



E-HEALTH

Závěrečná správa



28. APRÍLA 2019
FRANTIŠEK TOMANA 3AIN
Univerzita Komenského

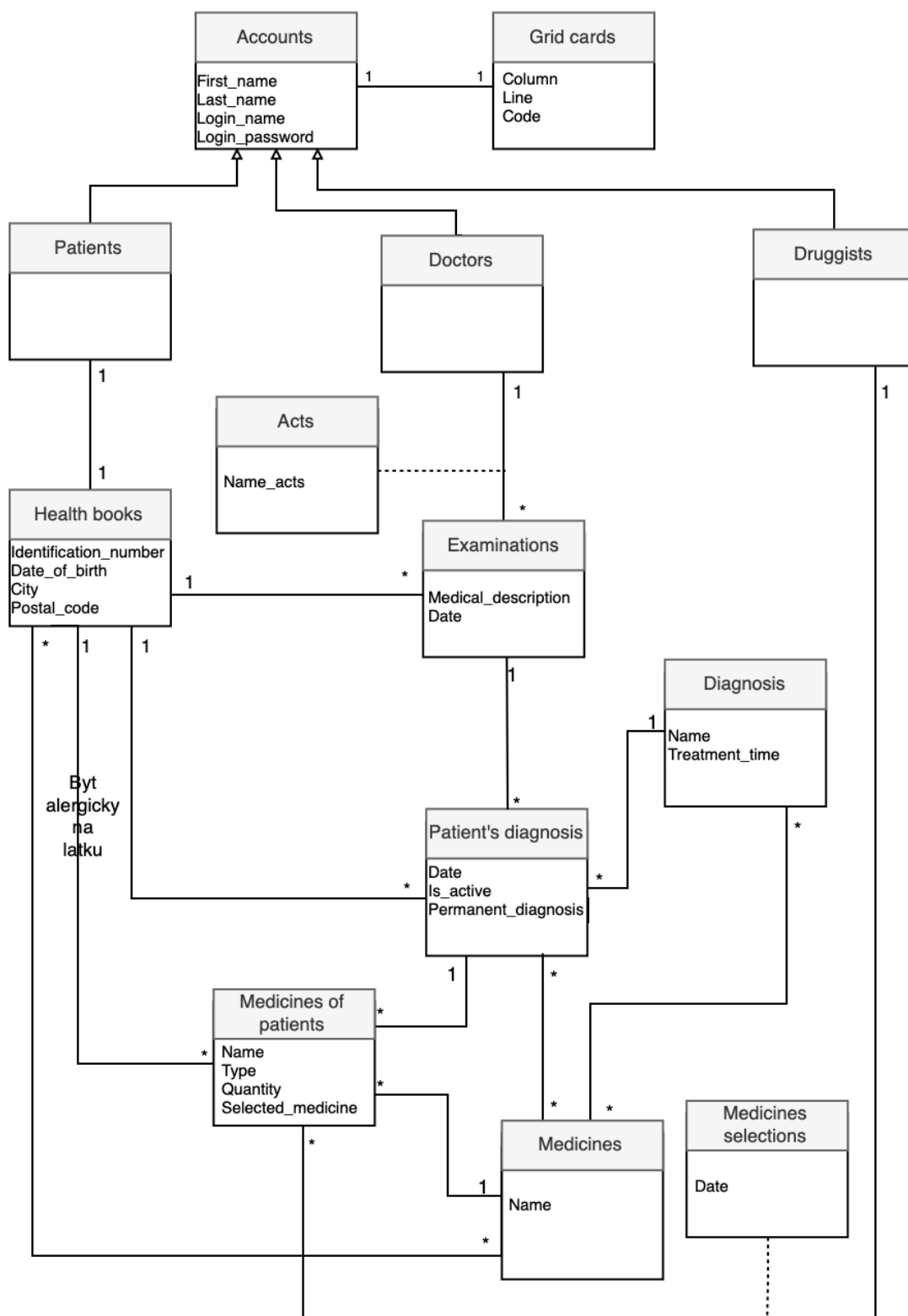
OBSAH

<u>PREDSLOV</u>	<u>3</u>
<u>DÁTOVÝ MODEL</u>	<u>4</u>
<u>ZOZNAM FUNKCIÍ</u>	<u>6</u>
<u>RELAČNÁ DATABÁZA.....</u>	<u>7</u>
<u>ORGANIZÁCIA KÓDU.....</u>	<u>8</u>
<u>OPTIMALIZÁCIA SQL.....</u>	<u>10</u>
<u>VÝBRANÝ RIEŠENÝ PROBLÉM.....</u>	<u>11</u>

PREDSLOV

Tento dokument obsahuje popis môjho systému, z názvom E-health, ktorý sa zaoberá problematikou zdravotníctva a rieši funkcionality troch entít lekárov, pacientov a lekárnikov.

DÁTOVÝ MODEL



Obr. 1: Entitno-relačný model dát pre systém E-health

Popis vzťahov medzi entitami:

Accounts: v aplikácii pod túto množinu spadajú tri entity lekári, pacienti a lekárnici.

Acts: väzobná tabuľka medzi vyšetreniami a doktormi.

Grid cards with accounts: sú vo vzťahu 1:1, pretože každá osoba potrebuje mať v systéme kartu, s ktorou sa bude verifikovať.

Health book with patient: sú tiež vo vzťahu 1:1, pretože každý pacient musí mať svoju vlastnú zdravotnú knižku, ktorý obsahuje zdravotný stav pacienta.

Examination with health book: k jednej zdravotnej knižke sa budú viazať viaceré vyšetrenia a preto obsahuje vzťah N:1.

Doctor with examination: k jednému vyšetreniu sa viaže konkrétny doktor, ale tiež tento doktor môže vykonávať viacero vyšetrení 1:N.

Health book with medicines: jeden pacient môže mať viacero liekov a konkrétny liek môže mať viacero pacientov teda N:N a vznikne nám väzobná tabuľka alergie pacienta.

Medicines of patients with health book: vzťah, ktorý sa priamo viaže na jedného pacienta a opisuje, ktoré konkrétne lieky pacient užíval alebo užíva N:1.

Patients diagnosis with health book: podobne ako medicines of patients with health book, akurát sa viaže na pacientove diagnózy N:1.

Examination with patients diagnosis: na základe jedného vyšetrenia môže byť stanovených viacero diagnóz a preto vzťah 1:N.

Patients diagnosis with diagnosis: pacientove diagnózy sú stanové z tabuľky všetkých diagnóz a preto vzťah N:1.

Diagnosis with medicines: vzťah, ktorý opisuje aké lieky sú na diagnózy na viacero diagnóz môže existovať jeden liek a naopak a preto N:N, z tohto vzťahu vznikne väzobná tabuľka ktorá bude obsahovať množinu liekov, ktoré sú dostupné na diagnózu.

Patients diagnosis with medicines: vzťah N:N popisujúci aké lieky sú na diagnózu a na základe tohoto vzťahu vznikne väzobná tabuľka, ktorá bude obsahovať lieky na konkrétnu diagnózu.

ZOZNAM FUNKCIÍ

Moja aplikácia zahŕňa tri entity pacientov, lekárov a lekárnikov. Na spravovanie systému som vytvoril admina, ktorý sa nemusí autentifikovať pomocou hesla a grid karty v systéme.

Každá z entít okrem admina sa musí v systéme autentifikovať a to pomocou svojho mena, priezviska a prihlasovacieho mena, hesla. Po splnení prvej časti sa musí druhou zložkou autentifikovať pomocou číselného kódu z grid karty.

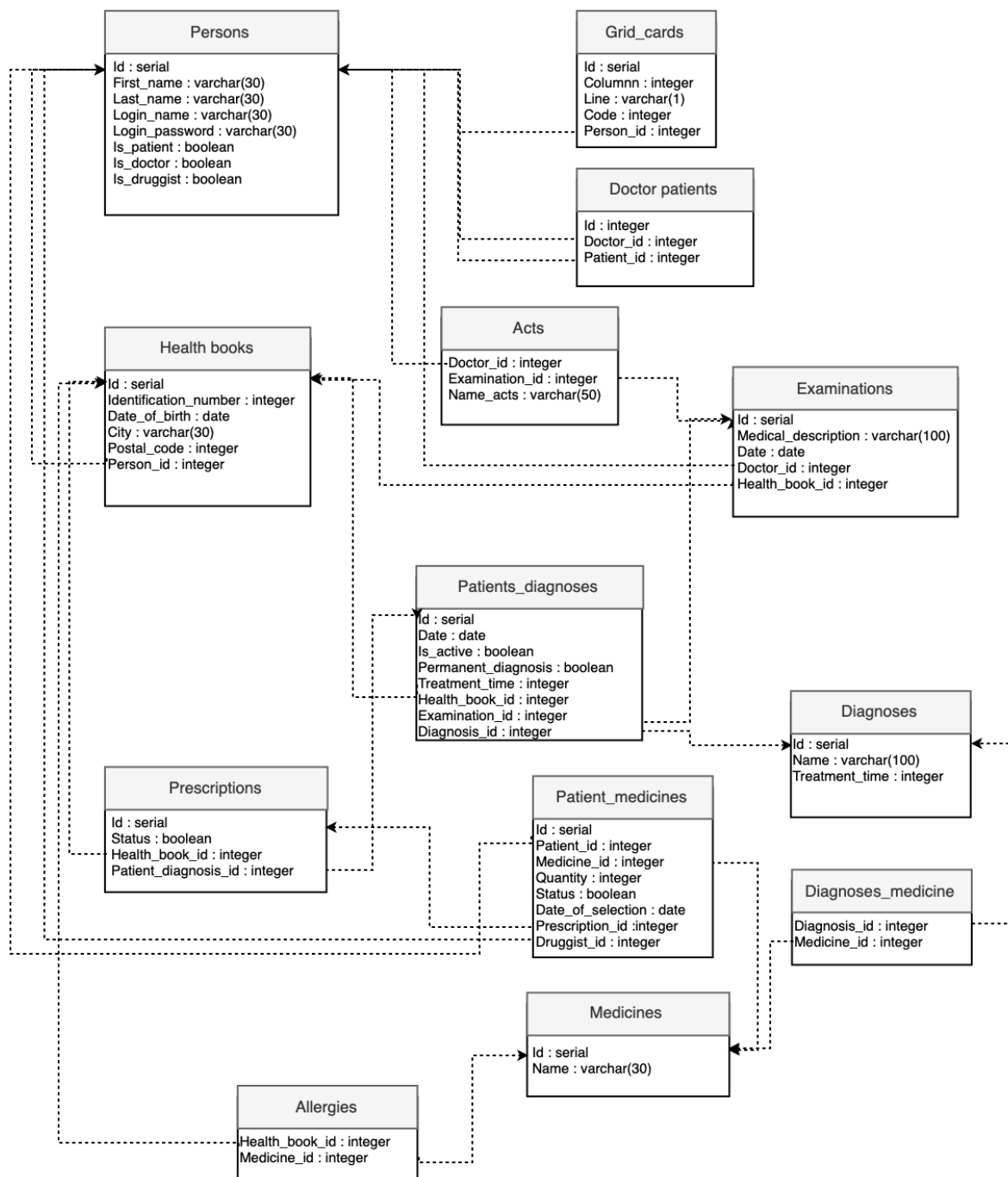
„Admin“: môže v systéme vy listovať všetky osoby, pridať osobu, ktorej musí vytvoriť zdravotnú kartu a tejto osobe sa automaticky vytvorí grid karta, každý v systéme má unikátnu grid kartu s číselnými kódmi. Ďalej môže osoby editovať, vyhľadávať osoby podľa mena a priezviska a zároveň aj odstrániť osobu zo systému. Admin má funkciu na prehľad štatistík lekárskeho výkonu pre každého lekára v jednotlivých mesiacoch a štatistiky najčastejšie predpisovaných troch liekov za štvrtrok pre každú diagnózu. Admin obsahuje funkciu, ktorá sa spúšťa každý deň o 00:00 a slúži na kontrolu diagnóz, ak sú po termíne doby liečenia, tak sa nastaví na už ukončené.

„Pacient“: môže si zobraziť svoj zdravotný záznam, upraviť si svoje osobné údaje v zdravotnom zázname. Skontrolovať stav alergií poprípade si pridať alergiu na konkrétne lieky alebo zmazať alergiu zo svojej zdravotnej knižky. Pacient si môže zobraziť svoje aktuálne vyšetrenia a aktívne diagnózy, ale aj ich históriu a taktiež svoje recepty.

„Doktor“: obsahuje funkciu na vy listovanie všetkých doktorových pacientov, zobrazenie vykonaných lekárskeho výkonu, vyšetrení a všetky aktívne diagnózy jeho pacientov. Doktor si môže pridať pacienta, ale pacient sa musí pri prvej návšteve u lekára autorizovať a po autorizovaní sa doktorovi zobrazí zdravotná karta pacienta. Lekár vykonáva vyšetrenie pre pacienta počas, ktorého môže vykonávať viacero lekárskeho výkonu. Na základe vyšetrení stanovuje lekár diagnózu, ku ktorým stanovuje dobu liečenia. Lekár po stanovení diagnózy predpisuje pacientovi lieky na ktoré nie je alergický alebo ich momentálne neužíva.

„Lekárnik“: lekárnik spúšťa funkciu výber liekov, kde si zvolí pacienta, ktorý sa musí autorizovať, pri každej návšteve u lekárnika. Pacient má možnosť si vybrať iné lieky aké mal predpísané, ktoré liečia danú diagnózu, ale zároveň nie je na nich alergický. Lekárnik má možnosť vidieť pacientove aktuálne lekárske predpisy, ale aj ich históriu.

RELAČNÁ DATABÁZA



Obr. 2: Relačný model dát pre systém E-health

Výsledný obrázok zobrazuje relačnú databázu, ktorá vznikla pri transformovaní entitno-relačného modelu Obr.1.

ORGANIZÁCIA KÓDU

- *Aplikácia je naprogramovaná v jazyku Java. Zdrojový kód je rozdelený do nasledovných balíkov.

Sk.tomana.e_health.application

- *Tento balík obsahuje Main triedu, cez ktorú sa aplikácia spúšťa. Aplikácia má po celý čas vytvorené jedno spojenie na databázový systém, ktoré sa udržiava v triede DbContext.

Sk.tomana.e_health.application.AbstrackClasses

- Obsahuje abstraktné triedy ako BaseGateway, BaseFinder a iné, ktoré sú použité v aplikácií.

Sk.tomana.e_health.application.Array

- Obsahuje pomocné pole lekárskeho výkonov, z ktorých lekár vyberá a následne sú uložené do databázy k jednotlivým vyšetreniam pacientov.

Sk.tomana.e_health.application.Exceptions

- Tento balík obsahuje triedy výnimiek, ktoré sa využívajú v aplikácií.

Sk.tomana.e_health.application.rdg

- *Tento balík obsahuje pre každú tabuľku v databáze Data Gateway. Row Data Gateway je realizovaný dvojicou tried – jedna trieda je pre samotný Gateway a druhá pre Finder. Opakujúci sa kód je vyčlenený do tried BaseGateway a BaseFinder, od ktorých ostatné triedy dedia.

Sk.tomana.e_health.application.Statistics

- Tento balík obsahuje štatistiky. Štatistiky sú realizované dvojicou tried – jedna trieda je pre samotný Gateway a druhá pre Finder, ktoré dedia z abstraktných tried.

Sk.tomana.e_health.application.ts

- Tento balík obsahuje jednotlivé transakcie, ktoré sú použité v aplikácií.

Sk.tomana.e_health.application.ui

- *Tento balík obsahuje kód používateľského rozhrania. Jedná sa o rozhranie v ASCII grafike ovládané z príkazového riadku. Jadrom aplikácie je menu v ktorom sa môžu prihlásiť admin, pacient, lekár a lekárnik. Každý má v systéme vlastné funkcie. Opakujúci kód je vyčlenený do tried Menu, od ktorého sa všetky menu dedia. Aby sa zabezpečila nezávislosť Row Data Gateway od realizácie používateľského rozhrania, ich vypisovanie je realizované pomocou printerov.

*text označený * znakom je podobný záverečnej správe zo vzorového projektu.

OPTIMALIZÁCIA SQL

Každý sa musí v systéme autorizovať a preto som sa rozhodol použiť index práve v časti kedy sa používateľ aplikácie prihlasuje do systému dvojzložkovo a pri hľadaní kódu by mohol dlhšie čakať na overenie.

Pri veľkom množstve dát bolo nájdenie kódu z grid karty pri autentifikácii používateľa pomalé. Tento problém som vyriešil vytvorením indexu.

Rýchlosť vykonania oboch variantov pred vytvorením indexu a po vytvorení zobrazuje nasledujúca tabuľa.

Časy boli merané príkazom EXPLAIN ANALYZE.

Veľkosť testovaných dát v riadkoch	Rýchlosť pred optimalizáciou v (ms)	Rýchlosť po optimalizácii v (ms)
900 000	72.536	0.055
9 000 000	530.313	0.045

Explain analyze pred vytvorením indexu:

Gather (cost=1000.00..123950.38 rows=2 width=18) (actual time=490.111..530.278 rows=1 loops=1)

Workers Planned: 2

Workers Launched: 2

-> Parallel Seq Scan on grid_cards (cost=0.00..122950.18 rows=1 width=18) (actual time=421.255..486.191 rows=0 loops=3)

Filter: ((columnn = 2) AND ((line)::text = 'A'::text) AND (person_id = 10200))

Rows Removed by Filter: 3000000

Planning time: 0.080 ms

Execution time: 530.313 ms

Explain analyze po vytvorení indexu:

Index Scan using grid_index on grid_cards (cost=0.56..12.34 rows=2 width=18) (actual time=0.023..0.025 rows=1 loops=1)

Index Cond: ((columnn = 2) AND ((line)::text = 'A'::text) AND (person_id = 10200))

Planning time: 0.085 ms

Execution time: 0.045 ms

VÝBRANÝ RIEŠENÝ PROBLÉM

Pôvodne som v dátovom modeli neuvažoval ako zabezpečím doktorových pacientov, kde si budem pamätať jednotlivých pacientov.

- Riešením môjho problému bolo zaviesť do systému tabuľku doktorových pacientov, ktorý obsahuje id doktora a id pacienta. Teraz už viem akých pacientov má doktor. Ak doktor nemá pacienta, tak sa musí pacient pri prvej návšteve doktora autorizovať a následne sa pridá do tabuľky. Doktor získa prístup k pacientovej zdravotnej knižke a môže vykonávať vyšetrenia.