

Spojená škola, Komárňanská 28, 940 75 Nové Zámky  
o.z. Stredná priemyselná škola elektrotechnická S.A.Jedlika  
Jedlik Ányos Elektrotechnikai Szakközépiskola

## **ELITE – English Language Interactive Training Experience**

Stredoškolská odborná činnosť

č. odboru: 14 – Tvorba učebných pomôcok, didaktické technológie

Nové Zámky  
2024

Riešiteľ:  
**Denis Soboslai**  
ročník štúdia: **štvrtý**

---

Spojená škola, Komárňanská 28, 940 75 Nové Zámky  
o.z. Stredná priemyselná škola elektrotechnická S.A.Jedlika  
Jedlik Ányos Elektrotechnikai Szakközépiskola

## **ELITE – English Language Interactive Training Experience**

Stredoškolská odborná činnosť

č. odboru: 14 – Tvorba učebných pomôcok, didaktické technológie

Nové Zámky  
2024

Riešiteľ:  
**Denis Soboslai**  
ročník štúdia: **štvrtý**  
Školiteľ:  
Mgr. Zoltán Soóky

---

### **Čestné vyhlásenie**

Vyhlasujem, že prácu stredoškolskej odbornej činnosti na tému ELITE – English Language Interactive Training Experience som vypracoval samostatne, s použitím uvedených literárnych zdrojov. **Prácu som neprihlásil a ani neprezentoval v žiadnej inej súťaži, ktorá je pod gestorstvom MŠM VVaŠ SR. Som si vedomý dôsledkov, ak uvedené údaje nie sú pravdivé.**

.....

### **Pod'akovanie**

Chcel by som sa pod'akovať môjmu konzultantovi práce, Mgr. Zoltánovi Soókymu, za vedenie a cenné pripomienky pri záverečnom spracovaní práce.

# Obsah

Úvod .....	6
1 Problematika a prehľad literatúry .....	8
1.1 Technická stránka problematiky .....	8
1.2 Jazyková stránka problematiky .....	10
1.3 Organizácia dát.....	12
1.4 Grafické rozhranie.....	13
2 Ciele práce .....	14
3 Materiál a metodika .....	15
3.1 Technická zložka.....	15
3.1.1 Knižnice.....	15
3.1.2 Databáza .....	18
3.2 Jazyková zložka.....	18
3.2.1 Flashcards .....	21
3.3 Grafické rozhranie.....	22
4 Výsledky a diskusia .....	23
5 Závery práce .....	25
Zhrnutie .....	27
Resume .....	28
Zoznam použitej literatúry.....	29

## Úvod

Najpodstatnejším dôvodom, prečo sme sa rozhodli vytvoriť náučnú aplikáciu, bola naša dlhodobá láska pre jazyky, hlavne pre angličtinu a preto nám bolo od začiatku jasné, aký smer náš projekt naberie.

Od malička sme mali výborný vzťah k jazykom, vždy nám dobre išli a keď sme sa chceli naučiť základné frázy v cudzom jazyku, nikdy nám to nerobilo problém. Veľmi často sme sa zúčastňovali jazykových súťaží, ako je napríklad olympiáda z anglického jazyka a aj napriek tomu, že nie vždy sme sa umiestnili najlepšie, dalo nám to nesmierne veľa skúseností, ktoré sme mohli neskôr využiť ako aj v reálnom svete, tak aj v našej práci. Avšak okrem programovacích znalostí, človek na vytvorenie náučnej aplikácie potrebuje aj určité jazykové zručnosti, ale tým, že sme ešte pred začiatkom získali medzinárodný certifikát na úrovni C2 po úspešnom ukončení skúšok v Nitre, bol tento krok pre nás už od začiatku splnený. Tieto znalosti boli kľúčové na to, aby sme vedeli kde vôbec začať. Trh, čo sa týka podobných aplikácií, nie je zrovna obšírny, okrem pár výnimiek, ako je napríklad svetovo známe Duolingo, ktoré umožňuje používateľom naučiť sa aj oveľa viac jazykov ako je len spomínaná angličtina. V Duolingo máte na výber z rôznych úrovní, menšie tematické celky, úlohy mierené na rôzne zručnosti, lenže čo sa nám osobne nepáči to, že čím ďalej tým viac je jednak Duolingo, ale aj väčšina aplikácií nesmierne komerčná. Medzi ďalšie patrí jedno z našich obľúbených, Anki, ktoré funguje na overenom systéme "flashcards", čo znamená, že používateľ si ako keby ťahá karty ktoré obsahujú slovíčka a pravidelne tie slovíčka preskúša, ale aplikáciu riadi len používateľ a nie je absolútne automatizovaná. Obe aplikácie sú podľa nás výborné, avšak majú svoje muchy, ako je napríklad potrebný prístup k internetu, alebo sťahovanie údajov, bez ktorých by aplikácie nefungovali. Preto sme sa rozhodli spojiť tie dobré časti z oboch aplikácií a vytvoriť našu vlastnú aplikáciu ELITE, kde proces preskúšania je automatizovaný, má presne definované jazykové úrovne a používateľ si nemusí sám hľadať úlohy. Okrem iného, na takýto projekt si človek samozrejme potrebuje vybrať aj programovací jazyk a knižnice, ktoré bude počas tvorby využívať na rôzne účely. My osobne sme mali rozhodnutie medzi Javou a Pythonom, ale nakoniec sme sa rozhodli pre Python, kvôli nadmernému množstvu knižníc, bez ktorých by sa pracovalo oveľa ťažšie. Samozrejme, takáto aplikácia sa dá vytvoriť aj v iných programovacích jazykoch ako je napríklad aj jazyk C, alebo spomínaná Java, taktiež sa dá podobná aplikácia vytvoriť na mobilné zariadenia pomocou Swift alebo Ruby, dokonca sa dá vo forme webovej stránky pomocou HTML, PHP a CSS, ale my sme od začiatku chceli čo najlepšiu možnú verziu na

desktop a preto sme si vybrali Python. Angličtina, ako každý iný jazyk má svoje vlastné zručnosti, potrebné znalosti a typy skúšok, preto sme si museli dôsledne premyslieť, aké knižnice použiť, ako sa na problematiku zamieriť a kde vlastne začať. Typov skúšok existuje nesmierne veľa, ako aj medzinárodne uznávané, tak aj pre jednotlivé krajiny. Taktiež, tieto jednotlivé skúšky môžu mať úplne iné systémy a preto bolo pre nás vybrať si jeden strašne ťažké. Ako sme už spomínali, máme certifikát na úrovni C2 a to konkrétne certifikát Cambridge, podľa ktorého sme samozrejme tvorili aj úlohy a predlohy. Nie sme prví, ktorí by sa pokúšali spojiť informatiku v tejto podobe a výučbu jazyka do jedného uceleného projektu a teda priebeh vývoja môže byť veľmi odlišný.

Hlavným cieľom našej práce je teda vytvoriť didaktickú desktopovú aplikáciu vo forme zbierky úloh, ktorá užívateľom z rôznych pozadí a kútov sveta umožní a bude podporovať učenie anglického jazyka vo forme elektronickej zbierky úloh.

# 1 Problematika a prehľad literatúry

Problematika ohľadom tvorby a programovania náučnej didaktickej desktopovej aplikácie je veľmi obširná a komplikovaná. Ako už bolo spomenuté v úvode, úplne prvé rozhodnutie je vybrať si a hlavne ten správny, programovací jazyk, ale tým, že ich v dnešnej dobe existuje strašne veľa, je tento výber veľmi komplikovaný. Taktiež, čo rozhodovanie veľmi silno ovplyvňuje, je výber tej ideálnej platformy a forma aplikácie.

Rozhodnúť sa správne a v správny čas je taktiež kľúčová časť našej práce, ako na začiatku s výberom technológií, tak aj priebežne s výberom knižníc a aby naša aplikáciu bola čo najviac optimalizovaná, predsa keď môžeme mať tú istú funkcionálnosť za 10 riadkov kódu alebo 100 riadkov, tak by sme si jednoznačne vybrali čo najmenej. Keby sme sa v jednom z veľkého množstva našich bodov pomýlili, mohlo by to byť znamenať koniec v krátkodobej budúcnosti, čo je možno aj lepšia alternatíva, alebo by sme ďalej pokračovali v tom smere, ale možno v dlhodobej budúcnosti by to znamenalo slepú uličku, čo by bolo oveľa horšie, a znamenalo by to obrovskú stratu času.

## 1.1 Technická stránka problematiky

Jedna z najznámejších didaktických aplikácií je Duolingo, ktorá je samozrejme vo forme webovej aplikácie. Naša téma na komplexnú odbornú prácu je ale desktopová aplikácia a odborná problematika ohľadom týchto dvoch rôznych druhov aplikácií je úplne odlišná. Na aplikáciu ako Duolingo by bolo potrebné použiť neskutočne veľa programovacích jazykov a obnášalo by to úplne iné problémy ako programovať v jednom jazyku, na desktop.

Ďalšia vec na ktorú sme museli dávať pozor je, že koľko programovacích jazykov si vybrať. Predsa, vybrať si zvlášť ideálny jazyk na grafické rozhranie a zvlášť jazyk na programovanie výpočtov. V dokonalom svete, by sa tieto dva komponenty dali bez problémov spojiť a fungovali by v symbióze. Na nešťastie, v reálnom svete to takto nefunguje a asi by to obnášalo viacej problémov ako by to vyriešilo.

Jedna z našich osobných favoritov, čo sa týka aplikácií na učenie jazykov, ktorý sme aj predtým spomínali, sa volá Anki a funguje na spôsob takzvaných “flaschards”. Sú to kartičky ktoré obsahujú na jednej buď slovo, alebo vetu, alebo viac menej hocičo, čo by ste si vedeli predstaviť a na druhej strane riešenie alebo odpoveď. Funguje na časovom princípe, kde používateľ má na výber zo štyroch políčok, podľa toho ako vedel danú kartu, ak ju vedel



perfektne, tak klikne na vhodné tlačidlo, tak sa vyradí z poradia a neobjaví sa v ponuke. Naopak, keď používateľ kartu nevedel vôbec, tak klikne na úplne iné tlačidlo a zaradí sa znova, čo znamená dvakrát takú šancu, že si ju používateľ vytiahne. Takto, používateľ dostane tie karty ktoré nevie a môže si všetko preskúšať podľa ich potreby. Jedna obrovská nevýhoda tohto systému je, že spomínané kartičky nevytvoril autor, je to spracované komunitou a preto možno práve pre tému ktorú potrebujete karty neexistujú. Samozrejme, môžete si spraviť svoje, ale je to časovo veľmi náročné a ťažko sa s tým pracuje.

Ak by človek chcel skôr vytvoriť aplikáciu podobnú Anki, tak by to samozrejme obnášalo úplne iné požiadavky ako napríklad také Duolingo. Každé má svoje výhody a nevýhody, pri Anki by bola oveľa väčšia požiadavka na technickú stránku, alebo teda na programovanie a možno menej na grafické rozhranie, kde pri Duolingo by sa muselo dbať na grafické rozhranie.

Keby sa človek rozhodne vytvoriť aplikáciu podobnú Duolingo, to znamená že by bola viac vo forme hry, programátor by si skôr vybral jeden z jazykov C, ktoré sa oveľa častejšie používajú na programovanie hier, ako náučných aplikácií. Naopak, keby sa niekto rozhodne pre tvorbu mobilnej aplikácie, alebo aplikácie na webovej stránke, obnášalo by to úplne iné problémy a prekážky. My, z dôvodu, že sme chceli aplikáciu vytvoriť pre desktop a že sme chceli spojiť najlepšie časti z našich obľúbených aplikácií, sme si vybrali Python, lebo je najlepšie vybavený programovací jazyk pre naše účely.

„Python je populárny programovací jazyk. Vytvoril ho Guido van Rossum a bol vydaný v roku 1991. Používa sa na: vývoj webových stránok (na strane servera), vývoj softvéru, matematiku a systémové skripty.“(1)

Python, ako programovací jazyk je jeden zo starších jazykov, vyvinutý najprv v roku 1991, ale je jeden z najlepších, najpoužívanějších a najviac užitočných čo sa týka vlastne všetkých záležitostí. Kým jazyky ako C alebo Java sú možno viac sústredené na pár účelov, v Pythone sa dá naprogramovať všetko čo si viete predstaviť. Obrovská výhoda Pythonu je, že má množstvo knižníc, ktoré sa dajú využiť pre hocijaký účel. Nech sa jedná o 2D alebo 3D hru, didaktickú aplikáciu ako je naša, alebo aj tvorba animácií, možností je nekonečne veľa. Malý faktor, ktorý nám taktiež uľahčil voľbu, bol fakt, že jazyk Python sme preberali v prvom ročníku strednej školy, čiže nám bol bližší ako iné alternatívy.

## 1.2 Jazyková stránka problematiky

Samozrejme, toto je len problematika z technického uhla a takýto projekt obsahuje viacero stránok, jedna z najvýznamnejších je tá jazyková. Angličtina je veľmi unikátny jazyk, ale je veľmi odlišný od slovenčiny a aj od viacerých jazykov, ako je napríklad nemčina, v tom, že tam skloňovanie a rody ako také, neexistujú. Používajú sa takzvané "gender neutral" slová, čo znamená že každodenné slová ako pes alebo auto nemajú žiadne pridružené rody.

Pre bežného človeka, ktorý sa rozhodne naučiť sa ľubovoľný jazyk, bude v ceste stáť neskutočne veľa stien a anglický jazyk sa v tomto vôbec neodlišuje od ostatných jazykov. Od úplných začiatkov, kde sa musíme naučiť „irregular verbs“, alebo teda slovesá ktoré majú nepravidelné časovanie, modálne slovesá, podmienky, až po komplikované koncepty ako zaužívané frazeologizmy, angličtina je neskutočne bohatý jazyk.

„Angličtina je jedným z najjednoduchších a najľahších prirodzených jazykov na svete. Samozrejme, všetko je relatívne. Závisí to od schopností študenta a jeho predchádzajúcich skúseností s učením sa jazykov. Napriek tomu je možné povedať, že angličtina je relatívne jednoduchý jazyk na učenie, porozumenie a hovorenie v porovnaní s veľmi zložitými jazykmi, ako sú arabčina, kantončina, mandarínčina, kórejčina a japončina. Anglický jazyk používa latinskú abecedu.

Je to najuniverzálnejšia, najkratšia a najjednoduchšia abeceda. Ďalšou výraznou charakteristikou angličtiny je (zvyčajne) pevný poriadok slov. Väčšina anglických viet (klauzúl) zodpovedá slovosledu SVO. Výslovnosť anglických slov ako this, thin, clothes, thirteenth, months nevyhnutne spôsobuje problémy žiakom, ktorí nepotrebujú používať špičku jazyka na tvorenie slov vo svojom jazyku.“(3)

Tým, že je taktiež jeden z najstarších a najviac využívaných na celom svete, existuje mnoho dialektov, nárečí, alebo aj neslávne rozdiely v spisovnosti. Rôzne kúty sveta používajú rôznu angličtinu, v Amerike sa použije slovo „color“, kým vo Veľkej Británii sa použije „colour“. Tieto rozdiely sú kvôli historickým spojením medzi Francúzmi a Britmi. Amerike sa tieto zbytočné samohlásky nepáčili a preto ich takto jednoducho odstránili. Austrália napríklad, používa úplne inú slovnú zásobu od zvyšku sveta a často sa stane, že môžeme síce rozumieť úplne v pohode angličtine, v Austrálii by sme sa tak či tak stratili. Toto isté platí aj pre Afriku alebo Latinskú Ameriku, kde tie nárečia vedia byť neskutočne namáhavé.

Čo sa týka skloňovania, angličtina sa k tomuto najviac približuje svojou verzou, čo je veľmi podobné slovenskému časovaniu slovies. Tým, že angličtina obsahuje veľmi veľa "časov", je celková štruktúra viet a textov úplne iná a komplikovanejšia, ako sme zvyknutý v slovenčine. Najťažšia časť angličtiny pre mnohých ľudí, je ale výslovnosť, ktorá je extrémne unikátna a žiadny jazyk sa k tomu ani nevie priblížiť.

Existuje mnoho zručností v každom jazyku a veľmi ťažko sa to dá usporiadať do kategórií, ale pre účel aplikácie tohto typu, sme si museli roztriediť úlohy na väčšie kategórie. Jednoznačne, existuje mnoho viacerých podkategórií, ktoré sme si ale mysleli, že sú menej podstatné. Zručnosti ktoré sme si my vybrali sú taktiež najzvyčajnejšie požadované na jazykových skúškach z angličtiny. Patrí sem čítanie s porozumením, kde máte text s určitou dĺžkou podľa úrovne, otázky k danému textu a možnosti, prípadne vlastná odpoveď. Ďalšia zručnosť je počúvanie, ktorá síce funguje podobne ako čítanie, no s takým rozdielom, že namiesto čítania textu a ľubovoľne dostupného času máte počúvanie nahratého textu a musíte v časovom rozmedzí odpovedať na otázky. Medzi hlavné zručnosti samozrejme patrí aj písanie slohovej práce, čo je nevyhnutné na osvedčenie sa, že vieme poriadne používať jazyk ako taký, rozprávanie, ktorý má nesmierne veľa podôb, od opisu obrázka, po vytvorenie príbehu na mieste, taktiež by som sem zaradil čítanie nahlas. Poslednou zručnosťou ktorú sme zvolili bol takzvaný „Use of English“, čo znamená niečo ako používaný jazyk. Museli sme sa taktiež uistiť, aby úlohy neboli len na jedinej úrovni a taktiež boli dostatočne obsiahne, čo sa týka obtiažnosti, predsa, keby používateľ na pokročilej úrovni dostáva úlohy pre začiatočníka, aplikácia by nebola veľmi funkčná.

Na docielenie vyváženého didaktického a náučného zážitku sme samozrejme museli pridať aj vymoženosť na vylepšovanie slovnej zásoby. Táto problematika je aj v reálnom svete veľmi náročná, vzhľadom na to, že neexistuje jeden spôsob ktorý by vyhovoval každému. Možnosť ako na toto zamieriť je veľmi veľa, študenti asi budú zvyknutý na neskutočne často uplatňovanú metódu „slovíčiek“, kde pani profesorka resp. pán profesor zadá pár desiatok slov, ktoré sa študent musí naučiť a neskôr z nich bude vyskúšaný. Tento spôsob je dosť dobrý, lenže v prostredí aplikácie ťažko realizovateľný, kvôli absencii učiteľa, alebo kontrolnej osoby. Ďalšia možnosť a už naša výherná, bola metóda tzv. „flashcards“, čo sú vlastne kartičky, kde na jednej strane máme slovo po slovensky a na druhej po anglicky.

Tento koncept je veľmi často využívaný v prostredí školy, najlepšie v malých skupinách, kde sa študenti môžu navzájom preskúšať. Samozrejme, táto aplikácia je určená

pre jedného používateľa a tak túto možnosť sme nemali. Toto sme ale nahradili vymoženosťami ako sú obrázky a ukážka výslovnosti. Taktiež sme pridali tlačidlá, pomocou ktorých používateľ vie tento systém ovládať.

### 1.3 Organizácia dát

Ďalšia prekážka, ktorá sa začiatčníkom možno ani nezdá, je organizácia dát, aby bol náš program čo najviac prehľadný. V tejto časti, by dáta znamenali úlohy, ktoré slúžia na preskúšanie predom spomínaných jazykových znalostí. Možnosti boli rôznorodé, ako napríklad zakotviť to do kódu, ale potom by sa naša aplikácia nedala rozvíjať v budúcnosti, nebola by škálovateľná a tomuto sme sa veľmi chceli vyhnúť. Ďalšia a už viac rozumná možnosť, bola vytvoriť databázu na počítači, či už na virtuálnom alebo fyzickom a spravovať dáta tak.

Táto možnosť už bola viac sofistikovaná, no stále mala tú chybu, že počítač na ktorom bola databáza uložená, musel byť stále zapnutý, inak by nám bola databáza zbytočná. Nakoniec sme prišli k riešeniu, čo bola služba "Aiven", ktorá nám sprístupnila online hosting na databázu, čiže nemuseli sme sa starať o správu databázy, iba by sme tam pridávali dáta, a pristupovali k nim.

Samotná správa databázy väčšinou prebieha v jazyku SQL (Structured Query Language), alebo MySQL, čo sú vyhradené jazyky pre databázy. My osobne, keďže sme sa cítili veľmi komfortne v prostredí Pythonu, sme to robili cez Python. Rozdiel nebol veľký, jednoducho sme zadávali príkazy do Python kódu, namiesto terminálu pre databázy.

„MySQL je veľmi populárny systém na správu relačných databáz (RDBMS) s otvoreným zdrojovým kódom. Čo je MySQL? Je to relačný systém správy databáz, je open-source, je zadarmo, je ideálny pre malé aj veľké aplikácie, je veľmi rýchly, spoľahlivý, škálovateľný a ľahko sa používa“(2)

Vytvoriť databázu tak, aby bola primeraná pre naše účely, je o dost' väčší problém. Stĺpec v databáze a v jazyku MySQL umožňuje veľmi veľa premenných, avšak, naša najviac používaná "varchar", ktorá sa používa pre text, má maximálnu veľkosť 255 znakov, ktorá pri dlhších textoch potrebných pre úlohy ako je čítanie s porozumením jednoducho nestačia. Na druhej strane, nechceli sme aby tá databáza bola obrovská, kde by možno obsahovala 5 alebo viac takýchto políček, čiže pre každú kategóriu úloh, sme si museli zvlášť stanoviť tabuľky.

## 1.4 Grafické rozhranie

Posledná a asi aj najnápadnejšia prekážka, je grafické rozhranie programu, aby aplikácia bola prívetivá pre používateľov. Chceli sme, aby aplikáciu mohol používať každý človek bez výnimky, preto pre nás táto časť bola jedna z najťažších, ale zároveň aj najviac naplňujúcich.

Tým že sme sa rozhodli použiť programovací jazyk Python a spájať viacero jazykov dokopy by bolo možno na ujmu technickej časti, museli sme si vybrať z ponuky knižníc, ktorých je síce veľmi veľa, no vyzerajú a fungujú celkom podobne.

Knižnica ktorú sme sa učili v škole sa volá Tkinter. Slúži na tvorbu viac menej všetkého, tvary, jednoduchšie animácie, kreslenie, no skratka dá sa tam robiť veľmi veľa vecí. Jeden malý problém je, že čo sa týka možností tlačidiel, alebo iných prvkov, ktoré by sme chceli použiť, tak sú veľmi staromódne a vyzerajú ako typický program z 90. rokov.

Ďalšia knižnica ktorú sme mohli použiť, je PyQt5 ktorá oproti Tkinter obsahuje v sebe viacero doplnkov a viacero modulov. Funguje podobne ako napríklad Microsoft Visual Studio, na báze takzvaného “Drag and Drop”, kde si programátor môže kliknúť napríklad na text a umiestniť ho hocikam a iba neskôr naprogramuje. Bola to síce lákavá možnosť, ale báli sme sa troška funkcionality a či by sme vedeli spraviť všetko čo sme chceli.

Nakoniec knižnica ktorú sme si vybrali sa volá Custom Tkinter, ktorá ako podľa mena sa dá zistiť, je vybudovaná na Tkinter. Obsahuje všetky vymoženosti Tkinter, ale taktiež tam pridá viacero modulov a oveľa väčšiu flexibilitu, čo sa týka jednotlivých modulov. Najväčšia výhoda je, že sa s tým dá vytvoriť moderné grafické rozhranie, čo je presne to čo sme chceli.

Čo sa týka podobných aplikácií, veľmi málo z nich má také možnosti čo sa týka dostupnosti ako naša. Ak je používateľ hluchý, máme na to zvlášť nastavenie. Taktiež máme nastavenia pre slabo vidiacich, alebo farboslepých na konkrétnu farbu. Samozrejme, nemohli sme vynechať nemých ľudí, pre ktorých sme preformátovali úlohy. Takto sme zabezpečili, aby každý používateľ mal nastavenie, ktoré im poskytuje tú najlepšiu skúsenosť. Chceli sme taktiež zabezpečiť, aby náš program vyzeral moderne, keďže niektoré aplikácie na dnešnom trhu vyzerajú, aspoň podľa nás, neprijemne.

## 2 Ciele práce

Stanoviť si ciele pri práci na takejto škále je nesmierne ťažké, ale taktiež veľmi dôležité. Nemôžeme si ich stanoviť príliš nízko, lebo potom by naša aplikácia absolútne nevynikala a stala by sa ďalšou potenciálne dobrou, no v realite zabudnutou aplikáciou. Na druhej strane sme taktiež nemohli mieriť veľmi vysoko, lebo ku koncu by sme prišli na to, že síce by naša aplikácia mohla mať viac vymožeností, no ani jedna by nefungovala na dokonalej úrovni. Museli sme si vybrať zlatú strednú cestu a uprednostniť kvalitu nad kvantitou.

Naša komplexná odborná práca by mohla spĺňať mnoho väčších cieľov, ale pre nás bol od začiatku hlavný cieľ jasný. Chceli sme vytvoriť vzdelávaciu aplikáciu, ktorá by nebola tak silne komerčná ako Duolingo, ale taktiež aby umožňovalo oveľa viac možností ako Anki, i keď sú tieto aplikácie podľa nás excelentné.

Náš projekt mal dve veľké ciele alebo časti, a to vytvoriť logickú časť a prezenčnú časť, čiže grafické rozhranie. Dosiahnuť tento cieľ sa nám podarilo pomaly, pomocou čiastkových úloh, ktoré nám krásne od začiatku diktovali tempo práce a aj trajektóriu projektu.

Prvým našim väčším bodom bola logická časť práce, kde sme sa uistili, že to, čo chceme vytvoriť, je vlastne možné. Toto zahŕňa aj vyhodnocovací algoritmus ktorý porovná vašu odpoveď so správnou a podľa potreby uzná ako vhodnú alebo nevhodnú. Ukáže aj správnu odpoveď, výber úloh, rozdelenie a rozčlenenie úloh na predom spomínané kategórie, aby sa používateľ mohol preskúšať v rôznych zručnostiach a spojenie všetkých týchto častí dokopy, aby nám aplikácia, na určitej miere, fungovala. Neskôr, bol bod druhý, a to grafické rozhranie, kde sme z aplikácie ktorá sa prezentovala len v termináli a pomocou klávesnice, bez použitia myši, spravili plnohodnotnú aplikáciu, v ktorej by sa vedel orientovať aj každodenný používateľ. Koniec koncov, chceli sme vytvoriť jednoduchú, no zároveň komplexnú didaktickú aplikáciu, ktorá by bola priateľská pre každého a podporovala by vzdelávanie anglického jazyka.

### **3 Materiál a metodika**

Na spomínané problémy a dilemy samozrejme existuje mnoho ciest a taktiež mnoho cieľov. Čo sa týka technickej stránky, vybrať si správny programovací jazyk, knižnice, objekty a procesy má obrovský význam pre nás, keďže chceme maximalizovať čo sa len dá. Podobne, ani jazyková stránka nie je o nič jednoduchšia, vybrať si správne úlohy, zabezpečiť bezchybovosť a správny bodový systém nie sú jednoduchou hračkou.

#### **3.1 Technická zložka**

Samozrejme, tým že naša učebná pomôcka má podobu desktopovej aplikácie, sme museli najprv spraviť samotný program, bez zabudnutia na jazykovú zložku. Dlhو sme váhali ktorý programovací jazyk si vybrať, lebo sme vedeli akú obrovskú váhu toto rozhodnutie bude mať počas priebehu celej práce. Nakoniec sme sa rozhodli pre Python, a to kvôli jednoduchému dôvodu, že Python čo sa týka knižníc je bezkonkurenčný. Má takú ponuku na ktorú sa ostatné jazyky len žiarlivo pozerajú a taktiež sme sa ho učili v prvom ročníku na strednej škole, čiže sme už boli ako tak zoznámený s daným jazykom.

Priebeh vývoju našej aplikácie je možno pre niektorých trochu nezvyčajný, veď predsa postupovať po malých častiach by bolo jednoduchšie ako si aplikáciu takto rozdeliť na logickú časť a grafické rozhranie, kde by mohli vzniknúť nezrovnalosti medzi filozofiami. Tohto sme sa celkom obávali, ale programovali sme konzistentne a táto obava nakoniec pre nás nič neznamenala. Naopak, kvôli tomuto sme sa vedeli uistiť, že to čo chceme spraviť je dosiahnuteľné a realizovateľné. Komponenty ako bodový systém, vyhodnocovací algoritmus a automatické ukladanie boli najprv odskúšané v tejto časti procesu.

Aplikácia v tej forme ako bola vtedy, by sa teraz pravdepodobne usúdila ako nefunkčná, z toho dôvodu, že mala nulovú interaktivitu a vstup bol len z klávesnice. Ukázalo nám to, že s čím máme do budúcnosti rátať a s čím nie, čo sa týka vymožeností pre aplikáciu.

##### **3.1.1 Knižnice**

Ako sme spomínali, Python ako programovací jazyk, má veľmi široký výber knižníc a pre nás to znamenalo, že aj na jeden cieľ sme mohli použiť viaceré. Nakoniec sme sa rozhodli použiť tieto knižnice.

Pre grafické rozhranie sme zvolili knižnicu CustomTkinter, ktorá je nadstavba na veľmi obľúbenú a využívanú knižnicu Tkinter, s tým rozdielom, že CustomTkinter narozdiel od Tkinteru nám ponúka možnosť vytvoriť modernú grafiku a prostredie pre používateľa,

aby mohol s našou aplikáciou interagovať. Na načítanie obrázkov, kvôli obohateniu časti flashcards, sme použili knižnicu pillow, z dôvodu, že výborne spolupracuje s CustomTkinter a tým sme sa vyhli problému ohľadom compatibility.

Na listening, alebo po slovensky počúvanie s porozumením, čo sa skladá z textu ktorý je prečítaný a otázok ohľadom prečítaného textu, sme potrebovali knižnicu ktorá nám umožnila prehrať text ako zvukový súbor. Existuje mnoho spôsobov, najpraciejšia by asi bola možnosť, kde sme si sami nahlas prečítali a nahrali text, čo by síce znelo neskutočne dobre, ale tie súbory by mali obrovskú veľkosť a taktiež by sa ťažko integrovali do databázy. Nakoniec sme sa rozhodli použiť modernú technológiu „Text to speech“, ktorá imituje ľudský hlas, síce to má svoje muchy a stáva sa, že položí váhu na zlé slovo, okrem týchto maličkých chýb funguje bez problému a je oveľa jednoduchší na integráciu. Použili sme knižnicu Google Text to Speech, ktorá je ako už meno napovedá vytvorená Googlom a využíva ten istý model ako samotný Google.

Týmto sme ale mali len samotný zvukový súbor a potrebovali sme taktiež spôsob ako to prehrať v aplikácii. Existuje knižnica playsound, lenže má taký problém, že keď beží zvukový súbor, aplikácia zamrzne a používateľ si nemôže vhodné odpovede vyberať. Presne kvôli tomuto sme sa ale rozhodli zvoliť knižnicu Pygame, čo je síce určená na tvorbu hier, ale má zložku „Mixer“, ktorá je pre nás ideálna. Tým, že aplikácia ďalej funguje ako má a vieme podľa vôle zastavovať a spúšťať prehrávanie zvuku, je táto knižnica ideálna.

Taktiež, na podobný účel sme použili knižnicu Pytsx3, ktorá je oveľa viac integrovaná s Pythonom a nepotrebuje prístup na internet, lenže má úplne tú istú chybu ako playsound, že aplikácia zamrzne. Pre naše účely, je toto síce v hlavnom programe nepoužiteľné, ale v časti flashcards už o toľko viac. Tam predsa, používateľ nebude nič robiť kým sa prehráva zvuk.

Problematika speaking-u, alebo rozprávania, je nesmierne široká a v prostredí aplikácie sa veľmi ťažko realizuje. Táto zložka je viacmenej nemožná bez prístupu na internet, to kvôli tomu, že reč môže byť veľmi rôznorodá a museli sme taktiež brať do úvahy vonkajšie faktory ako šum pozadia. My sme sa rozhodli použiť knižnicu Speech Recognition a to preto, že ponúka viaceré služby rozpoznávania hlasu, napríklad aj od Googlu. Pomocou tohto už sme len museli text dôrazne spracovať a neskôr prirovnať k tomu, čo používateľ musel prečítať.



Čo sa týka writing-u, alebo písanie slohových prác, je podobne ako speaking, nesmierne ťažká. Čo sa deje keď rozprávame, sa dá interpretovať aj do svetu písania. Ostáva tam tá istá subjektivita, lebo dvaja rôzni ľudia nikdy nenapíšu ten istý sloh. Museli sme rozmýšľať trochu nekonvenčne a vytvoriť algoritmy na hodnotenie slohových prác. Kontrolovať počet slov nebol veľký problém, ale to bola naša jedna výnimka.

Tematika slohu je niečo, na čo neexistuje ani do dneska dokonalé riešenie. My sme sa rozhodli zvoliť možno trochu laický spôsob, že sme mali jednu tému a na tú tému sme si vybrali 50 slov, ktoré sa často spájajú s danou témou. Keď používateľ dopísal sloh, tak by sme to slovo po slove prirovnávali a keď sa zhodovalo aspoň 10% práce s danou tematikou, vtedy sme začali dávať body za tematiku. Gramatika, našťastie bola oveľa jednoduchšia, na to existuje knižnica Language Tool Python, ktorá pomocou Java vie vytknúť gramatické chyby a taktiež ponúkne možnú náhradu.

Spomínali sme zhodu aj pri rozprávaní aj pri slohoch, ale situácia, kde sa dve slová úplne zhodujú je vzácna. Boolovská logika vyhodnocuje len 0 a 1, čiže nepravda alebo pravda a neberie do úvahy nič medzi. Knižnica Fuzzywuzzy nám umožnila použiť Fuzzy logiku, ktorá už berie do úvahy aj desatinné čísla medzi 0 a 1. Za zhodu sme uznali dve slová, keď mali hodnotu nad 75%, alebo 0,75. Týmto jednoduchým krokom sme taktiež eliminovali potrebu kontrolovať iné tvary rovnakých slov, ako je napríklad plurál, alebo časovanie.

Ukladací systém je pre každú aplikáciu dôležitá vec a tam sa naša aplikácia nelíši. My sme sa rozhodli tento proces spraviť automatizovaný a na to sme potrebovali knižnicu os.path, aby sme mohli pristupovať k vytvoreným textovým súborom, kde sa ukladali body, úroveň, zvolené nastavenia a podobne.

Ďalšia knižnica ktorá nám pomohla s pristupovaním k súborom je linecache, pomocou ktorej sme vedel čítať a samozrejme aj prepisovať celé riadky textových súborov kde sme to potrebovali, namiesto len jedného čísla.

Samozrejme, na vyváženú skúsenosť sme potrebovali zaistiť to, že používateľ dostane odlišné úlohy a nie len tie isté dookola. Toto sme docielili pomocou knižnice random, ktorá nám umožnila náhodne vybrať z usporiadaných čísiel len jednu a v prípade splnenia úlohy, taktiež spraviť to, aby dané číslo sa už znova nevybralo.

Na ukladanie úloh a rôznych údajov sme samozrejme použili štruktúrovanú databázu, ktorá by nám sama o sebe veľmi nepomohla, ale pomocou knižnice mySql Connector, sme vedel zabezpečiť prepojenie našej aplikácie a databázy, ktorú sme vytvorili.

Mohli sme zvoliť opis obrázka, kde by sme ale mali obrovský problém kontrolovať to, čo používateľ povedal. Tvorba príbehu by narazila na podobný koniec tým, že neexistuje objektívne hodnotenie na takéto úlohy.

### **3.1.2 Databáza**

Databáza, vo svete technológie, je dôležitý komponent pre každý projekt, nech je to aplikácia, webová stránka, alebo aj portály. Pre nás bolo vytvorenie a správa databázy nevyhnutným krokom, keďže sme nechceli mať úlohy a údaje ukladané lokálne, lebo by to pre nás znamenalo obrovskú veľkosť programu, čo sme samozrejme nechceli.

My ako programátori sme približne vedeli čo od nej vyžadujeme a preto sme si museli dobre vybrať. Existuje mnoho spôsobov, ako je databáza na vlastnom počítači, na vlastnom serveri, alebo online hosting databázy. Online hosting má ale takú nevýhodu že väčšinou nie je zadarmo a aj keď je, tak je buď nízkej kvality, alebo sa tam vôbec neberie do úvahy zabezpečenie. Pri najlepšom prípade, dostanete aj dobre zabezpečenú databázu, ale je limitovaná veľkosť. Nakoniec sme ale našli službu ktorá sa volá Aiven, ktorá ponúka jednu databázu zadarmo do 5gb, čo pre nás bolo ideálnym riešením.

V našej databáze sme chceli aby všetko bolo pekne oddelené, aby sme k nim neskôr mohli bez problémov pristupovať. Prihliadajúc na toto, sme si databázu rozdelili na jednotlivé tabuľky, jedna na každú úlohu, čiže dokopy 5, plus ešte tabuľky na flashcards.

Naša databáza je uzatvorená a to preto lebo sme chceli zabezpečiť bezchybovosť úloh. Niektorí si možno povia, že by bolo super mať možnosť tam pridať ešte viac úloh, napríklad pre učiteľa, ale jednoducho chceli sme si dať záležať na vybraných úlohách a myšlienka, že by to bolo nejakým spôsobom upravené, sa nám osobne nepáčila.

## **3.2 Jazyková zložka**

Hlavným záujmom našej aplikácie je ale samozrejme jazyk a to konkrétne angličtina. Ciest ako sa naučiť po anglicky existuje mnoho, od staromódnych kníh, personálnych lektorov, až po moderné webové stránky a aplikácie. Knihy alebo učebnice sa možno už v tejto dobe považujú za staromódny spôsob učenia, ale je stále nesmierne dobrý. Je to niečo fyzické, čo sa dá chytiť a podľa nás neexistuje lepší spôsob zapamätania si vecí ako je niečo fyzicky chmatať. Má to ale jednu veľkú nevýhodu, knihy niekedy môžu stáť dosť veľa

peňazí a po čase sa ich stav zhoršuje. Digitálna forma učenia sa čím ďalej tým viac popularizuje a my sme si ju tiež zvolili kvôli prístupnosti, kde používateľ si len musí stiahnuť inštaláciu, aplikáciu nainštalovať a už len používať. Taktiež tam degradácie po čase nehrajú žiadnu rolu, kým u knihách je to jeden z najväčších faktorov.

Samozrejme, prvým krokom pre aplikáciu ako našu, musí byť to, že zistíme na akej jazykovej úrovni používateľ je. Toto sme docielili takzvanej inicializačnej skúšky, čo je test, ktorý sa spustí len pri prvom spustení aplikácie. Museli sme sa uistiť, že ten test je čo najviac objektívny ako je len možné, ale stále dostatočne krátky aby to nebolo otravné. Kvôli tomuto sme tam dali 2 úlohy z každej zručnosti, na rôznych jazykových úrovniach. Bodový systém v tomto teste bol taký, že za každú úlohu je maximálne 20 bodov a tým že je 10 úloh, maximálny počet bodov je 200. Máme 4 úrovne dokopy, čiže naša stupnica sa rozdeľovala o 50 bodov. Taktiež, sme pridali tlačidlo na preskočenie tejto skúšky, ak sa niekto rozhodne, že aj tie najjednoduchšie úlohy sú pre neho ťažké, alebo jednoducho na to nemá čas.

Keď používateľ ukončil našu inicializačnú skúšku, dostane sa k oknu výberu úloh, kde sa mu zobrazí 5 okien s rôznymi úlohami, taktiež môže vidieť body a chyby, ešte ale žiadne nezískal. Týchto 5 okien je: Reading, Listening, Speaking, Writing, Use of English

My sme si tieto typy úloh vybrali z dôvodu, že sú nám najbližšie a taktiež sú to najčastejšie vyžadované zručnosti na jazykových skúškach. Samozrejme, existujú kombinované typy úloh, ale tým že naša aplikácia je určená ako pomôcka, sme si nechceli zbytočne komplikovať náš vlastný život.

Reading, alebo čítanie s porozumením, môže mať veľmi veľa podôb. Najčastejšie sa využíva podoba, kde máte zadaný text ktorý musíte prečítať a taktiež k tomu textu máte pár otázok, spolu s odpoveďami na výber. V podobe aplikácie to vyzerá tak, že máme obrazovku rozdelenú na dve časti, jedna s textom kde používateľ môže samozrejme aj scrollovať aby si prečítal celý text a na druhej časti sú otázky s možnosťami.

Listening, alebo počúvanie s porozumením, funguje celkom podobne ako čítanie, len s takým rozdielom že namiesto textu ktorý musí používateľ prečítať, má tlačidlo, na ktoré keď klikne tak sa mu začne prehrávať zvukový súbor, na ktoré sú potom samozrejme aj položené otázky. Tento spôsob sa nám páčil, lebo sme nemuseli rozdeľovať obrazovku na dve časti a zmestilo sa nám tam viac.

Speaking, alebo rozprávanie, má mnoho možných verzií, ktoré sme už spomínali, lenže z dôvodu, že toto je aplikácia a nie samostatný učiteľ, sme museli trochu premýšľať ako by sa to dalo spraviť čo najviac objektívne, ale taktiež funkčné. Rozhodli sme sa to spraviť vo forme čítanie nahlas a to kvôli tomu že to potrebuje úplne iné zručnosti ako čítanie a poriadne rozprávanie sa aj tak najlepšie trénuje s inými ľuďmi a nie s aplikáciou. Používateľ klikne na tlačidlo, potom prečíta text do mikrofónu a naša aplikácia vyhodnotí ako presne daný text používateľ prečítal.

Writing, alebo písanie slohových prác, je ďalšia veľmi náročná časť pre nás, z podobných dôvodov ako rozprávanie. Tým, že toto je aplikácia a nie reálny človek, nemáme šancu spraviť plne objektívne hodnotenie. Taktiež, písanie môže mať mnoho podôb, ale s týmto až taký problém nebol, lebo sme to vedeli generalizovať do jedného. Naše riešenie vyzerá tak, že používateľ dostane čas, počet slov a tematiku okolo ktorej musí napísať essay.

Use of English, po slovensky niečo ako „použitý jazyk“, bola jedna z najľahších pre nás a to kvôli tomu, že väčšinou tieto úlohy majú veľmi podobnú formu. Máte buď jednotlivú vetu, alebo celkový text, kde musí používateľ do vynechaného miesta doplniť slovo, alebo frázu. V našej aplikácii to vyzerá veľmi podobne, kde používateľ dostane vety s vynechanými miestami a taktiež dostane na výber z troch možností aby doplnil na prázdne miesto.

Keď používateľ dosiahne 5 bodov, čo je maximum, z každého typu úlohy, môže pristúpiť k „level up“ skúške, na ktorú má dva pokusy. Táto skúška pozostáva z jednej úlohy každého typu, obtiažnosťou približne medzi dvoma úrovňami. Ak používateľ získa nad 60%, tak sa posunie na vyššiu úroveň a ide zase od znova, teraz už ale na vyššej úrovni. Keď používateľ dvakrát za sebou získa skóre pod 60%, tak sa mu vynulujú body a musí znovu spraviť všetky úlohy na tej istej úrovni.

Tieto typy úloh slúžia ako super pomôcka, aby si používateľ viac osvojil jazyk a preskúšal rôzne zručnosti, hlavne keď sa naučil niečo v reálnom prostredí. Samozrejme, toto sú len úlohy a angličtina ako každý iný jazyk pozostáva nie len z gramatických vzorov alebo pravidiel, ale aj zo slovnej zásoby, ktorú sme vyriešili pomocou ďalšieho kroku.

### 3.2.1 Flashcards

Slovná zásoba sa v novom jazyku veľmi ťažko osvojuje, nech to je pomocou akýchkoľvek pomôcok. Asi každý študent je zvyknutý na písomky zo slovíčok, kde pani profesorka resp. pán profesor zadá pár desiatok slovíčok z pracovného zošita, ktoré sa študenti musia naučiť do dohodnutého dátumu. Je to celkom dobrý spôsob podľa nás, hlavne pre ľudí ktorú nadanie alebo záujem o jazyk nemajú, lebo to funguje na mechanickej báze. Sú ale situácie, kde niekoho ten jazyk môže zaujímať, ale aj tak sa musí takto mechanicky učiť tie slová, ktoré niekedy v pracovných zošitoch ani nemusia dávať zmysel. Tuto prichádzajú do hry knihy, ktoré si takýto človek môže prečítať vo svojom voľnom čase. Táto metóda je podľa nás najlepšia, je to aktívne obnovovanie slovnej zásoby a je to taktiež výborné využitie voľného času.

Kombinácia viacerých metód je asi najlepšia a podľa nás čítanie kníh, kombinované s konceptom „flashcards“, je najviac produktívna. Flashcards, v slovenčine kartičky, sú ako názov už napovedá, malé kartičky ktoré sa využívajú v širokej škále odborov. V jazyku to vyzerá asi tak, že na jednej strane je slovo v materinskom jazyku a na druhej je slovo v tom jazyku ktorý sa človek snaží naučiť.

Tento koncept je ešte viac populárny v dobe technologickej, kde takéto kartičky sú voľne dostupné na internete, ako v digitálnej tak aj vo fyzickej forme. Podľa nás tento spôsob by fungoval nesmierne dobre aj v prostredí školy, napríklad vo dvojiciach, kde jednu stranu by videl iba jeden používateľ. Takto by sa študenti mohli navzájom vyskúšať a overiť si či dané slová poznajú.

Naše prevedenie vyzerá približne tak, že na jednej strane je slovo po slovensky, na druhej po anglicky, kde sme pridali k tomu obrázok a taktiež tlačidlo na prehranie výslovnosti, keby je používateľ neistý. Máme tam tiež tlačidlá „good“ a „bad“, kde good znamená to, že používateľ dané slovo vie a karta sa znovu nezobrazí a bad znamená to, že používateľ slovo nevedel a tým sa karta znovu zaradí do poradia, kvôli čomu má dvojitú šancu, že ju dostane a tým pádom lepšie zapamätá.

V našej aplikácii tým, že kombinujeme viacero typov úloh, spolu s pomôckou na slovnú zásobu, sme chceli docieľiť to, aby to bolo čo najviac vyrovnaná pomôcka, bez očividných nedostatkov.

### 3.3 Grafické rozhranie

Samozrejme, toto sú len teoretické koncepty a východiská, bez zobrazovania na obrazovku. Spraviť grafické rozhranie tak, aby toto všetko podporovalo je celkom ťažký oriešok, no tým že sme programovali v jazyku Python, sme si vedeli vybrať mnoho knižníc ktoré nám toto zjednodušili.

Naše grafické rozhranie sme naschvál chceli spraviť čo najjednoduchšie, lebo z našej skúsenosti sa nám lepšie pracovalo s aplikáciou ktorá má jasné, moderné rozhranie ako s takou, ktorá je porozhadzovaná.

Vytvorili sme dva rôzne motívy, tmavý a svetlý motív, kvôli tomu, že my osobne skoro všade používame tmavý motív, máme taktiež kamarátov, ktorý sa nepohnú zo svetlého motívu. Ako sekundárnu farbu sme si vybrali tmavomodrú, lebo sa kombinuje veľmi dobre so šedou, ktorú sme použili v tmavom motíve. Čo sa týka úloh, tuto sme tiež zvolili jednoduchší prístup, tmavomodrá nám ostala, ale vyhodnocovací algoritmus využíva červenú a zelenú. Zelená indikuje správnu odpoveď a červená nesprávnu. Tieto farby sme použili lebo je jednoznačné čo znamenajú a zosilní celkový efekt aplikácie.

Čo sa týka rôznych rozlíšení, máme to ošetrené tak, že máme použité relatívne veľkosti, kde veľkosť daného objektu sa odvíja od veľkosti okna aplikácie, čím sme vedeli predísť veľmi veľkým komplikáciám.

Jeden malý problém nastal, keď sme chceli aplikáciu odskúšať na rôznych operačných systémoch a zistili sme, že táto relatívna veľkosť síce funguje, len niekedy sa zveličí buď na obrovské, alebo miniatúrne. Ošetrili sme to tak, že sme pridali v nastaveniach možnosť zmeniť veľkosť objektov, kde na výber sú možnosti:

- Malé objekty
- Veľké objekty

## 4 Výsledky a diskusia

Výsledkom našej roboty, ktorá sa začala ešte minulý rok cez leto, je teda didaktická desktopová aplikácia, ktorá síce narazila na dosť veľa problémov počas výrobného procesu, ale dokázali sme ich obísť pomocou rôznych metód. Pri niektorých zložkách našej práce, boli naše prvotné vízie a finálny produkt viacmenej totožné, pri iných sa dosť odlišovali, ale hlavná vec je, že sme úspešne vytvorili učebnú pomôcku.

Vyhodnocovací algoritmus pri niektorých úlohách bol veľmi jednoduchý, no pri písaní slohových prác bol veľmi odlišný od ostatných. Podľa nás, tam by sa dala výborne implementovať umelá inteligencia, je to niečo čo možno mnoho ľudí ešte neuznáva, ale bez pochyb by kvôli tomu bola naša aplikácia z tohto uhla pohľadu lepšia. Naše riešenie je síce funkčné, no nie vždy zhovievavé.

Bodový a ukladač sú úzko späté, mali sme síce možnosť ukladať body do databázy, ale kvôli tomu, že sme nechceli zahltiť databázu údajmi, sme sa rozhodli toto spraviť lokálne, pomocou čoho je to taktiež oveľa bezpečnejšie.

Mohli sme použiť metódu „threading“, ktorý by nám pomohol tak, že by nám umožnil previesť viacero procesov naraz v jednom programe, lenže sú tam určité požiadavky čo sa týka procesora a kvôli tomu, že sme chceli spraviť čo najviac prívetivú aplikáciu, sme sa rozhodli proti tomuto riešeniu. Síce by to používateľovi umožnilo napríklad prehrať zvuk a taktiež vyberať odpovede naraz, ale možno na niektorých starších systémoch by to vôbec nefungovalo a tomuto sme chceli predísť.

Rozmýšľali sme dlho, ako alebo pomocou čoho spraviť grafické rozhranie tým, že v Pythone existuje už spomínané neskutočne veľa knižníc. Tuto sme možno mohli spojiť dva rôzne programovacie jazyky dokopy, v jednom spraviť funkcionality programu a v druhom spraviť grafické rozhranie. Možno by to pre nás malo vo finále lepšie výsledky, no chceli sme čo najviac efektívne využiť čas a taktiež by sme nemali žiadnu istotu, že by to fungovalo. Takto ako sme to my spravili, máme úplnú istotu, že to bude vždy fungovať a podľa nás to stále zdôrazňuje body ktoré sme chceli.

Organizáciu dát sme vyriešili pomocou databázy, no podľa nás sa to kľudne dá aj CSV súborom, čo by bolo tiež lokálne uložené ako body a podobne, lenže z toho dôvodu, že tieto súbory by boli dosť veľké sme sa rozhodli proti tomuto riešeniu a radšej sme zvolili

online databázu. Prístup na internet sme aj bez databázy potrebovali, čiže to nám rozhodnutie neovplyvnilo.

Z uhla pohľadu jazyka je táto aplikácia možno časťami nedostatočná pre niektorých, kvôli nedostatku rozmanitosti úloh. My sme v minulosti mali najviac úspechu, čo sa týka učenie jazykov, keď sme používali tieto úlohy a preto sme ich použili do tejto aplikácie. Výsledkom tohto je možno trochu repetitívna, ale konzistentná skúsenosť.

Na čítanie sme mohli použiť viacero typ úloh, populárne sú typy, kde musí používateľ doplniť vystrihnutú časť z textu, alebo kde musí vybrať správne slovo alebo slovné spojenie. Na počúvanie tiež existuje mnoho typov, stretli sme sa už s takým, kde používateľ musí doplniť malú časť do vypočutého textu, alebo kde musí správne priradiť rozprávača k malému ústrižku. Na písanie sme využili vlastne všetko čo sa dalo, okrem teda spomínaného vyhodnocovacieho algoritmu, kde si myslíme, že sa naša aplikácia dá vylepšiť, ale z dôvodu časového obmedzenia sme to nevedeli aplikovať. Na rozprávanie tiež samozrejme existujú iné typy úloh, podobne ale ako v písaní, z dôvodu časovej tiesne sme nevedeli veľmi pridať iný tvar ako máme vo finálnej verzii, ktorý je síce podľa nás dosť dobrý, mohol by tiež byť lepší. Use of English, tak ako máme v aplikácii, tak sa aj prezentuje na ozajstných jazykových skúškach, tam by sme podľa nášho srdca absolútne nič nemenili.

Samozrejme, tieto typy úloh sú všetky dosť dobré, ba aj výnimočné, ale snažili sme sa našu aplikáciu vytvoriť tak, aby bola čo najviac prístupná a preto sme použili tie najviac dostupné typy úloh, aby používateľ nebol zmätený čo presne má robiť.

Finálna verzia obsahuje podľa nás nevyhnutné základy pre hocijakú náučnú skúsenosť. Máme tam inicializačnú skúšku na zistenie úrovne, rôzne typy úloh mierené na rôzne zručnosti, 4 jazykové úrovne rozdelené podľa obtiažnosti, možnosť pohybovať sa medzi danými úrovňami, samostatnú vymoženosť na slovnú zásobu a automatické vyhodnocovanie a ukladanie.



## 5 Závery práce

Naša práca mala pár väčších cieľov, ale hlavne dosť veľa menších cieľov. Hlavným cieľom našej práce bolo vytvoriť didaktickú desktopovú aplikáciu, ktorá sprevádza človeka, ktorý sa chce naučiť po anglicky. Síce táto cesta nebola jednoduchá, priebehom tohto školského roka sme to pomaly ale isto dokázali.

Naším prvým hlavným cieľom bolo teda tá spomínaná logická časť, kde bolo zahrnuté taktiež vyhotovenie vyhodnocovacieho algoritmu, bodového systému, rozdelenie úloh, úrovne a ukladací systém.

Vyhodnocovací algoritmus sme z väčšej časti splnili bez chýb, i keď by mohol možno niekde byť vylepšený, ale podľa nás z okolností ktoré sme mali, vytlačili sme maximum a náš algoritmus boduje presne a viditeľne. Bodový systém je úzko spojený s vyhodnocovacím algoritmom, pretože keď sa vyhodnotí, že používateľ splnil úlohu správne pričíta sa jeden bod, naopak keď ju nesplní tak sa pridá jedna chyba. Tieto menšie ciele boli síce jednoduchšie, no slúžili ako základ celej aplikácie.

Ukladací systém bolo tiež nevyhnutnou súčasťou aplikácie a už trochu ťažším komponentom, bez ktorej by sa nedala považovať za plnohodnotnú. Vyriešili sme to cez textové súbory, do ktorých sa zapisuje alebo vyčíta, že koľko bodov používateľ má a na akej úrovni používateľ je.

Tieto menšie ciele sme spojili do logickej zložky, ktoré keď sme dokončili, sme mali funkčnú aplikáciu, ale bez grafického rozhrania. Slúžilo to ako skvelý základ od ktorého sme sa vedeli ešte lepšie odraziť.

Naším druhým hlavným cieľom bola jazyková zložka a neskôr ukladanie nami zvolenými úlohami na čo slúžila databáza. Úlohy sme si buď vytvorili sami, alebo sme využili zdroje ktoré boli voľne dostupné na internete. Toto bolo, aspoň z nášho uhla pohľadu, jedna z najťažších častí, kvôli množstvu úloh ktoré sme museli spracovať a neskôr pridať do databázy. Samozrejme, keď sme už úlohy nahrali do databázy, sme ich taktiež museli najprv filtrovať a potom zobrazit' na obrazovke, čo tiež nebolo jednoduché, kvôli úprave veľkosti textu. Toto sme zvládli pomocou výberu veľkosti objektov v programe a relatívnej veľkosti. Keď sme toto docielili, zostávalo nám len pridať grafické rozhranie, pomocou čoho by sme to všetko vedeli pekne zobrazit'.

Spraviť grafické rozhranie bolo posledným našim veľkým cieľom, kde sme mali na výber z viacerých rôznych technológií. My sme si vybrali podľa toho, čo sme osobne považovali za dôležité. Mohli sme to spraviť veľmi staromódne, ale to by používateľov asi odradilo, preto sme spravili niečo modernejšie, s jednoznačným farebným označením.

V praxi by sa naša aplikácia mohla využiť v školách, ako pomôcka na hodine, kde učiteľ alebo učiteľka by prešli cez učebný plán a na konci hodiny by si žiaci spolu s učiteľom alebo učiteľkou spravili pár úloh, kde program by už sám vyhodnotil zdatnosť žiakov. Taktiež si myslíme, že komponent flashcards by mohol tiež skvele fungovať v prostredí školy, namiesto slovíčok z pracovného zošita.

Naším najdôležitejším cieľom, bolo vytvoriť učebnú pomôcku, ktorá pomáha hlavne samoukom, kvôli tomu, že aj my osobne patríme do tej kategórie. Okrem tohto, si myslíme že naša aplikácia by sa dala skvele využiť aj v prostredí školstva, na hodinách angličtiny alebo podobne. Podľa nás, naša aplikácia tieto úlohy splňa bez problémov a funguje ako super doplnok nie len pre samoukov, ale aj pre školské prostredia.

## **Zhrnutie**

Naša práca pozostáva z jazykovo orientovanej, ale technologicky zdokonalenej formy vzdelávania. Cieľ tohto projektu bol pri spätnom pohľade relatívne jednoduchý, vytvoriť výučbovú desktopovú aplikáciu, do ktorej by sme ukladali cvičenia, s ktorými sme sa počas rokov stretli, a využili ich na pomoc ľuďom, ktorí sa chcú naučiť anglicky. Aplikácia využíva známe jazykové kategórie, ako je čítanie, počúvanie, hovorenie, písanie a používanie angličtiny, a precvičovaním týchto jazykových zručností sa používateľ pripravuje aj na uznávané skúšky z anglického jazyka. Zamerali sme sa na prepojenie častí iných podobných aplikácií, ako je metodické učenie jazyka, flashcards, úrovne a grafické rozhranie. Náš vlastný postup spočíval v rozdelení projektu na niekoľko častí, a to na logickú časť, ktorá prebieha na pozadí, grafickú časť, vďaka ktorej môže používateľ s našou aplikáciou komunikovať a samozrejme, jazykovú časť, ktorá bola našou osobnou obľúbenou. Keď to všetko spojíme, dostaneme konečný výsledok, plne funkčnú aplikáciu na pomoc s anglickým jazykom, ktorá rieši nielen cvičenia, ale aj slovnú zásobu a to všetko pri zachovaní jednoduchého a prehľadného rozhrania, ktoré uľahčuje používanie.

## **Resume**

Our work consists of a language driven, but ultimately technologically enhanced form of education. The goal of this project was in hindsight, relatively straightforward, make an educational desktop app, where we would store exercises that we have encountered through the years, and put them to good use by trying to help people that want to learn english. The app uses well-known language categories like reading, listening, speaking, writing and use of english, and by practicing these language skills the user also prepares themselves for recognised English language exams. We focused on connecting parts of other similar apps, like methodical language learning, flashcards, levels and a graphical interface. Our own process consisted of dividing the project into a few parts, namely the logical part, which goes on in the background, the graphical part, which is how the user can interact with our app, and of course, the language part, which was our personal favorite. Add all of these together, and you have the final result, a fully functional, english language helper app, which not only tackles exercises, but vocabulary too, all the while keeping a simple and to the point interface for ease of use.

## **Zoznam použitej literatúry**

[1.] W3Schools. Python Tutorial. [online] [cit. 2024-02-08].

Dostupné na internete: <[https://www.w3schools.com/python/python\\_intro.asp](https://www.w3schools.com/python/python_intro.asp)>

[2.] W3Schools. MySQL Tutorial. [online] [cit. 2024-02-12].

Dostupné na internete: <[https://www.w3schools.com/mysql/mysql\\_intro.asp](https://www.w3schools.com/mysql/mysql_intro.asp)>

[3.] Magic Ears University. [online] [cit. 2024-02-13].

Dostupné na internete:

<<https://magicearsuniversity.teachable.com/courses/672322/lectures/12419332>>