

Проект:

Public-Localization-Services

Съдържание

1. АВТОРИ

2. ТЕМА

3. РЪКОВОДИТЕЛ

4. РЕЗЮМЕ:

4.1. ЦЕЛИ

4.2. ОСНОВНИ ЕТАПИ В РЕАЛИЗИРАНЕТО НА ПРОЕКТА

4.3. НИВО НА СЛОЖНОСТ

4.4. ЛОГИЧЕСКО И ФУНКЦИОНАЛНО ОПИСАНИЕ НА РЕШЕНИЕТО

4.5. РЕАЛИЗАЦИЯ

4.6. ОПИСАНИЕ НА ПРИЛОЖЕНИЕТО

4.7. ЗАКЛЮЧЕНИЕ

1. Автори:

Илия Владиславов Вълков

ЕГН: 0051216285

Адрес : ж.к. „Илинден“, бл. 2, ет. 3, ап.13

Телефон: 0888014990

Имейл: iliavulov@gmail.com

Училище: 57-СУ „Св. Наум Охридски“,
гр.София, 12^в клас

Радостин Бориславов Борисов

ЕГН: 0149173289

Адрес: ул. „Столетов“, №15

Телефон: 0897715719

Имейл: noob_120@abv.bg

Училище: ПМПП „Св. Климент
Охридски“, гр.Монтана, 11^в клас

2. Ръководител:

Илиян Ценков

Телефон: 0885927955

Имейл: i_cenkov@yahoo.com

Длъжност: Учител по информатика

- 3. Тема на проекта:** Проектът трябва да улеснява работата на планинските спасителни служби, като следи дали има бедствия и показва тяхната локация.

4.Резюме:

4.1. Цели - Идеята на проекта е да подпомогне българските планински спасителни служби на български червен кръст. Мобилно приложение трябва да взима координатите от дадено мобилно устройство. Информацията минава през сървър за филтрация на даните ако случайно попаднат невалидни дани. След филтрацията тези данни се съхраняват на база данни като те никога не се трият, за да не се получат грешки от към страна на индексатора. Като даните са запазени координатите на дадения потребител биват викнати от десктопно приложение, което трябва да ги изобрази на карта. Това се прави за улеснение на планските спасителни служби да намират по-бързо бедстващи или изчезнали. Например ако някой бедстващ планинар има нужда от помощ при натискане на бутон от мобилното приложение трябва да бъде маркиран на картата, като бедстващ и да се организира екип за спасяването му.

4.2. Основни етапи в реализирането на проекта – Започнахме с идеята за приложение с карта, андроид приложение и един сървър. Започнахме работа върху сървъра като идеята беше сървър да извършва цялата работа с данните получени от андроид приложението. В процеса на разработка на сървъра ни дойде идеята да създадем още един сървър и една база данни, за да улесним и разпределим работата, и да няма толокова трафик върху един сървър. Единия сървър приема информацията и я записва в базата данни, а другия взема информацията и я праща до десктопното приложение. След като изготвихме сървърите започнахме работа върху андроид приложението. Неговата цел е да се изпрати местоположението и някаква информация за идентификация. Спряхме се на телефонен номер, защото е уникален за всеки телефон. След завършване на андроид приложението започнахме десктопното приложение. Испитахме затруднения в изкарването на картата на

самото приложение, но успяхме да се справим с помощта на GMap библиотеката. Успяхме да изкараме зададена локация на картата и започнахме да създаваме връзките между сървърите и отделните приложения. В момента това е проблема който ни се опира най-много, но е в процес на разработка.

4.3. Ниво на сложност на проекта - Основните затруднения настъпиха, когато се стигна до създаване на връзките между сървърите и приложенията. Изискваше се сертификат за защита на данните и след прилагане на такъв се изисква домейн. Предполагаемото решение е да се защити потока от данни чрез блокчейн криптиране. В процес е на разработка и скоро ще бъде реализирано. С още затруднения се сблъскахме при работата с JAVA и Android Studio. Работихме с напълно непозната за нас среда, със времето решихме този проблем. Сложността на проекта за нас е много голяма, защото все още сме начинаещи програмисти.

4.4. Логическо и функционално описание на решението –

1. Базата дани – Създаваме базата дани изцяло автоматично, чрез използването на code first метода и entity framework, който представлява да опишеш чрез класове референциите на таблиците в базата. Всяко property в класът трябва да отговоря на колоните в таблицата. Code first метода е разделен в 3 библиотеки
 - 1.1. Model библиотеката – Тя съдържа класовете, които ще бъдат мапнати към таблиците в дадената база дани.
 - 1.2. Data библиотеката – В нея се намира dbContext-a, който по-късно ще се използва, за да се достъпи базата дани.
 - 1.3. Initilizer библиотеката – Чрез нея, ние запълваме таблиците с тествани дани, които знаем, че са валидни или се опитваме да вкараме грешни и да проверим да ли се прихващат. Това се случва чрез генератори класове, като трябва да има генератор за всяка таблица от базата данни.
2. ASP.NET Api – То представлява контролер на дани. Ние използваме два такива, за да разпределим работата и да има по-малко натоварване на интернет трафика. За събиране или изкарване на информация употребяваме viewmodel модела, който трябва да има properties, които да съответстват с тези на моделите от базата, за да могат да се мапнат без грешки.
 - 2.1. Единият от api-тата поема информацията, която бива изпращана от мобилните приложения и я запазва в базата дани, като трябва да проверява за грешни данни и да следи дали се изпраща от мобилно устройство. Ако не е от мобилно устройство трябва да игнорира тази информация.
 - 2.2. Другото api отговаря да изпраща информация на десктоп приложението.

3. Мобилното приложение – То е нормално gps приложение, само че то трябва да следи локацията на хората и да я изпраща до сървър, който да я обработи. То също трябва да има “help” бутон, който при натискането му да предупреди следящите на десктоп приложението, че има някой в беда.
4. Десктоп приложението – То приема информация от Asp.Net api-то. Чрез newtonsoft десириализираме json-на в сложен обект, който трябва да съдържа данните за user-а и неговите локации. След като вече имаме всички обекти запазени в лист ги изобразяваме на картата. За картата използваме Gmap библиотеката.

4.5.Описание на приложението - Мобилното приложение представлява нормален GPS софтуер. Избрахме да бъде GPS софтуер, за да примами повече планинари да го използват. Докато потребителите го използват техните локации ще биват изпращани на сървър, който ще ги филтрира и запази в база дани. Но това не е всичко. Желаяме също така да добавим малки тестове да проверяваме знанията на потребителите за първа помощ например и с тази информация БЧК може да разпространява обученията, ако резултатите са по-слаби или по-добри. За направата на това приложение използваме Android Studio, което разполага с много добри библиотеки на Google maps и благодарение на тези библиотеки ние можем да вземем координатите на даден потребител, също така библиотеките на Google maps ни позволяват да използваме GPS класове и методи, с което улеснява нашата разработка на приложението.

Сървърите на това приложение са два. Един приема заявки от мобилното приложение, а другият от десктопното.

-Сървър за мобилното приложение.

Той трябва да приема заявки от телефона за записване на информация.

Например регистрация. Това означава, че мобилното приложение ще прати телефонния номер на потребителя. Избрахме да използваме само телефоният номер понеже знаем, че той е уникален и се защитаваме един вид от невалидни регистрации.

Също така той трябва ще приема локациите на потребителите. Понеже знаем, че базата дани ще се претовари ако през 1 секунда сървър праща заявки към нея и вземе решение той да складира информацията и на всеки 5 секунди, да я изпраща на базата дани за запис, като по този начин намаляме трафика на заявки към базата дани.

-Сървър за десктопното приложение.

Неговата задача е на определено време (например 5 секунди) да взема цялата информация от базата дани и да я складира в себе си пак за намаляне трафика на дани към базата. Тези дани ще бъдат складирали в речник, който ще има за ключ телефония номер на потребителя и за стойност лист от локациите му. Когато десктопното приложение прати заявка сървър ще ги е приготвил предварително. Заявките ще бъдат хиширани, за да предотвратим изтичане на дани и по-трудно хакерите да разберат под, какъв формат изпращаме даните.

Сървърите са написани на ASP.NET CORE. Използваме API за него, защото има същите качества като нормално ASP.NET CORE приложение ,само че API няма view. Много прилича на конзолно приложение и най-важното, е че работи с https заявки. Използваме също така JSON формат за заявките, той е много по-разбираем и по-удобно се обработва.

Базата дани е важна част от проекта. В нея складираме цялата информация, която събираме от потребителите. Макар да филтрираме информацията през нашите сървъри ние слагаме допълнителна филтрация и в базата дани, за да преодвратим невалидни дани. Базата дани е удобна заради нейната индексация на ключовете. Тя може да намира изключително бързо информация по индекс. Малко за минусите на PostgreSQL. За разлика от SQLServer на Microsoft той не може да работи бързо с гуиди. Гуидите са много полезни, защото намалят шанса за повтаряне на индекс. Глобално шанса за създаване на 2 гуида е изключително малък. Той се използва ако имаме повече от един навързани сървъри и за да се избегнат повторения се използват гуидите. За тяхното създаване се използва вашия мак адрес и текущото време до милисекунди. Преди или след това време е невъзможно да се създаде еднакъв гуид. Но както казахме ние можеме да използваме само sql/express, който има ограничена памет и това ни затруднява. Затова се преместихме на PostgreSQL.

Десктоп приложението е най-важната част от целият проект. След всичките събрани дани и тяхната филтрация, най-накрая те идват тук за изобразяване на картата. Всеки потребител е изобразен с маркер на картата като маркерите на бедстващи и не бедстващи са различни. Също така спасителите ще бъдат отделени и ще се различават, за да може от централите да ги навигират по-лесно. Зада можем да постигнем тези операции ще използваме много колекции като речници и листове. Базата дани ще има поле, което следи дали дадения потребител е на линия и друго дали той е спасител. Зависимост от тези дани потребителите ще бъдат изобразени ако са на линия и ще се различават, ако са спасители и най-важното в базата ще има поле да ли са бедстващи. Ако са бедстващи екрана на десктоп приложението ще се насочи на там. Това е една автоматизация, която ще спести време на спасителите да търсят, къде е бедстващия.

4.7. Заключение – Основният проект все още е в разработка и има още какво да се добави, докато се превърне в краен продукт. Представяме една Beta версия на проекта, за да покажем основните функционалности. Веднъж щом проектът бъде завършен, той ще бъде предаден на Български Червен Кръст където ще намери огромно приложение във туристическите райони. Проектът ще бъде предоставен в перфектно състояние на организацията.

