SUMMARY

of the license thesis entitled:

**M-TICKETING BASED ON NFC TECHNOLOGY**

**Student:** Roxana BALAG, Computer Science

**Coordinator:** senior lecturer eng. Cosmina IVAN

1. **Project context**

Present transportation systems deal with issues like ticket queues, bus delays, overpacked vehicles, service disruptions. Mobile technologies and services can ease or even eliminate many of these issues that are happening in the daily lives of the travelling consumers.

1. **Project objectives**

In narrow terms, the objectives of this project break down into two categories: rendering ticketing and payment as simple as possible, and putting transit information at the fingertips of the passengers. The purpose of this project is to build a system capable of integrating tickets in the passenger’s smartphone, to make tickets purchasing easy, environment-friendly and available anytime and anywhere, to provide transit related information and a natural, intuitive way of usage.

1. **Bibliographic research**

Electronic ticketing is a form of electronic commerce for different kinds of tickets, having as main characteristic the fact that the tickets are sold and stored in an electronic device, such as smart cards or mobile phones (mobile ticketing). Currently, there are three different possibilities for mobile ticketing: SMS based (fare charged to the phone bill), OCR based (ticket represented as an image that functions as a code, e.g. barcode ) and Contactless NFC based (similar to OCR, compatible with existing contactless smart card infrastructure). Near Field Communication (NFC) is an international standard for short-range wireless connectivity that provides intuitive, simple, and safe communication between electronic devices.

Similar systems include: U’Go mobile application, launched in June 2013, in France, Strasbourg, which offers ticketing on NFC smart phones running Android, and the well-known Oyster Card in London, a contactless smart card using MIFARE technology.

1. **Used technologies**

The chosen development platform was ***Windows Phone 8***, because it promises an unaltered interface, offers great development tools, has a steady design guideline, takes advantage of the C# powerful features and makes cross-platform portability easier due to its switch to Windows NT kernel.

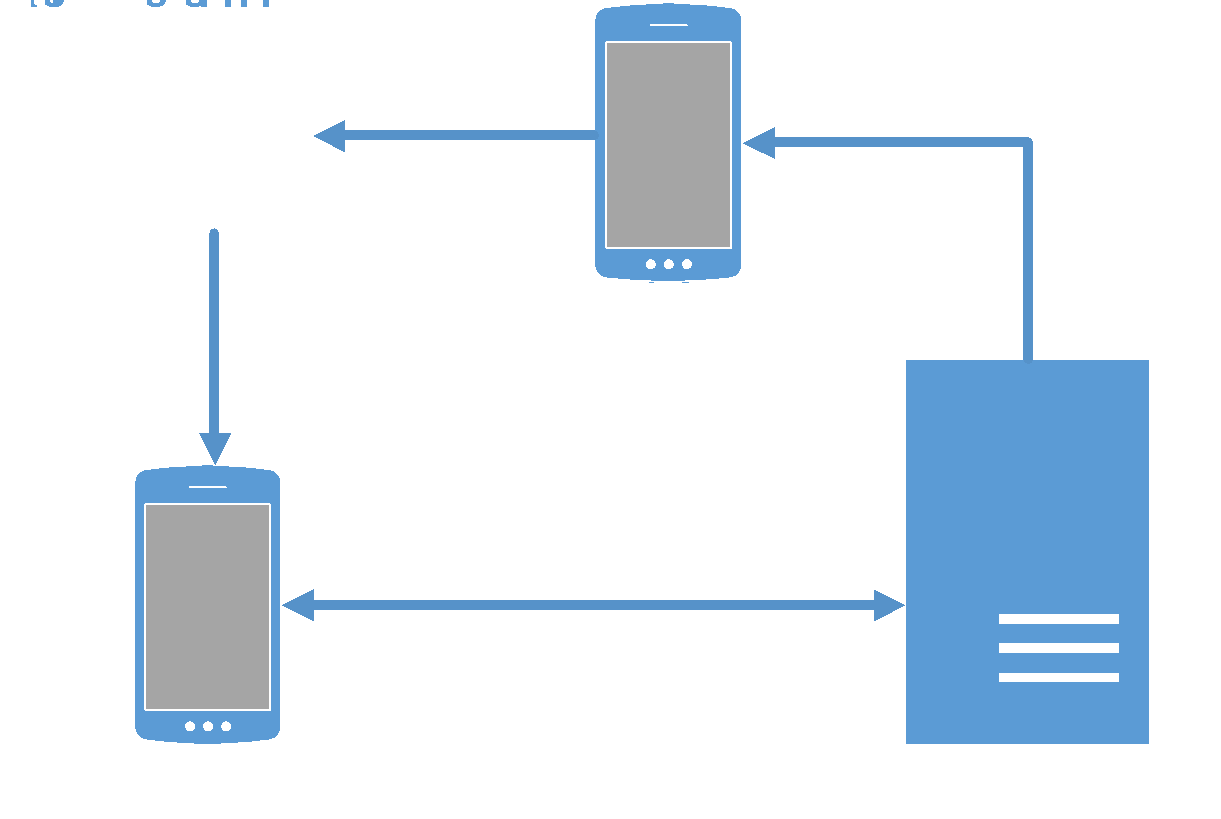
For the web service***, WCF*** (Windows Communication Foundation) was chosen because we have more control over the pieces within the system, it provides great tool support, supports multiple transport protocols, provides increased flexibility, increases interoperability potential and increases developer productivity due to its unified programming model.

***PayPal*** was chosen as the online payment service because it is trusted, easy to use and offers multiple options of payment.

As an ORM, ***Entity Framework*** was chosen because it reduces development time, removes the focus off the database, offers a more application-centric object model, provides independence of the physical/storage model, and offers great compile-time syntax validation for writing queries (LINQ).

For mapping objects between different application layers, ***AutoMapper*** was chosen because it can be used in a fairly generic way without the need to perform manual mapping.

The Proximity API provided by the Windows Phone 8 SDK provides NFC hardware communication and basic NDEF formatting for a very limited subset of the NDEF standards. The missing part, the support for the NDEF standards on top, was added by the ***NDEF Library for Proximity APIs/NFC.***

1. **System architecture**

The system’s architecture is a client-server

architecture. It is a modular achitecture,

composed of three subsytems: the application server, the ticketing application and the tag writing application.

The purpose of the tag writing application is to configure the NFC tag with the needed data from the server.

The purpose of the ticketing application is to provide the the traveler with tickets in an electronic format, along with the ability to use these tickets in an intuitive way.

The purpose of the application server is to expose a web service implemented on

the .NET Framework, that encapsulates the business rules and logic of the system.

1. **Implementation details**

The architecture of the ***application server*** is a 3-tier architecture. The bottom layer (Data Access Layer) was implemented using Entity Framework and applying the Repository Pattern. The Business Layer encapsulates all the business logic of the application and was implemented in a similar way to the repository from the DAL, the difference being that the Business Layer has as data source the DAL and that it works with Data Transfer Objects (DTOs) instead of entity objects (DTOs are mapped from entity objects using AutoMapper). The top layer is represented by the web service implemented using WCF, whose role is to expose the functionality from the Business Layer to the other components of the system.

Both the architecture of the ***ticketing application*** and the ***tag writing application*** are based on the Model-View-ViewModel architectural pattern. Using this pattern, the UI of the application and the underlying presentation and business logic is separated into three separate classes: the view, which encapsulates the UI and UI logic, the view model, which encapsulates presentation and business logic and state, and the model, which encapsulates the application's data. Both applications were implemented using Microsoft’s design guidelines.

1. **Conclusions and future development**

The obtained results were the expected ones. We obtained a system that meets all the functional and nonfunctional requirements expressed at the beginning of the project. Some future development could include: adding new functionalities (notifications, account editing, adding more payment options, etc.), providing offline usage, adding encryption of sensitive data, developing the system for other platforms.

1. **Testing**

A test suite, composed of multiple test cases, has been built for the most important functionalities of the ticketing application (the main focus of this project), which was tested thoroughly. All the applied test case passed this manual testing.

**Date**

July 8th, 2014