

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

UŽIVATELSKÉ ROZHRANÍ ŠKOLNÍHO INFORMAČNÍHO SYSTÉMU

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE
AUTHOR

JAN BUREŠ

BRNO 2013



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

UŽIVATELSKÉ ROZHRANÍ ŠKOLNÍHO INFORMAČNÍHO SYSTÉMU

USER INTERFACE OF SCHOOL INFORMATION SYSTEM

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

JAN BUREŠ

VEDOUCÍ PRÁCE

SUPERVISOR

Prof. Dr. Ing. PAVEL ZEMČÍK

BRNO 2013

Abstrakt

Tato bakalářská práce se věnuje návrhu uživatelského rozhraní školního informačního systému. Systém je určen především pro učitele a základní agendu. Cílem systému je ulehčit běžnou práci učitelů v hodině, protože dostupné informační systémy jsou hodně rozsáhlé a komplikované. V navrženém systému jsou dostupné funkce jako zápis do třídní knihy, zápis absence a zadávání klasifikace. Uživatelské rozhraní je navrženo ve stylu nového moderního vzhledu Windows 8. V práci je popsán proces návrhu a testování tohoto uživatelského rozhraní.

Abstract

This bachelor thesis is focused on design of user interface in school information system. This system is primarily designed for teachers and basic agenda. The purpose of this system is to simplify teacher's daily work, because the other available information systems are much more extensive and complicated. In aforementioned system are several functions, as a record in to the class books, record of a student absence and classification input. The said user interface is designed in the style of a modern appearance of Windows 8. In this document is contains description of a design and needed testing of user interface.

Klíčová slova

Uživatelské rozhraní, škola, informační systém, Windows 8.

Keywords

User interface, school, information system, Windows 8.

Citace

Jan Bureš: Uživatelské rozhraní školního informačního systému, bakalářská práce, Brno, FIT VUT v Brně, 2013

Uživatelské rozhraní školního informačního systému

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením Prof. Dr. Ing. Pavla Zemčíka. Další informace mi poskytli zaměstnanci firmy MP-Soft, a.s. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

.....

Jan Bureš
13. května 2013

Poděkování

Velké poděkování patří zaměstnancům firmy MP-Soft, a.s. - Ing. Jiřímu Soukenkovi, Ing. Radomilovi Maškovi a panu Tomáši Borkovi za vytvoření tématu bakalářské práce, odborné konzultace a praktické informace ze školního prostředí. Dále bych chtěl poděkovat Prof. Dr. Ing. Pavlu Zemčíkovi za užitečné rady a vedení bakalářské práce.

© Jan Bureš, 2013.

Tato práce vznikla jako školní dílo na Vysokém učení technickém v Brně, Fakultě informačních technologií. Práce je chráněna autorským zákonem a její užití bez udělení oprávnění autorem je nezákonné, s výjimkou zákonem definovaných případů.

Obsah

1 Úvod	2
2 Proces tvorby uživatelského rozhraní	3
2.1 Průběh tvorby uživatelského rozhraní	3
2.2 Typy uživatelských rozhraní	4
2.3 Grafické uživatelské rozhraní	4
2.4 Analýza uživatelského rozhraní	6
2.5 Návrh uživatelského rozhraní	8
2.6 Testování uživatelského rozhraní	9
2.7 Zásady tvorby uživatelského rozhraní Windows 8	10
3 Současné systémy pro správu školní matriky	12
3.1 SAS – Systém agend pro školy	12
3.2 Bakaláři	13
3.3 iŠkola	14
4 Specifikace aplikace	16
4.1 Počáteční požadavky na aplikaci	16
4.2 Požadavky vycházející z porovnání konkurence	17
5 Návrh uživatelského rozhraní školního informačního systému	18
5.1 Analýza školního prostředí	18
5.2 Návrh uživatelského rozhraní školního systému	19
5.3 Implementace ukázky uživatelského rozhraní	23
5.4 Testování uživatelského rozhraní školního systému	25
6 Závěr	26
A Finální verze volba hodiny	29
B Finální verze absence	30
C Finální verze klasifikace	31
D Obsah CD	32

Kapitola 1

Úvod

V současnosti jsou počítače rozšířeny do každé oblasti lidské působnosti. Člověk tedy přichází do styku s počítačovými aplikacemi denně ve své práci nebo ve svém volném čase. Každá počítačová aplikace komunikuje s člověkem pomocí uživatelského rozhraní. V dnešní době se tak stává, že kvalita uživatelského rozhraní je nejdůležitějším kritériem při hodnocení aplikace a závisí na něm úspěch či neúspěch celého jejího vývoje.

Tato práce obsahuje poznatky týkající se tvorby uživatelského rozhraní. Informace jsou uvedeny z pohledu člověka, který vytváří uživatelské rozhraní. Nejedná se o kompletní rozbor problematiky, ale pouze o souhrn informací, které jsem využil při návrhu aplikace pro vedení školních agend pro střední a základní školy. Práce také popisuje postup při návrhu uživatelského rozhraní této aplikace.

Téma jsem si vybral během mého působení ve společnosti, která vyvíjí systém pro správu školní agendy. Tento systém je určen jak pro základní školy, tak pro školy střední. Jejich aplikace pokrývá veškerou potřebu škol od klasifikace až po správu majetku. Velký rozsah funkcí systému se projevuje i na rozsáhlosti uživatelského rozhraní. Z tohoto důvodu se v něm mnoho uživatelů špatně orientuje a má různé problémy s jeho ovládáním. Běžný učitel potřebuje pro svou práci pouze zlomek nabízených funkcí. Proto jsem se rozhodl navrhnout uživatelské rozhraní pouze pro základní potřeby učitelů. Kompletní nabídka funkcí bude dostupná v původní aplikaci.

Mnou vytvořené uživatelské rozhraní bude sloužit především učitelům při zadávání klasifikace, absence a pro zápis do třídní knihy přímo v hodině, případně během volného času ve škole. Méně často využívané funkce, jako tvorba rozvrhu, tisk vysvědčení, evidence majetku a jiné, budou implementovány pouze v původním rozsáhlém systému. Uživatelské rozhraní by také mělo být spustitelné na různých zařízeních.

V následující kapitole číslo 2 se nachází teoretický přehled problematiky tvorby uživatelských rozhraní. Kromě vysvětlení pojmu uživatelské rozhraní jsou zde uvedeny informace, jak postupovat při jeho návrhu. V závěru kapitoly jsou vyjmenovány zásady pro tvorbu uživatelského rozhraní Windows 8. Kapitola 3 se zabývá porovnáním současných systémů pro správu školní agendy. Následuje rozpis požadavků na výslednou aplikaci v kapitole 4. Požadavky jsou rozděleny na dvě části. První částí je zadání od firmy MP-Soft, a.s. Druhou částí pak nároky na aplikaci na základě hodnocení podobných systémů. V 5. kapitole se nachází popis návrhu uživatelského rozhraní aplikace pro učitele. Dále jsou zde uvedeny informace ohledně analýzy, návrhu, implementace a testování kontrétního uživatelského rozhraní. Dále pak porovnání navrženého systému s již existujícími systémy. Poslední 6. kapitolou práce je závěr, ve kterém se nachází shrnutí dosažených výsledků a nápady na možné vylepšení aplikace.

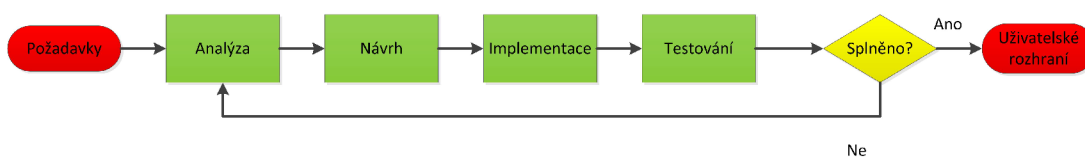
Kapitola 2

Proces tvorby uživatelského rozhraní

Kapitola stručně popisuje proces vytváření uživatelského rozhraní. Cíl tohoto procesu je většinou stejný – spokojený uživatel [1]. V kapitole jsou uvedeny informace, které potřebuje znát návrhář, aby se mohl tomuto cíli co nejvíce přiblížit. Na začátku jsou vyjmenovány jednotlivé části tvorby uživatelského rozhraní. Následuje vyjmenování typů uživatelských rozhraní a jejich výhody a nevýhody. Dále se kapitola věnuje pouze grafickému uživatelskému rozhraní a jednotlivým částem jeho vytváření. V závěru jsou pak uvedeny doporučení a pravidla pro tvorbu uživatelsky přívětivého rozhraní ve stylu Windows 8.

2.1 Průběh tvorby uživatelského rozhraní

Celý proces návrhu má několik společných bodů. Vstupem procesu jsou požadavky na výslednou aplikaci a výstupem je uživatelské rozhraní. Vlastní tvorba uživatelského rozhraní se skládá z analýzy, návrhu, implementace a testování. Po analýze následuje návrh. Z navrženého uživatelského rozhraní je vytvořen prototyp, který testují uživatelé. Zjištěné poznatky se přenesou do nového cyklu návrhu. Tento postup je znázorněn graficky na obrázku 2.1.



Obrázek 2.1: Diagram vývoje uživatelského rozhraní

Protože je velmi těžké určit, kdy je uživatelské rozhraní hotové, je potřeba na začátku procesu stanovit metodiku, jak změřit úspěšnost navrženého rozhraní [2]. Tato metodika je většinou založena na některých měřitelných faktorech. Jedná se buď o objektivní veličiny (chybovost, rychlost apod.) nebo subjektivní pocity uživatelů. Výsledné rozhraní se také může porovnat s již hotovou stejně zaměřenou aplikací (počet kliků k dosažení určitého cíle apod.) [3].

2.2 Typy uživatelských rozhraní

Dnes jsou počítače součástí každé oblasti lidské činnosti [4]. Lidé s počítačem komunikují téměř denně, protože jej využívají ke své práci. Komunikace člověka s počítačem je interaktivní [1]. Uživatel prostřednictvím ovládacích prvků definuje vstup a počítač na základě vstupu vytvoří výstup, který prezentuje uživateli [5]. V počátcích výpočetní techniky se jednalo o různé přepínače a tlačítka na straně vstupu a o světelné signály na straně výstupu [4]. V současnosti existuje několik typů uživatelských rozhraní. Nejčastěji jsou to tyto tři typy:

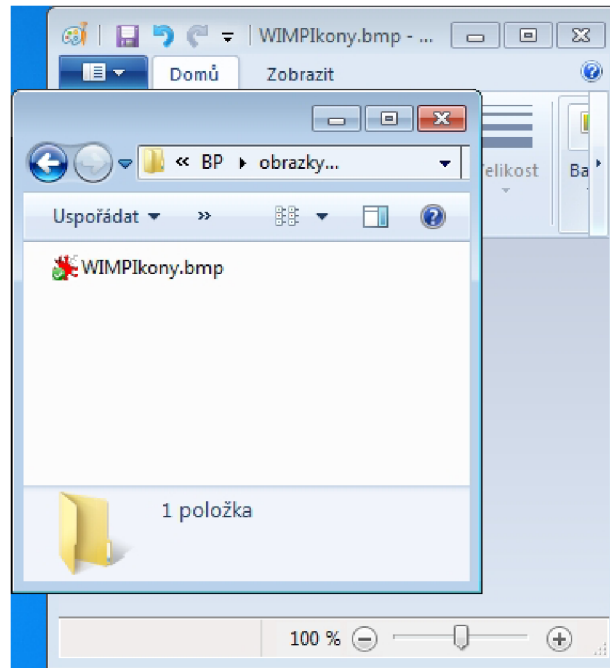
- Příkazový řádek: Rozhraní bez grafických prvků, které se ovládá pomocí textových příkazů. Výstup je taktéž textový. Je velmi efektivní tam, kde s počítačem pracuje odborník, který se naučí veškeré příkazy. Hardwarově nejméně náročné rozhraní. Historicky nejstarší uživatelské rozhraní.
- Grafické uživatelské rozhraní: Komunikace mezi člověkem a počítačem probíhá pomocí objektů na obrazovce. S tímto uživatelským rozhraním se nejlépe pracuje převážně většině uživatelů [5]. Náročnější na výpočetní výkon počítače. Nejvíce využívané v osobních počítačích. Nejmodernější uživatelské rozhraní.
- Textové uživatelské rozhraní: Jde o vývojový mezistupeň mezi příkazovým řádkem a grafickým uživatelským rozhraním. Na obrazovce se nachází několik znaků, které vytváří prvky uživatelského rozhraní. Mohou to být jak znaky textové, tak znaky reprezentující například roh okna. Nejvíce využívané bylo v době operačního systému DOS [4]. Dnes se již moc nevyskytuje.

2.3 Grafické uživatelské rozhraní

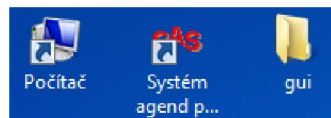
Grafické uživatelské rozhraní se skládá z objektů, se kterými uživatel manipuluje. Objekty jsou umístěny na obrazovce. Výsledek této akce pak počítač také zobrazí na této obrazovce [1]. Tento způsob ovládání se nazývá *přímá manipulace* [5]. Používají se zde techniky *Drag and drop* (uchop a přemísti), kdy uživatel vybraný objekt přesouvá stejně jako v reálném světě (přesouvání souborů do koše pro smazání nebo přesunutí souboru ze složky do složky). V poslední době se tato technika rozšířila na *Look and feel* (shlédni a porozuměj), kdy se kromě přetahování objektů používají i různá jiná gesta, která jsou podobná gestům v reálném světě. Uživateli tak připadá prostředí povědomé a lehce se v něm zorientuje [5].

V dnešní době je nejpoužívanějším konceptem grafického uživatelského rozhraní takzvaný WIMP [5]. Jedná se o zkratku slov windows (okna), icons (ikony), menu (nabídka) a pointer (polohovací zařízení). Pomocí oken (obrázek 2.2) je rozdělena plocha obrazovky mezi několik současně spuštěných aplikací. Ikony (obrázek 2.3) znázorňují programy, adresář, složky nebo zařízení jako například počítač. V menu (obrázek 2.4) nalezne uživatel dostupné akce, které slouží k ovládní aplikace. Pomocí polohovacího zařízení (nejčastěji se používá myš [5]) manipuluje s objekty, které vidí na obrazovce.

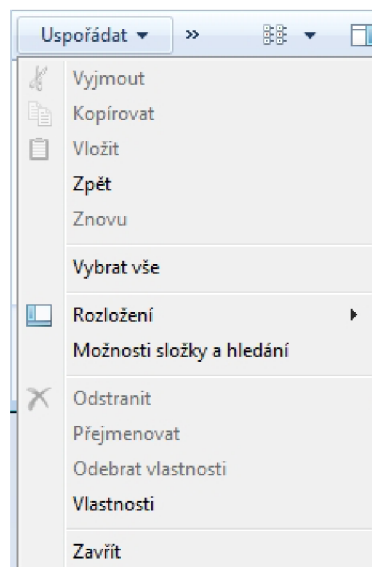
Při návrhu uživatelského rozhraní se využívají dva pohledy na problematiku. V počátcích vývoje výpočetní techniky se nejvíce využíval přístup *machine-centered*. Obsluhu počítačů tvořili technicky vzdělaní lidé a nebylo jich mnoho. Proto se uživatelské rozhraní přizpůsobovalo co nejvíce skutečné funkci stroje [4]. Ovládacími prvky se nastavovaly přímo vlastnosti zařízení. Uživatel proto musel vědět, jak počítač pracuje. Postupem času se výpočetní technika rozšiřovala mezi obyčejné lidi. Bylo tedy nutné uživatelská rozhraní při-



Obrázek 2.2: Okna v operačním systému Windows 7



Obrázek 2.3: Ikony v operačním systému Windows 7



Obrázek 2.4: Menu v operačním systému Windows 7

způsobit co nejvíce běžnému člověku. Tento přístup se nazývá *people-centered* [5]. Ovládací prvky se více zaměřovaly na způsob, jakým uživatel vykonává svou práci v reálném prostředí. Uživatel tedy pouze převádí úkony, které jsou mu známé, do prostředí počítače.

Studii, jak správně tyto prvky sestavit do výsledné aplikace tak, aby byla pro uživatele co nejpříjemnější a nejjednodušší, se zabývá obor nazvaný *Human-Computer Interaction* [2]. Tento obor se skládá z mnoha disciplín [1]. Návrhář (člověk, který uživatelské rozhraní navrhuje) musí znát základy psychologie, sociologie, fyzické možnosti, kulturní zvyklosti konkrétního člověka nebo skupiny lidí, pro které je výsledná aplikace určena [1]. Proto je způsob návrhu, kdy se do středu staví uživatel, náročnější, ale s výslednou aplikací se uživateli pracuje daleko lépe [5].

Existuje mnoho způsobů, jak vytvářet uživatelská rozhraní [2]. V této práci je dále uveden pouze přístup nazývaný *PACT*, protože na popis všech možných způsobů není dostatečný prostor. Zkratka *PACT* se skládá z těchto slov:

- People – lidé, kteří budou s aplikací pracovat (uživatelé).
- Activities – aktivity, které by měly být dostupné.
- Contexts – souvislosti, ve kterých budou aktivity vykonávány.
- Technologies – technologie, které použijeme.

Jedná se o 4 pohledy na uživatelské rozhraní, na které by se měl návrhář zaměřit. Všechny 4 body se navzájem ovlivňují.

2.4 Analýza uživatelského rozhraní

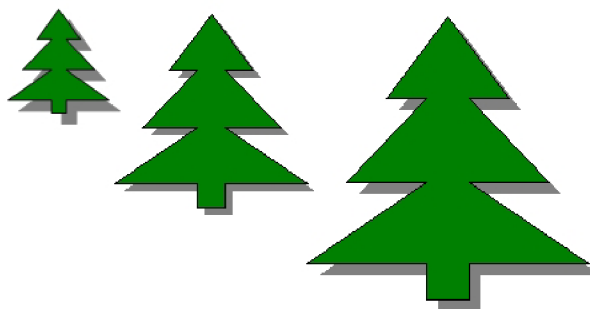
Při vytváření uživatelského rozhraní je možné využít přístup *PACT*. O takové analýze se hovoří jako o *PACT analýze* [5].

Lidé

Lidmi se myslí skupina lidí, kteří budou prostřednictvím navrženého uživatelského rozhraní komunikovat s aplikací (dále cíloví nebo koncoví uživatelé). Koncoví uživatelé jsou rozděleni do skupin (podle schopností, odbornosti, věku, pohlaví atd.). Každou skupinu zastupuje takzvaná *persona* (jedná se o fiktivní postavu, která zastupuje skupinu lidí) [1]. Je důležité si uvědomit, že i přes některé společné vlastnosti je každý člověk jedinečný. Návrhář by si měl uvědomovat právě tyto odlišnosti, aby mohl navrhnout přívětivé uživatelské rozhraní pro všechny uživatele [3].

Na první pohled jsou nejvíce viditelné různé fyzické vlastnosti člověka (velikost, váha atd.). Návrhář by se měl snažit vytvořit takové uživatelské rozhraní, aby s ním kromě průměrných lidí mohli pracovat i ti nějakým způsobem vyčnívající (nejmenší, nejvyšší, pravoruký, levoruký atd.) [1].

Kromě fyzických vlastností se lidé různí i ve vnímání světa pomocí smyslových orgánů (zrak, sluch, hmat, čich a chuť) [5]. Tyto vlastnosti mají největší vliv na to, jak zábavné, přístupné a použitelné bude uživatelské rozhraní pro konkrétního uživatele [5]. Každý člověk vnímá okolní svět trochu jinak než ostatní, ale jsou i určitá pravidla, která platí pro všechny. Například objekty, které jsou na obrazovce menší, jsou vnímány jako vzdálenější (obrázek 2.5). Do stejné skupiny patří i vnímání barev (např. červená = nebezpečí).



Obrázek 2.5: Nejmenší objekt se zdá být umístěn nejdále.

Vnímání okolního světa mohou ovlivnit i určitá onemocnění a postižení člověka. Mezi taková postižení patří barvoslepost, kterou trpí asi 8% západoevropské populace [5]. Většinou se však jedná o částečnou poruchu, kdy člověku dělá problém rozeznat jen některé barvy (nejčastěji červenou se zelenou) [1]. Z tohoto důvodu je dobré rozlišovat objekty kromě barvy i jiným poznávacím znamením (např. tvarem nebo textem). Existují i další znevýhodnění, která je však potřeba vzít v potaz podle specifických vlastností konkrétního systému (například vada řeči při hlasovém ovládnání aplikace).

To, co člověk uvidí nebo uslyší, si dále interpretuje na základě kulturního prostředí, ze kterého pochází. V uživatelském rozhraní se jedná hlavně o chápání různých symbolů [5]. Například Američané vnímají křížek jako zápor nebo odmítnutí, ale v našich podmínkách může znamenat i potvrzení správnosti odpovědi (formuláře na úřadech). Kultura a náboženství ovlivňují i celkové uvažování člověka a interpretaci informací [5].

Je také rozdíl, zda navrhované rozhraní bude sloužit pro zkušeného nebo pro začínajícího uživatele [5]. Tyto dvě skupiny se liší zejména nároky a očekáváním vzhledem ke svým dosavadním zkušenostem. Zejména u zkušeného uživatele můžeme předpokládat znalost různých symbolů, které se využívají napříč všemi aplikacemi v systému. Špatné použití takového symbolu více znepříjemní práci uživateli, který zná správný význam tohoto symbolu, než uživateli, který tento symbol vidí poprvé.

Uživatel si při interakci s nějakou věcí vytváří takzvaný *mental model* (jedná se o představu, jak daná věc uvnitř funguje) [5]. Pokud chce uživatel dosáhnout nějakého cíle, nejprve si představí provedení této akce na svém *mental modelu* a poté stejný postup aplikuje na model reálný. Proto musí návrhář svůj návrh co nejvíce přizpůsobit *mentálnímu modelu*, který si uživatel vytvoří. Návrhář tedy musí předvídat to, jak uživatel bude vnímat uživatelské rozhraní aplikace.

Informace o uživateli lze získat několika způsoby. Jedním z nich jsou dotazníky, které uživatelé vyplní buď sami nebo za asistence tazatele. Pomocí dotazníku je možné oslovit velké množství lidí, ale tato data mohou být ovlivněny mnoha vedlejšími aspekty [6]. Aby byl tazatel co nejúspěšnější, musí vhodně vybrat otázky. Těch by nemělo být mnoho a musí být jednoduše pochopitelné. Pokud uživatel volí z množiny odpovědí, musí tato množina být úplná (případně možnost doplnit dotazovaným vlastní odpověď) [6]. Nejlepších výsledků lze dosáhnout, pokud budou na sebe otázky navazovat a budou seřazeny od nejtěžší po nejtěžší [6].

Aktivity

Aktivity popisují, jak bude s uživatelským rozhraním nakládáno. Kromě způsobu interakce určují i jak často bude daný úkon prováděn, kdo bude úkon provádět, jak rychlá by měla být reakce systému a vlivy okolního prostředí. Často prováděné aktivity by měly být uživateli nejdostupnější. Naopak úkony prováděné méně často by uživateli neměly překážet při každodenní činnosti [7]. Mohou být zapsány formálně (případy užití) nebo neformálně jako scénář. Scénářem se rozumí stručný popis uživatele, prostředí, aktivity, kterou bude provádět, a důvod, proč ji chce provádět. Tyto scénáře často formulují i uživatelé jako svoje přání, jak by chtěli se systémem pracovat [5].

Souvislosti

Jedná se o souvislosti (kontext), během kterých probíhají aktivity. Kromě prostředí je analyzován i sociální a organizační kontext [5]. Prostor, ve kterém probíhá aktivita, může být různorodé. Návrhář musí zohlednit hlučnost, prašnost, světlo a jiné podmínky podle toho, kde se aktivita bude odehrávat, případně podle toho kde se bude dané zařízení nacházet. Z hlediska sociálního je sledováno, zda bude uživatel se systémem komunikovat sám nebo s více lidmi. Jednak je rozdílné chování lidí ve skupinách, ale také některá vstupní a výstupní zařízení nejsou vhodná do prostředí, kde se nachází větší počet lidí. Dále je také důležitá informace, zda budou v okolí jiní lidé, kteří mohou uživateli s ovládním pomoci, pokud si nebude vědět rady (například rozdíl aplikace v bance, kde může úředník pomoci, namísto mobilní aplikace). Organizační kontext je zejména otázkou oprávnění a bezpečnosti. Tedy kdo bude mít k aplikaci přístup a v jakém rozsahu.

Technologie

Do této části patří hardwarové a softwarové nástroje využití při komunikaci s aplikací. Analyzují se především vstupní (klávesnice, joystick atd.) a výstupní zařízení (obrazovka, tiskárna atd.), ale i způsob komunikace s okolním světem (síťové připojení) a uchování dat aplikace (zabezpečení, lokace atd.). Rozhodnutí se provádí na základě aktivity a kontextu používání aplikace. Špatně zvolené technologie mohou významně ovlivnit spokojenost uživatelů s výslednou aplikací (nejčastěji různé poruchy nebo pomalé reakce na uživatelský vstup) [7].

2.5 Návrh uživatelského rozhraní

Samotný návrh uživatelského rozhraní se liší podle toho, jaký účel bude aplikace plnit. První skupinu tvoří aplikace, které mají pobavit uživatele. Takových aplikací se na trhu nachází velké množství a často se liší pouze uživatelským rozhraním. Uživatelské rozhraní by mělo být tvořené co nejvíce originálně a jeho úkolem je zaujmout uživatele. Musí uživatele motivovat, aby aplikaci často využíval, aby se k ní vracel a aby si ji oblíbil [7]. Pokud uživatel nebude spokojen, poměrně lehce přejde k jiné aplikaci [3].

Druhou skupinu tvoří aplikace, které slouží jako pracovní nástroj uživatele. Zde je možné předpokládat, že uživatel bude tyto aplikace využívat každý den. Většinou není možné zvolit alternativní program. Proto musí být uživatelské rozhraní co nejvíce formální, aby s nimi bylo příjemné pracovat za různých okolností [5]. Při návrhu uživatelského rozhraní pro tento druh aplikací se zpravidla využívají standardní prvky operačního systému. Případně je možné vytvořit vlastní ovládací prvky, které nebudou vzhledově vyčnívat. Pokud to

návrhář dodrží, uživatel se mnohem snáze zorientuje a jednodušeji porozumí celé aplikaci a jejímu ovládání. Opačná situace může vést až k frustraci z používání aplikace [7].

Při výběru vhodného prvku a jeho použití se návrhář rozhoduje na základě poznatků z mnoha oborů (psychologie, sociologie atd.). Tyto poznatky musí vhodně aplikovat na konkrétní problém. Z tohoto důvodu je každé uživatelské rozhraní jedinečné a jeho návrh je náročný [2].

Malým pomocníkem mohou být příručky, které dodávají výrobci operačních systémů. Jsou v nich uvedena pravidla, jak prvky správně používat a jak tyto prvky vzájemně kombinovat. Jedná se však pouze o základní a obecné rady. Některé společnosti tato pravidla vyžadují a jejich dodržování kontrolují. Všechny aplikace tak musí projít schalovacím procesem před prodejem zákazníkovi. Tento postup platí například pro firmu Apple a operační systém iOS [8]. Firma Microsoft tato pravidla pouze doporučuje dodržovat.

2.6 Testování uživatelského rozhraní

Testování uživatelského rozhraní se například provádí pomocí testů použitelnosti [5]. Uživatelské rozhraní je použitelné, pokud je práce s ním jednoduchá, bezchybná, rychlá a lehce naučitelná [3].

K těmto testům jsou potřeba uživatelé, kteří systém testují používáním a plněním běžných úkolů pro daný systém. Uživatelské rozhraní by se mělo testovat ihned od začátku vývoje [2], respektive ihned po dokončení prvních návrhů.

Z počátku vývoje se často používají takzvané *lo-fi* (*low fidelity*) prototypy [5]. Zpravidla jde o návrh uživatelského rozhraní načrtnutý na několika papírech, kdy návrhář manipuluje s jednotlivými papíry, jako by se jednalo o reakci počítače na chování uživatele. *Lo-fi* testy mají za úkol odhalit chyby v požadavcích na uživatelské rozhraní a vyjasnit představy cílových uživatelů [3]. Oprava těchto chyb by byla v pozdějších fázích vývoje časově i finančně náročnější než na počátku [3].

V pozdějších fázích testy probíhají na takzvaných *hi-fi* (*high fidelity*) prototypoch. Tyto prototypy vzhledově odpovídají hotovému uživatelskému rozhraní. Na jejich sestavení se často používají prototypovací jazyky. Návrhář musí počítat s tím, že *hi-fi* prototypy se zpravidla po dokončení testování zahazují [5]. *Hi-fi* prototypy by měly obsahovat reálná data, aby co nejméně mátlá uživatele a testované rozhraní působilo co nejvíce věrohodně. Při testování sedí uživatel ve specializované místnosti a návrhář je sleduje z vedlejší místnosti nebo sedí uživatel společně s návrhářem před počítačem [5]. Ve specializované místnosti se nachází zařízení, které zaznamenává uživatelské reakce. Kromě kliků a vyplňovaných dat se často sleduje i pohyb očí uživatele. Díky tomu může návrhář zjistit, na jaké objekty a v jakém pořadí se uživatel díval. Návrhář by také v průběhu testování měl zjišťovat, jak uživatel vnímá rozhraní pocitově. Tyto informace pomáhají při optimalizaci uživatelského rozhraní [5].

V obou případech uživatel postupuje podle předem nachystaného scénáře a plní úkoly [5]. Návrhář sleduje uživatelské reakce, ale nesmí ovlivňovat uživatele svým pohledem na problematiku. Nejlepších výsledků lze dosáhnout, pokud se testování provádí na cílových uživateli [5]. Po dokončení testů se ověřuje, zda jsou již dostatečně splněny požadavky. Pokud ne, poznatky z testování promítneme do návrhu a celý cyklus se opakuje. V opačném případě se návrh uživatelského rozhraní může považovat za dokončené.

2.7 Zásady tvorby uživatelského rozhraní Windows 8

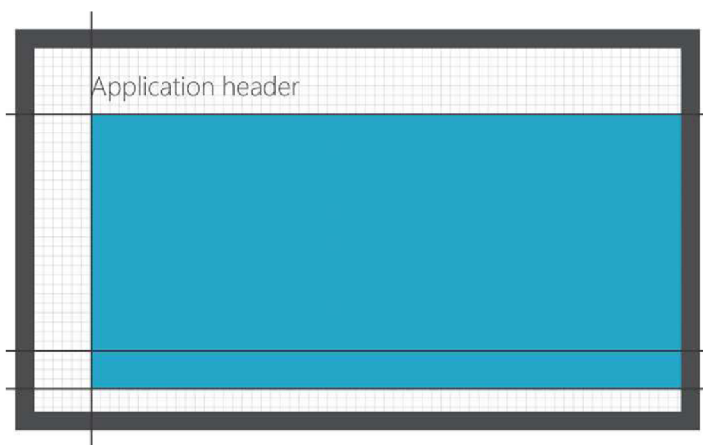
Každý operační systém obsahuje vizuální objekty, které mají typický vzhled. Tyto prvky musí návrhář kombinovat tak, aby odpovídaly zásadám operačního systému. Dodržení takových zásad a doporučení usnadní uživateli přechod mezi aplikacemi v rámci operačního systému.

Ve Windows 8 se poprvé objevilo nové uživatelské rozhraní z počátku označované jako *Metro*. V oficiálních zdrojích se však o tomto rozhraní mluví jako o *uživatelském rozhraní Windows 8*, protože *metro AG* je ochranná známka německé firmy [9].

Informace uvedené níže jsem čerpal z příručky [10] dodávané firmou Microsoft. Příručka obsahuje informace o rozložení prvků v aplikaci, rozměry jednotlivých prvků a doporučení, v jakých situacích a který prvek použít. Dodržení zásad a doporučení je nutnou podmínkou pro vystavení aplikace ve *Windows Store*. Jedná se o obchod, ve kterém mohou uživatelé připojení k internetu nakupovat aplikace přímo do svého systému.

Windows 8 je optimalizován pro dotykové obrazovky a úsporu energie [10]. Což z něj dělá ideální operační systém pro přenosná zařízení s dotykovou obrazovkou (v dnešní době převážně tablety, ale začínají se objevovat i notebooky s dotykovou obrazovkou). Vzhledem k držení tabletu doporučuje Microsoft umisťovat ovládací prvky do pravého a levého dolního rohu obrazovky. Při držení tabletu to má ruka uživatele nejbližší právě do těchto míst. Naopak informativní prvky je doporučeno umisťovat do středu horní části obrazovky tak, aby ruce uživatele tyto prvky nezakrývaly.

Minimální rozlišení pro Windows 8 je 1024 x 768 pixelů, kde ale nebudou fungovat některé pokročilé funkce systému. Pro plnohodnotný běh operačního systému je doporučeno rozlišení 1366 x 768 pixelů. Pro toto rozlišení Microsoft udává doporučené rozložení obrazovky aplikace (obrázek 2.6). Horní okraj je široký 140 pixelů, levý okraj 120 pixelů, dolní okraj není povinný. Pravý a spodní okraj není třeba specifikovat, protože obsah aplikace může být schován za okrajem obrazovky. Nadpis aplikace se nachází na lince, která je vzdálená 100 pixelů od horního okraje obrazovky.



Obrázek 2.6: Rozložení obrazovky aplikace ve Windows 8

Pro určení rozestupů mezi objekty na obrazovce jsou objekty rozděleny do dvou skupin. Jednu skupinu tvoří objekty s výrazným ohraničením jako například obrázky nebo dlaždice. Druhou skupinu pak tvoří objekty, u kterých je obtížné přesně určit, kde končí a kde začínají. Většinou jde o různé textové elementy. Mezi dvěma objekty s výrazným ohraničením

musí být mezera minimálně 10 pixelů, stejně jako mezi objektem s výrazným ohraničením a textovými objekty. Mezi dvěma textovými objekty musí být mezera minimálně 40 pixelů. Mezi většími organizačními celky by měl být rozestup alespoň 40 pixelů.

Další doporučení se pak převážně týká konkrétních prvků. Z velkého množství jsem vybral několik prvků, které využiji v navrhovaném uživatelském rozhraní:

- *Charm* – menu které se vysouvá z pravého okraje obrazovky. Obsahuje položky společné pro všechny aplikace systému Windows 8. Nalézají se v něm nástroje pro vyhledávání v ostatních aplikacích, sdílení obsahu s jinými aplikacem, nastavení aplikace a tlačítko pro odhlášení.
- *App bar* – panel vysunutý z dolního okraje obrazovky, měly by na něm být umístěny vedlejší funkce pro ovládání aktuální obrazovky. Tlačítka na něm umístěná by měla být shlukována do skupin a nejprve zarovnána vpravo, další skupina pak zarovnána doleva.
- *Navigation bar* – platí podobné zásady jako pro *App bar*, ale vysunuje se z horního okraje a slouží především pro navigaci aplikace.
- *Progress ring* – kolečko, které indikuje operace na pozadí aplikace. Na rozdíl od *progress baru* se využívá tam, kde není známá doba trvání výpočtu a uživatel nemůže během výpočtu s aplikací pracovat (typicky přihlašování na vzdálený server).
- *Combobox* – prvek určený pro zvolení jedné položky z nabídky. Doporučeno využívat, pokud v množině na výběr nebude více jak 10 prvků.
- *Radio buttons* – skupina prvků, z nichž může být vybrán pouze jeden. Doporučeno používat, pokud je na výběr ze dvou až sedmi prvků.
- *Edit* – prvek určený pro textový vstup. Používá se pro jednořádkové texty.
- *Memo* – obdobně jako *edit*, ale používá se pro víceřádkové vstupy.

V uživatelském rozhraní Windows 8 se hojně využívá prvek nazývaný *dlaždice*. Jedná se o univerzální grafický objekt obdélníkového nebo čtvercového tvaru. Dlaždice zastupuje nějaký objekt aplikace nebo reálného světa. Může například zastupovat aplikaci (pak se v ní zobrazí část obsahu aplikace nebo ikona aplikace) nebo žáka školy (zobrazí se jméno žáka a třída). Dlaždice je základní stavební prvek systému Windows 8.

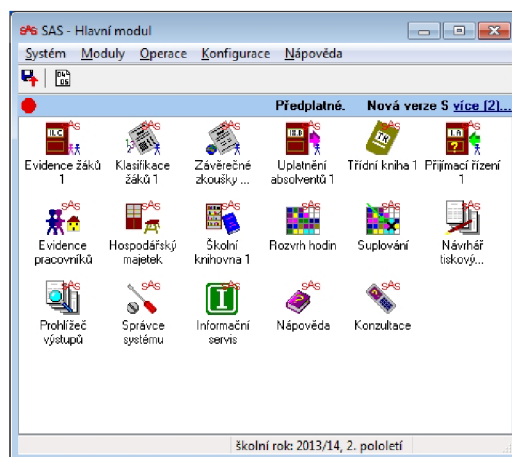
Kapitola 3

Současné systémy pro správu školní matriky

V současnosti existuje několik aplikací pro správu školní agendy. Převážně jde o zdarma dostupné internetové aplikace, které spíše slouží k informování rodičů o průběhu studia jejich dětí. Pro přehled jsem vybral tři placené systémy, které pokrývají alespoň základní potřeby učitelů. Jedná se o systémy SAS, Bakaláři a internetový projekt iŠkola. Další často používaný systém se nazývá Škola online¹. Bohužel se mi nepodařilo získat přístup ke spustitelné verzi tohoto systému, a proto není v přehledu zahrnut.

3.1 SAS – Systém agend pro školy

Tuto aplikaci vyvíjí firma MP-Soft, a.s.² Je to rozsáhlý systém, který svou funkčností pokrývá většinu potřeb základní a střední školy. Na obrázku 3.1 se nachází hlavní nabídka aplikace, ze které uživatel pokračuje do jednotlivých modulů. Nabídka působí docela příjemně. Vadou na kráse by mohlo být celkem velké množství modulů, ale tento počet lze ovlivnit nastavením práv uživatele.



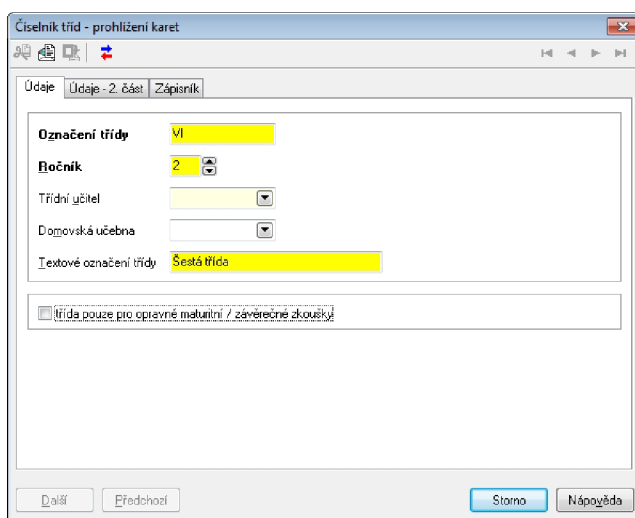
Obrázek 3.1: Hlavní nabídka SAS

¹<http://www.skolaonline.cz/>

²<http://www.mp-soft.cz/sas/>

Po zvolení modulu se zobrazí poměrně atypické okno, které tvoří jen menu, nástrojová lišta a stavový řádek. Jednotlivé moduly jsou vzhledově hodně podobné, což určitě usnadní orientaci uživatele. Tím však trpí rozložení menu modulu. V jedné z nabídek je 11 položek a 5 podnabídek. Naopak v druhé nabídce jsou pouze 3 položky a 1 podnabídka. Doporučené minimum jsou 4 položky, jinak je doporučeno využít jiný prvek [11]. Vhodnější rozložení by mohlo vylepšit orientaci uživatelů. Obecné názvy si také nemusí uživatel spojit s požadovanou funkcí.

Ostatní formuláře působí dobrým dojmem (obrázek 3.2). Jsou dobře zarovnané a dodržují doporučený vzhled formulářů ve Windows. Barevně zvýrazněny jsou položky, které jsou důležité pro sběr údajů ze školní matriky.



Obrázek 3.2: Formulář SAS

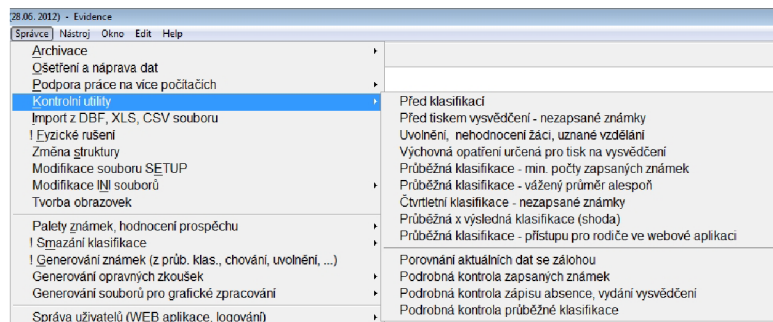
Z hlediska uživatelského rozhraní je velkou nevýhodou systému množství funkcí, které dramaticky znesnadňují orientaci uživatele. Až na tyto nedostatky působí celé uživatelské rozhraní aplikace přívětivě a odpovídá doporučeným zásadám. Uživatel by tedy práci s tímto systémem měl zvládnout bez větších obtíží.

3.2 Bakaláři

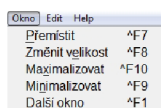
Stejně jako předchozí software jsou Bakaláři³ velice rozsáhlý systém s mnoha funkcemi. Hlavní nabídka zde chybí a je nahrazena zástupci, pomocí kterých se spouští jednotlivé moduly. Těchto modulů je 22, což je opravdu hodně. Menu (obrázek 3.3) obsahuje mnoho položek a názvy jednotlivých položek přesahují doporučenou délku (maximálně trojslovné názvy [11]). Přístupové klávesy by neměla tvořit úzká písmena, protože podtržení není dostatečně viditelné. Podtržení těchto písmen by nemělo být zobrazováno ihned, ale až po stlačení klávesy Alt. Klávesové zkratky jsou zobrazovány nestandardně (viz obrázek 3.4). Jako neodpovídající působí i nadměrně velký text použitý v názvech položek nabídky.

Jednotlivé formuláře jsou neuspořádané a vzhledově hodně nepříjemné (obrázek 3.5). Zejména velké množství barev a různá velikost objektů může pro uživatele působit odpudivě a nebude jednoduché se naučit s touto aplikací pracovat.

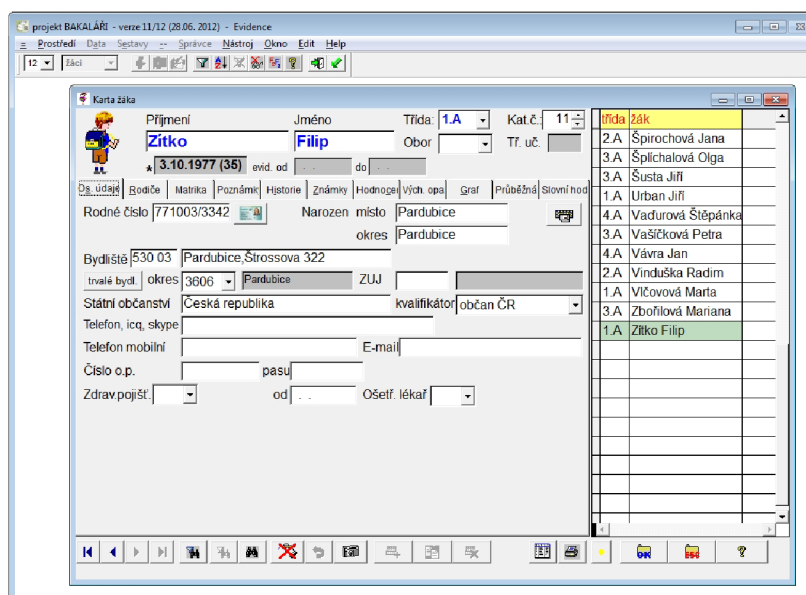
³<http://www.bakalari.cz>



Obrázek 3.3: Dlouhé položky a přístupové klávesy v menu Bakalářů



Obrázek 3.4: Klávesové zkratky v menu Bakalářů



Obrázek 3.5: Bakaláři formulář

3.3 iškola

iškola je placený systém provozovaný pouze na internetových stránkách⁴. Systém nenabízí tolik funkcí jako předchozí dva systémy. Nachází se zde funkce potřebné pouze pro práci učitelů (zadávání klasifikace, třídní kniha, maturita, ...). Na obrázku 3.6 je zobrazena stránka určená pro zápis do třídní knihy. V horní části se nachází hlavní menu aplikace a v pravém sloupci jsou volitelné nástroje, které mohou pomoci při práci s aplikací. Pod hlavní

⁴<http://www.iskola.cz/>

nabídkou jsou umístěny záložky pro práci s třídní knihou. Velké množství těchto záložek zhoršuje orientaci uživatele. Problémové může být i použití velmi malé velikosti písma na záložkách. Na této obrazovce je také vidět velká rozmanitost druhů, barev a velikostí písma. Ostatní stránky této aplikace jsou hodně podobné stránce pro editaci třídní knihy. Díky této vlastnosti se uživatel snadněji naučí s celý systémem pracovat.

The screenshot displays the 'iškola.cz' web application interface for editing a classbook. The main content area is titled 'Editace třídní knihy' and shows the class '1.A'. The date is 'Neděle, 07.04.2013', and it indicates that the classbook is currently empty for that day. A table is provided for recording absences, with columns for 'Absence žáka', 'Hodina' (1-11), 'Důvod absence', and 'Poznámka'. The table lists several students: Martina Bělochová, Aleš Brtník, Michal Kucharčík, Adam Nejděk, Marie Nováková, and Lucie Vondráčková. The interface also includes a sidebar with user information (Přihlášen: Rostislav Brandejs), a search function, and various settings. The footer contains the copyright information: '(c)2005-2013 Computer Media s.r.o. Všechna práva vyhrazena.'

Obrázek 3.6: Třídní kniha iškola

Kapitola 4

Specifikace aplikace

V této kapitole jsou upřesněny požadavky na výslednou aplikaci. První část specifikace byla známá hned na počátku práce. Druhou část tvoří požadavky, které vznikly až na základě porovnání konkurence.

4.1 Počáteční požadavky na aplikaci

V této sekci jsou požadavky na aplikaci, které měla přímo firma MP-soft, a.s. Cílem celé práce je vytvořit samostatnou aplikaci pro potřeby učitelů na základních a středních školách. Pro uživatele by měla tato aplikace sloužit jako jednoduchý nástroj pro běžnou práci ve třídě. Firma dlouhodobě vyvíjí aplikaci SAS, která pokrývá kompletní potřebu středních, základních i mateřských škol. Mnou vytvořené uživatelské rozhraní aplikace tedy nemá nabízet všechny funkce, ale musí pokrýt běžné potřeby učitelů. Dle zadání mezi základní funkce aplikace patří zápis klasifikace, absence, a popřípadě oprava zápisu hodiny v třídní knize. Tato aplikace bude fungovat na již vytvořené databázi systému SAS od firmy MP-Soft, a.s.

- Údaje v třídní knize budou obvykle předvyplněny z původní aplikace, ale téma hodiny bude pravděpodobně vyplňováno až na začátku hodiny. Aplikace musí umožnit i nový zápis do třídní knihy v případě neočekávaného suplování.
- Zadávání známek by mělo být přizpůsobeno pro zapsání jednotlivé známky v hodině (typicky po ústním zkoušení), tak i pro zadání známek všem žákům najednou (typicky pro písemné zkoušení). Kromě samotné hodnoty známky by měl systém evidovat i jiné údaje jako jsou typ zkoušení, váha známky, datum, téma.
- Absence se budou zadávat pouze jako přítomen a nepřítomen. Druh absence a omlouvání absence bude doplňováno třídním učitelem z rozsáhlé aplikace. Není tedy potřeba tuto funkci zahrnout v mé aplikaci.
- Kromě používání aplikace přímo v hodině nebo ve škole by ojedinele mohla být využita z domu nebo na cestě do/ze školy. Požadavkem tedy bylo, aby aplikace fungovala i na přenosném zařízení. Jelikož firma vyvíjí aplikaci pro platformu android a mobilní telefony, bylo rozhodnuto, že mnou vyvíjené uživatelské rozhraní má být vytvořeno pro tablety, osobní počítače a notebooky s operačním systémem Windows.

4.2 Požadavky vycházející z porovnání konkurence

Pro lepší přehled jsem všechny vyzkoušené systémy z kapitoly 3 zhodnotil v tabulce 4.1. Rozsah hodnotící škály je od 1 (nejlepší) až po 5 (nejhorší).

Název	Jednotnost	Dodržení standardů	Rušivé prvky	Seriózní vzhled	Funkčnost
SAS	1	2	2	2	1
Bakaláři	4	4	5	5	1
iŠkola	1	3	3	3	3

Tabulka 4.1: Porovnání dostupných systémů

Z tabulky je vidět, že z hlediska uživatelského rozhraní nejlépe skončil systém SAS. Nejhuře pak systém Bakaláři. Všechny aplikace obsahovaly menší či větší prohřešky proti zásadám návrhu přívětivého uživatelského rozhraní. Všechny systémy jsou značně rozsáhlé a proto bylo nutné se k požadovanému cíli proklikat přes několik nabídek. Na uživatelském rozhraní aplikací je také vidět, že se až na výjimky nemodernizuje.

Při porovnání existujících aplikací jsem zjistil, že mnoho firem si neuvědomuje důležitost uživatelského rozhraní a nebo mu nevěnují pozornost z jiných důvodů. Například proto, že většina zákazníků při nákupu hodnotí v první řadě funkčnost systému a až poté uživatelské rozhraní. Přitom špatně navržené uživatelské rozhraní může znehodnotit celou aplikaci.

Mnou vytvořené uživatelské rozhraní by se tedy mělo zaměřit především na zmíněné nedostatky hotových aplikací. Hlavní předností bude jednoduchost aplikace. Složitější a méně často využívané funkce budou dostupné pouze z rozsáhlého systému a mnou navržená aplikace bude sloužit jen jako doplněk pro pohodlnější práci učitelů. Další konkurenční výhodou bude moderní uživatelské rozhraní, které bude odpovídat zásadám tvorby přívětivého uživatelského rozhraní.

Kapitola 5

Návrh uživatelského rozhraní školního informačního systému

V této kapitole jsou popsány počítačící požadavky na celou aplikaci. Dále analýza školního prostředí a cílových uživatelů. V dalších částech kapitoly je popsán proces návrhu uživatelského rozhraní školního informačního systému. Popis testování navrženého rozhraní se nachází na konci kapitoly.

5.1 Analýza školního prostředí

Základní informace o uživateli a školním prostředí jsem se rozhodl čerpat od zaměstnanců firmy MP-soft, a.s. Další potřebné informace jsem zjišťoval pomocí online dotazníku [12] pro současné uživatele školního systému SAS. Jedná se tedy o cílovou skupinu uživatelů. Tento způsob se mi jevil jako časově i finančně nejméně náročný. Bohužel online dotazník mohli vyplnit pouze uživatelé, kteří používají email a internet, ostatní uživatelé nemohli dotazník nevyplnit. Dotazník byl využit i pro doplnění některých požadavků na výslednou aplikaci podle přání uživatelů. Celkem dotazník vyplnilo 178 respondentů, z toho 168 jich využívá původní aplikaci SAS. Při analyzování jsem postupoval podle principu PACT:

- **Lidé** – Typický uživatel nové aplikace bude učitel, který využívá osobní počítač nebo notebook. Při ovládání počítačů preferují myš. Uživatel umí pracovat se systémem SAS. Předpokládám tedy, že se jedná o mírně pokročilé uživatele. Dle statistiky Ministerstva školství, mládeže a tělovýchovy na základní a střední škole převážně vyučuje žena starší 46 let [13]. Většina učitelů je české národnosti. Předpokládám tedy, že pro tuto skupinu uživatelů platí stejné vlastnosti jako pro jiné lidi české národnosti.
- **Aktivita** – Nejčastěji využijí učitelé aplikaci k zadávání známek všem žákům současně. Učitel zvolí typ, téma, váhu a datum zkoušení společně pro všechny žáky, poté každému žákovi přiřadí známku. Způsob využívaný při písemném zkoušení, kdy jsou zkoušeni všichni žáci. Zadávání známek jednotlivým žákům bude druhou nejčastější učitelovou činností. V tomto případě učitel zvolí pouze váhu a typ zkoušení pro všechny žáky společně a při vkládání známky doplní téma a datum zkoušení. Způsob je určen především pro ústní zkoušení. Obě funkce mohou být využity i několikrát během vyučovací hodiny. Další funkce jako zadávání absence by měla být použita většinou jednou na začátku hodiny. Častější zadávání absence předpokládám pouze v případě příchodu nebo odchodu žáka v průběhu hodiny. Do třídní knihy bude učitel

zapisovat zpravidla jednou za hodinu. V případě suplování bude učiteli umožněno vytvořit zápis hodiny v třídní knize, ale nebude moct zapisovat známky.

- **Souvislosti** – Dostupnost původní aplikace ve všech učebnách školy je velmi ojedinelá. Avšak většina škol umožňuje přístup k původní aplikaci alespoň v několika učebnách. To je dáno především nedostupným počítačovým vybavením v učebnách. Předpokládám tedy, že minimálně z počátku bude nová aplikace využívána ve stejných podmínkách jako aplikace původní. Učitel tedy bude s aplikací pracovat jak ve sborovně tak i přímo ve třídě při výuce. Méně časté využití očekávám v přepravním prostředku nebo ve venkovním prostředí. S aplikací bude pracovat každý učitel samostatně a převážně s rozvrhem vytvořeným v původní aplikaci.
- **Technologie** – Rozhodl jsem se navrhnout uživatelské rozhraní v moderním stylu Windows 8, který je určen pro přenosná zařízení. Jelikož v současnosti jsou učitelé zvyklí na operační systém od společnosti Microsoft, mělo by pro ně být jednoduší naučit se ovládat aplikaci, která pracuje pod systémem právě od této firmy. Výhodou bude i možné rozšíření aplikace na tablety. Navržené rozhraní pro Windows 8 je možné implementovat v mnoha různých jazycích.

Navržené rozhraní optimalizuji pro obrazovky s rozlišením 1366 x 768 pixelů a více, protože se jedná o nejrozšířenější rozlišení [10]. Uživatel bude aplikaci ovládat zejména pomocí myši. Méně často pak využije klávesnici. Počítám i s ovládním pomocí dotykové obrazovky, pro kterou je celý systém Windows 8 přizpůsoben.

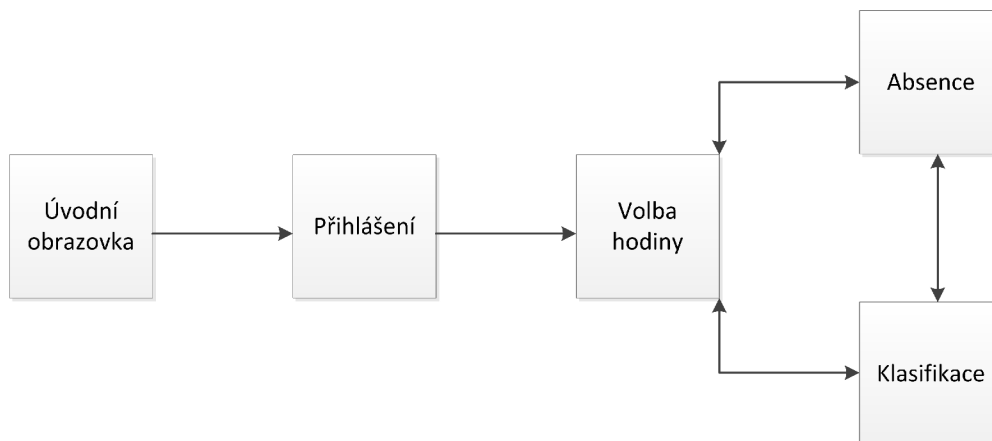
Jako datová základna poslouží existující databáze Firebird, kterou využívá již hotová aplikace SAS. Připojení na server s databází bude dostupné prostřednictvím sítě internet nebo vnitřní sítě školy.

5.2 Návrh uživatelského rozhraní školního systému

Při návrhu jsem vycházel z analýzy školního prostředí a ze zásad pro uživatelské rozhraní Windows 8. Na základě požadavků a funkcí, které měly být dostupné, jsem navrhl hierarchii celé aplikace (znázorněnou na obrázku 5.1). Z počátku jsem zvažoval i samostatný formulář (v prostředí Windows 8 se používá spíše pojem *obrazovka*) pro zápis do třídní knihy, který jsem ale vyřešil dialogem na obrazovce pro volbu hodiny. Pro jednoduchou aplikaci tedy postačí 5 obrazovek.

Okamžitě po spuštění aplikace se uživateli zobrazí takzvaný *splash screen*. Jedná se o úvodní obrazovku, na které se zobrazuje pouze logo aplikace a slouží pro ujištění uživatele, že spustil správnou aplikaci. Na pozadí *splash screenu* se alokují potřebné zdroje pro běh aplikace. Po načtení potřebných dat se může uživatel přihlásit ke svému účtu. To provede pomocí přihlašovacího jména a hesla (stejně přihlašovací údaje jako do původní aplikace). Po přihlášení učitel zvolí hodinu, ke které se budou vztahovat zadávané údaje. Poté se pohybuje mezi obrazovkami klasifikace a absence, až do doby, než se rozhodne změnit hodinu, odhlásit se nebo ukončit aplikaci.

U prvních dvou obrazovek nebyl prostor pro vlastní invenci, protože jsou zcela podle zásad doporučených Microsoftem. Vzhled dalších třech obrazovek tvoří společný základ dle zásad rozvržení aplikace pro Windows 8. Podrobnosti jsou uvedeny v následujících odstavcích.



Obrázek 5.1: Hierarchie aplikace

Společný základ

Pro všechny formuláře je společné rozložení základních prvků, zobrazování *app baru* a *navigation bar*. Rozložení, tedy nadpis, okraje a umístění hlavního obsahu obrazovky, se řídí předepsanými pravidly pro Windows 8. Pro všechny obrazovky je taky společná podkladová barva. Zvolil jsem tmavě modrou, která uklidňuje a zlepšuje schopnost se soustředit [5]. *Navigation bar* (zobrazený na obrázku 5.2) slouží pro navigaci v aplikaci. První ikona znázorňuje volbu hodiny. Mezera mezi první ikonou a dalšími dvěma symbolizuje nadřazenost volby hodiny oproti absenci a klasifikaci. Aktuální obrazovka je na *navigation baru* znázorněna oranžovým podsvícením. Uživatel k navigaci přistupuje přes horní okraj obrazovky. Naopak ze spodního okraje se vysouvá *app bar* (nástrojová lišta). Z lišty jsou dostupné méně často využívané funkce, které přímo neovlivňují průchod aplikací.



Obrázek 5.2: Navigation bar aplikace

Obrazovka volba hodiny

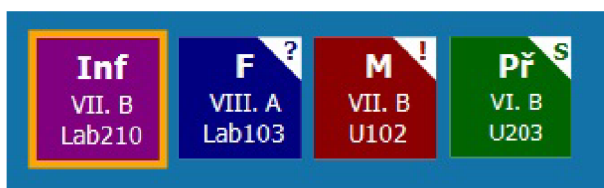
Hlavním úkolem této obrazovky je volba třídy, předmětu, data a vyučovací hodiny, ve které učitel právě vyučuje. V první verzi návrhu jsem použil speciální komponentu pro volbu data a 3 *combo boxy* pro zvolení třídy, předmětu a vyučovací hodiny. V tomto případě byla častá volba hodiny příliš zdlouhavá, což se projevilo při testech. Druhou variantou byla kombinace třídy a předmětu s následnou volbou data z kalendáře a vyučovací hodiny z *combo boxu* pro zadání absence. Toto řešení také nebylo ideální, protože učitelé by museli přemýšlet nad volbou vyučovací hodiny. Jako ideální se ukázala komponenta, která reprezentuje rozvrh učitele. Jedním klikem může učitel vybrat jak vyučovací hodinu, tak i třídu a předmět. Hodiny reprezentují *dlaždice*. Ovšem pro toto řešení je nutné nejprve v původní aplikaci SAS vyplnit rozvrh a dle rozvrhu předem vyplnit třídní knihu. Rozvrh většina škol vyplňuje

a předvyplnění třídní knihy je otázka pár kliků navíc jednou za týden nebo měsíc. Je to tedy malá daň za výrazné zjednodušení každodenní práce učitele.

Vždy se zobrazuje aktuální týden. Na předchozí nebo následující týden se rozvrh přesouvá pomocí tlačítek v *app baru*. Aktuální hodina je zvýrazněna rámečkem, aby pro uživatele bylo jednodušší vybrat hodinu při výuce. Po výběru hodiny se nejprve zkontroluje, zda je zapsáno téma vyučovací hodiny. Bez zapsaného tématu nemůže učitel pokračovat v další práci s hodinou. A to z toho důvodu, že v původní aplikaci SAS uživatelé často zapomínali provést zápis tématu hodiny a bylo velmi problematické doplnit chybějící údaje při uzavírání třídní knihy.

V případě, že učitel vyučuje ve třídě, kterou nemá v rozvrhu (ať už z důvodu, že třídní kniha nebyla předvyplněná nebo protože třídu supluje za jiného učitele), může se přepnout do editačního módu a svůj rozvrh si upravit. K samotné úpravě slouží dialogové okno, kde uživatel zvolí třídu, předmět, vybere, zda je hodina řádná nebo suplovaná, a vyplní název učebny. Při zobrazení dialogu se zmenší rozvrh, tak aby byla pořád vidět zvolená hodina.

U každé hodiny může být zobrazen příznak, zda je hodina suplovaná nebo zda byla zapsána a pokud nebyla, tak zda se jedná o budoucí hodinu nebo minulou. K tomuto účelu jsem využil pravý horní roh dlaždice hodiny a v ní zobrazený symbol podle příznaku (S – pro suplovanou hodinu, ? – pro nezapsanou budoucí hodinu, ! – pro nezapsanou minulou hodinu). Jednotlivé stavy prvku hodiny jsou zobrazeny na obrázku 5.3 (zleva – aktuální hodina, nezapsaná budoucí, nezapsaná minulá, suplovaná).

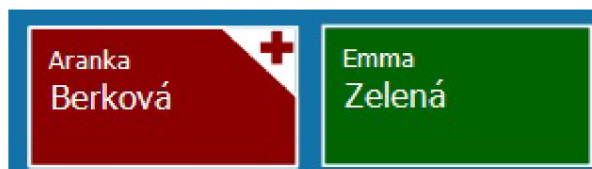


Obrázek 5.3: Jednotlivé stavy hodiny

Finální verzi návrhu obrazovky pro volbu hodiny můžete nalézt v příloze A. Původní návrhy pak na přiloženém CD.

Obrazovka absence

Na obrazovce absence budou učitelé vybírat žáky, kteří nejsou přítomni. Každý žák je reprezentován dlaždicí, která mění barvu podle toho, jestli je žák přítomen (zelená) nebo nepřítomen (červená). Pro lepší rozlišení jsem u nepřítomného žáka přidal červený křížek do pravého horního rohu dlaždice. Rozpoznat přítomnost žáka by tak měla být bezproblémová i pro barvoslepé uživatele (viz obrázek 5.4).



Obrázek 5.4: Jednotlivé stavy žáka pro absenci

Dále se na obrazovce nachází ukazatel, kolik žáků chybělo minulou hodinu a kolik žáků se nachází ve třídě podle vyplněných údajů. Tato informace slouží spíše pro kontrolu přepočítáním žáků ve třídě. Dalším navrženým usnadněním pro učitele je zobrazení žáků podle zasedacího pořádku. Aby tato možnost dobře fungovala, musí nejprve učitel přesunout žáky na správná místa. Tato činnost by měla být provedena pouze jednou pro každou třídu. Učitel pak klikne jen na žáky, po kterých v učebně zbylo prázdné místo a v programu toto místo bude červené. V případě příchodu, respektive odchodu se zaznamená i čas změny stavu a podle toho bude možné z původní aplikace omlouvat jen část hodiny.

Zadávání důvodu absence je dostupné pouze z původního systému SAS, protože omlouvání absencí je umožněno pouze třídním učitelům. Pro ostatní by tato funkce představovala pouze komplikaci nebo funkci navíc, kterou nevyužijí. Finální návrh obrazovky je přiložen jako příloha **B**.

Obrazovka klasifikace

Při zadávání klasifikace jsem uvažoval dva možné způsoby zadávání známek. První z nich jsem pracovním pojmenoval řádkový. Učitel zvolí žáka a může zadávat jednu známku za druhou (pro každou známku zvolí potřebné údaje). Tento způsob je vhodnější, pokud učitel nezkouší žáky rovnoměrně. Druhý způsob jsem pracovním nazval sloupcový. Sloupec známek je společný pro všechny žáky. Učitel nejprve nadefinuje sloupec tak, že zadá potřebné údaje pro celý sloupec, a pak již zadává známky jednotlivým žákům. Všichni žáci tak mají zhruba stejný počet známek. Výjimkou jsou pouze absentující žáci nebo žáci, kteří jsou dozkušováni, pokud se učitel nemůže rozhodnout, jakou konečnou známku jim udělit. V online dotazníku uživatelé spíše upřednostňovali druhou variantu – tedy sloupcové zadávání známek.

Na obrazovce klasifikace se nachází tabulka, ve které jsou na řádcích zobrazení všichni žáci a ve sloupcích pak jednotlivé známky. Pokud žák není v hodině přítomen, bude u něj zobrazen červený křížek. Ve druhém sloupci se nachází vážený průměr všech známek každého žáka. Při přidávání nových sloupců se zobrazí dialog, ve kterém uživatel zvolí parametry sloupce. Základními parametry jsou typ zkoušení, váha známky a název zkoušení. Sekundárními parametry jsou téma zkoušení a datum zkoušení. Téma nemusí být společné pro všechny známky, ale je možné je zadávat až při vkládání známek (především pro ústní zkoušení, kdy jsou žáci zkoušeni postupně v každé hodině určitý počet na jinou látku). Datum je v obou případech zadáno automaticky podle zvolené hodiny. Pokud se celá tabulka neveleze na obrazovku, bude možné informace rolovat, pomocí posuvníků nebo gesta. A to tím způsobem, že jména žáků a vážený průměr zůstane ukotven při horizontálním posouvání a názvy sloupců zůstanou ukotveny při vertikálním posunutí.

Známky budou tvořeny z dlaždic, ve kterých bude zobrazena pouze informace o známce. Pokud známka doposud nebyla žákovi zadána, zobrazí se jeden ze dvou symbolů. Znak mínus ($-$), pokud žák zatím nedostal známku. Ve druhém případě je v rohu známky znázorněn červený kříž, a to pokud se jedná o sloupec se společným datem pro všechny známky a žák v tento den není přítomen. Jednotlivé stavy známky jsou vidět na obrázku 5.5 (zleva – nezadaná známka, žák nepřítomen v den zkoušení, vyplněná známka). V případě, že žák bude dozkoušen z látky, na kterou chyběl, bude ke známce vždy přiřazeno datum hodiny, ve které byla známka zadána.

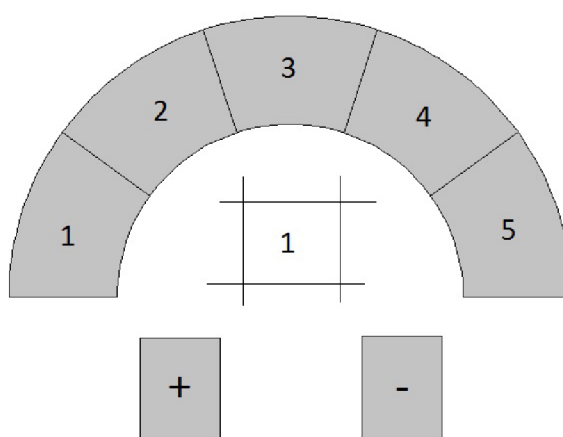
Po kliku na místo, kde má být známka, se zobrazí prvek znázorněný na obrázku 5.6. Pro jednoduchost bude možné zadávat pouze známky 1 až 5 s použitím plus nebo mínus za známku. Klikem na tlačítko $+$ nebo $-$ se změní hodnoty z celých známek na hodnoty se znaménkem. Hodnota známky je tedy stupňována po 0,33 bodech v rozsahu od 1,00 po



Obrázek 5.5: Jednotlivé stavy známky

5,00. Pro 1– je hodnota 1,33, pro 2+ pak 1,66, pro 2 je hodnotu 2,00. Vybraná známka se pak objeví v místě, na které bylo kliknuto před zobrazením ovládacího prvku.

Finální verze obrazovky klasifikace je zobrazena na v příloze C. Původní návrhy jsou uloženy na příloženém CD.



Obrázek 5.6: Ovládací prvek pro zadávání známek

5.3 Implementace ukázky uživatelského rozhraní

Rozhodl jsem se implementovat ukádku uživatelského rozhraní v Delphi XE4 na platformě Firemonkey 2. Již hotové objekty vzhledově odpovídají uživatelskému rozhraní Windows 8. Hlavní výhoda spočívá v tom, že pomocí tohoto prostředí je možné vytvořit aplikaci, která vzhledově odpovídá Windows 8, ale je možné ji spustit i v jiném operačním systému. Na základě rozšíření stávajících operačních systémů předpokládám, že testování bude probíhat pod operačními systémy Windows XP (38,31% trhu) nebo Windows 7 (44,72% trhu), protože Windows 8 (3,82% trhu) zatím tvoří jen minoritní podíl na trhu [14].

Kvůli absenci *Charm* na starších Windows bylo nutné při implementaci přidat tlačítka pro odhlášení a ukončení aplikace. Obě tlačítka jsem umístil v navigačním panelu. Bez nich by nebylo možné aplikaci ukončit, popřípadě se odhlásit, aby se mohl přihlásit jiný uživatel.

Protože v tomto prostředí nefunguje automatická změna rozměrů prvku podle DPI (dot per inch – počet bodů na palec), musel jsem implementovat vlastní funkci, která potřebné prvky zvětší nebo zmenší podle rozměru obrazovky. Poměr jsem spočítal jako velikosti ovládacího prvku ku maximální možné velikosti. Hodnoty jsou vypočteny jak pro šířku, tak pro výšku prvku. Z obou hodnot zvolím tu menší a přiřadím ji do atributu *scale* (měřítko)

daného prvku. Volím jednu hodnotu, aby zůstal zachován poměr stran prvku, a menší hodnotu z toho důvodu, aby prvek nepřesahoval mimo obrazovku.

Pro úvodní obrazovku a pro přihlašovací obrazovku jsem využil dostupné prvky, které odpovídají zásadám Windows 8. Pro další obrazovky jsem musel vytvořit vlastní ovládací prvky. Všechny jsou založeny na třídě *TPaint* z důvodu jednoduchého vykreslení vlastního obsahu. Pro obrazovku volby hodiny jsou vytvořeny třídy *TTyden* a *THodina*. *TTyden* reprezentuje rozvrh a práci s ním (např. změna týdne). Obsahuje několik prvků *THodina*, respektive dvakrát počet vyučovacích hodin krát počet pracovních dnů. Dvě sady prvků *THodina* slouží k demonstraci animace při změně týdne rozvrhu. V prvku *THodina* se kromě zkratky předmětu zobrazují i informace ohledně třídy a učebny. Volná hodina je znázorněna znaménkem plus. To se však zobrazuje pouze, pokud je prvek *TTyden* v módu úprav. Při úpravě zadání hodiny se vysune panel se standardními ovládacími prvky ze spodního okraje obrazovky. *TTyden* se zmenší tak, aby byla vždy vidět vybraná hodina, která se edituje.

Pro obrazovku absence jsem vytvořil taktéž dva prvky. *TAbsence* reprezentuje celou učebnu a obsahuje jednotlivá *TMista*. *TMisto* znázorňuje buď prázdné místo ve třídě nebo žáka, který na místě sedí. Pokud žák není přítomen, je *TMisto* podsvíceno červenou barvou a v rohu je zobrazen červený křížek. Pokud žák je přítomen, dané místo je podsvíceno zelenou barvou. Pokud se *TAbsence* nachází v módu úprav, je možné pomocí techniky *drag and drop* změnit pozici prvku *TMisto*. *TAbsence* může místo zasedacího pořádku zobrazit i abecedně seřazené žáky. V tomto pohledu se prázdná místa nezobrazují. V ukázce jsou použita testovací data pouze pro jednu třídu a jeden předmět. Tato data se zobrazí při zvolení libovolné třídy a předmětu.

Na obrazovce klasifikace se nachází jedna tabulka, kde řádky jsou nadepsány jmény žáků (prvek *TTabZak*). Druhý sloupec pak zobrazuje vážený průměr vypočítaný ze známek v ostatních sloupcích. Sloupce jsou reprezentovány prvkem *TSloupec*. Poslední sloupec slouží pro vytvoření nového sloupce. Po kliku na tento sloupec se zobrazí dialog, ve kterém uživatel zadá název sloupce, váhu sloupce na stupnici od 0 do 100, a pak zda má sloupec společné datum a téma pro všechny známky. Pokud zvolí společné téma, musí ještě zadat téma celého sloupce. Pokud zvolí společné datum, bude všem známkám připsáno datum hodiny, která je vybrána na obrazovce volba hodiny. Po vytvoření sloupce se všem žákům přiřadí prázdná známka.

Známky tvoří prvek *TZnamka*. Po kliku na *TZnamku* se nejdříve zobrazí panel pro zadání tématu zkoušení (pokud se nejedná o sloupec se společným tématem) a poté sada známek, ze kterých uživatel jednu vybere. Znamku je možné kdykoliv změnit nebo smazat.

V případě velkého množství sloupců nebo žáků je potřeba celou tabulku posouvat. Aby zůstaly sloupce ukotveny při posouvání, rozdělil jsem obsah do několika panelů a posouvání obsahu naprogramoval ručně. Pro panely posouvané pouze jedním směrem jsem využil dva panely. Jeden panel má pevnou šířku (*panelSloupce*) a pozici a druhý panel se roztahuje podle počtu sloupců (*PanelPosunSloupce*). Z obsahu panelu *PanelPosunSloupce* je zobrazena pouze ta část, která se nachází na panelu *panelSloupce*. Řešení je zobrazeno na obrázku 5.7. Obrys panelu *PanelPosunSloupce* je červený. Obrys panelu *panelSloupce* je černý. Viditelná část panelu *PanelPosunSloupce* je vybarvena světle modrou barvou. Stejně řešení jsem použil i při horizontálním posouvání, při kterém zůstávají ukotveny jména žáků a jejich průměr.



Obrázek 5.7: Zobrazení pouze části obsahu panelu

5.4 Testování uživatelského rozhraní školního systému

Testování probíhalo od první fáze vývoje na *lo-fi* modelech. Jako testující uživatelé mi pomohli zaměstnanci firmy MP-Soft, a. s. Testování probíhalo tak, že jsem měl původní návrhy jako výkresy na obrazovce. Simuloval jsem změnu prvků podle akcí uživatele. V této fázi se podařilo odhalit mnoho chyb a v podstatě změnit vzhled všech obrazovek.

Po dokončení *lo-fi* jsem implementoval ukázkou ve vývojovém prostředí Delphi XE4 s použitím grafických komponent Firemonkey 2. V modelu již byla implementována reálně vypadající data. Tento model měl být rozšířen mezi koncové uživatele společně s dotazníkem na jejich spokojenost s uživatelským rozhraním. Popřípadě využít při testování ve speciální místnosti, kde by byla sledována interakce uživatelů s modelem. Toto řešení by však bylo časově velmi náročné a proto jsem se rozhodl opět jako uživatele oslovit zaměstnance firmy MP-Soft, a. s. Při testování jsem byl přítomen u uživatele a dával mu úkoly, které má plnit. Při sledování jejich chování a na základě jejich připomínek jsem objevil chyby uvedené v tabulce 5.1.

Popis chyby	Chybu objevilo	Důsledek chyby	Řešení
Nevhodné rozložení hodnot známek	5 z 5 uživatelů	Chybné známkování žáků	Kromě celých známek jen známky s mínusem (pro 1– hodnota 1,5 apod.)
Splynutí dialogu s formulářem na pozadí	4 z 5 uživatelů	Špatná orientace uživatele	Překrytí formuláře šedým poloprůhledným obdélníkem
Nevýrazné označení aktuální hodiny v rozvrhu	3 z 5 uživatelů	Pomalejší volba aktuální hodiny	Zvýraznění aktuálního dne a hodiny v hlavičce rozvrhu
Uživatelé očekávali více možností zobrazení dat v klasifikaci	3 z 5 uživatelů	Neočekávané chování aplikace	Do budoucna dodělat filtrování a řazení po kliku na sloupec klasifikace

Tabulka 5.1: Nalezené chyby při testování

Kapitola 6

Závěr

Cílem této práce bylo navrhnout efektivní informační systém pro učitele na základních a středních školách. Vytvořil jsem tedy jednoduchou aplikaci bez zbytečných funkcí navíc. Nejprve jsem shromáždil údaje o cílových uživateli pomocí online dotazníku. Svůj návrh jsem pak upravil i podle takto zjištěných informací. Aby byl celý systém pro uživatele atraktivnější, návrh jsem vytvořil ve stylu moderního operačního systému Windows 8. Díky tomu bude systém možné využívat jak na tabletech, tak i na notebookech a stolních počítačích. Pokud bude mít systém nainstalován každý uživatel na svém tabletu, bude zápis informací v hodině velmi rychlý a efektivní. Navíc si bude učitel nosit důležité informace pořád u sebe.

Pro vypracování práce jsem nastudoval moderní trendy v návrzích uživatelských rozhraní. Shrnutí získaných znalostí je uvedeno v kapitole 2. Dále jsem vyzkoušel uživatelská rozhraní již hotových informačních systémů pro správu školní matriky a analyzoval reakce uživatelů na tato rozhraní v kapitole 3. Na základě získaných informací jsem navrhl vlastní uživatelské rozhraní nového systému. Návrh je popsán v kapitole 5. Implementovaná ukázka tohoto rozhraní je uložena na přiloženém CD a implementace je popsána v sekci 5.3.

Oproti dostupným systémům bych jako hlavní výhodu viděl jednoduchost a přímoučarost. Všechny systémy, které jsem mohl vyzkoušet, jsou komplikované a obsahují velké množství funkcí, které běžný učitel vůbec nevyužije. Pro mnou navržené uživatelské rozhraní postačí pouze 5 formulářů. První z nich je úvodní obrazovka, druhá slouží pro přihlášení a další tři využijí uživatelé pro zápis informací v hodině. Na těchto formulářích se nachází prvky, které využívají technik *drag and drop* a *look and feel*. Díky tomu si uživatelé při testování pochvalovali efektivnost a rychlost navrženého uživatelského rozhraní.

Implementoval jsem ukázku uživatelského rozhraní a podrobil ji testování uživatelům, kteří znají školní prostředí. Během testování byly odhaleny drobné nedostatky navrženého rozhraní. Tyto nedostatky jsem se pokusil co nejlépe opravit.

Do budoucna bych chtěl provést testy na koncových uživateli. Na tyto testy nebyl v bakalářské práci prostor. Taktéž by mohlo být užitečné navrhnout další pohledy na známky žáků. Například řazení žáků podle počtu známek nebo řazení žáků podle známek v určitém sloupci. Myslím že i filtrování známek by určitě našlo využití, zejména pak při rozhodování, koho v hodině vyzkoušet. U absence by bylo na zvážení navrhnout vazbu na předmět a učebnu, kdy by kromě standardního rozesazení mohlo být například v laboratořích speciální rozmístění lavic. K tomuto řešení by byla potřeba i editor umístění lavic v učebně.

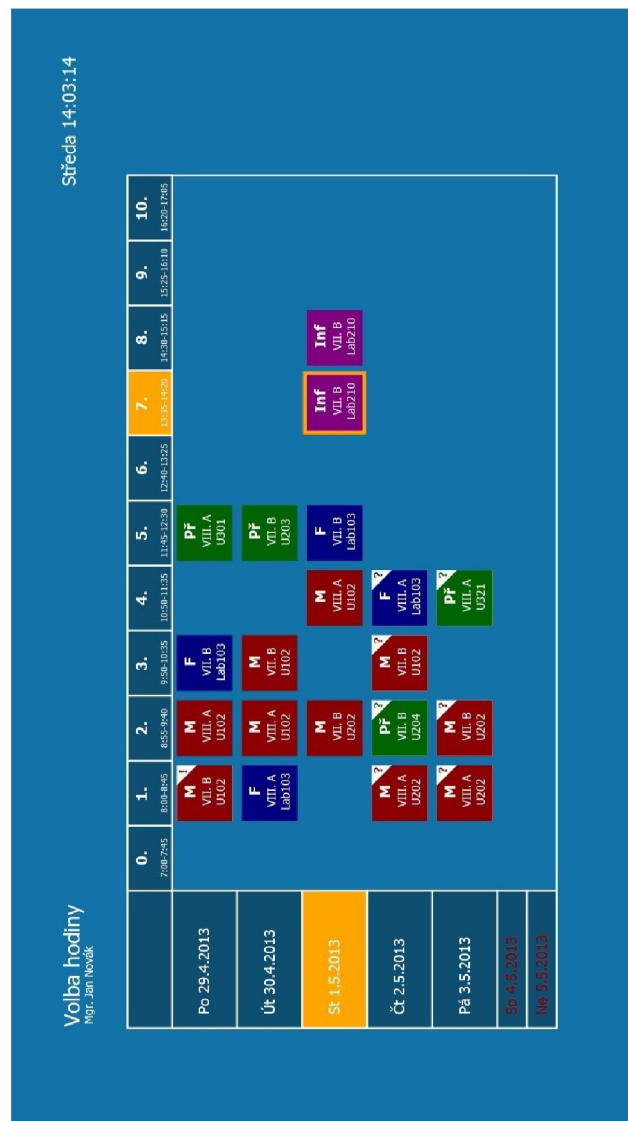
Literatura

- [1] HOŘAVA, Michal a kol.: *Uživatelsky přívětivá rozhraní*. Horava and Associates, Praha, 2009, iISBN 978-80-254-5295-0.
- [2] WOOD, L. E.: *User interface design*. CRC Press, New York, 1998, iISBN 0-8493-3125-0.
- [3] KRUG, S.: *Don't make me think*. New Riders, Berkeley, 2006, iISBN 0-321-34475-8.
- [4] ZELENÝ, Jaroslav a MANNOVÁ, Božena: *Historie výpočetní techniky*. Scientia, Praha, 2007, iISBN 80-86960-04-8.
- [5] BENYON, David, TURNER, Phil and TURNER, Susan: *Designing interactive systems: people, activities, contexts, technologies*. Essex, první vydání, Addison-Wesley, 2005, 789 s., iISBN 0-321-11629-1.
- [6] FORET, M.: *Marketingová komunikace*. Brno, první vydání, BizBooks, 2006, 464 s., iISBN 978-80-251-1041-6.
- [7] ANDERSON, S. P.: *Přitažlivý interaktivní design: jak vytvářet uživatelsky přívětivé produkty*. Brno, první vydání, Computer Press, 2012, 240 s., iISBN 978-80-251-3722-2.
- [8] APPLE INC.: Mac OS X Human Interface Guidelines [online]. Apple Inc. 2011, [cit. 2013-03-16].
URL <http://developer.apple.com/library/mac/#documentation/UserExperience/Conceptual/AppleHIGuidelines/Intro/Intro.html>
- [9] BBC: Microsoft to drop 'Metro' name for Windows 8. 2012, [online], rev. 3. srpna 2012 [cit. 2013-03-21].
URL <http://www.bbc.co.uk/news/technology-19108952>
- [10] MICROSOFT: Windows 8 User experience guidelines [pdf]. Microsoft 2012-08-14, [cit. 2013-01-16].
URL <http://msdn.microsoft.com/en-us/library/windows/apps/hh465424.aspx>
- [11] MICROSOFT: Windows User Experience Interaction Guidelines for Windows 7 and Windows Vista [pdf]. Microsoft 2010-09-27, [cit. 2013-01-16].
URL <http://www.microsoft.com/download/en/details.aspx?id=2695>
- [12] BUREŠ, J.: Nové uživatelské rozhraní aplikace SAS (výsledky průzkumu). 2012, [online], rev. 21. listopadu 2012 [cit. 2013-03-21].
URL <http://sas1.vyplnto.cz>

- [13] KLEŇHOVÁ, Michaela a kol.: *Krajská ročenka školství 2010*. Ústav pro informace ve vzdělání, Praha, 2011, iSBN 978-80-211-0618-5.
- [14] NET APPLICATIONS.COM: Desktop Top Operating System Share Trend. 2012, [online],[cit. 2013-08-05].
URL <http://www.netmarketshare.com/>

Příloha A

Finální verze volba hodiny



Obrázek A.1: Návrh obrazovky určené pro volbu hodiny

Příloha B

Finální verze absence



Obrázek B.1: Návrh obrazovky určené pro zadávání absence

Příloha C

Finální verze klasifikace

Neděle 12.47:16

Klasifikace VILLA Matamoras		+									
Umění	Vážený průměr	test Tomlin	test Smith	Převod jednotek	Energie	Ústř.	Čtyřlístek	Dů	Elektronika	Magnetsmus	
Aranika Berková	3,62	5	1-	4	1	5	+	5	4-	3-	
Kateřina Černíhková	2,66	2	+	5	2-	4-	1-	3-	1	2	
František Dohnal	3,32	4	2	3	4	1-	5	5	1-	3	
Květoslava Dostálová	1,93	3	1-	3-	1	1	+	4-	1-	1	
Věra Eichlerová	3,35	4-	2	1-	2	4-	4	2-	3-	4-	
Miloslav Faltejšek	2,96	+	3-	3-	2	4-	3	1	1	3-	
Marcela Fejfarová	3,18	1	3-	2	4-	1-	4	-	3-	5	
Alena Filousová	3,45	1-	5	1	4	4	5	1-	1-	4-	

Obrázek C.1: Návrh obrazovky určené pro zadávání klasifikace

Příloha D

Obsah CD

- Návrhy obrazovek
- Ukázka uživatelského rozhraní včetně zdrojových kódů a příručky k ovládání
- Text technické zprávy
- L^AT_EXové zdrojové kódy technické zprávy
- Výsledky online průzkumu