КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМЕНІ ТАРАСА ШЕВЧЕНКА

Кафедра інформаційних систем

Курсова робота

Розробка інтернет-застосунків для мобільної платформи Windows Phone 8.1

Керівник курсової роботи

доц. Галкін О. В.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

“\_\_\_\_” \_\_\_\_\_\_\_\_\_\_ 2015 р.

Виконав студент 3-го курсу

Кожухівський В. А.

“\_\_\_\_” \_\_\_\_\_\_\_\_\_\_ 2015 р.

**Київ - 2015**

|  |  |
| --- | --- |
| ЗМІСТ | № стор. |
| Реферат | 3 |
| ВСТУП | 3 |
| РОЗДІЛ 1 АНАЛІЗ ІСНУЮЧИХ РІШЕНЬ | 4 |
| РОЗДІЛ 2 ОПИС ЗАСТОСУНКУ | 5 |
| 2.1 Призначення застосунку | 5 |
| 2.2 Головні функції | 5 |
| РОЗДІЛ 3 ВИБІР ПРОГРАМНИХ ІНСТРУМЕНТІВ | 5 |
| 3.1 Загальний опис | 5 |
| 3.2 Бібліотека Json.NET | 6 |
| 3.3 Модуль System.Net | 8 |
| РОЗДІЛ 4 ПРОЦЕС РОЗРОБКИ ЗАСТОСУНКУ | 8 |
| 4.1 Запити до веб-сайту та перетворення відповідей в об'єкти | 8 |
| 4.2 Інтерфейс користувача | 11 |
| 4.3 Фонові процеси | 12 |
| 4.4 Процес тестування | 12 |
| 4.5 Середовище розробки | 12 |
| 4.6 Проблеми, що виникли в процесі, та їх вирішення | 13 |
| ДОДАТКИ | 13 |
| Додаток 1 - Знімки екрану | 13 |
| РЕСУРСИ | 14 |

**РЕФЕРАТ**

Курсова робота складається із вступу, чотирьох розділів та списку інтернет-ресурсів, що були використані в процесі розробки застосунку (в кількості **TODO**). Робота містить **TODO** рисунки та **TODO** таблиці. Загальний обсяг роботи становить **TODO** сторінок, основний текст викладений на **TODO** сторінках.

**Ключові слова:** WINDOWS PHONE 8.1, C#, WINRT, METRO, INTERNET-APP, JSON, .NET, КУПІВЛЯ ОНЛАЙН, ЗАЛІЗНИЦЯ, МОБІЛЬНА ПЛАТФОРМА.

**Реферат.** В роботі досліджено роботу існуючих на даних момент сервісів, котрі надають послуги із купівлі залізничних квитків онлайн, було виявлено незручність у системі продажу квитків Укрзалізницею та відсутність відповідних засобів у даних сервісів, що б позбавляли користувачів даної незручності. Метою роботи була розробка відповідного сервісу.

**ВСТУП**

Відомо, що для успішного придбання квитків на потяг потрібно брати їх заздалегідь, особливо якщо це такі популярні напрями, як львівський, київський чи кримський. Недавно Укрзалізницею було впроваджено деякі зміни до системи продажу квитків, опісля яких вагони додавались до потягів за необхідності. Таким чином, якщо спізнився на першу хвилю продажу квитків, треба по кілька разів на день перевіряти, чи не додали часом вагонів.

З незрозумілих причин з усіх онлайн-сервісів з продажу квитків, лишень один дає можливість "замовити" квитки, тільки в ньому дані про наявність будуть надіслані на електронну пошту, що є не дуже зручно, оскільки дуже невелика кількість користувачів використовує мобільні пристрої для доступу до їх електронних скриньок.

Уже давно виникала думка, що було б непогано, якби був сервіс, який би це робив замість нас. Саме така ідея і закладена як базова для даного застосунку. Його головна задача - це перевіряти наявність квитків, використовуючи онлайн сервіси, та повідомляти користувача.

**РОЗДІЛ 1 АНАЛІЗ ІСНУЮЧИХ РІШЕНЬ**

На даний момент є декілька існуючих сервісів придбання квитків онлайн:

* <http://booking.uz.gov.ua/>



* <http://miykvytok.com/ua/train>



* <https://www.ipay.ua/ua/zhdBiletyi/kupit_biletu_na_poezd>



* <https://bilet.privatbank.ua/train>



* <https://gd.tickets.ua/uk>



* <https://uticket.ua/>



* <http://www.plategka.com/uk/kupiti-bileti-na-poizd-onlajn>



Ні один із наведених вище сервісів не надає належних послуг. Даний функціонал зовсім відсутній у всіх сервісах, окрім TICKETS.UA. У останньому ж є спосіб попросити сповіщення, однак воно прийде на пошту, що не є зручним для більшості користувачів. І, на мою думку, дуже мала ймовірність появи сповіщень на телефон, оскільки висока ціна, а також користувачі з недовірою ставляться до розголошення номеру телефону в інтернеті.

**РОЗДІЛ 2 ОПИС ЗАСТОСУНКУ**

Зважаючи на все сказане вище, з'явилась ідея щоб, де людина не була б, вона завжди могла б бути вчасно сповіщена і не втратити нагоду вчасно придбати проїзний документ.

**2.1 Призначення застосунку**

Головним призначенням застосунку є сповіщення користувача про доступність квитків у відповідності до його бажань. Користувач має можливість задавати власні побажання (дати і години відправлення, дати та години прибуття, бажані місця).

**2.2 Головні функції**

* Пошук квитків за заданими станціями та на певні дати
* Додавання нагадувань про майбутні поїздки на потягу
* Додавання побажань щодо майбутніх поїздок

**РОЗДІЛ 3 ВИБІР ПРОГРАМНИХ ІНСТРУМЕНТІВ**

**3.1 Загальний опис**

Як мову програмування було обрано C# через особисті вподобання, а також через ряд переваг. Наприклад, в перспективі, застосунок, написаний даною мовою, можна портувати на інші мобільні системи такі, як Android i iOS.  
Також, дана мова дає більші можливості і легші способи їх реалізовувати. Однією із таких є можливість асинхронного програмування.

Windows Phone 8.1 було обрано, як платформу для розробки через наступні чинники:

* це платформа, яка швидко розвивається
* розробка застосунків під дану платформу набагато зручніша і приємніша
* наявність пристрою, що працює на даній платформі
* особисті вподобання

Як середовище розробки було обрано Visual Studio 2013, оскільки це на даний момент найкраще середовище розробки для мов родини С, а також поки що це єдине середовище із підтримкою .NET.

Дані для головного функціоналу (такі, як наявність потягів та квитків на них) бралися із відповідних веб-сервісів. Таким чином, поставали наступні задачі: зробити запит і розібрати відповідь. Для запитів використовувалась рідний модуль C# - System.Net, запити робились до ресурсу booking.uz.gov.ua, а, оскільки відповіді надсилались у форматі Json, то також була використана бібліотека Json.NET, яка на даний момент є найпопулярнішою бібліотекою для роботи із Json під платформу .NET.

**3.2 Бібліотека Json.NET**

Причини, чому була обрана саме дана технологія:

#### Серіалізатор JSON світового класу

Серіалізація та де-серіалізація будь-якого .NET об'єкта за допомогою потужного JSON серіалізатора бібліотеки Json.NET.

#### LINQ та JSON

Створення, розбирання, надсилання запитів та зміна JSON використовуючи JObject, JArray та JValue об'єкти бібліотеки Json.NET.

#### JSON Path

Надсилання зпитів до JSON за допомогою синтаксису, схожого на XPath. Більше про JSON Path тут (http://goessner.net/articles/JsonPath/).

#### High Performance

50% faster than DataContractJsonSerializer, and 250% faster than JavaScriptSerializer.

#### Легка у користуванні

Json.NET робить легке ще легшим і важке можливим.

#### Підтримка XML

При потребі, Json.NET підтримує конвертацію між XML і JSON.

#### Open Source

Json.NET - програмне забезпечення із відкритим кодом та є безкоштовним для комерційного використання (https://github.com/JamesNK/Newtonsoft.Json/blob/master/LICENSE.md).

#### Працює всюди

Json.NET підтримує Windows, Windows Store, Windows Phone, Mono, та Xamarin.

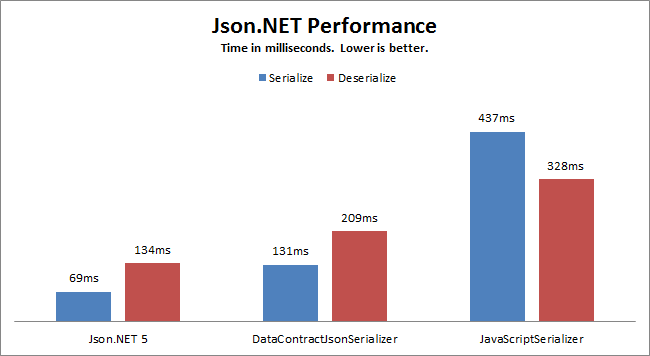
#### Найбільш популярна .NET бібліотека

Більше 10 мільйонів завантажень та в додачу, Json.NET посідає перше місце серед бібліотек в репозиторії NuGet.

Порівняння доступних властивостей:

|  |  |  |
| --- | --- | --- |
| Json.NET | DataContractJsonSerializer | JavaScriptSerializer |
| 63 | 21 | 11 |

Порівняння продуктивності:



Що було використано:

JsonConvert.SerializeObject(object) & JsonConvert.DeserializeObject<Type>(string) та анотації для позначення полів класу, які потрібно серіалізувати. Уривок коду для наглядного показу простоти реалізації функціоналу:

|  |
| --- |
| Серіалізація |
| Product product = new Product();  product.Name = "Apple";  product.Expiry = new DateTime(2008, 12, 28); product.Sizes = new string[] { "Small" }; string json = JsonConvert.SerializeObject(product);  // {  // "Name": "Apple",  // "Expiry": "2008-12-28T00:00:00",  // "Sizes": [  // "Small"  // ]  // } |
| Десеріалізація |
| string json = @"{  'Name': 'Bad Boys',  'ReleaseDate': '1995-4-7T00:00:00',  'Genres': [  'Action',  'Comedy'  ]  }";  dynamic m = JsonConvert.DeserializeObject<>(json);  string name = m.Name; // Bad Boys |

**3.3 Модуль System.Net**

Даний модуль містить клас HttpClient, який дає можливості із надсилання різного типу веб-запитів (POST, GET, PUT,…). Приклад простого запиту:

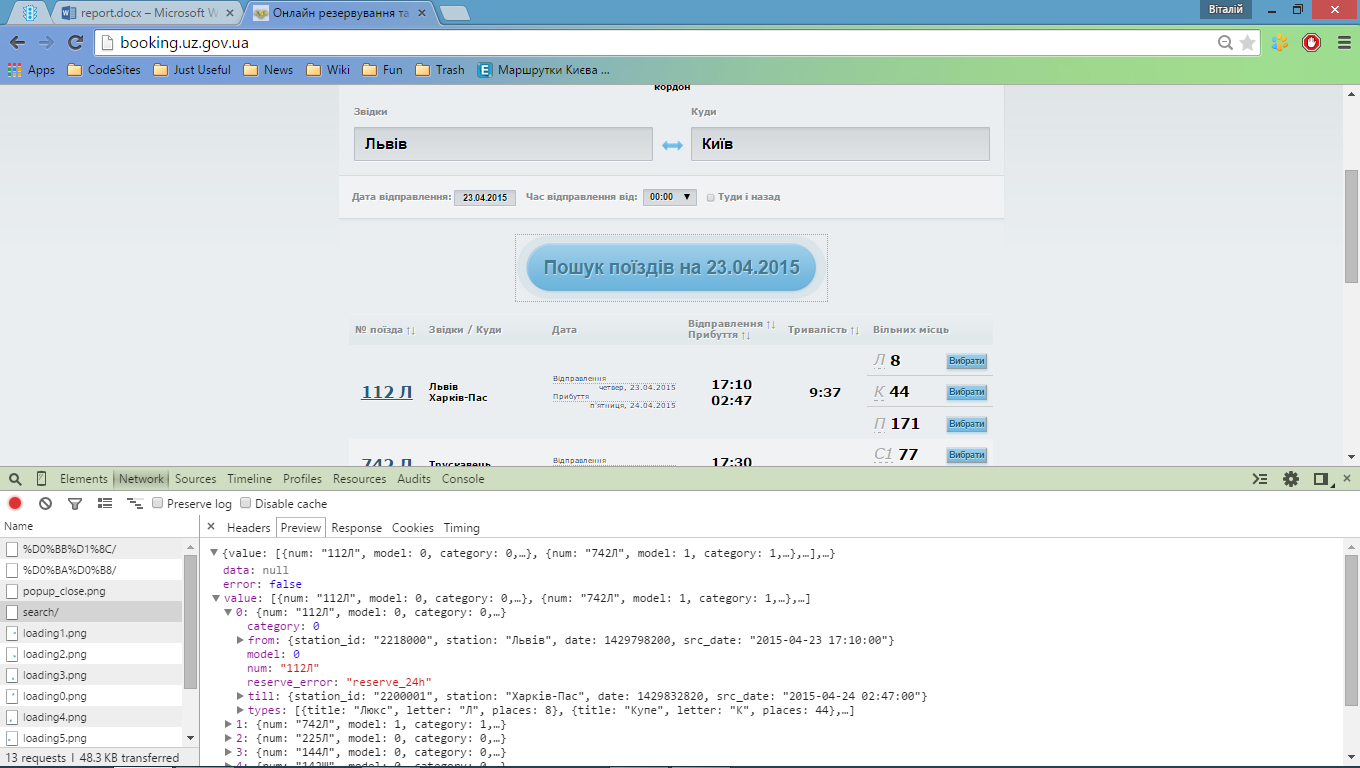
|  |
| --- |
| var httpClient = new HttpClient();  HttpResponseMessage response = await httpClient.GetAsync("https://google.com");  Console.WriteLine(await response.Content.ReadAsStringAsync()); |

**РОЗДІЛ 4 ПРОЦЕС РОЗРОБКИ ЗАСТОСУНКУ**

**4.1 Запити до веб-сайту та перетворення відповідей в об'єкти**

Першим етапом у розробці запитів до веб-сервісу, потрібно було визначити, який же формат запитів та відповідей. Оскільки ні у однієї із систем, представлених у розділі 1, немає офіційного API, то довелось вручну отримувати інформацію.

Задля цього на сайті booking.uz.gov.ua здійснювались звичайні дії для отримання списку доступних місць, а в той же час в консолі розробника в браузері у вкладці "Мережа" відслідковувались запити, що здійснювались до сервера.



Таким чином, було визначено наступне:

1. При введенні назви міста у відповідне поле на веб-сайті, відправляється запит, відповіддю на який є список усіх міст із даним префіксом разом із їх кодами.  
   При виборі міста із списку, його номер зберігається у приховане поле для подальших запитів. Формат:

|  |
| --- |
| Формат запиту |
| POST: http://booking.uz.gov.ua/purchase/station/Київ |
| Формат відповіді |
| {  'value':[  {  'title':'Київ',  'station\_id':2200001  },  {  'title':'Київська Русанівка',  'station\_id':2201180  }  ],  'error':null,  'data':{  'req\_text':['Київ','Rb]d','Киъв']  }  } |

1. Опісля того, як вибрано дві станції та дату відправлення, відбувається запит для отримання доступних потягів. Формат:

|  |
| --- |
| Формат запиту |
| POST: http://booking.uz.gov.ua/purchase/search |
| Параметри запиту |
| station\_id\_from:'2218000'  station\_id\_till:'2200001'  station\_from:'Львів'  station\_till:'Київ'  date\_dep:'01.03.2015'  time\_dep:'00:00'  time\_dep\_till:''  another\_ec:'0'  search:'' |
| Формат відповіді |
| {  'value':[  {  num: '013П',  model: 0,  category: 0,  from: {  station\_id:'2218000',  station:'Ужгород',  date:1425164880,  src\_date:'2015-03-01 01:08:00'  },  till: {  station\_id:'2200001',  station:'Київ-Пасажирський',  date:1425197040,  src\_date:'2015-03-01 10:04:00'  },  types: [  {  title:'Купе',  letter:'К',  places:16  },  {  title:'Люкс',  letter:'Л',  places:2  }  ]  }  ],  'error':false,  'data':null  } |

1. Для отримання доступних вагонів:

|  |
| --- |
| Формат запиту |
| POST: booking.uz.gov.ua/purchase/coaches/ |
| Параметри |
| station\_id\_from:'2218000'  station\_id\_till:'2200001'  train:'219П'  coach\_type:'К'  model:'0'  date\_dep:'1425242700'  round\_trip:'0'  another\_ec:'0' |
| Формат відповіді |
| {  value: {  content: '',  coach\_type\_id:4,  coaches: [  {  num:8, // carriage number  coach\_type\_id:4,  coach\_class:'�',  hasBedding:true,  services: [  'Б',  'Ч'  ],  places\_cnt:10,  prices: {  Б:17500  },  reserve\_price:1700,  allow\_bonus:false  }  ],  places\_allowed:8,  places\_max:8  },  error: null,  data: null  } |

1. Для отримання доступних місцю у вагоні:

|  |
| --- |
| Формат запиту |
| POST: booking.uz.gov.ua/purchase/coach/ |
| Параметри |
| station\_id\_from:'2218000'  station\_id\_till:'2200001'  train:'219П'  coach\_num:'8'  coach\_class:'К'  coach\_type\_id:'4'  date\_dep:'1425242700'  change\_scheme:'0' |
| Формат відповіді |
| {  value: {  places: {  Б: [  0:'32',  1:'34',  2:'36'  ]  },  css:'plac t4'  },  error:null,  data:null  } |

Для кожної відповіді сервера було створено класи, в об'єкти яких за допомогою бібліотеки Json.NET будуть конвертовані Json файли. Для запиту квитків чи нагадування про потяг також створені класи, які конвертуватимуться у Json для збереження у хмарі.

**4.2 Інтерфейс користувача**

Для розміщення елементів на сторінці використовується PivotControl.  
Є три головні сторінки.

1. Пошук (рис. 1)  
   Вводяться назви міст відправлення та прибуття у відповідні поля та вибираються із випадного списку. Підчас введення робляться запити першого типу, що за введеним префіксом шукають можливі станції та їх номери.  
   Потім обирається дата відправлення і натискається кнопка "Пошук".  
   Опісля знизу з'являється список, в якому при дотику до будь-якого елемента відбуватиметься перехід на відповідну сторінку із інформацією про потяг.
2. Придбані (рис. 2)  
   Відображається список придбаних квитків. При натисканні на елемент списку відбувається перехід на сторінку із детальною інформацією про цей квиток.  
   Через контекстне чи головне меню можна здійснювати додавання, видалення, редагування, перегляд інформації про квиток. При виборі певного пункту меню відбуватиметься перенаправлення на відповідну сторінку, на якій буде форма для даних про квиток.
3. Очікуються (рис. 3)  
   Відображається список поїздок, на які користувач бажав би придбати квиток.  
   Дана сторінка поводить себе в точності так само, як і попередня, окрім того, що в даному випадку редагується інформація не про існуючі квитки, а про запити на квитки.

Також є три допоміжні сторінки:

1. Інформація про потяг (рис. 6)  
   На даній сторінці відображається наступна інформація про потяг - номер потяга, пункт відправлення, пункт прибуття, дата і час відправлення, дата і час прибуття, час у дорозі, список доступних вагонів. При натисканні на певний вагон, показується список доступних місць у цьому вагоні. При натисканні на місце, показується його ціна.
2. Інформація про квиток (рис. 4)  
   На дані сторінці відображається наступна інформація про придбаний квиток - пункт відправлення, пункт прибуття, час і дата відправлення, час і дата прибуття. Так міститься поле із можливістю вибрати чи потрібне нагадування про даний потяг.
3. Інформація про запит на квиток (рис. 5)  
   На даній сторінці відображається наступна інформація про запит на квиток - бажаний пункт відправлення, бажаний пункт прибуття, припустимий інтервал дат відправлення, припустимий інтервал часу відправлення, припустимий інтервал часу прибуття, тип вагона, допустиме розміщення місця (нижнє чи верхнє, бокове чи ні, і т. д.), максимально допустима ціна і максимально допустимий час у дорозі.

На кожній із цих сторінок інформація піддається зміні, якщо на них відбувся перехід через пункт меню "Редагувати".

**4.3 Фонові процеси**

За стандартом застосунків для Windows Phone 8.1, за згодою користувача кожен застосунок може бути запущений у фоні за інтервалами часу, що повинні бути кратні 15 хв.   
Дана програма буде запускатись кожні 15 хв. і перевірятиме два пункти.

* для кожного квитка перевірятиметься час до відправлення і якщо він досить малий, то буде видано нагадування про майбутній потяг;
* для кожного запиту на квиток буде здійснено запит до веб-сервера для отримання наявних квитків, які будуть порівнюватись із збереженими локально і при вияві нових буде видано сповіщення про появу квитків для бажаної подорожі.

Формат сповіщень зображено на рис. 7.

**4.4 Процес тестування**

Для тестування у коді відслідковуються усі виключення із подальшим логуваннями результатів.  
Було використано технологію тестування, що називається Telemetry. Відповідно до принципів даної технології у коді програми втілюються методи, що за виникнення помилок надсилатимуть дані про помилки до розробників, які зможуть їх виправити в наступному оновленні.  
Було випущено дану програму для тестування та промодельовано всі можливі варіанти поведінки користувача, підчас яких було зібрано статистику та дані по помилках, проаналізовано та виправлено відповідні недоліки.

**4.5 Середовище розробки**

В якості середовища розробки було використано Visual Studio 2013 - потужне середовище, яке в більшості орієнтоване для розробки на мові C#, а також для платформи Windows Phone 8.1. У ньому є потужний графічний редактор, за допомогою якого можна легко редагувати зовнішній вигляд застосунку із миттєвим відображенням змін. У даному середовищі також є можливість розгортати і налагоджувати застосунок як на віртуальній машині, так і на реальному пристрої.

**4.6 Проблеми, що виникли в процесі, та їх вирішення**

Головною проблемою, що виникала під час розробки, була система захисту веб-сайту Укразалізниці від програм, що відправляють запити. Кожен запит повинен був містити cookie із числом-індикатором, яке визначало б відправника. Це число генерується веб-сайтом і не розповсюджується.  
З метою вирішення цієї проблеми я зв'язався із відповідними особами з персоналу Укрзалізниці, проте вони відмовилися розголошувати цю приватну інформацію.  
Опісля цього було вирішено використати іншу систему із аналогічним АРІ, яку було знайдено і успішно використано.

**ДОДАТОК**

**Знімки екрану**

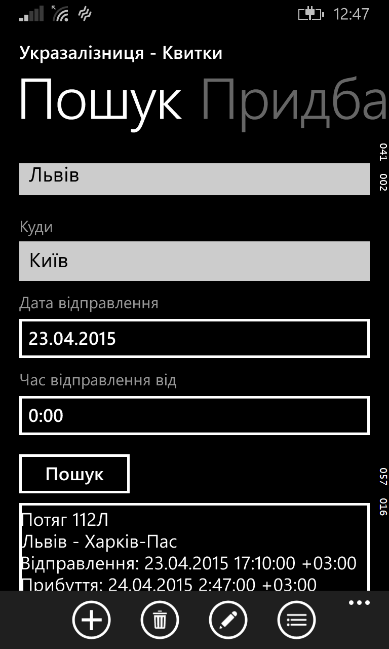
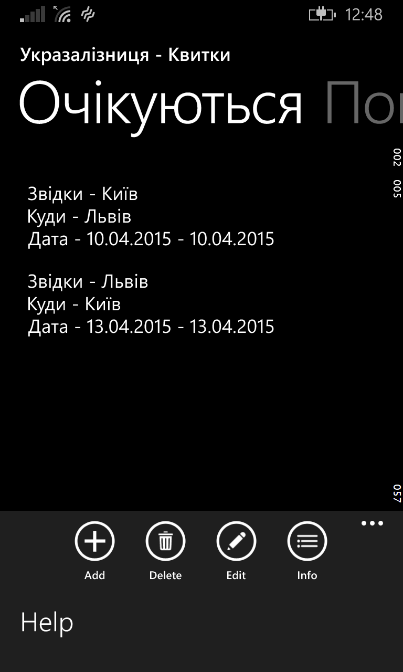
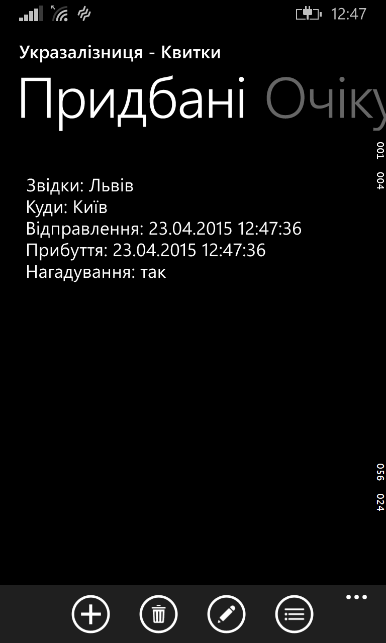
  

рис. 1.1 (сторінка "Пошук") рис. 1.2 (сторінка "Очікуються") рис. 1.3 (сторінка "Придбані")

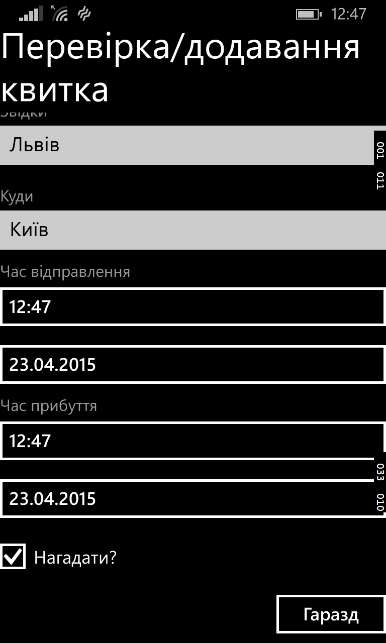
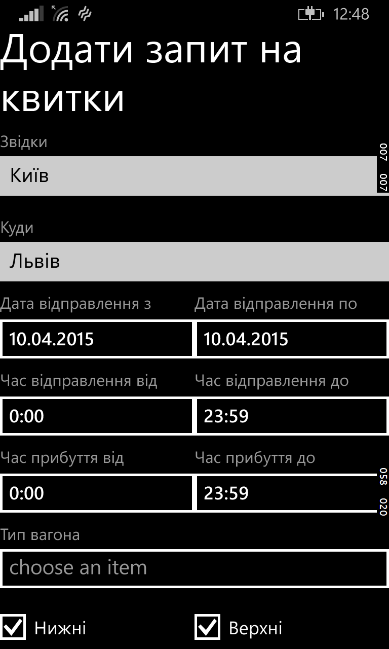
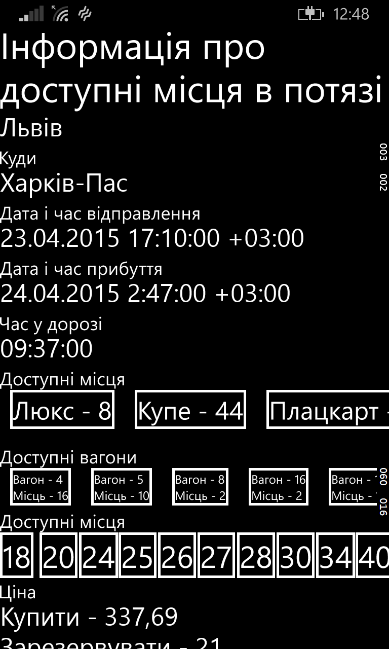
  

рис. 4 ("Квиток") рис. 5 ("Запит на квиток") рис. 6 ("Потяг")

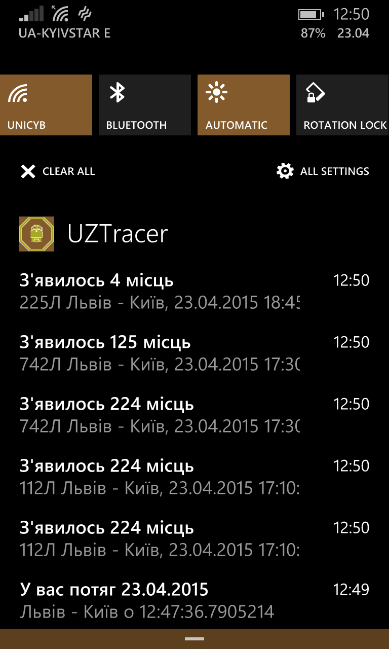


Рис. 7 (Сповіщення про близькі потяги чи появу квитків)

**ВИКОРСИТАНІ РЕСУРСИ**

* <http://www.newtonsoft.com/json/help/html/Introduction.htm>
* <https://msdn.microsoft.com>
* <http://stackoverflow.com/>