocou tlačidla, ktoré by po kliknutí vykresliło pop-up okno.

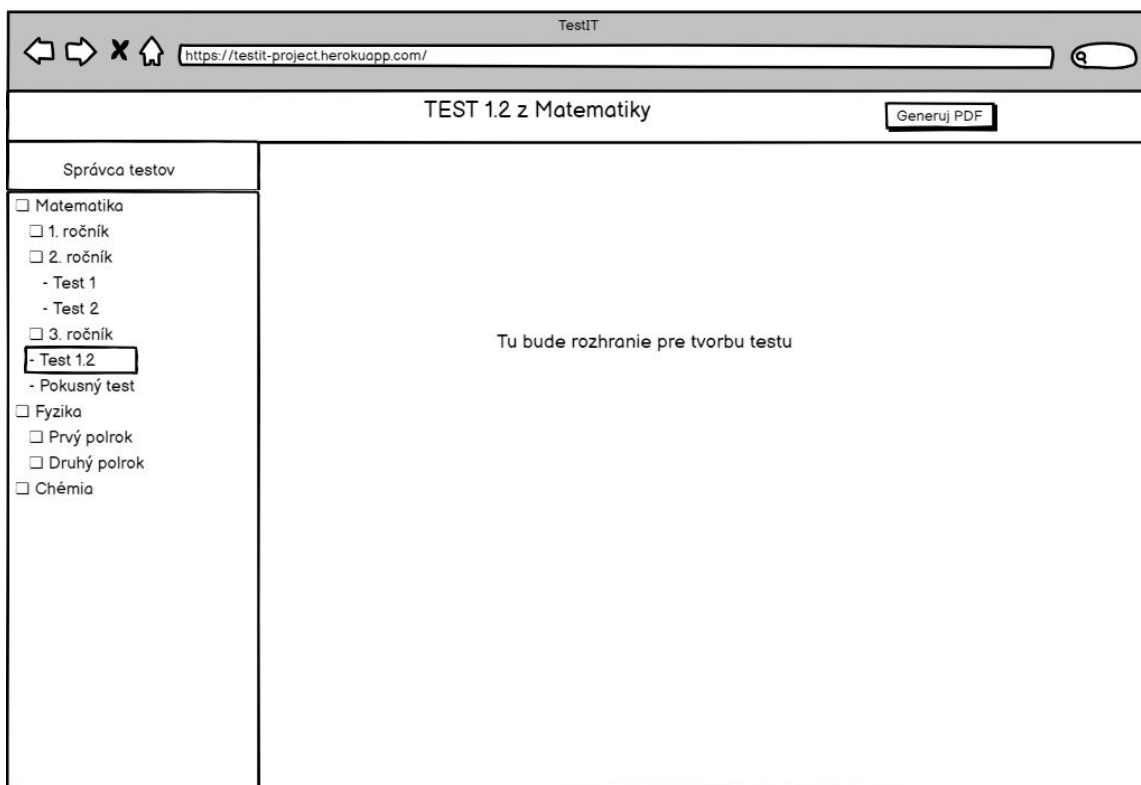
## 6.2 Makety

Prvotnou ideou bolo vytvoriť bočný panel, v ktorom by bol zobrazený len obsah aktuálne rozkliknutej kategórie/podkategórie. Mohli by sa vytvárať nové kategórie, alebo nové testy. Po kliknutí na daný test, by sa v hlavnej časti obrazovky zobrazil test, ktorý by sme mohli vytvárať a upravovať. Na základe toho bolo vypracovaná Varianta 1 (2. až po testoch).

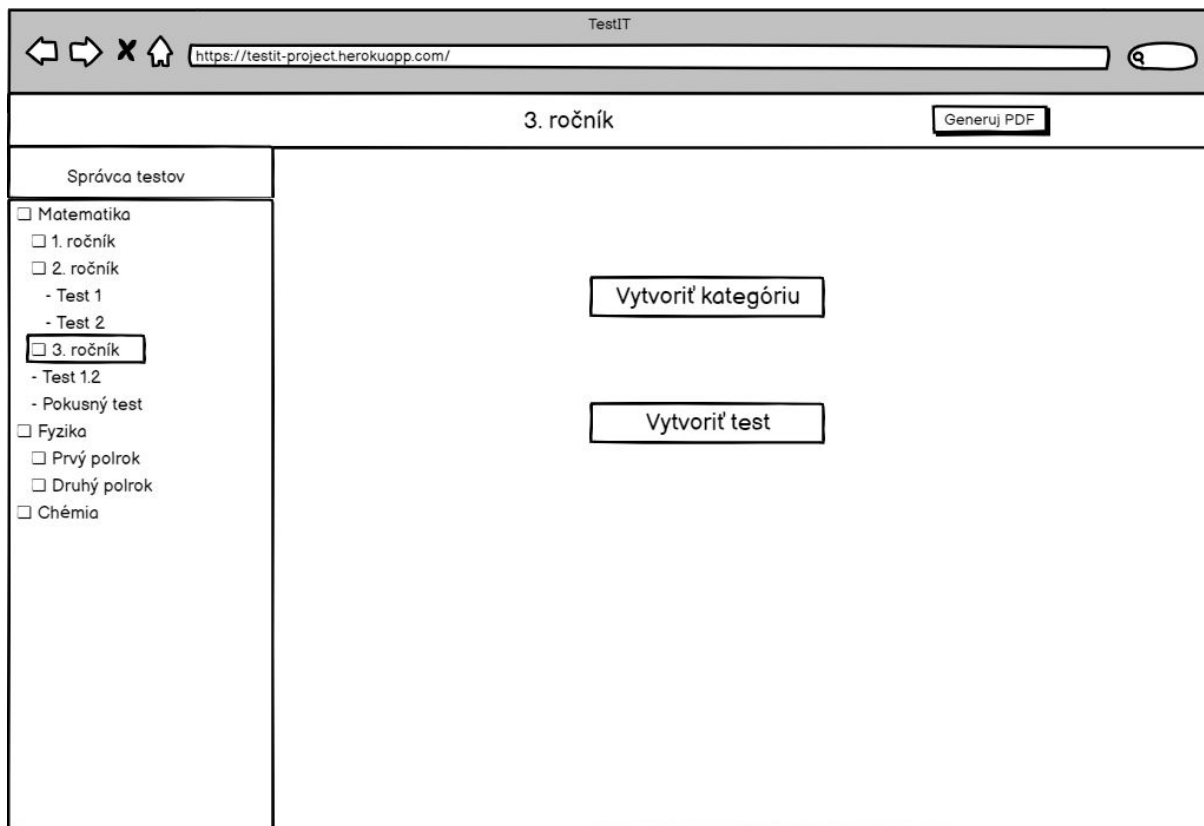
Varianta 1 (zobrazená len aktuálna kategória):



Varianta 2 (v paneli je zobrazená celá štruktúra kategórií a testov):



zobrazenie testu



vytváranie kategórie/testu po rozkliknutí kategórie

## 6.3 Pilotní test

Návrh a testovanie bolo realizované programom Balsamiq<sup>2</sup> na Variante 1 (Varianta 2 ešte neexistovala). Zúčastnila sa ho učiteľka slovenčiny cirkevného gymnázia.

Úlohou bolo vytvárať nové kategórie, podkategórie, testy (bez ich realizácie, čiže len pridávanie do panelu). Cieľom bolo zistiť, či je prostredie user-friendly, či je zjavné, ktoré tlačidlo ako funguje bez nápovedy, a vo všeobecnosti, či sa v prostredí užívateľ ľahko zorientuje.

## 6.4 Vyhodnocení testu a revize návrhu

Z testu vyplynulo, že užívateľ sa v prvej variante strácal, ak bol zanorený napríklad v 4. úrovni kategórii. Problém totiž nastal, ak mal pre predmety Slovenčina a Literatúra kategórie a podkategórie s rovnakými názvami a nebol si istý, kde sa vlastne nachádza. Ak by sme chceli použiť túto variantu, užívateľ navrhol, aby bola niekde v hornej časti vypísaná celá aktuálna cesta (ako napr. v programe Total Commander).

Následne bol návrh prerobený na Varianta 2, kedy je v paneli zobrazená celá štruktúra kategórií a testov. S výsledkom bol užívateľ spokojnejší. Išlo o princíp File Explorer na Windows, ktorý aj väčšina učiteľov základných a stredných škôl používa a sú naň zvyknutý. Komplikáciou by bola možno situácia, kedy by zanáranie do kategórií, resp. dlhé názvy, prinútili rozšíriť bočný panel, alebo prirobiť horizontálny scrollbar.

---

<sup>2</sup> <https://balsamiq.cloud/>

# Reference

<https://docs.microsoft.com/cs-cz/visualstudio/designers/getting-started-with-wpf?view=vs-2019>

<https://www.dotnetportal.cz/clanek/4994/MVVM-Model-View-ViewModel>

<https://balsamiq.cloud/>

<https://laravel.com/docs/8.x/>

<https://laravel-livewire.com/docs/2.x>