

Katedra informatiky
Přírodovědecká fakulta
Univerzita Palackého v Olomouci

BAKALÁŘSKÁ PRÁCE

Webová aplikace pro psychodiagnostiku



2023

Vedoucí práce:
Mgr. Jiří Zaccpal, Ph.D.

Jan Šándor

Studijní obor: Aplikovaná informatika,
prezenční forma

Bibliografické údaje

Autor: Jan Šándor
Název práce: Webová aplikace pro psychodiagnostiku
Typ práce: bakalářská práce
Pracoviště: Katedra informatiky, Přírodovědecká fakulta, Univerzita Palackého v Olomouci
Rok obhajoby: 2023
Studijní obor: Aplikovaná informatika, prezenční forma
Vedoucí práce: Mgr. Jiří Zaccal, Ph.D.
Počet stran: 46
Přílohy: 1 CD/DVD
Jazyk práce: český

Bibliographic info

Author: Jan Šándor
Title: Web application for psychodiagnostics
Thesis type: bachelor thesis
Department: Department of Computer Science, Faculty of Science, Palacký University Olomouc
Year of defense: 2023
Study field: Applied Computer Science, full-time form
Supervisor: Mgr. Jiří Zaccal, Ph.D.
Page count: 46
Supplements: 1 CD/DVD
Thesis language: Czech

Anotace

Práce popisuje návrh a implementaci webové aplikace pro sportovní psychodiagnostiku. Účelem aplikace je poskytnout vědeckým pracovníkům z FTK UPOL, odborníkům z Asociace psychologů sportu a sportovním psychologům on-line nástroj ke sběru dat od sportovců pomocí digitalizace dotazníkových metod ACSI-28 pro rozhodčí, ACSI-28 pro sportovce, PCDEQ, VMIQ-2, OMSAT-3 a jejich automatickou evaluaci. Aplikace je implementována s využitím jazyka Python a frameworku Django.*

Synopsis

The thesis describes the design and development of a web application for sports psychodiagnostics. The application aims to offer researchers from FTK UPOL, specialists from the Association of Sport Psychologists, and sports psychologists an on-line tool for collecting data from athletes through the digitization of questionnaires ACSI-28 for referees, ACSI-28 for athletes, PCDEQ, VMIQ-2 and OMSAT-3 along with their automatic evaluation. The application is implemented using the Python language and the Django framework.*

Klíčová slova: webová aplikace; Django; Python; Bootstrap

Keywords: web application; Django; Python; Bootstrap

Rád bych poděkoval panu Mgr. Jiřímu Zaccpalovi, Ph.D. za čas věnovaný mým dotazům a průběžným kontrolám jak praktické, tak teoretické části práce. Dále panu Mgr. Michalovi Vičarovi, Ph.D. za průběžné konzultace a dodání potřebných podkladů pro vývoj aplikace.

Místopřísežně prohlašuji, že jsem celou práci včetně příloh vypracoval/a samostatně a za použití pouze zdrojů citovaných v textu práce a uvedených v seznamu literatury.

datum odevzdání práce

podpis autora

Obsah

1	Cíle práce	8
2	Průzkum existujících řešení	9
2.1	Mindeval	9
2.2	Shrnutí	11
3	Specifikace řešení	12
3.1	Základní zadání a předpoklady	12
3.2	Návrh technologií	13
3.3	Rozbor zadání	13
3.3.1	Registrace	14
3.3.2	Omezení přístupu k datům podle rolí	15
3.3.3	Vyplnění dotazníku	15
3.3.4	Systém notifikací	15
3.3.5	Exportování dat	15
4	Návrh řešení	16
4.1	Diagram případů užití	16
4.2	Specifikace případů užití	16
4.3	Klient	17
4.3.1	Diagram případů užití	17
4.3.2	Specifikace případů užití	18
4.4	Psycholog	19
4.4.1	Diagram případů užití	19
4.4.2	Specifikace případů užití	19
4.5	Výzkumník	20
4.5.1	Diagram případů užití	21
4.5.2	Specifikace případů užití	21
4.6	Diagram tříd	23
5	Použité technologie	25
5.1	Jazyk Python	25
5.2	HTML/CSS	27
5.3	Bootstrap	27
5.4	Django	28
5.4.1	Balíčky	29
5.5	Databáze PostgreSQL a SQLite	31
5.6	Vue.js	31
5.7	Docker	32

6	Uživatelská příručka	33
6.1	Základní funkcionalita	33
6.1.1	Registrace	33
6.1.2	Přihlášení/Odhlášení	33
6.1.3	Zapomenuté heslo	33
6.2	Klienti	34
6.2.1	Žádosti o vyplnění dotazníku	34
6.2.2	Vyplnění a odeslání dotazníku	34
6.3	Psychologové	34
6.3.1	Přehled klientů	34
6.3.2	Pozvat klienta	35
6.3.3	Detail klienta	35
6.3.4	Detail responze	35
6.3.5	Dotazníky a manuály	35
6.4	Výzkumníci	35
6.4.1	Přehled výsledků	35
6.4.2	Detail responze	36
6.4.3	Dotazníky a manuály	36
6.5	Výzkumník-administrátor	36
6.5.1	Schvalování psychologů	36
6.5.2	Přehled výzkumníků	36
6.5.3	Deaktivovat/Reaktivovat účet výzkumníka	37
6.5.4	Vytvoření účtu výzkumníka	37
6.5.5	Dotazníky a manuály	37
6.5.6	Přidávání příloh k dotazníku	37
6.5.7	Povolení/zakázání sběru responzí	37
6.5.8	Odstranění dotazníku	38
7	Programátorská příručka	39
7.1	Struktura projektu	39
7.2	Konfigurační soubory aplikace	40
7.3	Modelové třídy	40
7.3.1	Modul <i>accounts</i>	40
7.3.2	Modul <i>sportdiag</i>	40
	Závěr	42
	Conclusions	43
A	Testování aplikace	44
A.1	Přístupové údaje	44
B	Obsah elektronických dat	44
	Literatura	45

Seznam obrázků

1	Desktopová aplikace Mindeval po spuštění na platformě Windows 10	10
2	Diagram případů užití – Uživatel a Klient	17
3	Diagram případů užití – Psycholog	19
4	Diagram případů užití – Výzkumník a Výzkumník-administrátor . . .	21
5	Diagram tříd modulu <i>accounts</i>	24
6	Diagram tříd modulu <i>sportdiag</i>	25
7	Struktura kořenového adresáře projektu	39

Seznam tabulek

1	Přístupové údaje pro testování	44
---	--	----

1 Cíle práce

Cílem práce je vytvořit webovou aplikaci pro Fakultu tělesné kultury Univerzity Palackého v Olomouci (FTK UPOL), kde se v rámci grantového projektu věnují adaptaci sady metod sportovní psychodiagnostiky z anglického originálu do českého sociokulturního prostředí. Sadu psychodiagnostických metod tvoří 6 dotazníků a je zaměřena na komplexní analýzu psychické složky výkonu sportovce.

Webová aplikace má být elektronickou alternativou ke standardní papírové verzi dotazníků a umožnit klientům zodpovídat psychodiagnostické dotazníky on-line, čímž se má výrazně rozšířit možnost spolupráce se sportovci. Dále má automatizovat vyhodnocení dotazníků, usnadnit administraci testování sportovců, urychlit interpretaci výsledků a především má být otevřena pro členy organizací sdružených v [Unii psychologických asociací ČR](#).

Data, která aplikace z responzí sportovců nasbírání mají podpořit vědecké pracovníky FTK UPOL při verifikaci, validaci a standardizaci adaptovaných metod sportovní psychodiagnostiky tak, aby byly výsledky základního výzkumu využitelné v klinické praxi, čímž se má pomoci rozšíření kvalitních psychodiagnostických metod ve sportu a omezit využívání nevalidních metod na území České republiky.

2 Průzkum existujících řešení

Jedinou aplikací, která se zabývá sportovní psychodiagnostikou a kterou jsem v rámci svého průzkumu objevil je aplikace Mindeval. V následující podkapitole si ji popíšeme.

2.1 Mindeval

Na domovské stránce www.mindeval.com autoři popisují Mindeval jako nejkomplexnější webovou aplikaci pro on-line testování sportovců pro účely sportovní psychologie. Testování probíhá pomocí dotazníků a má sportovcům z celého světa umožnit prostřednictvím validních dotazníkových metod sportovní psychologie vyvinutých odborníky na zahraničních univerzitách, získat okamžitou zpětnou vazbu, která přispívá ke zlepšení jejich výkonu.[1]

Stěžejním prvkem Mindeval je elektronický nebo též osobní klíč, který se přiřadí každému sportovci nebo obecně účastníkovi testování a zajišťuje anonymitu a bezpečnost – data nejsou propojena s osobními údaji respondenta a v případě nedokončení procesu vyplňování dotazníku jsou všechny dosud zaznamenané odpovědi zapomenuty.[2]

Pro profesionály (trenéry, psychology, konzultanty, ...) Mindeval nabízí webové rozhraní, kde mohou vytvářet testovací scénáře, tzv. „Evaluations“. Do nich lze přidávat jednotlivé dotazníky a elektronické klíče sportovců, kteří mají být testování podrobeni. Lze určovat pořadí, ve kterém se respondentům jednotlivé dotazníky v daném scénáři zobrazí a tak průběh testování navrhovat pro každý scénář. Mindeval disponuje 5 dotazníkovými metodami. Aby bylo možné dotazník do testovacího scénáře přidat, je nutné jej nejprve zakoupit. Čtyři z pěti dotazníků lze zakoupit za částku 25\$ každý a poslední zbývající stojí 45\$. K dispozici jsou množstevní slevy až 50% při koupi 100 a více dotazníků. V případě zakoupení 25–99 dotazníků sleva činí 25% a pro 10–24 dotazníků pak 10%.[3] Přes webové rozhraní mají profesionálové přístup k výsledkům všech testování, které administrují.

Elektronický klíč tvoří 12 místné číslo a ke sportovcům se dostane od jejich trenéra nebo sportovního psychologa (obecně administrátora testu) prostřednictvím elektronické pošty nebo vytištěný na papíře. K vyplňování dotazníků sportovcům slouží desktopová aplikace.

Aby tedy mohli sportovci dotazníky vyplňovat, musejí podniknout několik kroků:

- stáhnout a nainstalovat desktopovou aplikaci Mindeval, která je dostupná zdarma pro operační systémy:
 - Windows XP, Vista, 7, 8, 8.1 a 10
 - Mac OS X 10.6+
 - Ubuntu (32-bit i 64-bit)
- vložit do aplikace po spuštění osobní klíč. Klíč je přímo spárovaný s konkrétním testovacím scénářem, který může obsahovat jeden nebo více dotazníků. Po

vložení klíče do aplikace jsou sportovci ihned prezentovány otázky jedna po druhé.

- po poslední otázce jsou sportovci výsledky testování zaslány v PDF dokumentu, pokud tuto možnost povolil administrátor testu.



Obrázek 1: Desktopová aplikace Mindeval po spuštění na platformě Windows 10

Administrátorem testování je kdokoli, kdo provede na stránkách www.mindeval.com registraci. Ačkoliv Mindeval webové rozhraní deklaruje jako rozhraní pro profesionály, odbornost registrovaného není nijak ověřována. Po jednoduché registraci lze nakupovat dotazníky a libovolně generovat elektronické klíče. Generování klíčů není nijak omezeno a při vytváření klíče je nutné přiřadit jej existujícímu testovacímu scénáři. Pokud klíč respondent například ztratí, administrátor mu snadno vygeneruje nový a pouze jej přiřadí k testovacímu scénáři, kterému chce respondenta podrobit.

Elektronický klíč je přenositelný mezi testovacími scénáři. Administrátor testování tedy může existující klíč přeřadit k jinému vytvořenému testovacímu scénáři.

2.2 Shrnutí

Řešení Mindeval v podstatě umožňuje sportovcům, kteří nejsou svěřenci odborníka na sportovní psychologii:

- zaregistrovat se,
- zakoupit si některý z dotazníků,
- vygenerovat si elektronický klíč a sami se podrobit testování pomocí desktopové aplikace,
- nechat si zaslat výsledky ve formátu PDF.

Výsledky odpovídají osobnímu mentálnímu profilu sportovce pro danou dotazníkovou metodu a dle Mindeval jsou ve výsledcích k dispozici vysvětlivky. Obecně však ve sportovní psychologii není vždy žádoucí, aby byl respondent seznámen s výsledky testování bez účasti odborníka např. v rámci společné schůzky, protože by mohlo dojít k nesprávné interpretaci výsledků sportovcem. Každá z validních dotazníkových metod sportovní psychologie je standardně spjata s uživatelským manuálem určeným pro konzultanty a sportovní psychology, a popisuje jakým způsobem výsledky klientovi interpretovat a jak s ním vést konzultaci výsledků. Z tohoto důvodu bych výše zmíněné označil jako nevýhodu řešení Mindeval a lze se domnívat, že je za tímto finanční motivace tvůrců projektu. Dále považuji za nešikovné, že v dnešní moderní době jsou respondenti nuceni stahovat a instalovat desktopovou aplikaci, aby mohli vyplňovat dotazníky, která například na mém operačním systému Ubuntu 20.04 nefungovala, přesněji nešla ani nainstalovat. Na počítači s operačním systémem Windows 10 jsem se však s žádnými potížemi při instalaci a spuštění nesešel. Mé řešení formou webové aplikace s přístupem nejen pro odborníky, ale i pro respondenty by tedy tato úskalí řešení Mindeval mohlo překonat.

Jako inspirativní pro mou práci shledávám řešení bezpečnosti a anonymity respondentů prostřednictvím elektronického klíče.

3 Specifikace řešení

Řešení webové aplikace ze strany FTK UPOL zaštiťuje pan Mgr. Michal Vičar, Ph.D., který je zadavatelem a protože jedním z požadavků grantového projektu na aplikaci je její provoz a údržba po dobu 5 let po skončení projektu, předpokládá se, že ji do správy po mém vyhotovení převezme společnost OLC Systems s.r.o., která ji bude udržovat v provozu. Od pana Davida Pokorného z OLC Systems s.r.o. jsem také v rámci konzultace dostal doporučení na technologie, ve kterých mohu aplikaci realizovat.

3.1 Základní zadání a předpoklady

Aplikace by měla mít následující funkcionalitu:

- zajistit specifický přístup pro 3 uživatelské role (Klient, Psycholog, Výzkumník) a dle role omezit přístup pouze k odpovídajícím datům,
- umožňovat jednotlivým klientům vyplňovat dotazníky,
- obsahovat systém upozornění na vyplněné dotazníky, případně na žádosti o zřízení přístupu,
- umožňovat export anonymizovaných dat.

Z konzultací s panem Mgr. Michalem Vičarem, Ph.D. dále vyplynuly tyto požadavky:

- předpokládá se 500–1000 responzí na dotazník,
- klient při registraci povinně zadá jméno, příjmení, datum narození, e-mail a pohlaví,
- klient nemá přístup k výsledkům jeho testování,
- klient by měl mít přiřazený nějaký (řekněme anonymizující) identifikátor, kterým by bylo možné rozlišit, že různé responze pocházejí od téhož klienta, avšak bez prozrazení jeho totožnosti,
- klient musí být spárován se svým psychologem a jen ten může přistupovat k výsledkům jeho testování,
- psycholog při registraci zadá jméno, příjmení, e-mail a nahraje soubor potvrzující jeho odbornost v oblasti psychologie (například certifikát nebo diplom dosaženého vzdělání),
- výzkumným pracovníkům by se přístup zřizoval ručně v rozhraní aplikace (předpokládají se jednotky až nízké desítky účtů – v zásadě pracovní skupina pana Vičara),
- data získaná z testování klientů by byla výzkumným pracovníkům prezentována formou tabulky,

- data získaná z testování klientů je třeba exportovat do formátu, který lze otevřít v programu MS Excel,
- vzhledem k nízkému počtu dotazníkových metod (2–6) není vyžadováno rozhraní pro vytváření a úpravu dotazníků. Předpokládá se, že by nové dotazníky do aplikace vložil vývojář nebo správce aplikace.

Další požadavek, který si kladu za cíl sám a považuji jej za nezbytnou součást dnešní moderní webové aplikace je responzivita uživatelského rozhraní alespoň pro klienty, aby mohli vyplňovat dotazníky i z mobilních zařízení.

3.2 Návrh technologií

Vzhledem k předpokládané budoucí správě aplikace firmou OLC Systems s.r.o. jsem se ohledně výběru technologií držel doporučení, která mi byla poskytnuta zástupcem společnosti. Doporučené technologie jsou následující:

- databázová část aplikace bude realizována pomocí technologie PostgreSQL,
- logika aplikace může být realizována v jazyce PHP nebo Python, kde se v případě volby jazyka Python počítá s využitím frameworku Django,
- pro prezenční vrstvu aplikace (frontend) společnost OLC Systems využívá technologii Vue.js, avšak využití této technologie není nezbytně nutné.

Aplikaci jsem se rozhodl vytvořit pomocí jazyka Python a frameworku Django a to především kvůli z mého pohledu podstatně přívětivější a přehlednější uživatelské dokumentaci, než jak ji vnímám u jazyka PHP. Dalším důvodem pro volbu tohoto jazyka bylo, že jazyk Python se těší širokému využití nejen na poli vývoje webových aplikací, ale můžeme jej nalézt také v datové vědě a ve vývoji v oblasti umělé inteligence. Má tedy velký potenciál a také se o něm můžeme často dočíst, že se dobře učí. Jazyk Python a na něm založené technologie mají širokou komunitu uživatelů a proto je na internetu k nalezení spousta knihoven s otevřeným zdrojovým kódem (open-source) včetně jejich dokumentace. Na internetu lze také dohledat mnoho článků a návodů, případně kurzů zabývajících se řešením konkrétních programátorských úloh při vývoji v Pythonu a frameworku Django. To dohromady utváří bohatou základnu informací a použitelných hotových řešení, což může urychlit i usnadnit vývoj aplikace.

3.3 Rozbor zadání

V této kapitole nastíníme mou prvotní představu o technickém řešení aplikace vzhledem k zadání.

3.3.1 Registrace

Registrování klientů a psychologů lze realizovat tradičním registračním formulářem. Formuláře by se po obsahové stránce drobně lišily v závislosti na registrované uživatelské roli. Na konkrétní registrační formulář by se neregistrovaný uživatel mohl dostat přes odkazy/tlačítka „Registrovat se jako KLIENT“ a „Registrovat se jako PSYCHOLOG“.

Pro jméno a příjmení lze využít běžná textová vstupní pole. Pro e-mail bych využil speciální vstup pro e-mailovou adresu, kde by se ošetřila správnost tvaru vstupu jako e-mailové adresy. Každá e-mailová adresa by se v databázi měla vyskytovat nejvýše jednou a mohla by sloužit jako primární klíč. Přinejmenším se tak zabrání duplicitním registracím na jednu e-mailovou adresu, jako by tomu mohlo být v případě využití uživatelského jména jako unikátního identifikátoru uživatele.

Pro datum narození lze využít speciální vstupní pole pro výběr datumu. Volbu pohlaví je možné vyřešit rozbalovacím seznamem (dropdown) nebo tzv. „radio buttons“ s předpřipravenými volbami.

V registračním formuláři samozřejmě nesmí chybět vstupní textové pole pro vložení hesla a standardem bývá také druhé vstupní textové pole pro opakování hesla, aby mohl být uživatel upozorněn, pokud se hesla neshodují a tedy, že udělal při zadávání chybu.

Speciálně pro roli psychologa by byla ve formuláři možnost nahrát soubor, který by vybral ze svého zařízení.

Pro spárování klienta s psychologem se nabízí řešení, kdy by klient v registračním formuláři vybral psychologa z rozbalovacího seznamu obsahujícího psychology registrované v aplikaci. Takové řešení však nepovažuji za uspokojivé, protože:

- umožňuje registraci v roli klienta komukoliv a
- umožňuje klientovi vybrat si, resp. spárovat se s libovolným psychologem.

Psychologové by pak mohli být „obtěžováni“ notifikačními e-maily o zaregistrování nových klientů, které ani neznají, byť z důvodu omylu klienta, který si vybral v seznamu jiného psychologa než toho svého. Proto spárování klienta s psychologem realizují tak, že každý psycholog bude mít vygenerovaný unikátní osobní klíč (uuid4¹, který bude pro klienta povinné zadat při registraci. Přidělení osobního klíče psychologovi omezí registrace klientů pouze na ty, kteří jsou skutečně v kontaktu s některým zaregistrovaným psychologem a díky tomu obdrželi jeho osobní klíč, který pro registraci využijí. Navíc si klient nemůže vybrat libovolného psychologa, ale bude mu přiřazen aplikací v rámci procesu registrace. Unikátní osobní klíč lze využít pro vyhledání konkrétního psychologa v databázi.

Po registraci klienta by byl ním zvolenému psychologovi aplikací odeslán informativní e-mail o nové registraci.

¹uuid4 je unikátní 128 bitové číslo zapsané jako 32 hexadecimálních číslic, např. ba6eb330-4f7f-11eb-a2fb-67c34e9ac07c.[4]

Zmíněný návrh řešení registrace framework Django plně podporuje a z hlediska bezpečnosti uložení hesel se postará o jejich bezpečné uložení v databázi pomocí hashování.

3.3.2 Omezení přístupu k datům podle rolí

Pro omezení přístupu k některé části aplikace ve frameworku Django lze využít dekorátor `user_passes_test` nebo třídu `UserPassesTestMixin`.

3.3.3 Vyplnění dotazníku

Pro vyplňování dotazníku by sloužilo klientské rozhraní aplikace přístupné po přihlášení k uživatelskému účtu klienta. Po přihlášení by měl klient k dispozici seznam dostupných dotazníků a na základě dohody se svým psychologem by si vybral dotazník k vyplnění. Provedl by vyplnění a odeslání výsledků potvrdil tlačítkem na konci dotazníku. Psycholog by byl o nové responzi od svého klienta upozorněn e-mailem.

Zde by se mohlo uplatnit přiřazení klienta ke konkrétnímu dotazníku psychologem s možným omezením třeba jen na jediné vyplnění (dokud po klientovi nebude další vyplnění požadováno). Klient by pak nemusel dotazník vybírat (nemohl by „sáhnout vedle“) a nemusely by mu být všechny dostupné dotazníky prezentovány. Po přihlášení by byl buď vyzván k vyplnění konkrétního dotazníku nebo by byl informován například hláškou „Žádné dotazníky čekající na vyplnění.“, že se od něj responze na žádný dotazník neočekává.

3.3.4 Systém notifikací

Systém notifikací lze řešit prostřednictvím rozesílání e-mailů při registraci nového klienta, psychologa nebo vyplnění dotazníku klientem. Pro zaslání e-mailů existuje v jazyce Python knihovna `smtplib` a framework Django nad ní tvoří lehkou nadstavbu. Django pak přímo nabízí funkci `send_mail(subject, message, from_email, recipient_list)`, kde zmíněné argumenty funkce jsou povinné (nepovinné argumenty pro stručnost nezmiňuji). Pokud by bylo třeba využít pokročilejší funkce, je k dispozici třída `EmailMessage`, kterou funkce `send_mail` využívá, ale nedisponuje její plnou funkcionalitou.[5]

Pro odesílání e-mailů je potřeba v konfiguračním souboru aplikace nastavit připojení k SMTP serveru – adresu serveru a přihlašovací jméno a heslo. Pro vývoj je možné využít možnosti nastavení výpisu odeslaných e-mailů do konzole, kterou Django podporuje.[6]

3.3.5 Exportování dat

Pro exportování dat do souboru ve formátu vhodném k otevření v programu MS Excel by mohl posloužit standardní modul „`csv`“ jazyka Python.[7]

4 Návrh řešení

4.1 Diagram případů užití

Diagram případů užití (Use Case Diagram) slouží pro grafické a tedy názorné a přehledné vyobrazení stěžejních funkcionalit systému z pohledu uživatele (zpravidla v konkrétní roli). Vidíme na něm akce, které systém umožňuje uživateli v dané roli provádět. Těmto akcím říkáme případ užití.

Případ užití je znázorněn jako elipsa, kde uvnitř je název případu užití. Název bývá zvykem psát jako sloveso v infinitivu, případně jako krátkou větu se slovesem v infinitivu.

Aktér je v diagramu znázorněn jako postavička. Většinou je aktér uživatel, ale může ním být i externí systém², který s naší aplikací interaguje nebo část naší aplikace, která provádí (spouští) některý případ užití.³ Akteři jsou v diagramu znázorněni jako postavičky.

Vazba <<include>> šipkou ukazuje na případ užití, který se má provést vždy, když dojde k provedení případu užití, ze kterého vede šipka směrem ven.

Vazba <<extends>> šipka vede směrem k případu užití, který je doslova rozšiřován o jiný případ užití. Rozšiřovaný případ užití však musí být proveditelný samostatně a úplně za každých okolností nezávisle na rozšiřujícím případě užití. Vyjadřuje pouze volitelnou možnost⁴, třeba i záviselý na nějaké podmínce (i tu lze do diagramu zakreslit).[8]

Do diagramu lze zanést i **dědičnost** šipkou směřující od konkrétnějšího k obecnějšímu, v principu se stejným významem jako v OOP (objektově orientovaném programování). Linka vyjadřující interakci mezi aktérem a případem užití je **asociace**. Do diagramu případu užití je možné zanést také **násobnost** a další informace viz. [9].

4.2 Specifikace případů užití

Specifikace případů užití doplňuje diagram případů užití a vysvětluje přidanou hodnotu případu užití pro aktéra.

Následující podkapitoly obsahují stručné popisy uživatelských rolí systému, diagramy případů užití pro jednotlivé uživatelské role a podrobnější popisy případů užití (specifikaci).

²Např. platební brána, Administrativní registr ekonomických subjektů (ARES) – komunikace mezi systémy obvykle probíhá pomocí API.

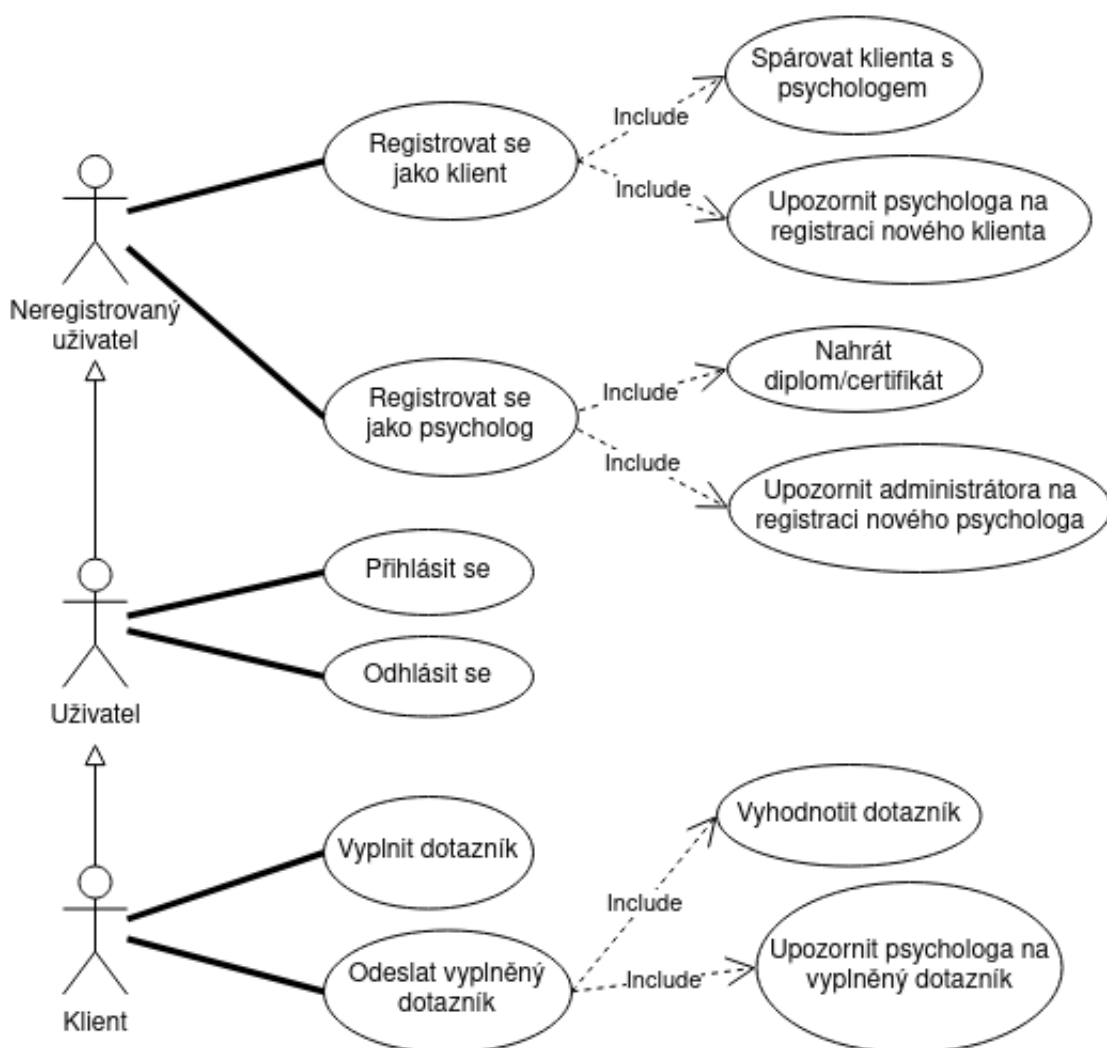
³Představme si třeba blog, ze kterého nám v pravidelných intervalech elektronickou poštou přichází souhrn nových článků zveřejněných za uplynulý týden. Část systému zodpovědná za rozesílání těchto zpráv může být rovněž aktér, byť se nejedná o živého uživatele a její případ užití by mohl být „Odeslat soupis nových článků“.

⁴Pro představu můžeme mít případ užití „Potvrdit objednávku zboží“ a s ním vazbou <<include>> provázaný případ užití „Souhlasit s obchodními podmínkami“ (s těmi je zpravidla nutné souhlasit) a vazbou <<extends>> „Zaslat dotazník spokojenosti“ (volitelné).

4.3 Klient

Role klienta reprezentuje sportovce (respondenta dotazníku/účastníka testování). Po registraci je spárován se svým psychologem, který ho může požádat o vyplnění dotazníku. Na novou žádost o responzi je klient upozorněn. Klient v zásadě v aplikaci může pouze vyplňovat dotazníky (být podroben testování, i opakovaně), pokud je o to požádán. Pokud klient test nedokončí, všechny doposud zadané odpovědi jsou zapomenuty.

4.3.1 Diagram případů užití



Obrázek 2: Diagram případů užití – Uživatel a Klient

4.3.2 Specifikace případů užití

UŽIVATEL

Registrovat se jako klient, Registrovat se jako psycholog – Každý uživatel má možnost vytvořit si uživatelský účet v jedné ze dvou rolí.

Spárovat klienta s psychologem – Každý klient je při registraci propojen se svým psychologem.

Nahrát diplom/certifikát – Každý psycholog musí prokázat svoji odbornost. Nahraný soubor slouží pro ověření odbornosti administrátorem aplikace. Ten pak může rozhodnout o zamítnutí registrace a trvalému vymazání ještě neaktivního účtu.

Upozornit psychologa na registraci nového klienta – Psycholog je upozorněn na registraci nového klienta.

Upozornit administrátora na registraci nového psychologa – Administrátor aplikace je upozorněn, že na schválení registrace čeká nový psycholog.

Přihlásit se – Každému uživateli musí být umožněno přihlásit se ke svému účtu, aby mohl využívat funkcionality a přistupovat k datům určeným pro jeho roli.

Odhlásit se – Každý uživatel musí mít možnost odhlásit se od svého účtu.

KLIENT

Vyplnit dotazník – Každý klient musí mít možnost vytvořit responzi dotazníku, čímž je podroben testování.

Odeslat vyplněný dotazník – Každý klient musí být schopen uložit své odpovědi a tedy dokončit proces testování.

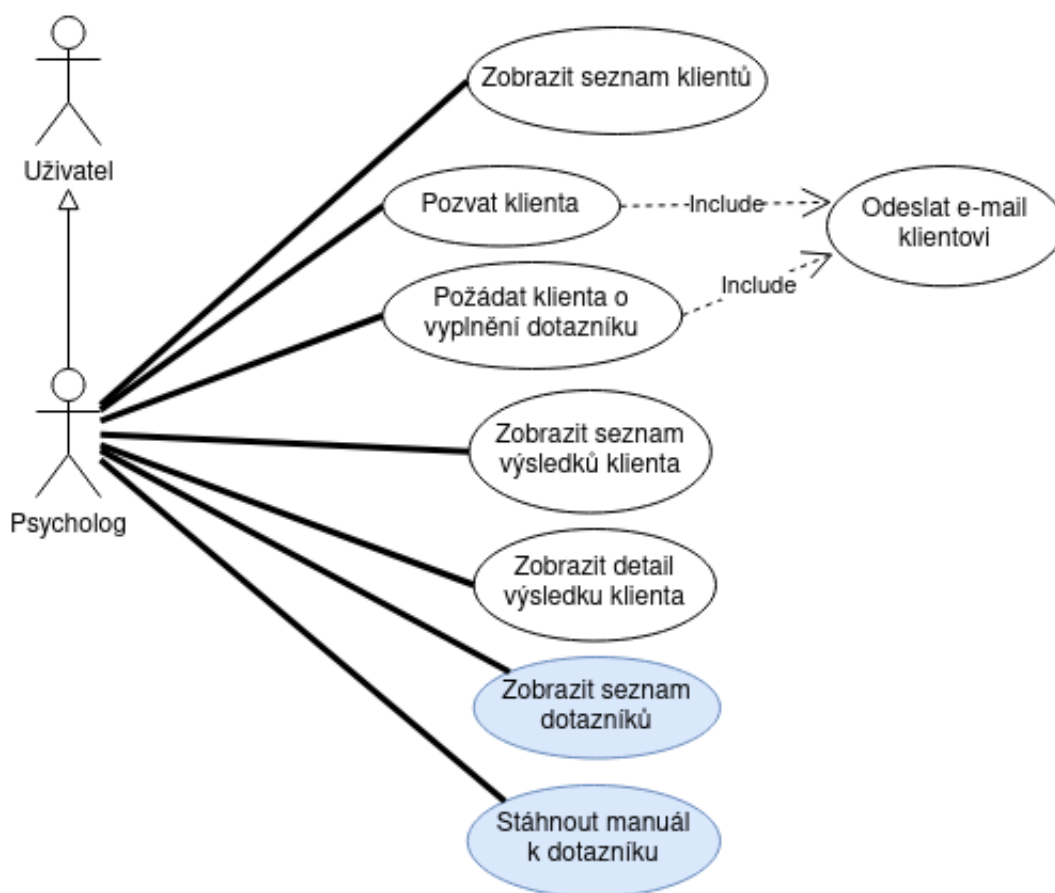
Vyhodnotit dotazník – Systém musí odpovědi klienta umět vyhodnotit, aby výsledky mohl prezentovat psychologům a výzkumníkům.

Upozornit psychologa na vyplněný dotazník – Psycholog je informován, že jeho klient dokončil testování.

4.4 Psycholog

Role psychologa reprezentuje odborníka na sportovní psychologii. Může své klienty podrobovat testování a následně si prohlížet výsledky. Má přístup k historii testování každého klienta. Ke každé testovací metodě (dotazníku) si může stáhnout příručku. Také má přístup k přehledu dostupných testovacích metod. Psycholog má možnost zobrazit si svůj osobní klíč, který slouží pro párování s klientem.

4.4.1 Diagram případů užití



Obrázek 3: Diagram případů užití – Psycholog

4.4.2 Specifikace případů užití

PSYCHOLOG

Zobrazit seznam klientů – Poskytuje psychologovi přehled svých klientů.

Pozvat klienta – Psycholog může odeslat pozvánku k registraci svému dosud neregistrovanému klientovi přímo z aplikace.

Požádat o vyplnění dotazníku – Umožňuje psychologovi testovat klienta.

Odeslat e-mail klientovi – Klient je upozorněn, pokud se po něm vyžaduje interakce se systémem.

Zobrazit seznam výsledků klienta – Umožňuje psychologovi prohlížet historii testování klienta.

Zobrazit detail výsledku klienta – Umožňuje psychologovi prohlížet konkrétní výsledek testování.

Zobrazit seznam dotazníků – Umožňuje psychologovi nahlížet na dostupné testovací metody. Tato funkcionality je dostupná pro Výzkumníka i Vyzkumníka-administrátora aplikace.

Stáhnout manuál k dotazníku – Zpřístupňuje psychologovi příručku k testovací metodě. Tato funkcionality je dostupná pro Výzkumníka i Vyzkumníka-administrátora aplikace.

4.5 Výzkumník

Výzkumník reprezentuje pracovníka FTK UPOL. Kromě toho, že sdílí dva případy užití s rolí psychologa, má k dispozici tabulku responzí (výsledků testování) zvoleného dotazníku. Výsledky může exportovat do formátu, který otevře aplikace MS Excel. Může si prohlížet detail vybrané responze, ale na něm ani v tabulce výsledků nesmí vidět údaje, které prozrazují totožnost klienta.

Výzkumník-administrátor je role, která má v aplikaci speciální význam. Proti standardnímu výzkumníkovi má rozšířené pravomoci. Může spravovat příručky dotazníků (nahrávat a odebírat je), dočasně zakázat nebo povolit užívání testovací metody anebo dotazník z aplikace odstranit. Kontroluje odbornost registrujících se psychologů a rozhoduje o schválení či zamítnutí jejich registrace. Také iniciuje proces vytvoření účtu výzkumníka (bez administrátorských práv). Účty jednotlivých výzkumníků může na libovolně dlouhou dobu deaktivovat a tím jim znemožnit další používání aplikace, nebo opět aktivovat.

4.5.1 Diagram případů užití



Obrázek 4: Diagram případů užití – Výzkumník a Výzkumník-administrátor

4.5.2 Specifikace případů užití

VÝZKUMNÍK

Zobrazit tabulku s výsledky vybraného dotazníku – Každý výzkumník si může zobrazit všechny výsledky zvolené testovací metody (dotazníku).

Exportovat výsledky vybraného dotazníku – Výzkumník má možnost exportovat výsledky pro další zpracování mimo aplikaci.

Zobrazit detail výsledku dotazníku – Výzkumníci mohou nahlížet podrobněji na data výsledků (response), které nejsou z logických důvodů uvedeny v tabulce. Např. kompletní texty otázek.

VÝZKUMNÍK-ADMINISTRÁTOR

Nahrát manuál k dotazníku – Administrátor má možnost zveřejňovat příručky dotazníků psychologům a výzkumníkům.

Odstranit dotazník – Administrátor může z aplikace trvale vymazat dotazník.

Povolit/zakázat sběr výsledků dotazníkem – Administrátor může rozhodnout, zdali mají nebo nemají psychologové možnost testovat své klienty konkrétní testovací metodou.

Zobrazit seznam nově registrovaných psychologů – Dává administrátorovi přehled psychologů, kteří čekají na vyřízení žádosti o přístup do aplikace (schválení registrace). Seznam obsahuje pouze psychology, kteří již potvrdily svou e-mailovou adresu.

Zobrazit certifikát psychologa – Administrátor má možnost posoudit odbornost psychologa, který projevil zájem aplikaci používat registrováním účtu.

Schválit/zamítnout registraci psychologa – Administrátor může kladně nebo záporně vyřídit žádost psychologa o přístup do aplikace.

Zobrazit seznam výzkumníků – Poskytuje administrátorovi přehled výzkumníků, kteří aplikaci využívají.

Vytvořit účet výzkumníka – Administrátor má možnost přidat do aplikace přístup pro výzkumníka, čímž je výzkumník vyzván k dokončení registrace.

Deaktivovat/aktivovat účet výzkumníka – Administrátor může (dočasně) znemožnit další používání aplikace výzkumníkem. K deaktivovanému účtu se nelze přihlásit.

Odeslat e-mail psychologovi – Psycholog je informován o kladném nebo záporném vyřízení jeho žádosti o zřízení účtu.

Odeslat e-mail výzkumníkovi – Výzkumník je v případě vytvoření účtu administrátorem vyzván k dokončení registrace. Taký je informován, pokud byl jeho účet deaktivován nebo opět aktivován.

4.6 Diagram tříd

Diagram tříd je ve své podstatě poměrně konkrétní implementační plán datového modelu aplikace. Proti diagramu případů užití nebo doménovému modelu již obsahuje implementační detaily specifické pro použité technologie, v našem případě pro framework Django. V kontextu Djanga třídě dědicí ze třídy `Model` říkáme *model*.

Třídy jsou v diagramu tříd vyobrazeny jako tabulky se dvěma sloupci. V záhlaví jsou tabulky opatřené názvem, případně i názvem jiné třídy ve dvojitém špičatých závorkách, ze které naše třída dědí. V prvním sloupci jsou názvy atributů třídy a ve druhém datové typy atributů. Obyčejnou čarou je označena vazba 1:1. Plná šipka na konci čáry značí N násobnost ve vazbě 1:N. Pokud je na obou koncích, jedná se o vazbu M:N. Prázdná šipka vyjadřuje dědičnost a vede od potomka k rodiči podobně jako u diagramu případů užití.

Některé třídy v diagramu jsou abstraktní (*abstraktní modely*) a dle konvence mají v názvu prefix „Abstract“. Abstraktní modely se uplatní tehdy, pokud chceme definovat atributy a metody společné pro více dědicích tříd, aniž bychom kvůli tomu vytvářeli duplicitní kód (a tedy porušovali princip DRY⁵). Někoho by mohlo napadnout, že téhož docílíme děděním z klasické třídy, avšak v kontextu Djanga se narozdíl od klasických tříd pro ty abstraktní nevytvoří v databázi tabulka, byť také dědí ze třídy `Model`. Dále není možné v jazyce Python vytvářet instance abstraktních tříd (v kontextu Djanga modelů), podobně jako v jiných programovacích jazycích. Abstraktní třída může obsahovat (ne)kompletní implementaci metod. Narozdíl od dědění rozhraní (*interface*) není při dědění abstraktní třídy nutné implementovat každou deklarovanou metodu. Python pojem *rozhraní* ani nezná. V případě, že dědíme neabstraktní třídu, je v databázi vytvořena tabulka pro rodiče i potomka a v Djangu se tento koncept nazývá „Multi-table inheritance“.[10]

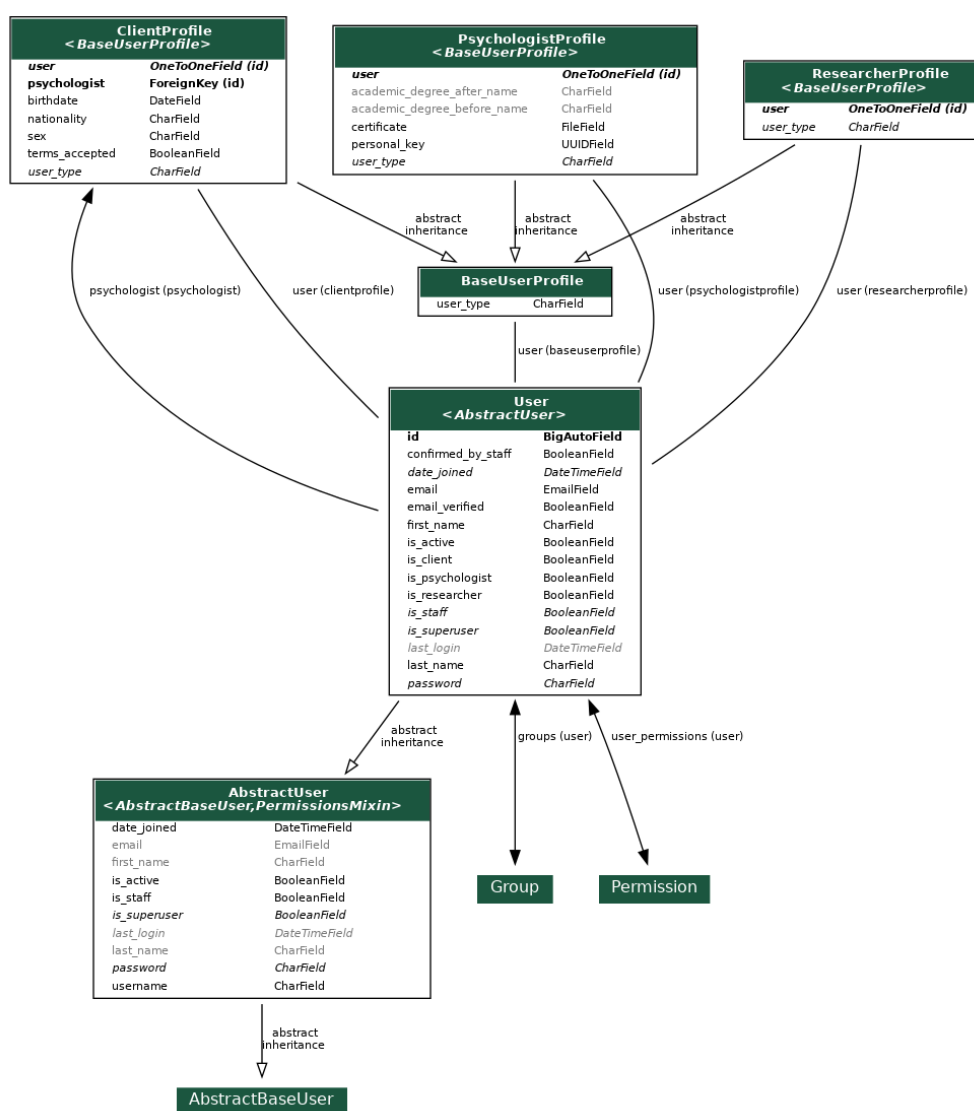
Protože Django implementuje vlastní ORM a databázové tabulky jsou vygenerovány automaticky z vytvořených tříd (modelů) při migraci databáze, datové typy atributů neodpovídají datovým typům použité databázové technologie – jako programátoři jsme od tohoto odstínění a mapování datových typů Djanga na datové typy konkrétní databázové technologie probíhá interně uvnitř frameworku na základě nastavené databáze (určeno v konfiguračním souboru aplikace). Díky ORM máme jednu definovanou třídu (modely) a pokud bychom se rozhodli změnit databázovou technologii, stačí upravit konfigurační soubor aplikace dle dokumentace. Samozřejmě je výběr databáze omezen zejména na ty, které Django podporuje, pokud se chceme vyhnout použití knihoven třetích stran a nechceme si psát vlastní databázový adaptér pro Django. Některá omezení Djanga vzhledem ke konkrétní databázi tvůrci frameworku uvádí viz. [11].

Aplikace *Sportdiag* je rozdělena na dva moduly. Modul *accounts* zajišťuje funkcionalitu potřebnou pro správu uživatelských účtů (registrace, obnovení hesla apod.) a modul *sportdiag*, který řeší procesy kolem dotazníků a implementuje související

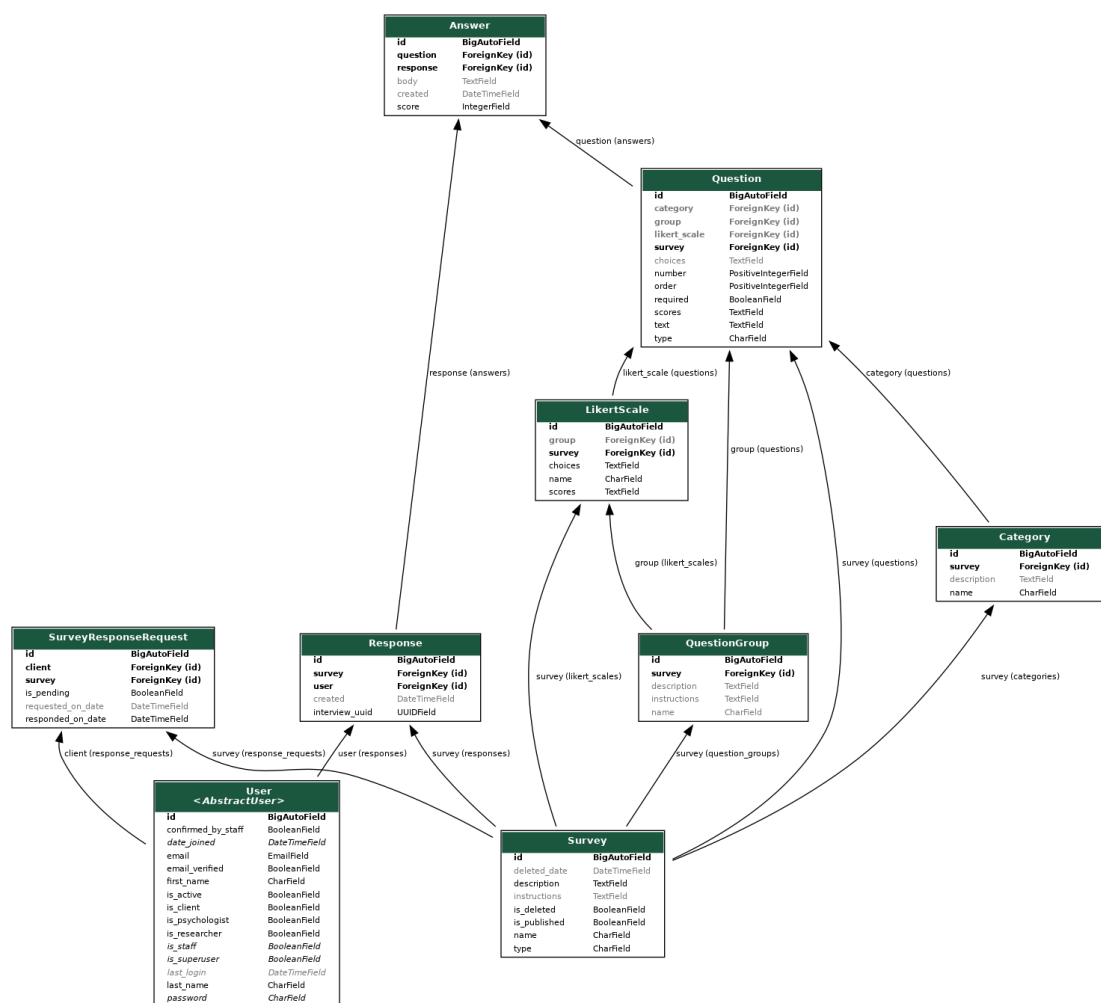
⁵Don't repeat yourself – zjednodušeně řečeno jde o programátorský princip, podle kterého by se v kódu neměly vyskytovat stejné bloky kódu více než jednou.

funkce, které jsou uživatelům k dispozici. Pracovně jsem si návrhy dělal převážně na papír nebo s využitím webové aplikace draw.io. Dříve popsané diagramy tříd umístěné níže jsou vygenerované pomocí balíčku *django-extensions*.

Všimněme si, že datové typy atributů tříd v diagramech jsou konkrétní datové typy, jenž poskytuje samotné Django (jsou popsány v Model field reference [12]). Rovněž třídy, jejichž tabulkám chybí „tělo“ jsou třídy implementované uvnitř frameworku a jejich dědění využijeme část funkcionalitu, kterou nám Django nabízí a popisuje v dokumentaci (v případě dědění ze třídy `Model` právě funkcionalitu ORM). Další zajímavostí je, že vygenerovaný diagram neobsahuje metody, čím se blíží spíše popisu datového modelu, než úplnému diagramu tříd.



Obrázek 5: Diagram tříd modulu *accounts*



Obrázek 6: Diagram tříd modulu *sportdiag*

5 Použité technologie

V této kapitole si stručně popíšeme technologie, které byly užity při vývoji webové aplikace Sportdiag.

5.1 Jazyk Python

Python je moderní, vysokoúrovňový, dynamicky typovaný, multiparadigmatický programovací jazyk. První verze Python 0.9 byla vydána v roce 1991 a její vývoj skončil verzí 1.6.1 v roce 2000. V témže roce následovala zpětně nekompatibilní druhá verze Python 2.0 a její vývoj skončil verzí 2.7.18 v dubnu 2020. Od roku 2008 se programátoři těší používání do dnes aktivně vyvíjené verze jazyka Python 3.0 (rovněž zpětně nekompatibilní s verzemi 2 a 1), přičemž v době psaní tohoto textu je nejnovější a stabilní verzí Python 3.10.2 ze 14. ledna 2022.[13] [14]

Python je vyvíjen společností Python Software Foundation jako open-source⁶ projekt a lze ho nainstalovat na většinu systémových platforem (Unix, Windows, macOS, ...). U distribucí systému GNU/Linux bývá navíc přímo součástí systému.[13] Je tedy nainstalován spolu s operačním systémem. Například na mém počítači s OS Ubuntu 20.04.3 LTS je k dispozici ve verzi 3.8.10 a pod touto verzí proběhl i vývoj aplikace Sportdiag.

Fakt, že Python umožňuje psát kód v objektovém, proceduralním i funkcionálním paradigma z něj dělá pružný nástroj pro vyjadřování myšlenek programátora. Umožňuje programátorovi vybrat si paradigma, které se pro řešení dané úlohy nejvíce hodí nebo které mu vyhovuje. Referenční implementace jazyka CPython je napsána v jazyce C. Instalace Pythonu zahrnuje několik standardních modulů pro usnadnění řešení různých programátorských úloh.

Následuje stručný popis vybraných termínů, které považuji za důležité a základní pro programátora v jazyce Python.

Python Enhancement Proposal (PEP)

Jedná se o verzovaný⁷ dokument, jehož prostřednictvím programátorská komunita podává návrhy na vylepšení jazyka, opravy chyb, ale slouží také jako dokumentace designových rozhodnutí učiněných při vývoji.[15] Pro každého programátora určitě stojí za nahlédnutí dokument PEP8, který se zabývá konvencemi pro psaní kódu v jazyce Python.

Python Package Index (PyPI)

PyPI je repozitář veřejně dostupných balíčků pro vývoj v jazyce Python. Najdeme v něm například i frameworky pro tvorbu webových aplikací v Pythonu jako jsou Django nebo Flask a mnoho dalšího. K instalaci a správě balíčků se používá konzolový nástroj „pip“ (package installer for Python).

Python Virtual Environment

Je izolované běhové prostředí jazyka, které umožňuje instalovat a používat balíčky aniž by došlo k ovlivnění ostatních Python aplikací běžících na stejném systému.[16] Při vývoji aplikace v jazyce Python je velice pravděpodobné, že budeme využívat některé z balíčků z PyPI. Ty pak tvoří závislosti (*requirements*) aplikace na těchto balíčcích. Aplikace, které na našem počítači vyvíjíme, mohou vyžadovat různé balíčky z PyPI a zejména pak jejich rozličné verze. Jediná instalace jazyka Python nám neumožňuje souběžné používání téhož balíčku v různých verzích. Pokud však pro každou naši aplikaci (nejlépe hned zpočátku) vytvoříme virtuální prostředí, můžeme mít v každém prostředí balíček v jiné verzi a ve výsledku udržovat všechny závislosti aplikace v tomto virtuálním prostředí, aniž bychom „znečišťovali“ systémovou instalaci jazyka. Standardní (konzolový) nástroj pro správu virtuálních prostředí je „venv“.

⁶Software s otevřeným zdrojovým kódem. Kód je veřejně k dispozici a lze ho upravovat a šířit v rámci konkrétní licence.

⁷Je uchovávána kompletní historie změn v dokumentu.

Volba jazyka Python byla v podstatě nezbytná, protože je na něm postaven framework Django, který jsem pro tvorbu aplikace použil.

5.2 HTML/CSS

HTML (Hyper Text Markup Language) a CSS (Cascading Style Sheets) jsou dvě pro vývoj webové stránky nebo aplikace zcela nezbytné a základní (frontendové) technologie, jejichž znalostí by měl každý webový vývojář disponovat. O zpracování („přečtení“ a vykreslení) kódu HTML a CSS se stará prohlížeč uživatele.

Kód HTML dává webové stránce strukturu a sémantiku. Prostřednictvím obvykle párových tagů (elementů) a jejich atributů uzavřených ve špičatých závorkách definuje, co je nadpis, odkaz, tlačítko, hlavní obsah stránky nebo odstavec textu. Toho pak využívají internetové vyhledávače, když vyhodnocují kvalitu stránky za účelem zobrazení ve výsledcích vyhledávání nebo asistivní technologie (např. čtecí programy) pro lidi s různými omezeními. Při psaní HTML kódu bychom se měli odstát od toho, jak výsledek zobrazuje prohlížeč, který na většinu HTML elementů aplikuje výchozí vizuální styl.

Vizualizace webové stránky je úloha CSS kódu. Prostřednictvím CSS selektorů a tříd s definovanými CSS vlastnostmi, aplikovanými na HTML elementy říkáme, jak má daný HTML element vypadat. Jakou má mít barvu, velikost, jak velké mají být vnitřní a vnější okraje nebo jaké má být použité písmo a jeho velikost.

Pomocí konceptu CSS Media Query lze dosáhnout změn vzhledu (zmenšit vlastností CSS tříd) na základě rozlišení (obvykle šířky) zobrazovacího zařízení uživatele. Díky tomu je možné vytvořit tzv. responzivní design a stránka se zobrazuje dobře na zařízeních s různě velkou obrazovkou.

5.3 Bootstrap

Bootstrap je front-end⁸ framework s jehož pomocí lze poměrně snadno dosáhnout uspokojivého vzhledu aplikace pro většinu rozlišení uživatelských zařízení. Zjednodušuje způsob, jakým lze docílit responzivního designu webové stránky nebo aplikace. Obvykle, pokud programátor usiluje o responzivní a vizuálně uspokojivý design webu, musí napsat značné množství CSS tříd a využít koncept CSS Media Query k dosažení responzivity. Názvy tříd pak musí správně přiřadit atributu *class* u elementů v HTML šabloně. Díky Bootstrapu psaní CSS tříd vesměs odpadá, protože představuje soubor předpřipravených tříd k přímému použití s vyřešenou responzivitou a zapouzdřuje řešení pro rozmísťování elementů na stránce (tvorbu „layoutu“) pomocí mřížky využívající koncept CSS Flexbox. Bootstrap je postaven na technologii CSS a jazyce Javascript. Od nejnovější verze 5 již nezávisí na javascriptové knihovně jQuery.^[17] Bootstrap také přebírá zodpovědnost za kompatibilitu stylu napříč prohlížeči. Programátor se tedy nemusí strachovat, že by se v hlavních používaných

⁸Front-end je v informatice část aplikace zodpovědná za její vizuální podobu, tvoří tzv. „User Interface“ (UI) – vizuální rozhraní mezi aplikací a uživatelem a jeho fungování zanechává „User Experience“ (UX), neboli „uživatelskou zkušenost“.

prohlížečích (Google Chrome, Firefox, Safari, Microsoft Edge, Opera) stránky nezobrazovaly správně.

Nutno však podotknout, že jelikož se jedná zejména o CSS framework, jsou jeho možnosti omezené a pokud bychom chtěli pracovat na straně klienta (v prohlížeči) se vzhledem a obsahem stránky více dynamicky, spravovat stav stránky a data ze serveru na straně klienta nebo dokonce server o data proaktivně žádat na základě uživatelem provedené akce, aniž by bylo nutné celou stránku překreslit, existují mocnější technologie založené na jazyce Javascript, například front-end frameworky React, Angular nebo Vue.js.

Bootstrap jsem si vybral proto, abych snadněji dosáhnul responzivity a hezcího, uceleného vzhledu aplikace a protože je jeho použití a nasazení jednodušší než v případě javascriptového frameworku. Navíc pro Django existuje balíček „django-crispy-forms“ pro snadné vykreslování formulářů ve vizuálním stylu Bootstrapu bez nutnosti psaní velkého množství HTML (doslova rozepisování jednotlivých formulářů kousek po kousku) a bez potřeby ručního stylování HTML kódu. Bootstrap byl použit ve verzi 5.1.3.

5.4 Django

Většina technologií, knihoven, balíčků a frameworků na poli vývoje software má za cíl usnadnit a urychlit vývoj aplikací a často i zajistit vyšší bezpečnost a nižší náročnost na hardwarové zdroje, než jakých by bylo dosaženo v případě vlastnoručně naprogramovaného řešení. Navíc programátoři řeší při vývoji aplikací tytéž úlohy stále dokola, např. správu uživatelů, uživatelských rolí a oprávnění, správu hesel apod. Proto je snaha řešení bezpečnosti a opakujících se úloh poskytovat skrze knihovny a frameworky, resp. jejich části. Jinak tomu není ani v případě frameworku Django, který je určen pro tvorbu webových aplikací v jazyce Python.

Podívejme se na některé vybrané výhody, jaké nám přináší použití Djanga:

Objektově-relační mapování

Django implementuje vlastní objektově-relační mapper (ORM) – vrstvu mezi naší aplikací a databází, díky které jsme odstíněni při tvorbě datového modelu aplikace a práce s ním (typicky operace CRUD – Create, Read, Update, Delete) od psaní skriptů v jazyce SQL. Jednotlivé databázové entity reprezentují třídy napsané v jazyce Python, dědící ze třídy *Model* a databázové operace nad modelem zapouzdřuje třída *Manager*. Django ORM se postará o vygenerování posloupnosti SQL příkazů nebo dotazu. Navíc Django oficiálně podporuje hned několik databázových systémů (PostgreSQL, MariaDB, MySQL, Oracle a SQLite) a lze poměrně jednoduše drobnými změnami v nastavení aplikace přecházet mezi nimi, aniž by bylo nutné zasahovat do definic modelových tříd.

Generované administrační rozhraní

Django umí na základě modelových tříd vygenerovat webové administrační rozhraní pro správu dat v databázi. Dle dokumentace není zamýšlené k produkčnímu užívání privilegovaným uživatelem (např. uživatelem s oprávněním

správce). Pro tento případ dokumentace doporučuje vytvořit vlastní administráční rozhraní. Nicméně generované rozhraní může sloužit vývojářům k rychlému přidání dat do databáze nebo jejich úpravě pro potřeby vývoje a ladění.

Bezpečnost

Django za dodržení doporučení daných dokumentací zajišťuje určitou míru ochrany:

Cross site scripting (XSS) – Django automaticky nahrazuje specifické znaky (např. `<`, `>`, `'`, `"`, `&`, ...) v proměnných vykreslovaných v HTML šablonách.

Cross site request forgery (CSRF) – Django vynucuje opatření každého formuláře tzv. CSRF tokenem a nastavuje CSRF cookie. Pro příchozí požadavky na server zejména typu POST, PUT, DELETE je přítomnost a správnost CSRF tokenu vyžadována a kontrolována. Pokud token není validní nebo přítomný, vrací server stavový kód HTTP 403 (operace je zakázána).

SQL injection – Ochrana je zajištěna tak, že Django databázové dotazy sestavuje pomocí parametrizace. Dotaz a jeho parametry jsou zpracovávány odděleně a parametry jsou tzv. escapovány s využitím nastaveného databázového adaptéru.

Clickjacking – Django ve výchozím nastavení zakazuje vykreslování stránky v elementu `<frame>` a `<iframe>`.

Podpora SSL/HTTPS – Django přichází s několika doporučeními, jak aplikaci nastavit pro komunikaci klienta a serveru prostřednictvím protokolu HTTPS.

Další informace k zabezpečení lze prozkoumat v dokumentaci frameworku. Aplikace je postavena nad verzí Django 3.2.8.

5.4.1 Balíčky

django-crispy-forms (v1.13.0)

Je balíček, díky němuž jsme velkou měrou ušetřeni od kódování formulářů v šablonách stránek (pomocí HTML a CSS, případně Bootstrapu nebo jiné podobné knihovny). Formulář má s užitím této knihovny aplikovaný výchozí vizuální styl dle nastaveného šablonového balíčku (template pack), který odpovídá jedné z podporovaných knihoven (Bootstrap 2, 3, 4, 5, Tailwind, ...). Pokud bychom chtěli v aplikaci vytvářet formuláře opticky hezčí, než jaké nabízí pouhé HTML a výchozí styly prohlížeče, znamenalo by to napsat značné množství HTML kódu a práci s aplikací CSS tříd, v našem případě z knihovny Bootstrap 5, nebo vytváření tříd vlastních. Tato knihovna zařídí, že HTML elementy formuláře tak, jak je vygeneruje Django, jsou správně ostylované CSS třídami z Bootstrapu 5 a tedy takřka bez práce vizuálně přijatelnější pro dnešního uživatele. Definici formuláře (vstupní pole, jejich pořadí, rozložení elementů

(layout), nastavení CSS třídy popisků polí atp.) je možné provádět zcela na straně serveru. To je užitečné, pokud chceme složitější formulář dynamicky sestavit na straně serveru na základě dat z databáze.

crispy-bootstrap5 (v0.6)

Je „šablonový balíček“ (template pack) pro django-crispy-forms. S jeho užitím lze django-crispy-forms nakonfigurovat tak, že se na formuláře aplikuje vizuální styl knihovny Bootstrap verze 5.

django-cors-headers (v3.11.0)

Díky balíčku je možné z klientské strany (prohlížeče uživatele) vykonávat asynchronní dotazy na server pomocí Javascriptu.

psycopg2 (v2.9.3)

Je potřeba, abychom mohli Django používat s databází PostgreSQL. Knihovna slouží jako databázový adaptér PostgreSQL pro jazyk Python.

python-decouple (v3.6)

V praxi můžeme provozovat více běžících instancí aplikace současně, např. jednu pro účely vývoje a testování a jednu produkční používanou skutečnými uživateli. Každou takovou instanci aplikace takřka určitě budeme chtít provozovat v odlišném *běhovém prostředí*, definovaném množinou proměnných (daného prostředí, tzv. „environment variables“). Tento balíček nám umožňuje načíst proměnné běhového prostředí ze souboru „.env“ umístěného na serveru, s proměnnými před použitím manipulovat a měnit je za běhu aplikace, aniž by bylo nutné opětovně nasazení aplikace. Za zmínku stojí, že Django udržuje konfiguraci aplikace běžně v souboru „settings.py“ včetně například přístupových údajů k databázi (uživ. jméno a heslo) nebo hodnotu proměnné SECRET_KEY, používanou jako „sůl“ pro hashování hesel v databázi. Na jednu stranu chceme soubor „settings.py“ mít verzovaný v repozitáři projektu (často veřejném). Na druhou stranu je trestuhodné hodnoty těchto proměnných zveřejňovat na internetu, protože tak zásadním způsobem narušujeme bezpečnost aplikace. Právě z těchto důvodů jsem se rozhodl použít balíček python-decouple, protože mi umožnil zveřejnit konfigurační soubor v repozitáři, ale bez expozice hodnot proměnných – ty se načtou až ze souboru „.env“ umístěného na serveru.

gunicorn (v20.1.0)

Django v sobě pro účely vývoje zahrnuje vývojový HTTP server, který najde využití při lokálním vývoji. Dle dokumentace však není vhodný pro produkční běh aplikace. Gunicorn slouží jako aplikační server a tvoří vrstvu mezi naší aplikací a webovým serverem (např. Apache nebo Nginx). V praxi pak přichází požadavky ze strany klienta nejprve dorazí na webový server a jsou dále propagovány k aplikačnímu serveru pro zpracování naší aplikací. Narozdíl od vývojového serveru Gunicorn dokáže zpracovávat paralelně více klientských požadavků, což se v produkčním nasazení aplikace projeví jejím vyšším výkonem (např. klienti nebudou čekat na vyřízení požadavku tak dlouho).

django-extensions (v3.1.5)

Balíček tvoří kolekci rozšíření pro Django. V práci byl použit pro vygenerování diagramů tříd.

Faker (v9.8.2)

Je balíček pro generování náhodných, ale smysluplných dat všeho druhu (jména, názvy, e-mailové adresy, hesla, ...). V aplikaci byl použit pro vytvoření testovacích dat (uživatelů) v databázi.

csv

Je standardní modul jazyka Python pro práci s CSV⁹ soubory. V aplikaci byl použit pro vytváření CSV souborů při implementaci funkcionality exportování dat.

5.5 Databáze PostgreSQL a SQLite

SQLite je multiplatformní knihovna napsaná v jazyce C, která implementuje práci se souborem na pevném disku tak, že se zvenčí zdá, že pracujeme se skutečnou databází (můžeme se dotazovat na data jazykem SQL). Pro svoji jednoduchost použití je vhodná pro lokální vývoj a menší projekty. Celou databázi reprezentuje jeden soubor.[18] V Django je nakonfigurována jako výchozí databáze po vytvoření nového projektu. Prohlížet a spravovat data můžeme pomocí nástroje DB Browser for SQLite.

PostgreSQL je již plnohodnotný systém řízení báze dat (SŘBD), vyvíjený jako open-source. Nad touto databází poběží aplikace Sportdiag v produkčním prostředí. Při vývoji byla použita databáze SQLite. PostgreSQL je možné provozovat na operačních systémech Linux, macOS, Windows, BSD a Solaris. V aplikaci je použita verze 14.

5.6 Vue.js

Vue je framework pro tvorbu uživatelských rozhraní postavený na technologiích HTML, CSS a jazyce Javascript. Lze v něm vytvářet plnohodnotné klientské aplikace běžící v prohlížeči a řadí se mezi oblíbené Javascriptové frameworky jako jsou React a Angular. Rozšiřuje klasické HTML o další atributy (tzv. direktivy), pomocí kterých lze psát v kódu HTML podmínky a cykly nebo vázat proměnné na standardní atributy HTML tagů.

Pokud se držíme základního konceptu fungování Django, kde se uplatňuje Server Side Rendering (SSR), dojde vždy při jakémkoliv požadavku na server k opětovnému načtení a překreslení celé stránky v okně prohlížeče (provede se „request-response kolečko“ mezi klientem a serverem). To je nepříjemné ve chvíli, kdy například chceme, aby uživatel aplikace na některé stránce třeba vybral hodnotu z rozbalovacího seznamu a poté potvrdil svoji volbu tlačítkem, čímž iniciuje požadavek na server. Kvůli

⁹CSV (Comma Separated Values) je formát souboru (s příponou .csv), kde jsou hodnoty v řádcích oddělené čárkou. Např. 1. řádek: „Jméno,Příjmení,Věk“ a 2. řádek: „Tomáš,Novák,21“ CSV souboru tabulkové editory (jako MS Excel) otevrou tak, že vytvoří tabulku o dvou řádcích a třech sloupcích.

opětovnému načtení celé stránky prakticky ztratíme informaci o vybrané hodnotě v seznamu (ztratíme stav aplikace na straně klienta). To není příliš žádoucí z pohledu UX. V takovém případě by se nám hodilo, aby požadavek na server proběhl asynchronně a na místo celé stránky server vrátil jako odpověď pouze změněná data. Toho lze docílit s využitím AJAXu. Nicméně stále není vyřešen problém s reakcí uživatelského rozhraní na změnu dat, která nám přišla v odpovědi a překreslení patřičné části uživatelského rozhraní. Stránka je stále stejná jako po jejím prvním načtení. Řešení nabízí Vue, protože jedna z jeho základních vlastností je reaktivita. Vue si pamatuje stav proměnných, které data obsahují a v případě, že v datech dojde k nějaké změně, automaticky překreslí část uživatelského rozhraní tyto data využívající.

Protože Vue bylo poměrně pohodlně integrovatelné prostřednictvím CDN do šablon Django bez nutnosti kroku sestavení celé javascriptové aplikace [19], rozhodl jsem se jej použít, byť velmi okrajově, v šablonách pouze dvou stránek (Přehled klientů [6.3.1] a Přehled výsledků [6.4.1]), kde jsem potřeboval provádět asynchronní dotazy na server a zároveň dosáhnout reaktivity, abych docílil uživatelsky přívětivějšího zážitku z používání aplikace.

5.7 Docker

Docker je open-source řešení, které umožňuje spouštět aplikace nezávisle na infrastruktuře, na které aplikace běží. Zahrnuje sadu nástrojů, díky kterým lze automatizovat proces nasazení aplikace. Pomocí Dockeru aplikaci spustíme v izolovaném kontejneru, který obsahuje všechny závislosti, které jsou k jejímu běhu potřeba (operační systém, běhové prostředí jazyka, knihovny, ...). Kontejnery narozdíl od virtuálních strojů, které abstrahují i hardware, tvoří pouze abstrakci aplikační vrstvy. Jsou tedy nenáročné na výkon stroje.

6 Uživatelská příručka

Rozmístění ovládacích prvků v aplikaci a některé funkce aplikace jsou běžné pro většinu webových aplikací a měly by uživateli být důvěrně známé. Uživatelské rozhraní nijak zvlášť nevybočuje od zažitých zvyklostí. Pro průměrného uživatele by tedy ovládání aplikace mělo být poměrně přímočaré a intuitivní. V opačném případě se uživateli může dostat pomoci v této uživatelské příručce, která se snaží poskytnout základní informace pro úspěšné používání aplikace a informace o umístění ovládacích prvků. Odkazy na hlavní stránky jsou zobrazeny v horní liště aplikace.

6.1 Základní funkcionalita

Nejprve uvedeme funkcionalitu společnou všem uživatelům nehledě na jejich roli. Funkce specifické pro jednotlivé role jsou popsány v samostatných podkapitolách.

6.1.1 Registrace

Vpravém horním rohu se nachází tlačítko „Registrovat se“. Kliknutím na něj je uživatel navigován na stránku pro výběr role (Klient nebo Psycholog). Kliknutím na jedno ze dvou tlačítek je uživatel navigován na registrační formulář pro vybranou roli. Formuláře jsou velmi podobné, v podstatě se liší jen v některých vstupních polích.

Registrace klienta

Pro klienta je důležité mít osobní klíč svého psychologa, aby bylo možné potvrdit a odeslat vyplněný registrační formulář. Pokud jej nemá, měl by se obrátit na svého psychologa. Pokud obdržel klient e-mailovou pozvánku k registraci (která osobní klíč psychologa obsahuje), může kliknout na odkaz ve zprávě. Při využití odkazu se v registračním formuláři automaticky předvyplní osobní klíč psychologa. Osobní klíč psychologa slouží k propojení klienta s psychologem.

Registrace psychologa

Psycholog při registraci nahrává soubor. Ten může být v libovolném formátu a využije se pouze pro ověření odbornosti administrátorem aplikace. Soubor by měl být např. diplom nebo certifikát stvrzující vzdělání v oboru psychologie.

6.1.2 Přihlášení/Odhlášení

Vpravém horním rohu aplikace je umístěno tlačítko „Přihlásit se“ nebo „Odhlásit se“, podle toho, zdali je uživatel přihlášený nebo ne. Kliknutím na „Přihlásit se“ je uživatel přesměrován na stránku s přihlašovací formulářem. Uživatel do něj zadává e-mail a heslo.

6.1.3 Zapomenuté heslo

Na přihlašovací stránce pod formulářem je odkaz na stránku pro obnovení hesla „Zapomněli jste heslo?“. Po kliknutí je uživatel přesměrován na stránku s formulářem,

do kterého vyplní e-mail, pod kterým má registrovaný účet. Po odeslání formuláře kliknutím na tlačítko „Obnovit heslo“ je mu systémem zaslán na uvedenou adresu e-mail s odkazem pro nastavení nového hesla. Po kliknutí na zasláný odkaz je uživateli zobrazen formulář pro zadání nového hesla včetně pole pro opakování hesla. Heslo uživatele není změněno, dokud nepotvrdí odeslání formuláře. Do té doby se může přihlásit se stávajícím heslem. Odkaz je platný pouze pro první otevření.

6.2 Klienti

6.2.1 Žádosti o vyplnění dotazníku

Hlavní stránka klienta. Po přihlášení do aplikace klient vidí seznam žádostí o vyplnění dotazníku, které by měl vyplnit. V seznamu jsou pouze ty dotazníky, u kterých po klientovi vyžádal vyplnění psycholog. Žádosti o vyplnění jsou seřazeny od nejstarší podle data vytvoření žádosti psychologem. Pakliže psycholog žádnou responzi nevyžádal, je uživateli prezentována hláška „Nemáte žádné dotazníky k vyplnění.“

6.2.2 Vyplnění a odeslání dotazníku

Kliknutím na tlačítko „Vyplnit“ na stránce „Žádosti o vyplnění dotazníku“ uživatel otevře dotazník, resp. formulář pro vytvoření nové responze. V pravém horním rohu je možné vyplňování ukončit kliknutím na křížek nebo na tlačítko „Ukončit a zapomenout odpovědi“ dole na konci formuláře. Obě možnosti zobrazí vyskakovací okno pro potvrzení, že klient skutečně chce ukončit testování. Vyskakovací okno má tlačítka „Pokračovat v responzi“ a „Ukončit“. Jejich popisek úplně vystihuje i funkci. Červená hvězdička u textu otázky značí povinnou odpověď. Kliknutí na tlačítko „Odeslat“ pod formulářem uloží klientovu responzi, pokud byly zodpovězeny všechny povinné otázky. V opačném případě se u problematických vstupů zobrazí chybová hláška.

6.3 Psychologové

6.3.1 Přehled klientů

Hlavní stránka psychologa. Obsahuje abecedně seřazený seznam klientů a kliknutím na jméno klienta je psycholog navigován na stránku „Detail klienta“. Nad seznamem je tlačítko „Pozvat klienta“ a přesměruje psychologa na stránku „Pozvat klienta“. Rozbalovací seznam vedle jména klienta obsahuje dotazníkové testovací metody, u kterých je aktuálně povolen sběr odpovědí. Psycholog vybere dotazník v seznamu a kliknutí na tlačítko „Požádat o responzi“ způsobí, že systém zašle informativní e-mail klientovi. Úspěšné odeslání žádosti je signalizováno textem „Responze vyžádána“ v tlačítku a v tuto chvíli nelze vytvořit další žádost, dokud klient nevyplní dotazník.

6.3.2 Pozvat klienta

Stránka, na které psycholog vidí svůj osobní klíč a formulář pro pozvání klienta. Do formuláře se vyplňuje e-mail klienta. Při potvrzení formuláře je kontrolováno, zdali již existuje účet se zadaným e-mailem. Pokud ano, pozvánka se neodešle a psycholog je informován chybovou hláškou. Pokud je vše v pořádku, klientovi je odeslán e-mail s osobním klíčem psychologa a odkazem na stránku s klientským registračním formulářem.

6.3.3 Detail klienta

Na stránce „Detail klienta“ psycholog vidí základní údaje klienta jako jméno a příjmení, e-mailovou adresu, věk a pohlaví. Tuto stránku vidí pouze psycholog daného klienta. Níže v sekci „Výsledky testování“ vidí seznam (historii) všech testování klienta. V seznamu je uveden název testovací metody (dotazníku), identifikátor responze a datum vytvoření responze. Výsledky jednotlivých testování jsou seřazeny od nejnovějšího po nejstarší. Tlačítko „Detail“ otevře stránku „Detail responze“. Pokud klient ještě nebyl testován, v sekci „Výsledky testování“ se zobrazí hláška „Klient dosud nebyl testován.“.

6.3.4 Detail responze

Na stránce „Detail responze“ jsou psychologovi prezentovány základní údaje o responzi včetně celkového dosaženého skóre. Sekce „Subškály“ obsahuje seznam odkazů na tabulku výsledků pro danou subškálu. Kromě navigace slouží jako přehled. Sekce „Shrnutí“ na konci stránky obsahuje souhrnnou tabulku s výsledky pro všechny subškály.

6.3.5 Dotazníky a manuály

Na stránku se psycholog dostane kliknutím na odkaz v horní liště. Stránka obsahuje seznam všech dostupných dotazníkových metod. Každá metoda má stručný popis a lze stáhnout příručku k dotazníku kliknutím na její název.

6.4 Výzkumníci

6.4.1 Přehled výsledků

Výzkumník se po přihlášení nachází na stránce „Výsledky testování“. Ta obsahuje formulář, sloužící jako filtr dotazníkových metod, pro kterou chceme zobrazit výsledky. Po výběru dotazníku provedeme filtraci kliknutím na tlačítko „Filtrovat“. Pod formulářem je tabulka vyfiltrovaných responzí. Pokud pro vybraný dotazník neexistují žádné responze, místo tabulky se zobrazí hláška „Žádná data. Můžete zkusit vybrat jiný dotazník.“. Vpravo nad tabulkou je také tlačítko „Exportovat do .csv“. Kliknutím na něj jsou vyfiltrované responze exportovány do souboru s příponou .csv, které je možné otevřít v MS Excel. V tabulce je sloupec „ID Klienta“. Pomocí ID Klienta může

výzkumník rozpoznat, které responze pocházejí od téhož uživatele, aniž by viděli jeho osobní údaje. Výsledky v tabulce jsou seřazeny od nejnovějších po nejstarší. Tabulka obsahuje zejména sloupce s čísly otázek pod zkratkou O<číslo otázky>. Čísla v řádcích pod těmito sloupci jsou bodová ohodnocení respondentem vybraných odpovědí. Pokud je za označením otázky červená hvězdička, odpovědět na otázku bylo pro respondenta povinné.

6.4.2 Detail responze

Podobně jako psycholog, i výzkumník může nahlížet na detail konkrétního výsledku testování. Dostane se na něj kliknutím na identifikátor responze ve sloupci „ID Responze“ v tabulce výsledků, který je současně odkaz na „Detail responze“. Narozdíl od psychologa výzkumník nevidí na stránce „Detail responze“ jméno a příjmení klienta, ale pouze identifikátor „ID Klienta“.

6.4.3 Dotazníky a manuály

I výzkumníci mají přístup na stránku „Dotazníky a manuály“ a mohou zde stahovat příručky k dotazníkům. Navíc proti psychologům vidí v seznamu dotazníků pod názvem štítek. Štítek může být buď červený s popiskem „Neveřejný“ nebo zelený s popiskem „Zveřejněný“. Výzkumník informuje, jestli má dotazník povolen sběr responzí.

6.5 Výzkumník-administrátor

Výzkumník-administrátor má proti obyčejnému výzkumníkovi přístupnou širší funkcionalitu. V horní liště vidí navíc vedle odkazů „Výsledky testování“ a „Dotazníky a manuály“ odkazy na stránky „Schvalování psychologů“ a „Přehled výzkumníků“.

6.5.1 Schvalování psychologů

Na stránce „Schvalování psychologů“ výzkumník-administrátor vidí tabulku psychologů, kteří provedli registraci, ověřili svoji e-mailovou adresu a nyní čekají na schválení registrace. Administrátor si může kliknutím na tlačítko „Zobrazit certifikát“ stáhnout soubor, který psycholog nahrál při registraci. Tlačítky „Schválit“, případně „Zamítnout“ poté může rozhodnout, zdali psychologovi umožní přístup do aplikace. V případě, že je registrace zamítnuta, vytvořený účet psychologa je vymazán a psycholog je informován e-mailem. Schválení nebo zamítnutí registrace je po kliknutí na příslušné tlačítko ještě nutné potvrdit ve vyskakovacím okně. Registrace čekající na schválení jsou v tabulce seřazeny od nejnovější po nejstarší.

6.5.2 Přehled výzkumníků

Na stránce „Přehled výzkumníků“ se nachází tabulka se zaregistrovanými výzkumníky. Výzkumníci v tabulce jsou seřazeni podle data registrace. Tlačítkem „Deaktivo-

vat/Reaktivovat účet“ může administrátor znemožnit výzkumníkovi v dalším používání aplikace nebo obnovit přístup.

6.5.3 Deaktivovat/Reaktivovat účet výzkumníka

Deaktivace nebo reaktivace účtu výzkumníka se provede kliknutím na tlačítko v tabulce „Přehled výzkumníků“. K deaktivovanému účtu se nelze přihlásit. Provedení operace je třeba potvrdit ve vyskakovacím dialogovém okně.

6.5.4 Vytvoření účtu výzkumníka

Tlačítko „Vytvořit účet výzkumníka“ vpravo nad tabulkou „Výzkumníci“ na stránce „Přehled výzkumníků“ naviguje administrátora na stránku s formulářem, kde administrátor vyplní jméno, příjmení a e-mailovou adresu výzkumníka. Je kontrolováno, zdali pro zadaný e-mail již existuje v aplikaci účet. Pokud takový účet existuje, administrátor je upozorněn chybovou hláškou při kliknutí na tlačítko „Vytvořit účet“ a účet se nevytvoří. Pokud je vše v pořádku, kliknutí na tlačítko „Vytvořit účet“ odešle výzkumníkovi e-mail s vygenerovaným heslem a odkazem pro aktivaci účtu (potvrzení e-mailové adresy). Jakmile výzkumník na odkaz klikne a aktivuje účet, může se pomocí svého e-mailu a zasláného hesla přihlásit do aplikace. Dokud výzkumník na aktivační odkaz v e-mailu neklikne, místo tlačítka „Deaktivovat/Reaktivovat účet“ administrátor vidí v tabulce výzkumníků modré „tlačítko“ s textem „Čekání na dokončení registrace“.

6.5.5 Dotazníky a manuály

Na stránce „Dotazníky a manuály“ jsou k dispozici výzkumníku-administrátorovi pro každý dotazník v seznamu navíc tlačítka „Přidat přílohy“, „Zveřejnit/Zneveřejnit“ a „Odstranit dotazník“.

6.5.6 Přidávání příloh k dotazníku

Po kliknutí na tlačítko „Přidat přílohy“ na stránce „Dotazníky a manuály“ se otevře dialogové okno, ve kterém může administrátor vybrat soubory k nahrání (může jich být více než jeden). Po potvrzení dialogu se soubory nahrají do aplikace k danému dotazníku a psychologové i výzkumníci mohou takto nahrané soubory na téže stránce stahovat.

6.5.7 Povolení/zakázání sběru responzí

Tlačítko „Zveřejnit“, případně „Zneveřejnit“ u daného dotazníku nastaví, zdali dotazník sbírá, případně nesbírá responze. Neveřejné dotazníky psychologové v aplikaci nevidí a nemohou požádat klienta o jejich vyplnění. Výzkumníci mají v tabulce „Výsledky testování“ na stejnojmenné stránce možnost filtrovat responze i pro dotazníky, které nejsou právě zveřejněné.

6.5.8 Odstranění dotazníku

Poslední možnost, kterou na stránce „Dotazníky a manuály“ výzkumník-administrátor má je kliknout na tlačítko „Odstranit dotazník“. Kliknutí vyvolá dialogové okno pro potvrzení operace s upozorněním. Pokud je odstranění potvrzeno, dotazník bude všem uživatelům aplikace trvale nedostupný a nebude se zobrazovat v seznamu na stránce. Nebude už ani možné pro tento dotazník filtrovat responze na stránce „Výsledky testování“. V případě nouze je však možné dotazník i responze opět zpřístupnit, musí tak ale učinit programátor nebo správce aplikace.

7 Programátorská příručka

Aplikace je kontejnerizovaná pomocí technologie Docker a podporuje dvě prostředí – vývojové a produkční.

7.1 Struktura projektu

Na následujícím obrázku vidíme obsah kořenového adresáře projektu „src/“. Názvy bez souborových přípon jsou složky.

```
src/
|-- accounts
|-- bp
|-- docker
|-- docker-compose.dev.yml
|-- docker-compose.prod.yml
|-- manage.py
|-- media
|-- README.md
|-- requirements
|-- sportdiag
|-- static
|-- templates
```

Obrázek 7: Struktura kořenového adresáře projektu

Adresář „accounts“ – obsahuje kód modulu „accounts“.

Adresář „bp“ – obsahuje modul „settings“, ve kterém jsou konfigurační soubory aplikace pro jednotlivá prostředí. Dále v něm nalezneme zdrojové kódy funkcionality sdílené moduly „accounts“ a „sportdiag“ (mixiny, dekorátory, utility, ...).

Adresář „docker“ – obsahuje skripty a soubory potřebné pro sestavení tzv. „docker image“ databázového, webového a aplikačního serveru.

Adresář „media“ – zde se nachází např. soubory nahrané uživatelem aplikace.

Adresář „requirements“ – obsahuje textové soubory popisující závislosti aplikace (název balíčku a verze). Závislosti je třeba nainstalovat pro sestavení a spuštění aplikace.

Adresář „sportdiag“ – obsahuje kód modulu „sportdiag“.

Adresář „static“ – obsahuje soubory CSS a zdrojové kódy jazyka Javascript.

Adresář „templates“ – zde jsou chybové HTML stránky.

7.2 Konfigurační soubory aplikace

Soubor „bp/settings/base.py“ – obsahuje konfiguraci společnou pro vývojové i produkční prostředí.

Soubor „bp/settings/dev.py“ – obsahuje konfiguraci aplikace specifickou vývojovému prostředí. Rozšiřuje base.py.

Soubor „bp/settings/prod.py“ – konfigurace aplikace specifická pro produkčnímu prostředí. Rozšiřuje base.py.

Soubor „bp/settings/local.py“ – obsahuje konfiguraci aplikace specifickou vývojovému prostředí. Rozšiřuje dev.py. Jediný rozdíl proti dev.py spočívá v nastavení databáze SQLite a zobrazování odeslaných e-mailů v konzoli.

7.3 Modelové třídy

Modely databázových tabulek (třídy) se nacházejí v modulech „accounts/models“ a „sportdiag/models“. Django ORM z modelových tříd generuje databázové tabulky. Názvům tabulek je přidán prefix dle názvu modulu, nejmenují se přesně podle tříd. V databázi se vyskytuje obvykle více tabulek, než definujeme modelů. Django si implicitně vytváří v databázi tabulky, které potřebuje pro svou funkci.

7.3.1 Modul *accounts*

Třída BaseUserProfile – bázová třída uživatelských profilů. Zavádí relaci 1:1 s modelem **User** a roli (typ) uživatele. Nevytváří se pro ni v databázi tabulka (je abstraktní).

Třídy ClientProfile, PsychologistProfile a ResearcherProfile – Třídy zajišťují uložení dat specifických pro jednotlivé uživatelské role.

Třída User – Reprezentuje uživatelský účet. Obsahuje mmj. příznaky pro určení role uživatele.

7.3.2 Modul *sportdiag*

Třída Answer – Reprezentuje odpověď na otázku. Drží informace o bodovém hodnocení odpovědi, obsahu odpovědi a datu vytvoření odpovědi.

Třída Category – Prostřednictvím této třídy jsou seskupovány otázky dotazníku do kategorií. Kategorie má kromě jména také popis.

Třída LikertScale – Třída reprezentuje Likertovu škálu v dotazníku. Otázky v liker-tově škále jsou ve formuláři nové responze zobrazeny formou tabulky/matice, kde lze v každém řádku vybrat pouze jedinou odpověď.

Třída Question – Ukládá text otázky, možné odpovědi, typ otázky (číslo, text, výběr z možností, ...), pořadí otázky při vykreslení ve formuláři, číslo otázky v dotazníku (číslo otázky je pevně dané testovací metodou a nesouvisí s pořadím, ve kterém je

otázka vykreslena ve formuláři pro vytvoření responze) apod. V databázi odpovídající tabulka drží informace o tom, kterému dotazníku, kategorii, skupině otázek nebo likertově škále otázka patří.

Třída QuestionGroup – Záznamy v databázové tabulce této třídy udržují dodatečná data pro vykreslení skupiny otázek ve formulaři nové responze. Např. název skupiny, popis a instrukce pro respondenta.

Třída Response – Reprezentuje výsledek testování, např. kdy bylo testování provedeno, identifikátor responze a má vazbu na dotazník a uživatele, který responzi vytvořil. Obsahuje metodu pro výpočet dosaženého skóre v kategorii nebo celkového dosaženého skóre (každá dotazníková metoda definuje vlastní způsob bodového zisku).

Třída Survey – Zavádí typ dotazníkové (testovací) metody. Ukládá název a popis dotazníku, instrukce pro vyplnění a příznak pro určení, zdali dotazník může sbírat responze nebo byl odstraněn administrátorem včetně data vymazání.

Třída SurveyResponseRequest – V odpovídající tabulce v databázi ukládá informace o žádostech psychologa o vyplnění dotazníku klientem. Drží datum pozvání klienta k vyplnění, datum vyplnění dotazníku a kterému klientovi žádost o vyplnění náleží. Má také příznak, který určuje, zdali je žádost stále aktivní nebo byla uspokojena. Data v tabulce slouží pro zobrazení seznamu žádostí o vyplnění klientovi.

Závěr

Výsledkem práce je verze webové aplikace pro psychodiagnostiku Sportdiag splňující základní požadavky zadavatele pana Mgr. Michala Vičara, Ph.D. a považuji ji tak za vhodnou pro nasazení do testovacího provozu, byť reálné nasazení bude vyžadovat ještě určité úsilí. Především by bylo vhodné aplikaci v reálném testovacím provozu zpřístupnit přes bezpečný protokol HTTPS. Podařilo se dosáhnout cíle vytvoření responzivního uživatelského rozhraní, takže by nemělo uživatelům činit potíže používání aplikace z mobilních zařízení.

Když jsem aplikaci začínal vyvíjet, neměl jsem s jazykem Python, frameworkem Django ani Dockerem žádné zkušenosti. Realizace aplikace mě tedy obohatila o nové znalosti. Za dobrý krok považuji kontejnerizaci aplikace s využitím Dockeru, protože díky tomuto rozhodnutí je aplikaci poměrně snadné nasadit na server.

Prostor k technickým vylepšením, která by pocítili i uživatelé, zde určitě je. Kdybych aplikaci vytvářel nyní, neváhal bych uživatelské rozhraní implementovat pomocí některého z javascriptových frameworků a serverovou část aplikace bych realizoval jako REST API.

Conclusions

The result of the thesis is a version of the web application for psychodiagnostics called Sportdiag, which fulfills the basic requirements of the client Mr. Mgr. Michal Vičar, Ph.D. I consider it suitable for deployment in a testing environment, although real deployment will require some additional effort. Above all, it would be appropriate to make the application accessible in a real testing environment through a secure HTTPS protocol. The goal of creating a responsive user interface has been achieved, so using the application on mobile devices should not cause any issues for the users.

When I started developing the application, I had no prior experience with the Python language, Django framework, nor Docker. The realization of the application has, therefore, enriched me with a new knowledge. I consider containerizing the application using Docker as a good decision because it makes the application relatively easy to deploy on a server.

There is certainly room for technical improvements that would benefit the users. If I were to create the application now, I would not hesitate to implement the user interface using one of the JavaScript frameworks and I would realize the server-side of the application as the REST API.

A Testování aplikace

Pro testování aplikace doporučuji využít verze běžící na virtuálním stroji na serveru univerzity. Aplikace je dostupná na adrese <http://158.194.92.116/>.

A.1 Přístupové údaje

Účet klienta v tabulce je spárováný s účtem psychologa „psycholog@example.com“. Je samozřejmě možné vytvořit si účty pro jednotlivé uživatelské role dle libosti.

Účet	Heslo
psycholog@example.com	B*%Ah\$q@Mr7W
vyzkumnik@example.com	%6SwAcz5@u1r
vyzkumnik-admin@example.com	staffresearcherdemo
lutz@example.com (klient)	BCJ2qTJoG)65

Tabulka 1: Přístupové údaje pro testování

B Obsah elektronických dat

doc/

Text práce ve formátu PDF, vytvořený s použitím závazného stylu KI PŘF UP v Olomouci pro závěrečné práce, včetně všech příloh, a všechny soubory potřebné pro bezproblémové vygenerování PDF dokumentu textu (v ZIP archivu), tj. zdrojový text textu, vložené obrázky, apod.

src/

Kompletní zdrojové texty webové aplikace SPORTDIAG se všemi potřebnými (příp. převzatými) zdrojovými texty, knihovnami a dalšími soubory potřebnými pro bezproblémové vytvoření spustitelných verzí programu / adresářové struktury pro zkopírování na webový server.

README.md

Instrukce pro nasazení webové aplikace SPORTDIAG na webový server, včetně všech požadavků pro její bezproblémový provoz, a webová adresa, na které je aplikace nasazena pro účel testování při tvorbě posudků práce a pro účel obhajoby práce.

Literatura

- [1] MINDEVAL. Dostupný z: <
<https://www.mindeval.com/en/>>.
- [2] MINDEVAL. *How does it work?* Dostupný z: <
<https://www.mindeval.com/en/content/howitworks/>>.
- [3] MINDEVAL. *Sport psychology questionnaires*. Dostupný z: <
<https://www.mindeval.com/en/shop/category/spsyq/>>.
- [4] KAMANI, Soham. *A Complete Guide to UUID Versions (v1, v4, v5) – With Examples*. 2021. [Online; navštíveno 5. 8. 2023]. Dostupný z: <
<https://www.sohamkamani.com/uuid-versions-explained>>.
- [5] *Django Documentation. : Sending email*. 2023. Dostupný z: <
<https://docs.djangoproject.com/en/3.2/topics/email/#module-django.core.mail>>.
- [6] *Django Documentation. : Email backends*. 2023. Dostupný z: <
<https://docs.djangoproject.com/en/3.2/topics/email/#console-backend>>.
- [7] *Django Documentation. : Outputting CSV with Django*. 2023. Dostupný z: <
<https://docs.djangoproject.com/en/3.2/howto/outputting-csv/#outputting-csv-with-django>>.
- [8] FAKHROUTDINOV, Kirill. *UML Use Case Extend*. 2022. Dostupný z: <
<https://www.uml-diagrams.org/use-case-extend.html>>.
- [9] FAKHROUTDINOV, Kirill. *UML Use Case Diagrams*. 2022. Dostupný z: <
<https://www.uml-diagrams.org/use-case-diagrams.html>>.
- [10] *Django Documentation. : Model inheritance*. 2023. Dostupný z: <
<https://docs.djangoproject.com/en/3.2/topics/db/models/#model-inheritance>>.
- [11] *Django Documentation. : Databases*. 2023. Dostupný z: <
<https://docs.djangoproject.com/en/3.2/ref/databases>>.
- [12] *Django Documentation. : Model Field Reference*. 2023. Dostupný z: <
<https://docs.djangoproject.com/en/3.2/ref/models/fields/#field-types>>.
- [13] WIKIPEDIE, Příspěvatelé. *Python — Wikipedie: Otevřená encyklopedie*. 2022. [Online; navštíveno 9. 02. 2022]. Dostupný z: <
<https://cs.wikipedia.org/w/index.php?title=Python&oldid=20855708>>.
- [14] WIKIPEDIE, Příspěvatelé. *History of Python — Wikipedia, The Free Encyclopedia*. 2022. [Online; navštíveno 9. 02. 2022]. Dostupný z: <
https://en.wikipedia.org/w/index.php?title=History_of_Python&oldid=1070233233>.
- [15] WARSAW, Barry; HYLTON, Jeremy; GOODGER, David; COGHLAN, Nick. *PEP 1 – PEP Purpose and Guidelines: What is a PEP?* 2000. [Online; navštíveno 9. 02. 2022]. Dostupný z: <
<https://www.python.org/dev/peps/pep-0001/>>.
- [16] *Python documentation*. 2023. Dostupný z: <
<https://docs.python.org/3/glossary.html#term-virtual-environment>>.

- [17] OTTO, Mark. *Bootstrap 5: JavaScript*. 2021. [Online; navštíveno 9. 02. 2022]. Dostupný z: <https://blog.getbootstrap.com/2021/05/05/bootstrap-5/>.
- [18] *What Is SQLite?* 2023. Dostupný z: <https://www.sqlite.org/index.html>.
- [19] *Vue.js In A Django Template*. 2018. Dostupný z: <https://vsupalov.com/vue-js-in-django-template>.