



Študentské Trénerské Centrum

Adaptovanie mobilnej aplikácie Noove na
desktop platformy pomocou Electron

Nagy Jakub

ZADANIE ABSOLVENTSKEJ PRÁCE

Študentské Trénerské Centrum	
Ročník:	2020
Meno a priezvisko študenta/študentky:	Jakub Nagy
Téma absolventskej práce v slovenskom jazyku:	Adaptovanie mobilnej aplikácie Noove na desktop platformy pomocou Electron
Téma absolventskej práce v anglickom jazyku:	Adapting of the Noove mobile app to desktop platforms using Electron
Cieľ absolventskej práce: (1 – 2 vety)	Mojím cieľom by bolo zobrať zdrojový kód mobilnej aplikácie a prerobiť ho na platformu Electron. To znamená, že by som nerobil všetko od základu, ale stále by to bolo dosť práce. Vzdelávacou časťou tohoto by bolo už len to, že platforma je určená pre rozvoj študentov, mohol by som ale aj proces vývoja zdokumentovať a spraviť z neho jednoduchý návod.
Stručný obsah, popis témy:	Noove je platforma, ktorá združuje študentov z celého sveta a umožňuje im pracovať na reálnych výskumných alebo komerčných projektoch, pri ktorých môžu rozvíjať svoje technické schopnosti, ako aj svoje soft skills. Rozhodli sme sa, že najlepší spôsob ako toto sprostredkovať je formou aplikácie, mobilnej a desktopovej. Na vývoj desktopovej aplikácie plánujeme využiť Microsoft technológiu - platformu Electron (na ktorej funguje napríklad aj MS Teams).

Meno a priezvisko konzultanta:	Tadeáš Dráb
Termín odovzdania témy:	05.02.2021
Termín odovzdania práce:	31.05.2021

Dátum: 5.2. 2021

Podpis študenta

A handwritten signature in black ink, appearing to read 'nagyz', written in a cursive style.

OBSAH

Zadanie absolventskej práce	2
1 Úvod	5
2 Vývoj Aplikácie	6
2.1 Školenie a práca s technológiami	6
2.2 Výber technológií	6
2.3 Programovanie HTML a štylovanie	8
2.4 Integrovanie React a generika	9
2.5 Prepojenie s backendom	9
2.6 Generovanie inštalátora	9
3 Výskum a spätná väzba	11
3.1 Príprava na získavanie spätnej väzby	11
3.2 Výsledky formulára a ich interpretácia	11
4 Záver	12
5 Zoznam použitej bibliografie a zdrojov	13
6 Prílohy	14

1 ÚVOD

Noove je naša spoločnosť, ktorá združuje študentov z celého sveta a umožňuje im pracovať na reálnych výskumných alebo komerčných projektoch, pri ktorých môžu rozvíjať svoje technické schopnosti, ako aj svoje soft skills. Rozhodli sme sa, že najlepší spôsob ako toto sprostredkovať je formou aplikácie, mobilnej a desktopovej. Na vývoj desktopovej aplikácie plánujeme využiť Microsoft technológiu - platformu Electron (na ktorej funguje napríklad aj MS Teams).

Od zadania absolventskej práce sa plán vývoja našej platformy trochu zmenil, ale aj napriek tomu sa mi naň podarilo dobre nadviazať a vytvoriť funkčné demo pre účel tejto práce.

Pri vývoji aplikácie sa využíva množstvo rôznych technológií od rôznych výrobcov. Pri vývoji našej aplikácie som sa rozhodol využiť platformu Electron, ktorá patrí pod spoločnosť Microsoft, vďaka jednoduchosti vývoja na viaceré operačné systémy pri použití tejto platformy.

2 VÝVOJ APLIKÁCIE

Vývoj aplikácie je veľmi zdĺhavý a komplikovaný proces, skladajúci sa z mnoho častí. Z tohto dôvodu som ho rozdelil na niekoľko podkapitol, kde jednotlivé jeho časti bližšie popíšem.

2.1 Školenie a práca s technológiami

Pred vývojom tejto aplikácie som už mal veľa skúseností s vývojom webov, dizajnovaním užívateľskej interakcie aj s programovaním. Napriek tomu som dovtedy pracoval s úplne inými technológiami, ako sú PHP a CMS systémy, ktoré sa pomerne dosť líšia od aplikácií.

Vedel som, že v každom prípade budem potrebovať vedieť **JavaScript**, keďže ten sa využíva v skoro všetkých moderných weboch ale aj v mnoho aplikáciách. Spravil som kompletný 30 hodinový kurz a certifikáciu z jazyka JavaScript cez službu **CodeCademy**.

Tak isto som vedel, že pravdepodobne budem využívať **TypeScript**, modifikovanú verziu JavaScriptu, ktorú vyvíja spoločnosť Microsoft. Spravil som ďalší 10 hodinový kurz a certifikáciu aj na tento jazyk.

Už som teda mal dobré základy na písanie samotnej logiky aplikácie. Aj keď som už ale predtým pracoval s jazykmi HTML a CSS, povedal som si, že na túto aplikáciu to nebude stačiť, keďže všetko štylovanie v našej aplikácii malo byť ručne napísané. Spravil som teda ďalších vyše 100 hodín kurzu sémantického HTML, komplexného štylovania s CSS a responzivity.

Dokopy som spravil asi **150 hodín** kurzov v priebehu 2 mesiacov, s tým, že v nich budem pokračovať aj po skončení práce.

2.2 Výber technológií

Výber využívaných technológií by vždy mal byť prvou časťou vývoja aplikácie. Je mnoho vecí ktoré je dobré mať dopredu dohodnuté, aby vývoj prebiehal hladko.

Prvá vec je výber platformy. Originálne sme mali mať natívne (inštalovateľné) aplikácie na mobil a desktop. Nakoniec to vyzerá tak, že tieto aplikácie budeme mať, ale až

trochu neskôr, keďže na začiatku je jednoduchšie získať užívateľov cez webovú platformu. Napriek tomu je kód pre web a pre desktopovú aplikáciu veľmi podobný, čiže vôbec neprekážalo, že som toto demo naprogramoval ako desktopovú aplikáciu.

Nasleduje výber softvérového frameworku pre túto platformu. Pri webových aplikáciách nie je takýto framework potrebný, keďže dokážu fungovať v prehliadači užívateľa. Pri mobilných aplikáciách je ale potrebné niečo, čo bude tento obsah zobrazovať užívateľovi na danom zariadení. Po dlhom rozhodovaní som vybral Electron. Electron funguje ako prepojenie medzi webovou aplikáciou a operačným systémom počítača. Využíva Chromium render engine, čo znamená, že dokáže zobraziť grafiku napísanú v jazyku HTML a CSS. Toto je užitočné, keďže tieto jazyky sú veľmi flexibilné a aj sa dobre dajú naučiť.

Zobrazovať obsah v modernej aplikácii iba pomocou HTML, CSS a JS ale nie je veľmi praktické riešenie. Väčšinou sa využívajú rôzne JavaScriptové front end knižnice, ktoré umožňujú oveľa väčšiu flexibilitu. Umožňujú jednoducho ukladať premenné, vytvárať generické komponenty (čo veľmi zníži dĺžku kódu) a umožňujú kombinovať jazyky HTML, CSS a JS v jednom súbore. Pre tento projekt som po konzultácii s ostatnými vývojármi v našom tíme vybral knižnicu React, z dielne spoločnosti Facebook. Táto knižnica sa oplatí používať hlavne z dôvodu, že okrem desktopu podporuje aj web a dokonca aj mobilné aplikácie, čo bude veľmi užitočné hlavne v budúcnosti.

Ďalšou dôležitou časťou sú podporné programovacie jazyky, takzvané preprocesory. Jazyky HTML, CSS a JS sú základom každej webovej aplikácie, no napriek tomu programovať v nich už dnes nie je veľmi efektívne. Využívajú sa rôzne podporné jazyky, ktoré dokážu pri programovaní skratiť alebo abstraktovať niektoré koncepty, čo skraca ich syntax. Namiesto klasického JS som využil jazyk **TS – TypeScript**, ktorý vyvíja spoločnosť Microsoft. Je veľmi podobný JS, s rozdielom, že dokáže rozlišovať rôzne typy premenných, čo pri fungovaní dokáže eliminovať veľa chýb. Namiesto klasického HTML som využil špeciálny formát **TSX**, čo je v podstate HTML vložené v jazyku TS. Posledný dôležitým jazykom je CSS. Klasickému CSS chýba veľa funkcií, a preto som pri vývoji využil **SCSS**.

Ďalšou dôležitou časťou je samotný backend (server) aplikácie. Pri tomto som sa už nemohol moc rozhodnúť, keďže už predtým sa pri vývoji mobilnej aplikácie pracovalo so službou **Firebase**, čiže z hľadiska kompatibilitu som túto službu musel využiť aj ja. Firebase je služba spoločnosti Google, ktorá je špeciálne orientovaná na jednoduchý vývoj backendu aplikácií, s tým, že obsahuje dôležité funkcie ako je autorizácia, databázy a úložisko pre aplikačné dáta s veľmi jednoduchou konfiguráciou a veľkými balíkmi zadarmo. Dal by sa použiť aj Microsoft Azure cloud, niektoré veci by to ale v tomto bode už mohlo zbytočne skomplikovať.

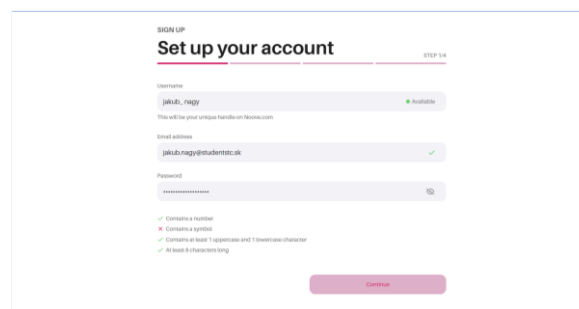
Poslednou, menej dôležitou vecou je knižnica na ikony, ktorú mi pomohli vybrať moji kolegovia až neskôr. Nakoniec sme sa rozhodli pre **Octicons**, čo je set ikon ktorý tiež patrí pod GitHub a teda aj spoločnosť Microsoft.

Dokopy v projekte využívam 3 rôzne Microsoft technológie, Electron, TypeScript a Octicons.

2.3 Programovanie HTML a štýlovanie

Ako prvé som vyvíjal HTML a SCSS na štýlovanie. Naši dizajnéri dopredu pripravili vizualizácie aplikácie v našom dizajnovom nástroji, Figma. Pozrel som sa na tieto dizajny a rozdelil ich na jednotlivé obrazovky (views). Následne som hľadal spoločné znaky medzi jednotlivými obrazovkami ako aj jednotlivými komponentami. Takto som vedel začať písať kód ktorý bol čo najviac generický a čo najmenej sa opakoval.

V HTML som sa snažil využívať čo najviac sémantických značiek pre jednoduchú čitateľnosť pre ostatných vývojárov. V SCSS som zas využil generické moduly, logiku, premenné a funkcie.



Obrázok 1 – ukážka užívateľského rozhrania v aplikácii

2.4 Integrovanie React a generika

Po dokončení predošlého kroku boli všetky dizajny preklopené na kód, ktorý sa mohol zobrazíť v akomkoľvek prehliadači a počítači, ale nedalo sa s ním ešte nijak interagovať. Začal som teda prerábať tento kód do Reactu. Pre toto som si musel prejsť ďalší krátky kurz. Nakoniec som bol schopný využiť aj časť modulov pre mobilnú aplikáciu.

Používaniu Reactu zjednodušilo vývoj vo veľa aspektoch. Mimo kratšieho kódu to znamenalo, že som nemusel aplikáciu vždy reštartovať pri zmenách v kóde, o to sa postaral sám React.

2.5 Prepojenie s backendom

Keď už bola aplikácia aj interaktívna, nastal čas ju pripojiť na backend, aby užívatelia mali s čím interagovať. Ako už bolo vyššie spomenuté, na náš backend sme využili službu **Firebase**. Firebase obsahuje mnoho nástrojov užitočných pri vývoji webovej, mobilnej alebo desktopovej aplikácie. Jedna zo služieb, ktorú využívame je databáza Firestore, v ktorej máme uložené užívateľské dáta. Ďalej využívame aj autentifikáciu, ktorá nám umožňuje zabezpečiť našu platformu a dať užívateľom viacero možností prihlásenia sa.

2.6 Generovanie inštalátora

Keď som začal vyvíjať túto aplikáciu, myslel som si, že generovanie inštalátora vôbec nebude problémom. Ukázalo sa ale, že to je celkom komplikovaný proces. Mal som už dokončenú skoro celú aplikáciu a bola plne funkčná vo vývojárskom režime. Pri generovaní inštalátora ale vznikalo veľa chýb a tak som bol nútený začať s novým základom pre aplikáciu a následne doň nakopírovať svoj starý kód.

Nakoniec to funguje tak, že kód v Reacte sa kompiluje pomocou **Webpacku**. Webpack je nástroj, ktorý je schopný zoberať komplexný kód zložený zo stoviek súborov vo viacerých programovacích jazykoch a zabaliť ho do jedného HTML a jedného JavaScript súboru, ktorý dokáže fungovať na každom počítači. Následne sa využije **Electron Builder**, ktorý tento skompilovaný kód zabalí do inštalátora.

Nakoniec som vygeneroval inštalátory na 2 rôzne platformy. Prvý inštalátor bol pre Windows a bol vo formáte **.exe**. Druhý inštalátor bol pre Mac OS a bol vo formáte **.dmg**. Jedným kliknutím na inštalátor sa dá nainštalovať celá aplikácia.

Oba inštalátory ako aj celý zdrojový kód aplikácie je dostupný na linku v prílohe.

3 VÝSKUM A SPÄTNÁ VÄZBA

Počas celého vývojového procesu som konzultoval aktuálny stav aplikácie s ľuďmi z našej firmy, s profesionálmi v danej oblasti ale aj s náhodnými ľuďmi, čo boli nablízku. Všetky pripomienky som kontinuálne pridával a vylepšoval tak celú aplikáciu.

V rámci tejto absolventskej práce som ale mal spraviť aj viac oficiálny prieskum, teda o tom budem hovoriť viac práve v tejto kapitole.

3.1 Príprava na získavanie spätnej väzby

Pôvodne som chcel robiť komplexnejšie testovanie aplikácie cez profesionálne analytické nástroje, akými sú **Google Analytics** alebo **Hotjar**. Na takéto komplexné testy ale už nezostalo moc času a nemal som ani dostatočnú vzorku testerov. Preto som sa nakoniec rozhodol vytvoriť jednoduchý formulár v **Microsoft Forms**. Tento anonymný formulár zisťuje, čo sa užívateľovi na aplikácii páčilo, čo mu chýbalo alebo vadilo a ako by celkovo ohodnotil danú časť aplikácie.

3.2 Výsledky formulára a ich interpretácia

Na získavanie spätnej väzby mi nezostalo dosť času, ale napriek tomu som rozdistribuoval inštalátory ako aj dotazník pre spätnú väzbu.

Medzi pozitívnymi aspektami hodnotení boli veci ako „*pekná paleta farieb*“ alebo „*všetko bolo funkčné a estetické*“.

Medzi negatívnymi aspektami hodnotenia boli veci ako „*tlačidlá nie sú dostatočne odlišené*“, „*neobsahovalo to hotové TOS (podmienky používania)*“, „*že zvyšok aplikácie ešte nie je hotový*“.

Celkové priemerné hodnotenie aplikácie užívateľmi bolo 4.33 / 5.

Všetky tieto výsledky hodnotím ako veľmi užitočné a určite ich budem implementovať pri ďalšom vývoji.

4 ZÁVER

Pri výbere tejto témy som očakával, že vývoj takejto aplikácie bude pomerne jednoduchší. Samotný vývoj nakoniec ale nezabral stále veľmi veľa, väčšinu času som sa práve učil pracovať s rôznymi technológiami potrebnými na vývoj aplikácie.

Ak by som robil tento projekt znova, snažil by som si nechať viac času na samotný vývoj. Napriek tomuto ale na vývoji budem pokračovať spolu s väčším tímom ľudí.

Počas tejto práce som sa naučil pracovať s veľa novými technológiami, ktoré sa dajú využiť v praxi na veľa rôznych miestach. Tak isto som sa naučil aplikovať proces testovania aplikácie a získavania spätnej väzby. Všetko toto bude užitočné pre pokračovanie vývoja tejto aplikácie ako aj pre vývoj potenciálnych budúcich aplikácií.

5 ZOZNAM POUŽITEJ BIBLIOGRAFIE A ZDROJOV

Kolektív autorov, 2021. Front-End Engineer path. In: Codecademy. Dostupné z: <https://www.codecademy.com/learn/paths/front-end-engineer-career-path>

Kolektív autorov, 2021. React Documentation. In: React. Dostupné z: <https://www.reactjs.org>

Kolektív autorov, 2021. Electron Documentation. In: Electron. Dostupné z: <https://www.electronjs.org>

Kolektív autorov, 2021. Webpack Documentation. In: Webpack. Dostupné z: <https://www.webpack.js.org>

Kolektív autorov, 2021. Electron Builder Documentation. In: Electron Builder. Dostupné z: <https://www.electron.build>

6 PRÍLOHY

Zdrojový kód aplikácie: <https://github.com/Jakub-Nagy/Noove-Electron-App-Demo>

Inštalátory k aplikácii: <https://1drv.ms/u/s!AmGR0UqomDo4qEexL3u7eTZTBss0?e=uhefsl>