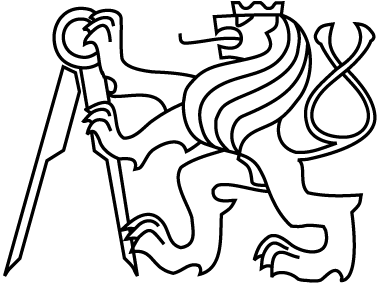
České vysoké učení technické v Praze  
Fakulta elektrotechnická  
Katedra počítačů



Bakalářská práce

Mobilní část k projektu SMART-FINE, systému pro evidenci dopravních přestupků a pohybu policistů

*Martin Štajner*

Vedoucí práce: Ing. Martin Komárek

Studijní program: Softwarové technologie a management

Obor: Softwarové inženýrství

12. února 2012

Poděkování

Zde můžete napsat své poděkování, pokud chcete a máte komu děkovat.

Prohlášení

Prohlašuji, že jsem práci vypracoval samostatně a použil jsem pouze podklady uvedené v přiloženém seznamu.

Nemám závažný důvod proti užití tohoto školního díla ve smyslu §60 Zákona č.121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon).

V Praze dne XX.XX.2012 Martin Štajner

Abstract

Projekt SMART-FINE se zabývá tvorbou klienta (mobilní aplikace) systému pro evidenci dopravních přestupků pomocí mobilních telefonů. Ten by měl usnadnit policistům práci s udělováním parkovacích lístků a kontrolou parkování. Eliminace papírových bločků, možnosti rychlých voleb a automatické před-vyplňování práci urychlí a dostupnost informací v digitální podobě usnadní následnou administrativu.

Práce se zabývá analýzou problému, návrhem aplikace, jejím samotným vývojem a testováním. Při vývoji byl použit programovací jazyk Java, jako cílová zařízení jsou telefony s operačním systémem Android. Aplikace spolupracuje se serverovou částí systému, kterou se zabývá ve své bakalářské práci kolega Pavel Brož [[1]](#Bakalarka_Pavel_Broz).

Abstrakt

Projekt SMART-FINE se zabývá tvorbou klienta (mobilní aplikace) systému pro evidenci dopravních přestupků pomocí mobilních telefonů. Ten by měl usnadnit policistům práci s udělováním parkovacích lístků a kontrolou parkování. Eliminace papírových bločků, možnosti rychlých voleb a automatické před-vyplňování práci urychlí a dostupnost informací v digitální podobě usnadní následnou administrativu.

Práce se zabývá analýzou problému, návrhem aplikace, jejím samotným vývojem a testováním. Při vývoji byl použit programovací jazyk Java, jako cílová zařízení jsou telefony s operačním systémem Android. Aplikace spolupracuje se serverovou částí systému, kterou se zabývá ve své bakalářské práci kolega Pavel Brož [[1]](#Bakalarka_Pavel_Broz).

Obsah

Zadání

Navrhněte, implementujte a testujte mobilní aplikaci představující „klientskou” část systému pro evidenci dopravních přestupků. Aplikace bude běžet na mobilním telefonu s operačním systémem Android a bude umožňovat vyplnění parkovacího lístku, kontrolu odcizení přenosné parkovací karty a kontrolu zaplacení parkování pomocí SMS. Mobilní klient bude využívat podporu „serverové” části, (převážně se jedná o odesílání dat na server), kterou bude zajišťovat kolega Pavel Brož v rámci své bakalářské práce [[1]](#Bakalarka_Pavel_Broz).

# Kapitola

Úvod

## Motivace

Téma „Evidence dopravních prostředků pomocí mobilních telefonů“ jsem si vybral ze seznamu nabízených témat pro předmět A7B36SI2 – Řízení softwarových projektů [[2]](#A7B36SI2), kde jsme na projektu pracovali v týmu. Téma nebylo před tím zpracováno jiným týmem, na který bychom museli navazovat, tudíž jsme začínali úplně od začátku. Po ukončení tohoto předmětu jsem se rozhodl aplikaci dále vyvíjet v rámci bakalářské práce, neboť shledávám téma jako zajímavé.

## Problém a cíl projektu

Kdykoliv v současnosti dojde k přestupku při parkování, policista na místě vyplňuje parkovací lístek v podobě papírového formuláře. Musí si pamatovat čísla paragrafů zákona, zjišťovat adresu, kde se událost odehrála, několikrát opisovat různé informace dokola atd. Papírové formuláře se mohou kdykoliv ztratit, jsou nepraktické na přenášení a zvyšují administrativu.

Cílem projektu je vytvoření systému pro mobilní zařízení (mobilní telefon), které bude usnadňovat kontrolu parkování a bude nahrazovat manuální psaní parkovacích lístků. V České republice mi době vývoje nebyl znám jiný elektronický systém řešící tuto problematiku.

Více v 2. kapitole a příloze Přehled o projektu [A].

## Struktura práce

Bakalářská práce je psána iterativním způsobem. Každá iterace má určité cíle, díky jimž dochází ke změnám v systému a dokumentaci. Může se jednat o úpravu čehokoliv, co již bylo předtím v projektu provedeno, přidávání nových cílů, nové funkčnosti systému, nebo dokončování cílu nesplněných v předešlých iteracích. Při přidávání nové funkčnosti se znovu provede analýza, návrh, implementace a testování daných změn. Většina kapitol tedy obsahuje popis a důvod všech provedených změn v projektu. Výsledkem spojení výstupů všech iterací dohromady pak vzniká finální projekt, tato bakalářská práce.

TODO???

Součástí bakalářské práce jsou přílohy, na které je na různých místech v tomto dokumentu odkazováno. Hlavními přílohami jsou Přehled o projektu [A], Analýza [B], Návrh [C] a Testování [D], které poskytují doplňující a detailnější informace k vývoji. Seznam všech příloh a přílohy samotné lze najít na konci dokumentu, spolu se seznamem použité literatury, využitých webových odkazů a seznamem zkratek.

# Kapitola

Iterace 1

## Úvod, cíle iterace

Text této kapitoly se zabývá veškerou prací udělanou v rámci předmětu SI2 [[2]](#A7B36SI2), který trval 1 semestr. Právě toto období jednoho semestru představuje 1. iteraci. Jak již bylo zmíněno v úvodní kapitole, na projektu se začalo pracovat úplně od začátku, neboť se doposud tímto tématem nikdo jiný nezabýval.

Jelikož se jedná o 1. iteraci, jejím cílem je hlavně vymezení samotného projektu a jeho zahájení. Po vymezení projektu a ujasnění si, čím se má vlastně zabývat je dalším důležitým cílem stanovení funkčních a nefunkčních požadavků na systém - sepsání jednotlivých funkcí, kterými má systém disponovat, co má vlastně umět.

### Seznam funkčních požadavků

1. Vyplnění parkovacího lístku
2. Nápověda při vyplňování čísel paragrafů zákonů nejběžnějších přestupků
3. Prohlížení lokálně uložených parkovacích lístků
4. Úprava lokálně uložených parkovacích lístků
5. Odeslání parkovacích lístků z mobilního zařízení na server
6. Tisk parkovacích lístků pomocí mobilní tiskárny
7. Pořízení fotodokumentace přestupku
8. Rozpoznání státní poznávací značky z fotografie
9. Automatické vyhledání adresy pomocí GPS
10. Kontrola odcizení přenosné parkovací karty opravňující parkovat v modré zóně
11. Přístup k funkcím aplikace po zadání identifikačního čísla

### Seznam nefunkčních požadavků

1. Mobilní zařízení (mobilní telefon)
2. Mobilní tiskárna s datovým připojením přes Bluetooth
3. Přístup k serveru Smart-Fine

Sběr požadavků byl bezproblémový, bylo na první pohled jasné, co by asi aplikace tohoto typu měla zvládat a co za funkce nabízet. Detailnější popis funkčních a nefunkčních požadavků představovaný vypsáním „pod-požadavků“ je v příloze Přehled o projektu [A].

Po určení požadavků je dalším cílem analýza, návrh implementace a samotná implementace co nejvíce z nich, počínaje od těch nejzákladnějších (zhruba stejné pořadí jako v seznamu). Součástí vývoje je samozřejmě také testování hotové naimplementované funkcionality.

## Vymezení projektu, problém, cíle, přínosy

Jak již bylo popsáno v úvodní kapitole, systém Smart-Fine se zabývá problematikou kontroly parkování a vypisování parkovacích lístků. Místo vypisování parkovacích lístků se informace budou zapisovat do mobilní aplikace, s možností archivace a zálohy na serveru. Tím se ulehčí práce s papírováním, opisováním věcí a celkově administrací. Aplikace si může za uživatele pamatovat, nebo zjišťovat informace a před-vyplňovat je. Např. automatické vyplnění času, aktuální polohy v podobě adresy atd. Další urychlení při vyplnění parkovacího lístku spočívá v použití nabídky nejčastějších voleb. Mobilní telefon v dnešní době většinou dokáže fotografovat, situaci lze tím pádem zdokumentovat i takovýmto způsobem. Všechny informace se odešlou a uloží na server, odkud budou snadno získatelné k další práci. K ovládání nebude potřeba žádných zvláštních dovedností, jen lehké zaškolení v ovládání, vše by mělo být intuitivní. Práce se systémem tak bude jednoduchá a efektivní zároveň.

Další informace k projektu (kontext systému, možné překážky vývoje) jsou v příloze Přehled o projektu [A].

## Případy užití

Po vypsání požadavků na systém v úvodu iterace došlo k určení tzv. případů užití. Jde o definování, jakým způsobem bude systém používán a jak s ním bude manipulováno.

Seznam případů užití:

1. Vyplnění nového parkovacího lístku
   * Uživatel bude schopen vypsat nový parkovací lístek systémem vyplňování formulářových polí. Tímto způsobem bude moct zadat veškeré potřebné informace.
2. Zobrazení seznamu parkovacích lístků
   * Uživatel bude schopen zobrazit seznam všech vyplněných parkovacích lístků od posledního úspěšného nahrání na server.
3. Prohlížení lokálních záznamů
   * Uživatel bude schopen prohlédnout všechny informace již vyplněných parkovacích lístků.
4. Úprava lokálních záznamů
   * Uživatel bude schopen upravit všechny informace již vyplněných parkovacích lístků.
   * Po vytištění parkovacího lístku ho již nebude možno dále upravovat.
5. Odeslání lokálních záznamů na server
   * Uživatel bude schopen, pokud bude připojen k internetu, odeslat všechny lokální záznamy do databáze na server.
   * Odeslat bude možné pouze všechny parkovací lístky najednou, nikoliv po jednom, z důvodu zamezení možnosti zadržování lístků v zařízení. Jde tedy o způsob jak zajistit, že nedojde ke korupčnímu jednání.
6. Kontrola odcizení parkovací karty
   * Systém bude umožňovat zkontrolovat, zda je přenosná parkovací karta odcizená.
7. Tisk parkovacích lístků
   * Uživatel bude schopen vytisknout všechny lokální záznamy na mobilní tiskárně.
   * Po vytištění již dále nebude možné parkovací lístek upravit. TODO: , nebo smazat.
8. Autentifikace uživatele
   * Uživatel se bude muset při přístupu do aplikace a některých funkcí přihlásit pomocí identifikačního čísla a číselného hesla (PIN). Jedná se o bezpečnostní a monitorovací prvek.
   * K přihlášení je potřeba internetové připojení. Některé funkce budou umožňovat přihlášení i bez připojení k internetu.

Diagram ukazuje případy užití a uživatelské role, a souvislosti mezi nimi. Uživatelská role je v tomto případě pouze jedna, policista v terénu, což znamená, že všechny případy užití, veškerou manipulaci s aplikací, bude prováděn pouze on. Vymodelováno v Enterprise Architect [4] pomocí jazyka UML:

D:\Dokumenty\skola\Bakalarska prace\Iterace1\Diagramy\Případy užití.emf

Diagram bla bla TODO spravne udelat popisek s cislem

Požadavky na systém a scénáře užití spolu úzce souvisí. Každý požadavek musí být zahrnut v nějakém případu užití, aby došlo k jeho realizaci. Jde vlastně říci, že případy užití využívají funkční požadavky, neboli pokud bude chtít uživatel provést určitou činnost, tak aplikace musí obsahovat příslušnou funkčnost. Diagram ukazuje mapování funkčních požadavků na jednotlivé případy užití. Vymodelováno v Enterprise Architect [4] pomocí jazyka UML:

D:\Dokumenty\skola\Bakalarska prace\Iterace1\Diagramy\Model závislosí požadavků a případů užití.emf

*Model závislostí požadavků a případů užití* **TODO**

S případy užití souvisí tzv. scénáře užití popisující posloupnost kroků, které musí být vykonány za účelem provedení dané činnosti. Scénář zahrnuje jak kroky uživatele, tak kroky systému. Jejich seznam je vypsán a jednotlivé scénáře popsány v příloze Analýza [B].

## Doménový model, datové entity v systému

V systému jsou zatím pouze 2 hlavní datové domény, se kterými je potřeba pracovat. Jedná se o entity, které v sobě ukládají nějaká data. Jsou jimi parkovací lístek (parking ticket) a odkaz na zákon (law). Každý parkovací lístek musí obsahovat právě 1 odkaz na příslušný zákon. Další informace, které charakterizují parkovací lístek, jsou vypsány v podobě atributů objektu. Např. parkovací lístek musí obsahovat datum, adresu, informace o vozidle apod. Odkaz na zákon musí mít např. číslo zákona, paragraf atd. Při zjišťování a sepisování správných atributů se vycházelo z parkovacího lístku, který momentálně používá policie města Prahy. Jeho položky jsou:

### Položky papírového parkovacího lístku

* Datum a čas
* SPZ – státní poznávací značka
* MPZ – mezinárodní poznávací značka
* Barva SPZ – např.: bílá, oranžová
* Druh vozidla - např.: osobní, nákladní
* Tovární značka vozidla - např.: Škoda, Ford
* Adresa - město, ulice, č. popisné
* Místo - určení místa, kde došlo k přestupku, např.: číslo lampy veřejného osvětlení
* Popis jednání (část pro policistu) - kód přestupku
* Označení, zda je přítomná dokumentace
* Označení, zda byl přestupek na pohyblivém dopravním značení
* Identifikační číslo policisty
* Popis jednání (část pro řidiče) – popis dopravního přestupku
* Podpis policisty

Doménový model je diagram, který obsahuje objekty daných entit, popisuje jejich vztahy a vypisuje atributy. Vymodelováno v Enterprise Architect [4] pomocí jazyka UML:

D:\Dokumenty\skola\Bakalarska prace\Iterace1\Diagramy\Doménový model.emf

*Domenovy model bla* TODO

Seznam a význam všech atributů jednotlivých entit je vypsán v příloze Analýza [B].

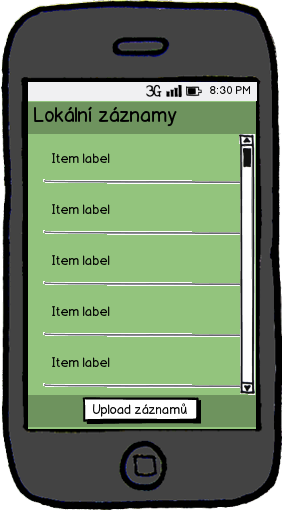
## Platforma, vývojové prostředí

Při výběru cílové platformy mobilních zařízení byl kladen důraz na velikou rozšířenost. Z toho důvodu byl vybrán Android OS, který se v posledních málo letech velmi rozrostl, konkrétně minimální verze 2.1. Aplikace pro tento operační systém jsou vyvíjeny v jazyku Java a pomocí speciálních knihoven, tzv. Android SDK [5], a tak tomu tak je i v tomto projektu. Jako vývojové prostředí bylo použito IDE Eclipse [6]. Testovat lze kromě vlastního mobilního zařízení také na emulátoru, který je součástí Android SDK. Na emulátoru však nemusí být přístupné všechny funkce, z důvodu jeho omezení. Při vývoji byl k dispozici mobilní telefon HTC Desire, vývoj i testování tedy probíhaly na tomto zařízení.

## Uživatelské prostředí

Návrh uživatelského prostředí byl vytvořen pomocí online nástroje Balsamiq Mockup [[3]](#Balsamiq_Mockup). Byly navrženy jednotlivé obrazovky, jak by asi mohly vypadat ve finální aplikaci. Jde pouze o orientační znázornění, naznačuje spíše strukturu, neboť tento nástroj zdaleka neumí vyjádřit pravý design aplikace. Návrh je k dispozici také jako příloha [I], včetně popisů jednotlivých obrazovek. Následující obrázky ukazují úvodní obrazovku, vytváření nového parkovacího lístku, seznam uložených parkovacích lístků, detail již vyplněného parkovacího lístku a odesílání lístků na server.

TODO zase nejakej popisek bla bla

Jak je vidět z obrázku, hlavní menu obsahuje pouze navigaci po aplikaci v podobě tlačítek. Vytváření nového parkovacího lístku je prakticky pouhé vyplnění několika formulářů, To lze usnadnit výběrem položky z nabídky nejčastějších voleb, která se vyvolá stiskem postraních tlačítek, nebo použitím automatického před-vyplnění. Celý lístek se uloží stiskem tlačítka ve spodu obrazovky. Seznam záznamů obsahuje všechny neodeslané parkovací lístky. Po kliknutí na položku se zobrazí její detail. Ten se dá vytisknout, či upravit stiskem tlačítka ve spodu obrazovky. Všechny záznamy lze naráz nahrát na server pomocí stisku tlačítka ve spodu na obrazovce seznamu lístků.

Z hotových návrhů obrazovek vznikl první prototyp [E] v podobě klikacího PDF souboru, který ukazuje a popisuje, jak by asi aplikace mohla fungovat ve velmi zjednodušené podobě. V prototypu není zahrnuta prakticky žádná funkčnost, opět se jedná pouze o primitivní návrh.

Jak celá aplikace vypadala po grafické stránce na konci 1. iterace, je ukázáno v příloze Návrh [C].

## Implementace

Struktura aplikace bude dodržovat návrhový vzor MVC (Model, View, Controller). Funkci části Controller budou zajišťovat třídy aktivit. Jde o třídy představující jednotlivé obrazovky aplikace, každá aktivita tedy bude mít svoje grafické rozhraní (View). Většina případů užití je implementována právě pomocí jedné aktivity. Část View bude pro lepší přehlednost řešena pomocí XML souborů obsahující popis grafického rozhraní, nikoliv napsaná přímo v kódu aplikace. Modelem budou datové třídy, představující jednotlivé entity, jak bylo dříve zmíněno v sekci Doménový model.

Při vývoji bude použito návrhových vzorů. Konkrétně návrhový vzor DAO (Data access object), který odstiňuje zdroje dat a umožňuje tak lehké použití různých implementací ukládání dat, např. do souboru, nebo do databáze. Dalším použitým návrhovým vzorem je Singleton (Jedináček), který umožňuje znovu použití určitých objektů.

### Seznam balíčků – struktura aplikace

* cz.smartfine - obsahuje převážně třídy aktivit
* dao - poskytuje přístup k souborové vrstvě
* model - obsahuje datové třídy entit
* model.util - obsahuje pomocné třídy aplikace
* model.validator - obsahuje validační třídy
* res.layout - obsahuje XML soubory s grafickým prostředím

Diagram ukazuje strukturu balíčku cz.smartfine, který má funkci části Controller. Jednotlivé třídy již obsahují datové typy svých proměnných a metody. Vymodelováno v Enterprise Architect [4] pomocí jazyka UML:

D:\Dokumenty\skola\Bakalarska prace\Iterace1\Diagramy\smartfine.emf

TODO bla bla

Diagram ukazuje strukturu balíčku dao. Vymodelováno v Enterprise Architect [4] pomocí jazyka UML:

D:\Dokumenty\skola\Bakalarska prace\Iterace1\Diagramy\dao.emf

TODO bla bla

Diagram ukazuje strukturu balíčku model, který má funkci části Model. V diagramu je oproti doménovému modelu nová třída Settings. Jde o preferenece, která si uživatel může nastavit, zatím lze pouze přednastavit město a adresu serveru Smart-Fine. Třída není obsažena v doménovém modelu, protože její objekt vyloženě neukládá data jako třídy parkovacího lístku a odkazu na zákon. Je však zařazena do modelu, jelikož zajišťuje přístup k datům uloženým v kontextu aplikace. Širší využití třídy settings nastane až v dalších iteracích. Vymodelováno v Enterprise Architect [4] pomocí jazyka UML:

D:\Dokumenty\skola\Bakalarska prace\Iterace1\Diagramy\model.emf

TODO bla bla

Diagramy ostatních balíčků jsou v příloze Návrh [C]. Kompletní popis implementace je zpracován za pomocí JavaDoc [F].

## Testování

Testování programátorem probíhá prakticky po celou dobu vývoje. Hned jak dojde k implementaci jakékoliv části systémů, dojde k jeho otestování. Ta však nejsou nikde uvedena, jelikož jsou drobná, samozřejmá a jsou součástí vývoje. Během první iterace však došlo ke dvěma větším testováním, která provedli uživatelé, jež se na vývoji nepodílí. Tak bylo možné získat informace od nezainteresovaných osob.

### Heuristická evaluace

V počátcích vývoje, kdy byl k dispozici pro testování pouze PDF prototyp [E], proběhla tzv. Heuristická evaluace. Dvou testerům byly zadány úkoly pro práci s prototypem, které provedli, a výsledkem jsou hodnocení obou zúčastněných.

**Pro testování byla použita tato kritéria:**

1. Viditelnost statusu systému
2. Uživatelské prostředí a svoboda pohybu
3. Prevence chyb
4. Design a struktura

**Pro testování byly použity tyto úkoly:**

1. Vytvořte nový lístek a uložte ho.
2. Najděte uložený lístek a upravte ho. Následně ho uložte.
3. Nahrajte všechny záznamy na server.

#### Evaluace - Tester 1 - student ČVUT (Softwarové inženýrství)

V prototypu se lze zatím prakticky pouze přesouvat mezi obrazovkami, veškerá funkčnost aplikace je v tomto typu prototypu nemožná. To samozřejmě značně omezuje zadané úkoly.

Ačkoliv je přesně vidět, kde se člověk zrovna nachází, není zcela jasné, jak se dostat zpět. Znalý uživatel androidu ví, že přes tlačítko “Zpět” na zařízení, neznalý tohoto však může mít problém. (Kritérium 1. Úkol 1, 2, 3)

Důležitou částí navigace jsou tlačítka ve spodní části obrazovky (při vytváření lístku, prohlížení záznamů, prohlížení a editaci lístku). Tlačítko je vidět neustále, což by mohl být problém, v případě, že si uživatel neuvědomí, že může obsahem nad tlačítkem pohybovat. (Kritérium 2, Úkol 1, 2, 3)

Chybí potvrzení pro uživatele, o nějaké vykonané akci. Úspěšné uložení záznamu atd. (Kritérium 3. Úkol 1, 2, 3)

#### Evaluace - Tester 2 - ČVUT (Komunikace a multimédia)

Systém nekontroluje, zda jsou zadaná veškerá data potřebná k identifikaci. (Kritérium 3, Úkol 1, 2)

Klást důraz na popisky, důležité, aby opravdu vystihovaly to, co mají. Např.: “Parkovací lístek” -> “Nový parkovací lístek”.

V aplikaci neexistuje jakékoliv potvrzování rozhodnutí typu “Opravdu chcete smazat lístek?”. Mohl bych omylem udělat něco, co bych nechtěl. (Kritérium 3. Úkol 1, 2, 3)

### Uživatelské testování

Podobně probíhalo také uživatelské testování v závěru iterace. Mělo by ukázat, zda je aplikace intuitivní na ovládání, přehledná a logicky strukturována. Testuje se celkově celá aplikace, uživatel bude provádět jednotlivé úkoly a komentovat je. Opět byli přítomni 2 testeři, kteří prováděli sadu úkolů, tentokrát však měli k dispozici funkční prototyp přímo na mobilních zařízeních. Celý průběh byl monitorován a je k dispozici v podobě videozáznamu [G][H].

**Seznam úkolů pro testování:**

1. Vytvořte a uložte nový lístek, pouze povinné údaje.
2. Vytvořte a uložte nový lístek, opatřete ho fotodokumentací a využijte nápovědy vyplnění.
3. Upravte první parkovací lístek, přidejte mu fotodokumentaci.
4. Vytiskněte druhý lístek.
5. Změňte v nastavení Město, a vytvořte nový lístek.
6. Odešlete všechny lístky na server.

**Výsledek testování:**

1. video - Tester 1 - student ČVUT - Samsung Galaxy S [G]
2. video - Tester 2 - student VŠE - HTC Desire [H]

Oba uživatelé si vedli při testování dobře. Práce byla přímočará a prakticky bez zádrhelů. Uživatelé vždy věděli, kam přesně kliknou a kde hledat tlačítka a formuláře. To poukazuje na intuitivní a přehlednou strukturu aplikace i pro ty, co s Androidem běžně nepracují. Design přišel oběma uživatelům ucházející, ale chtěli by ho nějakým blíže nespecifikovaným způsobem vylepšit.

## Závěr

Jak bylo zmíněno v úvodu kapitoly, 1. iterace probíhala poměrně dlouhou dobu. Během této doby byla z ničeho vytvořena nová aplikace se základním uživatelským rozhraním a funkčností, která bude v dalších přibývat. Byla sepsána dokumentace a vytvořen systém, kterým se vývoj aplikace bude nadále ubírat. I přes dlouhé trvání iterace se však nepodařilo implementovat všechny funkční požadavky. Seznam uvádí, co bylo a nebylo splněno, a co zůstalo do dalšího vývoje. V dalších iteracích bude také kladen důraz na výsledky testování a na komentáře uživatelů.

**Splněné funkční požadavky:**

1. Vyplnění parkovacího lístku
   1. Vyplnění informací o datu a času přestupku, SPZ, MPZ, barvě SPZ, druhu vozidla, tovární značky, města, ulice, čísla, místa, identifikačního čísla, přestupku (paragraf, odstavec, písmeno, zákon, sbírka, popis) a možnost výběru odtahu, přenosné DZ a fotodokumentace.
   2. Všechny vyjmenované informace (kromě výběrových) bude možné zadat za pomoci softwarové klávesnice nebo hardwarové klávesnice integrované do mobilního zařízení.
   3. Systém bude standardně před-vyplňovat údaje: datum, čas, město, identifikační číslo policisty.
   4. Systém bude vyplněné parkovací lístky ukládat do paměti mobilního zařízení
2. Nápověda při vyplňování čísel paragrafů zákonů nejběžnějších přestupků
   1. Systém bude uživateli nabízet výčet nejběžnějších přestupků a po vybrání uživatelem sám vyplní příslušná čísla zákonů a popis do parkovacího lístku
   2. Systém bude uživateli nabízet výčet nejběžnějších voleb v položkách: MPZ, Barva SPZ, Druh vozidla, Tovární značka
3. Prohlížení lokálně uložených parkovacích lístků
4. Úprava lokálně uložených parkovacích lístků
5. Pořízení fotodokumentace přestupku
   1. Systém bude umožňovat při vyplňování parkovacího lístku vytvořit fotodokumentaci přestupku
   2. Fotodokumentaci bude možné pořídit pomocí před-instalované aplikace fotoaparátu v systému mobilního telefonu

**Nesplněné funkční požadavky:**

1. Vyplnění parkovacího lístku
   1. Systém bude volitelně a podle aktuálních možností mobilního zařízení před-vyplňovat údaje: adresa, SPZ
2. Nápověda při vyplňování čísel paragrafů zákonů nejběžnějších přestupků
   1. Uživatel si bude moci přidat vlastní přestupky
   2. Uživatel si bude moci přidat vlastní volby v položkách: MPZ, Barva SPZ, Druh vozidla, Tovární značka
3. Odeslání parkovacích lístků z mobilního zařízení na server
4. Tisk parkovacích lístků pomocí mobilní tiskárny
5. Pořízení fotodokumentace přestupku
   1. Fotodokumentaci bude možné přidat vybráním předem nafocených fotografií ze před-instalované aplikace galerie v systému mobilního telefonu
   2. Systém bude fotodokumentaci ukládat do externí paměti (na paměťovou kartu)
6. Rozpoznání státní poznávací značky z fotografie
7. Automatické vyhledání adresy pomocí GPS
8. Kontrola odcizení přenosné parkovací karty opravňující parkovat v modré zóně
9. Přístup k funkcím aplikace po zadání identifikačního čísla

# Kapitola

Iterace 2

## Úvod, cíle iterace

Text této kapitoly se zabývá veškerou prací udělanou během 2. iterace. Hlavním cílem je pokračování v implementování funkcionality stanovené v iteraci 1. Jde o funkční požadavky, které byly dokončeny jen z části, případně jejich vývoj zatím vůbec nezačal. Nově se od této iterace objevuje nový požadavek na funkčnost, a to:

1. Kontrola parkovaní zaplaceného přes SMS
   1. Vyplnění SPZ vozidla.
   2. Systém bude umožňovat online ověření vozidla pomocí SPZ, zda dané vozidlo zaplatilo parkování. Pokud ano, v jakém časovém intervalu má povolení parkovat

## Případy užití

S novým funkčním požadavkem vznikl také nový případ užití:

1. Kontrola parkovaní zaplaceného přes SMS
   * Vyplnění SPZ vozidla.
   * Systém bude umožňovat online ověření vozidla pomocí SPZ, zda dané vozidlo zaplatilo parkování. Pokud ano, v jakém časovém intervalu má povolení parkovat.

Došlo k doplnění scénářů pro případy užití UC.06 a nový UC.09:

### Kontrola odcizení parkovací karty

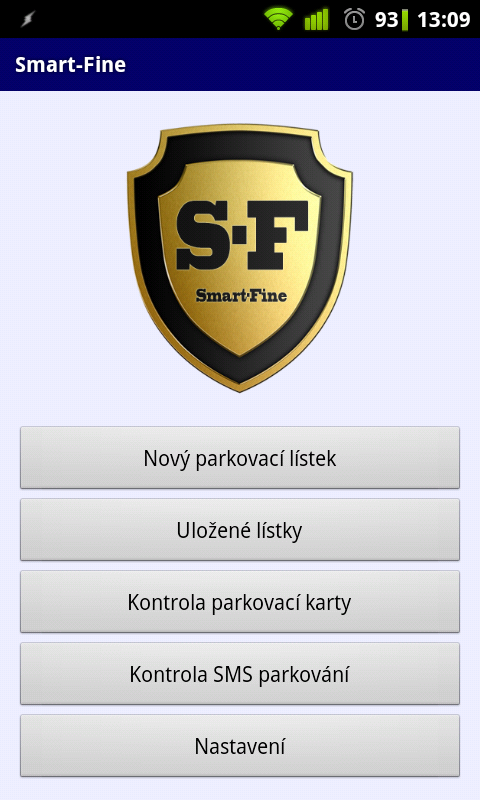
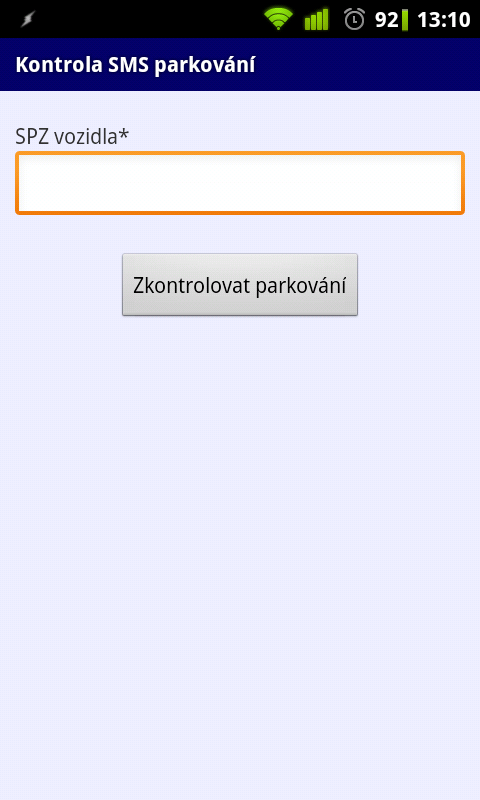
1. Uživatel chce ověřit, zda parkovací karta souhlasí s SPZ vozidla.
2. Systém zobrazí zadávací formulář na číslo parkovací karty.
3. Uživatel vyplní číslo parkovací karty.
4. Uživatel požádá systém o ověření čísla parkovací karty.
5. Systém ověří připojení k internetu.
   1. Pokud nalezne chybu, zobrazí chybovou hlášku.
   2. Pokračuje se bodem 4.
6. Systém získá informace ze serveru a zobrazí je.
   1. Pokud se získání informací nepodaří, systém zobrazí chybovou hlášku.
   2. Pokračuje se bodem 4.

### Kontrola parkovaní zaplaceného přes SMS

1. Uživatel chce ověřit, zda má vozidlo zaplacené parkování pomocí SMS.
2. Systém zobrazí zadávací formulář na SPZ vozidla.
3. Uživatel vyplní SPZ vozidla.
4. Uživatel požádá systém o parkování.
5. Systém ověří připojení k internetu.
   1. Pokud nalezne chybu, zobrazí chybovou hlášku.
   2. Pokračuje se bodem 4.
6. Systém získá informace ze serveru a zobrazí je.
   1. Pokud se získání informací nepodaří, systém zobrazí chybovou hlášku.
   2. Pokračuje se bodem 4.

## Uživatelské prostředí

Vzhledem k tomu, že je již část aplikace hotová, a vzhledem k jednoduchosti, nebyl udělán návrh uživatelského prostředí pro nové funkce v návrhovém editoru, jako tomu bylo v 1. iteraci, ale byl rovnou realizován přímo v aplikaci. Automatické určení polohy pomocí GPS se vyvolá tlačítkem při vytváření nového parkovacího lístku, dané tlačítko bylo navrženo a vytvořeno již v 1. iteraci. Naopak kontrola parkovací karty a kontrola parkování zaplaceného přes SMS vyžadovaly nové aktivity a tudíž i nové grafické rozhraní. Pro kontrolu se zadá číslo parkovací karty, případně SPZ do příslušných formulářů a stiskem tlačítka se provede kontrola. Výsledné informace o parkování, či parkovací kartě, se zobrazí v prázdném prostoru pod tlačítky. Z důvodu přibývající funkcionality se musela upravit také úvodní obrazovka s menu, aby nedošlo k znepřehlednění:

TODO zase nejakej popisek bla bla

## Implementace

Z 1. iterace zbyla rozdělaná práce implementace přidávání fotodokumentace při vytváření nového parkovacího lístku, případně jeho editace. Fotografie šly fotit přímo z aplikace za pomocí předem přeinstalované systémové aplikace na fotografování, která je standardní výbavou operačního systému Android. Nyní bude moci uživatel vybrat předem vyfocenou fotografii z galerie telefonu. Ta zůstane uložena na svém původním místě, k parkovacímu lístku se uloží pouze odkaz na ní. Při pořizování nových fotografií skrze aplikaci Smart-Fine se fotografie uloží na externí, nebo interní kartu, kde nebudou zabírat spoustu paměti, do složky „smartfine“. Po odeslání lístků na server se fotografie z této složky vymažou.

Dalším předmětem implementace byla funkce určení adresy pomocí GPS, při vytváření parkovacího lístku. To nastane po stisknutí tlačítka s kompasem u formuláře s adresou. Jelikož určení polohy není přímočará záležitost, je potřeba čekat na dobrý signál a GPS modulu chvíli trvá, než polohu nalezne, vznikl problém se „zamrznutím“ aplikace do doby, než byla poloha nalezena, nebo než vypršel timeout. To lze řešit pomocí nového vlákna, které běží na pozadí a hledá polohu, zatím co uživatel může dál bez problémů vyplňovat parkovací lístek. Pro získání polohy musí mít uživatel na mobilním telefonu zapnutý buď internetový, nebo GPS provider. V nejlepším případě oba. Vlákno má potom 30 vteřin na to, aby našlo polohu a vrátilo adresu. Pokud se tak nestane, uživatel bude muset znovu o polohu požádat stiskem onoho tlačítka s kompasem. Pro získání adresy z GPS souřadnic je potřeba připojení k internetu. Po úspěšném získání adresy se automaticky vyplní do formulářů.

Diagramy ukazující nové třídy v jednotlivých balíčcích jsou k dispozici v příloze Návrh [C]. Kompletní popis implementace je zpracován za pomocí JavaDoc [F].

## Závěr

Během 2. iterace bylo splněno několik dalších funkčních požadavků. Některé požadavky, které byly rozpracovány v 1. iteraci, byly doimplementovány, některé nové započaty. Opět se ovšem nepodařilo implementovat všechny. Seznam uvádí, co bylo a nebylo splněno, a co zůstalo do dalšího vývoje.

**Splněné funkční požadavky:**

1. Vyplnění parkovacího lístku
   1. Systém bude volitelně a podle aktuálních možností mobilního zařízení před-vyplňovat údaje: adresa
2. Pořízení fotodokumentace přestupku
3. Automatické vyhledání adresy pomocí GPS
4. Kontrola odcizení přenosné parkovací karty opravňující parkovat v modré zóně
   1. Vyplnění čísla karty.
5. Kontrola parkovaní zaplaceného přes SMS
   1. Vyplnění SPZ vozidla.

**Nesplněné funkční požadavky:**

1. Vyplnění parkovacího lístku
   1. Systém bude volitelně a podle aktuálních možností mobilního zařízení před-vyplňovat údaje: SPZ
2. Nápověda při vyplňování čísel paragrafů zákonů nejběžnějších přestupků
   1. Uživatel si bude moci přidat vlastní přestupky
   2. Uživatel si bude moci přidat vlastní volby v položkách: MPZ, Barva SPZ, Druh vozidla, Tovární značka
3. Odeslání parkovacích lístků z mobilního zařízení na server
4. Tisk parkovacích lístků pomocí mobilní tiskárny
5. Rozpoznání státní poznávací značky z fotografie
6. Kontrola odcizení přenosné parkovací karty opravňující parkovat v modré zóně
   1. Systém bude umožňovat online porovnání čísla zadané parkovací karty se seznamem odcizených přenosných parkovacích karet.
7. Přístup k funkcím aplikace po zadání identifikačního čísla
8. Kontrola parkovaní zaplaceného přes SMS
   1. Systém bude umožňovat online ověření vozidla pomocí SPZ, zda dané vozidlo zaplatilo parkování. Pokud ano, v jakém časovém intervalu má povolení parkovat

Seznam příloh

TODO: Tady bude jenom seznam priloh.. pos, analyza atd se bude přidávat na konec bakalářky jako prilohy (takze nikoliv externi, ale interni) Potom v priloze A bude napsano, ze tato tistena podoba je finální a ze predesle verze jde najit externe tady a tady a bla bla

**[A]** Přehled o projektu (Project Overview Statement) - Dostupné z:  
Iterace 1 **-** <[Iterace1\Project overview statement.docx](Iterace1/Project%20overview%20statement.docx)>  
Iterace 2 - TODO  
Iterace 3 - TODO

**[B]** Analýza - Dostupné z:  
Iterace 1 - <[Iterace1\Analyza.docx](Iterace1/Analyza.docx)>  
Iterace 2 - TODO  
Iterace 3 - TODO

**[C]** Návrh - Dostupné z:  
Iterace 1 - <[Iterace1\Navrh.docx](Iterace1/Navrh.docx)>  
Iterace 2 - TODO  
Iterace 3 - TODO

**[D]** Testování - Dostupné z:  
Iterace 1 - <[Iterace1\Testovani.docx](Iterace1/Testovani.docx)>  
Iterace 2 - TODO  
Iterace 3 - TODO

**[E]** PDF Prototyp aplikace – Dostupné z <[Iterace1\Klikaci protoyp.pdf](Iterace1/Klikaci%20protoyp.pdf)>

**[F]** JavaDoc, dokumentace implementace - Dostupné z:  
Iterace 1 - <[Iterace1\doc\index.html](Iterace1/doc/index.html)>  
Iterace 2 - TODO  
Iterace 3 - TODO

**[G]** Testování tester 1 - Dostupné z <[Iterace1\user\_test\_1.mp4](Iterace1/user_test_1.mp4)>

**[H]** Testování tester 2 - Dostupné z <[Iterace1\user\_test\_2.mp4](Iterace1/user_test_2.mp4)>

**[I]** Návrh základního uživatelského rozhraní - Dostupné z <[Iterace1\NavrhGui.pdf](Iterace1/NavrhGui.png)>

Příloha

Seznam literatury a webových odkazů

**[1]** BROŽ PAVEL, Bakalářská práce - Systém pro evidenci přestupků pomocí mobilních telefonů - serverová část, 2012 TODO: odkaz

**[2]** web: A7B36SI2 – Řízení softwarových projektů - Dostupné z: <<https://edux.feld.cvut.cz/courses/A7B36SI2/start>>

**[3]** web: Balsamiq Mockup – Návrh uživatelského prostředí - Dostupné z: <<http://www.balsamiq.com/products/mockups>>

**[4]** web: Enterprise Architect – Modelování v UML – Dostupné z: <<http://www.sparxsystems.com.au/>>

**[5]** web: Android SDK – Komponenty pro vývoj pro Android OS – Dostupné z: <<http://developer.android.com/sdk/index.html>>

**[6]** web: IDE Eclipse – Vývojový nástroj programování - Dostupné z: <<http://www.eclipse.org/>>

Příloha

Seznam použitých zkratek

**SI2** Předmět Řízení softwarových projektů - A7B36SI2

**IDE** Integrated Development Environment

**POS** Project Overview Statement

**PL** Parkovací lístek

**UML** Unified modeling language , Unifikovaný modelovací jazyk

**DAO** Data Access Object