

Tímový projekt
Webable

Prvý kontrolný bod - dokumentácia k inžinierskému dielu

Tím: stableFamily (tím č. 10)

Vedúci tímu: Ing. Jakub Šimko, PhD.

Členovia tímu: Bc. Michal Fabiš, Bc. Katarína Rafčíková, Bc. Daniela Sitárová, Bc. Maroš Vašš, Bc. Andrej Zaťko, Bc. Martin Žák

Ak. rok: 2018/2019

Úvod

V rámci predmetu Tímový projekt sa tím stableFamily zaoberá vývojom webového prehliadača pre zrakovo postihnutých ľudí. V tomto dokumente sa nachádzajú podklady, ktoré opisujú nami vyvíjaný softvér, t.j. inžinierske dielo. V nasledujúcich častiach sa uvádzajú ciele, ohraničenia, globálne ciele softvéru na zimný a letný semester, celkový pohľad na systém v zmysle architektúry softvéru, diagramu komponentov, opis konkrétneho modulu, ktorý sme integrovali do nášho prehliadača.

Webový prehliadač pre nevidiacich Webable je projekt, ktorý vznikol počas minulého akademického roka ako aplikácia určená do súťaže Imagine Cup. Preto je náš tím v odlišnej situácii oproti iným tímom, nakoľko pracujeme na softvéri, ktorý už bol vyvíjaný a preto sa stretávame s aj s inými problémami ako ostatné tímy. Vzhľadom na to, že počas minuloročného vývoja sme nedbali na testovanie nášho softvéru, narazili sme na niekoľko problémov, ktoré nám bránia vytvárať automatizované testy. Ide o problém, ktorý je skôr technického charakteru ako toho, že náš kód nie je testovateľný. Počas všetkých 3 šprintov sme sa snažili riešiť tento problém, no zatiaľ sa nám ho nepodarilo odstrániť.

Ciele a ohraničenia

Vzhľadom na to, že sme na softvéri pracovali počas súťaže ImagineCup 2018/2019, nemuseli sme ho vyvíjať od začiatku. Avšak ani zďaleka nespíňal to, čo by od neho naši potenciálni zákazníci očakávali. V rámci tímového projektu sme si preto stanovili nasledujúce ciele:

1. zabezpečiť to, aby náš prehliadač disponoval základnou funkcionalitou, na ktorú sú zrakovo postihnutí ľudia zvyknutí pri používaní súčasných moderných prehliadačov. Toto môžeme docieľiť tak, že náš prehliadač budeme často testovať s potenciálnymi používateľmi a zapracovávať ich pripomienky
2. zabezpečiť to, aby celková funkcionalita prehliadača bola prístupná (aby zrakovo postihnutý používateľ mohol plnohodnotne využívať náš prehliadač)
3. integrovať a zlepšiť algoritmus modulu *Automatická analýza a korekcia kódu*, ktorý bol vyvíjaný samostatne počas súťaže ImagineCup
4. zlepšiť použiteľnosť modulu *Mapa webovej lokality* a premyslieť, akým spôsobom oboznámiť nového používateľa s tým, na čo presne mapa slúži a ako sa používa
5. mať aspoň 1 používateľa

Globálne ciele pre ZS/LS

Ciele a ohraničenia, ktoré sú uvedené vyššie, sme si stanovili na celý rok. Na konci zimného semestra by sme chceli mať prehliadač v stave MVP. To znamená, že celý zimný semester sa budeme zaoberať bodmi č. 1, 2, 3. Prioritou je však pre nás bod č. 3. A to preto, lebo je to niečo inovatívne, vďaka čomu bude mať náš prehliadač pridanú hodnotu, bude iný oproti ostatným prehliadačom a budeme môcť povedať, že máme MVP.

Totíž takouto funkcionalitou nedisponuje žiadny prehliadač a ani žiadne rozšírenie do prehliadača. Bodmi č. 1, 2, 4, 5 sa chceme zaoberať hlavne v letnom semestri.

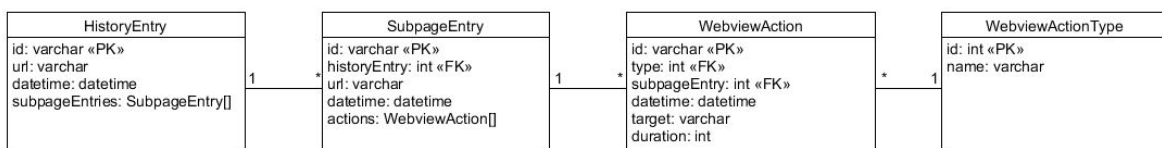
Celkový pohľad na systém

Náš prehliadač je riešený ako klientská aplikácia. Teda ide o spustiteľný súbor, ktorý si bude môcť používateľ stiahnuť z nášho webového sídla a po nainštalovaní hneď začať používať. Súčasťou našej aplikácie je aj čítačka obrazovky, ktorú my neimplementujeme, ale využívame existujúci softvér NVDA, ktorý je k našej aplikácii pribalovaný.

Primárne je prehliadač Webable programovaný v jazyku JavaScript, pretože využívame rámec Electron, ktorý používa NodeJS (javascript engine). Je niekoľko dôvodov, prečo sme sa rozhodli pre tento rámec. Prvým je ten, že zabezpečuje multiplatformovosť aplikácie a teda je jedno, či chce používateľ spustiť aplikáciu na operačnom systéme Windows, OSX alebo Linux. Ďalším je ten, že sme sa nechceli pri vývoji prehliadača zaoberať samotným spracovávaním HTML kódu, čo by bolo opätovné "vyvíjanie kolesa". Rámec Electron využíva Chromium - open-source webový prehliadač - ktorý za nás rieši elementárne veci, ktoré sú súčasťou každého webového prehliadača a preto sa môžeme sústrediť na vývoj našej vlastnej funkcionality.

Dátový model

V našom webovom prehliadači využívame 2 databázy. Prvou je dokumentovo orientovaná databáza PouchDB, v ktorej sú momentálne uložené len nastavenia prehliadača, ktoré si môže používateľ meniť. Druhou je relačná databáza SQLite, v ktorej sa ukladá história všetkých navštívených webových stránok, ktorú si vie používateľ zobraziť a navštíviť vybranú webovú stránku. Dátový model relačnej databázy je zobrazený na Obrázku č. 1.



Obrázok č.1 Dátový model nášho prehliadača

HistoryEntry

Do tabuľky sa vloží záznam vtedy, ak používateľ navštívi stránku zadaním url adresy do vstupného poľa alebo kliknutím na odkaz, ktorý používateľa presmeruje na inú doménu.

SubpageEntry

Do tabuľky sa vloží záznam vtedy, ak sa používateľ dostane na nejakú podstránku domény, ktorú práve navštívil, kliknutím na odkaz na stránke a nie zadaním url adresy do vstupného poľa.

WebviewAction

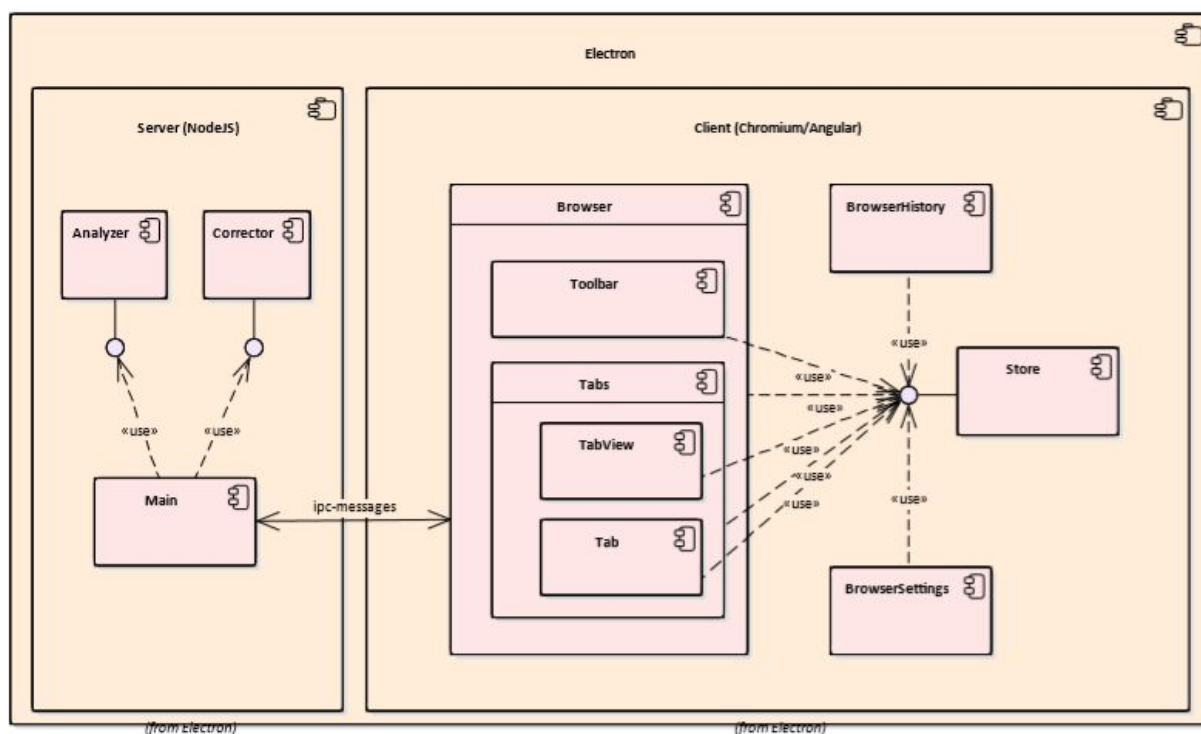
Do tabuľky sa vloží záznam vtedy, ak používateľ vykoná akciu na stránke typu *CLICK*, *SWIPE*, *SCROLL*, *HOVER*

WebviewActionType

Číselník, ktorý obsahuje 4 záznamy - *CLICK*, *SWIPE*, *SCROLL*, *HOVER*

Diagram komponentov

Architektúra prehliadača Webable je vizualizovaná na obrázku č. 2 pomocou diagramu komponentov.



Obrázok č.2 Diagram komponentov prehliadača Webable

Hoci ide o klientskú aplikáciu, je rozdelená na klientskú a serverovú časť.

Klient

Do klientskej časti patrí všetko, čo vidí používateľ. To znamená okno prehliadača s celou jeho funkcionalitou vrátane zobrazovania webových stránok. Z obrázku je možné vyčítať, že všetky komponenty využívajú rozhranie komponentu **Store**. Je to tak preto, lebo stav prehliadača je jeden veľký objekt, ktorý je týmto komponentom reprezentovaný. V prípade, že chcú ľubovoľné komponenty medzi sebou komunikovať, nekomunikujú priamo medzi sebou, ale cez komponent **Store** (ďalej centrálny objekt). A to tak, že sa vytvorí požiadavka nad centrálnym objektom, ktorá zmení jeho stav a tento objekt následne oboznámi príslušne

komponenty o zmene svojho stavu. Architektúru softvéru máme takto navrhnutú preto, aby sme zabezpečili modulárnosť systému.

Server

Do serverovej časti patrí časť aplikácie, ktorá zabezpečuje jej spustenie, inicializáciu spojenia s databázou SQLite, komunikáciu s rôznymi modulmi a kontrolu toho, či existuje čítačka obrazovky na počítači. V prípade, že neexistuje, spýta sa používateľa, či ju chce nainštalovať. V opačnom prípade ju naša aplikácia automaticky spustí.

Modul analýza a úprava zdrojového kódu

Momentálne náš prehliadač komunikuje len s modulom, ktorý analyzuje a upravuje kód. Ide o funkcionality na ktorej sa podieľala členka nášho tímu v rámci svojej bakalárskej práce. Preto analýzu, návrh a implementáciu v tomto dokumente nerozpracujeme ale uvedieme odkaz na bakalársku prácu, v ktorej sú všetky 3 časti podrobne rozpracované. V nasledujúcej podkapitole uvedieme len to, ako sme modul integrovali do nášho prehliadača.

Integrácia

Keďže sa tento modul od začiatku vyvíjal nezávisle od nášho prehliadača, na to, aby fungoval v našom prehliadači, ho bolo potrebné integrovať. Integráciu sme mohli spraviť buď na klientskej časti alebo na serverovej časti. My sme sa rozhodli vytvoriť komunikačné rozhranie s modulom na serverovej časti preto, lebo chceme, aby fungoval nezávisle od nášho prehliadača (klientskej časti aplikácie) a do budúcnosti by sme chceli z neho spraviť rozšírenie pre súčasné webové prehliadače.

Testovanie

Počas uplynulého akademického roka sme síce pracovali na vývoji prehliadača, ale pretože cieľom bolo vytvoriť softvér určený na súťaž v istom časovom období, príliš sme sa nezaoberali testovaním. Na začiatku boli vytvorené testovacie konfiguračné súbory, avšak postupom času sa stali neaktuálnymi a už viac nie sú kompatibilné s vytváraným softvérom. V súčasnosti nie je jednoduché zakomponovať do vyvíjaného softvéru testovací nástroj, pretože táto zmena si vyžaduje rozsiahle zásahy do existujúceho kódu.