



Dokumentace projektu IDS

Název projektu: Manažerský kalendář

1. část projektu

Tým: TrifOvs

Dmytro Trifonov

xtrifo00

Yelyzaveta Ovsianikova

xovsia00

Obsah

1	Zadání	2
2	Popis datového modelu	2
2.1	ER diagram	2
2.2	Diagram případů užití	3
3	Databázové operace	4
3.1	DROP	4
3.2	CREATE	4
3.3	INSERT	5
3.4	TRIGGER	5
3.5	EXPLAIN PLAN	5
3.6	PROCEDURE	5
3.7	PRIVILEGES	5
3.8	MATERIALIZED VIEW	5

1 Zadání

Manažerský kalendář

Ředitel vaší firmy se na vás obrátil s požadavkem na vývoj aplikace, kterou charakterizoval takto: Systém bude používat vedení firmy, tj. ředitel a vedoucí oddělení (manažeři) a sekretářky na obou úrovních. Ředitel a manažeři, případně jejich sekretářky budou vkládat informace o plánovaných akcích. Ředitel má možnost vidět kromě svého kalendáře i kalendáře všech manažerů, ti vidí pouze svoje, z ředitelova dostávají pouze informaci o termínech, kdy nebude ředitel přítomen. Ředitel má možnost manažerům sám některé akce plánovat. Systém musí v takovém případě zajistit odeslání zprávy příslušnému manažerovi. Systém také musí poskytovat informace o volných časech vybraných či všech manažerů, aby ředitel mohl plánovat některé společné akce.

2 Popis datového modelu

2.1 ER diagram

ER diagram obsahuje tyto entity:

Uživatel: Tato entita představuje každého uživatele systému, včetně ředitelů a manažerů a jejich sekretářek.

Sekretářky: mají v rámci tohoto systému stejná práva jako jejich nadřízení, mají také své vlastní účty a jsou plnohodnotnými uživateli, ale při každé akci se vygeneruje zprávu, aby bylo jasné, kdo akci provedl.

Ředitel: Ředitel může upravovat nejen svůj kalendář, ale také kalendáře všech manažerů (bude vygenerována zpráva), a díky atributu dostupnosti v entitě události může získat informace o dostupnosti manažerů v daném čase.

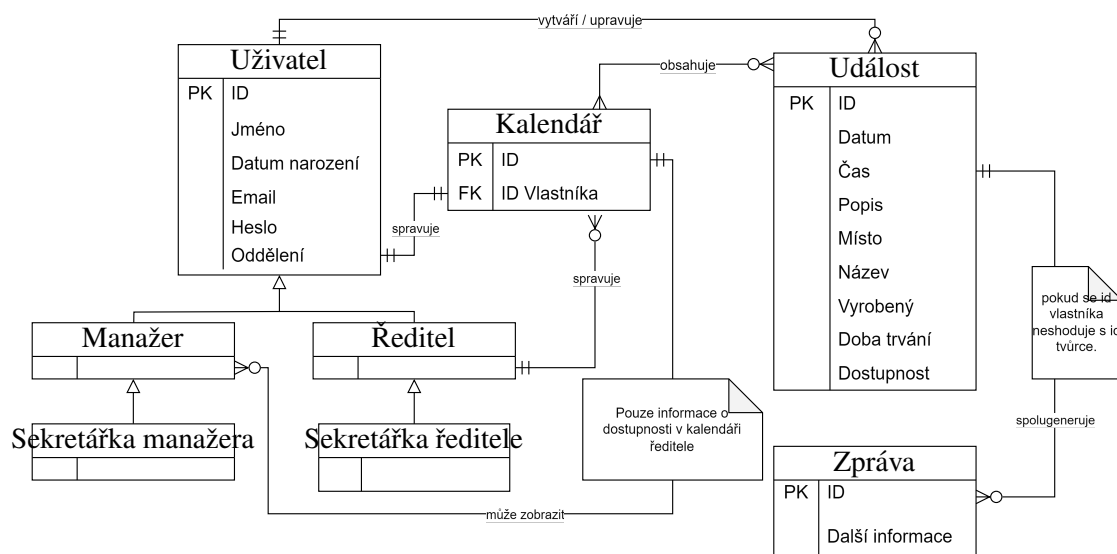
Manažer: Může upravovat pouze svůj vlastní kalendář a má také přístup k informacím o momentální nedostupnosti ředitele. Vypadá to jako stejný kalendář, ale všechny atributy kromě dostupnosti jsou nahrazeny standardní hodnotou (např. defaultní).

Kalendář: Každý kalendář je spojen s konkrétním uživatelem a oddělením. Umožňuje záznam plánovaných událostí.

Událost: Entita události obsahuje všechny důležité informace o plánovaných akcích a schůzkách v kalendářích.

Zpráva: Systém generuje zprávu jako zprávy o nově plánovaných akcích nebo o změnách v kalendáři, vyrobených osobami, které nejsou vlastníky.

Pozn.: V kontextu tohoto systému předpokládáme, že ředitel a manažer mohou mít několik sekretářek, které s nimi budou propojeny prostřednictvím oddělení.



Obrázek 1: ER diagram

2.2 Diagram případů užití

Diagram případů užití obsahuje následující funkce:

Přihlášení: Umožňuje uživatelům vstoupit do systému s ověřením jejich údajů.

Zobrazení událostí: Uživatelé mohou prohlížet události ve svém kalendáři.

Zobrazení událostí ostatních: Umožňuje vidět události v kalendářích ostatních uživatelů dle rolí a oprávnění.

Zjištění nepřítomnosti ředitele: Manažeři mohou zjistit, kdy ředitel nebude přítomen.

Volný čas manažera: Funkce pro zobrazení volných časových slotů v kalendáři manažera.

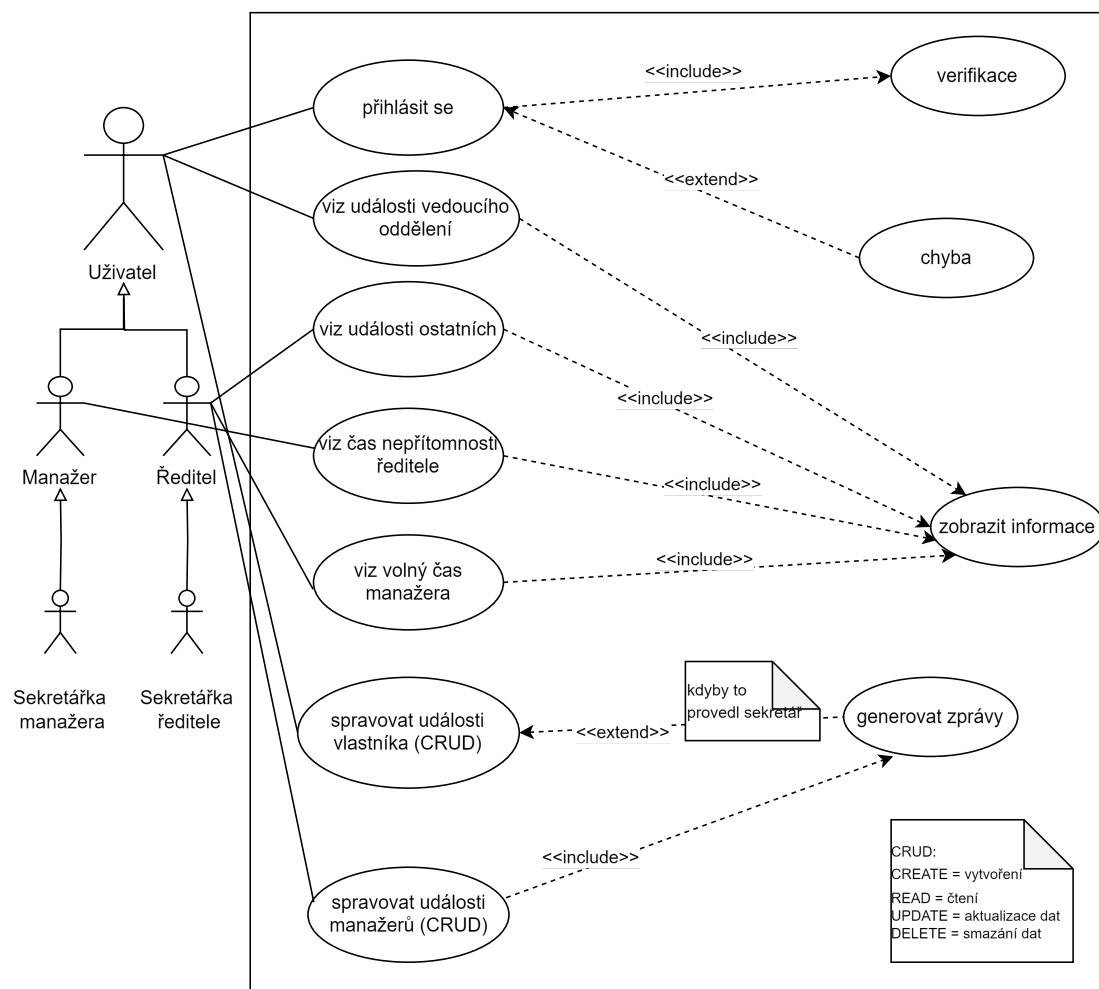
Správa vlastních událostí (CRUD): Uživatelé mohou vytvářet, číst, upravovat a mazat události.

Správa událostí jiných manažerů (CRUD): Ředitel může spravovat události manažerů.

Zobrazení informací: Zobrazuje informace spojené s vybranými funkcemi systému.

Generování notifikací: Pokud událost vytvoří sekretářka, systém vygeneruje notifikace.

Diagram také ukazuje, že některé případy užití ("zobrazení informací") jsou centrální pro zobrazení relevantních údajů a jsou podporovány dalšími akcemi.



Obrázek 2: Diagram případů užití

3 Databázové operace

Tato část dokumentuje klíčové operace použité při nastavení a údržbě databáze. Každá operace je pečlivě popsána s ohledem na její funkci a účel v systému.

3.1 DROP

Operace DROP se používají k odstranění existujících struktur v databázi, jako jsou tabulky, trigger a sekvence. Toto je nezbytné k zabránění konfliktům, pokud je potřeba tyto objekty znovu vytvořit. Například, DROP TABLE příkaz odstraní tabulku a všechna její data z databáze.

3.2 CREATE

CREATE příkazy se používají k definování nových struktur v databázi. Zahrnují vytváření tabulek, definování primárních klíčů, trigger, procedury a materializované pohledy. Tyto objekty jsou základem pro ukládání a manipulaci s daty v databázi.

3.3 INSERT

Příkazy `INSERT` slouží k vložení dat do tabulek. Tato data mohou být počáteční hodnoty potřebné pro funkčnost systému nebo testovací data použitá při vývoji.

3.4 TRIGGER

Triggery jsou speciální procedury spouštěné automaticky v reakci na určité události v databázi, jako je vložení nebo aktualizace záznamů. Triggery mohou být použity k udržení konzistence dat a automatickému provádění rutinních operací.

3.5 EXPLAIN PLAN

Příkaz `EXPLAIN PLAN` se používá k analýze a zobrazení plánu provádění dotazů. To pomáhá optimalizovat výkon dotazů tím, že odhaluje, jak databázový systém interpretuje a provádí specifické SQL příkazy.

3.6 PROCEDURE

Uložené procedury jsou skupiny předdefinovaných SQL příkazů, které lze vyvolat pro provedení databázových operací. Tyto procedury zjednodušují opakující se úkoly a zvyšují bezpečnost a efektivitu práce s databází.

3.7 PRIVILEGES

Přidělování oprávnění uživatelům nebo skupinám uživatelů je klíčové pro řízení přístupu k databázovým objektům. Oprávnění určují, co mohou uživatelé dělat, jako je čtení, úpravy, vkládání nebo mazání dat.

3.8 MATERIALIZED VIEW

Materializované pohledy jsou databázové objekty, které obsahují výsledky dotazů. Na rozdíl od standardních pohledů jsou data materializovaných pohledů uložena na disku, což umožňuje rychlejší přístup k datům na úkor vyšší nároky na úložiště.