Міністерство освіти і науки України

Харківський національний університет радіоелектроніки

Факультет комп’ютерних наук

Кафедра програмної інженерії

КУРСОВА РОБОТА  
пояснювальна записка

«Програмна система для керування ролетами»

Студент гр. ПІ-13-1 Рудь С.В.

Керівник роботи: доц. каф. ПІ Лещинський В.О.

Роботу захищено «\_\_»\_\_\_\_\_\_2016 р.

з оцінкою \_\_\_\_\_\_\_\_\_\_\_\_\_

Комісія:

проф. каф. ПІ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Дудар З.В.

доц. каф. ПІ \_\_\_\_\_\_\_\_\_\_\_\_\_\_Лещинський В.О.

ст. викл. каф. ПІ \_ \_ \_\_\_\_\_\_\_\_\_\_\_\_ Лещинська І.О.

Харків

2016 р.

Харківський національний університет радіоелектроніки

Факультет комп’ютерних наук

Кафедра програмної інженерії

Напрям підготовки Програмна інженерія

Курс 3 . Семестр 6 .

Навчальна дисципліна Аналіз та рефакторинг коду програмного забезпечення .

ЗАВДАННЯ  
НА КУРСОВУ РОБОТУ СТУДЕНТОВІ

*Рудю Сергію Віталійовичу\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_.*

1. Тема роботи: *«Сервіс для керування ролетами» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*

*.*

2. Термін узгодження завдання курсової роботи «*04*»\_\_*лютого*\_\_\_\_\_\_\_\_\_ 2016 р.

3. Термін здачі студентом закінченої роботи «*13*»\_\_\_*червня\_\_\_\_\_\_ \_\_* 2016р.

4. Вихідні дані до проекту (роботи): *Використовувати ОС Windows, середовище об'єктно-орієнтованого проектування Microsoft Visual Studio 2015 Community*

*.*

*.*

*.*

*.*

5. Зміст пояснювальної записки *(перелік питань, що належить розробити) вступ, аналіз предметної області, постановка задачі, моделювання програмного продукту, опис інтерфейсу та функціоналу, висновки, перелік посилань.*

*.*

*.*

6. Перелік графічного матеріалу (з точним зазначенням обов’язкових креслень)   
*діаграма класів, діаграма прецедентів, діаграма послідовностей, діаграма станів, діаграма компонентів, діаграма кооперації.*

*.*

*.*

*.*

*.*

КАЛЕНДАРНИЙ ПЛАН

|  |  |  |  |
| --- | --- | --- | --- |
| № | Назва етапів курсової роботи | Термін виконання етапів проекту (роботи) | Примітка |
| 1 | Функціональна специфікація програми | 05-02-2013 | виконано |
| 2 | Проектування програми | 20-02-2015 | виконано |
| 3 | Кодування програми | 20-03-2015 | виконано |
| 4 | Оформлення пояснювальної записки | 09-05-2015 | виконано |
| 5 | Захист курсової роботи | 13-06-2015 |  |

Дата видачі завдання «*04*»\_\_*лютого*\_\_\_\_\_\_\_\_2015 р.

Керівник роботи \_\_\_\_\_\_\_\_\_\_\_ доц. каф. ПІ Лещинський В.О.

Завдання прийняв до виконання

студент гр. ПІ-13-1 \_\_\_\_\_\_\_\_\_\_\_ Рудь С.В.

РЕФЕРАТ / ABSTRACT

Пояснювальна записка до курсової роботи: 25 с., 14 рис.,   
6 джер.

Метою роботи є проектування та розробка програмної системи для керування конструкцією захисних ролет.

Методи розробки базуються на технології .NET, MSSQL Server системі керування базами даних, середовищі розробки Visual Studio 2015 та SQL Server Management Studio 2014.

Результатом роботи є програмна система, що складається з серверу, веб-клієнта, мобільного додатку та емулятору пристрою, що відстежує зміну освітленності в приміщенні, що дозволить користувачу налаштувати автоматичний режим роботи конструкції захисних ролет.

ASP.NET, РОЛЕТИ, РОЗУМНИЙ ДІМ, МОБІЛЬНИЙ ДОДАТОК, ANDROID, ENTITY FRAMEWORK, ПРОГРАМНА СИСТЕМА.

The aim is to design and develop a software system for rollet systems managing.

The methods of development technologies are based on Java, .NET, MSSQL Server RDBMS, development environment Visual Studio 2015 and SQL Server Management Studio 2014.

The result of work is a software system that consisting server, web-client, mobile application and device emulator that tracks changing of light level in the house and allows user to configure automatic mode of rollet system working.

ASP.NET, ROLLETS, SMART HOUSE, MOBILE APPLICATION, ANDROID, ENTITY FRAMEWORK, SOFTWARE SYSTEM.

ЗМІСТ

Вступ….………………………………………………………….……….….….……......6

1 Аналіз предметної області…..……………………….…………….……...…….….…8

2 Постановка задачі….………………………………………………..….……….……13

3 Моделювання програмного продукту……..…..…………………………...….…….14

4 Реалізація програмного продукту…………….…...………………..……..…….…...18

5 Інтерфейс і функціонал……………………..………………………....….……….…20

5.1 Сервер…………………..………………….………………………..……….20

5.2 Веб-клієнт…………………………………….……...……....….…………..21

5.3 Додаток для Windows Phone……...………………………………………..23

Висновки……….……………………………...…………………………………...…....24

Перелік посилань………...…………………….………………..……...……….……...25

ВСТУП

Системи, що пов’язані з “розумним будинком”, з кожним роком набирають все більший рівень популярності і потребують для себе все більше програмних продуктів, що зможуть задовольнити вимоги користувачів.

Однією з невід'ємних частин кожної будівлі є його вікна, які, крім чистоти, потребують також і захисту. Одним з можливих розв’язків цієї проблеми є захисні ролети.

Захисні ролети є візитною карткою комфортних та сучасних будівель, побудованих із врахуванням європейських норм. Їх найчастіше встановлюють на перших поверхах багатоповерхівок, у приватних будинках, офісах, магазинах, кіосках тощо. Це пов’язано із головною функцією, яку виконують ролети – захист приміщення від небажаного проникнення. Таким чином, вони створюють комфорт, затишок та забезпечують Вашу безпеку.

Управляти такими пристроями можна в ручному режимі, за допомогою пультів або клавіш доступу. Завдячи цієї курсової роботи та розробленної в її ході системи відкривається ще одна можливість керування захисними ролетами – інтернет речей.

Інтернет речей (Internet of Things) − єдина мережа, що об’єднує техніку, якою ми користуємося щодня, та віртуальний світ. Технологія не лише дозволяє віддалено керувати різними приладами, а й пов’язує їх між собою. Обмінюючись даними, речі починають «спілкуватися» один з одним.

Ця ситуація може бути втілена в життя в тому випадку, якщо побутові прилади та інші електронні пристрої будуть підключені до Інтернету і навчаться обмінюватися між собою інформацією, виконувати команди, відправлені з інших пристроїв – компонентів цієї глобальної мережі. Реалізація цієї концепції отримала назву Інтернету речей – «Internet of Things». Вперше термін був запропонований в 1999-му році. А кількісний перехід від «Інтернету людей» до «Інтернету речей» відбувся в 2008-2009 році: саме в той період кількість пристроїв, підключених до Інтернету, перевищила кількість інтернет-користувачів.

В наш час ця технологія стає дедалі популярнішою, наприклад такі гіганти, як Apple чи Google виробляють системи для керування «розумним будинком».

Підключити інтернет зараз можна не тільки для дому. Вже зараз до мережі підключають автомобілі, які можуть у режимі онлайн отримувати мультимедійний контент, наприклад, відтворювати музику з iTunes, а також оновлювати інформацію про дорожню ситуацію і погодних умовах.

Футуристи і фахівці в технологіях називають Інтернет речей проміжним етапом на шляху до нового витка розвитку технологій – до мережі чергового покоління, названого «Інтернетом всього». У цій мережі просто не буде існувати пристроїв, що працюють в режимі «офлайн», а Інтернет буде те саме що електрика – без нього просто не зможе функціонувати жодний електронний пристрій. Сьогодні складно собі уявити, яким буде наше життя в «Інтернеті всього», проте вже зараз можна припустити, що сховатися від всевидющого ока такої воістину глобальної мережі буде практично неможливо.

Отже тема та засоби для виконання курсової роботи є актуальними. Метою курсової роботи є проектування та створення програмної системи, яка дозволить керувати захисними ролетами та відстежувати рівень освітленності приміщення для автоматичного режиму роботи з ролетами.

2 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

Як було зазначено вище обрана тема є актуальною на сьогодні і на близьке майбутнє. Щоб проаналізувати дану предметну область треба сказати про засоби, що все використовуються, їх плюси та мінуси.

Якщо казати про конкуренцію на ринку в даній сфері, то можна сміливо заявити, що наразі немає жодного аналогу, який міг би робити щось подібне. Проте конкурентами можна вважати системи, що дозволяють управляти ролетами в ручному режимі, за допомогою клавіші доступу чи дистанційного пульта керування.

Якщо розглядати плюси та мінуси «конкурентів» треба сказати, що єдиним плюсом є відсутність прив'язки до Інтернету. Якщо розглядувати мінуси, то іх у інших режимах роботи вистачає, розглянемо деякі з них.

При керуванні ролетами в автоматичному режимі широко використовується фізична сила людини і якщо ролет у домі багато, то є необхідність закривати і відкривати кожну з них використовуючи скінченні людьску силу.

Якщо керувати ролетами за допомогою клавіші доступу, то треба розмістити її в якомусь зазначеному місці в домі та спіткати необхідність кожного разу ходити до цього місця у разі необхідності змінити стан ролет. Також хазяїну дому треба заздалегідь продумати якою групою ролет буде керувати кожна з клавіш і змінити цю конфігурацію зможе тільки спеціаліст.

При керуванні ролетами за допомогою дестанційного пульту керування також існує вже описана проблема з конфігурацією ролет заздалегдіь. Крім того, пульт потребую енергії у вигляді батарейок, що не є ефективним та екологічним джерелом живлення, а сам пульт можна загубити і треба буде запрошувати професіонала для відновлення доступу до ролер та реконфігурації.

Отже, ця програмна система має функціонал, якому немає аналогів. Якщо казати про те, кому саме може знадобитись додаток, то можна впевнено сказати, що використовувати сервіс зможе всі, в кого є дім, смартфон чи ПК та доступ в інтернет.

Людина, що буде використовувати розроблений додаток зможе керувати ролетами за допомогою смартфону або додатку в Web. Для цього потрібно буде тільки встановити додаток у смартофон, увімкнути його за зв’язати його з відповідними ролетники конструкціями.

Отже, в цьому розділі було розглянуто предметну область. Можна підсумувати, що була обрана проблемна предметна область, яка потребує саме цього продукту, адже наразі немає жодного аналогу. Розглянуті приклади підтвердили, що ринок потребує цей продукт і він буде корисним дуже багатьом людям і навіть може використовуватися у великих масштабах. Враховуючи, що продукт буде розширюваний, його можна змінювати у залежності від потреб ринку та галузі використання.

3 ПОСТАНОВКА ЗАДАЧІ

Метою поданої курсової роботи є створення сервісу для керування захисними ролетами. З поставленої мети виходять мінімальні вимоги до користувачів.

По-перше для функціонування мобільного додатку необхідна наявність у користувача мобільного пристрою з операційної системою Android 4.0.x Ice Cream Sandwich або вищою, та наявність на мобільному пристрої GPS-датчика.

Дані користувачів передаються на сервер, який розміщено у хмарному середовищі, на серверах якого повинен бути встановлений та налаштований Internet Information Services Server. На сервері баз даних зберігаються дані користувачів у базі даних SQL, що також розміщується у хмарному середовищі. Дані, оброблені на серверах, передаються власне клієнтам. Мобільний додаток реалізований для подальшого розміщення у Play Market.

Веб частина повинна мати наступний функціонал: будь-який користувач може зайти в сервіс після авторизації. Якщо він ще не був зареєстрований, йому буде запропонована форма для реєстрації через соціальну мережу. Авторизований користувач має змогу переглядати свої конструкції захисних ролет та керувани ними, а також виставляти кофортний рівень освітленності. Також він може створювати та видаляти свої ролети.

Мобільна частина повинна мати наступний функціонал: авторизований користувач може переглядати свої ролети на карті та керувати ними, та також вибирати комфортний рівень освітлення.

Серверна частина повинна зберігати усіх користувачів системи та їхні ролети з налаштуваннями. Вона має надавати REST сервіси для використання веб і мобільною частинами для реєстрації, логіну, отримання маршрутів з бази.

У подальшому планується розширити соціальну та функціональну складову розробленої системи шляхом додавання наступних можливостей:

а) можливість групувати ролети та керувати групами;

б) вести статистику користувачей та девайсів додатку;

в) можливість виставити конфігурацію ролети в залежності від пори року чи часу дня.

Можлива також розробка мобільного додатку для платформ iOS та Windows Phone.

4 МОДЕЛЮВАННЯ ПРОГРАМНОГО ПРОДУКТУ

Для проектування програмної системи було використано засоби проектування Draw.IO та створено наступні діаграми UML:

а) Діаграма прецедентів (див. рис. 3.1) має одного актора – Користувача. Користу-вач – людина, яка увійшла у сервіс під своїм акаунтом, або встановила на пристрій додаток з Google Play Market. Після авторизації він має можливість працювати з ролетами або з датчиком освітлення. Він може переглядати свої ролети або змінювати їхній стан. Також є можливість налаштувати датчик освітлення для автоматичного режиму роботи.

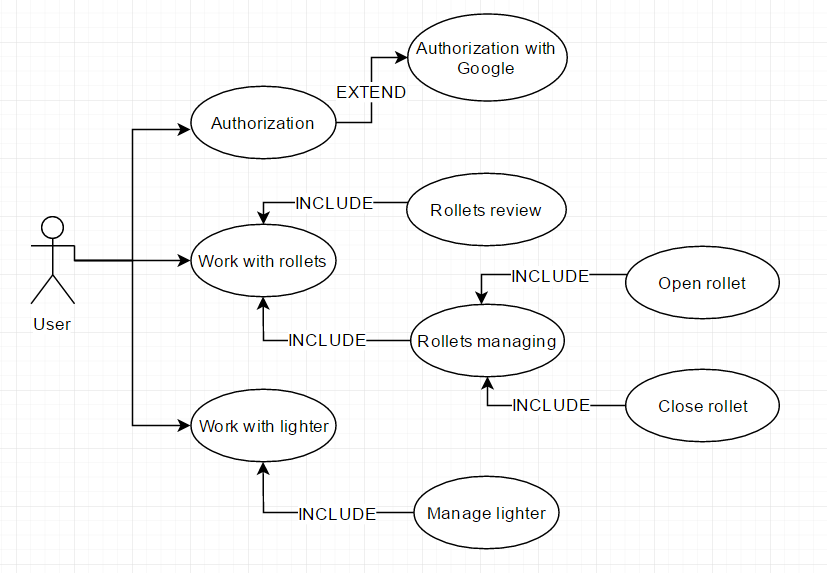


Рисунок 3.1 – Діаграма прецедентів

б) Діаграма класів (див. рис. 3.2) включає в себе серверні класи Roller, як модель ролети та User, як модель користувача, до якого входить номер, ім’я та прізвище користувача; IRolletRepository та IUserRepository – інтерфейси, що визначають всі операції, що повинні бути реалізовані для роботи з базою даних з об’єктами Rollet та User відповідно. RolletRepository та UserRepository – відповідні реалізації цих інтерфейсів.

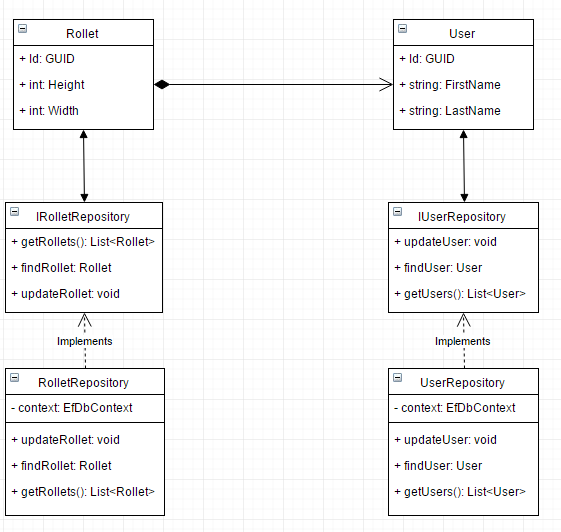


Рисунок 3.2 – Діаграма класів

в) На діаграмі послідовностей (див. рис. 3.3) зображена система в цілому на вищому рівні абстракції та взаємодія основних компонентів – клієнту, сервісу та менеджера бази даних;

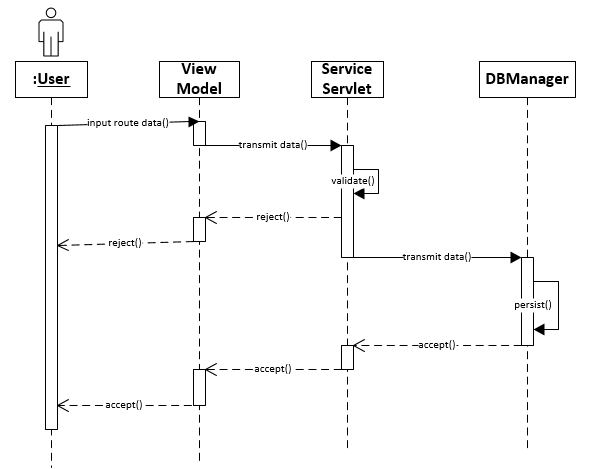


Рисунок 3.3 – Діаграма послідовностей

г) На діаграмі станів (див. рис. 3.4) зображений процес перегляду та конфігурування датчика освітлення та ролет. Для того, щоб користувачу було доступні можливості додатку, він повинен авторизуватися або зареєструватися у системі;

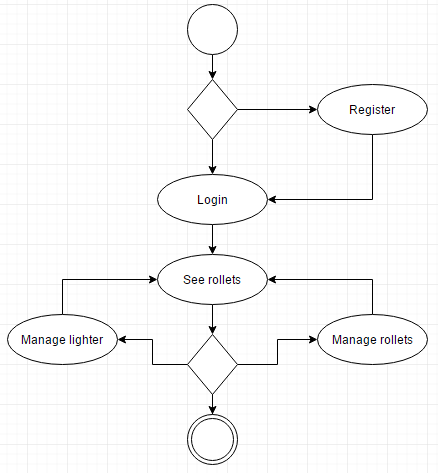


Рисунок 3.4 – Діаграма станів

д) На діаграмі компонентів (див. рис. 3.5) зображені основні частини системи веб-клієнт, мобільний клієнт, основний сервер системи, на якому розмішений сервіс та окремо класи даних, та сервер баз даних;

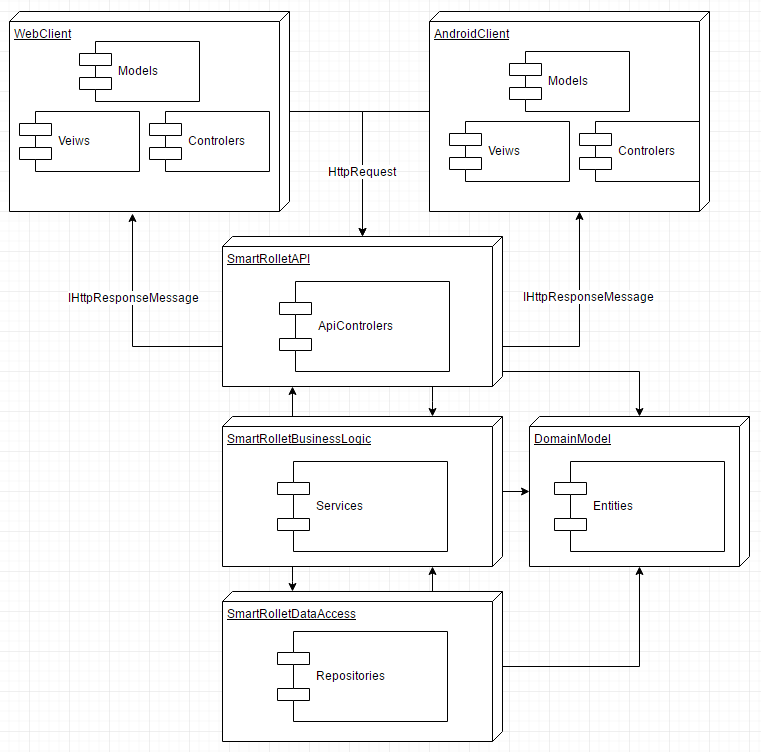


Рисунок 3.5 – Діаграма компонентів

е) На діаграмі кооперації (див. рис. 3.6) зображені стосунки між користувачем, датчиком освітлення та ролетою. Користувач може піднімати чи опускати ролету або налаштовувати датчик освітлення, який в свою чергу теж може піднімати чи опускати ролету відповідно до свого налаштування.

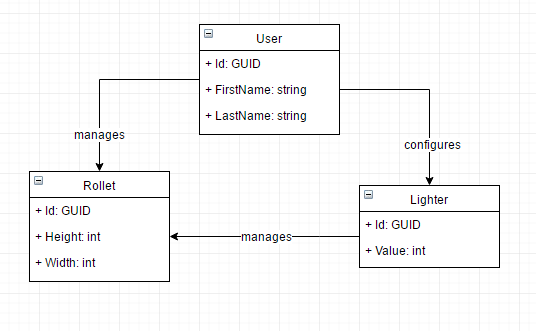


Рисунок 3.6 – Діаграма кооперації

4 РЕАЛІЗАЦІЯ ПРОГРАМНОГО ПРОДУКТУ

Під час планування та моделювання програмного продукту було визначено, що проект має складатися з трьох частин [4]: сервер, на якому будуть зберігатися дані, до серверу підключаються клієнти за допомогою REST API. На першому етапі реалізується веб-клієнт та мобільний додаток на платформі Android.

Для розробки серверу використовується технологія створення Web-застосувань мовою С# [5] з використанням .Net фреймворку.

Для розробки використовується середовище Visual Studio 2015 та платформа Microsoft .Net. Для виклику методів сервера використовуються REST контролери з фреймворку Spring, що дозволяє отримувати та відправляти різноформатні дані по протоколу http.

На сьогоднішній день прийнято використовувати REST - (скор. від англ. Representational State Transfer - «передача репрезентативного стану») - метод взаємодії компонентів розподіленого додатка в мережі Інтернет, при якому виклик віддаленої процедури являє собою звичайний HTTP-запит, а необхідні дані передаються як параметри запиту.

У свою чергу HTTP — протокол передачі даних, що використовується в комп'ютерних мережах. Назва скорочена від Hyper Text Transfer Protocol, протокол передачі гіпер-текстових документів.

HTTP — протокол прикладного рівня, схожими на нього є FTP і SMTP. Обмін повідомленнями йде за звичайною схемою «запит-відповідь». Для ідентифікації ресурсів HTTP використовує глобальні URI. На відміну від багатьох інших протоколів, HTTP не зберігає свого стану. Це означає відсутність збереження проміжного стану між парами «запит-відповідь». Компоненти, що використовують HTTP, можуть самостійно здійснювати збереження інформації про стан, пов'язаний з останніми запитами та відповідями. Браузер, котрий посилає запити, може відстежувати затримки відповідей. Сервер може зберігати IP-адреси та заголовки запитів останніх клієнтів. Проте, згідно з протоколом, клієнт та сервер не мають бути обізнаними з попередніми запитами та відповідями, у протоколі не передбачена внутрішня підтримка стану й він не ставить таких вимог до клієнта та сервера

Дані зберігаються у базі даних Sql типу, доступ до якої здійснюється за допомогою Entity Framework 6. Microsoft SQL Server — комерційна система керування базами даних, що розповсюджується корпорацією Microsoft. Мова, що використовується для запитів — Transact-SQL, створена спільно Microsoft та Sybase. Transact-SQL є реалізацією стандарту ANSI/ISO щодо структурованої мови запитів (SQL) із розширеннями. Використовується як для невеликих і середніх за розміром баз даних, так і для великих баз даних масштабу підприємства. Багато років вдало конкурує з іншими системами керування базами даних.

В якості системи контролю версій використовується Git.

Для розробки Front end веб-клієнту використовується Java Script, CSS та html, а для розробки серверу використовується фреймворк .Net.

Для розробки мобільного додатку на платформі Android використовується Eclipse Luna з Android SDK та мова Java. Додаток не має бази даних, усі дані для зберігання відправляються на сервер POST-запитом, а для відображання - GET-запитом.

Якщо описувати модель взаємодії усіх компонентів системи, то можна зрозуміти, що головною ланкою між усіма клієнтськими додатками є сервер з REST-сервісом. В усіх трьох компонентах системи присутні одні й ті ж класи-моделі для зручності передачі даних.

5 ІНТЕРФЕЙС ТА ФУНКЦІОНАЛ

Як вже було неодноразово сказано, програмна система складається з трьох компонентів: серверу, веб-клієнту та мобільного додатку. Під час розробки програмної системи однією з цілей було створити найбільш простий інтерфейс у мобільному додатку, шляхом перенесення більшості функціоналу у веб-клієнт. Тому було вирішено залишити можливість переглядати список своїх маршрутів і визначати своє положення під час поїздки у мобільному додатку, а створення маршрутів, пошук по публічних маршрутах реалізувати у веб-клієнті.

5.1 Сервер

На сервері реалізований REST-сервіс, за допомогою якого можна скористатися функціями системи. Методи, що вертають дані, вертають їх у форматі JSON. Наведемо список доступних методів для API:

а) GET “/rollet” – вертає всі ролети користувача;

б) GET “/rollet/{id}” – вертає всі ролети користувача з заданим ідентификатором;

в) PUT “/roller/{id}” – зберігає зміни, пов'язані з ролетом з заданим ідентификатором;

г) GET “/lighter” – повертає налаштування датчика освітлення;

д) PUT “/lighter” – зберігає налаштування датчика освітлення.

5.2 Веб-клієнт

Веб-клієнт є найбільш насиченим з точки зору інтерфейсу. Головна сторінка клієнту являє собою форму, за допомогою якої можна авторизуватися використовуючи Google-акаунт. Після авторизації користувач потрапляє на сторінку профілю (див. рис. 5.1), де він може побачити список всіх своїх маршрутів.

// TODO

5.3 Додаток для Android

Мобільний клієнт має досить простий інтерфейс, що повністю відповідає усім функціям, що він повинен виконувати. Після відкриття додатку користувач має авторизуватися через свій Google-акаунт, натиснувши кнопку «Log in with Google».

// TODO

ВИСНОВКИ

У результаті виконання курсової роботи був розроблений програмний продукт, що дозволяє керувати веломаршрутами та поїздками за ними. Система базується на сервісно-орієнтованній архітектурі, тому складається з окремих частин: сервера, веб-клієнта та мобільного додатку.

Сервер та клієнт для платформи Android було розроблено з використанням мови програмування Java. Сервер було розроблено у середовищі Visual Studio 2015, мобільний додаток у Android Studio. Сервер розроблявся з використанням фреймворка .Net. База даних розроблена в середовищі SQL Server Management Studio 2014.

В результаті розробки поставлену задачу було цілком виконано. Програмний продукт має зрозумілий інтерфейс для користувачів, не викликає труднощів з відправкою даних на сервер.

Як вже зауважувалося вище, програмний продукт навіть після реалізації усіх поставлених на курсову роботу задач, для комерційного запуску має дороблюватися. Перш за все, необхідно доробити додатковий функціонал та створити комерційну складову проекту. Другорядним завданням стоїть реалізація клієнтів для різних мобільних платформ (iOS, OS Android, Web), це дозволить покрити більшість сучасних смартфонів та планшетів. Також, необхідно доробити локалізацію.

ПЕРЕЛІК ПОСИЛАНЬ

1. Вікіпедія [Електронний ресурс]/ Режим доступу: www/URL: http://uk.wikipedia.org/wiki/GPS - GPS.
2. MongoDB docs [Електронний ресурс]/ Режим доступу: www/URL: http://docs.mongodb.org/manual/ - The MongoDB 3.0 Manual.
3. Фаулер, М. UML. Основы [Текст] : пер. с англ. А.: Петухов/ М. Фаулер, К. Скотт. - СПб.: Символ, 2006. - 184 с.
4. Мартин, Р. Чистый код. Создание, анализ и рефакторинг [Текст] : пер. с англ. Е.: Матвеев/ Р. Мартин. - Питер, 2010. - 464 с.
5. Еккель, Б. Философия Java [Текст] : пер. с англ. Е.: Матвеев/ Б. Еккель. - Спб: Питер, 2001.- 880с.
6. Смирнов, Н. И. JAVA 2 Enterprise. Основы практической разработки распределенных корпоративных приложений [Текст]/ Н.И. Смирнов. – М.: КУДИЦ-ОБРАЗ, 2002. – 240 с.