



Pedagogická  
fakulta  
Faculty  
of Education

Jihočeská univerzita  
v Českých Budějovicích  
University of South Bohemia  
in České Budějovice

## **Jihočeská univerzita v Českých Budějovicích**

Pedagogická fakulta

Katedra informatiky

### **Ovládání digitální zrcadlovky prostřednictvím mobilního zařízení**

### **Controlling of a Canon DSLR via a mobile device**

Bakalářská práce

**Vypracoval:** Petr Minář, DiS.

**Vedoucí práce:** Mgr. Václav Šimandl, Ph.D.

České Budějovice 2021

# Zadání bakalářské práce

JIHOČESKÁ UNIVERZITA V ČESKÝCH BUDĚJOVICÍCH

Pedagogická fakulta

Akademický rok: 2019/2020

## ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(projektu, uměleckého díla, uměleckého výkonu)

Jméno a příjmení: Petr MINÁŘ, DiS.  
Osobní číslo: P18481  
Studijní program: B7507 Specializace v pedagogice  
Studijní obor: Informační technologie a e-learning  
Téma práce: Ovládání digitální zrcadlovky prostřednictvím mobilního zařízení  
Zadávací katedra: Katedra informatiky

### Zásady pro vypracování

Cílem této bakalářské práce je vytvořit mobilní aplikaci pro platformu Android, prostřednictvím které by bylo možno ovládat digitální zrcadlovku značky Canon. Propojení mobilního telefonu a fotoaparátu bude provedeno pomocí Wi-Fi. Aplikace pro ovládání fotoaparátu bude umožňovat detekovat a připojit nebo odpojit fotoaparát, zobrazit informace o fotoaparátu (úroveň nabití baterie, volné místo na paměťové kartě), provádět vybraná nastavení fotoaparátu (upravení parametrů expozice, ovládání blesku, nastavení hodnoty ISO), zobrazení obrazu živého náhledu fotoaparátu, spuštění základních úkonů (zaostření, okamžité a odložené pořízení snímku, nastavení zoomu), komplexnější akce (časoběrné, sekvenční snímání, expozice s různými parametry) a možnost natáčení videa. Zvolená nastavení bude aplikace schopna uložit jako speciální tematické režimy pro pozdější použití. Dále bude aplikace podporovat přenos a uložení fotografií z fotoaparátu do mobilního telefonu. Programování aplikace se bude provádět na základě SDK balíčku vydaného společností Canon. Na základě průběžného testování vytvářené aplikace student vyhodnotí možnosti a meze ovládání digitální zrcadlovky z jiného zařízení prostřednictvím Wi-Fi.

Rozsah pracovní zprávy: 40  
Rozsah grafických prací: CD ROM  
Forma zpracování bakalářské práce: tištěná

### Seznam doporučené literatury:

1. Canon Inc. Camera Control API OperationGuide. Canon, 2019.
2. Canon Inc. Camera Control API Reference. Ver. 1.0.0. Canon, 2019.
3. Canon Inc. EDSKD 13.10.0 API Programming Reference. Canon, 2019.
4. Codelabs: Dokumentace k programování pro Android [online]. Google, 2020 [cit. 2020-04-01]. Dostupné z: <https://codelabs.developers.google.com/>
5. Herout, P. Testování pro programátory. České Budějovice: Kopp, 2016. ISBN 978-80-7232-481-1.
6. Lacko, L. Mistrovství Android: Kompletní průvodce vývojáře. Brno: Computer Press, 2017. ISBN 978-80-251-4875-4.
7. Lacko, L. Vývoj aplikací pro Android. 1. vyd. Brno: Computer Press, 2015. ISBN 978-80-251-4347-6.

Vedoucí bakalářské práce: Mgr. Václav Šimandl, Ph.D.  
Katedra informatiky



## Prohlášení

Prohlašuji, že svoji bakalářskou práci jsem vypracoval samostatně pouze s použitím pramenů a literatury uvedených v seznamu citované literatury.

Prohlašuji, že v souladu s § 47b zákona č. 111/1998 Sb. v platném znění souhlasím se zveřejněním své bakalářské práce, a to v nezkrácené podobě elektronickou cestou ve veřejně přístupné části databáze STAG provozované Jihočeskou univerzitou v Českých Budějovicích na jejích internetových stránkách, a to se zachováním mého autorského práva k odevzdanému textu této kvalifikační práce. Souhlasím dále s tím, aby toutéž elektronickou cestou byly v souladu s uvedeným ustanovením zákona č. 111/1998 Sb. zveřejněny posudky školitele a oponenta práce i záznam o průběhu a výsledku obhajoby kvalifikační práce. Rovněž souhlasím s porovnáním textu mé kvalifikační práce s databází kvalifikačních prací Theses.cz provozovanou Národním registrem vysokoškolských kvalifikačních prací a systémem na odhalování plagiátů.

V Českých Budějovicích dne .....

.....

Petr Minář

## **Anotace**

Cílem této bakalářské práce je vytvořit mobilní aplikaci pro platformu Android, prostřednictvím které by bylo možno ovládat digitální zrcadlovku značky Canon. Propojení mobilního telefonu a fotoaparátu je provedeno pomocí Wi-Fi. Aplikace pro ovládání fotoaparátu umožňuje detekovat a připojit nebo odpojit fotoaparát, zobrazit informace o fotoaparátu (úroveň nabití baterie, volné místo na paměťové kartě, připojení objektivu), může provádět vybraná nastavení fotoaparátu (upravení parametrů expozice, ovládání blesku, nastavení hodnoty ISO, nastavení barevného stylu fotografie), zobrazení obrazu živého náhledu fotoaparátu, spuštění základních úkonů (zaostření, okamžité a odložené pořízení snímku, nastavení zoomu), komplexnější akce (sekvenční snímání, expozice s různými parametry) a možnost natáčení videa. Zvolená nastavení je aplikace schopna uložit jako speciální tematické režimy pro pozdější použití. Dále aplikace podporuje přenos a uložení fotografií z fotoaparátu do mobilního telefonu. Programování aplikace je prováděno na základě SDK balíčku vydaného společností Canon. Na základě testování vytvářené aplikace jsem vyhodnotil možnosti a meze ovládání digitální zrcadlovky z jiného zařízení prostřednictvím Wi-Fi. V porovnání s aplikací od Canonu moje aplikace umožňuje více možností nastavení a také uživatelská nastavení, jednotlivých nastavení fotoaparátu.

## **Klíčová slova**

Mobilní aplikace, OS Android, Canon Control API, SDK, Vývoj aplikace, ovládání DSLR

## **Abstract**

The aim of this bachelor's thesis is to create a mobile application for the Android platform, through which it would be possible to control a Canon digital SLR camera. The connection between the mobile phone and the camera is made via Wi-Fi. The camera control application allows you to detect and connect or disconnect the camera, view camera information (battery level, free space on memory card, lens connection), can make selected camera settings (adjust exposure parameters, flash control, set ISO value, set color photo style), display a live view image of the camera, start basic operations (focus, immediate and delayed image capture, zoom adjustment), more complex actions (sequential shooting, exposure with various parameters) and the ability to record video. The application is able to save the selected settings as special thematic modes for later use. In addition, the application supports the transfer and storage of photos from the camera to a mobile phone. Application programming is based on an SDK package issued by Canon. Based on the testing of the created application, I evaluated the possibilities and limits of controlling a digital SLR camera from another device via Wi-Fi. Compared to the Canon application, my application allows more setting options as well stored user settings of individual camera settings.

## **Keywords**

Mobile application, OS Android, Canon Control API, SDK, Application development, controlling DSLR

## **Poděkování**

Rád bych poděkoval vedoucímu mé bakalářské práce panu Mgr. Václavu Šimandlovi, Ph.D. za rady, ochotu a odborné vedení práce. Dále panu Mgr. Jakubu Geyerovi za možnost přístupu k fotoaparátu a panu PhDr. Milanu Novákovi, Ph.D. za zapůjčení fotoaparátu.

# Obsah

1.	Úvod.....	10
1.1.	Cíle práce .....	10
1.2.	Metody práce.....	10
2.	Operační systém Android .....	12
2.1.	Vývoj OS Android .....	13
2.1.1.	Android 5 – Lollipop .....	14
2.1.2.	Android 6 – Marshmallow .....	14
2.1.3.	Android 7 a 7.1 – Nougat.....	15
2.1.4.	Android 8 a 8.1 – Oreo .....	15
2.1.5.	Android 9 – Pie .....	16
2.1.6.	Android 10 .....	16
2.1.7.	Android 11 .....	17
2.2.	Základní části Android aplikací .....	17
2.2.1.	Manifest .....	18
2.2.2.	Gradle.....	19
2.2.3.	Aktivity .....	19
2.2.4.	Fragment .....	20
2.2.5.	Navigace .....	21
2.2.6.	Layout .....	22
2.2.7.	ViewModel .....	22
2.2.8.	Shared Preferences.....	23
2.2.9.	Saved Instances.....	24
3.	Android Studio.....	25
3.1.	Android emulátor .....	25
3.1.1.	Android Virtual Device (AVD) .....	26



4.	Canon Camera Control API.....	27
5.	Praktická část .....	29
5.1.	Návrh uživatelského prostředí.....	29
5.2.	Princip fungování aplikace.....	30
5.2.1.	Živý náhled .....	33
5.2.2.	Funkce fotoaparátu .....	36
5.2.3.	Prohlížení obsahu.....	37
5.3.	Ukládání uživatelských nastavení fotoaparátu.....	39
5.4.	Řešené problémy .....	40
5.5.	Testování využitelnosti aplikace .....	41
5.6.	Porovnání s aplikací od Canonu.....	43
6.	Závěr .....	45
	Seznam použité literatury a zdrojů .....	46
	Seznam obrázků.....	49
	Přílohy.....	50

## 1. Úvod

Ve své práci se chci zaměřit na vývoj aplikace, která bude přívětivá pro uživatele, a navíc nabídne hodně funkcí, aby se co nejvíce zvýšila její využitelnost. Práce bude vycházet z API, který vydala společnost Canon pro komunikaci s jejími fotoaparáty, co tuto funkci podporují. Jelikož jejich vlastní aplikace nemá moc dobrá hodnocení mezi uživateli, budu se snažit v aplikaci napravit některé nedostatky, které obsahuje. Vývoj bude probíhat pro platformu Android s důrazem na použití nejnovějších technologií, které se na ní dají používat. V úvahu se také musí brát měnící se displeje mobilních telefonů v posledních letech kdy se zvětšuje úhlopříčka a mění se poměr stran displeje. Musí se tedy zajistit, aby se aplikace správně zobrazovala na všech zařízeních, ale také aby se co nejlépe využilo místo na displeji telefonu.

### 1.1. Cíle práce

Cílem této bakalářské práce je vytvořit mobilní aplikaci pro platformu Android, prostřednictvím které by bylo možno ovládat digitální zrcadlovku značky Canon. V teoretické části se budu zabývat platformou Android a představením samotného API. Cílem praktické části práce je navrhnout a naprogramovat mobilní aplikaci, která by měla zvládat základní i složitější fotografické úkony, zobrazovat živý náhled z fotoaparátu, prohlížení fotografií uložených na fotoaparátu i jejich stažení do mobilního telefonu. Jednotlivá nastavení fotoaparátu by si měl mít uživatel možnost uložit jako speciální tematické režimy pro pozdější využití. Dalším cílem je pomocí testování zjistit možnosti využití aplikace v běžném prostředí. Dále porovnam mnou naprogramovanou aplikaci s aplikací vydávanou společností Canon.

### 1.2. Metody práce

V úvodu práce jsem prostudoval možnosti a architekturu operačního systému Android. Dále jsem se zabýval všemi potřebnými nástroji k vývoji aplikace, mezi které se dají zařadit Java Development Kit, Software Development Kit, Android Virtual Device a jako vývojové prostředí jsem zvolil Android Studio. Poté jsem se zabýval funkcionalitou Canon Camera Control API a jejími charakteristikami. Následně jsem se pokračoval samotným návrhem a programování aplikace pro ovládání DSLR fotoaparátu. Po jejím

dokončení bylo důležitým faktorem zhodnocení možností a mezí používání této aplikace v běžném prostředí při různých fotografických situacích. Z tohoto testování by mělo být zřejmé, při kterých situacích se vyplatí využívat aplikaci a kdy bude lepší focení přímo z fotoaparátu. (Aplikace se hodí pro stacionární focení na stativu, naopak se příliš nehodí pro focení rychle se pohybujících věcí.) Nakonec jsem ještě provedl porovnání mnou vytvořené aplikace s aplikací, kterou vydává přímo společnost Canon.

## 2. Operační systém Android

Operační systém Android je mobilní operační systém vyvinutý společností Google, který se používá především pro zařízení s dotykovou obrazovkou, mobilní telefony a tablety. Jeho design umožňuje uživatelům intuitivně manipulovat s mobilními zařízeními pomocí pohybů prstů, které odrážejí běžné pohyby, jako je stlačení, přejetí prstem a dotyk obrazovky. Google také používá software pro Android v televizorech, automobilech a náramkových hodinkách, kde je každý z nich je vybaven jedinečným uživatelským rozhraním [1].

Android byl poprvé vyvinut společností Android, Inc., softwarovou společností se sídlem v Silicon Valley, než jej společnost Google získala v roce 2005. Investoři a analytici elektronického průmyslu zpochybňovali skutečné záměry společnosti Google vstoupit do prostoru mobilního trhu od koupi společnosti Android. V každém případě však brzy poté společnost Google oznámila blížící se uvedení svého prvního komerčně dostupného zařízení se systémem Android v roce 2007, ačkoli se telefon s tímto systémem na trh dostal v roce 2008 [1].

Od té doby mohou vývojáři softwaru a aplikací využívat technologii Androidu k vývoji mobilních aplikací, které se prodávají prostřednictvím obchodů s aplikacemi, jako je Google Play. A protože je vyvíjen jako produkt Google, uživatelé Androidu dostávají příležitost propojit svá mobilní zařízení s jinými produkty Google, jako je cloudové úložiště, e-mailové platformy a video služby [1].

Android se v průběhu let vyvíjel a rychle se stal dominantní mobilní platformou po celém světě. Podle StatCounter má Android v listopadu 2020 více než 71% podíl na globálním trhu mobilních operačních systémů. Android není telefon nebo aplikace, ale operační systém založený na linuxovém jádře. A také není jen verzí Linuxu kvůli mnoha změnám pod kapotou. Android je operační systém navržený s ohledem na mobilní zařízení, jako místo, kde žijí funkce a aplikace vašeho telefonu. Všechno, co vidíte na displeji vašeho zařízení, je součástí operačního systému. Když dostanete hovor, textovou zprávu nebo e-mail, operační systém tyto informace zpracuje a umístí je do čitelného formátu [2].

Operační systém Android je rozdělen na různé verze, které jsou číslovány, což znamená značné skoky ve funkcích, provozu a stabilitě a obvykle mají svoje přezdívky. Pokud tedy uslyšíte někoho říkat Android Marshmallow, Nougat, Oreo, Pie nebo nejnovější Android 11, je to jen název verze Androidu, kterou ve svém zařízení můžete mít. Moderní

smartphony a tablety vydané v roce 2020 většinou používají Android 10 nebo nejnovější Android 11 [2].

Výrobci zařízení s Androidem, jako jsou Samsung, Xiaomi, Sony, OnePlus a řada dalších, obvykle mají vzhledovou nastavbu nad operačním systémem. Vzhled nebo překrytí uživatelského rozhraní je v podstatě vlastní design, který přidává do vašeho telefonu další funkce, různé ikony a další vylepšení navržené tak, aby poskytovaly zážitek jedinečný pro vámi vybraného výrobce telefonu. Mezi nejoblíbenější nadstavby patří Samsung One UI a OnePlus OxygenOS. Telefon bez jakýchkoli větších úprav se obecně označuje jako „standardní Android“ a nadstavby společností s pouze malými změnami (například u telefonů Motorola) se často nazývají téměř čistým Androidem [2].

## 2.1. Vývoj OS Android

V roce 2003 založili Rich Miner, Nick Sears, Chris White a Andy Rubin společnost s názvem Android Inc. Název pochází z přezdívky Andyho Rubina, když pracoval ve společnosti Apple. Rubin měl vášeň pro roboty Android. Zakladatelé spustili Android, aby mohli vyvíjet „chytřejší mobilní zařízení, která si více uvědomují polohu a preference svého vlastníka“ [3].

V příštích dvou letech se společnost snažila prosadit v odvětví smartphonů. To bylo do té doby, než společnost Google koupila společnost v roce 2005. Rubin a jeho spoluzakladatelé zůstali s plným přístupem k vývoji a celé řadě produktů Google. Jedním z největších rozhodnutí učiněných v době akvizice bylo použití Linuxu jako základu pro operační systém Android [3].

V roce 2007 společnost Apple uvedla na trh první iPhone a zahájila novou éru v oblasti mobilních telefonů. V té době Google stále pracoval na Androidu tajně, ale v listopadu téhož roku společnost začala pomalu odhalovat své plány konkurovat Apple a dalším mobilním platformám. Jako hlavní vývojář vedl Google také vznik takzvané Open Handset Alliance [4].

Při tiskové konferenci předseda představenstva a generální ředitel společnosti Google Eric Schmidt uvedl: „Dnešní oznámení je ambicióznější než jakýkoli jiný „Google Phone“, o kterém v posledních týdnech tisk spekuluje. Naší vizí je, že výkonná platforma, kterou představujeme, bude pohánět tisíce různých modelů telefonů.“ [4]

Veřejná beta verze systému Android verze 1.0 byla pro vývojáře spuštěna 5. listopadu 2007. V září 2008 byl oznámen vůbec první smartphone s Androidem: T-Mobile G1. OS Android 1.0 obsahoval již několik aplikací Googlu pro mobilní zařízení. Integrovala řadu dalších produktů a služeb společnosti, včetně Map Google, YouTube a prohlížeče HTML (před Chrome), který samozřejmě používal vyhledávací služby Google. Měl také první verzi Android Market, obchodu s aplikacemi, který Google hrdě uvedl jako, „desítky jedinečných a prvotřídních aplikací pro Android.“ Všechny tyto funkce teď zní docela primitivně, ale to byl jen začátek růstu Androidu na trhu mobilních zařízení [4].

### **2.1.1. Android 5 – Lollipop**

Google v zásadě znovuobjevil Android s vydáním Android 5.0 Lollipop s API verzí 21 na podzim roku 2014. Společnost v Androidu 5 představila doposud platný standard Material Design, který přinesl zcela nový vzhled, který se rozšířil napříč všemi Androidy, jejich aplikacemi, a dokonce i dalšími Google produkty [5].

Koncept založený na kartách, který byl rozptýlen po celém systému, se stal základním vzorem uživatelského rozhraní, kterým by řídil vzhled všech věcí od oznámení, která se nyní zobrazovala na uzamčené obrazovce pro rychlý přístup, až po seznam s nedávnými aplikacemi, který také získal kartový vzhled [5].

Lollipop představil řadu nových funkcí pro Android, včetně skutečně hlasového ovládání hands-free pomocí příkazu „OK, Google“, podpory více uživatelů na jednom telefonu a prioritního režimu pro lepší správu oznámení. Změnilo se to natolik, bohužel, že to také představilo spoustu znepokojujících chyb, z nichž mnohé byly plně odstraněny až v příštím vydání 5.1 [5].

### **2.1.2. Android 6 – Marshmallow**

Podle dnešních měřítek byl Marshmallow z roku 2015 poměrně malým vydáním systému Android a vypadal spíš jako aktualizace na úrovni 0.1, než cokoli, co si zaslouží plné číslo. Ale touto verzí začal trend, že Google vydává jednu hlavní verzi systému Android ročně a tato verze vždy dostává své vlastní celé číslo [5].

Nejpřitažlivějším prvkem Marshmallow byla funkce vyhledávání pomocí kliknutí na obrazovku s názvem Now On Tap, tato funkce měla hodně potenciálu, který nebyl plně

využit. Google systém nikdy nezdokonalil a následující rok tuto funkci zařadil na vedlejší kolej [5].

### **2.1.3. Android 7 a 7.1 – Nougat**

Verze Google Android Nougat z roku 2016 poskytly Androidu nativní režim rozdělené obrazovky, nový systém dělaný pro organizaci oznámení a funkci Data Saver. Nougat přidal také některé menší, ale stále významné funkce, například klávesovou zkratku podobnou Alt-Tab pro přecházení mezi aplikacemi [5].

Snad nejvýznamnějším z vylepšení Nougatu však bylo spuštění Google Asistent, ten přišel spolu s oznámením prvního vyrobeného telefonu přímo společností Google. Pixel byl uveden asi dva měsíce po debutu Nougata. Google Asistent se stal kritickou součástí Androidu a většiny ostatních produktů Google a je dnes pravděpodobně nejdůležitější vývojovou větví společnosti [5].

### **2.1.4. Android 8 a 8.1 – Oreo**

Android Oreo přidal na platformu celou řadu drobností, včetně nativního režimu obraz v obraze, možnosti odložení oznámení a kanálů oznámení, které nabízejí kvalitnější kontrolu nad tím, jak vás aplikace mohou upozornit [5].

Vydání z roku 2017 zahrnovalo také některé pozoruhodné prvky, které podpořily cíl společnosti Google sladit Android a Chrome OS a zlepšit zážitek z používání aplikací pro Android v Chromebooku, a byla to první verze pro Android, která obsahovala Project Treble, jenž se snažil o vytvoření modulární základny pro kód Androidu s nadějí, že usnadní výrobcům zařízení včasné aktualizace softwaru [5].

Další přidanou funkcí bylo automatické vyplňování, které si s vaším svolením pamatovalo vaše přihlašovací údaje, abyste se mohli rychleji dostat do aplikací. V této verzi byla také představena funkce Oznamovací tečky, pomocí které rychle zjistíte, co je nového, a snadno jí vymažete přejetím prstem pryč. Ty začala rychle využívat aplikace Messenger pro zobrazování jednotlivých konverzací na obrazovce telefonu [6].

### 2.1.5. Android 9 – Pie

Android Pie, byl představen v srpnu roku 2018. Největší změnou byl jeho hybridní navigační systém gesta / tlačítka, který se změnil z tradičních kláves Android Back, Home a Overview na velké, multifunkční tlačítko Domů a malé tlačítko Zpět, které se podle potřeby objevilo vedle něj. Pie zahrnoval také některé pozoruhodné funkce produktivity, jako je univerzální systém doporučených odpovědí pro zasilání zpráv a inteligentnější systémy pro správu napájení a automatického jasu obrazovky. Dále chytřejší způsob zpracování hotspotů Wi-Fi, vylepšený režim úspory baterie v systému Android a řady vylepšení ochrany osobních údajů a zabezpečení [5].

Telefony se stávali velice populární a pro někoho se i jeho používání stávalo návykovou záležitostí, proto Google v této verzi uvedl Digitální rovnováhu, ve které bylo vidět kolik času trávíme v jednotlivých aplikacích a jaký celkový čas trávíme na mobilním zařízení. Byla i zde možnost stanovit časový limit pro používání určité aplikace, kde si můžete například určit, že můžete na Instagram koukat pouze hodinu denně a až vyprší čas používání, aplikace se pozastaví. Její ikona na domovské obrazovce bude šedá a nebudete z ní dostávat žádná oznámení. Následující den se časovač resetuje a aplikace se znovu odemkne [7].

### 2.1.6. Android 10

Android 10 byl první verzí pro Android, která se zbavila svého písemného označení a byla známá jednoduše pod číslem v září 2019. Tato verze přináší zcela přepracované rozhraní pro ovládací gesta systému Android, které je teď ovládáno pomocí přejíždění prstu po obrazovce [5].

Pod kapotou představuje Android 10 nové nastavení pro aktualizace ve stylu hot-fix, které nakonec umožňují rychlejší a konzistentnější zavádění malých úzce zaměřených oprav. Software má spoustu dalších důležitých vylepšení, včetně aktualizovaného systému oprávnění, který vám dává větší kontrolu nad tím, jak přesně a kdy budou aplikace schopny přistupovat k údajům o poloze, a rozšířený systém pro ochranu jedinečných identifikátorů zařízení, které lze použít k sledování vaší aktivity na zařízení v průběhu času [5].

Kromě toho Android 10 obsahuje temné téma pro celý systém, které šetří baterii u telefonu s OLED displeji. Toto nastavení můžete mít spuštěné po celý den, nebo se dá



nastavit jenom pro určité hodiny, kdy hlavně večer může být šetrnější při sledování obrazovky. U aplikací, které toto nastavení podporují může navíc tento mód změnit vzhled celé aplikace a přepnout jí na tmavé barvy [8].

### **2.1.7. Android 11**

Android 11, který byl uvedený na trh na začátku září 2020, je docela podstatná aktualizace systému Android jak pod kapotou, tak i na povrchu. Nejvýznamnější změny verze se točí kolem soukromí. Aktualizace staví na rozšířeném systému oprávnění zavedeném v systému Android 10 a přidává uživatelům možnost udělit aplikacím určitá oprávnění, která souvisejí s přístupem k poloze, přístupem k fotoaparátu a přístupem k mikrofonu, u kterých můžeme povolit používání stále, pouze při používání aplikace nebo jenom omezeně na jediné použití tohoto přístupu [5].

Android 11 také posouvá oprávnění k poloze na pozadí ještě hlouběji do systému a ztěžuje aplikacím její vyžádání. A zavádí novou funkci, ve které aplikacím, jež zůstaly neotevřené po dobu několika měsíců, budou automaticky zrušena jejich oprávnění, pokud se aktivně nerozhodneme je znovu autorizovat. Kromě toho Android 11 odstraňuje schopnost aplikace zjistit, jaké další aplikace jsou nainstalovány ve vašem telefonu [5].

A pokud jde o viditelnější funkce zaměřené na uživatele, Android 11 vylepšuje oblast oznámení systému, aby zdůraznila a zjednodušila upozornění související s konverzacemi. Zavádí nový efektivní multimediální přehrávač, který obsahuje ovládací prvky pro všechny aplikace pro přehrávání zvuku a videa na jednom místě. A přidává do nové kontextové nabídky ovládacích prvků připojeného zařízení pro všechny produkty chytré domácnosti spojené s vaším účtem (i když některé z těchto funkcí vyžadují trochu manuální úpravy, aby fungovaly optimálně). Android 11 má spoustu dalších malých, ale významných vylepšení včetně nové sekce Historie oznámení, nativní funkce nahrávání obrazovky a automatizovaného plánovacího systému pro temné téma celého systému [5].

## **2.2. Základní části Android aplikací**

V následujících kapitolách se budu věnovat základním částem Android aplikace, které se používají při jejím vývoji.

### 2.2.1. Manifest

Každý projekt aplikace musí mít soubor `AndroidManifest.xml` (s přesně tímto názvem) v kořenovém adresáři sady zdrojových souborů projektu. Soubor manifestu popisuje základní informace o aplikaci pro nástroje Android Build, operační systém Android a Google Play [9].

Soubor `AndroidManifest.xml` obsahuje informace o balíčkách, včetně komponent aplikace, jako jsou aktivity, služby, přijímače přenosového vysílání či poskytovatelé obsahu [10].

Spravuje také oprávnění, která aplikace potřebuje pro přístup k chráněným částem systému nebo jiným aplikacím. Rovněž deklaruje všechna oprávnění, která ostatní aplikace musí mít, pokud chtějí získat přístup k obsahu z této aplikace. Obsahuje informace o hardwarových a softwarových funkcích, jež aplikace vyžaduje, a to ovlivňuje, která zařízení mohou aplikaci nainstalovat z obchodu Google Play. Rovněž deklaruje Android API, kterou bude aplikace používat [9].

Soubor `AndroidManifest.xml` obsahuje části:

`<manifest>` - manifest je kořenový prvek souboru `AndroidManifest.xml`. Má atribut `balíčku`, který popisuje název balíčku třídy aktivity [10].

`<application>` - aplikace je dílčím prvkem manifestu. Zahrnuje deklaraci oboru názvů. Tento prvek obsahuje několik dílčích prvků, které deklarují komponenty aplikace, například aktivity [10].

`<activity>` - aktivita je dílčí prvek aplikace a představuje aktivitu, kterou je třeba definovat v souboru `AndroidManifest.xml`. Má mnoho atributů, jako je štítek, název, téma, `launchMode` [10].

`android:label` představuje štítek, který se zobrazuje na obrazovce.

`android:name` představuje název třídy aktivity. Je to povinný atribut.

`<intent-filter>` - `intent-filter` je dílčí prvek aktivity, který popisuje typ záměru, na který může aktivita, služba nebo přijímač přenosového vysílání reagovat [10].

`<action>` - Přidá akci pro `intent-filter`. `Intent-filter` musí obsahovat alespoň jeden prvek akce [10].

`<category>` - Přidá název kategorie do `intent-filteru` [10].

### 2.2.2. Gradle

Systém sestavení Android kompiluje prostředky aplikace a zdrojový kód a balí jej do souborů APK, které lze testovat, nasazovat, podepisovat a distribuovat. Android Studio používá Gradle, sadu nástrojů pro pokročilé sestavení, k automatizaci a správě procesu sestavení a zároveň umožňuje definovat flexibilní vlastní konfigurace sestavení. Každá konfigurace sestavení může definovat vlastní sadu kódu a prostředků, zatímco znovu používá části společné pro všechny verze aplikace. Plugin Android pro Gradle pracuje se sadou nástrojů pro sestavení a poskytuje procesy a konfigurovatelná nastavení, která jsou specifická pro vytváření a testování aplikací pro Android [11].

Každý projekt pro Android potřebuje gradle pro generování APK ze souborů .java a .xml v projektu. Jednoduše řečeno, Gradle vezme všechny zdrojové soubory (java a XML) a použije vhodné nástroje, např. Převěde java soubory na dex soubory a všechny zkomprimuje do jediného souboru známého jako APK, který se používá v mobilních zařízeních [12].

### 2.2.3. Activity

Třída Activity je zásadní součástí aplikace pro Android a způsob spouštění a sestavování aktivit je základní součástí aplikačního modelu platformy. Na rozdíl od programovacích paradigmat, ve kterých jsou aplikace spouštěny metodou main(), systém Android inicializuje kód v instanci třídy Activity vyvoláním konkrétních metod zpětného volání, které odpovídají konkrétním fázím jeho životního cyklu [13].

Zkušenosti s mobilní aplikací se liší od desktopového protějšku v tom, že interakce uživatele s aplikací nemusí vždy začínat na stejném místě. Místo toho začíná cesta uživatele často nedeterministicky. Například pokud otevřete e-mailovou aplikaci z domovské obrazovky, může se zobrazit seznam e-mailů. Naproti tomu, pokud používáte aplikaci pro sociální média, která poté spustí vaši e-mailovou aplikaci, můžete přejít přímo na obrazovku e-mailové aplikace pro psaní e-mailu [13].

Třída Activity je navržena k usnadnění tohoto paradigmatu. Když jedna aplikace vyvolá jinou, vyvolá volající aplikace aktivitu v druhé aplikaci, nikoli aplikaci jako atomický celek. Tímto způsobem aktivita slouží jako vstupní bod pro interakci aplikace s uživatelem. Aktivitu implementujete jako podtřídu třídy Activity [13].

Aktivita poskytuje okno, ve kterém aplikace kreslí své uživatelské rozhraní. Toto okno obvykle vyplňuje obrazovku, ale může být menší než obrazovka a vznášet se nad ostatními okny. Obecně platí, že jedna aktivita implementuje jednu obrazovku v aplikaci. Například jedna z aktivit aplikace může implementovat obrazovku Předvolby, zatímco jiná aktivita implementuje obrazovku Vybrat fotografii [13].

Většina aplikací obsahuje více obrazovek, což znamená, že obsahují více aktivit. Typicky je jedna aktivita v aplikaci brána jako hlavní aktivita, což je první obrazovka, která se zobrazí, když uživatel spustí aplikaci. Každá aktivita pak může zahájit jinou aktivitu, aby mohla provádět různé akce. Například hlavní aktivita v jednoduché e-mailové aplikaci může poskytovat obrazovku, která zobrazuje e-mailovou schránku. Odtud může hlavní aktivita spustit další aktivity, co poskytují obrazovky pro úkoly, jako je psaní e-mailů a otevírání jednotlivých e-mailů [13].

Ačkoli aktivity společně vytvářejí soudržný uživatelský zážitek v aplikaci, každá aktivita je pouze volně vázána na ostatní aktivity, to znamená, že mezi aktivitami v aplikaci jsou obvykle minimální závislosti. Ve skutečnosti aktivity často spouštějí aktivity patřící k jiným aplikacím. Například aplikace v prohlížeči může spustit aktivitu sdílení aplikace pro sociální média [13].

Chcete-li ve své aplikaci používat aktivity, musíte o nich zaregistrovat informace v manifestu aplikace a musíte odpovídajícím způsobem spravovat životní cyklus aktivit [13].

#### **2.2.4. Fragment**

Fragment představuje opakovaně použitelnou část uživatelského rozhraní aplikace. Je definován a spravován vlastním rozvržením, má svůj vlastní životní cyklus a může zpracovávat své vlastní vstupní události. Fragments nemohou žít samy, musí být hostitelem aktivity nebo jiného fragmentu. Hierarchie zobrazení fragmentu se stává součástí hierarchie pohledu hostitele nebo se k ní připojuje [14].

Fragments zavádějí do uživatelského rozhraní aktivity modularitu a opakovanou použitelnost tím, že umožňují rozdělit uživatelské rozhraní na diskrétní bloky. Aktivity jsou ideálním místem k umístění globálních prvků kolem uživatelského rozhraní aplikace, například navigační šuplík (výsuvná postranní lišta s dalšími nastaveními). Naopak,

fragmenty se lépe hodí k definování a správě uživatelského rozhraní jedné obrazovky nebo části obrazovky [14].

Pokud máme aplikaci, která reaguje na různé velikosti obrazovky. Na větších obrazovkách by aplikace měla zobrazit šuplík statické navigace a seznam v grid layoutu. Na menších obrazovkách by aplikace měla zobrazovat spodní navigační lištu a seznam v lineárním rozvržení. Správa všech těchto variací v aktivitách může být nepraktická. Oddělení navigačních prvků od obsahu může tento proces zlepšit. Aktivita je pak zodpovědná za zobrazení správného uživatelského rozhraní navigace, zatímco fragment zobrazí seznam se správným rozložením [14].

Rozdělení uživatelského rozhraní na fragmenty usnadňuje úpravy vzhledu aktivity za běhu. I když je aktivita ve stavu spuštěného životního cyklu, lze fragmenty přidávat, nahrazovat nebo i odebírat [14].

Fragment má vlastní rozvržení a vlastní chování s vlastními zpětnými voláními životního cyklu. Lze kombinovat více fragmentů v jedné aktivitě a vytvořit uživatelské rozhraní s více podokny. Fragmenty lze použít ve více aktivitách. Fragmenty byly přidány do Androidu v Honeycomb verzi Androidu, tedy od API verze 11 [15].

### 2.2.5. Navigace

Navigace odkazuje na interakce, které uživatelům umožňují procházet, do a zpět z různých částí obsahu v aplikaci. Komponenta Navigace systému Android může pomoci implementovat navigaci, od jednoduchých kliknutí na tlačítka až po složitější vzory, jako jsou pruhy aplikací a navigační lišta. Součástí Navigace také zajišťuje konzistentní a předvídatelný uživatelský zážitek dodržováním zavedené sady zásad [16].

Při procházení aplikace sdělujete NavController, že chcete navigovat buď po určité cestě v navigačním grafu, nebo přímo k určitému cíli. NavController pak zobrazí příslušný cíl v NavController [16].

Komponent navigace zpracovává fragmentové transakce, akcí zpět a vpřed, poskytuje standardizované zdroje pro animace a přechody, implementuje a zpracovává přímé odkazy včetně vzorů uživatelského rozhraní navigace, jako jsou navigační šuplík a navigační lišta. Dále podporuje ViewModel a může tedy zobrazit ViewModel na navigačním grafu a sdílet data související s uživatelským rozhraním mezi cíli grafu [16].

### 2.2.6. Layout

Layout definuje strukturu uživatelského rozhraní v aplikaci, například v aktivitě. Všechny prvky v rozložení jsou vytvořeny pomocí hierarchie objektů View a ViewGroup. View obvykle kreslí něco, co uživatel může vidět a komunikovat s tím. Zatímco ViewGroup je neviditelný kontejner, který definuje strukturu rozvržení pro View a další ViewGroup objekty [17].

Základním stavebním kamenem uživatelského rozhraní je objekt View, který je vytvořen z třídy View a zabírá obdélníkovou oblast na obrazovce a je zodpovědný za vykreslení a zpracování událostí. View je základní třída pro widgety, které se používají k vytváření interaktivních komponent uživatelského rozhraní, jako jsou tlačítka a textová pole [18].

ViewGroup je podtřídou View a poskytuje neviditelný kontejner, který obsahuje další Views nebo jiné ViewGroups a definuje jejich vlastnosti rozložení [18].

Na třetí úrovni máme různá rozvržení, která jsou podtřídami třídy ViewGroup a typické rozvržení definuje vizuální strukturu pro uživatelské rozhraní Androidu a může být vytvořeno buď za běhu pomocí objektů View / ViewGroup, nebo lze deklarovat rozvržení pomocí jednoduchého souboru XML main\_layout.xml, který je umístěn ve složce res/layout [18].

Deklarace uživatelského rozhraní v XML umožňuje oddělit prezentaci aplikace od kódu, který řídí její chování. Používání souborů XML také usnadňuje poskytování různých rozvržení pro různé velikosti a orientace obrazovky. Framework pro Android dává flexibilitu k použití jedné nebo obou těchto metod k vytvoření uživatelského rozhraní aplikace. Lze například deklarovat výchozí rozložení aplikace v XML a poté upravit rozložení za běhu [17].

### 2.2.7. ViewModel

ViewModel je třída, která je zodpovědná za přípravu a správu dat pro aktivitu nebo fragment. Zpracovává také komunikaci aktivity nebo fragmentu se zbytkem aplikace [19].

ViewModel je vždy vytvořen ve spojení s fragmentem nebo aktivitou a bude zachován, dokud nebudou ukončeny. Jinými slovy to znamená, že ViewModel nebude zničen,

pokud je jeho vlastník zničen, kvůli změně konfigurace (např. rotace). Nová instance vlastníka se znovu připojí ke stávajícímu modelu [19].

Účelem ViewModel je získat a uchovat informace, které jsou nezbytné pro aktivitu nebo fragment. Ty by měly být schopny sledovat změny ve ViewModelu. Ten tyto informace obvykle zveřejňuje prostřednictvím služby LiveData nebo Android Data Binding. Jedinou odpovědností ViewModelu je správa dat pro uživatelské rozhraní. Neměl by nikdy přistupovat k vaší View hierarchii nebo odkazovat zpět na aktivitu nebo fragment [19].

ViewModels lze také použít jako komunikační vrstvu mezi různými fragmenty aktivity. Každý fragment může získat model ViewModel pomocí stejného klíče prostřednictvím aktivity. To umožňuje komunikaci mezi fragmenty odpojeným způsobem, takže nikdy nemusejí komunikovat s druhým fragmentem přímo [19].

### 2.2.8. Shared Preferences

Shared Preferences jsou způsobem, jakým lze ukládat a načítat malé množství primitivních dat jako páry klíč/hodnota do souboru v úložišti zařízení, jako jsou String, int, float, Boolean, které tvoří předvolby v souboru XML uvnitř aplikace v úložišti zařízení. Shared Preferences lze považovat za slovník nebo pár dat klíč/hodnota. Budeme-li mít například klíč „uživatelské jméno“ a pro tuto hodnotu lze uložit uživatelské jméno uživatele. A pak ho lze získat pomocí jeho klíče (zde uživatelské jméno). Možností je také naprogramování jednoduchého API shared preferences, které je k dispozici pro použití k ukládání předvoleb a jejich stažení podle potřeby. Třída Shared Preferences poskytuje rozhraní API pro čtení, zápis a správu těchto dat [20].

Mají také vhodné využití v různých situacích. Například když je třeba uložit nastavení uživatele nebo uložit data, která lze použít v různých aktivitách v rámci aplikace. Metoda onPause() bude vždy zavolána, než bude aktivita umístěna na pozadí nebo zničena, takže pro trvalé ukládání dat dáváme přednost jejich uložení v onPause(), které lze obnovit v onCreate() aktivitě. Data uložená pomocí Shared Preferences jsou v rámci aplikace uchována jako soukromá. Shared Preferences se však liší od stavu instance dané aktivity [20].

### 2.2.9. Saved Instances

Zachování a obnovení stavu uživatelského rozhraní aktivity včas prostřednictvím aktivity iniciované systémem nebo zničení stavu aplikace je zásadní součástí uživatelské zkušenosti. V těchto případech uživatel očekává, že stav uživatelského rozhraní zůstane stejný, ale systém zničí aktivitu a jakýkoli stav v ní uložený [21].

Chcete-li překlenout propast mezi očekáváním uživatele a chováním systému, použijte kombinaci objektů ViewModel, metody `onSaveInstanceState()` nebo místního úložiště k přetrvávání stavu uživatelského rozhraní napříč takovými přechody instance aplikace a aktivity [21].

Metoda `onSaveInstanceState()` umožňuje přidat páry klíč/hodnota do `outState` aplikace. Metoda `onRestoreInstanceState()` poté umožní načíst hodnotu a nastavit ji zpět na proměnnou, ze které byla původně zaznamenána [22].

Chcete-li připravit aplikaci na jejich použití, je důležité vytvořit proměnné instance pro všechna data, která bude nutné zachovat, když stav aplikace způsobí zničení aktivity. Tyto proměnné jsou mimo jakékoli metody a jsou přístupné odkudkoli ve třídě [22].



### 3. Android Studio

Android Studio je oficiální integrované vývojové prostředí (IDE) pro vývoj aplikací pro Android založené na IntelliJ IDEA. Kromě výkonného editoru kódů a vývojářských nástrojů IntelliJ nabízí Android Studiu ještě více funkcí, které zvyšují produktivitu při vytváření aplikací pro Android. Těmi jsou například flexibilní systém sestavení na bázi Gradle, rychlý a na funkce bohatý emulátor, sjednocené prostředí, kde můžete vyvíjet pro všechna zařízení Android, použít změny k odeslání změn kódu a prostředků do spuštěné aplikace bez restartování aplikace, rozsáhlé testovací nástroje a frameworky a nástroje pro zachycování vláken, které zachycují výkon, použitelnost, kompatibilitu verzí a další problémy [23].

Každý projekt v Android Studio obsahuje jeden nebo více modulů se soubory zdrojového kódu a soubory prostředků. Mezi typy modulů patří: Moduly aplikací pro Android, moduly knihovny, moduly Google App Engine [23].

Všechny soubory sestavení jsou viditelné na nejvyšší úrovni pod skripty Gradle a každý modul aplikace obsahuje následující složky:

- manifesty: Obsahuje soubor `AndroidManifest.xml`.
- java: Obsahuje soubory procedurálního kódu Java, včetně testovacího kódu JUnit.
- res: Obsahuje všechny prostředky bez kódu, například rozložení XML, řetězce uživatelského rozhraní a bitmapové obrázky [23].

Android Studio podporuje anotace pro proměnné, parametry a návratové hodnoty, které pomohou zachytit chyby, jako jsou výjimky nulového ukazatele a konflikty typů prostředků. Správce sady Android SDK balí `Support-Annotations` library v Android Support Repository pro použití s Android Studiem. Android Studio také ověřuje nakonfigurované anotace během kontroly kódu. Když vytváříte a spouštíte svou aplikaci pomocí Android Studia, lze v okně Logcat zobrazit výstup Android Debug Bridge a device log zprávy [23].

#### 3.1. Android emulátor

Emulátor Android simuluje zařízení Android v počítači, takže lze otestovat aplikaci na různých úrovních zařízení a Android API, aniž byste museli mít každé fyzické zařízení. Emulátor poskytuje téměř všechny funkce skutečného zařízení Android. Lze simulovat

příchozí telefonní hovory a textové zprávy, určit umístění zařízení, simulovat různé rychlosti sítě, simulovat rotaci a další hardwarové senzory, přistupovat k obchodu Google Play a mnoho dalšího. Testování aplikace na emulátoru je v některých ohledech rychlejší a jednodušší než na fyzickém zařízení. Můžete například přenášet data rychleji do emulátoru než do zařízení připojeného přes USB [24].

### 3.1.1. Android Virtual Device (AVD)

Virtuální zařízení Android (AVD) je konfigurace, která definuje vlastnosti telefonu, tabletu, systému Wear OS, Android TV nebo Automotive OS zařízení Android, které lze simulovat v emulátoru Android. Správce AVD je rozhraní, které můžete spustit z aplikace Android Studio a které pomáhá vytvářet a spravovat AVD [25].

AVD obsahuje hardwarový profil, obraz systému, oblast úložiště, vzhled a další vlastnosti. Hardwarový profil definuje vlastnosti zařízení dodávaného z výroby. Správce AVD je dodáván s určitými předinstalovanými hardwarovými profily, jako jsou například zařízení Pixel, a podle potřeby můžete definovat nebo přizpůsobit hardwarové profily [25].

Obraz systému označený pomocí Google API zahrnuje přístup ke službám Google Play, včetně karty Google Play v dialogovém okně Rozšířené ovládací prvky, která poskytuje pohodlné tlačítko pro aktualizaci služeb Google Play v zařízení [25].

Aby bylo zajištěno zabezpečení aplikace a konzistentní prostředí s fyzickými zařízeními, jsou systémové obrazy se zahrnutým obchodem Google Play podepsány klíčem k vydání, což znamená, že u těchto obrazů nemůžete získat zvýšená oprávnění (root). Pokud potřebujete zvýšené oprávnění (root), která pomůžou s řešením problémů s aplikací, lze použít obrazy systému Android Open Source Project (AOSP), které neobsahují aplikace ani služby Google [25].

AVD má na vývojovém zařízení vyhrazený úložný prostor. Ukládá uživatelská data zařízení, například nainstalované aplikace a nastavení, stejně jako emulovanou SD kartu. V případě potřeby můžete pomocí aplikace AVD Manager vymazat uživatelská data, takže zařízení má stejná data, jako by bylo nové [25].

## 4. Canon Camera Control API

Rozhraní Camera Control API (CCAPI) je založeno na technologii HTTP a lze přes něj komunikovat na protokolu IP přes Wi-Fi, takže umožňuje vývojářům zvolit si pro obchodní řešení různé operační systémy nebo platformy. S CCAPI může hostitelská aplikace konfigurovat nastavení fotoaparátu, získávat obraz v živém zobrazení, spouštět fotografování a načítat obrázky z fotoaparátu. Ke komunikaci s fotoaparátem, který má povolené CCAPI nejsou vyžadovány žádné další speciální knihovny. Fotoaparát, u kterého je aktivována, funkce CCAPI může reagovat na požadavky z hostitelské aplikace [26].

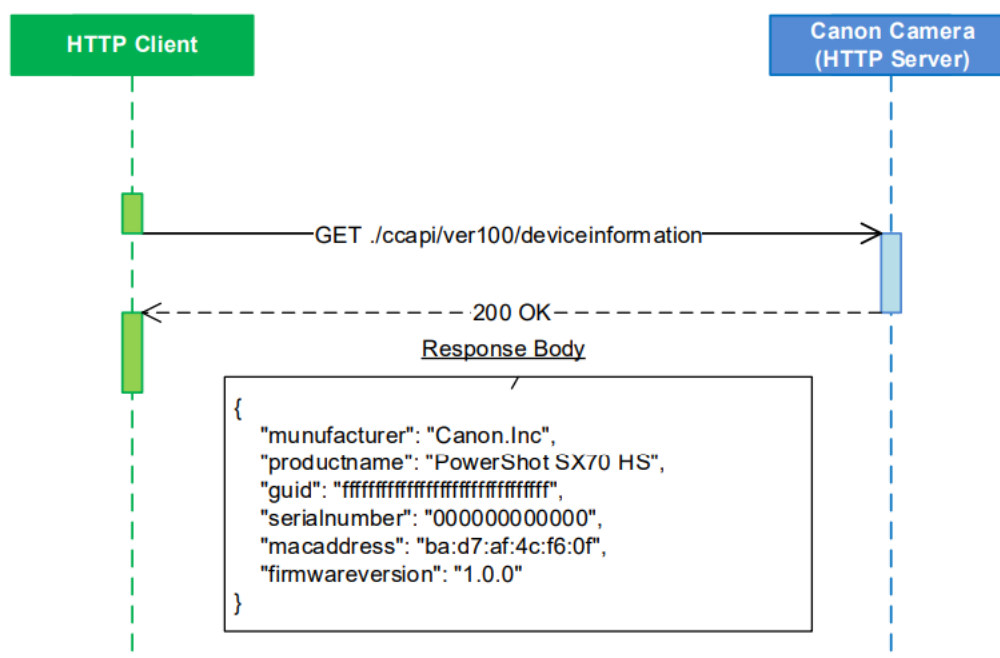
Komunikační protokoly používané k připojení fotoaparátu Canon a protilehlého zařízení jsou protokoly UPnP a HTTP. Tyto protokoly fungují pouze v síti IPv4. Fotoaparáty Canon nelze připojit pomocí protokolu HTTPS. Fotoaparáty Canon lze připojit pomocí ověřování HTTP s nastavením ve fotoaparátu. S fotoaparátem Canon lze připojit pouze jedno protilehlé zařízení. Komunikace není zajištěna, pokud je současně připojeno více zařízení [27].

Specifikace každého API jsou popsány rozdělením do sekcí pro každou z následujících funkcí:

- Získání seznamu API – Získá se seznam rozhraní API podporovaných fotoaparátem Canon.
- Získání informací o fotoaparátu (pevné hodnoty) - Tím se získávají pevné informace o fotoaparátu Canon. Umožňuje získat informace, které nezávisí na stavu fotoaparátu Canon, jako je název produktu a sériové číslo.
- Získání informací o fotoaparátu (proměnné hodnoty) - Získá se informace o úložišti médií připojených k fotoaparátu Canon, stavu baterie, připojeném objektivu a teplotě fotoaparátu.
- Nastavení fotoaparátu – Může získat, uložit nebo smazat informace o autorovi, copyrightu, majiteli, datumu, čase, Wi-Fi připojení a další možné nastavení.
- Operace s fotkami – Umožňuje provádět úkony s fotkami uloženými na kartě ve fotoaparátu.
- Ovládání snímání – Provádí focení, natáčení videa a nastavení automatického zaostřování.

- Nastavení snímání – Získá všechny současné hodnoty a možnosti rozsahu parametrů snímání podporovaných fotoaparátem Canon. Dále se dají tyto hodnoty změnit v aplikaci a tyto hodnoty poslat do fotoaparátu.
- Informace o záznamu – Zobrazí počet fotek, které můžeme ještě pořídit nebo jak dlouho můžeme nahrávat video.
- Živé zobrazení – Umožňuje provádět nastavení živého náhledu z fotoaparátu v aplikaci [27].

Na následujícím obrázku je příklad komunikace mezi aplikací a fotoaparátem pro získání informací o fotoaparátu. Aplikace pošle dotaz na získání informací a fotoaparát odpoví pomocí dat poslaných ve formátu JSON.



Obrázek 1: Komunikace mezi aplikací a fotoaparátem [27]

## 5. Praktická část

V praktické části mé bakalářské práce jsem měl za úkol vytvořit aplikaci pro mobilní zařízení, se kterou se bude moci ovládat fotoaparát značky Canon vzdáleně přes Wi-Fi, na základě API vydané přímo společností Canon. Aplikace je určena pro telefony s operačním systémem Android od verze 5.0. Uživatel by měl být díky aplikaci provádět jednoduchá fotografická nastavení, ale i složitější úkony. Výhodou bude, že bude mít všechna možná nastavení fotoaparátu na jednom místě a nebude se muset složitě navigovat přes různá menu, která jsou ve fotoaparátu. Jako funkcionalita navíc bude možnost si uložit celé nastavení scény jako uživatelské režimy přímo v aplikaci a bude je zde mít pro pozdější využití hned k dispozici.

### 5.1. Návrh uživatelského prostředí

Při návrhu základního menu jsem se chtěl držet jednoduchého designu a toho, aby se uživatel v aplikaci vyznal i při jejím prvním používání. Proto je základní menu rozdělené do 5 položek. První tlačítko slouží pro připojení fotoaparátu k telefonu. Aby se mohli spolu spojit mohou být buď připojeni na Wi-Fi síť, kterou vytvoří přímo fotoaparát. Nebo do libovolné sítě, kam budou připojena obě zařízení.

Druhým tlačítkem v menu je přímo vzdálené snímání z fotoaparátu. Tam je vidět živý náhled obrazu z fotoaparátu, dále se tam pod jednou ze záložek nachází seznam jednotlivých fotografických nastavení, které jsou podporovány API. Na další záložce, co se dají proklikávat na spodní části obrazovky se nacházejí tři tlačítka pro ukládání, načtení nebo smazání uživatelských režimů, pod kterými si může uživatel uložit aktuální nastavení fotoaparátu pro pozdější využití. Pod nimi se nacházejí informace o naklonění fotoaparátu a data týkající se živého náhledu. Na poslední záložce se nachází histogram náhledu.

Pod třetím tlačítkem se nacházejí informace o fotoaparátu. Ty se dělí do několika částí. První jsou samotné informace o fotoaparátu jako je verze firmwaru, model nebo sériové číslo. Ve druhé jsou data o baterii, která se nachází ve fotoaparátu. Jak je nabitá, nebo jestli je kvalitativně v pořádku. Ve dalších dvou částech jsou informace o objektivu a teplotě fotoaparátu. V poslední části jsou data o uložišti fotoaparátu. Kolik je na něm už vyfoceno fotografií, maximální možné místo nebo zbývající volné místo.

Čtvrté tlačítko slouží pro přístup k funkcím fotoaparátu. Těmi jsou jméno autora fotografií, možnost nastavení copyrightu, také možnost nastavení datumu a času ve fotoaparátu. Dále se zde dá i zformátovat SD karta, která je vložena ve fotoaparátu, nebo nastavení Wi-Fi a doba po níž se má fotoaparát automaticky vypnout.

Posledním tlačítkem na hlavní obrazovce aplikace je tlačítko pro možnost zobrazení fotografií, které jsou uloženy na fotoaparátu. Na mobilním zařízení si můžeme zobrazit jejich náhled, ale také s nimi provádět základní úkony, jako jsou otočení, zobrazení si informací o fotografii, její smazání nebo je tu možnost si ji stáhnout do mobilního zařízení a tam s ní dále pracovat.



Obrázek 2: Základní menu aplikace

## 5.2. Princip fungování aplikace

Základní princip fungování aplikace je na volání URL adres, přes které lze ovládat jednotlivé funkce fotoaparátu. Každé nastavení nebo informace, které jsou dostupné přes API mají vlastní URL adresu, nad kterou se dají provádět určité úkony. Přehled dostupných dat, které se mohou na fotoaparátu nastavit lze zjistit, pokud se připojíme

k Wi-Fi vytvořené fotoaparátem a v internetovém prohlížeči zadáme IP adresu, která je zobrazená na fotoaparátu pro připojení k API. Automaticky je tato adresa nastavena na `http://192.168.1.2:8080/ccapi`. Jednotlivé metody, které se dají volat nad adresami jsou GET, POST, PUT a DELETE. Ne nad všemi adresami se dají volat všechny metody pro získávání dat. Pokud máme dostat statické informace o fotoaparátu jako jsou verze firmwaru nebo model zařízení, tak nad touto adresou lze zavolat jenom GET, protože se tyto informace nemění, a proto nepodporuje jinou metodu. Příklad tohoto principu je vidět na obrázku 4. První URL je určeno pro zjištění informací o fotoaparátu a podporuje pouze metodu GET. Druhé URL na obrázku 4 je určeno pro zjištění stavu uložště (jeho velikost, volné místo, počet fotografií), a také podporuje pouze metodu GET.



Obrázek 3: Data o uložšti fotoaparátu [27]

Na obrázku 3 můžeme vidět odpověď fotoaparátu na dotaz metodou GET o data na uložšti fotoaparátu. V této struktuře je pošle fotoaparát do aplikace, kde se následně převedou do listu podle klíče a hodnoty.

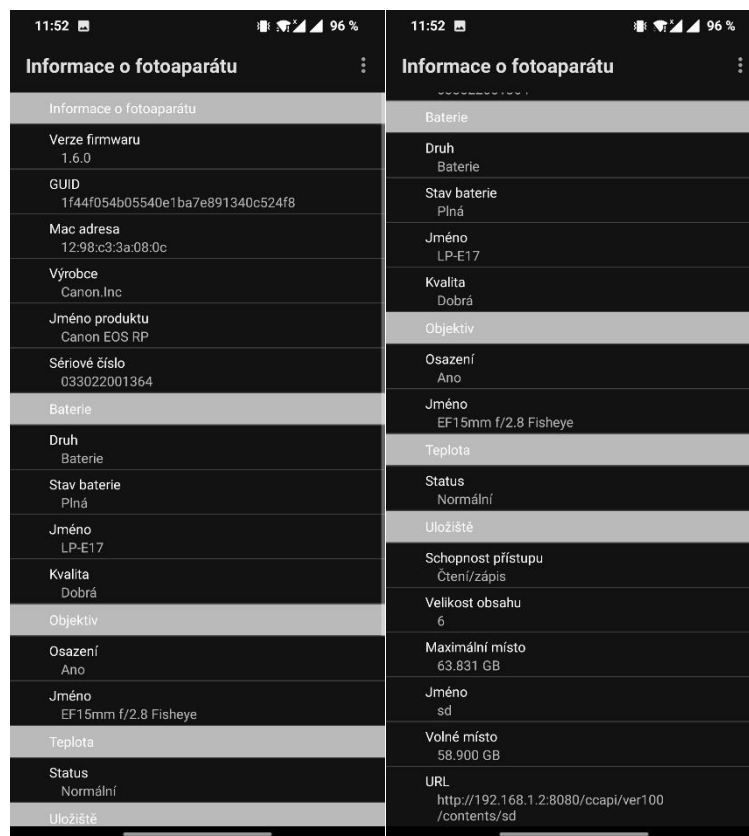
Tímto jednoduchým principem jsou zobrazeny v aplikaci data o informacích o fotoaparátu. Kdy jednotlivé skupiny mají vlastní adresu a data se přenášejí pomocí JSON ve skupinách klíč a hodnota. Na obrázku 5 je vidět snímek obrazovky telefonu a na něm jsou vidět jednotlivá data. Kde baterie je název skupiny a má vlastní URL adresu. Nad ní se zavolá GET a přijdou JSON data, ve kterých je klíč druh a hodnota je baterie

nebo je klíč stav baterie a hodnota je plná. Tyto data mohou být fixní (časem neměnná) a ty se dají najít pod adresou `.../ccapi/ver100/deviceinformation`. Nebo mohou být informace o fotoaparátu proměnná a ty se dají najít pod adresou `.../ccapi/ver100/devicestatus/`.

```
{
  "ver100": [
    {
      "url": "http://192.168.1.2:8080/ccapi/ver100/deviceinformation",
      "get": true,
      "post": false,
      "put": false,
      "delete": false
    },
    {
      "url": "http://192.168.1.2:8080/ccapi/ver100/devicestatus/storage",
      "get": true,
      "post": false,
      "put": false,
      "delete": false
    },
    ...
    ...
    ...
  ]
}
```

Obrázek 4: Ukázka výpisu z adresy `http://192.168.1.2:8080/ccapi` [27]



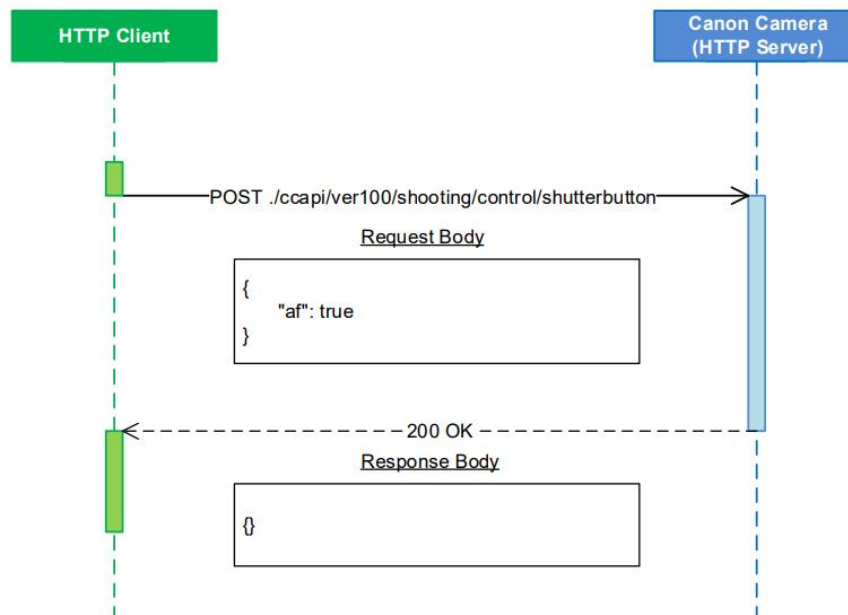


Obrázek 5: Informace o fotoaparátu

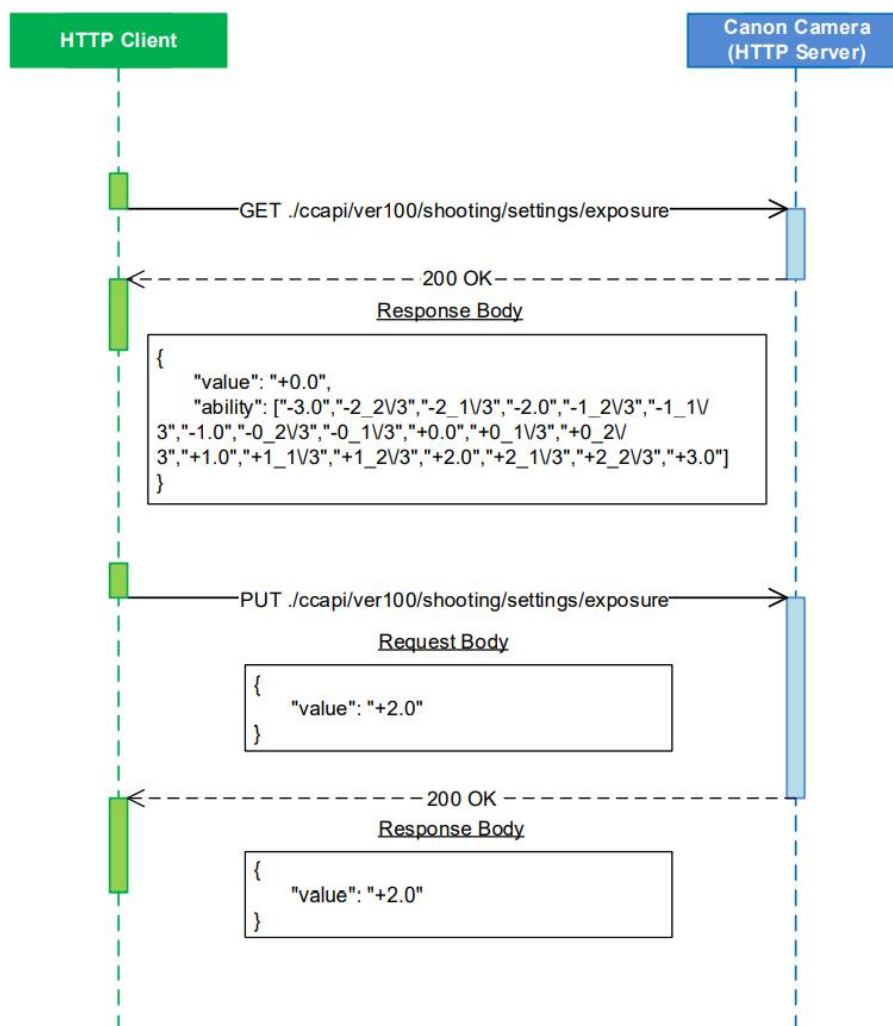
### 5.2.1. Živý náhled

U živého náhledu je hlavní část obrazovka zabraná samotným náhledem z fotoaparátu. Při kliknutí do obrazu můžeme přeastřovat na různá místa podle potřeby. Kliknutím do obrazovky se získá jeho pozice a tato hodnota se pošle do fotoaparátu pomocí URL pro pozici v rámečku automatického ostření. Pod ním se nachází rozložení dalších prvků, které se dají přepínat pomocí tlačítek na spodní straně (na obrázku 9 jsou to tlačítka s popisem ovládání, nastavení, info žn a histogram). V první máme ovládání focení. Kdy je jedno tlačítko pro vyfocení fotky a druhé pro začátek natáčení videa. Dále máme tři tlačítka pro ovládání závěrky fotoaparátu.

Na obrázku 6 je vidět, jak probíhá samotné vyfocení fotky z aplikace. U URL pro tlačítko spouště se zavolá metoda POST a do request body se uvede, pokud je zapnuto automatické ostření.



Obrázek 6: Akce po zmáčknutí tlačítka pro vyfocení fotografie [27]



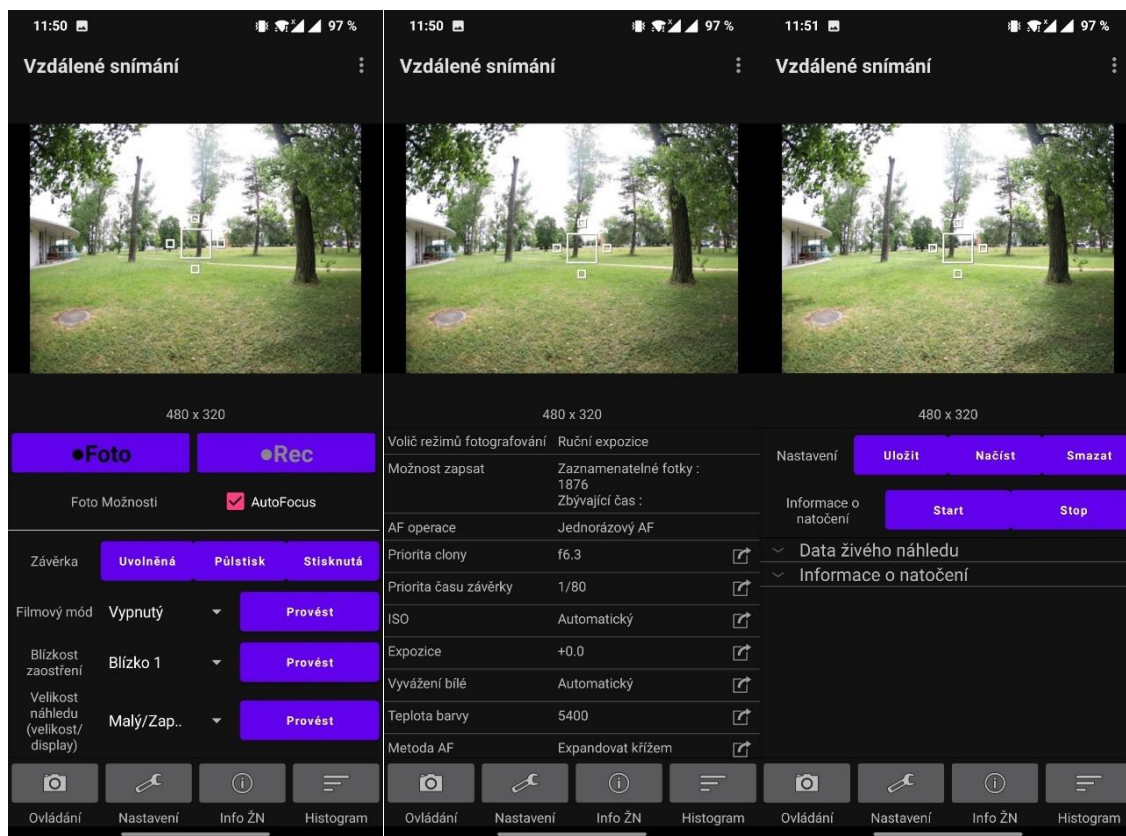
Obrázek 7: Data o expozici [27]

Na obrázku 7 můžeme vidět nejdříve dotaz na GET o datech o expozici. Ten nám vrátí aktuální hodnotu nastavení expozice na fotoaparátu a rozsah hodnot, který se na něm může nastavit. Tento rozsah se může měnit v závislosti na modelu fotoaparátu. Poté je vidět odpověď z aplikace do fotoaparátu. Stačí nad tou stejnou URL adresou zavolat PUT a poslat JSON s hodnotou, kterou na něm chceme změnit.

```
String url = result.getUrl();
String name;
if (url.contains("/shooting/settings/")) {
    String[] split = url.split("/shooting/settings/");
    name = split[split.length - 1];
} else {
    name = result.getRequestName();
}
```

Obrázek 8: Kód pro načtení možných nastavení

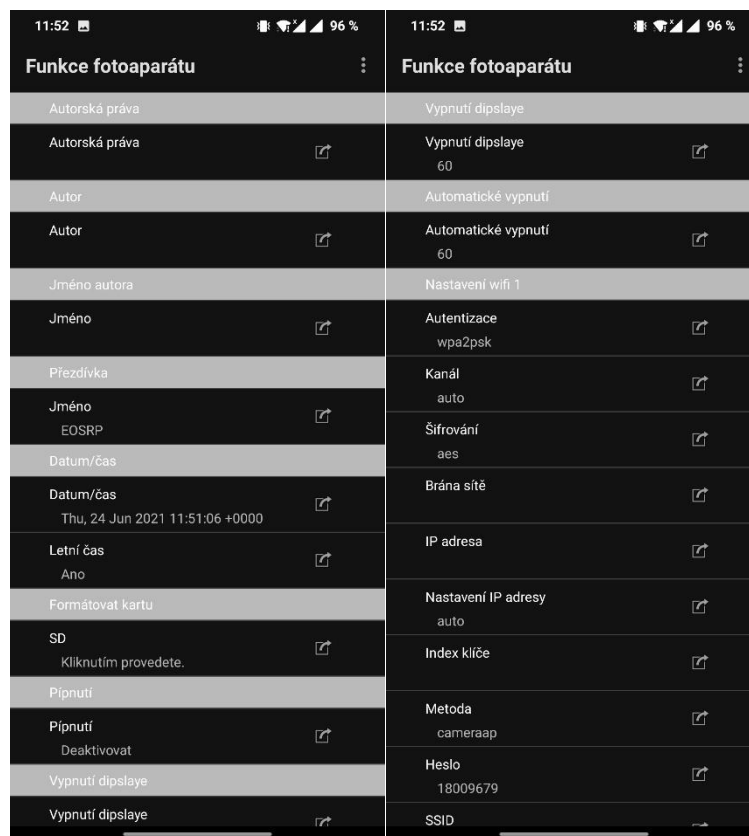
Na další záložce se nachází seznam se všemi možnými nastaveními fotoaparátu. Při nastavení manuálního režimu je možno nastavit 31 jednotlivých nastavení, které se načítají pomocí URL adres, jak je vidět na obrázku 8. Všechna nastavení mají společný základ a dají se najít pod adresou `.../ccapi/ver100/shooting/settings`, proto si jejich jména můžeme načíst do proměnné a k nim na základě JSON odpovědi získat hodnoty. Kromě základních nastavení jako je priorita clony, priorita času uzávěrky nebo ISO, tak se dá nastavit barevné podání nebo jednotlivé barevné styly fotografie. Při kliknutí na jednotlivá nastavení si lze vybrat ze seznamu podporovaných hodnot fotoaparátu, nebo v případě teploty barvy napsat požadované číslo z rozsahu. Toto nastavení se pošle do fotoaparátu ihned po jeho zvolení. Pokud bychom chtěli použít aplikaci pouze pro nastavení jednotlivých hodnot a potom fotit přímo z fotoaparátu, je tu tato možnost. Na obrázku 9 můžeme vidět snímky obrazovek z jednotlivých záložek, mezi kterými se lze přecházet. Levý snímek je pod záložkou ovládání a lze v něm vyfotografovat fotografii nebo přímo ovládat závěrku fotoaparátu. Prostřední snímek je pod záložkou nastavení a v ní se dají změnit jednotlivá nastavení fotoaparátu. Pravý snímek je ze záložky Info ŽN a v ní se dají ukládat, načítat a mazat jednotlivá uživatelská nastavení a data živého náhledu a informace o natočení fotoaparátu.



Obrázek 9: Živý náhled

### 5.2.2. Funkce fotoaparátu

Další částí aplikace jsou funkce fotoaparátu. Znovu probíhá jejich načtení přes jednotlivé URL adresy, ale na rozdíl od informací o fotoaparátu jsou tu data, které můžeme u fotoaparátu měnit. Jako je změna času nebo možnost formátovat kartu. Všechny možné funkce, které jsou dostupné u fotoaparátu nalezneme pod adresou `.../ccapi/ver100/functions`. Když chceme nastavit nějakou hodnotu vyskočí nám dialogové okno, kde může změnit data podle našich preferencí. Pokud to jednotlivé funkce podporují, tak se hodnoty se načítají a mění stejným principem jako u jednotlivých fotografických nastavení. Kdy při metodě GET nám přijde aktuálně nastavená hodnota a rozsah, ze kterého můžeme vybírat jiné hodnoty. Toto je například u funkce vypnutí displeje nebo automatické vypnutí.



Obrázek 10: Funkce fotoaparátu

### 5.2.3. Prohlížení obsahu

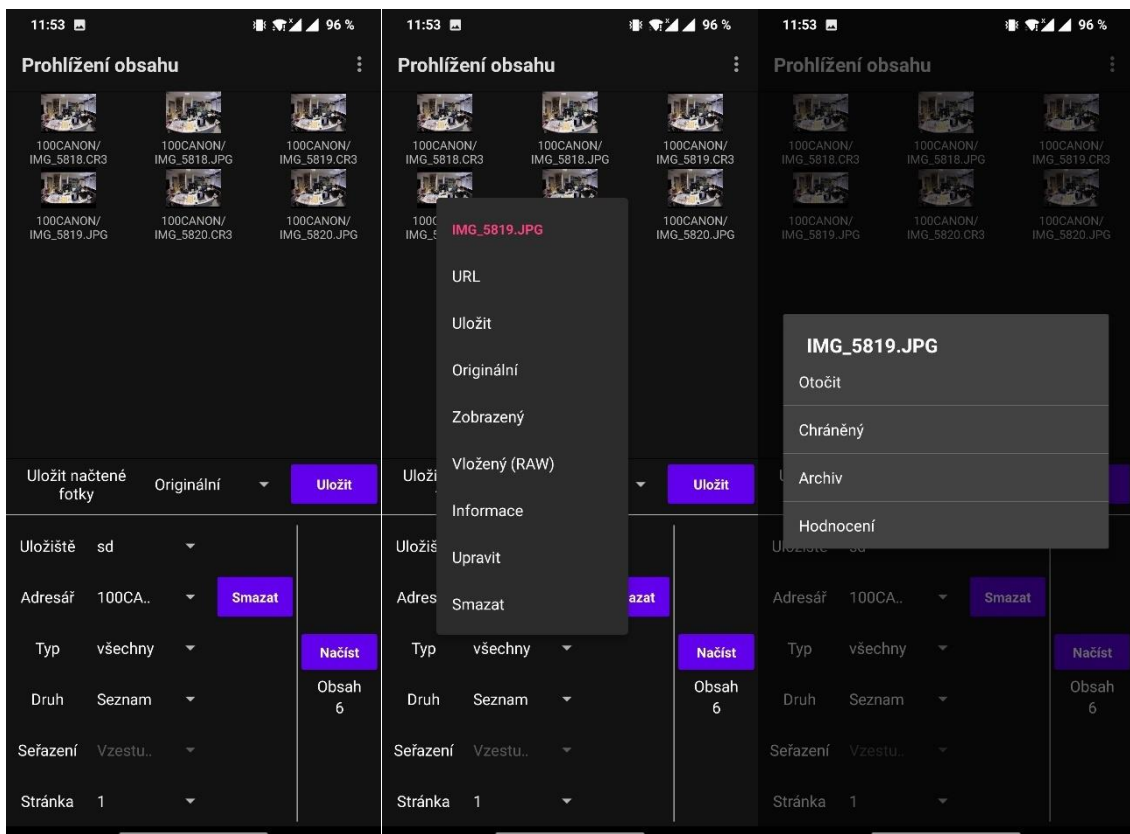
Poslední částí aplikace je prohlížení obsahu. V něm si můžeme prohlížet fotky uložené na kartě fotoaparátu a provádět základní operace, jako jsou otočení fotografie nebo hodnocení. Po kliknutí na fotografii se nám načte ve větší velikosti. Jednotlivé fotografie můžeme také smazat nebo si je uložit do mobilního zařízení. Pokud si je chceme uložit, vytvoří se v galerii složka podle názvu fotoaparátu, aby měl uživatel přehled, pokud má více fotoaparátů, s kterými tuto aplikaci využívá. Fotografie lze uložit v plné nebo zmenšené velikosti. Je také možné uložit všechny načtené fotografie do zařízení najednou.

Jednotlivé fotografie se načítají podle URL adres, které se načítají přes metodu GET a adresu `.../ccapi/ver100/contents/sd`. Na této adrese se nachází jednotlivé adresáře, pod kterými jsou fotografie uloženy a lze k nim přistupovat. Na obrázku 11 můžeme vidět jednotlivé URL adresy fotografií.

```
{
  "url": [
    "http://192.168.1.2:8080/ccapi/ver100/contents/sd/100CANON/IMG_0001.JPG",
    "http://192.168.1.2:8080/ccapi/ver100/contents/sd/100CANON/MVI_0002.MP4",
  ]
}
```

Obrázek 11: URL adresy jednotlivých obrázků [27]

Na obrázku 12 je v levé části vidět stránka s načtenými fotografiemi z fotoaparátu. Můžeme si vybrat také vybrat podle jaké přípony se nám budou fotografie načítat. Lze si například vybrat pouze načtení fotografií s příponou .JPG. Na prostředním snímku je zobrazeno menu, které se zobrazí po dlouhém stisku ikony fotografie. Zde si můžeme fotografii uložit do mobilního zařízení, smazat jí z fotoaparátu nebo zobrazit okno s informacemi o fotografii. Na pravém snímku je menu, které se objeví po stisknutí položky upravit z předchozího menu.



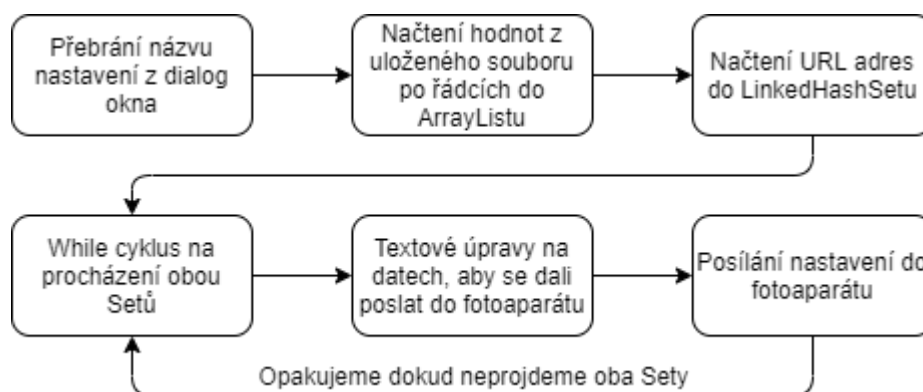
Obrázek 12: Prohlížení obsahu

### 5.3. Ukládání uživatelských nastavení fotoaparátu

Jako součást zadání nad možnostmi API bylo do aplikace udělat možnost ukládání a následné načítání nastavení fotoaparátu, které by chtěl uživatel využít pro pozdější využití. Pro tento důvod jsem se rozhodl naprogramovat tři dialogová okna, jež se budou zobrazovat po kliknutí na příslušná tlačítka v třetí záložce u vzdáleného snímání.

Prvním dialogovým oknem je ukládání nastavení, v kterém jenom uživatel zadá název, pod nímž se má nastavení uložit. Ten se potom uloží do Shared preferences, aby k němu mohl být přístup v ostatních oknech. Poté se restartuje aktivita, aby se mohl uložit seznam jednotlivých nastavení pod stejným názvem jako zvolil uživatel. Tyto nastavení se uloží do souboru se stejným názvem, aby k nim byl potom přístup. Tímto jsou jednotlivá data uložena ve formě klíč hodnota uložena do paměti zařízení.

Při načítání uložených dat se uživateli zobrazí seznam, ve kterém jsou načtena všechna jeho uložená nastavení ze seznamu Shared preferences. Z něj si může vybrat jedno nastavení, a to nechat načíst do fotoaparátu. To je komplikovanější úloha, protože fotoaparát potřebuje přesný formát na to, aby si data mohl načíst. Prvním krokem je načtení dat do ArrayListu. Podle názvu souboru se vždy jeden řádek načte do listu, aby se mohlo s každou hodnotou pracovat samostatně. Další věcí, kterou potřebujeme, pro poslání dat do fotoaparátu jsou URL adresy jednotlivých nastavení. Proto si je postupně načteme do LinkedHashSetu, aby nám sedělo pořadí URL adres a jednotlivých nastavení uložených v souboru. Kvůli tomuto musí být jednotlivá nastavení načítána ve stejném, režimu focení fotoaparátu jako byla uložena. V jiných režimech focení by nám totiž nesouhlasil počet nastavení, protože jsou některá dopočítávaná automaticky.



Obrázek 13: Diagram načtení dat uložených nastavení

Následně pomocí operací se řetězci převedeme uložená nastavení do správných formátů, které můžeme poslat do fotoaparátu. Do parametrů je potřeba přidat metoda, tou je v tomto případě PUT, poté potřebujeme URL adresu a JSON, který má klíč jako „value“ a hodnotou je buď řetězec nebo číslo, podle jednotlivých nastavení. Tyto hodnoty posíláme v cyklu po jedné do fotoaparátu a ten si je převezme a změní postupně jednotlivá nastavení.

```
Bundle args = new Bundle();
String[] params = new String[]{PUT, urls.next(), finalText};
args.putStringArray(Constants.RequestCode.ACT_WEB_API.name(), params);
mWebAPI.enqueueRequest(new WebAPIQueueDataSet(Constants.RequestCode.ACT_WEB_API, args, this));
```

Obrázek 14: Posílání dat do fotoaparátu

Posledním tlačítkem je tlačítko pro smazání nastavení. Po jeho stisku se objeví stejný seznam jako u načtení nastavení. Uživatel si jedno nastavení vybere a tímto výběrem se název nastavení vymaže z uloženého seznamu a smaže se i soubor pod tím názvem s uloženými daty.

## 5.4. Řešené problémy

Pro vyvarování se základním chybám doporučuji si pořádně projít dokumentaci, kterou vytvořila společnost Canon k tomuto API. Naleznete v ní seznam všech podporovaných nastaveních a také jak mají vypadat data, když se posílají do fotoaparátu a v jakém formátu fotoaparát zase posílá zpět. To je nejdůležitější při programování aplikace.

Největší problém, který jsem musel vyřešit byla lokalizace aplikace do češtiny. API jako takové potřebuje všechna data v určitém tvaru a vše co se posílá do fotoaparátu v něm musí být. Samotný fotoaparát také posílá tato data ve stejné podobě. Musel jsem tedy interně tyto data překládat, aby se v aplikaci zobrazovala česky a při výběru nějakého nastavení jsem opět toto nastavení musel zpátky převést zpět do formátu, který fotoaparát potřebuje. A to jsem musel udělat pro každý nápis uvnitř aplikace. Pro některé to bylo jednodušší, protože to byly statické hodnoty, ale například pro jednotlivá fotografická nastavení to bylo těžší. Když uživatel vybere, že chce fotit černobíle, tak musíme tuto hodnotu převést, tak aby jí mohl fotoaparát nastavit. Díky tomuto přístupu jsem ale dokázal aplikaci lokalizovat do českého i anglického jazyka. A pro lokalizaci do dalších jazyků stačí přidat XML soubor, kde bude stejný seznam řetězců a pouze v něm přepsat hodnoty do jakéhokoliv jazyka a aplikace bude stále fungovat stejně.



## 5.5. Testování využitelnosti aplikace

V průběhu vývoje aplikace jsem jí testoval převážně ve vnitřních prostorách, kvůli tomu, že jsem měl fotoaparát pouze zapůjčený a nemohl jsem si ho vzít domů nebo na nějakou akci, pro otestování aplikace ve více prostředí a při různých podmínkách. Až když jsem aplikaci dokončil vypůjčil jsem si fotoaparát alespoň na pár hodin, abych mohl provést nějaké testování ve venkovním prostředí. Zjistil jsem, že fotoaparát reaguje na změny nastavení velmi rychle a pokud byste měli nastavení měnit často, je jednodušší to dělat přes aplikaci, což je výhoda, pokud máte fotoaparát na stativu, odpadá úplně nutnost manipulace s ním a víte, že fotografie budou pořád ze stejné pozice. V tomto vidím největší výhodu aplikace, jelikož máte veškerá nastavení na jednom místě je jejich změna rychlejší než při nutnosti se proklikávat více menu při nastavování přímo ve fotoaparátu.

Další výhodou, kterou jsem zjistil při testování venku bylo zobrazení živého náhledu přímo na telefonu. Při přímém slunečním svitu na display fotoaparátu se těžko určovalo, jestli není přесvícená obloha a všechno dobře vypadá. Na telefon si můžete kouknout ve stínu a lépe se tam určovalo nastavení fotoaparátu, protože výsledná fotografie vypadá stejně jako při živém náhledu na telefonu. Při používání fotoaparátu například ve foto koutku je výhodou aplikace možnost okamžitého stažení fotografie do telefonu a možnost jí hned poslat hostům na sociální síť.

Největší nevýhodou při používání je prodleva mezi stiskem tlačítka pro vyfocení fotky na telefonu a samotným vyfocením fotky na fotoaparátu. Na následujících obrázcích je tato prodleva dobře viditelná. Tlačítko pro vyfocení jsem stiskl, když jsem viděl auto uprostřed objektivu. To samé jsem udělal při focení přímo z fotoaparátu. Je vidět zpoždění o skoro celou délku auta. S tímto zpožděním bude muset fotograf při používání aplikace počítat.



Obrázek 15: Foceno přes aplikaci



Obrázek 16: Foceno přímo na fotoaparátu.

Další nevýhodou je zastavení živého náhledu z fotoaparátu při focení fotografie. Je samozřejmé, že fotoaparát při snímání nemůže posílat data do aplikace, ale tato pauza, při které se živý náhled zmrazí v době pořízení fotografie je skoro čtyři sekundy, což se nehodí, pokud bychom chtěli dělat více fotografií za sebou a potřebovali bychom používat živý náhled.

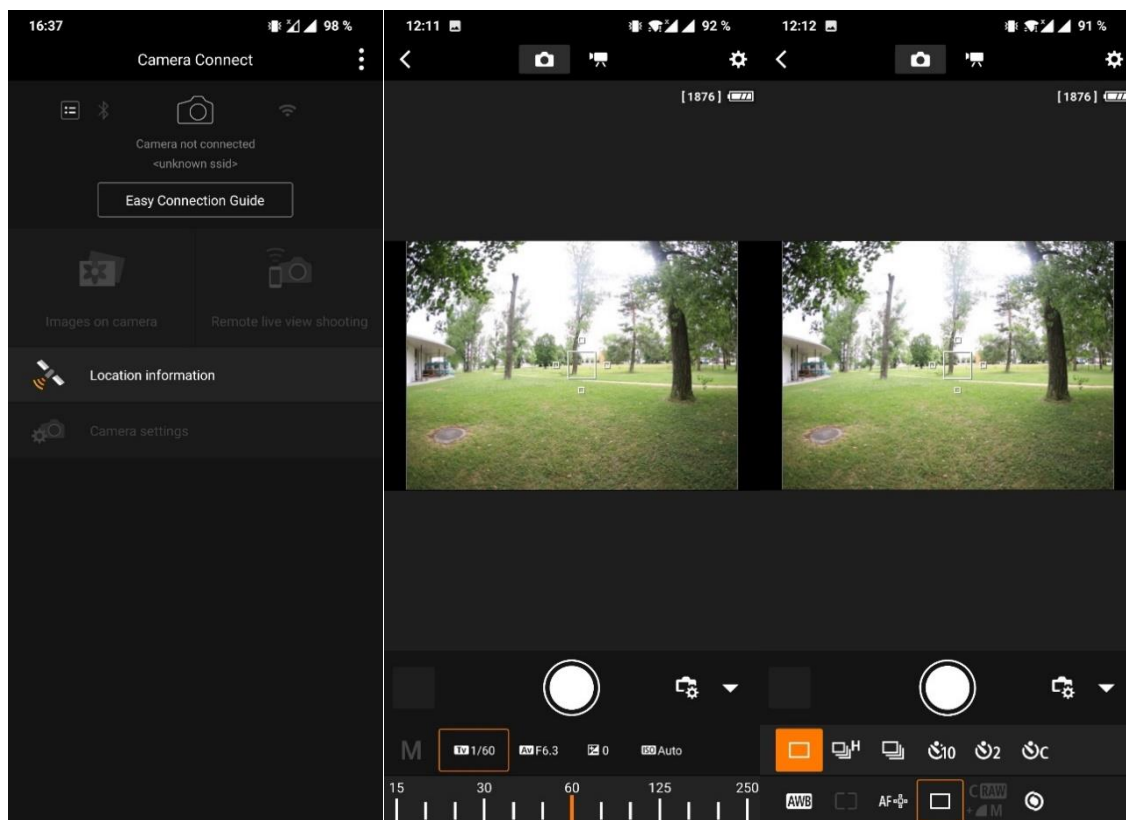
Z výsledků testování mohu říci, že se aplikace nehodí pro fotografy, kteří fotí rychle se pohybující předměty, hlavně kvůli zpoždění při focení fotografie. Ale pro všechny ostatní by měla aplikace přinést zjednodušení nastavování fotoaparátu, hlavně pokud se fotoaparát nachází na stativu, může se manipulace s ním zmenšit na minimum. Někteří také mohou využít možnost náhledu fotografií na větším displeji telefonu než přímo na fotoaparátu.

## 5.6. Porovnání s aplikací od Canonu

Pro svojí aplikaci používá firma Canon speciální komunikaci s fotoaparátem, kterou nevydala pro veřejné využívání. Proto jejich aplikace podporuje více modelů než API, které vydali, to podporuje většinou jenom modely vyšší řady od roku 2019. Aplikace komunikuje jak přes Wi-Fi, tak Bluetooth, proto nemusí mít fotoaparát ani Wi-Fi, aby mohl komunikovat s aplikací. Ta podporuje vzdálené fotografování a přenos obrázků z fotoaparátu do telefonu.

Co se týče možnosti nastavování fotoaparátu přes aplikaci, tak se v ní nachází méně možností nastavení, protože právě musí podporovat více zařízení. Což může být pro některé uživatele výhodou, protože je jednodušší se v nich vyznat, ale pokud by chtěl nastavovat více věcí bude je muset nastavit přes fotoaparát. Jak je vidět na obrázku 17 nastavování priority času uzávěrky se nastavuje pomocí posuvného výběru, na kterém nejsou vidět všechny hodnoty, což může být pro někoho matoucí, pokud se snaží najít určitou hodnotu. Aplikace je také v anglickém jazyce, a to může být pro někoho problém, hlavně při propojování aplikace a fotoaparátu. Jinak je v aplikaci hodně věcí intuitivních nebo mají dobré ikonky, takže i člověk s omezenou znalostí anglického jazyka by se měl v ní zvládnout pohybovat.

Aplikace také trpí na „zamrznutí“ živého náhledu při vyfocení fotografie. Podle mého měření je čas totožný se mnou vytvořenou aplikací. To samé platí i co se týče prodlevy, při stisknutí spouště na mobilním zařízení a vyfocení fotky. Aplikaci jsem měl připojenou přes Wi-Fi, ale nezkoušel jsem jí připojit přes Bluetooth, takže nevím, jaká je tato odezva při spojení aplikace a fotoaparátu touto technologií.



Obrázek 17: Aplikace od Canonu

Na obrázku 17 můžeme vidět prostředí aplikace Canon Connect. V levé části je základní menu aplikace, kde můžeme vidět dvě základní části aplikace, kterými jsou zobrazení snímků na fotoaparátu a živý náhled. Uprostřed vidíme samotný živý náhled z fotoaparátu a jak probíhá nastavování priority času uzávěrky. A v pravé části se nachází snímek s nastavením režimu snímání fotoaparátu.

Aplikace, kterou jsem vytvářel má úplně navíc oproti aplikaci od Canonu možnost uložit si nastavení fotoaparátu do speciálních uživatelských nastavení. Pokud fotograf používá fotoaparát ve více prostředích, může si svá oblíbená nastavení uložit a mít je kdykoliv k dispozici. Používá-li ho například někde v ateliéru může si toto nastavení uložit a když bude potom používat nějaká jiná nastavení při focení venku, stačí si při příchodu znovu do ateliéru jenom dát načíst jeho uložené nastavení a fotoaparát se přenastaví podle jeho uloženého nastavení a bude připraven k focení, podle jeho preferencí.

## 6. Závěr

Má bakalářská práce se zabývá vývojem aplikace pro vzdálené ovládání fotoaparátu značky Canon. Práce je rozdělená na teoretickou a praktickou část.

V teoretické části popisuji operační systém Android, jaké změny se v něm udělali od verze 5.0, která je i nejnižší kompatibilní verze, jakou má aplikace podporuje. Dále představuji nejdůležitější části Androidu, které se využívají při vývoji aplikací. Poté se zabývám vývojovým prostředím, kterým je Android Studio a jeho částmi důležitými při vytváření aplikace. V poslední kapitole teoretické části se zabývám představením samotného API, jenž je určeno ke komunikaci s fotoaparátem.

V praktické části jsem představil aplikaci pro ovládání fotoaparátu. Popsal jsem, z jakých částí se aplikace skládá a jaké jsou jednotlivé její funkce. Dále jsem psal o uživatelských nastavení, která jsem programoval a nejsou součástí API. V další části jsem se zabýval možnostmi využití aplikace v běžném prostředí a jaké z toho využívání pramení výhody a nevýhody. V posledním úseku praktické části jsem ještě mnou vytvořenou aplikaci srovnal s aplikací, kterou vydává přímo společnost Canon.

Myslím si, že aplikace si může najít místo v praktickém využívání fotografie. Má hodně funkcí, které se přes ní nastavují jednodušeji a fotografovi tak může šetřit čas. Záleží teď na společnosti Canon, jestli dál bude rozšiřovat seznam fotoaparátů, jenž toto API podporují. Aplikaci jsem vložil do obchodu Google Play pod názvem CCAPI APP.

Dalším vývojem by mohlo být předělání aplikace, aby jí podporovaly všechny systémy. Předělání by mohlo být do platformy Xamarin, která funguje jako abstraktní vrstva nad .NET. Díky tomu může aplikace běžet jak na OS Android, iOS, tak i přímo na Windows.

## Seznam použité literatury a zdrojů

- [1] Android Operating System. *Investopedia* [online]. New York: Investopedia, 2021 [cit. 2021-02-21]. Dostupné z: <https://www.investopedia.com/terms/a/android-operating-system.asp>
- [2] What is Android? Here's everything you need to know. *Android Authority* [online]. USA: Android Authority, 2021 [cit. 2021-02-21]. Dostupné z: <https://www.androidauthority.com/what-is-android-328076/>
- [3] A Brief History of Android Software Development. *Velvetech* [online]. Chicago: Velvetech, 2021 [cit. 2021-02-21]. Dostupné z: <https://www.velvetech.com/blog/brief-history-android-software-development/>
- [4] The history of Android: The evolution of the biggest mobile OS in the world. *Android Authority* [online]. USA: Android Authority, 2021 [cit. 2021-02-21]. Dostupné z: <https://www.androidauthority.com/history-android-os-name-789433/>
- [5] Android versions: A living history from 1.0 to 12. *Computerworld* [online]. USA: Computerworld, 2021 [cit. 2021-02-21]. Dostupné z: <https://www.computerworld.com/article/3235946/android-versions-a-living-history-from-1-0-to-today.html?page=2>
- [6] Android - 8.0 Oreo. *Android* [online]. USA: Android, 2017 [cit. 2021-02-25]. Dostupné z: <https://www.android.com/versions/oreo-8-0/>
- [7] Android - 9 Pie. *Android* [online]. USA: Android, 2018 [cit. 2021-02-25]. Dostupné z: <https://www.android.com/versions/pie-9-0/>
- [8] Android 10. *Android* [online]. USA: Android, 2019 [cit. 2021-02-25]. Dostupné z: <https://www.android.com/android-10/>
- [9] App Manifest Overview. *Android Developers* [online]. USA: Android Developers, 2021 [cit. 2021-03-02]. Dostupné z: <https://developer.android.com/guide/topics/manifest/manifest-intro>
- [10] AndroidManifest.xml file in android. *JavaTpoint* [online]. USA: JavaTpoint, 2011 [cit. 2021-3-2]. Dostupné z: <https://www.javatpoint.com/AndroidManifest-xml-file-in-android>

- [11] Configure your build. *Android Developers* [online]. USA: Android Developers, 2021 [cit. 2021-03-02]. Dostupné z: <https://developer.android.com/studio/build>
- [12] Android | build.gradle. *GeeksforGeeks* [online]. USA: GeeksforGeeks, 2021 [cit. 2021-3-2]. Dostupné z: <https://www.geeksforgeeks.org/android-build-gradle/>
- [13] Introduction to Activities. *Android Developers* [online]. USA: Android Developers, 2019 [cit. 2021-03-15]. Dostupné z: <https://developer.android.com/guide/components/activities/intro-activities>
- [14] Fragments. *Android Developers* [online]. USA: Android Developers, 2020 [cit. 2021-03-15]. Dostupné z: <https://developer.android.com/guide/fragments>
- [15] Android – Fragments. *Tutorialspoint* [online]. USA: Tutorialspoint, 2017 [cit. 2021-03-15]. Dostupné z: [https://www.tutorialspoint.com/android/android\\_fragments.htm](https://www.tutorialspoint.com/android/android_fragments.htm)
- [16] Navigation. *Android Developers* [online]. USA: Android Developers, 2021 [cit. 2021-03-22]. Dostupné z: <https://developer.android.com/guide/navigation>
- [17] Layouts. *Android Developers* [online]. USA: Android Developers, 2020 [cit. 2021-03-22]. Dostupné z: <https://developer.android.com/guide/topics/ui/declaring-layout>
- [18] Android UI Layouts. *GeeksforGeeks* [online]. USA: GeeksforGeeks, 2021 [cit. 2021-03-22]. Dostupné z: <https://www.geeksforgeeks.org/android-ui-layouts/>
- [19] ViewModel. *Android Developers* [online]. USA: Android Developers, 2021 [cit. 2021-03-26]. Dostupné z: <https://developer.android.com/reference/androidx/lifecycle/ViewModel>
- [20] Shared Preferences in Android with Example. *GeeksforGeeks* [online]. USA: GeeksforGeeks, 2021 [cit. 2021-03-26]. Dostupné z: <https://www.geeksforgeeks.org/shared-preferences-in-android-with-examples/>
- [21] Saving UI States. *Android Developers* [online]. USA: Android Developers, 2021 [cit. 2021-03-26]. Dostupné z: <https://developer.android.com/topic/libraries/architecture/saving-states>
- [22] Save and Restore Instance State Made Easy! *Medium* [online]. USA: Medium, 2018 [cit. 2021-03-26]. Dostupné z: <https://medium.com/@doyouseeitmyway/save-and-restore-instance-state-made-easy-cf6f175f54b0>

- [23] Meet Android Studio. *Android Developers* [online]. USA: Android Developers, 2021 [cit. 2021-04-04]. Dostupné z: <https://developer.android.com/studio/intro>
- [24] Run apps on the Android Emulator. *Android Developers* [online]. USA: Android Developers, 2021 [cit. 2021-04-04]. Dostupné z: <https://developer.android.com/studio/run/emulator>
- [25] Create and manage virtual devices. *Android Developers* [online]. USA: Android Developers, 2021 [cit. 2021-04-04]. Dostupné z: <https://developer.android.com/studio/run/managing-avds>
- [26] Camera Control API. *Canon Developers* [online]. USA: Canon Developers, 2020 [cit. 2021-04-09]. Dostupné z: <https://developers.canon-europe.com/s/camera>
- [27] Canon Inc. Camera Control API Reference. Ver. 1.0.0. Canon, 2019.



## Seznam obrázků

Obrázek 1: Komunikace mezi aplikací a fotoaparátem [27] .....	28
Obrázek 2: Základní menu aplikace .....	30
Obrázek 3: Data o uložišti fotoaparátu [27].....	31
Obrázek 4: Ukázka výpisu z adresy <a href="http://192.168.1.2:8080/ccapi">http://192.168.1.2:8080/ccapi</a> [27].....	32
Obrázek 5: Informace o fotoaparátu .....	33
Obrázek 6: Akce po zmáčknutí tlačítka pro vyfocení fotografie [27] .....	34
Obrázek 7: Data o expozici [27] .....	34
Obrázek 8: Kód pro načtení možných nastavení .....	35
Obrázek 9: Živý náhled.....	36
Obrázek 10: Funkce fotoaparátu.....	37
Obrázek 11: URL adresy jednotlivých obrázků [27].....	38
Obrázek 12: Prohlížení obsahu .....	38
Obrázek 13: Diagram načtení dat uložených nastavení.....	39
Obrázek 14: Posílání dat do fotoaparátu.....	40
Obrázek 15: Foceno přes aplikaci.....	42
Obrázek 16: Foceno přímo na fotoaparátu. ....	42
Obrázek 17: Aplikace od Canonu .....	44

## **Přílohy**

1. DVD – na přiloženém DVD se nachází plné znění bakalářské práce pod názvem bp\_minar.pdf a zdrojové kódy k aplikaci.