

# Slovenská technická univerzita

FAKULTA INFORMATIKY A INFORMAČNÝCH TECHNOLOGIÍ

Študijné odbory : Informačné systémy a Softvérové inžinierstvo

---

## Mobilný cestovný poriadok pre iPhone

Dokumentácia k inžinierskemu dielu

---

### **Tím 14 : iTEAM**

Martin Jačala

Marek Brandobúr

Michal Macko

Michal Hrdina

Martin Blažko

Hana Časnochová

### **Vedúci tímu :**

Ing. Michal Čerňanský, PhD.

**[tp14@googlegroups.com](mailto:tp14@googlegroups.com)**

3.11.2009

# OBSAH

<b>ÚVOD .....</b>	<b>1</b>
ÚČEL DOKUMENTU.....	1
SCRUM .....	1
ŠTRUKTÚRA DOKUMENTU .....	1
ZADANIE .....	1
SLOVNÍK POJMOV A SKRATIEK.....	2
<b>ŠPRINTY.....</b>	<b>3</b>
<b>1.ŠPRINT.....</b>	<b>3</b>
HELLO WORLD APLIKÁCIA .....	3
PROPAGÁCIA PRODUKTU.....	4
<b>2. ŠPRINT .....</b>	<b>5</b>
FUNKCIONALITA „NAJBLIŽŠIE SPOJE“ .....	5
PROPAGÁCIA PRODUKTU, DRUHÁ ČASŤ .....	10
<b>PRÍLOHA A : TESTOVACIE SCENÁRE.....</b>	<b>A-1</b>
<b>ROLY .....</b>	<b>A-1</b>
<b>POPIS PROSTREDIA.....</b>	<b>A-1</b>
<b>POPIS SLUŽBY.....</b>	<b>A-1</b>
<b>TESTOVACIE SCENÁRE .....</b>	<b>A-3</b>
TS01: SLUŽBA GETSERVICES – ÚSPEŠNÉ PREVZATIE SPRÁVY.....	A-3
TS02: SLUŽBA GETSERVICES – CHÝBAJÚCE PARAMETRE.....	A-6
TS03: SLUŽBA GETSERVICES – CHYBNÉ PARAMETRE .....	A-7
TS04: SLUŽBA GETSERVICES – CHYBA SPRACOVANIA .....	A-8
TS05 – KLIENTSKÁ ČASŤ – POUŽÍVATEĽSKÉ ROZHRANIE .....	A-8

TS06 – KLIENTSKÁ ČASŤ – ROZHRANIE KLIENT/SERVER.....	A-9
TS07 – KOMUNIKÁCIA KLIENT-SERVER .....	A-9
VÝSLEDKY TESTOV.....	A-10
<b><u>PRÍLOHA B : DIZAJN.....</u></b>	<b><u>B-1</u></b>

# ÚVOD

## Účel dokumentu

Tento dokument bol vytvorený v rámci predmetu Tvorba softvérového systému v tíme v akademickom roku 2009/2010. Slúži ako projektová dokumentácia pre riešený projekt a jeho cieľom je priblížiť priebeh riešenia daného projektu a jeho jednotlivé etapy.

## Scrum

Hneď v úvode je nutné podotknúť, že na vypracovanie projektu sme si zvolili agilnú metódu softvérového vývoja SCRUM. Agilný vývoj používa inkrementálny spôsob vývoja, časté testovanie a zároveň núti zákazníka byť zainteresovaným do vývoja. Jeho najväčšou výhodou je rýchla adaptácia na zmeny. Metóda Scrum pozostáva zo šprintov, v rámci ktorých sa vytvára inkrementálne komplexná časť softvéru. Výsledkom tohto čiastkového procesu je už otestovaná časť softvéru. Na tomto základe je aj ďalej členený dokument.

## Štruktúra dokumentu

Dokument je rozdelený na niekoľko častí. V nasledujúcej kapitole sa nachádza opis jednotlivých šprintov. V prílohe A je uvedené testovanie funkcionality, ktorá vznikla v druhom šprinte. V prílohe B sa nachádzajú návrhy dizajnu.

## Zadanie

Mnohí účastníci mestskej autobusovej prepravy vysoko oceňujú portál "imhd.sk" prevádzkovaný občianskym združením "mhd.sk". Portál poskytuje informácie týkajúce sa mestskej hromadnej dopravy pre viaceré slovenské mestá. Medzi jeho najvýznamnejšie služby patrí možnosť prehliadania a tlače cestovných poriadkov jednotlivých liniek a možnosť plánovania cesty mestskou hromadnou dopravou. Zaujímavým doplnením existujúcich služieb portálu by mohlo byť poskytovanie služieb pre mobilné zariadenia.

V poslednej dobe silnie ponuka aj popularita mobilných zariadení schopných vykonávať používateľom inštalované aplikácie. Možnosti mobilných zariadení sú čoraz širšie a tiež výrazne stúpa "príjemnosť" používania aplikácií (angl. User Experience). Tiež mobilní operátori poskytujú služby umožňujúce cenovo dostupné pripojenie na Internet prostredníctvom mobilných zariadení. Významným z hľadiska rozšírenia ale aj možností je zariadenie iPhone (Apple) s operačným systémom OS X pre iPhone disponujúce dotykovým displejom, GPS, akcelerometrom, a vo verzii 3GS aj kompasom. Na trhu je tiež stále väčšie množstvo zariadení s OS Microsoft Windows Mobile (napr. HTC Touch Pro 2) alebo s OS Symbian (napr. Nokia N97). Novým operačným systémom je Android OS vyvinutý spoločnosťou Google (napr. HTC Hero).

Navrhňte, implementujte a otestujte riešenie umožňujúce používateľom prístup k informáciám mestskej hromadnej dopravy prostredníctvom mobilných zariadení. Zamerajte

sa na "príjemnosť" použitia služby a možnosť využitia vlastností mobilného telefónu iPhone. Analyzujte vhodnosť a možnosti rozšírenia služby aj na linky medzimestskej dopravy či železničnej dopravy.

Zvážte potrebu sa podrobnejšie venovať nasledujúcim bodom:

- Overte možnosti použitia mobilného zariadenia ako polohovacieho zariadenia, okrem zemepisných súradníc z GPS overte možnosti použitia kompasu a tiež akcelerometra, overte presnosť poskytovaných údajov a tiež možnosť ich spresnenia filtrovaním.
- Navrhňte aplikáciu pre iPhone využívajúcu polohovanie na spresnenie oblasti záujmu používateľa.
- Navrhňte spôsob ukladania informácií do vyrovnávacej pamäte aplikácie automaticky na základe preferencií používateľa (modelovanie potrieb používateľa), riešte možnosť použitia služby aj bez pripojenia na Internet.
- Navrhňte spôsob analyzovania kvality modelovania potrieb používateľa, kvality navrhnutého používateľského rozhrania, spôsob analýzy dopytov realizovaných používateľom a pod.
- Venujte sa bezpečnosti prípadných citlivých údajov a špeciálne sa zamerajte na otázku možnej straty súkromia používateľov (sledovanie ich polohy, identifikácia používateľa na základe korelácie medzi zaznamenanými údajmi a pod.).

## **Slovník pojmov a skratiek**

**Scrum** – agilná metóda softvérového vývoja

**GPS** - satelitný navigačný systém používaný na zistenie presnej pozície

**Mac OS X** - operačný systém pre počítače Macintosh

**iPhone SDK** - softvérový balík na vývoj aplikácií pre iPhone a iPod touch

**XCode** – vývojové prostredie pre Mac OS X

**Users story** – požiadavka na softvérový systém formulovaná pár vetami v každodennom jazyku používateľa

**MHD** – mestská hromadná doprava

**REST** – z anglického Representational State Transfer

**Mock** – objekt, ktorý kontrolovane napodobňuje správanie reálneho objektu

# ŠPRINTY

## 1. šprint

Prvý šprint trval od 6.10.2009 do 20.10.2009. V rámci tohto šprintu sme zrealizovali dve Users stories. Zaoberali sme sa vytváraním rannej verzie aplikácie a snažili sa o oboznámenie používateľov so vznikajúcou aplikáciou.

### Hello world aplikácia

Používateľ spustí aplikáciu na mobilnom zariadení. Aplikácia sa bez problémov spustí a zobrazí sa okno, ktoré bude mať tri časti. V spodnej časti budú tlačidlá, v strednej obsah a v hornej kontextovo závislé možnosti. Zároveň prevezme súbor zo servera.

### *Analýza*

Už existuje viacero služieb, ktoré poskytujú jednoduchý prístup k cestovným poriadkom MHD. Na portáli imhd.sk je možné vyhľadávať spojenia medzi zastávkami, tak isto ako prezerat', či prípadne si vytlačiť cestovný poriadok podľa čísla linky. Je tu aj statická mapa s vyznačenými zastávkami. Portál však nie je veľmi prispôsobený takým ľuďom, ako napríklad turistom, ktorí nevedia názov zastávky, nevedia kde sa nachádza a ako vyzerá, prípadne vôbec netušia kde práve sú.

Na rad prichádza flexibilita mobilného zariadenia, ktoré je možné mať stále poruke. Myslíme si, že aplikácia podporovaná mobilným zariadením prináša zo sebou mnoho výhod oproti webovému portálu. Netreba si plánovať cestu vopred, je možné zistiť spojenie priamo v teréne.

Po zvážení sme sa rozhodli vyvíjať aplikáciu pre mobilné zariadenie iPhone od firmy Apple s operačným systémom OS X. Využijeme prostredie XCode a platformou iPhone SDK 3.0 ktoré, okrem iného, využíva pre nás podstatné GPS. Na vytvorenie grafického užívateľského prostredia použijeme nástroj Interface Builder, ktorý je súčasťou balíka určeného pre vývojárov iPhone. Distribúcia, ktorá je zabezpečená pomocou kanála App Store, umožňuje každému používateľovi prístup k aplikácii. A voľne šíriteľné aplikácie sú na App Store umiestnené bezplatne.

### *Návrh*

Hlavným cieľom je získanie zručností pri používaní nástrojov, v ktorých budeme ďalej implementovať aplikáciu. Architektúra klient-server.

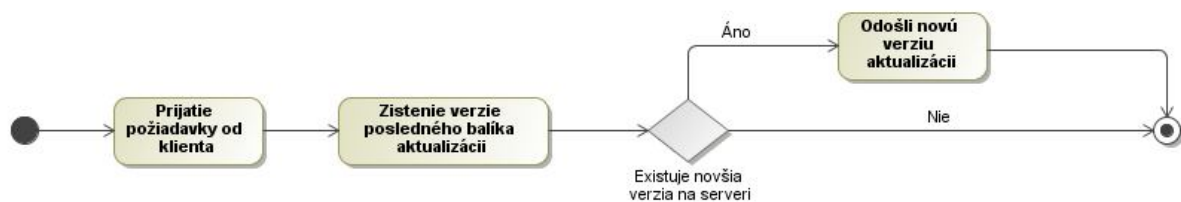
Samotná aplikácia po spustení bude obsahovať v spodnej časti obrazovky jednoduché tlačidlo, pričom po jeho stlačení sa vyšle požiadavka na sever a prevezme sa odpoveď - reťazec zo servera, ktorý sa vypíše v strednej časti obrazovky.

## **Implementácia**

Implementáciu sme rozdelili do dvoch častí, na klientskú a serverovú časť.

Klientskú časť sme implementovali v jazyku ObjectiveC v prostredí XCode, Aplikácia bola vytvorená pomocou šablóny „View based application“, ktorá je súčasťou vývojového prostredia. Používateľské grafické rozhranie obsahovalo dva prvky – tlačidlo a textové pole.

Serverová časť je implementovaná v jazyku php. Postupnosť krokov, ktoré vykonáva server, sa nachádza na Obr.1.



**Obr. 1.** Postupnosť krokov na serveri.

## **Testovanie**

Keďže ide o veľmi jednoduchú aplikáciu, test pripojenia na server spočíva v samotnom zobrazení verzie prevzatého súboru.

## **Propagácia produktu**

Zlepšiť informovanosť záujemcov o produkt.

## **Analýza**

Na to, aby sme získali čo najviac používateľov našej aplikácie, je nutné nájsť spôsob oslovenia tých potencionálnych. Dobré je osloviť obyvateľov Bratislavy. Zistiť zhruba koľko ľudí reálne používa iPhone, kvôli ďalšej funkcionalite a prípadnému zapojeniu používateľov. Vhodným spôsobom na interakciu je web. Webová stránka opisujúca produkt musí mať priliehavý a ľahko zapamätateľnú doménu.

## **Návrh**

Stránka produktu bude v slovenskej aj anglickej verzii. Jej dizajn musí byť prívetivý a prehľadný. Zároveň bude optimalizovaná pre vyhľadávače. V úvodnej fáze sa na nej budú nachádzať len informatívne údaje o plánovanej funkcionalite, neskôr prídu ďalšie podrobnosti a možnosť prevzatia samotnej aplikácie.

Za názov sme zvolili iTransport.

Anketa bude vytvorená tak isto v slovenskej aj anglickej verzii – chceme osloviť aj zahraničných turistov. Z ankety zistíme, ako rozšírené sú mobilné zariadenia, pre ktoré implementujeme aplikáciu, a akú ďalšiu funkčnosť, by budúci používatelia privítali. Anketa bude pozostávať z niekoľkých jednoduchých otázok.

### ***Implementácia***

Dizajn stránky sa nachádza v prílohe . Na jej implementáciu sme použili html a css.

Anketa, implementovaná v php, pozostáva z týchto otázok:

- Aký operačný systém používate v MT?
- Koľko ľudí používajúcich iPhone poznáte?
- Aká funkčnosť by sa vám páčila?

### ***Testovanie***

Aplikácia je určená pre širokú verejnosť, preto je potrebné zaručiť jej správne zobrazovanie v najčastejšie používaných webových prehliadačoch. Stránku sme otestovali pomocou nástroja Browser Shots – vybrali sme prehliadače Opera, Mozilla Firefox, Safari, Internet Explorer 7. Odhalené nedostatky pri zobrazovaní sme opravili, najviac opráv bolo nutných pre prehliadač Internet Explorer 7.

## **2. šprint**

Druhý šprint sme začali 20.11.2008 a ukončili 3.11.2008. Cieľom bolo dopĺňanie funkčnosti do aplikácie a ďalšia propagácia. Celkovo sme opäť identifikovali dve users stories.

### **Funkčnosť „najbližšie spoje“**

Používateľ chce informácie o odchodoch autobusov z najbližších zastávok v najbližšom čase. Klikne preto na prvé tlačidlo v spodnej (tlačidlovej) časti a v strednej (obsahovej) časti sa objaví zoznam s informáciami o spojoch. Ku každému spoju zobrazí:

- zastávku, z ktorej spoj ide
- číslo spoja (veľké a výrazné)
- smer spoja
- čas kedy ide
- čas, ktorý zostáva do odchodu
- 

### ***Analýza***

Na začiatok je nutné vytvoriť databázovú štruktúru, ktorá bude obsahovať zastávky, časy odchádzajúcich spojov a aké linky z akej zastávky odchádzajú. Pre každú zastávku musíme vedieť jej zemepisnú šírku a dĺžku. Tak isto je potrebné rozlišovať medzi zastávkami, ktoré



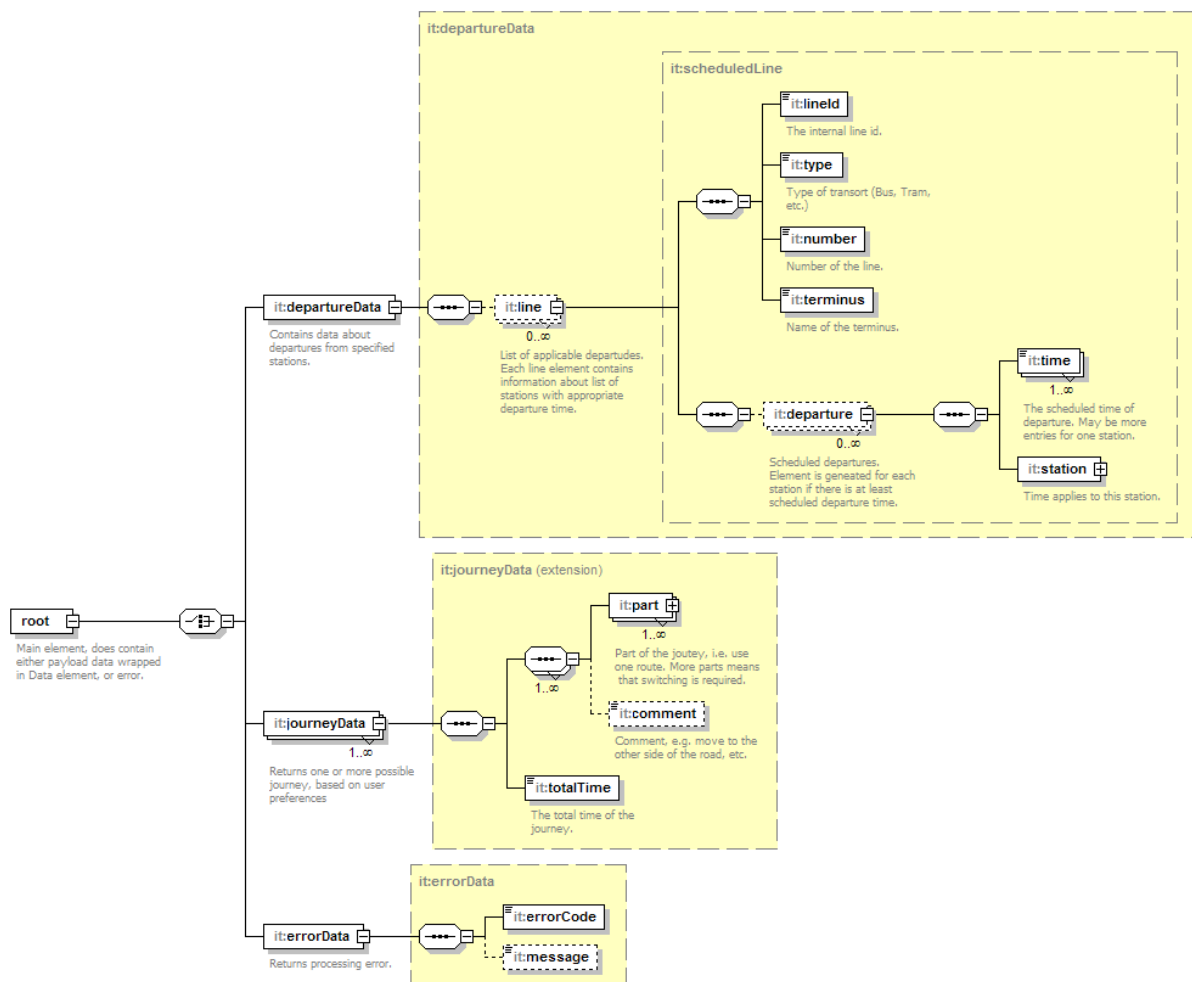
majú rovnaký názov, ale rôznu polohu, ako napríklad zastávky Račianske mýto. Zohľadniť treba aj smer linky, nielen jej číslo. Databázová štruktúra musí byť tiež ľahko rozšíriteľná o ďalšie typy údajov, ktoré budú potrebné v nasledujúcom vývoji aplikácie.

Zobrazovanie požadovaných informácií na obrazovke mobilného zariadenia musí byť jasné a prehľadné. Požadované informácie musia byť zobrazené ihneď po zapnutí aplikácie.

Pri návrhu protokolu komunikácie klient-server musíme brať do úvahy požiadavky na použiteľnosť a škálovateľnosť riešenia.

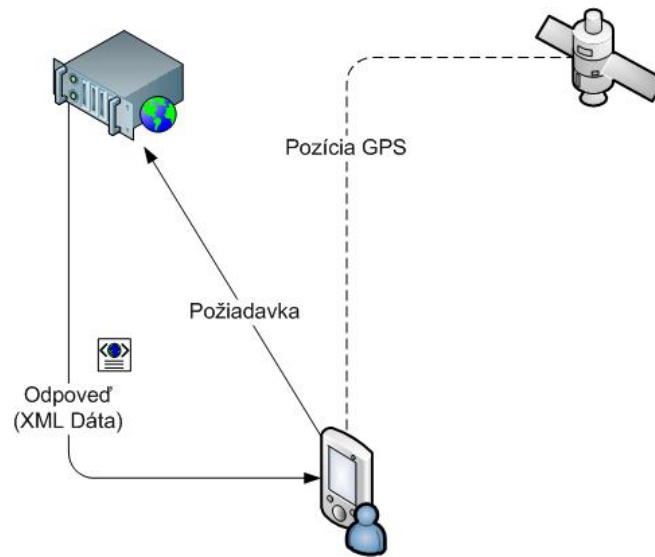
## Návrh

Prvý krok pri návrhu je vytvorenie sformalizovania komunikačného protokolu medzi klientom a serverom. S ohľadom na požiadavky definované v analýze, sme sa rozhodli využiť webové služby založené na štandarde REST, teda posielanie správ pomocou XML. Formát dát je špecifikovaný pomocou schémy XSD, ktorá zároveň slúži ako referencia pri implementovaní ostatných častí (Obr. 2).



Obr. 2. XSD schéma.

Klientská časť pozostáva s aplikácie, ktorá komunikuje so serverom pomocou protokolu HTTP. Po zavolaní webovej služby a predaní parametrov dostáva odpoveď vo formáte XML (**Obr. 3**).



**Obr. 3.** Model komunikácie klient – server.

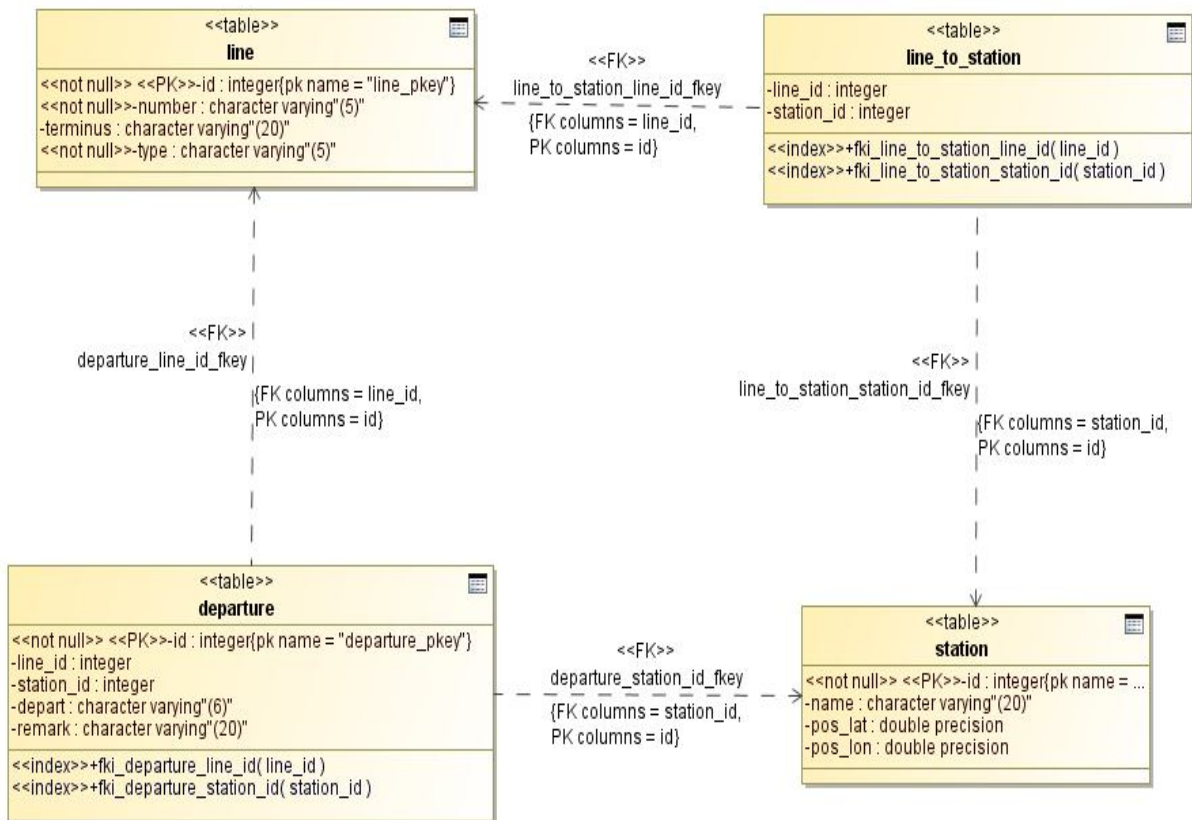
Pre samotnú funkcionálnosť bude vytvorená webová služba, ktorá poskytuje informácie o odchodoch spojov z lokality v špecifikovanom čase. Pre zadanú vstupnú polohu nájde všetky zastávky v okolí definovanej veľkosti. Pre každú zastávku vyhledá spoje obsluhujúce danú zastávku, ku každej dvojici spoj - zastávka bude nájdený najbližší odchod.

Použitie GPS v simulátore iPhone má obmedzené možnosti, pretože simulátor udáva vždy iba jednu GPS pozíciu v USA. Pozíciu teda bude nutné simulovať pomocou dočasného objektu (Mock). Počas prvých fáz vývoja klienta bude pozícia simulovaná v blízkosti školy.

### ***Implementácia***

Serverovú časť sme sa rozhodli implementovať v skriptovacom jazyku Python, kvôli lepšej výkonnosti, nakoľko na server budú kladené vysoké nároky, súvisiace s plánovaním cesty. Za databázový systém sme si zvolili voľne šíriteľný objektovo-relačný PostgreSQL 8.4.

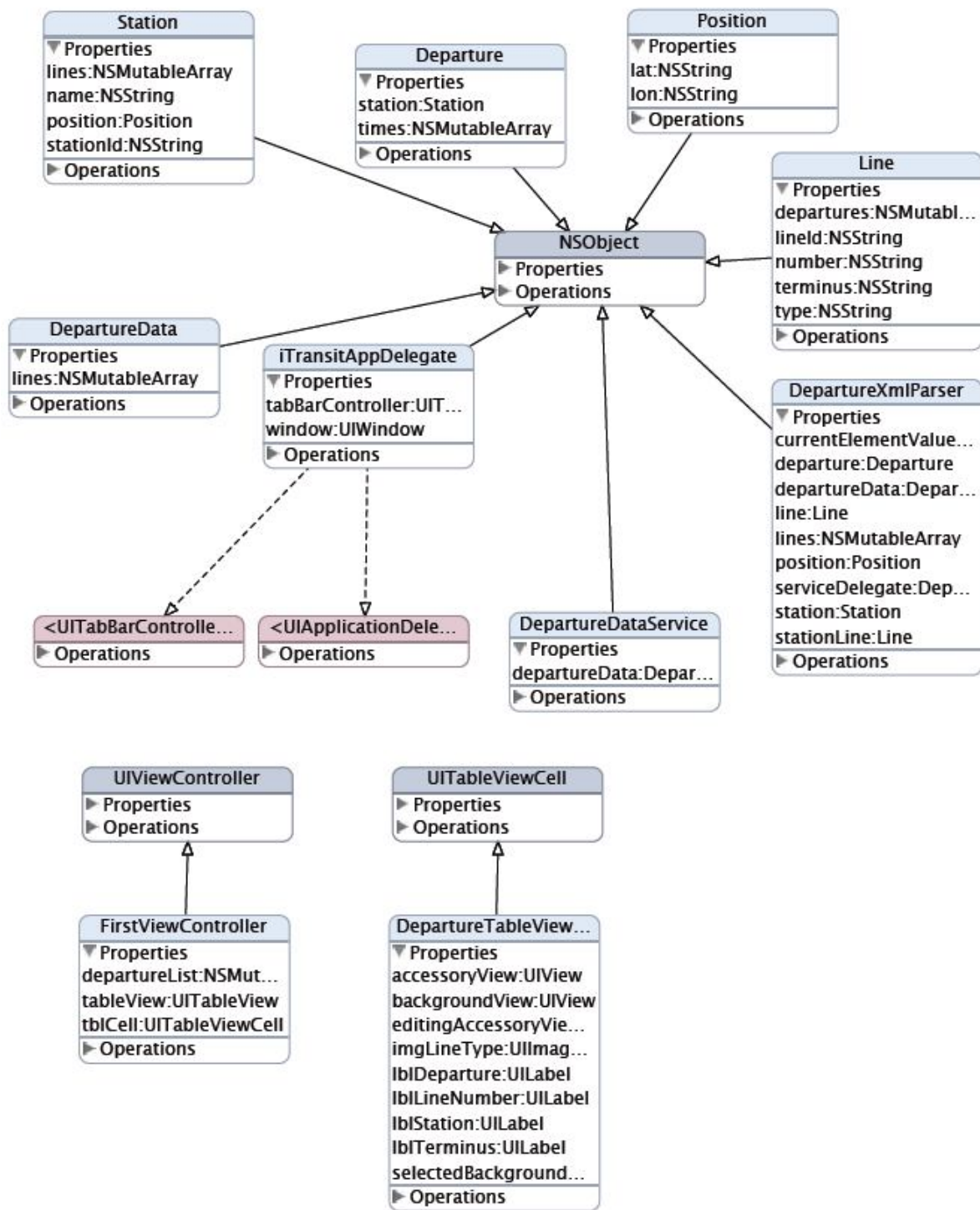
Implementácia logického modelu, špecifikovaného ako XSD schéma, je na strane databázových objektov vyjadrená pomocou fyzického dátového modelu (**Obr. 4**).



Obr. 4. Fyzický dátový model.

Klientská časť je implementovaná na základe návrhového vzoru „model-view-controller“.

Triedy Line, Position, Departure, Station a DepartureData predstavujú vrstvu „model“ v aplikácii. Sú implementované podľa špecifikácie služby definovanej v XSD schéme. Tieto objekty sú inšancované v triede DepartureXmlParser. Vrstva „controller“ manipuluje s dátami na základe požiadaviek GUI. Trieda DepartureDataService prijíma požiadavky z objektov „controller“ a slúži ako obaľovač (wrapper) na REST služby bežiace na aplikačnom serveri. Diagram tried sa nachádza na Obr. 5.



Obr. 5. Diagram tried.

Nami implementované používateľské rozhranie spĺňa požiadavky users story (Obr. 6.).



Obr. 6. Výstupná obrazovka aplikácie.

### ***Testovanie***

Testovacie scenáre, slúžiace ako integračné testy pri testovaní danej users story, sú uvedené v prílohe A.

### **Propagácia produktu, druhá časť**

Ešte viac zlepšiť informovanosť záujemcov o produkt.

### ***Analýza***

Súčasný dizajn stránky propagujúcej produkt je nevyhovujúci, preto je nutná jeho zmena. Keďže vyvíjame aplikáciu pre iPhone, aj stránka by sa mala podobať na stránky produktov určených pre iPhone.

Doména iTransport.sk nie je voľná, preto je nevyhnutné prehodnotiť názov.

Ďalším spôsobom zvýšenia záujmu o produkt je prezentácia stránky produktu pomocou reklamy. Forma prezentácie, ktorá podľa nás najviac osloví želanú skupinu potencionálnych používateľov, je reklamný prúžok (banner), zverejnený na už existujúcej a pomerne navštevovanej webovej stránke. Ďalšou formou je plagát, ktorý by tiež upozorňoval na existenciu stránky.

## ***Návrh***

Nový dizajn stránky sa nachádza v prílohe B.

Zmena ankety spočíva v rozšírení otázok a taktiež v ich postupnom zobrazovaní užívateľovi.

Názov sme zmenili na iTransit, pričom doména iTransit.sk je prístupná.

Plagátik na podporu návštevnosti webovej stránky produktu, v prílohe B, sa vyvesí na frekventované miesta v Bratislave..

## ***Implementácia***

Ukážka reklamného prúžku sa nachádza v prílohe B. Bol vytvorený technológiou Flash. Stránka, ktorá nám umožnila jeho zverejnenie je fei.sk.

## PRÍLOHA A : TESTOVACIE SCENÁRE

### Roly

Analytik:

- Vytvorí testovacie scenáre
- Navrhne spôsob testovania

Tester:

- Vykoná testy
- Zapíše výsledky testov

Vývojár:

- Zúčastňuje sa testovania
- Odstráni nájdené nedostatky

Zákazník:

- Akceptuje výsledky testov

### Popis prostredia

Testované riešenie sa konceptuálne skladá z dvoch častí. Prvou časťou je aplikácia nasadená na aplikačnom serveri. Funkcionalita, na ktorú sa vzťahuje tento dokument je zahrnutá v jednej REST službe na preddefinovanom endpointe. Táto služba poskytuje údaje druhej časti implementovaného riešenia. Klientská časť pozostáva s aplikácie, ktorá komunikuje so serverom pomocou protokolu HTTP. Po zavolaní služby a predaní parametrov dostáva odpoveď vo formáte XML. Dáta sú potom prezentované v grafickej forme používateľovi.

### Popis služby

Služba getDepartures poskytuje informácie o odchodoch spojov z lokality v špecifikovanom čase. Pre zadanú vstupnú polohu nájde všetky zastávky v definovanom okolí. Pre každú zastávku vyhledá spoje obsluhujúce danú zastávku, ku každej dvojici spoj - zastávka bude nájdený najbližší odchod.

#### *Input*

**Endpoint URI:**

<http://tp.ktokoho.info/api/getDepartures.api>

**Vstupné parametre:**

- lat - GPS súradnica - zem. šírka – povinný
- lon - GPS súradnica - zem. dĺžka – povinný
- time - čas klienta v tvare hh:mm - povinný
- offset - v minútach, časový limit pre odchod spojení (time+offset < každý z odchodov), v prípade nevyplnenia automaticky nastavený na 60 – nepovinný
- radius - V metroch, max. vzdialenosť zastávky od používateľa, aby bola zahrnutá do výsledkov, v prípade nevyplnenia automaticky nastavená na 500 - nepovinný

## **Output**

### **Example request:**

<http://tp.ktokoho.info/api/getDepartures.api?lat=48.154&lon=17.0754&time=16:50&offset=60&radius=500>

Výstupom je správa v nasledovnom formáte (v prípade, že pri spracovaní nenastala chyba):

```
<?xml version="1.0"?> <p1:root
xmlns:p1="http://www.itransit.sk/xsd/datamodel"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.itransit.sk/xsd/datamodel/datamodel.xsd">
<p1:departureData>
<p1:line>
<p1:lineId>string</p1:lineId>
<p1:type>NIGHTBUS</p1:type>
<p1:number>string</p1:number>
<p1:terminus>string</p1:terminus>
<p1:departure>
<p1:time>string</p1:time>
<p1:station>
<p1:stationId>string</p1:stationId>
<p1:name>string</p1:name>
<p1:position>
<p1:lat>-1E4</p1:lat>
<p1:lon>-1E4</p1:lon>
</p1:position>
</p1:station>
</p1:departure>
</p1:line>
</p1:departureData>
</p1:root>
```

## **Error ¶**

Code: 1 Message: Some parameter value is missing.

Code: 2 Message: Parameter lon has bad type. Maybe, it is not float number.

Code: 3 Message: Parameter lat has bad type. Maybe, it is not float number.

Code: 4 Message: Parameter time is bad.



Code: 5 Message: Parameter offset is bad.  
Code: 6 Message: Parameter radius is bad.  
Code: 7 Message: Cannot connect to DB.

## Testovacie scenáre

Táto časť dokumentu obsahuje jednotlivé testovacie scenáre, ktoré je potrebné vykonať. Každý scenár obsahuje postupnosť krokov, vstupy a očakávané výstupy.

### TS01: Služba `getServices` – úspešné prevzatie správy

Služba poskytuje dáta klientskej aplikácii. Je preto potrebné otestovať správanie služby a výstupné dáta bez pripojeného klienta.

#### Vstupné požiadavky:

Ako vstup slúži URI služby spolu s testovacími parametrami. Na serveri musí bežať testovaná služba. Databáza pre službu musí bežať a byť správne nakonfigurovaná. Databáza musí obsahovať testovacie dáta (súčasť dokumentu ako SQL skript, príloha 1). Tester musí mať prístup pre HTTP protokol na testovací server.

#### Testovacia URI:

<http://tp.ktokoho.info/api-dev/getDepartures.api?lat=48.154&lon=17.0754&time=16:50&offset=60&radius=500>

#### Očakávaný výstup:

XML správa obsahujúce požadované hodnoty. Správa je zároveň validný dokument podľa XSD schémy (príloha 2).

#### Očakávaná XML správa:

```
<?xml version="1.0" encoding="UTF-8"?>
<root xmlns="http://www.itransit.sk/xsd/datamodel"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.itransit.sk/xsd/datamodel/datamodel.xsd">
  <departureData>
    <line>
      <lineId>3</lineId>
      <type>BUS</type>
      <number>32</number>
      <terminus>Hlavna</terminus>
      <departure>
        <time>16:55</time>
        <time>17:07</time>
        <time>17:19</time>
        <time>17:31</time>
      </departure>
    </line>
  </departureData>
</root>
```

```

        <time>17:43</time>
        <station>
            <stationId>2</stationId>
            <name>ZOO</name>
            <position>
                <lat>48.154000</lat>
                <lon>17.075400</lon>
            </position>
        </station>
    </departure>
</line>
<line>
    <lineId>1</lineId>
    <type>BUS</type>
    <number>39</number>
    <terminus>Suhvezdna</terminus>
    <departure>
        <time>17:01</time>
        <time>17:21</time>
        <time>17:41</time>
        <station>
            <stationId>27</stationId>
            <name>Televizia</name>
            <position>
                <lat>48.156700</lat>
                <lon>17.071000</lon>
            </position>
        </station>
    </departure>
    <departure>
        <time>17:02</time>
        <time>17:22</time>
        <time>17:42</time>
        <time>17:02</time>
        <time>17:22</time>
        <time>17:42</time>
        <station>
            <stationId>2</stationId>
            <name>ZOO</name>
            <position>
                <lat>48.154000</lat>
                <lon>17.075400</lon>
            </position>
        </station>
    </departure>
</line>
<line>
    <lineId>2</lineId>
    <type>BUS</type>
    <number>31</number>

```

```

<terminus>TrnavskeMyto</terminus>
<departure>
  <time>16:56</time>
  <time>17:08</time>
  <time>17:16</time>
  <time>17:28</time>
  <time>17:36</time>
  <time>17:48</time>
  <station>
    <stationId>27</stationId>
    <name>Televizia</name>
    <position>
      <lat>48.156700</lat>
      <lon>17.071000</lon>
    </position>
  </station>
</departure>
<departure>
  <time>16:57</time>
  <time>17:09</time>
  <time>17:17</time>
  <time>17:29</time>
  <time>17:37</time>
  <time>17:49</time>
  <time>16:57</time>
  <time>17:09</time>
  <time>17:17</time>
  <time>17:29</time>
  <time>17:37</time>
  <time>17:49</time>
  <station>
    <stationId>2</stationId>
    <name>ZOO</name>
    <position>
      <lat>48.154000</lat>
      <lon>17.075400</lon>
    </position>
  </station>
</departure>
</line>
</departureData>
</root>

```

### Priebeh testu:

- Spustenie internetového prehliadača
- Vloženie URI –
- <http://tp.ktokoho.info/api-dev/getDepartures.api?lat=48.154&lon=17.0754&time=16:50&offset=60&radius=500>

- Prijatie výslednej požiadavky – xml správy
- Validácia oproti XSD schéme
- Porovnanie navrátených údajov so vzorovou správou uvedenou v tomto dokumente

## TS02: Služba `getServices` – chýbajúce parametre

Účel scenára je otestovať správanie služby v prípade, že nie sú zadané všetky parametre

Vstupné požiadavky:

Ako vstup slúži URI služby spolu s testovacími parametrami. Jeden z povinných parametrov chýba. Na serveri musí bežať testovaná služba. Databáza pre službu musí bežať a byť správne nakonfigurovaná. Databáza musí obsahovať testovacie dáta (súčasť dokumentu ako SQL skript, príloha 1). Tester musí mať prístup pre HTTP protokol na testovací server.

### Testovacia URI:

<http://tp.ktokoho.info/api-dev/getDepartures.api>

Očakávaný výstup:

Služba vygeneruje chybovú správu s chybovým kódom 1. Podľa tabuľky chybových kódov (príloha 3) je chyba označená ako `Some parameter value is missing`.

### Očakávaná XML správa:

```
<?xml version="1.0" encoding="UTF-8"?>
<p1:root xmlns:p1="http://www.itransit.sk/xsd/datamodel"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.itransit.sk/xsd/datamodel/datamodel.xsd">
  <p1:errorData>
    <p1:errorCode>1</p1:errorCode>
    <p1:message>Some parameter value is missing.</p1:message>
  </p1:errorData>
</p1:root>
```

Priebeh testu:

- Spustenie internetového prehliadača
- Vloženie URI – <http://aplikacny-server/api/getDepartures.api>
- Prijatie výslednej požiadavky – xml správy s informáciami o chybe
- Validácia oproti XSD schéme
- Výstupná správa musí obsahovať chybový kód podľa špecifikácie testu

## TS03: Služba getServices – chybné parametre

Účel scenára je otestovať správanie služby v prípade, že nie sú parametre zadané v správnom formáte.

Vstupné požiadavky:

Ako vstup slúži URI služby spolu s testovacími parametrami. Jeden z povinných parametrov je v nesprávnom formáte. Na serveri musí bežať testovaná služba. Databáza pre službu musí bežať a byť správne nakonfigurovaná. Databáza musí obsahovať testovacie dáta (súčasť dokumentu ako SQL skript, príloha 1). Tester musí mať prístup pre HTTP protokol na testovací server.

**Testovacia URI:**

<http://tp.ktokoho.info/api-dev/getDepartures.api?lat=8154&lon=17.0754&time=16:50&offset=60&radius=500>

**Očakávaný výstup:**

Služba vygeneruje chybovú správu s chybovým kódom 3. Podľa tabuľky chybových kódov (príloha 3) je chyba označená ako Parameter lat has bad type. Maybe, it is not float number.

**Očakávaná XML správa:**

```
<?xml version="1.0" encoding="UTF-8"?>
<p1:root xmlns:p1="http://www.itransit.sk/xsd/datamodel"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.itransit.sk/xsd/datamodel/datamodel.xsd">
  <p1:errorData>
    <p1:errorCode>3</p1:errorCode>
    <p1:message>Parameter lat has bad type. Maybe, it is not float
number</p1:message>
  </p1:errorData>
</p1:root>
```

**Priebeh testu:**

- Spustenie internetového prehliadača
- Vloženie URI – <http://aplikacny-server/api/getDepartures.api>
- Prijatie výslednej požiadavky – xml správy s informáciami o chybe
- Validácia proti XSD schéme
- Výstupná správa musí obsahovať chybový kód podľa špecifikácie testu

## TS04: Služba getServices – chyba spracovania

Účel scenára je otestovať správanie služby v prípade, že nastane chyba pri spracovaní požiadavky

### Vstupné požiadavky:

Ako vstup slúži URI služby spolu s testovacími parametrami. Na serveri musí bežať testovaná služba. Služba má nesprávne nakonfigurovaný prístup k databázovému serveru. Tester musí mať prístup pre HTTP protokol na testovací server.

### Testovacia URI:

<http://tp.ktokoho.info/api-dev/getDepartures.api?lat=48.154&lon=17.0754&time=16:50&offset=60&radius=500>

### Očakávaný výstup:

Služba vygeneruje chybovú správu s chybovým kódom 7. Podľa tabuľky chybových kódov (príloha 3) je chyba označená ako Cannot connect to DB.

### Očakávaná XML správa:

```
<?xml version="1.0" encoding="UTF-8"?>
<p1:root xmlns:p1="http://www.itransit.sk/xsd/datamodel"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.itransit.sk/xsd/datamodel/datamodel.xsd">
  <p1:errorData>
    <p1:errorCode>7</p1:errorCode>
    <p1:message>Cannot connect to DB.</p1:message>
  </p1:errorData>
</p1:root>
```

### Priebeh testu:

- Spustenie internetového prehliadača
- Vloženie URI – <http://aplikacny-server/api/getDepartures.api>
- Prijatie výslednej požiadavky – xml správy s informáciami o chybe
- Validácia oproti XSD schéme
- Výstupná správa musí obsahovať chybový kód podľa špecifikácie testu

## TS05 – klientská časť – používateľské rozhranie

Tento testovací scenár má za úlohu otestovať chyby v používateľskom rozhraní. Je preto potrebné eliminovať ostatné premenné (ako spracovanie správ, prípadne komunikáciu so serverom)

**Vstupné požiadavky:**

Aplikácia bežiaca v prostredí iPhone simulátora XCode. Pre naplnenie používateľského prostredia je použitý MOCK objekt obsahujúci predvyplnené údaje o spojoch.

**Očakávaný výstup:**

Používateľské rozhranie zobrazí všetky údaje z MOCK objektu bez viditeľných chýb v zobrazovaní.

**Priebeh testu:**

- Nastavenie MOCK objektu pri testovaní rozhrania
- Inicializácia objektu hodnotami
- Spustenie aplikácie v simulátore
- Vizuálna inšpekcia používateľského rozhrania

**TS06 – klientská časť – rozhranie klient/server**

Tento testovací scenár má za úlohu otestovať rozhranie klientskej aplikácie cez ktorú komunikuje so serverovou časťou. Zadaním požiadavky na získanie údajov do servera sa musí vyslať požiadavka, prijať výsledok a naplniť dátami príslušné dátové entity. Ako zdroj dát je potrebné použiť MOCK službu vracajúcu statické dáta – XML správu (príloha 3).

**Vstupné požiadavky:**

Aplikácia bežiaca v prostredí iPhone simulátora XCode. Serverová časť emulovaná MOCK službou, vracajúca statickú XML správu. Simulátor má prístup na MOCK službu.

**Očakávaný výstup:**

Rozhranie vykoná dopyt, získa dáta a naplní príslušné dátové entity.

**Priebeh testu:**

- Nastavenie MOCK služby
- Vykonanie požiadavky na rozhranie klient/server pomocou Unit Testu
- Overenie prijatia výsledku
- Kontrola dátových objektov UnitTestom

**TS07 – komunikácia klient-server**

Tento testovací scenár má za úlohu otestovať integráciu oboch súčastí do funkčného celku

Vstupné požiadavky:

Aplikácia bežiaci v prostredí iPhone simulátora XCode. Služba bežiaci na aplikačnom serveri. Simulátor má prístup na aplikačný server. Databáza pre službu musí bežať a byť správne nakonfigurovaná. Databáza musí obsahovať testovacie dáta (súčasť dokumentu ako SQL skript, príloha Tester musí mať prístup pre HTTP protokol na testovací server. GPS súradnice posielané ako parameter do služby sú emulované MOCK objektom zastupujúcim GPS senzor.

Súradnice sú nastavené na: 48.154,17.076.

### Očakávaný výstup:

Používateľské rozhranie zobrazí odchody spojov pre nasledovné zastávky:

- ZOO
- Televízia

### Priebeh testu:

- Nastavenie a spustenie služby
- Nastavenie MOCK objektu ako GPS senzora
- Spustenie aplikácie v simulátore
- Vizuálna inšpekcia používateľského rozhrania
- Kontrola výpočtu časov do odchodu. Každý riadok obsahujúci čas odchodu obsahuje v zátvorke zostávajúci čas, Výpočert AKTUÁLNY ČAS – ODCHOD musí byť rovný tomuto údaju.

### Výsledky testov

Táto časť dokumentu obsahuje tabuľku s výsledkami testovacích scenárov po ich vykonaní.

Tab. A-1. Výsledky testovacích scenárov.

Číslo	Názov	Tester	Výsledok
TS01	Služba getServices – úspešné prevzatie správy	Martin Jačala	OK
TS02	Služba getServices – chýbajúce parametre	Marek Brandobúr	OK
TS03	Služba getServices – chybné parameter	Michal Macko	OK
TS04	Služba getServices – chyba spracovania	Michal Macko	OK
TS05	klientská časť – používateľské rozhranie	Hana Časnochová	OK
TS06	klientská časť – rozhranie klient/server	-	netestované
TS07	komunikácia klient-server	Hana Časnochová	OK



## PRÍLOHA B : DIZAJN



Obr. B- 1. Dizajn stránky produktu.

The banner features a light blue background. At the top left, it says "domov | vlastnosti | o nás | prevziať". At the top right, it says "english version | slovenská verzia". The main title "iTransit" is in a large, black, sans-serif font, with a yellow starburst graphic to its right that says "Už žoskorol!". Below the title is the tagline "Už nikdy nezmeškaj žiadny spoj!". On the right, there is a navigation bar with the text "Tvoj osobný navigátor v spleti mestskej hromadnej dopravy". In the center, three smartphones are shown: the first displays a map with green location markers, the second shows a list of items, and the third shows a detailed view of an item. On the left, there are five bullet points describing the app's features: "Cestovné poriadky pre MHD v Bratislave", "Kompletný zoznam dostupných zastávok", "Množstvo rozličných bodov záujmu", "Jednoduché a prehľadné ovládanie", and "Mapy integrované s GPS prijímačom". At the bottom left, it says "... navyše, iTransit je zdarma!" and "Vyskúšaj hneď a odporúčaj svojim známym!".

Obr. B- 2. Zmenený dizajn stránky produktu.

Cestuješ MHD v Bratislave?

**Áno?**

Používaš iPhone,  
prípadne iný smartphone?

Si náš človek!

Sme tím mladých ľudí s ambíciou vytvoriť inteligentné cestovné poriadky pre mobilné telefóny a sprístupniť ju všetkým a **zadarmo**.

**Zaujalo Ťa to?** Môžeš nám **pomôcť!**

Potrebujeme počuť Tvoje nápady a postrehy,  
aby sme pri tvorbe aplikácie na niečo nezabudli.  
Za odmenu Ťa upozorníme e-mailom, keď bude aplikácia k dispozícii :)




[www.iTransit.sk](http://www.iTransit.sk)

Obr. B- 3. Plagát na podporu stránky produktu.

Cestuješ bratislavskou **MHD**?

Používaš iPhone alebo iný smartphone?



[www.iTransit.sk](http://www.iTransit.sk)

Obr. B- 4. Ukážky z reklamného prúžku.