

Projekt BookKeeper

Rozfázování vývoje

Tento dokument popisuje jednotlivé fáze vývoje projektu BookKeeper. Je zároveň primárním zdrojem informací k celé projektové dokumentaci.

Obsah

1. Seznam a popis jednotlivých dokumentů
2. Fáze vývoje
 - 2.1. Fáze 0. – Příprava projektu
 - 2.2. Fáze I. – Správa kontaktů
 - 2.3. Fáze II. – Správa faktur
 - 2.4. Fáze III. – Správa bankovních výpisů a hotovostních dokladů
 - 2.5. Fáze IV. – Účetní deník

Seznam a popis jednotlivých dokumentů

Název dokumentu	Účel dokumentu
BookKeeper_DevelopmentStages_v....docx	Tento dokument. Definuje postup vývoje. Zastřešuje dokumentaci.
BookKeeper_TechnicalSpecs_v....docx	Technická specifikace projektu. Jeho jednotlivé části. Použité technologie.
BookKeeper_DatabaseDesign_v....docx	Dodatek k technické specifikaci. Popis databázových tabulek a jednotlivých sloupců.
REST_HowDoesItWork_v....docx	Tutoriál k RESTovému API.
BookKeeper_FunctionalSpecs_v....docx	Funkční specifikace projektu. Rozdělení na funkční celky.
BookKeeper_FunctionalSpecs_Reporting_v....docx	Dodatek k funkční specifikaci. Příklad reportu pro účetní deník.

Fáze vývoje

Fáze 0. – Příprava projektu

GitHub

Vedoucí týmu si naklonuje výchozí projekt z GitHubu:

<https://github.com/sssvt-foltyn/bookkeeper-go>

Vedoucí týmu založí na GitHubu nový repozitář. Do nového repozitáře nakopíruje celý projekt (pozor – do nového projektu NEKOPÍROVAT složku .git z toho původního). Vedoucí pošle pozvánku pro přístup do repozitáře všem členům svého týmu a svému vyučujícímu (můj profil na GitHubu je sssvt-foltyn).

Databáze

Každý člen týmu si zprovozní svoji MySQL/MariaDB datbázi na serveru mysqlstudenti.litv.sssvt.cz. Každý student má uživatelské jméno pro přihlášení k databázi shodné se svým uživatelským jménem ve školní Active Directory. Použije ovšem maximálně prvních 16 znaků. Heslo jsme si říkali minule na hodině. Případně sdělím na vyžádání (přes email nebo přes MS Teams).

Ve výchozím projektu jsou skripty pro vytvoření databázových tabulek a skripty pro jejich naplnění. Ve skriptech je třeba upravit jméno databáze podle prostředí každého studenta.

V back-endové části projektu jsou uloženy připojovací řetězce k databázi v souborech appsettings.json. Připojovací řetězce je třeba všude náležitým způsobem upravit (jméno MySQL/MariaDB serveru, uživatelské jméno, heslo).

Týmové úkoly

Následují jednotlivé úkoly pro celý tým.
Práci rozdělujte a koordinuje vedoucí týmu.

Business objekty

První týmová práce:
Dokončit definici ORM mapování na všech business objektech. Učili jsme se v březnu a v dubnu. Případně se podívejte na business objekt User, který má mapování již hotové.

Datová vrstva

Týmová práce:
Dokončit definici repozitory tříd pro všechny business objekty. Vzorové řešení pro business objekt User máte ve třídě UserRepoMysql.

Aplikační vrstva

Týmová práce:

Připravit nějaké servisní třídy pro jednotlivé business objekty. Vzorové řešení pro business objekt User je ve třídě UserService.

Fáze I. – Správa kontaktů

Povinná část – Kontakty

Správa kontaktů bude zahrnovat všechny základní metody CRUD:

- Načtení a zobrazení seznamu všech kontaktů
- Zobrazení detailu kontaktu
- Editace existujícího kontaktu
- Přidání nového kontaktu
- Smazání vybraného kontaktu

Správu kontaktů pro business objekt Contact je třeba doimplementovat ve všech vrstvách aplikace:

Back-end

1. V datové vrstvě (repo-trída) by již mělo být hotovo.
2. Aplikační vrstvu (servisní třída) bude možná potřeba rozšířit.
3. Vytvořit nový controller, který definuje jednotlivé metody pro REST API.

Front-end

1. Doplnit interface, který bude reprezentovat data daného business objektu na front-endu
2. V servisní vrstvě doplnit příslušnou třídu pro komunikaci s RESTovým API na back-endu.
3. V prezentační vrstvě doplnit komponenty pro zobrazení seznamu, detailu, editaci detailu atd.

Vše lze dělat podle příkladu pro business objekt User.

Dobrovolná část – Přihlášení a odhlášení uživatele

Pro hbitější týmy: Za implementaci této dobrovolné části mohou členové týmu získat bonusovou jedničku.

Očekávaná funkcionalita:

Každá otevřená session uživatele proti aplikaci musí rozlišovat mezi dvěma stavy:

- uživatel je přihlášen
- uživatel není přihlášen

Nepřihlášeným uživatelům se zobrazí pouze login obrazovka.

Přihlášený uživatel má přístup ke všem funkcínostem aplikace.

Kombinace jména/hesla při přihlášení se kontroluje proti databázi.

Fáze II. – Správa faktur

Povinná část – Faktury

Správa faktur bude zahrnovat všechny základní metody CRUD:

- Načtení a zobrazení seznamu všech faktur
- Zobrazení detailu hlavičky faktury
- Editace hlavičky existující faktury
- Přidání nové faktury
- Smazání vybrané faktury (hlavičky i jejích položek)

Předchozí metody se týkaly hlaviček faktur. Je ovšem třeba přidat CRUD i pro položky určité faktury:

- Načtení a zobrazení seznamu všech položek vybrané faktury (zobrazí se např. na detailu hlavičky faktury)
- Zobrazení detailu položky dané faktury
- Editace existující položky k dané faktuře
- Přidání nové položky k dané faktuře
- Smazání vybrané položky dané faktury

Postup při implementaci všech výše uvedených use-case-ů bude víceméně podobný postupu pro implementaci modulu pro správu kontaktů.

Fáze III. – Správa bankovních výpisů a hotovostních dokladů

Povinná část – Bankovní výpisy

Správa bankovních výpisů je v podstatě identická se správou faktur. Požaduje se tatáž funkcionality pro CRUD metody hlaviček výpisů i CRUD metody položek výpisů.

U každé položky bankovního výpisu (při jejím přidání nebo při její editaci) musí mít uživatel možnost vybrat z číselníku kód účetní operace pro tuto položku.

Pokud je bankovní výpis označen jako "zaúčtovaný v účetním deníku", nesmí mít uživatel možnost upravovat ani mazat jeho hlavičku ani upravovat, mazat, nebo přidávat položky takového bankovního výpisu.

Povinná část – Hotovostní doklady

Správa hotovostních dokladů je taktéž identická se správou faktur. Opět je třeba funkcionality pro CRUD metody hlaviček dokladů i CRUD metody položek dokladů.

U každé položky hotovostního dokladu (při jejím přidání nebo při její editaci) musí mít uživatel možnost vybrat z číselníku kód účetní operace pro tuto položku.

Pokud je hotovostní doklad označen jako "zaúčtovaný v účetním deníku", nesmí mít uživatel možnost upravovat ani mazat jeho hlavičku ani upravovat, mazat, nebo přidávat položky takového hotovostního dokladu.

Fáze IV. – Účetní deník

Na rozdíl od kontaktů, faktur, bankovních výpisů a hotovostních dokladů nemá mít účetní deník na front-endu k dispozici CRUD operace. Do deníku se data dostávají jiným způsobem než přímým vkládáním od uživatele. Záznamy v deníku vznikají tzv. "zaúčtováním".

Povinná část – Zaúčtování bankovního výpisu do deníku

Uživatel má možnost zvolit si mezi nezaúčtovanými bankovními výpisy jeden výpis, který nechá zaúčtovat. Aplikace projde postupně všechny položky zvoleného výpisu a založí obdobné položky v účetním deníku s tím, že tak trochu "zkopíruje" data z položky bankovního výpisu do odpovídajících polí nové položky účetního deníku. Zároveň u této nové položky deníku zaznamená, že položka vznikla z bankovního výpisu, resp. jeho položky tak, že se uloží ID položky výpisu do příslušného sloupce v tabulce BK_JOURNAL.

Při účtování jednotlivých položek zvoleného bankovního výpisu se musí správně vyplnit odpovídající sloupce BANK_CREDIT, BANK_DEBIT a BANK_BALANCE. Nový zůstatek (BANK_BALANCE) se dopočítá ze zůstatku na účtu v bance, který vznikl při zaúčtování předešlé položky bankovního výpisu (buď tohoto stejného, anebo předešlého, pokud účtujeme první položku zvoleného výpisu).

Povinná část – Zaúčtování hotovostního dokladu do deníku

Uživatel má možnost zvolit si mezi nezaúčtovanými hotovostními doklady jeden doklad, který nechá zaúčtovat. Aplikace projde postupně všechny položky zvoleného dokladu a založí obdobné položky v účetním deníku s tím, že tak trochu "zkopíruje" data z položky hotovostního dokladu do odpovídajících polí nové položky účetního deníku. Zároveň u této nové položky deníku zaznamená, že položka vznikla z hotovostního dokladu, resp. jeho položky tak, že se uloží ID položky dokladu do příslušného sloupce v tabulce BK_JOURNAL.

Při účtování jednotlivých položek zvoleného hotovostního dokladu se musí správně vyplnit odpovídající sloupce CASH_CREDIT, CASH_DEBIT a CASH_BALANCE. Nový zůstatek (CASH_BALANCE) se dopočítá ze zůstatku ve fyzické pokladně, který vznikl při zaúčtování předešlé položky hotovostního dokladu (buď tohoto stejného, anebo předešlého, pokud účtujeme první položku zvoleného dokladu).

Dobrovolná část – Report z účetního deníku

Za vypracování této dobrovolné části mohou dostat členové týmu bonusovou jedničku.

Report je popsán v dokumentu BookKeeper_FunctionalSpecs_v....docx a jeho vizuální stránka je načrtnuta v dokumentu BookKeeper_FunctionalSpecs_Reporting_v....docx.