



Slovenská technická univerzita v Bratislave  
FAKULTA INFORMATIKY A INFORMAČNÝCH TECHNOLOGIÍ

## **Elektronická prihláška na štúdium - vyhodnocovanie prijímacej skúšky**

---

Tím číslo 7 HYDRA

Vedúci tímového projektu: Ing. Jaroslav Kuruc

December 2004

Bc. Daniel Brnák

Bc. Michal Holečka

Bc. Peter Mihalik

Bc. Juraj Pavlovič

Bc. Michal Petrov

Bc. Peter Ružička

## Zadanie projektu

Pre prijímanie študentov na akademický rok 2004/2005 použila naša fakulta prvýkrát systém elektronickej prihlášky. Ide o web-aplikáciu, ktorej cieľom je zefektívniť komunikáciu medzi uchádzačom a fakultou, ako aj poskytnúť uchádzačovi dôležitú spätnú väzbu. Uchádzači si tak prostredníctvom webu podávajú prihlášky na štúdium, ktoré sú potom ďalej spracovávané.

Súčasťou prijímacieho konania je vyhodnotenie písomnej prijímacej skúšky, ktoré je vzhľadom na veľký počet uchádzačov vhodné čo najviac zautomatizovať. Analyzujte súčasný spôsob vstupu a vyhodnocovania testov. Cieľom projektu je vytvoriť efektívny, pokiaľ možno čo najviac automatizovaný systém spracovania testových odpovedí. Pri jeho návrhu zohľadnite tieto ohraničenia:

- obmedzený čas na spracovanie testov,
- dosiahnutie akceptovateľnej chybovosti pri spracovaní testov,
- dostupné ľudské zdroje,
- dostupné technické prostriedky.

Vytvorený systém by mal spolupracovať s existujúcim systémom elektronickej prihlášky.

# Úvod

Študenti v rámci inžinierskeho štúdia na Fakulte informatiky a informačných technológií STU v Bratislave absolvujú aj predmety Tímový projekt I a Tímový projekt II.

Hlavnou náplňou týchto predmetov je práca študentov v tímoch na zadanej úlohe. Úlohou tímov je podrobne špecifikovať, navrhnuť a implementovať softvérový systém, ktorý bude nasadený v praxi a tak využívaný v reálnom prostredí. Navyiac si študenti vyskúšajú prácu vo väčšej skupine ľudí (4-6 členov) a zabezpečujú aj proces samotného riadenia projektu.

Predkladaný dokument je dokumentáciou k riešeniu projektu v rámci tímového projektu v školskom roku 2004/2005 tímom 7 - Hydra. Našou úlohou bolo riešenie úlohy elektronickej prihlášky na štúdium, konkrétne vyhodnocovanie prijímacej skúšky.

Dokumentácia je tvorená dvoma samostatnými časťami. Prvá sa zaoberá samotným výsledkom našej práce - predkladaným produktom. Ide teda o technickú dokumentáciu vytváraného informačného systému. V druhej je zaznamenaná práca členov tímu na riadení projektu a tímu ako celku.

**Elektronická prihláška na štúdium - vyhodnocovanie  
prijímacej skúšky**

Dokumentácia k softvérovému systému



## Obsah

<b>1 Úvod</b>	<b>iv</b>
1.1 Prehľad dokumentu . . . . .	iv
1.2 Použité skratky . . . . .	v
1.3 Použitá notácia . . . . .	vi
<b>2 Analýza problému</b>	<b>1</b>
2.1 Analýza súčasného stavu . . . . .	1
2.2 Ciele produktu . . . . .	2
2.3 Prehľad produktu . . . . .	3
2.3.1 Identifikácia uchádzača a testu v procese prijímacej skúšky . . . . .	3
2.3.2 Zadávanie šablón správnych odpovedí . . . . .	3
2.3.3 Manuálne vyhodnocovanie testov . . . . .	4
2.3.4 Automatizované vyhodnocovanie testov . . . . .	5
2.3.5 Zadávanie osobných údajov . . . . .	6
2.3.6 Vyhodnotenie zadaných testov . . . . .	6
2.3.7 Import a export údajov . . . . .	7
2.3.8 Tvorba výstupov a prehľad spracovania . . . . .	7
2.4 Analýza existujúcich systémov . . . . .	7
2.4.1 Komplexné systémy . . . . .	7
2.4.2 Hardwarové nástroje . . . . .	8
2.4.3 Softvérové nástroje . . . . .	11
<b>3 Špecifikácia požiadaviek</b>	<b>13</b>
3.1 Diagram prípadov použitia . . . . .	13
3.1.1 Roly používateľov . . . . .	15
3.1.2 Prípady použitia . . . . .	16
3.2 Kontrola vstupných údajov . . . . .	34
3.3 Nefunkcionálne požiadavky na systém . . . . .	35



3.3.1	Predpokladané ľudské zdroje . . . . .	36
3.3.2	Technické požiadavky . . . . .	36
3.3.3	Automatizované spracovanie testov . . . . .	37
3.3.4	Súhrn požiadaviek . . . . .	37
3.3.5	Súčasný stav . . . . .	37
3.3.6	Zhodnotenie . . . . .	38
<b>4</b>	<b>Hrubý návrh</b>	<b>39</b>
4.1	Konceptuálny diagram . . . . .	39
4.2	Model údajov . . . . .	40
4.2.1	Diagram modelu údajov ( logická úroveň ) . . . . .	41
4.2.2	Entity logického modelu údajov . . . . .	41
4.3	Diagram nasadenia . . . . .	43
<b>5</b>	<b>Prototyp</b>	<b>46</b>
5.1	Oblasť prototypovania . . . . .	46
5.2	Výber prostredia . . . . .	47
5.3	Prototyp používateľského rozhrania a dátového modelu . . . . .	48
5.3.1	Ciele prototypovania . . . . .	48
5.3.2	Spôsob prototypovania . . . . .	48
5.3.3	Výsledok prototypu . . . . .	50
5.4	Prototyp rozpoznávania čiarových kódov . . . . .	51
5.4.1	Ciele prototypovania . . . . .	51
5.4.2	Spôsob prototypovania . . . . .	51
5.4.3	Výsledok prototypu . . . . .	52
5.5	Prototyp rozpoznávania testových odpovedí . . . . .	53
5.5.1	Ciele prototypovania . . . . .	53
5.5.2	Spôsob prototypovania . . . . .	53
5.5.3	Výsledok prototypu . . . . .	53
5.6	Prototyp autentifikácie používateľov . . . . .	54
5.6.1	Ciele prototypovania . . . . .	54
5.6.2	Spôsob prototypovania . . . . .	54



5.6.3	Výsledok prototypu . . . . .	55
5.7	Zhodnotenie prototypovania . . . . .	56
<b>6</b>	<b>Používateľská príručka (k prototypu)</b>	<b>57</b>
6.1	Používateľské rozhranie . . . . .	57
6.2	Menu EAPP-EVAL . . . . .	58
6.2.1	Prihlásenie do aplikácie . . . . .	58
6.2.2	Odhlásenie . . . . .	59
6.3	Menu šablóny . . . . .	59
6.3.1	Pridanie šablóny . . . . .	59
6.3.2	Výber šablóny . . . . .	61
6.4	Menu Testy . . . . .	63
6.4.1	Zadávanie testov . . . . .	63
6.5	Menu Uchádzač . . . . .	64
6.5.1	Zadat' osobné údaje . . . . .	64
6.5.2	Zadat' vysvedčenia . . . . .	66
6.6	Menu Štatistiky . . . . .	68
6.6.1	Osobné údaje . . . . .	68
6.6.2	Prospech . . . . .	69
6.6.3	Výsledky testov . . . . .	70
6.7	Menu Skener . . . . .	71
6.8	Rozpoznanie dokumentov . . . . .	71
6.8.1	Riešenie konfliktov . . . . .	72
6.9	Menu Vyhodnotenie . . . . .	74



# 1 Úvod

Prijímacie konanie na vysokú školu je dôležitým a náročným procesom tak pre samotných uchádzačov ako aj pre pracovníkov fakulty a vysokej školy, ktorá prijímacie konanie organizuje. Vzhľadom na veľké množstvo uchádzačov je organizácia prijímacieho konania pre jednotlivé fakulty veľmi náročným organizačným úkonom. Zahŕňa úkony od podania prihlášky, cez samotné prijímacie testy až po ich vyhodnotenie a informovanie uchádzačov. Pre prijímanie študentov na akademický rok 2004/2005 použila FIIT STU prvýkrát systém elektronickej prihlášky. Súčasťou prijímacieho konania je vyhodnotenie písomnej prijímacej skúšky. Vzhľadom na obmedzený čas potrebný na vyhodnotenie prijímacích testov a spracovanie osobných údajov uchádzačov je potrebné tento krok podporiť zo strany informačného systému. Tento dokument sa zaoberá analýzou, špecifikáciou a návrhom systému pre vyhodnocovanie prijímacej skúšky.

## 1.1 Prehľad dokumentu

Analýza problému je popísaná v kapitole 2. Opisom problémového prostredia a procesom vyhodnocovania prijímacej skúšky tak ako v súčasnosti prebieha na FIIT STU sa zaoberá kapitola 2.1. Základné problémy a ciele produktu sú identifikované v kapitole 2.2. V kapitole 2.3 sú bližšie analyzované problémové oblasti spolu s krátkym návrhom možnosti riešenia. Analýza existujúcich riešení je zhrnutá v kapitole 2.4.

V kapitole 3 sa nachádza podrobná špecifikácia vytváraného informačného systému. Pozostáva z identifikovaných funkcionálnych požiadaviek kladených na systém v kapitole 3.1 v podobe diagramu prípadov použitia. Ďalej sú identifikované a charakterizované roly používateľov v systéme. Kapitola tiež zahŕňa podrobný opis prípadov použitia modelovanými diagramami aktivít. V kapitole 3.3 sú prehľadne uvedené nefunkcionálne požiadavky na systém.

Kapitola 4 je zameraná na hrubý návrh systému. V prvej časti sa nachádza konceptuálny diagram s jeho opisom. V kapitole 4.2 je uvedený logický model údajov. Diagram nasadenia zobrazujúci rozmiestnenie systému na uzly je uvedený v kapitole 4.3.

Kapitola 5 je určená opisu prototypovania. Opísaná je oblasť prototypovania, výber prostre-





dia a bližšie sú charakterizované ciele, spôsob a výsledky vyvíjaných prototypov. V kapitole 6 je uvedená používateľská príručka.

## 1.2 Použité skratky

Táto kapitola obsahuje vysvetlenie skratiek použitých v tomto dokumente.

ADF - automaticky podávač dokumentov

CVS - systém na správu verzií (Concurrent Versions System)

DPH - daň z pridanej hodnoty

DPI - jednotka rozlíšenie, určuje počet bodov na palec (Dots Per Inch)

FIIT - Fakulta informatiky a informačných technológií

GB - jednotka určujúca množstvo dát - giga bajty

GNU GPL - typ softvérovej licencie (GNU General Public License)

ID - identifikátor

JAVA, C, C++ - programovacie jazyky

LDAP - množina protokol na prístup k informačným adresárom (Lightweight Directory Access Protocol)

OCR - nástroj na konvertovanie vytlačeného textu do súborov spracovateľných počítačom (Optical Character Recognition)

PC - osobný počítač (Personal Computer)

PDF - typ dokumentov (Portable Document Format)

PostgreSQL - voľne dostupný databázový softvér

SK - Slovenská koruna

SR - Slovenská republika

STU - Slovenská technická univerzita v Bratislave

STUDAPP - systém elektronickej prihlášky

TCP/IP - sieťový protokol na riadenie prenosu (Transmission Control Protocol / Internet Protocol)

TIF, BMP, JPG, GIF - rôzne formáty súborov určených na uchovanie grafických obrázkov

TWAIN - platformovo nezávisle rozhranie na získavanie obrázkov zo skeneru (Technology without an interesting name)

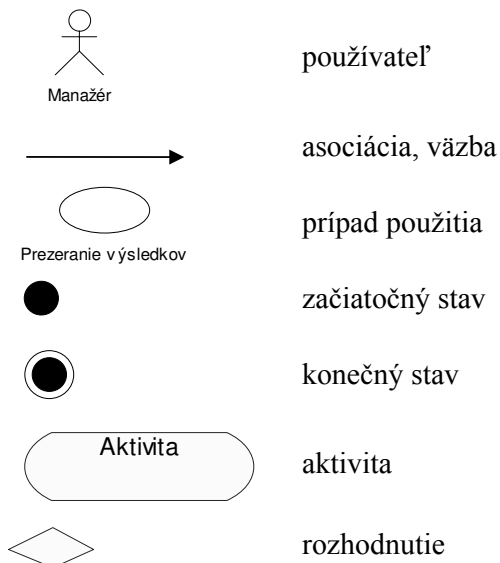


USB - počítačové rozhranie na pripojenie periférnych zariadení (Universal Serial Bus)

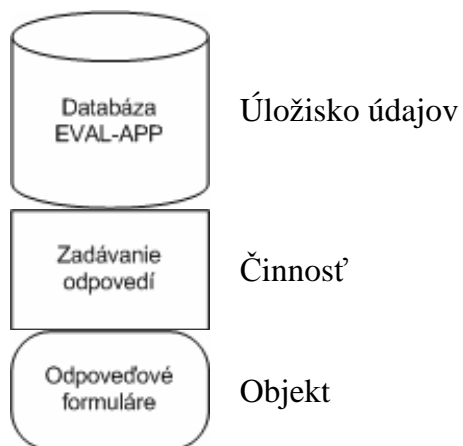
### 1.3 Použitá notácia

Opis notácie použitej pri vytváraní diagramov uvedených v dokumente.

#### Diagram prípadov použitia a aktivít



#### Konceptuálny diagram



#### Diagram logického modelu údajov

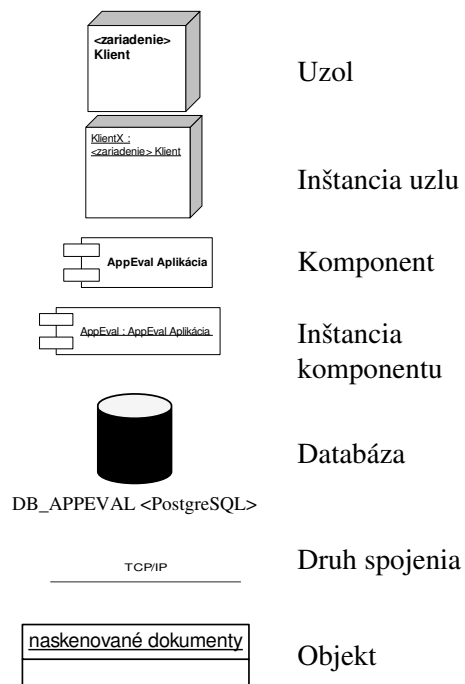


Vzor testu
predmet
miestnosť
čas

údajová entita

asociácia medzi entitami

## Diagram nasadenia





## 2 Analýza problému

Táto kapitola sa zaoberá analýzou súčasného stavu spôsobu vyhodnocovania prijímacej skúšky na FIIT STU. Taktiež sú tu analyzované niektoré v praxi používané systémy vyhodnocovania informácií a prostriedky na to použité.

### 2.1 Analýza súčasného stavu

Pre prijímanie študentov na akademický rok 2004/2005 použila FIIT STU prvýkrát systém elektronickej prihlášky. Ide o web-aplikáciu, ktorej cieľom je zefektívniť komunikáciu medzi uchádzačom a fakultou, ako aj poskytnúť uchádzačovi dôležitú spätnú väzbu. Uchádzači si tak prostredníctvom webu podávajú prihlášky na štúdium, ktoré sú potom ďalej spracovávané.

Súčasťou prijímacieho konania je vyhodnotenie písomnej prijímacej skúšky. Záverečný krok vyhodnocovania odpovedí uchádzačov však veľmi úzko súvisí s celým procesom konania prijímacej skúšky, preto je vhodné priblížiť si celú postupnosť krokov od podania prihlášky po jej vyhodnotenie tak ako sa v súčasnosti vykonáva v rámci prijímacieho konania na FIIT STU.

Uchádzač o štúdium, sa zaregistruje cez webovské rozhranie v systéme elektronickej prihlášky STUDAPP. Systém mu umožňuje podať jednu prihlášku na jednu vysokú školu. Keďže v súčasnosti funguje tento systém len pre Fakultu informatiky a informačných technológií STU jeho výber je obmedzený len ponukou študijných odborov. V systéme sa kontrolujú údaje, ktoré zadal v prihláške a potvrdené sa exportujú do dokumentu formátu PDF. Potvrdené ich pošle spolu s ostatnými požadovanými dokumentami (potvrdenie od lekára, kópia vysvedčení zo strednej školy, atd.) na adresu fakulty. Po prijatí a kontrole potrebných dokumentov je mu zaslaná odozva s jeho registračným číslom v systéme v podobe emailu.

V požadovanom termíne sa študent dostaví na prijímaciu skúšku. Pri registrácii obdrží formulár na vyplnenie a kontrolu osobných údajov, zadanie otázok na prijímaciu skúšku a hárok s predtlačným formulárom na vyplnenie odpovedí. Oba formuláre majú predtlačené náhodne generované jedinečné čísla, ktorými sú študent a jeho hárok s odpoveďami identifikovaní. Existuje viac alternatív zadaní prijímacích skúšok, pričom číslo alternatívy sa vpisuje na há-



rok s odpoveďami. Počas prijímacej skúšky si uchádzač skontroluje osobné údaje, potvrdí ich správnosť prípadne vykoná požadované úpravy v predznačenom formulári. Formulár s osobnými údajmi odovzdá späť už pri registrácii a informácie na nich sa môžu spracovávať už počas priebehu samotnej skúšky.

Po absolvovaní skúšky odovzdá uchádzač vyplnené formuláre, ktoré sa zozbierajú a rozdelia na vyhodnotenie. Pri vyhodnocovaní sa hárky s odpoveďami porovnávajú s príslušnými šablónami a zapisuje sa počet správnych odpovedí do súboru aplikácie Excel spolu s identifikačným číslom uvedeným na hárku. Takáto kontrola prebieha opakovane iným zamestnancom povereným vyhodnocovaním prijímacej skúšky. Vyhodnotenia zapísané v súboroch Excel sa zozbierajú do systému kontroly verzií CVS a výsledky sa porovnávajú. Prípadné rozdiely vo výsledkoch prijímacej skúšky pre jedného uchádzača z dvoch zdrojov opravy sa opätovne prehodnocujú. Takto revidované výsledky sa exportujú do databázy aplikácie STUDAPP kde sa podľa identifikačného čísla uvedeného na hárku s odpoveďami priradia k príslušnému študentovi.

## 2.2 Ciele produktu

Analýzou súčasného stavu a problémového prostredia vyplynulo niekoľko problémov, ktoré bude mať vyvíjaný systém za úlohu odstrániť alebo aspoň v čo najväčšej možnej miere minimalizovať ich dopad na proces vyhodnocovania prijímacej skúšky.

### Identifikované problémy:

- výskyt subjektívnych chýb (pri vyhodnocovaní testov pomocou šablón, pri zadávaní údajov do systému)
- zdĺhavý proces odstraňovania chýb
- náročnosť na ľudské zdroje

### Cieľ zlepšenia:

- čo najväčšou možnou mierou automatizovať systém vyhodnocovania prijímacej skúšky
- znížiť náročnosť na ľudské zdroje pri vyhodnocovaní prijímacej skúšky
- odstrániť chybovosť v procese vyhodnocovania prijímacej skúšky



## 2.3 Prehľad produktu

Kapitola poskytuje stručný prehľad problémových oblastí spolu s analýzou možných riešení v navrhovanom systéme.

### 2.3.1 Identifikácia uchádzača a testu v procese prijímacej skúšky

Pri prijímacej skúške musí byť v čo najväčšej miere zabezpečená anonymita uchádzača vzhľadom na osobu(y), ktorá vyhodnocuje jeho odpovede zo skúšky. V súčasnosti je systém identifikácie zabezpečený náhodne generovaným číslom uvedeným na formulári s testovými odpoveďami a na formulári s osobnými údajmi. Tento identifikátor testu je evidovaný v systéme STUDAPP a nie je uvedené na žiadnom inom dokumente. Formulár s osobnými údajmi a testové odpovede sa spracovávajú oddelene, čím sa zaručuje anonymita pri vyhodnocovaní skúšky.

Druhým bezpečnostným prvkom je opakované vyhodnocovanie testov rôznymi osobami. Tým sa čiastočne eliminuje nebezpečenstvo zámernej manipulácie s výsledkami ako aj prípadná chyba pri zadávaní výsledkov do systému.

Systém identifikácie uchádzačov je jednou z kľúčových otázok pri riešení systému a preto mu bola pri analýze venovaná veľká pozornosť. Navrhnutých bolo niekoľko alternatív identifikácie uchádzača a testu, tak aby bolo pri zachovaní anonymity umožnené automatizované spracovanie výsledkov testov pomocou skenovacieho zariadenia. Jednotlivé návrhy boli diskutované zákazníkom a zhodnotením výhod a možných úskalí bolo vyšpecifikované najvýhodnejšie riešenie. Stručný prehľad analýzy problému spolu s navrhnutým výsledným riešením sa nachádza v Prílohe A.

### 2.3.2 Zadávanie šablón správnych odpovedí

Do systému musia byť jednoduchým spôsobom vložené šablóny správnych odpovedí pre všetky sady testov, slúžiace na kontrolu odpovedí testov uchádzačov. Systém na príkaz používateľa vyhodnocuje správnosť odpovedí testov podľa čísla sady a pridelí body. Získané údaje zapisuje do príslušnej tabuľky. Zadávanie šablón do systému by malo byť umožnené len obmedzenej skupine používateľov, čím by sa obmedzil prístup k správnym odpovediam a teda



znížilo riziko ich prípadného zneužitia.

Šablóny správnych odpovedí musia byť vložené do systému pred začiatkom vyhodnocovania testov, teda predtým ako začnú zamestnanci zadávať odpovede uchádzačov do systému. Pri zadávaní šablón správnych odpovedí do systému musia byť špecifikované parametre jednotlivých sád testov. Tieto parametre budú zahŕňať najmä:

- číslo sady testu
- predmet
- počet otázok
- počet možných odpovedí pre každú otázku
- typ otázok (jedna odpoveď/viac odpovedí)
- správne odpovede
- systém bodovania odpovedí

### 2.3.3 Manuálne vyhodnocovanie testov

Cieľom systému je dosiahnuť čo možno najväčšiu pohodlnosť a rýchlosť zadávania odpovedí z testov odovzdaných uchádzačmi. Systém musí zabezpečovať také používateľské rozhranie, ktoré umožňuje zadávať odpovede z testu uchádzača čo najrýchlejšie pričom systém by sám, na príkaz oprávneného používateľa, vyhodnocoval podľa zadaného čísla šablóny testu správnosť odpovedí a prideloval by body na základe pravidiel určených v šablónach testov.

Po odovzdaní vypracovaných testov študentami sa odpoveďové hárky rozdelia niekoľkým zadávateľom. Tieto skupiny hárkov sa označia tak aby sa nestalo, že jeden zadávateľ zadáva rovnakú skupinu hárkov viackrát. Zadávatel spracováva postupne jednotlivé odpoveďové formuláre. Pre každý odovzdaný formulár zadáva do systému jeho číslo, predmet, číslo sady. Podľa identifikácie sady sa zobrazí príslušný počet polí pre odpovede. Zadávatel'ovi by malo byť čo najjednoduchším a najrýchlejším spôsobom umožnené zadať odpovede z formuláru do systému. Po zadaní všetkých odpovedí sú údaje kontrolované s údajmi, ktoré už mohli byť do systému vložené predchádzajúcim zadávateľom. Ak sa podobný prípad vyskytne musí byť používateľovi umožnené opraviť prípadnú chybu.



Výhodou navrhovaného postupu je urýchlenie procesu zadávania údajov. Systém sám vygeneruje vstupné polia na obrazovke pre zadávanie odpovedí teda zadávateľ sa môže sústrediť len na zadávanie odpovedí čím sa výrazne eliminuje možnosť chyby. Do systému vkladá len odpovede a nepracuje so žiadnou fyzickou šablónou správnych odpovedí, dokonca nemá prístup ku správnym odpovediam ani v elektronickej podobe v systéme. Tým sa podstatne zvyšuje bezpečnosť celého procesu vyhodnocovania oproti súčasnému riešeniu s použitím fyzických šablón. Správne odpovede do systému má možnosť zadať len obmedzená skupina používateľov s vyššími právomocami.

### 2.3.4 Automatizované vyhodnocovanie testov

Systém by mal poskytovať riešenie pre automatizované vyhodnocovanie testov pomocou skenera. Táto časť systému bude implementovaná s nižšou prioritou ako manuálne zadávanie výsledkov testu, ktoré je pre fungovanie systému nevyhnutné.

Od systému sa očakáva identifikácia čísla testu, predmetu, čísla sady a vyplnených odpovedí uchádzačom. Takéto požiadavky by mali zabezpečiť moduly systému pre skenovanie a rozpoznávanie, pričom postup by mohol byť nasledovný, rozdelený do dvoch fáz:

**Skenovanie dokumentov:** Všetky odovzdané formuláre s odpoveďami je potrebné spracovať a vložiť do systému. Vložia sa do podávača skenera. Vyplnené hárky s čiarovým kódom by sa vložili do skenera a naskenovali v postačujúcom rozlíšení.

**Vyhodnocovanie dokumentov:** Takto uložené v systéme by sa postupne vyhodnocovali, pričom sa roznáva čiarový kód (resp. číslo uchádzača na prijímacej skúške), predmet a číslo sady testu (šablóny), odpovede uchádzača, pričom sú evidované dobré aj zlé odpovede. V prípade, že systém by nedokázal postačujúcou mierou istoty vyhodnotiť odpoveď prípadne celý test systém musí umožňovať manuálny spôsob opravy pri náhlade na naskenovaný dokument. Je náročné zabezpečiť rozpoznávanie textu a preto by číslo testu mohlo byť zakomponované v čiarovom kóde, prípadne na vyznačenie sady testov by mohlo byť použité zaškrtávacie pole na hárku s odpoveďami.





### 2.3.5 Zadávanie osobných údajov

Popri zadávaní odpovedí z vypracovaných testov je potrebné v procese prijímacej skúšky evidovať aj ďalšie údaje o študentoch. Systém by mal podporovať zadávanie minimálne dvoch typov údajov, ktoré zákazník definoval ako najdôležitejšie:

- kontaktné telefónne číslo mobilného telefónu, kam bude uchádzačovi odoslaný výsledok prijímacej skúšky. Nachádza sa na formulári s osobnými údajmi, ktorý študent odovzdal pri registrácii
- priemery predmetov za jednotlivé školské roky na strednej škole. Uchádzač odovzdáva pri registrácii vysvedčenia, ktoré ešte nie sú evidované v systéme

Prípadne by mohli byť doplnené ďalšie typy údajov, ktoré sú potrebné pri vyhodnocovaní prijímacej skúšky.

Systém musí umožňovať čo najrýchlejšie spracovanie týchto údajov, keďže sú nevyhnutne potrebné k vyhodnoteniu prijímacej skúšky. Údaje sa nachádzajú na rôznych dokumentoch, bolo by teda výhodné zadávať ich osobitne. V prvom prípade by sa zadávala dvojica registračné číslo uchádzača - telefónne číslo. V druhom prípade vzhľadom na to, že na vysvedčení nie je uvedené registračné číslo uchádzača v systéme STUDAPP, uchádzač musí byť identifikovaný rodným číslom. Zadávalo by sa teda rodné číslo a n-tica známkov, ktoré sú zapísané na vysvedčení. Údaje budú po zadaní uchovávané na databázovom serveri a na príkaz používateľa exportované do systému STUDAPP.

### 2.3.6 Vyhodnotenie zadaných testov

Systém musí zabezpečovať vyhodnotenie odpovedí zadaných testov. Po zadaní príkazu používateľom systém začína porovnávať zadané testy a so šablónami správnych odpovedí pre jednotlivé sady testov. Pre každú otázku vyhodnocuje úplnú alebo čiastočnú správnosť odpovede (v prípade otázok s viacerými možnými odpoveďami) a prideluje body na základe pravidiel definovaných v šablóne. Takto vyhodnotenú testy zapisuje do príslušných výsledných tabuliek tak, aby bolo možné určiť jeho správne a nesprávne odpovedané otázky ako aj výsledný počet bodov za absolvovaný test.



### 2.3.7 Import a export údajov

System bude v procese vyhodnocovania prijímacej skúšky úzko spolupracovať s niekoľkými ďalšími systémami. Je teda nevyhnutnosťou zabezpečiť import a export údajov. Príkladom je import inicializačných údajov (registračné čísla, čísla testov, predmety prijímacích skúšok, osobné údaje študentov) do nami navrhovaného systému zo systému STUDAPP. Podobne bude prebiehať export výsledkov vyhodnotených testov z databázy do systému STUDAPP.

### 2.3.8 Tvorba výstupov a prehľad spracovania

System by mohol umožňovať tvorbu výstupov o vyhodnocovaní testov. Príkladom by mohol byť dokument o štruktúre odpovedí uchádzača na testoch, výsledku vyhodnotenia testov a bodového ohodnotenia jednotlivých častí prijímacieho konania. Uchádzač má možnosť žiadať tieto informácie po zverejnení výsledkov prijímacích skúšok. Dokument by mohol byť generovaný systémom a vystavený zamestnancom fakulty pre uchádzača na požiadanie.

System by mal byť v každom okamihu spracovania testov vyhodnotiť stav spracovania. Pre privilegovaného používateľa by teda mal byť schopný zobrazit', ktoré testy už boli zadané a kým, prípadne pre ktorých používateľov nebol ešte test zadaný. Takto by používateľ získal prehľad o stave spracovania testov a mohol by vhodnými krokmi tento proces ovplyvniť.

## 2.4 Analýza existujúcich systémov

Táto kapitola obsahuje prehľad existujúcich systémov z oblasti vyhodnocovania dokumentov. Keďže podobných komplexných systémov je nie veľa a ich špecifikácia je veľmi ťažko dostupná, venuje sa táto kapitola aj systémom, ktorých činnosť predstavuje aspoň časť procesu vyhodnocovania prijímacej skúšky. Analyzujeme teda komplexné systémy vyhodnocovania dokumentov, skenovací zariadenia použiteľné pri zadanej úlohe a softvérové riešenia príp. dostupné knižnice použiteľné pri rozoznávaní formulárov.

### 2.4.1 Komplexné systémy

Jediný komplexný systém o ktorom sa nám podarilo zistiť aspoň čiastočné informácie, bol použitý pri sčítaní ľudu 2001 Štatistickým úradom SR. Konkrétne hárky spracovávala firma



Infostat, pričom samotný systém zapožičal Scanservis z Českej republiky. Ten istý systém sa v dnešnej dobe (r. 2004) používa Daňovým úradom na spracovanie formulárov pre odpočty DPH.

**Parametre systému:**

- cena viac miliónov korún
- rýchlosť skenovania okolo 500 strán za minútu (obojstranný sken)
- použitý softvér AFPSPRO

Typ scanera s podávačom a jeho rozlíšenie sa nepodarilo zistiť.

Procesy skenovania a vyhodnocovania boli nezávislé. Výstup zo skenera sa ukladal na server vo forme obrázkov, v tejto fáze neboli zaznamenané nedostatky. Program na ďalšom počítači následne analyzoval tieto obrázky.

**Proces analýzy unifikovaných formulárov:**

- čiarové kódy a krížiky boli rozpoznávané bez akýchkoľvek problémov
- väčšina zadávaných údajov bola na formulároch vo forme číselných hodnôt, chyby v rozpoznávaní vyžadovali občas zásah operátora (najprv sa analyzovali číslice 0, potom 1, atd..)
- údaje vyplnené paličkovým písmom mali najmenšiu mieru úspešnosti rozpoznania
- v prípade nejasnosti sa zobrazilo prislúchajúce políčko z obrázku a navrhnuté rozpoznané písmeno či číslica
- kalibračným znakom pri skenovaní boli okraje formulárov

**2.4.2 Hardwarové nástroje**

Preskúmané boli ponuky skenerov najvýznamnejších a hlavne u nás dostupných výrobcov HP, Canon, UMAX (Avison), EPSON, Microtek, Mustek. Analýza bola orientovaná najmä na skenery strednej triedy s podávačom papiera.

Z našej strany je po analýze odporúčaný najmä skener HP scanjet 8250/8290 prípadne Avision základný model AV600U, ktorý majú minimálnu rýchlosť skenovania 12 strán za minútu. V nasledujúcej časti sú vypísané stručné charakteristiky vybraných skenerov doplnené o krátke



opisy prevzaté zo stránok produktov.

### **HP Scanjet 8250/8290**

Skener je určený na obojstranné ploché skenovanie dokumentov rýchlosťou až 25 strán za minútu. Obsahuje automatický podávač dokumentov (ADF) s rýchlosťou 25str./min s obojstranným režimom Balík produktu obsahuje softvér na správu dokumentov. Skener podporuje rozhranie ISIS/TWAIN, Kofax sVRS. Systém ďalej podporuje rozhrania na pripojenie k PC USB Hi-Speed alebo SCSI-2.

Základné parametre:

- rýchlosť skenovania náhľadu 4 s
- bitová hĺbka 48 bitová
- rozlíšenie 4 800 dpi
- zásobník 50 strán
- rýchlosť skenovania 15 str./min
- obojstranný automatický podávač dokumentov s kapacitou 50 listov
- orientačná cena: 36 215,- Sk

Skener je odporúčaný zákazníkovi.

### **Avision AV600U**

Dokumentový obojstranný skener formátu A4. Softvér PaperCom Document Manager dodávaný ku skeneru je výkonným nástrojom na organizovanie toku dokumentov. Umožňuje vytvoriť si vlastnú adresárovú štruktúru pre ukladanie dokumentov jednotlivých kategórií, export dokumentov v štandardných formátoch (TIF, BMP, JPG, GIF ...), tlač na zvolených tlačiarňach, poprípade priame spracovanie dokumentu prostredníctvom OCR. Spolu so skenerom je tiež dodávaný profesionálny OCR software Text Bridge, ktorý plne podporuje slovenčinu. Skener podporuje TWAIN rozhranie. Balík obsahuje aj softvér na rozpoznávanie textu OCR Text Bridge a úpravy skenovaných obrázkov MGI PhotoSuite SE. Skener podporuje rozhranie pripojenia USB 1.1.

Základné parametre:

- bitová hĺbka 24 bitová
- rozlíšenie 600 x 1200 dpi



- kapacita ADF 25 strán/min
- rýchlosť 12 strán/min
- orientačná cena s DPH 23000 Kč

System je odporúčaný zákazníkovi.

### **DR 2080C - základný model**

Kompaktný farebný skener Canon ponúka zvýšenú kvalitu a vysokú rýchlosť. Skener podporuje rozhranie ISIS/TWAIN. V balíku je dodávaný obslužný program Capture Perfect, s novými funkciami, napr. dávkovým skenovaním do e-mailu alebo ukladaním obrázkov v PDF formátu s plným textovým OCR atd.

Základné parametre:

- rýchlosť skenovania 12 strán/min
- rozlíšenie 600x 600 dpi
- orientačná cena 28 000.-

System nie je odporúčaný zákazníkovi. Neobsahuje zásobník a podávač dokumentov.

### **Epson GT 15000 58 000**

GT-15000 je farebný skener A3 s vysokou rýchlosťou určený pre profesionálov. Je ideálnym riešením pre skenovanie veľkých formátov (A3 alebo dvojito A4). Skener je rýchly, digitalizácia prebieha pri rýchlosti 16 ppm a je vybavený podávačom papierov A3 s kapacitou 100 strán. Počítačové rozhranie je USB 2.0 a SCSI. GT-15000 sa vyznačuje systémom MCBF (Mean Cycles Between Failure) s 100 000 cyklami, ktoré umožňuje digitalizáciu všetkých typov dokumentov (brožúry, kótovanie, listy, faxy).

Základné parametre:

- bitová hĺbka 48 bitov
- rozlíšenie 600 dpi
- rýchlosť skenovania 16 strán/min
- voliteľný podávač papiera

System je odporúčaný zákazníkovi.



### 2.4.3 Softvérové nástroje

Táto časť sa zameriava na analýzu softvérových riešení využiteľných pri riešení projektu. Niektoré druhy softvéru dodávané ku skenerom majú možnosť vyhodnocovania skenovaných dokumentov, prípadne spracovanie formulárov, preto by bolo výhodné tieto druhy softvéru tiež preskúmať. Nie je to však možné vzhľadom na to, že tento druh softvéru je dostupný len vo voľne šíriteľných verziách, ktoré nepodporujú jeho úplnú funkcionálnosť. Plné verzie sú dostupné buď pri zakúpení skenera alebo zakúpením od predajcu pričom cena takýchto riešení sa pohybuje vo výške niekoľko desiatok tisíc korún. Takáto suma sa nám zdá nevyhovujúca a preto sa v tejto kapitole zameriame na analýzu existujúcich knižníc na rozpoznávanie textu OCR, na rozpoznávanie čiarového kódu a na prácu so skenerom prostredníctvom rozhrania TWAIN, ktoré by bolo možné využiť pri implementácii nami navrhovaného systému.

#### GOOCR

Knižnica naprogramovaná v jazyku C++ určená na rozpoznávanie textu a naskenovaných dokumentov. Táto knižnica je voľne dostupná pod licenciou GNU GPL. Nevýhodou je že nepodporuje čiarový kód. Posledná verzia 0.72 je nestabilná a knižnica sa pravdepodobne ďalej nevyvíja.

Odkaz na zdroj: <http://jocr.sourceforge.net/index.html>

#### JavaOCR

Knižnica, už ako názov napovedá je naprogramovaná v jazyku JAVA. Podporuje rozpoznávanie čiarového kódu ako aj tlačeného a čiastočne písaného textu. Je k nej dostupná komplexná dokumentácia. Knižnica však nie je voľne šíriteľná a pri nekomerčnom využití stojí približne 700 USD.

Odkaz na zdroj: <http://www.javaocr.com/>

#### TWAIN knižnice

Lepšia situácia je v oblasti knižníc určených na prácu so skenerom. Tu existuje viacero voľne dostupných knižníc v jazyku JAVA aj C++. Niektoré z nich sú: C++ Wrapper for TWAIN, Twain.NET, ActiveX scanner control.

**Čiarový kód**

Medzi voľne dostupnými knižnicami na rozpoznávanie čiarového kódu, sú len knižnice naprogramované v jazyku JAVA prípadne iných. Avšak žiadna v jazyky C/C++. Voľne dostupné knižnice v jazyky JAVA sú: JBarcode Recognition, BarBara, Bar code reader.



## 3 Špecifikácia požiadaviek

Táto kapitola obsahuje požiadavky na vytváraný informačný systém. Je rozdelená na dve časti. Prvá časť obsahuje špecifikáciu funkcií požadovaného riešenia vo forme modelu prípadov použitia. Druhá časť kapitoly ponúka pohľad na nefunkcionálne požiadavky na vytváraný systém.

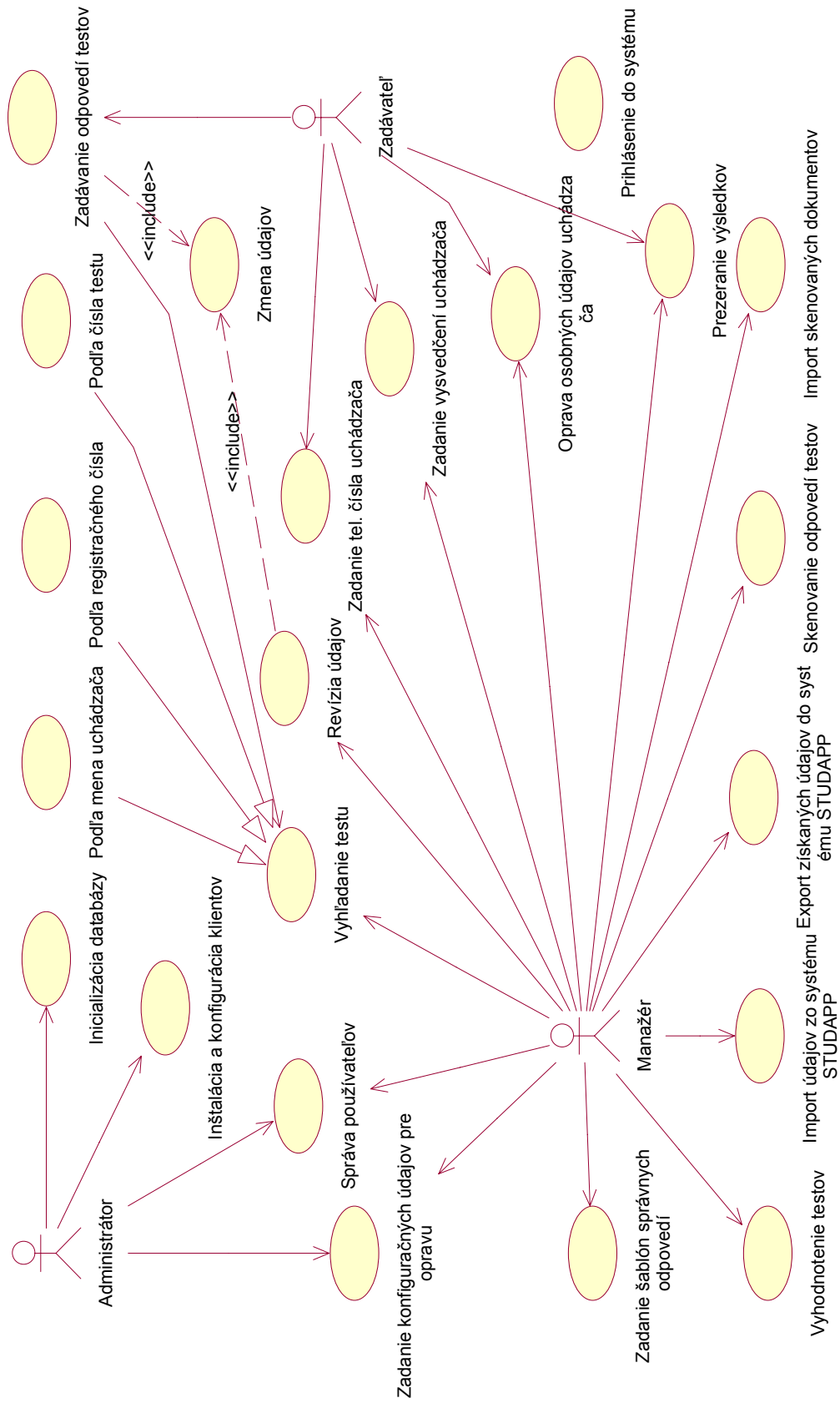
### 3.1 Diagram prípadov použitia

Na obrázku 1 je znázornený diagram prípadov použitia. Poskytuje prehľadnú informáciu o funkcionalite systému vo vzťahu k jednotlivým hráčom.

Typickí používatelia sú v diagrame prípadov použitia reprezentovaní jednotlivými hráčmi. Majú presne vymedzené postavenie v systéme. Toto postavenie obmedzuje ich prístup k niektorým častiam informačného systému. Konkrétne prípady použitia sú na obrázku znázornené oválmi, ktoré predstavujú funkcie systému. Diagram ďalej zobrazuje väzby hráčov na im prislúchajúce prípady použitia i väzby medzi prípadmi samými.

Informačný systém identifikuje jednotlivých používateľov na základe prihlasovacieho mena. Prihlásenie do systému je povinné. Na základe prihlasovacieho mena klientská aplikácia prístupní povolené časti systému pre daného používateľa. Tieto prístupové práva sme pre jednotlivých používateľov zhrnuli v tabuľke číslo 1.





Obrázok 1: Diagram prípadov použitia.



### 3.1.1 Roly používateľov

#### Zadávateľ:

Fyzická osoba, ktorej úlohou je zadávanie odpovedí jednotlivých testov manuálne alebo pomocou skenera, v prípade zistenia navzájom si odporujúcich údajov v systéme počas procesu zadávanie odpovedí má možnosť zmeniť tieto údaje. Samozrejmosťou je prezeranie výsledkov.

- vkladá do systému údaje z opravovaných testov
- môže meniť dáta iných zadávateľov ale len v tom prípade, ak práve opravuje test, ktorého výsledky sú už systéme zadaný a údaje sa nezhodujú so skutočnosťou
- opravuje odpovede, ktoré boli predtým zle zadané alebo je v nich konflikt

#### Manažér:

Fyzická osoba, reprezentujúca úzku skupinu zamestnancov, ktorý majú najväčšie právomoci v systéme. Majú možnosť zasahovať do všetkých fáz spracovávania testov, keďže môže meniť výsledky testu kedykoľvek, nielen počas zadávania.

- zadáva šablóny správnych odpovedí pre jednotlivé sady testov
- mení nastavenia parametrov opravy (počet opravujúcich)
- vkladá a nastavuje údaje o vstupných formulároch (počet otázok, počet možných odpovedí), určuje vstupné polia na formulári pri skenovaní
- riadi skenovanie formulárov do systému a ich rozpoznávanie
- zadáva príkaz na vyhodnotenie testov (porovnanie zadaných údajov so šablónami testov)
- obnova údajov zo zálohy na lokálnom disku počítača
- zadáva príkazy na import/export údajov do/zo spolupracujúcich systémov (STUDAPP)
- správa používateľov

#### Administrátor:

Fyzická osoba zodpovedná najmä za inštaláciu a konfiguráciu ako klientských aplikácií tak aj databázového serveru.

- správa používateľov
- nastavenie databázy



Informačný systém identifikuje jednotlivých používateľov na základe prihlasovacieho mena. Prihlásenie do systému je povinné. Na základe prihlasovacieho mena klienta aplikácia prístupní povolené časti systému pre daného používateľa. Tieto prístupové práva sme pre jednotlivých používateľov zhrnuli v tabuľke 1.

	Zadávateľ	Manažér	Administrátor
Prihlásenie do systému	Áno	Áno	Áno
Export získaných údajov do systému STUDAPP	Nie	Áno	Nie
Import údajov zo systému STUDAPP	Nie	Áno	Nie
Zadanie šablón správnych odpovedí	Nie	Áno	Nie
Zadanie odpovedí opravovaného testu	Áno	Áno	Nie
Zmena údajov	Áno	Áno	Nie
Skenovanie odpovedí z opravovaných testov	Áno	Áno	Nie
Import skenovaných dokumentov	Nie	Áno	Nie
Vyhodnotenie naskenovaných údajov	Nie	Áno	Nie
Správa používateľov	Nie	Áno	Áno
Inštalácia a konfigurácia klientov	Nie	Nie	Áno
Inicializácia databázového serveru	Nie	Nie	Áno
Vyhľadanie testu (podľa mena, registračného čísla, čísla testu)	Áno	Áno	Nie
Zadanie konfiguračných údajov pre opravu	Nie	Áno	Nie
Oprava osobných údajov uchádzača	Áno	Áno	Nie
Zadanie vysvedčenia uchádzača	Áno	Áno	Nie
Zadanie telefónneho čísla uchádzača	Áno	Áno	Nie

Tabuľka 1: Prístupové práva používateľov systému

### 3.1.2 Prípady použitia

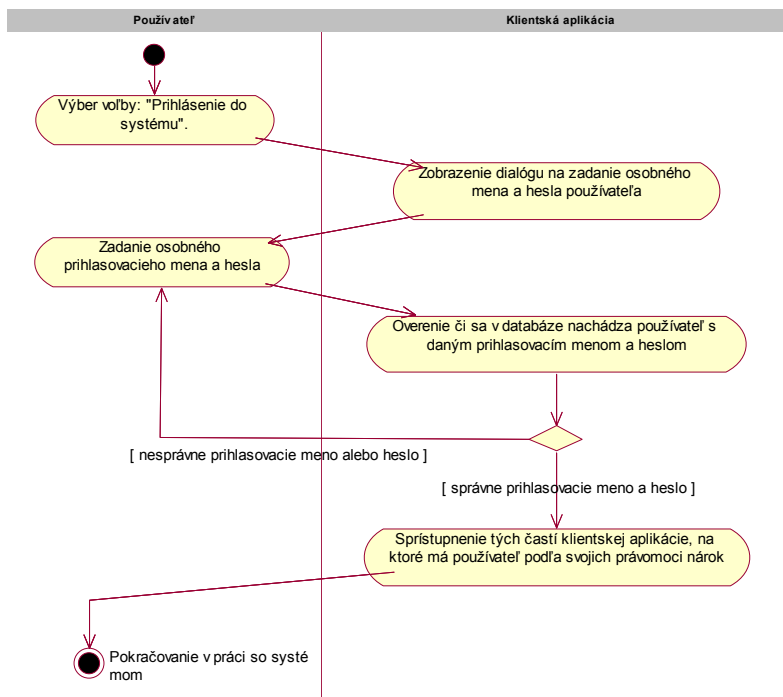
V tejto časti sú podrobne popísané všetky prípady použitia uvedené v diagrame prípadov použitia. Na názornú reprezentáciu prípadov je použitý dynamický model použitím diagramov aktivít.



## Prihlásenie do systému

Identifikátor	UC00		
Názov	Prihlásenie do systému		
Opis	Používateľ sa pred začatím práce s aplikáciou nainštalovanou na klientskom počítači musí najprv prihlásiť do systému		
Priorita	1 = vysoká	Frekvencia	stredná, vždy pred začiatkom akejkoľvek činnosti na klientovi
Vstup. podm.	Aplikácia musí byť na klientskom počítači správne nainštalovaná t. j. administrátor počas inštalácie správne nastavil údaje: adresu servera, prihlasovacie meno do databázy a heslo.		
Výstup. podm.	Používateľ zadal správne osobné prihlasovacie meno a heslo, ktoré sa nachádza v databáze		
Používatelia	Zamestnanec, zadávateľ, Manažér, administrátor		
Základná postupnosť	Krok	Činnosť	
	1	Používateľ vyberie voľbu: "Prihlásenie do systému".	
	2	Klientska aplikácia zobrazí používateľovi dialóg na zadanie jeho osobného prihlasovacieho mena a hesla.	
	3	Používateľ zadá svoje osobné prihlasovacie meno a heslo.	
	4	Klientska aplikácia overí, či sa v databáze nachádza používateľ so zadaným prihlasovacím menom a heslom.	
	5a	Prihlasovacie meno a heslo je správne. Klientska aplikácia sprístupní tie časti, na ktoré má používateľ podľa svojich právomocí nárok.	
	6a	Používateľ pokračuje v práci so systémom.	
Alternatívna postupnosť	Krok	Činnosť	
	5b	Prihlasovacie meno alebo heslo nie je správne. Návrat na krok 3.	
Poznámky		Pre prehľadnosť nie je v tomto scenári zahrnutá situácia, keby sa klientska aplikácia nemohla pripojiť na databázový server. V tomto prípade by bol používateľ znovu požiadaný o zadanie adresy servera, prihlasovacieho mena a hesla do databázy.	

Tabuľka 2: Import údajov zo systému STUDAPP

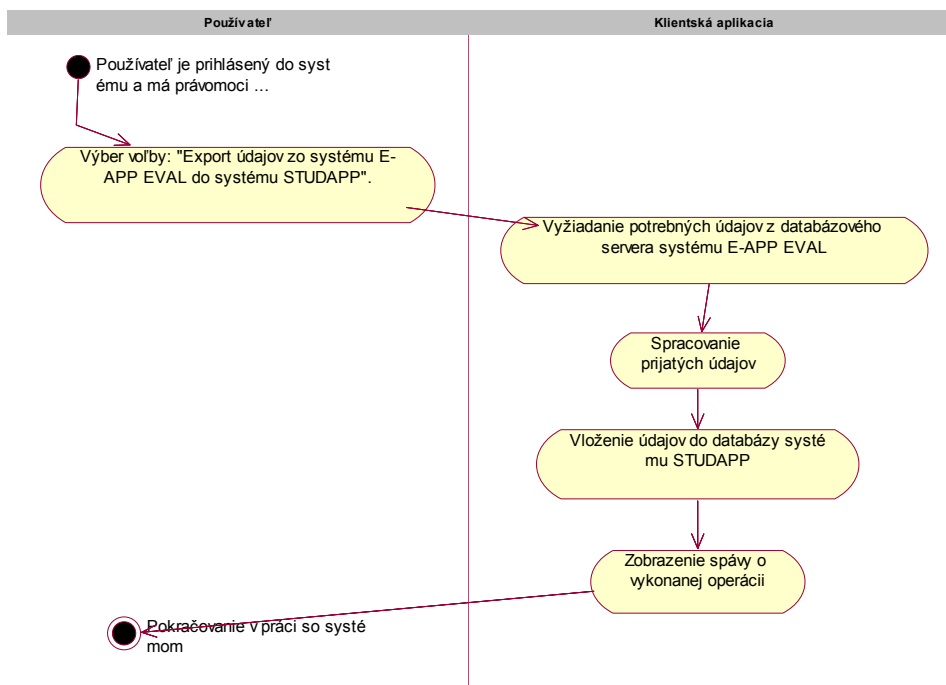


Obrázok 2: Diagram aktivít pre prihlásenie do systému

**Export získaných údajov do systému STUDAPP**

Identifikátor	UC01		
Názov	Export získaných údajov do systému STUDAPP		
Opis	Export získaných údajov a spracovaných údajov z databázového servera systému do databázového servera STUDAPP.		
Priorita	1 = vysoká	Frekvencia	nízka
Vstup. podm.	Používateľ je prihlásený v systéme a má potrebné právomoci. Klientska aplikácia pozná adresu, prihlasovacie meno a heslo pre databázový server systému ako aj pre databázový server STUDAPP. Databázový server systému už obsahuje získané a spracované údaje.		
Výstup. podm.			
Používatelia	Manažér		
Základná postupnosť	Krok	Činnosť	
	1	Používateľ vyberie voľbu: "Export údajov zo systému E-APP EVAL do systému STUDAPP".	
	2	Klientska aplikácia si vyžiada potrebné údaje z databázového servera systému E-APP EVAL.	
	3	Klientska aplikácia spracuje prijaté údaje.	
	4	Klientska aplikácia vloží spracované údaje do databázy systému STUDAPP.	
	5	Klientska aplikácia zobrazí správu o vykonanej operácii.	
	6	Používateľ pokračuje v práci so systémom.	

Tabuľka 3: Export získaných údajov do systému STUDAPP



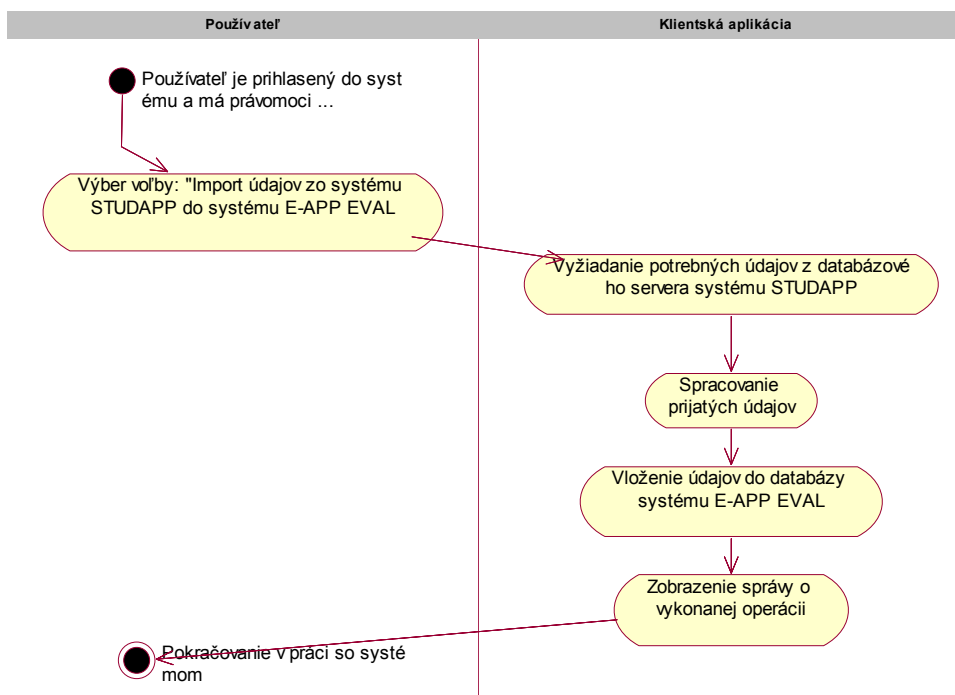
Obrázok 3: Diagram aktivít pre export získaných údajov do systému STUDAPP



### Import údajov zo systému STUDAPP

Identifikátor	UC02		
Názov	Import údajov zo systému STUDAPP		
Opis	Import údajov z databázového servera STUDAPP do databázového servera systému.		
Priorita	1 = vysoká	Frekvencia	nízka
Vstup. podm.	Používateľ je prihlásený v systéme a má potrebné právomoci. Klientska aplikácia pozná adresu, prihlasovacie meno a heslo pre databázový server systému ako aj pre databázový server STUDAPP. Databázový server STUDAPP obsahuje požadované údaje.		
Výstup. podm.			
Používatelia	Manažér		
Základná postupnosť	Krok	Činnosť	
	1	Používateľ vyberie voľbu: "Import údajov zo systému STUDAPP do systému E-APP EVAL".	
	2	Klientska aplikácia si vyžiada potrebné údaje z databázového servera systému STUDAPP.	
	3	Klientska aplikácia spracuje prijaté údaje	
	4	Klientska aplikácia vloží spracované údaje do databázy systému E-APP EVAL.	
	5	Klientska aplikácia zobrazí správu o vykonanej operácii.	
	6	Používateľ pokračuje v práci so systémom.	

Tabuľka 4: Import údajov zo systému STUDAPP

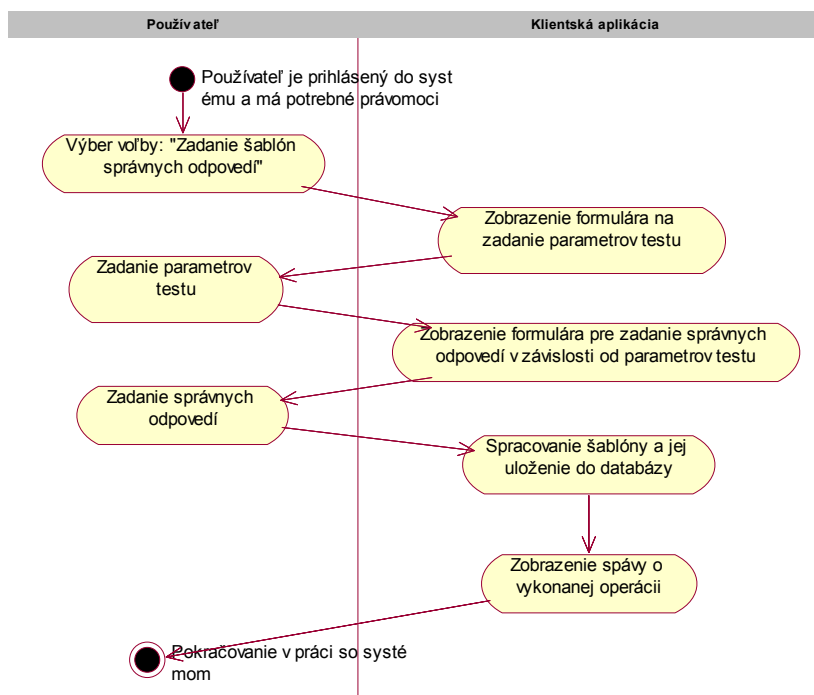


Obrázok 4: Diagram aktivít pre import údajov zo systému STUDAPP

**Zadanie šablón správnych odpovedí**

Identifikátor	UC03		
Názov	Zadanie šablón správnych odpovedí.		
Opis	Zadanie šablón správnych odpovedí, podľa ktorých sa budú vyhodnocovať testy.		
Priorita	1 = vysoká	Frekvencia	stredná
Vstup. podm.	Používateľ je prihlásený v systéme a má potrebné právomoci. Klientska aplikácia pozná adresu, prihlasovacie meno a heslo pre databázový server systému.		
Výstup. podm.			
Používatelia	Manažér		
Základná postupnosť	Krok	Činnosť	
	1	Používateľ vyberie voľbu: "Zadanie šablón správnych odpovedí".	
	2	Klientska aplikácia zobrazí formulár na zadanie parametrov testu (predmet, beh, sada, počet otázok)	
	3	Používateľ zadá parametre testu.	
	4	Klientska aplikácia podľa zadaných parametrov zobrazí formulár na zadanie správnych odpovedí.	
	5	Používateľ zadá správne odpovede.	
	6	Klientska aplikácia spracuje šablónu a uloží ju do databázy.	
	7	Klientska aplikácia zobrazí správu o vykonanej operácii.	
	8	Používateľ pokračuje v práci zo systémom.	

Tabuľka 5: Zadanie šablón správnych odpovedí



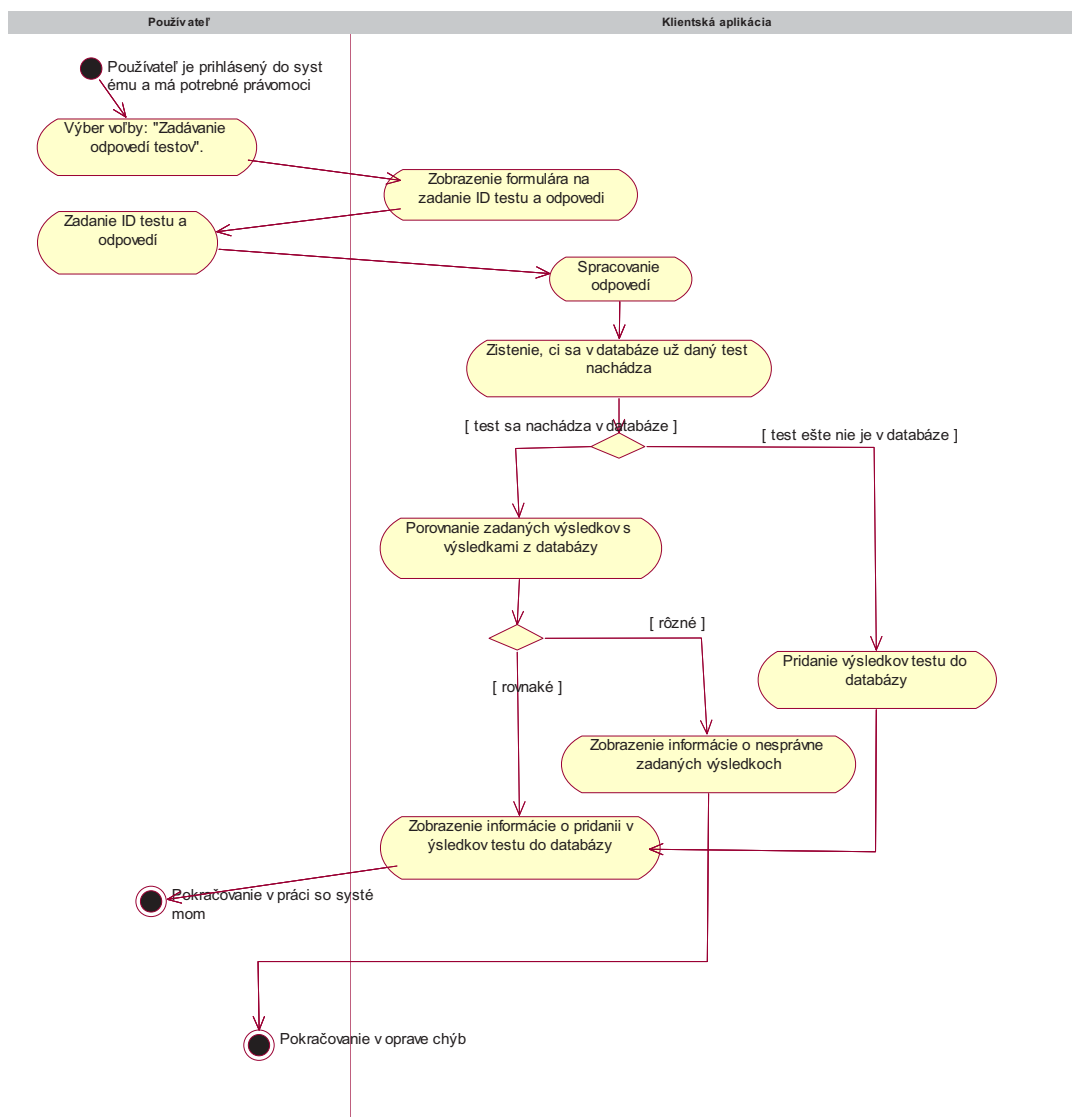
Obrázok 5: Diagram aktivít pre zadanie šablón správnych odpovedí

**Zadanie odpovedí testov**

Identifikátor	UC04		
Názov	Zadanie odpovedí testov.		
Opis	Používateľ zadáva odpovede opravovaných testov. Jeden testu musí zadať viac používateľov, pričom sa kontroluje či sa zadané výsledky pre konkrétny test zhodujú.		
Priorita	1 = vysoká	Frekvencia	vysoká
Vstup. podm.	Používateľ je prihlásený v systéme a má potrebné právomoci. Klientska aplikácia pozná adresu, prihlasovacie meno a heslo pre databázový server systému.		
Výstup. podm.			
Používatelia	zamestnanec, Manažér		
Základná postupnosť	Krok	Činnosť	
	1	Používateľ vyberie voľbu: "Zadanie odpovedí testov".	
	2	Klientska aplikácia zobrazí formulár na zadanie ID testu a odpovedí.	
	3	Používateľ zadá ID testu a odpovede.	
	4	Klientska aplikácia spracuje odpoveď a zistí, či sa v databáze už daný test nenachádza.	
	5a	Test sa už v databáze nachádza. Klientska aplikácia porovná zadané výsledky z výsledkami z databázy.	
	6a	Výsledky sú rovnaké. Klientska aplikácia zobrazí správu o pridaní testu do databázy.	
Alternatívna postupnosť	Krok	Činnosť	
	6b	Výsledky sa líšia. Klientska aplikácia zobrazí správu o odlišných výsledkoch. Pokračovanie v oprave chýb.	
	5c	Test ešte nie je v databáze. Klientska aplikácia pridá výsledky testu do databázy.	
	6c	Klientska aplikácia zobrazí správu o pridaní testu do databázy.	

Tabuľka 6: Zadanie odpovedí testov



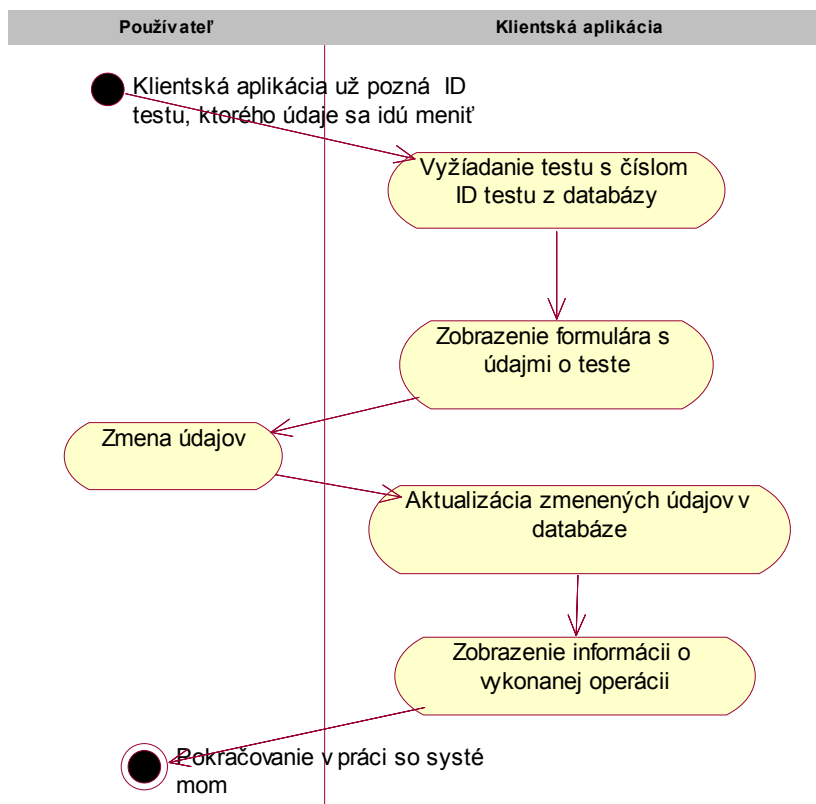


Obrázok 6: Diagram aktivít pre zadanie odpovedí testov

**Zmena údajov**

Identifikátor	UC05		
Názov	Zmena údajov		
Opis	Používateľ mení výsledky testu, ktorý je už v systéme zadaný. Scenár začína až potom, keď už je známe číslo testu, ktorý sa ide meniť.		
Priorita	1 = vysoká	Frekvencia	stredná
Vstup. podm.	Používateľ je prihlásený v systéme a má potrebné právomoci. Klientska aplikácia pozná adresu, prihlasovacie meno a heslo pre databázový server systému. ID testu, ktorý chceme meniť je známe. Test sa už nachádza v databáze.		
Výstup. podm.			
Používatelia	zamestnanec, Manažér		
Základná postupnosť	Krok	Činnosť	
	1	Klientská aplikácia si z databázy vyžiada test s daným ID testu.	
	2	Klientská aplikácia zobrazí formulár s údajmi o danom teste.	
	3	Používateľ zmení údaje.	
	5	Klientská aplikácia aktualizuje zmenené údaje v databáze.	
	6	Klientská aplikácia zobrazí informácie o vykonanej operácii.	

Tabuľka 7: Zmena údajov

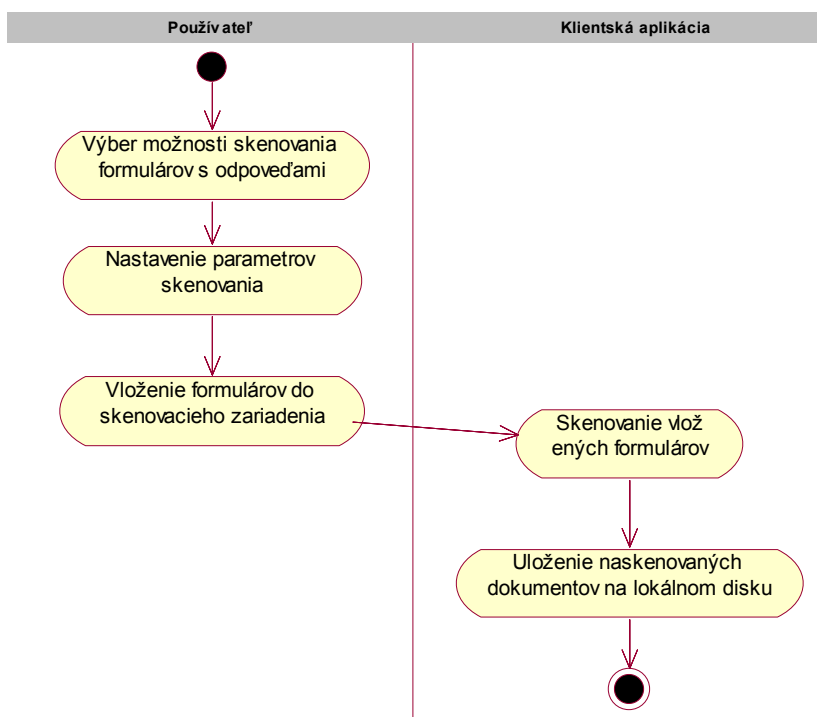


Obrázok 7: Diagram aktivít pre zmena údajov

**Skenovanie odpovedí z opravovaných testov**

Identifikátor	UC06		
Názov	Skenovanie odpovedí z opravovaných testov		
Opis	Formuláre s odpoveďami na testové otázky odovzdané uchádzačmi o štúdium sú skenované do systému		
Priorita	2 = stredná	Frekvencia	Pre každý odovzdaný test raz
Vstup. podm.	Používateľ je prihlásený do systému s príslušnými právomocami, skenovacie zariadenie funkčné, pripojené ku klientovi a správne nakonfigurované		
Výstup. podm.			
Používatelia	Zadávatel', Manažér		
Základná postupnosť	Krok	Činnosť	
	1	Používateľ si vyberie možnosť skenovania dokumentov a zvolí možnosť skenovania vyplnených formulárov s odpoveďami	
	2	Používateľ nastaví parametre skenovania, vloží vyplnené formuláre do skenovacieho zariadenia	
	3	Skenovacie zariadenie naskenuje vložené formuláre a ukladá ich do súborov na lokálnom disku počítača ku ktorému je pripojené skenovacie zariadenie	
Poznámky			

Tabuľka 8: Skenovanie odpovedí z opravovaných testov

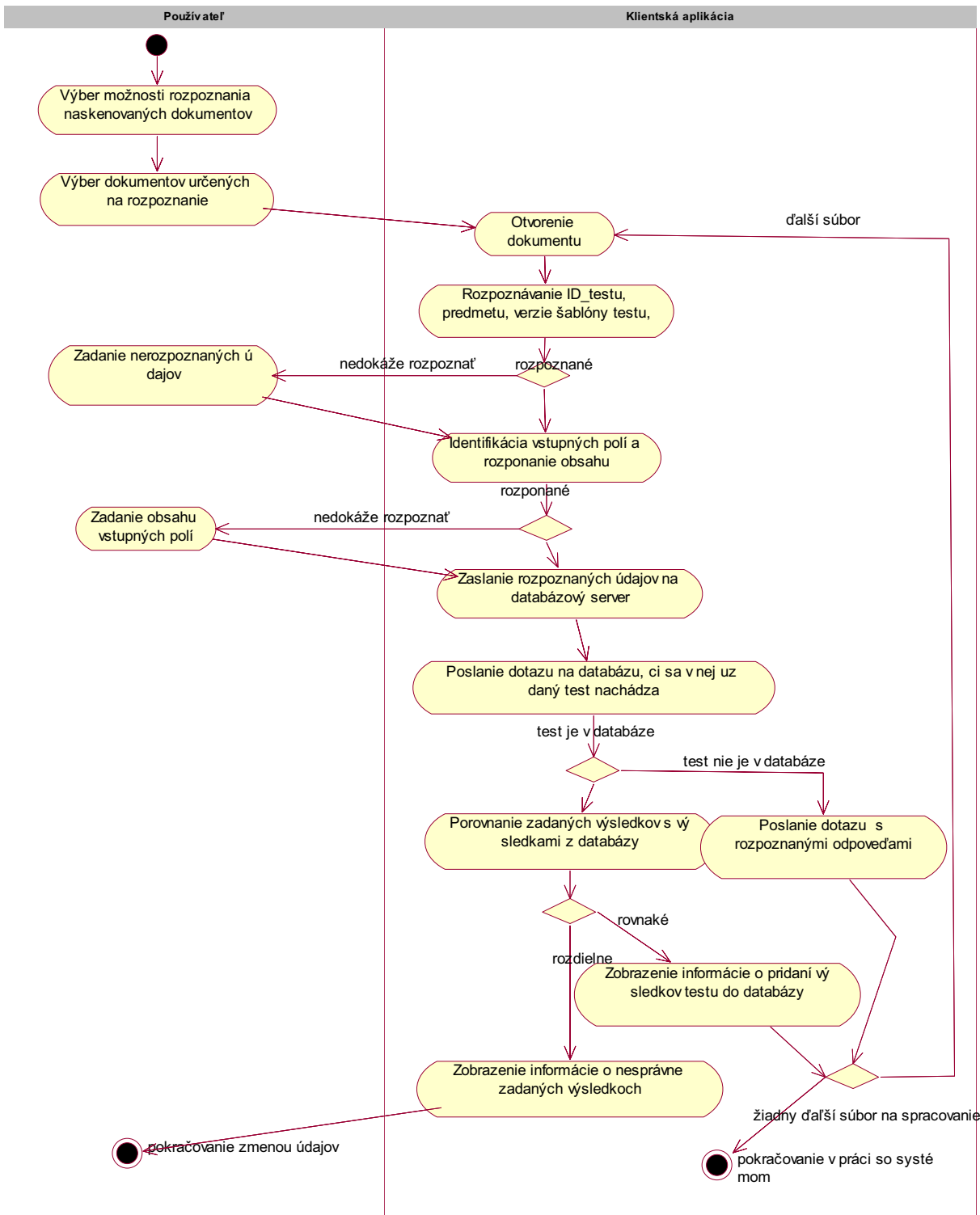


Obrázok 8: Diagram aktivít pre skenovanie odpovedí z opravovaných testov

**Import skenovaných dokumentov**

Identifikátor	UC07		
Názov	Rozoznávajúce a import skenovaných dokumentov do databázy		
Opis	Naskenované formuláre s odpoveďami na testové otázky odovzdané uchádzačmi o štúdium naskenované a uložené na lokálnom disku počítača sú rozpoznávané systémom, vyhodnotené podľa šablón a údaje z nich získané sú importované na databázový server		
Priorita	2 = stredná	Frekvencia	Raz v procese vyhodnocovania
Vstup. podm.	Formuláre s odpoveďami na testové otázky sú odovzdané uchádzačmi o štúdium a sú naskenované a uložené na lokálnom disku počítača. Používateľ je prihlásený do systému s príslušnými právomocami		
Výstup. podm.			
Používatelia	Manažér		
Základná postupnosť	Krok	Činnosť	
	1	Používateľ si vyberie možnosť rozpoznania naskenovaných dokumentov a vyberie príslušnú množinu naskenovaných dokumentov	
	2	Systém prechádza jednotlivé formuláre z množiny zvolených naskenovaných dokumentov pokiaľ nie sú rozpoznávané všetky.	
	3	Systém rozpoznáva identifikačné číslo na hlavičke formuláru vo forme čiarového kódu, predmet a sadu testu.	
	4	Podľa rozpoznávaného čísla verzie šablóny testu systém identifikuje vstupné polia formuláru a rozpoznáva ich obsah.	
	5	Systém posiela všetky rozpoznávané údaje z formuláru na databázový server.	
	6	V prípade, že sa v databáze už nachádza záznam pre rozpoznávaný test, údaje sa porovnávajú s existujúcou verziou.	
	7	Ak nenastal konflikt medzi importovanými údajmi a údajmi uloženými v databáze, údaje sú uložené a pokračuje sa ďalším z množiny rozpoznávaných testov v bode 2.	
Alternatívna postupnosť	Krok	Činnosť	
	3.0 b	Systém nedokáže rozpoznať niektorý z údajov na hlavičke formuláru vo forme čiarového kódu, zobrazí naskenovaný dokument na obrazovke a vyzve používateľa na ich zadanie.	
	3.1 b	Používateľ prezrie naskenovaný dokument a zadá údaje	
	4.0 c	Systém nedokáže rozpoznať obsah vstupného poľa definovaného v šablóne príslušnej verzie testu	
	4.1 c	Systém zobrazí naskenovaný dokument na obrazovke a vyzve používateľa na zadanie obsahu rozpoznávaného vstupného poľa (zvolené odpovede sú vo vstupných poliach označené krížikmi)	
	4.2 c	Používateľ prezrie naskenovaný dokument a zadá obsah vstupného poľa (či bolo označené krížikom alebo nie)	
	8 d	Údaje v databáze a rozpoznávané údaje sa líšia. Pokračuje sa scenárom zmeny údajov.	

Tabuľka 9: Import skenovaných dokumentov

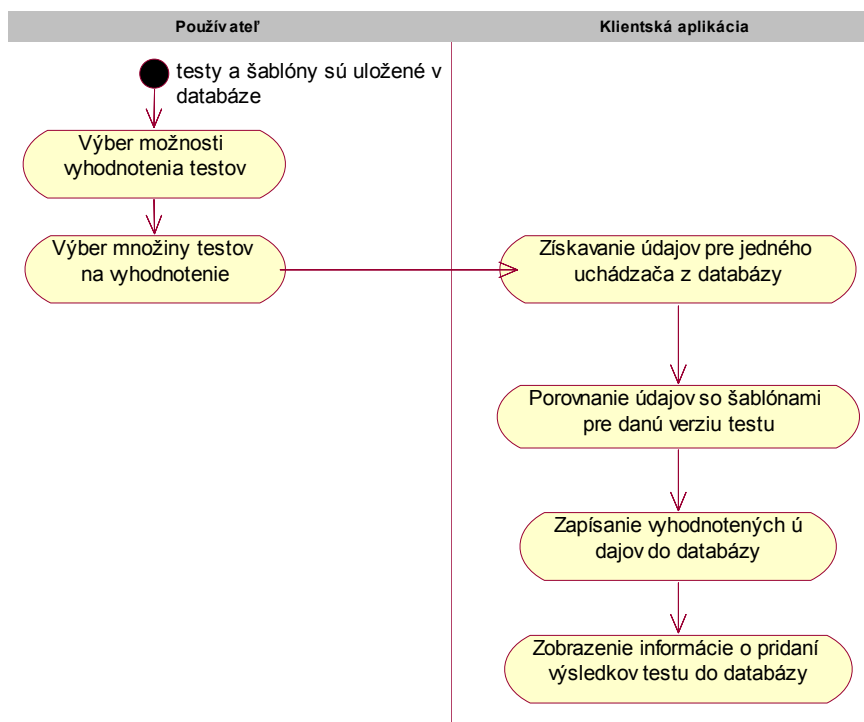


Obrázok 9: Diagram aktivít pre import skenovaných dokumentov

**Vyhodnotenie získaných údajov, porovnanie a opravenie konfliktov**

Identifikátor	UC08		
Názov	Vyhodnotenie získaných údajov, porovnanie a opravenie konfliktov		
Opis	Získané údaje vložené na databázovom serveri sa spracovávajú, vyhodnucujú a opravujú sa prípadné konflikty v prípade rôzneho vyhodnotenia		
Priorita	2 = stredná	Frekvencia	Pre každý odovzdaný test raz
Vstup. podm.	Údaje sú vložené do databázy, Používateľ je prihlásený do systému s príslušnými právomocami		
Výstup. podm.			
Používatelia	Manažér		
Základná postupnosť	Krok	Činnosť	
	1	Používateľ si vyberie možnosť vyhodnotenia získaných údajov	
	2	Systém porovnáva údaje v systéme s vloženými šablónami verzií správnych odpovedí testov, pričom správna verzia šablón je identifikovaná pri každom teste	
	3	Systém vyhodnocuje výsledky testov a vkladá ich do príslušných tabuliek databázy	
Poznámky			

Tabuľka 10: Vyhodnotenie získaných údajov

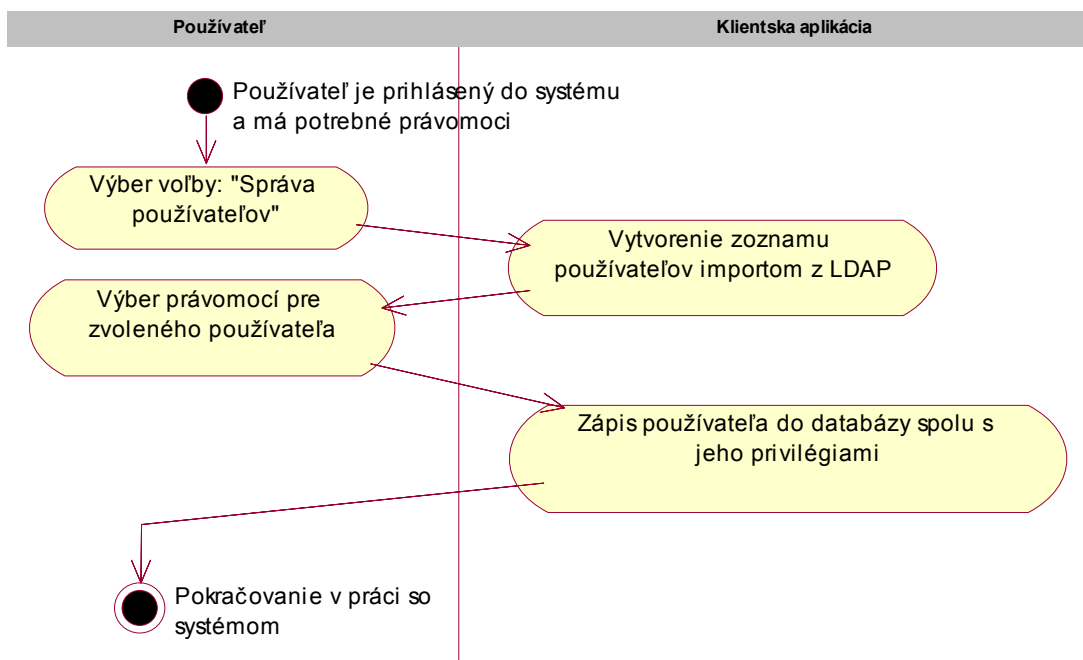


Obrázok 10: Diagram aktivít pre vyhodnotenie získaných údajov

**Správa používateľov**

Identifikátor	UC09		
Názov	Správa používateľov		
Opis	Manažér resp. administrátor (ďalej v tabuľke uvádzaný iba manažér) priraduje práva používateľom.		
Priorita	3 = vysoká	Frekvencia	Priamoúmerne závisí od počtu používateľov
Vstup. podm.	Používateľ musí byť prihlásený do systému.		
Výstup. podm.	Aktualizované kontá používateľov.		
Používatelia	Manažér, Administrátor		
Základná postupnosť	Krok	Činnosť	
	1	Používateľ si vyberie možnosť systému pre správu používateľov systému.	
	2	Klientská aplikácia zobrazí zoznam používateľov importom z LDAP.	
	3	Používateľ vyberie práva pre zvoleného používateľa.	
	4	Klientská aplikácia zapíše do databázy informácie o používateľovi s jeho právomocami.	

Tabuľka 11: Správa používateľov

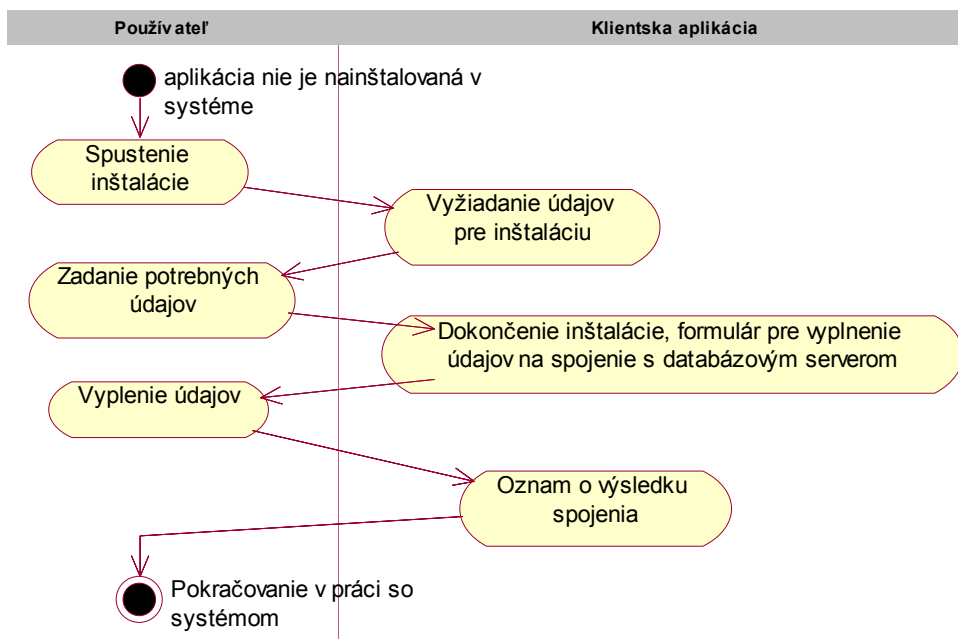


Obrázok 11: Diagram aktivít pre správu používateľov

**Inštalácia a konfigurácia klientov**

Identifikátor	UC10		
Názov	Inštalácia a konfigurácia klientov		
Opis	Administrátor nainštaluje klienta a nakonfiguruje prihlasovacie meno a heslo do databázy.		
Priorita	3 = vysoká	Frekvencia	Jeden krát pre každú pracovnú stanicu.
Vstup. podm.	Aplikácia nie je v systéme nainštalovaná. Administrátor pozná meno databázového servera na sieti, prihlasovacie meno a heslo.		
Výstup. podm.	Aplikácia je pripravená na komunikáciu s databázovým serverom.		
Používatelia	Administrátor		
Základná postupnosť	Krok	Činnosť	
	1	Používateľ spustí inštalátor aplikácie a zadá cestu, kam sa aplikácia nainštaluje.	
	2	Klientská aplikácia potvrdí správny priebeh inštalácie a spustí konfiguračnú obrazovku aplikácie pre spojenie s databázovým serverom.	
	3	Používateľ zadá názov databázového servera na sieti, prihlasovacie meno a heslo do databázy.	
	4	Klientská aplikácia uloží údaje v šifrovanej podobe na strane klienta.	
	5	Používateľ môže zadať možnosť otestovania spojenia s databázovým serverom.	
	6	Klientská aplikácia podá výsledok pokusu o spojenie s databázovým serverom.	
Poznámky	Proces inštalácie a konfigurácie klientov môže robiť i osoba používateľom poverená.		

Tabuľka 12: Inštalácia a konfigurácia klientov



Obrázok 12: Diagram aktivít pre inštaláciu a konfiguráciu klientov

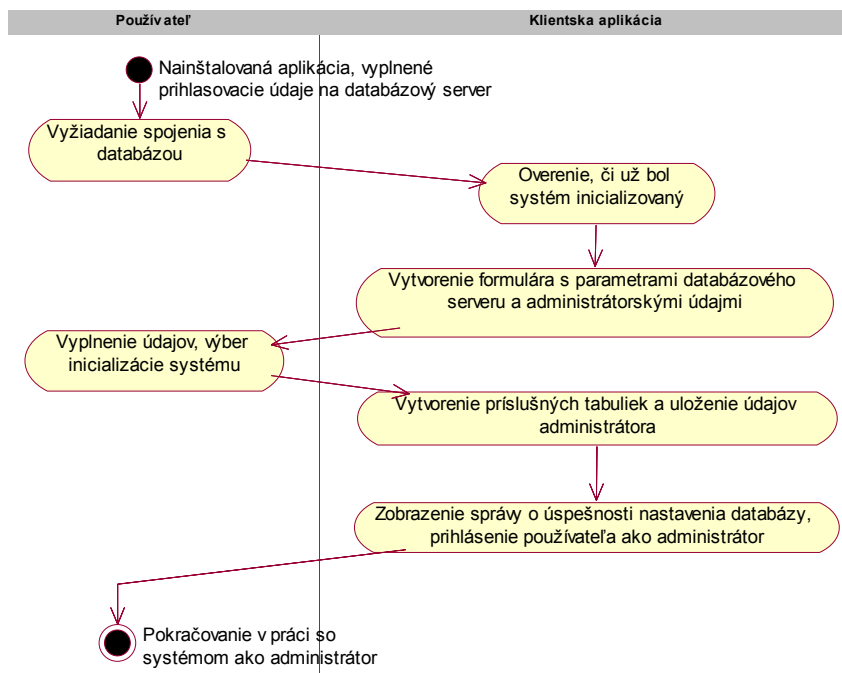




### Inicializácia databázového serveru

Identifikátor	UC11		
Názov	Inicializácia databázového serveru		
Opis	Administrátor sa spojí s databázovým serverom a aplikácia mu pri neexistencii potrebných údajov na databázovom serveri umožní vytvoriť a inicializovať databázu ako i prihlasovacie meno a heslo administrátora.		
Priorita	3 = vysoká	Frekvencia	Raz pri inicializácii systému.
Vstup. podm.	Administrátor pozná meno databázového servera na sieti, prihlasovacie meno a heslo. Je vytvorená databáza na databázovom serveri pre E-APP EVAL, ktorá nie je inicializovaná.		
Výstup. podm.	Databázový server je inicializovaný a pripravený na prácu s aplikáciou.		
Používatelia	Administrátor		
Základná postupnosť	Krok	Činnosť	
	1	Používateľ si pomocou klienta vyžiada spojenie s databázovým serverom.	
	2	Klientská aplikácia sa spojí s databázovým serverom. Overí, či už bola databáza inicializovaná.	
	3	Klientská aplikácia zobrazí úvod do používania aplikácie, formulár na zadanie prihlasovacieho mena a hesla používateľa systému a ponúkne používateľovi inicializáciu databázy.	
	4	Používateľ vyplní prihlasovacie meno i heslo a potvrdí možnosť inicializácie systému.	
	5	Klientská aplikácia overí údaje a vytvorí príslušné tabuľky, uloží údaje používateľa na databázový server. Aplikácia prihlási používateľa s administrátorskými právami	
Poznámky	Inicializáciu systému sa z používateľa stáva administrátor. Pod overením, či už bola inicializácia databázy vytvorená, sa rozumie overenie existencie administrátora v systéme.		

Tabuľka 13: Inicializácia databázového serveru

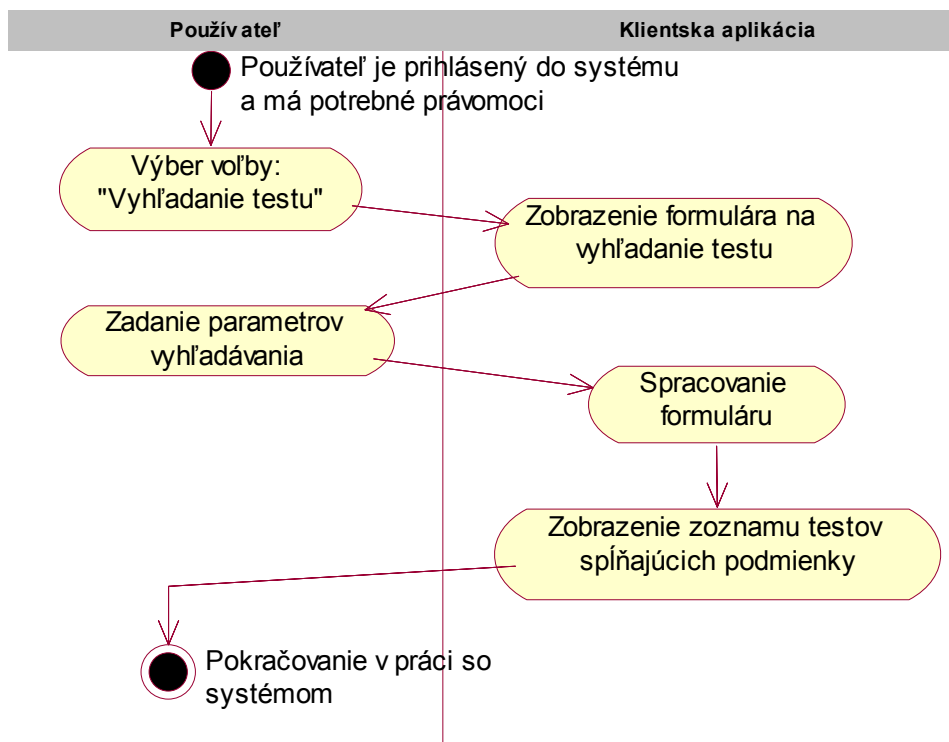


Obrázok 13: Diagram aktivít pre inicializáciu databázového serveru

**Vyhľadanie testu (podľa mena, registračného čísla študenta, čísla testu)**

Identifikátor	UC012		
Názov	Vyhľadanie testu (podľa mena, registračného čísla študenta, čísla testu)		
Opis	Používateľ systému vyhľadáva všetky opravy testu študenta na základe mena, registračného čísla študenta alebo čísla testu.		
Priorita	3 = vysoká	Frekvencia	Závisí od používateľov.
Vstup. podm.	Používateľ musí byť prihlásený do systému.		
Výstup. podm.	Zobrazený zoznam testov zadaného študenta.		
Používatelia	Manažér, Zadávatel'		
Základná postupnosť	Krok	Činnosť	
	1	Používateľ si vyberie možnosť systému pre vyhľadanie testov študenta.	
	2	Klientská aplikácia zobrazí formulár s možnosťou zadania mena, registračného čísla študenta alebo čísla testu.	
	3	Používateľ zadá meno a/alebo registračného čísla študenta a/alebo čísla testu.	
	4	Klientská aplikácia spraví logický súčin nad vyhľadávajúcimi údajmi. Vyhľadá testy podľa kritérií.	
	5	Systém vytvorí zoznam testov a zobrazí ho.	

Tabuľka 14: Vyhľadanie testu (podľa mena, registračného čísla študenta, čísla testu)

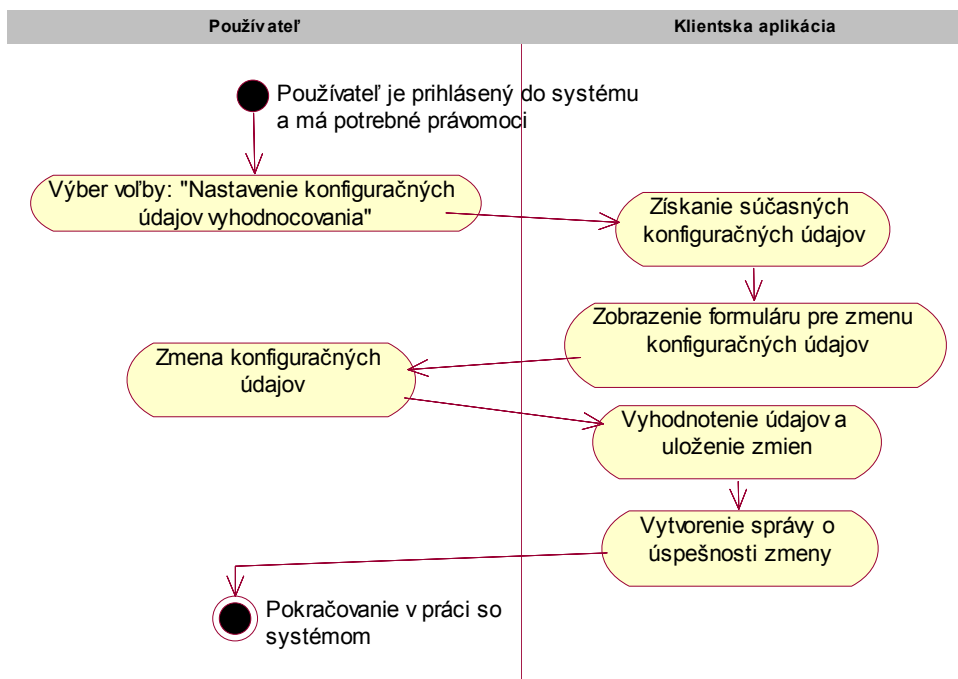


Obrázok 14: Diagram aktivít pre vyhľadanie testu (podľa mena, registračného čísla študenta, čísla testu)

**Zadanie konfiguračných údajov pre opravu**

Identifikátor	UC13		
Názov	Zadanie konfiguračných údajov pre opravu		
Opis	Manažér nastaví spôsob práce so systémom pre všetkých zadávateľov.		
Priorita	2 = stredná	Frekvencia	Pred začatím opravy testov
Vstup. podm.	Manažér musí byť prihlásený do systému.		
Výstup. podm.	Nastavené parametre.		
Používatelia	Manažér		
Základná postupnosť	Krok	Činnosť	
	1	Používateľ si vyberie možnosť systému pre nastavenie parametrov systému.	
	2	Klientská aplikácia zobrazí formulár s prednastavenými parametrami, ktoré je možné editovať.	
	3	Používateľ upraví požadované vlastnosti.	
	4	Systém zaznamená zmenu parametrov, informuje o výsledku konfigurácie.	
Poznámky	Nastavenie parametrov je nevyhnutné pred začatím opravovania testov. Parametrami systému sú napr.: počet testových behov, počet predmetov, z ktorých sa robí prijímacia skúška, ...		

Tabuľka 15: Zadanie konfiguračných údajov pre opravu

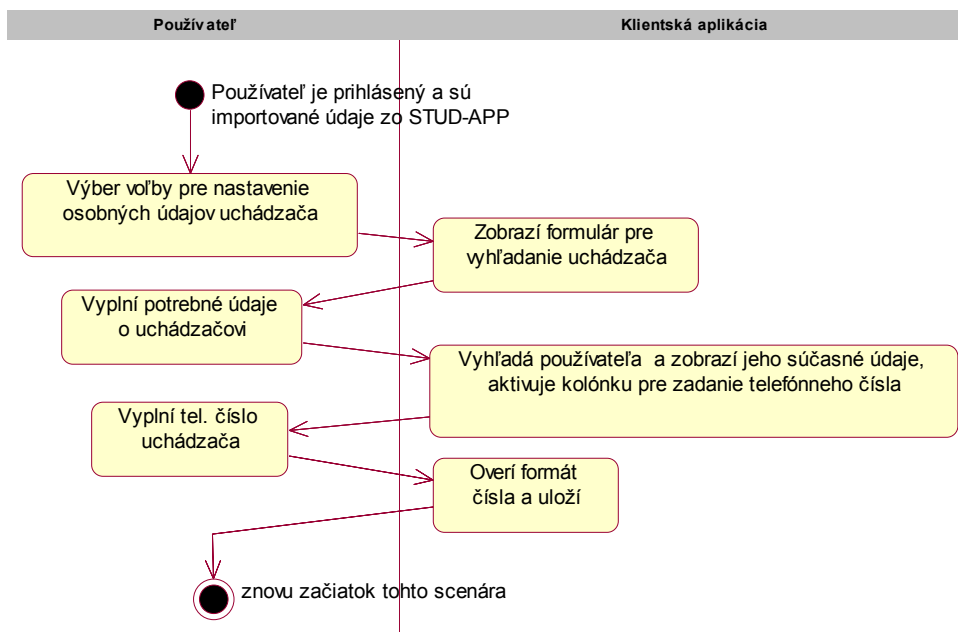


Obrázok 15: Diagram aktivít pre zadanie konfiguračných údajov pre opravu

**Zadanie telefónneho čísla uchádzača**

Identifikátor	UC14		
Názov	Zadanie telefónneho čísla uchádzača.		
Opis	Používateľ zadá do profilu uchádzača telefónne číslo.		
Priorita	2 = stredná	Frekvencia	Pre každého uchádzača aspoň raz.
Vstup. podm.	Používateľ musí byť prihlásený do systému. V systéme E-APP EVAL sú importované údaje zo STUD-APPu.		
Výstup. podm.	Nastavené parametre.		
Používatelia	Manažér, Zadávateľ		
Základná postupnosť	Krok	Činnosť	
	1	Používateľ si vyberie možnosť systému pre nastavenie údajov uchádzača.	
	2	Systém zobrazí formulár s možnosťou vyhľadania uchádzača.	
	3	Používateľ vyplní potrebné údaje o uchádzačovi.	
	4	Klientská aplikácia zobrazí formulár so súčasnými údajmi uchádzača. Kolónku s telefónnym číslom nastaví ako aktívnu. Pri priemeroch vysvedčení za jednotlivé roky je tlačidlo s názvom „Zmeň“.	
	5	Používateľ vyplní do správnej kolónky telefónne číslo uchádzača.	
	6	Klientská aplikácia overí formát čísla a zaznamená ho.	
Poznámky	Vo formulári s údajmi uchádzača je vždy aktívna kolónka s telefónnym číslom z dôvodu najväčšej pravdepodobnosti zmeny práve tohto údaje. Kolónka je naplnená tel. číslom uchádzača, ktoré zadal pred prijímacím konaním.		

Tabuľka 16: Zadanie telefónneho čísla uchádzača



Obrázok 16: Diagram aktivít pre zadanie telefónneho čísla uchádzača



### **Oprava osobných údajov uchádzača**

Prípad použitia pre opravu osobných údajov je podobný scenáru pre zadanie telefónneho čísla uchádzača s tým rozdielom, že používateľ zmení osobné údaje namiesto telefónneho čísla.

### **Zadanie vysvedčení uchádzača**

Prípad použitia pre zadanie vysvedčení do systému je podobný scenáru pre zadanie telefónneho čísla uchádzača s tým rozdielom, že používateľ si v kroku 4 zvolí tlačidlo "Zmeň" pri vysvedčení, ktoré chce zmeniť. Aplikácia zobrazí dialóg pre zadanie predmetov a známok. Vypočíta priemer.

## **3.2 Kontrola vstupných údajov**

Pri zadávaní všetkých vstupných údajov do systému je potrebná kontrola, ktorá nielen zabezpečuje platnosť údajov vstupujúcich do systému ale kontrolou údajov priamo na vstupe odpadá následná réžia pri vyhl'adávaní a oprave chýb. Údaje sa dajú kontrolovať dvoma spôsobmi:

- kontrola na definičný obor, kde sa kontroluje či vstupné údaje sú predpísaného formátu alebo či sú z požadovaného rozsahu
- druhou možnosťou je kontrola na jedinečnosť, teda či už v databáze existuje rovnaká položka so zadávanými údajmi a či sa

Pričom niektoré vstupné údaje sa kontrolujú oboma spôsobmi. Je vhodné spísať kontrolu vstupných údajov postupne tak, ako sa zadávajú do systému podľa scenárov prípadov použitia.

### **UC03 Zadanie šablón správnych odpovedí**

- predmet - môže byť len z množiny predmetov z ktorých sa daná prijímacia vykonáva
- beh - definuje sa miestnosť z množiny dostupných miestností a čas z množiny dostupných časov, žiadna iná miestnosť ani iný čas nie je prípustný
- číslo sady - číslo sady v rámci určeného behu, toto číslo musí byť pre danú sadu jedinečné a musí byť v intervale 0-9
- počet otázok - môže byť ľubovoľné kladné celé číslo väčšie ako 0
- počet možných odpovedí - celé číslo v intervale od 2-10

### **UC04 Zadanie odpovedí testov**



- ID testu - číslo testu uchádzača, obmedzené rozsahom pridelených ID
- beh - 2 čísla v rozsahu 0-9, ktoré identifikujú miestnosť a čas konania testu, v prípade generovania jedinečných testov pre každého uchádzača sa toto číslo nezadáva
- sada - číslo sady v rámci behu v rozsahu (0-9), takéto číslo sady musí pre daný beh existovať, v prípade generovania jedinečných testov pre každého uchádzača sa toto číslo nezadáva ani nekontroluje
- odpoveď na otázku - zadáva sa číslo odpovede v rozsahu možných odpovedí pre danú otázku (uvedené je to v šablónach správnych odpovedí), jednomiestne číslo (0-9)

**UC05 Zmena údajov** kontrola zadávaných údajov je rovnaká ako pri scenári zadávania odpovedí testov

**UC07 Import skenovaných dokumentov** kontrola zadávaných údajov je rovnaká ako pri scenári zadávania odpovedí testov

#### **UC014 Zadávanie osobných údajov**

- registračné číslo - registračné číslo študenta z aplikácie STUDAPP
- telefónne číslo - telefónne číslo na mobilný telefón, 10 miestne v príslušnom tvare, prípadne medzinárodné, kontrola na jedinečnosť zadaného čísla, je nepravdepodobné aby dvaja uchádzači mali rovnaké číslo mobilného telefónu, možná kontrola prvého štvorčísla či existuje taká predvoľba na mobilného operátora
- rodné číslo - 10 miestne číslo v príslušnom tvare, kontrola na jedinečnosť, dvaja ľudia by nemali mať rovnaké rodné číslo, kontrola na deliteľnosť 11
- ročník strednej školy - hodnota obmedzená intervalom pre danú školu
- známka z predmetu - hodnota v rozmedzí 1-4, prípadne iné obmedzenie zápisu hodnotenia v prípade maturitných vysvedčení

### **3.3 Nefunkcionálne požiadavky na systém**

V tejto podkapitole zhrnieme nefunkcionálne požiadavky kladené na vyvíjaný informačný systém. Zároveň popíšeme do akej miery sú už tieto požiadavky splnené a čo je ešte potrebné zabezpečiť pre zabezpečenie fungovania systému v predpokladaných podmienkach.



### 3.3.1 Predpokladané ľudské zdroje

Odhadovaný počet uchádzačov o štúdium, ktorí sa zúčastnia na prijímacích testoch je približne 1500. Pri každom uchádzačovi je potrebné zadať do systému dokument s osobnými údajmi (zadanie čísla testu a telefónneho čísla, v prípade potreby zmena ostatných údajov), vysvedčenia (zadanie rodného čísla a známkov na maturitnom vysvedčení, prípadne aj na vysvedčení z posledného ročníka) a dva dokumenty s vyznačenými testovými odpoveďami (každý by mali spracovať aspoň dvaja zamestnanci). Dokopy je potrebné pre každého uchádzača spracovať minimálne 6 dokumentov. Pri 1500 uchádzačoch o štúdium to predstavuje 9000 dokumentov. Aby zadanie jednotlivých dokumentov do informačného systému prebehlo dostatočne rýchlo, musí sa na ňom podieľať súčasne viacero zamestnancov fakulty. Nami odhadovaný počet súčasne pracujúcich zamestnancov je 10 až 20 pri ručnom zadávaní testov. Pri použití automatického zadávania pomocou skeneru by tento počet mierne klesol, avšak stále by sa niekoľko zamestnancov muselo venovať zadávaniu dokumentov s osobnými údajmi. Ďalšia skupina zamestnancov (jeden až dvaja) by sa musela venovať obsluhu skenera a spracovaniu nerozpoznaných testov. Predpokladaný počet zamestnancov pri použití automatického zadávania testov by teda bol približne 5 až 10 zamestnancov. Každý zamestnanec musí mať k dispozícii osobný počítač, na ktorom bude spustená klientská aplikácia. Z dôvodu zabránenia falšovania výsledkov by bolo vhodné ak by všetci pracovali v jednej miestnosti.

### 3.3.2 Technické požiadavky

Osobný počítač, na ktorom bude spustená klientská aplikácia musí mať nainštalovaný operačný systém MS Windows 2000/XP/2003. Aby bolo možné zadané údaje uchovávať v databáze spustenej na databázovom serveri, je potrebné aby fungovalo TCP/IP spojenie z osobného počítača na databázový server. V prípade firewallu medzi týmito dvoma uzlami je potrebné povoliť TCP port 5432. Pre beh aplikácie je potrebné poznať informácie o zamestnancoch fakulty, ktorý budú zo systémom pracovať. Tieto informácie sa získajú z LDAP systému fungujúceho na fakulte. Preto je potrebné aby fungovalo TCP/IP spojenie z osobného počítača na LDAP server. V prípade firewallu medzi týmito dvoma uzlami je potrebné povoliť TCP port 389.

Systém bude na uchovanie údajov používať databázu PostgreSQL. Preto je potrebné aby na počítači, ktorý bude slúžiť ako databázový server bola táto databáza nainštalovaná a bolo



možné sa na server pripojiť z osobných počítačov určených na zadávanie údajov. Takisto musí existovať server, na ktorom bude systém LDAP a bude sa na neho dať pripojiť z osobných počítačov určených na zadávanie údajov. PostgreSQL a LDAP môžu byť nainštalované na tom istom serveri alebo oddelené.

### 3.3.3 Automatizované spracovanie testov

Namiesto ručného zadávania údajov sa môže použiť automatické rozpoznanie údajov s použitím skenera. Skenovali by sa len dokumenty s odpoveďami, čo pri 1500 uchádzačoch o štúdium predstavuje 3000 listov papiera. Pri použití skenera s rýchlosťou skenovanie 20 strán za minútu by naskenovanie všetkých listov trvalo 150 minút. Skener musí mať automatický podávač a na pripojenie k počítaču používať rozhranie USB 2.0. Počítač ku ktorému bude skener pripojený musí spĺňať tie isté požiadavky ako osobný počítač popísaný vyššie. Navyše musí mať port USB 2.0 a voľnú diskovú kapacitu približne 10 GB.

### 3.3.4 Súhrn požiadaviek

Nefunkcionálne požiadavky sa dajú zhrnúť do niekoľkých bodov:

- 10 - 20 (5 - 10 pri automatickom zadávaní testov) osobných počítačov s operačným systémom MS Windows XP/2000/2003 v jednej miestnosti
- každý osobný počítač musí byť schopný pripojiť sa k databázovému severu PostgreSQL cez TCP port 5432
- každý osobný počítač musí byť schopný pripojiť sa k LDAP serveru cez port TCP 389
- databázový server PostgreSQL umožňujúci pripojenie z osobných počítačov
- LDAP server umožňujúci pripojenie z osobných počítačov
- skener s automatickým podávačom aspoň 20 strán za minútu s USB 2,0 rozhraním (platí len pre automatické zadávanie testov)
- osobný počítač na pripojenie skenera s portom USB 2.0 a 10 GB voľnej diskovej kapacity (platí len pre automatické zadávanie testov)

### 3.3.5 Súčasný stav

Na fakulte sa nachádza miestnosť CISCO Lab. s 20 osobnými počítačmi spĺňajúcimi naše požiadavky. V miestnosti je navyše jeden počítač s portom USB 2.0 a dostatočnou diskovou kapacitou, ktorý by sa mohol použiť na pripojenie skenera. Správca počítačovej siete v CISCO





Lab. nám prisľúbil povolenie požadovaných portov na pripojenie k serveru LDAP a PostgreSQL. Na fakulte sa nachádza server Projects, na ktorom je nainštalovaná databáza PostgreSQL aj LDAP.

Skener s požadovanými parametrami nemá fakulta k dispozícii a je potrebné ho zakúpiť. Cena skenera s požadovanými parametrami je približne 30 000 až 40 000 SK.

### **3.3.6 Zhodnotenie**

Splnené sú všetky nefunkcionálne požiadavky okrem požiadavky na skener. Záleží na zákazníkovi, či je pre neho bude postačujúce ručné spracovanie formulárov, alebo bude požadovať automatické rozpoznanie formulárov. V tomto prípade je potrebné zakúpiť skener.



## 4 Hrubý návrh

### 4.1 Konceptuálny diagram

Koncept systému EVAL-APP pozostáva v identifikovaní jeho hlavných činností, uložisk údajov, fyzických objektov, s ktorými sa bude manipulovať a väzieb medzi nimi. Konceptuálny diagram systému sa nachádza na obrázku ??

EVAL-APP bude pracovať s týmito uložiskami údajov:

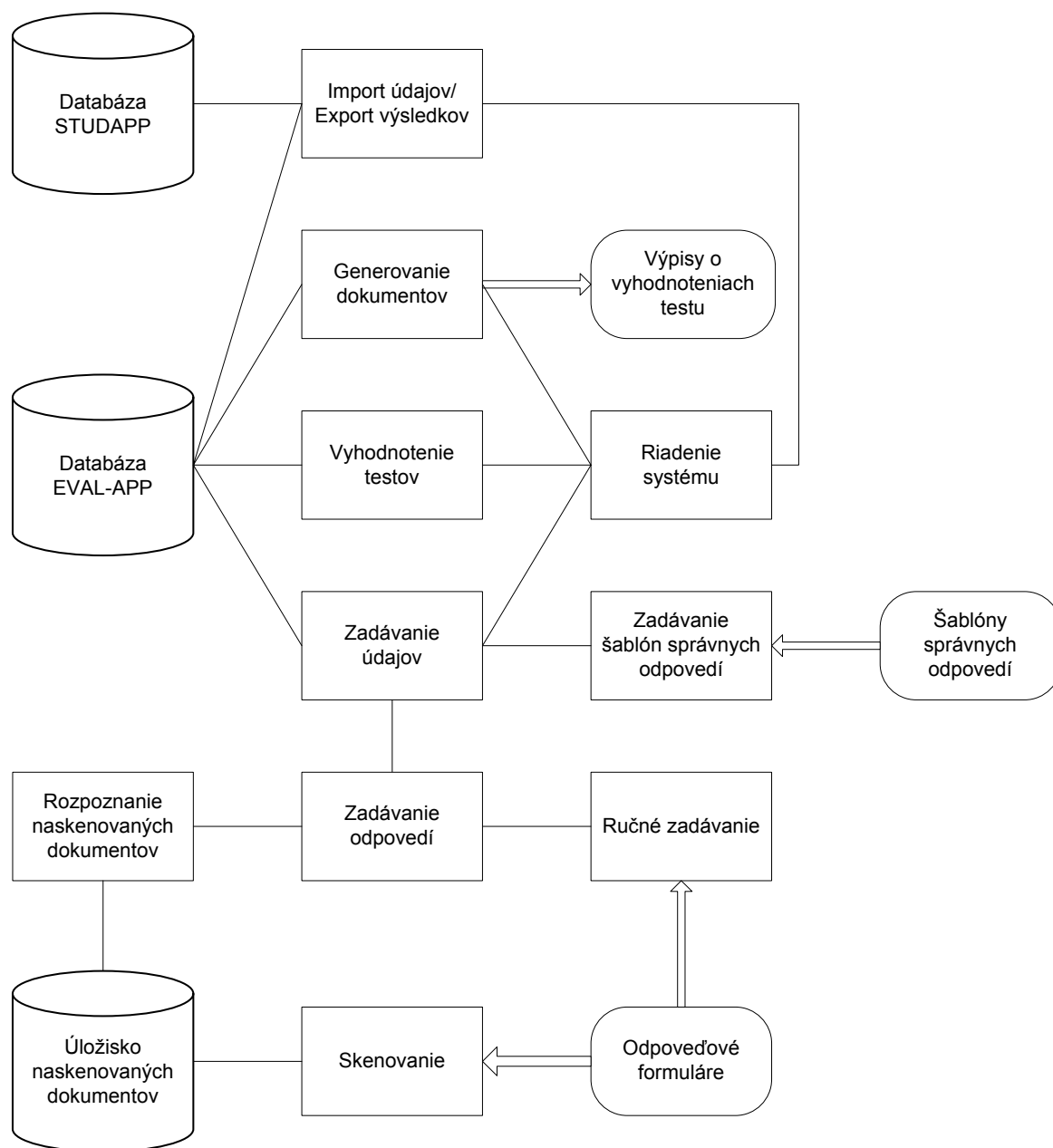
- Databáza STUADPP, odtiaľto sa budú čítať údaje o uchádzačoch a zapisovať výsledky vyhodnotenia prijímacej skúšky
- Databáza EVAL-APP, pracovná databáza systému obsahujúca vzory testov, odpovede uchádzača a ďalšie informácie (pozri logický model)
- Úložisko naskenovaných dokumentov, na súborovom systéme uložené naskenované vyplnené odpoveďové hárky uchádzačov.

Ako vstupné fyzické médiá pre EVAL-APP budú vyplnené odpoveďové formuláre uchádzačov a šablóny správnych odpovedí pre jednotlivé sady testov. Výstupnými fyzickými objektami budú dokumenty s vyhodnotením prijímacej skúšky uchádzača.

Pod spracovaním vyplnených odpoveďových formulárov rozumieme *zadávanie uchádzačovských odpovedí* z týchto formulárov do systému EVAL-APP. Môže sa to udiť ručným vkladáním údajov do systému, alebo nasledovným automatickým procesom:

1. Zoskenovanie odpoveďového formulára
2. Uloženie tohto formulára do úložiska v elektronickej podobe.
3. Rozpoznanie vyznačených odpovedí v týchto dokumentoch

Spolu so *zadávaním šablón správnych odpovedí* patrí *zadávanie uchádzačovských odpovedí* k činnosti *zadávaniu údajov do systému*. Tieto údaje sa následne uchovávajú v databáze EVAL-APP. *Riadením systému* je ďalej ovládané *vyhodnocovanie testov* uchádzača a *generovanie adresného dokumentu* s bodovým vyhodnotením jednotlivých otázok uchádzača. Po vyhodnutí testov sú konečné údaje *importované* do aplikácie STUADPP.



Obrázok 17: Konceptuálny diagram systému

## 4.2 Model údajov

Táto kapitola obsahuje pohľad na uchovávané údaje. Identifikuje základné entity, ich atribúty a vzťahy. Je rozdelená na dve časti. Prvá časť obsahuje predstavu spôsobu riešenia daného modelu údajov, entít a vzťahov medzi nimi. Druhá časť kapitoly obsahuje podrobné vysvetlenie jednotlivých entít diagramu.



#### 4.2.1 Diagram modelu údajov ( logická úroveň )

Diagram na obrázku 18 pozostáva z entít *Výsledok*, *Vzor testu*, *Otázka*, *Možnosť*, *Pracovník*, *Odpoveď*, *Uchádzač*. Tie sú navzájom poprepájané relačnými vzťahmi, ktoré určujú, že dané entity sú v navzájom relácií (s určitou kardinalitou). Entity okrem mena obsahujú atribúty, ktoré sú uvedené pod čiarou. Relačné vzťahy vyjadrujú nasledovné skutočnosti:

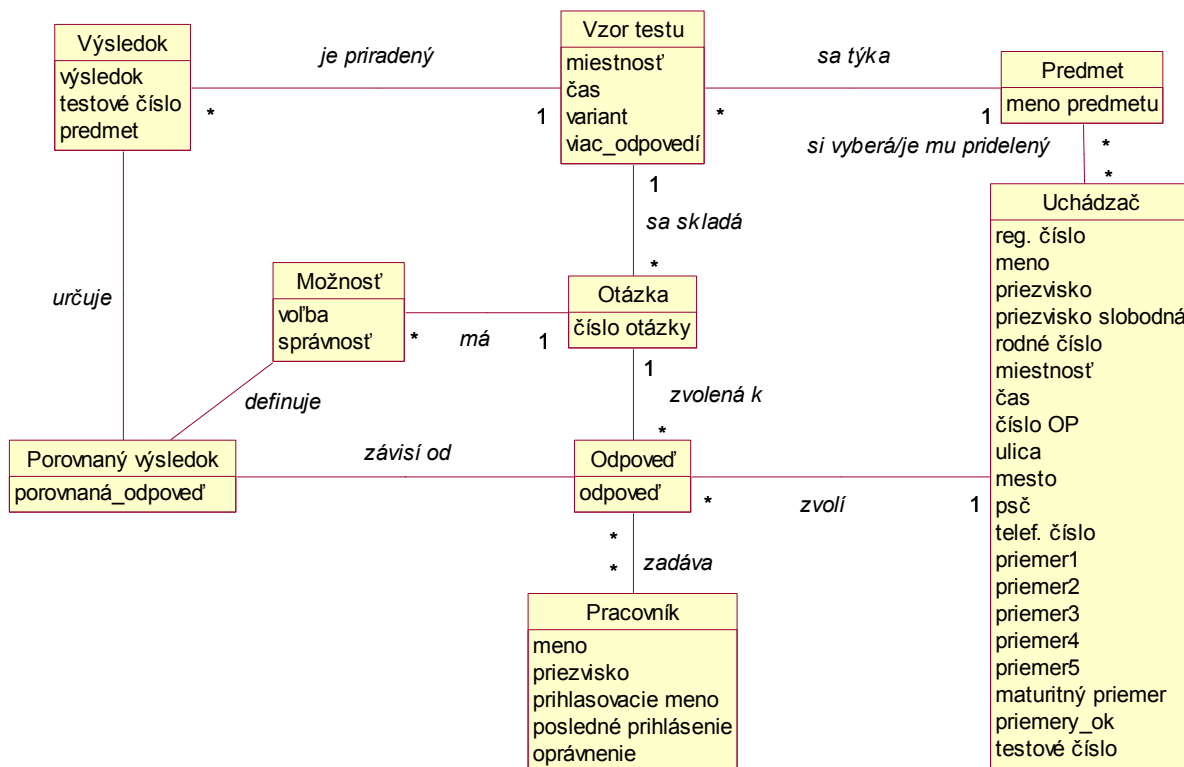
Uchádzač o štúdium vyplňa počas prijímacieho konania test z viacerých predmetov. Tieto testy sú definované entitou *Vzor testu*. Vzor testu definuje z akej oblasti je daný test pomocou atribútu *Predmet* a taktiež definuje kde a kedy sa daný test konal. Test sa skladá z určitého počtu otázok, čo zachytáva entita *Otázka*. Táto entita je navrhnutá s možnosťou doplniť neskôr texty otázok. Testové otázky môžu mať rôzny počet možných odpovedí. Možné odpovede na otázku popisuje entita *Možnosť*. Tá ďalej definuje aj to, či je daná možnosť správna, či sa jedná o otázku s jednou alebo viacerými správnymi odpoveďami. Týmto sa jednoznačne definuje test, jeho otázky, možnosti odpovedí na otázky a správne odpovede. Konkrétne odpovede uchádzačov sú uložené v entite *Odpoveď*. Výsledok testu sa po vyhodnotení uloží prostredníctvom entity *Výsledok*.

#### 4.2.2 Entity logického modelu údajov

Táto kapitola obsahuje stručný opis entít logického modelu údajov a ich atribútov tak ako boli uvedené v diagrame logického modelu údajov na obrázku 18.

**Entita Uchádzač:** je nositeľom základných informácií pre jednotlivých uchádzačov o štúdium. Dôležité je registračné číslo študenta a testové číslo študenta. Testové číslo identifikuje uchádzača v procese vyhodnocovania testov. Pri jednotlivých uchádzačoch je potrebné zadávať aj chýbajúce priemery zo strednej školy, preto sa tieto tiež nachádzajú v entite. Stredné školy majú rôznu dĺžku štúdia. Preto nemožno ani na základe typu strednej školy rozpoznať, či uchádzač má všetky potrebné priemery zo stredných škôl zaevidované v systéme. Pri aktualizácii osobných údajov počas prijímacej skúšky môže nastať situácia, že uchádzač nedodal potrebné vysvedčenia na výpočet priemerov. Túto skutočnosť zachytáva atribút *priemery\_ok*.

Ďalej sa tu nachádzajú osobné údaje uchádzača kvôli možnej oprave preklepov. Informácie o



Obrázok 18: Diagram logického modelu údajov

jednotlivých uchádzačoch sa načítajú zo systému elektronickej prihlášky STUDAPP.

**Entita Pracovník:** Predstavuje osobu, ktorá pracuje so systémom. Je identifikovaný pomocou svojho prihlasovacieho mena. Na autentifikáciu pracovníkov sa predpokladá použiť LDAP server, preto sa v entite nenachádza atribút heslo. Atribút *oprávnenie* určuje stupeň právomocí jednotlivých používateľov, pričom používateľ s vyšším stupňom oprávnení môže pracovať so systémom rovnako ako používateľ z nižším stupňom a okrem toho má prístup ďalším funkciám.

**Entita Vzor testu :** definuje test, ktorý vypracovávajú uchádzači. Test je definovaný pomocou predmetu testu, miestnosti, kde sa daný test bude písať a času. Ku konkrétnemu testu sa viaže sada testových otázok. Atribút *viac odpovedí* určuje, či majú otázky daného testu iba jednu správnu odpoveď alebo viacero.

**Entita Otázka:** popisuje jednotlivé otázky testu resp. ich poradové číslo v konkrétnom teste a počet bodov, ktorý je možné získať za danú odpoveď. Zatiaľ sa neuvažuje o evidovaní textov k



otázkam.

**Entita Možnosť:** zachytáva možnosti odpovede na testovú otázku. Definuje počet možností odpovede, ich správnosť. Taktiež definuje, či je na otázku len jedna správna odpoveď alebo viacero.

**Entita Odpoveď:** reprezentuje konkrétnu odpoveď uchádzača na testovú otázku, teda ktoré možnosti zvolil.

**Entita Výsledok:** predstavuje vyhodnotenie testu z konkrétneho predmetu. Jednotlivé testy sú definované pomocou testového čísla a predmetu testu.

**Entita Porovnaný výsledok** predstavuje porovnané výsledky uchádzača. Odpovede uchádzača sa porovnajú so vzorom testu. Ku každej možnosti každej otázky vznikne teda akýsi medzivýsledok v podobe jednej z variant:

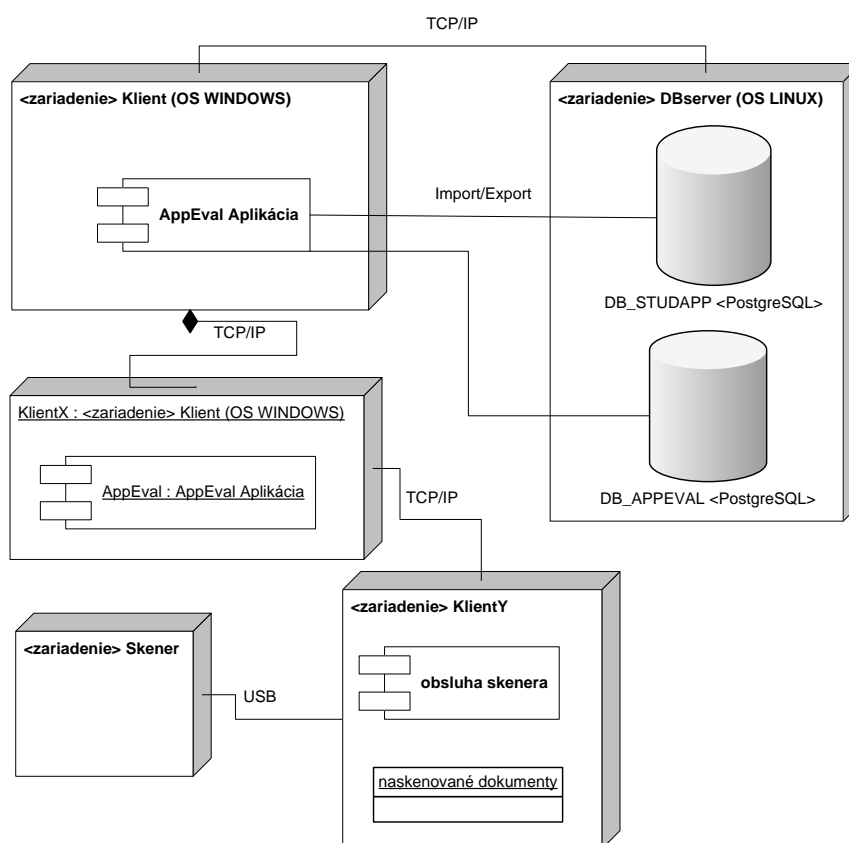
- možnosť bola správna a uchádzač ju zvolil
- možnosť bola správna a uchádzač ju nezvolil
- možnosť bola nesprávna a uchádzač ju zvolil
- možnosť bola nesprávna a uchádzač ju nezvolil

Na základe informácií z tejto entity a na základe stratégie opravovania testov a priradí každej otázke bodové ohodnotenie. Výsledné bodové ohodnotenie na celý test uchováva entita *Výsledok*.

### 4.3 Diagram nasadenia

Diagram nasadenia na obrázku 19 zobrazuje statický pohľad na konfiguráciu uzlov a komponentov systému, ktoré na týchto uzloch bežia. Uzly predstavujú prevažne hardvérové zariadenia, na ktorých bežia softvérové riešenia.

**Klient:** množina zariadení (počítačov). Uzol zovšeobecňuje počítače zapojené v sieti, na ktorých beží softvérový komponent, aplikácia APP-EVAL. Na počítačoch je nainštalovaný ope-



Obrázok 19: Diagram nasadenia systému APP-EVAL

račný systém Windows. Tento uzol predstavuje vlastne všetky počítače, zapojené v sieti, na ktorých môžu používatelia pracovať s aplikáciou APP-EVAL. Množina klientov je pripojená k externému zariadeniu, databázovému serveru, ktorý beží pod operačným systémom LINUX.

**DBserver:** uzol predstavuje zariadenie, na ktorom beží databázový server. Na tomto serveri bežia dve PostgreSQL databázy a to DB-STUDAPP, databáza aplikácie STUDAPP a vlastná databáza aplikácie APP-EVAL. Server je pripojený na sieť zahŕňajúcu všetky klientské počítače s bežiacou aplikáciou APP-EVAL. Databáza DB-APPEVAL obsahuje všetky informácie aplikácie APP-EVAL, ako výsledky testov, osobné údaje uchádzačov, šablóny testov a iné. Aplikácia APP-EVAL prístupuje do databázy systému STUDAPP len v prípade importu alebo exportu údajov.

**KlientX:** tento uzol predstavuje inštanciu jedného zo zariadení typu Klient, teda jeden počítač.



tač zapojený do siete s bežiacou inštanciou aplikácie APP-EVAL. Má prístup k naskenovaným dokumentom na zariadení typu KlientY, a má možnosť tieto dokumenty spracúvať pomocou inštancie aplikácie APP-EVAL. Získané údaje sa exportujú do vlastnej databázy na databázovom serveri.

**KlientY:** uzol predstavuje počítač pripojený do siete. Na tento počítač je pomocou rozhrania USB pripojené zariadenie Skener. Na lokálny disk klienta sa pomocou obslužného softwaru skenera skenujú dokumenty (vypracované testy). Skener je pripojený pomocou rozhrania USB. Prístup k naskenovaným dokumentom by mal mať najmenej jeden z Klientov, na ktorých beží aplikácia APP-EVAL aby tak bolo umožnené rozpoznanie naskenovaných testov a ich import do databázy.

**Skener:** zariadenie typu Skener, určené na skenovanie vypracovaných testov a ukladanie ich na lokálny disk KlientaX, kde ich spracúva inštancia aplikácie APP-EVAL.





## 5 Prototyp

Táto kapitola sa venuje vývoju prototypu. Bližšia pozornosť je venovaná opisu oblasti a jednotlivých cieľov prototypovania, spôsobu prototypovania systému a nakoniec dosiahnutým výsledkom. Prototyp bol vyvíjaný v druhej časti zimného semestra.

### 5.1 Oblasť prototypovania

Prototyp by mal vo všeobecnosti poskytnúť lepšiu predstavu o fungovaní systému pre zákazníka, nájsť chyby a nedostatky v špecifikácii a návrhu, ako aj experimentovať v oblastiach vývoja systému, ktorých realizácia ešte nie je celkom jednoznačná.

V priebehu analýzy a špecifikácie sme identifikovali niekoľko oblastí, ktoré sme sa rozhodli zahrnúť do prototypu systému. Vzhľadom na to, že tieto oblasti nie sú vzájomne previazané a odlišujú sa najmä spôsobom prototypovania (evolučné, experimentálne), bolo vytvorených niekoľko programov, pričom každý sa zameriava na určitú oblasť vývoja systému.

Prvou oblasťou prototypovania bolo používateľské rozhranie. Už pri analýze a špecifikácii požiadaviek vznikali nejasnosti v otázkach návrhu používateľského rozhrania a funkcií, ktoré má poskytovať používateľom. Ďalšou oblasťou prototypovania je dátový model. Je vhodné overiť, či je model údajov navrhnutý tak, aby umožňoval vykonávať funkcie systému uvedené v špecifikácii. Spomenuté dve oblasti používateľského rozhrania a dátového modelu sme sa rozhodli spojiť do jedného funkčného evolučného prototypu, ktorý by mal slúžiť ako základ pre ďalší vývoj systému.

Počas analýzy a návrhu vznikla potreba overiť možnosť implementácie niektorých špecifických funkcií, ktoré by mal systém poskytovať. Potrebné bolo otestovať možnosti rozpoznávania čiarových kódov, ktoré slúžia na identifikáciu testov. Podobne vznikla potreba overenia možnosti rozpoznávania testových odpovedí v podobe krížikov na formulári s odpoveďami. Takto vznikli dva experimentálne prototypy, určené na zahodenie, ktoré overili možnosti implementácie v danej oblasti.



Poslednou oblasťou prototypovania bola identifikácia možných používateľov v rámci lokálnej siete a zaradenie ich práv v rámci vyvíjaného systému. Tento modul bol vyvíjaný mimo evolučného prototypu, ale s možnosťou zakomponovania do cieľového systému.

Po výbere prostredia implementácie, ktoré je spoločné pre všetky prototypy, sa venujeme bližšiemu opisu jednotlivých prototypov samostatne, keďže každý sa zaoberá inou oblasťou vývoja systému.

## 5.2 Výber prostredia

Výber prostredia pre vývoj prototypov v značnej miere závisí na výbere prostredia pre implementáciu celého systému. Prostredie a použité technológie veľmi úzko súvisia s výsledkami analýz najmä experimentálnych prototypov. Analyzovali sa existujúce dostupné prostriedky pre vývoj prototypov v uvažovaných prostrediach.

Pri výbere implementačnej technológie sa do úvahy brali najmä objektovo orientované technológie vývoja softvérových systémov použitím jazyka C Sharp, Java alebo C++. V konečnom dôsledku bol zvolený jazyk C++ s použitím vývojového prostredia Visual C++ 6.0. Pre túto možnosť sme sa rozhodli z viacerých dôvodov:

- poskytuje jednoduchú implementáciu používateľského rozhrania pomocou knižnice MFC
- jednoduchosť inštalácie výsledného produktu, nepotrebuje žiadne virtuálne stroje ako napr. Java
- poskytuje vysokú rýchlosť implementovaného riešenia, pri vykonávaní frekventovaných operácií, ktorá je dôležitou požiadavkou zo strany zákazníka, napr. v porovnaní s Javou
- znalosť jazyka všetkými členmi tímu

Rozhodnutie v konečnej fáze záviselo aj od dostupných knižníc vhodných na použitie pri implementácii. Zvolený bol jazyk C++ a vývojové prostredie Visual C++ 6.0.

Druhým krokom bola voľba technológie implementácie databázy. Táto závisela najmä od dostupných prostriedkov pre nasadenie aplikácie. Zvolená bola databáza PostgreSQL. Hlavným dôvodom pre výber tohto riešenia je, že tento typ databázy je už na fakulte nasadený a má



zo strany fakulty technickú podporu. Databáza PostgreSQL je voľne dostupná a postačuje požiadavkám, kladeným na databázu vyvíjanou aplikáciou. Členovia tímu majú s databázou PostgreSQL skúsenosti.

### 5.3 Prototyp používateľského rozhrania a dátového modelu

Evolučný prototyp, ktorý bude slúžiť ako základ pre ďalší vývoj aplikácie.

#### 5.3.1 Ciele prototypovania

Vzhľadom na zvolenú oblasť prototypovania sme identifikovali základné ciele:

- implementácia používateľských rozhraní
- implementácia databázy podľa návrhu modelu údajov
- implementácia funkcií systému pre spoluprácu s databázou
- ukážka funkcionality systému previazaním zvolených používateľských obrazoviek s databázou

V rámci implementácie používateľských rozhraní by mali byť podrobne navrhnuté jednotlivé obrazovky systému a ich prvky. Určená by mala byť logická nadväznosť a previazanie jednotlivých formulárov alebo ich častí. Návrh a implementácia formulárov by sa mali zameriavať na možnosti rýchleho a čo možno najefektívnejšieho zadávania údajov so zaručením čo najmenej novej chybovosti. Prezentovaná by mala byť funkčnosť obrazoviek spolu s čiastočnou kontrolou vstupných údajov. Takto prezentované používateľské rozhranie poskytuje zákazníkovi reálnu predstavu o cieľovej podobe systému a funkcií, ktoré bude poskytovať.

Implementované by mali byť aj funkcie na spoluprácu s databázou, ktoré by umožňovali pohyb údajov medzi aplikáciou a databázou. Existencia týchto funkcií by v plnej miere umožnila prezentáciu funkcionality zvolených obrazoviek systému. Takto by bola overená správnosť návrhu modelu údajov a implementovaných funkcií.

#### 5.3.2 Spôsob prototypovania

Spôsob vývoja prototypu pozostával z troch hlavných častí a to:

- návrh používateľských obrazoviek



- návrh fyzického modelu údajov a implementácia databázy
- návrh architektúry aplikácie a funkcionality zvolených obrazoviek

Pri návrhu a implementácii používateľských obrazoviek bola použitá knižnica MFC. Použitá bola architektúra tvorby používateľského rozhrania SDI (Single Document Interface) pričom prvky na jednotlivých formulároch pre vstup a zobrazovanie údajov sú dynamicky generované. Pozornosť bola zameraná na formuláre zadávania údajov z prijímacej skúšky a osobných údajov uchádzačov. Testové odpovede z formulárov odovzdaných študentmi je potrebné čo najrýchlejšie spracovať. Preto boli testované rôzne formy zadávania testových odpovedí, tak aby bola zaručená čo najmenšia možná chybovosť. Možnosť výskytu chýb sa výrazne znížila zavedením vstupnej kontroly údajov. Implementované používateľské rozhrania a práca s nimi sú bližšie popísané v kapitole používateľskej príručky k prototypu.

Pri návrhu spôsobu zadávania správnych odpovedí sa uvažovalo nad rôznymi spôsobmi tak, aby bolo zadávanie čo najefektívnejšie:

- zadávanie pomocou čísla pol'a - každé vstupné políčko na formulári s odpoveďami by malo svoje číslo, teda čísla by boli v rozmedzí 1 až  $n*m$ , pričom  $n$  by bol počet otázok testu a  $m$  počet odpovedí pre každú otázku. Takto by sa zadávali postupne len čísla uchádzačom označených odpovedí. Tento spôsob je výhodný vzhľadom na nízku chybovosť a dobrú rýchlosť zadávania, avšak pri väčšom množstve otázok a možných odpovedí dochádza k označovaniu vstupných polí rádovo veľkými číslami, čím sa tento spôsob stáva neprehľadným
- zadávanie pomocou čísla možnosti v rámci otázky - do pol'a odpovedí pre jednotlivé otázky by sa zadávali odpovede uvedené uchádzačom. Pre každú otázku by teda existovalo vstupné pole odpovedí. Každá cifra v čísle v poli odpovedí by znamenala jednu odpoveď na otázku. V prípade, že odpovede na otázku sú označované písmenami, je zvolená ekvivalentná konverzia (1-a,2-b,3-c,atď.). Táto možnosť by poskytovala dostatočnú rýchlosť zadávania ako nízku chybovosť pri zadávaní
- zadávanie pomocou zaškrtnutia vstupného políčka myšou - toto riešenie zadávania je veľmi prehľadné, avšak nie je dostatočne efektívne a dochádza pri ňom k veľkej chybovosti, keďže si od používateľa vyžaduje citlivé ovládanie myši. Vhodná by bola jeho kombi-



nácia s niektorým z predchádzajúcich riešení napríklad na opravu vzniknutých chýb, teda zadávanie len malého množstva údajov

Testovaním všetkých spôsobom zadávania bola implementovaná kombinácia analyzovaných spôsobov. Prvá možnosť bola zamietnutá vzhľadom na veľký rozsah čísel pri označovaní vstupných políček pri veľkom počte otázok a možnosti odpovedí. Druhý spôsob bol rozšírený o rozdielne možnosti zadávania singlechoice a multichoice odpovedí. Spôsob zadávania pomocou označenia vstupného políčka myšou bol tiež implementovaný a je vhodný hlavne na použitie pri oprave konfliktov medzi odpoveďami pri zadávaní testov.

Vytvorená bola databáza podľa fyzického návrhu tabuliek. Implementované boli funkcie na prácu s databázou. Využitá bola existujúca knižnica libpq.dll, ktorá umožňuje jednoduchý spôsob komunikácie s databázou PostgreSQL.

Na overenie funkčnosti prepojenia dátového modelu s rozhraniami boli implementované niektoré funkcie formulárov. Príkladom je napĺňanie roliet s možnosťami údajov pri vytváraní nových šablón z číselníkov. Z číselníkov v databáze sa získavajú údaje ako predmet, časy a miesta konania skúšok, z ktorých používateľ vyberá pri vytváraní novej šablóny. Umožnené je aj uloženie vytvorenej šablóny do databázy. Ostatné funkcie neboli v prototypu implementované.

### 5.3.3 Výsledok prototypu

Implementovaný prototyp splnil v plnej miere ciele prototypovania. Navrhnuté a implementované boli obrazovky systému, ktoré utvrdili predstavu o budúcej kompletnej funkcionalite systému. Bol vytvorený efektívny spôsob interakcie s používateľom najmä pri zadávaní testových odpovedí, ktorý umožňuje rýchle zadávanie údajov pri nízkom výskyte chýb. Možnosť výskytu chýb sa výrazne znížila aj zavedením vstupnej kontroly údajov pri niektorých prvkoch formulárov. Bola implementovaná databáza podľa návrhu a do prototypu boli zahrnuté knižnice na prácu s databázou PostgreSQL. Tie umožnili overenie funkčnosti komunikácie s databázou v niektorých formulároch.



## 5.4 Prototyp rozpoznávania čiarových kódov

Experimentálny prototyp, určený na overenie možnosti rozpoznávania čiarových kódov a preskúmania vhodných riešení.

### 5.4.1 Ciele prototypovania

Vzhľadom na zvolenú oblasť prototypovania sme identifikovali základné ciele:

- zvoliť vhodný čiarový kód na identifikáciu testov
- implementovať algoritmus na rozpoznávanie zvoleného čiarového kódu z naskenovaného obrázku
- určiť spoľahlivosť zvoleného implementovaného riešenia

Analyzovať dostupné typy čiarových kódov a dostupné knižnice na ich spracovanie pre zvolené vývojové prostredie. Určiť kód vhodný na reprezentáciu údajov potrebných na identifikáciu testu. Následne je potrebné implementovať algoritmus na jeho rozpoznávanie z naskenovaného obrázku, pričom je potrebné uvažovať s nižšou kvalitou obrázku alebo prípadnými inými nedostatkami. Určiť pravdepodobnosť chyby pri rozpoznávaní čiarového kódu.

### 5.4.2 Spôsob prototypovania

V prvej časti prototypovania sme museli zvoliť vhodný čiarový kód. Tento bude kódovať nasledujúce informácie:

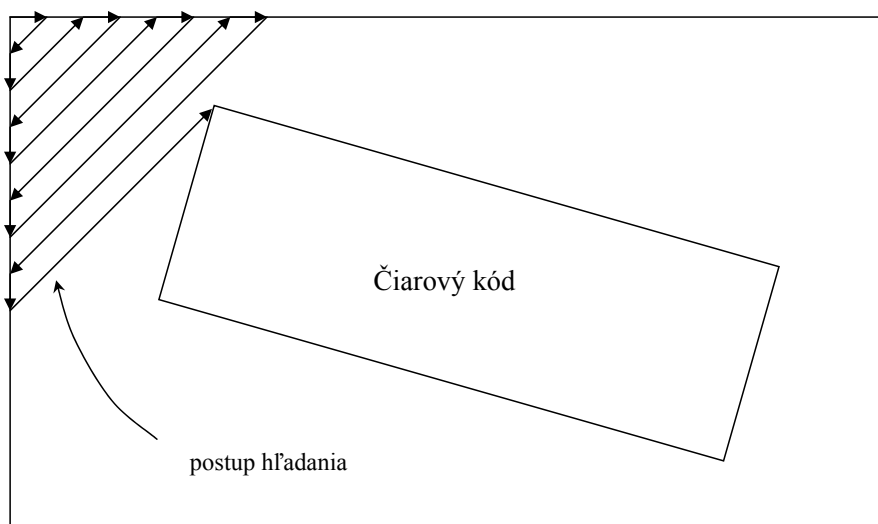
- 4 - ciferný identifikačný kód uchádzača
- 1 - ciferný kód identifikujúci predmet
- 1 - ciferný kód identifikujúci miestnosť
- 1 - ciferný kód identifikujúci beh

Na zápis týchto informácií by postačil prakticky hociktorý existujúci čiarový kód. A tak sme sa pri výbere sústredili hlavne na kódy umožňujúce jednoduchý zápis a čítanie. Nakoniec sme sa rozhodli pre kód Code39. Dôvodov bolo viacej:

- kód má variabilnú dĺžku, čo umožňuje budúce rozšírenie kódovanej informácie
- kód obsahuje iba čiary dvoch hrúbok, čo uľahčuje jeho načítavanie
- voľne dostupný font na zápis kódu



Načítavanie čiarového kódu pracuje v dvoch fázach. V prvej program hľadá na obrázku pozíciu kódu a v druhej prečíta nájdený kód. Hľadanie funguje na princípe hľadania rohov čiarového kódu, pričom sa využíva predpoklad, že na načítanom obrázku je iba samotný čiarový kód. Spôsob hľadania je uvedený na obrázku. Pri rozpoznávaní čiarového kódu sme vyskúšali dve metódy. Prvá metóda pracovala nasledovným spôsobom: Z načítaného čiarového kódu sa vytvorila jeho horizontálna projekcia a z nej sa pomocou prahovania načítali šírky jednotlivých čiar a medzier. Táto metóda ale bola veľmi citlivá na natočenie čiarového kódu a tak sme boli nútení túto metódu pozmeniť. Možností bolo viacero od rôznych šikmých projekcií cez natáčanie čiarového kódu až po emuláciu čítačiek kódov. My sme sa rozhodli pre poslednú. Cez čiarový kód postupne preložíme niekoľko náhodných čiar, načítame intenzitu bodov cez ktoré čiary prechádzajú a spriemerujeme ich. Ďalej sa už postupuje podobným spôsobom ako v predošlej metóde.



Obrázok 20: Príklad rozpoznávania čiarového kódu

### 5.4.3 Výsledok prototypu

Uvedená metóda poskytuje veľmi dobré výsledky a pri testoch sa jej podarilo správne načítať 100% predložených nepoškodených čiarových kódov. V budúcnosti sa podľa potreby bude metóda rozširovať o filtre korigujúce najbežnejšie poškodenia čiarových kódov.



## 5.5 Prototyp rozpoznávania testových odpovedí

Experimentálny prototyp na zahodenie, určený na overenie možnosti rozpoznávania testových odpovedí v podobe krížikov a preskúmania vhodných riešení.

### 5.5.1 Ciele prototypovania

Vzhľadom na zvolenú oblasť prototypovania sme identifikovali základné ciele:

- analyzovať metódy na rozpoznávanie krížikov ako testových odpovedí
- implementovať vhodný algoritmus na rozpoznávanie krížikov
- určiť spoľahlivosť zvoleného implementovaného riešenia

Cieľom je vytvoriť prototyp na zahodenie, ktorý by overil možnosti rozpoznávania testových odpovedí v podobe krížikov z naskenovaných formulárov. Jednu alebo viac metód je potrebné implementovať vo zvolenom vývojovom prostredí. Určiť, ktoré z metód sú prípustné z hľadiska percentuálnej úspešnosti správneho rozpoznania krížika.

### 5.5.2 Spôsob prototypovania

Načítavanie testových odpovedí pracuje v troch fázach. V prvej sa podobne ako pri načítavaní čiarového kódu hľadajú rohy matice s odpoveďovými políčkami. V druhej fáze sa nájdená matica natočí do vodorovnej polohy a hľadajú sa v nej jednotlivé odpoveďové políčka. Hľadanie prebieha na základe informácií o veľkosti matice a na základe analýzy horizontálnej a vertikálnej projekcie obrázka. V tretej fáze sa analyzuje jedno odpoveďové políčko. Úlohou programu je rozhodnúť čo sa v ňom nachádza, možnosti sú: Políčko je prázdne, krížik, políčko je začiernené.

### 5.5.3 Výsledok prototypu

Prvé dva kroky sme úspešne naimplementovali a otestovali. Identifikácia obsahu políčka zatiaľ nie je dokončená a práce na nej pokračujú. Úspešne sa nám darí odlíšiť prázdne políčko od začierneného, problémy sú ale s detekciou krížika.





## 5.6 Prototyp autentifikácie používateľov

Experimentálny prototyp určený na overenie spôsobu autentifikácie používateľov pri prihlasovaní do systému. Vhodné riešenie by malo byť zakomponované do systému.

### 5.6.1 Ciele prototypovania

Vzhľadom na zvolenú oblasť prototypovania sme identifikovali základné ciele:

- analyzovať spôsob autentifikácie používateľov

### 5.6.2 Spôsob prototypovania

Keďže na fakulte nie je možné autentifikovať používateľa oproti serveru LDAP, rozhodli sme sa využiť existujúci Samba server, ktorý implementuje protokoly SMB/CIFS, umožňujúce interoperabilitu medzi Unix/Linux servermi a Windows servermi/klientami.

Server Samba vo fakultnom prostredí funguje ako kontroler NT domény. Autentifikovať sa k NT doméne dá buď natívnym LogonUser API, ktorý operačný systém Windows poskytuje alebo pomocou API poskytovaným inými implementáciami. Iné existujúce implementácie sú:

- JCIFS - <http://jcifs.samba.org>, implementované v jazyku JAVA
- Samba - <http://samba.org>, bežiaci na operačných systémoch iných ako Windows

Voľbou jazyka C++ a prostredia operačného systému Windows nám zostáva využiť natívne LogonUser API. Ďalšie nájdené implementácie vždy vychádzali zo spomínaného natívneho, takže neposkytovali žiadnu výhodu (napr. NutValidatePassword, <http://www.mkssoftware.com>).

### Funkcia LogonUser

Táto funkcia sa pokúsi prihlásiť používateľa na lokálny počítač (počítač, z ktorého je táto funkcia zavolaná). Používateľa špecifikujeme prihlasovacím menom a doménou, autentifikujeme heslom. Ak funkcia skončí úspešne, získame token, ktorý reprezentuje prihláseného používateľa. Ak skončí neúspešne, bolo zadané zlé prihlasovacie meno alebo heslo.

```
BOOL LogonUser (  
    LPTSTR lpszUsername,  
    LPTSTR lpszDomain,  
    LPTSTR lpszPassword,
```



```
DWORD dwLogonType,  
DWORD dwLogonProvider,  
PHANDLE phToken  
);
```

*lpSzDomain* ak obsahuje reťazec ".", funkcia validuje používateľský účet použitím lokálnej databázy účtov (a nie pomocou domény NT)

*dwLogonType* použijeme hodnotu LOGON32\_LOGON\_INTERACTIVE

*dwLogonProvider* bude LOGON32\_PROVIDER\_DEFAULT

#### Podmienky použitia LogonUser API

1. Počítač s programom, používajúcim LogonUser API musí byť členom NT domény, oproti ktorej sa chceme autentifikovať.
2. Program používajúci LogonUser API musí byť spustený z procesu už majúceho privilégium "Act as Part of the Operating System". V opačnom prípade sa generuje chyba, oznamujúca nedostatočné oprávnenie. Normálne je toto privilégium dané iba účtu SYSTEM.

Ak chceme splniť druhú podmienku, je potrebné pridelit' oprávnenie "Act as Part of the Operating System" účtu, pod ktorým sa interaktívne prihlasujeme do systému (používateľskému účtu, z ktorého spúšťame program využívajúci LogonUser API, nie účet, na ktorý sa snažíme pomocou tohto API prihlásiť).

Na zmenu privilégií využijeme okno Start | Settings | Control Panel | Administrative Tools | Local Security Policy | Security Settings | Local Policies | User Rights Assignment. Nájďme oprávnenie "Act as Part of the Operating System" a pridáme používateľov, prípadne celé grupy, ktoré budú používať program, vyžadujúci LogonUser API. Ak sme pridali účet, pod ktorým práve pracujeme, je potrebné sa odhlásiť a znovu prihlásiť. Oprávnenia sa totiž zisťujú v čase prihlásenia a sú cachované LSA.

#### 5.6.3 Výsledok prototypu

Prototyp za splnenia podmienok uvedených vyššie je plne funkčný, autentifikuje používateľa oproti NT doméne. Jeho nasadenie, ako súčasť celej aplikácie bude potrebné schváliť a povoliť



bezpečnostným administrátorom fakultnej LAN. Pri voľbe programovacieho jazyka C++ a klientskeho operačného systému Windows, nie je možné použiť iné riešenie autentifikácie oproti NT doméne.

Za zmienku stojí možnosť nainštalovania servera LDAP na fakulte (ktorý tam momentálne nie je) a autentifikovať sa oproti tomuto LDAP serveru. Tento prístup by nevyžadoval špeciálne privilégia od systémového administrátora.

## 5.7 Zhodnotenie prototypovania

Prototypovanie v tejto fáze vývoja projektu pozostávalo z niekoľkých prototypov. Prototyp na overenie používateľského rozhrania umožnil overenie predstavy o obrazovkách aplikácie. Implementácia databázy a komunikácie s ňou umožnila prezentáciu funkčnosti niektorých častí rozhrania. Prototyp je teda použiteľný v ďalšom vývoji aplikácie a je možné rozširovať ho o ďalšiu funkcionálnosť.

Implementované boli aj experimentálne prototypy určené na overenie možností implementácie špecifických funkcií aplikácie. Prototyp na rozpoznávanie čiarových kódov dosiahol vysokú úspešnosť a implementácia bude použitá v riešení projektu. Prototyp rozpoznávania krížikov doposiaľ nepreukázal vhodné riešenie s dostatočnou mierou spoľahlivosti a práca na ňom bude pokračovať. Prototyp autentifikácie používateľov odhalil úskalia v použití autentifikácie využitím samba serveru. Jeho ďalšie použitie, vzhľadom na potrebu rozšírenia práv všetkých používateľov je zatiaľ otáznave.



## 6 Používateľská príručka (k prototypu)

Táto kapitola slúži ako používateľská príručka k prototypu k aplikácii EAPP-EVAL. Aplikácia slúži na zadávanie a vyhodnocovanie testov prijímacích skúšok na FIIT STU v Bratislave. Systém bol vyvíjaný v prostredí Microsoft Visual C++ 6.0, pre operačný systém Windows.

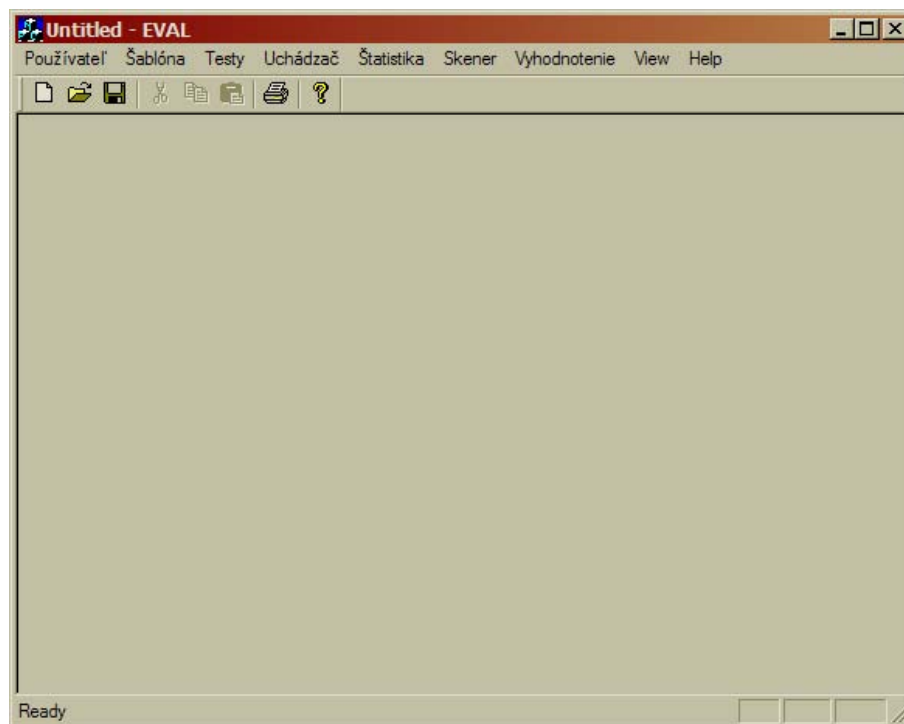
### 6.1 Používateľské rozhranie

V aplikácii je použité štandardné používateľské rozhranie, známe aj neskúseným používateľom z iných aplikácií pod systémom Windows. Rozhranie zabezpečuje jednoduchú interakciu používateľa s programom. Pozostáva z niekoľkých hlavných prvkov, ktoré je vhodné si opísať aj vzhľadom na lepšie porozumenie ďalších častí príručky:

- Aplikáčne okno - Hlavné okno programu, v ktorom beží aplikácia. Každá aplikácia má práve jedno také okno.
- Pruh Menu - Zoznam všetkých kategórií povelov pre aplikáciu.
- Menu - Zoznam povelov prislúchajúcich k jednej kategórii z pruhu menu.
- Modálny dialóg - Dialóg otvárajúci sa v novom okne aplikácie.
- Formulár - formulár otvárajúci sa v hlavnom okne aplikácie na pracovnej ploche pod Menu.
- Systémové Menu - povel pracujúce s oknom aplikácie, napríklad zmena veľkosti alebo zavretie okna.
- Titulný pruh - Pruh na hornom okraji okna, v ktorom je názov aplikácie.
- Tlačidlo minima a maxima - Tlačidlo minima sa používa na minimalizovanie aplikácie a tlačidlo maxima na zväčšenie aplikácie na celú pracovnú plochu.

Vzhľadom na väčšie množstvo prvkov v dialógových formulároch aplikácie je vhodné spúšťať aplikáciu maximalizovane, prípadne aplikáciu po spustení maximalizovať použitím tlačidiel.

V ďalších kapitolách sú podrobne rozpísané jednotlivé ponuky menu aplikácie, reprezentujúce logické časti funkcií poskytovaných systémom.



Obrázok 21: Príklad používateľskej obrazovky systému

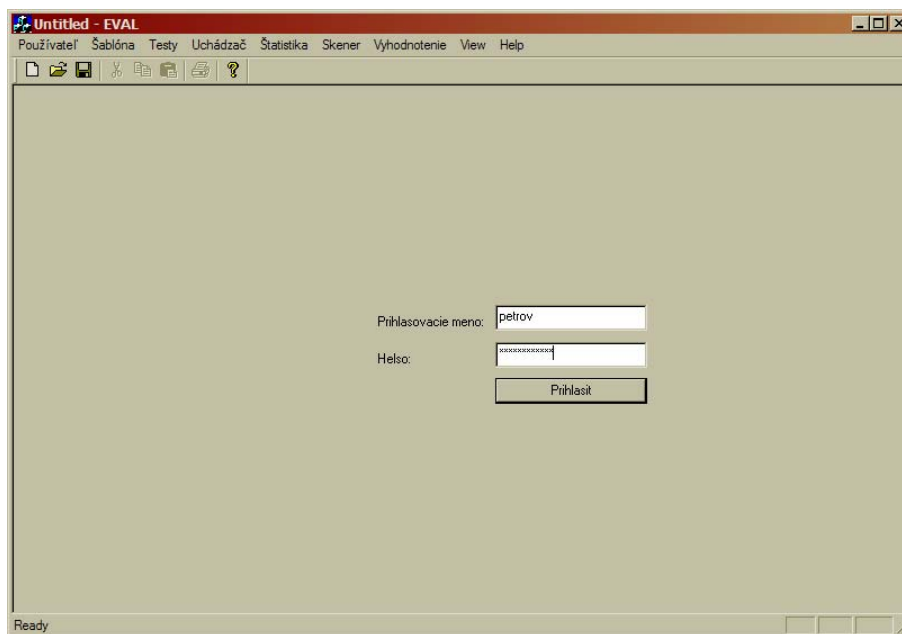
## 6.2 Menu EAPP-EVAL

Táto časť menu poskytuje povelý aplikácie na vyvolanie dialógov prihlásenia a odhlásenia zo systému ako aj nastavenia základných parametrov a ukončenia aplikácie.

### 6.2.1 Prihlásenie do aplikácie

Používateľia sa za účelom používania funkcií, ktoré aplikácia poskytuje, musia prihlásiť do systému.

V ponuke menu EAPP-EVAL povelom Prihlásiť sa zobrazí formulár požadujúci od používateľa prihlasovacie meno a heslo. Používateľ na prihlasovanie použije rovnaké údaje ako používa pri prihlasovaní do domény na počítačoch pripojených do fakultnej siete. Zadané údaje sú autentifikované a podľa prístupových práv, ktoré sú používateľovi v systéme priradené, sú mu sprístupnené príslušné časti systému.



Obrázok 22: Obrazovka prihlasovania sa do systému

### 6.2.2 Odhlásenie

Odhlásením používateľ a zo systému sa uzamknú funkcie systému prístupné len prihlásenému používateľovi. Aplikácia zostáva spustená. Prihlásiť sa následne môže znovu rovnaký alebo iný používateľ.

## 6.3 Menu šablóny

Menu šablóny poskytuje povely na otvorenie dialógov na prácu so šablónami testov, ktoré určujú štruktúru testov a správnych odpovedí. Menu zahŕňa funkcie vytvárania, editovania a mazania šablón testov. Toto menu je prístupné len používateľom s oprávnením typu Manažér.

### 6.3.1 Pridanie šablóny

Po zvolení povely Pridať z menu šablóny sa otvorí na pracovnej ploche aplikácie dialógový formulár vytvorenia novej šablóny.

Vytvorenie šablóny pozostáva z niekoľkých krokov, ktoré na seba veľmi tesne nadväzujú:

- Zadanie parametrov šablóny - zadávajú sa základné parametre pre vytvorenie šablóny
- Zadanie správnych odpovedí - pre každú otázku sa špecifikuje počet možných a správnych



odpovedí, ako aj bodové ohodnotenie

	1.	2.	3.	4.	5.	Odpovede	Body	Možnosti
1.	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	3	3	5
2.	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	1	1	5
3.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		1	2
4.	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	25	1	5
5.	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	34	1	5
6.	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	1	1	5
7.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		1	3
8.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		2	5
9.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		3	5
10.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		1	5
11.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		1	2
12.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		1	5
13.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		1	5
14.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		1	5
15.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		1	5

Obrázok 23: Obrazovka pridania novej šablóny

Skupina prvkov na formulári Parametre šablóny obsahuje zadanie hodnôt:

- Predmet - výber predmetu pomocou roletky. Predmety z ktorých sa vykonáva prijímacia skúška sú evidované v číselníku v databáze.
- Miestnosť - výber miestnosti konania prijímacej skúšky pre ktorú je vytváraná šablóna.
- Čas - čas konania prijímacej skúšky
- Variant - výber možných variantov
- Počet otázok - počet otázok testu
- Max. možností - maximálny počet možností pre každú otázku. Je možné znížiť pre každú otázku samostatne
- Body za otázku - štandardný počet bodov za otázku. Je možné zmeniť neskôr pre každú otázku samostatne
- SingleChoiceMultichoice - výber typu odpovedí na otázku, jedna/viac správnych odpovedí na otázku

Potvrdenie všetkých zadaných parametrov tlačidlom vytvor šablónu, vygeneruje v dialógu časť



na zadávanie odpovedí na jednotlivé otázky. Zobrazený je formulár s počtom otázok stanoveným v parametroch a so štandardným počtom možností tak ako boli zadané v predchádzajúcom kroku. Možné odpovede pre každú otázku sú reprezentované radom zaškrťovacích políčk. Správna možnosť odpovede na otázku je vyznačená zaškrtnutím príslušného políčka. V poli odpovedí sa je zároveň reprezentovaná číslom. Každá cifra v čísle v poli odpovedí znamená jednu správnu odpoveď na otázku. V prípade, že odpovede na otázku sú označované písmenami, je zvolená ekvivalentná konverzia (1-a,2-b,3-c,atď.).

Zadávanie správnych odpovedí je umožnené viacerými spôsobmi:

- vyznačením krížikov - používateľ vyznačí správne odpovede vyznačením krížiku v poli odpovedí pre príslušnú otázku. Zároveň sa číslo odpovede zobrazí v poli správnych odpovedí.
- zadaním čísla do pol'a odpovedí - tu sa dajú rozlišovať dva módy zadávania správnych odpovedí, Zadávací mód sa vyberá zaškrťovacím pol'om Multi mode:
  - single mód - pre každú otázku sa zadáva len jedna správna odpoveď. Po zadaní čísla správnej odpovede v rozmedzí možných odpovedí, je kurzor automaticky premiestnený na zadanie odpovede ďalšej otázky.
  - multi mód - dostupný len v prípade, že vytváraná šablóna, je typu multichoice, pre každú otázku sa zadávajú jedna alebo viaceré správne odpovede, po zadaní čísel správnych odpovedí v rade za sebou bez medzier, je stlačené predvolené tlačidlo na potvrdenie zadania všetkých správnych odpovedí pre danú otázku (pôvodne nastavená klávesa je 0)

Pre každú otázku je možné znížiť počet možných odpovedí a upraviť body pridelené danej otázke.

### 6.3.2 Výber šablóny

Zvolením povelu Vybrať z menu šablóny sa otvorí na pracovnej ploche aplikácie formulár na výber šablóny.

Zobrazené sú všetky šablóny uchovávané v databáze. Prehľadne v tabuľke sú uvedené ich identifikátory, predmet, miestnosť, čas a variant. Možné je zvoliť jednu zo šablón z tabuľky. Následne je možné upraviť alebo zmazať zvolenú šablónu použitím tlačidiel Upraviť a







## 6.4 Menu Testy

Menu šablóny poskytuje povel na otvorenie dialógu na zadávanie testov.

### 6.4.1 Zadávanie testov

Výberom povelu Zadávanie testov z menu Testy, sa zobrazí formulár na zadávanie testov. Slúži na manuálne zadávanie údajov z testov vypracovaných uchádzačmi. Predpokladá sa, že zadávateľ má k dispozícii fyzickú verziu testu a zadáva údaje podľa nej.

Proces zadávania testov rovnako ako formulár na zadávanie šablón pozostáva z dvoch krokov zadávania.

- zadanie čísla testu
- zadanie variantu testu
- zadanie odpovedí uchádzača

Po otvorení nového formuláru je používateľ vyzvaný na zadanie čísla testu. Po jeho zadaní sa overí či takéto číslo testu je platné. Podľa čísla sa identifikuje čas a miestnosť vykonávania testu. Zobrazí sa ponuka dostupných variantov testov. Variant testu je uvedený na odpoved'ovom formulári uchádzača. Používateľ vyberie variant testu a potvrdí výber. Systém vyberie údaje šablóny prislúchajúce zvolenému variantu testu a podľa nich vygeneruje formulár na zadávanie odpovedí pre jednotlivé otázky. Tento formulár je veľmi podobný tomu na zadávanie správnych odpovedí šablóny testov.

Zadávateľ môže využiť dva módy zadávania testu rovnako ako to je pri zadávaní šablón.

Po potvrdení zadaných odpovedí prebieha kontrola vstupných údajov. Ak bol už test s rovnakým číslom zadaný porovnávajú sa zadané údaje z oboch verzií. V prípade konfliktu je na to zadávateľ upozornený vysvietením konfliktných odpovedí. Má možnosť znovu overiť platnosť zadaných údajov a takto potvrdené údaje sú uložené. Uchovávané v systéme sú obe verzie zadania ale za platné sú považované tie čo boli zadané neskôr, vzhľadom na to, že tu bol zadávateľ upozornený na vzniknuté konflikty.



The screenshot shows a software window titled 'Untitled - EVAL' with a menu bar (Používateľ, Šablóna, Testy, Uchádzač, Štatistika, Skener, Vyhodnotenie, View, Help) and a toolbar. The main area is divided into several sections:

- Zadaj IDB číslo:** A text box containing '1234567' and an 'OK' button.
- Udaje o zadávanom teste:** A form with fields for TestNumber (1234), Predmet (Matematika), Miestnosť (AB 300), Čas (10.00 - 12.00), Počet otázok (15), Max. počet možností (5), Už zadaný (1 - krát), and MultiChoice (Áno).
- Mod:** A checkbox for 'MultiMod' which is currently unchecked.
- Buttons:** 'Pridať', 'Zrušiť zadávanie', and 'Vymazať'.
- Variant Selection:** A grid with 5 columns (1-5) and 15 rows. Each cell contains a checkbox and a text box for the answer. The grid shows the following selections:

Row	1	2	3	4	5	Answer
1.	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	1
2.	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	3
3.	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	1
4.	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	3
5.	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	1
6.	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	2
7.	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	3
8.	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	1
9.	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	2
10.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	4
11.	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	1
12.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	4
13.	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	1
14.	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	2
15.	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	2

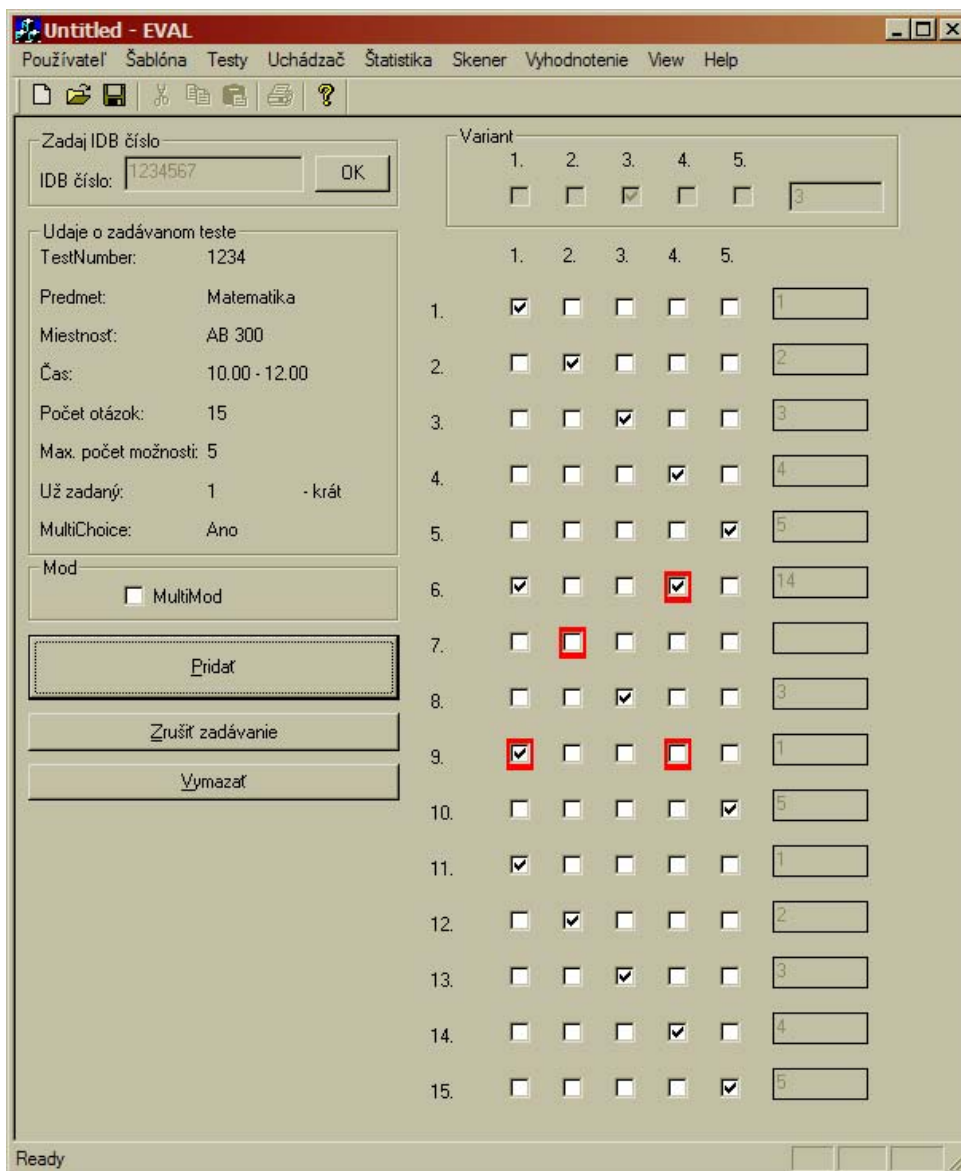
Obrázok 26: Obrazovka zadávania testu s odpoveďami

## 6.5 Menu Uchádzač

Menu Uchádzač poskytuje povely na otvorenie formulárov zadávania údajov o samotných uchádzačoch, ktorých zber je potrebný pri vyhodnocovaní prijímacej skúšky a ďalšom spracovaní údajov. Menu zahŕňa funkcie zadávania osobných údajov a priemerov vysvedčení.

### 6.5.1 Zadať osobné údaje

Výberom povelu Zadať osobné údaje z menu Uchádzač, sa zobrazí formulár na zadávanie a aktualizáciu osobných údajov o uchádzačovi. Tento zahŕňa:



Obrázok 27: Obrazovka opravy konfliktov pri zadávaní testu

- Meno
- Rodné priezvisko
- Priezvisko
- Telefónne číslo
- SMS notifikácia
- Číslo OP
- Ulica
- Mesto



- PSC

Uchádzač	
Meno:	Michal
Rodné priezvisko:	
Priezvisko:	Holečka
SMS notifikácia:	0904 123412
Telefónne číslo:	
Číslo OP:	DE 430 250
Ulica:	Pod Mostom 34
Mesto:	Bratislava
PSC:	04432

Obrázok 28: Obrazovka zadávania osobných údajov

Dôraz sa kladie zadávanie čísla SMS notifikácie keďže zmena ostatných údajov pri nie je až tak bežná. Po zobrazení formuláru sa požaduje zadanie identifikátora uchádzača, ktorý sa nachádza na formulári s osobnými údajmi. Po potvrdení sa overí platnosť čísla a zobrazia sa dostupné osobné údaje pre uchádzača. Kurzor sa presunie automaticky do pol'a zadávania SMS notifikácie. Po zadaní čísla SMS notifikácie, by mal zadávateľ overiť aj ostatné osobné údaje a v prípade zisteného rozdielu, ich aktualizovať. Potvrdením sa údaje uložia a otvorí sa nový formulár na zadávanie osobných údajov pričom kurzor je opäť v poli na zadanie identifikátora uchádzača. Takto je umožnené rýchle zadávanie osobných údajov uchádzačov, najmä v prípade telefónneho čísla SMS notifikácie.

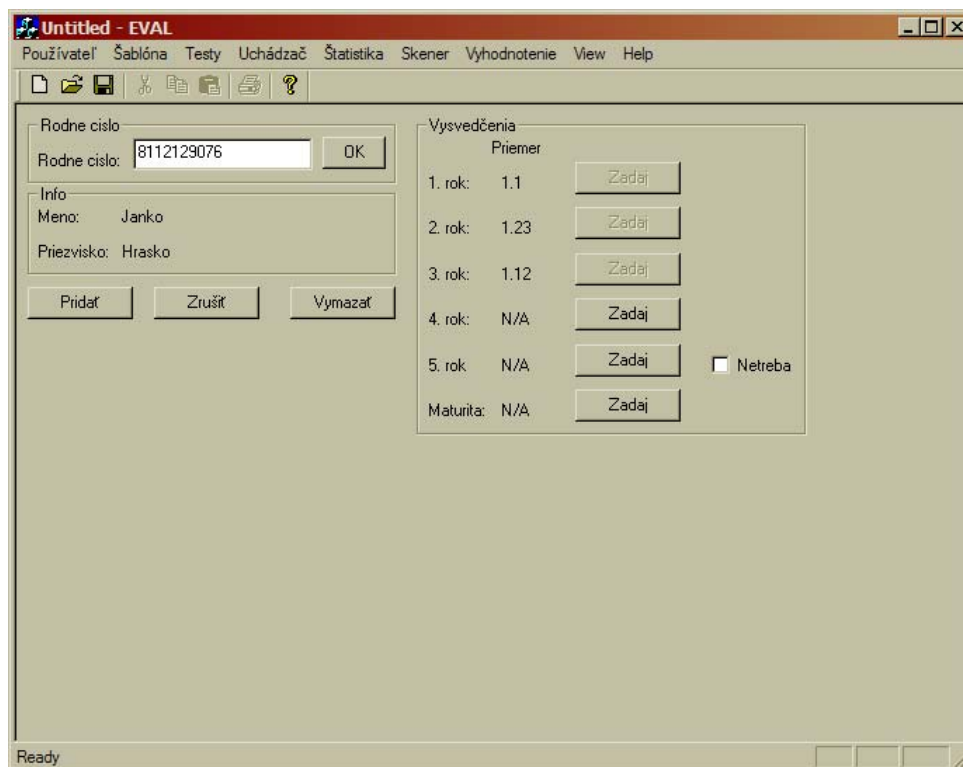
### 6.5.2 Zadat' vysvedčenia

Zvolením položky zadania vysvedčení z ponuky menu Uchádzač sa otvorí formulár na zadanie priemerov vysvedčení uchádzača. Predpokladá sa, že zadávateľ má fyzicky k dispozícii vysvedčenia uchádzača.

Jediným jednoznačným identifikátorom uchádzača na vysvedčení je rodné číslo. Po jeho zadaní do vstupného pol'a, je aktualizované okno formuláru, pričom sú zobrazené už zadané priemery známok za jednotlivé roky. Zadávateľ má možnosť zadať priemer známok za ktorý-



koľvek z ročníkov zvolením tlačidla Zadaj pri príslušnom ročníku.



Obrázok 29: Obrazovka zadávania priemerov vysvedčení

Následne sa objaví dialógové okno na zadanie známok z vysvedčenia. Dialógové okno je odlišné pre zadávanie vysvedčení za jednotlivé ročníky a pre maturitné vysvedčenie.

Pri zadávaní známok za ročník, je možné zadať ľubovoľný počet známok. Výber predmetov, ktorých známky je potrebné zadať je na zadávateľovi, ktorí by mal byť na túto úlohu patrične inštruovaný. Všetky známky majú rovnakú váhu pri výpočte priemeru. Potvrdením zadaných známok sa vypočíta ich priemer a ten sa zapíše do prehľadu priemerov.

Pri zadávaní známok z maturity je zadávanie výrazne odlišné. Určuje sa level maturitnej skúšky. Pre internú maturitu sa pre každý maturitný predmet určuje známka z písomnej časti a ústnej časti. Pre prípadnú externú maturitu sa určuje percentuálna úspešnosť pre každý z predmetov. Zo zadaných hodnôt sa vypočíta vážený priemer, ktorý určuje priemer maturitného vysvedčenia.



Zadávatel' pokiaľ vie určiť (podľa odovzdaných vysvedčení), či uchádzač odovzdal už všetky vysvedčenia teda minimálne za prvé štyri roky štúdia a maturitu, môže označiť vysvedčenie z piateho ročníka ako nepotrebné. Ak teda študent navštevoval štvorročnú strednú školu toto vysvedčenie od neho nie je už ďalej požadované.

## 6.6 Menu Štatistiky

Menu Štatistiky obsahuje povely na vyvolávanie formulárov prehľadov zadaných údajov. Používateľ má takto možnosť kontrolovať priebeh zadávania a vyhodnocovania údajov, prípadne ho prehľady upozorňujú na konflikty v zadaných údajoch.

Prehľady sú rozdelené do niekoľkých kategórií, každý z nich sa zameriava na jednu oblasť údajov. Poskytujú nástroje na filtrovanie a zoradenie záznamov v zobrazených prehľadoch.

### 6.6.1 Osobné údaje

Príkazom Osobné údaje v menu Štatistiky sa zobrazí formulár obsahujúci prehľad informácií o zadaných osobných údajoch. Zameriava sa na kontrolu zadávania telefónneho čísla a prípadných konfliktov medzi zadanými údajmi.

Priezvisko	Meno	Registračné číslo	Telefónne číslo	konflikt	Typ konfliktu
Ponozka	Dezider	1111113	0904546890	Ano	duplicitne tel. cislo
Brnák	Daniel	1111112	0904546890	Ano	duplicitne tel. cislo

Obrázok 30: Obrazovka prehľadu osobných údajov





Obsahuje filter, ktorý umožňuje obmedzenie položiek na základe výskytu reťazca. Ďalej usporiadanie položiek podľa Priezviska, Mena, Registračného čísla a Telefónneho čísla. Ďalej je dostupný aj filter konfliktov, teda zobrazuje položky všetky, konfliktné alebo len nekonfliktné.

Dvojitým kliknutím na záznam v tabuľke sa používateľovi otvorí formulár zadávania osobných údajov s už predvyplnenými dostupnými údajmi pre príslušného uchádzača.

### 6.6.2 Prospech

Zvolením príkazu Prospech v menu Štatistika sa zobrazí formulár obsahujúci prehľad informácií o vysvedčeniach zo strednej školy.

Priezvisko	Meno	Rodné číslo	Registračné číslo	1. ročník	2. ročník	3. ročník	4. ročník	5. ročník	maturita	konflikt	typ konfliktu
Štárek	Daniel	62030176590	1111112	1,2	1,2	1,20		1,20	1,25	ano	nezadany 4. ročník

Obrázok 31: Obrazovka prospechu jedného študenta

V tomto prehľade je pozornosť upriamená na vysvedčenia uchádzača a priemery známok z nich zadané do systému. Ako položky zoznamu sa zobrazujú uchádzači a k nim príslušné priemery známok za jednotlivé ročníky. Konflikt v tomto prípade môže nastať v troch prípadoch:

- pre uchádzača nebol zadaný priemer za 4.ročník
- pre uchádzača nebol zadaný priemer za 5.ročník
- pre uchádzača nebol zadaný priemer z maturitného vysvedčenia

Používateľovi je umožnené filtrovať medzi zobrazeniami jedného alebo všetkých konfliktov. A podobne ako v prípade všetkých prehľadov umožnené je vyhládavanie reťazca, rôzne spôsoby usporiadania v zozname a zobrazenie konfliktných, bezkonfliktných a všetkých záznamov.





Dvojitým kliknutím na záznam v tabuľke sa používateľovi otvorí formulár zadávania priemerov vysvedčení s už predvyplnenými dostupnými údajmi pre príslušného uchádzača.

### 6.6.3 Výsledky testov

Zvolením príkazu Prospech v menu Štatistika sa zobrazí formulár obsahujúci prehľad informácií o vysvedčeniach zo strednej školy.

Test ID	Matematika	Fyzika	Informatika	Konflikt	Typ konfliktu
1234	2x [42:42]	0x	0x	Áno	Zadany len jeden predmet

Obrázok 32: Obrazovka prehľadu výsledkov a spracovania testov

Prehľad poskytuje informáciu o testoch prijímacích skúšok. Zobrazené sú všetky identifikátory uchádzačov. Pre každého uchádzača je uvedený počet zadania testov a bodové ohodnotenie z jednotlivých predmetov.

Zobrazenie prehľadu sa čiastočne odlišuje pre používateľa typu manažér a zadávateľ. Zadávatel má v zozname výsledkov testov zobrazené len položky uchádzačov, ktoré sám zadával.



Pre manažéra sú zobrazené položky všetkých uchádzačov a je doplnený stĺpec identifikujúci zadávateľ a údajov.

Identifikované sú štyri situácie, kedy sú prvky zoznamu označené ako konflikty:

- pre uchádzača nebol zadaný test zo žiadneho predmetu
- pre uchádzača bol zadaný test len jedného predmetu
- pre uchádzača bol zadaný test z predmetu, z ktorého z ktorého prijímaciu skúšku nemal vykonávať
- zobrazí informáciu pri ktorých testoch došlo k oprave odpovedí pri viacnásobnom zadávaní, a aký je rozdiel v počte bodov

Dvojitým kliknutím na položku zoznamu sa otvorí formulár na riešenie príslušného druhu konfliktu. Pre prvé dva konflikty sa jedná o formulár manuálneho zadávania testov, kde môže byť opakovane zadaný test. Pre tretí konflikt sa otvorí formulár zobrazenia testu a je možné zmeniť predmet, pre ktorý bol test zadaný, prípadne vymazať záznam testu. Používateľovi je pri prehľade opäť umožnené filtrovať hodnoty v zozname na základe rôznych kritérií.

## 6.7 Menu Skener

Menu Skener poskytuje povel na otvorenie dialógu umožňujúceho rozpoznanie naskenovaných dokumentov odpoveďových formulárov a riešenie prípadných konfliktných situácií keď nie je možné niektoré údaje rozpoznať.

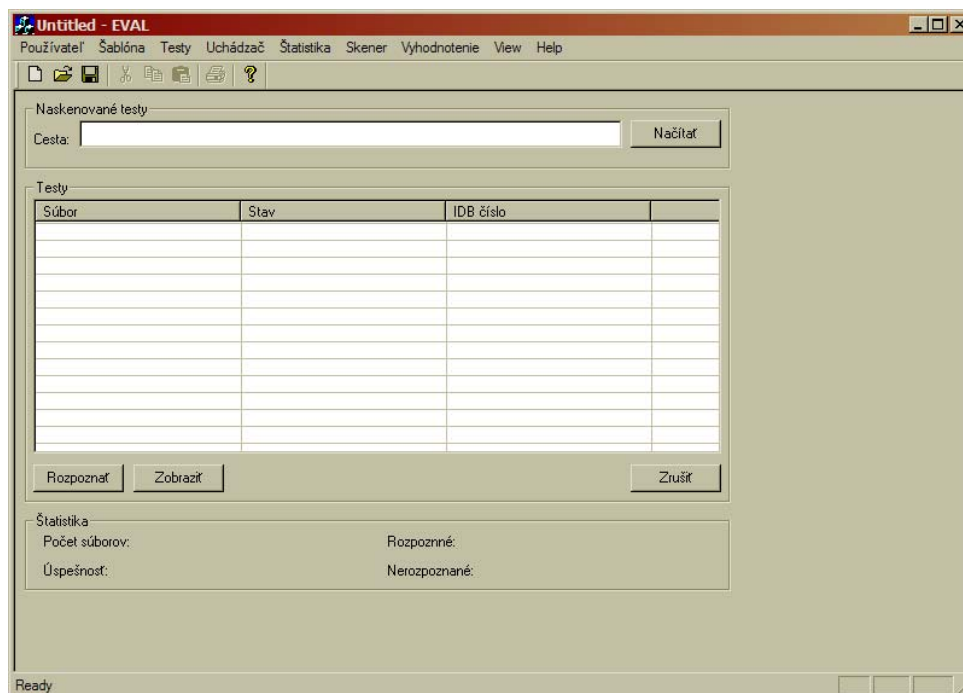
## 6.8 Rozpoznanie dokumentov

Výberom povelu Rozpoznania dokumentov z menu Skener, sa otvorí formulár umožňujúci riadenie rozpoznávania dokumentov.

Používateľ je vyzvaný zadať cestu k adresáru v ktorom sa nachádzajú súbory určené na rozpoznanie. Po určení cesty a stlačením tlačidla Načítať sa zobrazí zoznam súborov, ktoré sa nachádzajú v určenom adresári. Príkazom Rozpoznať, začne proces rozpoznávania súborov. V prípade, že súbor bol úspešne rozpoznaný, premenuje a presunie sa do podadresára ROZPOZNANE. V tabuľke súborov sa označí zelenou farbou a zobrazia sa aj niektoré z rozpoznávaných parametrov (napr. číslo testu). V prípade, že súbor nebolo možné rozpoznať, premenuje a



presunie sa do podadresára NEROZPOZNANE. Po prejdení všetkých súborov v hlavnom adresári a teda ukončení rozpoznávania sa zobrazí počet rozpoznaných, nerozpoznaných súborov a úspešnosť rozpoznávania.



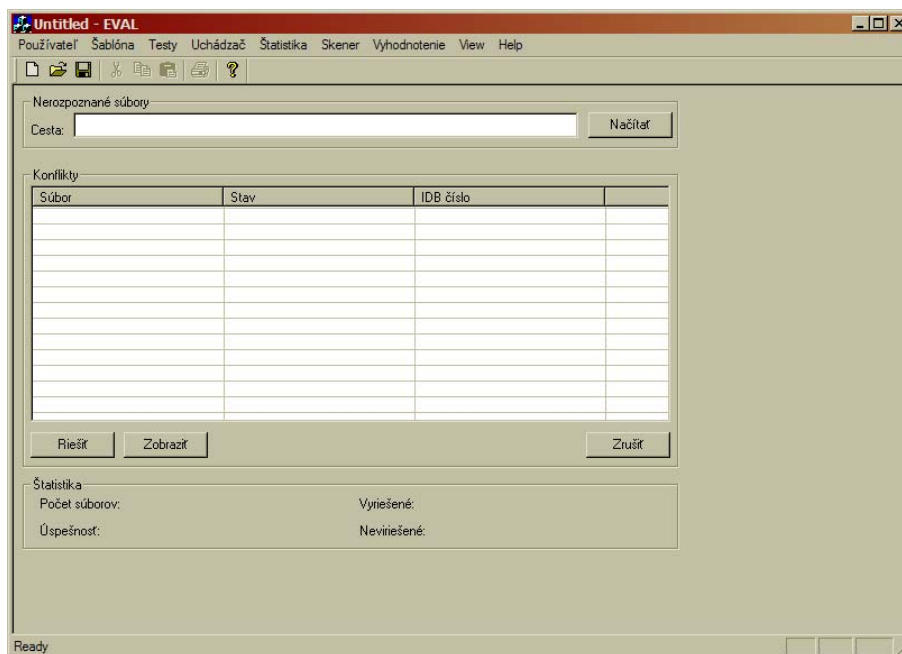
Obrázok 33: Obrazovka rozpoznávania údajov

Pre každý rozpoznaný súbor si môže používateľ zobrazit' rozpoznané parametre a obrázok súboru. Pre nerozpoznané súbory sa po príkaze Zobraziť zobrazí len samotný súbor.

Rozpoznané údaje zo súborov sa okamžite po rozpoznaní importujú do databázy. Nerozpoznané sa do databázy neimportujú. Preto je potrebné ich rozriešenie, ktoré sa vykonáva povelom Riešiť konflikty v menu Skener.

### 6.8.1 Riešenie konfliktov

Povel Riešiť konflikty z menu Skener zobrazí formulár na riešenie konfliktov v nerozpoznaných súboroch. Rovnako ako pri rozpoznávaní je potrebné určiť cestu k adresáru, v ktorom sú uložené súbory ktoré nebolo možné rozpoznať.



Obrázok 34: Príklad obrazovky riešenia konfliktov

Tieto sa zobrazia v zozname súborov. Pri rozpoznávaní každého súboru v prípade, že program nedokáže s dostatočnou mierou pravdepodobnosti rozpoznať údaj na obrázku, otvorí formulár manuálneho zadávania testov a v dialógovom okne zobrazí obrázok. Používateľ zadá údaj na základe obrázku a potvrdí. Program pokračuje v rozpoznávaní testu, a po jeho úplnom rozpoznaní prechádza na ďalší súbor v poradí. V prípade, že ani používateľ nedokáže rozpoznať údaj na obrázku, je tento súbor označený za nerozpoznaný.

Testy, ktoré boli úspešne rozpoznané, za pomoci používateľa sú po rozpoznaní označené v zozname súborov žltou farbou. Tie, ktoré nemohli byť rozpoznané ani po riešení používateľom sú označené červeným. Súbor, ktorý bol rozriešený bez asistencie používateľa sú označené zelenou farbou.

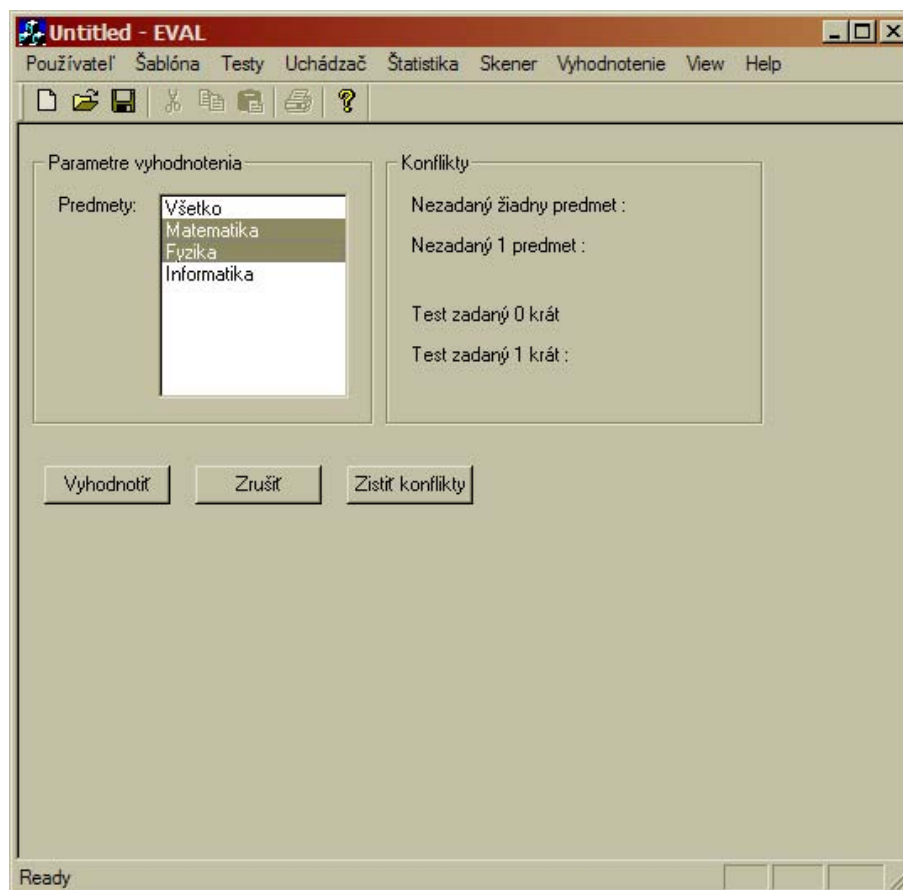
Pre každý rozpoznaný súbor si môže používateľ zobrazíť rozpoznané parametre a obrázok súboru. Pre nerozpoznané súbory sa po príkaze Zobrazíť zobrazí len samotný súbor.



## 6.9 Menu Vyhodnotenie

Menu Vyhodnotenie poskytuje povel na otvorenie formuláru vyhodnocovania zadaných testov.

Táto časť menu je prístupná len používateľom s oprávnením typu Manažér.



Obrázok 35: Príklad obrazovky vyhodnotenia testov

Testy je možné vyhodnocovať všetky naraz, prípadne rozdelené po predmetoch. Prípadné existujúce konflikty v údajoch sú zobrazované na pravej strane. Konflikty sú zisťované vždy pre novú kombináciu zadaných predmetov po stlačení tlačidla Zistiť konflikty. V prípade, že používateľ chce vyhodnotiť testy aj na napriek existujúcim prípadným konfliktom stlačí tlačidlo Vyhodnotiť. Všetky testy spĺňajúce zadanú podmienku sa vyhodnotia vzhľadom na existujúcu politiku vyhodnocovania.



## Referencie

- [1] Bieliková M.: Softvérové inžinierstvo. Princípy a manažment. Bratislava: STU, 2000.
- [2] Programming Windows Security - Keith Brown, June 30, 2000 (ISBN: 0201604426)
- [3] LogonUser API: <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/secauthn/security/logonuser.asp>



## Príloha A : Návrh formulárov prijímacej skúšky

### Analýza

Z analýzy rôznych variantov formulárov použitých na prijímacej skúške, sme izolovali dva rôzne pohľady na tieto formuláre, z ktorých každý dáva iné alternatívy z pohľadu organizácie priebehu skúšky.

Nasleduje **pohľad na párovanie** dokumentu osobných údajov a odpovedového formulára pomocou identifikačného čísla (ID) vo forme čiarového kódu. Ak uchádzač dostane čiarovým kódom dopredu označené formuláre odpovedí a osobných údajov, síce sa dozvie svoje ID, ale formulár osobných údajov už môže byť posunutý na spracovanie počas konania prijímacej skúšky. Ak ID bude nalepené na konci prijímacej skúšky, uchádzač sa ho nedozvie, ale vznikne organizačná potreba lepenia týchto ID čiarových kódov, ktorá bude vzhľadom na počet uchádzačov enormná.

*Záver pohľadu:* dopredné predtlačenie ID testu na odpovedový formulár a na formulár s osobnými údajmi.

Ak uchádzač znehodnotí svoj odpovedový hárok, bude mať možnosť na opravný formulár prepísať svoje odpovede, pričom ID bude napísané manuálne a neskôr zadané do systému.

Zaoberali sme sa i **pohľadom na test prijímacej skúšky** z konkrétneho predmetu, či ho bude mať uchádzač predgenerovaný, alebo určený priamo na skúške *polohou miesta, na ktorom bude sedieť*. Vzhľadom na priority projektu je potrebné sa sústrediť sa na druhú možnosť (test bude vybraný z N možných sád), pričom treba identifikovať zmeny, ktoré je potrebné vykonať, ak by sa v budúcnosti uvažovalo o generovaní unikátneho testu pre každého uchádzača.

Ak uchádzač dostane test s vyznačeným identifikátorom sady, bude ho musieť vyznačiť na svojom odpovedovom hárku vo forme krížika pri príslušnej sade. Je tiež žiadúce skontrolovať či uchádzač správne vyznačil identifikátor sady, z dôvodu možného odpísania *všetkých krížikov* od suseda. Obe tieto činnosti by odpadli, v prípade generovania testov.

*Záver pohľadu:* bude existovať niekoľko sád testov z konkrétneho predmetu a uchádzač bude musieť vyznačiť identifikátor sady na odpovedovom hárku.

### Návrh

Budeme disponovať týmito údajmi na jednotlivých formulároch.

1. Formulár s osobnými údajmi:
  - TXT - Registračné číslo, ID
  - FIELD – telefónne číslo
  - FIELD – osobné informácie (adresa, rodné číslo)
2. Test s otázkami:
  - TXT – predmet, sada, beh



- TXT\* - otázka
- 3. Formulár s odpoveďami
  - ČK (čiarový kód s číslom) – Identifikačný čiarový kód IDB (ID, predmet, beh, miestnosť)
  - TXT – predmet, beh, miestnosť
  - CHECKBOX – sada
  - CHECKBOX\* - odpoveď

Príklad týchto dokumentov je v ďalšej kapitole.

V prípade neskoršej požiadavky na generovanie testu pre každého uchádzača unikátne, z formulára 2 sa odstráni TXT sada a beh, pričom pribudne TXT ID; z formulára 3 sa odstráni CHECKBOX sada, TXT beh a miestnosť, z IDB ubudne 6. a 7. číslica vyjadrujúca beh a miestnosť.

### Príklad dokumentov prijímacej skúšky

Formulár osobných údajov

Registračné číslo: 1034
ID: 1321
Meno: Janko Hraško
Tel: <i>0905 234567</i>
Podpis: <i>Hraško</i>
Adresa: <i>Pod Mostom 6 84545 Bratislava</i>

Test

MATEMATIKA
Termín: 8.00
Skupina A
1. Koľko je $2 + 2$ ? a) 3 b) 4 c) 1 d) 5
2. Koľko je $3 + 3$ ? a) 6 b) 4 c) 5 d) 8

Formulár odpovedí

IDB: 1321 231
MATEMATIKA
Termín: 8.00
Miestnosť: AB150
A B C D
Skupina: <input checked="" type="checkbox"/> [ ] [ ] [ ] [ ]
a b c d
1. [ ] <input checked="" type="checkbox"/> [ ] [ ] [ ]
2. <input checked="" type="checkbox"/> [ ] [ ] [ ] [ ]

**Červené písmo:** Vyplní uchádzač o štúdium.

**Identifikačné číslo (ID)** je 4 – miestne, jednoznačne identifikujú uchádzača (bude spárované s registračným číslom). **Identifikačný čiarový kód (IDB)** je zakódované číslo, ktoré má 7 číslic. Prvé 4 číslice sú ID, 5. číslica identifikuje predmet, 6. číslica beh a 7. číslica vyjadruje miestnosť, v ktorej bude uchádzač písať prijímaciu skúšku. ID je potrebné tlačiť aj na dokument s osobnými údajmi, aby zamestnanec vedel spárovať dokument s osobnými údajmi a dokument určený na odpovede a dať ich do jednej obálky.