

# Platforma na správu dotazníků

---

*Tým Problem Solvers*

## *Architektura*

<b>Verze</b>	<b>Datum</b>	<b>Úprava</b>	<b>Autor</b>
1.0	24.4.2017	Vytvoření dokumentu	Miroslav Liška
1.1	2.5.2017	Úprava dokumentu po konzultaci s mentorem. Úprava modelu nasazení. Popis produkčního prostředí. Zdůvodnění použitých technologií. Odebrání příliš podrobné implementační části.	Miroslav Liška

## Produkční prostředí

Na úvod - výsledkem ASWI projektu je webová aplikace pro dotazování uživatelů ZČU.

V rámci projektu je nutné zajistit přihlašování uživatelů. V produkčním prostředí jsou uživatelé zvyklí se přihlašovat Orion loginem a heslem. V rámci produkčního prostředí se využívá ověřovacího systému WebAuth. Napojení na tento systém ověřování je zajištěno administrátorem produkčního prostředí.

Vzhledem k tomu, že se v rámci výsledného produktu budou po síti posílat citlivá data, je nutné, aby byl přenos dat šifrovaný. K tomu bude využito protokolu HTTPS. Jeho integrace bude zajištěna administrátorem produkčního prostředí.

Administrátor produkčního prostředí vyžaduje vývoj webové aplikace s využitím programovacího jazyka PHP z důvodu nízké režie. Dále požaduje použití databázového systému MySQL též z důvodu nízké režie.

Administrátor produkčního prostředí z důvodu bezpečnosti silně doporučuje pro komunikaci mezi jednotlivými moduly využití webových služeb. Komunikace s webovými službami by tak byla jediná možnost pro přístup k databázovému systému.

## Možné způsoby řešení

### Google Apps

Zákazníkem nám bylo doporučeno prozkoumat možnosti Google aplikací, konkrétně aplikaci pro tvorbu dotazníků (Google Forms).

Univerzita má zakoupenou prémiovou licenci pro Google Apps (licenci plánuje univerzita dlouhodobě podporovat) a přihlašovací systém univerzity je napojen na přihlašovací systém Google Apps.

V rámci analýzy byla aplikace prozkoumána. Aplikace v základu umožňuje tvorbu otázek s odpověďmi typu text a jedno-výběrové (radiobutton) a více-výběrové odpovědi (checkbox). Aplikace umožňuje výběr správné odpovědi a následné vyhodnocení dotazníků s případným exportem dat.

Google Forms tedy poskytují v rámci požadavků zákazníka potřeby, kromě modulů odpovědi pro psaní a vyhodnocování kódu. Google Forms umožňují uživatelům si vlastní modul naprogramovat. V rámci analýzy bylo ale zjištěno, že dostupné implementační prostředky nejsou dostačující. Konkrétně – do formuláře na straně respondenta je potřeba přidat tlačítko pro vyhodnocení správnosti kódu napsaného respondentem a na toto tlačítko navěsit události, které tento kód zvalidují, případně zjistí výsledek kódu. To ale Google přímo nepodporuje. Z tohoto důvodu jsme se rozhodil Google Apps nepoužít. Dalším důvodem nepoužití Google Apps je nejasnost, jakým Google ukládá odpovědi respondenta.

## Kandidátní architektura

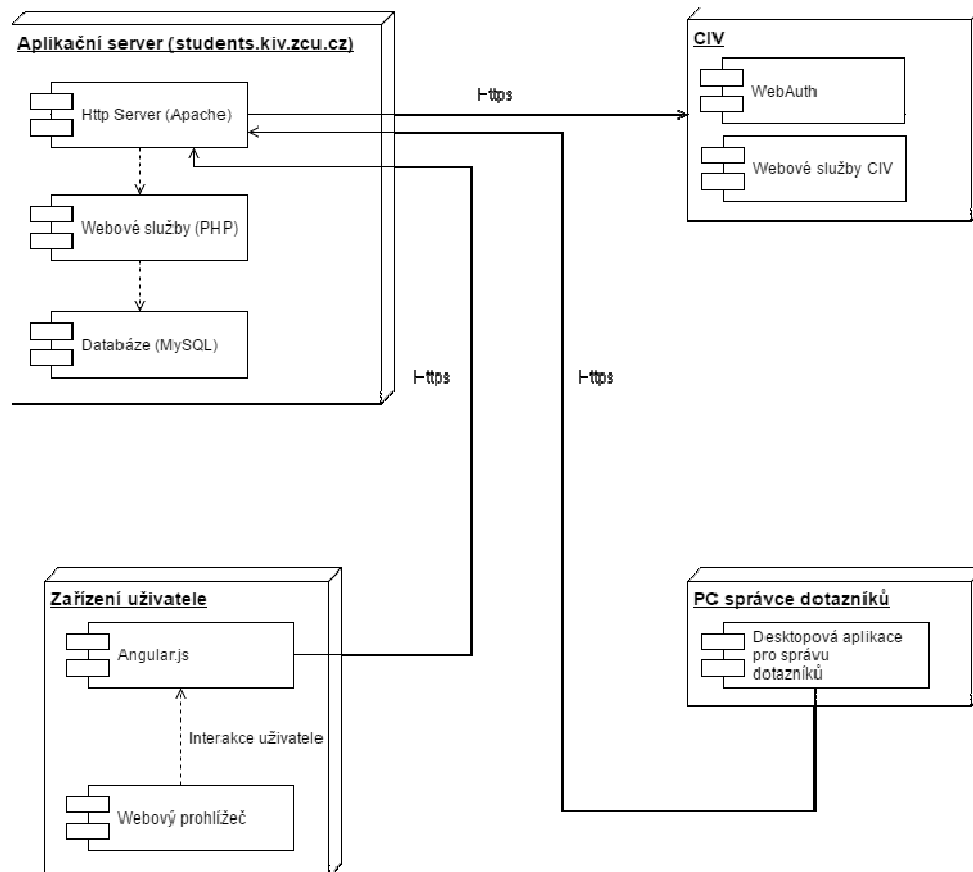
Jak bylo zmíněno v sekci [Produkční prostředí], administrátor produkčního prostředí nám silně doporučil pro komunikaci mezi jednotlivými částmi využít aplikačních služeb. Tyto služby by zajišťovaly přístup k databázovému systému.

Projekt je rozdělen na následující části:

- Desktopová aplikace – slouží pro tvorbu dotazníků. Přistupuje k ní pouze správce dotazníků

- Webová aplikace – slouží pro běžné uživatele, zobrazuje dotazníky a umožňuje je uživateli vyplnit
- Aplikační služby – zajišťuje business logiku projektu

Komunikace mezi jednotlivými částmi probíhá s využitím webového rozhraní REST. Business logika bude schovaná za tímto rozhraním. Jednotlivé části budou volat metody, které jsou REST rozhraním definované. Volání těchto metod bude zprostředkované HTTPS protokolem. Každý požadavek bude prověřen ověřovacím systémem WebAuth, který zajistí autentizaci uživatelů. Obsah přenášených zpráv mezi aplikačními službami a uživatelskými aplikacemi bude ve formátu JSON. Komunikace je zobrazena na obrázku [Obrázek 1].



Obrázek 1 Model nasazení

## Architektura aplikačních služeb

Architektura aplikačních služeb je navržena podle návrhového vzoru MVC (Model-View-Controller). Ten zajišťuje oddělení datové, prezentační a logické vrstvy aplikace. Návrhový vzor bude navíc doplněn o tzv. servisní vrstvu. Tato vrstva bude zajišťovat business logiku aplikačních služeb.

Jak je možné vidět na obrázku [Obrázek 2], desktopová nebo webová část pošle požadavek konkrétní api. Ta předá požadavek servisní vrstvě. Servisní vrstva zajistí business logiku, například komunikaci s databází.

Takováto architektura nám logicky oddělí kód aplikačních služeb. Servisní služby nám navíc snadno dovolí změnit případný zdroj dat. Například controller nebude komunikovat s využitím servisní vrstvy přímo s databází, ale například s tzv. Mock daty. Tento návrh nám umožní velkou flexibilitu a robustnost kódu a zároveň bude kód snazší otestovat.

Jak bylo zmíněno v sekci [Produkční prostředí], je požadováno, aby byly aplikační služby psané v jazyce PHP.

## **Rozhraní aplikačních služeb**

Aplikační služby jsou rozděleny na tři rozhraní:

- ManageQuestionnaire.php – rozhraní pro správu dotazníků
- UserQuestionnaire.php – rozhraní pro komunikaci s uživatelem webové aplikace
- ExportQuestionnaire.php – rozhraní pro získání dat k exportu

Požadavek od klienta je podle URL zpracován odpovídajícím REST rozhraním. V rozhraní je ověřena validita požadavku a požadavek je předán do servisní vrstvy. Servisní vrstva podle potřeby upraví datový model a výsledek předá zpět REST rozhraní, které vrátí odpověď klientovi.

### **Datové struktury pro komunikaci s webovými službami**

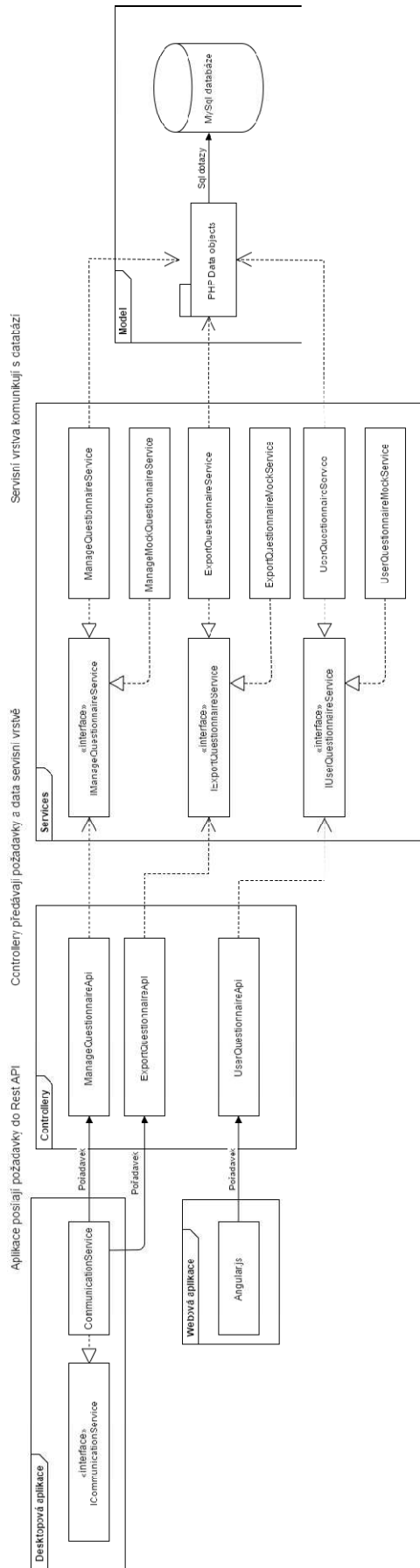
Datové struktury pro komunikaci s webovými službami jsou formátu JSON. Jejich podrobný popis je součástí programátorské dokumentace.

## **Datový model**

Datový model projektu bude reprezentován relační databází. V projektu je použit databázový systém MySQL.

Pro přístup k datovému modelu bude využito rozhraní PHP Data Objects (PDO), které slouží ke snadnějšímu a bezpečnějšímu přístupu k datům.

Era model databáze je popsán v sekci ERA model databáze.



Obrázek 2 Navržená architektura aplikačních služeb

## Architektura desktopové aplikace

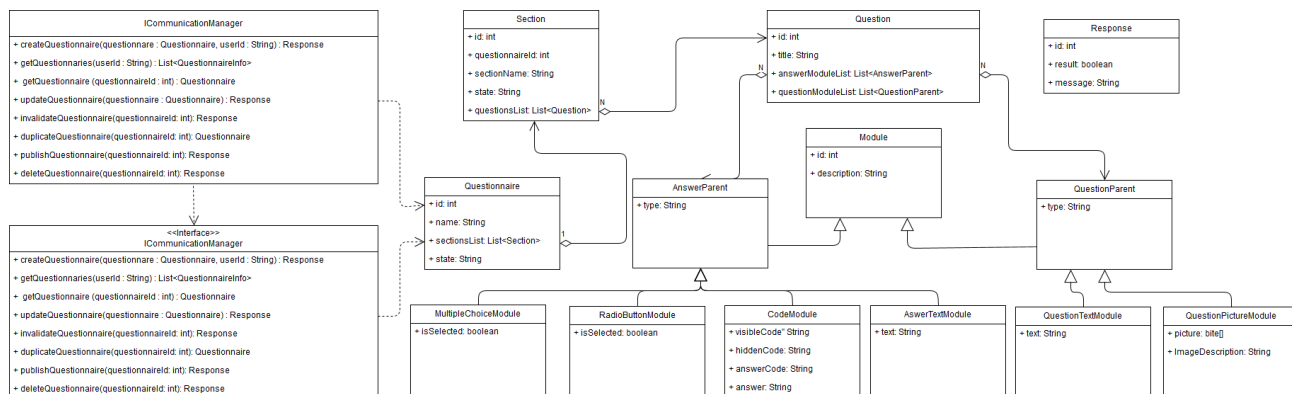
Tuto část projektu jsme se rozhodli implementovat v jazyce Java a používat návrhový vzor Model-View-Controller. Tato architektura je použita hlavně z důvodu značného zpřehlednění vývoje a možnou modularitu jednotlivých částí, které jsou na sobě víceméně nezávislé.

Při tvorbě grafického rozhraní bylo hojně využíváno rozhraní, aby bylo možno jednotlivé části jednoduše přidávat či nahrazovat.

Celá aplikace pracuje díky spolupráci jednotlivých controllerů. Všechny controllery implementuje jednotné rozhraní IController, díky kterému veškeré důležité instance udržujeme pomocí dynamického pole v hlavní třídě aplikace a máme k nim tedy přístup a v případě potřeby je využíváme.

V aplikaci se vyskytuje několik základních datových struktur, které jsou z důvodu kompatibility téměř totožné s těmi, které se vyskytují u architektury na webové části projektu, a tak není důvod tyto architektury znovu uvést.

V aplikaci se dále nachází rozhraní ICommunicationManager, jehož implementující třída má za úkol veškerou komunikaci s webovým klientem. Data, která toto rozhraní musí poskytovat, byla představena výše v architektuře webové aplikace. Niž na obrázku [obrázek 3] se můžete podívat na schéma této architektury.



Obrázek 3: Náhled architektury desktopové aplikace

## Návrh uživatelského rozhraní

### Webová aplikace

Uživatelské rozhraní webové aplikace bude využívat framework Bootstrap, který usnadňuje tvorbu responzivního webu s čistým a elegantním vzhledem. Cílem návrhu je, aby bylo uživatelské rozhraní maximálně intuitivní a respondent byl schopen dotazník vyplnit i bez předchozího zaškolení.

Na obrázku [4] je ukázka prototypu uživatelského rozhraní. Stránka se skládá z následujících částí:

1. **Hlavní menu v hlavičce stránky** – Po kliknutí na odkaz „Dotazníky“, se respondentovi zobrazí stránka se seznamem jemu přiřazených dotazníků. Dále toto menu obsahuje odkazy pro získání informací o dotaznících a kontaktu na pověřenou osobu. Toto menu slouží k přístupu k informacím o stránce a návodu jak vyplňovat dotazníky
2. **Seznam sekcí dotazníku** – Levé menu obsahuje seznam sekcí aktuálně otevřeného dotazníku, společně s indikací, zda uživatel danou sekci již vyplnil, rozpracoval, nebo dosud neotevřel.
3. **Hlavní obsah stránky**– Obsah stránky ukazuje seznam otázek právě otevřené sekce. Pod otázkami jsou umístěna tlačítka pro uložení rozpracované sekce pro pozdější editaci a pro odeslání odpovědí aktuální sekce. Před odesláním odpovědí bude uživatel varován, že

Dotazníky   Informace   Kontakt

Sekce 1 ✓

**Sekce 2 (rozpracována)**

Sekce 3 (rozpracována)

Sekce 4 ?

Dotazník XXX > Sekce 2

**Otázka 1 ✓**

Jak se jmenujete?

**Otázka 2 ?**

Doplňte kód následující metody tak, aby vracela druhou mocninu vstupu.

Programovací jazyk:  Java    C#    Pascal

```
public static int squareNumber(int number) {
```

```
}
```

**Otázka 3 ?**

S jakými technologiemi/jazyky jste se setkali při výuce na střední škole?

- Java
- C#
- C/C++
- HTML
- JavaScript
- SQL, Microsoft Access
- Pascal
- Python
- Linux
- Jiné

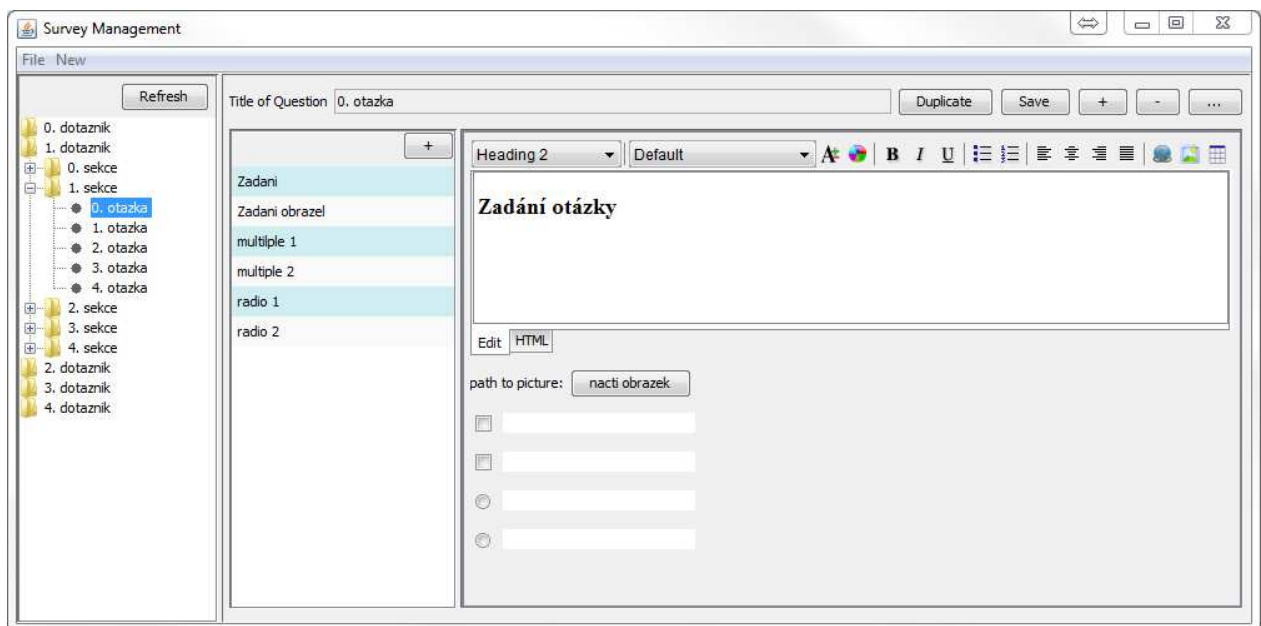
odpovědi jsou finální a nebude možné je dodatečně změnit.

## Desktopová aplikace

Aplikace se skládá ze 2 hlavních částí, které se velmi jednoduše ovládají.

1. První část je tvořena stromem dotazníků, včetně jejich podsekcí a otázek. Ihned po startu aplikace dojde ke stažení základních informací o všech dotaznících, tj. jejich název a identifikátor. Všechny informace jsou tedy obsaženy právě v tomto stromě. Pokud uživatel klikne poprvé na ještě nezobrazený dotazník v rámci spuštění aplikace, dojde nejprve k jeho stažení pomocí webové služby. Strom implementuje celkem 3 typy uzlů (Dotazník, sekce a otázka) a v závislosti na pohybu po tomto stromě dochází k překreslování druhé části této aplikace. Součástí stromů jsou také vyskakovací okna, která umožňují manipulaci s jednotlivými uzly.
2. Tato část je vykreslována v závislosti na zvoleném uzlu ve stromové struktuře popisované v první části
  - 2.1. Pokud uživatel klikne na některý dotazník ve stromové struktuře, pak dojde k přenesení tohoto objektu do pravého okna a dojde k zobrazení jeho základních informací. Uživatel zde má možnost pomocí tlačítek přidávat a odebírat dotazníky případně zobrazit si detailní informace o tomto dotazníku
  - 2.2. Pokud uživatel klikne na uzel obsahující sekci, pak mu bude nabídnuté téměř totožné možnosti jako v předchozím případě. Dojde opět k natažení aktuální instance ve stromě do okna, kde se mu vše potřebné zobrazí.
  - 2.3. Tato část patří mezi nejdůležitější a dochází zde k samotné tvorbě dotazníků. Uživateli se nejen zobrazí stejné možnosti jako v předchozích dvou případech, ale zobrazí se mu také seznam modulů, které otázka obsahuje. Opět dochází k natažení zvoleného uzlu ve stromě, konkrétně instance otázky.

Návrh této desktopové aplikace můžete vidět v následující části u návrhu uživatelského rozhraní desktopové aplikace. Návrh desktopové uživatelského rozhraní je k vidění na obrázku [5].



Obrázek 5 Návrh hlavního okna desktopové aplikace

## ERA model databáze

Návrh databázového modelu projektu je zobrazen na obrázku [6]. V modelu nejsou pro přehlednost uvedené datové typy. Model je rozdělen na dvě logické části – část pro zpracování odpovědí uživatele a část pro ukládání dat dotazníků. Použité názvy tabulek v databázi budou v následujícím popisu uváděny v závorce. V pravé části je vidět následující posloupnost tabulek: Dotazník (Questionnaire) je



tvořen sekcemi (Section), sekce jsou tvořeny otázkami (Question). Otázky jsou tvořeny moduly otázek (QuestionModule) a odpovědí (AnswerModule). Modul otázky může být typu (QuestionType):

- Text (QuestionTextModule)
- Obrázek (QuestionPictureModule)

Modul odpovědi může být typu (AnswerType):

- MultipleChoice (MultipleChoiceModule)
- Code (CodeModule)
- RadioButton (RadioButtonModule)
- Text (TextModule)

Dotazník během vytváření (Questionnaire) nabývá stavů (QuestionnaireState):

- Založen
- Publikován
- Smazán

V levé části je posloupnost tabulek následující: Uživatel (User) má přiřazený (UserQuestionnaire) dotazníky (Questionnaire). Přiřazené dotazníky uživatelů nabývají stavů (UserQuestionnaireStatus). Konkrétní stavy jsou:

- Neotevřen
- Rozpracován
- Odeslán

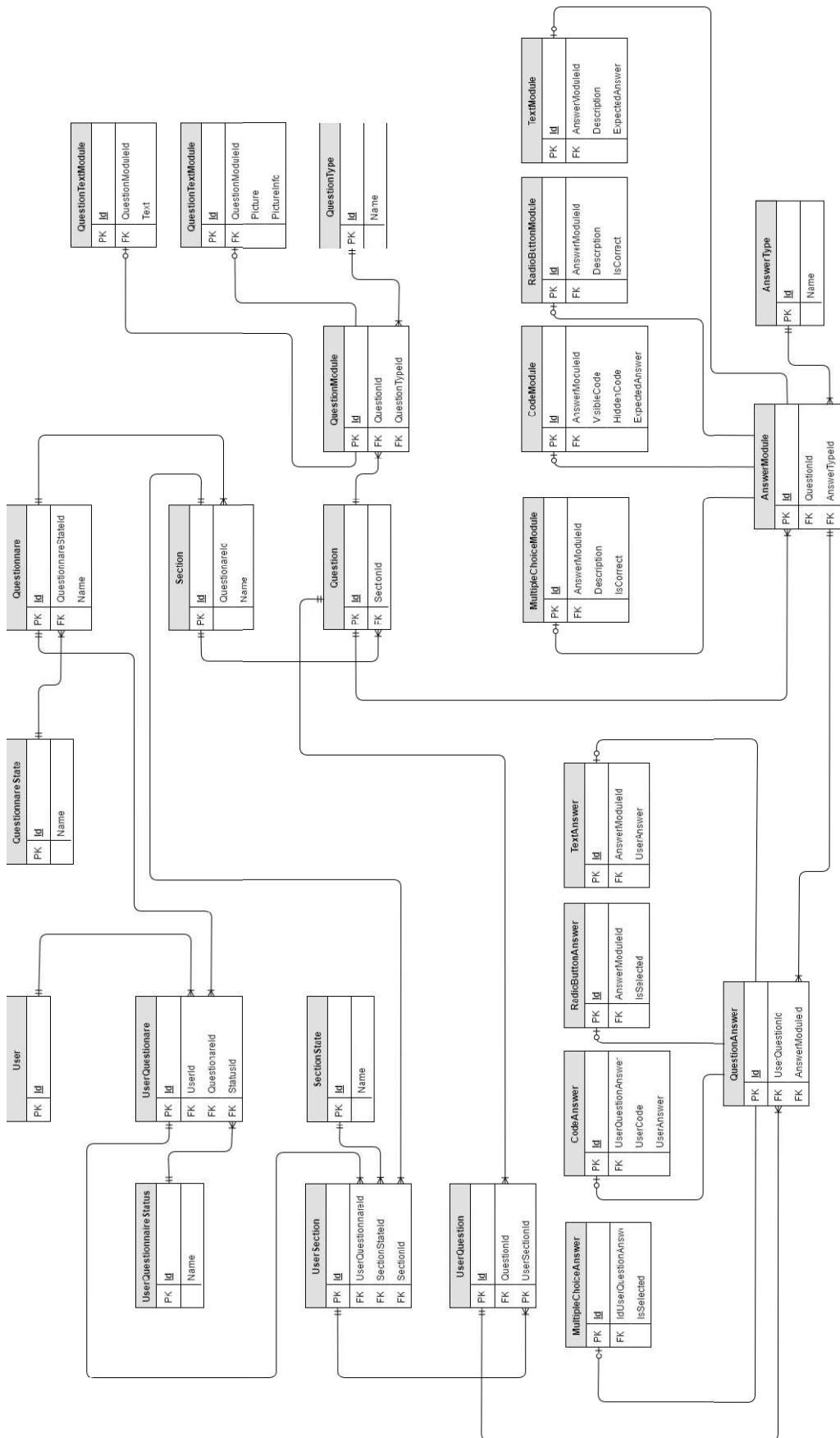
Přiřazené dotazníky uživateli se skládají ze sekcí (UserSection). Přiřazené sekce nabývají stavů (SectionState):

- Neotevřen
- Rozpracován
- Odeslán

Přiřazené sekce jsou tvořeny otázkami (UserQuestion). Odpovědi uživatele na jednotlivé otázky jsou uloženy v tabulkách, která jsou v relaci 0,1:1 s tabulkou QuestionAnswer. Konkrétně to jsou:

- MultipleChoiceAnswer
- CodeAnswer
- RadioButtonAnswer
- TextAnswer

Tabulka QuestionAnswer je provázána s tabulkou AnswerModule. Touto vazbou je možné zjistit, zda odpověď uživatele byla správná.



Obrázek 4 ERA model projektu