

Slovenská technická univerzita
Fakulta informatiky a informačných technológií

**Testovanie používateľského zážitku pomocou
sledovania pohľadu**

Dokumentácia k dielu

Vedúci tímu: Ing. Róbert Móro

Členovia tímu: Bc. Peter Kiš, Bc. Matej Kucek, Bc. Viktória Lovasová,
Bc. Peter Kušnír, Bc. Marek Šulek, Bc. Šimon Valíček, Bc. Martin Svrček

Akademický rok: 2014/2015

Obsah

- [1 Úvod](#)
- [2 Globálne ciele](#)
 - [2.1 Podpora používateľa](#)
 - [2.2 Vizualizácie](#)
- [3 Celkový pohľad na systém](#)
- [4 Moduly systému](#)
 - [4.1 Modul API](#)
 - [4.2 JS knižnica na komunikáciu systému s externými aplikáciami](#)
 - [4.2.1 Zobrazovanie Alertov](#)
 - [4.2.2 Analýza rozšírenia](#)
 - [4.2.3 Komunikácia](#)
 - [4.2.4 Zapnutie alertov](#)
 - [4.2.5 Zhrnutie](#)
 - [4.3 Agregácie](#)
 - [4.3.1 Implementácia](#)
 - [4.4 Vizualizácie](#)
 - [4.5 Vzory čítania](#)
 - [4.6 IVT filter](#)
 - [4.6.1 Výber oka](#)
 - [4.6.2 Vypĺňanie medzier](#)
 - [4.6.3 Použité objekty](#)
- [5 Inštalačná príručka](#)
 - [5.1 Inštalačná príručka pre browser addon](#)
 - [5.1.1 Krok 1 - stiahnutie inštalačného súboru](#)
 - [5.1.2 Krok 2 - rozbalenie inštalačného súboru](#)
 - [5.1.3 Krok 3 - spustenie inštalačného súboru a postup inštalácie](#)
 - [5.2 Inštalačná príručka pre CarrotsClient](#)
 - [5.2.1 Krok 1 - stiahnutie inštalačného súboru](#)
 - [5.2.2 Krok 2 - rozbalenie inštalačného súboru](#)
 - [5.2.3 Krok 3 - spustenie inštalačného súboru a postup inštalácie](#)
 - [5.3 Inštalačná príručka pre simulátor](#)
 - [5.3.1 Krok 1 - stiahnutie inštalačného súboru](#)
 - [5.3.2 Krok 2 - rozbalenie inštalačného súboru](#)

[5.3.3 Krok 3 - spustenie inštalačného súboru a postup inštalácie](#)

[5.4 Inštalačná príručka pre aplikáciu GazeAdmin](#)

[5.4.1 Krok 1 publishovanie ViewTracking.Wcf](#)

[5.4.2 Krok 2 nastavenie webconfigu Viewtracking.Wcf](#)

[5.4.3 Krok 3 publishovanie ViewTracking.Wcf](#)

[5.4.4 Krok 4 nastavenie web configu ViewTracking.Web](#)

[6 Používateľská príručka](#)

[6.1 Používateľská príručka pre browser addon](#)

[6.1.1 Základná funkcionlita](#)

[6.1.2 Odlišnosti v implementácii rozšírení](#)

[6.2 Používateľská príručka pre Carrots Client](#)

[6.2.1 Spustenie aplikácie](#)

[6.2.2 Obrazovka pre prihlásenie](#)

[6.2.3 Výber zariadenia pre sledovanie pohľadu](#)

[6.2.4 Kalibrácia zariadenia](#)

[6.2.5 Obrazovka pre výber sedenia](#)

[6.2.6 Obrazovka so stavom aplikácie](#)

[6.2.7 Nastavenia klienta](#)

[6.3 Používateľská príručka pre Simulátor](#)

[6.3.1 EyeTracker Simulator Information](#)

[6.3.2 GazeData Information](#)

[6.3.3 Nastavenia spustenia sledovania](#)

[6.4 Používateľská príručka pre aplikáciu Gaze Admin](#)

[6.4.1 Administrácia experimentov](#)

[6.4.2 Administrácia používateľov](#)

[6.4.3 Rozhranie štatistík](#)

[6.4.4 Rozhranie pre automatické anotácie](#)

1 Úvod

Použitelnosť softvéru je pomerne rozšírenou oblasťou, pri ktorej sa využívajú rôzne techniky na jej overenie. V súčasnosti je jednou z najzaujímavejších sledovanie pohľadu pri práci s daným softvérom alebo aplikáciou. V kontexte sledovania pohľadu existuje priestor pre veľké uplatnenie tejto metódy aj v bežnom živote. Napríklad pri ovládaní elektronických zariadení pohľadom.

My sa v našom projekte chceme zamerať na tieto základné oblasti:

- *Analýza použiteľnosti softvéru* – V tomto kontexte chceme využiť eye-tracker na sledovanie pohľadu používateľa pri použití určitého softvéru a následným vyhodnocovaním získaných dát. Tento scenár sa dá využiť v oblasti UX firiem zaoberajúcich sa použiteľnosťou alebo na výskumné účely ako časť overenia pre rôzne články a iné projekty.
- *Prispôsobovanie obsahu na webe* – Táto časť súvisí s prepojením nástroja eye-tracker s určitými aplikáciami (aj webovými) a následným reagovaním na jednotlivé udalosti zachytené sledovaním pohľadu. Klasickým príkladom prispôsobovania môže byť generovanie odporúčaní.
- *Identifikácia problematických miest v danej doméne* – Klasickým využitím je v tomto smere identifikácia nezrozumiteľných častí textu v rôznych vzdelávacích systémoch a podobne.

V rámci tejto kapitoly spíšeme základné ciele, ktoré budú určovať naše smerovanie počas práce na projekte. A následne opíšeme pohľad na architektúru systému obohatenú o naše nové moduly.

2 Globálne ciele

V rámci práce na tomto projekte sme si stanovili dva základné ciele, ktoré chceme dosiahnuť. Obidva ciele vzišli z pomerne podrobnej analýzy. Snažili sme si predstaviť, čo by bolo potrebné pre klasického používateľa nášho systému. Avšak neostali sme len pri tom, a preto sme si dohodli stretnutie s „našimi zákazníkmi“, ktorými sú profesori a študenti používajúci sledovače pohľadu pri overovaní ich projektov. Na základe tejto analýzy sme identifikovali dva základné ciele nášho projektu, ktoré boli z nášho pohľadu najzaujímavejšie, avšak predstavovali aj aspekty potrebné pre jednoduchšiu prácu so získanými údajmi zo sledovania:

- Podpora používateľov
- Vizualizácia získaných dát

V prípade prvej úlohy ide z nášho pohľadu o veľmi zaujímavú snahu podporiť používateľa pri prehliadaní webových stránok alebo iných aplikácií. V prípade druhej úlohy ide o funkcionálnu vizualizáciu zozbieraných dát, ktorá je veľmi potrebná pre jednoduchú prácu s dátami a hlavne ich vyhodnocovanie.

2.1 Podpora používateľa

V tomto kontexte bolo našou snahou využiť možnosti sledovania pohľadu na podporu používateľov pri používaní rôznych aplikácií a webových stránok prostredníctvom identifikovania problematických alebo nezaujímavých častí webových stránok. Sledovanie pohľadu nám v tomto smere umožňuje získať informáciu o tom, kde sa používateľ pozerá, alebo nepozerá napríklad pri prehliadaní stránky. Identifikáciou takýchto udalostí sme schopní formou rôznych upozornení alebo odporúčaní podporiť používateľa v jeho aktivite.

Podpora používateľov je však aktuálne možná aj formou manuálnej analýzy výsledkov určitého experimentu, kedy sledujeme správanie používateľov a následne sa snažíme upraviť systém tak, aby sme odstránili odhalené problémy. Veľkou nevýhodou je v tomto prípade to, že ak má používateľ určitý problém, vo väčšine prípadov ho potrebuje riešiť čo najskôr, ideálne ihneď. Preto sme chceli aby naše riešenie dokázalo sledovať určené udalosti a reagovať na ne automaticky ihneď ako nastanú, bez potreby manuálnej analýzy.

Ako už bolo spomenuté, oblasťou nášho záujmu boli v tomto smere hlavne výučbové systémy. Avšak pri podpore používateľov siaha potenciál nášho riešenia aj do ďalších oblastí, a preto sme si stanovili, že naše riešenie bude od začiatku vytvárané tak, aby ho bolo možné použiť na ľubovoľnú aplikáciu alebo webovú stránku.

Celkovú ideu podpory používateľa sme teda rozšírili o dve vylepšenia s cieľom zvýšiť kvalitu systému vzhľadom k používateľovi:

- Reakcia na udalosti v reálnom čase
- Prenositeľnosť riešenia

Takto definovaný cieľ sme si rozdelili na dve časti z hľadiska identifikácie jednotlivých aktivít, ktoré odhaľujeme:

1. Identifikácia fixácie pohľadu na určitých oblastiach záujmu na stránke

2. Identifikácia vzorov čítania

Identifikácia fixácie pohľadu na určitých špecifických funkciách spočíva hlavne v snahe odhaliť aktivity spojené s prehliadaním webu. Príkladom je zle umiestnená funkcionalita. Do výučbového systému umiestnime nový element umožňujúci študentom riešenie testových otázok a pri sledovaní ich pohľadu zistíme, že si tento element vôbec nevšímajú. V klasickom prípade by sme toto mohli odhaliť po manuálnej analýze a následne zmeniť umiestnenie elementu. Avšak naše riešenie umožní vygenerovanie upozornenia v reálnom čase, kde bude tento element spomenutý a študent ho tak môže využívať ihneď.

V prípade identifikácie vzorov čítania je snahou odhaliť základné vzory čítania a následne primerane reagovať na tieto zistenia. Ak napríklad zistíme, že študent opakovane číta určitý odsek, môžeme na to reagovať a odporučiť mu napríklad ďalšie študijné materiály.

Prvou a základnou úlohou alebo cieľom je prepojenie existujúceho systému s ľubovoľnými externými aplikáciami cez API, ktoré chceme vytvoriť.

- **Externá aplikácia**
 - V rámci tejto časti sa konkrétne budeme snažiť vytvoriť API na prepojenie s výučbovým systémom ALEF, ktorý sa využíva na našej fakulte. Avšak celkové riešenie bude všeobecne použiteľné na rôzne typy aplikácií.
 - Tento systém sme si vybrali hlavne preto, že je reálne využívaný na našej fakulte na viacerých kurzoch alebo predmetoch. Takisto je tu možnosť, že by sa mohli sledovače pohľadu využívať priamo na cvičeniach pri práci so systémom ALEF.
- **Čo bude potrebné spraviť ?**
 - V rámci tohto cieľa je potrebné mať možnosť vytvorenia projektov, sedení a pridávania používateľov do systému.
 - Následne sa budú počas používania systému ALEF dopredu prepočítavať agregácie anotovaných dát, ktoré budú pri používaní systému vytvárané (napríklad čas, počet pozretí v čase pre danú oblasť).
 - Na základe zozbieraných dát a informácií sa budú následne vykonávať rôzne prispôbenia priamo v systéme ALEF, alebo sa budú generovať upozornenia. Využitie teda vidíme hlavne v zlepšení práce so systémom ako celkom a s tým súvisiacim zlepšením učenia sa.
- **Prenosnosť systému (portability)**
 - Pri vývoji API však chceme dbať na to, aby bolo čo najuniverzálnejšie v kontexte jeho prenositeľnosti na iné aplikácie ako je ALEF. Toto je dôležitý bod, pretože bude mať veľký vplyv na vývoj systému.
 - Celá biznis myšlienka za týmto spočíva v tom, že ak niekto bude chcieť používať náš systém pre jeho vlastnú aplikáciu, budeme schopný poskytnúť mu naše API bez potreby vykonávania nejakých veľkých úprav.

Ciele pre zimný semester v kontexte prepojenia cez API:

- Vytvorenie základnej infraštruktúry

- Našou snahou v zimnom semestri bolo v rámci API umožnenie vytvárania projektov, sedení, používateľov oblastí záujmu a agregácií.
- V kontexte agregácií sme chceli umožniť samotné počítanie agregácií nad získanými dátami.
- Vytvorenie základnej javascript knižnice
 - Takisto sme chceli aby bola na konci zimného semestra možná základná komunikácia medzi externou aplikáciou a gaze-tracking softvérom.

Ciele pre letný semester v kontexte prepojenia cez API:

- Generovanie upozornení
 - Našou snahou v letnom semestri bolo umožniť generovanie upozornení v externej aplikácii prostredníctvom nášho riešenia.
 - V tomto kontexte bolo potrebné sfunkčniť celé riešenie aj na strane servera aj v rámci JavaScript knižnice, ktorú sme vytvorili.
 - Základným prístupom bolo vygenerovanie upozornenia pri určitých vzoroch správania.
 - Rovnako bolo naším cieľom implementovať algoritmus detekcie vzorov čítania a ten následne využiť ako zdroj vzorov správania pre generovanie upozornení.

2.2 Vizualizácie

V rámci minuloročného projektu sa podarilo vytvoriť základnú infraštruktúru na sledovanie pohľadu a ukladanie dát z takéhoto pozorovania. Našou úlohou bude využiť získané dáta a nejakým spôsobom ich spracovať a vizualizovať a hlavne ich týmto spôsobom spraviť užitočné.

Pri sledovaní pohľadu rovnako potrebujeme získané dáta vizualizovať pre učiteľov a ukázať im rôzne štatistiky v pochopiteľnej podobe, pretože dáta zo sledovania pohľadu sú pomerne zložité. Prostredníctvom rôznych štatistík a vizualizácií vo forme grafov alebo tabuliek môžeme rovnako pomôcť aj študentom pri výučbe a pri ich výskumnej činnosti prostredníctvom identifikácie problematických oblastí vo vzdelávacom obsahu.

Samotná idea vizualizácie je v našom prípade zameraná hlavne na oblasť analýzy dát, kedy sme chceli zjednodušiť proces vyhodnocovania experimentov spojených so sledovaním pohľadu. Takéto spracovanie dát má pre nás tri hlavné významy spojené najmä s oblasťou výučby a výskumu:

- Prvým významom takto spracovaných dát je poskytnutie rozumného pohľadu na dáta aj pre ľudí, ktorí nie sú výskumníci a chceli by vidieť nejaký výstup zo sledovania pohľadu.
- Druhým využitím je umožnenie získania zaujímavých vizualizácií, ktoré si môžu výskumníci vložiť do práce a zdokumentovať tak overenie práce.
- Ďalším využitím je pomoc učiteľom pri tvorbe učebných textov, alebo pri získaní informácií o tom, čo je pre používateľov problémová oblasť, a čo je potrebné rozobrať bližšie.

Spôsob spracovania dát:

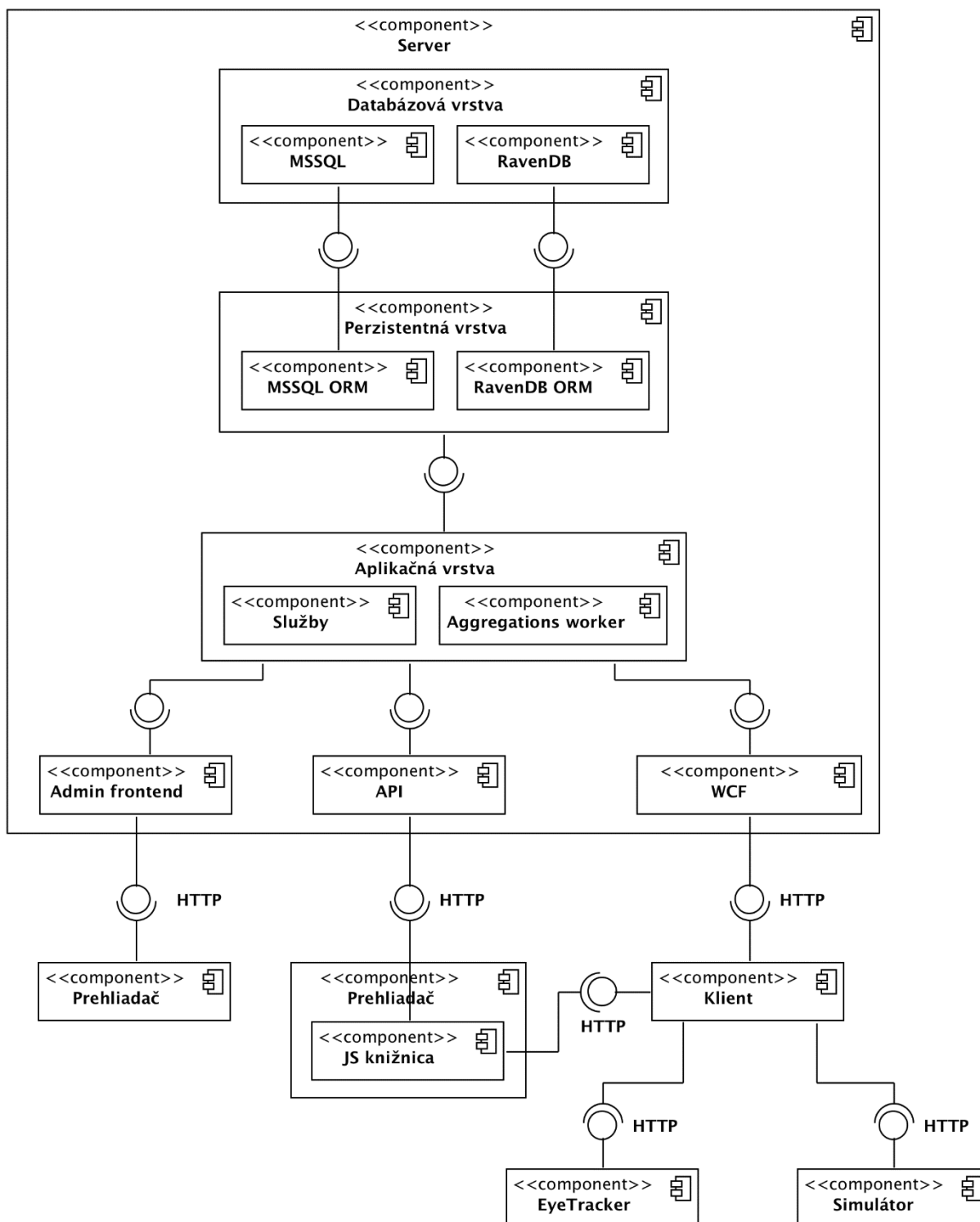
- Štatistiky
 - Štatistiky predstavujú základný prehľad o projektoch, sedeniach, používateľoch a ich aktivite v systéme, ktorá bola zaznamenaná sledovaním pohľadu.
 - Zaznamenávali by sa takto napríklad počty fixácií pohľadu alebo trvanie v čase pre danú oblasť záujmu.
- Vizualizácie
 - Vizualizácie predstavujú rôzne grafy a teplotnémapy zaznamenávajúce poradie fixácií, ale aj pomery časov sledovania jednotlivých oblastí.
 - V tomto kontexte bude potrebné zaznamenávanie zmeny pohľadu v čase (pomocou videa alebo obrázkov).

Ciele pre zimný semester v kontexte vizualizácie sme nemali stanovené.

Ciele pre letný semester v kontexte vizualizácií:

- Generovanie grafov
 - V rámci letného semestra bolo našim cieľom úplne sfunkčniť možnosť generovania grafov.
 - V prvom kroku sme identifikovali potrebu spracovať dáta do potrebnej podoby pre generovanie rôznych typov grafov.
 - Následne sme vyriešili spôsob výberu typu grafu vo webovej aplikácii.
 - Posledným krokom bolo samotné generovanie grafov.

3 Celkový pohľad na systém



Obrázok 1: Základná architektúra systému gaze-tracking.

Služba je navrhnutá ako server-klient riešenie, jej architektúru zobrazuje Obrázok 1. Z pohľadu toku dát je jedným z hlavných komponentov celého systému NOSQL databáza RavenDB. Do tejto sa ukladajú jednotlivé súradnice pohľadu sledovaného používateľa. Zdrojom údajov je senzor na sledovanie pohľadu, z ktorého sú dáta čítané prostredníctvom desktopovej klientskej aplikácie. Táto aplikácia následne komunikuje s rozšírením vo webovom prehliadači, ktorý jej poskytuje informácie o elementoch DOM stromu webovej stránky, ktoré sa nachádzajú na aktuálnych súradniciach pohľadu. Klientska aplikácia následne súradnicové dáta doplnené o identifikáciu elementov odosiela prostredníctvom HTTP volaní na server. Na serveri je za týmto účelom implementované proprietárne API rozhranie slúžiace na komunikáciu s klientskou aplikáciou, z ktorého sa prijaté dáta ukladajú do RavenDB. Okrem RavenDB riešenie zahŕňa navyše relačnú databázu, do ktorej sa ukladajú agregované dáta, vypočítané za pomoci asynchrónnych workerov z dát uložených v RavenDB. Tento prístup bol zvolený za cieľom zvýšenia priepustnosti systému pri vyberaní štatistických dát o sedeniach z databázy. V relačnej databáze sa takisto nachádzajú administratívne údaje ohľadom projektov sledovania pohľadu, konkrétnych sedení a sledovaných používateľov. Pre účely manažmentu administratívnych údajov slúži frontend aplikácia.

Novými komponentami v systéme sú API modul pre komunikáciu s aplikáciami tretích strán, javascriptová knižnica slúžiaca ako náhrada rozšírenia do prehliadača a zobrazovanie vizualizácií získaných údajov v administratívnom webovom rozhraní. Demonštratívne bude implementovaná komunikácia medzi službou a aplikáciou Alef. Okrem toho bude frontend aplikácia rozšírená o vizualizáciu štatistických dát agregovaných v relačnej databáze.

MSSQL

Štandardná relačná databáza slúžiaca k uchovávaní údajov o používateľoch, projektoch, sedeniach, oblastiach záujmu a upozorneniach. Uchovávajú sa tu takisto agregované dáta vypočítané zo surových dát z EyeTrackeru.

RavenDB

NOSQL databáza slúžiaca k uchovávaní surových dát z EyeTrackeru. Tieto dáta sa skladajú z identifikácie sedenia a používateľa, od ktorého pochádzajú, súradníc pohľadu pre ľavé a pravé oko a časovej pečiatky.

Aggregations worker

Táto súčasť systému má za úlohu spracúvať prichádzajúce surové dáta z EyeTrackeru a realizovať nad nimi analýzu. Medzi sledované vlastnosti patria časy pohľadu do jednotlivých oblastí záujmu a tento modul takisto uskutočňuje detekciu fixácií pohľadu na sledovanej stránke.

Admin frontend

Webové rozhranie určené pre administrátorov a klientskych používateľov slúži k správe používateľov, projektov, sedení, oblastí záujmu a upozornení.

Klientská aplikácia

V rámci projektu z minulého roku sa podarilo vytvoriť desktopový klient, ktorý je prvou inštanciou, s ktorou sa používateľ dostáva do styku. Nachádza sa teda na prvej vrstve architektúry, najbližšie k používateľovi.

Jeho úlohou je komunikácia so zariadením na sledovanie pohľadu. Aby takáto komunikácia mohla byť naviazaná, je potrebná synchronizácia so zariadením, ktorú taktiež tento desktopový klient zabezpečuje.

V rámci komunikácie zaznamenáva, kde sa daný používateľ pozeral a následne tieto informácie vo forme dát odosiela na serverovú časť systému.

Rozšírenie do prehliadača

V rámci práce minuloročného tímu sa podarilo vytvoriť rozšírenia pre webové prehliadače GoogleChrome a MozillaFirefox. Samotné rozšírenie sa delí na administrátorskú a klientsku časť (rozšírenie pre Firefox obsahuje obe tieto časti v jednom rozšírení, zatiaľ čo Chrome rozšírenia sú oddelené). Hlavnou funkciou rozšírenia je obohacovanie dát klientskej aplikácie, ktoré získava zo zariadenia na sledovanie pohľadu. Pre každú súradnicu pohľadu je v rozšírení určený identifikátor HTML elementu, na ktorý sa používateľ pozerá, pričom sú tieto údaje spárované s dátami prijatými z klientskej aplikácie. Druhou funkciou rozšírenia je možnosť definovania oblastí záujmu pre jednotlivé stránky. Myšlienkou je špecifikovanie oblastí webovej stránky na základe identifikujúceho elementu, samotnej adresy stránky a dopĺňujúcimi informáciami ako sú názov a popis.

JS knižnica

Rozšírenie do prehliadača bolo nahradené JavaScriptovou knižnicou, ktorá poskytuje všetku funkcionálnu ponúkanú rozšírením do prehliadača. Umožňuje vytvárať oblasti záujmu a následne detegovať pohľad používateľa na oblasti záujmu.

WCF (Windows Communication Foundation)

WCF predstavuje súbor služieb v rámci .NET, ktorý umožňuje vytváranie aplikácií orientovaných na služby. Minulý rok vytvoril predchádzajúci tím metódy pre správu dopytov na WCF službu v súbore *Viewtracking.wcf*. Táto služba má na starosti zachytávanie dopytov a následné odpovede, pričom komunikuje prostredníctvom protokolu HTTPS. Samotné odpovede alebo dáta v nich obsiahnuté sa posielajú vo formáte JSON.

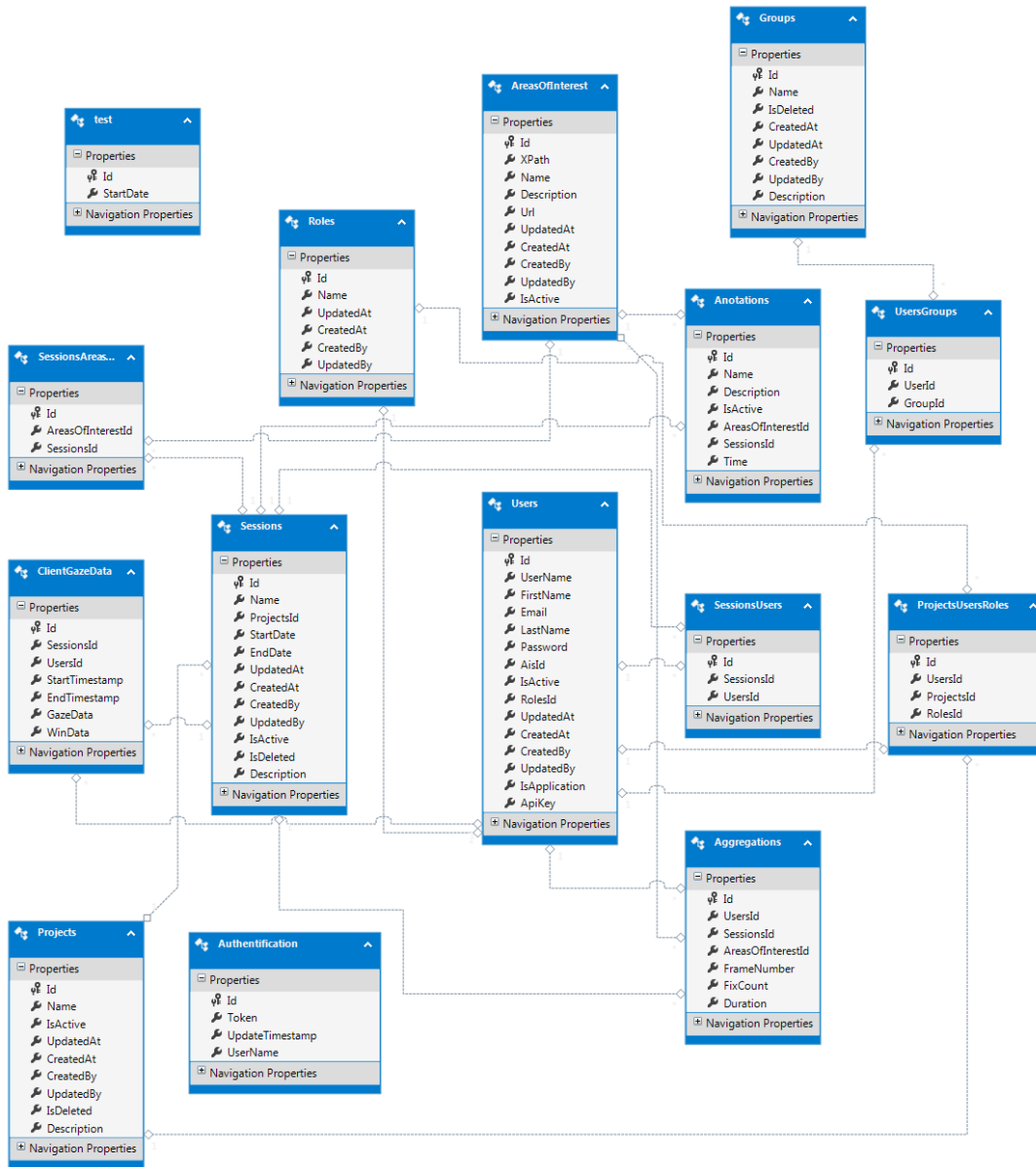
Simulátor

Tím z minulého roka spravil simulátor, ktorý predstavuje náhradu za eyetracker pričom zaznamenáva súradnice myši a ukladá ich do databázy. Predpokladá sa, že pozícia myši predstavuje používateľov pohľad, pričom sa berie do úvahy aj určitý šum v okolí myši a nie jej presné súradnice.

Dátový model

Centrálnymi entitami dátového modelu sú projekt, sedenie a používateľ. Používateľ je základným prvkom pri prístupe k systému. Zastupuje aplikácie tretích strán a takisto jednotlivých používateľov vstupujúcich do procesu sledovania pohľadu. V prípade aplikácií

tretích strán je autentifikácia vykonávaná na základe poskytnutého API kľúča. Takýto používateľ môže následne vytvárať projekty, sedenia, a ďalších používateľov, ktorých pohľad bude v rámci jednotlivých sedení sledovaný. Entita projekt slúži na uchovávanie základných údajov o projektoch sledovania pohľadu. V rámci jedného projektu je spravovaných viacero nezávislých sedení, do ktorých sú pridávaní používatelia.



Obrázok 2: Dátový model systému GazeTracking.

4 Moduly systému

V tejto kapitole sú popísané jednotlivé moduly systému na sledovanie pohľadu vzhľadom na ciele, ktoré sme si stanovili. Každý z týchto modulov predstavuje oblasť, ktorú aktuálne riešime, alebo chceme riešiť v blízkej budúcnosti.

4.1 Modul API

Modul API slúži na komunikáciu aplikácii tretích strán so službou gaze-tracking. Prostredníctvom jednotlivých volaní API modulu je možné spravovať projekty, sedenia, agregácií, oblastí záujmu a používateľov prístupujúcej aplikácie. Aplikácia sa musí pri každom prístupe k API modulu autentifikovať za pomoci API kľúča, ktorý je odosielaný v tvare reťazca.

API modul je implementovaný za použitia technológie WCF. Prijíma parametre prostredníctvom HTTP POST požiadaviek a odpovedá vo forme JSON reťazcov. Špecifikácia jednotlivých API volaní je uvedená nižšie.

Volania GazeTracking API

- **Manažment Projektov**
 - **CreateProject (Vytvoriť projekt)**
 - **požiadavka:** POST application/x-www-form-urlencoded
 - **url:** /api/project/create
 - **parametre:**
 - apiKey:String(notnull) - autentifikačný token
 - name:String(notnull) - názov vytváraného projektu
 - description:String - popis vytváraného projektu
 - **odpoveď:**
 - application/json
 - id:integer - identifikátor vytvoreného projektu
 - **HTTP stavové kódy:**
 - 201 - projekt bol vytvorený
 - 400 - nesprávny formát požiadavky
 - 401 - nesprávny api kľúč
 - 500 - chyba pri spracovaní
 - **ListProjects (Vypísanie všetkých projektov)**
 - **požiadavka:**POST application/x-www-form-urlencoded
 - **url:**/api/project/list
 - **parametre:**
 - apiKey:String(notnull) - autentifikačný token
 - active:Boolean(notnull) - vypísať len aktívne projekty
 - **odpoveď:**
 - application/json
 - projects - pole nájdených projektov

- id:integer - id projektu
 - name:String - názov projektu
 - isActive:boolean - stav projektu
 - **HTTP stavové kódy:**
 - 200 - všetko v poriadku
 - 400 - nesprávny formát požiadavky
 - 401 - nesprávny api kľúč
 - 500 - chyba pri spracovaní
 - **GetProjectById (Vypísať podrobnosti projektu)**
 - **požiadavka:**POST application/x-www-form-urlencoded
 - **url:**/api/project/getbyid
 - **parametre:**
 - apiKey:String(notnull) - autentifikačný token
 - id:integer(notnull) - id hľadaného projektu
 - **odpoveď:**
 - application/json
 - id:integer - id projektu
 - name:String - názov projektu
 - isActive:boolean - stav projektu
 - description:String - popis projektu
 - createdAt:integer - čas vytvorenia projektu v unix formáte
 - updatedAt:integer - čas poslednej zmeny projektu v unix formáte
 - **HTTP stavové kódy:**
 - 200 - všetko v poriadku
 - 400 - nesprávny formát požiadavky
 - 401 - nesprávny api kľúč
 - 500 - chyba pri spracovaní
- **Manažment sedení**
 - **CreateSession(Vytvoriť sedenie)**
 - **požiadavka:**POST application/x-www-form-urlencoded
 - **url:**/api/session/create
 - **parametre:**
 - apiKey:String(notnull) - autentifikačný token
 - projectId:integer(notnull) - id nadradeného projektu
 - **odpoveď:**
 - application/json
 - id:integer - id vytvoreného sedenia
 - **HTTP stavové kódy:**
 - 201 - sedenie bolo vytvorené
 - 400 - nesprávny formát požiadavky
 - 401 - nesprávny api kľúč

- 500 - chyba pri spracovaní
- **ListSession(Vypísať existujúce sedenia)**
 - **požiadavka:**POST application/x-www-form-urlencoded
 - **url:**/api/session/list
 - **parametre:**
 - apiKey:String(notnull) - autentifikačný token
 - active:boolean(notnull) - vypísať len aktívne sedenia
 - **odpoveď:**
 - application/json
 - sessions - pole sedení
 - id - identifikátor sedenia
 - name - názov sedenia
 - isActive - stav sedenia
 - **HTTP stavové kódy:**
 - 200 - všetko v poriadku
 - 400 - nesprávny formát požiadavky
 - 401 - nesprávny api kľúč
 - 500 - chyba pri spracovaní
- **GetSessionById(Vypísať podrobnosti sedenia)**
 - **požiadavka:**POST application/x-www-form-urlencoded
 - **url:**/api/session/getbyid
 - **parametre:**
 - apiKey:String(notnull) - autentifikačný token
 - id:integer(notnull) - identifikátor sedenia
 - **odpoveď:**
 - application/json
 - id - identifikátor sedenia
 - projectId - id nadriadeného projektu
 - name - názov sedenia
 - isActive - stav sedenia
 - createdAt:integer - čas vytvorenia sedenia v unix formáte
 - updatedAt:integer - čas poslednej zmeny sedenia v unix formáte
 - **HTTP stavové kódy:**
 - 200 - všetko v poriadku
 - 400 - nesprávny formát požiadavky
 - 401 - nesprávny api kľúč
 - 500 - chyba pri spracovaní
- **Manažment používateľov**
 - **CreateUser (Vytvoriť používateľa)**
 - **požiadavka:**POST application/x-www-form-urlencoded
 - **url:**/api/user/create
 - **parametre:**

- apiKey:String(notnull) - autentifikačný token
 - username:String(notnull) - používateľské meno
 - firstname:String(notnull) - krstné meno používateľa
 - lastname:String(notnull) - priezvisko používateľa
 - email:String - email používateľa
- **odpoveď:**
 - application/json
 - id:integer - id vytvoreného používateľa
- **HTTP stavové kódy:**
 - 201 - používateľ bol vytvorený
 - 400 - nesprávny formát požiadavky
 - 401 - nesprávny api kľúč
 - 500 - chyba pri spracovaní
- **ListUsers (Vypísať existujúcich používateľov)**
 - **požiadavka:**POST application/x-www-form-urlencoded
 - **url:**/api/user/list
 - **parametre:**
 - apiKey:String(notnull) - autentifikačný token
 - active:boolean(notnull) - vypísať len aktívnych používateľov
 - sessionId:integer - identifikátor sedenia
 - **odpoveď:**
 - application/json
 - users - pole používateľov
 - id:integer - identifikátor používateľa
 - username:String - používateľské meno
 - isActive:Boolean - stav používateľa
 - **HTTP stavové kódy:**
 - 200 - všetko v poriadku
 - 400 - nesprávny formát požiadavky
 - 401 - nesprávny api kľúč
 - 500 - chyba pri spracovaní
- **GetUserById (Vypísať podrobnosti používateľa)**
 - **požiadavka:**POST application/x-www-form-urlencoded
 - **url:**/api/user/getbyid
 - **parametre:**
 - apiKey:String(notnull) - autentifikačný token
 - id:integer(notnull) - identifikátor používateľa
 - **odpoveď:**
 - application/json
 - id:integer - identifikátor používateľa
 - username:String - používateľské meno
 - firstname:String - krstné meno používateľa
 - lastname:String - priezvisko používateľa

- email:String - email používateľa
 - isActive:Boolean - stav používateľa
 - createdAt:integer - čas vytvorenia používateľa v unix formáte
 - updatedAt:integer - čas poslednej zmeny používateľa v unix formáte
- **HTTP stavové kódy:**
 - 200 - všetko v poriadku
 - 400 - nesprávny formát požiadavky
 - 401 - nesprávny api kľúč
 - 500 - chyba pri spracovaní
- **Manažment oblastí záujmu**
 - **CreateAoi**
 - **požiadavka:**POST application/x-www-form-urlencoded
 - **url:**/api/aoi/create
 - **parametre:**
 - apiKey:String(notnull) - autentifikačný token
 - xpath:String(notnull) - XPath adresa AOI
 - name:String(notnull) - názov AOI
 - description:String - popis AOI
 - url:String(notnull) - url stránky obsahujúcej AOI
 - **odpoveď:**
 - application/json
 - id:integer - id AOI
 - xpath:String(notnull) - XPath adresa AOI
 - name:String(notnull) - názov AOI
 - description:String - popis AOI
 - url:String(notnull) - url stránky obsahujúcej AOI
 - **HTTP stavové kódy:**
 - 201 - všetko v poriadku
 - 400 - nesprávny formát požiadavky
 - 401 - nesprávny api kľúč
 - 500 - chyba pri spracovaní
 - **UpdateAoi**
 - **požiadavka:**POST application/x-www-form-urlencoded
 - **url:**/api/aoi/update
 - **parametre:**
 - id:integer - id AOI
 - apiKey:String(notnull) - autentifikačný token
 - xpath:String(notnull) - XPath adresa AOI
 - name:String(notnull) - názov AOI
 - description:String - popis AOI
 - url:String(notnull) - url stránky obsahujúcej AOI

- **odpoveď:**
 - application/json
 - id:integer - id AOI
 - xpath:String(notnull) - XPath adresa AOI
 - name:String(notnull) - názov AOI
 - description:String - popis AOI
 - url:String(notnull) - url stránky obsahujúcej AOI
- **HTTP stavové kódy:**
 - 200 - všetko v poriadku
 - 400 - nesprávny formát požiadavky
 - 401 - nesprávny API kľúč
 - 500 - chyba pri spracovaní
- **GetAoiById**
 - **požiadavka:** POST application/x-www-form-urlencoded
 - **url:** /api/user/getbyid
 - **parametre:**
 - apiKey:String(notnull) - autentifikačný token
 - id:integer(notnull) – id AOI
 - **odpoveď:**
 - application/json
 - id:integer - id AOI
 - xpath:String(notnull) - XPath adresa AOI
 - name:String(notnull) - názov AOI
 - description:String - popis AOI
 - url:String(notnull) - url stránky obsahujúcej AOI
 - **HTTP stavové kódy:**
 - 200 - všetko v poriadku
 - 400 - nesprávny formát požiadavky
 - 401 - nesprávny API kľúč
 - 500 - chyba pri spracovaní
- **GetAoiBySessionId**
 - **požiadavka:** POST application/x-www-form-urlencoded
 - **url:** /api/user/getbysessionid
 - **parametre:**
 - apiKey:String(notnull) - autentifikačný token
 - sessionId:integer(notnull) - id sedenia
 - **odpoveď:**
 - application/json
 - aois - pole AOI
 - **HTTP stavové kódy:**
 - 200 - všetko v poriadku
 - 400 - nesprávny formát požiadavky
 - 401 - nesprávny API kľúč
 - 500 - chyba pri spracovaní

- **DeleteAoi**
 - **požiadavka:**POST application/x-www-form-urlencoded
 - **url:**/api/user/list
 - **parametre:**
 - apiKey:String(notnull) - autentifikačný token
 - id:integer(notnull) - id AOI
 - **odpoveď:**
 - application/json
 - id:integer - id AOI
 - xpath:String(notnull) - XPath adresa AOI
 - name:String(notnull) - názov AOI
 - description:String - popis AOI
 - url:String(notnull) - url stránky obsahujúcej AOI
 - **HTTP stavové kódy:**
 - 200 - všetko v poriadku
 - 400 - nesprávny formát požiadavky
 - 401 - nesprávny api kľúč
 - 500 - chyba pri spracovaní
- **AddAoiToSession**
 - **požiadavka:**POST application/x-www-form-urlencoded
 - **url:**/api/user/addtosession
 - **parametre:**
 - apiKey:String(notnull) - autentifikačný token
 - id:integer(notnull) - id AOI
 - sessionId:integer(notnull) - id sedenia
 - **odpoveď:**
 - application/json
 - id:integer - id AOI
 - xpath:String(notnull) - XPath adresa AOI
 - name:String(notnull) - názov AOI
 - description:String - popis AOI
 - url:String(notnull) - url stránky obsahujúcej AOI
 - **HTTP stavové kódy:**
 - 200 - všetko v poriadku
 - 400 - nesprávny formát požiadavky
 - 401 - nesprávny api kľúč
 - 500 - chyba pri spracovaní
- **Manažment upozornení**
 - **GetAlertsByAoiId**
 - **požiadavka:**POST application/x-www-form-urlencoded
 - **url:**/api/user/list
 - **parametre:**
 - apiKey:String(notnull) - autentifikačný token

- aoiId:integer(notnull) - id AOI
- **odpoveď:**
 - application/json
 - alerts - pole upozornení
- **HTTP stavové kódy:**
 - 200 - všetko v poriadku
 - 400 - nesprávny formát požiadavky
 - 401 - nesprávny api kľúč
 - 500 - chyba pri spracovaní

4.2 JS knižnica na komunikáciu systému s externými aplikáciami

V tejto časti opíšeme spôsob komunikácie nášho systému s externou (webovou) aplikáciou prostredníctvom JavaScript knižnice.

4.2.1 Zobrazovanie upozornení

Jedným z našich cieľov bolo vytvorenie externej knižnice, ktorá by bola ľahko prenosná a umožňovala použitie v akejkolvek webovej aplikácii. Jadro tejto knižnice vychádza z rozšírenia do google chrome, ktoré bolo implementované už skoršej fáze projektu. Knižnica je písaná v jazyku javascript. Je dôležité, aby po pridaní tejto knižnice mohla aplikácia komunikovať so systémom gaze.

V rozšírení do google chrome, už bola implementovaná logika pre komunikáciu so systémom. Úlohou bolo teda ju vytiahnuť do samostatnej javascriptovej knižnice a následne pridať funkcionality, ktorá by umožňovala si vybrať oblasť záujmu a sledovať pohľad užívateľa a zaznamenať, či sa do oblasti užívateľ pozrel alebo nie.

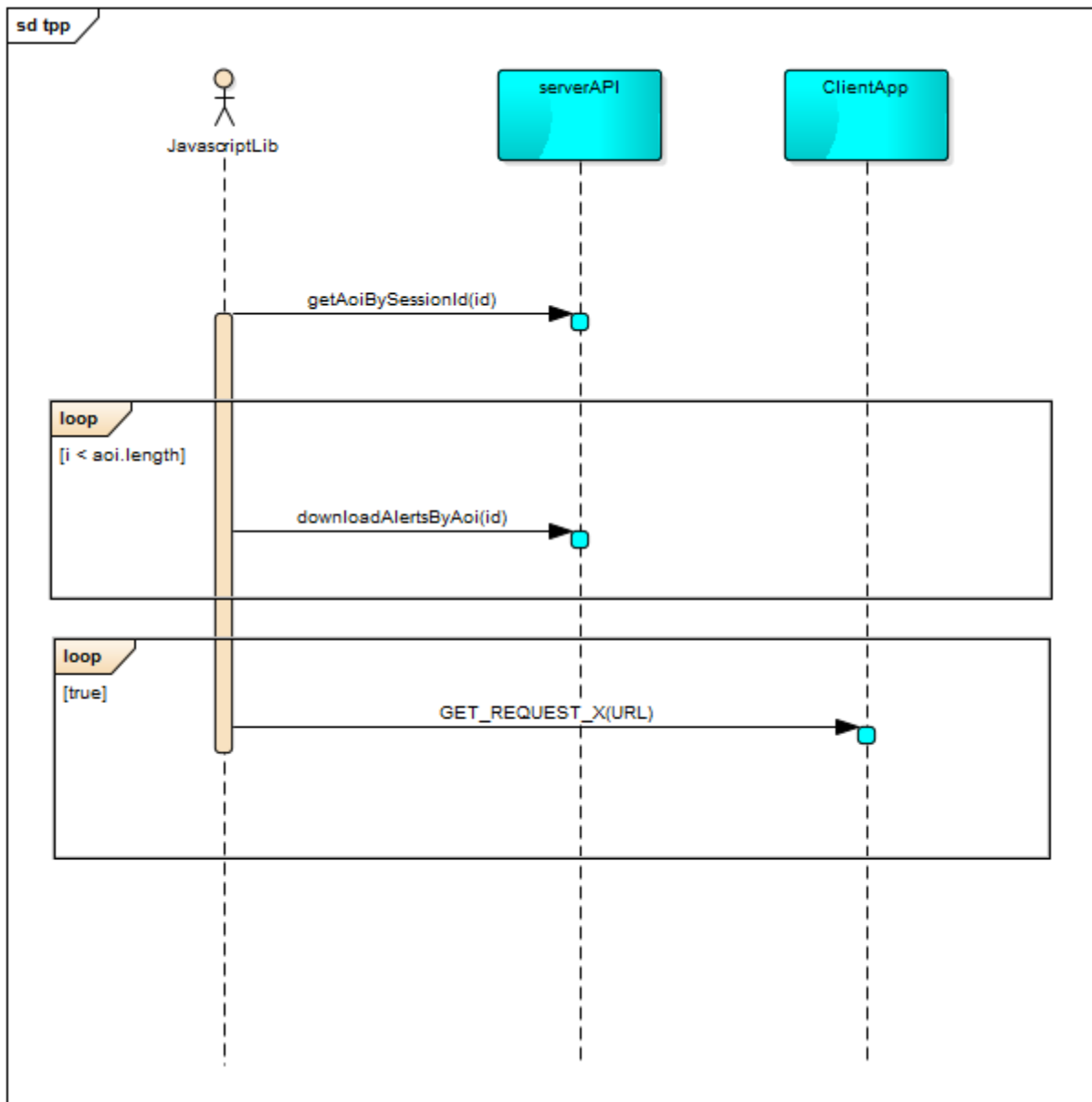
4.2.2 Analýza rozšírenia

Vykonalí sme analýzu rozšírenia pre google chrome. Jeho hlavnou nevýhodou je to, že používateľ si musí nainštalovať rozšírenie do internetového prehliadača, ktoré je však implementované len pre prehliadač google chrome. Potrebovali sme užívateľa oslobodiť od nutnosti inštalovať ďalší nástroj. Rozhodnutie bolo presunúť logiku do samostatnej knižnice.

4.2.3 Komunikácia

Pre komunikáciu je nutné mať spustenú klientsku aplikáciu a ak nevlastníte eye tracker aj simulátor. Pri spustení je potrebné prihlásenie do klienta a vybratie správneho sedenia.

Knižnica komunikuje s klientskou aplikáciou. Klientska aplikácia, ktorá monitoruje pohľad používateľa, sa po spustení zapne na lokálnej adrese `http://localhost:62582/server`, z ktorej sa dá získať súradnica toho, kam sa používateľ momentálne pozerá. Knižnica po spustení posiela XMLHttpRequest na localhost. Bolo nutné povoliť v klientskej aplikácii cross domain volania, pretože javascript túto vlastnosť v základe nepovoľuje, na rozdiel od predtým použitého google chrome rozšírenia. Aplikácia vráti aktuálne súradnice kam sa klient pozerá. Toto sa opakuje v pravidelných cykloch a tak vie knižnica kam sa aplikácia momentálne pozerá. Tak isto sa na začiatku vykoná JQUERY požiadavka na stiahnutie zadaných oblastí záujmu pre dané sedenie pomocou funkcie `getAoiBySessionId` z nášho API. Oblasť záujmu nám príde vo formáte JSON, ktorý si uložíme do poľa. Ďalej sa dotiahnú alerty pre dané oblasti pomocou funkcie `downloadAlertsByAoi`, ktorej parameter id si zistíme z danej oblasti, ktorú sme si uložili.





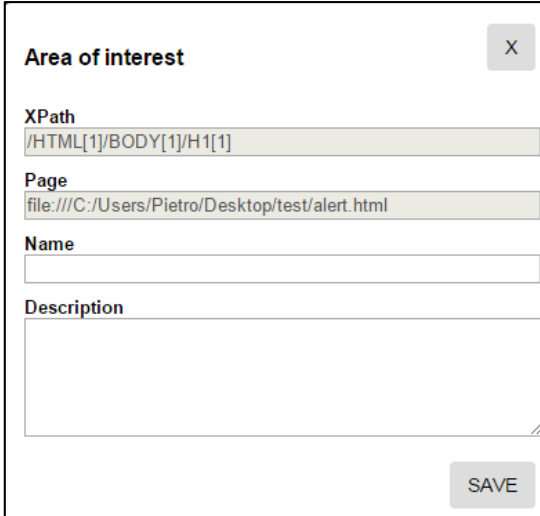
Obrázok 3: Sekvenčný diagram komunikácie.

4.2.4 Zapnutie alertov

Stiahnuté alerty v sebe obsahujú správu, ktorá sa má vypísať, časový limit po akú dobu sa používateľ do danej oblasti má alebo nemá pozeráť, atribút sa volá wasLooking. Následne sa podľa danej správy nastaví v javascripte timer, ktorý zobrazí alert ak sa užívateľ do danej oblasti nepozeral. Tento timer sa vypne ak sa používateľ do danej oblasti začne pozeráť, a teda ani daný alert sa nezobrazí. Ak je atribút alertu wasLooing alertu nastavený na možnosť true, umožní zapnutie ďalšieho timeru, ktorý sa spustí ak sa užívateľ do oblasti záujmu pozerá. Ak je teda táto možnosť povolená pre dané id sa spustí timer, v momente ak nám dôjde z clientskej xpath a ten skontrolujeme či sa nejedná o väčší celok. Tento timer sa vypne ak sa užívateľ prestal pozeráť do oblasti záujmu.

4.2.5 Vytváranie oblastí záujmu

Ďalšou funkcionalitou využívanou v rozšírení bolo definovanie oblastí záujmu pre konkrétnu stránku. Pri použití našej knižnice v projektoch tretích strán je po načítaní stránky v pravom hornom rohu zobrazený malý symbol oka . Kliknutím na túto ikonu je možné spustiť samostatný proces definovania oblastí záujmu. Po kliknutí na ikonu je vykonaná autentifikácia používateľa pomocou identifikačného kódu. Tento kód je generovaný v správcovi oblastí záujmu vo webovej aplikácii GazeAdmin (Projects -> Edit -> Sessions/experiments -> Edit -> Areas of interest -> Generate verifying code). Po úspešnej autentifikácii je možné kliknutím na požadovaný element stránky zahájiť proces vytvorenia novej oblasti záujmu (ikona oka sa zmení na zelenú ). Na zadenovanie novej oblasti záujmu slúži jednoduchý formulár, v ktorom sa špecifikuje názov vytváranej oblasti záujmu a pridá sa jej detailnejší popis (Obrázok 4).



The image shows a web form titled "Area of interest" with a close button (X) in the top right corner. The form contains the following fields:

- XPath**: A text input field containing the value `/HTML[1]/BODY[1]/H1[1]`.
- Page**: A text input field containing the value `file:///C:/Users/Pietro/Desktop/test/alert.html`.
- Name**: An empty text input field.
- Description**: An empty text area.

A "SAVE" button is located at the bottom right of the form.

Obrázok 4: Formulár na vytváranie oblasti záujmu.

Kliknutím na tlačidlo SAVE sa odošle požiadavka na vytvorenie novej oblasti záujmu na server, ktorý podľa zaslaných údajov o oblasti a používateľovi vytvorí a uloží nový záznam ku konkrétnemu sedeniu a projektu. Následne je možné pokračovať v definovaní ďalších oblastí záujmu predstaveným postupom. Kliknutím na aktívnu ikonu (zelené oko) je možné ukončiť proces vytvárania oblastí záujmu.

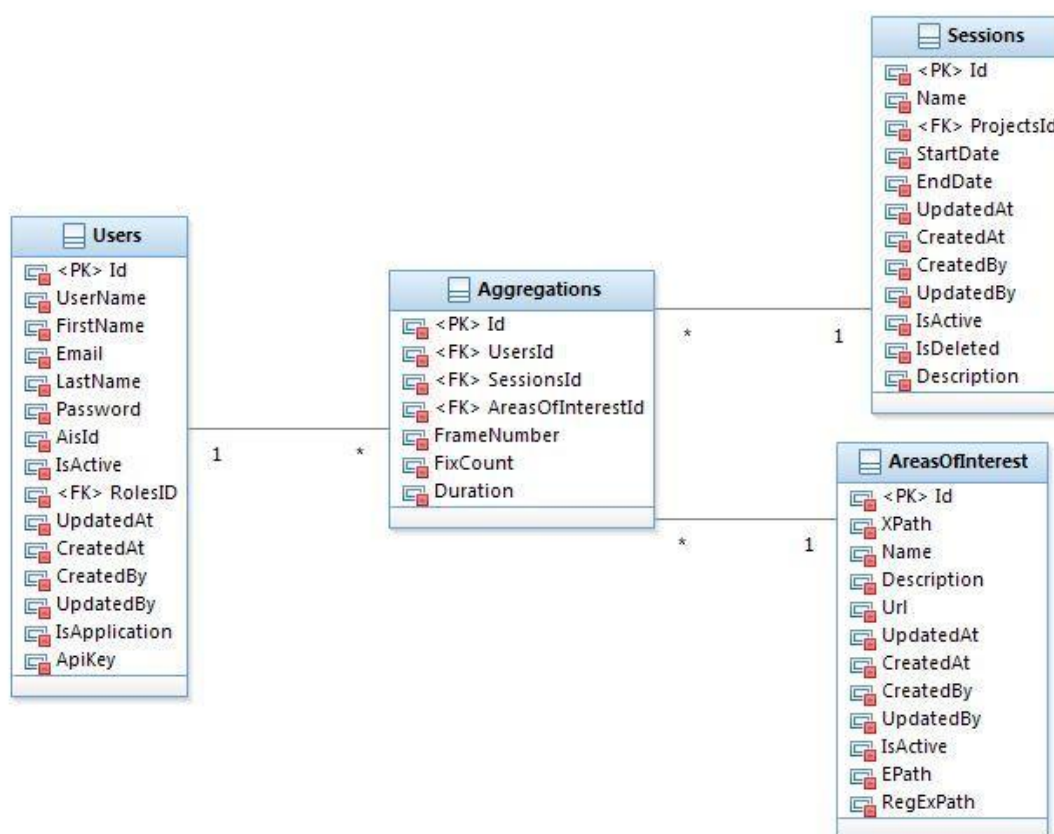
4.2.6 Zhrnutie

Podarilo sa nám tak vytvoriť prenosnú knižnicu, ktorá sleduje užívateľa a na základe toho ho upozorní či sa do nami zvolenej oblasti už pozrel alebo či ju odignoroval, čo nám prináša veľké využitie. Rovnako bola pridaná časť pre definovanie oblasti záujmu pre testovanú stránku. Testovanie kódu pomocou nástroja sa nám však z časových dôvodov zatiaľ nepodarilo implementovať. Avšak vykonali sme podrobnú analýzu možných nástrojov na testovanie, z ktorých sme si vybrali nástroj JASMINE. Samotná analýza tohto nástroja sa nachádza v prílohe A.

4.3 Agregácie

4.3.1 Implementácia

MSSQL databáza bola rozšírená o novú tabuľku Aggregations, v ktorej budú ukladané priebežné agregované dáta pre konkrétne sedenie (SessionsId), probanta (UsersId) a záujmovú oblasť (AreasOfInterestId). Agregované dáta predstavujú počet fixácií pohľadom na konkrétny AOI (FixCount) a čas trvania týchto fixácií (Duration). Nakoľko sa počas jedného sedenia budú vytvárať agregácie v 1 minútových časových intervaloch, poradie týchto agregácií bude uložené vo (FrameNumber) (pozri Obrázok 5)



Obrázok 5: Dátový diagram agregácií.

Z dátového modelu bola vygenerovaná trieda Aggregations, slúžiaca pre potreby ukladania a načítania agregáčnych objekt z MSSQL databázy a pre potreby kontextu Entity frameworku. Pre prácu s agregáčnymi bola vytvorená trieda AggregationDto (Data transfer object). Všetky operácie nad agregáčnymi objektmi sú obsiahnuté v triede AggregationsRepository, implementujúcej rozhranie IAggregationRepository. Trieda momentálne obsahuje nasledovné dopytovacie metódy, ktoré bude možné kedykoľvek v prípade potreby rozšíriť o ďalšie:

- **GetAggregationById (int Id)**
 - vráti agregáčny objekt so zadaným Id
- **AddAggregation (AggregationDto aggregation)**
 - uloží agregáčny objekt do databázy

- **GetAggregationsForSession (int sessionId)**
 - vráti všetky agregáčn  objekty pre dan  sedenie
- **GetAggregationsForSessionAndUser (int sessionId, int userId)**
 - vráti všetky agregáčn  objekty dan ho probanta pre dan  sedenie
- **GetAOITimeStatistic (int AOIId, int sessionId)**
 - vráti pre zadan  AOI a pre zadan  sedenie celkov  hodnoty po tu a trvania fixáci  v etk ch probantov po as cel ho trvania sedenia.
- **GetAOITimeStatistic (int AOIId, int sessionId, int startTime, int endTime)**
 - vráti pre zadan  AOI a pre zadan  sedenie celkov  hodnoty po tu a trvania fixáci  v etk ch probantov po as trvania sedenia v  asovom rozsahu startTime a endTime.

Pre potreby vytv rania agregáci  boli triedy AreaOfInterestRepository, AnnotationRepository, SessionRepository a k nim pridr zen  rozhrania roz iren  o nov  metody dopytuj ce sa do datab zy MSSQL.

AreaOfInterestRepository:

- **GetSessionAreaOfInterestId (int sessionId)**
 - vráti ID v etk ch aktivn ch oblast  z ujmu pre zadan  sedenie

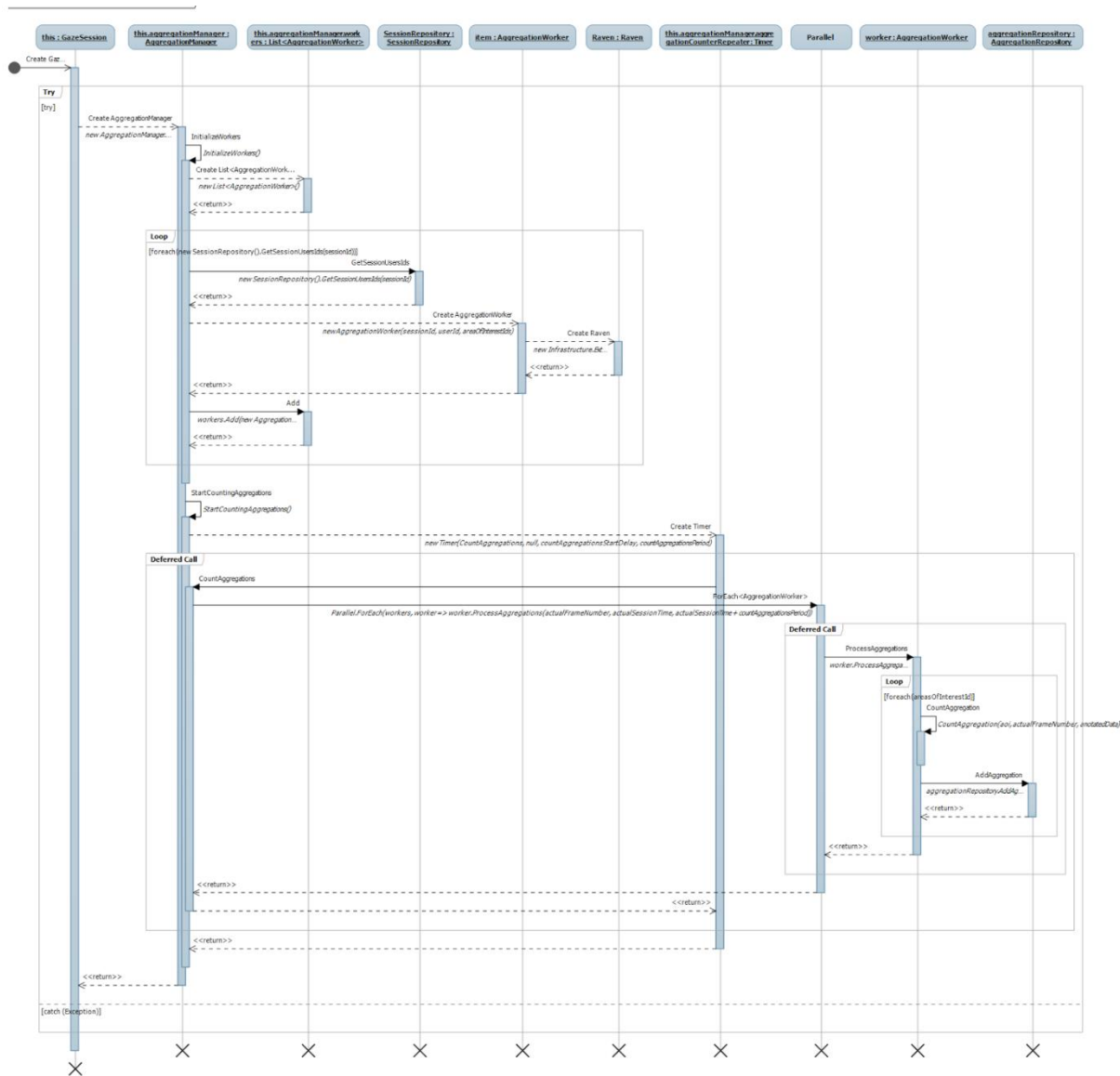
AnnotationRepository:

- **GetAnnotationsForSessionAndAOI(int sessionId, int AOIId)**
 - vráti v etky anota n  objekty pre dan  oblasť z ujmu a dan  sedenie

SessionRepository:

- **GetSessionUsersIDs(int id)**
 - vráti ID v etk ch probantov pre dan  sedenie

Proces vytv ranie agregáci  za ina v triede GazeSession, ktorá v kon truktoze vytv ra okrem in tancie AnnotationManager (star  sa o anotovanie pr du d t) aj nami vytvoren  in tanciu triedy AggregationManager, ktorej kon truktor je naplnen  ID aktu lnego sedenia a aktu lnego  asu. Tento kon truktor m  na starosti vytvorenie kolekcie in tanci  novovytvorenej triedy AggregationWorker, ktor ch po et je rovn  po tu probantov dan ho sedenia. Takto sa pri vytv ran  agregaa n workera do jeho kon truktoru pred  ID sedenia, Id probanta a kolekcia ID oblast  z ujmu pre dan  sedenie. N sledne AggregationManager spust  cyklick   asova , ktor  po uplynut  1 min ty za ina v 1-min tov ch intervaloch volať metodu CountAggregation, ktorá asynchr nne vol  metody ProcessAggregations v etk ch agregaa n workerov. Ka d  agregaa n worker na ita z RavenDB datab zy anotovan  d ta vzniknut  za posledn  min tu pre svoje sedenie a probanta. N sledne pre ka d  oblasť z ujmu vypo ita zo ziskann ch d t agregaa cie a ulo i ich do MS SQL datab zy. Po skon en  sedenia AggregationManager nech  posledn  raz zbehn ť spo itanie agregaa ci  a ukon i proces.



Obrázok 6: Sekvenčný diagram pre vytváranie agregácií.

4.4 Vizualizácie

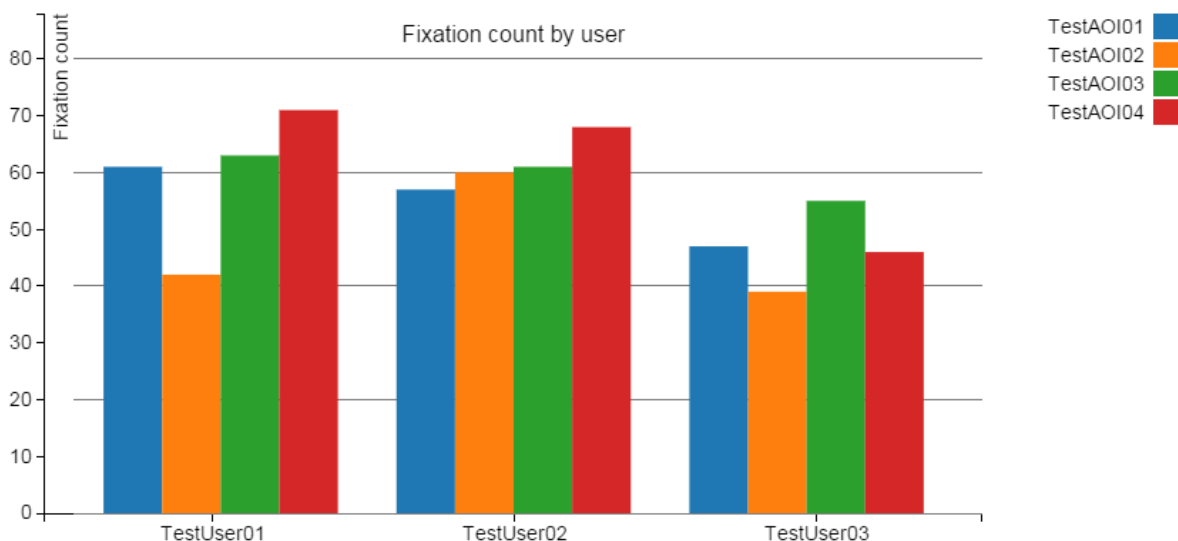
Existujúca implementácia štatistík bola z dôvodu predchádzajúceho vytvorenia agregácií prerobená tak, aby miesto pôvodne vypočítavaných štatistík zo surových eye-tracker dát boli tieto hodnoty vypočítavané z dát agregovaných. Preto bolo získavanie surových dát z RavenDB nahradené za dopyty do MSSQL databázy, čo malo za následok aj zjednodušenie celkového vypočítavania štatistických dát a tým aj zjednodušenie a sprehľadnenie zdrojových kódov.

Ďalšou zmenou oproti predchádzajúcemu riešeniu boli zmeny, ktoré sme zapracovali do spôsobu filtrovania zobrazených dát vo filtri štatistík. Modálne výberové okná boli nahradené za prehľadnejšie rozbaľovacie výberové polia.

Samotné zobrazovanie štatistík bolo zmenené do podoby, keď sa k vybraným sedeniam zobrazuje prehľad počtu k nim priradených používateľov ako aj preddefinovaných oblastí záujmu. Rovnako tak sa pre každé sedenie zobrazuje prehľad počtu fixácií a dĺžky ich trvania pre každú kombináciu používateľa a oblasti záujmu.

Pôvodne existujúci tab pre grafy je v novej implementácii rozdelený do viacerých tabov, z ktorých každý obsahuje grafy pre znázornenie inej skúmanej veličiny, momentálne sú pripravené taby pre zobrazovanie grafov „počtu fixácií“ a pre grafy „dĺžky trvania fixácií“. Každý z týchto grafových tabov obsahuje prepínač medzi sedeniami. Pri výbere sedenia v prepínači sa pre vybrané sedenie vykreslia vždy tri grafy:

- Prehľad celkovej sumy vybranej metriky pre všetky oblasti záujmu
- Závislosť vybranej metriky od oblasti záujmu
- Závislosť vybranej metriky od používateľa



Obrázok 7: Graf počtu fixácií podľa používateľa pre všetky oblasti záujmu.

4.5 Vzory čítania

Analýza čítania vychádza z algoritmu predstaveného v článku A Robust Algorithm for Reading Detection od Campbella. Fungovanie algoritmu je založené na prúdovom spracovaní fixačných bodov. Kde analýza prebieha nad rozdielmi vlastností medzi dvoma fixačnými bodmi. V globále existovala premenná, ktorá reprezentovala bodové ohodnotenie, či je identifikované čítanie. Bodové hodnotenie fungovalo na základe predpokladu ideálnosti pohybu aký nastáva pri čítaní. Teda najideálnejší je krátky pohyb v smere textu, plus treba brať do úvahy aj prechod pohľadu na nový riadok. Nasleduje tabuľka (pozri Tabuľka 1) ohodnotenia jednotlivých pohybov fixácií. Reset predstavuje znovu nastavenie hodnoty na nulu.

Tabuľka 1: ohodnotenia fixácií.

Pohyb	Bodové ohodnotenie
Krátky pohyb doprava	+10
Stredný pohyb doprava	+5
Veľký pohyb doprava	reset
Krátky pohyb doľava	+5
Stredný pohyb doľava	-5
Veľký doľava s kúsok nadol	+5
Zvyšok	reset

Krátky pohyb doľava má v rámci chyby merania ohodnotenie +5, lebo môže ísť o chybu merania. Štandardný prah nad ktorý som považoval sekciu za čítanie bol nastavený na 15 bodov. Čítanie bolo identifikované iba ak je väčšie ako tento prah.

Vstupom algoritmu je teda vektor fixačných bodov spolu s id oblasti záujmu. Okrem toho je tam začiatkový a koncový index a tiež aj počet fixácií. Parametre, ktoré sú štandardne nastavené, ale je možné ich dodatočne upraviť sú nasledovné (pozri Tabuľka 2).

Tabuľka 2: Parametre.

Názov premennej	Štandardná hodnota	Opis
lineHeight	20	Výška riadku
shortParameter	20	Veľkosť krátkého pohybu
mediumParameter	40	Veľkosť stredného pohybu
longParameter	60	Veľkosť dlhého pohybu
horizontalDeviation	20	Horizontálna odchýlka
verticalDeviation	20	Vertikálna odchýlka
pointLimit	15	Bodový hodnota čítania
DistanceLimitForRepeatedReadingFixation	40	Vzdialenosť pri opakovanom čítaní
fontSize	12	Veľkosť fonu

Algoritmus detekcie opakovaného čítania

Opakované čítanie je podmnožina čítania. Preto detekcia opakovaného čítania pracovala iba nad množinou považovanou za čítanie. Algoritmus funguje nasledovne. Je zostavený z viacerých cyklov. Vonkajší začína nie od prvej fixácie, ale od tretej a túto fixáciu porovnávam so všetkými predošlými v poradí od prvej. Porovnanie spočíva nad ich vzdialenosťou. V prípade, že takáto vzdialenosť je menšia tak sa vbehne do ďalšieho cyklu, ktorý porovnáva vzdialenosti v rámci obidvoch sekvencií. Tento cyklus beží buď po poslednú fixáciu, alebo pokým je vzdialenosť v rámci tolerancie vzdialenosti, považovanej za opakované čítanie. Okrem vzdialenosti sa berie do úvahy aj to, aby išlo o fixáciu v rovnakom riadku. Po dobehnutí cyklu sa v prípade, že bolo identifikovaných viac ako dve fixácie v rámci tolerancii, som prehlásil dané fixácie za opakované čítanie. Tieto fixácie boli označené, aby sa predišlo znovu označeniu tých istých fixácií.

Po dobehnutí oboch algoritmov, nasledovalo počítanie štatistík. Toto fungovalo tak, že len sa pripočítavalo počítadlo pre jednotlivé fixácie, podľa toho kam fixácia patrila, v rámci vektorov čítania a skákania. Výsledkom boli štatistiky o počte fixačných bodov typov čítania, skákania a zvyšku pre jednotlivé oblasti záujmu.

Implementácia v rámci infraštruktúry

Na podporu detekcie čítania som implementoval dve dátové entity. Prvá je tabuľka reprezentujúca stav počítania analýzy čítania. Druhá už sú samotné dáta. K manažovaniu dát som implementoval do vrstvi infraštruktúry rozhranie a triedu `ReadingAnalyticsRepository` a `ReadingStateRepository`.

ReadingAnalyticsRepository

Obsahuje nasledujúce metódy:

- `MessageDto AddReadingAnalytics(ReadingAnalyticsDto readingAnalytics);`
Realizuje pridanie štatistík do databázy.
- `List<ReadingAnalyticsDto> getReadingAnalyticsForSession(int sessionId);`
Vráti všetky štatistiky o čítaní pre dané sedenie.

ReadingStateRepository

Obsahuje nasledujúce metódy:

- `MessageDto addNewSession(Sessions sessions);`
Pridanie nového sedenia. Stav je štandardne nastavený na nevypočítaný.
- `ReadingStateDto getSessionState(Sessions sessions);`
Vráti stav pre dané sedenie.
- `MessageDto setSessionStateToFinished(Sessions sessions);`
Nastaví stav na vypočítaný.

Nasledujú triedy vrstvy aplikácie zastrešujúce čítanie..

ReadingAnalyticsManager

Realizuje manažment počítania analýzy čítania. Obsahuje nasledujúce metódy:

- `private void processReadingAnalytics()`
Spustí samotné počítanie štatistík, od naplňovania dát až po vypočítanie.
- `public void startCountingReadingAnalytics(int sessionId)`
Verejná metóda, ktorá realizuje neblokujúce volanie spustenia predošlej metódy počítania.

ReadingAnalyticsWorker

Obsahuje triedu počítania štatistík čítania, ktorá obsahuje nasledujúce metódy:

- `public void setFontSize(int fontSize)`
Štandardné nastavenie veľkosti písma je dvanásť. Táto trieda umožňuje prispôbiť si veľkosť fonu.
- `public void setPointLimit(int limit)`
Nastavuje veľkosť tolerancie za ktorou je sekcia považovaná za čítanie.

-public List<ReadingAnalyticsDto> ProcessReadingAnalytics(List<Fixation> fixations, int startIndex, int endIndex, int numberOfAOIs, int sessionId)

Realizuje samotné počítanie. Na vstupe sú fixácie v type vnhiezdenej triedy, začiatkový a koncový index počítania a id sedenia. Výstupom je list štatistík pre jednotlivé oblasti záujmu.

Okrem toho má vnhiezenú triedu fixácií, kvôli synchronizácií očakávaného vstupu a taktiež triedu výstupných štatistík.

ReadingAnalyticsStateManager

Obsahuje nasledujúce metódy:

-public void addNewSessionState(int sessionId)

Pridá nový záznam stavu sedenia pre dané sedenie.

4.6 IVT filter

Velocity-Threshold Identification (I-VT) algoritmus pre klasifikáciu fixácií je klasifikačný algoritmus založený na rýchlosti. Základná myšlienka I-VT filtra je klasifikovať pohyby oka na základe rýchlosti posunov oka. Rýchlosť je najčastejšie uvádzaná vo vizuálnych stupňoch za sekundu ($^{\circ} / s$). Ak je nad určitou hranicou rýchlosti, tak je tento pohyb oka klasifikovaný ako sakáda (saccade) a ak sa nachádza pod touto hranicou tak je považovaný za súčasť fixácie (fixation).

Pri samotnej implementácii I-VT filtra sme pracovali s funkciami v jazyku Matlab. Na základe týchto funkcií sme následne reimplementovali algoritmus do jazyka C#, ktorý používame v našom projekte.

I-VT filter pozostáva z 5 základných funkcií:

- Výber oka (angl. eye selection)
- Vypĺňanie medzier (angl. gap fill in)
- Hľadanie medzier (angl. check for gaps)
- Zlúčenie fixácií (angl. merge fixations)
- Vizualizácia uhlov (angl. visual angle)

4.6.1 Výber oka

V ideálnom prípade pri klasifikovaní fixácií predpokladáme, že sa obe oči správajú identicky. Avšak v skutočnosti sa obe oči nie vždy správajú rovnako. Často existuje nepatrný rozdiel medzi jedným a druhým okom, pokiaľ ide o začiatok a koniec doby fixácie, rovnako ako pri žmurkaní. Taktiež sa môžu objaviť prípady, keď existujú medzery v údajoch z jedného oka a z druhého nie. Preto sa musí zvoliť spôsob akým sa budú dáta z oboch očí konvertovať na jediný súbor dát pri klasifikácii vzoriek.

V kontexte implementácie je možné dosiahnuť tri základné možnosti (pozri Obrázok 8):

- Ľavé oko – iba ľavé oko bude uvažované pri výpočte
- Pravé oko - iba pravé oko bude uvažované pri výpočte
- Priemer –priemerná hodnota z ľavých a pravých pozícií oka, iba ľavého alebo iba pravého oka alebo žiadneho z nich ak nie sú validné.


```

switch (eyeSelection.ToLower())
{
    case "left":
        data.ValidityEyeSelected = data.ValidityLeft;
        data.EyePositionEyeSelected3Dx = data.EyePosLeft3Dx;
        data.EyePositionEyeSelected3Dy = data.EyePosLeft3Dy;
        data.EyePositionEyeSelected3Dz = data.EyePosLeft3Dz;
        data.GazePointEyeSelected3Dx = data.GazePointLeft3Dx;
        data.GazePointEyeSelected3Dy = data.GazePointLeft3Dy;
        data.GazePointEyeSelected3Dz = data.GazePointLeft3Dz;
        data.GazePointEyeSelected2Dx = data.GazePointLeft2Dx;
        data.GazePointEyeSelected2Dy = data.GazePointLeft2Dy;
        break;
    case "right":
        data.ValidityEyeSelected = data.ValidityRight;
        data.EyePositionEyeSelected3Dx = data.EyePosRight3Dx;
        data.EyePositionEyeSelected3Dy = data.EyePosRight3Dy;
        data.EyePositionEyeSelected3Dz = data.EyePosRight3Dz;
        data.GazePointEyeSelected3Dx = data.GazePointRight3Dx;
        data.GazePointEyeSelected3Dy = data.GazePointRight3Dy;
        data.GazePointEyeSelected3Dz = data.GazePointRight3Dz;
        data.GazePointEyeSelected2Dx = data.GazePointRight2Dx;
        data.GazePointEyeSelected2Dy = data.GazePointRight2Dy;
        break;
    case "average":

```

Obrázok 8: Výber oka.

4.6.2 Vyplňanie medzier

Ďalším problémom, ktorý je potrebné zväžiť pri klasifikácii fixácií s dátami zo sledovania pohľadu je, že v digitálnych meracích systémov je takmer nevyhnutné, že dôjde k strate niektorých dát. To znamená, že vzorka nemôže byť získavaná v každom momente merania.

V kontexte sledovania pohľadu je strata dát často spôsobená žmurkaním, odvrátením pohľadu alebo vloženým predmetom medzi očami a kamerou. Tieto straty dát sa následne prejavajú ako krátke medzery. Avšak straty dát nastávajú aj z dôvodu oneskorenia pri prenose dát alebo z dôvodu odrazov zo zrkadla a podobne, pričom tieto straty sú kratšie než ostatné straty dát.

V prípade, že nastane strata dát v strede fixácie, tak by mohla byť fixácia klasifikovaná ako dve samostatné fixácie. Preto je potrebné tieto medzery vyplňať.

V kontexte implementácie sa sledujú začiatky a konce fixácií a medzier (pozri Obrázok 9) a na základe nich sa menia typy udalostí (*eventType*) aj ich trvania (*eventDuration*) a tak sa vlastne vyplňajú medzery.

```

// Calculate visual angle between fixation and merge if needed
for (i = 0 ; i < gapStart.Length ; i++){
    visualAngle[i] = visualanglefixation(data, fixationPointXmm, fixationPointYmm, fixationPointZmm, gapStart[i], gapFinish[i]);
    j = gapStart[i] - 1;

    if (visualAngle[i] < maxAngleBtFix){
        for (k = gapStart[i]; k <= gapFinish[i]; k++){
            {
                eventType[k] = 1;
            }
        }

        // Look for the beginning of the previous fixation
        while (eventType[j]==1 && j > 1){
            j = j-1;
        }
        FixationStart = j+1;
        // Look for the end of the previous fixation
        j = gapFinish[i]+1;
        while ( ( eventType[j]==1) && (j <= eventType.Length) ){
            j = j+1;
        }
        FixationFinish = j - 1;
        for (k = FixationStart; k <= FixationFinish; k++) {
            fixationPointX[k] = Math.Round((fixationPointX[gapStart[i] - 1] + fixationPointX[gapFinish[i] + 1]) / 2);
            fixationPointY[k] = Math.Round((fixationPointY[gapStart[i] - 1] + fixationPointY[gapFinish[i] + 1]) / 2);
            eventDuration[k] = eventDuration[gapStart[i] - 1] + eventDuration[gapStart[i]] + eventDuration[gapFinish[i] + 1];
        }
    }
}
}
}

```

Obrázok 9: Vypĺňanie medzier.

Samotné vyhľadávanie medzier sa vykonáva vo funkcii *checkforgaps()*, ktorej časť je zobrazená na obrázku (pozri Obrázok 10).

```

//Check the validity of the data and calculate the gaps starting and finish samples
for (i = 1; i < Validity.Length; i++)
{
    if (Validity[i] > 0.0 && gap == 0){
        // If we are in a gap and last sample wasn't, register the starting sample of the possible gap
        if (Validity[i-1] < 1 || i == 1)
        {
            gapStart[n] = i;
            if (numGapStart <= numGapFinish){
                numGapStart++; //MS
            }
            gap = 1;
        }
    }else{
        // If the sample is not missed, we assess if the last sample was missing
        if (Validity[i-1] > 0 && gap == 1)
        {
            // If the length is less than the maxGapLength, register the finish
            if (maxGapLength >= ((TimeStamp[i] - TimeStamp[gapStart[n]]) / 1000))
            {
                gapFinish[n] = i-1;
                numGapFinish++; //MS
                n = n + 1;
                gap = 0;
            }
        }
    }
}
}

```

Obrázok 10: Vyhľadanie medzier.

4.6.3 Použité objekty

Na účely tohto algoritmu sme si vytvorili aj samostatné objekty reprezentujúce údaje získané zo sledovania pohľadu (Obrázok 11).

```

class EyeTrackingData
{
    public int[] trigger;
    public double[] EtTimeStamp;
    public double[] LocalTimeStamp;
    public double[] EyePosLeft3Dx;
    public double[] EyePosLeft3Dy;
    public double[] EyePosLeft3Dz;
    public double[] EyeRelLeft3Dx;
    public double[] EyeRelLeft3Dy;
    public double[] EyeRelLeft3Dz;
    public double[] GazePointLeft2Dx;
    public double[] GazePointLeft2Dy;
    public double[] GazePointLeft3Dx;
    public double[] GazePointLeft3Dy;
    public double[] GazePointLeft3Dz;
    public double[] PupilLeft;
    public double[] ValidityLeft;
    public double[] EyePosRight3Dx;
    public double[] EyePosRight3Dy;
    public double[] EyePosRight3Dz;
    public double[] EyeRelRight3Dx;
    public double[] EyeRelRight3Dy;
    public double[] EyeRelRight3Dz;
}

class FilterParameters
{
    public int windowLength;
    public int velocityThreshold;
    public int maxGapLength;
    public String eyeSelection;
    public double maxAngleBtFix;
    public int maxGapBtFix;
    public int minFixDuration;

    public FilterParameters()
    {
        windowLength = 20;
        velocityThreshold = 30;
        maxGapLength = 75;
        eyeSelection = "average";
        maxAngleBtFix = 0.5;
        maxGapBtFix = 75;
        minFixDuration = 60;
    }
}

```

Obrázok 11: Použité objekty.

5 Inštalčná príručka

V tejto kapitole uvediem inštalčné príručky pre jednotlivé moduly nášho systému. Tieto príručky vypracoval minuloročný tím a keďže sa nič pri inštalácii nezmenilo nevypracovávali sme nové.

5.1 Inštalčná príručka pre browser addon

Inštalácia rozšírenia pre prehliadač je pomerne jednoduchá. V prípade Google Chrome, ale aj Mozilla Firefox je nutné stiahnuť požadovanú verziu rozšírenia.

5.1.1 Krok 1 - stiahnutie inštaláčného súboru

Súbory potrebné pre inštaláciu rozšírení je možné stiahnuť z webovej stránky:

<http://team01-13.ucebne.fiit.stuba.sk/web/Home/Downloads>

Podľa zvoleného prehliadača je možné stiahnuť dve verzie:

- Google Chrome verzia
- Firefox verzia

Súbory stiahnite na ľahko prístupne miesto, napríklad na plochu.

5.1.2 Krok 2 - rozbalenie inštaláčného súboru

Stiahnutý súbor je nutné rozbaľiť použitím komprimačného nástroja, napríklad WinRAR (Pravdepodobne však postačí aj základný nástroj dodávaný s Windows). Obsah súboru si dočasne umiestnite na plochu. V prípade Firefox verzie súbor obsahuje iba jednu inštalčku s príponou .xpi a Google Chrome verzia obsahuje dve inštalčky (klientskú a administrátorskú verziu) s príponou .crx.

5.1.3 Krok 3 - spustenie inštaláčného súboru a postup inštalácie

Inštalácia rozšírenia prebieha jednoducho. V prvom rade je nutné otvoriť požadovaný internetový prehliadač (podľa verzie rozšírenia). Ak sa jedná o firefox, súbor s príponou .xpi stačí uchytením na ploche presunúť do okna prehliadača. V prípade Google Chrome je nutné otvoriť v prehliadači panel rozšírení (Menu / Nástroje / Rozšírenia). Následne stačí opäť premiestniť požadovaný súbor z plochy do okna prehliadača.

5.2 Inštalčná príručka pre CarrotsClient

V tejto časti je popísané, ako aplikáciu CarrotsClient nainštalovať.

5.2.1 Krok 1 - stiahnutie inštaláčného súboru

Inštaláčny súbor je možné stiahnuť na adrese: <http://team01-13.ucebne.fiit.stuba.sk/web/Home/Downloads>

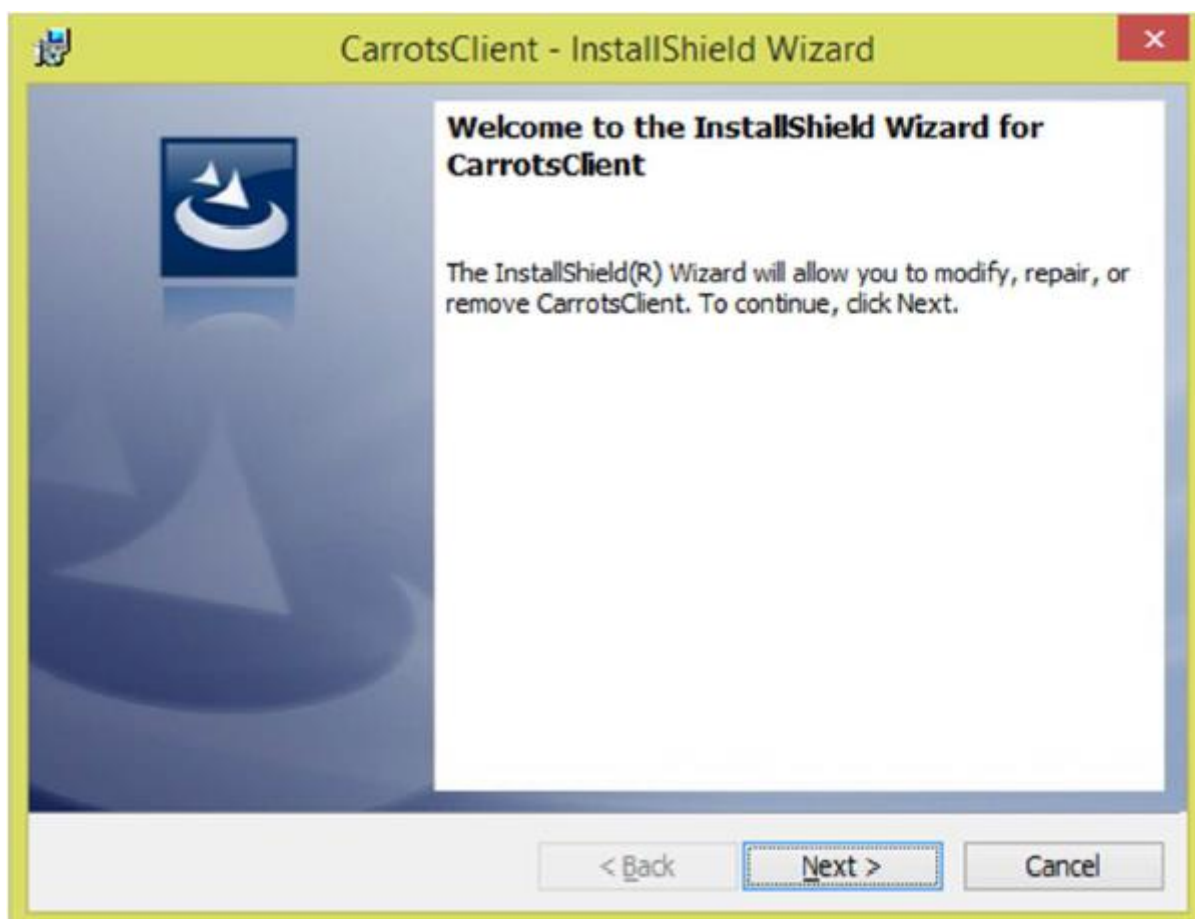
V sekcii Desktop Client. Súbor sa stiahne zabalený vo formáte zip a je ho potrebné rozbaľiť.

5.2.2 Krok 2 - rozbalenie inštaláčného súboru

Súbor je najprv potrebné rozbaľiť pomocou vášho obľúbeného komprimačného programu. Použiť je možné napr. [7-zip](#).

5.2.3 Krok 3 - spustenie inštaláčného súboru a postup inštalácie

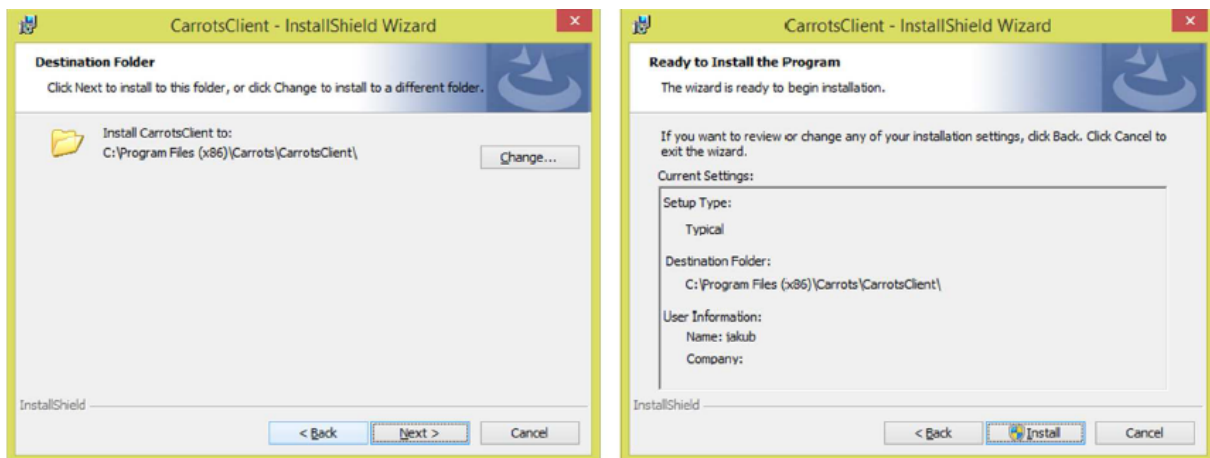
Po rozbaľení inštaláčného súboru ho spustíme dvojklikom a zobrazí sa nám nasledovné okno.



Obrázok 12: Uvítacie okno inštalátora.

V tomto okne jednoducho klikneme na Next > a pokračujeme ďalej.

V ďalšom okne si môžeme zvoliť adresár, do ktorého sa má aplikácia nainštalovať, kliknutím na Browse. Keď sme spokojný s adresárom, klikneme znova na Next > a následne na Install.



Obrázok 13: Výber inštaláčného adresára a spustenie inštalácie.

Po prebehnutí inštalácie si na poslednej obrazovke môžeme vybrať, či chceme nainštalovanú aplikáciu aj priamo spustiť a kliknutím na Finish inštaláciu ukončíme. Inštalátor automaticky vytvorí pre spustenie aplikácie odkaz na pracovnej ploche.



Obrázok 14: Odkaz na spustenie aplikácie na pracovnej ploche.

5.3 Inštalčná príručka pre simulátor

Táto časť uvádza postup inštalácie aplikácie simulátora.

5.3.1 Krok 1 - stiahnutie inštalateľného súboru

Stiahneme aktuálnu verziu Simulátora zo stránky projektu: <http://team01-13.ucebne.fiit.stuba.sk/web/Home/Downloads>

5.3.2 Krok 2 - rozbalenie inštalateľného súboru

Súbor rozbalíme pomocou obľúbeného archivovacieho nástroja.

5.3.3 Krok 3 - spustenie inštalateľného súboru a postup inštalácie

Spustíme inštalateľný súbor aplikácie a riadime sa pokynmi sprievodcu inštalácie.

5.4 Inštaláčn prručka pre aplikáciu GazeAdmin

Inštalácia webovej aplikácie Gaze admin je pomerne jednoduchá, stačí zo zdrojového kódu vytvoriť release verziu projektu ViewTracking.Web, a aby fungovala aplikačná vrstva tak tiež projektu ViewTracking.Wcf. Nasledne upnúť do IIS na serveri obe aplikácie zvlášť a nastaviť v projekte web v jeho webconfigu cestu na svc subor Wcf projektu.

5.4.1 Krok 1 publishovanie ViewTracking.Wcf

Po pravom kliknutí na projekt sa zobrazí kontextové menu kde sa nachádza publish po nastavení suboru pre publish stačí skopirovať kódu.

5.4.2 Krok 2 nastavovanie webconfigu Viewtracking.Wcf

1. Nastavenie connection stringu na databázu:

```
<add name="ModelContainer" connectionString="ms sql metadata" />
```

2. Nastavenie enpointov pre raven db v časti client:

```
<endpoint address="net.pipe://localhost/raven/GazeDataService.svc"
contract="GazeDataService.IGazeDataService" name="NetNamedPipeBinding_Raven" />
```

3. Nastavenie Entity Frameworku v časti configuration:

```
<section name="entityFramework"
type="System.Data.Entity.Internal.ConfigFile.EntityFrameworkSection, EntityFramework,
Version=6.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
requirePermission="false" />
<entityFramework>
  <defaultConnectionFactory
type="System.Data.Entity.Infrastructure.LocalDbConnectionFactory,
EntityFramework">
<parameters>
  <parameter value="v11.0" />
</parameters>
</defaultConnectionFactory>
<providers>
<provider invariantName="System.Data.SqlClient"
type="System.Data.Entity.SqlServer.SqlProviderServices, EntityFramework.SqlServer" />
</providers>
</entityFramework>
<runtime>
  <assemblyBinding xmlns="urn:schemas-microsoft-com:asm.v1">
  <dependentAssembly>
  <assemblyIdentity name="EntityFramework"
```



```

    publicKeyToken="b77a5c561934e089" culture="neutral" />
    <bindingRedirect oldVersion="0.0.0.0-6.0.0.0" newVersion="6.0.0.0" />
  </dependentAssembly>
</assemblyBinding>
</runtime>

```

4. Nastavenie service model časti vo webconfigu:

```

<system.serviceModel>
  <client>
    binding="netNamedPipeBinding"
    bindingConfiguration="NetNamedPipeBinding_Raven"
    contract="GazeDataService.IGazeDataService" name="NetNamedPipeBinding_Raven"
  />
  </client>
  <bindings>
    <basicHttpBinding>
      <binding name="BasicHttpBinding_IGazeDataService" />
      <binding name="BasicHttpsBinding_IGazeDataService">
        <security mode="Transport" />
      </binding>
    </basicHttpBinding>
    <netNamedPipeBinding>
      <binding name="NetNamedPipeBinding_Raven" />
    </netNamedPipeBinding>
    <webHttpBinding>
      <binding name="pokus">
        <security mode="Transport"/>
      </binding>
      <binding name="http" />
    </webHttpBinding>
  </bindings>
  <services>
    <service behaviorConfiguration="serviceBehavior"
      name="ViewTracking.Wcf.ViewTrackingService">
      <endpoint address=""
        binding="webHttpBinding" bindingConfiguration="pokus"
        behaviorConfiguration="pokus2"
        contract="ViewTracking.Wcf.IViewTrackingService" />
      <endpoint address="" binding="webHttpBinding" bindingConfiguration="http"

```

```

behaviorConfiguration="pokus2" contract="ViewTracking.Wcf.IViewTrackingService"
/>
</service>
</services>
<behaviors>
<serviceBehaviors>
<behavior name="serviceBehavior">
<serviceMetadata httpsGetEnabled="true" httpGetEnabled="true"/>
<serviceDebug includeExceptionDetailInFaults="true"/>
</behavior>
</serviceBehaviors>
<endpointBehaviors>
<behavior name="pokus2">
<webHttp/>
</behavior>
</endpointBehaviors>
</behaviors>
<serviceHostingEnvironment multipleSiteBindingsEnabled="true" />
</system.serviceModel>

```

5. Uputie do IIS:

- pravým klikom na stránku v IIS a klik na Add application
- nastaviť cestu k suboru
- nastaviť application pool na .net framework 4.0 classic
- nastaviť názov, ktorý bude v URL

5.4.3 Krok 3 publishovanie ViewTracking.Wcf

Po pravom kliku na projekt sa zobrazí kontextové menu, kde sa nachádza publish po nastavení súboru pre publish stačí skopírovať kód.

5.4.4 Krok 4 nastavenie web configu ViewTracking.Web

1. Nastavenie connection stringu pre WCF službu

Name parameter musí byť rovnaký ako v uvedenom príklade:

```

<add name="WcfConnnection" connectionString="http://team01-
13.ucebne.fiit.stuba.sk/wcf/ViewTrackingService.svc/" />

```

2. Nastavenie autentifikačnej chybovej správy:

```

<authentication mode="Forms"><forms loginUrl="~/account/login" timeout="2880"
/></authentication>

```

3. Uputie do IIS:

- pravým kliknutím na stránku v IIS a klik na Add application
- nastaviť cestu k súboru

- c. nastaviť application pool na .net framework 4.0 classic
- d. nastaviť názov, ktorý bude v URL

6 Používateľská príručka

V tejto časti uvedieme používateľské príručky pre jednotlivé časti nášho systému.

6.1 Používateľská príručka pre browser add-on

Hlavná funkcionálnosť oboch verzií rozšírení je rovnaká, avšak existujú menšie rozdiely v prevedení rozšírení. Niektoré odlišnosti pri používaní rozšírení sú uvedené v ďalšej časti.

6.1.1 Základná funkcionálnosť

Rozšírenie pre prehliadač plní dve funkcie.

- Hlavnou funkciou je obohacovanie dát zo sledovania pohľadu, ktoré zo zariadenia získava náš desktopový klient. Nazbierané dáta sú odoslané rozšíreniu a to následne pre jednotlivé súradnice pohľadu k dátam priradí identifikátor konkrétneho elementu stránky, na ktorý sa používateľ pozerá.
- Druhou funkcionálnosťou je umožnenie vytvárania takzvaných oblastí záujmu. Tie sú reprezentované identifikátorom elementu, ktorý tvorí ohraničenie oblasti, url adresou stránky a doplnujúcimi informáciami ako sú názov a popis.

Obohacovanie dát zo sledovania je možné jednoduchým aktivovaním rozšírenia v prehliadači. Definovanie oblastí záujmu vyžaduje prihlásenie sa cez gazeadmin a zvoliť si sedenie, pre ktoré chceme oblasť definovať (Projects -> Edit project / Sessions -> Edit session / Areas Of Interest). V pravej časti záložky sa nachádza ovládacie tlačidlo pre aktiváciu rozšírenia.



Obrázok 15: Tlačidlo pre aktiváciu a deaktiváciu rozšírenia.

Po aktivovaní rozšírenia kliknutím na tlačidlo je možné vytvárať oblasti záujmu (nie na gazeadmin stránke). Pozastavenie a aktivovanie rozšírenia pre posledné aktivované sedenie je možné kliknutím na ikonku rozšírenia v prehliadači.

Zmenu sedenia, pre ktoré je vyžadované vytvorenie oblasti záujmu je možné urobiť len cez gazeadmin podobne ako pri prvej aktivácii. Rozšírenie dokáže editovať oblasti len pre jedno sedenie súčasne. Aktivovaním editácie pre sedenie sa zmení stav editácie predchádzajúceho sedenia na neaktívny.

6.1.2 Odlišnosti v implementácii rozšírení

Mozilla Firefox

Obe funkcionality rozšírenia sú implementované v jednom rozšírení

Google Chrome

Rozšírenie je rozdelené podľa funkcionality na dve časti

- administrátorská - zabezpečuje definovanie oblastí záujmu
- klientská - umožňuje obohacovanie dát zo sledovania

6.2 Používateľská príručka pre Carrots Client

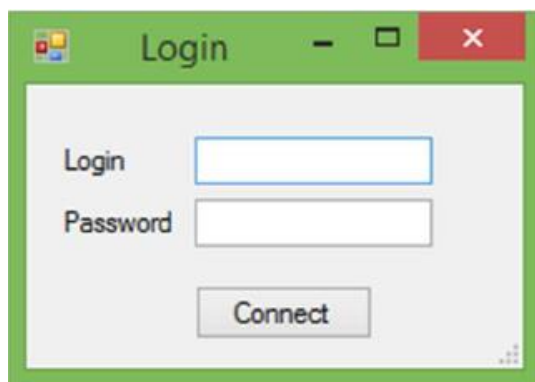
Táto časť opisuje každú obrazovku aplikácie a aké akcie sa v nej dajú vykonať.

6.2.1 Spustenie aplikácie

Aplikáciu je možné jednoducho spustiť pomocou odkazu na pracovnej ploche. Po spustení aplikácie sa zobrazí obrazovka pre prihlásenie.

6.2.2 Obrazovka pre prihlásenie

Do obrazovky treba zadať používateľské meno a heslo a následne kliknúť na tlačidlo connect. Prihlásiť sa je možné pomocou AIS konta, no používateľ sa musí nachádzať v našej databáze. Ak potrebujete vytvoriť konto, môžete nám poslať email na adresu tp-1415-16@centrum.sk



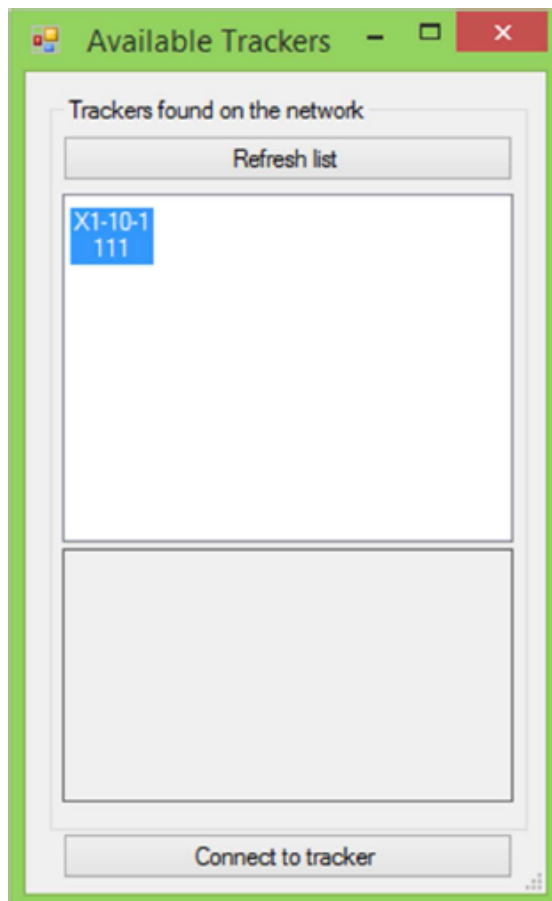
Obrázok 16: Obrazovka pre prihlásenia.

Po úspešnom prihlásení môžu nastať dva scenáre:

- ak má používateľ pripojené práve jedno zariadenie pre sledovanie pohľadu, automaticky sa aplikácia pripojí na toto zariadenie a spustí sa kalibrácia
- ak má používateľ pripojených viac, alebo žiadne zariadenie pre sledovanie pohľadu, zobrazí sa mu obrazovka pre výber zariadenia, na ktoré sa chce pripojiť

6.2.3 Výber zariadenia pre sledovanie pohľadu

Zariadenia môžeme zobrazit' kliknutím na tlačidlo Refresh list. Následne kliknutím môžeme vybrať zariadenie a pripojiť sa naň kliknutím na tlačidlo Connect to tracker.

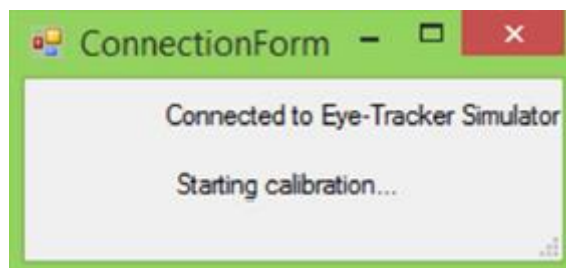


Obrázok 17: Obrazovka pre výber zariadenia pre sledovanie pohľadu.

Po pripojení na zariadenie sa automaticky spustí kalibrácia zariadenia.

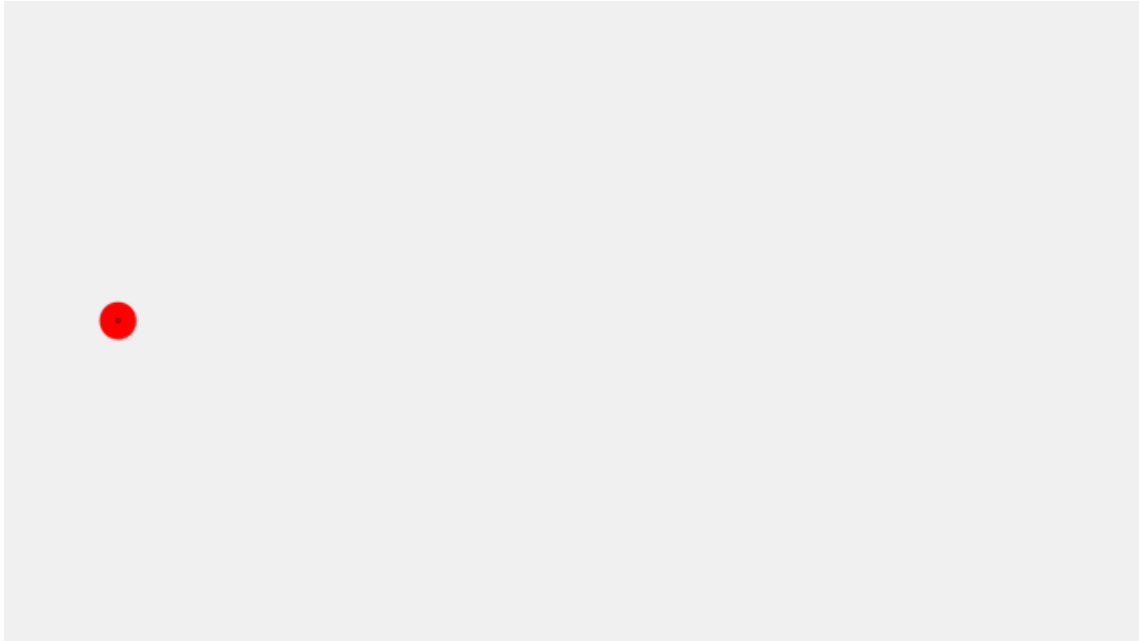
6.2.4 Kalibrácia zariadenia

Po pripojení na zariadenie sa zobrazí oznámenie o spustení kalibrácie.



Obrázok 18: Oznámenie o spustení kalibrácie.

Následne sa zapne samotná kalibrácia. Pri kalibrácii musí používateľ sledovať stred červenej guľičky, kým sa nenakalibruje 9 bodov kalibrácie.

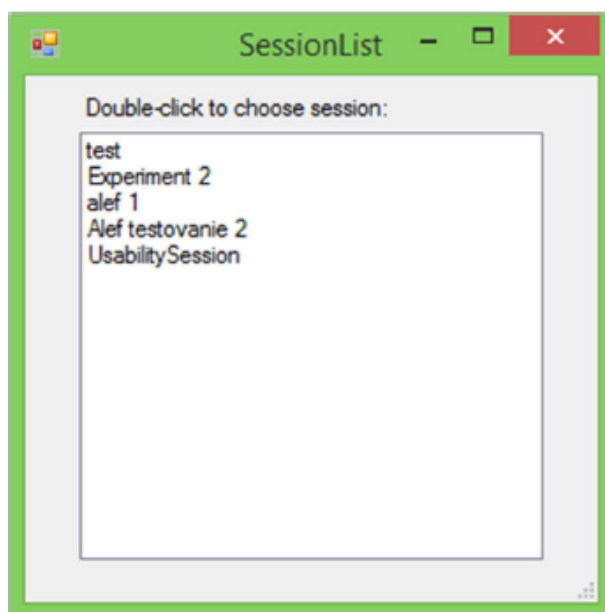


Obrázok 19: Priebeh kalibrácie zariadenia.

Po kalibrácii zariadenia sa zobrazí výsledok kalibrácie. Výsledok kalibrácie vyzerá rôzne pre rôzne zariadenia. Vždy ale obsahuje dve tlačidlá - jedno pre potvrdenie kalibrácie a jedno pre opätovné nakalibrovanie. V prípade, že kalibrácia nebola dobrá, môžeme zariadenie nakalibrovať znovu. Po ukončení kalibrácie sa automaticky zobrazí obrazovka pre výber sedenia (session).

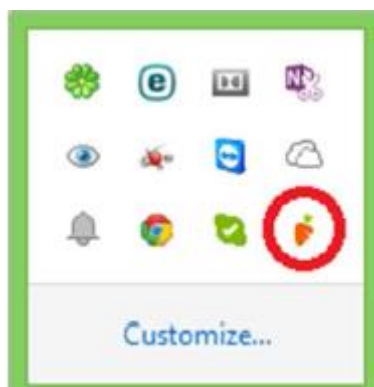
6.2.5 Obrazovka pre výber sedenia

V tejto obrazovke sú jednoducho vypísané aktívne sedenia, na ktoré je používateľ prihlásený. Pre pokračovanie si používateľ musí dvojklikom musí vybrať sedenie, na ktoré sa chce aktuálne prihlásiť.



Obrázok 20: Výber sedenia.

Po výbere session sa aplikácia automaticky minimalizuje a začne sledovať používateľov pohľad. Aplikáciu je možné zobrazit' pomocou dvojkliku na ikonu mrkvičky v system tray.

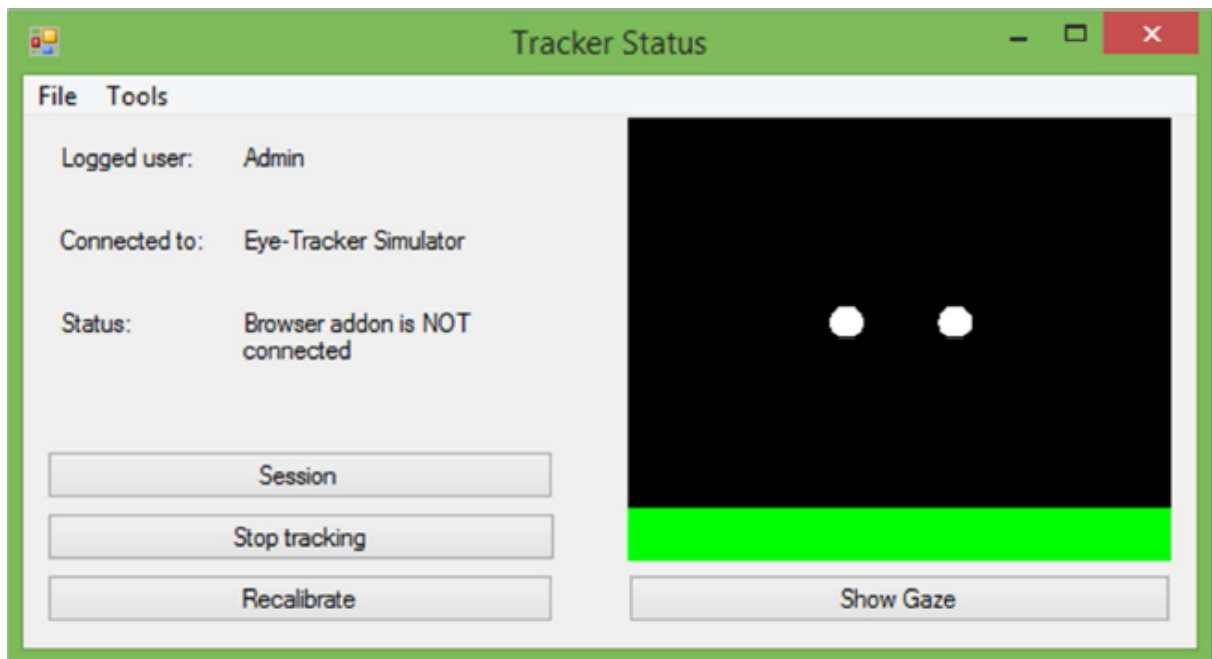


Obrázok 21: Ikona aplikácie v system tray.

Po dvojkliku na ikonu sa zobrazí obrazovka so stavom aplikácie

6.2.6 Obrazovka so stavom aplikácie

Obrazovka so stavom aplikácie predstavuje hlavnú obrazovku aplikácie a obsahuje všetky dôležité informácie o stave aplikácie.



Obrázok 22: Obrazovka stavu aplikácie.

Obrazovka so stavom aplikácie obsahuje v ľavom hornom rohu rôzne informácie:

- Logged user: prihlásený používateľ - meno zadávané v prvej obrazovke
- Connected to: meno trackera, na ktorý sa používateľ pripojil
- Status: zobrazuje, či je pripojené rozšírenie pre prehliadač

V ľavom dolnom rohu sa nachádza niekoľko tlačidiel:

- Session - tlačidlo pre zmenu sedenia, ak je to nutné
- Start/Stop tracking - tlačidlo pre spustenie/zastavenie zachytávania dát z pohľadu používateľa
- Recalibrate - tlačidlo, pre opätovnú kalibráciu zariadenia, ak je potrebná

V pravej časti sa ďalej nachádza grafické znázornenie umiestnenia používateľových očí v priestore, zobrazené ako biele body v čiernom poli. Pod čiernym poľom sa nachádza farebný pásik, ktorý svojou farbou znázorňuje stav zachytávania pohľadu:

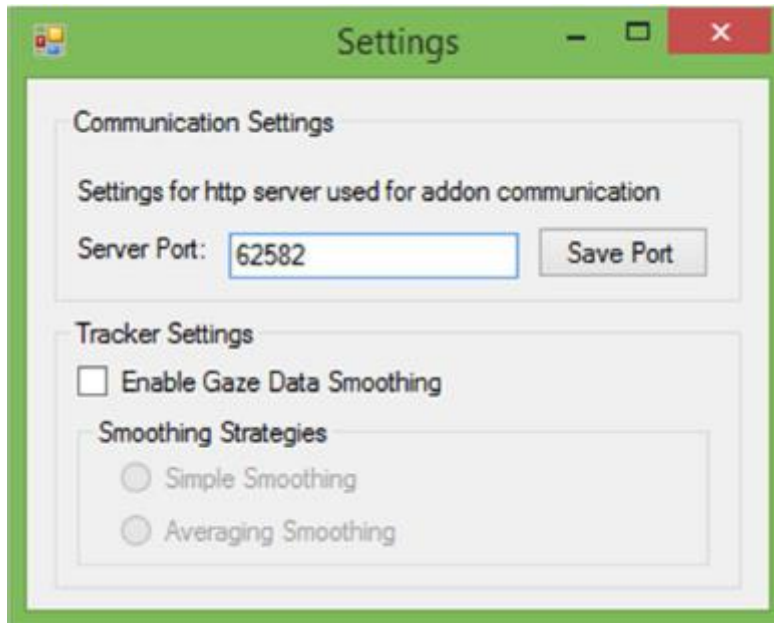
- šedá farba - zachytávanie pohľadu je vypnuté
- červená farba - zariadenie má problém so zachytením očí
- zelená farba - zariadenie v poriadku sleduje oči

Pod touto časťou sa nachádza tlačidlo Show Gaze, ktoré zobrazí formulár podobný kalibrácii. V tomto prípade ale nemá používateľ sledovať červenú guľičku, naopak sa červená guľička nachádza tam, kam sa používateľ práve pozerá. Zavrieť sa táto obrazovka dá buď pozretím sa do pravého dolného rohu, alebo stlačením klávesovej skratky ALT+F4.

Okrem týchto častí obsahuje na vrchu okna stavu aplikácie navigačné menu. V položke File je zatiaľ možné iba vypnúť aplikáciu. V položke Tools sa dá kliknutím na Settings prístup k nastaveniam klienta, popísaným v nasledovnej časti.

6.2.7 Nastavenia klienta

Nastavenia klienta momentálne obsahujú dve položky: nastavenie HTTP servera, ktorý sa používa na komunikáciu s rozšíreniami pre prehliadač a nastavenie vyhladzovacej stratégie pre pripojené zariadenie pre sledovanie pohľadu.



Obrázok 23: Nastavenie HTTP servera.

Pri serveri je možné v prípade potreby zmeniť port. Port je možné zmeniť prepísaním hodnoty Server Port na požadovaný port a stlačením tlačidla Save. Pre povolenie vyhladzovania dát je potrebné zaškrtnúť Enable Gaze Data Smoothing a následne vybrať stratégiu vyhladzovania z dvoch možných:

- **Simple Smoothing** - berie sa každých 5 gaze data, ktoré prišli zo zariadenia a spraví sa z nich priemer. Tým pádom nám prakticky 5-násobne klesne framerate zariadenia.
- **Averaging Smoothing** - podobne ako pri predchádzajúcej stratégii sa počíta priemer z piatich gaze data, ale je to vždy 5 posledných. Framerate teda zostáva rovnaký. Je odporúčané využívať túto stratégiu.

6.3 Používateľská príručka pre Simulátor

Táto časť uvádza použitie aplikácie s popisom jej jednotlivých prvkov.

6.3.1 EyeTracker Simulator Information

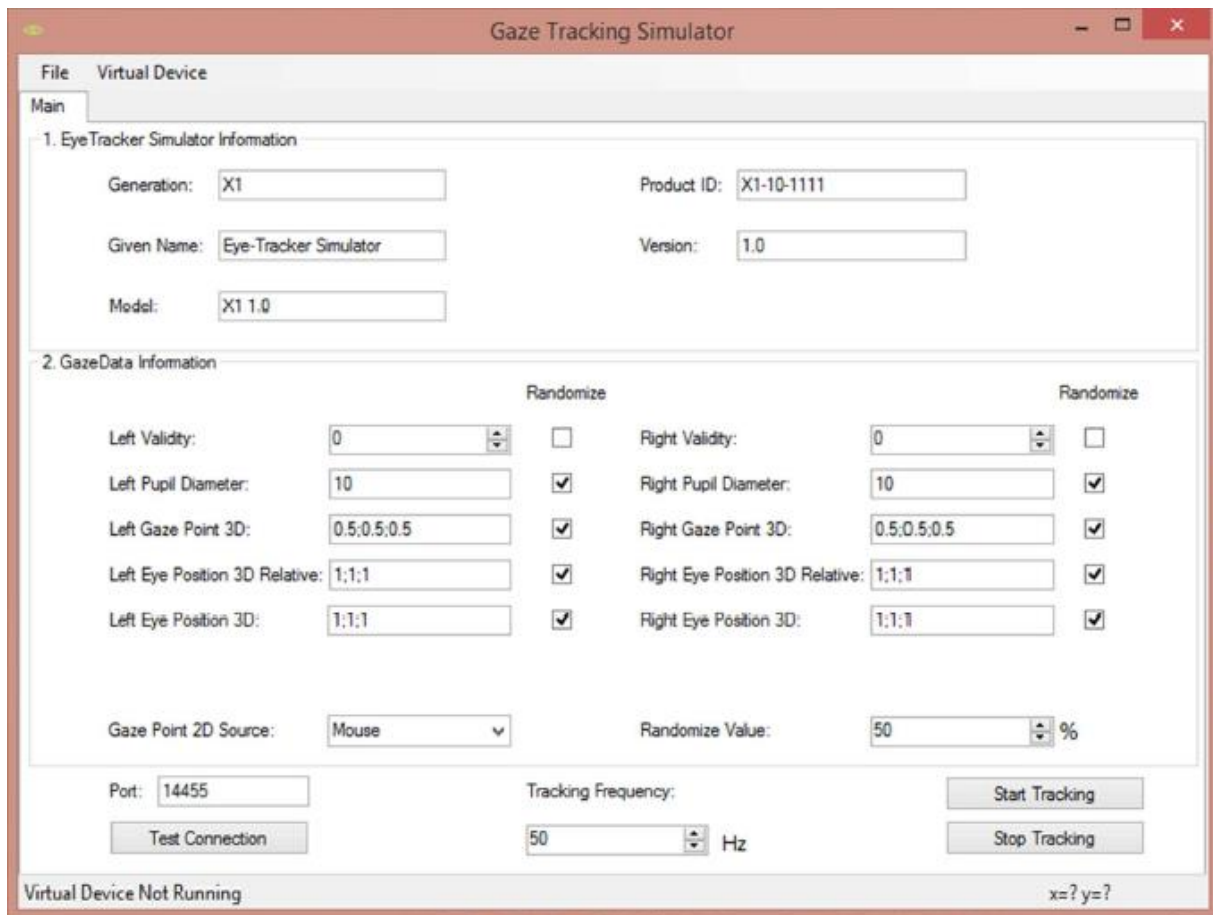
Táto časť nie je povinná. Obsahuje informatívne údaje o simulátore aby ho bolo možné nájsť jednoduchšie pomocou browsera poskytovaného firmou Tobii. Tieto údaje zodpovedajú údajom reálnych zariadení od firmy Tobii.

6.3.2 GazeData Information

V tejto časti sa nachádzajú informácie o pohľade pre obe oči, ktoré je možné simulovať. Validity môže nadobúdať hodnoty 0 až 4, pričom 0 predstavuje zaznamenané oko a hodnota 4, nezaznamenané oko. GazePoint 3D určuje polohu očí v priestore v rámci imaginárnej kocky projektovanej zariadením. Najdôležitejší údaj je Gaze Point 2D Source, ktorým vieme nastaviť vstup dát, ktoré určujú kam na obrazovku sa používateľ "pozeral".

6.3.3 Nastavenia spustenia sledovania

Je možné nastaviť číslo portu, na ktorom má prebiehať komunikácia simulátora s ďalšími aplikáciami. Simulátor po spustení automaticky spúšťa komunikáciu. Broadcastom odosiela informácie z prvej časti simulátora na danom porte aby sa mohli iné aplikácie pripojiť k simulátoru. Tlačidlom "Test connection" môžeme testovať, či je simulátor prepojený s inou aplikáciou a tým pádom, či môžeme spustiť simuláciu sledovania pohľadu. Tlačidlami "Start Tracking" a "Stop Tracking" spúšťame a zastavujeme simuláciu sledovania pohľadu. Na najspodnejšej časti aplikácie sa nachádza riadok o stave simulátora spolu so súradnicami simulujúcimi pohľad na obrazovku.



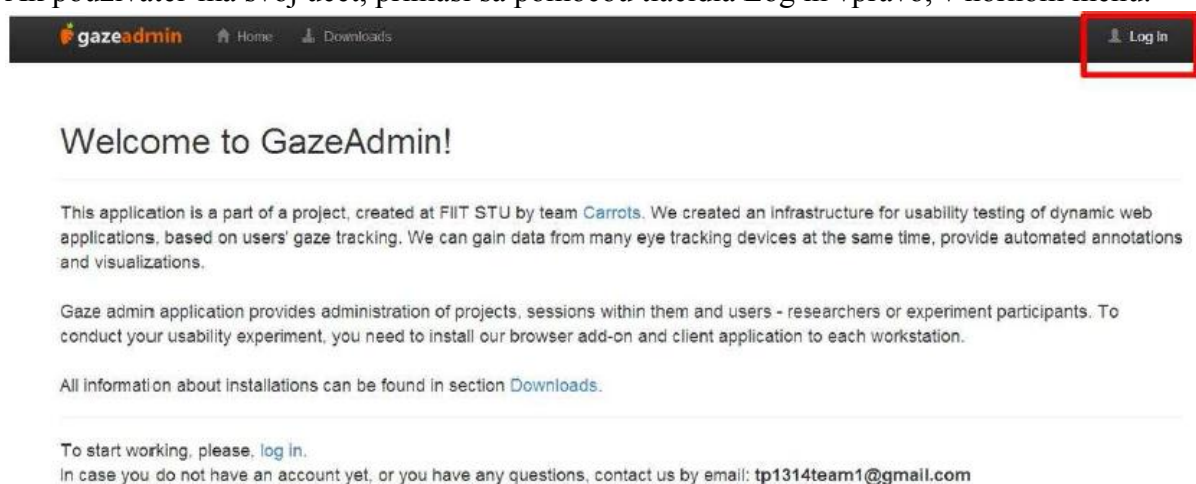
Obrázok 24: Grafické rozhranie Simulátora.

6.4 Používateľská príručka pre aplikáciu Gaze Admin

Aplikácia GazeAdmin je dostupná na adrese: <http://team01-13.ucebne.fiit.stuba.sk/web>. Úvodná stránka obsahuje základné pokyny pre začiatok práce s aplikáciou. Je nutné najprv sa prihlásiť. Ak používateľ doposiaľ nemá konto, je na stránke uvedený kontakt, kam sa môže obrátiť so žiadosťou o jeho vytvorenie. Pre neprihláseného používateľa je dostupná okrem úvodnej aj stránka Downloads, kde sú dostupné na stiahnutie verzie inštalovateľných komponentov našej infraštruktúry:

- rozšírenia pre prehliadače Google Chrome a Mozillu Firefox
- desktopový klient
- simulátor

Ak používateľ má svoj účet, prihlási sa pomocou tlačidla Log in vpravo, v hornom menu.



Obrázok 25: Úvodná stránka s vyznačeným tlačidlom na prihlásenie používateľa.

Po prihlásení sa používateľovi sprístupnia ďalšie časti aplikácie prostredníctvom položiek menu. Základné sú administrácia experimentov, administrácia používateľov a zobrazenie štatistík. Všetky tri skupiny popisujeme v nasledujúcich častiach.

6.4.1 Administrácia experimentov

Na uskutočnenie experimentu je potrebné:

















- vytvoriť projekt a pridať k nemu používateľov
- vytvoriť rámci projektu sedenie/sedenia (sedenie = 1 UX experiment)
- pridať k sedeniu participantov
- určiť oblasti záujmu sledovanej aplikácie

Vytvorenie projektu

Na prístup k projektom slúži v hornom menu položka *Projects*. Používateľ uvidí zoznam všetkých projektov, ktoré vytvoril alebo bol na ne pridelený. V tabuľke v slpci *Actions* má možnosť upraviť alebo vymazať projekt a pomocou tlačidla *Add project* môže pridať nový projekt.

Projects

My projects ▾ + Add project

Name	Description	Created at	Owner	Is active	Actions
Test2	what?	5.2.2014 14:25:08	Jakub Daraz	Yes	 
projekt moj	Lorem ipsum dolor sit amet, co...	6.2.2014 1:18:40	Admin Admin	Yes	 
test	test	18.2.2014 12:16:06	Admin Admin	Yes	 
blabla	bla	21.3.2014 23:15:30	Admin Admin	No	 
UsabilityTestovanie01		20.4.2014 21:06:23	Admin Admin	No	 
project_1		20.4.2014 23:02:12	Admin Admin	No	 
Testovanie UX	pokus o vzorovy projekt	28.4.2014 9:21:00	Admin Admin	Yes	 
ahoj		15.5.2014 0:16:08	Admin Admin	No	 

Obrázok 26: Zoznam projektov, ktorý sa zobrazí po kliku na položku Project horného menu.

Projects - New

Name Is active

Description

<< Back to List Create

Obrázok 27: Vytváranie projektu.

Po kliknutí na tlačidlo *Create* sa vytvorí nový projekt, ktorý sa rovno otvorí, pripravený na editovanie.

The record have been successfully added

Edit Project

Details Sessions Users

[← Back to Projects](#) [Save details](#)

Name: Is active Owner: Admin Admin

Description: Created at: 20.5.2014 11:01:39

Obrázok 28: Editovanie projektu.

Stránka *Edit Project* obsahuje tri záložky

- Details - úprava základných informácií o projekte, uloženie zeleným tlačidlom Save details
- Sessions - možnosť prezerat' vytvorené a pridavat' nové sedenia (experimenty). Postup je analogický ako pri projekte
- Users - možnosť pridať k projektu používateľov s rôznou rolou

Pridávanie používateľov k projektu

V záložke Users používateľ vidí ľudí, pridaných na projekt/sedenie. Pridať ďalších môže pomocou tlačidla *Add users*. Pridať k projektu možno len tých používateľov, ktorí sú registrovaní v našej databáze.

Postup: zadať meno > potvrdiť tlačidlom *Search* > zobrazí sa zoznam nájdených používateľov, z ktorých je možné vybrať a pridať ich k projektu alebo sedeniu s vybranou rolou. Existujú dva druhy rolí používateľa na projekte:

- *Project admin* - môže prezerat' a upravovat' daný projekt a sedenie v rámci neho
- *Researcher* - môže si prezerat' dáta o danom projekte, nemôže upravovat'

Projects - Add User

Find user name: [Search](#)

User Name	First Name	Last Name	Email	User Role	
xmeszarosm1	Michal	Meszaros	null	ProjectAdmin	Add
xdarazj	Jakub	Daraz	bla@bla.sk	ProjectAdmin	Add
xgregorovic	Lukáš	Gregorovič	gri@chello.sk	ProjectAdmin	Add

[<< Back to project](#)

Obrázok 29: Pridávanie používateľa k projektu.

Vytvorenie sedenia v rámci projektu

Pridávanie session (sedenia) funguje podobne, ako vytváranie projektov. Po vytvorení sa podobne ako pri projekte otvorí stránka Edit, kde možno upraviť details, pridať používateľov - participantov a definovať oblasti záujmu vo svojej aplikácii.

Pridávanie participantov k sedeniu

K sedeniu sa pridávajú len používatelia-participanti, nie je možné nastaviť im rolu. Pridať používateľov k sedeniu v časti Edit session dvoma spôsobmi:

1. *po jednom* - pomocou tlačidla *Add users*, podobne ako pri projektoch
2. *po skupinách* - pomocou tlačidla *Add group of users* je možné pridať vopred definovanú skupinu používateľov. Pridanie používateľov pomocou skupiny je výhodné najmä v prípade nutnosti pridania viacerých používateľov naraz. Skupina musí byť definovaná vopred (definovanie skupiny je popísané v časti Administrácia používateľov nižšie). Po kliknutí na *Add group of users* sa zobrazí zoznam vytvorených skupín. Jednoduchým klikom na tlačidlo *Add* sa pridajú všetci používatelia príslušnej skupiny k sedeniu.

AddGroupToSession

Name	Description	Number of users	Id
skupina pokusna	Lorem ipsum dolor sit amet, consectetur adipiscing elit. Suspendisse sodales ultrices arcu, quis laoreet sapien blandit non. Sed mattis tempus enim at accumsan. Pellentesque ullamcorper posuere leo. Id posuere justo aliquet in. Sed vel ipsum nisi. Integer euismod nibh in dignissim porta. Mauris vitae euismod massa, non pulvinar lacus. Fusce bibendum euismod urna ac placerat. Nunc imperdiet iorem diam, quis sollicitudin est tristique in. Curabitur convallis iaculis consectetur. Nulla tempor turpis pellentesque est vehicula semper. Aenean tempor leo non tincidunt interdum.	5	Add
test	test	0	Add

[Back to session details](#)

Obrázok 30: Pridávanie skupiny používateľov k sedeniu.

Definovanie oblastí záujmu (areas of interest) k sedeniu

Definovanie a zobrazovanie oblastí záujmu k sedeniu prebieha v časti Edit session v záložke Areas of interest. Na definovanie oblastí je potrebné mať nainštalovaný Addon pre ľubovoľný prehliadač. Samotné definovanie potom prebieha pomocou neho (podrobnejší popis - v príručke pre Addon).

6.4.2 Administrácia používateľov

Gaze Admin aplikácia umožňuje vytvárať používateľské účty a skupiny či importovať používateľov z CSV. Administrácia používateľov sa nachádza v hornom menu v položke Administration.

Vytváranie používateľských účtov

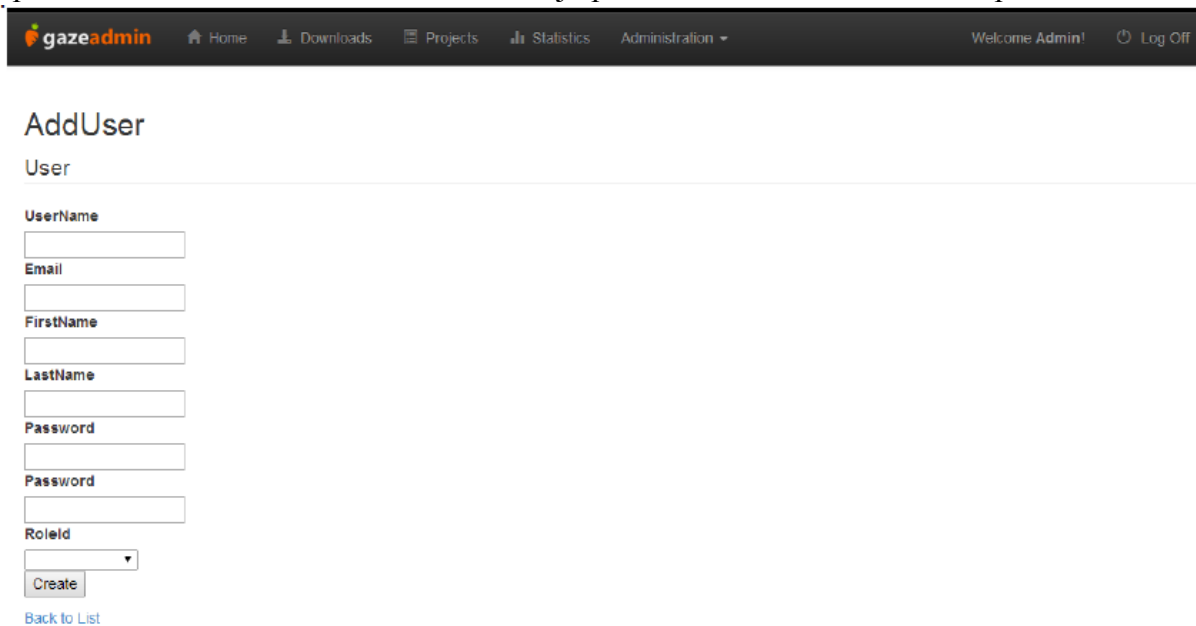
Používateľský účet je možné vytvoriť v časti Administration > Users kde je zobrazený zoznam používateľov a nad nim sa nachádza odkaz pre vytvorenie používateľov. Všetky

položky na obrazovke pre vytváranie používateľov je treba vyplniť, vybrať rolu a následne odoslať.

Používateľ môže mať nasledujúce roly:

- User
- Project admin
- Researcher
- Admin

Používateľ s právami User nemá v gaze admine žiadne práva a môže byť priradený len k sedeniu. Project admin vidí projekty a sedenia. Admin má prístup ku všetkým častiam aplikácie. Researcher môže prezerať projekty, ku ktorým je priradený ale nemôže ich upravovať. Akcie Delete a Edit sa nachádzajú pri každom zázname v tabuľke používateľov.

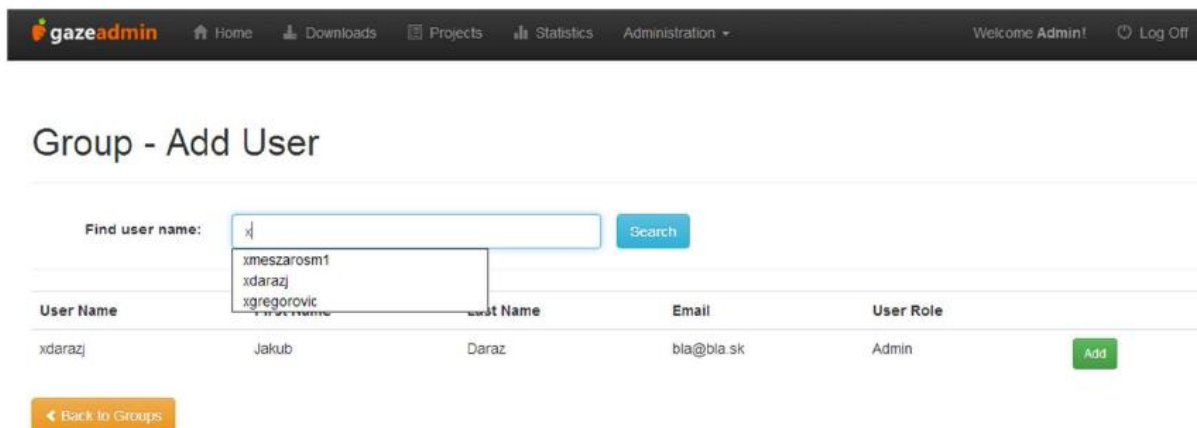


The screenshot shows the 'AddUser' page in the GazeAdmin application. At the top, there is a navigation bar with the GazeAdmin logo and menu items: Home, Downloads, Projects, Statistics, and Administration. On the right side of the navigation bar, it says 'Welcome Admin!' and 'Log Off'. Below the navigation bar, the page title is 'AddUser'. Underneath, there is a sub-header 'User'. The form contains the following fields: 'UserName' (text input), 'Email' (text input), 'FirstName' (text input), 'LastName' (text input), 'Password' (text input), another 'Password' (text input), and 'RoleId' (a dropdown menu). At the bottom of the form, there is a 'Create' button and a 'Back to List' link.

Obrázok 31: Pridávanie používateľov do aplikácie.

Vytváranie skupín používateľov

Pre vytvorenie skupiny používateľov je potrebné ísť do časti Application > User Groups, kde sa nachádza zoznam skupín. Nad zoznamom je odkaz, ktorý zobrazuje formulár pre vytvorenie skupiny. Pre vytvorenie skupiny je potrebné zadať hodnoty len do názvu. Description netreba vyplňať. Po vytvorení skupiny je možné v tabke nad formulárom pri editovaní skupiny, pridať do skupiny viacero používateľov. Po kliku na tlačidlo + Add users sa zobrazí prvok Autocomplete, kde stačí zadať časť alebo celé používateľské meno, ak zadáte časť tak program automaticky odporučí zvyšok mien, ktoré sa podobajú.



Obrázok 32: Rozhranie pridávania používateľov.

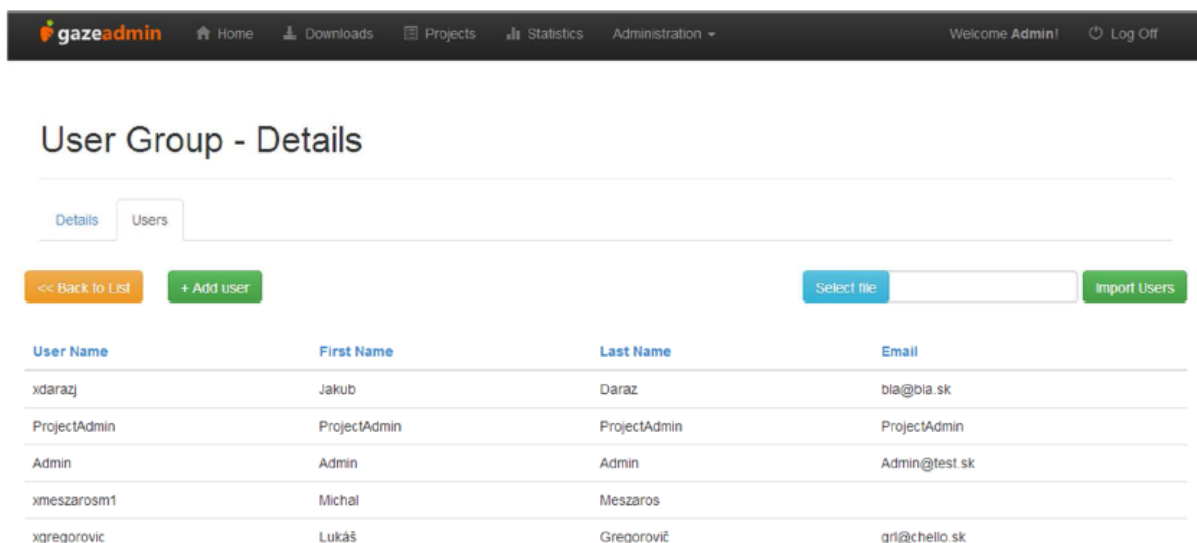
Po vybratí odporúčaného používateľa stačí kliknúť na search a v tabuľke sa zobrazí tento používateľ. Po kliku na tlačidlo Add, ktoré sa nachádza v tabuľke bude používateľ pridaný.

Vytváranie aplikácií a priradovanie API kľúčov

Vytváranie aplikačných účtov sa nachádza v časti Administration > Application user kde je zoznam vytvorených aplikácií. Princíp je rovnaký ako pri vytváraní používateľov, pričom Gaze Admin aplikácia po vyplnení všetkých položiek formulára priradí aplikácii unikátny API kľúč, ktorý je možné si pozrieť v zozname aplikácii(Application user)

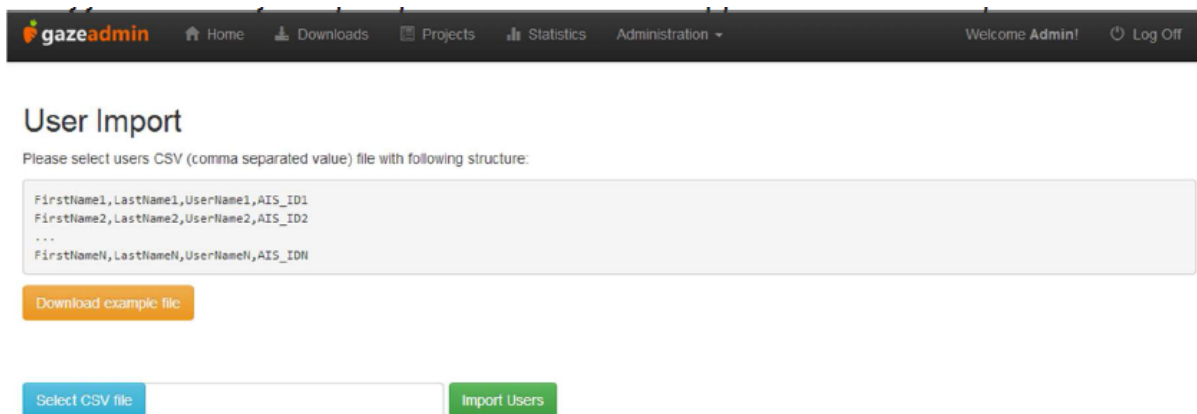
Import používateľov

Import používateľov je možné nájsť na dvoch miestach. V časti Application > User Group kde sa nachádza zoznam skupín pri každom zázname skupiny je možné kliknúť na ikonu detail. Po kliku na túto ikonu sa zobrazí zoznam priradených používateľov a nad zoznamom je možné importovať používateľov, ktorý sa rovno priradia ku skupine. Importovaný súbor musí byť vo formáte CSV.



Obrázok 33: Zoznam používateľov priradených k skupine.

Ďalej je možné nájsť import používateľov v časti Application > User import



Obrázok 34: Rozhranie pre import používateľov.

Rozhranie pre import používateľov, ktoré je uvedené na obrázku 9 znázorňuje v prvej časti príklad CSV súborov, ktorý treba nahráť a je možné si stiahnuť príklad CSV. Po nahratí sa zobrazí prehľad nahratých používateľov a tí ktorí sa nenahráli budú v tomto zozname zvýraznení.

6.4.3 Rozhranie štatistík

Gaze Admin umožňuje aj prezeranie dát z ukončených sedení pomocou rozhrania štatistík (položka *Statistics* v hornom menu). Na zobrazenie štatistík je potrebné: vyplniť filter > potvrdiť tlačidlom *Submit*. Cez filter je možné vybrať:

- projekt
- sedenia
- areas of interest
- konkrétnych participantov

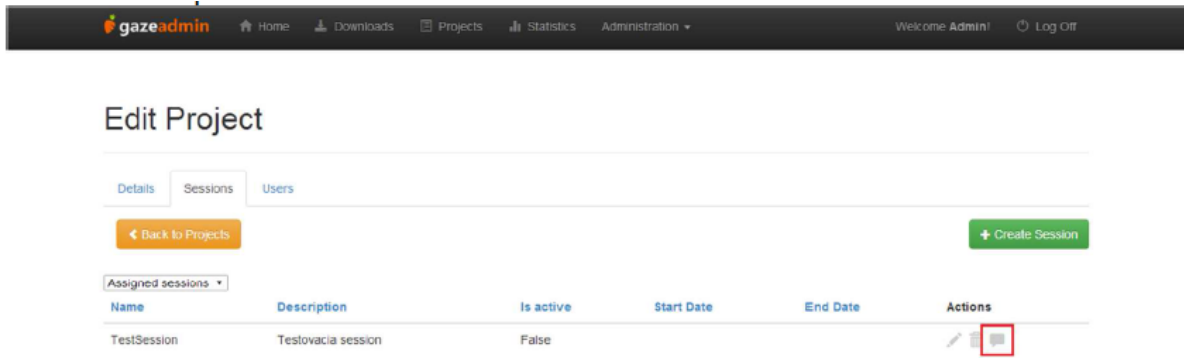
Po odoslaní formulára sa zobrazia vyfiltrované údaje zobrazia v tabuľke. Prepnutím záložky na *Chart – fixation count* alebo na *Chart – fixation duration* je možné vidieť grafové zobrazenie. V grafových zobrazeniach je možné prepínať medzi grafmi pre jednotlivé sedenia prepínaním výberu sedenia.



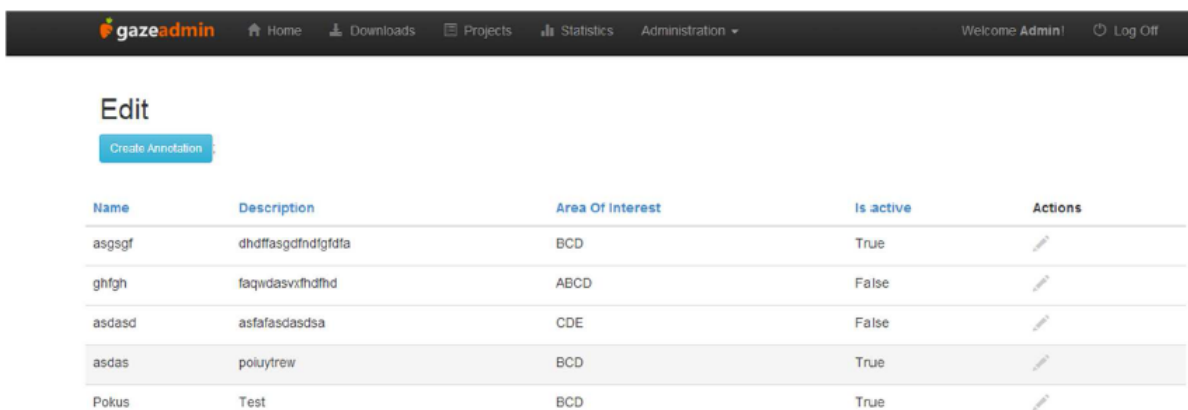
Obrázok 35: Rozhranie štatistík.

6.4.4 Rozhranie pre automatické anotácie

Gaze Admin podporuje aj vytváranie automatických anotácií definovanými Projektovým administrátorom. Aby prístup k definovaniu anotácie bol čo najjednoduchší k rozhraniu na spravovaní anotácií sa dostaneme po vybratí príslušného projektu a následne v záložke Sessions vyberieme možnosť Annotation v stĺpci Actions. V prípade, že používateľ nemá ProjectAdmin práva ale len Researcher práva dostane sa po kliknutí na túto ikonu len do rozhrania kde sú anotácie zobrazené a nemá právo ich editovať alebo pridávať nové.

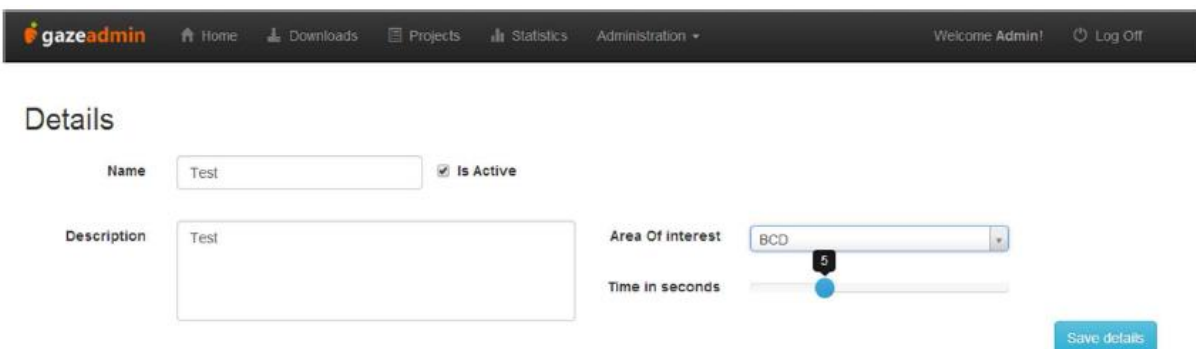


Obrázok 36: Prístup k rozhraniu pre anotácie.



Obrázok 37: Rozhranie pre editáciu anotácií respektíve ich pridávanie.

Nakoľko anotácie sú viazané na oblasti záujmu pri jej vytváraní je potrebné aby používateľ vybral oblasť ku ktorej sa majú anotácie vytvárať. Taktiež je potrebné určiť čas, po ktorom sa vytvorí anotácia. Defaultná hodnota je určená 1 sekunda pričom granularita pre čas je 1 sekunda.



Obrázok 38: Rozhranie pre vytváranie anotácie.

Pre uľahčenie párovania oblasti záujmu na anotáciu je v pri v vyberaní oblasti možné vyhľadávať ich pomocou fultextového vyhľadávania.

Details

Name Is Active

Description

Area Of interest
Time in seconds

Obrázok 39: Možnosť fulltextového vyhľadávania pri vyberaní oblasti záujmu.