

.NET Code Camp 2013

Restaurants @ LUT

Final Report for the .NET Code Camp 2013 project

Group 4

Ilkka Tommola	0208429
Tulibako Paiti	0392517
Doreen Mushi	0392504

Table of Contents

1 Introduction.....	2
2 Motivation.....	2
3 Functionality and Features.....	2
3.1 Viewing Restaurant's Menus.....	3
3.2 Viewing Restaurant Info.....	4
3.3 Viewing Web cam Feeds.....	5
4 Technical Design.....	5
5 Reflection.....	6
6 Conclusion.....	7

1 Introduction

There are about four restaurants/ cafeterias at Lappeenranta University of Technology (LUT) and each restaurant have different operating hours, also each restaurant offer different types of menu which happen to be prepared already on a day-to-day basis and on a weekly basis.

Restaurants @ LUT is the Windows Phone 8 application for restaurants/cafeterias available at LUT. We developed the application using Visual Studio 2012, C# programming language and JSON (JavaScript Object Notation) libraries.

The application make possible for users to do the following:

- View information about available restaurants at the University
- View information about operating hours (opening and closing hours)
- View information about daily and weekly menus
- Extract menus using JSON Technology
- View Live Web cam from restaurant locations

2 Motivation

Due to the setting of the activities at LUT, sometimes a person may need to know the available menus and/or know the restaurant's operating hours and also know the status of the queues to the restaurants from anywhere before visiting the restaurants so that he/she can properly plan for the Lunch and breakfast breaks, also he/she can properly manage the time.

Moreover, students and lecturers who are attending a certain lecture can easily access the restaurant information from a phone than logging-in to a laptop or go out of the classroom and access desktop computers.

Also, the application will make possible for users who are outside the school campus who does not have access to computers and want to come to the campus for eating. They can just access the application through their mobile phones.

Therefore in order to simplify the accessibility of the restaurants information we thought of developing a Windows Phone 8 application so that users can be able to access restaurant's information from any location at any time through their phones.

3 Functionality and Features

The main functionality of the application is to show menus for different restaurants and cafeterias at LUT so that the user may decide what to eat. Fetching the menus to be shown in the application is limited to Sodexo restaurants because Aalef restaurant doesn't offer the menus on Web in easily parseable form and Aalef's menu is not translated to English.

User can also see information additional information about restaurant's opening hours for each restaurant.

By viewing the web cam feed user can quickly see if there are long lines in the restaurant. Web cam feed is provided by computer club Ruut and feed is available for the Sodexo restaurant at the main building (feed is currently broken) and Aalef restaurant at the Student's union house.

Screen shots of application's all main functionalities can be seen on the code camp wiki page http://codecamp.fi/doku.php/dotnet2013/group4/start#user_interface_examples

3.1 Viewing Restaurant's Menus

There are three ways to view info about restaurant's menu.

First user can click one of the restaurant buttons from the application's Main page – see illustration 1 – and select “Day's menu”. The user is shown a page that has a scrollable list which consists of multiple food categories (e.g. “Lounas”, “Vegetarian”, ...) and English names for courses.

Second user can click one of the restaurant buttons from the application's Main page and select “Week's menu”. The user is shown a panorama page that has week's menu for selected restaurant. Page can be scrolled both up and down. There is one panorama item for each day so swiping to the left will show next day's menu and swiping to the right will show the last day's menu. This feature wasn't working fully by the end of code camp, because the lists weren't populated properly, but we managed to fix this right after the code camp had officially finished.

Third user may click “Eat” button from Application bar. A panorama page for all today's menus for all three Sodexo restaurants is shown. First panorama item shows the menu for Sodexo restaurant at the university's main building and next item can be seen by panning to the left. This feature wasn't presented on Friday's demo day at all because it was fully implemented only after code camp.



Illustration 1: Main page of the application. There are four clickable restaurant buttons and two buttons ("eat" and "settings") on the application bar. Note: "eat" button (the empty circle) doesn't have a proper icon and the application doesn't have any settings just yet.

3.2 Viewing Restaurant Info

The user can click one of the restaurant buttons from the application's Main page and view additional information about each restaurant. Additional information may include information about opening hours, lunch hours and contact information. All information on info page is that is shown to the user is static information that has to be typed manually by developers before the release of application.

3.3 Viewing Web cam Feeds

There are two restaurants that have the “Web cam” button. Clicking the button will open a minibrowser inside the application so that web cam feed can be shown to the user.

4 Technical Design

Sodexo restaurants provide the information about menus through web by serializing the data by using JSON. A snippet of real JSON data is given in illustration 2. First there is some meta data that we aren't interested in. Then all courses for requested date are listed in an array. This array is easy to transform into a C# class, see illustration 3.

```
{"meta":{
  "generated_timestamp":1359728392,
  "requested_timestamp":1359669600,
  "ref_url":"http://www.sodexo.fi/lrannanyliopisto",
  "ref_title":"Lappeenrannan teknillinen yliopisto"
},
"courses":[
  {
    "title_fi":"Mustajuurikeittoa",
    "title_en":"Black salsify pure soup",
    "category":"Lounas",
    "price":"0,97 \\/ 3,21 \\/ 5,64",
    "properties":"G, L"
  },
  {
    "title_fi":"Papuja sweet & sour kastikkeessa",
    "title_en":"Beans with sour&cream sauce",
    "category":"Vegetarian",
    "price":"1,74 \\/ 3,98 \\/ 5,79"
  }
],
```

Illustration 2: Example of real JSON data from Sodexo web service.

```

public class Cours
{
    public string title_fi { get; set; }
    public string title_en { get; set; }
    public string category { get; set; }
    public string price { get; set; }
    public string properties { get; set; }

    public Cours(String titleEn, String Category)
    {
        this.title_en = titleEn;
        this.category = Category;
    }
}

```

Illustration 3: Class constructed from JSON data, written in C#.

Whenever new JSON data is received through Web, the data is stored into phone's permanent storage called **Isolated Storage** (Windows Phone 7.1 / Windows Phone 8) or **Windows.storage** (only available in WP8). JSON data is cached so that the application doesn't need to make a new Web request every time the user changes pages in the application. Data is stored into permanent store because Windows Phone stops running the application when user exits the application or shuts down the phone which means that the data is lost if it's not stored into permanent store. Every time information about restaurant's menu is requested, application first checks if information is in cache and only makes Web request if data is not available or is too old (more than five days). Each day's JSON data for each restaurant is stored into a separate file.

All Web requests are made in an asynchronous manner which means that if multiple consecutive requests are made there's no waiting time between requests. For some Web requests we use **Reactive** extension (**Microsoft.Phone.Reactive**) which gives powerful options to observe the Web request, but on the other hand learning curve of using this extension properly is quite steep. Because of that in some cases Web requests are just "normal" Web requests.

When the data needs to be presented to the user, we use **Newtonsoft.Json** library to deserialize JSON data into a data type that is easier to manipulate in application.

5 Reflection

This was the first code camp experience for all team members, therefore further effort was needed to cope and form groups with other participants.

Idea brainstorming was rather not successful on the first day, this was due to the fact that the team could not come up with innovative suggestions that would be possible to implement in the 5 days span. On the second day, Ilkka came up with the Restaurants at LUT idea which was directly accepted by the rest of the team. The idea was feasible for the time limit, and it was observed that the availability of RSS and JSON feeds technologies in the Sodexo website would be useful in supporting implementation of the main features of the application.

The team had to spend the second day for planning, assigning of tasks and by going through tutorials on Windows Phone 8 developer tools, options and elements. This was important as most of us were new in the area. Some technical problems were encountered especially with the Emulators, but the problem was solved rather quickly by the administrators.

Tulibako and Doreen worked on the UI Design, Web cam and Information pages while Ilkka worked on the actual coding bit of menu extraction features (for daily and weekly menus). The integration of JSON technology in the application was a challenge and that took a lot of time to make the menu extraction features complete. It was also difficult to incorporate menus from Aalef Cafeteria due to the nature of HTML content from the website and we soon decided to concentrate on Sodexo restaurants.

Storing the application data to permanent storage proved out to be more difficult than expected, because by default it's not possible to store list structures to a file. We tried to use additional libraries that should make it possible to store more complicated structures (like lists and object) directly, but as time was running short, we decided to fall back into simpler, less elegant solutions.

By the end of the code camp, the features of daily menus were successfully implemented, while the parts for weekly menus required more work and time. One week time span was short but it gave the team an opportunity to aim and achieve bigger goals.

The experience has also brought some lessons that could be applicable in future code camps. We have realized that preparation beforehand is very important in order to save time during the actual code camp. Having variety of ideas, knowing the environment, tools, technologies and even teams beforehand is recommended.

6 Conclusion

The overall development process for Restaurants @ LUT application during .NET Code Camp 2013 was challenging but also successful. Team members encountered different challenges such as idea conceptualization, application design and mastering the right tools and technologies (use of JSON technology for extraction of menus). These specific factors contributed to hardships that resulted to having fewer features implemented than the anticipated ones.

One team member however remained optimistic and continued with implementation even right after the Code Camp challenge. More features have been implemented and the application is much more feasible and complete. Still there are many minor things that need to be considered before releasing the application to the public. At minimum all exceptions should be handled properly: for example now if downloading the menus fails or web service returns invalid data, application doesn't notify the user that there is a problem. Also some parts of code needs to be rewritten. In order to save time, some code parts were just created by copy-pasting code and changing some parameters to functions, which makes maintaining the code too expensive if the application needs to be extended.

There are growth opportunities for the product. The future plan is therefore more advanced and directed

towards the market.

The main feature of the current pilot application is menu extraction for Sodexo cafeterias located at LUT only; the team however considers advancing the application by providing Sodexo users with wider options. These options would involve all Sodexo restaurants/cafeterias available in Finland.

Among the deployed technology for this purposes is Bing Maps Silverlight Control, which will be used to integrate Bing Maps by specifying location points for all registered Sodexo Cafeterias. This will enable the user with a power feature of searching for the nearest restaurant depending on his/her current locations and wishes to eat.

One feature that we wanted to implement was a feature which would make it possible for the user to select favourite cafeterias and favourite courses. Based on the selection the application would pick only the interesting courses from the data and show them on a live tile on the phone's start page. Because we didn't even time to try implementing this feature this is something that could be done in the future.

All content in the application is currently written in English, but it should be easy to translate everything in Finnish also.

This has conclusively been a great opportunity for the team. The one week hands-on experience has improved our knowledge and skills in the application and use of ASP.NET platforms and SDK tools. Interaction with other students has also widened our ways of thinking in conceptualizing great ideas into working applications.