

# Specifikace zadání a uživatelských požadavků

Tvorba uživatelských rozhraní

# Individuální průzkum

Vojtěch Kališ, xkalis03@stud.fit.vutbr.cz

## Téma – mobilní aplikace Time Planner

Jakožto téma jsem se rozhodl vybrat nějakou aplikaci, jejíž funkcionality spočívá ve vytváření plánů, upomínek či jakéhokoliv jiného druhu poznámek za účelem zapsání si či připomenutí důležité události, schůze, nebo i trivialit jako co je potřeba koupit při příští návštěvě obchodu apod.; čili jakýkoliv například digitální kalendář, poznámkový blok a jiné aplikace umožňující výše popsané. Zároveň proběhla snaha najít takovou aplikaci, na níž byly na první pohled patrné chyby z pohledu GUI a uživatelského procesu, a to za účelem zjednodušení analýzy problémů dané aplikace a následného návrhu jejich řešení.

## Způsob zkoumání uživatele

Jako subjekt pro svůj uživatelský průzkum jsem si vybral svého bratra, jemuž jsem aplikaci na chvíli předal a následně jej podrobil několika předem přichystaným stručným otázkám, které měly za účel zjistit jeho pocity z aplikace, připomínky, požadavky a jiné.

## Požadavky uživatele

- Přehlednost
- Jednoduchost použití
- Není zapotřebí převeliké množství funkcí
- Možnost změny barevného scématu a pozadí
- Rychlá editace položek

## Způsob používání současné aplikace

Momentální aplikace je založena na třídění všech nových i již vytvořených úloh do skupin podle různých kondic, převážně pak tedy místa jejich vykonání, čili např. na poznámky vytvořené pro školu, práci, na doma, volný čas a tak dále. Postrádá možnost zobrazit je všechny chronologicky za sebou a tím zbytečně vytváří zmatek, což jde proti jejímu účelu. Ihned po prvotním spuštění aplikace na uživatele vyskočí několikakrokový tutoriál na vytvoření nové úlohy, načež je uživatel stejně zmaten co to vlastně vytvořil a proč, a poté navíc zjistí že jej čeká ještě zkouknout dalších 10 tutoriálů. Odhaduji, že právě v tenhle moment by to obvyklý uživatel vzdal a aplikaci zavřel, čemuž napovídá i nepěkné hodnocení aplikace.

## Identifikované potíže

- Aplikace má zbytečně příliš mnoho funkcí, což ji dělá nepřehlednou a komplikovanou

- Pouze dvě barevná schémata, skoro žádné možnosti kustomizace
- Zákazník očekává jednoduché řešení jednoduchého problému, ale čeká na něj složité monstrum

## Vlastní návrh nového řešení

Aplikace má v hlavním panelu nástrojů pouze tři hlavní funkce, a to konkrétně funkci pro vytvoření úlohy, zobrazení kalendáře s podrobnostmi k úlohám v následujících dnech a také možnost grafického přepnutí mezi více režimy zobrazení úloh, přesně v tomto pořadí. Vytvořené úlohy obsahují v hlavním menu pouze jejich jméno a čas konání, přičemž je možná každou úlohu rozkliknout, což otevře textový editor v němž je možné i doplnit přibližnější specifikaci k dané ploze. V pravém rohu položky je také tlačítko, které po rozkliknutí otevře jednoduché dropdown menu s možnostmi přejmenování položky, nastavení upomínky a nebo odstranění položky. V pravém horním rohu menu je tlačítko jež rozkliknutí otevře Nastavení, kde je dále umožněno například měnit barevné schéma aplikace výběrem z RGB palety, změnit používaný jazyk, nastavení zvuku apod.



# Individuální průzkum

Jan Lutonský, xluton02@stud.fit.vutbr.cz

## Téma – školní administrativní systém

Při hledání témat projektu jsem se ptal lidí v mém okolí jestli neznají nějaké aplikace, které by podle nich potřebovali zlepšit a každý kterého jsem se zeptal našel alespoň jednu aplikaci která by potřebovala redesign. Ze všech byla ale nejhorší aplikace pro správu informací o studentech, učitelích a majetku školy jedné nejmenované hudební školy. Proto jsem se rozhodl že se pokusím systém vylepšit ve svém projektu.

## Sledování uživatele

Sledoval jsem uživatele při práci s jejich dosavadním administrativním systémem, bylo zjevné že nejsou se systémem vůbec spokojeni. systém je příliš složitý pro běžnou kancelářskou práci.

## Uživatelé systému

Se systémem pracují dvě skupiny uživatel, administrativní pracovníci a učitelé, kteří ve škole učí. Administrativní pracovníci mají zkušenosti s prací na počítači a většinu dne tráví editací údajů v systému. Učitelé na druhou stranu jsou většinou starší lidé a mají problém při práci s počítačem proto si zvykli se systémem nepracovat vůbec a veškerou svou práci převedly na administrativní pracovníky. Učitelé ze systému potřebují číst informace o rozvrhu, seznamy žáků a kontaktní informace na rodiče a popřípadě zadávat známky ke konkrétnímu žákovi.

## Požadavky uživatele

- Snadná manipulace se systémem, eliminace manuálního vypisování známých informací
- Snadné vyhledávání informací a jejich filtrování
- Snadné tisknutí informací na tiskárně
- Implementovat historii úpravy systému, možná reverzace akcí
- Zjednodušené rozhraní pro učitele

## Způsob používání současné aplikace

Systém slouží pro vyhledávání studentů a modifikace informací o nich například školné, předměty které navštěvují, do jaké pobočky dochází, známky, atd.. A k generování informací nutných pro práci učitelů: rozvrhy, seznamy žáků, vysvědčení.

Sepsal jsem jednotlivé kroky které uživatel musí podstoupit při změně školného jednoho žáka.

### *Změna školného žáka*

1. uživatel musí stisknout tlačítko studenti a vyhledat jméno žáka
2. v seznamu stejných jmen najít správného žáka manuálním porovnáváním čísla studenta
3. po té co byl vybrán správný žák je nutno stisknout tlačítko upravit
4. dále je nutné manuálně vypočíst a manuálně vepsat školné, toto školné je možno vypočíst z informací uložených v systému automaticky
5. zmáčknout tlačítko uložit nebo tlačítko zavřít, tyto tlačítka provádí stejnou akci
6. zmáčknout tlačítko faktury u žáka a tlačítko opravit dále kliknout na tlačítko řádky
7. manuálně upravit řádek cena, stejná hodnota jako školné, které bylo vyplněno v kroku 4)
8. zmáčknout tlačítko uložit nebo tlačítko zavřít
9. stáhnout fakturu a odeslat mailem

### *Tento postup by mohl vypadat následovně*

1. zmáčknout tlačítko osoby
2. vyhledat podle čísla žáka konkrétního žáka
3. upravit informace o studiu, které automaticky změní cenu školného
4. zmáčknout tlačítko vygenerovat fakturu, které vygeneruje fakturu v pdf
5. odeslat pdf mailem

## Vlastní návrh nového řešení

Hlavním cílem je zjednodušit a zpříjemnit práci administracním pracovníkům a to tak že sjednotím podsystemy. Implementuji historii změn v systému aby bylo možné měnit chyby a vytvořit nový přívětivější podsystem pro učitele.

ID	Jméno	Příjmení	Datum Narození	Číslo studia
0	Adam	Ringer	06-06-2016	10

# Specifikace zadání a uživatelských požadavků

## téma Školní administrativní systém

### Důvod výběru tématu

Téma „Školní administrativní systém“ jsme se rozhodli vybrat hned z několika důvodů, a to větší škály možností v oblasti implementace různých vylepšení, zajímavosti daného tématu, a v neposlední řadě také jelikož druhé téma se nám v porovnání zdálo až příliš primitivní.

### Analýza uživatele

Uživatel aplikace je učitel nebo administrativní pracovník, který aplikaci používá k zobrazování a/nebo modifikaci informací o studentech, zaměstnancích školy, majetku školy a studijních programech.

### Potřeby uživatele

Uživatel potřebuje co nejjednodušší grafické rozhraní pro učitele a automatizaci pro administrativní pracovníky do níž by bylo možné vkládat informace a poté je nazpět číst. Preferována je funkčnost a přehlednost, ovšem nebylo by od věci zařídit i vizualisticky uspokojivý vzhled; ne však na úkor přehlednosti. Dále je kladen důraz na efektivní tisk informací a sdílení informací přes mail.

### Popis současného řešení

The screenshot displays a web-based administrative interface for a school. It is divided into several sections:

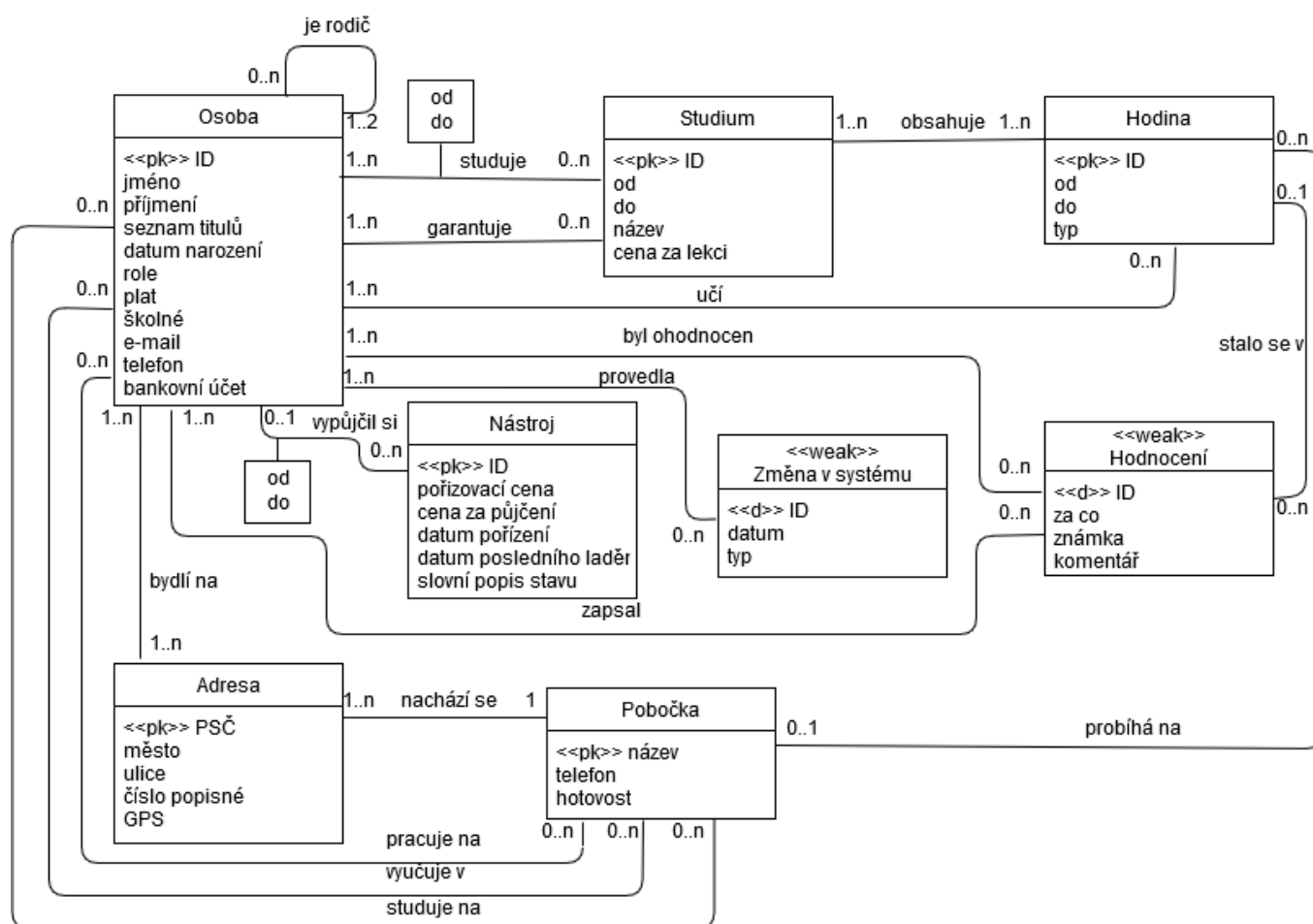
- Top Left:** A search bar labeled "hledat" and a list of students with their IDs and names.
- Top Right:** A student profile for "Student #3697" showing details like "Hra na klavír" and "Hudební nauka".
- Bottom Left:** A form for editing student information, including fields for "Identifikační údaje", "Kontaktní údaje", "Platební údaje", and "Adresa".
- Bottom Right:** A table showing the "Předměty" (Subjects) for the selected student, including "Hra na klavír" and "Hudební nauka".

## Návrh zadání

Proběhne snaha implementace sjednocení co nejvíce podsystémů do jednoho (ne však na úkor přehlednosti), automatizace vypisování již předem známých nebo vypočítatelných položek, jednoduššího a srozumitelnějšího vyhledávání žáků, snadněji navigovatelného uživatelského rozhraní, a v neposlední řadě také možnosti zobrazení historie změn a jejich případné reverzaci. Uživatelé tak bude umožněno systém využívat efektivněji, bude mít přístup k více možnostem načtení a editace informací, a k vykonání potřebných změn mu bude stačit značně menší počet manuálních kroků.

Konkrétní příklad úpravy a zjednodušení uživatelského procesu lze nalézt v Individuálním průzkumu tohoto tématu, strana 2, sekce Způsob používání současné aplikace. Je zde patrná snaha o zmenšení počtu nutně vykonávaných kroků ze strany uživatele pro zařízení jednoho konkrétního výstupu.

## Návrh řešení (předběžný)



# API

## Popis rozhraní systému

### 1 Vstupní funkce

#### 1.1 add\_person

funkce přijme informace o osobě zabalené do objektu a pokusí se je vložit do systému.

**vstupy:** objekt popisující osobu

**výstupy:** *bool* – **true** povedlo se přidat  
– **false** nepovedlo se přidat

**Možné chyby:** chyba databáze  
nekorektní informace o uživateli

#### 1.2 add\_instrument

funkce přijme informace o hudebním nástroji zabalené do objektu a pokusí se je vložit do systému.

**vstupy:** objekt popisující hudební nástroj

**výstupy:** *bool* – **true** povedlo se přidat  
– **false** nepovedlo se přidat

**Možné chyby:** chyba databáze  
nekorektní informace o hudebním nástroji

#### 1.3 add\_studium

funkce se pokusí přidat nové studium popsané objektem do systému.

**vstupy:** objekt popisující studium

**výstupy:** *bool* – **true** povedlo se přidat  
– **false** nepovedlo se přidat

**Možné chyby:** chyba databáze  
nekorektní informace o studiu

#### 1.4 add\_address

funkce se pokusí přidat novou adresu do systému.

**vstupy:** objekt popisující adresu

**výstupy:** *bool* – **true** povedlo se přidat  
– **false** nepovedlo se přidat

**Možné chyby:** chyba databáze  
nekorektní informace o adrese

#### 1.5 add\_branch\_office

funkce se pokusí přidat novou pobočku do systému.

**vstupy:** objekt popisující pobočku

**výstupy:** *bool* – **true** povedlo se přidat  
– **false** nepovedlo se přidat

**Možné chyby:** chyba databáze  
nekorektní informace o pobočce

#### 1.6 person\_rent\_instrument

vytvoří záznam o půjčení hudebního nástroje osobou.

**vstupy:** objekt popisující osobu  
objekt popisující hudební nástroj  
objekt popisující půjčku

**výstupy:** *bool* – **true** povedlo se přidat  
– **false** nepovedlo se přidat

**Možné chyby:** chyba databáze  
nekorektní informace o vstupu

#### 1.7 rate\_student

vytvoří hodnocení studenta učitelem.

**vstupy:** objekt popisující osobu student  
objekt popisující osobu učitel  
objekt popisující hodnocení studenta

**výstupy:** *bool* – **true** povedlo se přidat  
– **false** nepovedlo se přidat

**Možné chyby:** chyba databáze  
objekt studenta není student  
objekt učitele není učitel  
chybné vstupní informace

### 2 Nastavovací funkce

#### 2.1 studium\_set\_students

přidělí množinu žáků ke studiu.

**vstupy:** množina žáků  
množina žáků  
množina studií

**výstupy:** *bool* – **true** povedlo se  
– **false** nepovedlo se

**Možné chyby:** prázdná množina žáků nebo studií

#### 2.2 studium\_set\_lectures

přidělí dané vyučovací hodiny k danému studiu.

**vstupy:** množina vyučovacích hodin  
množina studií

**výstupy:** *bool* – **true** povedlo se  
– **false** nepovedlo se

**Možné chyby:** prázdná množina hodin nebo studií

## 3 Aktualizační funkce

### 3.1 end\_rent\_instrument

ukončí vypůjčení hudebního nástroje.

**vstupy:** záznam o půjčení nástroje  
stav nástroje

**výstupy:** *bool* – **true** povedlo se  
– **false** nepovedlo se

**Možné chyby:** chyba databáze  
prázdná množina hodin nebo studií

### 3.2 update

přepočítá všechny hodnoty v databázi, co dává smysl přepočítat.

**vstupy:** –

**výstupy:** *bool* – **true** povedlo se  
– **false** nepovedlo se

**Možné chyby:** chyba databáze

## 4 Tvořící funkce

### 4.1 generate\_classification

vygeneruje vysvědčení pro skupinu žáků z jednoho studia a uloží je jako PDF.

**vstupy:** množina studentů  
studium

**výstupy:** *bool* – **true** povedlo se  
– **false** nepovedlo se

**Možné chyby:** chyba databáze  
jedna nebo více osob z množiny studentů není student  
jedna nebo více osob z množiny studentů není zapsána na dané studium  
chyba převodu informací do pdf

### 4.2 generate\_tuition\_bill

vygeneruje fakturu za školné pro množinu žáků a uloží je jako PDF.

**vstupy:** množina žáků  
studium

**výstupy:** *bool* – **true** povedlo se  
– **false** nepovedlo se

**Možné chyby:** chyba databáze  
jedna nebo více osob z množiny studentů není student  
chyba převodu informací do pdf

### 4.3 generate\_salaries

vygeneruje výplaty a uloží je jako PDF.

**vstupy:** množina zaměstnanců  
studium

**výstupy:** *bool* – **true** povedlo se  
– **false** nepovedlo se

**Možné chyby:** chyba databáze  
jedna nebo více osob z množiny zaměstnanců není zaměstnanec  
chyba při vytváření PDF

### 4.4 generate\_schedule

vygeneruje rozvrh studentů jako PDF.

**vstupy:** množina studentů  
studium

**výstupy:** *bool* – **true** povedlo se  
– **false** nepovedlo se

**Možné chyby:** chyba databáze  
jedna nebo více osob z množiny studentů není student  
chyba při vytváření PDF

## 5 Odstraňovací funkce

### 5.1 remove\_person

odstraní záznam o osobě ze systému.

**vstupy:** množina osob

**výstupy:** *bool* – **true** povedlo se  
– **false** nepovedlo se

**Možné chyby:** chyba databáze

### 5.2 remove\_instrument

odstraní záznam o hudebním nástroji ze systému ze systému.

**vstupy:** množina nástrojů

**výstupy:** *bool* – **true** povedlo se  
– **false** nepovedlo se

**Možné chyby:** chyba databáze

### 5.3 remove\_studium

odstraní záznam o studiu ze systému ze systému.

**vstupy:** množina nástrojů

**výstupy:** *bool* – **true** povedlo se  
– **false** nepovedlo se

**Možné chyby:** chyba databáze



## 5.4 remove\_addres

odstraní záznam o adrese ze systému ze systému.

vstupy: množina nástrojů

výstupy: *bool* – **true** povedlo se  
– **false** nepovedlo se

Možné chyby: chyba databáze

## 5.5 remove\_branch\_office

odstraní záznam o pobočce ze systému ze systému.

vstupy: množina nástrojů

výstupy: *bool* – **true** povedlo se  
– **false** nepovedlo se

Možné chyby: chyba databáze

# 6 práce s historií

## 6.1 get\_history

vrátí seznam změn podle filtru.

vstupy: filtr

výstupy: množina změn

Možné chyby: chyba databáze  
špatný filtr

## 6.2 try\_undo

Pokusí se o odělení změny

vstupy: změna

výstupy: *bool* – **true** povedlo se  
– **false** nepovedlo se

Možné chyby: chyba databáze  
změnu nelze vrátit

# 7 získávání informací

## 7.1 quarry\_person

Vyhledá osobu v databázi.

vstupy: filtr

výstupy: množina osob

Možné chyby: chyba databáze

## 7.2 quarry\_instrument

Vyhledá hudební nástroje v databázi .

vstupy: filtr

výstupy: množina hudebních nástrojů

Možné chyby: chyba databáze

## 7.3 quarry\_studium

Vyhledá studium v databázi.

vstupy: filtr

výstupy: množina studií

Možné chyby: chyba databáze

## 7.4 quarry\_addres

Vyhledá adresu v databázi.

vstupy: filtr

výstupy: množina adres

Možné chyby: chyba databáze

## 7.5 quarry\_branch\_office

Vyhledá pobočku v databázi.

vstupy: filtr

výstupy: množina poboček

Možné chyby: chyba databáze