

Bakalářská práce



**České
vysoké
učení technické
v Praze**

F3

**Fakulta elektrotechnická
Katedra počítačové grafiky a interakce**

Kontextově závislá adaptivní struktura aplikace

Pratashchyk Dzmitry

Leden 2018

Vedoucí práce: Ing. Jiří Šebek

Obsah

1.	Úvod	1
1.1	Motivace.....	1
1.2	Cíle	1
2.	Rešerše.....	2
2.1	Uživatelské rozhraní	2
2.2	Adaptivní uživatelské rozhraní	2
2.3	Uživatelský context.....	3
2.4	Senzory a technologie Androidu.....	5
2.5	Fyzické sensory.....	7
2.5.1	Senzor okolního osvětlení.....	7
2.5.2	Akcelerometr	7
2.5.3	Mikrofon	9
2.5.4	Gyroskop.....	11
2.5.5	Barometr	12
2.5.6	Magnetometr.....	13
2.5.7	GPS	14
2.6	Technologie které používá Android.....	15
2.6.1	Bluetooth.....	15
2.6.2	Wi-fi.....	16
2.6.3	NFC.....	17
3.	Struktura aplikace pro systém Android	19
3.1	Architektura platformy Android	19
3.2	Struktura Android aplikace	21
3.3	Aktivity	22
4.	Související práce	26
5.	Analýza a implementace framework	27
5.1	Použitý framework.....	27

5.2	Package diagram	28
5.3	Případy užití	29
5.3.1	Use-case adaptace menu na základě hlasového kontextu.....	29
5.3.2	Use-case adaptace menu na základě kontextu vedlejších.....	33
5.3.3	Use-case adaptace uživatelského rozhraní na základě polohy zařízení....	35
5.3.4	Use-case adaptace uživatelského rozhraní pomocí NFC-tagů.....	37
6.	Testování.....	40
6.1	Úvod.....	40
6.2	Popis algoritmu testování.....	40
6.3	Nulová hypotéza	41
6.4	Cílová skupina.....	41
6.5	Množina otázek	41
6.6	Množina odpovědí.....	42
6.7	Nasbíraná data.....	42
6.8	Vyhodnocení získaných dat	43
7.	Instalace	45
8.	Závěr	46
9.	Literatura.....	47
10.	Obsah priloženého CD	51

Poděkování

Chtěl bych poděkovat Ing. Jiřímu Šebkovi za cenné rady, věcné připomínky a vstřícnost při konzultacích pro vypracování bakalářské práce. Chtěl bych poděkovat Ing. Ludíku Tesařovi, Ph.D. za cenné rady a odborný dohled při psaní v českém jazyce. Také bych chtěl poděkovat Ing. Nikolaji Volkovovi za jeho zvědavost o chodu psaní této práce.

Prohlášení

Prohlašuji, že jsem předloženou práci vypracoval samostatně a že jsem uvedl veškeré použité informace zdroje v souladu s Metodickými pokyny o dodržování etických principů při přípravě vysokoškolských závěrečných prací.

V Praze dne 09.12.2017

Abstrakt

Bakalářská práce má za úkol vylepšení existujícího nástroje pro adaptace uživatelského rozhraní na základě dat získaných se snímačů mobilního zařízení uživatele. Framework, na který navazuje tato práce je psaný s ohledem na modulární architekturu, proto poskytuje snadný a rychlý přístup ke kontextu senzorů mobilního zařízení. Tato bakalářská práce se zabývá rozšířením seznamu podporovaných snímačů mobilního zařízení, a jejich následným použitím při návrhu případu užití pro adaptace uživatelského rozhraní. Byl zachován postup navržení a čitelnost kódu frameworku, na které se vztahuje tato bakalářská práce

Klíčová slova: kontext, adaptace uživatelského rozhraní, snímače, senzory

Abstract

The bachelor thesis aims to improve the existing tool for adaptation of user interface based on the data obtained with the user's mobile device sensors. The framework, which this work builds, is written with respect to modular architecture and, therefore, allows easy and quick access to the context of mobile device sensors. This bachelor thesis focuses on the extension of the list of supported mobile device sensors and their subsequent use when designing a case of usage for adaptation of the user interface. The process of designing and legibility of the framework code covered by this bachelor thesis has been retained.

Key words: context, user interface adaptraion, sensors.

1. Úvod

1.1 Motivace

Za posledních několik let využití mobilních zařízení vzrůstá exponenciálně. Ke konci roku 2013 byl podíl uživatelů chytrých telefonů na světě 22%. Takže každý čtvrtý uživatel je majitelem chytrého mobilního telefonu. To je 18% vzrůst v porovnání s obdobím od 2009 až do 2013 [7].

S růstem výroby a použití mobilních zařízení se objevila nutnost usnadnění komunikace mezi člověkem a zařízením. Není překvapením, že jedním z nejrozšířenějších zařízení jsou mobilní telefony. Většina z nich jsou chytré telefony, které jsou vybavené poměrně velkým množstvím senzorů, které se dají využít pro už zmíněné usnadnění komunikace chytrého telefonu a uživatele. Toto usnadnění lze provést pomocí adaptací User interface na základě dat, které dodávají senzory telefonu.

Člověk očekává, že telefon mu bude nejen pomáhat v běžném životě, ale i rozumět tomu, co člověk právě v tu chvíli potřebuje, a nějak se bude tomu adaptovat.

S budoucím rozvojem hardwaru případně senzorů a algoritmů, pomocí kterých lze snadněji a přesněji získávat data, vzniká možnost lepšího využití pro vývoj adaptací uživatelského rozhraní.

1.2 Cíle

Cílem práce je vylepšit existující nástroj pro adaptace uživatelského rozhraní na základě dat ze senzorů a mikrofonu telefonu. Je potřeba se seznámit a nastudovat existující nástroje na platformě Android a tyto znalosti využít při práci. Vylepšení musí probíhat v rámci návrhu a testování situací užití tohoto nástroje. Tento framework musí být snadno modifikovatelný a rozšiřitelný.

2. Rešerše

2.1 Uživatelské rozhraní

Uživatelské rozhraní je nástroj, který zajišťuje komunikaci mezi člověkem a zařízením. Obsahuje v sobě sadu příkazů a pravidel, pomoci kterých se tato komunikace realizuje.

Uživatelské rozhraní je jedna z nejdůležitějších částí aplikace, protože určuje jak může uživatel dosáhnout přesně to co potřebuje [1]. Jedná se o jednu z mnoha složek human-computer interaction (interakce člověk - počítač).

Komunikace uživatele s uživatelským rozhraní musí být co nejjednodušší a pochopitelná pro uživatele. Programátor při plánování a implementaci aplikace se snaží udělat uživatelské rozhraní co nejobecnější, aby co nejvíce uživatelů mohlo pohodlně užívat aplikaci. Je třeba si uvědomit, že každý člověk má své preference. To nás vede k využití adaptivního uživatelského rozhraní. Adaptivní uživatelské rozhraní je flexibilní nástroj, který je schopný realizovat interakci s uživatelem v širším kontextu okolního prostředí. Podpora senzorů a snímačů slouží jako dobrý základ při adaptaci uživatelského rozhraní.

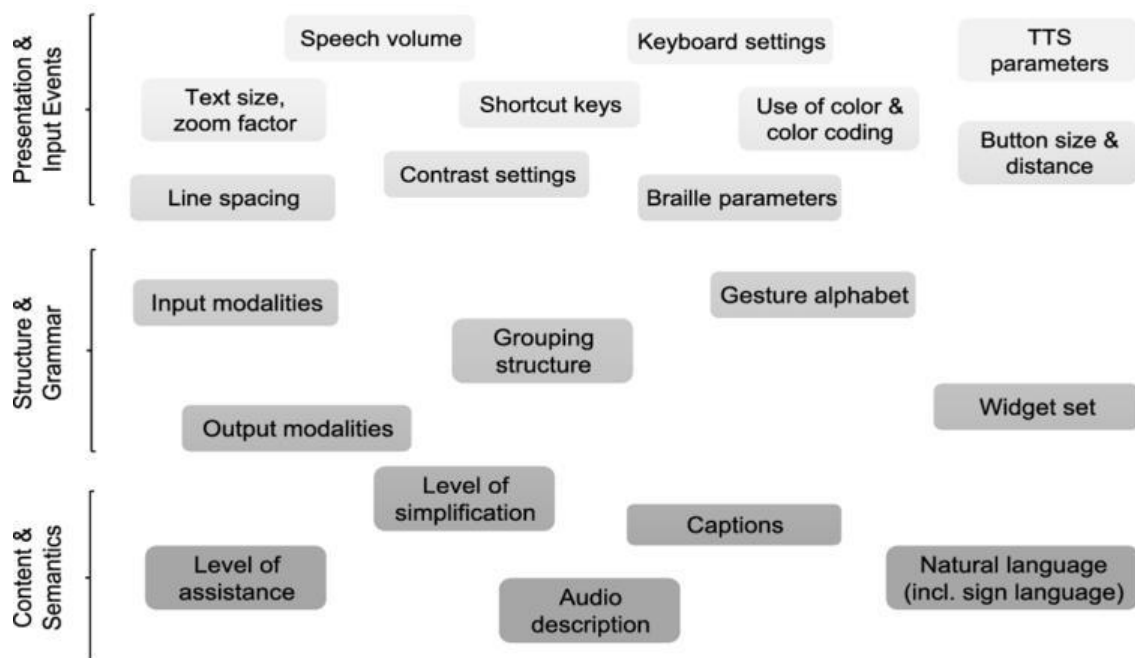
2.2 Adaptivní uživatelské rozhraní

Adaptivní uživatelské rozhraní je nástroj, který dovoluje adaptovat elementy uživatelského rozhraní podle potřeb a preferencí uživatele. Jedním z příkladů adaptivního uživatelského rozhraní může sloužit změna barvy pozadí aplikaci na základě času nebo volbě uživatele.



Obrázek 2.1. Adaptace UI

Adaptivní uživatelské rozhraní může ovlivnit jakýkoli element uživatelského rozhraní. A to od menších změn (rozměr textu, řádkování, kontrast, hlasitost) až k větším změnám (změna jazyku systému, audio nastavení atd.)

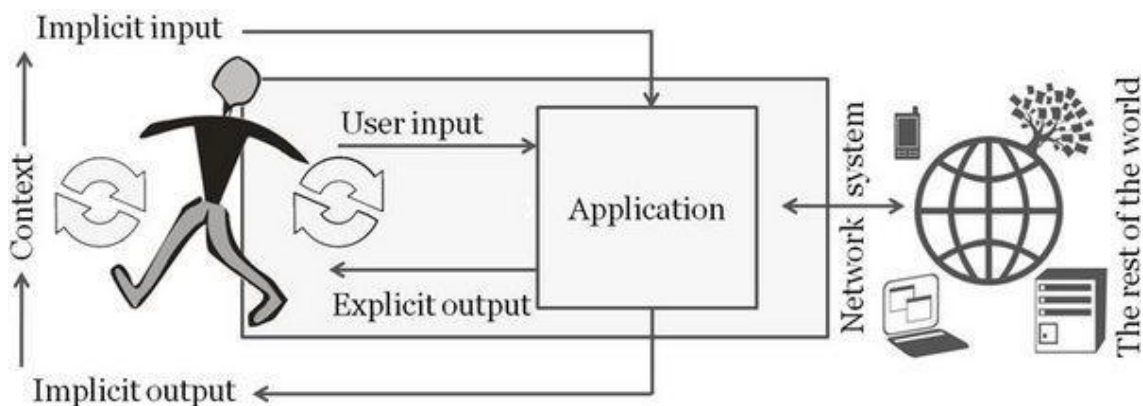


Obrázek 2.2. Některé aspekty uživatelského rozhraní

2.3 Uživatelský context

Uživatelský context je jakákoli relevantní informace která může být využita pro popis okolí, ve kterém se nachází uživatel. Důležité aspekty pro popis okolí jsou kupříkladu: kde se nachází uživatel, jestli se nachází v pohybu, jakého charakteru je tento pohyb, množství světla, které dopadá na zařízení uživatele atd. [5].

Sběr informací o chování uživatele je základem funkčnosti adaptivního uživatelského rozhraní a probíhá buď explicitně, což je přímý zásah uživatele nebo implicitně - sběr dat na pozadí bez účasti uživatele.

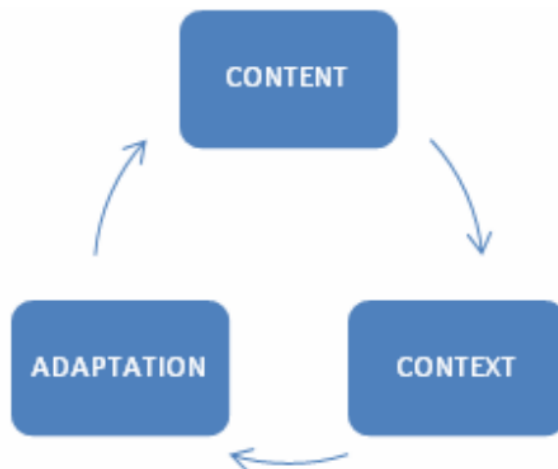


Obrázek 2.3. Explicitní a implicitní získání kontextu

Správně navržená aplikace, která podporuje adaptivní uživatelské rozhraní, musí umět uložit kontextová data, zpracovat je, a udělat potřebné změny uživatelského rozhraní. Cílem takové aplikace je přizpůsobit uživatelské rozhraní tak, aby uživatel dostával informace, které potřebuje s nejmenší interakcí s aplikací. Jinak řečeno, aplikace, která je schopna kontextově adaptovat uživatelské rozhraní, má za účel vytvořit co nejméně práce na straně uživatele při komunikaci s danou aplikací.

Adaptace může zahrnout nejen změny uživatelského rozhraní ale i změnit chování systému. Kontext ze kterým pracuje zařízení může být vícenásobný. To znamená, že pro adaptaci analyzuje nejen vlastnosti okolního prostředí, ale i stav systému a data z analýzy práce uživatele se zařízením.

Na obrázku 2.4 je vidět standardní cyklus pro adaptaci aplikace na základě kontextu [6].



Obrázek 2.4. Cyklus použití kontextu

Uživatel dostává obsah, který mu nabízí aplikace, následně během změny kontextu nebo jiných vnějších změn, které může vyvolat sám uživatel svým chováním, se adaptuje aplikace a následně se tyto změny provádí v obsahu, který se zobrazuje uživateli. Jak bylo řečeno, tento kontext se může skládat z více změn a způsobu chování uživatele, ale tento princip zůstává stejný. Při adaptaci probíhá analýza dat, které byly k dispozici při daném kontextu, a následně se zpracovává na základě algoritmu který určí programátor nástroje.

Jakákoli adaptace na základě kontextových dat musí přinášet snazší ovládní aplikace uživatelem a zrychlovat interakci, proto veškerá adaptace musí dávat smysl a být otestována. V opačném případě tato adaptace zmate uživatele a zvýší strávený uživatelem na potřebnou operaci.

V této práci se bude probírat adaptace uživatelského rozhraní na základě kontextu, který přináší senzory telefonu. Proto v další kapitole budeme probírat senzory pomocí kterých zjistíme kontext.

2.4 Senzory a technologie Androidu

Jak bylo zmíněno v předchozí kapitole získávání uživatelského kontextu a jeho následná analýza je zásadním procesem pro adaptaci uživatelského rozhraní.

Senzor je měřicí zařízení určené ke generování měřicího informačního signálu ve formě, která je vhodná pro přenos, další užití, zpracování a / nebo ukládání, ale kterou pozorovatel nemůže přímo vnímat [9].

Většina zařízení se systémem Android má vestavěné senzory, které měří pohyb, orientaci a různé podmínky prostředí. Tyto snímače jsou schopné poskytovat hrubá data s vysokou přesností.

Například hra může sledovat odečty z gravitačního senzoru zařízení, aby vyvodila složitá gesta a pohyby uživatelů, jako je náklon, otřesy, otáčení nebo houpání. Podobně aplikace počasí může používat snímač teploty a snímač vlhkosti pro výpočet a hlášení rosného bodu nebo aplikace pro navigaci může používat senzor geomagnetického pole a akcelerometr pro zobrazení světových stran pomocí kompasové růžice [11].

Pro získávání takového kontextu mohou sloužit senzory, kterými je vybaveno moderní “chytré” mobilní zařízení.

Data získaná ze senzorů, počet senzorů a jejich určení se může lišit v závislosti na výrobci chytrého telefonu. Sensory poskytují data o teplotě, poloze, osvětlení, úrovni šumu atd. Než začneme popisovat jednotlivé senzory musíme pochopit co je MEMS. MEMS je zkratka anglických slov "Micro-Electro-Mechanical Systems", v češtině Mikro-Elektricko-Mechanické-Systemy [15].

Zařízení MEMS jsou obvykle vyráběna na křemíkovém substrátu prostřednictvím různých výrobních technologií, podobně jako technologie mikročipů. Typické velikosti mikromechanických prvků leží v rozmezí od 1 mikrometru do 100 mikrometrů, zatímco rozměry čipu MEMS mikročipů mají rozměry od 20 mikrometrů do jednoho milimetru.

Jako materiály pro výrobu MEMS lze použít jak tradiční křemík, tak další materiály: například polymery, kovy a keramiku. Nejčastěji jsou mechanické systémy vyrobeny z křemíku. Jeho hlavními výhodami jsou fyzikální vlastnosti. Křemík je velmi spolehlivý. Může zpracovat miliardy operací aniž by došlo k poruše. Polymery jsou rovněž vhodným materiálem, protože může být vyráběn ve velkém množství, a co je nejdůležitější, s různými charakteristikami pro specifické úlohy. Naproti tomu kovy (zlato, měď, hliník) poskytují vysoké ukazatele spolehlivosti, i když mají horší fyzikální vlastnosti než křemík [17].

A právě na MEMS založena výroba veškerých snímačů, které se používají v mobilních zařízeních, a to z následujících důvodů [16]:

- Rozměry - jak bylo zmíněno takové snímače mají rozměry od 20 mikrometrů do jednoho milimetru, což je v současné době řešením rozmístění množství snímačů v rámci jednoho mobilního zařízení.
- Univerzálnost
- Spolehlivost;
- Malá spotřeba energie;
- Nízká cena.

Senzory můžeme rozdělit do třech skupin:

- Fyzické sensory. Tato kategorie senzorů označuje skutečná zařízení, která dokáží zachytit fyzikální vlastnosti okolního prostředí.
- Virtuální sensory. Tato kategorie senzorů nepředstavuje skutečná zařízení, ale dostává data analýzou a monitoringem dat aplikaci, operačního systému, a jiného softwaru. Příkladem funkčnosti takového senzoru je následující situace: uživatel má zakoupenou letenku na určitý den a čas, tak že systém může předpokládat stanoviště uživatele aniž by byl k dispozici fyzický senzor, který zodpovídá za zjištění polohy zařízení [10].
- Logické senzory se zpracovávají data z několika fyzických nebo virtuálních senzorů anebo jejich kombinaci.

2.5 Fyzické sensory

2.5.1 Senzor okolního osvětlení

Senzor okolního osvětlení je základní senzor, který se běžně používá skoro v každém moderním telefonu.

Používá se pro měření okolního světla a získaná data se například používají pro určení jasu displeje. Tím, že se sníží jas tak se sníží úroveň spotřeby energie. Na chytrých telefonech může displej využívat až 50 procent celkového výkonu telefonu, což má zásadní vliv na provozní dobu telefonu na baterii - což je v současné době velký problém pro všechny uživatele. Naopak, pokud je to potřeba, osvětlovací senzor musí navýšit úroveň jasu například při světlejším okolí, aby uživatel mohl pohodlně vidět informace, zobrazené na displeji.

2.5.2 Akcelerometr

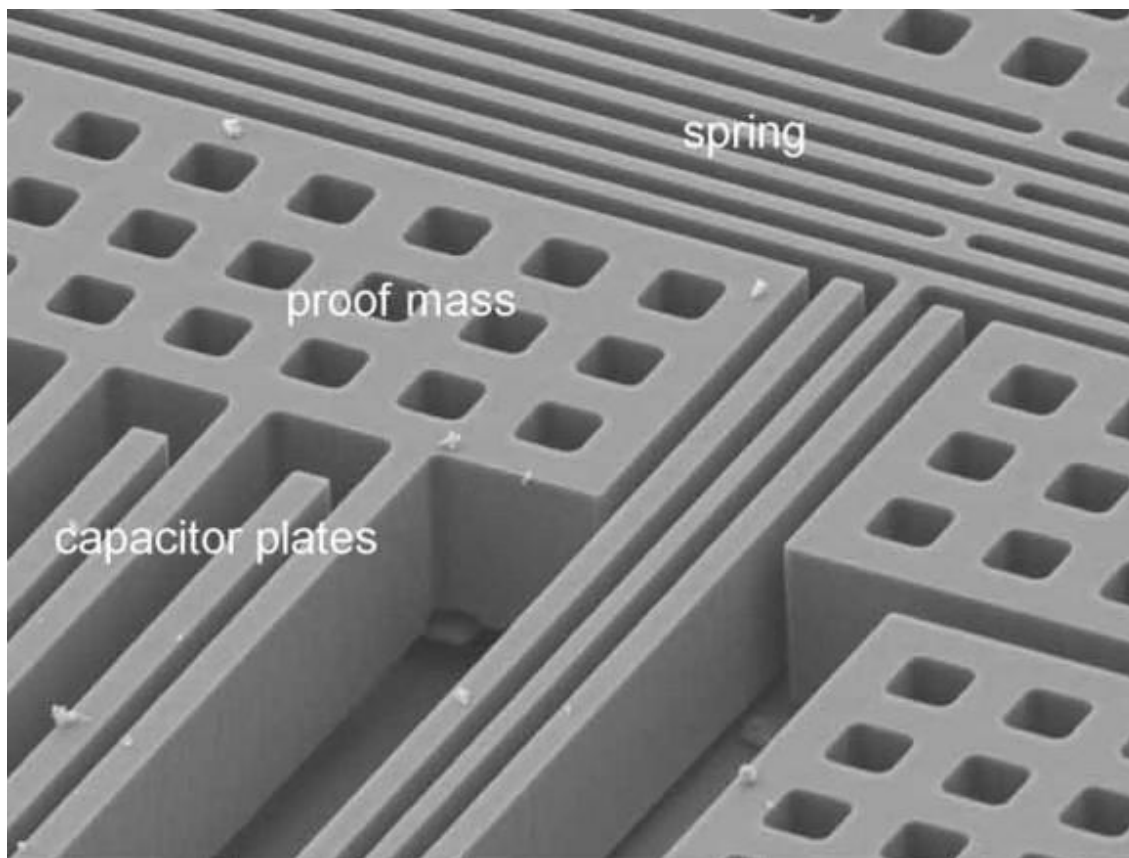
Akcelerometr je snímač, který umí měřit zrychlení, ale v chytrých telefonech dokáže zjistit změnu orientace a informovat o tom. Jedná se o téměř nejrozšířenějším MEMS zařízením. Přitom oblast jeho užití je velice široká. Například se používá v pevných discích za účelem detekce náhlé nepřítomnosti tíhového zrychlení (“zero g”).

Nepřítomnost tíhového zrychlení je ve většině případů způsoben pádem, takže se očekává pád spojený s impulsem poškozením mechaniky hlaviček disku a ploten disku spojeným s masivní ztrátou dat. V takovém případě disk provede příkaz na vrácení hlavy pevného disku do bezpečné polohy, která zabraňuje jak poškození zařízení, tak i ztrátě dat. Podobný princip se používá u zařízení pro registraci havárie ve vozidlech (t.j. kamery). Když vozidlo náhle zrychluje, případně náhle brzdí, nebo náhle mění jízdní pruh, je video označeno zvláštním příznakem, který ho chrání před vymazáním a přepsáním, což značně usnadňuje další analýzu silničních nehod. V porovnání s mobilními zařízeními, kde se akcelerometr používá přednostně pro zábavní účely, ve vozidlech se akcelerometr používá skoro pro každý systém bezpečnosti, což je systém airbagů, proti-blokovací brzdový systém (ABS), stabilizační systém, atd. Na rozdíl od toho v mobilních zařízeních se akcelerometr většinou používá pro přizpůsobení orientace. Například pokud se pozice mobilního zařízení mění z vertikální na vodorovnou, otočí se obraz tak aby uživatel mohl pohodlně pracovat se zařízením.

Funkce akcelerometru se rovněž často používá při výrobě aplikací nebo her, dovolí tak uživateli ovládat aplikace nakloněním zařízení.

Nyní se podíváme jak akcelerometr funguje. Práce akcelerometru může být založena na principu kondenzátoru. Pohyblivou částí takového systému je zátěž, která se pohybuje a mění kapacitu kondenzátorů v závislosti na náklonu a pohybu zařízení ve všech třech osách.

Při posunu se změní kapacita kondenzátorů, která se měří obvykle pomocí oscilátorů. Z těchto dat, je možné získat okamžité zrychlení.



Obrázek 2.4. Akcelerometr

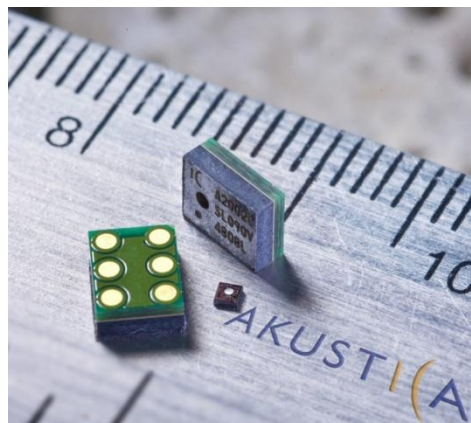
Nejrozšířenější typ akcelerometru je založen na piezoelektrickém jevu. Stejně jako u akcelerometru založeným na kondenzátoru akcelerometr založený na piezoelektrickém jevu má závaží, která vytvoří napětí na piezoelektrickém krystalu. Pod tlakem piezoelektrický krystal generuje elektrický proud, který umožňuje vypočítat zrychlení, při znalosti ostatních parametrů systému [17].

2.5.3 Mikrofon

Mikrofon je zařízení pro přeměnu akustického signálu na signál elektrický.

Princip mikrofonu spočívá v tom, že tlak zvukových vibrací vzduchu působí na tenkou membránu mikrofonu. Vibrace membrány vyvolávají oscilace.

V závislosti na typu použitého mikrofonu se používá jevu elektromagnetické indukce na vinutí, nebo ke změně kapacity kondenzátoru, nebo k piezoelektrickému efektu na krystalu [13].



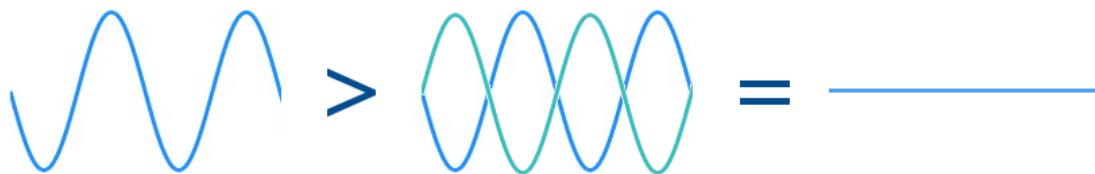
Obrázek 2.5. MEMS mikrofon

V mobilních zařízeních se často používá kondenzátorový mikrofon, je skoro nejrozšířenějším druhem mikrofonu, kvůli svoji jednoduchosti, ceně výroby a minimální možných rozměrech, což je pro mobilní zařízení jedna s nejdůležitějších věcí. Základem kondenzátorového mikrofonu je kondenzátor. Pracuje na principu změn elektrického napětí, způsobených změnou kapacity. Mikrofon je tvořen membránou z kovu či pokovené plastové fólie, umístěnou 0,02mm od druhé pevné elektrody. Vodivá membrána se akustickým tlakem deformuje, což způsobuje změnu kapacity mikrofonu. Mění-li se kapacita mikrofonu, mění se i akumulovaný náboj, který se vyrovnává přes zdroj napětí a to způsobuje průchod proudu obvodem. Přiváděné napětí však musí být relativně vysoké (řádu desítek voltů), což je značnou nevýhodou tohoto typu mikrofonu [14].

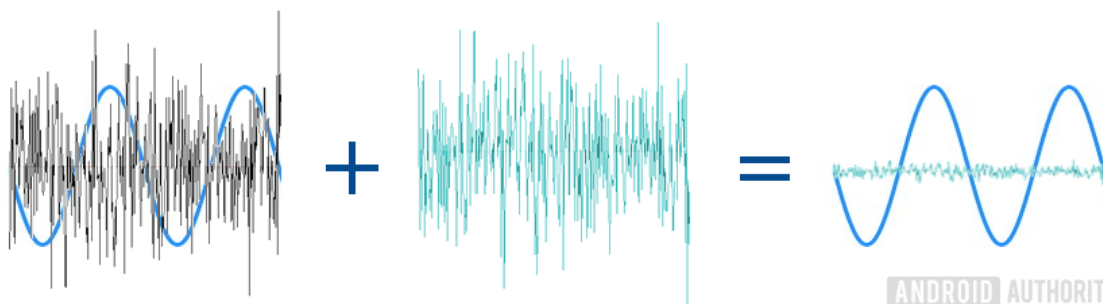
Často moderní chytré telefony jsou vybavené více mikrofony. A slouží především pro rozdělení účelu použití. Hlavní mikrofon, který se nachází ze předu, se používá při snímání hlasu při rozhovoru. Mikrofon, který se nachází na zadní ploše přístroje, se používá pro snímání zvukového signálu, například při snímání videí. Tyto mikrofony mohou rovněž pracovat spolu a to za účelem potlačení šumu při rozhovoru. Funguje to následujícím způsobem : jelikož druhý mikrofon se nachází v zadní části zařízení, hlavní mikrofon zachycuje hlas uživatele silněji, naopak zadní mikrofon bude zachytávat šum okolí silněji. Speciální čip pro zpracování audio signálu z mikrofonu analyzuje frekvenční spektrum z obou mikrofonů.

Pomocí speciálního algoritmu lze rozlišit hlas uživatele od šumu na úrovni programu.

Waves of equal amplitude and opposite phase cancel out



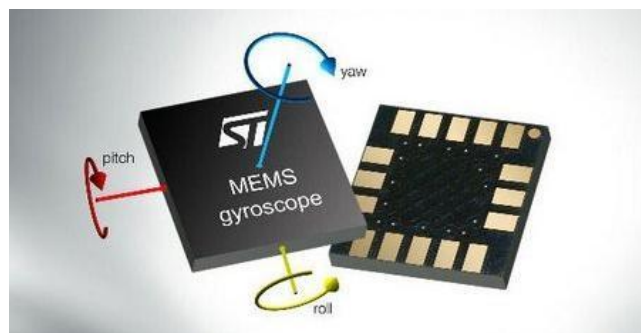
Recording and inverting noise leaves you with your desired signal



Obrázek 2.6. Filtrování šumu

2.5.4 Gyroskop

Hlavním účelem gyroskopu [17] je měřit úhlové rychlosti vzhledem k jedné nebo více osám. Gyroskop je podobné zařízení jako akcelerometr a často se používají dohromady, například spolupráce gyroskopu s akcelerometrem umožní sledovat a zjišťovat pohyb v třírozměrném prostoru. Architektura gyroskopů je dosti podobná architektuře akcelerometru. Mnoho z těchto zařízení je založených na kondenzátorech. Takový návrh například používá ve svých produktech STMicroelectronics [18]. Základem jejich gyroskopu je mechanický prvek, který funguje na principu ladičky a používá efekt Coriolisovy síly. To znamená, že transformuje úhlovou rychlost na pohyb citlivé struktury.



Obrázek 2.7. Gyroskop

Dvě pohyblivé hmoty jsou v neustálém pohybu ve třech osách, které jsou na obrázku vyznačeny modře a červeně. Když se změní úhlová rychlost, začíná působit Coriolisova síla, označená žlutě. Směr Coriolisovy síly je kolmý na směr pohybu hmoty. Coriolisova síla způsobuje hmotnostní posun úměrný velikosti úhlové rychlosti. Vzhledem k tomu, že systém má strukturu kondenzátoru, jakékoli posunutí způsobí změnu elektrické kapacity. A tak se úhlová rychlost mění na elektrický parametr. Dále je třeba poznamenat, že díky použití speciálních ladiček nejsou gyroskopy STMicroelectronics citlivé na náhodné vibrace. Princip je, že obě pohybující se hmoty se budou pohybovat ve stejném směru, čímž se nezmění kapacita kondenzátoru [17].

2.5.5 Barometr

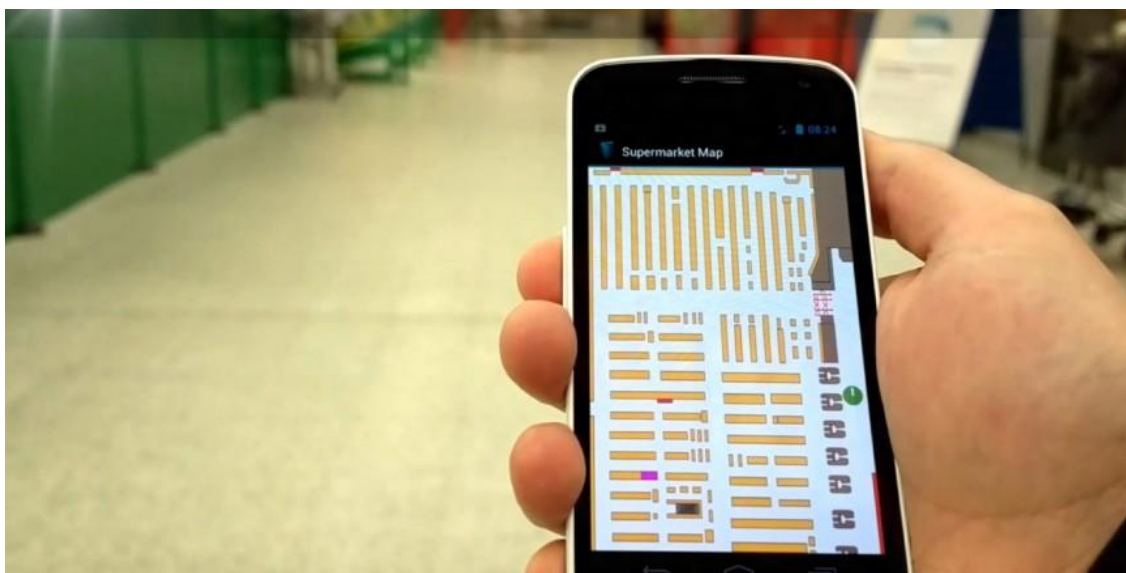
V telefonu se rovněž často nachází digitální barometr který slouží pro měření atmosférického tlaku. Měření tlaku se probíhá na základě detekce změny elektrického náboje při deformaci tenzometrického převodníku, který se deformuje se změnou atmosférického tlaku.

Barometr se používá v mnoha situacích. Například zkracuje dobu potřebnou ke zjištění polohy pomocí systému GPS. Samotná práce snímače spočívá ve srovnání vnějšího atmosférického tlaku s ohledem na vakuovou komoru uvnitř samotného snímače. Umožňuje určit polohu uživatele s přesností 50 cm na výšku a značně rozšiřuje navigační schopnosti uživatele, protože umožňuje také určit umístění vertikálně. Například mobilní telefon s barometrem vám pomůže určit trasu ve všech podlažích nákupního centra, které systém GPS nedokáže zpracovat.

Pro orientaci v budovách je však nutný speciální algoritmus implementovaný v software telefonu. Hlavní myšlenkou orientace uvnitř budov je, že každá místnost má svůj jedinečný geomagnetický vzhled způsobený rozdíly v interiérech

a architektonických podobách. Stojí za zmínku, že ani kovové konstrukce uvnitř budov, ani elektrické vedení nezasahují do správného fungování takového algoritmu. Na základě geomagnetické komponenty se vygeneruje mapa místnosti a zapamatuje se pro budoucí použití. Samozřejmě, pro plné fungování služby je potřeba nejprve vytvořit mnoho mapových podkladů. Tento úkol leží na bedrech samotných uživatelů a jeho řešení jistě zabere mnoho času. Podle vývojářů může trvat až dvě hodiny, než se vytvoří geomagnetická mapa velké místnosti [17].

Při práci s takovými mapovými podklady ovšem může být poloha uživatele určena s přesností na 2 metry [17].



Obrázek 2.8. Využití barometru

2.5.6 Magnetometr

Magnetometr, tak jako běžný magnetický kompas, sleduje orientaci zařízení v prostoru vzhledem k magnetickým pólům Země. Přijaté informace se používají hlavně v kartografických a navigačních aplikacích. Magnetometr v systému Android je velmi užitečný pro implementaci funkcí kompasu.

2.5.7 GPS

Globální polohovací systém GPS [20] je systém, který umožňuje určit polohu objektu s přesností nejméně několika desítek metrů, tj. jeho zeměpisné šířky, délky a nadmořské výšky, stejně tak lze zjistit směr a rychlost jeho pohybu. Kromě toho lze pomocí GPS určit čas s přesností na 1 nanosekundu.

GPS se skládá ze několik umělých zemních družic a pozemních stanic, které jsou spojeny ve společné síti. Protože uživatelským zařízením jsou individuální GPS přijímače schopné přijímat signály ze satelitů a z přijatých informací vypočítat jejich polohu.

GPS systém má alespoň 24 stanic, které se nachází v různých kruhových drahách, jejichž roviny se od sebe liší v podélném směru až 60° , a sklon k rovníkové rovině o 55° . Období otáčení jednoho družice je asi 12 hodin.



Obrázek 2.9. GPS

Pravidelně satelity posílají následující informace do Zemi:

- aktuální datum
- aktuální čas
- přesný čas odeslání celé sady zpráv
- vypočtené souřadnice jeho polohy v tomto okamžiku

GPS přijímač na základě informací získaných ze satelitů určuje vzdálenost jednotlivých satelitů a vypočítává jejich souřadnice. Za účelem určení dvou souřadnic (zeměpisné šířky a délky) stačí přijímat signály ze třech satelitů, a pro určení výšky nad zemským povrchem stačí přijímat signály ze čtyř družic.

Vzhledem k šíření rádiových signálů je vzdálenost k satelitům určena zpožděním v čase příjmu zprávy pomocí přijímače GPS vzhledem k době odeslání zprávy ze satelitní desky. Samozřejmě, pro přesné určení tohoto zpoždění musí být synchronizovány hodiny na hodinách satelitů a v přijímači GPS, což je zajištěno synchronizací hodin přijímače s informacemi obsaženými v satelitních signálech, jak bylo uvedeno výše.

Vzhledem k takové přesnosti GPS se používají docela často v mobilních zařízeních a telefonech. Používají se přednostně pro navigační aplikaci.

2.6 Technologie které používá Android

2.6.1 Bluetooth

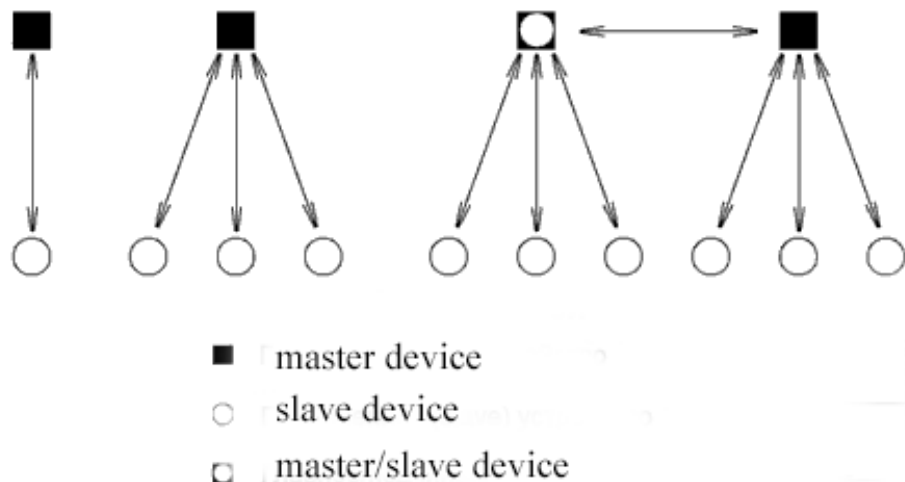
Bluetooth je standard bezdrátové rádiové komunikace pro připojení zařízení různých typů na krátké. Je určen pro vytváření osobních sítí. Technologie Bluetooth používá stejnou provozní frekvenci 2,4 GHz jako u bezdrátových standardů 802.11b a 802.11g, nicméně spotřebovává méně energie během provozu.

Nízká spotřeba energie vede k mnohem nižším přenosovým rychlostem, ale tato metoda připojení získala své místo a je používána téměř všeobecně.

Existují tři třídy standardu Bluetooth. Třída 1, která se používá pro průmyslové účely a poskytuje rozsah do 100 metrů a má vyzařovaný výkon 100 mW. Třída 2 je nejrozšířenější třídou standardu Bluetooth, vyzařovaný výkon je pouze 2,5 mW a dosahuje až 10 metrů. Třída 3 vyzařuje maximálně 1 mW, ale dosah je pouze 1 metr [32].

Několik zařízení, kombinovaných s Bluetooth, tvoří tzv. Piconet (Piconet) - síť dvou až osmi zařízení. V rámci Piconetu je jedním z přístrojů hlavní a další jsou vedlejší. Bluetooth může podporovat několik vedlejších zařízení - maximálně sedm. Hlavní je uživatelské zařízení, které iniciuje komunikaci. Řídí také provoz zařízení v piconetu.

Topology



Obrázek 2.10 Bluetooth topologie

2.6.2 Wi-fi

Bezdrátovou lokální síť standardu IEEE 802.11 označujeme Wi-Fi [34].

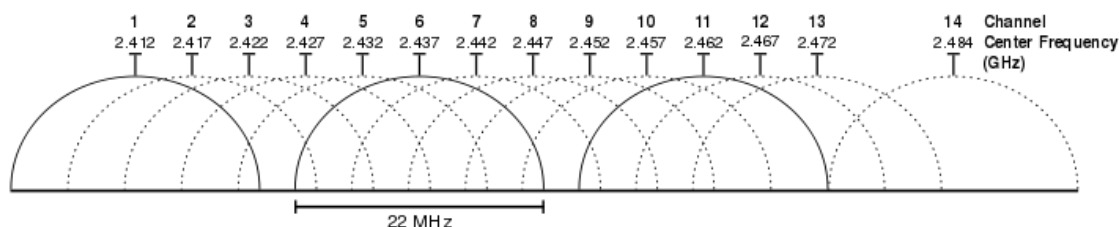
Schéma sítě Wi-Fi obvykle obsahuje alespoň jeden přístupový bod a alespoň jednoho klienta. Je také možné propojit dva klienty v režimu bod-bod (Ad-hoc), když přístupový bod není použit a klienti se připojují přímo prostřednictvím síťových adaptérů. Přístupový bod přenáší jeho identifikátor sítě (SSID) se speciálními signálovými pakety rychlostí 0,1 Mb/s každých 100 ms. Proto je 0,1 Mb/s nejnižší přenosovou rychlostí pro Wi-Fi. Při znalosti SSID sítě může klient určit, zda je možné připojení k tomuto přístupovému bodu. Pokud dva přístupové body se stejnými identifikátory SSID spadají do zóny pokrytí, přijímač si může vybrat mezi nimi na základě úrovně signálu. Standardní Wi-Fi poskytuje klientovi naprostou volnost při výběru kritérií pro připojení [35].

Pro vzájemnou komunikaci mezi zařízeními využívá rádiové vlny v kmitočtovém pásmu 2,4 GHz a 5 GHz [36].

standard	teoretická rychlost	kmitočet
802.11a	do 54 Mb/s	5 GHz
802.11b	do 11 Mb/s	2,4 GHz
802.11g	do 54 Mb/s	2,4 GHz
802.11n	do 600 Mb/s	2,4 i 5 GHz

Obrázek 2.11 wi-fi standarty

Hlavním prvkem sítě Wi-Fi je přístupový bod (AP), vedle kterého se vytvoří bezdrátová síť. Přístupový bod spolupracuje s klientskými adaptéry (bezdrátové síťové karty v počítačích). Wi-Fi umožňuje přenos dat rychlostí kanálu 54-100 Mb/s, je kompatibilní s 802.11b. Pro usnadnění přenosu dat je frekvence rozdělena na tzv. kanály [37].



Obrázek 2.12 wi-fi kanály

Ve frekvenčním pásmu Wi-Fi 2,4 GHz jsou k dispozici 3 nepřekrývající se kanály: 1, 6, 11. Toto přidělení je založeno na požadavku IEEE poskytovat minimálně 25MHz pro oddělení center bezdrátových kmitočtových kanálů Wi-Fi. Šířka kanálu je 22MHz.

2.6.3 NFC

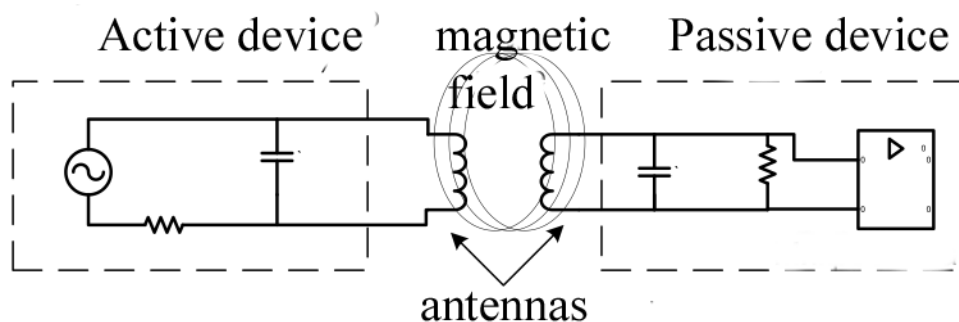
NFC (near field communication) je standardizovaná technologie pro výměnu dat na krátké vzdálenosti, což umožňuje jednoduchou a intuitivní interakci mezi dvěma elektronickými zařízeními. Například s smartphonem vybaveným technologií NFC můžete nakupovat, rozdávat vizitky, dostávat kontextová data [39].

NFC je založen na technologii RFID s frekvencí 13,56 MHz a pracovní vzdáleností až 10 cm. Rychlost přenosu dat je až 424 kb/s. Ve srovnání s jinými komunikačními technologiemi je hlavní výhodou NFC jeho rychlost a snadné používání.

V rámci zařízení, kromě aktivních regulátorů NFC existují pasivní varianty (obvykle označované jako tagy), které přijímají napájení z aktivního zdroje. Příkladem jsou karty, které se používají v hromadné dopravě (např. tzv. Lítačka). Tagy jsou pouze úložištěm dat, jejichž objem obvykle nepřesahuje 4 KB. Nejčastěji je pro ně k dispozici pouze režim čtení, ale existuje i podpora pro psaní.

Princip funkčnosti NFC je založen na elektromagnetické indukci. Vysílač čtečky prostřednictvím antény nepřetržitě vydává signál sinusové formy s frekvencí 13,56 MHz. NFC-tag má také smyčkovou anténu. Když je snímač přiveden do čtečky, obě cívky tvoří vzduchový transformátor. Střídavý proud v první cívkce generuje magnetické pole, které vytváří proud ve druhé cívkce. Tato energie je dostatečná pro provoz mikroobvodu.

Během komunikace hlavní tranzistor vysílacího zařízení periodicky budí anténu na základě řídicího programu a přenášených dat. To způsobuje modulaci vysokofrekvenčního signálu, který je snímán čtečkou [39].



Obrázek 2.13 NFC

Existují dva režimy komunikace - pasivní a aktivní. V prvním případě čtečka vytvoří elektromagnetické pole a NFC-tag díky tomuto poli se zapne a vytvoří odpověď. Velmi výhodnou vlastností je, že není potřeba napájení tagu, takže může být velmi malé.

3. Struktura aplikace pro systém Android

3.1 Architektura platformy Android

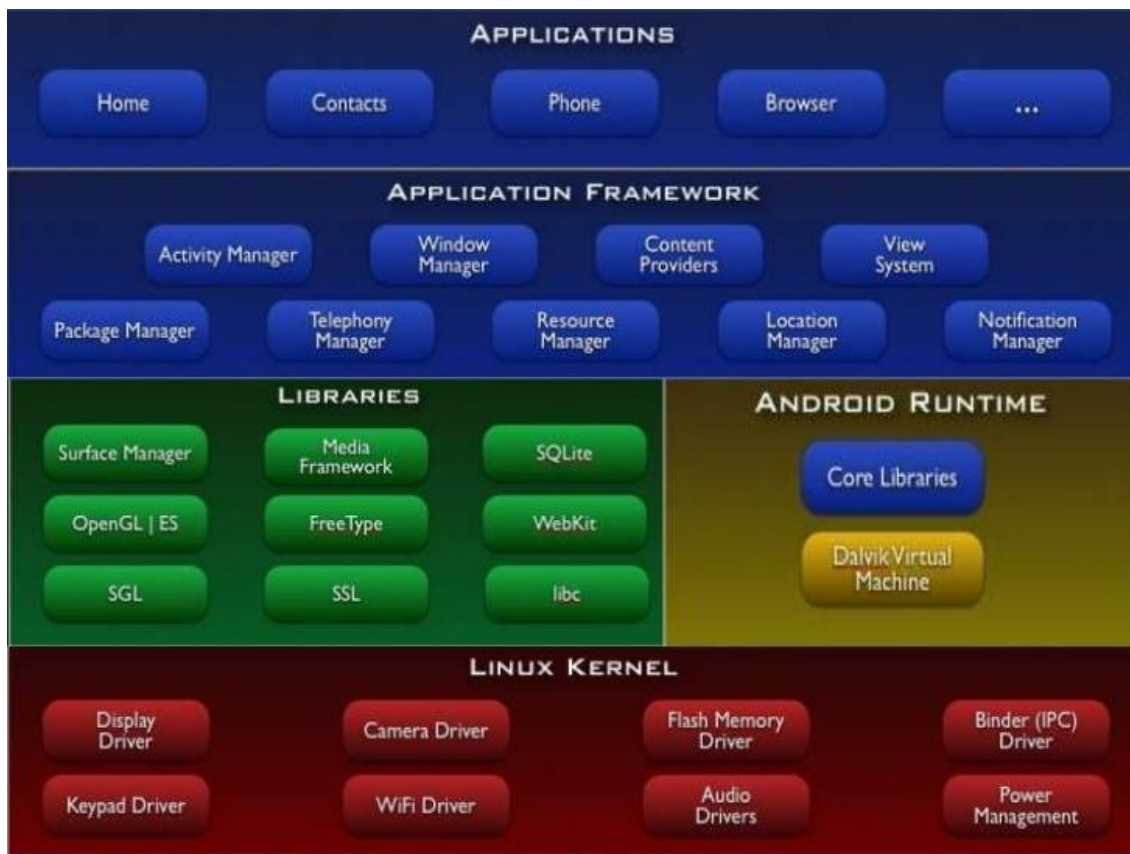
Platforma Android je balík softwaru pro mobilní zařízení, který zahrnuje:

- operační systém
- middleware
- hlavní uživatelské aplikace, které jsou součástí mobilního telefonu (nebo PDA), jako je kalendář, prohlížeč, kontaktní databáze, SMS zprávy atd.

Architektura Androidu je rozdělena do čtyř úrovní:

- **Úroveň jádra.** Android je založen na jádře Linuxu se základními systémovými službami - zabezpečení, správě paměti, správě procesů a modelu driverů. Vývojáři Android modifikovali jádro Linuxu, přidali hardwarovou podporu zařízení, používaných v mobilních zařízeních, ale neobvyklých pro počítače.
- **Úroveň runtime.** Android obsahuje soubor základních knihoven, které poskytují většinu funkcí dostupných v knihovnách Java. Každá aplikace Android se spouští jako vlastní proces, s vlastní instancí virtuálního stroje Dalvik (v novějších verzích Androidu se místo Dalviku používá jiný virtuální stroj). Dalvik byl napsán tak, že zařízení může pracovat efektivně s několika virtuálními stroji najednou. Dalvik byl navržen speciálně pro platformu Android. Virtuální stroj je optimalizován pro nízkou spotřebu paměti a práci s mobilním hardwarem zabezpečení. Dalvik používá svůj vlastní bytecode. Aplikace pro Android se překládají kompilátorem do speciálního nízkoúrovňového kódu nezávislého na stroji. Tento kód Dalvik interpretuje a provádí takový program. Navíc, pomocí speciálního nástroje, který je součástí sady Android SDK, je Dalvik schopen překládat byte-kódy Java do svých vlastních formátových kódů a také je spouštět ve svém virtuálním prostředí
- **Úroveň knihoven** Android obsahuje sadu knihoven C / C ++ používaných různými komponentami systému. Tyto knihovny jsou vývojářům k dispozici v rámci Android Application Framework. Některé hlavní knihovny jsou uvedeny níže:
 - Knihovna médií - tyto knihovny poskytují podporu pro přehrávání a nahrávání mnoho populárních audio a video formátů a formátů obrázků, včetně MPEG4, MP3, AAC, AMR, JPG, PNG a další.

- Surface Manager - ovládá přístup k subsystému zobrazování 2D a 3D grafických vrstev.
 - LibWebCore - moderní webový engine, na kterém je postavený prohlížeč Android.
 - SGL - hlavní grafický engine 2D.
 - 3D knihovny - implementované na bázi OpenGL; knihovny používají hardware 3D akcelerace (je-li k dispozici).
 - FreeType - podpora rastrových a vektorových fontů.
 - SQLite je databázový stroj, který je přístupný všem aplikacím.
-
- **Úroveň aplikace.** Poskytnutím otevřené vývojové platformy Android dává vývojářům možnost vytvářet flexibilní a inovativní aplikace. Vývojáři mohou používat hardwarové možnosti zařízení, získávat informace o umístění, provádět úlohy na pozadí, nastavovat notifikace a ještě mnohem více. Vývojáři mají úplný přístup ke stejnému rozhraní API, které se používá v systémových aplikacích. Aplikační architektura je navržena tak, aby zjednodušila opětovné použití komponenty; může každá aplikace "publikovat" své funkce a jakákoli jiná aplikace pak může tyto funkce používat. Stejný mechanismus vám umožňuje nahradit standardní komponenty vlastními.



Obrázek 2.14 Architektura Android

3.2 Struktura Android aplikace

Aplikace pro Android mohou být jednoduché nebo složité, ale aplikační struktura bude vždy stejná. Některé prvky aplikací jsou povinné a některé jsou volitelné. ty se používají podle potřeby. Aplikace pro Android se skládá z několika základních komponent: manifest aplikace, sada zdrojů (resource) a zdrojový kód programu.

Název folderu nebo souboru	Funkce
gen	Soubory generované samotnou Java.
AndroidManifest.xml	Soubor manifestu AndroidManifest.xml poskytuje základní informace o programu. Každá Aplikace musí mít vlastní soubor manifestu.
src	Adresář obsahující zdrojový kód aplikace
assets	Adresář obsahující katalogy a libovolné soubory
res	Adresář obsahující zdroje aplikací. V tomhle adresáři může se nacházet podsložky drawable, anim, layout, menu, values, xml a raw

Tabulka 1. Struktura Android aplikace

3.3 Aktivity

Aktivita [23] je uživatelské rozhraní pro jednu akci kterou může uživatel udělat. Například aplikace pro textové zprávy může mít jednu aktivitu pro zobrazení seznamu kontaktů a další pro psaní zprávy vybranému kontaktu, třetí pro zobrazení zpráv a další pro změnu nastavení. Všechny tyto aktivity tvoří jedno uživatelské rozhraní, ale nezávisí na sobě. Aplikace se může skládat z jedné aktivity nebo několika. Záleží na typu aplikace a její návrhu.

Jedna aktivita může vyvolat další. Aktivita většinou koresponduje s oknem, které se zobrazuje na celý displej, ale může být menší okno, případně plovoucí okno. Aktivita může využívat další okna, například rozbalovací dialog, který vyžaduje reakci uživatele nebo okno, které zobrazí uživatelům důležité informace při výběru položky, která si zaslouží zvláštní pozornost.

Vizuální rozhraní je postaveno na základě hierarchie vizuálních komponent - objektů, odvozené od třídy základního zobrazení. Android má řadu komponent připravených k použití, včetně tlačítek, textových polí, zaškrtačkových políček a mnoha dalších.

Aktivita může být v jednom ze třech stavů:

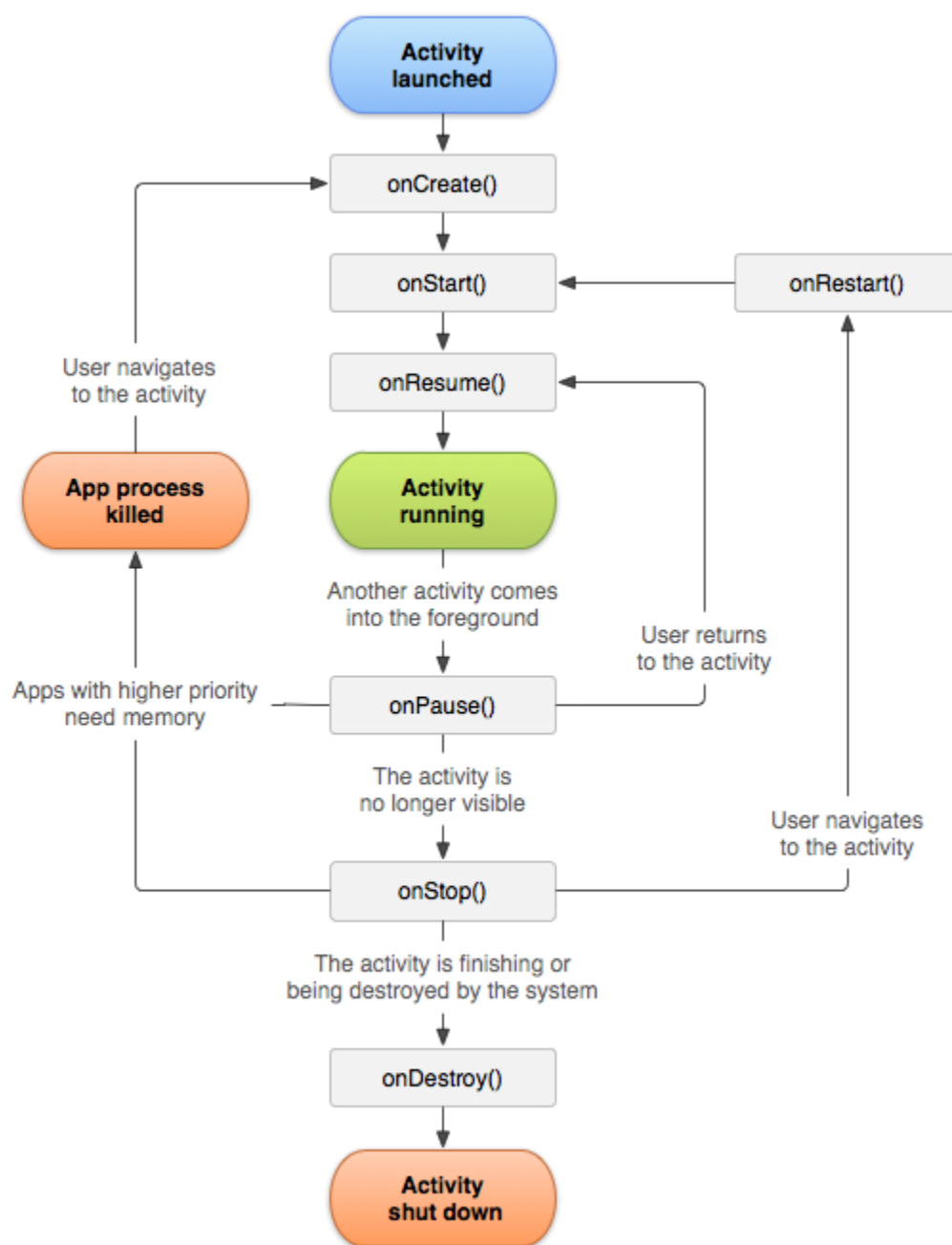
- **active** nebo **running** (bežící stav)- nachází se na předním planu a má aktivní fokus pro interakci s uživatelem.
- **paused** (přerušovaný stav) - neaktivní pro interakci, ale stále viditelné pro uživatele. Nahoře je další aktivita, která je transparentní nebo překrývá celou obrazovku.
- **přerušovaná** činnost zcela "živá" (její stav je zachován), ale může být v případě nedostatku paměti systémem zrušen.
- **finished** nebo **killed** (zrušený stav) - zcela zablokováno jinou aktivitou. Už není vidět pro uživatele a je zničen systémem, pokud bylo potřeba dodatečnou paměť.

Pokud je aktivita pozastavena nebo zrušená, systém ji může vymazat z paměti nebo poslat požadavek na její ukončení nebo jednoduše zničit její proces. Je-li aktivita znovu zobrazena uživateli to znamená že její stav je plně obnoven.

Při přechodu mezi stavy o tom aktivita informuje voláním následujících metod:

- **void onCreate()** - volá se jednou, když je aktivita vytvořena. Zde se probíhá nastavení - vytváření instancí třídy *View* (uživatelské rozhraní), zápis dat do seznamy a podobně. Následně následuje volání metody *onStart()*.
- **void onStart()** - volá se vždy když aktivita se spouští po předchozím zastavení.
- **void onStart()** - volá se před tím než aktivita začne být viditelná pro uživatele. Následně probíhá volání *onResume()*, pokud je aktivita na popředí nebo *onStop()*, pokud je skrytá.
- **void onResume()** - volá se před zahájením interakci aktivity s uživatelem. Z toho okamžiku aktivita je v horní části zásobníku a přijímá veškeré vstupy uživatele. Je následována voláním *onPause()*.
- **void onPause()** - volá se systémem před obnovením další aktivity. Tato metoda se obvykle používá k uložení změněných dat, zastavení animace a dalších operaci, které mohou spotřebovat procesorový čas. Další aktivita nebude obnovena, dokud není dokončena předchozí. Metoda *onResume()* může být zavolaná ihned potom pokud se aktivita vrátí do popředí, nebo metodu *onStop()*, pokud se stane neviditelné pro uživatele.

- ***void onStop()*** - volá se, když není aktivita viditelná uživateli. To se může stát, pokud je aktivita v procesu zastavení, nebo proto, že jiná aktivita byla obnovena a překryla aktuální. Potom může být zavolaná metoda *onRestart()* pokud je aktivita obnovena pro interakci s uživatelem, nebo metoda *onDestroy()*, pokud se běh aktivity ukončuje.
- ***void onDestroy()*** - volá se před zničením aktivity. To je poslední volání, který obdrží aktivita. Může být volaná z důvodu, že aktivita se ukončuje (něco zavolalo metodu *finish()*), nebo protože tento systém dočasně ukončuje tuto aktivitu pro šetření paměti.



Obrázek 2.15 Životní cyklus aktivit Androidu

4. Související práce

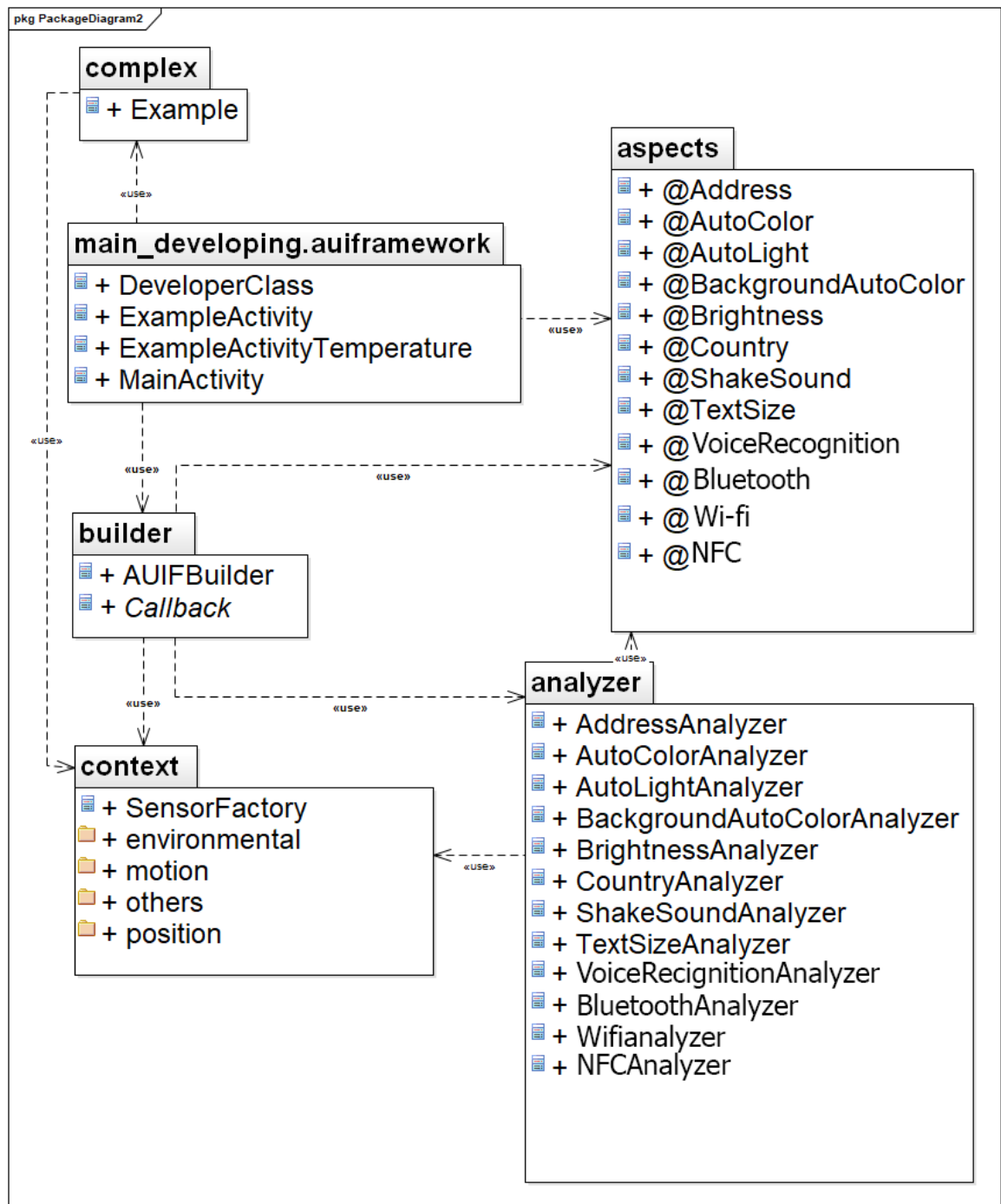
V rámci této bakalářské práce byla potřeba rozšířit funkcionalitu existujícího nástroje navrženého v rámci bakalářské práce T.Titova. “Využití aspektově orientovaného přístupu a lokálních dat ve vývoji adaptivního uživatelského rozhraní pro Android” [40].

5. Analýza a implementace framework

5.1 Použitý framework

Framework byl vyvíjen v prostředí Android Studio a je určen pro vývoj aplikací pro mobilní zařízení s operačním systémem Android. Ke splnění požadavků univerzálnosti a praktického použití, framework má modulární strukturu, která umožňuje používat nástroje jednotlivých komponent. Framework využívá princip aspektově orientovaného přístupu programování a slouží ke generování adaptivního rozhraní prostřednictvím kontextu, získaného ze senzorů mobilního zařízení.

5.2 Package diagram



Obrázek 2.16 Package diagram

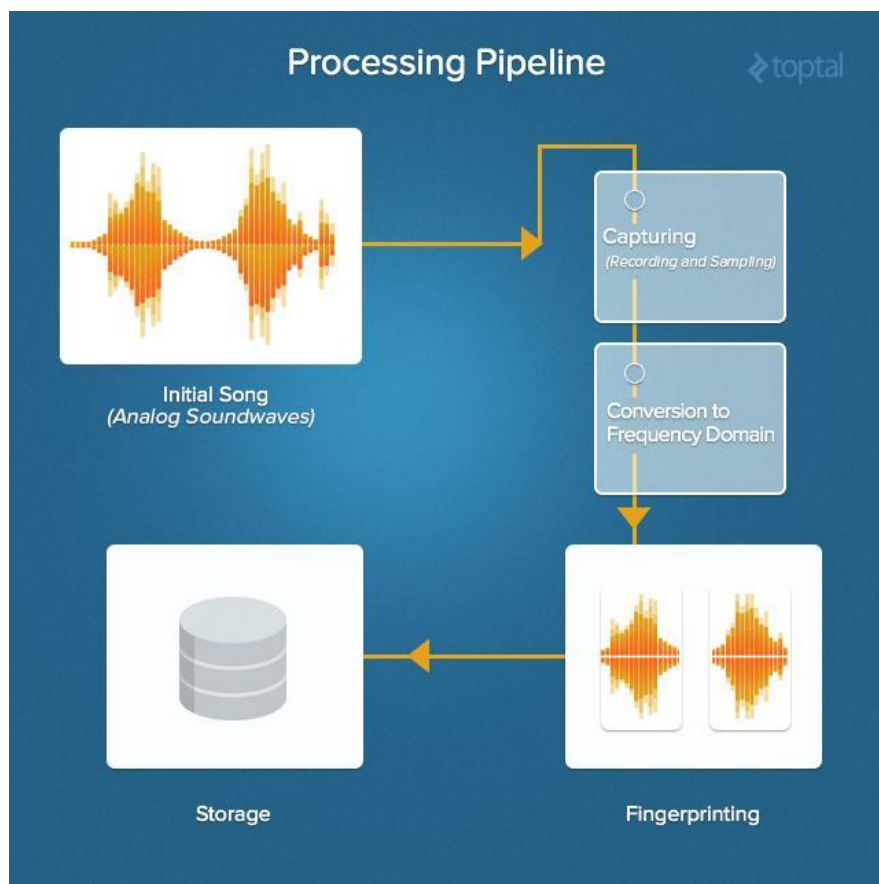
5.3 Případy užití

5.3.1 Use-case adaptace menu na základě hlasového kontextu

Jak bylo zmíněno v předchozích kapitolách, používání okolního kontextu je cestou k adaptaci uživatelského rozhraní.

V tomto případě užití se používá uživatelský kontext převzatý prostřednictvím mikrofonu. Získat, správně zanalyzovat a využít kontextové informace je úlohou tohoto případu užití. Pro kvalitní kontext z mikrofonu je potřeba získaný signál oddělit od šumu a získaný hlas rozpoznat jako jednotlivá slova.

Proces oddělení šumu od získaného zvukového signálu a následující zpracování výsledku lze sledovat na příkladě aplikace, které rozpozná název skladby z jejího krátkého úseku.



Obrázek 2.17 Proces rozpoznání zvuku

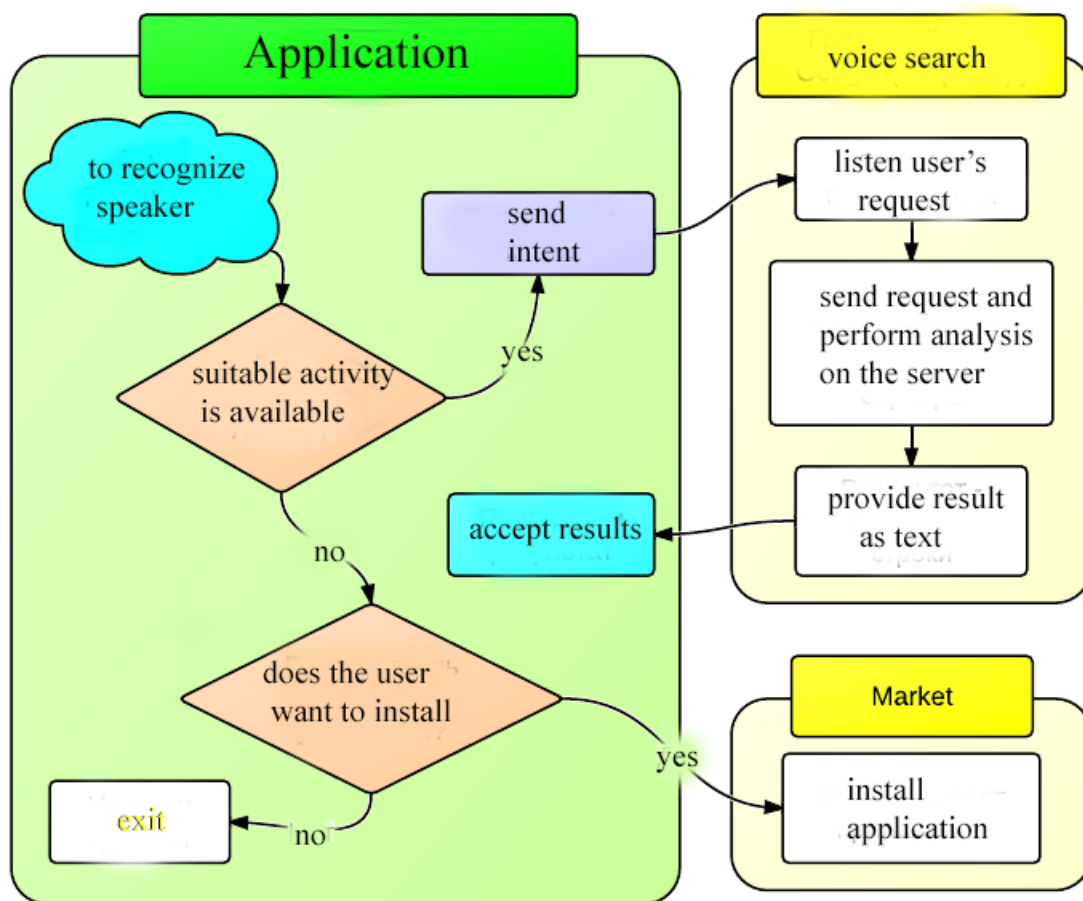
Jak je vidět na obrázku, zpracování a rozpoznání hudby začíná získáním původního zvukového signálu. Jakmile je zachycen, naleznou se frekvenční charakteristiky, a vypočítá se z nich hash kód, který se následně porovná je s těmi, které jsou uloženy v hudební databázi.

Databáze pro podobnou aplikaci může být poměrně velká, takže je důležité používat řešení, které je škálovatelné. Neexistuje zde potřeba komplexního databázového modelu, datový model je velmi jednoduchý, takže je vhodná databáze NoSQL [24]. Pro přímé rozpoznání hlasu a jeho následné zpracování a aby se dalo pracovat s kontextem ve formátu textu, v Androidu existuje speciální třída “*SpeedRecognizer*”. Tato třída poskytuje přístup ke službě rozpoznávání řeči a dědí se od rozhraní *android.speech*. Zvuk se přenáší na vzdálené servery za účelem rozpoznávání řeči. Android při zpracování řeči pomocí *android.speech* používá následující metody:

Název metody	Popis metody a případ použití
<i>onBeginningOfSpeech()</i>	Uživatel začal mluvit
<i>onBufferReceived(byte[] buffer)</i>	Byl přijat hlasitější zvukový signál. Účelem této funkce je umožnit uživateli zpětnou vazbu týkající se zachyceného zvuku. Neexistuje žádná záruka, že tato metoda bude volána.
<i>onEndOfSpeech()</i>	Zavolá se když uživatel přestane mluvit.
<i>onError(int error)</i>	Došlo k chybě sítě nebo chybě během rozpoznávání.
<i>onEvent(int eventType, Bundle params)</i>	Vyhrazeno pro přidání budoucích událostí.
<i>onPartialResults(Bundle partialResults)</i>	Voláno, pokud jsou k dispozici výsledky částečného rozpoznávání. Zpětné volání lze kdykoli volat mezi příkazy <i>OnBeginningOfSpeech()</i> a <i>onResults(Bundle)</i> , když jsou připraveny částečné výsledky. Tuto metodu lze volat jednou nebo vícekrát pro každý hovor pro <i>startListening(Intent)</i> , v závislosti na implementaci služby rozpoznávání řeči.
<i>onReadyForSpeech(Bundle params)</i>	Voláno, když je nastroj připraven, aby mohl uživatel začít mluvit.
<i>onResults(Bundle results)</i>	Voláno, když jsou výsledky rozpoznání připraveny.

Jak je vidět na obrázku 4.3, funkčnost třídy v rámci aplikace zaručuje následující postup zpracování řeči:

- přijme požadavek na zpracování řeči
- ověří dostupnost aplikace schopné rozpoznat řeč
- poslouchá uživatele
- po úspěšném skončení poslechu odešle požadavek z výslednými daty na server
- pak se vrátí výsledek rozpoznání v textové formě



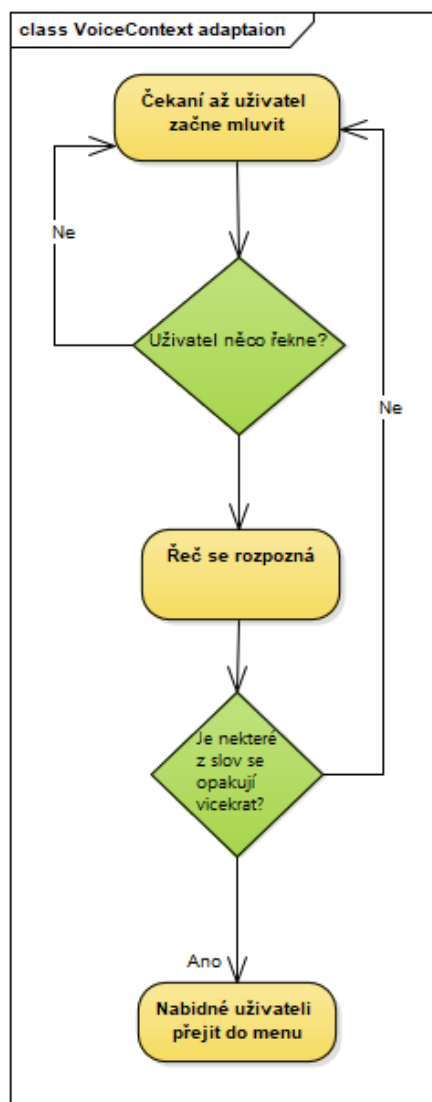
Obrázek 2.18 Cyklus API pro rozpoznání zvuku

Tento případ užití využívá popsaný algoritmus získání kontextu k adaptaci uživatelského rozhraní prostřednictvím poslechu uživatelských komentářů během

použití aplikace. Nástroj čeká až uživatel něco řekne. Na základě získaného kontextu nástroj provede následující analýzu:

- Zapiše slova do tabulky u které jeden sloupec budou slova a druhý počet opakování
- Při detekci více opakování slov se prohlédne, jestli v aplikaci existuje menu s názvem, který je totožný slovu, které se opakovalo vícekrát, a v případě úspěchu nabídne uživateli přejít do toho menu.

Na obrázku č.2.19 je znázorněn algoritmus analýzy uživatelského kontextu s následující úpravou uživatelského rozhraní.



Obrázek 2.19 VoiceContext adaptace

Za účelem získání kontextu se používá třída *VoiceContext*. Mikrofon zachytává hlas a na základě toho framework vyhodnotí jaké slova uživatel řekl a uloží výsledek do *ArrayListu*, ve kterém se nachází rozpoznané slova v pořadí od nejvíce podobného slova až do nejméně podobného. Pomocí metody *getSaidWords()* framework dostane kontext v textové podobě a na základě těchto dat nabídne uživateli přejít do hledané obrazovky.

```
public String getSaidWords() {  
    return this.matches.get(0);  
}
```

Musí být uděleno oprávnění k použití mikrofonu. Za tímto účelem se přidává do manifestu řádka

```
<uses-permission android:name="android.permission.RECORD_AUDIO" />
```

5.3.2 Use-case adaptace menu na základě kontextu vedlejších

V tomto případě užití se používá uživatelský kontext obdrženy prostřednictvím Bluetooth a barometru.

Pomocí Bluetooth lze navzájem přenášet data mezi mobilními zařízeními. Data získaná z jednoho zařízení lze použít jako zdroj kontextu pro adaptace uživatelského rozhraní druhého zařízení (vedlejšího), t.j. adaptovat jej. Programátor bude mít možnost nastavit jakoukoliv součást uživatelského rozhraní podle obdržných dat. Například se dá synchronizovat barvu pozadí a rozměr textu hlavních elementů aplikace. Tato funkce se dá používat pro synchronizaci mobilních zařízení jednoho uživatele, pokud bude chtít použít v jiném mobilním zařízení svoje přednastavené uživatelské rozhraní.

Aby byl zajištěn přenos dat a aplikace mohla předpokládat následující bezchybný přenos, je možné stanovit další podmínky:

- Pomocí vestavného snímače tlaku je možné zajistit, že zařízení se nacházejí ve stejné výšce, což by mohlo znamenat že se přístroje opravdu nachází vedle sebe a uživatel bude chtít použít funkci synchronizace uživatelských rozhraní svých zařízení.

Pro použití Bluetooth v aplikaci musí programátor provést následující kroky:

- Nejprve je třeba připojit odpovídající balík API.

```
import android.bluetooth.*;
```

- Kromě toho musí být uděleno oprávnění k použití modulu Bluetooth. Přidávají se do manifestu následující řádky

```
<uses-permission android:name="android.permission.BLUETOOTH" />  
<uses-permission android:name="android.permission.BLUETOOTH_ADMIN" />  
<uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION" />
```

Třídě *BluetoothAdapter* povoluje provádět hledání vzdáleného zařízení Bluetooth spuštěním skenování.

Při skenování se vyhledávají dostupné moduly Bluetooth blízko zařízení. V případě, že se dosáhne zařízení s povoleným Bluetooth, odešle se jako odpověď informace v následující podobě: jméno, třída, unikátní MAC adresa. Na základě těchto informací se může uspořádat datové připojení a přenos.

İhned po vytvoření připojení ke vzdálenému zařízení bude uživatel automaticky požádán o připojení. V případě úspěchu, informace, které byly získané při odpovědi (název, třída, a MAC adresa), budou uloženy, a pak mohou být použity přes Bluetooth rozhraní API. Takže při příští komunikace se vzdáleným zařízením již není nutné provádět opakované párování zařízení, protože v databázi telefonu již bude existovat záznam o MAC adrese

Je třeba rozlišovat mezi zařízeními, které jsou spárované a připojené. Spárovaná zařízení ví o sobě, mají klíč, což je odkaz, který lze použít pro autentizaci, a jsou schopna vytvořit šifrované připojení. Připojená zařízení sdílejí jeden rádiový kanál a mohou přenášet data navzájem. Současná implementace rozhraní Bluetooth API vyžaduje spárování zařízení před připojením.

Pro práci v rámci toho případu užití mobilního zařízení musí být předem spárované. Při přiblížení zařízení proběhne vzájemná výměna kontextové informace, které určí barvu pozadí aplikace, a to v podobě pole o třech hodnotách typu *int*. Položky v poli obsahují informace o barvě v číselné podobě od 0 do 255, a to tak že nultá položka pole obsahuje informace o červeně barvě, první obsahuje informace o zelené barvě, a druhá má v sobě informace o modré barvě. Toto pole využívá framework pro určení barvy a obdrží jej voláním metody *getContext()*.

```
public static int[] getContext(){
    return bluetoothMessage;
}
```

5.3.3 Use-case adaptace uživatelského rozhraní na základě polohy zařízení

Tento use-case určuje adaptaci uživatelského rozhraní v závislosti na poloze zařízení. V rámci tohoto use-case pro získání kontextu polohy se bude používat wi-fi. Samotná data získaná pomocí wi-fi nedávají přesné informace o umístění zařízení. Programátor bude mít možnost přednastavit jakoukoli změnu uživatelského rozhraní, na základě názvu (SSID) wi-fi sítě, ke které je zařízení aktuálně připojeno. Tímto způsobem se dá předpokládat současnou polohu uživatele a následně provést odpovídající adaptaci uživatelského rozhraní. Pro možnost získání informace pomocí wi-fi je potřeba provést následující kroky:

- Do *AndroidManifest.xml* přidat povolení:

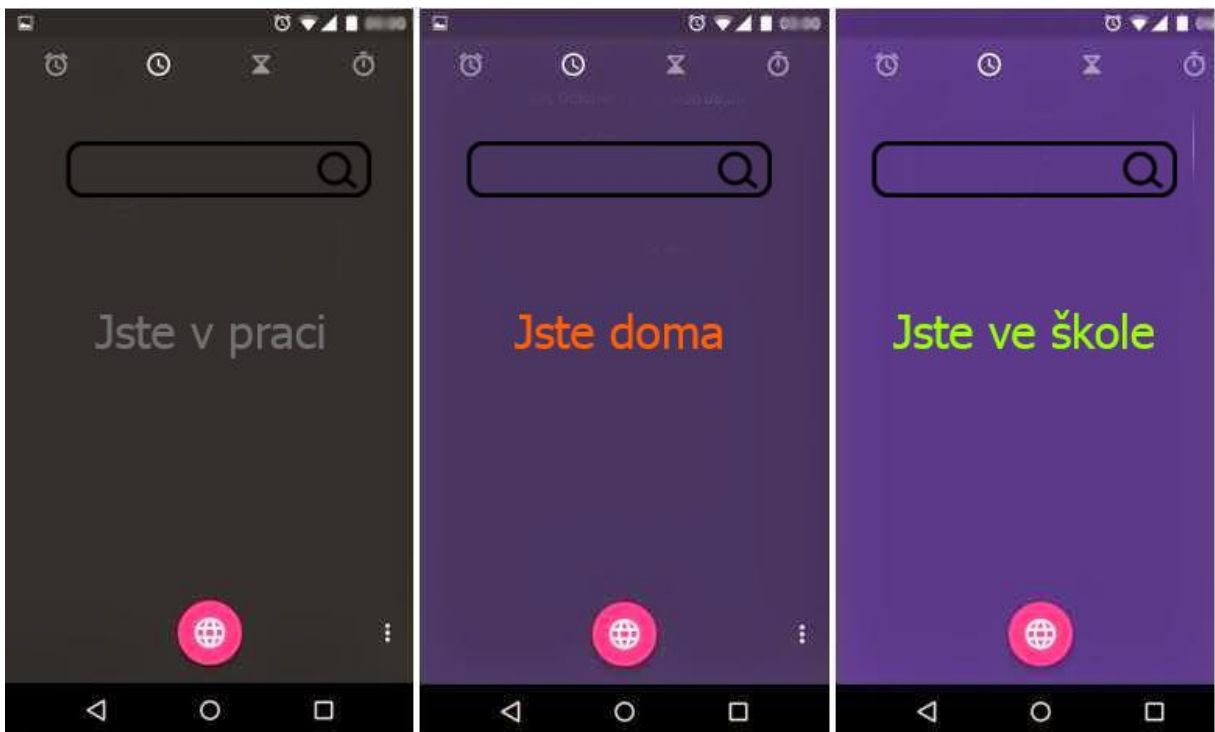
```
<uses-permission android:name="android.permission.INTERNET"/>
<uses-permission android:name="android.permission.ACCESS_WIFI_STATE"/>
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE"/>
<uses-permission android:name="android.permission.ACCESS_FINE_LOCATION"/>
```

- Programátor musí určit počet a názvy profilů a odpovídajících názvům bezdrátových sítí (SSID). Framework bude podle profilů adaptovat uživatelské rozhraní zařízení uživatele. Profil obsahuje svůj název a název sítě.

```
public @interface Wifi {
    String work() default "cetin_work";
    String home() default "LTNET";
    String school() default "eduroam";
}
```

Příkladem adaptace uživatelského rozhraní může být například změna barvy pozadí aplikace na základě toho, ke které wi-fi síti je zařízení připojeno. Lze stanovit více profilů pro více sítí.

Po detekci připojení k bezdrátové síti, která je v seznamu profilů, se vytvoří instance třídy *WifiContext* a následně se využije metoda *getSSID()*, která vrátí název sítě, ke které je zařízení právě připojeno. Jestliže bude síť s takovým názvem v seznamu profilů, vzápětí proběhne úprava uživatelského rozhraní, kterou dříve označil programátor. Jestliže nebude stanoveno jinak, proběhnou výchozí úpravy uživatelského rozhraní spočívající ve změně barev *TextView* elementů, změně barvy pozadí a zvýšení hlasitosti aplikace pro domácí síť a snížení hlasitosti pro pracovní nebo školní síť.



Obrázek 2.19 Příklad adaptace

5.3.4 Use-case adaptace uživatelského rozhraní pomocí NFC-tagů

Podobně jako u předchozího případu užití, i pomocí NFC-tagů lze stanovit polohu zařízení a uživatele. Na rozdíl od zjišťování polohy pomocí wi-fi signálu, který se může šířit na větší vzdálenosti a tudíž je možné stanovit zda se uživatel nachází v oblasti šíření bezdrátové sítě, NFC-tag dovoluje stanovit polohu v rámci jedné oblasti (místnosti) mnohem přesněji.

NFC-tag může obsahovat informace pro konkrétní aplikace a vyvolat v nich určitou funkcionalitu.

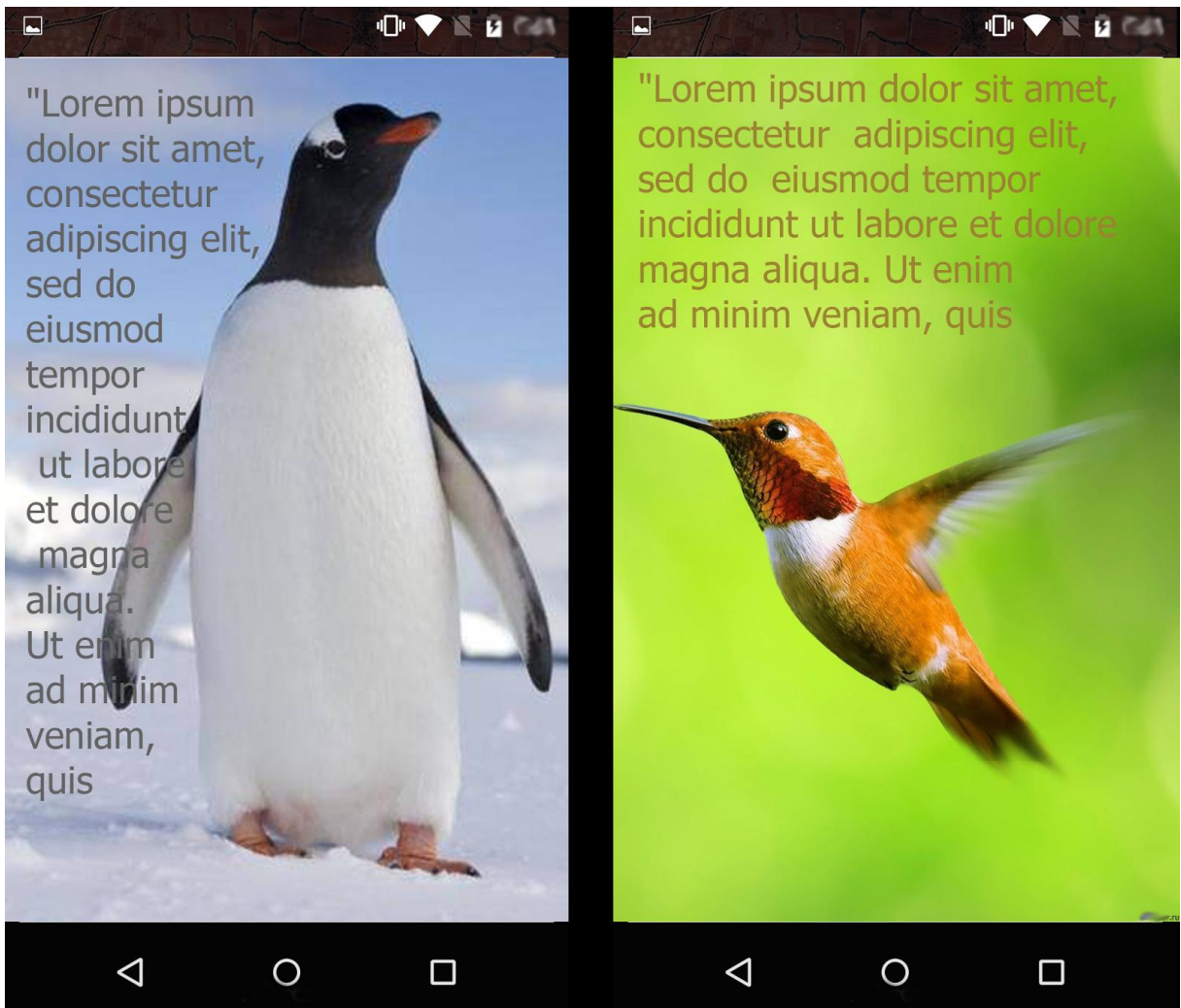
Podobně lze používat informaci uchovanou v NFC-tagu pro adaptaci uživatelského rozhraní.

Aby funkce třídy *NFCManager* byly přístupné je potřeba v *AndroidManifest.xml* přidat jejich povolení:

```
<uses-permission android:name="android.permission.NFC" />
```

Při přiložení mobilního zařízení k NFC-tagu se vytvoří instance třídy *NFCContext* a spustí se metoda *readid()*, která přečte unikátní identifikátor NFC-tagu, a následně vyvolá další metodu *checkAUI(byte[])* pro zjištění adaptace uživatelského rozhraní pro zodpovědný NFC-tag.

Konkrétním příkladem může být aplikace pro návštěvníky muzea, která si podle informace z NFC-tagu, ke kterému uživatel přiloží mobilní zařízení, adaptuje uživatelské rozhraní na základě dat z NFC tagu. Například NFC-tag může být umístěn vedle vystaveného exponátu, a při přiložení mobilu proběhnou změny uživatelského rozhraní aplikace ve formě nastavení pozadí aplikace na fotografii exponátu. Popřípadě, mohou vzniknout nové elementy uživatelského rozhraní, například textové pole, které předtím nebylo k dispozici pro hodnocení uživatelem daného exponátu.



Obrázek 2.20 Příklad adaptace pomocí NFC

6. Testování

6.1 Úvod

Cílem této kapitoly je kvantitativně otestovat interakci se dvěma typy uživatelského rozhraní : bez užití adaptace popsané v předchozím článku; s jejím užitím. Smyslem je porovnat je z hlediska použitelnosti a uživatelské přívětivosti. Pro testování byl využit T-test (Studentův t-test).

6.2 Popis algoritmu testování

T-test (Studentův t-test) [27], slouží k testování statistických hypotéz. Studentův t-test se používá k určení statistické významnosti rozdílů v průměrných hodnotách. Může být použita v případech porovnání nezávislých veličin (např. Skupin pacientů s cukrovkou a skupiny zdravých).

Pro uplatnění Studentova t-testu je nezbytné, aby počáteční data byla normálně rozdělena.

Pro porovnání průměrných hodnot se hodnota t-testu vypočte podle následujícího vzorce:

$$t = \frac{M_1 - M_2}{\sqrt{m_1^2 + m_2^2}}$$

kde M_1 je aritmetický průměr první srovnávané skupiny, M_2 je aritmetický průměr druhé srovnávané skupin, m_1 a m_2 je směrodatnou odchylku.

6.3 Nulová hypotéza

Nulová hypotéza je stanovena takto: Při použití testovací aplikace s vypnutou funkcí pro adaptaci uživatelského rozhraní se uživatelská přívětivost neliší od případu, že se funkce pro adaptaci rozhraní zapne.

6.4 Cílová skupina

Cílovou skupinou tohoto testování jsou uživatelé mobilních telefonů. Jedná se o aktivní uživatele operačního systému Android ve věku 18-99 let. Uživatelé musí mít zkušenost interakcí s tímto operačním systémem.

Jelikož testovací aplikace, která byla navržena na daném frameworku, neměla velkou funkcionalitu jediným zadáním pro cílovou skupinu bylo projít položky aplikace obsahující kapitoly s textem a jinými prvky a hodnotit jak se pracuje s aplikací. Testovací skupině bylo povoleno provádět veškeré změny uživatelského rozhraní, adaptovat ho podle svých potřeb s tím, že by mohli porovnat jestli se jim líbila následovná automatická adaptace uživatelského rozhraní.

6.5 Množina otázek

V rámci testování měli uživatelé vyzkoušet testovací aplikaci se zapnutou a vypnutou funkcí adaptace uživatelského rozhraní.

Otázky byly zvoleny takto : Pozitivní (P – 1, 3, 5, 7, 9) a Negativní (N – 2, 4, 6, 8, 10) – podle jejich hodnoty v Likertově škále [30]. Bylo provedeno střídání pozitivních a negativních otázek. Na všechny otázky bylo nutné odpovědět.

Na základě zkušenosti při interakci s testovací aplikací měli uživatelé odpovědět na řadu otázek [31]:

1. Myslím, že bych chtěl/a tuto aplikaci používat často.
2. Myslím, že aplikace je zbytečně složitá.
3. Myslím, že je jednoduché používat danou aplikaci.
4. Myslím, že je zapotřebí pomoc technicky zdatného člověka, abych aplikaci mohl/a používat.
5. Přejde mi, že různé funkce a vlastnosti jsou do dané aplikace správně zakomponované.
6. Myslím si, že je daná aplikace je nekonzistentní.

7. Myslím si, že většina lidí se naučí používat tuto aplikaci velmi rychle.
8. Myslím si, že daná aplikace je velmi těžkopádná při používání.
9. Cítím se velmi jistě při používání této aplikace
10. Musel/a jsem se naučit mnoho věcí, než jsem tuto aplikaci mohl/a používat.

6.6 Množina odpovědi

Množina odpovědí se skládá z 5 možností:

Odpověď	Hodnocení	Hodnocení
Zcela souhlasím	4	0
Spíše souhlasím	3	1
Nevím	2	2
Spíše nesouhlasím	1	3
Zcela nesouhlasím	0	4

Tabulka 2. Množina odpovědi

6.7 Nasbíraná data

Ohodnocení otázek následně sečteme a součet vynásobíme číslem 2.5, ohodnocení pak bude v intervalu 0-100

Uživatel/otázka	1	2	3	4	5	6	7	8	9	10	Suma	2.5*Sum
1	4	2	4	4	3	3	4	1	4	4	33	82.5
2	3	3	4	4	4	4	4	4	4	4	38	95
3	4	4	3	3	3	4	3	3	3	4	34	85
4	4	4	4	3	4	4	3	3	3	3	35	87.5
5	4	4	4	3	3	3	2	3	3	3	32	80

Tabulka 3. Funkce adaptace uživatelského rozhraní je vypnuto

Uživatel/otázka	1	2	3	4	5	6	7	8	9	10	Suma	2.5*Sum
1	2	3	4	4	4	4	4	4	4	4	37	92.5
2	1	1	2	2	4	3	3	3	3	0	23	57.5
3	2	2	3	3	2	2	3	3	2	2	24	60
4	3	3	3	2	3	3	4	2	4	1	28	70
5	3	3	3	2	3	3	4	3	2	3	29	72.5

Tabulka 4. Funkce adaptace uživatelského rozhraní je zapnuté

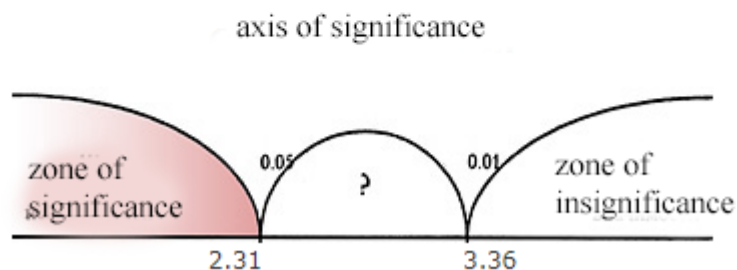
6.8 Vyhodnocení získaných dat

Získaná data byla vyhodnocena pomocí párového t-testu s hladinou významnosti $\alpha=0,05$. Samotné vyhodnocení bylo provedeno použitím programu R [29].

№	Groups					
	G1	G2	G1	G2	G1	G2
1	82.5	92.5	-3.5	22	12.25	484
2	95	57.5	9	-13	81	169
3	85	60	-1	-10.5	1	110.25
4	87.5	70	1.5	0.5	2.25	0.25
5	80	72.5	-6	2	36	4
Sum	430	352.5	0	0	132.5	767.5
Average	86	70.5				

Tabulka 4. Výsledky výpočtu programem R [29].

t_{kp}	
$p \leq 0.05$	$p \leq 0.01$
2.31	3.36

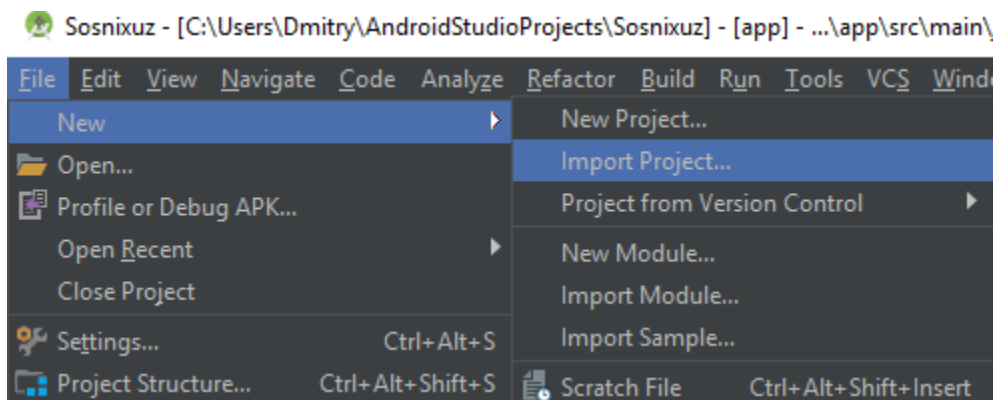


Jelikož vypočtená hodnota p je vyšší než stanovená hladina významnosti, nulovou hypotézu nezamítáme. Na základě získaného vzorku dat lze konstatovat, že testovací aplikace se zapnutou a vypnutou funkcí adaptace uživatelského rozhraní je stejně uživatelsky přívětivá a pohodlně použitelná. Přesnost testu mohla být ovlivněna malým množstvím uživatelů pro testování současně s nedokonalostí testovací aplikace.

7. Instalace

Pro instalaci frameworku stačí mít nainstalované vývojové prostředí pro programování v jazyce Java s podporou Android. Doporučuje se používat Android Studio, které je založeno na IntelliJ IDEA. Android Studio je dostupné na několika operačních systémech, ale návrh a testování bylo provedeno pro operační systém Windows. Instalace se provádí následujícím postupem:

- Po zkopírování frameworku z CD je potřeba spustit vývojové prostředí
- Ve vývojovém prostředí je třeba udělat import projektu pomocí vestavěné funkce vývojového prostředí



Obrázek 7.1. Import projektu

8. Závěr

Cílem této bakalářské práce bylo prostudovat a vylepšit existující nástroj pro adaptace uživatelského rozhraní, který byl vyroben v rámci bakalářské práce, kterou vypracovala T. Titova “Využití aspektově orientovaného přístupu a lokálních dat ve vývoji adaptivního uživatelského rozhraní pro Android“ [40].

Na základě závěru bakalářské práce [40], na kterou se navazuje, byly vyvinuty další případy užití (use-case), rozšířil se počet snímačů mobilního zařízení podporovaných frameworkem .

V první části práce byla prozkoumaná teorie, která je důležitým podkladem pro následující vylepšení frameworku. Byla prozkoumána teorie týkající se snímačů mobilních zařízení, jejich funkcí, a principů, na kterých jsou postavené, a bylo prozkoumáno použití pro adaptaci uživatelského rozhraní.

V následujících částí práce byly popsány jednotlivé případy užití frameworku, a byl zanalyzován přínos a možná realizace v aplikacích. Dále byl popsán návrh a využití nových funkcí frameworku a v dalších kapitolách bylo popsáno jejich testování.

Struktura frameworku, na který práce navazuje, byla zanechaná ve stejné podobě, a to z důvodu dobrého rozpracování a použitého aspektového pohledu programování. Aspektové programování urychluje následující implementaci dalších aspektů a zachovává čitelnost kódu.

9.Literatura

[1] *What is User Interface (UI)?* Webopedia Definition. Webopedia: Online Tech Dictionary for IT Professionals [online].

Dostupné z: https://www.webopedia.com/TERM/U/user_interface.html

[2] Raheel, Saeed. "Improving the user experience using an intelligent Adaptive User Interface in mobile applications." Multidisciplinary Conference on Engineering Technology (IMCET), IEEE International. IEEE, 2016.

[3] *Model-Based Design of Adaptive User Interfaces through Connectors* | SpringerLink. Home - Springer [online]. Copyright © 2017 Springer International Publishing AG. Part of [cit. 08.08.2017]. Dostupné z: https://link.springer.com/chapter/10.1007/978-3-540-39929-2_17

[5] Yuping, Yuping Yang. *Handbook of Research on Mobile Multimedia, Second Edition*. 2009.

[6] Uday Bhaskar , U.S.N Raju2 , P.Govindarajulu , B.V.Ramana Reddy. *Aspects of Content, Context and Adaptation Modeling in Mobile Learning Application Design.*, 2(4):29-37, 2013.

[7] John Heggustuen. *One In Every 5 People In The World Own A Smartphone, One In Every 17 Own A Tablet*. *Bus. Insid.*, 15, 2013. URL <http://www.businessinsider.com/smartphone-and-tablet-penetration-2013-10>.

[8] Context-Aware Computing | *The Encyclopedia of Human-Computer Interaction*, 2nd Ed.. UX Design Courses & Global UX Community | Interaction Design Foundation [online]. Dostupné z: <https://www.interaction-design.org/literature/book/the-encyclopedia-of-human-computer-interaction-2nd-ed/context-aware-computing-context-awareness-context-aware-user-interfaces-and-implicit-interaction>

[9] *Датчики и преобразователи физических величин электронные. Термины и определения* [online]. Copyright © [cit. 08.10.2017]. Dostupné z: <http://vak.ru/pub/gost/gost-r-51086-97.pdf>

[10] *Location management in pervasive systems*. ACM Digital Library [online]. Copyright ©2003 [cit. 08.10.2017]. Dostupné z: <https://dl.acm.org/citation.cfm?id=828003>

[11] *Sensors Overview*[online]. Content is licensed under Creative Commons Attribution 2.5:df, df [cit. 2017-04-24]. Dostupné z: https://developer.android.com/guide/topics/sensors/sensors_overview.html

- [12] *BrightnessGate for the iPhone & Android Smartphones and HDTVs*. DisplayMate Monitor Projector and TV Set Up Calibration and Testing [online]. Dostupné z: http://www.displaymate.com/AutoBrightness_Controls_2.htm
- [13] *ГОСТ Р 53566-2009 | НАЦИОНАЛЬНЫЕ СТАНДАРТЫ*. Стандарты за Декабрь 2017 года | Страница 1 | НАЦИОНАЛЬНЫЕ СТАНДАРТЫ [online]. Copyright © [cit. 08.10.2017]. Dostupné z: <http://protect.gost.ru/document.aspx?control=7&id=176057>
- [14] *Беседка №169. Шумоподавление в смартфонах* — android.mobile-review.com. android.mobile-review.com — Всё про Android [online]. Dostupné z: <http://android.mobile-review.com/articles/52769/>
- [15] MEMS - díl 1. - *Co to je a jak to vypadá ?* | Vývoj.HW.cz. Vývoj.HW.cz | Vše o elektronice a programování [online]. Copyright © 1997 [cit. 09.08.2017]. Dostupné z: <https://vyvoj.hw.cz//clanek/2006111901>
- [16] МЭМС-устройства. *Главная - НПК Фотоника* [online]. Copyright © [cit. 09.08.2017]. Dostupné z: <http://www.npk-photonica.ru/content/products/mems>
- [17] *Как смартфоны чувствуют мир. Часть 1: акселерометры, гироскопы и другие сенсоры* [online]. Dostupné z: <https://ferra.ru/ru/techlife/review/mems-part-1>
- [18] *LIS3MDL - ultra low-power, high performance 3-axis magnetometer* [online]. Copyright © 2017 STMicroelectronics [cit. 09.08.2017]. Dostupné z: <http://www.st.com/en/mems-and-sensors/lis3mdl.html>
- [19] *Гироскоп в смартфоне может использоваться в качестве «скрытого» микрофона. Все самое интересное из мира IT-индустрии* [online]. Copyright © 1997 [cit. 09.08.2017]. Dostupné z: <https://3dnews.ru/900176>
- [20] *Global Positioning Systems, Inertial Navigation, and Integration* - Mohinder S. Grewal, Lawrence R. Weill, Angus P. Andrews - Книhy Google. Книhy Google [online]. Dostupné z: https://books.google.cz/books?id=6P7UNphJ1z8C&redir_esc=y
- [21] *Как это работает?* | Спутниковая система навигации – Винтересном.com. Винтересном.com – Интересные факты, новые открытия, неизведанные рубежи, загадки и теории - только самое интересное [online]. Dostupné z: <https://vinteresnom.com/kak-eto-sdelano/kak-eto-rabotaet-sputnikovaya-sistema-navigacii/>

[22] *Android Architecture and Libraries Every Android Developer Should Know*. SIMPLE DEVELOPER - Making Sense of The Complex Universe [online]. Dostupné z: <http://simpledeveloper.com/android-architecture/>

[23] *Activity* [online]. Content is licensed under Creative Commons Attribution 2.5: df, df [cit. 2017-04-24]. Dostupné z: <https://developer.android.com/reference/android/app/Activity.html>

[24] *Shazam: алгоритмы распознавания музыки, сигнатуры, обработка данных* / Блог компании Wunder Fund / Хабрахабр. Лучшие публикации за сутки / Хабрахабр [online]. Copyright © 2006 [cit. 09.12.2017]. Dostupné z: <https://habrahabr.ru/company/wunderfund/blog/275043/>

[25] *Распознавание речи в Андроид* / PandaCoder [online]. Dostupné z: http://www.pandacoder.com/android_speech_recognition/

[26] *RecognitionListener* [online]. Content is licensed under Creative Commons Attribution 2.5: df, df [cit. 2017-04-24]. Dostupné z: <https://developer.android.com/reference/android/speech/RecognitionListener.html>

[27] *t-критерий Стьюдента. Медицинская статистика* [online]. Dostupné z: http://medstatistic.ru/theory/t_criteria.html

[28] *System Usability Scale* [online]. [cit 2017-12-20] <https://www.usability.gov/how-to-and-tools/methods/system-usability-scale.html>

[29] R: *The R Project for Statistical Computing*. R: *The R Project for Statistical Computing* [online]. Copyright © The R Foundation. [cit. 01.01.2018]. Dostupné z: <https://www.r-project.org/>

[30] HAYES, Nicky. *Základy sociální psychologie*. 1998. ISBN 80-7178-198-3. Kapitola Měření postojů, s. 112.

[31] *HCI Semestrálky* [online]. Copyright © [cit. 01.01.2018]. Dostupné z: <http://hcisemestralky.felk.cvut.cz/system/mems/7623/original/B.pdf>

[32] *Что такое Bluetooth и как им воспользоваться на своём компьютере?* [online]. Copyright © [cit. 01.01.2018]. Dostupné z: <http://procomputer.su/seti-i-svyaz/179-chto-takoe-bluetooth>

[33] Как это работает: Bluetooth [online]. Copyright © [cit. 01.01.2018]. Dostupné z: <https://trashbox.ru/topics/40763/kak-eto-rabotaet-bluetooth>

[34] Six Wi-Fi Interoperability Certifications Awarded By The Wireless Ethernet Compatibility Alliance (WECA) | Wi-Fi Alliance . *Wi-Fi Alliance* [online]. Copyright © 2018 Wi [cit. 01.01.2018]. Dostupné z: <https://www.wi-fi.org/news-events/newsroom/six-wi-fi-interoperability-certifications-awarded-by-the-wireless-ethernet>

[35] IEEE 802.20.3-2010 - IEEE Standard for Minimum Performance Characteristics of IEEE 802.20 Terminals and Base Stations/Access Nodes. *IEEE-SA - The IEEE Standards Association - Home* [online]. Copyright © Copyright 2018 IEEE [cit. 01.12.2017]. Dostupné z: <http://standards.ieee.org/findstds/standard/802.20.3-2010.html>

[36] Předmět IKT | Informační a komunikační technologie [online]. Copyright © [cit. 09.08.2017]. Dostupné z: http://www.gioivt.cz/studium/pocsit_2.pdf

[37] Кое-что о Wi-Fi / Хабрахабр. Лучшие публикации за сутки / Хабрахабр [online]. Copyright © 2006 [cit. 08.09.2017]. Dostupné z: <https://habrahabr.ru/post/117761/>

[38] Разработка NFC приложений для Android / Блог компании Intel / Хабрахабр. *Лучшие публикации за сутки / Хабрахабр* [online]. Copyright © 2006 [cit. 08.09.2017]. Dostupné z: <https://habrahabr.ru/company/intel/blog/194344/>

[39] Что такое NFC и как его готовить / Хабрахабр. Лучшие публикации за сутки / Хабрахабр [online]. Copyright © 2006 cit. 08.09.2017]. Dostupné z: <https://habrahabr.ru/article/272217/>

[40] TITOVA, Tamara. *Využití aspektově orientovaného přístupu a lokálních dat ve vývoji adaptivního uživatelského rozhraní pro Android* Prague, 2016. Diploma thesis. Department of Computer Science, CTU FEE. Leadership J. Šebek.

[41] A comparative study of systems for the design of flexible user interfaces - IOS Press. Home - IOS Press [online]. Copyright ©2015 IOS Press All rights reserved. [cit. 07.08.2017]. Dostupné z: <https://content.iospress.com/articles/journal-of-ambient-intelligence-and-smart-environments/ais370>

10. Obsah přiloženého CD

Framework code.zip.	AUIF framework
source.zip.	obrázky a diagramy
pratadzm2018bach.pdf.	Text bakalářské práce
pratadzm2018.pdf.	Zadání bakalářské práce