

ZADÁNÍ PROJEKTU Z PŘEDMĚTU ITU

Vítězslav Beran, Michal Kapinus

email: {beranv, ikapinus} @fit.vutbr.cz

18. září 2022

Obecné informace

Zadání projektu do ITU

Řešitelské týmy

Postup řešení projektu

Zadání (1.-6. týden)

Téma projektu (1.-2. týden)

Průzkum uživatelských potřeb a rozdělení práce (2.-3. týden)

Návrh GUI, tvorba makety a testování (3.-5. týden)

Návrh technického řešení (5.-6. týden)

Odevzdání + Registrace zadání (6. týden)

Prezentace (pitch) návrhů projektů (7. týden)

Implementace a testování + Odevzdání (7.-13. týden)

Obhajoba (leden)

Hodnocení

Podmínky splnění projektu

Klíčové vlastnosti programového řešení

Klíčové aspekty řešení projektu

Slovníček pojmů

Changelog

2022-09-26

2022-09-30

Obecné informace

Představte si, že: *jste vývojář v malé SW firmě, která právě dokončila velkou zakázku a chystá se začít pracovat na nové. Každý člen Vašeho týmu byl pověřen, aby ve svém okolí našel nějakého pravidelného uživatele libovolné aplikace (mobilní, na PC, apod.), analyzoval způsob, jakým danou aplikaci uživatel používá a navrhnul změny v uživatelském rozhraní dané aplikace nebo její podčásti, které by uživateli usnadnily její používání. Taktéž lze najít díru na trhu - něco, co by nějaká skupina uživatelů potřebovala a na trhu to ještě není.*

Zadání projektu do ITU

- Najděte si ve svém okolí uživatele a aplikaci, kterou budete inovovat a provedte analýzu používání této aplikace konkrétním uživatelem. Pokud najdete uživatelský problém, pro který zatím žádná aplikace neexistuje, provedte analýzu uživatelským potřeb a jak uživatelé postupují nyní, bez aplikace.
- Navrhněte nový uživatelský proces (aby byl rychlejší a/nebo přirozenější), k tomu informační strukturu UI a GUI.
- Navržené změny implementujte v rámci řešitelského týmu na základě pokynů a požadavků specifikovaných v následujících podkapitolách. Dbejte zejména na:
 - vhodný výběr a správné použití vybrané technologie pro tvorbu GUI
 - architekturu aplikace a správné použití návrhového vzoru MVC
 - pečlivou implementaci a kvalitu zdrojového kódu.
- Výsledek otestujte na uživateli, vyhodnoťte a zdokumentujte.
- Vše zdokumentujte v technické zprávě.
- Výsledek je nutné obhájit v závěrečné prezentaci.

Řešitelské týmy

- Projekt je řešen tříčlennými týmy.
 - V odůvodněných případech je možné řešit projekt ve dvoučlenném týmu, nelze však počítat s úlevou v hodnocení.
- Úvodní část projektu - průzkum a vyhodnocení používání stávající aplikace, provádí a zpracovává každý člen samostatně, viz Postup řešení projektu. Každý člen si najde jiného uživatele a své situace a aplikace, které použije k průzkumu a testování a samostatně tuto analýzu zpracuje a vyhodnotí. Výsledky samostatného průzkumu budou využity celým týmem k výběru společného směru vývoje a dopřesnění zadání.

- Tímy se nebudou vytvářet a registrovat v žádném systému. Tímy vytvořte sami a uvedete je v Zadání (viz níže).

Postup řešení projektu

Následuje popis důležitých termínů v průběhu řešení projektu společně s popisky požadovaných úkonů a povinně odevzdávaných souborů.

Zadání (1.-6. týden)

Téma projektu (1.-2. týden)

Každý člen týmu samostatně

- **ve svém okolí najde** nějakou SW aplikaci s GUI (mobilní, desktopovou, aplikaci na wearable zařízení apod.) a jejího typického uživatele (každý člen týmu jinou aplikaci a jiného uživatele).
- Pokud najdete uživatelský problém, pro který zatím žádná aplikace neexistuje, seznámte se s problematikou v daném tématu.

Společně v týmu

- **vyberte jedno téma**, ujasněte si typické uživatele a proveďte průzkum.

Průzkum uživatelských potřeb a rozdělení práce (2.-3. týden)

Odted' již pracujete POUZE se společně vybraným tématem.

Každý člen týmu samostatně

- **provede průzkum** s uživatelem (každý člen týmu s jiným uživatelem, ale stejnou aplikací) pomocí rozhovorů, pozorování, detailním konkrétním dotazníkem, snímání videa nebo záznamu obrazovky při využívání aplikace apod.
 - Výzkum by měl odpovědět na otázky typu: K čemu uživatel aplikaci potřebuje? Jak ji používá/je nucen ji (aplikaci) používat? Co by vlastně opravdu potřeboval a co je s touto aplikací "nucen" dělat?
- Na základě úvodního vyhodnocení **provede analýzu uživatelských potřeb a klíčových problémů** tohoto řešení.
- **Provede průzkum 2-5 existujících aplikací**, nástrojů nebo řešení, které se týkají vybraného tématu, které nějakým způsobem uživatelské potřeby řeší nebo mohou řešit,
 - Popíše výhody a nevýhody vybraných existujících řešení.

- Alespoň 2 vybrané aplikace budou v rámci týmu jedinečné, tedy žádný jiný člen týmu jejich analýzu neprovádí.
- **Navrhne sadu změn nebo zcela nové řešení** pro vybranou aplikaci nebo její části tak, aby se s ní danému uživateli lépe pracovalo.

Společně v týmu

- diskutujete jednotlivé výsledky průzkumu a zjištění,
- **definujete uživatelské potřeby a klíčové problémy**,
- specifikujete sadu změn nebo zcela nové řešení.

Společně v týmu rozdělíte práci na výsledné aplikaci. Jsou dvě možnosti:

1. **jedna aplikace** (nebo stejná část aplikace), ale **více možností řešení**
 - všichni členové týmu pracují na stejné aplikaci nebo její stejné části,
 - každý člen týmu pracuje na svém řešení stejné části - svůj návrh i realizace GUI,
 - aplikace nebo její část bude mít více řešení, např. pro srovnání apod.
2. **různé části aplikace**, každá část má **jedno řešení**
 - všichni členové týmu pracují na různých částech aplikace,
 - každý člen týmu pracuje na svém řešení své části - svůj návrh i realizace GUI,
 - každá řešená část aplikace bude mít pouze jedno řešení.

POZOR! Každý člen týmu musí v projektu sám autorsky realizovat/implementovat nějakou část aplikace s GUI. Práce v týmu slouží ke snížení režie s realizací backendu, ke vzájemné podpoře a sdílení implementačních zkušeností a především k získání zkušenosti z práce v týmu.

Návrh GUI, tvorba makety a testování (3.-5. týden)

Každý člen týmu samostatně

- **navrhne informační strukturu a GUI**, tj. rozložení informací a interakce do oddílů (stránek/obrazovek/sekcí atd.) a rozložení GUI prvků v jednotlivých oddílech,
- vytvoří maketu navrženého rozhraní,
- provede testování navrženého GUI pomocí vytvořené makety alespoň na 2 uživateli.

Společně v týmu

- diskutujete dosažené výsledky a využijte společně získané zkušenosti.

Návrh technického řešení (5.-6. týden)

Společně v týmu

- navrhnete technické řešení a vyberte technologie k realizaci,
- navrhnete architekturu aplikace, použijte návrhový vzor MVC,

- specifikujte vlastnosti/chování logiky aplikace (Model, Controller) a navrhnete potřebné datové struktury, funkce a API,
- zdokumentujte veškerou dosavadní práci a výsledky do technické zprávy a odevzdejte.
 - Pozor! Pečlivě dbejte na rozdělení individuálních a týmových částí, ať je zcela jasné, jakou činnost a výsledky dělali jednotliví členové týmu.

Odevzdání + Registrace zadání (6. týden)

DO 30. 10. 2022, 23:59:59

Kapitán týmu

- musí zaregistrovat tým a zadání pomocí tohoto [registračního formuláře](#).

Každý člen týmu samostatně

- Odevzdejte technickou zprávu se Zadáním společně s Pitch-prezentací (viz dále). Odevzdejte v IS VUT k aktivitě Projekt - Prezentace zadání. Odevzdávají se 2 soubory a **odevzdává každý člen týmu zvlášť**:
 - Technická zpráva - Zadání (*login_zadani.pdf*), stejná pro všechny členy týmu
 - Pitch-prezentace (*login_pitch.pdf*), individuální, každý člen týmu bude prezentovat svoji práci na zadání (viz níže [Prezentace \(pitch\) návrhů projektů \(7. týden\)](#))
- Technická zpráva obsahuje především:
 - Seznam všech členů týmu, označený kapitán týmu,
 - detailní popis práce odvedené jednotlivými členy týmu:
 - výběr tématu, popis provedeného průzkumu s uživatelem, analýza uživatelských potřeb a klíčových problémů, navržená sada změn
 - popis současného řešení - jaké nástroje uživatel používá, popř. obrázky/screenshotty současného řešení/reálné situace,
 - návrh uživatelského rozhraní, makety rozhraní, testování pomocí maket rozhraní
 - detailní popis společně odvedené práce
 - důvod vybrání tématu týmem,
 - analýza uživatele - co uživatel dělá v reálném životě, jaký je proces/činnost, kterou vykonává a kde by mu příp. vhodný SW mohl pomoci,
 - potřeby uživatele - co je cílem činnosti, čeho chce dosáhnout,
 - shrnutí současných řešení a výsledné pro a proti, inspirace, nápady
 - návrh zadání - co má nové řešení zlepšit, co uživateli umožní, v čem mu konkrétně pomůže, co uživateli konkrétně přinese, co má být konkrétním výstupem, jak má vypadat nový uživatelský proces,
 - návrh řešení (předběžný) - náčrty vlastního řešení, datový model (datové struktury), popis API.

Prezentace (pitch) návrhů projektů (7. týden)

Každý člen týmu samostatně

- v době svého cvičení provést velmi krátkou prezentaci (1,5 min.),
- aktivita je individuální - budete prezentovat svoji vlastní samostatnou práci na týmovém projektu,
- prezentace by měla obsahovat:
 - vlastní návrh na téma projektu a výsledné téma vybrané týmem
 - vlastní průzkum a analýzu uživatelských potřeb, a průzkum konkurenčních aplikací
 - vlastní návrh změn nebo nového řešení
 - týmový návrh výsledné aplikace a rozdělení práce
 - vlastní návrh informační struktury a GUI
 - plánované použité technologie (velmi stručně).

Implementace a testování + Odevzdání (7.-13. týden)

Odevzdání:

- *Projekt*
- *Technická zpráva*
- *Video*

Implementace projektu může být provedena pomocí libovolné technologie na libovolném zařízení (desktop, mobilní zařízení, wearable zařízení apod.), pokud dává daná technologie a zařízení pro vybranou aplikaci smysl. Každý člen týmu musí při obhajobě prokázat, že rozumí funkčnosti jeho části řešení, že ovládá klíčové aspekty zvolené technologie a že je prokazatelně využil ve své části řešení (např. v případě využití webových technologií musí každý člen ve své části použít asynchronní komunikaci pro čtení a úpravu dat).

Společně s programovým řešením se odevzdává také vypracovaná technická zpráva (jedna za celý tým), která popisuje vlastní řešení:

- Obsahuje kompletní obsah technické zprávy z první poloviny semestru (Technická zpráva - téma a návrh)
 - Pokud je to nutné, jeho obsah aktualizujte a vložte jej do technické zprávy
- Konkrétní rozdělení práce v týmu
- Popis použitých nástrojů
- Popis implementace
- Screenshoty výsledné aplikace
- Report z testování
- Použitá literatura

S vytvořenou aplikací provedete **testování**, tj. experiment s minimálně 4-7 uživateli (skuteční potenciální uživatelé nebo alespoň co nejvíce relevantní uživatelé). Experiment má testovat vlastnosti aplikace z pohledu uživatelského použití, tedy nikoliv absenci chyb a pádů

aplikace, ale jak se uživatel s aplikací pracuje, jak často neví, co má udělat, jak moc se ptá. Součástí technické zprávy je také kapitola *Report z testování*, obsahující zejména:

- Na kom bylo testování provedeno.
- Jakým způsobem testování proběhlo a co konkrétně se testovalo.
- Metriky použité pro měření kvality / použitelnosti výsledného řešení v porovnání s referenčním řešením.
- Konkrétní výstupy z testování a konkrétní návrhy na úpravy a další vývoj.

Pro účely vyhodnocení bude součástí odevzdání také video, prezentující práci s aplikací a její klíčové funkce. Je nezbytné, aby video jasně demonstrovalo správné využití použitých technologií (např. MVC, asynchronní práce s daty - čtení i změna - v případě užití webových technologií apod.). Video neslouží jako reklama a jeho estetická stránka nebude hodnocena. Postačující je záznam obrazovky při používání aplikace, volitelně je možné doplnit o stručné titulky nebo popisky vysvětlující, co se na obrazovce děje (ukázka používání, běhu aplikace, ne slajdy s obrázky) - doporučená délka 1 minuta, maximálně 2 minuty.

Projekt se odevzdává v IS VUT, prostřednictvím příslušné aktivity Projekt.

TERMÍN ODEVZDÁNÍ: 14. 12. 2022, 23:59

- Programové řešení i technická zpráva jsou stejné pro celý tým a odevzdává ji **KAŽDÝ** člen týmu.
- Každý člen týmu odevzdává 3 soubory:
 - programové řešení v archivu (*login_src.zip*),
 - obsahuje pouze zdrojové kódy implementovaného řešení, zdrojové kódy musí obsahovat komentáře a hlavičky s konkrétním autorem daného zdrojového kódu.
 - pokud se výjimečně stane, že na jednom zdrojovém souboru pracovali dva autoři a tento zdrojový soubor nelze rozdělit na dva "čistě" autorské, doplňte autory do komentářů k funkcím či k částem autorských kódů.
 - *readme.txt* (nebo *README.md*) s popisem instalace a odkazy na další knihovny.
 - multimediální a další zdrojové soubory.
 - technickou zprávu (*login_tz.pdf*),
 - video (*login_video.mp4* nebo *login_video.avi*).

Obhajoba (leden)

Obhajoby projektů budou probíhat v lednovém termínu. Celková doba na jeden tým (včetně zhodnocení a vystřídání týmů) je 15 minut.

- **Obhajoba slouží k obhájení autorství řešení každého člena týmu a relevantních znalostí dané problematiky.**
- Obhajoby se účastní celý řešitelský tým.
 - Pouze osobně přítomní členové mají nárok na bodové ohodnocení.

- V odůvodněných případech (doložených cestou studijního oddělení formou překážky ve studiu) lze dohodnout jiný termín obhajoby těm členům týmu, kteří se z vážných důvodů nemohli obhajoby zúčastnit.
- Na obhajobu se za tým hlásí pouze kapitán a to ve zveřejněném termínu v IS VUT v době **před** odevzdáním projektu.

TERMÍN REGISTRACE NA OBHAJOBY: 12. 12. 2022 21:00 - 14. 12. 2022 23:59

- Doba prezentace je stanovena na maximálně 6 minut (toto je HARD limit, po kterém bude prezentující komisi vyzván k ukončení prezentace).
- Následná diskuze obvykle trvá max. 7 minut.
- Zbýlý čas je určený k přípravě dalšího týmu, vykrytí případných časových skluzů a podobně.
- Komise určená k hodnocení dané prezentace a projektu se skládá z minimálně **dvou** osob.
- Hodnotí se zejména:
 - kvalita prezentace klíčových aspektů projektu,
 - schopnost odpovídat na relevantní dotazy,
 - dodržení časového limitu.

Hodnocení

Do 2. 1. 2023 obdrží všechny týmy návrh hodnocení na základě odevzdaného programového řešení a podpurných materiálů (specifikace zadání, technická zpráva a video). V případě, že by hodnocení projektu bylo <30 bodů, bude řešitelům projektu nabídnuta možnost dopracovat požadovanou chybějící funkcionalitu (na základě návrhu hodnocení) do termínu obhajob a změnit tak výsledné hodnocení.

Podmínky splnění projektu

Vypracování, odevzdání a provedení následujících aktivit/dokumentů.

- Specifikace týmu, zadání a uživatelských požadavků
- Technická zpráva - zadání + Pitch+prezentace (7. týden)
- Technická zpráva + Zdrojové kódy + Video (13. týden)
- Obhajoba (leden)

Klíčové vlastnosti programového řešení

Základním technickým požadavkem na řešení je důsledné oddělení frontendu (dále jen FE, určeného ke komunikaci s uživatelem) a backendu (dále jen BE, určeného pro uchovávání dat a manipulaci s nimi) aplikace, které spolu komunikují pomocí jasně definovaného rozhraní (API). Oddělení logiky aplikace (modelu) od uživatelského rozhraní (view) je

klíčové pro obhájení projektu. Inspirovat se lze například návrhovým vzorem MVC nebo ekvivalentním.

- Není podstatné, jakým způsobem jsou data uložena (DB, textový soubor, json ve zdrojovém kódu), podstatné je, aby byla uložena na BE a manipulace s nimi byla možná POUZE skrze definované API. Plně funkční DB je sice nice-to-have, ale v ITU se hodnotit nebude.
- V případech, kdy to není nutné pro smysluplný chod aplikace, není vyžadována perzistence dat mezi jednotlivým spouštěním aplikace - např. v případě tvorby hry piškvorky, není třeba ukládat rozehrané ani historické hry.
- Data NESMÍ být uložena přímo na FE, GUI s nimi nesmí ani přímo manipulovat - např. v případě hry piškvorky není stav hry určen stavem jednotlivých komponent GUI, které reprezentují herní políčka, ale modelem umístěným na BE. V případě žádosti o změnu stavu některého políčka ze strany uživatele předá FE skrze volání API na BE požadavek, BE jej vyhodnotí (např. zkontroluje jestli nebyla vytvořena řada koleček nebo křížků vedoucí k ukončení hry, případně jestli nebylo dosaženo remízy, kdy už není možné hru vyhrát žádným hráčem atd.), BE aktualizuje data a informuje FE o výsledku. FE aktualizuje GUI podle aktuálních dat (modelu) - upraví jednotlivé komponenty.
- Pozor, BE nemusí znamenat server apod. BE je principiální oddělení logiky aplikace (dat a funkcí s daty) od GUI (zobrazení a interakce). BE jsou často jen třídy (datové struktury a metody) pro správu dat a logiky aplikace, je-li potřeba, tak i datové úložiště. Velmi zhruba - model MVC (bude probíráno ve výuce) to právě takto specifikuje - Model (data a metody) a View (zobrazení a interakce). Controller pak mapuje akce uživatele (interakci) na konkrétní funkce Modelu a zpětně zajišťuje zobrazení výsledku do View.
- V případě, že pro Vaši nově vznikající aplikaci již existuje vhodný a použitelný BE (např. pokud tvoříte nové GUI již existující aplikace nebo k již existující webové aplikaci tvoříte alternativní mobilní aplikaci), je možné jej využít a v rámci ITU tedy BE vůbec neimplementovat. I v takovém případě je ale nutné v technické zprávě definovat potřebné datové struktury a aplikační rozhraní pro Vaši aplikaci, které zvolený backend implementuje (nebo lze Vámi definované potřeby na datové struktury/API vhodnou úpravou realizovat i s existujícím BE).
- Pokud projekt není přímo zaměřen na nějaké nové řešení pro **registraci, autentizaci či správu rolí**, tak tomu **se nevěnujte**, v ITU to hodnoceno nebude. Toto je vyřešená problematika a jistě to není něco, co uživatelé pro svoji činnost potřebují a proto tomu v projektu nevěnujte pozornost. V ITU svoji energii a pozornost zaměřte na potřebnou interakci pomáhající uživateli dosáhnout svých cílů.

V případě použití webových technologií je pro akceptování řešení podmínkou využití moderních postupů, jako např. asynchronní komunikace mezi FE a BE a to jak pro čtení, tak na úpravu dat. Není akceptovatelné odevzdat aplikaci, která bude pro úpravu jedné položky vyžadovat znovunačtení celé stránky. Taková aplikace bude hodnocena 0 body. V případě využití frameworku, který řeší asynchronní komunikaci za Vás (např. React), bude po

autorech vyžadována znalost, jakým způsobem daný framework asynchronní komunikaci provádí.

Klíčové aspekty řešení projektu

Při řešení projektu budou sledovány a hodnoceny následující aspekty, které budou mít přímý vliv na hodnocení technické zprávy, implementace a obhajoby podle následujícího klíče.

Prezentační kvalita (až 6 bodů)

- Formální a jazyková úroveň technické zprávy.
- Kvalita zdrojového kódu (dekompozice, komentáře atd.).
- Odborná úroveň a obsahová náplň bude hodnocena v rámci Aplikace.

Zadání - “vím, co a proč dělám” (až 10 bodů)

- Z jakých požadavků téma projektu, zadání a návrh vychází (potřeby uživatele).
- Pečlivost a rozsah provedeného průzkumu.
- Rozsah zkoumaných konkurenčních aplikací (dle typu aplikace a možností tak 2-5).
- Kvalita specifikace cíle projektu, uživatelských potřeb a návrhu GUI?
- Jak dobře návrh GUI reflektuje uživ. potřeby? Tedy vysvětlení, že je GUI navrženo právě takto s ohledem na cíle projektu.
- Kvalita vytvořených maket, testování a zapracování nových zjištění o navrženém GUI a návrh technického řešení.

Aplikace - “vím, jak aplikaci správně vytvořit” (až 33 bodů)

- Implementace MVC,
- zdůvodnění výběru technologie,
- popis klíčových specifik/aspektů vybrané technologie pro vytvořenou aplikaci,
- jaké GUI prvky byly využity a jakým způsobem.
- Tyto výsledky, jejich odborná kvalita a obsahová náplň budou prezentovány v Technické zprávě, Zdrojových kódech a na Obhajobě.

Testování - “vím, jak se mi to podařilo” (až 6 bodů)

- Co mělo testování zjistit?
- Jakým způsobem bylo testování provedeno a na jakém vzorku respondentů?
- Jaká jsou zjištění? Co jste se dozvěděli?
- Jak hodnotíte přínos testování a co byste udělali příště jinak?
- Tyto výsledky budou prezentovány v Technické zprávě, kapitola *Report z testování*.

Důležité je dbát na rozdělení činností v týmu. Každý člen musí samostatně realizovat některou část GUI výsledného řešení, ale také velmi dobře znát i práci kolegů v týmu ve všech klíčových aspektech řešení projektu. Není akceptovatelné, aby libovolný člen týmu pracoval pouze na BE. V takovém případě bude daný člen hodnocen 0 body. Ačkoliv je BE nutnou součástí projektu, hodnocena bude pouze práce odvedená na FE a způsob jeho napojení na BE.

Slovníček pojmů

Krátké vysvětlení některých pojmů, které se vyskytují v tomto textu.

- Frontend
 - Uživatelské rozhraní aplikace, prezentační vrstva (View v MVC), zodpovědná za interakci s uživatelem a prezentování dat.
- Backend
 - Vrstva pracující s daty (Model v MVC), uživatel k ní nemá přímý přístup. Frontend komunikuje s backendem výhradně přes jasně specifikované rozhraní. Může, ale nemusí být realizován jako server.
- API
 - Aplikační rozhraní. Sbíрка funkcí, tříd nebo protokolů vybrané knihovny, které jsou dostupné programátorovi při tvorbě aplikace založené na vybrané knihovně.
- MVC
 - Model-View-Controller - SW architektura dělící data aplikace, uživatelské rozhraní a řídicí logiku. Podrobněji prezentována na přednáškách.
- Technická zpráva
 - Dokument odevzdávaný společně s vypracovaným projektem. Detaily viz “Postup řešení projektu”
- Zadání
 - Dokument, který se odevzdává cca v půlce semestru. Obsahuje mimo jiné detailní popis vlastního zadání. Podrobněji viz “Postup řešení projektu”.

Changelog

2022-09-26

- kap. Klíčové vlastnosti programového řešení - přidání informace ohledně autentizace apod.

2022-09-30

- zpřesnění zadání v části: [Průzkum uživatelských potřeb a rozdělení práce \(2.-3. týden\)](#) - *“Alespoň 2 vybrané aplikace budou v rámci týmu jedinečné, tedy žádný jiný člen týmu jejich analýzu neprovádí.”*
- zpřesnění informací ohledně odevzdávání ([Odevzdání + Registrace zadání \(6. týden\)](#))
- zpřesnění zadání pro aktivitu: [Prezentace \(pitch\) návrhů projektů \(7. týden\)](#), včetně větší časové alokace na prezentaci

2022-10-21

- upřesnění požadavků na BE v části [Klíčové vlastnosti programového řešení](#) - *“V případě, že pro Vaši nově vznikající aplikaci již existuje vhodný a použitelný BE (např. pokud tvoříte novou GUI již existující aplikace nebo k již existující webové aplikaci tvoříte alternativní mobilní aplikaci), je možné jej využít a v rámci ITU tedy BE vůbec neimplementovat. I v takovém případě je ale nutné v technické zprávě definovat potřebné datové struktury a aplikační rozhraní pro Vaši aplikaci, které zvolený backend implementuje (nebo lze Vámi definované potřeby na datové struktury/API vhodnou úpravou realizovat i s existujícím BE).”*