UNIVERSITY OF ZAGREB

FACULTY OF ORGANISATION AND INFORMATICS

V A R A Ž D I N

**Viktor Lazar,** [**vlazar@foi.hr**](mailto:vlazar@foi.hr)

**Martina Šestak,** [**msestak2@foi.hr**](mailto:msestak2@foi.hr)

**Goran Vodomin,** [**gvodomin@foi.hr**](mailto:gvodomin@foi.hr)

**Matej Vuković,** [**mvukovic2@foi.hr**](mailto:mvukovic2@foi.hr)

BLUETOOTH LE SHOWCASE

SOFTWARE ANALYSIS AND DEVELOPMENT PROJECT

Varaždin, 2014.

UNIVERSITY OF ZAGREB

FACULTY OF ORGANISATION AND INFORMATICS

V A R A Ž D I N

**Team number:** T1

**Team name:** Heisenbug

**Team members:**

Viktor Lazar, 0108063551

Martina Šestak, 0016091250

Goran Vodomin, 0016092445

Matej Vuković, 0016094754

BLUETOOTH LE SHOWCASE

SOFTWARE ANALYSIS AND DEVELOPMENT

Mentors:

Dr.sc. Zlatko Stapić

Ivan Švogor, mag.inf.

Varaždin, November 2014.

Content

[1. Introduction 1](#_Toc403518062)

[2. Bluetooth Low Energy Technology 2](#_Toc403518063)

[3. User Requirements Specification 4](#_Toc403518064)

[3.1. Introduction 4](#_Toc403518065)

[3.2. Organisational / Functional Areas Affected 4](#_Toc403518066)

[3.3. Requirements 5](#_Toc403518067)

[4. Software development methodology 6](#_Toc403518068)

[4.1. Scrum team 6](#_Toc403518069)

[4.2. Product backlog 7](#_Toc403518070)

[4.3. Tasks 7](#_Toc403518071)

[5. Project diagrams 9](#_Toc403518072)

# Introduction

Evolaris is a company that has been established in 2000 and since then its teams of experts have been developing many web and mobile applications. Besides that, they are also constantly studying new technologies and evaluating if it's possible to apply them to build useful and modern applications.

One of Evolaris' projects is NFC bonus programme that was recently launched at the Shopping City Seiersberg located in Graz. Customers can, by using this application, register online and become „Friend of Seiersberg“. When registering every customer receives a traditional plastic card that contains a “Friend Chip”. When he comes to Shopping City Seiersberg he has to find a so called “Friends Kiosk” and place his “Friend Chip” over the terminal to sign in and begin using the application on his device. Every time he signs in, the user gets some points and vouchers which he can later spend in stores.

A “Friend Chip” which each customer has on his card supports Near Field Communication (hereinafter referred to as NFC) technology. NFC represents a form of contactless communication between devices. To start a communication session a user only needs to wave his device over another device that supports NFC technology. That device acts as a reader/interrogator and he is sending information to the passive user’s device using specific radio frequency.

Figure 1. NFC card,

Source: http://bestnfchardware.com/wp-content/uploads/2013/12/NFCBusinesscards.jpg

Currently Evolaris installed four terminals (readers) inside the shopping center so, whenever a user signs in with his card with NFC tag at that terminal, he can see how many points he has and he can also print his vouchers.

Even though the current system with NFC technology has proven to be successful in practice, they would like to use alternative technologies for the same functionality. This is the area our team will be dealing with; we will do some research about Bluetooth Low Energy (hereinafter referred to as Bluetooth LE) technology and build a mobile application that will mostly have the same functionality as Evolaris’ current application, but instead, it will work using Bluetooth LE.

Our code will be private because of the security policy and uploaded to Team Foundation Server, while everything regarding project documentation and notes from Scrum meetings will be stored in public Github repository available at <https://github.com/MartinaSestak/AiR_BLE.git>.

# Bluetooth Low Energy Technology

Bluetooth Low Energy (also called Bluetooth Smart) is a technology that came public in 2010 and became popular because of lower battery power consumption compared to its predecessor Bluetooth. Thanks to this technology Bluetooth can also be used on smaller devices like watches and toys. It was first supported in mobile devices with Android version 4.3 (Android API 18), but nowadays it’s already available on iOS via iBeacon feature.

Figure 2. Bluetooth LE,

Source:http://upload.wikimedia.org/wikipedia/en/thumb/2/20/Bluetooth\_Smart\_Logo.svg/1280px-Bluetooth\_Smart\_Logo.svg.png

Some features that made this technology stand out from classical Bluetooth technology are:

* Ultra-low peak, average and idle mode power consumption
* Ability to run for years on coin-cell batteries
* Lower implementation costs
* Multi-vendor interoperability
* Enhanced range

Compared to NFC technology, both Bluetooth LE and NFC define protocols for establishing a radio link and transferring data between two devices, where one device is considered to be a master (reader, scanner) and the other one is a slave (card, advertiser).

However, when it comes to range Bluetooth LE is better than NFC because it works over several meters of distance, while NFC will only work over a couple of centimeters. Also, a NFC communication session between two devices will start only after the user himself holds his device above the other device (reader) and a bidirectional data transfer will start immediately. Unlike that, Bluetooth LE doesn’t require the user to do anything, communication between devices can start when the user’s device is in reader’s range.

However, data transfer begins after the “pairing” process which establishes the connection between devices (for instance, exchanging keys). There is another difference that can be noticed when comparing these technologies, and it’s regarding devices’ power supply. Bluetooth LE demands both devices to be powered with less power consumption compared to the standard Bluetooth. On the other side, when using NFC one device (usually NFC tag) can be passive and inactive until it starts communicating with the other device (reader).

Bluetooth LE has the potential to change the way costumers and retailers interact, but only for costumers with the latest smartphones and tablets.

# User Requirements Specification

### Introduction

#### Objectives

This is the User Requirements Specification for Bluetooth LE Showcase project, for use by Evolaris GmbH, team members and project mentors. In this section we will be determining the project’s scope, user requirements that need to be satisfied in our mobile application and describing our task that was assigned to us by Evolaris’ mentors.

#### History

Evolaris GmbH already developed the entire system and installed terminals inside The Shopping City Seiersberg. New users register online or at a terminal and receive a card with NFC tag. When they arrive at the shopping center, they need to sign in at a terminal in order to gain shopping benefits. After that they can see their “loyality” points and vouchers status and print vouchers. Our assignment is to try to implement the same functionality using Bluetooth LE technology by building a mobile application.

#### Scope

While doing this project we are expected to build a mobile application that will start when the terminal recognizes the user with Bluetooth LE turned on in its range. When using the application for the first time, the user has to sign in and after that his credentials will be stored in the database. A signed in user can see how many points and what kind of vouchers he has and receive additional points or vouchers each time he is inside the shopping center.

### Organisational / Functional Areas Affected

#### Assumptions

The application that we build will periodically need to be tested in real environment (The Shopping City Seiersberg) and we will need access to customers’ database and server instance.

### Requirements

All requirements are defined in point form and are rated either Mandatory (M) or Highly Desirable (HD) or Desirable (D), dependent on business need and University Policy.

#### Functional Requirements

##### 3.3.1.1. Common Features

| **Requirement** | **Preference** |
| --- | --- |
| * + - 1. User login | M |
| * + - 1. User can see his points status and receive new ones | M |
| * + - 1. User can see his points history | M |
| * + - 1. User can see his personal info | M |
| * + - 1. User can see his vouchers and receive new ones | D |

##### 3.3.1.2. Reporting

| **Requirement** | **Preference** |
| --- | --- |
| * + - 1. Project documentation | M |
| * + - 1. Notes from SCRUM meetings | M |

#### 

#### System Requirements

##### Hardware

| **Requirement** | **Preference** |
| --- | --- |
| * + - 1. Mobile device with BLE support | M |
| * + - 1. Beacon device | HD |

##### Software

| **Requirement** | **Preference** |
| --- | --- |
| * + - 1. Eclipse IDE | M |
| * + - 1. Min. Android API 18 | M |

# Software development methodology

During this project we decided to use agile software development methodology, which emphasizes team communication and continuous collaboration, functional software product and the flexibility to adapt to emerging business needs. Agile methodology includes methodologies like Scrum, Extreme Programming, Dynamic Systems Development etc.

Since Scrum is currently very popular and enables teams to dynamically plan everything regarding the project (releases, resources and functionalities etc.) in cooperation with stakeholders (in this case Evolaris and project mentors), we decided to use this software development methodology.

Among many tools and services, we decided to use Microsoft Team Foundation Service (Visual Studio Online available at <http://www.visualstudio.com/>) provided by Team Foundation Server (hereinafter referred to as TFS) to implement Scrum methodology in our project. As mentioned in [Introduction](#_Introduction), our application code will be stored on TFS, as well as everything related to our software development process.

### Scrum team

According to Scrum, a Scrum team consists of 3 roles: Product Owner, Scrum Master and Development Team. Each of these roles is assigned to a team member, so every team member has some responsibilities shown in Figure 3. However, in accordance with courses’ policies, each team member will be working on developing some part of the final application.

Figure 3. Scrum roles and assignments

We defined our sprints (iterations) according to course’s checkpoints, so our first sprint called Phase 1 started on 7th November 2014 and ended on 14th November [[1]](#footnote-1). Also, each team member’s working capacity is set to 3 hours per day from Monday to Saturday.

### Product backlog

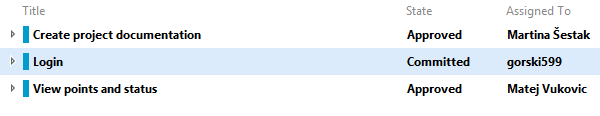
On a product backlog refinement meeting the Product Owner defined and prioritized product backlog items (hereinafter referred to as PBI) that need to be implemented at the end of this sprint (shown in Figure 4).

Figure 4. Product Backlog for Phase 1 sprint

### Tasks

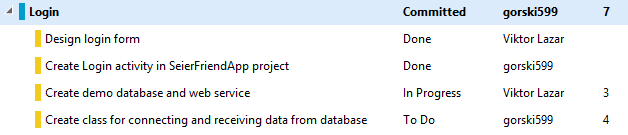
On a sprint planning meeting the Scrum Master in agreement with the entire team analyzed each product backlog item and defined tasks for each PBI. Each task and PBI was assigned to a team member and he has to continuously update task’s status and remaining work time. Figure 5 contains a list of tasks for Login PBI, which is assigned to Goran Vodomin and at that time in “Committed” state with 7 hours of remaining work.

Figure 5. Tasks of a PBI

As mentioned before, all notes from Scrum team meetings can be found in our Github repository.

# Project diagrams

In Graz our team was introduced with user specifications. Based on these information’s we created some UML diagrams. Firstly we will create use case diagram which describes how user can use our application. After that based on use case diagrams, activity diagrams were made. Each activity diagram describes specific use case.

* 1. Use case diagram

In use case diagram we have two actors, user and the system. System is made up of two parts, web service and terminal. Main role of system is to provide data for user about discounts, points and friend status. It also, through terminal, notifies the user when the user arrive into shopping center. On the other hand, user have to be registered so that he could use the application. When the user is registered there are three use cases that describes how to user can interact with application, these are “Starts application”, “Checks points” and “Checks for discounts”. Listed use cases are described on activity diagrams in next chapter.

* 1. Activity diagram

Activity diagram is used to describe dynamic aspects of system. Activity diagram shows flow of control or object flow with emphasis on the sequence and conditions. Based on use case diagram we can see three dynamic aspects of our system, they are “Starts application”, “Checks points” and “Checks for discounts”

* + 1. Starts application

To use application user has to be signed in, so for that purpose application has login form where users enter their username and password. When the data is entered, application passes data to web service. Web service verifies the data and sends information if the entered data is correct. Application receives data and based on information proceeds on or display error message. During entire time applications listens for beacon signals that terminals transmit. If entered data is correct and application received beacon signal, welcome message is displayed and user can use the application. Anytime user can send request for termination which closes the application.

* + 1. Checks points

To check points and see friend status user has to start module for that. In the beginning application instantiates form and send data request to web service. Web service further creates queries on database to retrieve data. When the data is retrieved, web service sends it to application which display the data. Users then reads number of points and friend status. Anytime user can send request for termination which closes the application.

* + 1. Checks for discounts

In order to see list of discounts user starts module for this purpose. Further logic is quite the same as at previous case, application instantiates form, send data request to web service. After that web service creates queries on database to retrieve data. In the end when is retrieved, web service sends it to application which displays the data. User reads list of discounts and chooses two options, to see discount details or goes to menu. Base on the choice application creates menu form or discount details form. Anytime user can send request for termination which closes the application.

* + 1. Checks for discount details

For every discount displayed user can see discount details and for that purpose starts module. After module is started, application instantiates form and sends data request to web service. Web service creates queries to retrieve data, and retrieved data proceeds to application. Application displays data which user read. Anytime user can send request for termination which closes the application.

1. Phase 1 sprint started later because we didn’t know our project scope before our meeting with Evolaris mentors, which took place on 31st October and because we were unofficially studying BLE technology earlier. [↑](#footnote-ref-1)