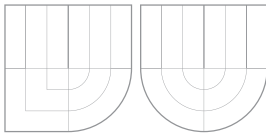


VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ



FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

ZPRACOVÁNÍ GPS DAT NA IPHONE

GPS DATA PROCESSING ON IPHONE

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

RADEK LIŠKA

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. RUDOLF KAJAN

BRNO 2010

Abstrakt

Tématem této bakalářské práce jsou moderní webové technologie a jejich využití v mobilních zařízeních se zaměřením na Apple iPhone. Aplikace pracuje jako organizér závodů. Mezi hlavní funkce patří tvorba a správa tratí, pomoc závodníkům při jejich absolvování a uchování rekordů, kterých dosáhli.

Abstract

Subject of this thesis is modern web technologies and their usage in mobile devices, with focus on the Apple iPhone. Application works as a race organizer. It's main functions consists of creation a management of tracks, support to racing users and storage of records.

Klíčová slova

GPS, WGS-84, iPhone, webová aplikace

Keywords

GPS, WGS-84, iPhone, web application

Citace

Radek Liška: Zpracování GPS dat na iPhone, bakalářská práce, Brno, FIT VUT v Brně, 2010

Zpracování GPS dat na iPhone

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením pana Ing. Rudolfa Kajana. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

.....

Radek Liška
17. května 2010

Poděkování

Rád bych poděkoval panu Ing. Rudolfovi Kajanovi za poskytnutou podporu při tvorbě práce.

© Radek Liška, 2010.

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	Analýza aplikace	3
2.1	Výběr implementačního prostředí	3
2.2	Specifikace funkce aplikace	4
2.3	Analýza cílové platformy	4
2.3.1	Historie a budoucnost	5
2.3.2	Hardware	6
2.3.3	Software	8
2.4	Analýza dostupných technologií	9
2.5	Geodetický systém WGS84	11
3	Návrh aplikace	14
3.1	Návrh principu závodění	14
3.2	Návrh uživatelského rozhraní	15
4	Implementace aplikace	20
4.1	jQuery	20
4.2	Google Maps API v3	20
4.3	Navigator Geolocation	21
4.4	Offline aplikace	21
4.5	Implementace uživatelského rozhraní	21
5	Závěr	24

Kapitola 1

Úvod

Technologický pokrok v oblasti chytrých mobilních zařízení pro každodenní použití zažívá v posledních několika letech veliký rozmach. Tato problematika mě vždy velice zajímala, a proto jsem se rozhodl věnovat se tomuto tématu ve své bakalářské práci.

Tato práce se zabývá výrobou webové aplikace pro mobilní telefon Apple iPhone založené na práci s GPS daty. Jejím cílem je vytvoření aplikace typu klient-server s maximálním využitím moderních technologií, které přístroj podporuje. Důraz budu klást také na přehledné dotykové uživatelské rozhraní a alespoň částečnou dostupnost i na jiných platformách.

Celý text je rozdělen tak, že kapitoly odpovídají jednotlivým částem vývoje. První část se věnuje analýze. V ní specifikuji funkci aplikace a seznámím čtenáře s dostupnými možnostmi jak u klienta, tak na serveru. Součástí této kapitoly jsou i základní informace o systémech GPS a WGS84. Z analýzy vychází návrh, ve kterém na základě zjištěných informací vytvořím základní konstrukce, jež budou využity v realizaci aplikace. V této části diskutuji především návrh logiky aplikace a uživatelského rozhraní. Předposlední kapitola se věnuje samotné implementaci projektu. Je zde podrobně řešen způsob vypracování projektu, včetně popisu použitých knihoven třetích stran. Závěrečná kapitola sumarizuje dosažené výsledky a přínos aplikace. Na konec jsou uvedeny plány na možné budoucí vylepšení.

Kapitola 2

Analýza aplikace

Na začátku celé práce je nutné provést analýzu aplikace. V této kapitole stanovím cíle aplikace, pokusím se odhalit možné problémy a získat co nejvíce informací o možnostech jejich řešení. Tím pološím základ návrhu aplikace, který bude z této kapitoly vycházet.

2.1 Výběr implementačního prostředí

První krok, který jsem musel udělat, bylo určit, v jakém prostředí implementuji moji aplikaci. Při studiu této části jsem používal webové stránky vývojářských center pro iPhone[2] a Safari[5], ze kterých jsem čerpal základní informace o možnostech vývoje aplikací na iPhone. Z těchto podkladů vyplynuly následující dva technologické směry, jejichž stručnou charakteristiku nyní uvedu.

- **Nativní aplikace** - pro implementaci se používá jazyk Objective-C, s knihovnou Cocoa. Vývoj je možný pouze na strojích Macintosh. Veřejně distribuovat hotové aplikace mají povoleno pouze členové iPhone Developer Program a k zveřejnění dojde až po schválení ze strany Apple. Hlavní výhodou tohoto způsobu tvorby aplikací je maximální využití výkonu a všech periférií, který iPhone nabízí.
- **Webová aplikace** - prostřednictvím vestavěného prohlížeče internetových stránek Safari, je možné na iPhone spouštět i takzvané webové aplikace. Při jejich vývoji se používají klasické webové technologie jako je (X)HTML, CSS a JavaScript. Hotová aplikace pak může být přístupná nejen na iPhone, ale i z jiných zařízení. Distribuce je snadná, stačí aplikaci umístit na webový server, jehož URL adresu uživatel zadá do prohlížeče. Nevýhody tohoto řešení jsou především v zabezpečení a omezené možnosti ve srovnání s nativní aplikací. Konkrétní tuto problematiku diskutuji v sekcích [2.3.2](#) a [2.3.3](#).

Po zhodnocení všech výhod a nevýhod obou přístupů, jsem se rozhodl pro webovou aplikaci. Hlavními důvody byla relativní snadnost vývoje a distribuce. Zároveň pro mě bude výzva vytvořit takovou aplikaci, která bude snadno použitelná i na více platformách a nebude náročná na výkon klienta. Cílovou platformou bude v souladu se zadáním této práce iPhone, u kterého bude zaručeno využití všech možností aplikace. Výčet všech podporovaných typů naleznete v sekci [2.3.1](#). U ostatních platform by měla být zajištěna alespoň základní funkčnost (např. tvorba tratí), za předpokladu že bude použitý prohlížeč podporovat alespoň některé stěžejní technologie ze Safari.

2.2 Specifikace funkce aplikace

Ze zadání vyplývá, že hlavní úlohou aplikace bude zpracování GPS dat. Konkrétní téma jsem po pečlivém zvážení specifikoval jako organizér závodů. Mezi hlavní důvody pro tuto volbu patří především absence podobné aplikace řešící tento problém a dobrá perspektiva reálného využití.

Aplikace bude zabezpečovat logiku organizace závodů pro libovolný počet závodníků. Tím mám na mysli tvorbu tratí pro závodění, vedení uživatelů při jejich absolvování, správu závodníků a výkonů, kterých dosáhli. Volba dopravních prostředků by měla být zcela ponechána na uživatelích, správná funkce tak musí být zajištěna jak při pomalém pohybu (např. běh, jízda na kole), tak i ve větších rychlostech (např. auto, vlak). Jednotlivé tratě musí být veřejně a perzistentně uchovány v systému, aby bylo možné jejich absolvování více uživateli zároveň. Každý závodník absolvuje trať izolovaně, může si tak vybrat okamžik kdy odstartuje. Výsledný čas se prezentuje ve formě intervalu mezi dosažením startu a cíle. Systém musí vhodným způsobem prezentovat výsledné časy závodníků v závislosti na konkrétní trati.

Pro splnění výše uvedených požadavků bude muset aplikace pracovat na architektuře klient-server. Na serveru budou uložena data (tratě, rekordy, závodníci), ke kterým budou mít přístup všichni uživatelé. Klienti budou k serveru přistupovat pomocí webových stránek. Po úvodním načtení musí být následná komunikace pouze asynchronní. Zabrání se tak zbytečnému opětovnému přenosu celého obsahu stránek a také ztrátě kontextu, protože protokol HTTP je bezstavový.

2.3 Analýza cílové platformy

V této sekci se budu věnovat popisu cílové platformy projektu, mobilnímu telefonu Apple iPhone. Podrobně budu prezentovat možnosti, jenz přístroj nabízí a zaměřím se na ty, které plánuji v projektu využít. Zmíním se i o historii iPhone a plánech do budoucna na možné další přírůstky do rodiny těchto přístrojů.



Obrázek 2.1: Pohled na Apple iPhone 3G.

Mezi podklady, které jsem při studiu této sekce používal, patří především webové stránky společnosti Apple prezentující iPhone[1] a také bakalářská práce Ondřeje Fabiána [6].

2.3.1 Historie a budoucnost

iPhone je produkt společnosti Apple Inc., který v sobě spojuje funkce mobilního telefonu, digitálního fotoaparátu, videokamery, multimediálního přehrávače a zařízení pro komunikaci s internetem. V současné době existují již tři generace tohoto přístroje. Rozdíly mezi nimi stručně popíši v této sekci.

iPhone 2G První verze spatřila světlo světa v létě roku 2007. Původně byla nazývána pouze iPhone, přídomek 2G byl přidán až zpětně, kvůli potřebě odlišit tento model od názvu celé rodiny přístrojů, jenž na jeho základě vznikla. Z pohledu této bakalářské práce, je bohužel první generace iPhone nepoužitelná. Chybí jí totiž to hlavní: GPS lokátor. Stále by ale mělo být možné používat přístroj k operacím, které ho nevyžadují, např. vytváření tras a závodů.

iPhone 3G Druhá verze byla vydána o rok později, tedy v létě 2008. Sklidila obrovský úspěch a za první rok bylo prodáno téměř 20 miliónů přístrojů. Mezi hlavní novinky v této verzi patří podpora sítí 3. generace (rychlejší přenos dat) a integrace GPS lokátoru. Tato verze přístroje je v mé bakalářské práci považována za cílovou. To je dáno především díky jejím masovému rozšíření a splněním základních požadavků kladených touto aplikací.

iPhone 3GS V zažitém tempu, kdy každý rok vychází nová generace, byl v létě roku 2009 vydán iPhone 3GS. Písmeno S v názvu má znamenat Speed, což předznamenává jeho hlavní inovaci a to zvýšení výkonu hardwaru (hlavně procesoru, paměti a grafického koprocesoru). Za zmínku stojí také lepší akumulátor a fotoaparát. GPS modul konečně obsahuje také digitální kompas. Ve vztahu k mé bakalářské práci je tento model již nadstandardem. To ale neznamená, že pro něj nebude určen. Pouze z důvodu zpětné kompatibility s předchozí generací, nebudu využívat novinky, které iPhone 3GS nabízí.

iPhone 4G V poslední době se na webech zaměřených na iPhone začíná diskutovat o nové generaci přístroje. Nepochybně bude zveřejněna letošní léto, ale žádné velké změny se od ní nečekají. Předpokládám ale, že přístroj bude zpětně kompatibilní a tudíž nebude problém na něm mojí aplikaci provozovat.

iPod Touch Společnost Apple nabízí v současnosti několik řad sofistikovaných mobilních přístrojů. Z pohledu mé aplikace je kromě rodiny iPhone zajímavý ještě iPod Touch. V podstatě jde o ořezaný iPhone, kterému chybí většina komunikačních periférií jako GPS nebo mobilní datové sítě. Zachován je pouze Bluetooth a WIFI. Tím je dáno, že tento přístroj půjde využít pouze pro prohlížení výsledků nebo vytváření map. Konkrétní optimalizaci ovšem nejsem schopen zajistit, především pro velmi malé rozšíření tohoto přístroje v našich podmínkách.

iPad Na závěr této sekce bych se ještě rád zmínil o novince, která se dostala před nedávnem na trh. iPad je multimediální počítač typu tablet. Primárně je určený pro prohlížení internetu a e-knih, běží ovšem na stejném operačním systému jako rodina iPhone, a tudíž by mělo být možné provozovat mojí aplikaci i na něm. Jeho velký displej (úhlopříčka

25cm) by k tomu byl více než vhodný. Bohužel verze se zabudovaným GPS lokátorem ještě není v prodeji, a tak optimalizace aplikace i pro tuto platformu zůstává pouze plánem do budoucna.

2.3.2 Hardware

Po hardwarové stránce se iPhone skládá z vnitřních komponent a několika vstupních a výstupních zařízení. V následující sekci jednotlivé části stručně rozeberu a u těch z hlediska aplikace zajímavých uvedu i podrobnější informace.

Přístroj má rozměry $115\text{mm} \times 62\text{mm} \times 12\text{mm}$ a hmotnost 133g. Od verze 3G má výhradně plastový kryt (kvůli lepšímu příjmu GSM a GPS signálu).

Processor

Patří do architektury RISC, u které je preferována malá, ale vysoce optimalizovaná sada strojových instrukcí. Konkrétně jde o model z řady ARM, kterou vyvíjí společnost ARM Limited. V první a druhé generaci přístroje nalezneme procesor Samsung S5L8900 s frekvencí 412MHz. iPhone 3GS má zabudován lepší procesor Cortex-A8 s frekvencí 600MHz. U obou je integrován grafický koprocesor PowerVR MBX 3D (u 3GS ve výkonnější verzi Lite 3D).

Operační paměť

Přístroj je osazený pamětí typu DRAM. U 2G a 3G má velikost 128 MB, 3GS pak dokonce 256 MB. Tento typ paměti má poměrně velkou spotřebu energie, na druhé straně má nízké výrobní náklady. V příštích verzích iPhone proto můžeme docela reálně očekávat přechod na jiný typ paměti.

Flash paměť

Pro permanentní uchování dat je v iPhone instalována flash paměť, typicky ve velikostech 8, 16 nebo 32 GB.

Komunikační periferie

S přihlédnutím k faktu, že iPhone je především mobilní komunikační zařízení, nebude jistě velkým překvapením přítomnost mnoha bezdrátových komunikačních periferií. Popis jednotlivých technologií je nad rámec této práce, proto zde uvedu jen prostý výčet: WiFi (IEEE 802.11 b,g), Bluetooth a GPRS (respektive UMTS). Přístroj automaticky používá nejrychlejší dostupné připojení, což na jedné straně přináší výhodu v tom, že aplikace se nemusí starat o připojení k internetu, na druhé straně nemá možnost toto připojení ovlivňovat (např. vyžadovat WiFi pro přenos větších dat).

GPS přijmač

Před popisem této komponenty by bylo vhodné specifikovat co vlastně pojem GPS znamená. Je to zkratka z anglického *Global Positioning System*, neboli Globální Poziční Systém. Jde o polohový družicový systém, jeho hlavní funkcí je určování polohy na zemském povrchu. Byl primárně vyvinut pro vojenské potřeby Spojených států amerických. Od svého

nasazení v roce 1994 je však přístupný i civilním složkám, i když do roku 2000 pouze s omezenou přesností. V dnešní době lze pomocí systému GPS zjistit polohu s přesností cca 3 metrů.

Celý systém se skládá ze tří segmentů: *kosmického* (32 vzájemně komunikujících satelitů na oběžné dráze země), *řídícího* (většina umístěna na území USA, dozoruje a ovládá funkci kosmického segmentu) a *uživatelského* (v tomto případě iPhone). Uživatelský segment je striktně pasivní, pouze přímá signály z jednotlivých družic a na základě získaných dat určuje aktuální polohu. Ta je interpretována na geografickém referenčním systému WGS 84. Teto systém podrobně rozebírám v sekci 2.5.

V iPhone je integrován GPS přijímač kompatibilní s technologií A-GPS (Assisted GPS), ta je určena pro mobilní přijímače a vylepšuje rychlost zjišťování pozice, obzvláště v nepříznivém prostředí jako je například město. Zde totiž dochází k odrazům družicových signálů mezi budovami a určení přesné polohy je velmi obtížné. Tento problém se řeší tak, že k výpočtu aktuální pozice přispívají i data o poloze získané z přístupových bodů WIFI (technologie WPS) a také pomocí triangulace GSM signálu.

U starších modelů (2G a 3G) bohužel není součástí GPS přijímače kompas. Nelze tak detekovat, jakým směrem je přístroj momentálně natočen. Jedinou možností je porovnávat aktuální pozici s poslední známou a určit jejich azimut. Z diskuzí u aplikací založených na tomto přístupu jsem vypořádal, že u uživatelů taková funkčnost zůstává nepochopená. Většina při pohledu na kompas očekává aktuální informace, které bohužel přístroj nedokáže v klidové poloze poskytnout. Proto ve své aplikaci tento přístup nebudu využívat a o správném směru budu uživatele informovat pomocí mapy a aktuální vzdálenosti k příštím bodu tratě.

Displej

Dotykový displej je nejdůležitější periferií přístroje. Jde o vstupně-výstupní zařízení, které se využívá v drtivé většině komunikace mezi přístrojem a uživatelem. Má velikost úhlopříčky 89mm a rozlišení 480px × 320px (poměr 3:2). Barevná paleta obsahuje 262144 barev. Mezi hlavní přednosti displeje patří patentovaná technologie *multi-touch*. Ta umožňuje uživateli používat u vstupních operací několik prstů zároveň a poskytuje tak intuitivní rozhraní pro specifikaci komplexních příkazů (rotace, zvětšení).

Tlačítka

Pro někoho může být velkým překvapením, že na iPhone nalezneme pouhá čtyři tlačítka a jeden přepínač. Není zde přítomna hardwarová klávesnice, která je neodmyslitelnou součástí drtivé většiny ostatních mobilních přístrojů. Navíc ani jedno není přístupné pro aplikace, protože veškeré tlačítkové vstupy odchyťává operační systém. Z toho vyplývá, že pro komunikaci s uživatelem je silně preferován dotykový displej. Je klíčové si tento fakt uvědomit a odpovídajícím způsobem ho zohlednit ve způsobu realizace ovládání uživatelského rozhraní.

Akcelerometr

V iPhone je integrován i akcelerometr, což je vstupní zařízení, které detekuje změnu natočení přístroje v prostoru. Mezi jeho nejčastější využití patří přepínání mezi zobrazením

na výšku (portrait) a na šířku (landscape). Uživatel si tak může vybrat, které zobrazení mu více vyhovuje a přístroj o tom informovat prostým natočením do požadované pozice. Akcelerometr ovšem nabízí i mnohem větší přesnost a lze díky němu naprogramovat aplikaci pracující například jako vodováha.

Ostatní periferie

Ačkoliv se výčet jednotlivých periférií je již teď velmi dlouhý, ještě pořád jsem neuvedl všechny. Jejich výčet zde pro úplnost uvedu. Především jde o *fotoaparát* (2Mpx respektive 3Mpx u 3GS) a *reproduktory s mikrofonem*. Mezi ty méně zajímavé periferie patří například *senzor intenzity světla* (používaný pro automatické nastavení jasu displeje), nebo *senzor blízkých předmětů* (díky němu se automaticky vypíná displej při přiložení k uchu).

2.3.3 Software

Následující sekci věnuji seznámení se softwarovou výbavou iPhone. Stručně popíši i jednotlivé historické verze a budu diskutovat jejich kompatibilitu s mojí aplikací.

Operační systém

V poslední době stalo standardem, že mobilní telefony v sobě mají zabudovaný plnohodnotný operační systém. iPhone v tomto směru není výjimkou, běží na *iPhone OS*. Jak již vyplývá z názvu, jde o operační systém vyvinutý speciálně pro iPhone. Vznikl ze systému *Mac OS X*, který je určený pro počítače Mac. Oba operační systémy patří do rodiny UNIX. Od této skutečnosti se odvíjí celá řada atributů: např. organizace souborového systému (běžnému uživateli je ale zatajena), orientace na síťovou komunikaci nebo důraz na vztahy mezi programy (mnoho navzájem propojených jednoúčelových nástrojů). S Unixovou filozofií se naopak rozchází v jeho uzavřenosti (není open-source), absencí víceúlohovosti (multi-tasking) nebo nepřítomností podpory pro více uživatelů na jednom přístroji.

Stejně jako hardware tak i iPhone OS podléhá pravidelným aktualizacím. Výčet jednotlivých změn zasahuje nad rámec této bakalářské práce. Z hlediska mojí aplikace je klíčová verze 3.0, která přináší podporu geolokace prostřednictvím internetového prohlížeče. Z toho důvodu je tato verze považována za cílovou a je vyžadována pro bezproblémový chod aplikace.

Nativní aplikace

Součástí softwarové výbavy iPhone jsou nativní aplikace a widgety, které mohou ostatní programy využívat. Za zmínku stojí například softwarová klávesnice nebo různá pop-up okna. Z aplikací je přístupný internetový prohlížeč Safari, emailový klient, přehrávač YouTube videí, poznámkový blok nebo aplikace pro uskutečňování hovorů či ovládání fotoaparátu. Některé funkce těchto aplikací jsou veřejně přístupné a určitě zvážím jejich použití v mém projektu.

Safari

Safari je webovým prohlížečem vyvinutým společností Apple. Jeho počátky sahají do roku 2003 a od roku 2005 je jediným prohlížečem dodávaným s operačním systémem OS X. Kromě toho ho najdeme i na mobilních zařízeních jako je iPod, iPhone nebo iPad. V nedávné

době vyšel také ve verzi pro operační systémy Windows XP a novější. V současnosti je prohlížeč dostupný ve verzi 4.

Z technologického pohledu je Safari poháněno renderovacím jádrem **Webkit**[12], což je opensource projekt, jehož počátky můžeme nalézt u prohlížeče Konqueror z linuxového prostředí KDE. Kromě Safari ho používají i prohlížeče *Google Chrome*, *Marthon* a dokonce i v *Android Web browser*, *Palm WebOS* nebo *Symbian S60*. V současnosti jde o jedno z nejpokročilejších zobrazovacích jader, které podporuje spoustu moderních technologií, a tím posunuje možnosti webových aplikací na novou úroveň. V mé aplikaci se budu snažit těchto technologií co nejvíce využít. Verze instalovaná v iPhone se ovšem od ostatních v některých ohledech liší. Nejvíce omezující je absence technologií Flash a Java, Apple to zdůvodňuje nedostatečným výkonem přístroje.

2.4 Analýza dostupných technologií

XHTML

Neboli *Extensible HyperText Markup Language*[14] je značkovací jazyk pro tvorbu hypertextových dokumentů vytvořený jako nástupce jazyka HTML4[16]. Momentálně se chystá specifikace nové verze HTML5[17], a právě některé připravované vlastnosti této specifikace podporují webové prohlížeče již dnes. Já se přirozeně zaměřím na ty implementované v jádře Webkit. Ve výsledku tak sice vznikne invalidní dokument, někde na pomezí XHTML a HTML5, ale pokud chci tyto pokročilé webové technologie využívat, tak do zveřejnění konečné specifikace pro HTML5 jinou možnost nemám.

CSS

Cascading Style Sheets[13], neboli česky *Kaskádové styly*, jsou jazyk pro popis zobrazení dokumentů napsaných ve značkovacích jazycích HTML, XHTML a XML. Opět se jedná o výtvar konzorcium W3C. Jeho hlavní myšlenkou je oddělit vzhled dokumentu od jeho obsahu. To je realizováno pomocí definice pravidel, které se skládají ze selektor a bloku deklarácí. Selektor určuje na jaké elementy z DOM (*Document Object Model*) se má pravidlo aplikovat. Blok deklarácí obsahuje seznam deklarácí, jež přiřazují jednotlivým vlastnostem a nové hodnoty. V současné době je nejrozšířenější verze CSS2, ale s blížícím se příchodem HTML5, se připravuje i verze CSS3, která obsahuje řadu velmi zajímavých vylepšení, které si prostě nemohu dovolit nevyužít. Oproti HTML5 je zde způsob podpory v jednotlivých prohlížečích velmi odlišný. V drtivé většině se totiž nepoužívají názvy vlastností podle CSS3 specifikace, ale každé vykreslovací jádro podporuje své vlastní názvy (např. s prefixem `-webkit` u jádra Webkit nebo `-moz` u jádra Gecko).

Javascript

Javascript je objektově orientovaný skriptovací jazyk. Jeho autorem je Brendan Eich z dnes již neexistující společnosti Netscape. V současnosti se jeho hlavní využití jako interpretovaný programovací jazyk pro webové stránky. K tomu ho budu využívat i ve své aplikaci. Jeho podpora v prohlížeči je striktně vyžadována, protože zajišťuje veškeré interakce a funkce klientské části aplikace. Syntaxe jazyka patří do rodiny C/C++/Java.

Přítomnost slova Java v názvu je pouze z marketingových důvodů a kromě některých podobností v syntaxi, spolu tyto jazyky nemají nic společného. Existuje také standardizovaná verze s názvem *ECMAScript*.

Z pohledu objektově orientovaného programování je JavaScript dynamickým, funkčním a prototypově orientovaným jazykem. Z toho vyplývají následující vlastnosti: Typické je pro něj dynamické přiřazení typů, to znamená, že za běhu může jedna proměnná obsahovat různé datové typy. Existuje v něm funkce `eval`, která je schopná provádět příkazy předávané jako řetězce, toho se využívá například u technologie AJAX. Každá funkce v JavaScriptu je ve skutečnosti objekt, při použití klíčového slova `new` před jejím voláním se vytvoří kopie tohoto objektu. Objekty jsou v JavaScriptu reprezentovány jako asociativní pole rozšířené o prototypy. Pomocí nich je realizována dědičnost, nejedná se však o klasickou dědičnost jako například v jazycích Java nebo C++. Prototyp slouží jako zvláštní objekt, který je přidružen ke každému konstruktoru (funkci) a všechny objekty skrze něj inicializované dědí (respektive sdílejí) všechny položky obsažené v daném prototypu. Metody a atributy, které nejsou uvedeny v prototypu, se při vytváření instance třídy do nového objektu zkopírují. Na závěr stojí za zmínku také přítomnost konstrukcí pro zpracování výjimek, které se používají pro ošetření chybových stavů.

PHP

PHP: Hypertext Preprocessor je objektově orientovaný skriptovací jazyk. Jeho první verzi vytvořil Rasmus Lerdorf. Vychází z jazyka Perl a je určený pro programování dynamických internetových stránek. PHP kód je spouštěn na serveru a jeho výstupem je strukturovaný dokument (většinou HTML), který se odesílá klientovi. Ve své aplikaci ho budu využívat pro implementaci serverové části. Pomocí něj se budou vyřizovat požadavky od klientů a přistupovat se do serverové databáze. Mezi hlavní výhody tohoto jazyka, které mě přesvědčili k jeho použití, patří jeho velké rozšíření mezi poskytovateli webhostingu, neměl by tak být problém aplikaci v případě potřeby přenést na jiný server. Dále pak jeho svobodná licence a poměrně slušná dokumentace (i když paradoxně neexistuje přesná definice jazyka).

MySQL

MySQL je databázový systém vycházející z jazyka SQL. V současnosti ho vlastní společnost Sun Microsystems. Jeho hlavní funkcí je uchování strukturovaných dat. V mojí aplikaci bude tato databáze uložena na serveru, odkud bude prostřednictvím PHP přístupná klientům. Pro tvorbu a správu MySQL databáze použiji aplikaci phpMyAdmin.

AJAX

Asynchronous JavaScript and XML je označení technologie pro vývoj interaktivních webových aplikací. Jeho podstatou je změna obsahu webové stránky bez nutnosti znovu načtení. Toho se dosáhne pomocí rozhraní `XMLHttpRequest`, přes které se mezi serverem a klientem přenášejí data v dohodnutém formátu, většinou XML, HTML, JSON nebo v prostém textu. Je ovšem nutné, aby bylo toto rozhraní podporováno prohlížečem. AJAX představuje velmi výrazné oživení mojí aplikace, pomocí něj jsem schopen omezit síťovou komunikaci na nutné minimum, což je u aplikace určené pro přístup přes mobilní datové přenosy nutností.

JSON

JavaScript Object Notation, neboli česky *JavaScriptový objektový zápis*, je způsob reprezentace dat založený na jeho snadném uložení a přenosu po síti. Principiálně je velice podobný algoritmu serializace. Vstupem může být libovolná datová struktura jazyka JavaScript (např. číslo, pole, objekt), ta je transformována na výstup, který je vždy typu řetězec. Výstup je kódován pouze za pomoci horní poloviny ASCII tabulky, toho dosahuje pomocí nahrazení nestandardních znaků za odpovídající escape sekvenci (např. znak `á` se převede na `\u00e1`). Hlavní využití JSON spočívá v rámci technologie AJAX, kde se využívá pro kódování přenášených dat. Pro tento účel jsem měl zpočátku v plánu použít značkovací jazyk XML, ale nakonec padla volba na JSON. Hlavní výhodou je snadné dekodování pomocí funkce `eval` a také mnohem menší velikost výstupního řetězce ve srovnání s XML.

2.5 Geodetický systém WGS84

Na závěr analýzy bych ještě rád popsal souřadný systém WGS84, který používá GPS. Ve studiu této sekce mi nejvíce pomohla oficiální dokumentace[9] a bakalářské práce Michala Kuchty[11] a Ondřeje Ilčíka[8].

Popis systému

Geodetický standard WGS84 vydaný ministerstvem obrany USA definuje souřadnicový systém pro popis polohy bodu na Zemi. Z důvodu nepravidelnosti jejího povrchu by ale bylo technicky příliš náročné určovat skutečnou polohu bodu, proto je zde použita abstrakce v podobě elipsoidu. Elipsoid je prostorové těleso tvořené množinou bodů daných následující nerovnicí:

$$\frac{x^2}{a^2} + \frac{y^2}{b^2} + \frac{z^2}{c^2} \leq 1$$

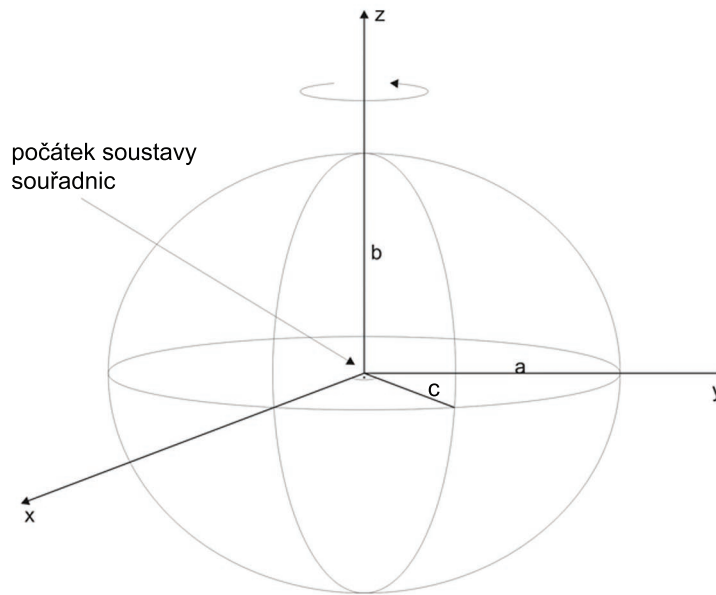
Konstanty a , b a c jsou kladná reálná čísla, určující délky jednotlivých poloos geoidu. V systému WGS84 mají následující hodnoty:

$$a = 6378137m$$

$$b = 6356752,3142m$$

$$c = 6399593,6258m$$

Jednotlivé souřadnice na elipsoidu jsou popsány pomocí dvou hodnot. První je *zeměpisná délka*, která se udává v rozmezí 0° až 90° . 0° je na rovníku, 90° na pólech. Druhou hodnotou je *zeměpisná šířka*. Ta se udává v rozmezí hodnot 0° až 180° , přičemž nultý stupeň prochází hvězdárnou Greenwich v Londýně.



Obrázek 2.2: Vizulizace souřadného systému WGS84.

Souřadnice se obvykle udávají jako dvě čísla ve stupních s určením, jestli se jedná o severní/jižní a východní/západní souřadnice. Formátování čísel může být buď pouze ve stupních, ve stupních a minutách nebo ve stupních, minutách a vteřinách. Pro demonstraci zde uvedu příklad zobrazení jedné polohy v různých formátech:

$$N49,2265666^\circ, E16,5958333^\circ$$

$$N49^\circ13,594', E16^\circ35,735'$$

$$N49^\circ13'60,64'', E16^\circ35'12,25''$$

Výpočet vzdálenosti dvou bodů

Jedinou početní operací, kterou budu v systému WGS84 provádět, je zjišťování délky *ortodromy*, což je nejkratší spojnice dvou bodů na zakřivené ploše. Úhlová délka ortodromy se dá vypočítat pomocí následujícího vztahu ze sférické trigonometrie za předpokladu, že si krajní body ortodromy označím jako $[\varphi_1, \lambda_1]$ a $[\varphi_2, \lambda_2]$:

$$\sigma = \arccos(\sin(\varphi_1) \cdot \sin(\varphi_2) + \cos(\varphi_1) \cdot \cos(\varphi_2) \cdot \cos(\lambda_2 - \lambda_1))$$

Tento vztah je bohužel velmi nepřesný pro malé vzdálenosti, proto se v geografii používá výrazně přesnější *harvesinova rovnice*, která představuje rozumný kompromis mezi přesností a složitostí výpočtu. Má následující tvar:

$$\sigma = 2 \cdot \arcsin\left(\sqrt{\sin^2\left(\frac{\varphi_2 - \varphi_1}{2}\right) + \cos(\varphi_1) \cdot \cos(\varphi_2) \cdot \sin^2\left(\frac{\lambda_2 - \lambda_1}{2}\right)}\right)$$

Převod z úhlové délky ortodromy na její délku na povrchu elipsoidu můžeme zjednodušeně provést jejím vynásobením s délkou hlavní poloosy

$$a = 6378137m$$

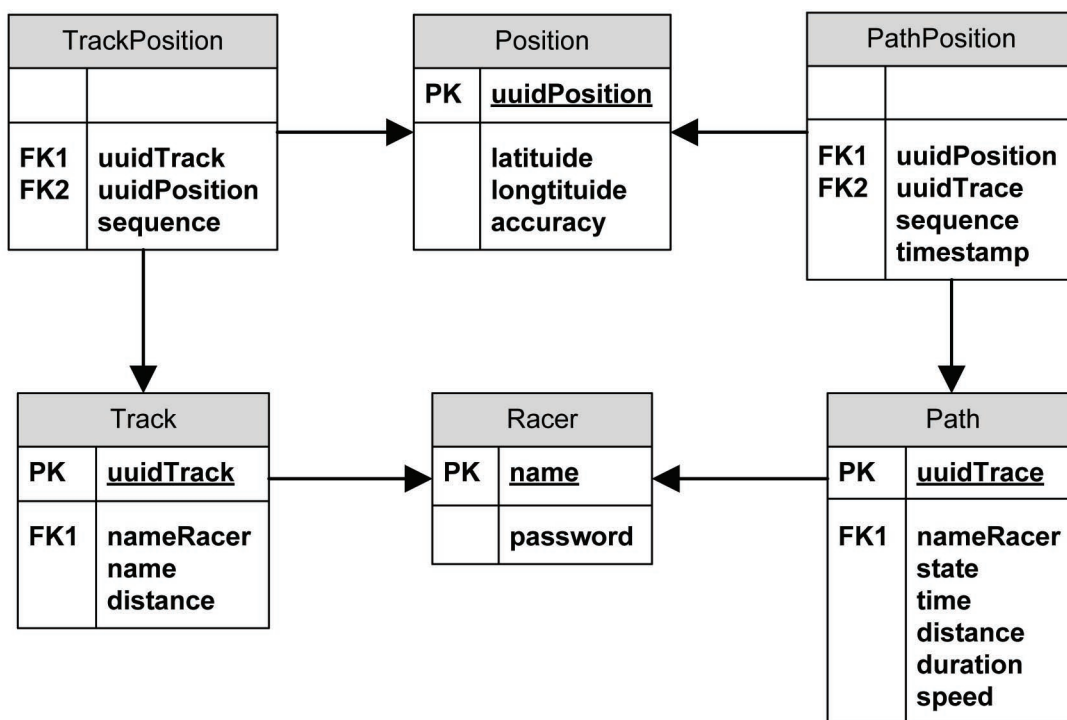
Kapitola 3

Návrh aplikace

V následující kapitole seznámím čtenáře s návrhem aplikace. Věnovat se budu především způsobu organizace závodů a také návrhu uživatelského rozhraní.

3.1 Návrh principu závodění

Následující sekce se bude věnovat návrhu principu, na jakém budou organizovány závody. Při jeho specifikaci jsem vycházel z požadavků formulovaných v sekci 2.2.



Obrázek 3.1: Model databáze reprezentující logiku principu závodění.

Pozice

Objekt sloužící k uchování informace o pozici v systému WGS84 diskutovaném v sekci 2.5. Z pozic jsou sestaveny všechny složitější konstrukce jako trať a trasa. Každou pozici

definuje její přesnost v metrech a zeměpisná šířka a délka.

Trať

Trať reprezentuje lineární posloupnost pozic, které musí závodník v průběhu závodu projít. V této posloupnosti jsou dva význačné prvky: startovní pozice a cílová pozice (po jejím dosažení závodníkem pro něj závod končí). Vytvořená trať, která je umístěna na server, už nelze upravit, protože jí může nějaký uživatel stáhnout a začít absolvovat. V takovémto stavu je pochopitelně nežádoucí jakákoliv změna tratě. Pro potřeby identifikace může autor nově vytvořené trati přiřadit název, automaticky je doplněna i její celková délka.

Trasa

Trasa popisuje pohyb závodníka v prostoru a času. Skládá se z lineární posloupnosti pozic rozšířené o sekvenční číslo a časový otisk reprezentující okamžik, ve kterém byli dosaženi. Z logiky věci vyplývá, že tyto časové otisky musejí být neklesající. Každá trasa je vázána na konkrétní trať a závodníka. Do trasy se započítávají všechny pozice, kterými závodník projde od dosažení startovního bodu až do okamžiku, kdy dorazí do cíle. Můžeme sledovat několik zajímavých atributů. Konkrétně celkovou délku trasy, čas po který byla absolvována, čas dosažení startu a cíle, průměrnou a maximální rychlost. Trasa vzniká absolvováním celé trati je uživatelům prezentována jako rekord a to včetně všech uvedených atributů, na jejichž základě se mohou závodníci vzájemně porovnat.

Závodník

Závodník je reprezentace uživatele v systému. Je s ním spojena konkrétní osoba využívající aplikaci. Závodníci mohou vytvářet tratě a pak je absolvovat. Počet absolvování není omezen. Každý závodník má svoji přezdívku a heslo, které je vyžadováno při každé operaci s daty na serveru (vytvoření tratě, nahrání rekordu atd.).

3.2 Návrh uživatelského rozhraní

Cílem této sekce je vytvořit návrh uživatelského rozhraní webové aplikace. Při jejím studiu jsem vycházel z velmi kvalitně zpracované publikace od společnosti Apple[3], která se tomuto tématu podrobně věnuje. iPhone používá inovativní model interakce uživatele s přístrojem a poskytuje tak velkou výzvu pro vytvoření přehledného a efektivního uživatelského rozhraní. To vyžaduje dokonalé pochopení principu dotykového ovládání a způsobu jak k němu uživatelé přistupují.

Specifika uživatelského rozhraní iPhone

Vzhledem k odlišnostem iPhone od ostatních přístrojů diskutovaných v kapitole 2.3.2 je evidentní, že uživatelé přistupují k jeho ovládání odlišně než k běžnému počítači nebo notebooku. Uvědomění této skutečnosti je důležitým prvním krokem k tvorbě dobrého uživatelského rozhraní. Mezi hlavní rozdíly patří dotykové ovládání a z toho plynoucí absence kurzoru, relativně malý displej nebo závislost na prohlížeči Safari. Tyto vlastnosti dávají velký prostor k experimentování, musím ovšem pamatovat na to, že uživatelé iPhone jsou zvyklí na jednoduché, přehledné a především rychlé uživatelské rozhraní. Zde je dobré vzít

si příklad z nativních aplikací, se kterými jsou uživatelé obeznámeni a s jejichž jednotlivými elementy umějí pracovat.



Obrázek 3.2: Ukázka nativní aplikace AppStore.

Povaha přístroje udává další specifikum: uživatel je při jeho používání mobilní. Bude aplikaci využívat kdekoliv se právě nachází, tedy nejen v klidu, ale třeba i při chůzi, jízdě autem nebo vlakem. Celý problém je ještě umocněn zaměřením mojí aplikace na zpracování GPS. Z tohoto faktu vyplývá následující závěr: uživatel může být při používání uživatelského rozhraní rozptýlen podněty z okolí a nemusí věnovat aplikaci svojí plnou pozornost. S tím musím počítat a snažit se vyvarovat obsahu, který vyžaduje velkou koncentraci uživatele. Například je nevhodné používat příliš malých tlačítek, uživatel se pak musí hodně soustředit, aby se na tlačítka trefil dotekem svého prstu. Takový problém se u běžného počítače neřeší, myš má nesrovnatelně vyšší přesnost než prst a obecně platí, že co uživatel vidí, na to je schopný kliknout. Na tento fakt je nutné při tvorbě uživatelského rozhraní pamatovat, mělo by být především rychlé a extrémně snadné na použití. Musí upoutat uživatelskou pozornost a rychle ho navést na získání požadovaných informací bez ohledu na to, co právě dělá.

Další zajímavá vlastnost související s prezentací webových stránek je takzvaný *průhled* (anglicky viewport). Průhled je obdélníková oblast, která určuje jak je obsah uspořádán a kde se zalamuje text. U prohlížeče na běžném počítači je průhled daný šířkou okna a uživatel se v něm pohybuje pomocí posuvníků. Na iPhone ovšem žádné posuvníky nejsou, pohyb po stránce je realizován pomocí zažitých dotekových gest (více se dozvíte v následujícím odstavci). Uživatelé mohou průhled zvětšovat, nebo zmenšovat, ale nemůžou měnit jeho velikost (s výjimkou natočení přístroje do horizontální polohy). Z těchto zjištění plyne nutnost správného nastavení průhledu. Protože moje aplikace je pro iPhone primárně určena, vypnul jsem možnost jeho zvětšování a nastavil výchozí poměr zobrazení na 1:1. Obsah tak nebude nijak zkreslený a bude budit dojem nativní aplikace, u kterých toto zvětšování neprobíhá. Na druhé straně je nutné zajistit, aby byl veškerý text dobře čitelný, když uživatel nebude mít možnost jeho zvětšení.

Jak jsem se již zmínil v předchozím odstavci, s uživatelským rozhraním na iPhone souvisí i zažitá gesta pro jeho ovládání. Jejich smyslem je nahradit akce, které nejsou prostým

dotekem realizovatelné (například funkce kolečka myši). V tomto ohledu je velmi zajímavá patentovaná technologie zvaná *multi-touch*, která dovoluje zpracovávat libovolné množství doteků v jeden okamžik. Předpokládám, že s těmito gesty jsou běžní uživatelé seznámeni, protože se vyskytují ve většině nativních aplikací a při ovládání přístroje je běžně používají. Pro úplnost nyní uvedu seznam těchto gest, aby i neznalý čtenář pochopil základy ovládání iPhone aplikací:

- **Klepnutí** (Tap) - Prostý krátký dotek displeje jedním prstem. Používá se pro aktivaci formulářových prvků a odkazů. Zde můžeme nalézt analogii s levým tlačítkem myši.
- **Dvojité klepnutí** (Double tap) - Rychlé dvojité ťuknutí na obrazovku by mělo způsobit rychlé přiblížení na vybranou část stránky. Vzhledem k zákazu zvětšení průhledu v mojí aplikaci toto gesto nefunguje, protože není potřeba.
- **Švihnutí** (Flick) - Rychlým pohybem jedním prstem po obrazovce dává uživatel najevo, že momentálně zobrazené informace ho nezajímají a chce se rychle přesunout na jinou část stránky. Tuto velice užitečnou funkci by měla moje aplikace podporovat.
- **Posunutí** (Drag) - Pomalý pohyb prstu se používá k jemnému posouvání po stránce. Jde o absolutní základ pro ovládání zobrazení stránky.
- **Roztažení** (Pinch open) - Dotyk dvou prstů a následné roztažení od sebe je učebnicovou ukázkou technologie *multi-touch*. Toto gesto je zařité pro přiblížení obsahu, změna vzdálenosti mezi prsty odpovídá míře zvětšení. Velice užitečné například pro ovládání měřítka mapy.
- **Stažení** (Pinch close) - Stejně jako předchozí bod jen pozpátku. Prsty se umístí daleko od sebe a poté se přibližují. Toto gesto invokes zmenšení obsahu.
- **Dlouhý dotek** (Touch and hold) - Dlouhé přidržení prstu na jednom místě způsobí zobrazení kontextové nabídky pro ovládání funkce copy/paste. Pro uživatele velice důležité gesto, které by měla moje aplikace bezpodmínečně podporovat. Vzhledem k ovládání myši je toto gesto analogií ke kliknutí pravým tlačítkem.
- **Posun dvěma prsty** (Two-finger scroll) - Toto gesto by mělo být analogií k funkci kolečka myši. Z vlastní zkušenosti vím, že ho většina uživatelů nezná. Pravděpodobně kvůli tomu, že pro pohyb na stránce je pohodlnější používat gesta Flick a Drag.

Další důležitou vlastnost, kterou musí uživatelské rozhraní mojí aplikace splňovat je flexibilita. A to nejen v rámci zobrazení na iPhone, ale také na ostatních platformách. Je důležité, aby aplikace působila kompaktním dojmem a byla bez problému použitelná na malých i velkých rozlišeních. U těch malých by se měl vzhled aplikace upravit, aby navigační prvky nezabírali příliš velkou plochu na úkor obsahu.

Navigace v uživatelském rozhraní

Po seznámení se specifickými vlastnostmi uživatelského rozhraní na iPhone se nyní budu věnovat návrhu navigace v rámci aplikace. Správný přístup k tomuto problému je stěžejní pro vytvoření kvalitního uživatelského rozhraní. Je nutné maximálně omezit přesun mezi jednotlivými obrazovkami aplikace, aby byl minimalizován počet nutných uživatelských vstupů a tím i čas, který potřebný pro zjištění požadovaných informací. Jako nejlepší řešení



Obrázek 3.3: Ukázka flexibilního uživatelského rozložení mojí aplikace.

se mi tak jeví využití již zavedených schémat navigace běžně používaných v nativních aplikacích. Obrázek 3.4 znázorňuje způsob navigace v rámci nastavení telefonu.



Obrázek 3.4: Typické zpracování navigace iPhone aplikace.

V tomto schématu je stránka rozdělena do tří částí: horní panel, obsah a spodní panel. Horní panel slouží k identifikaci obrazovky, na které se uživatel nachází, navigaci v historii, a také může obsahovat tlačítka související s aktuální obrazovkou. Nadpis by měl být výstižný, ideálně jednoslovný. Přesun na předchozí obrazovku je aktivován pomocí tlačítka umístěného v levé části horního panelu, jeho název je shodný s nadpisem obrazovky, na kterou se po jeho stisknutí vrátíme. V pravé části je místo pro jedno tlačítko, obvyklé je například přidání položky do vypisovaného seznamu, nebo zobrazení nastavení. Spodní panel slouží pro rychlé přepínání mezi sekcemi aplikace. Je realizován pomocí ikon s krátkým popisem. Vzhledem k velikosti displeje v orientaci na výšku nelze do tohoto panelu umístit více jak 4-5 položek.

Prezentace informací

Kromě navigace v aplikaci je hlavním cílem uživatelského rozhraní prezentace informací. Ty musejí být podány strukturovaně a přehledně, aby uživatel mohl rychle najít co potřebuje. Je třeba použít vhodný font s dostatečnou velikostí, Apple doporučuje font Helvetica o velikosti 17 až 22 pixelů. Kromě něj podporuje Safari i poměrně velké množství dalších fontů, které jsou ale většinou vhodné pouze pro krátké úseky textu (například imitace ručního písma Zapfino). Důležitý je také kontrast mezi barvou pozadí a barvou textu. Z hlediska strukturovanosti je doporučeno používání seznamů a zvýraznění textu pomocí tučného písma a kurzívy. Apple nedoporučuje používání dekorace podtržením, protože text potom prý vypadá přeplněně. Všechny tyto zásady se budu snažit ve své aplikaci dodržet a dosáhnout tak efektivního a přehledného uživatelského rozhraní.

Kapitola 4

Implementace aplikace

V následující kapitole bych rád seznámil čtenáře se způsobem, jakým je aplikace implementovaná. Popíši použité knihovny třetích stran a zdůvodním jejich použití. Dále se zaměřím na zajímavé problémy, na které jsem v průběhu implementace narazil, a nastíním způsob jejich řešení.

4.1 jQuery

jQuery[10] je JavaScriptová knihovna, která klade důraz na interakci mezi JavaScriptem a HTML. Její filozofií je oddělení skriptů od struktury HTML na podobném principu jako CSS. Například místo přímé specifikace on-click události na element přímo v HTML kódu je použit selektor, který tento element vybere a poté změní jeho manipulátor zmíněné události. Toho se dosahuje pomocí funkce \$, což je v podstatě namespace pro všechny operace nabízené knihovnou. Většinou se této funkci předává jeden parametr typu řetězec reprezentující selektor a poté se může volat libovolné množství metod. jQuery tento selektor zpracuje, vybere odpovídající elementy z DOM a aplikuje na ně požadované operace.

V mojí aplikaci je knihovna jQuery použita pro zjednodušení zápisu JavaScriptového kódu. Konkrétně pro nastavování reakcí na události nebo změnu tříd jednotlivých elementů. Využil jsem také funkce pro práci s technologií AJAX, které tato knihovna nabízí.

4.2 Google Maps API v3

Google Maps je JavaScriptová knihovna, kterou jsem ve své aplikaci použil pro zobrazování mapy. Hlavním důvodem pro její implementaci byl fakt, že pro nekomerční využití je dostupná zcela zdarma. Zároveň k ní existuje velice přehledná referenční příručka[7], ze které jsem čerpal. V mojí aplikaci je implementovaná třetí verze této knihovny, mezi jejíž hlavní přednosti patří optimalizace pro mobilní zařízení. Poskytuje velké množství funkcí, které usnadňují adaptaci mapy pro potřebné účely. Nejvíce jsem využil funkce pro práci s ukazateli (markers), lomenými čarami a kruhy. Na druhé straně během implementace jsem narazil na problém s informačními okny poskytované knihovnou, které jsou na iPhone téměř nepoužitelné z důvodu jejich velmi dlouhého načítání. Také celkový výsledný výkon při zobrazování mapy mohl být lepší, velké problémy nastávají, pokud se má zobrazit velké množství bodů, aplikace má pak značně velkou odezvu.

4.3 Navigator Geolocation

Základním stavebním kamenem celého projektu je podpora geolokace neboli zjišťování aktuální pozice na zemském povrchu. Touto problematikou se zabývá specifikace [15] konsorcia W3C. Definuje objekt `navigator.geolocation`, pomocí kterého může skript přistupovat k datům z GPS lokátoru. Využívá k tomu dvě funkce, `getCurrentPosition` pro jednorázové zjištění aktuální pozice a `watchPosition` pro odstartování sledování pozice. V projektu využívám druhé jmenované, je však plně podporovaná jen u zařízení s GPS jednotkou, tedy u iPhone a Android.

4.4 Offline aplikace

Jedno z W3C doporučení [18] obsahuje specifikaci pro podporu takzvaných offline aplikací. Jde v podstatě o to, donutit prohlížeč uložit vybrané části prezentace (xhtml soubory, soubory s kaskádovými styl, obrázky, skripty...) do stálé paměti klienta, odkud jsou při dalším požadavku načteny bez nutnosti připojování k internetu. Tímto způsobem je možné vytvořit aplikaci, která bude vyžadovat pouze prvotní stažení všech jejích částí a její následné používání bude probíhat bez dalšího načítání dat ze serveru. Přesně to ve svém projektu potřebuji. Uživatel pak může aplikaci používat i v místech, kde nemá žádný signál pro připojení k internetu. Zároveň se bude aplikace podstatně rychleji spouštět (s výjimkou prvního stahování) a výrazně se zmenší množství přenesených dat. Konkrétně je tento mechanismus realizován uvedením atributu `manifest` v kořenovém elementu `html`. Jeho obsahem je cesta k souboru, ve kterém jsou vypsány všechny soubory, které si má prohlížeč uložit.

4.5 Implementace uživatelského rozhraní

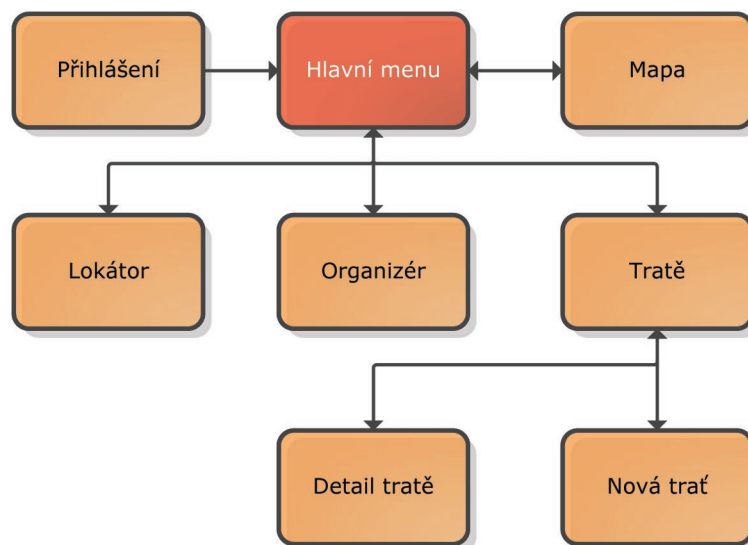
V této sekci bych rád čtenáře seznámil s konečnou podobou uživatelského rozhraní a s prostředky v něm využitými. Při jeho studiu jsem vycházel především z návrhu, který diskutuji v sekci 3.2.

Výsledná podoba uživatelského rozhraní

Na základě znalostí ze sekcí 3.1 a 3.2 jsem vytvořil strukturu uživatelského rozhraní mé aplikace. Rozdělil jsem jí do několika sekcí, jak je vidět na následujícím obrázku:

Mapa Hlavním obsahem této sekce je instance knihovny Google Maps API. Lze na ní zobrazit probíhající závod, náhled trati nebo lze použít i pro návrh nové trati. Ovládání mapy probíhá jednak pomocí standardních gest (diskutovaných v sekci 3.2) a také pomocí vysouvací nabídky, kde uživatel nalezne pokročilé nastavení jako je vycentrování mapy nebo volba mapového podkladu.

Lokátor Účelem této sekce je prezentace informací získaných z GPS lokátoru. Uživatel zde může zjistit svou poslední známou polohu a rychlost jakou se pohyboval. Nalezneme zde i ovládání GPS jednotky včetně volby obnovovací frekvence. Vzhledem k povaze sekce je přístupná pouze na přístrojích, které podporují sledování GPS.



Obrázek 4.1: Struktura uživatelského rozhraní mojí aplikace.

Organizér V této sekci uživatel nalezne všechny potřebné nástroje pro nastavení závodu a zároveň získá i informace o jeho aktuálním výkonu, jako je průměrná rychlost nebo uražená vzdálenost.

Tratě Tato poslední sekce obstarává napojení na server. Nalezneme zde výpis veřejných tratí, včetně jednoduchého vyhledávání. U jednotlivých tratí si uživatel může zobrazit dosažené rekordy, může si zobrazit náhled jednotlivých tratí nebo si je stáhnout do lokální paměti a odtud je poté pomocí organizéru absolvovat. Nachází se zde také tlačítko pro vytvoření nové tratě.

Prostředky použité při implementaci

Při studiu této části mi byla velice nápomocná příručka od společnosti Apple zabývající se vizuálními efekty prohlížeče Safari[4] a také dokumentace knihovny jQuery[10].

Celé uživatelské rozhraní je implementováno pomocí technologií HTML, CSS a JavaScript popsanych v sekci 2.4. HTML zapouzdřuje obsahovou část, jako jsou jednotlivé popisky tlačítek a texty. CSS je použit pro popis vzhledu obsahové části a to i včetně animací. Odezvu na uživatelské podněty obstarává JavaScript pomocí odchyty událostí. Reakce na tyto události je poté závislá jednak na obsahu dokumentu (pozice v DOM a třída elementu) a také na stavu vnitřních proměnných. Pro zjednodušení celého tohoto procesu je použita knihovna jQuery, která exceluje především ve zjednodušení a tím pádem i zpřehledněním zápisu algoritmu. Tato knihovna nabízí i funkce pro animaci, které jsem ale nevyužil. Důvodem byl především jejich slabý výkon na mobilních zařízeních, a pak také přítomnost mnohem elegantnějších řešení, které nabízí prohlížeč Safari, respektive jeho jádro Webkit. Tyto možnosti se nyní pokusím v krátkosti shrnout.

Animace a přechody Jedny z nejzajímavějších novinek připravovaných v CSS3. Přináší revoluci v zobrazování webových stránek, díky nim lze animovat většinu CSS vlastností. K tomu se dosud využíval pouze JavaScript, který ale k tomuto účelu není primárně

určený. Navíc animace založené na JavaScriptu jsou na iPhone zoufale pomalé a nepoužitelné, vzhledem k jejich náročnosti na výpočetní výkon. Proto budu veškeré animace provádět pomocí CSS transformací, u kterých je zaručena plynulost i na méně výkonných zařízeních, jako je iPhone.

Gradients Gradient je vizuální efekt, který se aplikuje na povrch a způsobuje vznik barevného přechodu mezi definovanými barvami. Tuto vlastnost lze nastavit libovolnému elementu. Při jeho implementaci není potřeba jakýchkoliv obrázků, postačuje prostý výčet barev a jejich pozic. Barevný přechod pak dopočítá a zobrazí sám prohlížeč. K dispozici jsou přechody lineární a radiální. Přechody lze zobrazit nejen na pozadí, ale také v okrajích a odrážkách seznamů. Hlavní výhodou tohoto řešení spočívá v odstranění potřeby realizovat barevné přechody pomocí obrázků a tím snížit množství přenášených dat.

Border radius Dlouho očekávaná vlastnost, která umožňuje jednoduchou tvorbu zakulacených rohů, které se v CSS2 musely řešit pomocí obrázků. Nyní to již není potřeba, zpřehlední se tak CSS kód a také se nemusí používat žádné obrázky.

Border image Velice užitečná vlastnost. Dovoluje jeden obrázek rozdělit na 9 oblastí (4 rohy, 4 strany a střed), které jsou poté použity jako roztahovací pozadí libovolného elementu. Tato vlastnost je ideální pro použití u různých flexibilních tlačítek, které nelze jednoduše realizovat pomocí obyčejných okrajů nebo pozadí.

Průhlednost Jedna z nejpodporovanějších CSS3 vlastností. V dnešní době ji implementuje již většina prohlížečů, Webkit v tomto ohledu nemůže být pozadu. Umožňuje nastavit procentuální průhlednost libovolného elementu. Ve spojení s animacemi je tak možné provést například postupné objevení, nebo zmizení libovolného elementu uživatelského rozhraní.

Stínování Další skupina vlastností podporuje vytváření stínů, ať už u textu i u celých bloků. Z estetického hlediska jde o velmi povedenou vlastnost, která se opět snaží vyhnout zbytečnému používání obrázků.

Kapitola 5

Závěr

Cílem této bakalářské práce bylo seznámit čtenáře s platformou Apple iPhone a analyzovat možnosti, jež nabízí. Na základě těchto znalostí byla poté vytvořena aplikace se zaměřením na zpracování GPS dat. Klientská část byla poté rozšířena o serverové pozadí, které funguje jako globální úložiště dat.

Téma této práce zasahuje do mnoha oblastí, které bylo nutné obsáhnout. Musel jsem nastudovat značné množství literárních pramenů z různých oblastí. Bylo třeba seznámit se s podstatou systému GPS a se souřadným systémem WGS84. Asi nejvíce času mi zabrala problematika tvorby webových aplikací a následná aplikace těchto znalostí do specifických podmínek iPhone. Další studovanou oblastí byla tvorba a správa databázových systémů.

Výsledkem mého snažení je funkční webová aplikace splňující podmínky zadání. Má zároveň atraktivní a přehledné uživatelské rozhraní, které je jednoduše použitelné. Mezi další přednosti mé aplikace patří šetrné využití datového přenosového pásma, schopnost funkčnosti i bez stálého připojení k síti internet nebo alespoň částečná podpora i jiných platform než iPhone.

V budoucnu by bylo dobré aplikaci rozšířit o další funkce jako například větší interakce mezi závodníky, třeba formou hromadných závodů, popřípadě obohacené o možnost sociální interakce v podobě chatu. Další oblastí, kterou by bylo dobré propracovat je detailnější podpora ostatních mobilních zařízení, ať už třeba iPad od společnosti Apple nebo i přístroje s operačním systémem Android a prohlížečem na jádře Webkit.

Literatura

- [1] Apple Inc.: Apple - iPhone - Mobile phone, iPod, and Internet device. [online], rev. 3. května 2010 [cit. 03-04-2010].
URL <http://www.apple.com/iphone/>
- [2] Apple Inc.: iPhone Dev Center. [online], rev. 31. března 2010 [cit. 31-03-2010].
URL <http://developer.apple.com/iphone/>
- [3] Apple Inc.: iPhone Human Interface Guidelines for Web Applications. [online], rev. 6. ledna 2010 [cit. 04-04-2010].
URL <http://developer.apple.com/safari/library/documentation/InternetWeb/Conceptual/iPhoneWebAppHIG/iPhoneWebAppHIG.pdf>
- [4] Apple Inc.: Safari CSS Visual Effects Guide. [online], rev. 24. února 2010 [cit. 15-04-2010].
URL <http://developer.apple.com/safari/library/documentation/InternetWeb/Conceptual/SafariVisualEffectsProgGuide/SafariVisualEffectsProgGuide.pdf>
- [5] Apple Inc.: Safari Dev Center. [online], rev. 31. března 2010 [cit. 31-03-2010].
URL <http://developer.apple.com/safari/>
- [6] Fabián, O.: *Internetová aplikace na mobilním zařízení iPhone*. Diplomová práce, FIT VUT v Brně, 2009.
- [7] Google: Google Maps API V3 Reference. [online], rev. 5. dubna 2010 [cit. 05-04-2010].
URL <http://code.google.com/intl/cs/apis/maps/documentation/v3/reference.html>
- [8] Ilčík, O.: *Jednoduchá navigace v rastrové mapě pomocí GPS v PDA*. Diplomová práce, FIT VUT v Brně, 2008.
- [9] Imagery, N.; Agency, M.: Department of Defense World Geodetic System 1984. [online], rev. 3. ledna 2000 [cit. 04-04-2010].
URL <http://earth-info.nga.mil/GandG/publications/tr8350.2/wgs84fin.pdf>
- [10] jQuery Team: jQuery. [online], rev. 5. dubna 2010 [cit. 05-04-2010].
URL <http://jquery.com/>
- [11] Kuchta, M.: *Podpora Geocachingu pro kapesní počítače s GPS*. Diplomová práce, FIT VUT v Brně, 2009.

- [12] Webkit Development Team: The Webkit Open Source Project. [online], rev. 15. dubna 2010 [cit. 15-04-2010].
URL <http://webkit.org/>
- [13] World Wide Web Consortium: Cascading Style Sheets. [online], rev. 18. března 2010 [cit. 15-04-2010].
URL <http://www.w3.org/Style/CSS/>
- [14] World Wide Web Consortium: The Extensible HyperText Markup Language (Second Edition). [online], rev. 1. srpna 2002 [cit. 15-04-2010].
URL <http://www.w3.org/TR/xhtml1/>
- [15] World Wide Web Consortium: Geolocation API Specification. [online], rev. 7. července 2009 [cit. 15-04-2010].
URL <http://www.w3.org/TR/geolocation-API/>
- [16] World Wide Web Consortium: HyperText Markup Language 4. [online], rev. 24. prosince 1999 [cit. 15-04-2010].
URL <http://www.w3.org/TR/html4/>
- [17] World Wide Web Consortium: HyperText Markup Language 5. [online], rev. 4. dubna 2010 [cit. 15-04-2010].
URL <http://www.w3.org/TR/html5/>
- [18] World Wide Web Consortium: Offline Web Applications. [online], rev. 30. května 2008 [cit. 15-04-2010].
URL <http://www.w3.org/TR/offline-webapps/>