



Zadání diplomové práce

| | |
|-----------------------------|---|
| Název: | Vývoj aplikace na chytré televize Samsung pro neziskovou TV |
| Student: | Bc. Matúš Viščor |
| Vedoucí: | Ing. Tomáš Vondra, Ph.D. |
| Studijní program: | Informatika |
| Obor / specializace: | Softwarové inženýrství |
| Katedra: | Katedra softwarového inženýrství |
| Platnost zadání: | do konce letního semestru 2023/2024 |

Pokyny pro vypracování

CATVUSA je nezisková TV propagující ČR v USA. Nabízí tři internetové televizní kanály (cestopisný, jazykovědný a kuchařský) a dvě radiové stanice.

Ve spolupráci s producentem a technikem televizní stanice sepište požadavky na vzhled a funkčnost aplikace.

Požadavky analyzujte a navrhněte jejich řešení pro chytré televize.

Aplikace by měla splňovat doporučení pro tvorbu grafického rozhraní a použít API na hlasové ovládání. Bude se napojovat na Wordpressové API webu CATVUSA. Aplikaci implementujte.

Napište jednotkové testy a integrační testy, které půjdou periodicky automatizovaně spouštět a odhalí i změny webu, které by rozbily aplikaci.

Funkčnost zahrne úvodní stránku s upoutávkami na aktuální pořady synchronizovanou s webem. Oznámí uživateli existenci nového obsahu. Hlavní funkcí bude streaming videa ze zmíněných televizních kanálů a rádio se zvýhodněním podporovatelů nadace. Veškerý obsah je již připraven v databázi webu.



ČVUT

ČESKÉ VYSOKÉ
UČENÍ TECHNICKÉ
V PRAZE

F8

**Fakulta informačních technologií
Katedra softvérového inženýrstva**

Diplomová práce

VÝVOJ APLIKÁCIE NA INTELIGENTNÉ TELEVÍZIE SAMSUNG PRE NEZISKOVÚ TV

Bc. Matúš Višcor

Máj 2023

Vedúci práce: Ing. Tomáš Vondra, Ph.D.

Podakovanie / Prehlásenie

Chcel by som sa poďakovať vedúcemu Ing. Tomášovi Vondrovi, Ph.D. za vedenie tejto práce a cenné rady poskytnuté počas tvorby diplomovej práce. Ďalej by som chcel poďakovať zadávateľovi Johnovi Honnerovi za možnosť pripojiť sa k tomuto zaujímavému projektu. Veľká vďaka patrí Pánu Bohu, že ma dostal až sem. V neposlednom rade ďakujem mojim rodičom, priateľke a všetkým blízkym za podporu počas môjho štúdia.

Prehlasujem, že som predloženú prácu vypracoval samostatne a že som uvedený všetky použité informačné zdroje v súlade s Metodickým pokynom o dodržiavaní etických princípov pri príprave vysokoškolských záverečných prác. Beriem na vedomie, že na moju prácu sa vzťahujú práva a povinnosti vyplývajúce zo zákona č.121/2000 Z.z., autorského zákona, v znení neskorších predpisov. V súlade s ust. §2373 odst.2 zákona č.89/2012 Z.z., občiansky zákonník, v znení neskorších predpisov, týmto udelením nevýhradného oprávnenia (licencie) k používaniu mojej práce, a to vrátane všetkých počítačových programov, ktoré sú jej súčasťou alebo prílohou a všetky ich dokumentácie (ďalej iba "dielo"), a to všetkým osobám, ktoré si prajú dielo používať. Tieto osoby majú právo používať dielo akýmkoľvek spôsobom, ktorý neznižuje hodnotu diela a za akýmkoľvek účelom (vrátane použitia pre účely zisku). Toto oprávnenie je časovo, teritoriálne aj množstevne neobmedzené.

V Prahe 4. mája 2023

.....

.....

Abstrakt / Abstract

Táto diplomová práca sa zaoberá návrhom a implementáciou aplikácie pre smart televízory značky Samsung. Hlavným cieľom práce je navrhnuť a vytvoriť aplikáciu, ktorá bude užívateľsky prívetivá a zároveň bude schopná ponúknuť dostatočný počet funkcií a možností pre užívateľa. V práci sa podrobne opisujú technické aspekty ako napríklad vývoj vo vývojovom prostredí pre operačný systém Tizen. Využívajú sa aj webové technológie a knižnica CAPH. Aplikácia sa vytvára pre projekt CATVUSA. CATVUSA je nezisková televízia propagujúca Českú republiku v USA. Ponúka tri internetové televízne kanály (cestovateľský, jazykovedný a kuchársky) a dve rozhlasové stanice.

Kľúčové slová: Smart TV, Samsung, Tizen, Wordpress, CAPH, CATVUSA

This thesis deals with the design and implementation of an application for Samsung smart TVs. The main goal of the work is to design and create an application that is user-friendly and at the same time capable of offering sufficient functions and options for the user. The technical aspects of development environment for the Tizen operating system are described in detail. Web technologies and the CAPH library are also used. The application is created for the CATVUSA project. CATVUSA is a non-profit television promoting the Czech Republic in the USA. It offers three internet TV channels (travel, language, and cooking) and two radio stations.

Keywords: Smart TV, Samsung, Tizen, Wordpress, CAPH, CATVUSA

Title translation: Samsung smart tv application development for non-profit tv.

Obsah /

| | | | |
|--|-----------|--|--|
| 1 Úvod | 1 | | |
| 2 Cieľ práce | 2 | | |
| 2.1 Zadavateľ | 2 | | |
| 2.2 Požiadavky od zákazníka | 3 | | |
| 2.2.1 Prehrávanie videí | 3 | | |
| 2.2.2 Prehrávanie rádia | 3 | | |
| 2.2.3 Členstvo | 4 | | |
| 2.2.4 Hlasové ovládanie | 4 | | |
| 2.2.5 Funkčné požiadavky | 5 | | |
| 2.2.6 Nefunkčné požiadavky | 5 | | |
| 3 Princípy dizajnu pre Smart TV | 6 | | |
| 3.1 Základné princípy dizajnu | 6 | | |
| 3.1.1 Jednoduchosť | 7 | | |
| 3.1.2 Jednoznačnosť | 7 | | |
| 3.1.3 Konzistentnosť | 7 | | |
| 3.1.4 Spätná väzba | 7 | | |
| 3.2 Uživatelské prostredie | 9 | | |
| 3.2.1 Priemerná vzdialenosť medzi televízorom a užívateľom | 9 | | |
| 3.2.2 Televíziu používa viac osôb | 9 | | |
| 3.2.3 Prechod medzi obrazovkami | 9 | | |
| 3.2.4 Ovládacie prvky televízora | 10 | | |
| 3.3 Kontrolný zoznam od Samsungu | 11 | | |
| 3.3.1 Všeobecné | 11 | | |
| 3.3.2 Vstup | 11 | | |
| 3.3.3 Prehrávač médií | 11 | | |
| 3.3.4 Spätná väzba | 12 | | |
| 3.3.5 Všeobecne odporúčania | 12 | | |
| 4 Analýza a návrh užívateľského rozhrania | 13 | | |
| 4.1 Aplikácia pre Android TV | 13 | | |
| 4.1.1 Hlavná obrazovka | 13 | | |
| 4.1.2 Obrazovka detailu videa | 14 | | |
| 4.1.3 Obrazovka prihlásenia a registrácie | 15 | | |
| 4.1.4 Obrazovka videa a rádia | 16 | | |
| 4.1.5 Obrazovka vyhľadávania | 16 | | |
| 4.2 Návrh zmien | 17 | | |
| 4.2.1 Hlavná obrazovka | 17 | | |
| 4.2.2 Obrazovka detailu videa | 18 | | |
| 4.2.3 Obrazovka prehrávača videa | 19 | | |
| 4.2.4 Obrazovka prehrávača rádia | 20 | | |
| 4.2.5 Obrazovka vyhľadávania | 21 | | |
| 4.2.6 Obrazovka prihlásenia a registrácie | 21 | | |
| 4.2.7 Úvodná obrazovka | 22 | | |
| 5 Použité technológie | 24 | | |
| 5.1 Hypertext Markup Language 5 (HTML5) | 24 | | |
| 5.2 Cascading Style Sheets (CSS) | 25 | | |
| 5.2.1 JavaScript | 25 | | |
| 5.3 Tizen | 25 | | |
| 5.3.1 Typy aplikácií | 26 | | |
| 5.4 Tizen studio | 27 | | |
| 5.4.1 TV Simulator | 28 | | |
| 5.4.2 TV Emulator | 28 | | |
| 5.4.3 Webový inšpektor | 29 | | |
| 5.5 CAPH | 29 | | |
| 5.5.1 Základné podporované komponenty | 30 | | |
| 5.6 Wordpress REST API | 30 | | |
| 5.6.1 WordPress | 30 | | |
| 5.6.2 API | 31 | | |
| 5.6.3 REST | 31 | | |
| 5.6.4 JSON | 31 | | |
| 5.6.5 REST API | 32 | | |
| 6 Implementácia | 33 | | |
| 6.1 Komunikácia so serverom | 33 | | |
| 6.1.1 Koncové body | 33 | | |
| 6.1.2 Globálne parametre | 34 | | |
| 6.1.3 Stránkovanie | 35 | | |
| 6.1.4 Realizácia požiadavkov | 35 | | |
| 6.2 Navigácia | 36 | | |
| 6.2.1 Rolovateľné zoznamy | 38 | | |
| 6.3 Realizácia jednotlivých častí | 40 | | |
| 6.3.1 Práca s údajmi | 40 | | |
| 6.3.2 Hlavná obrazovka | 41 | | |
| 6.3.3 Obrazovka detailu videa | 42 | | |
| 6.3.4 Vyhľadávania | 43 | | |
| 6.3.5 Prehrávač videa | 43 | | |
| 6.3.6 Prehrávač rádia | 45 | | |
| 6.3.7 Úvodná obrazovka | 46 | | |
| 6.3.8 Hlasové ovládanie | 46 | | |
| 6.4 S2Member Pro API | 47 | | |
| 6.4.1 Obrazovka prihlásovania | 48 | | |

| | |
|---|-----------|
| 7 Testovanie | 49 |
| 7.1 Jednotkové testovanie | 49 |
| 7.1.1 Typy jednotkového tes- tovania | 49 |
| 7.1.2 Realizácia jednotko- vých testov | 50 |
| 7.2 Integračné testovanie | 51 |
| 7.2.1 Realizácia integrač- ných testov | 51 |
| 7.3 Užívateľské testovanie | 52 |
| 8 Záver | 53 |
| Literatúra | 54 |
| A Skratky | 59 |

/ Obrázky

| | | |
|-------------|--|----|
| 2.1 | Use case diagram. | 4 |
| 3.1 | Dobrý príklad zamerania | 8 |
| 3.2 | Zlý príklad zamerania..... | 8 |
| 3.3 | Príklad plochej navigácie | 10 |
| 3.4 | Zmätok spôsobený diagonálnym umiestnením | 11 |
| 4.1 | Hlavná obrazovka pre Android TV | 14 |
| 4.2 | Obrazovka detailu videa pre Android TV | 15 |
| 4.3 | Obrazovka detailu videa pre Android TV | 15 |
| 4.4 | Obrazovka videa a rádia pre Android TV | 16 |
| 4.5 | Obrazovka vyhľadávania pre Android TV | 17 |
| 4.6 | Návrh užívateľského rozhrania hlavnej obrazovky..... | 18 |
| 4.7 | Návrh užívateľského rozhrania obrazovky detailu videa ... | 19 |
| 4.8 | Návrh užívateľského rozhrania obrazovky prehrávača videa pri pozastavenom videu ... | 20 |
| 4.9 | Návrh užívateľského rozhrania obrazovky prehrávača rádia pri zobrazovaní dostatočných informácií | 21 |
| 4.10 | Návrh užívateľského rozhrania obrazovky prihlásovania a registrácie | 22 |
| 4.11 | Sekvečný diagram zobrazujúci proces vykreslenia úvodnej obrazovky | 23 |
| 5.1 | Model architektúry Tizen..... | 27 |
| 6.1 | Hlavná obrazovka smart TV Samsung aplikácie..... | 42 |
| 6.2 | Hlavná obrazovka smart TV Samsung aplikácie so zameraním na prvok | 42 |
| 6.3 | Obrazovka detailu videa smart TV Samsung aplikácie .. | 43 |
| 6.4 | Obrazovka pozastaveného videa | 44 |
| 6.5 | Obrazovka rádia smart TV Samsung aplikácie..... | 46 |

| | | |
|------------|--------------------------|----|
| 6.6 | Obrazovka rádia smart TV | |
| | Samsung aplikácie..... | 48 |

Kapitola 1

Úvod

V Spojených štátoch amerických žije teraz viac ako milión Američanov českého pôvodu a práve pre túto národnostnú menšinu vysiela prostredníctvom káblovej televízie a internetu svoje pravidelné vzdelávacie programy Czech-American TV (CATVUSA). Úlohou tejto neziskovej organizácie je predovšetkým vzdelávať a informovať o českej kultúre, tradíciách, folklóre alebo miestach v Čechách, na Morave a v Sliezske. Pri jej zrode stál brnenský rodák John Honner. CATVUSA vysiela výhradne v angličtine a svojich divákov si hľadá cez internet po celom svete. Mimo USA je najväčšia sledovanosť v Kanade, Austrálii, Anglicku a tiež v Ruskej federácii. Vysielanie je v angličtine, pretože televízia chce prevažne zaujať Čechoameričanov, občanov USA, ktorí sa hlásia k českému pôvodu a už nevedia dobre po česky, a ďalej pre všetkých Američanov, ktorí sa chcú niečo viac dozvedieť o českej kultúre. Podľa slov zakladateľa je cieľom krajanových aktivít je osveta a šírenie dobrého mena Českej republiky v zahraničí so zameraním na krajanov, rodiny českých prisťahovalcov, ktorí svoje deti vychovávajú v USA a všetkých záujemcov o Českú Republiku žijúcich v zahraničí, a to nielen v USA ale aj v ďalších krajinách sveta . V súčasnosti je tento obľúbený program vysielaný na káblovej televízii v šesťdesiatich rôznych amerických mestách a jeho dosah je dva a pol milióna domácností.[1]

Éra televízorov, ktoré väčšinou užívateľ spustil a sledoval programy, ktoré mu boli naservírované jednotlivými televíziami vo vysielacom čase, je dávno preč. Nastúpila éra Smart TV, ktorá ponúka viac, než prepínanie medzi kanálmi. Užívateľia teraz môžu venovať svoj čas zábave podľa ich výberu. Prívetivé užívateľské rozhranie pomáha nájsť obľúbený seriál, rozprávku, film alebo program, ktorý si môžu dokonca nahráť a pustiť neskôr alebo sa vrátiť na jeho začiatok. Podobne aj obsah produkovaný CATVUSA je vhodné ponúknuť divákovi nie len cez internet prostredníctvom obrazovky monitoru, ale aj pomocou inteligentných televízií. Medzi najväčších poskytovateľov Smart TV patrí Samsung. Práve televízory tejto spoločnosti sú v USA veľmi populárne. Organizácia CATVUSA by tak s aplikáciou pre chytré TV mohla ešte viac robiť dobré meno Českej republike a to nie len v USA ale aj v iných krajinách.

Táto diplomová práca sa bude zaoberať analýzou požiadavkou od organizácie CATVUSA, návrhom, implementáciou pomocou vývojového prostredia Tizen Studio pre vývoj natívnych a webových aplikácií Tizen. Práca bude brať do úvahy aj základné dizajnové princípy, obmedzenia a odporúčania spoločnosťou Samsung. Nakoniec sa aplikácia otestuje a publikuje do online obchodu.

V druhej kapitole sa predstavuje zadávateľ a jeho požiadavky na prácu, zatiaľ čo v tretej kapitole sú rozobrané princípy dizajnu pre Smart TV a odporúčania a požiadavky výrobcu Samsung. V štvrtej kapitole je analyzované riešenie aplikácie pre Android TV a navrhnuté upravené užívateľské rozhranie. V piatej kapitole sú predstavené použité technológie, ktoré sú založené na webových technológiách a sú kompatibilné s operačným systémom Tizen. Aplikácia bude komunikovať pomocou WordPress REST API. V predposlednej kapitole sa podrobne popisuje programová realizácia aplikácie a v poslednej kapitole sú informácie o jednotkovom a integračnom testovaní.

Kapitola 2

Ciel' práce

Obsahom tejto diplomovej práce je analyzovať požiadavky na televíznu aplikáciu a navrhnúť ich riešenie pre zariadenia od spoločnosti Samsung. Aplikácia by mala byť vzhľadovo aj obsahovo podobná aplikácii CATVUSA pre platformu Android, na ktorej pracoval Viktor Sinelnikov. [2] Výstupom práce potom bude hotová aplikácia dostupná v obchode pre TV aplikácie od výrobcu Samsung. Podoba aplikácie by mala byť založená na vzhľade aplikácie pre Android, ale z hľadiska užívateľskej prívetivosti to nie je podmienkou. Podľa slov zakladateľa spoločnosti CATVUSA Johna Honnera, je možné upraviť vzhľad tak, aby aplikácia bola viac praktická a uľahčila tak užívateľom prácu s ňou. Zakladateľ tiež spomína, že CATVUSA by chcela do určitej miery vzhľad jednotlivých aplikácií na rôznych platformách zachovať. Aplikácia bude vytvorená podľa štandardu návrhových pravidiel a obmedzení pre Samsung smart TV dostupných z oficiálnych stránok. [3] [4]

Na rozdiel od iných aplikácií pre portál CATVUSA od študentov ČVUT, obsah tejto aplikácie bude priamo riadený z administrátorskej plochy systému Wordpress pre správu obsahu rovanko ako u aplikácie pre Android TV.

2.1 Zadavateľ

Czech-American TV (CATVUSA) je Česko-americká televízia, ktorá produkuje pravidelne vysielanie od roku 2003. Je prvým a jediným programom svojho druhu v Spojených štátoch amerických, ktorý cieľi svoj obsah nielen na česko-americkú komunitu, ale aj na každého, kto sa zaujíma o české regióny, históriu, tradície, módu, kuchyňa, hudba, prírodné a architektonické krásy, obchod alebo iné produkty Českej republiky. [5]

CATVUSA ponúka okrem programov vysielaných na káblovej televízii aj svoje webové stránky, ktoré sú prístupné na adrese catvusa.com. Na týchto stránkach sa nachádza archív videozáznamov programov káblovej televízie, recepty českej kuchyne, lekcie češtiny, informácie o význame českých priezvisk alebo o českých tradíciách. Tiež je možné si vypočuť dve internetové rádiá s klasickou a ľudovou hudbou a podcast o českej kultúre.

Czech-American TV funguje ako nezisková charitatívna organizácia, ktorá vysielá vzdelávacie programy pre každého, kto sa chce dozvedieť viac o českom dedičstve. Televízia vysielala online po USA a do celého sveta cez internet. Jej obsah je týždenne aktualizovaný. Program na káblovej televízii je známy ako John Honner's CATV Show a vysielala sa každú stredu v 60 mestách USA. Medzi kľúčové programy patria Objavuj české kraje, České pamätihodnosti, České destinácie, České varení, České tradície, Poznaj české zeme, Česká hodina, Reportáž z Česka, Slávni Česi a iné. Všetky programy sú len v angličtine. Česko-americkú televíznu show založil John Honner. [5]

Na vývoji a správe týchto webových stránok už viac ako desať rokov pracujú študenti českých vysokých škôl vrátane ČVUT. Bolo vyvinutých niekoľko aplikácií pre Android a iOS, pluginy a šablóny Wordpressu použité na portáli.

2.2 Požiadavky od zákazníka

Veľmi dôležitým procesom pri vývoji akejkoľvek aplikácie alebo systému je špecifikácia požiadaviek na produkt. Je kľúčové zoznámiť sa s prostredím spoločnosti, analyzovať ako funguje, pozrieť si ich už používané aplikácie napr. na iných platformách, vypočuť si pozitívne skúsenosti s používaním ale aj negatívne. Na konci tohoto procesu by mal mať vývojár jasno, aký produkt zákazník požaduje.

V spolupráci so zakladateľom spoločnosti, Johnom Honnerom, boli spísané nasledujúce požiadavky na televíznu aplikáciu.

2.2.1 Prehrávanie videí

Spoločnosť CATVUSA ponúka zaujímavý a edukačný obsah nie len o varení a regiónoch v Českej republike. Prostredníctvom videí, by aplikácia mohla byť pre užívateľov zaujímavým a vzdelávacím nástrojom. Vďaka nim sa môžu užívatelia dozvedieť o miestnych tradičných receptoch, kultúre a histórii, čím si rozšíria svoje vedomosti a zároveň získajú nové zážitky.

Základom aplikácie je teda možnosť prehrávania videí dostupných na webovej stránke. Keďže televízia produkuje vysielanie rozdelené podľa rôznych kanálov (cestopisný, jazykovedný a kuchársky), tak je požadované, aby hlavná stránka aplikácie tiež zobrazovala videá oddelene od seba podľa kategórií. Podľa týchto kategórií bude užívateľovi umožnené filtrovať videa a ľahšie sa tak dostať k cieľu, ktorý hľadá.

Niektoré z videí musia byť obsiahnuté nie len vo svojej kategórii, ale aj za účelom propagácie nových videí v špeciálnej kategórii. Táto kategória by mala byť na začiatku všetkých kategórií, aby ju užívateľ spozoroval čo najskôr. Pred každým videom musí byť prehrané špeciálne krátke video (intro) a po každom videu musí byť prehrané ďalšie krátke video (outro). Intro a outro sú nastavené pre každé video zvlášť správcom systému pri pridaní nového videa.

Každé video bude mať stránku s detailom, na ktorej bude užívateľ môcť video spustiť. Ďalej tu tiež uvidí sprievodný textový popis, ktorý mu napovie, čo vo videu môže očakávať.

2.2.2 Prehrávanie rádia

Pre obyvateľov USA s českými koreňmi môže byť učenie o českej ľudovej a klasickej hudbe dôležitým spôsobom, ako sa priblížiť ku kultúre ich predkov. Česká hudba má svoje miesto v histórii svetovej hudby a je považovaná za veľmi kvalitnú. Učenie sa o tejto hudbe môže poskytnúť estetickú hodnotu a umožniť lepšie porozumenie a ocenenie tohto umenia.

Ďalšou funkcionalitou aplikácie je počúvanie rádia so zvýhodnením podporovateľov nadácie. V databáze webového portálu sa nachádza bohatý obsah českých hudobných klasík a tiež známa ľudová hudba. Aplikácia bude obsahovať playlist týchto nahrávok, čo by malo pripomínať funkciu rádia. Podľa požiadavok zadávateľa bude užívateľovi obmedzená možnosť pretočiť nahrávku v oboch smeroch, môže to pripomínať funkciu rádia, ktoré neumožňuje preskočenie alebo vrátenie sa späť na predošlú skladbu.

Toto obmedzenie môže byť účelné pre rôzne dôvody. Napríklad, môže to zabezpečiť, aby užívateľ skutočne venoval pozornosť obsahu a nemohol nahrávku jednoducho pretočiť alebo preskočiť časti, ktoré mu neprekážajú. Obmedzenie by malo viac priblížiť efekt rádia.

2.2.3 Členstvo

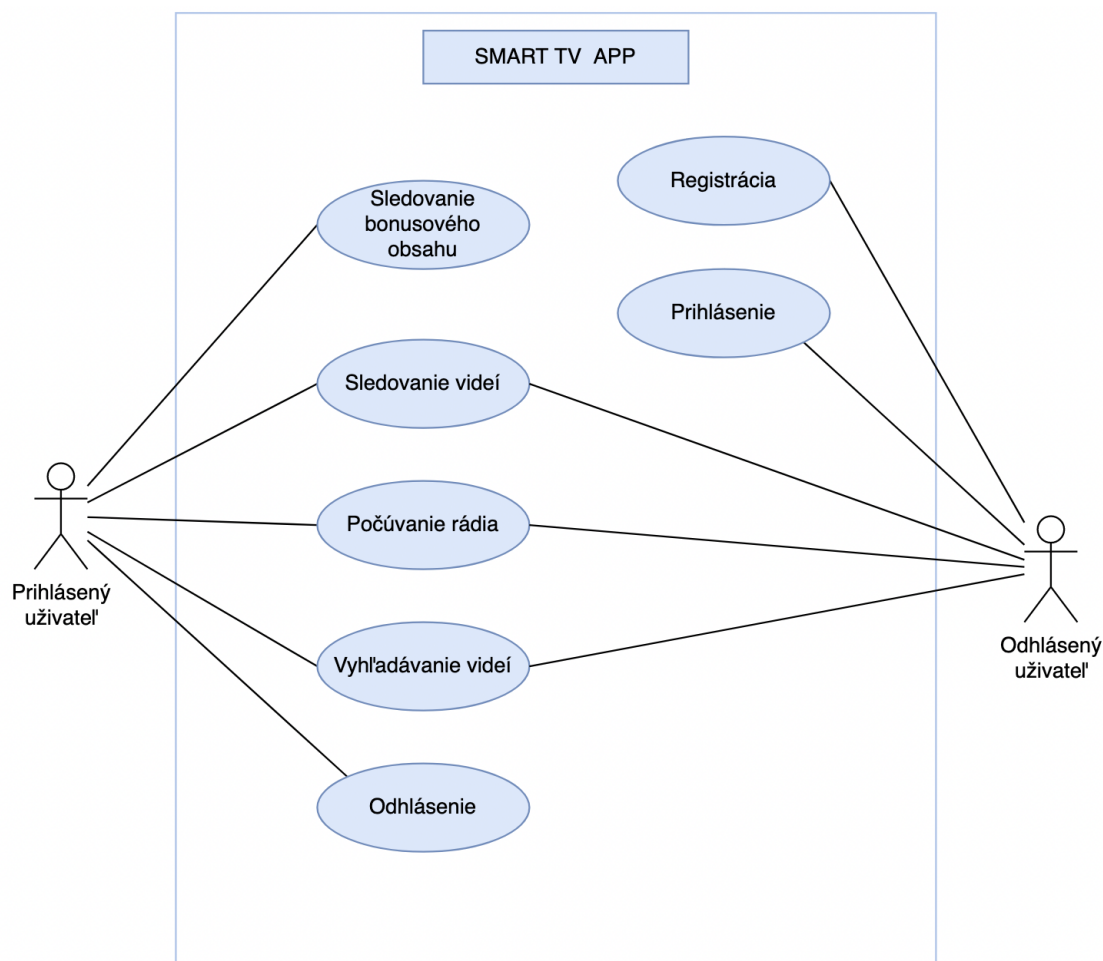
Po zvážení všetkých možností sa organizácia rozhodla, že určitý obsah bude prístupný len pre tých užívateľov, ktorí sa prihlásia cez svoj účet. Prihlasovanie bude jednoduché a rýchle, preto nezisková spoločnosť nepríde o veľa potenciálnych návštevníkov.

Prihláseným užívateľom to tiež dodáva niekoľko výhod. Tieto výhody spočívajú v tom, že prihlásení užívatelia budú mať prístup k niektorým videám z obmedzenej sekcie. Tiež prihlásení užívatelia budú mať možnosť neobmedzeného prehrávania rádia, kdežto neprihlásení užívatelia sú obmedzení len na desať minútovú lehotu.

2.2.4 Hlasové ovládanie

Užívateľom bude ponúknuté hlasové ovládanie, pomocou VoiceControl API, ktoré poskytuje rozhrania a metódy na rozpoznávanie hlasových príkazov. Hlasové ovládanie API ponúka funkcie na rozpoznanie hlasu a odoslanie výsledku ako preddefinovaný príkaz. [6]

Navyše väčšina novších modelov smart TV značky Samsung obsahuje hlasového asistenta, ktorý umožňuje používateľom aj písať text do vyhľadávača pomocou hlasového vstupu. Stačí stlačiť tlačidlo mikrofónu na diaľkovom ovládači a vysloviť hľadaný výraz. Hlasový asistent potom automaticky prepíše hovorený text na textový výraz a vyhledá výsledky.



Obrázek 2.1. Use case diagram.

■ 2.2.5 Funkčné požiadavky

Funkčná požiadavka v softvérovom a systémovom inžinierstve je vyhlásenie o zamýšľanej funkcii systému a jeho komponentov. Na základe funkčných požiadaviek inžinier určí správanie (výstup), ktoré sa od zariadenia alebo softvéru očakáva v prípade určitého vstupu. Dizajn systému je ranou formou funkčnej požiadavky. [7]

Aplikácia pre smart TV pre značku Samsung bude mať nasledujúce funkčné požiadavky:

- Aplikácia bude umožňovať prehrávanie videa a rádia z wordpress stránky organizácie CATVUSA.
- Aplikácia bude umožňovať užívateľovi vyhľadávať konkrétny obsah na základe kľúčových slov alebo frázy.
- Aplikácia bude poskytovať užívateľovi prihlásenie a umožniť mu tak prístup k bonusovému obsahu.
- Aplikácia bude umožňovať užívateľovi presmerovania na oficiálnu webovú stránku spoločnosti CATVUSA, kde sa môže dozvedieť viac o organizácii a podporiť ju.
- Aplikácia bude umožňovať užívateľovi hlasové ovládanie pomocou zabudovaného hlasového asistenta.
- Aplikácia bude byť po čase aktualizovaná, aby sa zabezpečila jej bezchybná funkčnosť a ochrana pred bezpečnostnými hrozbami.

■ 2.2.6 Nefunkčné požiadavky

Nefunkčné požiadavky (Non Functional Requirements) popisujú ako by systém, produkt alebo služba mali fungovať v kontexte jeho použitia. Napríklad požiadavky na výkonnosť, bezpečnosť, prevádzka atď. Dopĺňajú funkčné požiadavky. [8]

Aplikácia pre smart TV pre značku Samsung bude mať nasledujúce nefunkčné požiadavky:

- Aplikácia bude jednoduchá na používanie a užívateľsky prívetivá.
- Aplikácia bude dostatočne rýchla a efektívna, aby poskytovala plynulé užívateľské rozhranie a rýchlu odozvu na užívateľske vstupy.
- Aplikácia bude flexibilná a rozšíriteľná. Aplikácia bude navrhnutá tak, aby bola schopná rásť a rozvíjať.
- Aplikácia bude naprogramovaná s čistým a efektívnym kódom a mala by dodržiavať dobré programátorské praktiky a štandardy.
- Užívateľ bude schopný jednoducho navigovať medzi rôznymi kategóriami a funkciami aplikácie.
- Užívateľ bude schopný ovládať aplikáciu pomocou diaľkového ovládača, napríklad pomocou ovládacích tlačidiel.

Kapitola 3

Princípy dizajnu pre Smart TV

Ak sa pri vývoji aplikácii objavia pochybnosti, je dobré postupovať podľa pokynov výrobcu zariadenia. V prípade tejto práce je to výrobca Samsung. Väčšina odporúčaní a obmedzení v tejto kapitole je inšpirovaná článkami od vývojárov spoločnosti Samsung. Takmer každá platforma má svoje vlastné špecifické vzory a pokyny na dizajn, ktoré sa musia dodržiavať, aby sa zabezpečilo, že aplikácia bude prijatá a certifikovaná ako v obchode s aplikáciami. Dobrou správou je, že výrobcovia medzi sebou zdieľajú mnoho princípov dizajnu.

Dizajn pre Smart TV sa riadi súborom prísnych a neúprosných pravidiel. Ak sa tieto pravidlá nedodržia, môže to mať za následok v najlepšom prípade nepoužiteľnú aplikáciu a v horšom prípade aj stratu zákazníka alebo dokonca aj celého podnikania. Zvláštnosti televíznej obrazovky a spôsobu, akým ju užívatelia používajú, znemožňujú jednoduché skopírovanie osvedčených praktík dizajnu pre bežné mobilné alebo webové aplikácie.

Vďaka dobrému dizajnu vyzerajú aplikácie jednoducho na používanie a zároveň sa zvyšuje pravdepodobnosť ich opätovného použitia. Návrh aplikácii špeciálne na používanie v televízore musí venovať osobitnú pozornosť niektorými dôležitým faktorom, ktoré sa líšia od tých, ktoré sa používajú v prostredí stolných počítačov a mobilných zariadení.

3.1 Základné princípy dizajnu

V tejto podkapitole sa budú rozoberať kľúčové zásady a pravidlá, ktoré je potrebné mať na pamäti pri navrhovaní rozhrania pre inteligentné televízie. S rozvojom technológií sa televízne zariadenia stávajú čoraz sofistikovanejšími a ponúkajú užívateľom stále viac možností, čo zvyšuje dôležitosť pohodlného a intuitívneho rozhrania.

Rozhranie televízie by malo byť navrhnuté tak, aby bolo pre užívateľov jednoduché a intuitívne a aby umožňovalo rýchle a efektívne používanie aplikácie. Táto kapitola sa bude venovať kľúčovým faktorom, ktoré ovplyvňujú použiteľnosť rozhrania televízie, ako sú veľkosť a rozlíšenie obrazovky, ovládacie prvky, navigácia, použiteľnosť hlasového ovládania a ďalšie faktory.

Taktiež sa bude zaoberať pravidlami pre navrhovanie rozhrania pre inteligentné televízie, ako sú jednoduchosť, zrozumiteľnosť, konzistentnosť, prispôbitelnosť a ďalšie. Cieľom tejto kapitoly je poskytnúť návod na návrh pohodlného a užívateľsky prívetivého rozhrania pre inteligentné televízie, ktoré budú užívateľom pohodlné a jednoduché v používaní.

V tejto časti o správnom návrhu aplikácie pre Smart TV sa ukážu obrázky a príklady, ktoré zobrazujú správne riešenia, väčšinou podľa spoločnosti Samsung, pre užívateľské rozhranie a navigáciu v aplikácii. Tieto príklady budú ukazovať, ako správne usporiadať ovládacie prvky, ako optimalizovať použiteľnosť a ako minimalizovať námahu užívateľa.

■ 3.1.1 Jednoduchosť

Ludia si pri sledovaní televízie už tradične snažia urobiť pohodlie a relaxovať. Preto nie je vhodné aby televízni diváci využívali zložité rozloženie obrazovky alebo zložité ovládacie prvky.

Zložitejšia aplikácia nemusí byť nevyhnutne lepšia. Grafické rozvrhnutie (angl. layout) by malo byť užívateľsky prívetivé s jasnými, ľahko prístupnými funkciami, ktoré sú usporiadané pohodlným a intuitívnym spôsobom. Rozloženie jednotlivých obrazoviek by nemalo byť tiež príliš komplikované. Nepotrebné úrovne obrazovky musia byť odstránené a prechod na rôzne úrovne alebo návrat znova na predošlú úroveň musí byť zrejmý. Aplikácia by mala byť na toľko jednoduchá aby si nevyžadoval samostatný návod alebo samostatnú príručku. [3]

■ 3.1.2 Jednoznačnosť

Jedným z najdôležitejších faktorov pri navrhovaní TV aplikácie je zahrnutie jasnej a presnej navigácie pre užívateľské operácie. Navigácia pre inteligentné televízie by mala byť navrhnutá tak, aby ponúkala čo najkratšiu cestu k obsahu. To v podstate znamená, že aplikácie by nemali prinútiť užívateľa prejsť cez množstvo obrazoviek, kým sa dostane k požadovanému obsahu. Zároveň by každá obrazovka mala mať konzistentný dizajn a mala by sa dať jednoducho ovládať pomocou vstupného zariadenia. Ak je navigácia nejednoznačná, užívatelia sa budú cítiť zmätení a neistí. Pravdepodobne zatvoria aplikáciu a možno ju už nikdy nebudú používať. Stručne povedané, užívatelia by mali vždy presne vedieť, kde sa v rámci aplikácie nachádzajú. Pohyb medzi jednotlivými úrovňami obrazoviek, návrat späť, potvrdenie určitej akcie a ďalšie základné navigačné funkcie musia byť jednoznačné. Užívatelia musia byť schopní ihneď rozpoznať aká akcia sa vykoná pri spustení určitej operácie. [3]

■ 3.1.3 Konzistentnosť

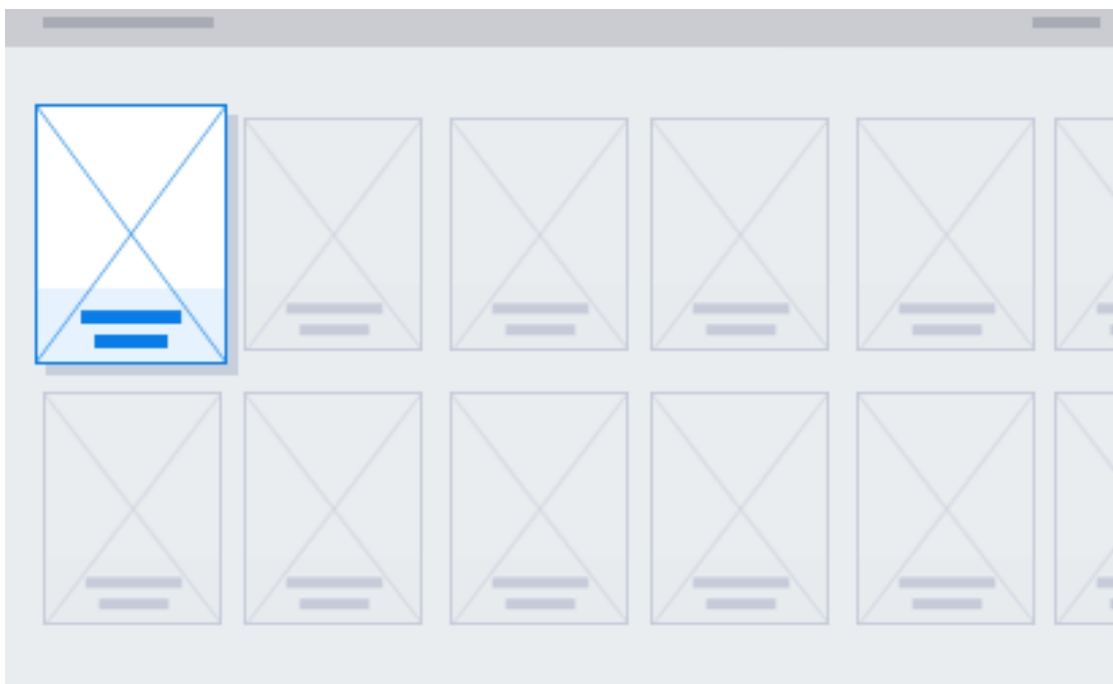
Konzistentnosť úzko súvisí so zlepšením použiteľnosti a učenia. Pri používaní novej aplikácie skúsenosti s podobnou aplikáciou zabezpečia užívateľovi rýchle učenie. Neexistujú žiadne veľké obmedzenia týkajúce sa dizajnu tlačidiel. Na druhej strane však dodržiavanie rovnakého návrhu pre tlačidlá, ktoré sa týkajú rovnakých funkcií však pomáha užívateľom rýchlo, jednoducho a efektívne sa naučiť používať aplikáciu. Aplikácia by mala byť celkovo konzistentná nie len že každá obrazovka má konzistentný dizajn, ale aj spolu so všetkými potrebnými základnými navigačnými prvkami, ako je štvorsmerná navigácia, prechod do jednotlivých úrovní alebo návrat na predchádzajúcu obrazovku. [3]

■ 3.1.4 Spätná väzba

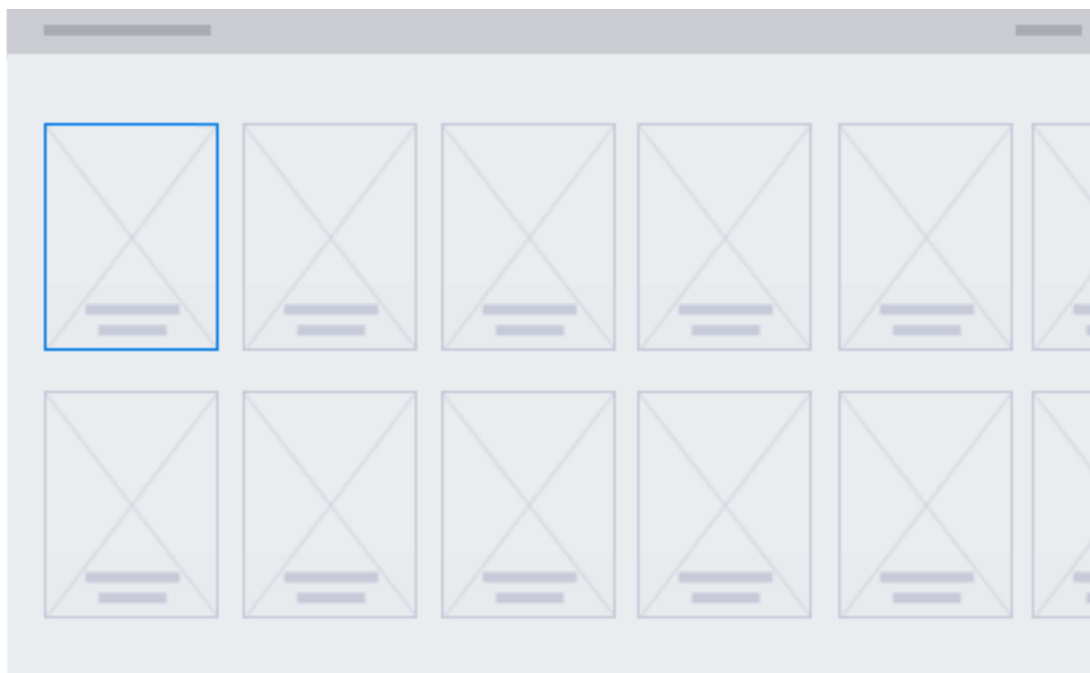
Keď čas potrebný na otvorenie aplikácie alebo načítanie údajov prekročí určitú dobu, musí sa zobrazit animácia načítania, ktorá vizuálne potvrdí, že sa obrazovka čoskoro zmení podľa požiadavky užívateľa.

Pretože na obrazovke Smart TV nie je žiadny kurzor myši ani dotykový vstup, rozhranie musí tiež implementovať niečo známe ako stav zaostrenia – vždy musí byť používateľovi jasné, ktorý prvok je momentálne zaostrený, zvyčajne zvýraznením prvok alebo pridanie rôznofarebného okraja okolo neho. Pri zobrazovaní položiek, ktoré je možné vybrať pomocou zamerania, musí byť stav položiek zobrazený spôsobom, ktorý umožňuje užívateľom ľahko identifikovať rozdiel medzi položkou, ktorá je vybraná pomocou diaľkového ovládača a položkou, ktorá naopak nebola vybraná. [3]

Forma zobrazenia zaostrenia sa môže líšiť v závislosti od komponentu. Vyžaduje sa však konzistentnosť v rámci aplikácie. Jasné a dobre viditeľné zaostrenie pomáha užívateľovi rýchlo rozpoznať aktuálnu polohu na obrazovke a uľahčuje navigáciu. [3]



Obrázek 3.1. Dobrý príklad zamerania.



Obrázek 3.2. Zlý príklad zamerania.

3.2 Uživatelské prostredie

Samsung uvádza niekoľko dôležitých odporúčaní pre uživatelské prostredie, ktoré pomôžu vytvoriť uživatelský prívetivé prostredie pre televízne aplikácie. Uživatelská prívetivosť je to, ako sa používateľ cíti pri používaní systému – webovej stránky, mobilnej aplikácie, desktopového softvéru alebo v podstate akéhokoľvek mysliteľného rozhrania, s ktorým ľudia interagujú. Je to mnohostranné a dosť ťažko uchopiteľné, no zároveň rozhodujúce pre úspech aplikácie. [3]

3.2.1 Priemerná vzdialenosť medzi televízorom a užívateľom

Samsung uvádza, že priemerná vzdialenosť medzi televízorom a divákmi by mala byť približne tri metre. [3]

Pri navrhovaní aplikácie je potrebné mať na pamäti, že väčšina užívateľov sedí približne 10 stôp alebo 3 metre od obrazovky. Štýly, ako sú farby, písma a veľkosti prvkov, by sa mali testovať na obrazovkách zo vzdialenosti až 3 metrov. [9]

Televízory sa dodávajú v mnohých rôznych veľkostiach obrazovky, od malých televíznych monitorov až po inteligentné televízory s uhlopriečkou takmer 100 palcov. Aj keď sa veľkosť obrazovky môže líšiť od zariadenia k zariadeniu, medzinárodný štandardný pomer strán je dohodnutý na 16:9. Aplikácia je zvyčajne navrhnutá s veľkosťou obrazovky 1920 x 1080 pixelov, potom je možné v rámci kódu jednoducho proporcionálne zmeniť veľkosť všetkých prvkov pre obrazovky HD alebo 4K. [9]

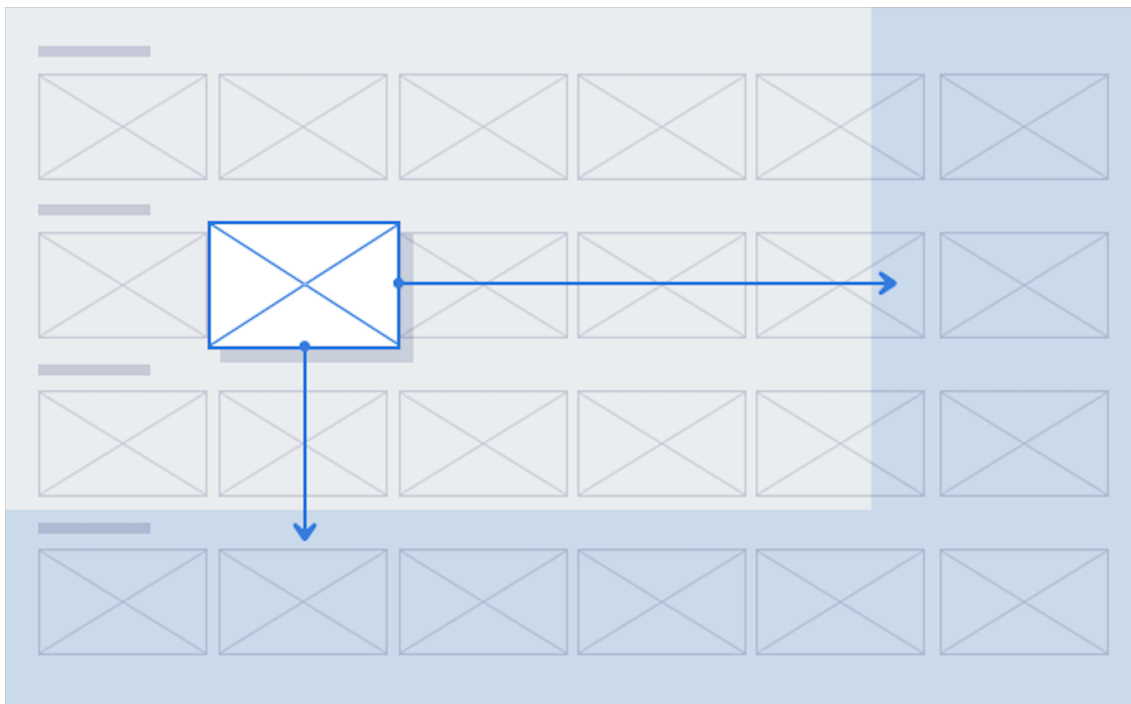
3.2.2 Televíziu používa viac osôb

Je potrebné vziať do úvahy situácie, keď zariadenie používa viac osôb. Pre aplikácie, ktoré môžu zobrazovať osobné informácie, musia byť poskytnuté preventívne opatrenia na ochranu súkromia. V prípade aplikácií, ktoré vyžadujú prihlásenie, musí byť jasné, kto je prihlásený používateľ. V prípade potreby musí byť pre odhlásených užívateľov poskytnuté jednoduché opätovné prihlásenie. [3]

3.2.3 Prechod medzi obrazovkami

Je vhodné myslieť na to, že časté prechody medzi obrazovkami spôsobia, že sa používatelia budú cítiť stratení a sťažia navigáciu v aplikácii. Prechody obrazoviek by preto mali byť zahrnuté len vtedy, keď sú absolútne nevyhnutné.

Ak je kategória aplikácií a príslušný zoznam obsahu rozdelené do dvoch úrovní, potom sa odporúča použiť plochú navigáciu (kde tlačidlá nahor a nadol prepínajú medzi kategóriami a ľavé a pravé tlačidlo na navigáciu v obsahu v rámci každej kategórie). To znamená, že užívateľ môže navigovať pomocou štvorsmerných pohybov, pričom má neustále prehľad o svojej aktuálnej polohe na obrazovke. [3]



Obrázek 3.3. Príklad plochej navigácie.

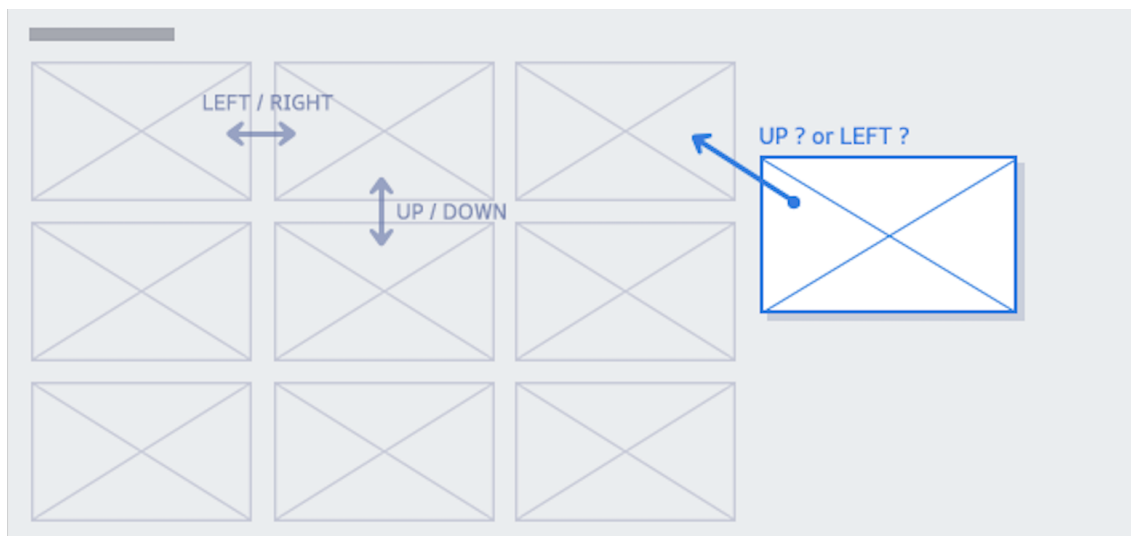
■ 3.2.4 Ovládacie prvky televízora

Štandardný diaľkový ovládač je základným ovládacím prvkom televízora. Dobrá navigácia je na televíznych platformách kľúčová. Pretože písanie na diaľkovom ovládači je zvyčajne vyčerpávajúca úloha, dobrý dizajn by mal užívateľom ušetriť čo najviac problémov.

Dobré rozhrania televíznych aplikácií sú zvyčajne obmedzené iba na navigáciu pomocou „križníka“ a „tlačidla OK“ na diaľkovom ovládači. Využitie pomocou hlasového ovládania sa tiež stáva čím ďalej tým viac populárne. Diaľkové ovládanie má aj iné množstvo tlačidiel, ktoré nie je vhodné pri návrhu využiť a to z jednoduchého dôvodu. Existuje niekoľko značiek, ktoré vyrábajú televízory, a diaľkové ovládače sa dodávajú vo všetkých tvaroch a veľkostiach. Zriedkavo sú kompatibilné a nehovoria rovnakým dizajnovým jazykom.

Spôsob ovládania by mal zodpovedať ovládaciemu zariadeniu a podľa toho musí byť poskytnuté intuitívne rozloženie. Aby bolo dosiahnuté intuitívna interakcia s užívateľom, názvy tlačidiel a ikony na diaľkovom ovládači sa musia zhodovať s akciami, ktoré sa vykonávajú na obrazovke. Smer pohybu objektov, ktoré sa vyskytujú na obrazovke, by mali byť v smere, ktoré očakáva užívateľ.

Dôležité je najmä rozmiestnenie komponentov. V prostredí ovládania televízora, kde je najnákladnejším ovládaním štvorsmerové ovládanie, je použitie mriežky najlepším spôsobom, ako minimalizovať zmätok. Ak má rozloženie diagonálne umiestnenie, ako je to na obrázku 3.4, užívateľ bude zmätený, pokiaľ ide o to, kam sa jeho stlačenie tlačidla bude pohybovať ďalej. [3]



Obrázek 3.4. Zmätok spôsobený diagonálnym umiestnením.

3.3 Kontrolný zoznam od Samsungu

Pri vytváraní aplikácie pre Samsung Smart TV bolo postupované podľa zásad popísaných v príručke pre návrh užívateľského rozhrania. Tento kontrolný zoznam požiadaviek a odporúčaní uvedených nižšie sú základnými kritériami, ktoré by mali byť splnené, aby bola aplikácia schválená. Pred registráciou aplikácie by mali byť skontrolované nasledujúce špecifikácie. Zoznam obsahuje povinné požiadavky ktoré musia byť splnené ale aj odporúčania, ktorých splnenie sa nevyžaduje. [4]

Ak nasledujúce požiadavky nie sú splnené, žiadosť o publikáciu aplikácie môže byť zamietnutá. [4]

3.3.1 Všeobecné

- Ikony aplikácií musia mať 1920 x 1080 pixelov.

3.3.2 Vstup

- Všetky akcie tlačidiel na diaľkovom ovládači musia byť v súlade s týmto zoznamom.
- Užívatelia musia byť schopní nájsť všetky označiteľné objekty pomocou štyroch navigačných tlačidiel (šípky do všetkých smerov).

3.3.3 Prehrávač médií

- Na ovládanie prehrávania musia byť užívateľom poskytnuté ovládacie prvky prehrávania na obrazovke.
- Všetky tlačidlá prehrávania na diaľkovom ovládači musia správne fungovať, aby bolo možné správne ovládať prehrávanie multimediálnych súborov.

■ 3.3.4 Spätná väzba

- Musí byť poskytnutá jednoznačná spätná väzba pre interakcie alebo stav aplikácie.
- V jednom momente musí byť možné vybrať práve jeden objekt.
- Vybraný objekt musí byť jasne rozpoznateľný oproti ostatným objektom súborov.
- Vzor pre vybraný objekt alebo obrázok by mal byť podobný alebo konzistentný v rámci celej aplikácie. Je potrebné sa vyhnúť používaniu rôznych vzorov.
- Užívateľom by sa mala poskytnúť animácia priebehu, ak akcia prebieha alebo ak to chvíľu trvá.
- Keď sa vyskytne chyba, mali by sa poskytnúť informácie o tom, prečo k chybe došlo a ako ju vyriešiť.

Nasledujúci zoznam obsahuje len odporúčania, ktoré by podľa vývojárov spoločnosti Samsungu mala obsahovať aplikácia pre inteligentné televízie. Predošli zoznam obsahoval povinné požiadavky. [4]

■ 3.3.5 Všeobecne odporúčania

- Ak zoznam alebo položky obsahu presahujú jednu stranu, musí sa zobrazíť ukazovateľ pre rolovanie.
- Akékoľvek iné oblasti, ktoré je možné označiť by sa nemali nachádzať v hornej a dolnej časti zvisle rolovateľných zoznamov ani napravo a vľavo od vodorovne rolovateľných zoznamov.
- Všetky zaostriteľné objekty by mali byť zarovnané v mriežkach, nie diagonálne
- Príliš veľa štýlov písma na rovnakej obrazovke alebo okne by sa nemalo používať.
- Vzory používateľského rozhrania aplikácie by mali byť v súlade so základnými vzormi používateľského rozhrania televízora Samsung Smart TV, aby sa predišlo zmätku používateľa. Tieto vzory sú zhrnuté v podkapitole 3.2 a 3.1 .
- Celkové vizuálne informácie na obrazovke by sa mali upraviť tak, aby sa znížilo pracovné zaťaženie užívateľa.
- Podobné informácie by mali byť zlúčené alebo zoskupené pre efektívnu navigáciu.
- Aby sa predišlo chybám užívateľa, jednotlivé obrazovky by mali byť konzistentné.
- Na otváranie alebo ovládanie užívateľského rozhrania by sa od užívateľov nemalo vyžadovať nadmerné pohybové interakcie.

Kapitola 4

Analýza a návrh užívateľského rozhrania

Je dôležité analyzovať problémy a navrhovať riešenia, pretože to môže pomôcť zlepšiť novú aplikáciu a dosiahnuť určitý cieľ, ktorý je v tomto prípade vytvoriť jednoduchú a intuitívnu aplikáciu, ktorá zachová farby spoločnosti CATVUSA. Usporiadanie prvkov by malo byť podobné ako pri aplikácií pre Android TV, nie je to však úplnou podmienkou zo strany zadávateľa.

Bez analýzy a návrhu riešenia sa môže stať, že sa budú opakovať rovnaké chyby a problémy, a to môže mať negatívny dopad na efektívnosť. Pri vývoji novej aplikácie je dôležité nielen sa sústrediť na funkčnosť, ale aj na to, aby bola aplikácia pre užívateľa jednoduchá a intuitívna. Jedným z dôležitých prvých krokov pri vývoji aplikácie je navrhnutie užívateľského rozhrania. V tomto prípade, sa aplikácia pre Samsung Smart TV bude snažiť zachovať farby a usporiadanie prvkov, na ktoré sú niektorí užívatelia zvyknutí z používania predchádzajúcej verzie určenej na Android TV. Na druhej strane, aby bolo zabezpečený čo najlepší návrh a aby bola zvýšená účinnosť aplikácie, je nutné v prípade potreby urobiť niektoré zmeny. Jednou z týchto zmien môže byť zvýraznenie určitých prvkov, ako napríklad tlačidlo vďaka, ktorému sa užívateľ dostane na stránky organizácie. Táto zmena môže mať vplyv aj na marketingové ciele, keďže to môže prilákať pozornosť a zvýšiť angažovanosť skladovateľov.

Vzhľadom na tieto faktory, sa bude vynakladať úsilie vyvinúť aplikáciu, ktorá bude zohľadňovať nielen pôvodné farebné schémy a usporiadanie prvkov, ale aj urobiť niektoré zmeny, ktoré zlepšia užívateľskú prijateľnosť a splnia marketingové ciele.

V tejto kapitole sa bude teda porovnať riešenie aplikácie pre Android TV, podľa ktorej zákazník požaduje aj riešenie, s novým návrhom. Budú sa analyzovať ich rozdiely a predstaví sa aj návrh upraveného užívateľského rozhrania.

4.1 Aplikácia pre Android TV

Pri navrhovaní užívateľského rozhrania pre aplikáciu pre Smart TV je dôležité analyzovať predchádzajúce riešenie aplikácie a zvážiť jeho silné a slabé stránky.

Pri zadávaní požiadavok na aplikáciu, sa zadávateľ tejto práce opieral o riešenie pre Android TV a vysvetľoval slabé a silné stránky. V tejto časti bude predstavené riešenie tejto aplikácie. Následne bude predstavené nové riešenie, a bude sa aj poukazovať na novo-vzniknuté zmeny.

4.1.1 Hlavná obrazovka

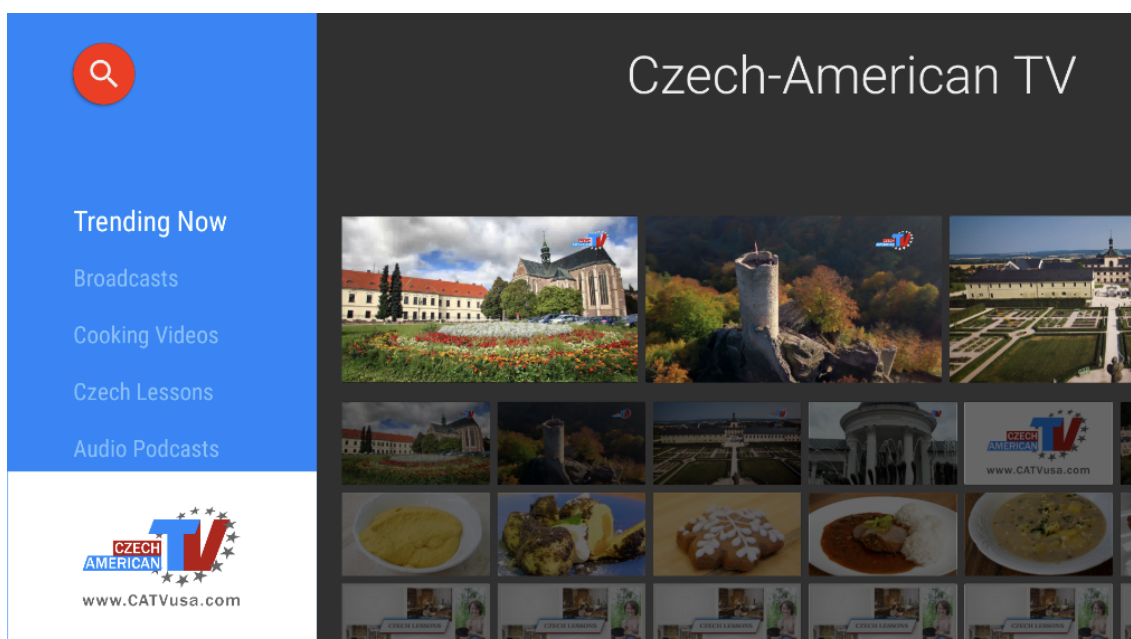
Aplikácia má pevne umiestnenú navigáciu na ľavej strane obrazovky, ktorá obsahuje celkovo 6 položiek. Prvá položka predstavuje posledné tri vysielania, ktoré sú zvýraznené svojou veľkosťou, aby bolo jasné, že ide o aktuálne videá, ktoré boli nedávno pridané. Nasledujúce tri položky predstavujú rôzne kategórie videí, ktoré sú v aplikácii k dispozícii. Každá z týchto kategórií má svoj vlastný názov a obsahuje videá zodpovedajúce

danému tématu. Okrem toho, v navigácii sa nachádza aj položka, ktorá odkazuje na rádio. Pod poslednou položkou sa nachádzajú niekoľko ďalších možností, ako je napríklad podporiť neziskovú organizáciu, alebo možnosť prechodu na stránky organizácie.

V ľavom dolnom rohu obrazovky sa nachádza logo aplikácie, ktoré sa vždy zobrazuje, bez ohľadu na to, akú položku z navigácie používateľ práve zvolil. Navigácia mizne, keď sa používateľ zameria na určitú položku mimo menu a vstúpi do jej obsahu.

Na obrazovke sa nachádzajú obdĺžniky s videami, ktoré sú rozdelené do kategórií podľa zvolenej položky v navigácii. Každé video má svoj vlastný obrázok, názov a popis, aby užívateľ vedel, o čom dané video je.

Celkovo je táto obrazovka prehľadná a umožňuje užívateľom jednoduchý prístup k videám a hudbe v rôznych kategóriách a žánroch. Navigácia a rozdelenie obsahu do kategórií robia aplikáciu jednoduchou na použitie a zároveň zaručujú, že užívateľ sa ľahko zorientuje v obsahu a nájde to, čo hľadá. Je viac než pravdepodobné, že si na používanie tejto aplikácie sledovatelia rýchlo zvyknú. Aj to je jeden z dôvodov prečo je dobre zachovať určitú jednotnosť medzi jednotlivými aplikáciami.



Obrázek 4.1. Hlavná obrazovka pre Android TV.

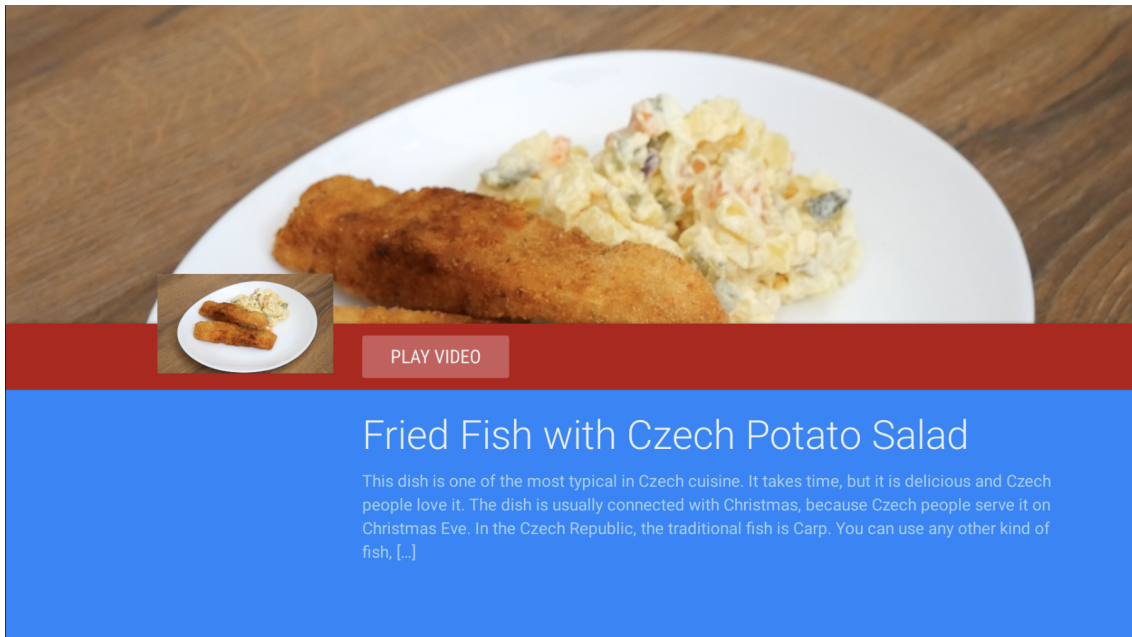
4.1.2 Obrazovka detailu videa

Obrazovka detailu videa v aplikácii pre Android TV má nasledovný dizajn. V dolnej časti obrazovky sa nachádzajú detailné informácie o videu, ktoré obsahuje názov videa a jeho popis. Táto časť je zvyčajne naplnená relevantnými informáciami o videu, aby sa užívateľ mohol ľahko rozhodnúť a vybrať si, či chce dané video sledovať.

Nad popisom sa nachádza miniatúra titulného obrázku spolu s tlačidlom prehrať video, ktoré slúži na spustenie videa. Po stlačení tohto tlačidla sa video spustí a prehrá sa na celú obrazovku.

Titulný obrázok sa tiež nachádza na pozadí, čím tvorí zvyšok obrazovky. Tento prvok pridáva na vizuálnom dojme a umožňuje užívateľovi vidieť, aké video si práve vybral.

Celkovo je dizajn obrazovky detailu videa v aplikácii pre Android TV výrazne zjednodušený, aby užívateľ mohol ľahko a rýchlo nájsť a prehrať video, ktoré hľadal.

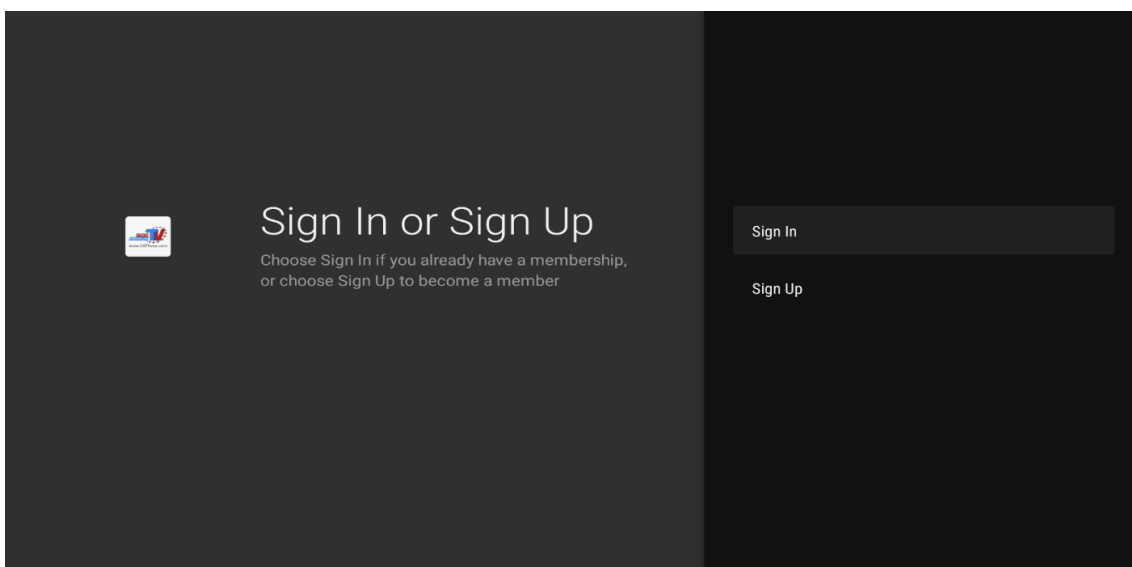


Obrázek 4.2. Obrazovka detailu videa pre Android TV.

4.1.3 Obrazovka prihlásenia a registrácie

Obrazovka, kde sa užívatelia môžu prihlásiť a získať prístup k bonusovému obsahu, je veľmi jednoduchá a zrozumiteľná. Na ľavej strane obrazovky sa nachádza logo spoločnosti a krátky popis, ktorý informuje užívateľov o tom, že ide o obrazovku určenú na prihlasovanie a registráciu.

Na pravej strane obrazovky sa nachádzajú tlačidlá s nápisom prihlásiť sa alebo registrovať sa. Tieto tlačidlá užívateľovi zobrazia vstupné polia, kde je možné zadať príslušné prihlasovacie údaje, ako sú meno a heslo, a následne sa prihlásiť.

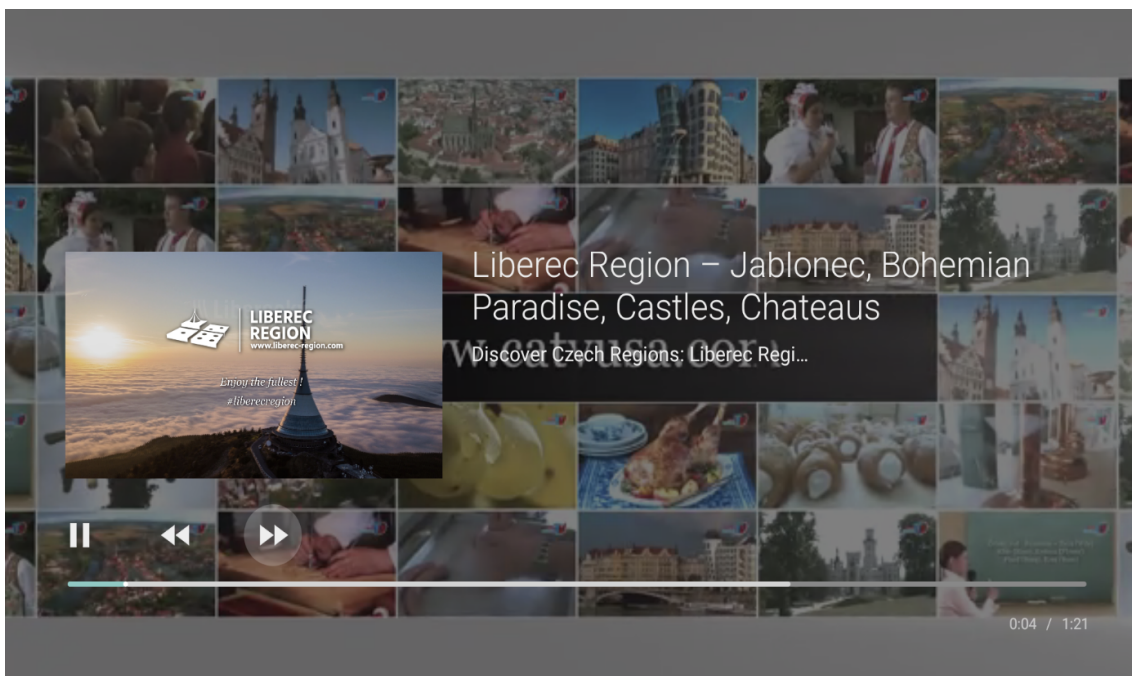


Obrázek 4.3. Obrazovka detailu videa pre Android TV.

4.1.4 Obrazovka videa a rádia

Na obrazovke videa sa nachádza obrázok videa, tlačidlá na ovládanie ako je pozastaviť video alebo ho spustiť, a názov videa. Tlačidlá na ovládanie, ako je pozastavenie alebo spustenie videa, sú umiestnené pod obrázkom videa a umožňujú užívateľom plne ovládať prehrávanie videa. Názov videa je umiestnený tiež v dolnej časti obrazovky a slúži na identifikáciu a orientáciu v obsahu videa. Celkovo sa na obrazovke videa nachádza viacero dôležitých prvkov, ktoré umožňujú užívateľom plne ovládať prehrávanie videa a orientovať sa v jeho obsahu.

Takmer identický vyzerá aj obrazovka radia, vzhľadom na to, že neexistuje žiadny pohyb na obrazovke, používa sa často len statický obraz, ktorý zobrazuje názov skladby alebo iné informácie.

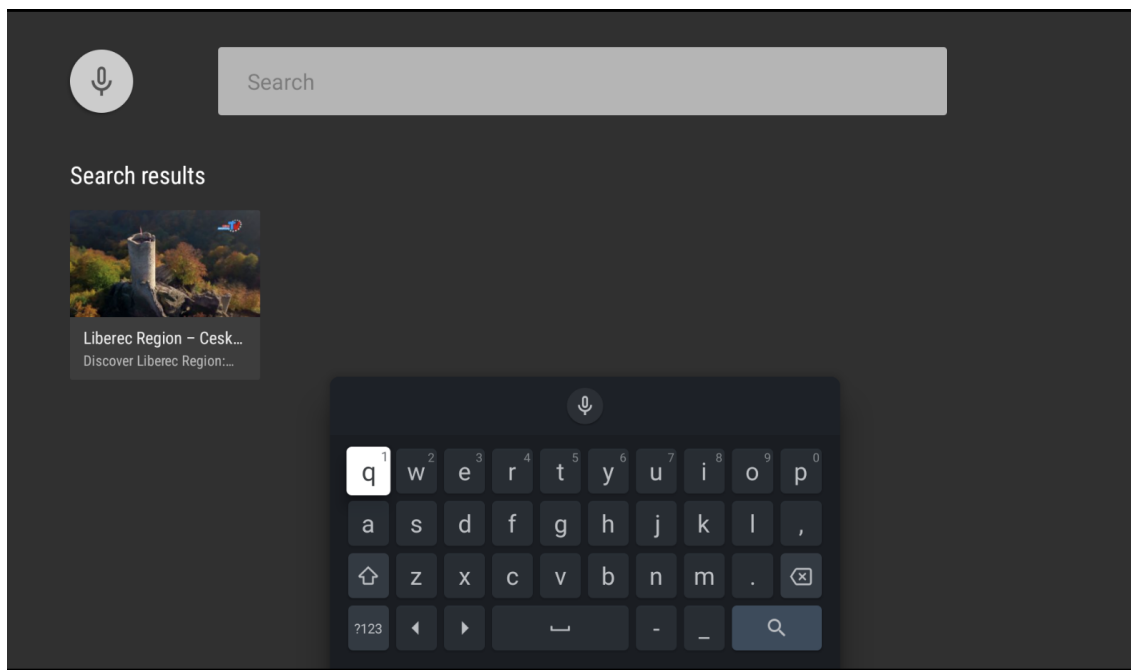


Obrázek 4.4. Obrazovka videa a radia pre Android TV.

4.1.5 Obrazovka vyhľadávania

Obrazovka vyhľadávania umožňuje vyhľadávanie videí na základe zadaného textu, je usporiadaná tak, že má v hornej časti vyhľadávacie pole, kam si užívateľ môže napísať hľadaný výraz. V nej sa zobrazujú aj posledné hľadané výrazy pre prípad, že by užívateľ chcel vyhľadávanie opakovať.

Vyhľadávanie sa spustí keď užívateľ potvrdí, že zadal požadovaný text. Po spustení vyhľadávania sa pod vyhľadávacím poľom zobrazí zoznam výsledkov vyhľadávania. Výsledky sú zvyčajne zoradené podľa relevancie, pričom najrelevantnejšie výsledky sú zobrazené na vrchu zoznamu. Každý výsledok zobrazuje miniatúru videa, jeho názov a krátky popis. Kliknutím na miniatúru videa sa spustí prehrávanie.



Obrázek 4.5. Obrazovka vyhľadávania pre Android TV.

4.2 Návrh zmien

Táto časť diplomovej práce bude zameraná na vylepšenie navigácie a užívateľskej prívetivosti aplikácie pre Samsung Smart TV, ktorá sa podobá na aplikáciu pre Android TV. Cieľom je vytvoriť užívateľsky prívetivejšiu aplikáciu, ktorá bude lepšie uspokojovať požiadavky užívateľov a umožní im jednoduchšie a intuitívnejšie používanie.

Tento návrh zmien bol zostavený spoločne s riaditeľom organizácie CATVUSA. Návrh na zlepšenie vzhľadu a funkčnosti aplikácie bol vytvorený na základe spoločných skúseností spolu s riaditeľom, ako aj na základe odporúčaní, ktoré Samsung odporúča pre vylepšenie užívateľského zážitku.

4.2.1 Hlavná obrazovka

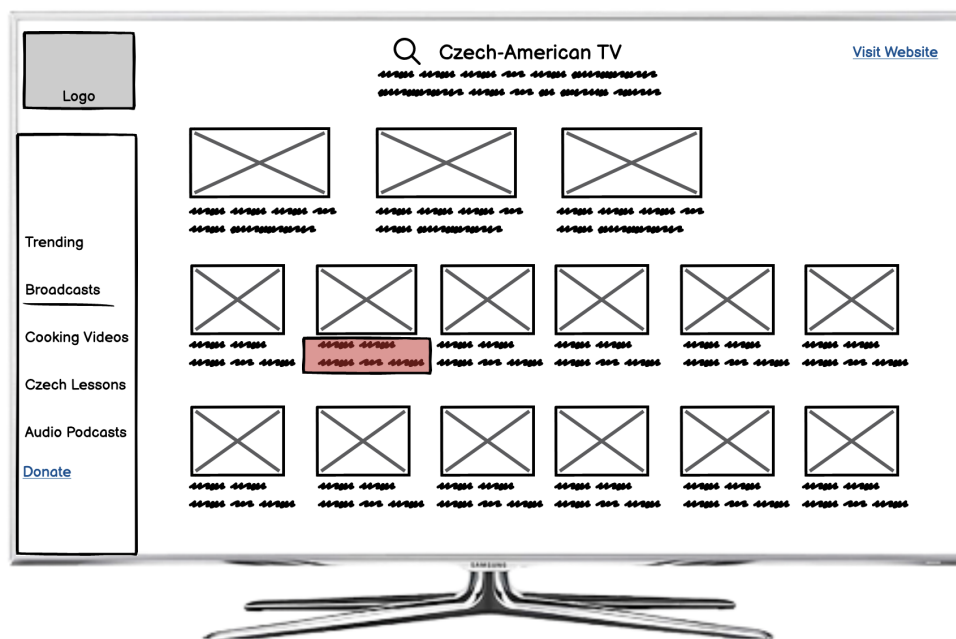
- Jedným z hlavných problémov, ktoré boli identifikované v pôvodnej aplikácii, bolo zlepšenie navigácie. V aplikácii pre Android bolo ťažké nájsť, ktorá položka v menu bola zvolená, pretože položky len menili svoj jas. Čo môže byť dostatočne viditeľné, ale je ťažko predpokladať, aké budú mať užívatelia svetelné podmienky alebo ako bude nastavený jas televíznej obrazovky. Tiež sa môže stať že užívateľ bude mať zdravotné problémy v oblasti zraku. V aplikácii pre Samsung Smart TV bude preto pridané podčiarknutie inou farbou pre aktuálnu položku, čo umožňuje užívateľom rýchlejšie a ľahšie vidieť, ktorá položka je aktuálne vybraná. To by mohlo mať pozitívny vplyv na celkovú použiteľnosť aplikácie, pretože užívatelia sú schopní jednoduchšie a rýchlejšie navigovať v rámci aplikácie. Okrem toho prispieva k vizuálnej prívetivosti aplikácie a robí ju esteticky príjemnejšou pre užívateľov.
- Ďalšou zmenou v návrhu je umiestnenie loga v ľavom hornom rohu. Túto požiadavku požadoval aj samotný zadávateľ a to hneď z niekoľkých dôvodov. Prvým dôvodom je orientácia televízora a zvyklosti užívateľov. Pri používaní televízora si užívatelia často sadajú alebo stoja pred televízorom, čo znamená, že ich zorné pole je viac v hornej časti obrazovky. To znamená, že logo v ľavom hornom rohu by bolo viac videné a

umožnilo by užívateľom ľahšie rozpoznať aplikáciu. Ďalším dôvodom by mohla byť dôležitosť vizuálneho usporiadania. Umiestnenie loga v ľavom hornom rohu dáva logo významnejšiu pozíciu v hierarchii obsahu. Vizuálne sa stáva prvým prvkom, ktorý užívatelia vidia, a to môže pomôcť zvýšiť identifikáciu značky a zapamätateľnosť aplikácie.

- V aplikácii bude tiež premiestnené tlačidlo, ktoré umožní užívateľom podporiť neziskovú organizáciu. Toto tlačidlo bude umiestnené v menu aplikácie, kde bude ľahko viditeľné a užívateľ si ho môže jednoducho všimnúť. Cieľom tohto tlačidla je zvýšiť povedomie o neziskovej organizácii a umožniť užívateľom presmerovať sa na stránku, kde je možné prispieť k jej činnosti.

Okrem toho bude vytvorené nové tlačidlo, ktoré bude umiestnené v pravom hornom rohu aplikácie. Toto tlačidlo povedie užívateľov na webové stránky spoločnosti, kde sa budú môcť dozvedieť viac o ich aktivitách a projektoch. Toto tlačidlo bude umiestnené výraznejšie, aby ho užívatelia jednoduchšie našli.

Týmto spôsobom aplikácia umožní užívateľom ľahko pristupovať k informáciám o neziskovej organizácii a prispieť k jej činnosti. Tlačidlá budú umiestnené tak, aby boli pre užívateľov dostatočne viditeľné a jednoducho prístupné. V prípade, že užívateľ bude mať záujem sa dozvedieť viac, bude mať možnosť kliknúť na tlačidlo, ktoré ho presmeruje na webové stránky organizácie, kde sa dozvie viac o ich projektoch a aktivitách. Celkovo, pridaním týchto tlačidiel do aplikácie sa podporí nezisková organizácia a zvýši sa jej povedomie.



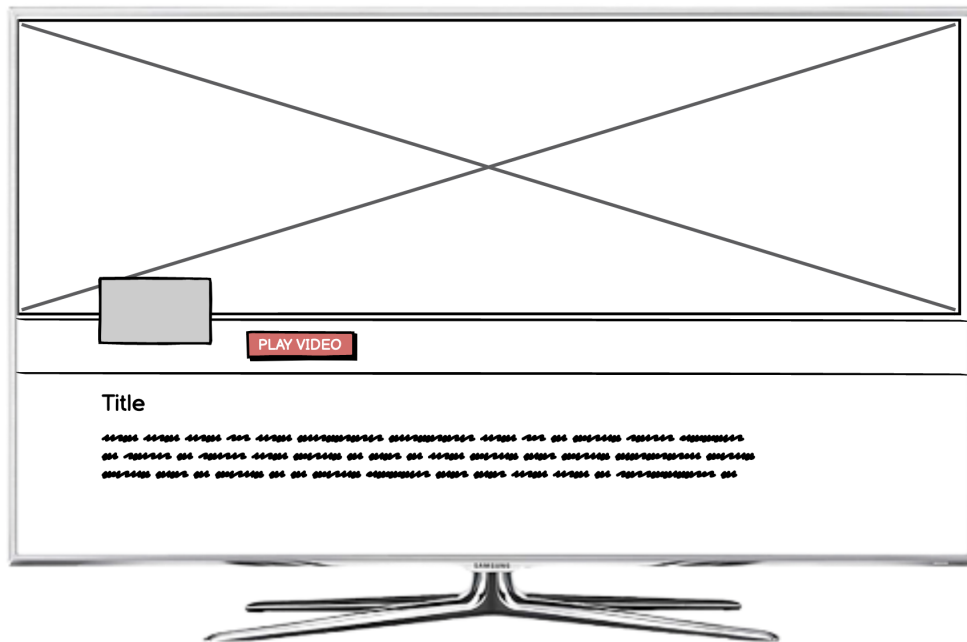
Obrázek 4.6. Návrh užívateľského rozhrania hlavnej obrazovky.

4.2.2 Obrazovka detailu videa

Táto zmena v návrhu obrazovky detailu videa zahŕňa nahradenie dvojitého zobrazenia titulného obrázka videa automatickým prehrávaním videa v hornej časti obrazovky. Cieľom tejto zmeny je zvýšiť angažovanosť užívateľa a uľahčiť mu rozhodnutie, či si chce video pozrieť. Tento prvý dojem môže byť rozhodujúci pre to, či si užívateľ video pozrie alebo nie, pretože v prvých sekundách si vytvorí názor na základe toho, čo vidí v práve prehrávanom videu.

Je však dôležité, aby bolo automatické prehrávanie videa implementované tak, aby bolo pre užívateľa príjemné a aby mu nebránilo v prezeraní informácií o videu, ktoré sú umiestnené v dolnej časti obrazovky. Napríklad, automatické prehrávanie by malo byť zvukovo potlačené alebo vypnuté, aby užívateľ nebol rušený zvukom videa, kým si prečíta informácie.

Jednou ďalšou zmenou, ktorú by bolo vhodné implementovať do aplikácie oproti Android TV, je zmena farby tlačidla pre prehrávanie videa. V súčasnosti je tlačidlo rovnakej farby ako pozadie, s odlišným odtieňom. Pre zlepšenie používateľskej skúsenosti by malo byť tlačidlo vždy výrazne odlišné farbou od ostatných prvkov v okolí.



Obrázek 4.7. Návrh užívateľského rozhrania obrazovky detailu videa.

4.2.3 Obrazovka prehrávača videa

Je dôležité pamätať na to, že prekryvanie videí veľkým textom, ako je napríklad názov videa, cez samotné video, ktoré je práve pozastavené, môže byť pre divákov veľmi rušivé a nepraktické. Tento prístup môže spôsobiť divákovi pocit frustrácie.

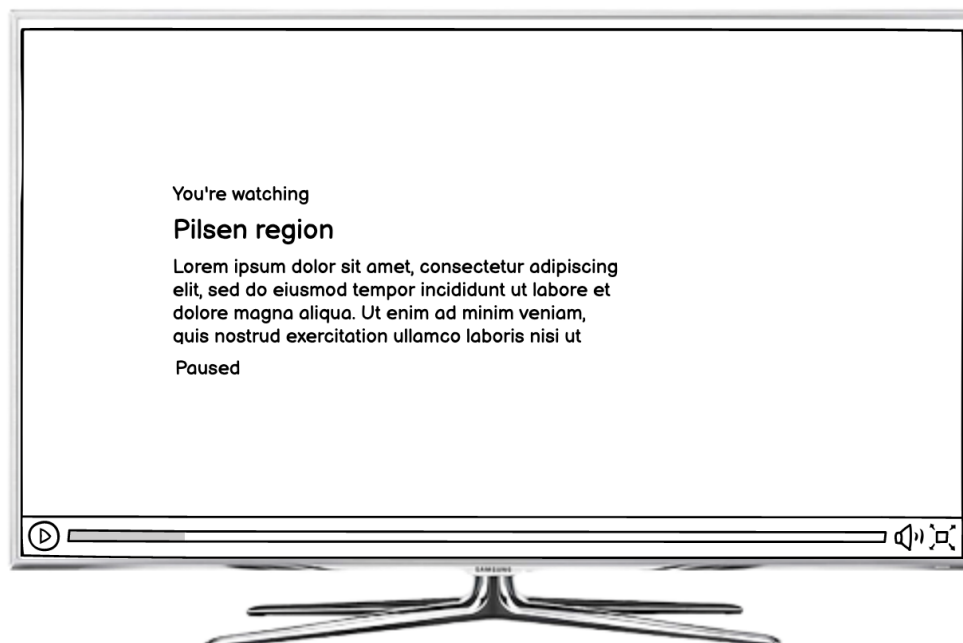
Vo výsledku je dôležité udržiavať rovnováhu medzi zdieľaním informácií o videu a zachovaním plynulého a príjemného zážitku pre divákov. Preto je dôležité premýšľať o tom, ako zdieľať informácie o videu, aby boli čo najmenej rušivé pre divákov, a aby to neprejavilo negatívny vplyv na sledovanie videa.

Je potrebné zvoliť vhodnú veľkosť písma a farbu tak, aby bol text čitateľný a zároveň nebol príliš rušivý pre divákov.

Nasledujúce riešenie bolo inšpirované americkou spoločnosťou Netflix, ktorá je predplatiteľská streamovacia služba, umožňujúca členom sledovať televízne programy a filmy na zariadení pripojenom na internet. [10] Keď užívateľ stlačí pauzu počas prehrávania filmu, zobrazí sa mu obrazovka s názvom "Paused" ("Pozastavené"), ktorá obsahuje informácie o prehrávanom titule, ale až po určitom časovom intervale. Netflix využíva tento prístup pri zobrazovaní informácií o seriáloch a filmoch, kde po určitom časovom intervale, keď je video pozastavené, sa zobrazia informácie o epizóde alebo filme. To môže byť užitočné pre užívateľov, ktorí sledujú množstvo videí po seba a strácajú informácie, čo vlastne sledujú. Takmer rovnaký princíp bude zahrnutý aj vo vyvíjanej

aplikácii. Na rozdiel od Netflixu budú vynechané, niektoré informácie ako je napríklad dĺžka videa, ktoré je možné vidieť na spodnej časti obrazovky spolu s tlačidlami pre prehrávanie.

Ďalej sú zobrazené už tradične možnosti prehrávania alebo zastavenia zvuku, ako aj ovládače pre pretáčanie vpred alebo vzad v rámci prehrávaného média.



Obrázek 4.8. Návrh užívateľského rozhrania obrazovky prehrávača videa pri pozastavenom videu.

4.2.4 Obrazovka prehrávača rádia

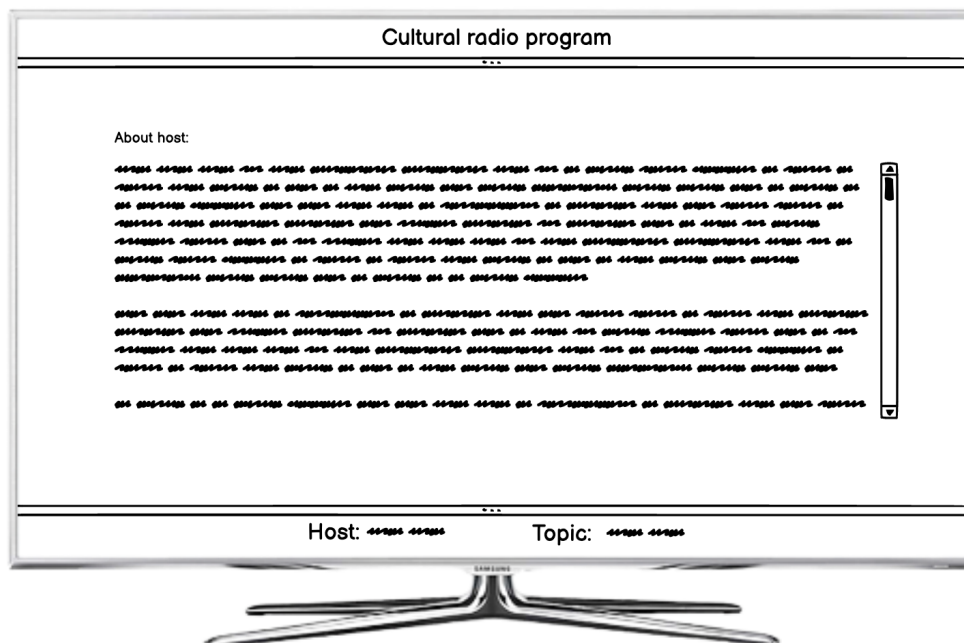
Súbežne s touto aplikáciou bolo aj vyvíjaný ďalší projekt študentom ČVUT. Jedná sa o API pre prístup k obsahu rádia. Toto API umožňuje užívateľom vybrať si z niekoľkých kategórií rádia a navyše, okrem samotných nahrávok, im poskytuje informácie o autoroch podcastov, skladateľoch klasickej hudby a ďalších podrobnostiach o skladbách a nahrávkach.

Vzhľadom na to, že sa jedná o rádio integrované do televízie, užívateľovi nie sú poskytnuté len textové informácie, ale aj iné obrazové prvky, ktoré zvyšujú celkový zážitok a pomáhajú užívateľom sa ešte viac ponoriť do obsahu šírené CATVUSA prostredníctvom týchto rádiových nahrávok. [11]

Obrazovka pre rádio, je zameraná na prehrávanie nahrávok pomocou nového API. Dizajn užívateľského rozhrania sa bude zhodovať s návrhom na webovej stránke, čo by mohlo byť pre užívateľov pohodlné, pretože sú zvyknutí na toto rozloženie. Celkovo webové stránky navštevuje viac užívateľov ako v prípade aplikácie pre Smart TV.

Na obrazovke je celkovo päť rôznych častí. Prvá časť zobrazuje názov kategórie videa alebo rádia, ktoré sa práve prehráva. Nasledujúca časť obsahuje krátku výzvu pre užívateľa. Napríklad, aby sa dozvedel viac o českej kultúre, jazyku a pod. Stredná a najväčšia časť obrazovky obsahuje obrázok, ktorý ukazuje, čo sa práve prehráva. Táto časť obrazovky pravdepodobne upúta pozornosť užívateľa a pomôže mu zorientovať sa v aplikácii. Posledné dve časti obsahujú informácie o ovládaní rádia a o prehrávanom obsahu. Zobrazujú sa tu informácie, ako napríklad názov autora, názov skladby a podobne. Užívateľia boli zvyknutí, že po kliknutí na strednú a najväčšiu časť sa im ukážu

par zaujímavých informácií. V prípade aplikácie pre Smart TV tak môžu urobiť pomocou šípok na diaľkovom ovládači. Je im aj umožnené pohybovať s obrazovkou smerom nadol a nahor, ak text bude príliš dlhý. Ovládanie pomocou šípok sa dalo využiť preto, lebo prehrávanie dopredu a späť rádiu nie je možné. Vo výsledku táto obrazovka by mala poskytovať užívateľom prehľadný a zrozumiteľný spôsob, ako prehrávať rádio v aplikácii pre Smart TV.



Obrázek 4.9. Návrh užívateľského rozhrania obrazovky prehrávača rádia pri zobrazovaní dodatočných informácií.

4.2.5 Obrazovka vyhľadávania

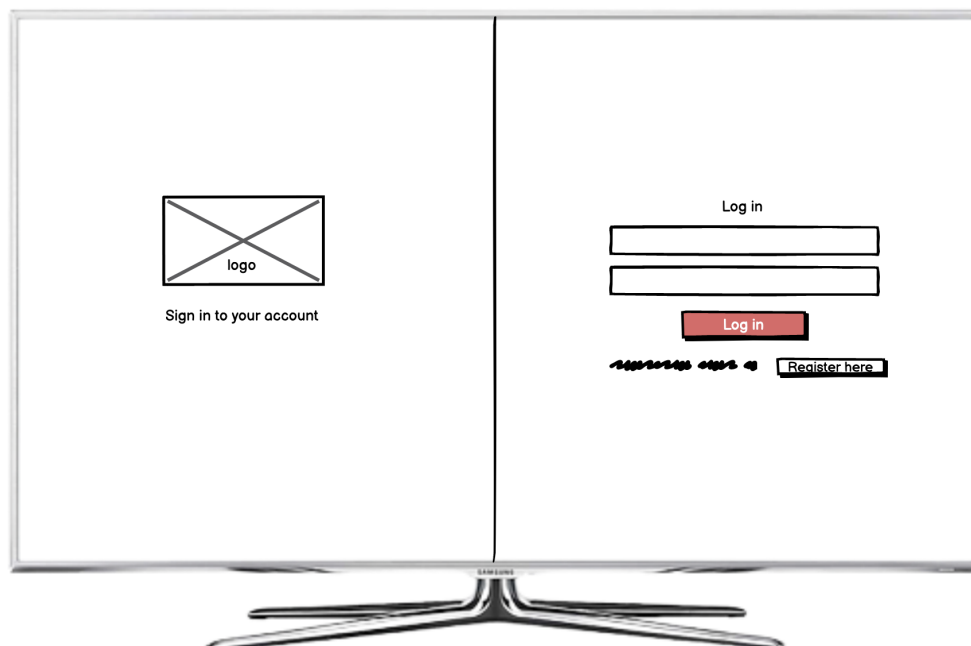
Čo sa týka zmien pre obrazovky vyhľadávania, tak táto obrazovka vyhľadávania videí má podobný vzhľad ako vyhľadávacie rozhranie v minulej aplikácii pre Android TV. Toto riešenie môže pomôcť užívateľom, ktorí majú skúsenosti s predchádzajúcimi verziami alebo aplikáciami pre Android TV.

Toto užívateľské rozhranie je veľmi jednoduché a intuitívne. S týmto rozhraním sa užívateľom uľahčuje navigácia na stránke a umožňuje sa im rýchlo a efektívne orientovať vo výsledkoch vyhľadávania.

4.2.6 Obrazovka prihlásenia a registrácie

Ďalšie zmeny sa konkrétne týkajú obrazovky, kde sa používateľ môže prihlásiť. Bolo presunuté tlačidlo na registráciu, pretože toto tlačidlo iba ukazovalo informáciu, že je potrebné sa zaregistrovať na oficiálnych stránkach. Plugin S2meber to neumožňuje, takže bolo rozhodnuté presunúť túto informáciu rovno na obrazovku. Tlačidlo, ktoré presmeruje užívateľa na stránku, kde sa môže registrovať, bolo presunuté do pravej časti pod prihlasovacie polia, ako je dobrým zvykom pri klasických stránkach tohto typu.

Druhou zmenou je, že tlačidlo na prihlásenie sa môže teraz odstrániť a namiesto neho sa priamo pridávajú vstupy na užívateľské meno a heslo spolu s tlačidlom na potvrdenie, že používateľské údaje už boli zadané. To spĺňa aj odporúčanie Samsungu, ktoré hovorí, že minimalizovať počet prechodov na obrazovky je dobrou praxou. (viz. 3.3.4) Pri Android TV, bolo potrebné urobiť jeden prechod navyše.



Obrázek 4.10. Návrh užívateľského rozhrania obrazovky prihlásovania a registrácie.

4.2.7 Úvodná obrazovka

Jedným z dôvodov, prečo aplikácie pridávajú svoje logá alebo úvodné animácie, by mohlo byť to, že tým vytvárajú rozpoznateľnosť značky alebo to môže mať aj efekt zabavenia užívateľa kým sa načítavajú dáta v aplikácii.

Animácia načítania má užívateľa informovať o dobe čakania, dôvode a o stave prebiehajúcich procesov, ako je načítanie webovej stránky, odoslanie formulára alebo ukladanie aktualizácií. To môže poskytnúť vynikajúce rozptýlenie pozornosti, aby sa zabránilo netrpezlivosti na dlhé čakanie. Navyše je to skvelý čas na zdôraznenie značky a vyjadrenie hlasu spoločnosti. [12]

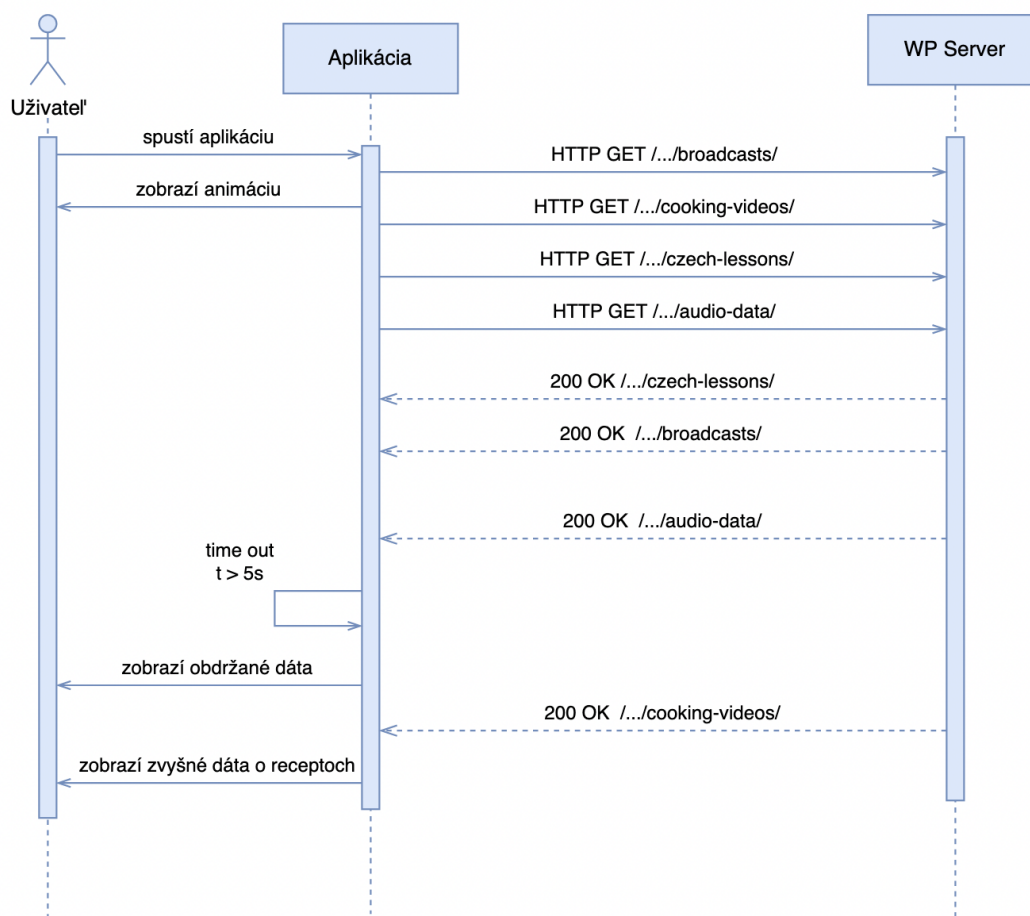
Indikátor priebehu je prvok grafického užívateľského rozhrania, ktoré je určené na informovanie užívateľa, že prebieha určitá operácia. [12]

Ludia majú 3 hlavné časové limity, 0,1 sekundy predstavuje okamžitý pocit, 0,1–1 sekunda umožňuje používateľom uvedomiť si oneskorenie, zatiaľ čo 1–10 sekúnd je potrebné poskytnúť užívateľom spätnú väzbu, aby sa znížila netrpezlivosť. [12]

Aj kvôli týmto dôvodom bolo rozhodnuté pridať na úvodnú obrazovku animáciu alebo obrázok pred načítaním dát. Použitie animácie môže udržať užívateľa zainteresovaného a zapojeného, kým sa dáta načítavajú. Vzhľadom na to, že načítanie dát môže byť pre niektorých užívateľov frustrujúce a môže to trvať dlhšie ako očakávali, animácia môže zabrániť tomuto problému a udržať ich pozornosť na obrazovke. V aplikácii ale bude nastavený limit, ak sa všetky dáta nenačítajú do určitého časového limitu, tak užívateľovi budú zobrazené aspoň tie dáta, ktoré už načítané boli a zvyšné dáta sa načítajú asynchrónne. Zvyšuje sa tým šanca, že užívateľ po zmiznutí animácie uvidí celú aplikáciu načítanú a bude to naňho pôsobiť pozitívne a vytvárať dojem profesionality a kvality aplikácie. Správne zvolená animácia môže tiež poskytnúť užitočné informácie o aplikácii a zvyšovať jej užitočnosť. Výber animácie bude presunutý na zadávateľa tejto práce.

Nie len Samsung odporúča dávať užívateľom spätnú väzbu (viz. 3.3.4), ale je to trendom moderných aplikácií už niekoľko rokov a to z pádných dôvodov.

Nasledujúci sekvenčný diagram zobrazuje proces zobrazenia úvodnej obrazovky a spomínanej animácie.



Obrázek 4.11. Sekvenčný diagram zobrazujúci proces vykreslenia úvodnej obrazovky.

Kapitola 5

Použité technológie

Pri vývoji takmer všetkých softvérových produktov je veľmi dôležité si správne zvoliť technológie, ktoré umožnia programátorom splniť požiadavky zákazníka a vytvoriť efektívny a funkčný produkt. V tomto projekte boli použitých niekoľko súčasne dostupných technológií, ktoré boli zvolené na základe požiadaviek projektu a autorových skúseností.

Jednou z technológií, ktorá bola využitá, je Tizen Studio. Tento vývojový nástroj umožnil vytvoriť aplikáciu pre operačný systém Tizen, ktorý beží na rôznych typoch zariadeniach, ako sú napríklad televízory alebo chytré hodinky. S jeho pomocou bolo možné jednoducho vytvoriť užívateľské rozhranie a implementovať potrebné funkcie. Jeho nástroje tiež umožnili vyskúšať a otestovať aplikáciu v simulovanom alebo emulovanom prostredí, čo zvýšilo kvalitu a spoľahlivosť aplikácie a znížilo potrebný čas na testovanie na skutočnom zariadení.

Okrem Tizen štúdia boli použité aj základné webové technológie, ako sú HTML, CSS a JavaScript. Tieto technológie boli použité na tvorbu webovej časti projektu a na implementáciu funkcií, ktoré nie sú špecifické pre Tizen OS. Značkovací jazyk HTML bol využitý na vytvorenie štruktúry webovej stránky, CSS na jej vizuálny dizajn a JavaScript na interakciu s užívateľom. Pre zjednodušenie vývoja bola použitá aj knižnica jQuery, ktorá umožnila jednoduchšie manipulovať s DOM objektmi a vykonávať rôzne akcie pomocou jednoduchých funkcií. Zlepšiť užívateľskú prívetivosť pomohol aj framework CAPH 3.0, ktorý ponúka množstvo výhod a funkcií pri vývoji aplikácií pre inteligentné televízory.

V neposlednom rade bolo využité aj WordPress web API, ktoré umožnilo získavať dáta z WordPress stránky spoločnosti CATVUSA a zobrazovať ich v aplikácii. Toto API poskytlo jednoduchú a efektívnu cestu, ako integrovať WordPress do tohto projektu a umožniť užívateľom rýchly prístup k obsahu, ktorý publikuje spoločnosť CATVUSA. Podobne vďaka použitiu JSON vo WordPress web API bolo možné jednoducho prenášať dáta v rôznych formátoch, ako napríklad text, obrázky alebo videá.

V tejto kapitole bude ponúknutý bližší pohľad na tieto technológie a ich využitie v tomto projekte.

5.1 Hypertext Markup Language 5 (HTML5)

HTML5 je konečná verzia HTML, čo je hypertextový značkovací jazyk používaný pri vytváraní a štruktúrovaní obsahu, ktorý sa má zobrazovať na webových stránkach. Tí, ktorí ho používajú, môžu popri ňom využívať aj kódovanie, ako sú kaskádové štýly (CSS) a skriptovacie jazyky, ako je Javascript. [13]

HTML5 sa bežne považuje za piatu verziu alebo vydanie jazyka HTML (Hypertext Markup Language), čo je štandardizovaný popisný jazyk, ktorý špecifikuje spôsob štruktúry webových stránok. Napriek tomuto použitiu sa výraz HTML5 v súčasnosti považuje skôr za módný výraz než za verziu, ktorá voľne popisuje súbor moderných webových technológií vrátane HTML. [13]

HTML5 poskytuje spoločné rozhranie, ktoré uľahčuje vykresľovanie rovnakej webovej stránky v rôznych prehliadačoch (hoci medzi prehliadačmi stále môžu existovať nezrovnalosti a funkčné rozdiely).

Webová stránka HTML je dokument s obvyčajným textom, ktorý sa skladá z prvkov, ktoré definujú štruktúru dokumentu. Každý prvok je obklopený zodpovedajúcimi otvorenými a zatvorenými značkami, pričom každá značka je uzavretá v lomených zátvorkách. Páry značiek zvyčajne zahŕňajú nejaký typ textu alebo jeden alebo viacero ďalších prvkov, hoci niektoré páry značiek neobsahujú ani jeden. Prvky môžu byť tiež rozšírené o atribúty, ktoré poskytujú dodatočné informácie alebo pokyny. Webová stránka môže napríklad obsahovať prvok, ktorý obsahuje vložený prvok. [13]

5.2 Cascading Style Sheets (CSS)

Cascading Style Sheets (CSS) je jazyk šablóny štýlov, ktorý sa používa na popis prezentácie dokumentu napísaného v HTML alebo XML (vrátane dialektov XML, ako sú SVG, MathML alebo XHTML). CSS popisuje, ako by mali byť prvky vykreslené na obrazovke. CSS patrí medzi základné jazyky otvoreného webu a je štandardizované vo webových prehliadačoch podľa špecifikácií W3C. [14]

CSS sa používa na štýl a rozloženie webových stránok ako napríklad na zmenu písma, farby, veľkosti a medzier publikovaného obsahu, jeho rozdelenie do viacerých stĺpcov alebo pridanie animácií a iných dekoratívnych prvkov. Tieto vlastnosti sa aplikujú na jednotlivé elementy HTML kódu pomocou selektorov. [14]

5.2.1 JavaScript

JavaScript (JS) je interpretovaný alebo načas kompilovaný programovací jazyk s prvotriednymi funkciami. Hoci je najznámejší ako skriptovací jazyk pre webové stránky, používa ho aj mnoho prostredí bez prehliadača, ako napríklad Node.js, Apache CouchDB a Adobe Acrobat. JavaScript je dynamický jazyk, ktorý podporuje objektovo orientované, imperatívne a deklaratívne (napr. funkčné programovanie) štýly. [15]

Dynamické možnosti JavaScriptu zahŕňajú vytváranie objektov za behu, zoznamy parametrov premenných, premenné funkcie, vytváranie dynamických skriptov a obnova zdrojového kódu. [15]

V projekte bola asi najviac využívaná knižnica jQuery a to nie len pre manipuláciu s DOM objektami, ale aj na pri používaní frameworku CAPH, ktorý využíva túto knižnicu na svoju implementáciu. [16]

jQuery je knižnica JavaScriptu, ktorá uľahčuje manipuláciu s dokumentom HTML, správu animácií, udalostí a komunikáciu s webovými servermi pomocou AJAX. Jednou z výhod použitia jQuery je jednoduchá syntax a schopnosť zlepšiť výkon stránky. To robí z jQuery populárnu knižnicu na tvorbu interaktívnych webových stránok. [17]

Lodash je JavaScriptová knižnica, ktorá pomáha pri práci s polami, reťazcami, objektmi, číslami atď. Poskytuje rôzne vstavané funkcie pre kolekcie, polia, na manipuláciu s objektmi a ďalšie pomocné metódy, ktoré je možné použiť priamo namiesto toho, aby ich bolo potrebné písať od začiatku. Celkovo teda uľahčuje iteráciu polí, reťazcov a objektov. Jeho modulárne metódy umožňujú jednoduchšie vytváranie kompozitných funkcií. [18]

5.3 Tizen

Tizen je operačný systém založený na linuxovom jadre a softvérovej knižnici GNU C Library. Je zameraný na vstavané systémy, vrátane smartfónov, tabletov, automobilo-

vých zariadení, inteligentných televízií, fotoaparátov a laptopov. Jeho licenčný model zahŕňa softvér, ktorý používa rôzne licencie a proprietárny vývojový kit (SDK). To má zabezpečiť hladké fungovanie Tizenu na rôznych platformách a zariadeniach. [19]

Projekt Tizen je prevádzkovaný Linux Foundation a spravovaný Technical Steering Group (TSG), v ktorej sú zástupcovia Samsungu, Intelu a ďalších. [19]

Je užívateľsky interaktívny a na služby orientovaný open source projekt, ktorý umožňuje vytvárať aplikácie bohaté na funkcie pre viacero kategórií zariadení. Vývojárom sa ponúka možnosť začať s vývojom vlastných aplikácií Tizen pre nositeľne zariadenia (napr. Inteligentné hodinky), pre televíziu alebo pre mobilný telefón. [19]

■ 5.3.1 Typy aplikácií

Platforma Tizen podporuje tri primárne typy aplikácií:

- **Aplikácia Tizen .NET** predstavuje nový spôsob vývoja aplikácií pre operačný systém Tizen, ktorý beží na 50 miliónoch zariadení Samsung, vrátane televízorov, nositeľných zariadení, mobilných zariadení a mnohých ďalších. Existujúce frameworky Tizen sú založené buď na jazyku C alebo na HTML5 s menším počtom funkcií a nižším výkonom ako riešenie založené na jazyku C. [20]

S Tizen .NET je možné používať programovací jazyk C# a štandardy Common Language Infrastructure a využívať výhody behového prostredia pre rýchlejší vývoj aplikácií a podobne aj efektívne a bezpečné spúšťanie kódu. [20]

- **Webová aplikácia** je v podstate webová stránka uložená v zariadení a vytvorená pomocou webových natívnych jazykov, ako sú HTML5 (HyperText Markup Language 5), CSS (Cascading Style Sheet) a JavaScript. Webovú aplikáciu využíva Tizen Web Framework na interakciu s natívnymi podsystémami. Webové aplikácie využívajú Web API, čo je štandardná štruktúra projektu webovej aplikácie so základnými prvkami. Webové rozhranie API je navrhnuté tak, aby umožnilo jednoducho vytvárať aplikácie pomocou webových natívnych jazykov. [21]

Program vytvorený pomocou webového rozhrania API je usporiadaný podobne ako štandardná webová stránka. Má *index.html* súbor, ktorý slúži ako koreňový a samostatné oddelené adresáre pre zdroje, ako sú JavaScript súbory, CSS súbory, obrázky a rovnako aj zvukové zdroje. Tento prístup robí vývoj webových aplikácií v Tizen mimoriadne intuitívnym pre vývojárov so skúsenosťami vo vývoji webu a uľahčuje rýchle písanie jednoduchých aplikácií pomocou jazykov na vysokej úrovni. [21]

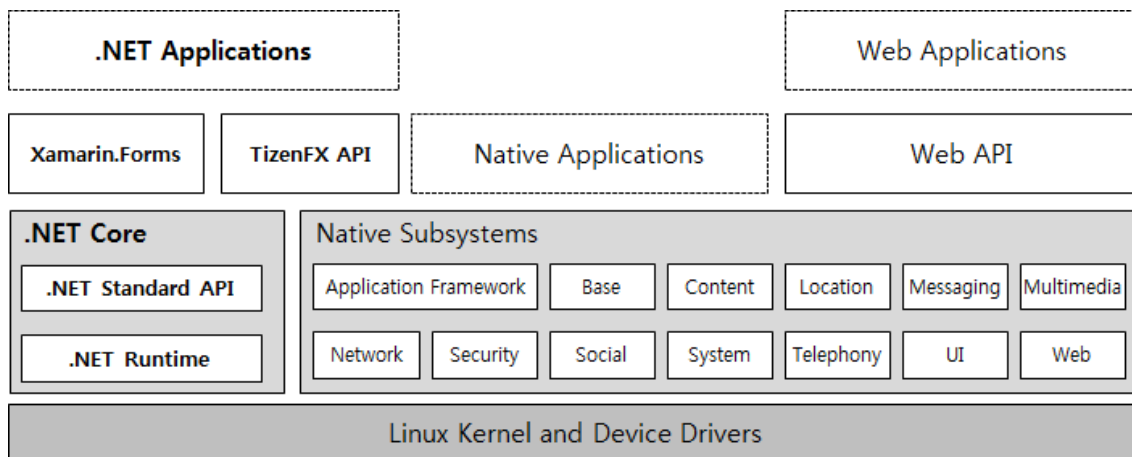
- **Natívna aplikácia** je vyvinutá pomocou jazyka C a okrem pokročilejších nastavení systému a funkcií má prístup k pokročilejším funkciám špecifickým pre zariadenie, ako je fotoaparát, GPS a podobne. [22]

Natívne aplikácie používajú Native API, ktoré poskytuje všetky výhody správy pamäte a výkonu, ktoré prichádzajú s vytváraním aplikácií pre Linux v C. [22]

Natívne API je mimoriadne užitočné, pretože obsahuje desiatky modulov API, ktoré pokrývajú širokú škálu možností. Poskytuje množstvo rozhraní pre väčšinu hardvéru, ktorý sa nachádza v moderných mobilných a nositeľných zariadeniach, a to v prostredí, ktoré je prispôbené pre obmedzené zdroje. [22]

Platforma Tizen tiež umožňuje vyvinúť hybridný balík aplikácií, kde sú natívne a webové aplikácie zabalené spolu, aby vytvorili výkonnejšie aplikácie. Platforma Tizen zaisťuje, že všetky aplikácie Tizen budú mať konzistentný vzhľad a dojem, bez ohľadu na to, či sa na ich vytvorenie používa natívny alebo webový rámec. [22]

Nasledujúci obrázok znázorňuje model architektúry Tizen podporujúci tri typy aplikácií.



Obrázek 5.1. Model architektúry Tizen podporujúci tri typy aplikácií. [23]

5.4 Tizen studio

Tizen Studio je oficiálne integrované vývojové prostredie používané pre tvorbu webových aplikácií a natívnych aplikácií pre Tizen. Tizen Studio je komplexná sada nástrojov na vývoj natívnych a webových aplikácií Tizen. Pozostáva z vyvojového prostredia, emulátora, toolchainu, vzorového kódu a dokumentácie. Tizen Studio beží na Windowse, Ubuntu, alebo aj na macOS. Aplikácie Tizen je možné vyvíjať bez spoliehania sa na oficiálne Tizen Studio, pokiaľ aplikácia spĺňa určité pravidlá. [24]

Ak niekto uprednostňuje rozhranie príkazového riadka pred grafickým oknom Tizen Studio, počas celého procesu vývoja umožňuje použiť aj nástroj Rozhranie príkazového riadka (CLI). Rozšírenie Tizen Studio TV podporuje všetky bežné príkazy CLI Tizen Studio, ale niektoré príkazy sa pri práci s televíznymi aplikáciami používajú inak ako s aplikáciami pre mobilné zariadenia alebo nositeľné aplikácie.

Tizen Studio poskytuje rôzne nástroje, ktoré pomáhajú s rôznymi fázami procesu vývoja webových aplikácií:

- **Riadenie projektu** je možné pomocou sprievodcu projektom, ktorý dokáže vytvárať nové aplikačné projekty a spravovať existujúce aplikácie v zobrazení Project Explorer v Tizen Studio. Môže tiež spravovať pripojené zariadenia a registrovať certifikáty pre svoje aplikácie, aby ich bolo možné publikovať v obchodoch s aplikáciami.
- **Písanie a úprava kódu** teda súborov HTML, JS a CSS je možné upravovať v konkrétnych editoroch a zobraziť ukážku toho, ako meniaci sa kód ovplyvní obrazovku aplikácie. Na úpravu aplikácie počas jej spustenia sa môže použiť aj funkcia úprav naživo.

Na urýchlenie písania a overovania kódu je vhodné použiť nástroje na pomoc s obsahom, kontrolu privilégií a čistenie kódu. Pomocou nástroja REST Viewer sa dá emulovať interakcia s webovou službou. Okrem toho je možné upraviť rôzne preferencie Tizen Studia, aby sa vývojári uistili, že prostredie na úpravu kódu vyhovuje ich potrebám.

- **Konfigurácia aplikácie** je veľmi jednoduchá a umožňuje rôzne funkcie a vlastnosti aplikácie spravovať prostredníctvom konfiguračného súboru aplikácie. Okrem toho vývojári môžu aplikáciu lokalizovať tak, aby používala rôzne jazyky a zdroje.
- **Ladenie aplikácie** je tiež možné prostredníctvom rôznych nástrojov. Aplikáciu je možné ladiť a optimalizovať pomocou nástrojov ako Dynamic Analyzer, Web Inspector a JavaScript Analyzer. Tizen Studio poskytuje rôzne zobrazenia (napr. log zobrazenie) navrhnuté tak, aby pomáhali vývojárom rýchlo identifikovať a opraviť chyby.
- **Spustenie a testovanie aplikácie** je možné pomocou webového simulátora, ktorý umožní otestovať, ako sa aplikácia správa, a dokonca aj odosielať udalosti a správy. Je možné tiež použiť jednotkové testovanie, kde sa dajú vytvárať testovacie projekty a overiť tak spoľahlivosť kódu.

[25]

■ 5.4.1 TV Simulator

Samsung TV Simulator je jednoduchý nástroj, ktorý poskytuje množstvo nastavení a funkcií na testovanie webových aplikácií Samsung TV na počítači. Samsung TV Simulator simuluje prevádzkové prostredie webových aplikácií Samsung TV. Vyvíjané aplikácie si je možné otestovať na simulátore a ušetriť tak značný čas na vývoj a testovanie. Simulátor je užitočný, keď je potrebný rýchly vývoj prototypu. Je vhodný na vývoj užívateľského rozhrania alebo testovanie funkcií aplikácií. [26]

Simulátor podporuje štandardné webové technológie, ako sú HTML5, JS a CSS. Simuluje tiež niektoré rozhrania API televíznych produktov, ale rozhrania API, ktoré sú prísne závislé od hardvéru televízora alebo základných modulov špecifických pre televízor, nie sú podporované. [26]

V počiatočnej fáze tejto diplomovej práce bol značne využívaný Samsung TV Simulator. Hlavne z dôvodu ľahšej dostupnosti TV zariadenia, ale aj úspory času. Nie je však vhodné vynechávať testovanie priamo na zariadení, niektoré časti aplikácie môžu na simulátore pôsobiť inak. Vyvíjaná aplikácia nie je však veľmi závislá od hardvéru a nepoužíva ani špecifické TV API, v takýchto prípadoch by sa testovanie na simulátore vysoko nedoporučovalo.

Samsung TV Simulator poskytuje možnosť otestovať, ako sa aplikácia zobrazuje v rôznych mierkach obrazovky alebo otestovať na simulovanej obrazovke ako aplikácia reaguje v orientácii na výšku. Čo oproti klasickej televízii ponúka výhodu, kedy je možné otestovať, či je aplikácia responzívna. Simulátor dokonca ponúka aj možnosť vyskúšať viaceré typy diaľkových ovládačov alebo otestovať rôzne konfigurácie televízora.

■ 5.4.2 TV Emulator

Emulátor TV je nástroj, ktorý na testovacie účely poskytuje prostredie podobné skutočnému zariadeniu. Ak nie je k dispozícii skutočné zariadenie, existuje možnosť otestovať aplikáciu na virtuálnom zariadení s podobnými funkciami. [27]

Televízny emulátor je virtuálny stroj, ktorý sa veľmi podobá skutočnému televízornému softvéru. Znižuje nepohodlie pri testovaní aplikácie na skutočnom zariadení tým, že poskytuje prostredie podobné skutočnému zariadeniu. Emulátor poskytuje skutočný hardvér podobný ovládacím prvkom skutočného zariadenia a je možné skontrolovať funkcie aplikácie. Emulátor je založený na open source projekte QEMU. [27]

Emulátor je však pomalší ako skutočný televízor, nepodporuje všetky rozhrania API a ani inteligentnú interakciu ako sú hlas, gestá alebo dlhšie stlačenie. Počas vývoja boli zistené problémy so zariadením MacBook, kedy sa emulátor neočakávane ukončil.

Ak bola aplikácia otestovaná na emulátore, tak je pripravená na nahranie a testovanie aj v televízore. Je potrebné povoliť režim vývojára na televízore a pripojiť televízor k súprave SDK. Aj keď je možné väčšinu testovania aplikácie dokončiť na emulátore, dôrazne sa odporúča otestovať aplikáciu aj na cieľovom zariadení. Emulátor a televízne prostredie sú veľmi podobné, existujú však určité rozdiely na úrovni hardvéru. Aplikáciu môže vývojár spustiť na cieľovom zariadení v normálnom režime alebo režime ladenia [28]

■ 5.4.3 Webový inšpektor

Webový inšpektor je dobre známa, hlavne vývojárom webových stránok, softvérová pomôcka alebo funkcia vo webovom prehliadači, ktorá ľuďom umožňuje zobrazit kód HTML na webovej stránke. Vo väčšine prípadov je možné údaje zmeniť a stránka sa vykreslí inak, akonáhle sa však stránka obnoví, zmeny sa stratia.

Rozšírenie Tizen Studio TV poskytuje Web Inspector ako nástroj na ladenie. Webový inšpektor pomáha vývojárom ladit a optimalizovať ich webové aplikácie. Poskytuje prostredie na ladenie, kde je možné kontrolovať CSS a rozloženie stránok HTML, monitorovať využitie pamäte, siete, pozastaviť vykonávanie JavaScriptu a sledovať hodnoty premenných tak, ako sú definované. Webový inšpektor je založený na Webkit Web Inspector a Google Chrome Developer Tools. [29]

Všetky televízory Samsung od roku 2015 podporujú Webový inšpektor a je plne podporovaný aj na emulátore TV. [29]

Počas tohto projektu bol webový inšpektor veľmi využívaná pomôcka, ktorá poskytuje prostredie na ladenie, v ktorom je možné napríklad kontrolovať vykonávanie JavaScriptu a monitorovať rôzne informácie o aplikácii počas jej spustenia. Okrem toho boli využité aj nasledujúce funkcie, ktoré sú usporiadané od kariet:

- Karta s elementami umožňuje zobrazit komponenty webovej stránky (štýl CSS, sledovanie udalostí a ďalších vlastností).
- Karta s konzolou umožňuje skontrolovať činnosť JavaScriptu a interagovať so stránkou programovo, zobrazuje tiež všetky chyby alebo upozornenia na stránke.
- Karta so zdrojmi umožňuje skontrolovať zdrojovú stránku JavaScriptu a odladiť kód JavaScript. Táto karta podporuje výrazy sledovania, zásobník hovorov, premenné rozsahu a operáciu bodu zlomu. Okrem toho podporuje základné operácie ladenia: pokračovať, prejsť, vstúpiť a vystúpiť.
- Karta sieť umožňuje kontrolovať zdroje stiahnuté cez sieť.
- Karta časová os umožňuje vykonávať pokročilé analýzy časovania a rýchlosti aplikácie. Je možné sledovať, ako dlho prehliadaču trvá spracovanie udalostí DOM a vykresľovanie elementov.
- Karta profily umožňuje kontrolovať analýzy výkonnosti JavaScriptu. Je možné skontrolovať CPU a pamäť.

[30]

■ 5.5 CAPH

CAPH je pomocný framework pre užívateľské rozhrania webových aplikácií navrhnutý tak, aby zjednodušoval a zrýchľoval vývoj aplikácií určené pre inteligentné televízie.

Tvorcovia uvádzajú, že CAPH je vyvinutý konkrétne pre aplikácie určené Samsung Tizen TV. Webová aplikácia by teda mala mať dobrý výkon. [16]

Tretie verzia tohoto frameworku (CAPH 3.0) bola vydaná v októbri v roku 2015. Oproti predchádzajúcim verziám nepoužíva rovnaký základný engine. Namiesto toho ho možno implementovať pomocou technológie jQuery alebo AngularJS, čo umožní jednoducho pridať podporované komponenty do vyvíjanej aplikácie. Všetky komponenty sú voľne upraviteľné.[16]

■ 5.5.1 Základné podporované komponenty

- **Navigácia pomocou klávesov** pomôže užívateľom ľahko zvládnuť zameranie prvku užívateľského rozhrania pomocou šípok na diaľkovom ovládači.
- **Rolovateľný zoznam** je komponent užívateľského rozhrania s vysokovýkonnou animáciou posúvania. Je veľmi praktický pre užívateľov a preto je aj populárny v aplikáciách pre filmy a seriály ako napríklad Netflix. Vývojári si ho môžu veľmi jednoducho prispôbiť pre svoje vlastné užívateľské rozhranie. Podporuje spracovanie veľkých dát.
- **Dialógové okno** je vyskakovacie okno na interakciu s užívateľom. Do dialógového okna je možné umiestniť ďalšie komponenty.
- **Rôzne druhy tlačidiel**, sa radia medzi veľmi základné komponenty, ktoré už pozná takmer každý užívateľ (napr. klasické tlačidlo, zaškrťavacie políčko alebo prepínač, ktorý má dva stavy).
- **Kontextové menu**, ktoré sa v prípade potreby zobrazí na určenom mieste.
- **Rozbalovacia ponuka** obsahuje zoznam možností, z ktorých je možné znovu vybrať.

[16]

Všetky tieto komponenty spolupracujú, vďaka CAPH, spolu s navigáciou, sú teda ľahko zamerateľné a tak umožňujú užívateľom jednoducho používať aplikácie pre inteligentné televízie.

■ 5.6 Wordpress REST API

Názov WordPress REST API vznikol z kombinácie názvu WordPress a dvoch koncepcií ako je REST (Representational State Transfer) a API (Application Programming Interface). V tejto podkapitole bude ponúknutý bližší pohľad na toto rozhranie.

■ 5.6.1 WordPress

WordPress je jeden z najjednoduchších a najpopulárnejších systémov na správu obsahu (angl. Content Management System). To znamená, že niekto poskytuje obsah, zatiaľ čo platforma robí všetko kódovanie a organizáciu miesto neho. Ak užívateľ hľadá správneho tvorca webu, aby mohol začať publikovať svoj obsah na web bez znalosti programovania, WordPress sa často označujú ako tá správna cesta.

Na vylepšovaní tejto platformy každý deň pracujú stovky dobrovoľníkov po celom svete. Znamená to tiež, že sa dá bezplatne používať bez licenčného poplatku. [31]

■ 5.6.2 API

API (Application Programming Interface) je rozhranie, ktoré spája dve rôzne počítačové programy alebo aplikácie za účelom komunikácie. Je to typ softvérového rozhrania, ktoré ponúka službu iným častiam softvéru. [32]. Na internete je jeho použitie veľmi rozšírené, ak napríklad je príspevok alebo video zdieľané na Youtube cez aplikáciu tretej strany, využíva sa Youtube API.

Vďaka tomu, že API umožňuje komunikáciu s ostatnými technológiami, je možné získavať obsah takmer odkiaľkoľvek. WordPress platformu vďaka tomu môžu programátori použiť ako redakčný systém pre vlastný obsah, ktorý je uložený mimo WordPress, alebo naopak obsah z WordPress webu je možné využiť v inej aplikácii

■ 5.6.3 REST

REST je skratka pre Representational State Transfer a odkazuje na daný štýl budovania API. Takmer všetky hlavné webové služby, ako sú Google, Facebook a Twitter, sa spoliehajú na REST a to pravdepodobne preto, že REST je založený na HTTP (čo je zhodou okolností protokol, ktorý poháňa takmer všetky internetové pripojenia). REST je navyše ľahký a flexibilný a ľahko zvládne veľké objemy aktivít. REST sám o sebe nie je novým trendom a na webe sa používa na prepojenie služieb už dosť dlhú dobu. [33]

Pre užívateľov WordPressu využitie sily REST API znamená, že aplikácie môžu interagovať s množstvom služieb priamo z WordPress, s pomocou REST API. [33]

REST je teda architektonická paradigma pre webové služby a služby, ktoré využívajú takúto architektúru, sú známe ako RESTful služby.[33]

Základnou myšlienkou REST je, že namiesto spoliehania sa na zložité webové služby ako sú SOAP alebo XML-RPC, sa na vytváranie pripojení používa jednoduchý protokol HTTP. Preto všetky aplikácie RESTful využívajú požiadavky HTTP na spracovanie všetkých štyroch operácií CRUD, menovite na vytváranie, čítanie, aktualizáciu a mazanie. Vďaka tomu je REST mimoriadne všestranný a každý môže zaviesť svoju vlastnú verziu REST pomocou programovacieho jazyka podľa vlastného výberu, ako je JavaScript, Perl alebo PHP. Navyše je REST plne nezávislý od platformy, takže ho je možné používať v prípadoch, kde server môže byť Linux, ale klient môže používať Windows atď. Keďže požiadavka RESTful je založená na štandardoch a je aj jazykovo nezávislá.[33]

■ 5.6.4 JSON

JSON je skratka pre JavaScript Object Notation. Ako už názov napovedá, ide o formu formátu výmeny údajov, ktorý je založený na JavaScripte. S pribúdajúcimi knižnicami a službami JavaScriptu stúpa popularita JSON na webe. Najlepšie na JSON je, že je vhodný pre stroje aj pre ľudí, pokiaľ ide o čítanie a porozumenie. Vývojári ho môžu čítať aj písať rovnako, ako by pracovali s akýmkoľvek iným programovacím jazykom, a počítače ho môžu tiež ľahko analyzovať a spracovať. V skutočnosti mnoho populárnych programovacích jazykov ponúka svoje vlastné funkcie, ktoré transformujú výstup do JSON a späť. Vďaka tomu je JSON ideálny pre interakciu medzi platformami. Aplikácia kódovaná v jednom programovacom jazyku a aplikácia kódovaná v druhom programovacom jazyku, môžu spolu jednoducho interagovať konvertovaním svojich dátových štruktúr na JSON a späť. Táto funkcia JSON z neho urobila univerzálny konektor na webe. [33]

Pre užívateľov WordPress môže byť JSON tiež použitý na nahradenie takmer zastaraného štandardu XML-RPC.[33]

■ 5.6.5 REST API

Po krátkom predstavení, čo znamenajú pojmy API, REST a JSON, je vhodné sa vrátiť na pojem REST API.

WordPress REST API poskytuje aplikáciám rozhranie, ktoré umožňuje interakciu s webovou stránkou WordPress odosielaním a prijímaním údajov v podobe objektov vo formáte JSON (JavaScript Object Notation). [31]

Jednoducho povedané, REST API umožňuje prepojiť WordPress s ostatnými aplikáciami bez ohľadu na to, akými jazykmi sú písané. Zatiaľ čo WordPress je postavený na jazyku PHP, REST API sa stáva akýmsi mostom medzi PHP jadrom WordPressu a webovými aplikáciami stavanými na JavaScripte, rovnako ako natívnymi mobilnými a desktopovými aplikáciami.

Je zrejmé, že REST API je prelomová a šikovná inovácia, ktorá má schopnosť zmeniť spôsob, akým komunikujeme s WordPress. Keďže spoločnosť CATVUSA momentálne produkuje svoj obsah cez webové stránky, ktoré sú tvorené vo WordPress, ponúka sa práve využiť REST API, ktoré môže pomôcť k rýchlejšiemu vývoju.

Kapitola 6

Implementácia

Kapitola sa zameriava na podrobný popis jednotlivých častí aplikácie, ako je hlavná obrazovka, detail videa, prehrávač rádia a ďalšie.

Okrem popisu týchto častí aplikácie sa kapitola tiež zameriava na komunikáciu so serverom pomocou WordPress REST API. Táto časť popisuje koncové body a globálne parametre, ktoré boli použité na získanie dát zo serveru. Taktiež sa venuje realizácii požiadaviek pomocou technológie AJAX.

Ďalšou časťou kapitoly je opis realizácie plochej navigácie a rolovateľných zoznamov s využitím knižnice CAPH a jQuery. Budú popísané prvky implementácie a využitie ich funkcií a vlastností pre zlepšenie užívateľského rozhrania.

Na konci kapitoly bude predstavené, ako je možné pomocou S2Member Pro API overiť prihlasovacie údaje užívateľa. Bude aj ukázané, ako odoslať HTTP POST požiadavku s použitím API kľúča a požadovaných parametrov.

6.1 Komunikácia so serverom

Pri implementácii funkcionality na získanie jednotlivých videí a ďalších informácií potrebných pre aplikáciu pre smart TV bolo rozhodnuté použiť WordPress REST API. Toto API umožňuje komunikáciu so serverom pomocou HTTP požiadaviek a návrat dát vo formáte JSON. (viz. 5.6)

Táto aplikácia obsahuje niekoľko kategórií videí, ktoré sú k dispozícii pre užívateľov. Každá kategória reprezentuje jeden požiadavok na server a získava informácie o videách v danej kategórii. Kategórie zahŕňajú vysielanie o rôznych regiónoch, videá o českých jedlách a videá pre učenie českého jazyka.

Použitie WordPress REST API umožnilo jednoducho získať potrebné dáta zo servera. Použité boli HTTP požiadavky na získanie dát o videách a odpoveďou boli dáta vo formáte JSON. Tieto dáta boli následne použité na zobrazenie informácií o videách, alebo odkaz na samotné video v aplikácii.

Pri implementácii sa bolo potrebné zaoberať niekoľkými problémami, ako napríklad s tým, ako získať správne dáta pre každú kategóriu videí a tiež tým, ako sa zachovať ak odpoveď nebude prichádzať príliš dlho.

6.1.1 Koncové body

WordPress REST API používa koncové body (angl. endpoint) na poskytovanie rôznych funkcií a možností na získanie alebo aktualizáciu dát. Koncové body sú URL adresy, ktoré poskytujú prístup k rôznym zdrojom dát, vrátane článkov, stránok, kategórií a iných obsahových entít. [34]

Pri použití WordPress REST API je možné použiť koncové body na rôzne účely. Napríklad, ak je potrebné získať všetky články, je možné použiť GET požiadavku na koncový bod `/wp/v2/posts`. Ak ide o konkrétny článok, je možné použiť GET požiadavku na koncový bod `/wp/v2/posts/id`, kde id je identifikátor konkrétneho článku.

Koncové body, ktoré boli použité, poskytujú prístup k rôznym kategóriám na stránkach *catvusa.com* pomocou WordPress REST API. Konkrétne ide o koncové body:

- Tento koncový bod poskytuje prístup k dátam, ktoré sú spojené s vysielaním na *catvusa.com*. V tejto kategórii sa nachádzajú informácie o vysielaní rôznych programov, napríklad informácie o ich názve, dátume vysielania a čase, popisy programov, obrázky a iné.

```
https://www.catvusa.com/wp-json/wp/v2/broadcasts.
```

- Tento koncový bod poskytuje prístup k dátam, ktoré sú spojené s kategóriou o varení. V tejto kategórii sa nachádzajú informácie o rôznych receptoch.

```
https://www.catvusa.com/wp-json/wp/v2/cooking.
```

- Tento koncový bod poskytuje prístup k dátam, ktoré sú spojené s kategóriou o vyučovaní českého jazyka na *catvusa.com*.

```
https://www.catvusa.com/wp-json/wp/v2/class.
```

6.1.2 Globálne parametre

API obsahuje niekoľko globálnych parametrov (tiež nazývaných *meta-parametre*), ktoré riadia spôsob, akým API spracúva požiadavky a odpovede. Tieto parametre fungujú na úrovni nad samotnými zdrojmi a sú dostupné pre všetky zdroje. [35]

Zdroj REST, ako je napríklad príspevok, obsahuje veľké množstvo údajov: základné informácie, ako je obsah, názov a ID autora, ale aj registrované metadáta a polia, informácie o médiách a odkazy na iné zdroje. Aplikácie nemusia potrebovať všetky tieto informácie pri každej žiadosti. [35]

Ak vývojár chce prikázať WordPress, aby vrátil iba podmnožinu polí v odpovedi, môže použiť `__fields` parameter dopytu. Ak napríklad vytvára zobrazenie archívu a potrebuje iba ID, názov, trvalý odkaz, autora a úryvok pre kolekciu príspevkov, pomocou tohto dotazu na polia môže obmedziť odpoveď len na niektoré vlastnosti. [35]

V priebehu vývoja aplikácie bol použitý parameter `__fields` na filtrovanie potrebných údajov v odpovedi, aby sa zvýšil výkon aplikácie a zlepšila sa rýchlosť spracovania požiadavkov. Koncový bod spolu s parametrami by mohol vyzeráť nasledovne:

```
https://www.catvusa.com/wp-json/wp/v2/broadcasts?__fields=id,title,
,content,excerpt
```

- **id** - unikátny identifikátor príspevku
- **title** - názov príspevku
- **content** - obsah príspevku vo formáte HTML, obsahuje blok videí prehrávača WPLYr, z ktorého je možné získať URL adresy videí
- **excerpt** - krátky popis príspevku vo formáte HTML

Väčšina zdrojov obsahuje odkazy na súvisiace zdroje. Príspevok môže napríklad odkazovať na nadradený príspevok alebo na komentáre k príspevku. Aby sa znížil počet požiadaviek HTTP, klienti môžu chcieť získať zdroj, ako aj prepojené zdroje. Parameter `__embed` indikuje serveru, že odpoveď by mala zahŕňať tieto vložené prostriedky. [35]

Odpovede v režime vkladanie (*embed*) budú obsahovať ďalší kľúč `__embedded` vedľa kľúča `__links` obsahujúci prepojené zdroje. Iba odkazy s parametrom `embeddable`, ktoré sú nastavené na `true` budú vložené. [35]

V aplikácií pre Smart TV, bolo potrebné získať obrázky, pomocou režimu vkladania. V opačnom prípade by miesto jedného požiadavku, bolo potrebné použiť stovky. Čo enormne skraca čas na získanie dát od serveru. Odkaz na dáta, v ktorých je odkaz

na obrázkov sa nachádzal pod kľúčom `wp:featuredmedia`. Nový koncový bod spolu s parametrami by mohol vyzeráť nasledovne:

```
https://www.catvusa.com/wp-json/wp/v2/broadcasts?_fields=id,title,content,excerpt,featured_media,_links,_embedded&_embed=wp:featuredmedia
```

- **featured_media** - identifikátor prepojeného zdroja, teda identifikátor obrázku, ktorý je priradený k príspevku
- **_links** - objekt s odkazmi na súvisiace entity
- **_embedded** - obsahuje prepojené zdroje, teda informácie o obrázku, ktorý je priradený k príspevku

6.1.3 Stránkovanie

Stránky WordPress môžu mať veľa obsahu – oveľa viac, ako by si vývojár chcel stiahnuť v jednej žiadosti. Koncové body API predvolene poskytujú obmedzený počet položiek na žiadosť, rovnako ako web WordPress predvolene nastaví 10 príspevkov na stránku v zobrazeniach archívu. [36]

Akákoľvek odpoveď rozhrania API, ktorá obsahuje viacero zdrojov, podporuje niekoľko bežných parametrov dotazu na spracovanie stránkovania prostredníctvom údajov odpovede:

- **page** špecifikuje sa stránku s výsledkami, ktorá sa má vrátiť, napríklad, `page=2` je druhá stránka výsledkov
- **per_page** špecifikuje počet záznamov, ktoré sa majú vrátiť v jednej požiadavke, zadané ako celé číslo od 1 do 100, napríklad, `per_page=1` vráti iba prvý príspevok v kolekcii. [36]

V aplikácii sa používajú koncové body s parametrami `page` a `per_page`. Parameter `per_page` je nastavený na maximum na 100, takže pri dotazovaní na danú kategóriu sa zobrazí maximálne 100 príspevkov. V každej kategórii sa momentálne nachádza 70-80 príspevkov (videí).

V prípade, že sa v budúcnosti počet videí v kategórii zvýši, aplikácia bude postupne zvyšovať parameter `page`. To zabezpečí, že užívatelia budú môcť prehľadávať všetky videá v kategórii bez straty prehľadnosti a efektívnosti použitia aplikácie.

Aby bolo možné určiť, koľko stránok údajov je k dispozícii, rozhranie API vráti dve polia hlavičky s každou odpoveďou:

- **X-WP-Total** - celkový počet záznamov v kolekcii
- **X-WP-TotalPages** - celkový počet strán zahŕňajúci všetky dostupné záznamy

Kontrolou týchto polí hlavičky môžete určiť, koľko ďalších údajov je dostupných v rámci API. [36]

Finálny koncový bod spolu s parametrami bude vyzeráť nasledovne:

```
https://www.catvusa.com/wp-json/wp/v2/broadcasts?_fields=id,title,content,excerpt,featured_media,_links,_embedded&_embed=wp:featuredmedia&per_page=100&page=1
```

6.1.4 Realizácia požiadavkov

Na realizáciu požiadavku aplikácia využíva funkciu `$.ajax` z knižnice jQuery na načítanie dát pomocou technológie **AJAX**.

AJAX umožňuje asynchrónnu aktualizáciu webových stránok výmenou údajov s webovým serverom v zákulisí. To znamená, že je možné aktualizovať časti webovej stránky bez opätovného načítania celej stránky. [37]

Ako bolo spomínané v časti 6.1.3, tak je potrebné zistiť, koľko stránok obsahuje odpoveď serveru. Na začiatku sa zavolá funkcia, ktorá načíta všetky dáta od prvej stránky. Vykoná sa AJAX požiadavka na API a vráti dáta. Tie sa uložia do premennej a z HTTP odpovede sa získa počet stránok `totalPages`. Na základe tohto počtu sa rozhodne, či sa má rekurzívne volať funkcia na načítanie ďalšej stránky alebo sa končí načítavanie dát.

Nakoniec, ak všetky dáta boli úspešne načítané, vytvorí sa zoznam videí na obrazovke so všetkými potrebnými informáciami.

6.2 Navigácia

V kapitole o odporúčaní užívateľského návrhu od spoločnosti Samsung (viz. 3.2) sa spomínalo, že ak je kategória aplikácií a príslušný zoznam obsahu rozdelený do dvoch úrovní, potom sa odporúča použiť plochú navigáciu (kde tlačidlá nahor a nadol prepínajú medzi kategóriami a ľavé a pravé tlačidlo v obsahu v rámci každej kategórie).

Na tento typ navigácie bol použitý pomocný framework CAPH (viz. 5.5). Caph má zabudovanú funkciu pre navigáciu, ktorá umožňuje užívateľom rýchlo sa pohybovať medzi prvkami pomocou diaľkového ovládača. To všetko je možné implementovať pomocou knižnice jQuery.

Knižnica jQuery obsahuje rozširujúci modul jQuery („\$.caph.focus“), ktorý poskytuje funkcie na navigáciu zaostriteľných prvkov pomocou diaľkového ovládača, ktorý je hlavným vstupným zariadením televízora. [38]

Pozostáva z 3 častí. Atribútu `focusable`, ktorý predstavuje zaostriteľné prvky, `nearestFocusableFinder`, ktorý má na starosti vyhľadávanie najbližších zaostriteľných prvkov, a `controller`, ktorý poskytuje správanie súvisiace so zameraním.

Pre jednoduché použitie navigácie, stačí pridať do atribút `focusable` do cieľového HTML elementu.

```
<div> focusable> \ div>
```

Všetky prvky, ktoré sú špecifikované týmto atribútom, majú automaticky tri ďalšie vlastnosti:

- **Hĺbka** sa používa pri presúvaní zaostrenia na iný zaostriteľný prvok, zaostrenie sa zmení iba medzi rovnakou hĺbkou. Ak teda neexistuje prvok s rovnakou hĺbkou zaostrenia, zaostrenie sa nezmení.
- **Skupina** sémanticky oddeľuje zameranú oblasť. Aj keď sa skupina navzájom nerovná, zaostrenie sa môže zmeniť, ak je každá hĺbka rovnaká. Oddelenie skupín je užitočné, keď sa s predchádzajúcou históriou zaostriteľných prvkov každej skupiny volaním rozhrania API „setGroup“ ovládača.
- **Názov** sa používa pri nastavovaní ďalšieho zamerania na konkrétny prvok alebo pri zmene zamerania na iný zaostriteľný prvok. Každý názov by mal mať jedinečnú hodnotu.

[38]

Nasledujúci HTML element vybraný z vyvíjaného kódu ukazuje ako je možné nastaviť prvok v navigácii pre framework CAPH. V aplikácii sa používa rôzna funkcionalita pre získanie zamerania a určenie poradia elementov. Taktiež sa používajú rôzne atribúty, ktoré umožňujú určiť nasledujúci element v danom smere navigácie

a hĺbku daného elementu v rámci hierarchie zmerateľných elementov. Tieto atribúty tiež umožňujú určiť konkrétny názov elementu pre použitie v JavaScript funkciách.

```
<div focusable data-focusable-initial-focus='true'
data-focusable-name='Lessons' data-focusable-depth="0"
data-focusable-next-focus-down='Audio'
data-focusable-next-focus-up='null'
data-focusable-next-focus-right='group:BroadcastGroup'> \ div>
```

- **data-focusable-name='Lessons'** - Ide o definovanie názvu elementu. Teraz je možné zamerať prvok manuálne z iného prvku alebo zamerať prvok pomocou javascriptu.
- **data-focusable-depth='0'** - Tento atribút určuje hĺbku tohto elementu v strome zmerateľných elementov. Hĺbka 0 znamená, že tento element je najvyššou úrovňou a ostatné zmerateľné elementy sú pod ním.
- **data-focusable-initial-focus='true'** - Tento atribút určuje, či by element mal získať zameranie ako predvolený, keď je stránka načítaná.
- **data-focusable-group='BroadcastGroup'** - Ide o definovanie názvu skupiny elementov. Teraz je možné zamerať prvok z danej skupiny.
- **data-focusable-next-focus-down='Broadcasts'** - Tento atribút určuje, ktorý element by mal získať zameranie po stlačení dolnej šípky. V tomto prípade je hodnota nastavená na null, čo znamená, že sa zameranie nezmení.
- **data-focusable-next-focus-right='group:BroadcastGroup'** - Ak by bolo potrebné zamerať skupinu (to znamená zamerať prvok, ktorý bol zameraný ako posledný z danej skupiny), je potrebné nastaviť hodnotu na group:nazov skupiny.

Pri implementácii tejto aplikácie bolo potrebné definovať v kóde aj funkcie, ktoré sa volajú pri rôznych udalostiach, ako je napríklad získanie zameranie na element alebo jeho výber. Tieto funkcie môžu byť využité na naprogramovanie rôznych akcií, ktoré sa majú vykonať pri navigácii v aplikácii pre Smart TV. V prípade tejto aplikácie, ktorá bola vyvíjaná pomocou knižnice CAPH a ovládania zamerania, sa to mohlo využiť na získanie dát o vybranom videu. Napríklad, pri výbere konkrétneho videa sa zavolala funkcia, ktorá získa potrebné dáta o videu z databázy a následne ich zobrazí na obrazovke pre zobrazenie detailov videa. Tieto dáta obsahujú informácie, ako sú názov videa, popis videa, náhľadový obrázok a podobne. Tiež bolo možné využiť funkciu na zmenu vzhľadu zameraného prvku.

Každý zmerateľný prvok vyvolá udalosť, keď je zameraný, rozmazaný alebo vybraný. Takže je možné pripojiť obsluhu udalosti na prijímanie zodpovedajúcich udalostí. [38]

```
$(document).ready(function() {
    $.caph.focus.activate(function(nearestFocusableFinderProvider,
    controllerProvider) {
        controllerProvider.onFocused(function(event, originalEvent) {
            // ďalší kód
        });

        controllerProvider.onBlurred(function(event, originalEvent) {
            // ďalší kód
        });
    });
});
```

```

    });

    controllerProvider.onSelected(function(event, originalEvent) {
        // ďalší kód
    });
});
});

```

Tento kód definuje funkciu, ktorá aktivuje ovládanie navigácie v aplikácii pre Smart TV. Funkcia je definovaná pomocou jQuery syntaxe.

V rámci funkcie sa volá metóda `activate` z knižnice CAPH, ktorá inicializuje ovládanie. V tejto metóde boli definované tri funkcie pre spracovanie rôznych udalostí, ktoré môžu nastať pri navigácii v aplikácii:

- `onFocused` - táto funkcia sa zavolá, keď je element zameraný
- `onBlurred` - táto funkcia sa zavolá, keď element stratí zameranie
- `onSelected` - táto funkcia sa zavolá, keď je vybraný daný element

Každá z týchto funkcií má definované dva parametre - `event` a `originalEvent`, ktoré poskytujú informácie o udalosti. Tieto funkcie môžu byť využité na naprogramovanie rôznych akcií, ktoré sa majú vykonať pri navigácii v aplikácii pre Smart TV.

Nasledujúci kód slúži na nastavenie zamerania na element s identifikátorom `playvideo`. To znamená, že po jeho vykonaní bude tento element označený ako aktívny a užívateľ bude môcť interagovať s ním pomocou diaľkového ovládača. Tento príkaz sa zvyčajne používa v prípade, keď chcete explicitne určiť, na ktorom prvku v aplikácii sa má nachádzať zameranie. Toto môže byť užitočné napríklad v prípade, keď je potrebné, aby sa po otvorení stránky zameranie automaticky presunulo na konkrétny prvok v aplikácii.

```
$.caph.focus.controllerProvider.getInstance().focus("playvideo");
```

6.2.1 Rolovateľné zoznamy

Využíva sa doplnok jQuery (tzv. `caphList`), ktorý poskytuje vysoko výkonný komponent rolovateľného zoznamu. Tento doplnok vytvorí kontajner a prvok obalu na usporiadanie zobrazených položiek a vypočíta počet zobrazení na stránku, aby sa prispôbil veľkosti kontajnera. Následne vytvorí aktuálnu, predchádzajúcu a nasledujúcu stránku. Zobrazenia položiek na každej stránke sa aktualizujú daným zobrazením šablóny a údajmi o položke a potom sa usporiadajú do zodpovedajúcej pozície na každej stránke. [39]

Tento doplnok aktualizuje zobrazenia položiek na každej stránke pri prechode na predchádzajúcu alebo nasledujúcu stránku posúvaním prvku na predchádzajúcu alebo nasledujúcu pozíciu. [39]

Je možné ho jednoducho vytvoriť iba pomocou možností nastavenia a zobrazenia šablóny.

Najskôr je potrebné pripraviť údaje na zobrazenie. Pri použití `CaphList` pre zobrazenie dynamických dát, je nutné nahrávať tieto dáta z externého zdroja, v tomto prípade pomocou technológie AJAX (viz. 6.1.4). Po úspešnom načítaní dát v JSON formáte, je ich možné spracovať a vykresliť v `CaphList`.

Po načítaní dát z externého zdroja a ich spracovaní sa používa šablóna pre vykreslenie dát v CaphList. Šablóna je HTML kód, ktorý určuje, ako majú byť dáta zobrazené v CaphList. Táto šablóna môže obsahovať textové polia, obrázky, tlačidlá a ďalšie prvky, ktoré sa budú zobrazovať pre každú položku v CaphList.

```

<div id="list1" style="position: absolute; left: 525px; top: 220px;"
// šablona sa vloží tu
<\div>

<script>
var items;

$.ajax({
  url: 'https://www.catvusa.com/wp-json/wp/v2/broadcasts?_fields=...',
  method: 'GET',
  success: function(response) {
    items = response;
  },
  error: function(error) {
    console.log('There was an error: ' + error);
  }
});

$('#list1').caphList({
  template:<div class="item" data-focusable-name="<%=item.focusName%"
  id="<%= item.id %" data-focusable-group="<%=item.focusGroup%"
  ...
  <div class="movie-img"><%= item.img %><\div>
  <div class="card-body"><%=item.title%> <%=item.desc%> <\div>
  ...'
  <\div> ',
  items : items,
  containerClass : 'list',
});
<\script>

```

Tento kód, je mierne upravený pre lepšie porozumenie, ale veľmi podobne sa používal aj pre vytvorenie zoznamu CaphList komponenty. Prvá časť kódu vytvára základ pre zoznam v HTML kóde, s div elementom s ID `list1`, ktorý sa bude používať ako kontajner pre zoznam.

Potom nasleduje JavaScript kód, ktorý zahŕňa AJAX požiadavku na určitú URL adresu, ktorá obsahuje dáta pre položky zoznamu. Po úspešnej odpovedi sa dáta uložia do premennej `items`. Následne je inicializovaná CaphList komponenta s použitím jQuery knižnice. Konfigurácia komponenty zahŕňa template reťazec, ktorý sa použije na vykreslenie každej položky, pole s dátami pre položky zoznamu (`items`), triedu kontajnera pre zoznam (`containerClass`). Podľa dokumentácie je možné pridávať ďalšie vlastnosti. Celkový výsledok kódu je zoznam v aplikácii s položkami načítanými z dátovej služby pomocou AJAX požiadavky a vykreslenými pomocou CaphList komponenty a šablóny.

Šablóna obsahuje kód s použitím `%= item.propertyName %`. Ide o vlastnosť ako jazyk JavaScript umožňuje vkladať dynamické hodnoty do reťazcov.

Na vykreslenie `CaphListu` sú potrebné len dáta z jednej požiadavky, ktorá súvisí s jeho obsahom. Táto požiadavka sa načíta pri spustení aplikácie. Napríklad, ak sa zobrazuje zoznam s videami z kategórie o českej kuchyni, treba počkať na túto požiadavku, ktorá vráti všetky dáta ohľadom týchto videí. `CaphList` navyše zobrazuje dáta postupne. To znamená že najskôr vloží do HTML dokumentu len tie položky, ktoré sa zmestia na obrazovku. Ak užívateľ začne hýbať so zoznamom, tak sa začnú načítavať ďalšie. Veľkosť HTML dokumentu je teda minimálna a to aplikáciu zťažuje tiež minimálne.

6.3 Realizácia jednotlivých častí

Aplikácia je štruktúrovaná pomocou niekoľkých `wrapper` `div` elementov, ktoré poskytujú prehľadné a usporiadané rozloženie. Obsahuje hlavnú obrazovku s menu, obrazovku s detailom videa, obrazovku vyhľadávania, obrazovku prihlásenia používateľa a obrazovku prehrávania médií. Pomocou jQuery metód `show()` a `hide()` sa v určitých prípadoch jeden `wrapper` objaví a ostatné sa skryjú. Pri výbere videa zo zoznamu, sa html elementy v `div` elemente `wrapper` pre detail videa naplnia údajmi, hlavná obrazovka sa skryje a zobrazí sa obrazovka s detailom videa.

6.3.1 Práca s údajmi

Aplikácia používa jQuery na dynamické zmeny vzhľadu webovej stránky. Súbor `index.html` obsahuje základnú štruktúru stránky a jQuery slúži na pridávanie a odstraňovanie HTML elementov, úpravu ich obsahu a štýlu, a tým mení celkový vzhľad stránky. V takomto prípade, ak aplikácia uchováva nejaké dáta na strane klienta, je dobré použiť `Session Storage`. Uľahčí to presúvanie vybraných dát naprieč elementami.

`Session Storage` objekt ukladá údaje len pre jednu reláciu. Znamená to, že údaje uložené v `session Storage` sa vymažú po zatvorení prehliadača. Relácia stránky trvá tak dlho, kým je otvorený webový prehliadač, a tiež dokáže prežiť obnovenie stránky. [40]

Keď užívateľ otvorí stránku na novej karte alebo okne, webový prehliadač vytvorí novú reláciu. Ak užívateľ otvorí viacero kariet alebo okien s rovnakou adresou URL, webový prehliadač vytvorí samostatnú `session Storage` pre každú kartu alebo okno. Údaje uložené na jednej karte webového prehliadača teda nemôžu byť prístupné na inej karte. Keď užívateľ zatvorí kartu alebo okno, webový prehliadač ukončí reláciu a vymaže údaje v `session Storage`. Údaje uložené v `session Storage` je špecifický pre protokol stránky. Napríklad tá istá stránka `javascripttutorial.net` má iný `sessionStorage` pri prístupe s `http` a `https`. [40]

Na prácu s aplikáciou pre Smart TV stačilo použiť metódy `setItem()` a `getItem()` pri ukladaní a načítavaní dát zo `session Storage`. Keď užívateľ zameria určitý prvok na stránke, aplikácia uloží informácie o danom prvku do `Session Storage` pomocou metódy `setItem`. Následne, ak používateľ zvolí daný prvok, aplikácia získa uložené informácie o prvku z úložiska pomocou metódy `getItem` a zobrazí ich v správnom elemente na stránke.

Okrem toho sa využíva aj `local Storage` na ukládanie dát o prihlásenom užívateľovi. Ten sa od `session Storage` líši tým že údaje ostanú uložené aj po vypnutí aplikácie. Užívateľ sa tak stále nemusí prihlasovať znovu po vypnutí aplikácie

6.3.2 Hlavná obrazovka

Stránka je rozdelená do dvoch hlavných častí pomocou `div` elementov. Bočné menu a hlavný obsah, ktorý obsahuje videá. Tieto dve časti sú reprezentované v HTML kóde pomocou `<div>` elementov s pridelenými triedami `sidebar` a `mainContent`. Na realizáciu menu sa používa element `` pre zobrazenie menu položiek.

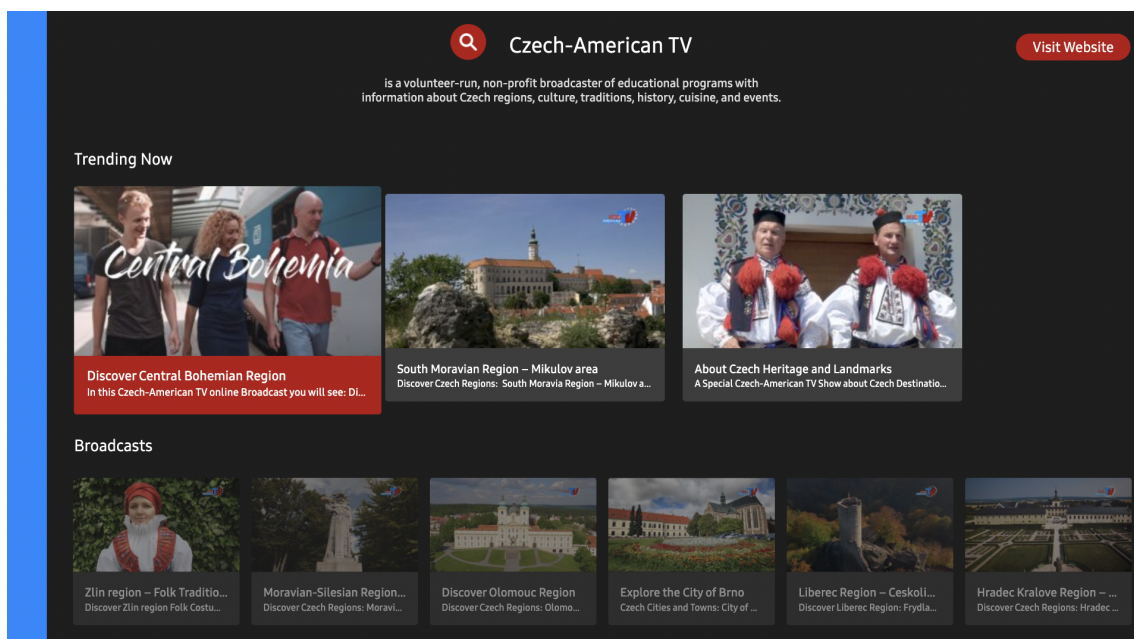
Hlavná časť obsahuje menu položky v HTML elemente ``. Každá položka aplikácie má pridelenú triedu `focused`, ktorá sa zvýrazní pomocou CSS, keď je daná položka zameraná. Položka sa môže zväčšiť, zmeniť farbu alebo prípadne sa obsah položky len podčiarkne. V hornej časti obrazovky sa nachádza element `<h2>`, ktorý slúži ako názov aplikácie, a kratší popis v `<p>` elemente. Pod nimi sú jednotlivé rolovateľné zoznamy, ktoré obsahujú obrázok a telo položky. Telo položky, ktoré sa nachádza pod obrázkom, zobrazuje názov videa ako `<h2>` element a popis videa ako `<p>` element. V prípade príliš dlhého popisu sa text skráti na tri bodky. To bolo zabezpečené pomocou CSS pravidla `text-overflow: ellipsis`, ktoré zúži text na konci riadku pomocou trojbodkového významu. V prípade položiek, ktoré nemajú náhľadový obrázok, ako je napríklad rádio, tak sa použila len ikona a názov kategórie rádia.

Okrem navigácie a rolovateľných zoznamov, pri implementácii hlavnej obrazovky bolo potrebné tiež zohľadniť, kde sa užívateľ so svojim zameraním nachádza. Počet zoznamov je až šesť, čo znamená, že by nemalo dôjsť k situácii, keď sú všetky zoznamy zobrazené na jednej fyzickej obrazovke. Je potrebné myslieť na posúvanie obsahu na obrazovke. To bolo zabezpečené pomocou metódy `scrollIntoView()` v jazyku JavaScript, ktorá posúva stránku na zvolenú pozíciu na obrazovke pre lepší kontakt s užívateľom. Túto metódu je potrebné zavolať na konkrétny prvok, ktorý je potrebné posunúť do zorného poľa. V prípade tejto aplikácie, sa zoznam posunie vždy do stredu obrazovky, okrem úplne vrchného a spodného zoznamu. V prípade týchto zoznamov sa totiž obrazovka nemá kde posunúť.

Na túto funkciu bol implementovaný autorom práce skript `checkfocus.js`, ktorý zabezpečuje kontrolu polohy prvku, na ktorý je užívateľ zameraný, a následne podľa toho posúva obrazovku respektíve v prípade potreby vyroluje menu. To sa dosiahne pomocou animácie, zmeny triedy a pridania pravidiel CSS. Tento skript teda zabezpečuje, že užívateľ má na obrazovke vždy zobrazené len to, čo potrebuje, a to na správnej pozícii.



Obrázek 6.1. Hlavná obrazovka smart TV Samsung aplikácie.



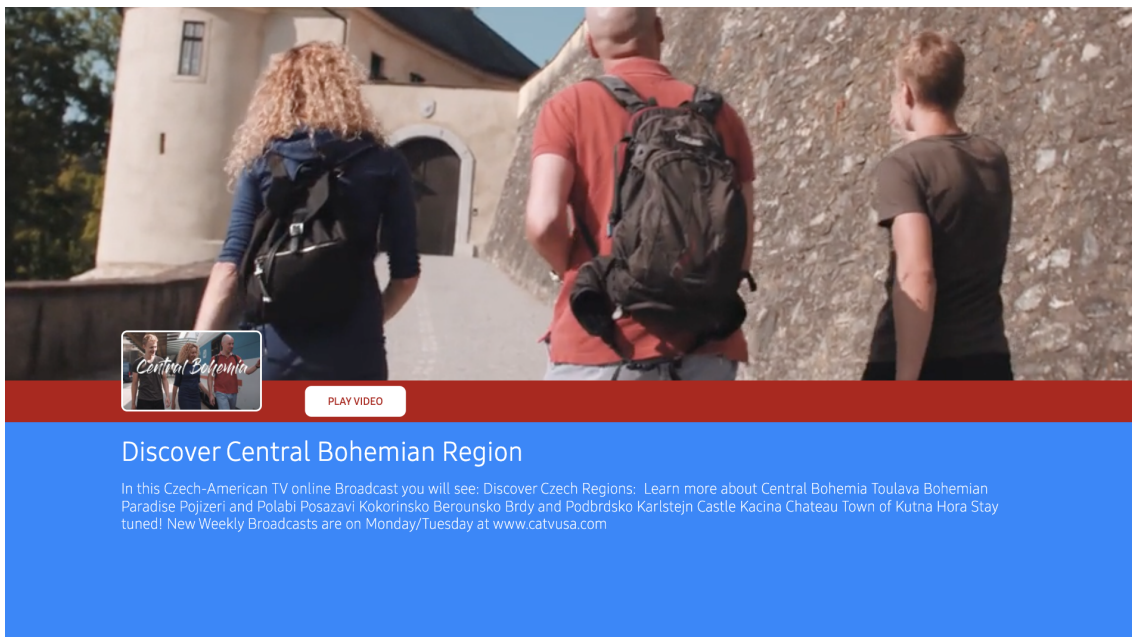
Obrázek 6.2. Hlavná obrazovka smart TV Samsung aplikácie so zameraním na prvok.

6.3.3 Obrazovka detailu videa

Bolo potrebné vykonať zmenu obrazovky detailu videa, ktorá zahŕňala dvojité zobrazenie titulného obrázka, a to automatickým prehrávaním videa v hornej časti obrazovky. Cieľom tejto zmeny bolo zvýšiť angažovanosť užívateľa a uľahčiť mu rozhodnutie, či si chce video pozrieť.

Obrazovka detailu videa ma tri základné prvky. Prehrávané video, časť s obrázkom a tlačidlom na prehratie videa a časť s popisom videa a jeho názvom. Najvrchnejšia prvá časť obsahuje prvok videa s atribútmi `autoplay`, `muted` a `loop`, spolu s prvkom zdroja, ktorý špecifikuje adresu URL súboru videa a jeho typ.

Keď sa kód vykoná a na obrazovke sa zobrazia elementy, ktoré sú súčasťou obrazovky pre detail videa, tak sa zobrazí prehrávač videa s načítaným videom, ktoré sa automaticky začne prehrávať v slučke a bez zvuku. Zadávateľ ešte požadoval, aby sa video začalo prehrávať po tridsiatich sekundách, aby sa nezobrazovala úvodná upútavka. To sa podarilo zabezpečiť pomocou atribútu `currentTime`, ktorý ma každý html prvok video.



Obrázek 6.3. Obrazovka detailu videa smart TV Samsung aplikácie.

6.3.4 Vyhľadávania

V tejto aplikácii pre Smart TV napísanú aj v JavaScripte a s využitím knižnice jQuery sa nachádza vyhľadávanie. Pre tento účel boli využité dobre známe regulárne výrazy.

Bola vytvorená globálna premenná, ktorá bude obsahovať všetky videá a ich názvy a popisy. Táto premenná je zoznam, ktorý obsahuje všetky videá a ich názvy a popisy, teda zoznam všetkého možného obsahu, ktorý si užívateľ môže spustiť. Následne bola vytvorená funkcia, ktorá bude vyhľadávať v tejto globálnej premennej na základe textu, ktorý zadal užívateľ do vyhľadávacieho poľa. Na vytvorenie regulárneho výrazu sa použije zadaný text. Na základe získanej hodnoty zadaného textu v poli sa vytvorí regulárny výraz. Tento regulárny výraz by mal hľadať zhodu s textom v názve alebo popise videa. Ďalej prechádza funkcia každý objekt v globálnej premennej a porovnáva názov a popis každého videa s regulárnym výrazom. Ak sa názov alebo popis videa zhoduje s regulárnym výrazom, pridá sa do poľa výsledkov. Ostáva už len výsledky vyhľadávania zobraziť v užívateľskom rozhraní aplikácie v podobe rovnakého rolovateľného zoznamu ako pri hlavnej obrazovke

6.3.5 Prehrávač videa

Spoločnosť CATVUSA ma vo svojej databáze ku každému videu uložené aj videá, ktoré nepredstavujú obsah, ktorý si užívateľ vybral. Preto bude potrebné vytvoriť

akési krátke playlisty, ktoré sa spúšťajú po výbere konkrétneho videa v aplikácii. Tieto playlisty zvyčajne obsahujú tri videá, ale to nie je pravidlo. Často prvým videom je reklama, ktorá hovorí o neziskovej organizácii CATVUSA a snaží sa priblížiť užívateľovi, kto je zodpovedný za obsah v aplikácii. Po nej nasleduje samotný obsah, ktorý si užívateľ vybral. Tretie video môže byť vyzvanie na podporu neziskovej organizácie alebo upozornenie, že všetok obsah a ešte viac si užívateľ môže pozrieť aj na webových stránkach.

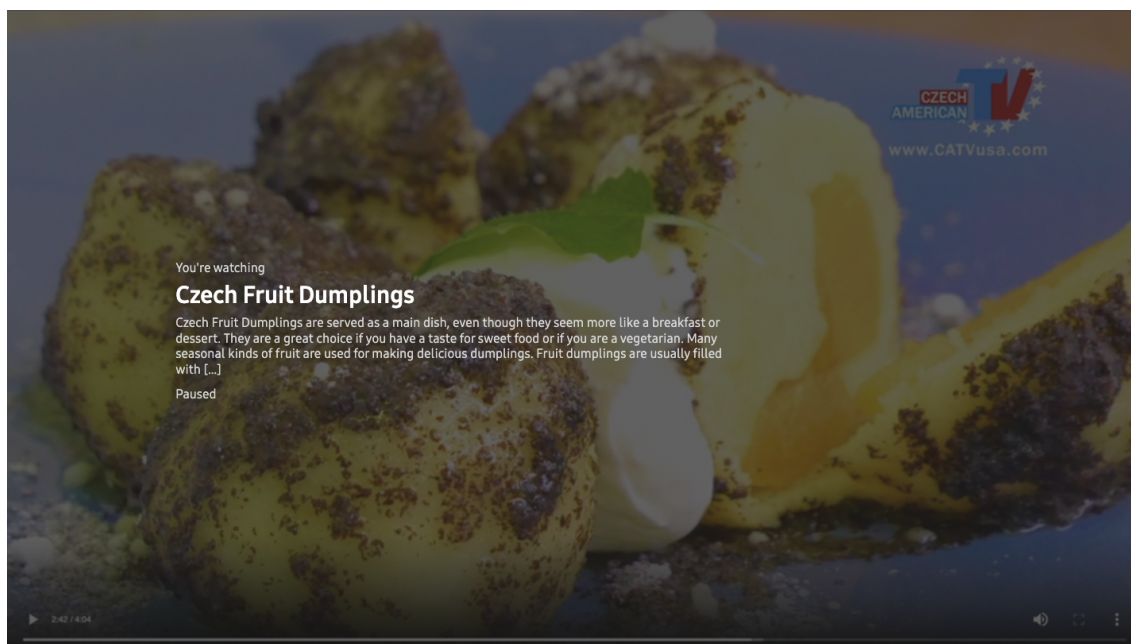
Na spúšťanie videa v HTML kóde sa používa HTML tag `<video>` a jeho príslušné atribúty ako napríklad `src`, `controls`, `autoplay`, `loop`, `muted`, a pod. Okrem toho bol využitý aj JavaScript a jeho udalosti, aby sa pridali interakcie s videom, ako napríklad prehrávanie, pozastavenie, pretočenie a podobne.

Udalosť je dôležitou súčasťou JavaScriptu. Webová stránka reaguje podľa udalosti, ktorá nastala. Poslucháč udalostí je procedúra v JavaScripte, ktorá čaká na výskyt udalosti. Jednoduchým príkladom udalosti je kliknutie užívateľa alebo stisnutie klávesu na klávesnici. [41] Veľmi často, a hlavne pri implementácii obrazovky videa, sa používala funkcia `AddEventListener()`.

`AddEventListener()` je vstavaná funkcia v JavaScripte, ktorá počúva udalosť a druhý argument, ktorý sa má volať vždy, keď sa spustí opísaná udalosť. [41] Takáto vstavaná funkcia, ktorá počúva udalosť, bola použitá aj na vytvorenie už spomínaného playlistu.

Udalosť, na ktorú sa počúva, je `ended`, to znamená keď sa video skončí. Keď sa táto udalosť stane, vykoná sa anonymná funkcia, ktorá pustí ďalšie video v poradí.

Ďalšia udalosť, ktorá sa používa, nie len pri obrazovke videa, je obsluha pri stisnutí tlačidla na diaľkovom ovládači. To platí aj pri zobrazení informácií o videu, ktoré sa práve prehráva. Ak užívateľ pozastaví video, zobrazia sa mu po určitom časovom intervale tieto informácie. Ak užívateľ opäť stisne nejaké tlačidlo, obsluží sa udalosť tak, že tieto informácie zmiznú. Podobne sa obsluží udalosť, aj keď užívateľ chce pretočiť video alebo pozastaviť. HTML element `<video>` ma množstvo atribútov s ktorými je možné pracovať ako napríklad `currentTime`, `volume` alebo metódy `pause()` či `play()`.



Obrázek 6.4. Obrazovka pozastaveného videa.

6.3.6 Prehrávač rádia

Ako už bolo spomínané (viz. 4.2.4), vzhľadom na to, že sa jedná o rádio v televízii, užívateľovi nie sú poskytnuté len textové informácie, ale aj iné obrazové prvky, ktoré zvyšujú celkový zážitok a pomáhajú užívateľom.

Aplikácie v rámci implementácie rádia, bude využívať API, ktoré bolo súbežne vyvíjané v rámci inej záverečnej práce. [11] Toto API poskytuje, ďalšie koncové body, ktoré je možné využiť:

- 1) https://catvusa.com/wp-json/rm_extended_api/v1/radio/data
- 2) https://catvusa.com/wp-json/rm_extended_api/v1/radio/player_data
- 3) https://catvusa.com/wp-json/rm_extended_api/v1/radio/playlist_data
- 4) https://catvusa.com/wp-json/rm_extended_api/v1/radio/warning_data
- 5) https://catvusa.com/wp-json/rm_extended_api/v1/radio/settings_data

- 1) vráti všetky dáta
- 2) vráti názov rádia, logo, dĺžku nahrávky a popis o autorovi a skladbe
- 3) vráti playlist dáta, vďaka ktorým je možné prehrávať nahrávky v určitom poradí
- 4) vráti upozornenie pre počúvajúceho, ako je napríklad, či ešte stále počúva
- 5) vráti zaujímavosti o prehrávanej nahrávke aj spolu s obrázkom

Je dôležité dobre porozumieť, aké dáta sa nachádzajú v odpovedi. Každá odpoveď obsahuje tri kategórie rádia (klasická hudba, ľudová hudba a nahrávky o českej kultúre). Na to, aby sa rozlišovalo, o akú kategóriu ide, sa využíva názov kľúča `genres`. Taktiež je potrebné použiť informáciu, ktorá sa skrýva pod kľúčom `slug`, aby bolo možné rozlíšiť, či ide o samotnú skladbu, jingle alebo upútavku.

Po načítaní dát do príslušných premenných, bolo potrebné tieto dáta spracovať a vypísať ich do užívateľského rozhrania. Na to bol použitý už dobre známa JavaScriptová knižnica jQuery, ktorá sa používa na interakciu s dokumentom HTML. Nielenže uľahčuje manipuláciu s HTML elementami na webovej stránke, ale vďaka jQuery bolo možné počas realizácie rádia jednoduchšie vypisovať text do HTML elementov. Najskôr sa získal prístup k určitému HTML elementu pomocou jeho identifikátora a potom sa do neho jednoducho vložil text pomocou funkcie `.text()` alebo `.html()`.

Bola využitá aj možnosť, že v HTML elementoch je možné ukladať informácie pomocou atribútov. Napríklad v audio elemente môžu byť uložené informácie o názve a umelcovi skladby pomocou atribútov `data-name` a `data-artist`. Tieto informácie môžu byť následne použité pomocou jQuery, v rôznych súvislostiach. Napríklad pomocou funkcie `.attr()` v jQuery je možné získať hodnotu atribútu a následne ju použiť. V prípade ľudovej hudby, boli využité tieto atribúty na to, aby sa vedelo, ktorá skladba v poradí sa prehráva. Rádio na stránkach CATVUSA je totiž implementované tak, že po troch skladbách jedného interpreta nasleduje oznámenie o práci spoločnosti alebo upútavka na iný obsah. Okrem toho sa prehráva aj krátka zvuková ukážka (tiež známa ako jingle), ktorá je charakteristická pre organizáciu.

Interpret aj jeho skladba sa vyberá náhodne, aby sa zvýšil efekt rádia. Na generovanie náhodného čísla sa využíva vstavaná funkcia jazyka JavaScriptu `Math.random()`, ktorá generuje náhodné desatinné číslo medzi 0 (vrátane) a 1. Následná sa výsledná hodnota vynásobí určitým maximum. Aby výsledok bol celé číslo tak sa ešte využíva tiež vstavaná funkcia jazyka JavaScriptu `Math.floor()`, ktorá zaokrúhľuje desatinné číslo na najbližšie celé číslo smerom nadol.

Celkovo je implementácia rádia kombinácia vypisovania dát do html elementov pomocou jQuery a využívania atribútov.



Obrázek 6.5. Obrazovka rádia smart TV Samsung aplikácie.

6.3.7 Úvodná obrazovka

Úvodná obrazovka je veľmi dôležitý prvok pre užívateľskú prívetivosť, pretože pomáha zaujať pozornosť užívateľa, kým sa načítajú ďalšie dáta (viz. 4.2.7).

Po skončení všetkých asynchrónnych volaniach sa volá časť kódu, ktorá sa nazýva **callback funkcia** a definuje sa pomocou metódy `done()` alebo `success()` v objekte, ktorý reprezentuje AJAX požiadavku. Táto funkcia sa spustí po dokončení AJAX požiadavky a spracuje dáta vrátené zo servera. V tejto funkcii môže byť definovaný kód na spracovanie a zobrazenie dát, ktoré boli získané z AJAX požiadavky. Taktiež môže byť definovaná aj funkcia pre ošetrenie chýb, ktoré sa môžu vyskytnúť počas komunikácie so serverom, a to pomocou metódy `fail()` alebo `error()`. [42]

Používa sa teda callback funkcia. Po dokončení požiadavky sa volá lokálne vytvorená funkcia `isLoading()`, ktorá si len inkrementuje premennú. Ak bude počet inkrementácií rovný počtu žiadostí na server, tak to znamená, že dáta sú všetky načítané a zobrazené. Ak však čakanie na odpovede presiahne interval piatich sekúnd, úvodná obrazovka zmizne a zvyšné dáta sa načítajú asynchrónne. Pri testovaní s priemerným internetovým pripojením, boli všetky požiadavky spracované skôr ako ubehol tento interval.

6.3.8 Hlasové ovládanie

Zariadenia Samsung, ako sú Smart TV, podporujú rôzne webové rozhrania API, ktoré umožňujú vývoj webových aplikácií. Okrem rozhrania W3C/HTML5 API, vďaka ktorému je možné vytvoriť užívateľské rozhranie aplikácie pomocou HTML a CSS, je ďalším takým rozhraním aj VoiceControl API poskytujúce rozhrania a metódy na rozpoznávanie hlasových príkazov. Hlasové ovládanie API ponúka funkcie na rozpoznávanie hlasu a odoslanie výsledku ako preddefinovaný príkaz. [43] [6]

Postup bol prebratý z dokumentácie a všetky nasledujúce metódy sú súčasťou VoiceControl API. [6] Na začiatku je potrebné vytvoriť objekt na správu hlasového ovládania pomocou metódy `tizen.voicecontrol.getVoiceControlClient()`. Táto metóda vytvára novú inštanciu objektu `VoiceControlClient`, ktorá sa použije na správu hlasových ovládacích príkazov a udalostí. Ďalej bolo vytvorené pole objektov `tizen.VoiceControlCommand`, každý s iným názvom (napr. "play random video"). Tieto príkazy budú zaregistrované v rámci systému hlasového ovládania a môžu byť rozpoznané aplikáciou, keď ich užívateľ vysloví.

Tiež je potrebné volať metódu `setCommandList()` na objekt pre správu hlasového ovládania vytvorený na začiatku. Táto metóda potrebuje ako argumenty pole príkazov a reťazec `BACKGROUND`. Toto nastavuje zoznam príkazov a špecifikuje, že príkazy by mali byť rozpoznané v popredí, teda keď je aplikácia spustená. Ďalší blok kódu, ktorý je potrebné pridať je funkcia callback `resultListenerCallback()`, ktorá sa zavolá, keď ovládacie rozhranie hlasového ovládania zistí, že užívateľ vyslovil príkaz, ktorý zodpovedá jednému z registrovaných príkazov a môže spustiť obsluhu tohto príkazu. Na koniec stačilo pridať metóda `addResultListener()`, ktorú sa volá s argumentom callback funkcie. Tým sa pridá funkcia callback ako poslucháč výsledkov. Funkcia vráti špecifický identifikátor, ktorý je vhodný na kontrolu, či registrácia poslucháča prebehla správne.

6.4 S2Member Pro API

S nainštalovaným s2Member Pro je možné mať prístup k s2Member Pro API pre vzdialené operácie. Toto je k dispozícii pre vývojárov, ktorí chcú dynamicky vytvárať používateľské/členské účty prostredníctvom vlastných skriptov. [44]

API poskytuje nasledujúce operácie

- `auth_check_user` (overenie existujúcich používateľov/členov)
- `get_user` (získať údaje o existujúcich užívateľoch/členoch)
- `create_user` (vytvoriť používateľa, ak neexistuje, inak aktualizovať)
- `modify_user` (aktualizuje existujúcich používateľov/členov)
- `delete_user` (vymaže existujúcich používateľov/členov)

Využíva sa iba jeden koncový bod a telo požiadavky sa odovzdáva vo formáte JSON ako pole `x-www-form-urlencoded` pod kľúčom `s2member_pro_remote_op`. Je teda potrebné serializovať JSON formát na reťazec a uložiť pod kľúč `s2member_pro_remote_op`. Na pripojenie k API sa potom využíva koncový bod spolu s datmi:

```
POST https://www.catvusa.com/?s2member_pro_remote_op=1

op: 'get_user',
  api_key: 'api_key',
  data: {
    user_id: "user_id",
    user_login: 'username',
    user_pass: 'password'
  }
```

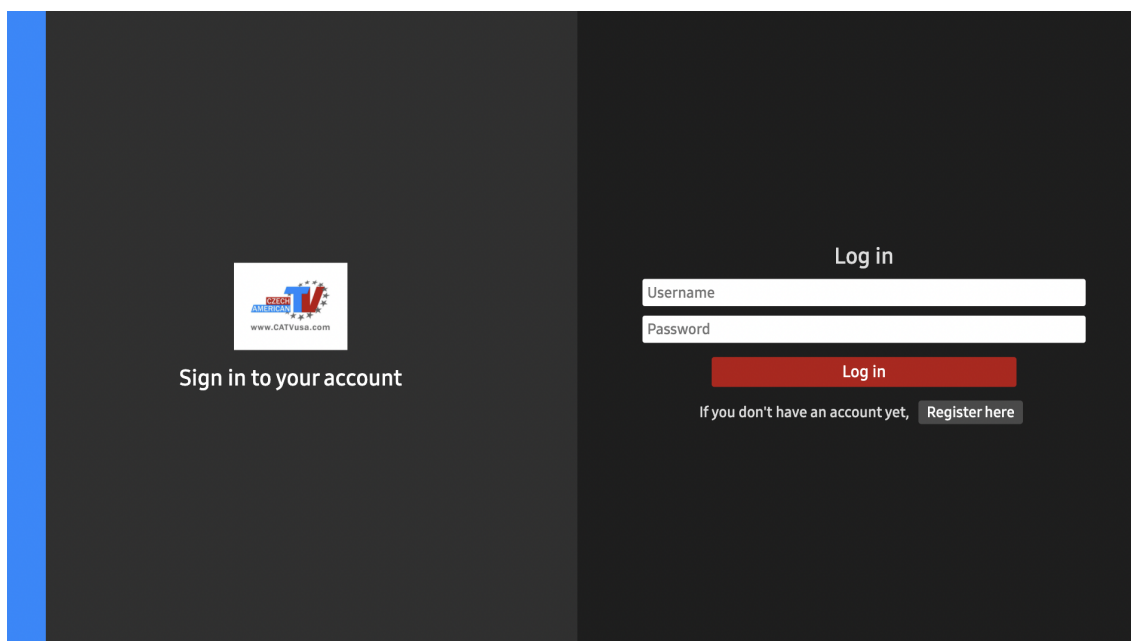
[44] V aplikácií sa využíva iba metóda `get_user` a `auth_check_user`, ktorá vyzerá takmer rovnako, akurat bez `user_id` položky.

Toto experimentálny API neumožňuje zistiť požadovanú úroveň členstva pre zobrazenie konkrétnych príspevkov, avšak v PHP knižnici tohto pluginu existuje metóda, ktorá vráti nastavenú požadovanú úroveň podpory pre príspevok. Autor aplikácie pre Android TV, Viktor Sinelnikov vytvoril nasledujúci koncový bod: GET <https://www.catvusa.com/isprotected2member.php>, ktorý prijíma postako parameter požiadavky a vráti požadovanú úroveň podpory, alebo -1, ak je príspevok prístupný pre všetkých. Tento koncový bod bol využitý aj pri vývoji tejto aplikácie.

6.4.1 Obrazovka prihlásovania

Táto obrazovka obsahuje dve polia. Jedno pre zadanie prihlasovacieho mena a druhé pre zadanie hesla. Pod týmito poliami sa nachádza tlačidlo **Prihlásiť sa**, ktoré slúži na potvrdenie prihlasovacích údajov a prihlásenie používateľa. Okrem toho sa tu nachádza aj tlačidlo **Registrácia**, ktorého kliknutím sa používateľ presmeruje na stránku registrácie.

Tieto prvky sú všetky implementované pomocou HTML a CSS. Pole pre login a heslo sú textové polia, ktoré sa vzhľadom prispôbujú zvyšku obrazovky. Tlačidlo **Prihlásiť sa** je implementované ako tlačidlo, ktorého farba a tvar sú definované pomocou CSS. Podobne aj tlačidlo **Registrácia** je tlačidlo s vlastným štýlom pomocou CSS.



Obrázek 6.6. Obrazovka prihlásovania smart TV Samsung aplikácie.

Kapitola 7

Testovanie

Manuálne testovanie zdrojového kódu JavaScript je časovo náročný proces. To spomaľuje proces vývoja aplikácie. Ak nastane situácia, že sa zavedú nejaké zmeny v jednej funkcii a túto funkciu využívajú niektoré ďalšie funkcie, tak by sa malo skontrolovať, ako tieto zmeny ovplyvňujú súvisiace funkcie. V opačnom prípade môže aplikácia zlyhať alebo vrátiť nesprávny výsledok. Keď sa zavádzajú akékoľvek opravy do kódu, mali by sa skontrolovať, ako ovplyvňujú prácu aplikácie. V takýchto prípadoch sa ukáže ako veľmi užitočné sú jednotkové testy. Je možné napísať test a spustiť ho pri každej zmene kódu. [45]

Na testovanie vyvíjanej aplikácie pre Smart TV Samsung boli požadované automatizované jednotkové a integračné testy.

7.1 Jednotkové testovanie

Unit testing je proces vývoja softvéru, v ktorom sú najmenšie testovateľné časti aplikácie, nazývané jednotky, individuálne a nezávisle kontrolované, či správne fungujú. Túto testovaciu metodiku vykonávajú počas procesu vývoja vývojári softvéru a niekedy aj zamestnanci QA. Hlavným cieľom testovania jednotiek je izolovať napísaný kód na testovanie a určiť, či funguje podľa plánu. Testovanie jednotiek je dôležitým krokom v procese vývoja, pretože ak sa vykoná správne, môže pomôcť odhaliť skoré chyby v kóde, ktoré môže byť ťažšie nájsť v neskorších fázach testovania. Jednotkový test zvyčajne pozostáva z troch fáz: plán, prípady a skriptovanie a samotný jednotkový test. V prvom kroku sa pripraví a skontroluje test jednotky. Ďalším krokom je vytvorenie testovacích prípadov a skriptov, potom sa otestuje kód. [46]

Každý testovací prípad sa testuje nezávisle v izolovanom prostredí, aby sa zabezpečilo, že v kóde nie sú žiadne závislosti. Vývojár softvéru by mal kódovať kritériá na overenie každého testovacieho prípadu a testovací framework možno použiť na hlásenie všetkých neúspešných testov. Vývojári by nemali robiť test pre každý riadok kódu, pretože to môže zabrať príliš veľa času. Vývojári by potom mali vytvoriť testy zamerané na kód, ktorý by mohol ovplyvniť správanie vyvíjaného softvéru. [46]

7.1.1 Typy jednotkového testovania

Jednotkové testy je možné vykonávať manuálne alebo automatizovane. Tí, ktorí používajú manuálnu metódu, môžu mať vytvorený inštinktívny dokument s podrobnosťami o každom kroku procesu; automatizované testovanie je však bežnejšou metódou jednotkových testov. Automatizované prístupy bežne používajú testovací rámec na vývoj testovacích prípadov. Tieto rámce sú tiež nastavené tak, aby označili a nahlásili všetky neúspešné testovacie prípady a zároveň poskytli súhrn testovacích prípadov. [46]

7.1.2 Realizácia jednotkových testov

V kapitole o použitých technológiach, v časti o tizen študiu, bolo spomínané, že tizen štúdio tiež umožňuje vytvárať testovacie projekty a overiť tak spoľahlivosť kódu. (viz. 5.4)

Pri vytvorení testovacieho projektu v Tizen Studiu sa vytvorí štruktúra projektu, ktorá obsahuje súbory potrebné na vytvorenie a spúšťanie testov. Pri otvorení súboru `test.html` sa zobrazí prostredie na testovanie, ktoré slúži na spúšťanie testov a zobrazovanie výsledkov. Toto prostredie sa podobá na prostredie od qUnit, pretože Tizen Unit Test Framework, ktorý je súčasťou rozšírení v tizen študiu, používa qUnit ako základný testovací nástroj pre aplikácie napísané v HTML5 a v JavaScripte.

QUnit je framework na testovanie jednotiek JavaScriptu na testovanie projektov jQuery alebo akéhokoľvek kódu JavaScript. Bol vyvinutý Johnom Resigom a pôvodne bol súčasťou jQuery. Od roku 2009 je to samostatná knižnica. Je vydaný pod licenciou MIT. [45]

V priečinku `qunit` sú umiestnené skripty `qunit.js` a `qunit-reporter.js`, ktoré sú súčasťou qUnit frameworku. `qunit.js` obsahuje hlavnú funkcionálnu časť pre testovanie, ako sú funkcie pre definovanie testovacích prípadov, asertívne funkcie pre overenie výsledkov a ďalšie. `qunit-reporter.js` slúži na vizualizáciu výsledkov testovania a generovanie správ o úspechu alebo zlyhaní testov.

Testovacie nástroje v Tizen študiu teda využívajú qUnit ako testovací nástroj pre aplikácie napísané v HTML5 a v JavaScripte a umožňujú generovanie testovacích súborov v qUnit formáte.

Na zostavenie testu jednotky potrebné použiť metódu `test()`, ktorá vyžaduje tri parametre: názov testu, očakávaný počet tvrdení (ktoré test obsahuje) a funkciu obsahujúcu tvrdenia. [45].

Keďže sa jedná o aplikáciu, ktorá sa programuje veľmi podobne ako webová stránka, tak jednotkové testovanie bolo navrhnuté tak, aby pokrývalo funkcionálnu časť webovej aplikácie, ktorá pracuje s DOM dokumentom a využíva predovšetkým jQuery. Preto boli jednotky navrhnuté ako funkcie, ktoré pracujú s DOM a zahrňujú rôzne operácie. Pri práci s HTML dokumentom sa testovali nasledujúce vlastnosti, ktoré boli zaobalené do funkcií, ktoré bolo možno volať:

- pridanie alebo odstránenie prvku
- zmenu viditeľnosti prvku
- prechod na inú obrazovku
- zmena CSS pravidiel
- ovládanie videa a zvuku
- posúvanie (scroll) dokumentu
- vkladanie textu do elementov
- zmena zamerania elementu
- výpočet nasledujúceho prvku

Tieto funkcie by mali pokryť veľkú časť udalostí, ktoré sa budú diať, keď užívateľ bude interagovať s aplikáciou. Každá funkcia obsahovala logiku, ktorá zabezpečovala správne vykonanie požadovanej operácie a vracala boolean hodnotu `true` v prípade, že sa operácia podarila. Tak bolo aj testovanie jednoduchých funkcií pohodlnejšie.

Boli vytvorené aj testy, ktoré pracujú s dátami, ktoré prišli zo servera. Funkcia mohla najskôr roztriediť JSON súbor a potom sa využila opäť funkcia na zápis do HTML elementu a vrátenie hodnoty v prípade úspechu. Okrem toho boli otestované aj funkcie ako výpočet nasledujúcej hodnoty v prípade rádia. Pretože rádio by malo

pôsobit náhodne, ale v niektorých prípadoch by malo prehrávať rovnakého interpreta aspoň trikrát za sebou.

Pre každú funkciu boli napísané testy, ktoré pokrývali všetky možné scénáre a situácie, ktoré mohli nastať. Testy boli spúšťané pravidelne. Tento prístup zabezpečil, že aplikácia bola spoľahlivá a bez problémov fungovala. Celkový prístup k jednotkovému testovaniu bol zameraný na spoľahlivosť aplikácie, čo je veľmi dôležité, najmä v prípade aplikácií, ktoré pracujú s DOM a využívajú JavaScriptové frameworky, ako napríklad jQuery.

7.2 Integrované testovanie

Integrované testovanie je proces testovania rozhrania medzi dvoma softvérovými jednotkami alebo modulmi. Zameriava sa na určenie správnosti rozhrania. Účelom testovania integrácie je odhaliť chyby v interakcii medzi integrovanými jednotkami. Po otestovaní všetkých modulov sa vykoná integračné testovanie. [47]

Integračné testovanie je technika testovania softvéru, ktorá sa zameriava na overenie interakcií a výmeny údajov medzi rôznymi komponentmi alebo modulmi softvérovej aplikácie. Cieľom integračného testovania je identifikovať akékoľvek problémy alebo chyby, ktoré vzniknú, keď sa rôzne komponenty kombinujú a vzájomne pôsobia. Integrované testovanie sa zvyčajne vykonáva po testovaní jednotky a pred testovaním systému. Pomáha identifikovať a riešiť integračné problémy na začiatku vývojového cyklu, čím znižuje riziko závažnejších a nákladnejších problémov neskôr. Testovanie integrácie je možné vykonať výberom modulu po module. Dá sa to urobiť tak, že by sa mala dodržiavať správna postupnosť. A tiež, ak nechcete zmeškať žiadne integračné scénáre, musíte postupovať podľa správnej postupnosti. Odhalenie defektov je hlavným zameraním testovania integrácie a časom interakcie medzi integrovanými jednotkami. [47]

7.2.1 Realizácia integračných testov

Ak Smart TV aplikácia používa len WordPress REST API ako zdroj dát, potom je potrebné testovať integráciu medzi aplikáciou a WordPress REST API. Integrované testovanie by sa preto malo zamerať na overenie, ako aplikácia komunikuje s WordPress REST API a či správne získava, spracováva a zobrazuje dáta z tohto zdroja. Celkovo by sa aplikácia mala správať stabilne pri obdržaní rôznych dat zo servera.

Aby sa vykonali integračné testovanie pre Smart TV aplikáciu, bolo potrebné vytvoriť testy, ktoré simulujú interakciu aplikácie s WordPress REST API. Bolo potrebné vytvoriť simuláciu WordPress REST API. To bolo možné pomocou nástrojov ako napríklad JSON Server. Tento nástroj umožňuje vytvoriť falošný server, ktorý bude simulovať WordPress REST API a bude poskytovať falošné dáta. Testy by potom overili, či aplikácia správne interaguje s touto simulovanou verziou WordPress REST API. Ak aplikácia bude testovaná na dostatočnom množstve rôznych dat, potom by aplikácia mala správne fungovať, keď sa použije v reálnom prostredí s WordPress REST API ako zdrojom dát.

Situácia pri testovaní aplikácie pomocou integračných testov bola veľmi úspešná vďaka použitiu JSON servera ako falošného WordPress servera. Bolo vytvorené dostatočné množstvo dát, ktoré umožnili testovať rôzne prípady, ako napríklad chýbajúce atribúty alebo prázdne refazce. Toto testovanie sa uskutočnilo až na konci vývoja, a tak stačilo len jednoducho nahradiť odkaz skutočného zdroja dát za falošný server. Vytvorenie testov bolo rýchle a jednoduché a testy overili, či sa dáta zobrazujú

správne a aplikácia funguje stabilne. Vďaka týmto testom boli odhalené chyby, ktoré boli následne ihneď opravené. Celková skúsenosť z testovania bola veľmi pozitívna a ukázala, že použitie integračných testov s JSON serverom je efektívne a spoľahlivé.

7.3 Užívateľské testovanie

Užívateľské testovanie, alebo testovanie použiteľnosti sa používa pri vývoji počítačových aplikácií. Tieto metódy sledujú správanie samotných užívateľov, čím môžu odhaliť chyby, ktoré zostali vývojárom skryté. Testovanie je vhodné nasadiť už v ranej fáze vývoja aplikácie. Užívateľské testovanie tak môže určovať správny smer pri vývoji a predchádzať tak vznikajúcim problémom. Podľa Steva Kruga môžeme užívateľské testovanie prirovnať k návšteve cudzincov vo vašom meste: Keď ich budete vykonávať po vašom meste, nevyhnutne uvidíte veci, ktoré ste si obvykle nevšimli, pretože ste na ne boli zvyknutí. A súčasne si uvedomíte, že veľa vecí, ktoré vám pripadajú jasné, nebýva zrejmá všetkým. [48]

Užívateľské testovanie bolo vykonané s autorom aplikácie a dvoma ďalšími respondentmi. Počas testovania bola aplikácia prezentovaná respondentom a boli im dané úlohy, aby zistili, ako sa s aplikáciou pracuje, ako ju používajú a aké sú ich reakcie na dizajn a funkcie aplikácie.

Počas testovania bola identifikovaná chyba v navigácii späť v histórii zobrazených položiek, kedy používatelia očakávali, že po návrate na predchádzajúcu zobrazenú položku sa zvýrazní tá položka, ktorá je najbližšie k aktuálnej položke a nie tá, ktorá bola ako posledná zobrazená. Takto si CaphList ukláda históriu. Bola tiež zistená potreba výraznejšieho zvýraznenia vybraných položiek v menu, aby bolo jasnejšie vidieť, ktorá položka bola vybraná.

Celkovo bola aplikácia hodnotená ako jednoduchá a ľahko použiteľná. Respondenti sa rýchlo naučili, ako aplikáciu používať a nevyskytli sa žiadne výrazné problémy s ovládaním aplikácie.

Kapitola 8

Záver

V tejto diplomovej práci bola vyvinutá aplikácia pre Samsung Smart TV pre neziskovú organizáciu, ktorá popularizuje Českú republiku v USA. Požiadavky na aplikáciu zahŕňali prehrávanie videí z kategórií týkajúcich sa českých regiónov, českých receptov a učenia sa českého jazyka. Okrem toho bolo požadované aj vytvorenie rádia zameraného na klasickú, ľudovú hudbu a na podcasty o českej kultúre. Prihlásený užívateľia majú výhodu a to v podobe prístupu k niektorým videám z obmedzenej sekcie a počúvanie rádia bez prerušenia.

V práci bola použitá široká škála technológií pre vývoj aplikácie pre inteligentné televízie Samsung. Vývoj prebiehal v prostredí Tizen Studio, kde bola aplikácia testovaná na simulátore alebo emulátore a následne aj na fyzických zariadeniach. Zariadenia Samsung podporujú rozhrania W3C/HTML5 API, ktoré umožňujú vytváranie užívateľského rozhrania aplikácie pomocou HTML, CSS a Javascriptu spolu s jQuery. Pre plošnú navigáciu a rolovatelné zoznamy bola použitá knižnica CAPH. V priebehu práce s ňou sa vyskytli niektoré problémy, najmä pri pridávaní dynamických HTML elementov, ktoré zamedzili správne zameriavaniu prvkov. Tieto problémy však boli vyriešené nepridávaním elementov a navigácia bola otestovaná úspešne. Aplikácia komunikuje so serverom pomocou WordPress REST API. Na pripojenie k serveru boli použité rôzne koncové body s určitými parametrami, čo umožnilo skrátiť rozsah odpovede. Pre získanie dát pre radio bola použitá iná sada koncových bodov, ktorá bola vytvorená v rámci inej záverečnej práce a mohla byť využitá na konci vývoja. Vďaka tomu môže aplikácia, na rozdiel od aplikácie pre Android TV, zobrazovať zaujímavé informácie o počúvanom obsahu. Napríklad informácie o ľudových piesňach alebo o autoroch klasickej hudby.

Analýza a návrh užívateľského rozhrania aplikácie pre Samsung Smart TV boli opreté o dizajn aplikácie pre Android TV a o odporúčania od spoločnosti Samsung. Vylepšenia návrhu rozhrania boli navrhnuté na základe analýzy a zohľadnenie potrieb cieľovej skupiny užívateľov a požiadavok od zadávateľa.

Aplikácia bola testovaná pomocou jednotkových testov, ktoré testovali najmenšie funkcie a metódy, vďaka ktorým bolo postavené užívateľské rozhranie a funkcionálnosť aplikácie. Okrem toho boli prevedené aj integračné testy, kedy bolo potrebné využiť falošný server (JSON server), kde sa mohli uložiť rôzne dáta a testovať tak, či aplikácia bude stabilná aj pri obdržaní neočakávaných dát. Navyše, aplikácia prešla aj užívateľským testovaním, avšak len autorom aplikácie a pár ďalšími respondentmi. Navrhne sa však riaditeľovi organizácie CATVUSA, aby spustil užívateľské testovanie na budúcich užívateľoch, ktorí už majú s organizáciou CATVUSA skúsenosti. Tak by sa mohli odhaliť aj chyby, ktoré sú závislé na hardware. Po úspešnom testovaní by aplikácia mala byť pridaná do obchodu značky Samsung.

Celkovo som si z tejto práce odniesol cenné skúsenosti a vedomosti o vývoji aplikácií pre Smart TV a o prispôsobovaní sa špecifickým požiadavkám organizácií. Verím, že táto práca bude užitočná pre organizáciu a bude ju používať veľa spokojných užívateľov, ktorí sa chcú dozvedieť viac o Českej republike.

Literatúra

- [1] *Již osmnáct let vzdělává Czech-American TV v USA české krajany*. [online]. [cit. 1.4.2023]. Dostupné z: <https://brnozurnal.cz/archiv/rozhovory-vzpominky-medailonky/jiz-osmnact-let-vzdelava-czech-american-tv-v-usa-ceske-krajany/>.
- [2] Sinelnikov Viktor. *Vývoj aplikace na chytré televize pro neziskovou TV*. Praha: České vysoké učení technické v Praze, 2021.
- [3] *Samsung Developers - Design Principles*. [online]. [cit. 15.2.2023]. Dostupné z: <https://developer.samsung.com/smarttv/design/design-principles.html>.
- [4] *Samsung Developers - UX Checklist*. [online]. [cit. 23.2.2023]. Dostupné z: <https://developer.samsung.com/smarttv/design/ux-checklist.html>.
- [5] *About Czech-American TV*. [online]. [cit. 25.2.2023]. Dostupné z: <https://catvusa.com/about-czech-american-tv/>.
- [6] *Samsung Developers - VoiceControl API*. [online]. [cit. 25.4.2023]. Dostupné z: <https://developer.samsung.com/smarttv/develop/api-references/tizen-web-device-api-references/voicecontrol-api.html##VoiceControlVoiceControlCommand>.
- [7] *Funkčné požiadavky*. [online]. [cit. 1.3.2023]. Dostupné z: <https://sk.theastrologypage.com/functional-requirement>.
- [8] *Nefunkčné požiadavky (Non Functional Requirements)*. [online]. [cit. 1.3.2023]. Dostupné z: <https://managementmania.com/sk/nefunkcne-poziadavky-non-functional-requirements>.
- [9] *UI/UX Design on Smart TV*. [online]. [cit. 15.3.2023]. Dostupné z: <https://noriginmedia.com/2021/06/21/10-tips-for-ui-ux-design-on-smart-tv/>.
- [10] *What is Netflix?* [online]. [cit. 16.4.2023]. Dostupné z: <https://help.netflix.com/en/node/412>.
- [11] Gorbatenko Margarita. *Creating a plugin for managing partners on the web*. Praha: České vysoké učení technické v Praze, 2023.
- [12] Hao Yanbin. *Understanding loading animations — types and applications*. 2019. <https://uxdesign.cc/understanding-loading-animation-types-and-its-application-e41ba914b847>.
- [13] *WhatIs.com HTML5*. [online]. [cit. 23.3.2023]. Dostupné z: <https://www.techtarget.com/whatis/definition/HTML5>.
- [14] *Mozilla Developer Network. CSS: Cascading Style Sheets*. [online]. [cit. 21.3.2023]. Dostupné z: <https://developer.mozilla.org/en-US/docs/Web/CSS>.

- [15] *Mozilla Developer Network. (2022). JavaScript.* [online]. [cit. 25.3.2023]. Dostupné z: <https://developer.mozilla.org/en-US/docs/Web/JavaScript> .
- [16] *CAPH is a Web UI framework for TV, CAPH3 Documentation.* [online]. [cit. 3.4.2023]. Dostupné z: <https://developer.samsung.com/onlinedocs/tv/caphdocs/index.html> .
- [17] *jQuery Foundation. (2022). jQuery.* [online]. [cit. 11.4.2023]. Dostupné z: <https://jquery.com/> .
- [18] *Mozilla Developer Network. CSS: Cascading Style Sheets.* [online]. [cit. 30.3.2023]. Dostupné z: <https://developer.mozilla.org/en-US/docs/Web/CSS> .
- [19] *Tizen Docs - Introduction to Tizen.* [online]. [cit. 6.4.2023]. Dostupné z: <https://docs.tizen.org/platform/what-is-tizen/overview/> .
- [20] *Tizen Docs - Tizen Web Application.* [online]. [cit. 10.4.2023]. Dostupné z: <https://docs.tizen.org/application/web/index>.
- [21] *Tizen Docs - Learn About .NET.* [online]. [cit. 11.4.2023]. Dostupné z: <https://docs.tizen.org/application/dotnet/learn-about-dotnet/>.
- [22] *Tizen Docs - Tizen Native Application.* [online]. [cit. 11.4.2023]. Dostupné z: <https://docs.tizen.org/application/native/index>.
- [23] *Tizen Docs - Tizen architecture.* [online]. [cit. 11.4.2023]. Dostupné z: <https://docs.tizen.org/application/> .
- [24] *Tizen Docs - Tizen studio.* [online]. [cit. 11.4.2023]. Dostupné z: <https://docs.tizen.org/application/tizen-studio/> .
- [25] *Web Tools in Tizen studio.* [online]. [cit. 3.3.2023]. Dostupné z: <https://docs.tizen.org/application/tizen-studio/web-tools/>.
- [26] *Samsung Developers - Using SDK, TV simulátor .* [online]. [cit. 14.4.2023]. Dostupné z: <https://developer.samsung.com/smarttv/develop/getting-started/using-sdk/tv-simulator.html> .
- [27] *Samsung Developers - Using SDK, TV Emulator .* [online]. [cit. 15.4.2023]. Dostupné z: <https://developer.samsung.com/smarttv/develop/getting-started/using-sdk/tv-emulator.html> .
- [28] *Samsung Developers - Using SDK, TV Device .* [online]. [cit. 16.4.2023]. Dostupné z: <https://developer.samsung.com/smarttv/develop/getting-started/using-sdk/tv-device.html> .
- [29] *Samsung Developers - Web inspector.* [online]. [cit. 14.4.2023]. Dostupné z: <https://developer.samsung.com/smarttv/develop/getting-started/using-sdk/web-inspector.html> .
- [30] *Samsung Developers - Web Inspector Features.* [online]. [cit. 15.4.2023]. Dostupné z: <https://developer.samsung.com/smarttv/develop/getting-started/using-sdk/web-inspector/web-inspector-features.html> .
- [31] *Developer Resources - Using the WordPress REST API.* [online]. [cit. 11.4.2023]. Dostupné z: <https://developer.wordpress.org/rest-api/> .

- [32] Martin Reddy. *API Design for C++*. Elsevier Science. , 2014. ISBN 978-0-12385-004-1.
- [33] Sufyan bin Uzayr. *Learning WordPress REST API*. Packt Publishing Ltd. , 2016. ISBN 978-1-78646-924-3.
- [34] *REST API Developer Endpoint Reference*. [online]. [cit. 11.4.2023]. Dostupné z:.
<https://developer.wordpress.org/rest-api/reference/>.
- [35] *Using the REST API - Global Parameters*. [online]. [cit. 12.4.2023]. Dostupné z:.
<https://developer.wordpress.org/rest-api/using-the-rest-api/global-parameters/>.
- [36] *Using the REST API - Pagination*. [online]. [cit. 12.4.2023]. Dostupné z:.
<https://developer.wordpress.org/rest-api/using-the-rest-api/pagination/>.
- [37] *What is Ajax*. [online]. [cit. 12.4.2023]. Dostupné z:.
https://www.w3schools.com/whatis/whatis_ajax.asp .
- [38] *Samsung Developers - How To Use Key Navigation With jQuery*. [online]. [cit. 12.4.2023]. Dostupné z:.
<https://developer.samsung.com/onlinedocs/tv/caphdocs/main.html?type=jquery&doc=tutorial&p1=7> .
- [39] *Samsung Developers - caphList Documentation*. [online]. [cit. 11.1.2023]. Dostupné z:.
<https://developer.samsung.com/onlinedocs/tv/caphdocs/main.html?type=jquery&doc=tutorial&p1=7###>.
- [40] *Introduction to JavaScript sessionStorage* . [online]. [cit. 26.3.2023]. Dostupné z:.
<https://www.javascripttutorial.net/web-apis/javascript-sessionstorage/>.
- [41] *JavaScript addEventListener*. [online]. [cit. 11.1.2023]. Dostupné z:.
<https://www.geeksforgeeks.org/javascript-addeventlistener-with-examples/>.
- [42] *Perform an asynchronous HTTP (Ajax) request*. [online]. [cit. 20.4.2023]. Dostupné z:.
<http://api.jquery.com/jquery.ajax/> .
- [43] *Samsung Developers - Web API References*. [online]. [cit. 25.4.2023]. Dostupné z:.
<https://developer.samsung.com/smarttv/develop/api-references/web-api-references.html> .
- [44] *Pro API for Remote Operations* . [online]. [cit. 11.1.2023]. Dostupné z:.
<https://s2member.com/kb-article/pro-api-for-remote-operations/>.
- [45] *QUnit - JavaScript unit testing framework*. [online]. [cit. 1.5.2023]. Dostupné z:.
<https://developer.tizen.org/community/tip-tech/qunit-%E2%80%93-javascript-unit-testing-framework>.
- [46] *Unit testing By TechTarget Contributor*. [online]. [cit. 1.5.2023]. Dostupné z:.
<https://www.techtarget.com/searchsoftwarequality/definition/unit-testing>.

-
- [47] *Software Engineering / Integration Testing*. [online]. [cit. 1.5.2023]. Dostupné z:
<https://www.geeksforgeeks.org/software-engineering-integration-testing/> .
- [48] Steve KRUG. *Nenutte uživatele přemýšlet!: praktický průvodce testováním a opravou chyb použitelnost [sic] webu*. . 1 vyd.. Brno: Computer Press , 2010 . ISBN 978-80-251-2923-4.



Príloha **A**

Skratky

| | |
|---------|--------------------------------------|
| CATVUSA | Czech-American TV |
| API | Application Programming Interface |
| REST | Representational State Transfer |
| WP | Wordpress |
| HTML | The HyperText Markup Language |
| CSS | Cascading Style Sheets |
| JS | JavaScript |
| TV | Televízia |
| HTTP | Hypertext Transfer Protocol |
| URL | Uniform Resource Locator |
| ID | Identifikátor |
| JSON | JavaScript Object Notation |
| XML | Extensible Markup Language |
| AJAX | Asynchronous JavaScript and XML |
| DOM | Document Object Model |
| SVG | Scalable Vector Graphics |
| XHTML | Extensible Hypertext Markup Language |
| W3C | The World Wide Web Consortium |
| MATHML | Mathematical Markup Language |
| CLI | A command-line interface |
| SDK | Software development kit |
| CPU | Central processing unit |
| SOAP | Simple Object Access Protocol |
| CRUD | Create, Read, Update, Delete |
| PHP | Hypertext Preprocessor |