



Meowrent

Dokumentace k ročníkové práci

Autor: Filip Schod

Třída: 3ITC

Vedoucí práce: Tomáš Ulrich

2025/2026

Prohlášení

Prohlašuji, že jsem ročníkovou práci na téma „Meowrent“ vypracoval samostatně a s použitím uvedené literatury a pramenů.

V Ústí nad Labem dne 15.01.2026

.....

Poděkování

Chtěl bych poděkovat Tomášovi Ulrichovi za vedení mé ročníkové práce, cenné rady a odborný dohled.

Anotace

Tato dokumentace popisuje vývoj a realizaci ročníkové práce zaměřená na vytvoření torrentové aplikace. Úvod představuje zaměření práce a její cíle. Rešerše shrnuje inspirační zdroje, které ovlivnily UI a UX prvky aplikace. Technologická část popisuje použitý software a nástroje. Praktická část se věnuje návrhu, vlastnímu vývoji a nasazení aplikace. Závěr hodnotí dosažené výsledky. Dokument také obsahuje použitou literaturu, seznam obrázků, a seznam kódu.

Klíčová slova

torrent, aplikace, desktop, Electron, JavaScript, Nodejs, WebTorrent, peer, stahování, sdílení

Obsah

Úvod	7
1 Teoretická část	8
1.1 Rešerše	8
1.1.1 qBittorrent	8
1.1.2 WebTorrent Desktop	8
1.2 Technologie	9
1.2.1 Electron	9
1.2.2 Node.js	9
1.2.3 Npm	9
1.2.4 Javascript	9
1.2.5 Typescript	9
1.2.6 Neovim	9
1.2.7 ChatGPT	10
1.2.8 WebTorrent	10
1.2.9 WebTorrent Hybrid	10
2 Praktická část	11
2.1 Návrhy	11
2.1.1 Uživatelské rozhraní	11
2.1.2 Uživatelské rozhraní formuláře	12
2.2 Produktizace	13
2.2.1 Funkčnost aplikace	13
2.3 Popis pro uživatele	15
2.3.1 Hlavní okno aplikace	15
2.3.2 Vytváření torrentů	16
2.3.3 Stahování torrentů	16
Závěr	18
Seznam použitých zdrojů	19
Seznam obrázků	20
Seznam kódů	20

Úvod

Hlavním cílem mého ročníkového projektu je vytvořit desktopovou aplikaci. Aplikace bude vyvinuta pomocí frameworku Electron, který umožňuje tvorbu multiplatformních desktopových aplikací s využitím webových technologií. Jednou z klíčových funkcí klienta bude možnost přizpůsobit vzhled klienta. Uživatel bude moci měnit vzhled prostředí a vytvářet vlastní motivy a exportovat je či sdílet s ostatními uživateli.

Torrentový protokol, kolem kterého je klient postaven, je technologie určená pro sdílení dat. Funguje na principu peer-to-peer sítě, kde uživatelé přímo sdílejí jednotlivé části souborů namísto stahování z jednoho centrálního serveru. Díky tomu je přenos dat rychlejší a odolnější vůči výpadkům. Každý uživatel, který data stahuje, se zároveň stává zdrojem pro ostatní.

Téma této práce jsem si vybral, protože se v poslední době zajímám o přizpůsobitelnost a estetiku operačních systémů a také mě zajímalo, jak fungují torrentové soubory. Proto jsem se rozhodl vytvořit přizpůsobitelný torrentový klient, který bude fungovat na většině operačních systémech.

K hlavním očekáváním od projektu patří vytvoření funkčního a upravitelného klientu, který budu moci využívat k sdílení souborů. Také bych rád zlepšil své dovednosti ve vývoji desktopových aplikací, abych mohl vyvíjet další užitečné aplikace.

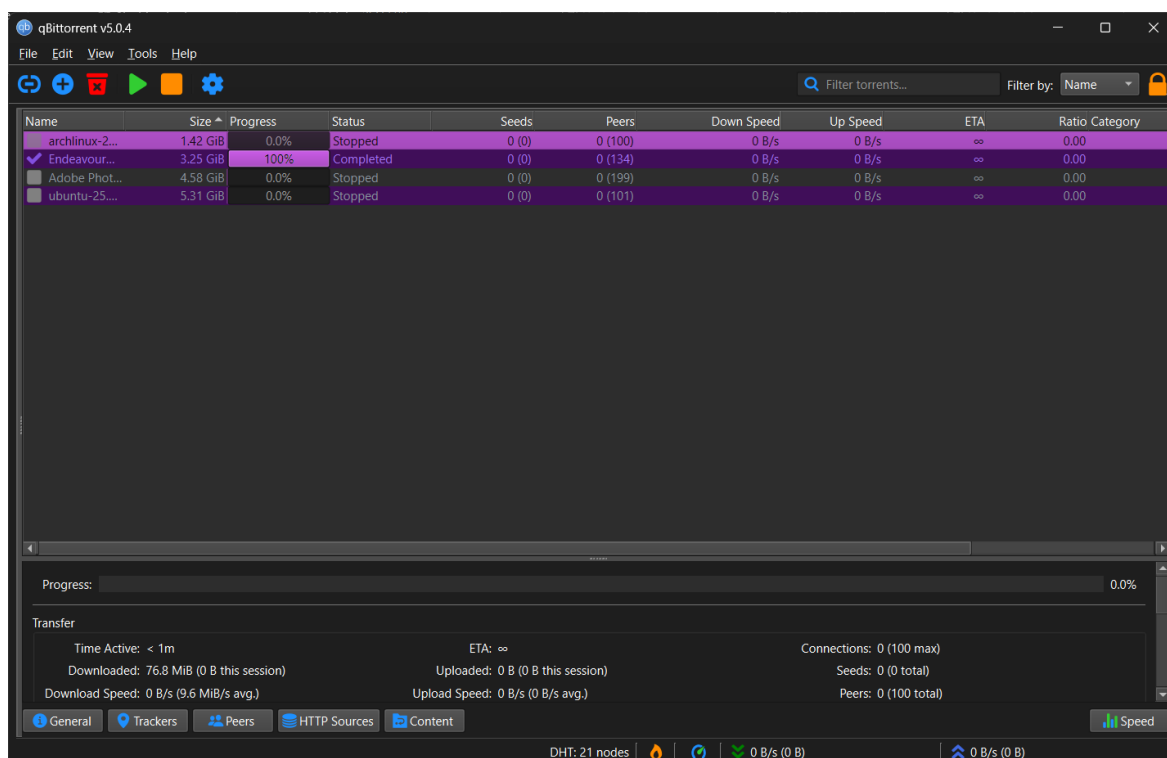
1 Teoretická část

1.1 Rešerše

Tato kapitola se věnuje všem existujícím aplikacím a technologiím, které se zabývají torrentovým stahováním, které jsem použil jako inspiraci k vytvoření svého vlastního klienta. Cílem bylo porozumět jejich uživatelskému rozhraní a technickému řešení, které sloužilo jako podnět pro vytvoření mého projektu.

1.1.1 qBittorrent

qBittorrent je bittorrent klient vytvořený v C++. Snaží se být dobrou alternativou pro všechny bittorrentové klienty. qBittorrent je rychlý, stabilní a podporuje mnoho funkcí. (1)



Obrázek 1 - UI qBittorrent, Zdroj: Vlastní

Nabízí přehledné uživatelské rozhraní, správu torrentů, práci s magnet linky a pokročilé možnosti nastavení. Inspirací pro projekt Meowrent byla především struktura seznamů aktivních torrentů a způsob zobrazování informací o rychlosti stahování a počtu peerů.

1.1.2 WebTorrent Desktop

WebTorrent Desktop je streamovací torrentový klient pro web i desktopové počítače. Webtorrent je napsán kompletně v Javascriptu, takzvaném jazyku webu, a používá WebRTC pro P2P připojení, pokud je možné. Funguje bez doplňků do prohlížeče a bez jakékoliv instalace ve vašem prohlížeči. (2)

Umožňuje práci s torrent protokolem bez nutnosti nízko úrovně sítě implementace. Tento projekt byl klíčovou inspirací pro samotnou technickou realizaci Meowrentu, jelikož poskytuje hotovou knihovnu pro práci s torrenty v prostředí Node.js.

1.2 Technologie

V této kapitole jsou popsány technologie, které byly využity při vývoji aplikace Meowrent.

1.2.1 Electron

Electron je framework, který umožňuje vytvářet multiplatformní desktopové aplikace pro MacOS, Windows, Linux kombinací webových technologií jako jsou HTML, CSS, Javascript s Node.js a nativním kódem. (1)

V projektu Meowrent jsem Electron využil jako základní stavební kámen, na kterém aplikace stojí. Používám tak jeho mezi procesovou komunikace na přenášení dat mezi frontendem a backendem aplikace.

1.2.2 Node.js

Node.js je mezi platformní a open-source Javascriptové běhové prostředí. Je to populární nástroj pro skoro jakýkoliv projekt. (2)

V mém projektu slouží jako základ aplikační logiky, především pro komunikaci s torrentovou knihovnou správu stahovaných souborů a práci se souborovým systémem. Node.js také umožňuje snadnou integraci knihoven pomocí balíčkového systému npm.

1.2.3 Npm

Npm je balíčkový systém pro Node.js. Tento nástroj slouží vývojářům jako usnadnění při vývoji aplikací. Díky němu jsem jednoduše mohl stahovat a nastavovat knihovny, které projekt Meowrent využívá. (3)

1.2.4 Javascript

Javascript je programovací jazyk webu. Může počítat, manipulovat a validovat s daty. Může upravovat HTML a CSS. (4)

V projektu Meowrent je Javascript použit jako hlavní programovací jazyk pro tvorbu uživatelského rozhraní i aplikační logiky. Jeho výhodou je dobrá čitelnost, široká podpora knihoven a snadná integrace s ostatními použitými technologiemi. Vybral jsem si ho, protože jsem s ním dobře seznámen a tako pro mě vyhrál nad druhou možností Typescriptem.

1.2.5 Typescript

Typescript je syntaktický super set Javascriptu, který přidává statické typování. To jednoduše znamená, že dostanete syntaxi navíc k Javascriptu, která umožňuje vývojářům přidávat typy. (5)

Tento jazyk byl zvažován v návrhové části projektu jako možná alternativa k použití čistého Javascriptu. Nakonec však zvolen nebyl, jelikož mám větší zkušenosti s jazykem Javascript, což mi umožnilo rychlejší a plynulejší vývoj aplikace. Dalším důvodem bylo omezení složitosti projektu, protože použití Typescriptu by vyžadovalo dodatečnou konfiguraci, který nebyl pro rozsah ročníkové práce nezbytný.

1.2.6 Neovim

Neovim je refaktor a někdy i redaktor v tradici Vimu (který sám vychází ze Stevieho). Nejedná se o přepracování, ale o pokračování a rozšíření Vimu. Existuje mnoho klonů a derivátů, některé velmi chytré – ale žádný z nich není Vim. Neovim je vytvořen pro uživatele, kteří chtějí mít ty nejlepší stránky Vimů a ještě více. (6)

V rámci projektu Meowrent byl Neovim využit jako hlavní vývojové prostředí pro psaní a úpravu kódu. Editor umožňuje vysokou míru přizpůsobení, práci s více soubory současně a podporu programovacích jazyků pomocí pluginů. Díky tomu zvyšuje produktivitu práce a usnadňuje orientaci v rozsáhlejší kódu projektu.

1.2.7 ChatGPT

ChatGPT je technologií umělé inteligence od společností OpenAI, který využívá nejmodernější algoritmy pro zpracovávání přirozeného jazyka k vytváření smysluplného textu. Tato technologie je určena k automatizaci konverzací a generování textových odpovědí na různé otázky a úkoly. (7)

Sloužil především k vysvětlování technických konceptů, hledání řešení problémů a konzultaci návrhu jednotlivých částí aplikace. ChatGPT byl využíván jako zdroj inspirace a pomoci při ladění chyb. Použití tohoto nástroje přispělo k lepšímu pochopení použitých technologií a urychlení vývojového procesu.

1.2.8 WebTorrent

Webtorrent je streamovací torrentový klient pro Node.js a prohlížeč. Ano, to je správně. Funguje i ve vašem prohlížeči. Je napsán pouze v Javascriptu, takzvaným jazykem webu, takže ten stejný kód funguje v obou běhových prostředích. (8)

V projektu Meowrent tvoří WebTorrent klíčovou součást aplikace, jelikož zajišťuje samotné stahování a sdílení dat v P2P síti. Knihovna poskytuje jednoduché rozhraní pro přidávání torrentů, správu peerů a sledování průběhu přenosu. Díky WebTorrentu nebylo nutné implementovat nízko úrovněnou síťovou komunikaci, což výrazně zjednodušilo vývoj aplikace.

1.2.9 WebTorrent Hybrid

V Node.js je balíček webtorrent schopen se napojovat pouze na normální TCP/UDP peery, nemůže se napojovat na WebRTC peery. Pokud se chcete napojovat na všechny typy peerů, včetně WebRTC pomocí Node.js, musíte použít tento balíček. (9)

Tato technologie byla během návrhu projektu zvažována jako možná alternativa, avšak nakonec nebyla využita. Důvodem bylo ukončení podpory této knihovny.

2 Praktická část

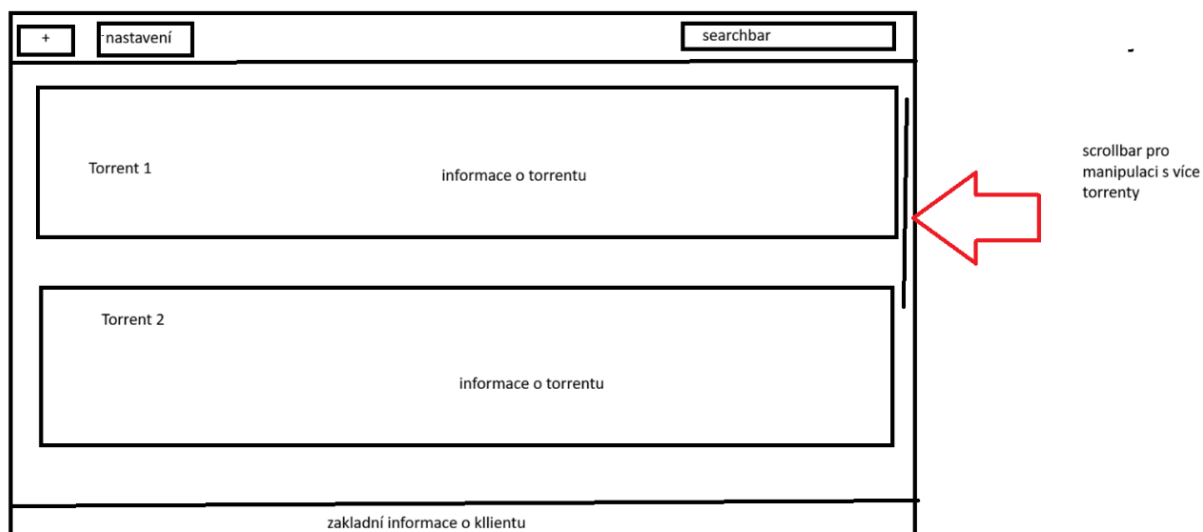
Praktická část dokumentace se věnuje vlastní realizaci projektu Meowrent Tato kapitola popisuje postup vývoje aplikace, návrh jejího uživatelského rozhraní, implementaci klíčových funkcí a výsledné chování aplikace z pohledu uživatele.

2.1 Návrhy

V této kapitole se zaměřuji na návrh uživatelského rozhraní, které uživatel používáme pro ovládání aplikace. Také zde projdu jakým návrhovým potížím jsem čelil při tvorbě formulářů a uživatelského prostředí.

2.1.1 Uživatelské rozhraní

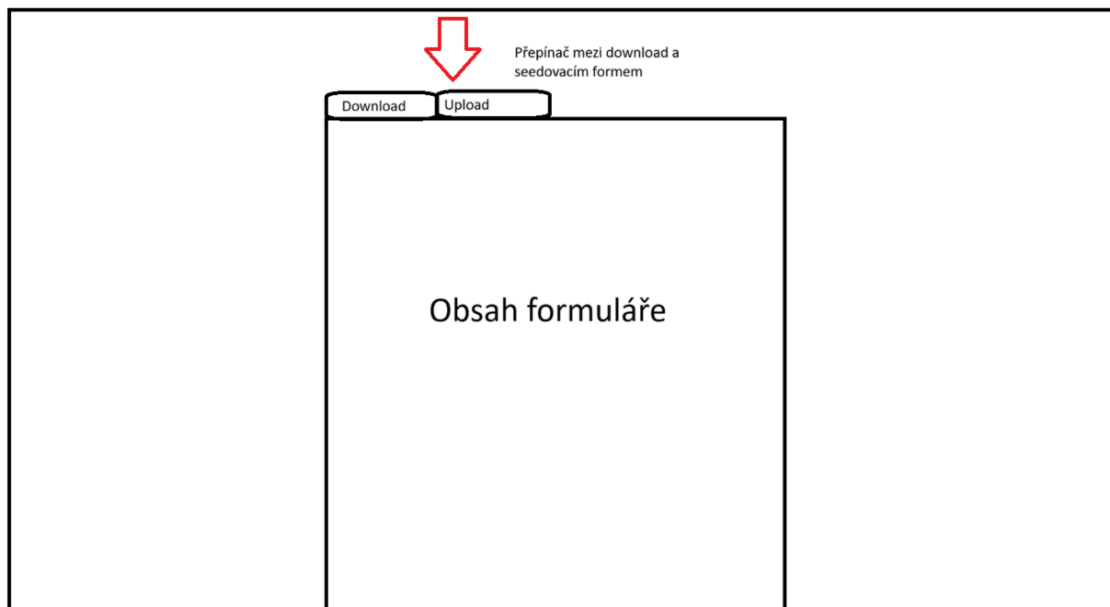
Jako první jsem si vytvořil velice jednoduchý náčrt aplikace. Můžete na něm vidět, že rozvržení aplikace se moc nezměnilo. Stále je zde vidět list torrentů, který automaticky přidává torrenty do listu. Také zde vidíme horní a spodní lištu které obsahují různé funkce a informace klienta.



Obrázek 2 - Návrh UI, Zdroj: Vlastní

2.1.2 Uživatelské rozhraní formuláře

Tento obrázek obsahuje návrh formulářového okna, které slouží k stahování a vytváření torrentů. Tato stránka také obsahuje dvě přepínací tlačítka, kterými si uživatel může vybrat, zda chce stahovat nebo vytvářet torrent.



Obrázek 3 - Návrh formuláře, Zdroj: Vlastní

Obsah tohoto formuláře se během vývoje několikrát změnil, protože jsem ještě nebyl plně rozhodnut, jaké funkce uživateli povolím a jaké pole potřebuji k vytvoření torrentu. Nyní však ve formuláři najdeme veškeré základní věci jako jsou trackery, které pomáhají najít ostatní seedery, nebo místo kde se nacházejí složky které chcete sdílet do torrentu a nebo název torrentu. V stahovací části jsou najdeme možnosti pro vybrání torrentových souborů nebo textové pole pro vložení magnet linků také zde můžete zadat kam chcete torrent uložit.

2.2 Produktizace

V této části se budu zabývat samotným programováním aplikace.

2.2.1 Funkčnost aplikace

Tento kód představuje obsluhu IPC funkce `downloadTorrent`, která zajišťuje přidání a správu stahování torrentů. Funkce je registrována pomocí `ipcMain.handle` a je volána z renderer procesu, odkud přijímá veškeré potřebné parametry, jako je cílová složka pro uložení dat, seznam torrent souborů a magnetických odkazů a uživatelské volby ovlivňující chování stahování.

```
1. ipcMain.handle("downloadTorrent", async (event,
  saveLocation, fileList, linkList,
  startTorrent, topQueue, hashCheck) => {
2.   let torrentList = fileList.concat(linkList)
3.   for (let i = 0; i < torrentList.length; i++) {
4.     const torrentThing = torrentList[i]
5.
6.     client.add(torrentThing, {
7.       path: saveLocation,
8.       skipVerify: hashCheck,
9.       paused: !startTorrent,
10.    }, (torrent) => {
11.      mainWindow.webContents.send('addTorrentToList',{
12.        torrentName: torrent.name
13.      })
14.
15.      let lastDownloadProgress = 0
16.      torrent.on('download', () => {
17.        if(lastDownloadProgress !== Math.floor(torrent.progress * 100)){
18.          lastDownloadProgress = Math.floor(torrent.progress * 100)
19.          console.log(`${torrent.name}: ${lastDownloadProgress}%`)
20.        }
21.      })
22.
23.      torrent.on('done', async () => {
24.        console.log(`${torrent.name} is done downloading`)
25.      })
26.    })
27.  })
28.  formWindow.close()
```

Kód 1 - `downloadTorrent`

Na začátku dochází ke spojení vstupních seznamů do jednoho společného pole, které následně slouží jako fronta torrentů ke zpracování. Pomocí cyklu jsou torrenty postupně přidávány do torrent klienta metodou `client.add`. Při přidání jsou nastaveny konfigurační volby.

Po úspěšném přidání aplikace odešle informace zpět do renderer procesu, čímž aktualizuje uživatelské rozhraní a přidá nový torrent do seznamu stahování. Dále je na objekt `torrent` navázán posluchač událostí, který sleduje průběh stahování. Tento posluchač bude později vylepšen aby zobrazoval veškeré informace přímo do uživatelského rozhraní.

Jakmile je stahování dokončeno, vyvolána další událost, která slouží k informování aplikace o úspěšném dokončení torrentu. Na závěr celé metody je uzavřeno formulářové okno, ze kterého byl proces stahování spuštěn.

Tento kód zajišťuje obsluhu IPC funkce createTorrent, které v aplikaci slouží k vytvoření nového torrentu a jeho následnému seedování. Funkce je vyvolána z renderer procesu a přenáší do hlavního procesu všechny potřebné informace o datech, která mají být sdílena, i o parametrech samotného torrentu.

```
1. ipcMain.handle("createTorrent", async (event,  
    itemsToUpload, torrentName, trackerURLs, torrentComment,  
    pieceLength, privateTorrent, startSeeding) => {  
2.     const autoValue = ""  
3.     if(pieceLength !== autoValue){  
4.         pieceLength = Number(pieceLength)  
5.     }  
6.  
7.     client.seed(itemsToUpload, {  
8.         name: torrentName,  
9.         announce: trackerURLs,  
10.        comment: torrentComment,  
11.        pieceLength: pieceLength,  
12.        private: privateTorrent,  
13.        paused: !startSeeding  
14.        createdBy: "Meowrent - PirateCZ"  
15.    }, (torrent) => {  
16.        console.log("torrent started seeding")  
17.        console.log("Here's the magnet: " + torrent.magnetURI)  
18.        mainWindow.webContents.send('addTorrentToList', {  
            torrentName: torrent.name  
        })  
19.    })  
20.    formWindow.close()  
21. })
```

Kód 2 - createTorrent

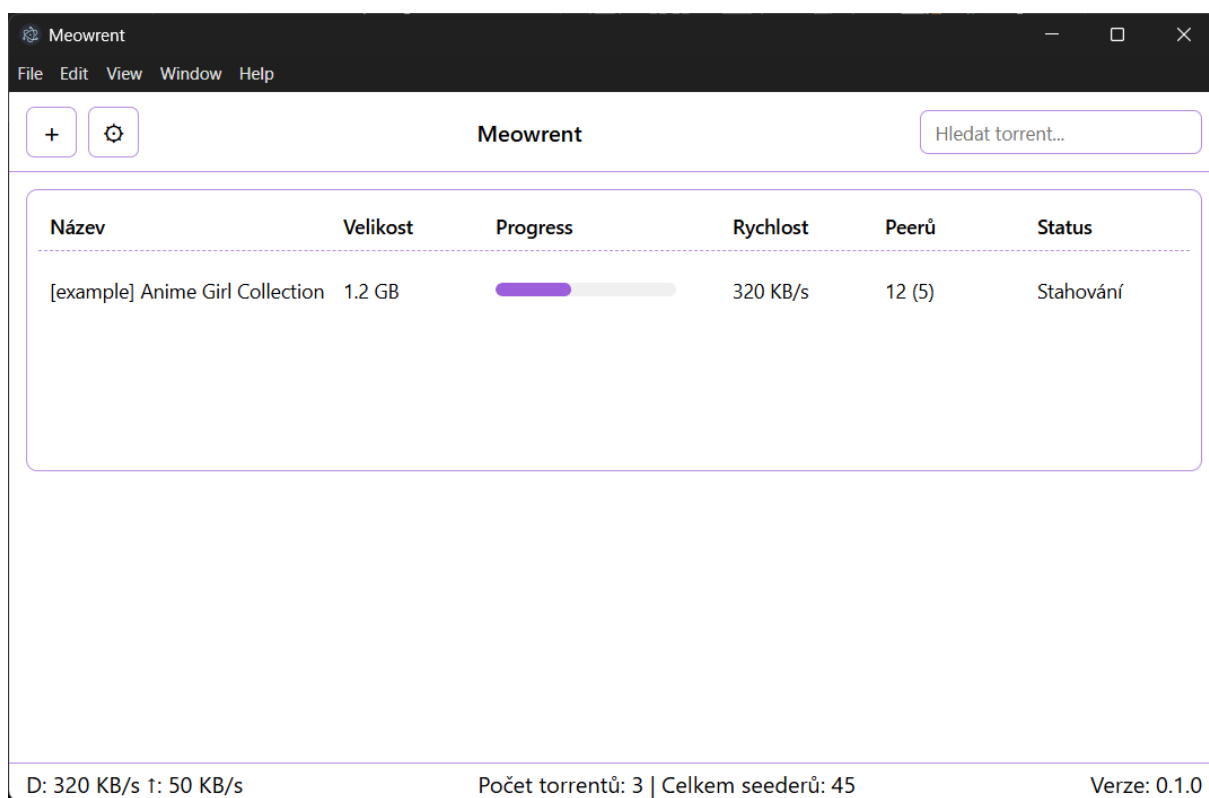
Nejdříve tento kód zaregistruje funkci createTorrent aby mohla být volána z renderer procesu a přenesla veškerá data. Poté správně převedeme délky dílku na číslo, aby nenastaly potíže při vybírání délky. Potom už můžeme zavolat metodu client.seed která automaticky vezme složky nebo soubory a vytvoří z nich torrentový soubor a magnet link. Následně torrent začne seedovat a zde pošleme údaje o torrentu zpátky do renderer procesu abychom je mohli zobrazit zpátky uživateli.

2.3 Popis pro uživatele

Tato kapitola popisuje způsob ovládání aplikace Meowrent z pohledu běžného uživatele. Aplikace je navržena tak aby byla přehledná a snadno ovladatelná i pro méně zkušené uživatele, přičemž zároveň nabízí dostatečné množství funkcí pro běžnou práci s torrentovými soubory.

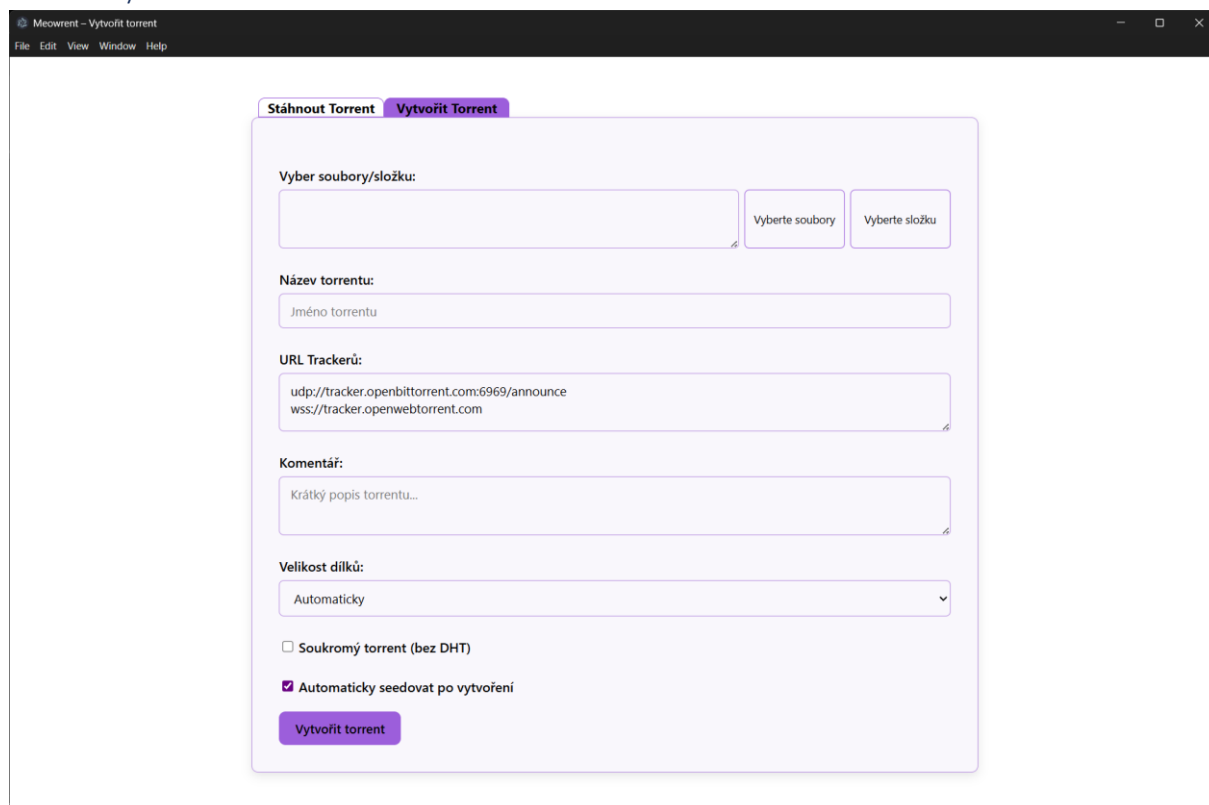
2.3.1 Hlavní okno aplikace

Po spuštění aplikace se uživateli zobrazí hlavní okno, které slouží jako centrální místo pro správu všech torrentů. V této části aplikace je zobrazen seznam aktivních torrentů spolu se základními informacemi o jejich stavu. U každého torrentu je uveden jeho název, celková velikost dat, průběh stahování znázorněný grafickým indikátorem, aktuální rychlost přenosu dat, počet připojených peerů, a aktuální stav torrentu, například stahování nebo seedování. Ve spodní části hlavního okna se nachází stavový řádek, který poskytuje souhrnné informace o celkové rychlosti stahování a odesílání, počtu aktivních torrentů celkovému počtu seedů a aktuální verzi aplikace. Horní část aplikace dále obsahuje ovládací prvky pro přidání nového torrentu, otevření nastavení a vyhledávání v seznamu torrentů. V nastavení si uživatel bude moci upravit vzhled a fungování jeho torrentového klienta.



Obrázek 4 - Hlavní okno, Zdroj: Vlastní

2.3.2 Vytváření torrentů

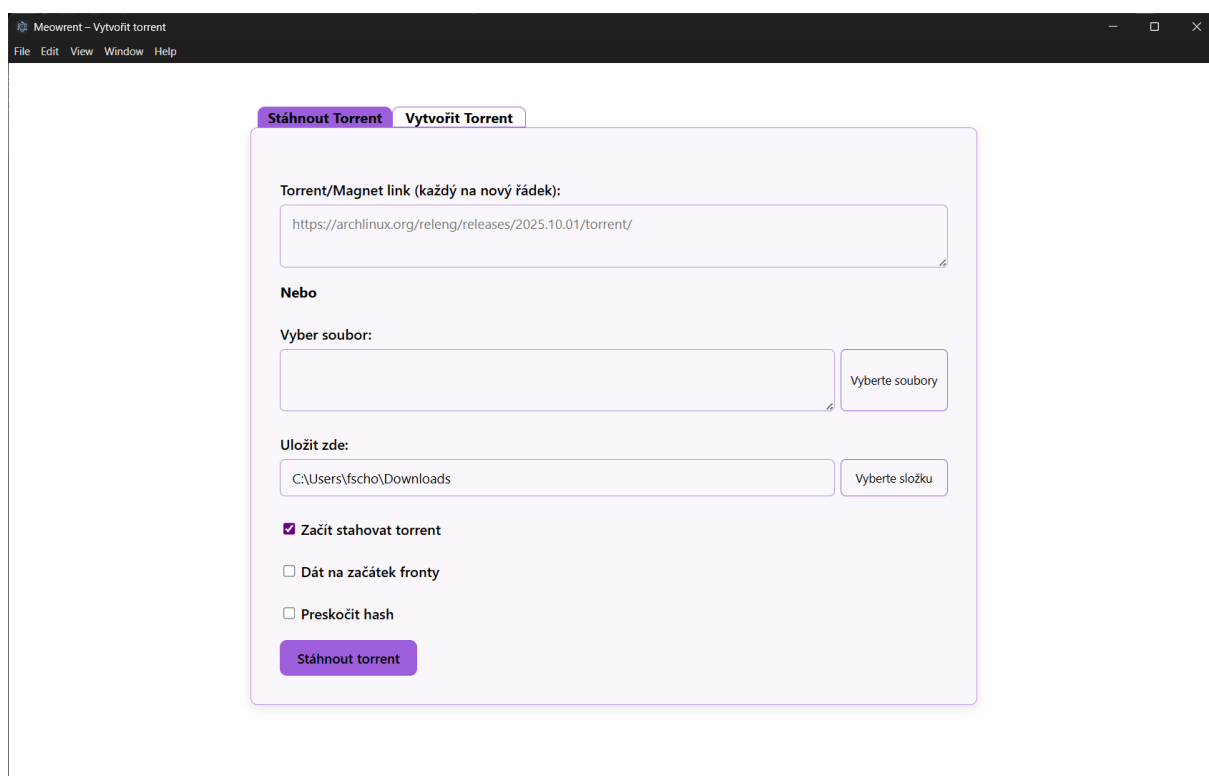


Obrázek 5 - Vytváření torrentů, Zdroj: Vlastní

Aplikace Meowrent umožňuje také vytváření vlastních torrentů určených ke sdílení souborů nebo složek. V dialogu pro vytvoření torrentu uživatel nejprve vybere soubory nebo složku, které chce do torrentu zahrnout. Následně zadá název torrentu a seznam URL adres trackerů, které zajišťují komunikaci mezi jednotlivými klienty v torrentové síti. Volitelně lze doplnit krátký komentář popisující obsah torrentu a upravit velikost datových dílků, přičemž výchozí automatické nastavení je vhodné pro většinu běžných případů. Aplikace dále nabízí možnost vytvořit soukromý torrent bez podpory DHT a nastavit automatické seedování ihned po dokončení vytváření torrentu. Po potvrzení vytvoření se torrent uloží a v případě zapnutého seedování se automaticky objeví v hlavním okně aplikace jako aktivní.

2.3.3 Stahování torrentů

Pro stahování torrentů je v aplikaci k dispozici samostatné dialogové okno, které uživateli umožňuje přidat nový torrent několika způsoby. Uživatel může vložit magnet odkaz nebo URL adresu torrent souboru, přičemž aplikace umožňuje zadat více odkazů současně, každý na samostatný řádek. Alternativně lze torrent přidat také výběrem torrent souboru z lokálního úložiště počítače. Dále je možné zvolit cílovou složku, do které budou stahovaná data uložena. Před samotným zahájením stahování může uživatel nastavit doplňující volby, například automatické spuštění stahování po přidání torrentu, přesunutí torrentu na začátek fronty nebo přeskočení kontroly hash hodnot již existujících dat. Po potvrzení se torrent automaticky přidá do hlavního seznamu aplikace, kde lze sledovat jeho další průběh.



Obrázek 6- Stahování torrentů

Závěr

Tento projekt pro mě představoval výraznou výzvu, a to nejen z technického hlediska, ale také z hlediska osobní organizace práce a přístupu k plnění zadaných cílů. V průběhu realizace jsem se musel naučit efektivně pracovat i v obdobích, kdy jsem nepociťoval dostatečnou motivaci, a postupně si osvojit systematičtější přístup k práci. Významnou zkušeností pro mě bylo také dodržování stanoveného termínu odevzdání, se kterým jsem dříve neměl větší zkušenosti, jelikož jsem na projektech tohoto rozsahu dosud nepracoval. Díky tomu jsem si uvědomil důležitost plánování jednotlivých kroků projektu a rozdělení práce do menších, lépe zvládnutelných částí.

Z odborného hlediska mi projekt umožnil výrazně prohloubit znalosti v oblasti torrentové technologie. Během vývoje aplikace jsem se podrobněji seznámil s principy fungování peer-to-peer sítí, rolí trackerů, významem seedování a samotným procesem distribuce dat mezi jednotlivými uživateli. Tyto poznatky pro mě byly přínosné nejen z hlediska realizace samotného projektu, ale také pro lepší pochopení fungování moderních distribuovaných systémů obecně.

Dalším významným přínosem projektu bylo získání praktických zkušeností s vývojovým prostředím Electron. Práce v tomto prostředí mi umožnila lépe porozumět propojení webových technologií s desktopovými aplikacemi a osvojit si základy struktury Electron aplikací. Během vývoje jsem se setkal s různými technickými problémy, jejichž řešení mi pomohlo rozvinout schopnost samostatně vyhledávat informace, analyzovat chyby a hledat vhodná řešení v dostupné dokumentaci a odborných zdrojích.

Projekt Meowrent mi celkově poskytl cenné zkušenosti, které mohu využít i při další práci na podobných projektech. Uvědomil jsem si, že kromě samotných technických dovedností je při vývoji softwaru velmi důležitá také schopnost plánovat, dodržovat termíny a pracovat systematicky. Do budoucna bych projekt rád dále rozšířil o další funkce a optimalizace.

Seznam použitých zdrojů

1. **sledgehammer999**. qbittorrent/qBittorrent: qBittorrent BitTorrent client. *GitHub*. [Online] qBittorrent. [Cited: Prosinec 29, 2025.] <https://github.com/qbittorrent/qBittorrent>.
2. **WebTorrent, LLC**. WebTorrent - Streaming browser torrent client. *WebTorrent - Streaming browser torrent client*. [Online] WebTorrent, LLC. [Cited: Prosinec 29, 2025.] <https://webtorrent.io/>.
3. **Foundation, OpenJS**. <https://www.electronjs.org/docs/latest/why-electron>. <https://www.electronjs.org/>. [Online] OpenJS Foundation. [Cited: Prosinec 26, 2025.] <https://www.electronjs.org/docs/latest/why-electron>.
4. —. Node.js — Introduction to Node.js. *Node.js — Run JavaScript Everywhere*. [Online] OpenJS Foundation. [Cited: Prosinec 26, 2025.] <https://nodejs.org/en/learn/getting-started/introduction-to-nodejs>.
5. **npm, Inc.** npm About. *npm / Home*. [Online] npm, Inc. [Cited: Prosinec 26, 2025.] <https://www.npmjs.com/about>.
6. **W3Schools**. JavaScript Introduction. *W3Schools Online Web Tutorials*. [Online] W3Schools. [Cited: Prosinec 28, 2025.] https://www.w3schools.com/Js/js_intro.asp.
7. —. TypeScript Introduction. *W3Schools Online Web Tutorials*. [Online] W3Schools. [Cited: Prosinec 28, 2025.] https://www.w3schools.com/typescript/typescript_intro.php.
8. **Neovim**. Vision - Neovim. *Home - Neovim*. [Online] [Cited: Prosinec 26, 2025.] <https://neovim.io/>.
9. **Zuzana**. Co je ChatGPT a jak funguje - Lukáš Barda. *Lukáš Barda / Kurzy AI, digitální rovnováhy a dovedností*. [Online] [Cited: Prosinec 28, 2025.] <https://lukasbarda.cz/co-je-chatgpt-a-jak-funguje/>.
10. **Webtorrent, LLC**. webtorrent - npm. *npm / Home*. [Online] Webtorrent, LLC. [Cited: Prosinec 28, 2025.] <https://www.npmjs.com/package/webtorrent>.
11. —. webtorrent-hybrid - npm. *npm / Home*. [Online] Webtorrent, LLC. [Cited: Prosinec 28, 2025.] <https://www.npmjs.com/package/webtorrent-hybrid>.

Seznam obrázků

OBRÁZEK 1 - UI QBITTORRENT, ZDROJ: VLASTNÍ	8
OBRÁZEK - NÁVRH UI, ZDROJ: VLASTNÍ	11
OBRÁZEK - NÁVRH FORMULÁŘE, ZDROJ: VLASTNÍ	12
OBRÁZEK - HLAVNÍ OKNO, ZDROJ: VLASTNÍ	15
OBRÁZEK - VYTVÁŘENÍ TORRENTŮ, ZDROJ: VLASTNÍ	16
OBRÁZEK - STAHOVÁNÍ TORRENTŮ	17

Seznam kódů

KÓD 1 - DOWNLOADTORRENT	13
KÓD 2 - CREATETORRENT	14