

Zápis z jednání / Meeting Minutes: [MN002 v0.2 Schuzka 20080226.doc](#)

Datum konání / Date:	26.2.2008 18:45	Místo konání / Place:	MS – Rodunda
Přítomní / Participants:	David Stach, Martin Kontsek, Ondřej Dušek, Rastislav Wartiak, Son Tung Nguyen		
Téma jednání / Topic:	Pravidelná schůzka		

Deleted: MN002 v0.1 Schuzka 20080226.doc

Obecné informace a rozhodnutí

Byl založen projekt na SourceForge. Je k dispozici SVN, Tracker, webové diskusní fórum, úložiště dokumentů a webová stránka. Webová stránka je docela pomalá, pravděpodobně využijeme jinou. Rozhodnutí ohledně SVN prozatím nepadlo.

Pro výstup z projektu bude použita licence GPL. Umožní nám využít knihovny a funkce vytvořené v rámci GPL a LGPL.

Deleted: ¶

Formatted: Heading 3

Zjištění pro jednotlivé části mobilní aplikace

Data transfer application

Zajímavou možností pro práci s Bluetooth v rámci desktopové aplikace se jeví využití BlueCove (<http://code.google.com/p/bluecove/>). Jedná se o Java knihovnu, která poskytuje JSR 82 rozhraní na Windows, Linuxu i MacOS X. Díky tomu by bylo možné využít stejný kód pro práci s Bluetooth v mobilní i desktopové aplikaci. Praktický test ukázal bezproblémový provoz na Windows, ale problémy s využitím na Linuxu. Je ale možné očekávat, že vývojový tým tyto problémy dořeší. Aplikace bude pracovat s jednoduchým GUI.

Deleted: ¶

Formatted: Heading 3

Deleted: :

Communication with the desktop

Pro Bluetooth komunikace bude použito rozhraní JSR 82. Při komunikaci mobilní a desktopové aplikace bude mobil klientem a desktop serverem. Při komunikaci mezi dvěma mobilními aplikacemi bude nutné u každé zvolit režim, pokud možno ale v uživatelské terminologii, např. Přijmout pro server a Odeslat pro klienta.

Modul bude nabízet funkce, které budou volány z menu aplikace, dále se bude muset postarat o navázání spojení s desktopovou aplikací a přenos dat. Očekává se tedy i implementace částí uživatelského rozhraní. Po ukončení přenosu modul zobrazení dat aktualizuje zobrazení dat. Pro práci (čtení i zápis) s uloženými mapami, trasami a dalšími objekty bude využívat třídy rozhraní z modulů uložení dat, pro odstínění od způsobu uložení těchto dat.

Communication with the server

Pro komunikaci bude využita třída StreamConnection. Prozatím máme jenom zkušenosti s využitím pro HTTP komunikaci, MIDP 2.0 ale podporuje i TCP komunikaci (prefix socket://). Spojení se serverem bude vždy navazovat mobilní aplikace. V rámci konfigurace bude možné určit jak často má být pozice odesílána. V rámci potvrzení přijatých dat bude server k odpovědi přikládat i data, ke kterým je mobilní aplikace přihlášená, tj. informace o poloze sledovaných objektů. Zároveň bude možné v odpovědi zaslat i administrativní příkazy, např. pozastavení odesílání polohy u aplikací, které sledují polohu vozidla. Modul bude využívat rozhraní GPS třídy a zároveň bude nabízet pro modul zobrazování registraci listeneru, kterým bude oznamovat změnu sledovaných objektů (waypointů).

Vector map/Track storage

Mapy budou uloženy přímo v JAR souboru, který tak bude muset být vytvořen v aplikaci pro přípravu dat nebo je mobilní aplikace přijme z aplikace pro přenos dat a uloží pomocí třídy RecordStore. Uložení do souborového systému pomocí JSR 75 se jeví jako problémové, proto s ním prozatím nebudeme počítat. Práce s RecordStore se jeví jako rychlejší, je ale nutné zjistit omezení pro velikost ukládaných dat.

GPS communication

Mobilní aplikace bude obsahovat třídu pro práci s GPS, která bude poskytovat jednotné rozhraní pro komunikaci s GPS moduly pomocí Location API (LAPI, JSR 179), Bluetooth nebo sériového portu. S LAPI se pracuje pomocí registrace listenerů, které systém volá při vybraných událostech – změna stavu modulu, polohy, výšky, atd. Pro práci se sériovým portem je možné využít StreamConnection s prefixem comm:. GPS modul následně v pravidelných intervalech (obvykle cca. 1 s) posílá informace o poloze. Aplikace data jenom pasivně přijímá. S Bluetooth se po nalezení zařízení a spojením s ním otevře komunikační port a komunikace následně probíhá stejně. Naše třída bude pracovat obdobně jako LAPI, tedy bude od GPS modulu přijímat zprávy a aplikace si bude moci zaregistrovat funkci, která bude zavolána při výskytu definované události. Zároveň ale bude možné kdykoliv zjistit aktuální stav všech hodnot voláním definovaných metod.

Třída bude také poskytovat pro aplikaci simulátor GPS. I bez připojeného modulu tedy bude možné aplikaci testovat. Simulátor bude generovat jenom jednoduchý pohyb, např. pohyb po kružnici s definovaným středem, poloměrem a rychlostí.

User interface

Všechny texty budou uloženy tak, aby bylo možné aplikaci jednoduše přeložit do jiného jazyka. Jako minimum se počítá s angličtinou a češtinou. Pro uložení se pravděpodobně použije pole řetězců a pojmenované konstanty jednotlivých indexů.

SN v rámci jiné aplikace vytvořil graficky zajímavé menu, které můžeme v aplikaci použít. Pro některé nabídky bude využito i standardní menu.

Pokud budou některé moduly vyžadovat konfiguraci uživatelem, budou v rámci svých rozhraní nabízet funkce, kterými bude možné dané parametry zjistit a/nebo změnit. Zároveň musí v rámci analýzy definovat seznam a typ těchto parametrů. V rámci implementace bude vytvořeno jednotné uživatelské rozhraní pro konfiguraci.

Vývoj

Všechny funkce budou okomentovány, Doxygen musí pro všechny funkce vygenerovat smysluplnou dokumentaci.

Základní návrh kódovacího standardu:

- názvy tříd začínají velkým písmenem
- názvy proměnných začínají malým písmenem
- [názvy tříd i proměnných stylem kazdeDalsiSlovoVelkymPismenem](#)
- [konstanty jenom z velkých písmen, mezery pomocí podtržítka](#)
- [kódování UTF-8](#)
- [formátování kódu bude nutné definovat pravidly, např. definice v NetBeans](#)

Deleted: ¶

Formatted: Heading 3

Formatted: Bullets and Numbering

Návrh pro analýzy vytvořené v rámci projektu

Na začátku by měl být slovní popis funkce a rozhraní aplikace/modulu. Na základě toho vznikne návrh logiky rozhraní, např. která strana bude co volat. Následovat by měl popis jednotlivých tříd rozhraní, včetně potřebných parametrů.

Funkce jednotlivých aplikací a jejich modulů by měly být popsány pomocí požadavků na ně. Požadavky jsou slovní popisy jednotlivých funkcí, např. „Mobilní aplikace při prvním spuštění umožní uživateli vybrat jazyk, ve kterém bude komunikovat“. Na základě těchto požadavků bude vytvořen návrh rozhraní (kód, data, uživatelské) a také bude možné na základě těchto požadavků jednotlivé části testovat.

Upravené rozdělení prací na analýze mobilní části aplikace

Úkol z plánu	Odpovědný
Vector map storage	David Stach, Son Tung Nguyen
Vector map display	David Stach
POI storage and search	Ondřej Dušek
GPS communication	Rastislav Wartiak, Son Tung Nguyen
Communication with the server	Ondřej Dušek
Waypoints	David Stach
Communication with the desktop	Martin Kontsek
Track storage	David Stach, Son Tung Nguyen
External data display	David Stach
Extra display functionality	David Stach

Upravený harmonogram projektu: PL003.

Doplňeny úkoly:

- [Configuration interface \(Mobile application\) – DS, NS](#)
- [Simulator \(Mobile application / GPS communication\) – NS, RW](#)

[Změněny délky trvání některých úkolů z 7,5 dne na 8 dnů. Úkoly pro implementaci komunikace mobilní aplikace se serverovou prodlouženy na 12 dnů, na žádost OD. Beze změny termínu dokončení projektu.](#)

[U některých úkolů v mobilní aplikaci doplněn NS jako další řešitel.](#)

Termín další schůzky: 4.3.2008 19:00

Úkoly				
Pořadí ID	Popis úkolu Description	Řeší Responsible	Termín Date	Poznámka Remark
002/1	Provéřit možnosti RecordStore – omezení na velikost ukládaných dat	OD	4.3.2008	
002/2	Připravit první návrh požadavků a slovní popis funkcí rozhraní pro jednotlivé části mobilní aplikace. Vytvořené popis zaslat ostatním.	dle rozdělení prací	2.3.2008	
002/3	Zaslat kód vytvořené aplikace/aplikací ostatním	NS	28.2.2008	

001/1	Podívat se na rozdělení prací v plánu projektu, zaslat návrhy na doplnění/úpravu	všichni	trvale	
001/2	Začít s přípravou podkladů pro analýzu mobilní části	dle rozdělení úkolů z plánu	26.2.2008	