

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

Fakulta elektrotechniky  
a komunikačních technologií

SEMESTRÁLNÍ PRÁCE



**VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ**

BRNO UNIVERSITY OF TECHNOLOGY

**FAKULTA ELEKTROTECHNIKY**

**A KOMUNIKAČNÍCH TECHNOLOGIÍ**

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

**ÚSTAV TELEKOMUNIKACÍ**

DEPARTMENT OF TELECOMMUNICATIONS

**MOBILNÍ APLIKACE PRO SUBJEKTIVNÍ MĚŘENÍ KVALITY  
ZÁŽITKU STREAMOVANÉHO VIDEO**

**SEMESTRÁLNÍ PRÁCE**

SEMESTRAL THESIS

**AUTOR PRÁCE**

AUTHOR

**Bc. Pavel Šeda**

**VEDOUCÍ PRÁCE**

SUPERVISOR

**Ing. Dominik Kováč**

**BRNO 2016**



# Semestrální práce

magisterský navazující studijní obor **Telekomunikační a informační technika**

Ústav telekomunikací

**Student:** Bc. Pavel Šeda

**ID:** 154208

**Ročník:** 2

**Akademický rok:** 2016/17

## NÁZEV TÉMATU:

### Mobilní aplikace pro subjektivní měření kvality zážitku streamovaného videa

## POKYNY PRO VYPRACOVÁNÍ:

Práce se bude zabývat subjektivním měřením kvality zážitku streamovaných videí pomocí mobilní aplikace. Teoretická část práce se bude zabývat problematikou kvality zážitku (QoE) a jejich metodami měření. Cílem práce bude vytvoření mobilní aplikace na operačním systému Android, aplikace bude vycházet z dostupného vývojového API YouTube Android Player. Aplikace bude umožňovat uživatelům hodnotit jejich subjektivní dojem o kvalitě streamovaného videa a výsledky se budou ukládat do vzdálené databáze.

## DOPORUČENÁ LITERATURA:

[1] GRIFFITHS, Dawn a David GRIFFITHS. Head First Android Development. O'Reilly Media, 2015. ISBN 1449362184.

[2] ITU-T, "Recommendation P.910: Subjective video quality assessment methods for multimedia applications," April 2008

**Termín zadání:** 19.9.2016

**Termín odevzdání:** 14.12.2016

**Vedoucí práce:** Ing. Dominik Kováč

**Konzultant semestrální práce:**

**doc. Ing. Jiří Mišurec, CSc., předseda oborové rady**

## UPOZORNĚNÍ:

Autor semestrální práce nesmí při vytváření semestrální práce porušit autorská práva třetích osob, zejména nesmí zasahovat nedovoleným způsobem do cizích autorských práv osobnostních a musí si být plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č.40/2009 Sb.

## ABSTRAKT

Semestrální práce se zabývá subjektivním měřením kvality zážitku ze streamovaného videa. Pro tyto potřeby byla vytvořena mobilní aplikace pomocí, které je možné získávat uživatelské hodnocení streamovaných videí. Tato uživatelská hodnocení jsou ukládána do databáze. Na základě výsledků z provedených testů je možné detailněji zkoumat subjektivní hodnocení kvality zážitku.

## KLÍČOVÁ SLOVA

subjektivní Quality of Experience, adaptivní streaming, Java, Android

## ABSTRACT

This thesis is focused on the subjective measurement of the quality of experience on streaming video. For these purposes was created mobile application with which is possible to obtain user ratings of these streaming videos. These user ratings are stored in database. Based on these results from tests it's possible to investigate subjective assessment of quality of experience in more detail.

## KEYWORDS

Subjective Quality of Experience, Adaptive Streaming, Java, Android

ŠEDA, Pavel *Mobilní aplikace pro subjektivní měření kvality zážitku streamovaného videa*: semestrální projekt. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav telekomunikací, 2017. 67 s. Vedoucí práce byl Ing. Dominik Kováč

## PROHLÁŠENÍ

Prohlašuji, že svůj semestrální projekt na téma Mobilní aplikace pro subjektivní měření kvality zážitku streamovaného videa jsem vypracoval(a) samostatně pod vedením vedoucího semestrálního projektu a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor(ka) uvedeného semestrálního projektu dále prohlašuji, že v souvislosti s vytvořením tohoto semestrálního projektu jsem neporušil(a) autorská práva třetích osob, zejména jsem nezasáhl(a) nedovoleným způsobem do cizích autorských práv osobnostních a/nebo majetkových a jsem si plně vědom(a) následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon), ve znění pozdějších předpisů, včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

Brno .....

.....

podpis autora(-ky)

## PODĚKOVÁNÍ

Rád bych poděkoval Ing. Dominiku Kováčovi za seznámení s problematikou subjektivního měření kvality zážitku streamovaných videí, cenné rady, věcné připomínky, poskytnuté množství materiálů a vstřícný přístup při konzultacích a vypracování semestrální práce.

Brno .....

.....

podpis autora(-ky)

# OBSAH

<b>Úvod</b>	<b>12</b>
<b>1 Úvod do problematiky streamingu</b>	<b>13</b>
1.1 Streaming	13
1.2 Služby pro streamování videa	13
1.3 Techniky streamování videa	14
<b>2 Quality of Experience</b>	<b>17</b>
2.1 Faktory ovlivňující QoE	17
2.2 Objektivní QoE	18
2.3 Subjektivní QoE	20
2.3.1 Metody měření subjektivního QoE	20
2.3.2 Metody korelace získaných výsledků	26
<b>3 Vývoj aplikace</b>	<b>29</b>
3.1 Analýza projektu	29
3.1.1 Myšlenková mapa	29
3.1.2 Diagram případů užití	30
3.1.3 Stavový diagram	32
3.1.4 Diagram tříd	32
3.1.5 Sekvenční diagram	33
3.1.6 Entitně relační model	34
3.1.7 Gantův diagram	35
3.2 Použité technologie	35
3.2.1 Git	35
3.2.2 Java	36
3.2.3 Android a Android Studio	37
3.2.4 YouTube API	39
3.2.5 Gradle	40
3.2.6 MySQL a MySQL Workbench	41
3.3 Implementace aplikace	41
3.3.1 Prerekvizity k vytvoření Android projektu	41
3.3.2 Vytvoření Android projektu	42
3.3.3 Kompilace Android aplikace	47
3.3.4 Integrace YouTube Android API do projektu	48
3.3.5 Úprava aplikace pro potřeby subjektivního testování	55
<b>4 Závěr</b>	<b>58</b>

<b>Literatura</b>	<b>59</b>
<b>Seznam symbolů, veličin a zkratk</b>	<b>63</b>
<b>Seznam příloh</b>	<b>65</b>
<b>A MySQL databáze</b>	<b>66</b>
A.1 Návrh MySQL databáze na základě ERD v MySQLWorkbench . . .	66
A.2 MySQL dotazy k vytvoření databáze . . . . .	66



# SEZNAM OBRÁZKŮ

1.1	Jeden z možných průběhů změny bitrate během streamingu. . . . .	15
2.1	Faktory ovlivňující QoE [21]. . . . .	18
2.2	Stimulovaná prezentace ACR metody [16]. . . . .	22
2.3	Stimulovaná prezentace DCR metody [16]. . . . .	24
2.4	Stimulovaná prezentace PC metody [16]. . . . .	25
3.1	Myšlenková mapa . . . . .	30
3.2	Diagram případů užití znázorňující vytvářenou mobilní aplikaci . . .	31
3.3	Diagram stavů . . . . .	32
3.4	Diagram tříd mobilní aplikace . . . . .	33
3.5	Sekvenční diagram znázorňující použití YouTube Android API . . . .	34
3.6	Entitně relační diagram znázorňující fyzickou podobu databáze. . . .	34
3.7	Gantův diagram znázorňující náplánování posloupnosti činností v čase. .	35
3.8	Životní cyklus aktivity . . . . .	38
3.9	Ukázka SDK Manageru . . . . .	42
3.10	Vytváření nového projektu v Android Studiu . . . . .	43
3.11	Výběr cílových zařízení . . . . .	44
3.12	Výběr Android aktivity do projektu . . . . .	44
3.13	Dokončení vytvoření nového Android projektu . . . . .	45
3.14	Struktura nově vytvořeného projektu ve vývojovém prostředí . . . . .	46
3.15	Výběr cílového zařízení pro kompilaci emulátor/připojené zařízení. . .	47
3.16	Nastavení cílového zařízení na testování . . . . .	48
3.17	Generování podepsaného .apk souboru nastavení část 1. . . . .	49
3.18	Generování podepsaného .apk souboru nastavení část 2. . . . .	50
3.19	Vygenerovaný SHA1 soubor pro podepsaný .apk soubor . . . . .	51
3.20	Vložený SHA1 heše a package name . . . . .	52
3.21	Znázornění využití API klíče ke Google službě . . . . .	52
3.22	Vložení YouTube API do layout souboru . . . . .	53
3.23	Screen save z mobilní aplikace po integraci YouTube API . . . . .	55
3.24	Aplikace při použití WebView místo YouTube Android API. . . . .	56
A.1	Databáze vytvořená v programu MySQL Workbench. . . . .	66

# SEZNAM TABULEK

1.1	Srovnání HTTP a UDP streamingu . . . . .	14
2.1	Pětiúrovňová stupnice pro hodnocení kvality metodou ACR . . . . .	23
2.2	Pětiúrovňová stupnice pro hodnocení zhoršení metodou DCR . . . . .	25

# SEZNAM ZDROJOVÝCH KÓDŮ

1	Finální vyřazující kritéria pro pozorovatele . . . . .	27
---	--	----

# ÚVOD

Kvalita zážitku (Quality of Experience – QoE) se v poslední době stalo hlavní paradigma, které klade důraz na uživatelské chápání kvality komunikačních kanálů a služeb.

Quality of Experience původně patřilo do problematiky kvality služeb (Quality of Service – QoS), která se zabývá spíše výkonnostními charakteristikami služeb jako je latence, míra chybovosti, šířka pásma, pravděpodobnost výpadku atd. QoS se tedy nezabývá koncovými uživateli, ale spíše výkonnostními charakteristikami poskytovaných služeb.

S nárůstem množství služeb a uživatelů se stalo velmi žádoucí pro poskytovatele zkoumat služby nejen z pohledu QoS, ale i QoE, protože je vhodné, aby se kromě výkonnostních charakteristik formoval i celkový uživatelský dojem z jimi poskytovaných služeb.

Cílem této práce je seznámit se s vývojem mobilních aplikací na platformě Android, dostupné API YouTube Android Player a problematikou subjektivního QoE. Po počátečním prostudování bude vytvořena mobilní aplikace, ve které si bude možné spustit YouTube video, které bude mít pozměněné některé faktory jako je rozlišení nebo typ streamingu a poté bude moci uživatel ohodnotit kvalitu zážitku takto pozměněného videa. Na základě dat z aplikace od uživatelů se bude dát lépe zhodnotit kvalita zážitku v rámci daného pozměňujícího faktoru na videu.

# 1 ÚVOD DO PROBLEMATIKY STREAMINGU

Kapitola úvod do problematiky streamingu je rozdělena na sekce streaming, služby pro streamování videa a adaptivní streamování. Nejprve je uveden krátký popis, co je to streaming, poté jsou představeny nejpoužívanější streamingové služby a v poslední části jsou objasněny techniky streamování se zaměřením na techniku adaptivního streamingu.

## 1.1 Streaming

Streaming je technologie přenosu audiovizuálního materiálu ke koncovým uživatelům přes komunikační kanál, jako je například internet nebo vyhrazená IP (Internet Protocol) síť, která je řízena poskytovatelem služeb. Přenos může být v reálném čase (televize, rádio) nebo systémem video na vyžádání (video on demand), což zprostředkovává například služba YouTube. Při streamování videa musíme mít k dispozici také streamovací server, který zajišťuje komunikaci s cílovými uživateli [1]. Služby, které poskytují streamovací servery jsou popsány v sekci 1.2.

## 1.2 Služby pro streamování videa

V současné době existuje mnoho služeb pro streamování videa, které nabízejí streamování on demand, na základě předdefinovaných playlistů nebo například podle žánrů. Mezi nejznámější streamovací služby patří:

- [Netflix](#),
- [Amazon Prime Instant Video](#),
- [YouTube](#).

Například YouTube je jedna z nejpobulárnějších aplikací dnešního internetu.

YouTube dnes představuje více než 30 % provozu na celém internetu. YouTube je streamovací platforma, která nabízí hlavně malé až středně velké videoklipy. Video je zašifrované dle H.264/MPEG-4 kódování (Advanced Video Coding (AVC)) jako výchozí kompresní formát videa. YouTube používá HTTP (Hypertext Transfer Protocol) jako streamovací technologii. Pro úplnost je zde uvedeno v tabulce 1.1 porovnání HTTP streamování s často vyskytujícím se UDP (User Datagram Protocol) streamováním [2].

Tab. 1.1: Srovnání HTTP a UDP streamingu

HTTP	UDP
spolehlivý přenos	nespolehlivý přenos
kvalita videa není ovlivněna	kvalita videa je ovlivněna
většina poruch má dočasný charakter	poruchy způsobují vizuální degradaci

Klient stáhne data videa přes HTTP spojení a současně spustí video ještě předtím, než jsou data plně stažena. Klientská aplikace je předkompilovaná s využitím Adobe Flash player assembly, která běží u klienta ve webovém prohlížeči. Stažená data jsou uložena v dočasných souborech, které slouží jako buffer pro spuštění určité části videa. YouTube provádí před-bufferování, což znamená, že klientovi se spustí video potom, co určitá část videa je dostupná v bufferu. Čas od požadavku na spuštění videa do doby, než video začne hrát (úspěšně se naplní buffer), se nazývá počáteční zpoždění (startup delay). Když je video spuštěné, tak server znovu naplňuje buffer periodicky přenášenými bloky s daty videa ke klientovi. Aktuálně příchozí data ke klientovi jsou regulována TCP (Transmission Control Protocol) protokolem a také dostupnou šířkou pásma [3].

Pro služby, které poskytují streamování videa, je rovněž podstatné, jaké používají techniky pro streamování videa. Tyto techniky jsou detailněji popsány v sekci 1.3. Volba správné techniky streamování je jeden z klíčových faktorů, které vedou k vyšší kvalitě zážitku ze streamovaného videa.

## 1.3 Techniky streamování videa

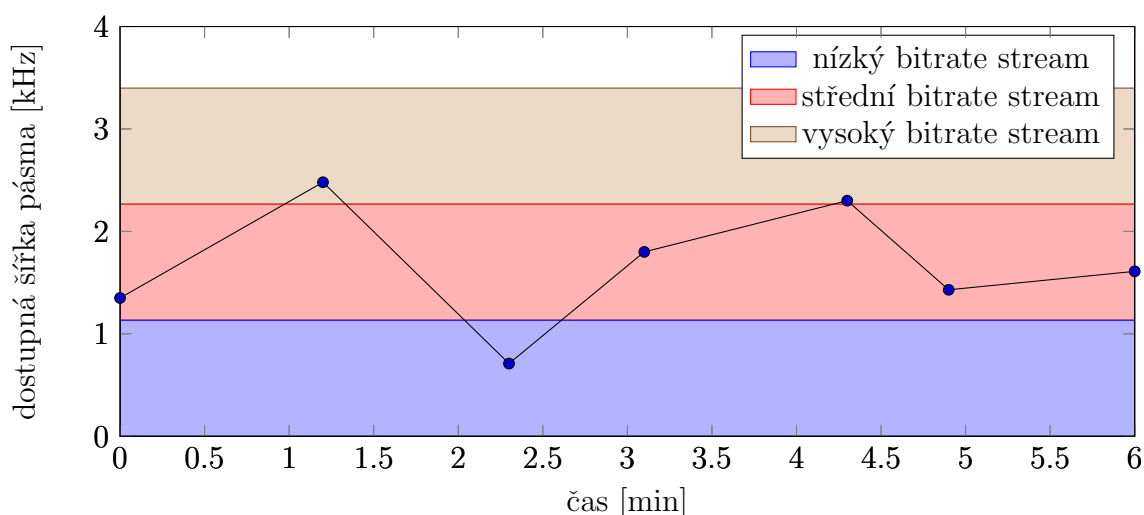
Služby pro streamování videa používají streamovací techniky, které se dají rozdělit na 3 základní kategorie:

- streaming,
- adaptivní streamování,
- progresivní stažení.

Princip techniky streamingu funguje tak, že když uživatel klikne na webové stránce na play u videa, které si chce spustit, tak video začne okamžitě hrát a hraje hladce až do konce. Aby to bylo možné, musí být rychlost přenosu dat kódovaného souboru menší než kapacita pásma vzdáleného diváka, jinak se video často zastavuje [4].

Jako prevence proti zastavování videa se v současné době velmi často používá technika adaptivního streamování, která řeší několik kritických aspektů při přenosu dat. Jedním z nejvýznamnějších je, že ze stejného zdrojového souboru vytváří

několik souborů, které jsou distribuovány uživatelům různých zařízení s různou přenosovou rychlostí. Neméně důležitá je adaptivní distribuce souborů, to znamená, že mění rychlost přenosu dat nebo přenosovou rychlost videa v závislosti na rychlosti připojení jednotlivého uživatele. Výhodou je transparentnost. Uživatel jedním klikem vybere předem bitrate a kvalitu videa, čímž se na pozadí nastaví přepínání streamování. Pokud video zaplní vyrovnávací paměť a využití procesoru je nízké, technologie adaptivního streamování může přepnout na vyšší kvalitu přenosu a zvýšit tak kvalitu zážitku. Klesne-li vyrovnávací paměť pod určitou úroveň nebo využití Central Processing Unit (CPU) přesáhne mezní hranice, může tato technologie naopak přepnout na nižší kvalitu streamu. Jeden z možných průběhů změny bitrate je vidět na obrázku 1.1, kdy mezi body 2 a 2.5 na ose znázorňující čas vidíme přetížení sítě, které vyvolalo právě snížení bitrate.



Obr. 1.1: Jeden z možných průběhů změny bitrate během streamingu.

Klíčovým rozdílem v implementaci napříč různými technologiemi je streamovací server. Některé technologie vyžadují streamovací server a neustálou komunikaci mezi serverem a přehrávači. Pokud je požadováno přepnutí streamu, server tento požadavek zpracovává odesláním jiného streamu. Ostatní technologie fungují bez streamovacího serveru. Streamy různých kvalit jsou zveřejněny na různých adresách webového serveru nebo několika webových serverech. Přehrávač sleduje provozní heuristiky, jako je využití procesoru nebo stav vyrovnávací paměti a rozhoduje, kdy je nutné provést změnu a začít načítat data z jiného streamu.

Poskytovatelé služeb pro adaptivní streaming přehrávače spadají do tří hlavních kategorií: technologie vývojáře, poskytovatelé služeb a standard-based technologies. K prominentním technologickým poskytovatelům se řadí:

- Adobe s Flash-based Dynamic Streaming,

- Apple s HTTP Live Streaming (HLS),
- Microsoft s technologií Smooth Streaming pro Silverlight.

Několik dalších možností založených na WebM-based HTML5 je dostupných nebo se vyvíjejí a to včetně technologií od Anevia and Quavlive. Primárním poskytovatelem služeb je hlavně Akamai s jejich Akamai HD Network, což je platforma která může být použita na iOS zařízeních, Flash a Silverlight klientech. Některé společnosti, jako například Netflix, vyvinuly své vlastní technologie adaptivního streamování pro interní použití. Standard-based technologie zahrnují Scalable Video Coding (SVC) - rozšíření specifikace H.264. Kromě toho předložil Apple HTTP Live Streaming protocol to the Internet Engineering Task Force (IETF), který pracuje na základě standardizace procesů [5][6].

Technika progresivního stažení je založena na stažení videa z HTTP web serveru upřednostněného před streaming serverem. Ve většině případů video doručené touto technikou je uloženo na uživatelově pevném disku, jakmile je video přijato. Poté je video přehráváno přímo z pevného disku. V kontrastu se streaming videem, které obvykle není uloženo lokálně, takže pokud uživatel nemůže načíst video a přehrát ho v reálném čase, tak si ho nemůže přehrát vůbec [4].

Výše uvedené techniky streamingu ovlivňují kvalitu zážitku uživatele, která vzhledem k velkému rozmachu streamovaných videí vede k zkoumání problematiky Quality of Experience (QoE), která je detailněji popsána v kapitole 2.



## 2 QUALITY OF EXPERIENCE

Quality of Experience (QoE) se v poslední době stalo hlavním paradigmatem, které klade důraz na uživatelské chápání kvality komunikačních kanálů a služeb. Pomocí QoE jsme schopni popsat uživatelské vnímání a výslednou spokojenost se službami v komunikační síti. Toho často využívají poskytovatelé internetu a operátoři, kteří věnují stále více pozornosti modelování a vyhodnocování QoE, aby mohli dosáhnout konkurenční výhody [7]. Standardně se QoE dělí na 2 přístupy:

- objektivní,
- subjektivní.

V případě subjektivních metod se kvalita služeb hodnotí pozorovateli (čili lidmi) a v případě objektivních metod se vyhodnocuje kvalita služeb strojově. Tyto přístupy jsou popsány detailněji v sekcích níže.

Výzkum problematiky QoE vyšel původně z Quality of Service (QoS), která se zabývá spíše výkonnostními faktory sítě jako jsou:

- šířka pásma,
- latence,
- míra chybovosti,
- provozuschopnost / pravděpodobnost výpadku aj.

QoS narozdíl od QoE se tedy nezabývá koncovými uživateli, ale spíše výkonnostními charakteristikami služeb, které jim poskytují. QoS je zvláště důležité pro poskytovatele internetových služeb, kteří nabízejí velké množství provozně náročného obsahu jako je VoIP, streamování médií, online hraní, videokonference, čili cokoliv, co je závislé na konstantním a stabilním rychlém připojení.

Z popisu výše vidíme, že problematiky QoS a QoE spolu úzce korelují. Pro poskytovatele služeb je tedy důležité, aby měli výkonnostní charakteristiky sítě na vysoké úrovni, ale neméně důležitá je také kvalita zážitku, která formuje celkový uživatelský dojem, z jimi poskytovaných služeb [8] [9]. Tato práce je zaměřena na problematiku QoE a zejména na problematiku subjektivního QoE, která je popsána v sekcích níže.

### 2.1 Faktory ovlivňující QoE

Vnímání kvality streamovaného videa pro uživatele na internetu je ovlivněno celou řadou faktorů. Tyto ovlivňující faktory jsou rozděleny do čtyř základních kategorií:

- kontext
- uživatel
- systém
- obsah

Kontextová úroveň zvažuje aspekty jako je prostředí, ve kterém uživatel konzumuje danou službu. Uživatelská úroveň zahrnuje psychologické faktory jako jsou očekávání uživatele, paměť, historie používání aplikace. Technické ovlivňující faktory jsou abstraktní na systémové úrovni. Pokrývají vlivy na přenosové sítě, zařízení, ale také implementace aplikace samotné, která může mít jiné strategie na bufferování videa. Pro doručení videa, obsahová úroveň zkoumá video kodeky, formát, rozlišení, ale také délku videa, obsah videa a typ videa [21]. Všechny výše zmíněné kategorie ovlivňujících faktorů jsou přehledně znázorněny na obrázku 2.1.



Obr. 2.1: Faktory ovlivňující QoE [21].

## 2.2 Objektivní QoE

Objektivní QoE se vyhodnocuje zpravidla strojově, což vede na rozdíl od subjektivního QoE k menší náročnosti na lidské a finanční zdroje, proto se někdy upřednostňují objektivní metody před subjektivními. Objektivní metody měření kvality se dají rozčlenit do 3 skupin:

- plná reference,
- redukovaná reference,

- bez reference.

Některé z těchto metod jsou založeny na zahrnutí původního referenčního signálu v měření.

Plná reference vyžaduje původní materiál v celku. Operují na základě porovnávání původního s pozměněným či narušeným signálem k vypočítání degradace. Výpočty této degradace se pohybují od jednoduchých algoritmů typu odhadu chyb v signálu k velmi komplexním.

Redukovaná reference používá část původního videa pro výpočet degradace. Jsou vhodné pro situace, kde je původní obsah těžké uložit nebo přenést na místo určení nebo výpočetní výkon je limitován.

Objektivní metoda bez reference nepoužívá žádnou část původního obsahu. Nezávisí na porovnání s původním signálem, ale na měření externích faktorů k vytvoření modelu QoE. Modely bez reference jsou obvykle určeny pro specifické aplikace a nejsou vhodné k obecnému použití, ale vyžadují nejméně zdrojů a jsou užitečné v případě, kdy původní obsah není dostupný [10]. Mezi metody objektivního měření QoE patří:

- PSNR (Peak Signal to Noise Ratio),
- VQM (Video Quality Metric),
- MPQM (Moving Picture Quality Metric),
- SSIM (Structural Similarity Index),
- NQM (Noise Quality Measure).

Vzhledem k tomu, že práce není zaměřena na objektivní metody QoE, tak zde bude popsána pouze metoda PSNR, která patří mezi jedny z nejrozšířenějších.

Metoda PSNR je navržena pro obecnější použití, protože dokáže detekovat chyby v jakémkoliv typu signálu. Je často používána pro hodnocení kvality obrazu a hodnocení kvality videa, díky své jednoduchosti. PSNR je odvozena nastavováním střední mocniny chyby (MSE), ve vztahu k maximální možné hodnotě jasu (například pro 8-bitové hodnoty je to  $2^8 - 1 = 255$ ) jako v následující rovnici:

$$MSE = \frac{\sum_{i=1}^M \sum_{j=1}^N [f(i, j) - F(i, j)]^2}{M \times N} \quad (2.1)$$

$$PSNR = 20 \cdot \log_{10} \left( \frac{255}{\sqrt{MSE}} \right) \quad (2.2)$$

kde:

$f(i, j) :$	je původní signál na pixelu $(i, j)$
$F(i, j) :$	je rekonstruovaný signál
$M \times N :$	je velikost obrázku

Výsledkem je číslo v decibelech v rozmezí od 30 do 40 decibelů pro středně až vysoce kvalitní videa [10][11][12].

## 2.3 Subjektivní QoE

Metody subjektivního QoE jsou založeny na hodnocení služeb samotnými pozorovateli, buď může jít o laboratorní testy, kde jsou pozorovatelé (lidé), kteří hodnotí jednotlivé představené služby nebo může jít o hodnocení v rámci takzvaného crowdsourcingu viz sekce 2.3.1. Výsledky jsou poté vyhodnoceny (popřípadě ještě dodatečně korelovány viz sekce 2.3.2) a je z nich sestaveno MOS<sup>1</sup> (metody měření subjektivního QoE jsou detailněji popsány v sekci 2.3.1).

Nevýhodou laboratorních testů je, že jsou zpravidla náročné na finanční, časové a lidské zdroje. Mezi nevýhody také patří, že pozorovatel se může účastnit testovací události pouze jednou, protože pokud by se testů účastnil opakovaně, tak by se zkresloval výsledek v rámci procesu ("učení se").

Výhodou subjektivního QoE je, že pokud jsou dodrženy veškeré podmínky, tak je měření subjektivními metodami nejpřesnější ze všech současných způsobů měření, to je důvodem, proč se tyto metody často používají jako referenční metody u nových zařízení, multimediálních kodeků atd. [13][14][16].

### 2.3.1 Metody měření subjektivního QoE

V současné době bylo publikováno poměrně velké množství metod měření subjektivního QoE. V této práci budou nastíněny metody, které patří mezi nejpoužívanější měřicí metody subjektivního QoE současnosti. Jde o metody Crowdsourcing, Absolute Category Rating (ACR), Absolute Category Rating with hidden reference (ACR-HR), Degradation Category Rating (DCR), Pair comparison method (PC), The Double-Stimulus Continuous Quality-Scale method (DSCQS). U všech těchto metod je vysvětlen jejich základní princip [15] [16].

---

<sup>1</sup>Mean opinion score (MOS) se vyhodnocuje na základě výsledků hodnocení služeb při použití nejčastěji 5úrovňových stupnic, které jsou uvedeny v metodách ACR konkrétně tabulka 2.1 a DCR tabulka 2.3.

## Crowdsourcing

Crowdsourcing je poměrně nová metoda používaná k získávání dat od velkého množství uživatelů, kteří nejsou nuceni se účastnit laboratorních testů, jako je tomu v případě metod ACR, ACR-HR, DCR, PC, DSCQS aj. Crowdsourcing je založen na drobných operacích prováděných prostřednictvím služeb na internetu, které jsou snadno splnitelné:

- napsat článek / blok / komentář,
- umístit banner nebo link na webovou stránku,
- nahrát/stáhnout média,
- využívat služby sociálních sítí,
- instalovat software,
- provést YouTube Stalling test.

Mezi výhody crowdsourcingu oproti běžným laboratorním testům pro získávání dat pro studie subjektivního QoE patří široké spektrum uživatelů, rozmanitost uživatelů (např. z jiných geografických lokací nebo výrazně rozdílný věk atd.), uživatelské studie mohou být provedeny v krátkém čase, nízké náklady v porovnání s laboratorními testy a testování QoE pro internetové aplikace s reálnými nastaveními.

Crowdsourcing se v praxi využívá pro širokou škálu funkcionalit:

- online zlepšování pověsti / dobrého jména dané služby,
  - hodnocení pro videa nebo články,
  - pozitivní hodnocení profilu na základě hlasování,
- hodnocení obrázků,
- získávání nových uživatelů / využívání nových platforem,
- online reklamy / optimalizace vyhledávacích nástrojů,
  - umísťování informací na twitter,
  - tvorba zpětných odkazů,
- uživatelské dotazníky pro marketing / QoE studie,
- vytváření obsahů,
  - příspěvky na fórech či blozích,

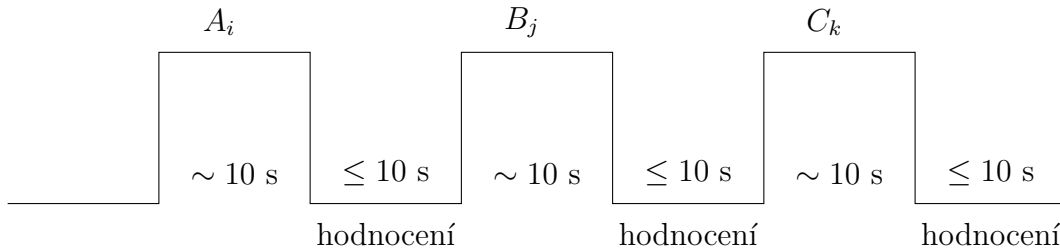
– články,

- výzkum a vývoj.

Z bodů výše vidíme, že crowdsourcing je často používaná metoda skýtající mnoho výhod, avšak k jejím rizikům patří možnost ovlivňování komunity, jednostranná zaměřenost komunity atd., což může vést k irelevantním výsledkům [2] [20]. V takovém případě nepřinese Crowdsourcing názorový průměr, ale názory jednotlivců, proto je vhodné provádět i laboratorní testy, které jsou popsány níže.

### Absolute Category Rating (ACR)

Absolute Category Rating (ACR) je metoda, při které jsou testované sekvence prezentovány lidskému subjektu jedna po druhé a poté jsou sekvence nezávisle hodnoceny na stupnici přiřazené dané kategorii (tato metoda se také nazývá metoda jediného stimulu). Subjekty jsou požádány, aby po zhlédnutí nebo poslechu ohodnotily kvalitu prezentace a to na základě očekávané úrovně kvality pro danou sekvenci. Časový průběh testování metodou ACR je vidět na obrázku 2.2. Pokud je zvoleno konstantní hodnocení pak by hodnotící čas měl být menší nebo rovno 10 s. Doba prezentace sekvence může být redukována nebo zvyšována v závislosti na obsahu testovaného materiálu.



Obr. 2.2: Stimulovaná prezentace ACR metody [16].

kde:

$A_i$ :	Sekvence $A$ za testovacích podmínek $i$
$B_j$ :	Sekvence $B$ za testovacích podmínek $j$
$C_k$ :	Sekvence $C$ za testovacích podmínek $k$

Nejčastěji se používá pětiúrovňová stupnice hodnocení pro celkovou kvalitu. Tato stupnice je znázorněna v tabulce 2.1.

Tab. 2.1: Pětiúrovňová stupnice pro hodnocení kvality metodou ACR

5	Excellent (vynikající)
4	Good (dobrá)
3	Fair (průměrná)
2	Poor (špatná)
1	Bad (mizerná)

Je-li požadováno vyšší rozlišení hodnocení, pak se používají 9 nebo 11úrovňové hodnocení [15] [16].

### Absolute Category Rating with hidden reference (ACR-HR)

Tato metoda je založena na principu posuzování, kdy je subjektu předkládán jeden vzorek, který je následně nezávisle hodnocen dle dané stupnice. Předložená testovaná sekvence musí zahrnovat i referenční verzi každé sekvence. Toto je nazýváno hidden reference condition. Při analýze dat bude hodnocena kvalita mezi každou testovací sekvencí a jí odpovídající referenční sekvencí. Tato metoda je známá jako hidden reference. Metoda určuje, že subjekt hodnotí kvalitu prezentované sekvence po každé prezentaci.

Je-li použit konstantní hodnotící čas (několik subjektů se kouká současně), pak by čas pro hodnocení měl být menší nebo roven 10 s. Čas prezentace může být zkrácen nebo prodloužen a to na základě obsahu testovaného materiálu. Pro tento typ hodnocení by měla být použita 5stupňová stupnice viz tabulka 2.1. DV za použití následující rovnice:

$$DV(PVS) = V(PVS) - V(REF) + 5 \quad (2.3)$$

kde:  $V$  – ACR hodnocení subjektu

V této rovnici odpovídá DV hodnota 5 vznikající kvalitě a DV hodnota 1 odpovídá špatné kvalitě. Jakákoliv hodnota vyšší než 5, (když hodnocená sekvence je hodnocena lepší kvalitou než skrytá referenční sekvence), bude obecně považována za platnou. Alternativně může být použita 2point crushing function k zabránění těmto individuálním ACR-HR hodnocením (DV) neoprávněně ovlivňovat mean opinion score:

$$Crushed_{DV} = (7 \cdot DV) / (2 + DV) \text{ kde } DV > 5 \quad (2.4)$$

Je-li požadováno vyšší rozlišení, pak je použita 9stupňová stupnice ACR.

Pro metodu ACR-HR se získá potřebný počet replikací opakováním stejné zkušební podmínky v různých časových intervalech během trvání testu. Tato metoda by měla

Metoda ACR-HR nemusí být vhodná pro analýzu neobvyklých zhoršení kvality vyskytujících se v první a poslední sekundě sekvence. Neobeznámenost subjektu s referenční sekvencí může způsobit, že jinak zřejmá poškození budou přehlédnuta (sekvence se zastaví těsně před koncem, subjekt nemusí být schopen rozeznat zda se jedná o záměr nebo chybu sítě).

Differential viewer scores (DV) jsou kalkulovány na subjekt a zpracovanou videosekvenci [16].

V metodě Degradation Category Rating (DCR) jsou testované sekvence uvedeny v párech. První stimul v každé dvojici je vždy zdrojový bez jakýchkoliv vad. Druhý z páru je ze stejného zdroje, ale zhoršen zkušebními podmínkami (tato metoda se také nazývá Double Stimulus Impairment Scale (DSIS)). Časový průběh testování metodou DCR je vidět na obrázku 2.3.



kde:

- $A_i$  :                      Sekvence  $A$  za testovacích podmínek  $i$

24



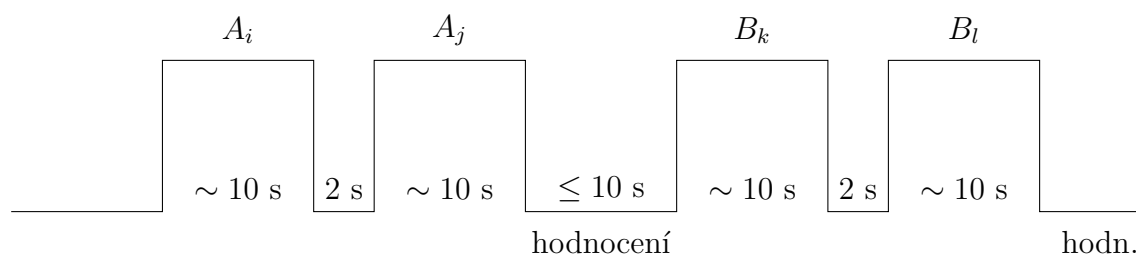
parametry. Prezentační čas může být redukován nebo zvýšen v závislosti na obsahu testovaného materiálu stejně jako u metody ACR [16].

Tab. 2.2: Pětiúrovňová stupnice pro hodnocení zhoršení metodou DCR

5	Imperceptible (nepatrné)
4	Perceptible but no annoying (znatelné, ale ne nepříjemné)
3	Slightly annoying (mírně nepříjemné)
2	Annoying (nepříjemné)
1	Very annoying (velmi nepříjemné)

### Pair comparison method (PC)

Metoda pair comparison zahrnuje, že testované sekvence jsou prezentovány v párech, sestavených ze stejné sekvence prezentované nejprve jedním testovaným systémem a následně dalším systémem. Testované systémy (A, B, C) jsou zpravidla kombinovány ze všech možných systémů  $n(n - 1)$  kombinací AB, BA, CA atd. Takto by všechny testované páry měly být prezentovány v obou možných pořadích (AB, BA). Po každém páru se hodnotí, který element dvojice je preferován v rámci zkušebního scénáře. Je-li použit konstantní hodnotící čas (několik subjektů se kouká současně), pak by čas pro hodnocení měl být menší nebo roven 10 s. Čas prezentace může být zkrácen nebo prodloužen a to na základě obsahu testovaného materiálu. Časový průběh testování metodou ACR je vidět na obrázku 2.4.



Obr. 2.4: Stimulovaná prezentace PC metody [16].

Použije-li se menší rozlišení (CIF, QCIF, SIF), je vhodné zobrazit každý pár sekvencí současně na stejném monitoru. Pro PC metodu není obecně stanoveno počet opakování, a to protože metoda sama o sobě obsahuje opakované prezentace sekvencí se stejnými podmínkami, ačkoliv v různých párech. Různé variace metody PC využívají kategorické měřítko pro další měření rozdílů mezi páry sekvencí [16].

## The Double-Stimulus Continuous Quality-Scale method (DSCQS)

Metoda DSCQS je užitečná, není-li možné zajistit zkušební podmínky zahrnující celé spektrum dostupné kvality. Tato metoda může být použita například v síti, která zaručuje určitou úroveň QoS, například, když MOS je větší než 3 na stupnici s rozsahem 5 bodů. Stejně jako v metodě DCR jsou i zde sekvence uvedeny ve dvojicích: referenční a zhoršená sekvence. Subjekty jsou žádány aby posoudily kvalitu obou sekvencí v páru (nikoli však hodnotit zhoršenou sekvenci vzhledem k referenční jako je tomu u DCR).

Nepozměněná z nich slouží jako referenční, ale pozorovateli není řečeno, která z nich slouží jako referenční. V sérii testů se náhodně mění pozice referenčního videa. Pozorovatelé jsou požádáni, aby ohodnotili celkovou kvalitu každé testované sekvence vložením hodnocení dle hodnotící stupnice (nejčastěji 5úrovňové) [16].

### 2.3.2 Metody korelace získaných výsledků

Během testování se získává velké množství dat, které zpravidla obsahuje i určitý zlomek šumových dat, a ta mohou mít za následek zkreslení výsledků a tím negativně ovlivnit vzešlé závěry. Pro jejich redukci je vhodné zařadit některou z metod korelace výsledků, která by měla pomoci odstranit tato šumová data. Mezi nejpoužívanější metody korelace patří Pearsonova korelace a Spearmanova korelace, které jsou popsány v sekcích níže.

#### Pearson korelace

Pro použití Personovy korelace se předpokládá, že vztah mezi rozsahem kvality a rozsahem skóre z hodnocení od pozorovatelů je lineární. Hlavním cílem je jednoduchou metodou ověřit, jestli skóre jednoho pozorovatele je konzistentní se skóre ostatních pozorovatelů z testovací události [17].

$$r(x, y) = \frac{\sum_{i=1}^n x_i y_i - \frac{\left(\sum_{i=1}^n x_i\right) \cdot \left(\sum_{i=1}^n y_i\right)}{n}}{\sqrt{\left(\sum_{i=1}^n x_i^2 - \frac{\left(\sum_{i=1}^n x_i\right)^2}{n}\right) \cdot \left(\sum_{i=1}^n y_i^2 - \frac{\left(\sum_{i=1}^n y_i\right)^2}{n}\right)}} \quad (2.5)$$

kde:

- $x_i$  : průměrné skóre od pozorovatelů z trojice (algo, bitrate, scéna)
- $y_i$  : skóre individuálního pozorovatele ze stejné trojice
- $n$  : (počet algo)  $\times$  (počet scén)
- $i$  : druh kodeku, bitrate, číslo scény [17][18]

### Spearman rank korelace

Spearman rank korelace může být na rozdíl od Pearsonovy korelace použita, i když vztah mezi rozsahem kvality a rozsahem skóre pozorovatelů není lineární<sup>2</sup> [17].

$$r(x, y) = \left( 1 - \frac{6 \cdot \sum_{i=1}^n [R(x_i) - R(y_i)]^2}{n^3 - n} \right) \quad (2.6)$$

kde:

- $x_i$  : průměrné skóre od pozorovatelů z trojice (algo, bitrate, scene)
- $y_i$  : skóre individuálního pozorovatele ze stejné trojice
- $n$  : (počet algo)  $\times$  (počet scén)
- $R(x_i$  nebo  $y_i)$  : sestupné pořadí
- $i$  : druh kodeku, bitrate, číslo scény [17][19]

### Finální kritéria pro vyřazení pozorovatele z testu

Spearman rank korelace a Pearson korelace jsou použity pro odstranění hodnocení pozorovatele dle následujícího pseudókódu.

---

**Algorithm 1** Finální vyřazující kritéria pro pozorovatele

---

```
if [mean(r) - sdt(r)] > max korelační práh (MTC) then
    práh odmítnutí  $\leftarrow$  max korelační práh
else
    práh odmítnutí  $\leftarrow$  mean(r) - sdt(r)
end if
if [r(pozorovatel)] > práh odmítnutí then
    pozorovatel "i" není vyřazen z testu
else
    pozorovatel "i" je vyřazen z testu
end if
```

---

<sup>2</sup>Obecně dává Pearson korelace velmi podobné výsledky jako Spearman rank korelace

kde:

r :	min(Pearson korelace, Spearman rank korelace)
mean(r) :	průměrná hodnota korelace všech pozorovatelů testu
std(r) :	směrodatná odchylka všech korelací pozorovatelů testu
MTC :	0.85

MTC (Maximal Threshold Correlation) 0.85 je validní např. pro metodu DSCQS, ale pro metodu DCR by měla být použita hodnota MCT 0.7 [\[17\]](#).

## 3 VÝVOJ APLIKACE

Vývoj aplikace je v dnešní době komplexní proces, který vyžaduje náležitou analýzu vytvářeného projektu, čili to, k čemu má být určen, jaké jsou návaznosti jednotlivých komponent, jaké zdroje bude projekt vyžadovat, jaké by měly být použity technologie a až poté je vhodné začít programovat. Proto je kapitola vývoj aplikace rozdělena na sekce analýza projektu, použité technologie a samotná implementace aplikace.

### 3.1 Analýza projektu

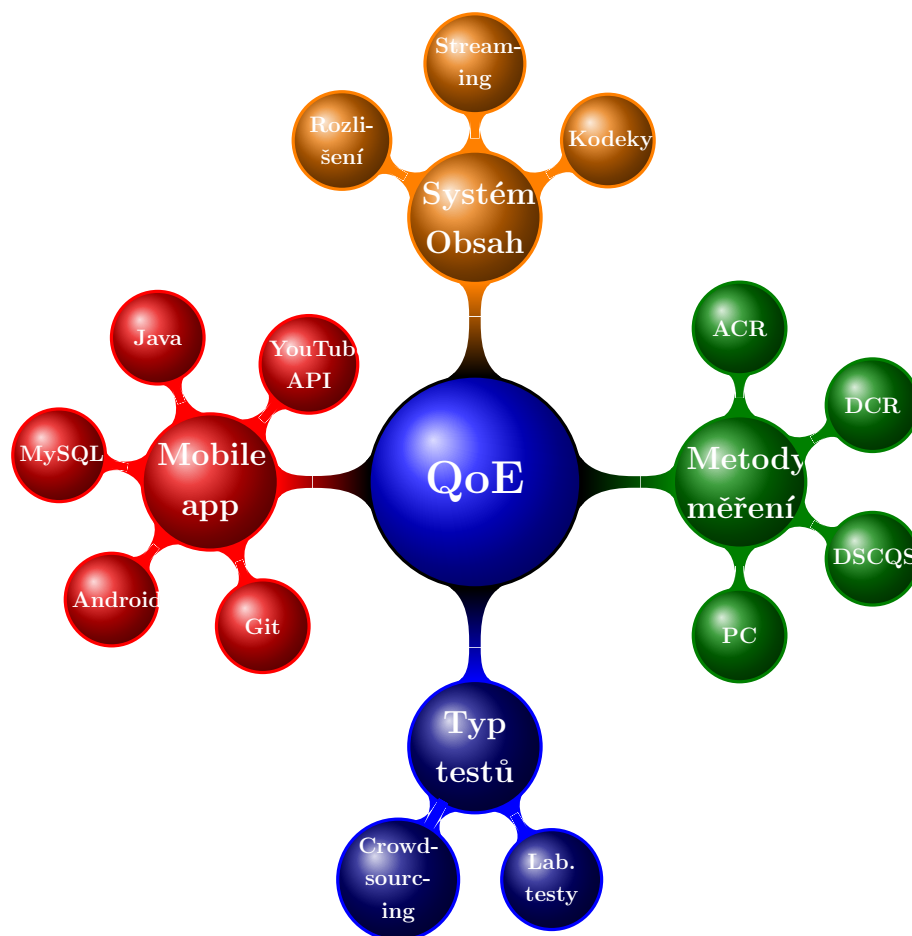
V každém větším softwarovém projektu je vhodné nejprve analyzovat problém a dekomponovat ho na menší podproblémy. K tomu slouží následující diagramy, které zobrazují uživateli činnosti v rámci běhu aplikace, rozložení logiky kódu aj. Záleží na typu diagramu, ke kterým z těchto aspektů je určen. Pro potřeby této práce jsou použity diagramy myšlenková mapa, diagram případů užití, stavový diagram a fyzický model databáze.

#### 3.1.1 Myšlenková mapa

Myšlenková mapa při návrhu systému slouží k uspořádání myšlenek, shrnutí důležitých pojmů a přehledné vizualizaci řešeného tématu. Myšlenková mapa je hierarchická a ukazuje vztahy mezi jednotlivými částmi celého systému. Při tvorbě myšlenkové mapy je vhodné se držet následujícího doporučení:

1. začněte uprostřed s obrázkem znázorňujícím téma,
2. používejte obrázky, symboly, kódy a dimenze skrze myšlenkovou mapu,
3. vyberte slova a vepište je do struktury,
4. každé slovo/obrázek nejlépe v samostatné čáře či obrázku,
5. obrázky nebo čáry by měly být spojené začínaje od počátku,
6. slova by měla být přiměřené délky,
7. používejte více barev napříč myšlenkovou mapou pro vizuální stimul a také pro seskupení různých částí tématu [22].

Na myšlenkové mapě vytvořené na obrázku 3.1 je vidět ve středu nápis QoE, který značí, že tématem této práce je Quality of Experience. V levé části, vyznačené červenou barvou, je poukázáno, že má jít o mobilní aplikaci, jíž jsou přiřazeny technologie, které by mohly být použity při vývoji aplikace.

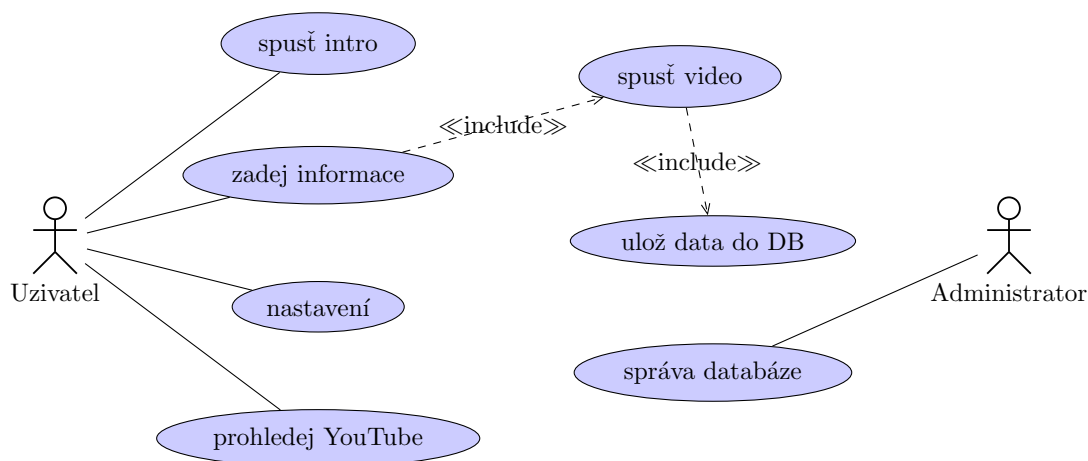


Obr. 3.1: Myšlenková mapa

Ve spodní části obrázku je vidět modře vyznačené rozdělení na typy testů, které mohou být provedeny v rámci řešené problematiky. V pravé části obrázku, vyznačené zelenou barvou jsou vidět některé z možných metod měření při laboratorních testech. A v horní části jsou vidět některé prvky, které se dají zkoumat u multimediálního obsahu. Z výše uvedeného je vidět, že myšlenková mapa velmi přehledně shrnuje základní tematické body z teoretické části a vztahy mezi nimi a rovněž znázorňuje technologie, které by mohly být použity v implementační části pro zkoumanou problematiku.

### 3.1.2 Diagram případů užití

Diagram případů užití se skládá z množiny možných interakcí mezi systémem a uživatelem. Měl by obsahovat všechny systémové aktivity, které mají významný vliv na uživatele. Tento diagram si lze představit jako množinu možných sekvencí, které vedou k určitému cíli.



Obr. 3.2: Diagram případů užití znázorňující vytvářenou mobilní aplikaci

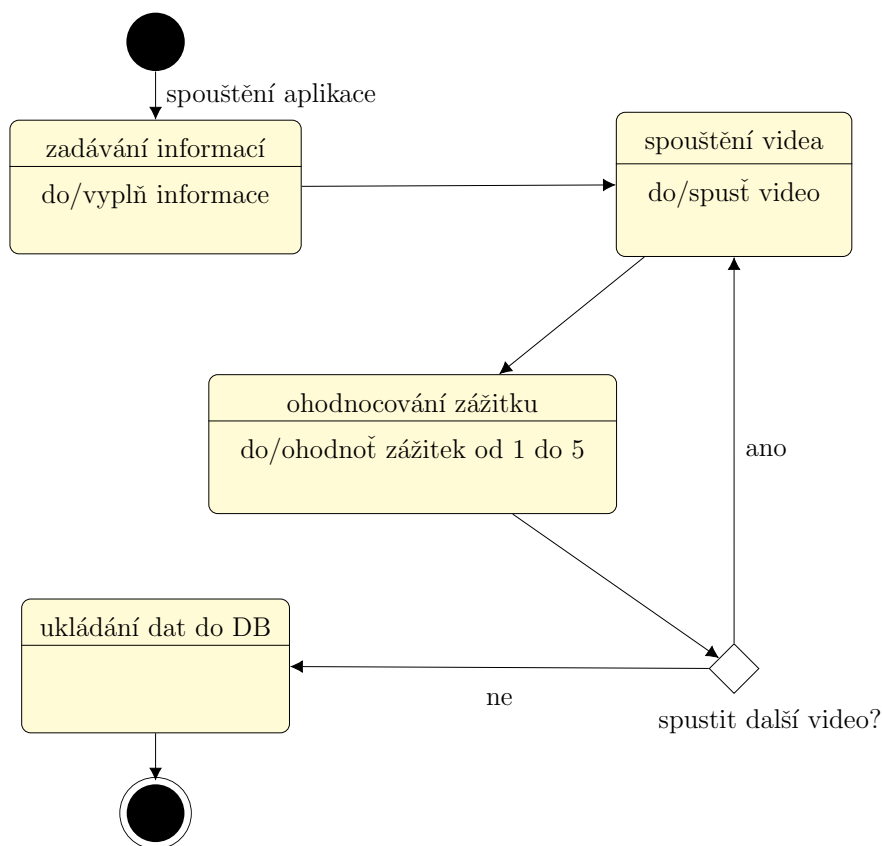
Pomocí tohoto diagramu kromě sekvencí určitých akcí, také znázorňujeme různé typy uživatelů v našem případě uživatel a administrátor, ale obecně jich může být daleko více. V diagramu případů užití se vyskytují tyto prvky:

- Případ užití (use case) – popisuje sled akcí, které poskytují nějakou měřitelnou hodnotu pro uživatele. Případ užití se obvykle kreslí jako horizontální elipsa.
- aktor (actor) – osoba, organizace nebo externí systém, který hraje určitou roli v jedné nebo více interakcích se systémem. Aktor se obvykle kreslí jako panáček.
- asociace (association) – asociace mezi aktory a případy užití jsou indikovány v diagramu případů užití tenkými čarami. Asociace existuje vždy pokud je daný aktor nebo případ užití propojený s danou akcí. Indikace include značí, že po vykonání dané akce se nezbytně poté vykoná akce další.
- systém (system) – je volitelný a můžeme jej nakreslit jako obdélník kolem prvků případu užití. Tento obdélník je nazýván systém. Všechno uvnitř tohoto obdélníku znázorňuje funkcionalitu, která je v rozsahu daného systému a cokoliv mimo obdélník je mimo rozsah [23].

Na diagramu případů užití, který zobrazuje mobilní aplikaci viz obrázek 3.2, která bude vytvořena v této diplomové práci vidíme 2 aktory, uživatele a administrátora. Administrátor v systému funguje tak, že může přistupovat do databáze a manipulovat s daty. Uživatel naopak funguje v roli uživatele aplikace případně testera QoE. Uživatel si může spustit intro, může jít do nastavení, může si vyhledávat videa na serveru YouTube nebo může v roli testera pro QoE zadat informace, spustit a ohodnotit video, na základě čehož se bude vyhodnocovat QoE.

### 3.1.3 Stavový diagram

Stavový diagram je v počítačových vědách používán k popisu chování systému, zvažující všechny možné stavy objektu, kdy dojde k události. Toto chování je reprezentováno a analyzováno v sérii událostí, které se objeví v jednom nebo více možných stavech. Každý stavový diagram začíná s počátečním stavem, což je stav, kde je objekt vytvořen, a končí koncovým stavem, kdy je daný objekt (program – část aplikace) ukončen [23].



Obr. 3.3: Diagram stavů

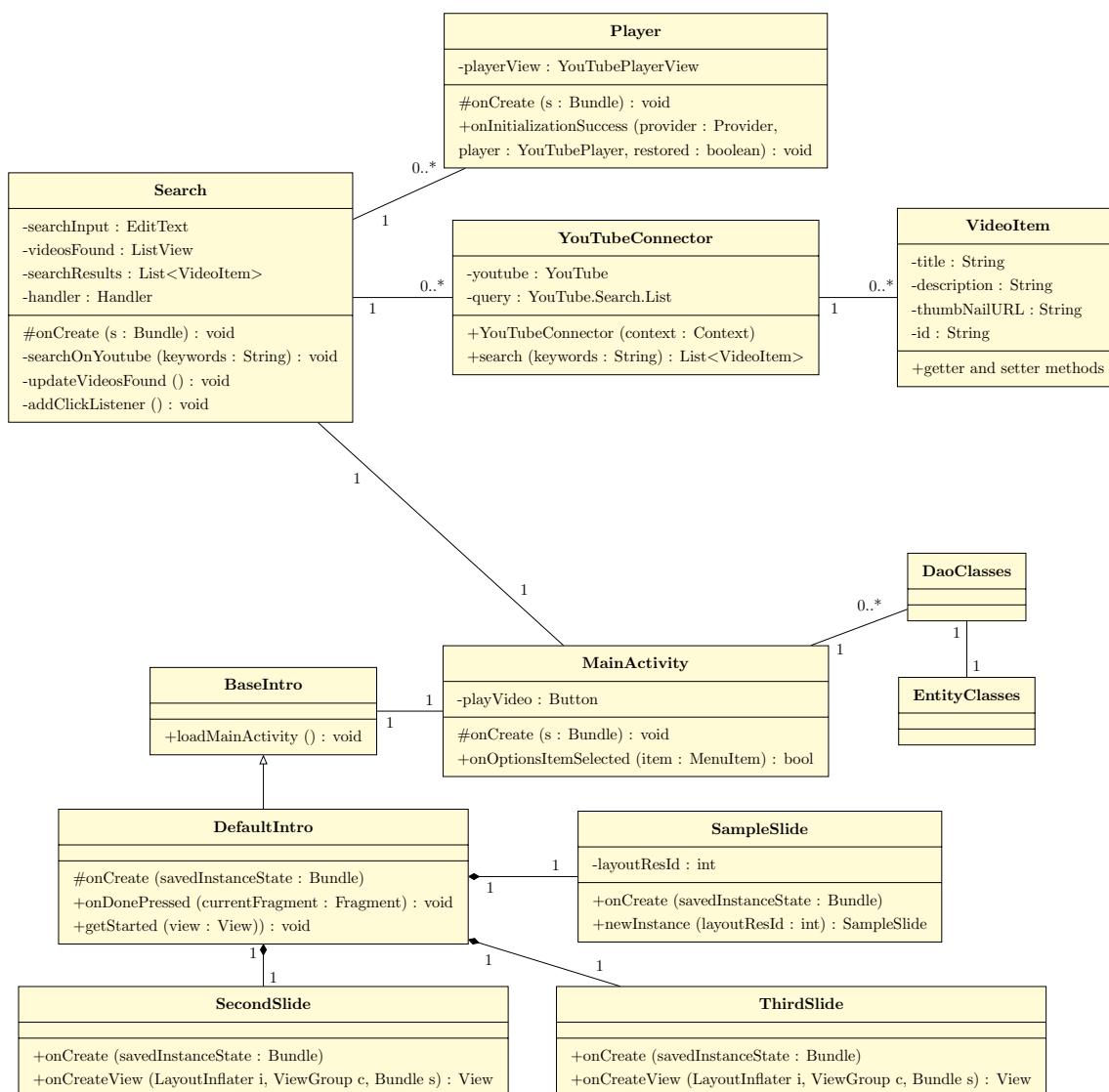
Na stavovém diagramu znázorněném na obrázku 3.3 vidíme průběh testování QoE. Uživatel nejprve zadá základní informace o sobě poté spustí video, ohodnotí ho, a poté pokračuje dalším videem až se nakonec data uloží do databáze.

### 3.1.4 Diagram tříd

Diagram tříd znázorňuje návrh implementace aplikace. Měl by tedy obsahovat všechny třídy, které bude obsahovat aplikace. Z diagramu na obrázku 3.4 lze vidět, že z třídy MainActivity si můžeme pustit intro, vyhledávat videa, spustit video a uložit data do databáze pomocí tříd DaoClasses a EntityClasses, kde Entity představují



entity z ERD popsaného v podsekcí 3.1.6 a DaoClasses jsou třídy, které využívají tyto entity a provádí operace nad databází.

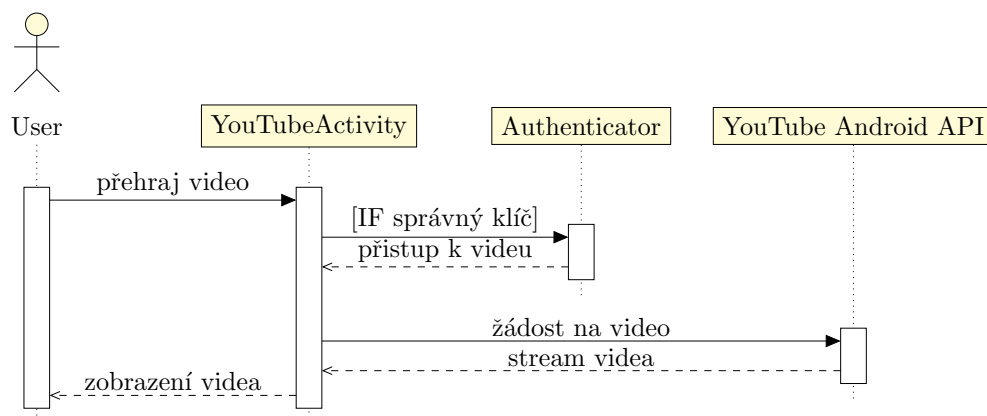


Obr. 3.4: Diagram tříd mobilní aplikace

Tento diagram by tedy měl sloužit jako podrobný návod, jak implementovat aplikaci [23].

### 3.1.5 Sekvenční diagram

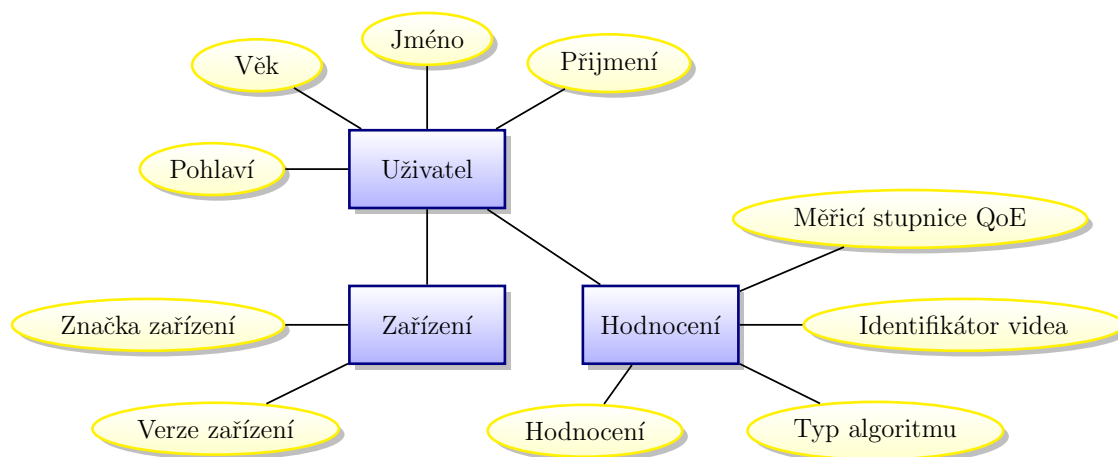
Sekvenční diagram zachycuje časově uspořádanou posloupnost zaslanych zpráv mezi objekty [23]. Na obrázku 3.5 je znázorněna časová řada zpráv zasílaných v případě uživateli žádosti o přehrání videa.



Obr. 3.5: Sekvenční diagram znázorňující použití YouTube Android API

### 3.1.6 Entitně relační model

Entitně relační model (ERD) se používá pro konceptuální znázornění dat. Obvykle se pomocí ERD modelují relační databázové systémy [23]. Pro potřeby Android projektu je vhodné využít nějaký z dostupných nástrojů pro vytváření ERD například MySQL Workbench, z kterého se následně dají vygenerovat potřebné SQL (Structured Query Language) příkazy, které vytvoří modelované tabulky v databázovém systému MySQL.

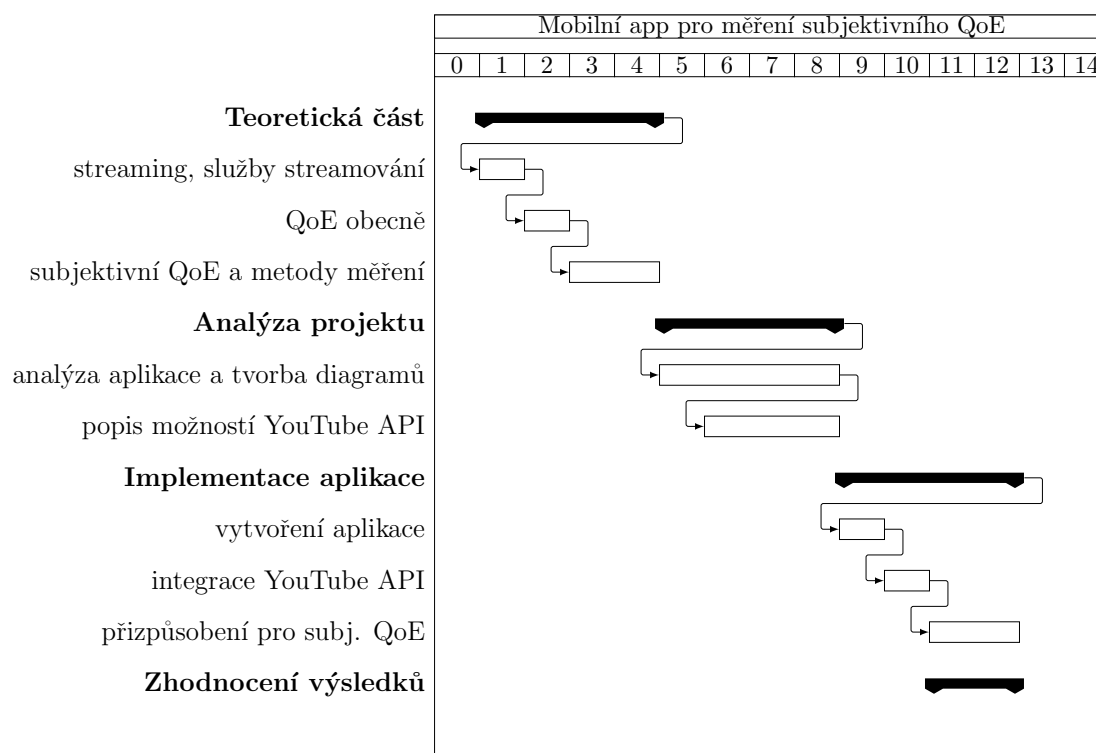


Obr. 3.6: Entitně relační diagram znázorňující fyzickou podobu databáze.

Na obrázku 3.6 lze vidět, že v aplikaci budou ukládány základní údaje o uživateli jako je pohlaví, věk, jméno, příjmení, značka zařízením, verze zařízení a jeho hodnocení videa na základě typu algoritmu a měřicí stupnice.

### 3.1.7 Gantův diagram

Gantův diagram je druh pruhového diagramu, který graficky znázorňuje naplánování posloupnosti činností v čase. V základní podobě nealokuje Gantův diagram vztahy mezi činnostmi, ale často se spojuje se síťovým diagramem, který tyto vztahy dotváří. Gantův diagram pro tento projekt lze vidět na obrázku 3.7.



Obr. 3.7: Gantův diagram znázorňující naplánování posloupnosti činností v čase.

Čísla 1..13 na obrázku výše vyjadřují týdny v roce 2016 začínající od 20. října.

## 3.2 Použité technologie

V této sekci budou nastíněny technologie, které byly použity pro vytvoření softwarové části projektu. Pro hlubší pochopení jednotlivých technologií je vhodné si přečíst citovanou literaturu k dané technologii.

### 3.2.1 Git

Git byl vytvořen Linusem Torvaldsem v roce 2005 k vývoji operačního systému Linux. Je licencován pod GNU General Public License version 2. Jde o distribuovaný systém pro správu verzí určený zejména k softwarovému vývoji. Mezi základní příkazy gitu patří [25]:

- `git init` – inicializace repozitáře gitu v existujícím adresáři,
- `git clone [url]` – vytvoří kopii existujícího repozitáře gitu do aktuální složky, například pro naklonování projektu této práce stačí dát příkaz:  
`git clone https://github.com/SedaQ/Measurement-of-User-Experience`,
- `git status` – příkaz, pomocí kterého zjišťujeme stav jednotlivých souborů čili to, které soubory jsou změněné, které jsou přidáné nebo odebrané atd.,
- `git add .` – po příkazu `git add .` přidáme všechny změněné soubory k sledovaným, což znamená, že jsou připraveny k zapsání a můžeme je zapsat na server nebo commitnout do lokálního repozitáře,
- `git add README` – tento příkaz slouží k vytvoření README souboru, který zpravidla obsahuje konfigurační instrukce, instalační instrukce, copyright a licenční informace, kontaktní informace na distributora nebo programátora, známe buggy aj.,
- `git commit -m "Initial commit."` – zaznamená změny do lokálního repozitáře,
- `git push` – pomocí tohoto příkazu zašleme soubory, které jsme přidali přes příkaz `git add` na server.

Projekt byl nahrán na server github, kde jej můžeme najít na adrese:

<https://github.com/SedaQ/Measurement-of-User-Experience>.

V projektu byl Git upřednostněn před verzovacím systémem SVN (Subversion), protože vývojáři umožňuje plnou lokální kontrolu (např. lokální větve, historie aj.), je rychlejší a umožňuje intuitivnější manipulaci s větvemi než SVN [26].

### 3.2.2 Java

Jako programovací jazyk při tvorbě aplikace byl zvolen programovací jazyk Java, který je nativním jazykem pro platformu Android. Programovací jazyk java byl vytvořen Jamesem Goslingem v roce 1991. Java je široce rozšířena zejména díky své jednoduchosti, robustnosti, bezpečnosti, vysoké výkonnosti, nezávislosti na platformě, efektivní podpoře vícevláknových aplikací a své dynamičnosti. Byla použita verze Java 7, jakožto nejnovější verze javy podporovaná bez větších úprav platformou Android. Verze javy 7 umožňuje oproti dřívějším verzím tzv. diamond operátor, kdy nemusíme v instanciovaném objektu vkládat typ viz kód níže:

---

```
//před verzí Java 7
```

```
List<String> bezDiamondOperatoru = new ArrayList<String>();
```

```
//verze Java 7 a výš (umožnění diamond operátoru)
```

```
List<String> diamondOperator = new ArrayList<>();
```

---

Podporuje takzvaný try-with-resources blok, který využívá Autocloseable interface a tudíž nemusíme zavírat stream ve finally bloku. Také umožňuje řetězení typů výjimek v jednom catch bloku viz kód níže:

---

```
//před verzí Java 7
```

```
try{  
    //vytvoř stream k souboru, databázi aj., zpracovávaj data  
} catch(IOException e){ //zpracuj výjimku  
} catch(Exception e){ //zpracuj výjimku  
} finally{  
    //zavři vytvořený stream  
}
```

```
//verze Java 7 a výš (umožnění try-with-resources bloku a zřetězených  
    výjimek)
```

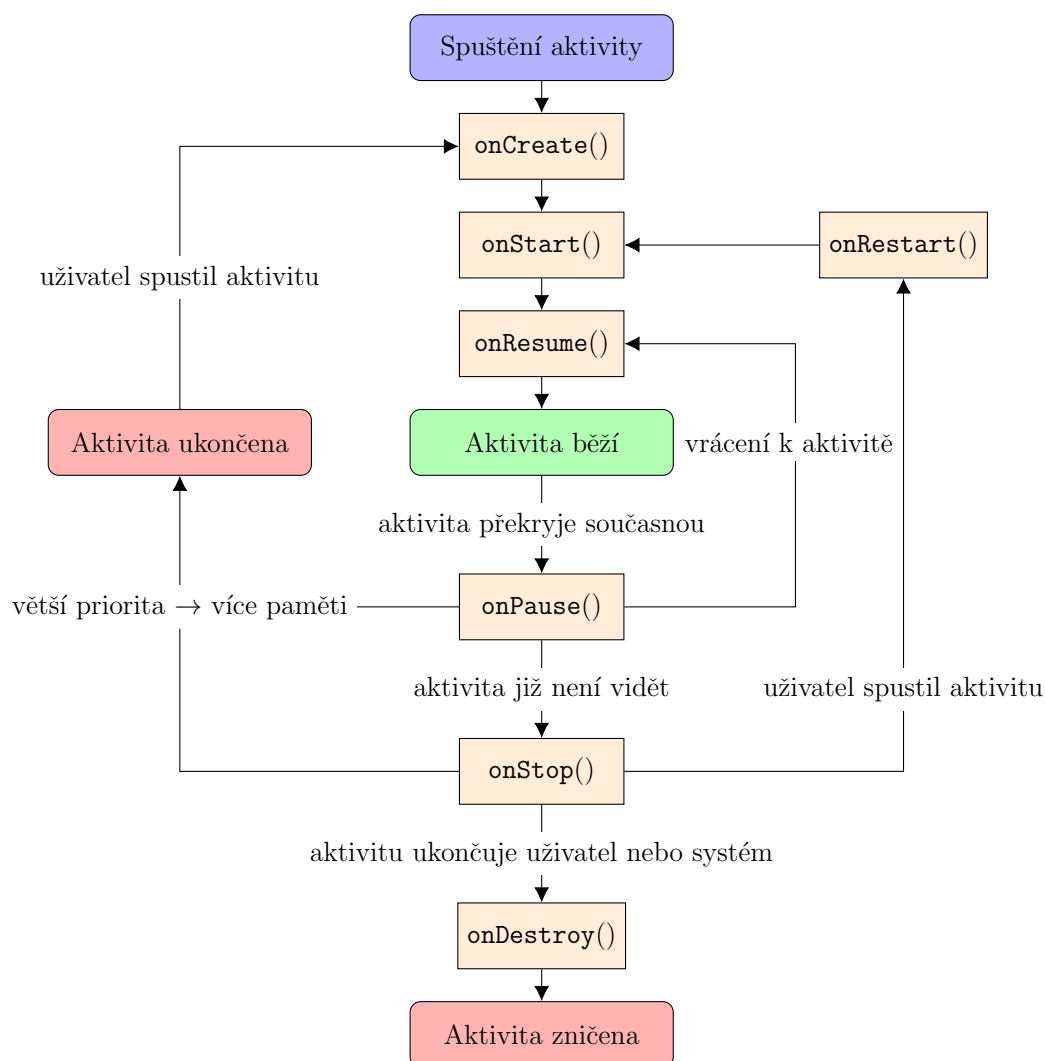
```
try(vytvoř stream k souboru, databázi aj.){  
    //zpracovávaj data  
} catch(IOException | Exception e){  
    //zpracuj výjimku  
}
```

---

Dále byla v Javě 7 přidána bezpečnostní a kryptografická implementace eliptické křivky (Elliptic Curve Cryptography - ECC) atd. [28][29][30].

### 3.2.3 Android a Android Studio

Android je operační systém pro mobilní zařízení vyvinut firmou Google. Mezi základní prvky operačního systému Android patří aktivity, které představují akci, která se vykoná po volbě uživatele například v menu. Tyto aktivity mají daný svůj životní cyklus, který je znázorněn obrázkem 3.8.



Obr. 3.8: Životní cyklus aktivity

Z obrázku lze vidět, že první metoda, která se volá po vytvoření aktivity je metoda `onCreate()` a při ukončení aktivity se volá metoda `onStop()` popřípadě `onDestroy()`. Z tohoto faktu lze jednoduše odvodit, že metoda `onCreate()` by měla být přepisovaná ve většině tříd už jen z důvodů nastavení cesty k layout souboru, který se zobrazuje uživateli na displeji jeho zařízení. V diagramu tříd na obrázku 3.4 lze vidět, že většina tříd přepisuje metodu `onCreate()` [27]. Pro potřeby znázornění životního cyklu Android aktivity v této práci, stejně jako u všech předchozích obrázků byl použit samotný program  $\text{\LaTeX}$  a balíček `tikz`<sup>1</sup>.

<sup>1</sup>Balíček `tikz` je v současné době hlavní balíček v distribuci  $\text{\LaTeX}$  pro kresbu vektorové grafiky. Dokumentace k balíčku `tikz` je na následující adrese: <http://www.texample.net/media/pgf/builds/pgfmanualCVS2012-11-04.pdf>. Pro snadnost použití balíčku `tikz` je též vhodné se podívat na příklady obrázků, které jsou dostupné na stránce: <http://www.texample.net/tikz/examples/>

Android Studio je oficiální Integrated Development Environment (IDE) pro Android. Android studio poskytuje nejrychlejší nástroje pro kompilování aplikací na každý typ Android zařízení. Editace tříd, debuggování, výkonnostní nástroje, flexibilní kompilovací systém atd. [24].

### 3.2.4 YouTube API

YouTube API (Application Programming Interface) umožňuje vývojářům přistupovat k videím na serveru YouTube. Vývojáři využívají YouTube přehrávače, data a nástroje, které poskytuje YouTube API.

#### YouTube přehrávač

YouTube videa obvykle zobrazujeme v různých přehrávačích, jako je IFrame, Android Player nebo iOS player.

IFrame se používá zejména, pokud chceme vkládat video ze serveru YouTube do webové stránky a poté řídit toto video pomocí JavaScriptu. Požadavkem na možnost využití tohoto API pomocí tagu `<iframe>` je, že koncový uživatel musí mít prohlížeč s podporou Hyper Text Markup Language 5 (HTML5). Pomocí funkcí JavaScript rozhraní API můžeme videa přehrát, pozastavit, upravit hlasitost přehrávače, změnit kvalitu videa a získávat informace o přehrávaném videu. Můžeme přidat i různé posluchače událostí (listeners), které vykonají určitou akci po spuštění určité akce např. pokud spustíme video, tak změníme kvalitu přehrávání na High Definition (HD). Příklad použití tagu `<iframe>` je vidět na kódu níže:

---

```
<!DOCTYPE html>
<html>
  <body>
    <iframe id="p" type="text/html" width="640" height="390"
      src="http://www.youtube.com/embed/M7lc1UVf-VE?
        enablejsapi=1&origin=http://example.com"
      frameborder="0"></iframe>
  </body>
</html>
```

---

Android player a iOS player umožňují přistupovat k videu v Android a iOS aplikaci. Android player, jeho možnosti a integrace do projektu je popsána v podsekcí 3.3.4.

## YouTube data a zdroje

YouTube Data API (v3) slouží k přidávání vlastností zahrnujících možnost nahrávání videa na YouTube server, vytváření a vedení playlistů atd.

YouTube Analytics API se využívá k získání statistik, populárních metrik a více pro YouTube videa a kanály. Metriky mohou být například: Jak frekventovaně jsou vaše videa zobrazována? Jak dlouho uživatelé typicky stráví pozorováním vašeho videa? Jak se zobrazení odlišuje v závislosti na zemi?

YouTube Live Streaming API umožňuje vytvářet, aktualizovat a spravovat živé přenosy na YouTube. Pomocí tohoto API můžete naplánovat události a asociovat je s video streamy.

Server YouTube podporuje 2 typy volání této API, a to Representational state transfer (REST) a XML-RPC (Extensible markup language - remote procedure call). Volání pomocí REST API si předává zprávy ve formátu JavaScript Object Notation (JSON), což je v současnosti upřednostňováno před XML na tyto výměny zpráv vzhledem k menší paměťové náročnosti formátu JSON [33].

### 3.2.5 Gradle

Gradle je open source automatizovaný systém pro buildování projektů, který staví na konceptu Apache Antu a Apache Mavenu a představuje doménově specifický jazyk (DSL), narozdíl od Antu a Mavenu, které jsou založeny na XML konfiguracích [31]. Část souboru build.gradle z projektu je vidět na kódu níže.

---

```
apply plugin: 'com.android.application'
repositories {
    mavenCentral()
}
dependencies {
    compile fileTree(dir: 'libs', include: ['*.jar'])
    compile 'com.android.support:appcompat-v7:24.2.1'
    compile 'com.github.paolorotolo:appintro:4.0.0'
}
```

---

Na kódu vidíme v části dependencies {...} definované knihovny, které budou zkompilovány a rovněž část je zde definován typ aplikace jako Android aplikace pomocí apply plugin 'com.android.application'



### 3.2.6 MySQL a MySQL Workbench

MySQL je open source relační databázový systém vytvořený švédskou firmou MySQL AB v roce 1995. V současnosti vlastní MySQL společnost Oracle Corporation, která patří mezi přední poskytovatele relačních databázových systémů. V tomto projektu byl databázový systém MySQL použit z důvodů jeho volné rozšiřitelnosti a dostačující podpory požadovaných funkcionalit. Jako volně dostupný MySQL databázový hosting byl použit hosting na serveru: <https://www.freemysqlhosting.net/>, který umožňuje ukládat data bez zpoplatnění do 5 MB.

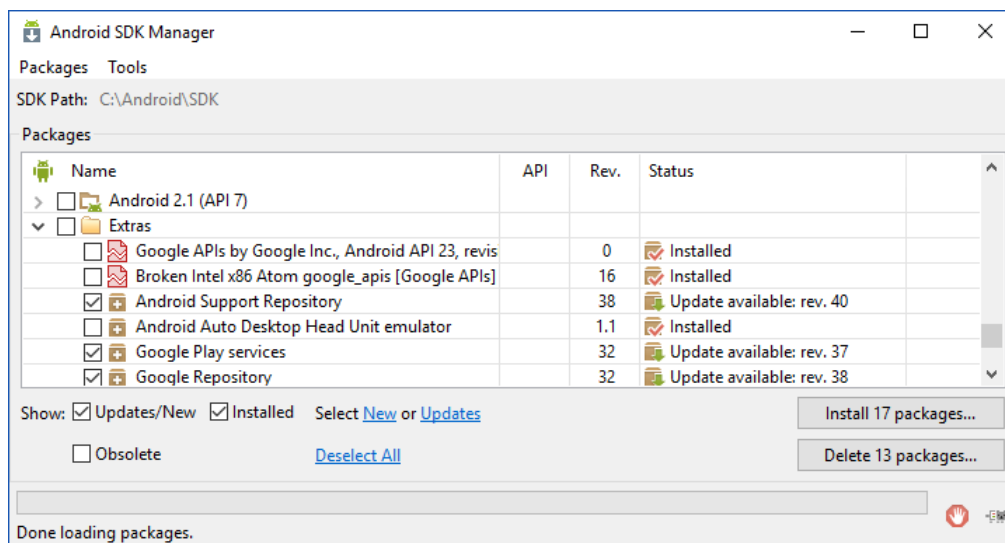
MySQL Workbench je nástroj určený pro databázové architekty a vývojáře. Umožňuje navrhnout strukturu databáze na základě níž se dají vygenerovat SQL příkazy, poté se dá připojit k určené databázi a vygenerovat tabulky na základě SQL příkazů v příslušné MySQL databázi [32].

## 3.3 Implementace aplikace

V následující části této práce jsou popsány prerekvizity k vytvoření Android projektu, stručný postup vytvoření, kompilace projektu a následná integrace YouTube API do takto vytvořeného projektu.

### 3.3.1 Prerekvizity k vytvoření Android projektu

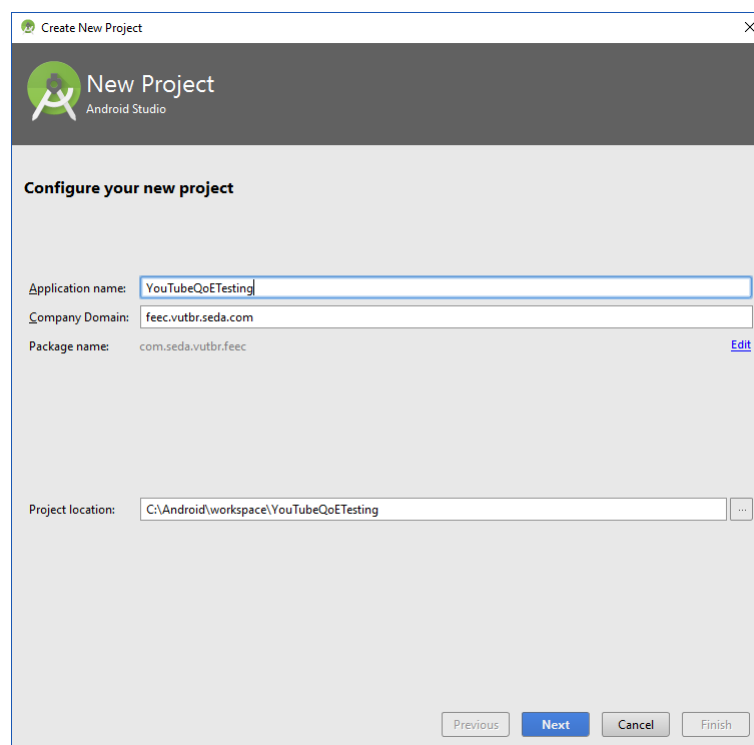
Pro vytvoření Android aplikace bude potřeba si nejprve stáhnout Java Development Kit (JDK) například na stránce: <http://www.oracle.com/technetwork/java/javase/downloads/jdk7-downloads-1880260.html>, kde v části Java SE Development Kit 7u79, budou přijaty licenční podmínky a stažen jdk-7u79 balíček dle operačního systému. Po stažení následuje instalace JDK na OS. Dalším krokem je nainstalování vývojového prostředí, ve kterém bude vyvinuta aplikace. Pro účely projektu je použito vývojové prostředí Android studio, které se dá stáhnout např. na stránce: <https://developer.android.com/studio/index.html>. Po stažení Android Studia je nutné zkontrolovat, jestli součástí balíčku byl i Software Development Kit (SDK), který představuje množinu vývojářských nástrojů používaných k vývoji aplikací na platformě Android. Obsahuje emulátor, debugger, požadované knihovny, dokumentaci pro aplikační programová rozhraní Android API atd. Součástí SDK by měl být i SDK Manager, který se používá pro stahování požadovaných knihoven zabudovaných v Android API. Na obrázku 3.9, zobrazujícím SDK Manager, je vidět, že SDK Manager nabízí aktualizování Google Play services atd.



Obr. 3.9: Ukázka SDK Manageru

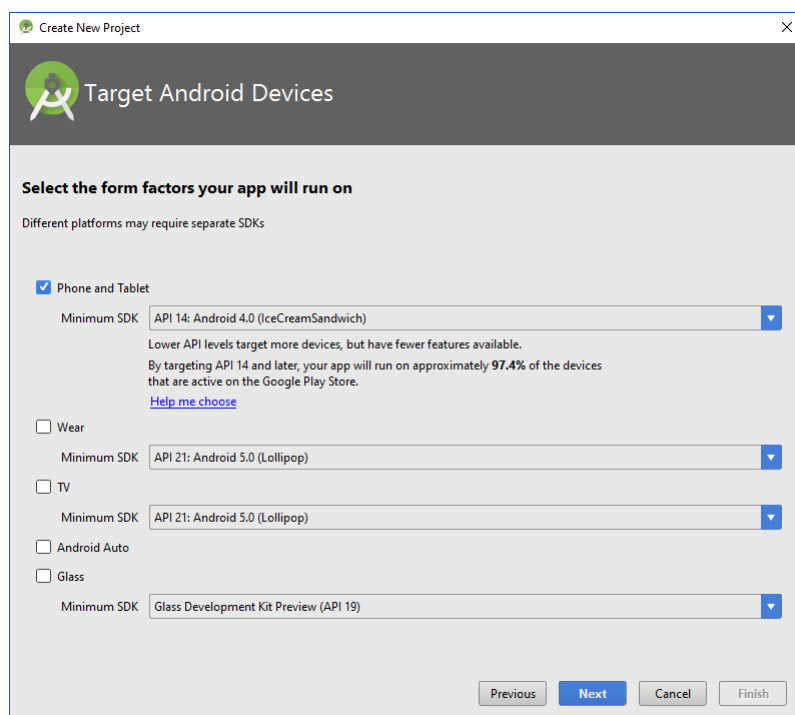
### 3.3.2 Vytvoření Android projektu

Po spuštění Android studia bude kliknuto na **Start a new Android Studio project**. Otevře se okénko znázorněné na obrázku 3.10. Na obrázku je vidět, že Application name bylo zvoleno YouTubeQoETesting a Company domain je feec.vutbr.seda.com. Z Application name a Company domain se generuje Package name, které bylo editováno na com.seda.vutbr.feec z com.seda.vutbr.feec.youtubeqoetesting.



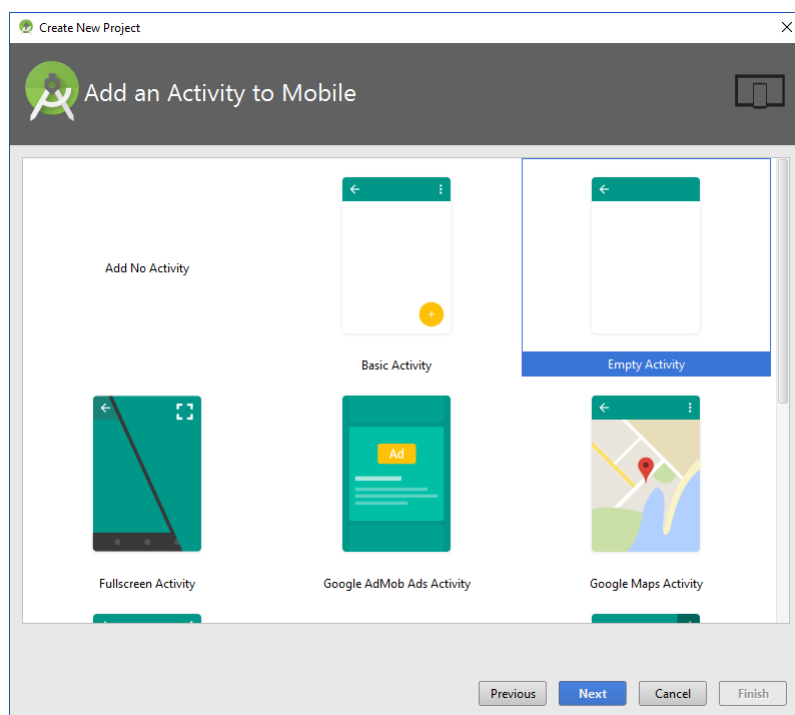
Obr. 3.10: Vytváření nového projektu v Android Studiu

Po vyplnění údajů na obrázku 3.10 je kliknuto na tlačítko **Next**. Otevře se nové okno, které je vidět na obrázku 3.11. Z obrázku je vidět, že cílová zařízení jsou mobily a tablety. Tato cílová zařízení musí mít verzi Androidu 4.0 (IceCreamSandwich) a vyšší. Dle informací poskytnutých Android Studiem by restrikce na verzi 4.0 a výše měla pokrýt přibližně 97.4% zařízení, která jsou aktivována na Google Play Store.



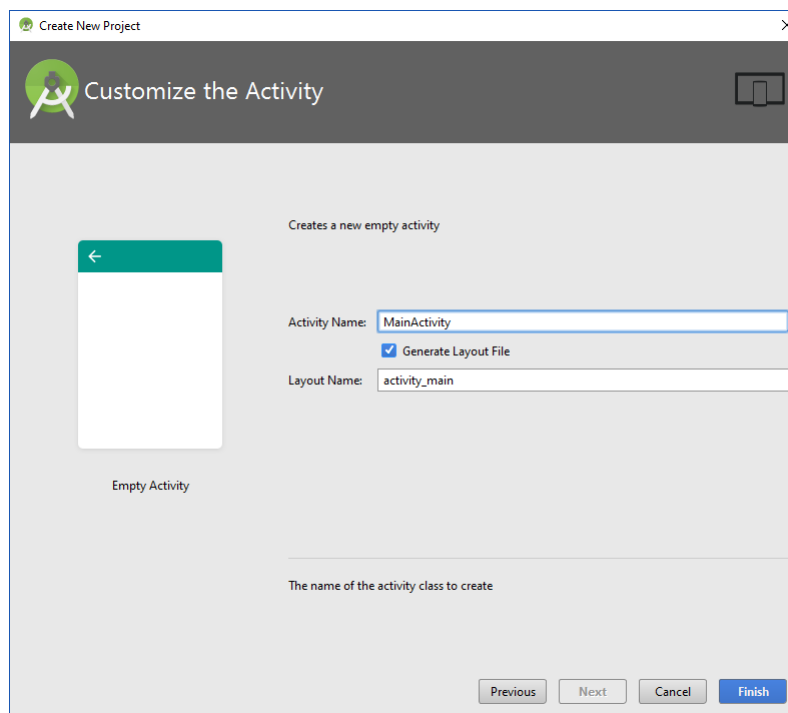
Obr. 3.11: Výběr cílových zařízení

Po selekci cílových zařízení a požadavků na cílová zařízení je kliknuto na tlačítko **Next**. Otevře se okno s výběrem základní aktivity projektu viz obrázek 3.12



Obr. 3.12: Výběr Android aktivity do projektu

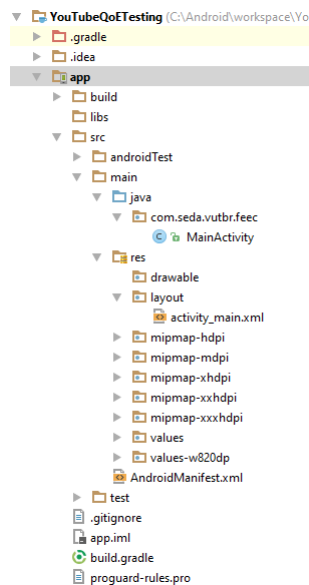
Jako základní aktivita je vybrána aktivita typu Empty Activity a je kliknuto na tlačítko **Next** (výběr této aktivity znamená, že je projekt čistý a neobsahuje žádné předpřipravené aktivity, které nabízí Android Studio). Zobrazí se okno, které je vidět na obrázku 3.13, kde do pole Activity Name je vloženo MainActivity, do pole Layout Name je vloženo activity\_main a je zaškrtnuto Generate Layout File, což způsobí, že se vytvoří základní layout soubor pro aktivitu MainActivity. Poté je kliknuto na tlačítko **Finish**, čímž se vytvořil nový Android projekt.



Obr. 3.13: Dokončení vytvoření nového Android projektu

Struktura takto vytvořeného projektu ve vývojovém prostředí nyní vypadá jako na obrázku 3.14, kde je vidět třída<sup>2</sup> MainActivity, layout soubor activity\_main.xml, AndroidManifest.xml, build.gradle a další. Třída MainActivity je javovský soubor, který představuje aktivitu, která po vybuildování a spuštění aplikace načte soubor activity\_main.xml. V tomto .xml souboru je definován vzhled spuštěné aplikace, tzn. do tohoto souboru přidáváme různá tlačítka, textová pole a další, která pak uživatel vidí ve své aplikaci (javovské soubory dodávají logiku k těmto nadefinovaným tlačítkům, polím atd.).

<sup>2</sup>Třída je množina objektů s určitými vlastnostmi. Přitom samotná třída nedefinuje nějaké konkrétní objekty třídy, jen udává, jaké vlastnosti bude mít každý objekt třídy (podobně představa o tom, co je telefon, může existovat, i kdyby všechny existující telefony zanikly).



Obr. 3.14: Struktura nově vytvořeného projektu ve vývojovém prostředí

Soubor AndroidManifest.xml popisuje základní charakteristiky aplikace a definuje jednotlivé komponenty. Pro nově vytvořený projekt manifest soubor vypadá následovně:

---

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.seda.vutbr.feec">

    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:supportsRtl="true"
        android:theme="@style/AppTheme">
        <activity android:name=".MainActivity">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER"
                    />
            </intent-filter>
        </activity>
    </application>

</manifest>
```

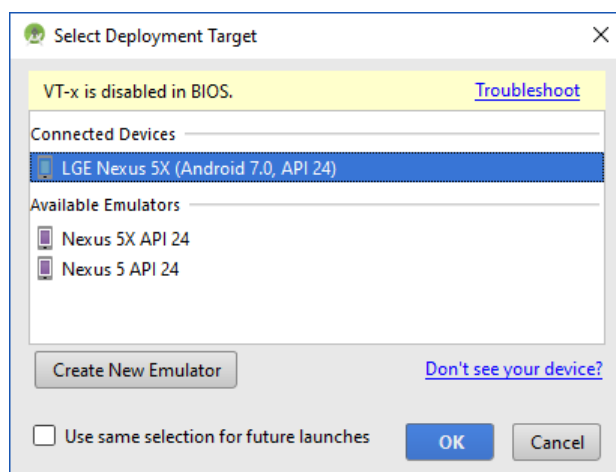
---

Na kódu výše je vidět, že v tomto souboru se dá definovat ikonka aplikace, její jméno, základní kompozice projektu a všechny aktivity, které jsou naprogramovány v javovských souborech.

Soubor build.gradle je buildovací soubor, který slouží k překompilování/stáhnutí závislostí na různé knihovny, konfigurace minimální verze Androidu na cílových zařízeních atd.

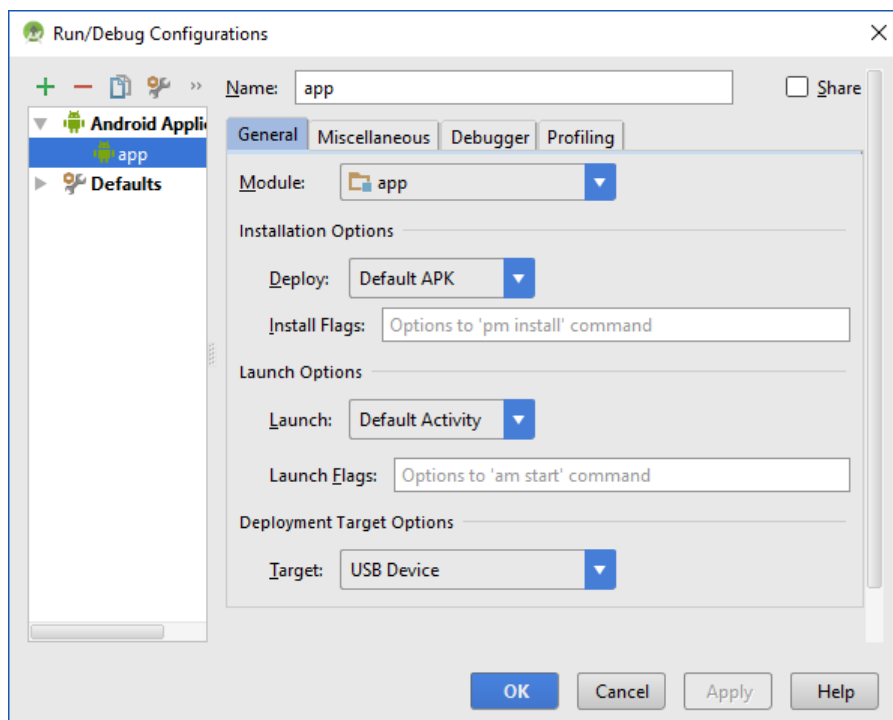
### 3.3.3 Kompilace Android aplikace

Pro kompilaci aplikace v Android studiu se používá buď emulátor nebo připojené zařízení jako lze vidět na obrázku 3.15 (Toto dialogové okno se objeví pokud je v Android Studio v daném projektu kliknuto na Run a není vybrán žádný Deployment Target).



Obr. 3.15: Výběr cílového zařízení pro kompilaci emulátor/připojené zařízení.

V případě, že bude pro testovací účely vybráno připojené zařízení, tak je nutné u tohoto zařízení povolit tzv. USB debugging. USB debugging bývá zpravidla v nastaveních telefonu pod Informace o telefonu. Pro kontrolu je ještě nutné se podívat do Run Configurations, kde je nutné v případě, že už tak není učiněno, nastavit Deployment Target Options → Target na USB Device jako lze vidět na obrázku 3.16. Po těchto nastaveních, připojení cílového zařízení a stisknutém tlačítku Run v Android studio je nahrán na cílové zařízení .apk soubor, který představuje spustitelný soubor pro mobilní aplikace a je spuštěna vytvořená aplikace.



Obr. 3.16: Nastavení cílového zařízení na testování

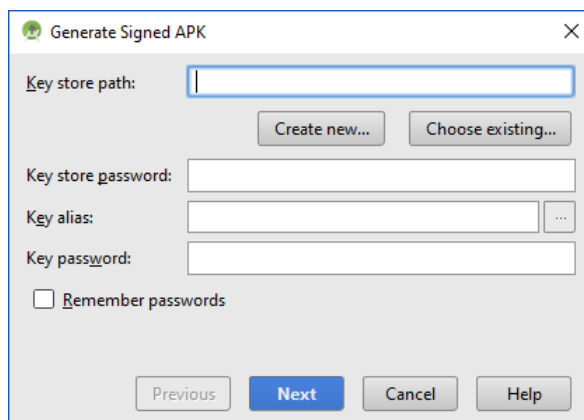
### 3.3.4 Integrace YouTube Android API do projektu

Prvním stěžejním bodem integrace YouTube API do vlastního Android projektu je stažení balíčku YouTubeAndroidPlayerApi, který se dá stáhnout na adrese <https://developers.google.com/youtube/android/player/downloads/>. Dalším bodem je registrace aplikace, včetně digitálně podepsaného souboru `.apk` (spustitelný soubor pro Android aplikace) veřejným klíčem na Google Developers Console, aby se dalo použít YouTube Android Player API z aplikace. Pro registraci aplikace je nutné postupovat podle kroků níže:

1. Na stránce <https://console.developers.google.com> se klikne v levém horním rohu na **Dashboard** a poté na tlačítko **Enable API** (nebo stačí jít rovnou přes **Library**). Ze široké nabídky poskytovaných API od společnosti Google se vybere sekce YouTube APIs a konkrétně YouTube Data API, která poskytuje přístup k datům YouTube, jako jsou videa, playlisty a kanály. V nově otevřeném okně se klikne na tlačítko **Enable**. Tím je zajištěno, že je povolené využívat službu YouTube Data API v3.
2. Nyní k nově vytvořenému projektu YouTubeQoETesting se nastaví přihlašovací údaje k YouTube Data API v3. YouTube Data API v3 umožňuje 2 způsoby přístupu buď OAuth 2.0 nebo pomocí API klíče. Na stránce je nutné vytvořit

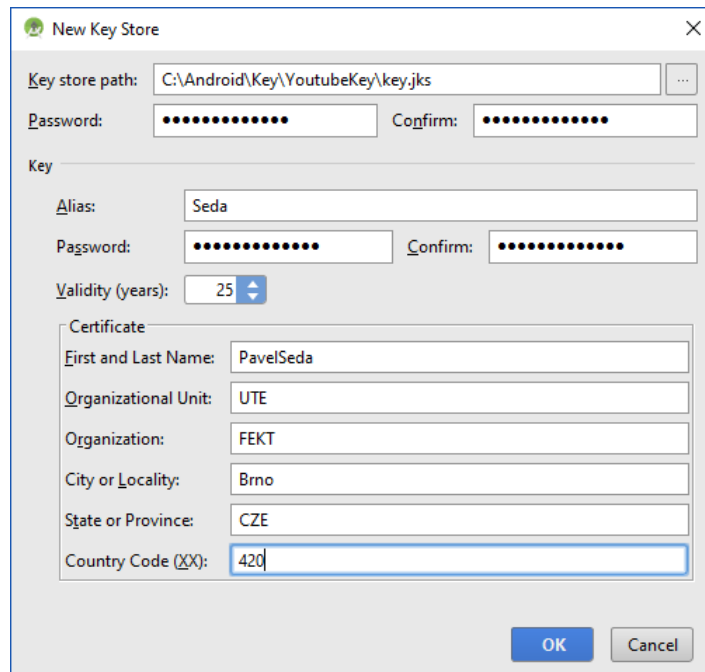


projekt s názvem YouTubeQoETesting a poté pro tento projekt v levé liště nahoře se klikne na **Credentials**, poté na tlačítko **Create credentials** vybere se typ např. API key, klikne se na **RESTRICT KEY** pokud není žádoucí, aby tento API klíč mohla použít neautorizovaná osoba. V sekci key restriction se vybere Android apps. Nyní ve spuštěném Android studiu se klikne na **Build** → **Generate Signed APK**, což otevře okno, které lze vidět na obrázku 3.17.



Obr. 3.17: Generování podepsaného .apk souboru nastavení část 1.

Klikne se na tlačítko **Create new...** a otevře se okno, které se vyplní obdobně jako lze vidět na obrázku 3.18. Zapiše se heslo, protože bude potřebné pro vygenerování SHA1 (Secure Hash Algorithm 1) klíče.

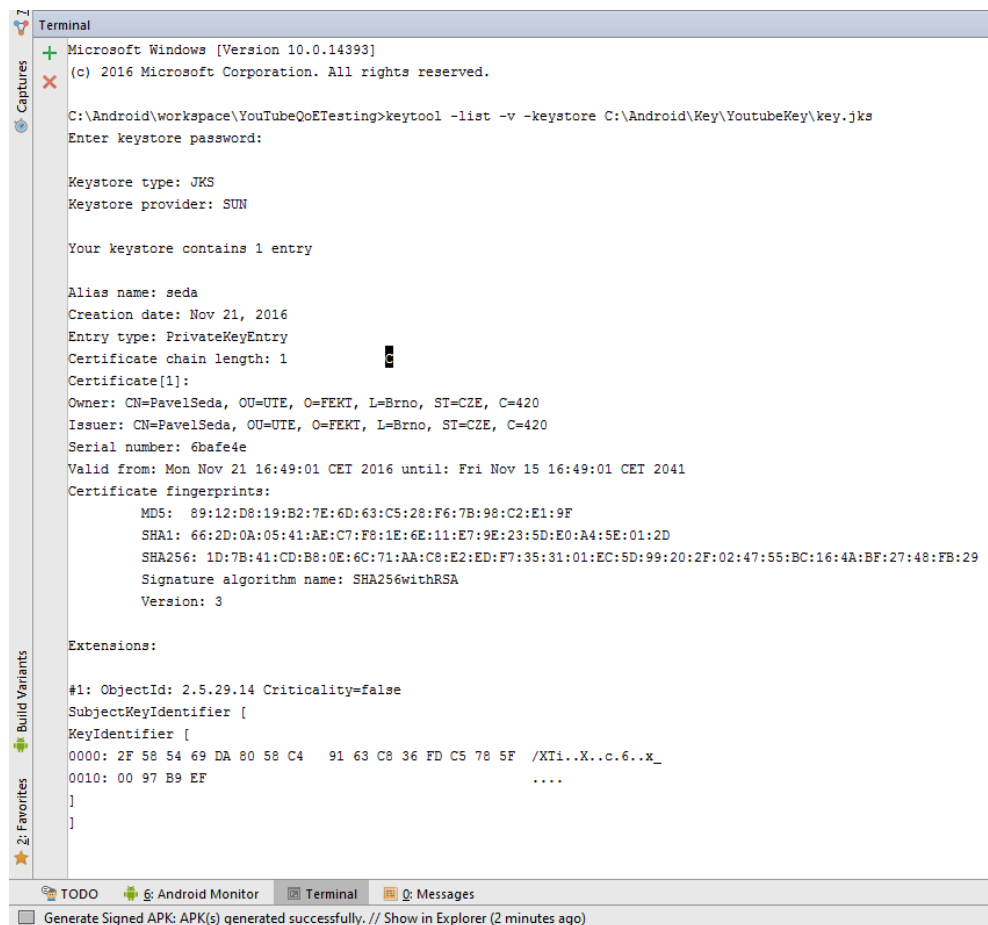


Obr. 3.18: Generování podepsaného .apk souboru nastavení část 2.

Otevře se terminál v Android studiu, kde se vloží příkaz dle následujícího vzoru:

```
keytool -list -v -keystore mystore.keystore.path
```

Terminál z android studia poté bude vyžadovat heslo, které jsme si zapsali při generování podepsaného .apk souboru v krocích výše, zadá se heslo a poté lze vidět v terminálu vygenerovaný SHA1 klíč obdobně jako na obrázku [3.19](#)



```
Terminal
+ Microsoft Windows [Version 10.0.14393]
(c) 2016 Microsoft Corporation. All rights reserved.

C:\Android\workspace\YouTubeQoETesting>keytool -list -v -keystore C:\Android\Key\YoutubeKey\key.jks
Enter keystore password:

Keystore type: JKS
Keystore provider: SUN

Your keystore contains 1 entry

Alias name: seda
Creation date: Nov 21, 2016
Entry type: PrivateKeyEntry
Certificate chain length: 1
Certificate[1]:
Owner: CN=PavelSeda, OU=UTE, O=FEKT, L=Brno, ST=CZE, C=420
Issuer: CN=PavelSeda, OU=UTE, O=FEKT, L=Brno, ST=CZE, C=420
Serial number: 6baf4e
Valid from: Mon Nov 21 16:49:01 CET 2016 until: Fri Nov 15 16:49:01 CET 2041
Certificate fingerprints:
    MD5:  89:12:D8:19:B2:7E:6D:63:C5:28:F6:7B:98:C2:E1:9F
    SHA1: 66:2D:0A:05:41:AE:C7:F8:1E:6E:11:E7:9E:23:5D:E0:A4:5E:01:2D
    SHA256: 1D:7B:41:CD:B8:0E:6C:71:AA:C8:E2:ED:F7:35:31:01:EC:5D:99:20:2F:02:47:55:BC:16:4A:BF:27:48:FB:29
    Signature algorithm name: SHA256withRSA
    Version: 3

Extensions:
#1: ObjectId: 2.5.29.14 Criticality=false
SubjectKeyIdentifier [
KeyIdentifier [
0000: 2F 58 54 69 DA 80 58 C4  91 63 C8 36 FD C5 78 5F  /XI...X..c.6..x_
0010: 00 97 B9 EF                ....
]
]
```

Obr. 3.19: Vygenerovaný SHA1 soubor pro podepsaný .apk soubor

Zkopíruje se vygenerovaný SHA1 heš a na stránce, kde je vytvořený YouTube-QoETesting projekt se klikne na tlačítko **Add package name and fingerprint** a vloží se tam package name vytvořeného projektu a vygenerovaný SHA1 heš, klikne se na tlačítko **Save**. Stránka by nyní měla vypadat jako na obrázku 3.20.

**API key**

AIzaSyDFDNMnxc0nDswQQKCbGsgZQuBxYeF1ONE

**Name**

YouTubeQoETesting

**Key restriction**

Key restriction lets you specify which web sites, IP addresses, or apps can use this key. [Learn more](#)

☐ None  
☐ HTTP referrers (web sites)  
☐ IP addresses (web servers, cron jobs, etc.)  
☒ Android apps  
☐ iOS apps

**Restrict usage to your Android apps** (Optional)

Add your package name and SHA-1 signing-certificate fingerprint to restrict usage to your Android apps  
 Get the package name from your AndroidManifest.xml file. Then use the following command to get the fingerprint:

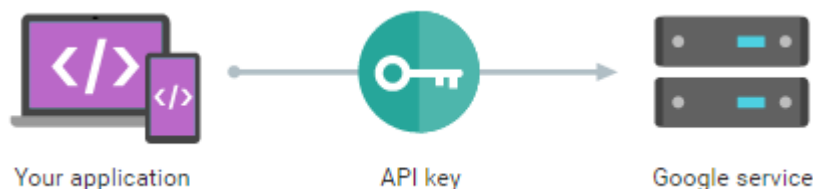
```
$ keytool -list -v -keystore mystore.keystore
```

Package name	SHA-1 certificate fingerprint
com.seda.vutbr.feec	66:2D:0A:05:41:AE:C7:F8:1E:6E:11:E7:9E:23:5D:E0:A4:5E:01:2D

+ Add package name and fingerprint

Obr. 3.20: Vložený SHA1 heše a package name

a je zaručeno, že žádná neautorizovaná osoba nevyužije tento API key na přístup k YouTube Android Player API. Využití samotného API klíče je znázorněno na obrázku 3.21.



Obr. 3.21: Znázornění využití API klíče ke Google službě

- Posledním krokem je vložení API klíče a YouTube Data API v3 do projektu YouTubeQoETesting [34] [35]. Ve vytvořeném Android projektu se vloží knihovna YouTubeAndroidPlayerApi.jar do složky libs, API klíč se uloží v souboru strings.xml v následující podobě:

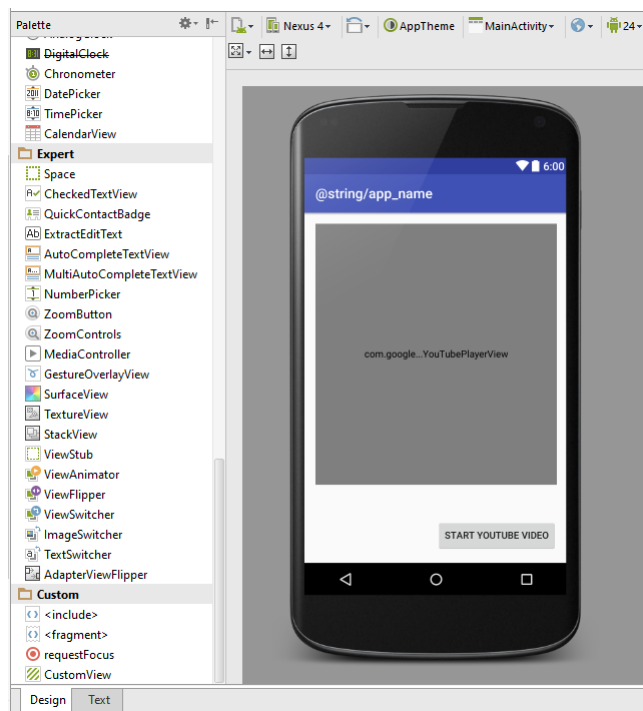
---

```
<string name="YOUTUBE_API_KEY">Generated API key </string>
```

---

Otevře se soubor activity\_main, v kterém se otevře Design perspektiva a na paletě komponent v sekci Custom se vybere CustomView, otevře se dialogové

okno, ve kterém se vybere `YouTubePlayerView` a vloží se do obrazovky, poté by měla vypadat Design perspektiva zhruba obdobně jako na obrázku 3.22.



Obr. 3.22: Vložení YouTube API do layout souboru

V perspektivě Text by nyní měla být vidět následující část kódu, ve které se upravilo id na `youtube_view`:

---

```
<com.google.android.youtube.player.YouTubePlayerView
    android:id="@+id/youtube_view"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_weight="0.29" />
```

---

Jako poslední konfigurační záležitost je nutné nastavit v souboru `AndroidManifest.xml` uvnitř tagu `manifest`, následující blok kódu, který vyžaduje, že dané zařízení musí mít přístup k internetu

---

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

---

Nyní je nezbytné napojit na layout soubor `activity_main` logiku v javovském souboru `MainActivity.java`. Tato třída po implementování vypadá následovně:

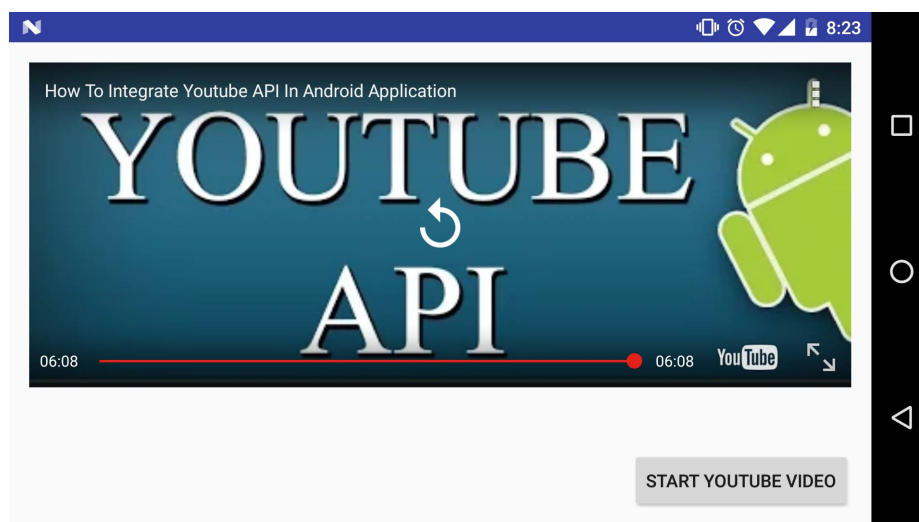
---

```
public class MainActivity extends YouTubeBaseActivity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        YouTubePlayerView youTubePlayerView = (YouTubePlayerView)
            findViewById(R.id.youtube_view);
        YouTubePlayer.OnInitializedListener initListener = new
            YouTubePlayer.OnInitializedListener(){
                @Override
                public void onInitializationSuccess
                    (YouTubePlayer.Provider provider, YouTubePlayer
                     youTubePlayer, boolean b) {
                    youTubePlayer.loadVideo("a4NT5iBFuZs");
                }
            };
    }
}
```

---

Pro každou Android aktivitu, která má přistupovat k YouTube API je nutné, aby dědila z třídy `YouTubeBaseActivity`, což v kódu výše vidíme `MainActivity extends YouTubeBaseActivity`. V kódu je spuštění YouTube videa implementováno jako událost, která se spouští po kliknutí na tlačítko start youtube video, které po stisknutí inicializuje `youTubePlayerView` s uloženým API klíčem (pro zkrácení ukázky kódu bylo toto tlačítko vyňato z ukázkového kódu stejně jako nutnost `@Override onInitializationFailure` při inicializaci třídy `YouTubePlayer.OnInitializedListener()`). Při úspěšné inicializaci je spuštěno video s klíčem `a4NT5iBFuZs`, což je identifikátor určitého videa na YouTube, celá adresa tohoto videa je: <https://www.youtube.com/watch?v=a4NT5iBFuZs>, pro spuštění ovšem stačí zadávat pouze identifikátor videa. Nyní je vše nastavené a po zkompileování aplikace bychom ji měli vidět obdobně jako na obrázku 3.23.



Obr. 3.23: Screen save z mobilní aplikace po integraci YouTube API

### 3.3.5 Úprava aplikace pro potřeby subjektivního testování

Pro subjektivní testování je důležité zajistit, aby bylo v přehrávaném videu možné měnit rozlišení, nastavovat adaptivní streamování aj. Bohužel při bližším prozkoumání YouTube Android API bylo zjištěno, že toto API pro Android v současné době neumožňuje z programového kódu měnit rozlišení videa ani nastavení na adaptivní streamování. Absence této možnosti jde vidět přímo ve výčtu podporovaných metod z dokumentace [YouTube Android API](#) v části public methods. Bohužel v této třídě `YouTubePlayer` při volání metody `loadVideo` nebo `cueVideo` nejde spustit video s určitým rozlišením ani při použití náhradního řešení v podobě přidání řetězce např. `"?vq=hd1080"`, k identifikátoru videa, který definuje typ rozlišení, v kterém má být video spuštěno (toto funguje pouze prostřednictvím webového rozhraní, ale YouTube Android API to nepodporuje).

I přesto bude YouTube Android API využito při dalších možnostech aplikace, jako je vyhledávání videí na YouTube viz podsekcce ??.

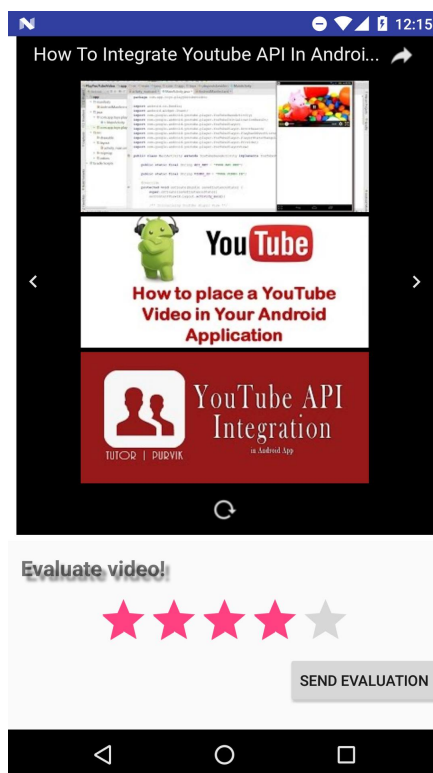
Východiskem nemožnosti použití YouTube Android API pro subjektivní testování v rámci nastavování typu streamingu a kvality videa je použití třídy `WebView`, která umožňuje pouštět webové stránky z mobilní aplikace. Pomocí této třídy se dá spustit YouTube video s příznakem `/embed/`, což značí, že se video zobrazí na celé stránce a bude tedy působit celá stránka pouze jako video. V této variantě už se dá využít `<iframe>` tag, viz. podsekcce 3.2.4, a aplikovat příznak definující rozlišení videa. Příklad použití třídy `WebView` s puštěním YouTube videa je vidět na kódu níže:

---

```
String frameVideo = "<html><body><iframe width=\"100%\"  
    height=\"100%\"  
    src=\"https://www.youtube.com/embed/a4NT5iBFuZs?vq=large\"  
    frameborder=\"0\" allowfullscreen></iframe></body></html>";  
WebView displayYoutubeVideo = (WebView) findViewById(R.id.webview1);  
displayYoutubeVideo.setWebViewClient(new WebViewClient() {  
    @Override  
    public boolean shouldOverrideUrlLoading(WebView view, String url) {  
        return false;  
    }  
});  
displayYoutubeVideo.loadData(frameVideo, "text/html", "utf-8");
```

---

Screenshot z aplikace používající WebView je vidět na obrázku 3.24



Obr. 3.24: Aplikace při použití WebView místo YouTube Android API.

Je použito stejné video jako na obrázku 3.23 v integraci YouTube Android API. Zajímavé je, jak se změnila URL, kterou voláme na <https://www.youtube.com/embed/a4NT5iBFuZs?vq=large>, kde je vidět nutnost vložení /embed/ a ?vq=large pro zobrazení videa ve specifikovaném rozlišení.



Pro testování subjektivního QoE bude zřejmě nutné přidat javascriptové skripty, které budou po určité době měnit rozlišení či streamování.

## 4 ZÁVĚR

V semestrální práci byla zkoumána problematika měření subjektivní kvality zážitku u streamovaných videí pomocí mobilní aplikace využívající YouTube API. V úvodní části je popsána problematika streamingu, základní rozdělení QoE (Quality of Experience), charakteristiky QoE, rozdíly oproti QoS (Quality of Service). Detailněji popsáno subjektivní QoE, metody jeho měření a případné korelace získaných výsledků.

V další části této práce je navržena analýza k vytvářené aplikaci, kde je nastíněno, jak by měla aplikace fungovat, v jakých stavech by se měl účastník testování pro měření subjektivního QoE nacházet, jaké třídy by měly být v aplikaci zahrnuty a návrh databáze, ve které se budou ukládat data z měření aj. Stručně jsou popsány technologie, které by měly být použity pro vývoj aplikace a v neposlední řadě je zde znázorněno jakým způsobem se aplikace vytvářela, jak se integruje YouTube Android API do aplikace a co bylo potřebné nastavit, aby aplikace byla uzpůsobena subjektivnímu měření kvality zážitku. V této části bylo zjištěno, že YouTube Android API se nedá pro měření subjektivního QoE použít, protože nedovoluje z programového kódu měnit rozlišení ani typ streamování. V závislosti na těchto zjištěních bylo navrženo jiné možné řešení, které pomocí třídy `WebView` a `html5` s příslušnými značkami (tagy) iniciuje komunikaci s YouTube API, kde už se pomocí javascriptu dá přizpůsobovat video pro potřeby subjektivního měření.

V další práci by bylo možné zaměřit implementaci mobilní aplikace na komplexní řešení, pomocí kterého by bylo možné získat data z měření. Tato data by byla uložena ve vzdálené databázi s přístupem prostřednictvím například REST API. Bylo by možné také provést řadu laboratorních testů, kde by se získalo množství věrohodných dat. Na základě nich by se následně prováděla analýza např. vlivu adaptivního streamování na subjektivní kvalitu streamovaného videa, a poté by byly výsledky přehledně zobrazeny v grafech a stanoveny patřičné závěry.

# LITERATURA

- [1] MACK, Steve. *Streaming media bible*. New York, NY: Hungry Minds, c2002. ISBN 0764536508.
- [2] HOSSFELD Tobias, Raimund SCHATZ, Michael SEUFERT, Matthias HIRTH, Thomas ZINNER, Phuoc TRAN-GIA. *Quantification of YouTube QoE via Crowdsourcing*. IEEE International Workshop on Multimedia Quality of Experience - Modeling, Evaluation, and Directions (MQoE 2011), Dana Point, CA, USA, December 2011.
- [3] WAMSER, F., D. HOCK, M. SEUFERT, B. STAEHLE, R. PRIES a P. TRAN-GIA. Using buffered playtime for QoE-oriented resource management of YouTube video streaming. In: *Transactions on Emerging Telecommunications Technologies*. 2013, 24(3), s. 288-302. DOI: 10.1002/ett.2636. ISSN 21613915. Dostupné také z: <http://doi.wiley.com/10.1002/ett.2636>
- [4] OZER, Jan. *Streaming Vs. Progressive Download Vs. Adaptive Streaming* [online]. [cit. 2016-10-21]. Dostupné z: <http://www.onlinevideo.net/2011/05/streaming-vs-progressive-download-vs-adaptive-streaming/>
- [5] STOCKHAMMER, Thomas. Dynamic adaptive streaming over HTTP –. In: *Proceedings of the second annual ACM conference on Multimedia systems - MMSys '11* [online]. New York, New York, USA: ACM Press, 2011, s. 133- [cit. 2016-10-16]. DOI: 10.1145/1943552.1943572. ISBN 9781450305181. Dostupné z: <http://portal.acm.org/citation.cfm?doid=1943552.1943572>
- [6] OZER, Jan. *What is Adaptive Streaming?* [online]. [cit. 2016-10-02]. Dostupné z: <http://www.streamingmedia.com/Articles/Editorial/What-Is-.../What-is-Adaptive-Streaming-75195.aspx>
- [7] SCHATZ, Raimund, Tobias HOSSFELD a Pedro CASAS. Passive YouTube QoE Monitoring for ISPs. In: *2012 Sixth International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing* [online]. IEEE, 2012, s. 358-364 [cit. 2016-10-12]. DOI: 10.1109/IMIS.2012.12. ISBN 978-1-4673-1328-5. Dostupné z: <http://ieeexplore.ieee.org/document/6296879/>
- [8] ORGANIZERS, ETRI a IEEE ComSoc SPONSORS. *The 12th International Conference on Advanced Communication Technology ICT for Green Growth and Sustainable Development, Phoenix Park, Korea, Feb. 7-10, 2010, proceedings: ICACT 2010*. Piscataway, N.J: IEEE, 2010. ISBN 9788955191462.

- [9] WU, Hong Ren. QoE Subjective and Objective Evaluation Methodologies. *Multimedia Quality of Experience (QoE)* [online]. Chichester, UK: John Wiley & Sons, Ltd, 2015, s. 123 [cit. 2016-10-28]. DOI: 10.1002/9781118736135.ch6. ISBN 9781118736135. Dostupné z: <http://doi.wiley.com/10.1002/9781118736135.ch6>
- [10] MENKOVSKI, Vlado. Objective QoE Models. *Computational Inference and Control of Quality in Multimedia Services* [online]. 2015, s. 15 [cit. 2016-10-10]. DOI: 10.1007/978-3-319-24792-2\_2. ISBN 978-3-319-24792-2. Dostupné z: [http://link.springer.com/10.1007/978-3-319-24792-2\\_2](http://link.springer.com/10.1007/978-3-319-24792-2_2)
- [11] WINKLER, S. a P. MOHANDAS. The Evolution of Video Quality Measurement: From PSNR to Hybrid Metrics. *IEEE Transactions on Broadcasting* [online]. 2008, 54(3), 660-668 [cit. 2016-10-10]. DOI: 10.1109/TBC.2008.2000733. ISSN 0018-9316. Dostupné z: <http://ieeexplore.ieee.org/document/4550731/>
- [12] PIAMRAT, K., VIHO, C., BONNIN, J.M., KSENTINI, A.: *Quality of Experience Measurements for Video Streaming over Wireless Networks*. (Apr 2009) 1184–1189
- [13] KONEČNÝ, Jakub. *Optimalizace univerzitní bezdrátové sítě pro provoz hlasových služeb*. Brno, 2015. Diplomová práce. Mendelova univerzita v Brně. Vedoucí práce Ing. Petr Zach.
- [14] ITU-T, *Methods for subjective determination of transmission quality*, Recommendation ITU-T P.800., September 1996
- [15] CHOE, Ji-Hwan, Tae-Uk JEONG, Hyun-Soo CHOI, Eun-Jae LEE, Sang-Wook LEE a Chul-Hee LEE. Comparison of subjective video quality assessment methods for multimedia applications. *Journal of Broadcast Engineering* [online]. 2007, 12(2), 177-184 [cit. 2016-10-28]. DOI: 10.5909/JBE.2007.12.2.177. ISSN 1226-7953. Dostupné z: <http://koreascience.or.kr/journal/view.jsp?kj=BSGHC3&py=2007&vnc=v12n2&sp=177>
- [16] ITU-T, *Subjective video quality assessment methods for multimedia applications*, Recommendation ITU-T P.910, April 2008
- [17] ITU-R, *Methodology for the subjective assessment of video quality in multimedia applications*, Recommendation ITU-R BT.1788, 2007

- [18] GLEN, Stephanie. *Pearson Correlation: Definition and Easy Steps for Use* [online]. 2016 [cit. 2016-10-29]. Dostupné z: <http://www.statisticshowto.com/what-is-the-pearson-correlation-coefficient/>
- [19] BOLBOACA, Sorana-Daniela a Lorentz JÄNTSCHI. *Pearson versus Spearman, Kendall's Tau Correlation Analysis on Structure-Activity Relationships of Biologic Active Compounds* [online]. [cit. 2016-10-29]. Dostupné z: [http://ljs.academicdirect.org/A09/179\\_200.htm](http://ljs.academicdirect.org/A09/179_200.htm)
- [20] HIRTH, Matthias, Tobias HOSSFELD, Phuoc TRAN-GIA. *Anatomy of a Crowdsourcing Platform - Using the Example of Microworkers.com*. Workshop on Future Internet and Next Generation Networks (FINGNet), Seoul, Korea, June 2011. Also available as technical report *Human Cloud as Emerging Internet Application - Anatomy of the Microworkers Crowdsourcing Platform*. Technical Report 478, January 2011.
- [21] BIERSECK, Ernst et al.: *Data traffic monitoring and analysis: from measurement, classification, and anomaly detection to quality of experience*. 1st ed. New York: Springer, 2013, pp. 264-301, ISBN 3642367836.
- [22] TONY BUZAN WITH BARRY BUZAN. *The mind map book: how to use radiant thinking to maximize your brain's untapped potential*. New York: Plume, 1996. ISBN 9780452273221.
- [23] ARLOW, Jim a Ila NEUSTADT. *UML 2.0 and the unified process: practical object-oriented analysis and design*. 2nd ed. Boston: Addison-Wesley, c2005. Addison-Wesley Object Technology Series. ISBN 978-0-321-32127-5.
- [24] Google. *Android Studio*. [online] [cit. 2016-11-24]. Dostupné také z: <https://developer.android.com/studio/index.html>
- [25] CHACON, Scott. *Pro Git*. Praha: CZ.NIC, c2009. CZ.NIC. ISBN 978-80-904248-1-4.
- [26] PEARCE, Shawn. *GitSvnComparison* [online]. 2013. vyd. [cit. 2016-09-30]. Dostupné z: <https://git.wiki.kernel.org/index.php/GitSvnComparison>.
- [27] GRIFFITHS, Dawn a David GRIFFITHS. *Head first Android development*. Sebastopol: O'Reilly, 2015. Head first series. ISBN 1449362184.
- [28] SIERRA, Kathy. a Bert. BATES. *Head first Java*. 2nd ed. Sebastopol, CA: O'Reilly, 2005. ISBN 05-960-0920-8.

- [29] FINEGAN, Edward. a Robert. LIGUORI. *OCA Java SE 7 programmer I study guide (exam 1Z0-803)*. New York: McGraw-Hill, 2013. ISBN 0071789421.
- [30] GUPTA, Mala. *OCF Java SE 7 programmer II certification guide: prepare for the 1Z0-804 exam*. Shelter Island, NY: Mannning, 2015. ISBN 9781617291487.
- [31] BERGLUND, Tim a Matthew MCCULLOUGH. *Building and Testing with Gradle* [online]. Sebastopol, CA: O'Reilly Media, Inc., 2011 [cit. 2016-09-30]. Dostupné z: [http://berddk.ru/media/doc/2013/11/27/Building\\_and\\_testing\\_with\\_Gradle.pdf](http://berddk.ru/media/doc/2013/11/27/Building_and_testing_with_Gradle.pdf)
- [32] Oracle, *What is MySQL?* [online]. [cit. 2016-11-13]. Dostupné z: <http://dev.mysql.com/doc/refman/5.7/en/what-is-mysql.html>
- [33] YouTube, *YouTube Developer Documentation* [online]. listopad 16, 2015 [cit. 2016-11-6]. Dostupné také z: <https://developers.google.com/youtube/documentation/>
- [34] YouTube, *YouTube Android Player API* [online]. květen 23, 2015 [cit. 2016-10-4]. Dostupné také z: <https://developers.google.com/youtube/android/player/>
- [35] VUJOVIC, Filip. (2015, prosinec). *How To Integrate YouTube API In Android Application*. Dostupné také z: <https://www.youtube.com/watch?v=a4NT5iBFuZs>

## SEZNAM SYMBOLŮ, VELIČIN A ZKRATEK

ACR	Absolute Category Rating
ACR-HR	Absolute Category Rating with hidden reference
API	Application Programming Interface
Authenticator	Autentizátor
CPU	Central Processing Unit
ECC	Elliptic Curve Cryptography
DB	Databáze
DCR	Degradation Category Rating
DSCQS	The Double-Stimulus Continuous Quality-Scale method
DSIS	Double Stimulus Impairment Scale
DSL	Domain Specific Language
HD	High Definition
HTTP	Hypertext Transfer Protocol
HTML5	Hyper Text Markup Language 5
IDE	Integrated Development Environment
IETF	Internet Engineering Task Force
IP	Internet Protocol
JDK	Java Development Kit
JSON	Javascript Object Notation
MOS	Mean Opinion Score
MPQM	Moving Picture Quality Metric
MSE	Mean Square Error
MTC	Maximal Threshold Correlation
MySQL	My Structured Query Language

NQM	Noise Quality Measure
ORM	Object Relation Mapping
PC	Pair comparison method
PSNR	Peak Signal to Noise Ratio
QoE	Quality of Experience
QoS	Quality of Service
REST	Representational State Transfer
SDK	Software Development Kit
SHA1	Secure Hash Algorithm 1
SQL	Structured Query Language
SSIM	Structural Similarity Index
SVM	Subversion
TCP	Transmission Control Protocol
UDP	User Datagram Protocol
User	Uživatel
VoIP	Voice over Internet Protocol
VQM	Video Quality Metric
XML	Extensible Markup Language
XML-RPC	Extensible Markup Language Remote Procedure

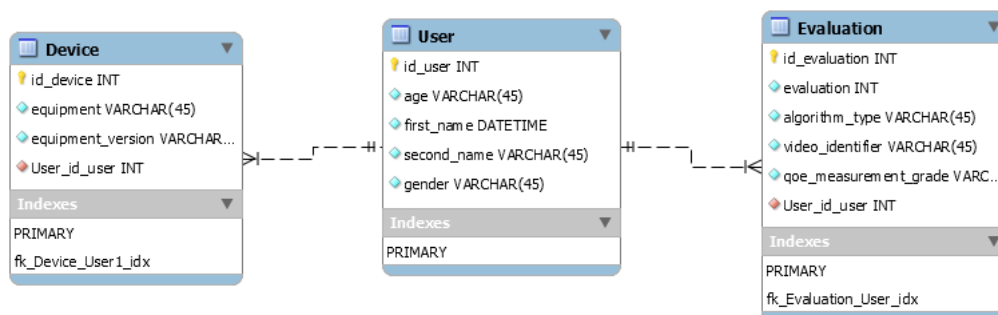


# SEZNAM PŘÍLOH

<b>A</b>	<b>MySQL databáze</b>	<b>66</b>
A.1	Návrh MySQL databáze na základě ERD v MySQLWorkbench . . .	66
A.2	MySQL dotazy k vytvoření databáze . . . . .	66

## A MYSQL DATABÁZE

### A.1 Návrh MySQL databáze na základě ERD v MySQLWorkbench



Obr. A.1: Databáze vytvořená v programu MySQL Workbench.

### A.2 MySQL dotazy k vytvoření databáze

```
-- phpMyAdmin SQL Dump
-- version 3.5.2.2
-- http://www.phpmyadmin.net
-- Verze MySQL: 10.0.22-MariaDB
-- Verze PHP: 5.2.17
```

```
SET SQL_MODE="NO_AUTO_VALUE_ON_ZERO";
SET time_zone = "+00:00";
```

```
--
-- Databáze: 'u785958671_mydb'
--
```

```
-- -----
```

```
--
-- Struktura tabulky 'Device'
--
```

```

CREATE TABLE IF NOT EXISTS 'Device' (
    'id_device' int(11) NOT NULL AUTO_INCREMENT,
    'equipment' varchar(45) COLLATE utf8_unicode_ci NOT NULL,
    'equipment_version' varchar(45) COLLATE utf8_unicode_ci NOT NULL,
    'User_id_user' int(11) NOT NULL,
    PRIMARY KEY ('id_device'),
    KEY 'fk_Device_User1_idx' ('User_id_user')
) ENGINE=MyISAM DEFAULT CHARSET=utf8 COLLATE=utf8_unicode_ci
    AUTO_INCREMENT=1;

-- -----

--
-- Struktura tabulky 'Evaluation'
--

CREATE TABLE IF NOT EXISTS 'Evaluation' (
    'id_evaluation' int(11) NOT NULL AUTO_INCREMENT,
    'evaluation' int(11) NOT NULL,
    'algorithm_type' varchar(45) COLLATE utf8_unicode_ci NOT NULL,
    'video_identifier' varchar(45) COLLATE utf8_unicode_ci NOT NULL,
    'qoe_measurement_grade' varchar(45) COLLATE utf8_unicode_ci NOT NULL,
    'User_id_user' int(11) NOT NULL,
    PRIMARY KEY ('id_evaluation'),
    KEY 'fk_Evaluation_User_idx' ('User_id_user')
) ENGINE=MyISAM DEFAULT CHARSET=utf8 COLLATE=utf8_unicode_ci
    AUTO_INCREMENT=1;

-- -----

--
-- Struktura tabulky 'User'
--

CREATE TABLE IF NOT EXISTS 'User' (
    'id_user' int(11) NOT NULL AUTO_INCREMENT,
    'age' varchar(45) COLLATE utf8_unicode_ci NOT NULL,
    'first_name' datetime NOT NULL,
    'second_name' varchar(45) COLLATE utf8_unicode_ci NOT NULL,
    'gender' varchar(45) COLLATE utf8_unicode_ci NOT NULL,
    PRIMARY KEY ('id_user')

```

```
) ENGINE=MyISAM DEFAULT CHARSET=utf8 COLLATE=utf8_unicode_ci  
  AUTO_INCREMENT=1;
```

---