

University of Glamorgan  
Prifysgol Morgannwg

Faculty of Advanced Technology  
M.Sc. Project

*Photographic game Lila*

*Matúš Vančo*

*12003166*

First Supervisor: *Dr. Gaius Mulley*

Second Supervisor: *Dr. Alhad Kuwadekar*

Year of Study: *2012-2013*

Scheme: *Computing & Information Systems*

# **Abstract**

The project critically evaluates principles of current technologies. Research deals with Android platform, Web technologies and their mutual communication, as well as related technologies. Essential part of the work is design and implementation of simple fictional game for Android, which randomly notifies user once a day and enables him to take a picture with phone camera within limited time period. The server collects these photos and registered users are able to view, comment and vote for the pictures of their friends.

# Acknowledgements

I would really like to thank my supervisor Gaius, who enthusiastically accepted my proposal for M.Sc. project, guided me and motivated for further study. Thanks to him, I was able to start and finalize this work successfully.

University of Glamorgan  
Prifysgol Morgannwg

Faculty of Advanced Technology

STATEMENT OF ORIGINALITY

This is to certify that, except where specific reference is made, the work described in this project is the result of the investigation carried out by the student, and that neither this project nor any part of it has been presented, or is currently being submitted in candidature for any award other than in part for the M.Sc. award, Faculty of Advanced Technology from the University of Glamorgan.

Signed .....

Matúš Vančo

# Contents

<b>1</b>	<b>Introduction</b>	<b>7</b>
1.1	Background . . . . .	7
1.2	Aim and objectives . . . . .	8
<b>2</b>	<b>Literature review</b>	<b>10</b>
2.1	Basic Concepts of Android platform . . . . .	10
2.1.1	Application blocks . . . . .	10
2.1.2	Specific components of application package . . . . .	14
2.1.3	Alarm Manager . . . . .	16
2.2	Web technologies . . . . .	17
2.2.1	Physical layer . . . . .	19
2.2.2	Software platform . . . . .	19
2.2.3	HTTP protocol . . . . .	19
2.2.4	HTML and structure of web . . . . .	21
2.2.5	PHP language . . . . .	22
2.2.6	Database system . . . . .	23
2.3	Pseudorandom number generator . . . . .	23
2.3.1	Introduction . . . . .	23
2.3.2	Linear congruential generator . . . . .	24
2.3.3	Transformation to other distributions . . . . .	25
<b>3</b>	<b>Software requirements specification and analysis</b>	<b>27</b>
3.1	Brief description . . . . .	27
3.2	Technical requirements . . . . .	27
3.3	Progress of work . . . . .	28
3.4	Client analysis . . . . .	29
3.4.1	Run on background . . . . .	29
3.4.2	Daily challenge . . . . .	29
3.4.3	Settings . . . . .	30
3.5	Server analysis . . . . .	30
3.6	Comparison with existing applications . . . . .	30
3.6.1	Snapchat . . . . .	31
3.6.2	Instagram . . . . .	33
3.6.3	Conclusion . . . . .	34

<b>4</b>	<b>Software design and development</b>	<b>35</b>
4.1	Design of client application . . . . .	35
4.1.1	Application screens . . . . .	35
4.1.2	Application logic . . . . .	39
4.2	Server application and communication . . . . .	44
4.2.1	Database schema . . . . .	44
4.2.2	Server communication . . . . .	46
4.2.3	Server structure . . . . .	46
4.3	Implementation . . . . .	48
4.3.1	Preparation of the IDE . . . . .	48
4.3.2	Collaboration . . . . .	50
4.3.3	Administration of Webhosting . . . . .	51
4.3.4	Progress of work . . . . .	51
<b>5</b>	<b>Conclusions</b>	<b>54</b>
5.1	Contribution and benefit of the project . . . . .	55
5.2	Assessment of functionality and security risks . . . . .	56
5.2.1	Vulnerabilities towards the system . . . . .	56
5.2.2	Vulnerabilities towards the application . . . . .	57
5.3	Review of further work . . . . .	58
<b>A</b>	<b>Application screens</b>	<b>60</b>
<b>B</b>	<b>CD content</b>	<b>64</b>
<b>C</b>	<b>Project Logbook</b>	<b>65</b>
<b>D</b>	<b>Git history of Android Application</b>	<b>74</b>

# Chapter 1

## Introduction

### 1.1 Background

Android platform in past few years became the market leader in mobile platforms. Commercial share of this operating system is growing, unlike other mobile platforms. In the same time, more and more people are becoming active users of smart phones and devices called *tablet*, which still supports their development and reduce prices. This leads to even greater sales and the circle is closed. It is not difficult to estimate, that in a few years, there will be only smart phones and we will see even more massive technological madness. We can imagine, that in 100 years every human on planet will have his own tablet available, and through it, he will be carried out daily duties as an order of made meal, home monitoring and a number of others. Great impact in the future will have GPS technology or improved equivalent of this technology, that has currently become an indispensable technology in logistics.

But to have seen it, there is a lot of work ahead of us. Even though, there is already a myriad of various applications and devices, especially for the Android platform, which is very popular between programmers. Just for the popularity of Android, which is caused by support of latest technologies for programmers and very comfortable work with it, there are emerging risks associated with such widespread use. In particular, it is security issue like information abuse. In this area is still potential for improvement.

Research of impact of these technologies and their security problems is carried out through specifically designed application related to Android and Web technology. The reason for it is

the fact, that applied research has often the greatest impact and better results. Motive for this project is to acquire a good knowledge about the extensive possibilities offered by the Android system. This is also the essential part to become part of the Android developer community and thus contribute to the further development and innovation.

## **1.2 Aim and objectives**

The aim of project is to develop an Android application, which randomly notifies user once a day and enables him to take a picture with phone camera within limited time period. There will be developed also Web server, which collects these photos. Users are able to view, comment and vote for the pictures of their friends.

Application is designed with motivation to be different and to be able to meet as much technologies as possible. Namely, these are technologies like HTML, PHP, CSS, SQL and Android platform, where is used various hardware like phone camera, GPS system, vibrator and speakers of phone. Main purpose is to obtain a broader view on different connected technologies as well as to highlight the importance of security and evaluate some vulnerabilities and risks. Another motivation for the future may be also getting valuable data samples from the lives of people which may be subjected to further analysis. This would be more realistic sample of people's lives, unlike presentation on common social networks, because uploaded images of people and places are captured accidentally. This, however, requires further adaptations of application focusing on target group of users. Other requirement is an effective marketing, which does not fall within the scope of this project. But there are found already now some possible motives why users would like to use this application like desire to receive daily pictures of each friend and to have random sample of their live every day. The application is trying to completely distinguished from other, even if it is not entirely possible as there is plenty of applications for every area of life.

Main objectives are research about Web technologies and Android platform development in terms of carrying out the design and implementation of actual project as well as critical evaluation of existing similar Android applications. In application will be made various decisions about the most suitable technical solution as well as design of user interface and application



logic for both parts of the project. Other objective is evaluation of final application functionality in terms of technologies used and propose further development on the basis of learned knowledge.

First part of work, *Basic Concepts of Android platform*, will acquaint the reader with the basic pillars of this platform. In chapter *Requirements Specification and Analysis* are summarized specified requirements for application and analysed in terms of usability, commercial success, user base. There are compared with similar applications and according to that there are designed possible further enhancements. Next chapter *Application Design* designs user interface and designs which technologies should be used for certain problem. It describes principles of the most important algorithm used as well. Chapter *Work flow and implementation* divides the entire application solution to sub-problems, which are easier to solve and mentions the necessary steps, arrangements and measurements with software but also hardware equipment that was necessary to carry out. Here is also description of the environment in which the application was developed and its settings. Next chapter *Results and findings* is the most important chapter as it evaluate whole developed application and findings during development, identifies the most important security issues and design further development. The last chapter *Summary and conclusion* is final chapter and the task is to summarize all the work and point out outputs of each chapter.

# Chapter 2

## Literature review

### 2.1 Basic Concepts of Android platform

*Android* is operating system, which is being developed and sponsored by company Open HandsetAlliance (2013). It includes basic user programs like email client, calendar, integrated internet explorer and more. Android allows programmer to develop very sophisticated applications with using of various hardware, which is available on today's mobile devices (Burnette, 2010). Operating system uses Linux kernel. In this kernel, every application is started under unique user. Every user has rights only for his started application which he originated for. Only user with ID assigned to application can access to private application's files. There is even every application started in its own process implicitly. This process is started when there is started some block of application. Cancel of process is carried out after usage, when it is not more needed or system has to release new memory for other applications.

This paragraph and next sub-chapters are benefiting from book of Murphy (2011) named *Android 2* and many articles on the site *Android Developers* (Google, 2012) (Google, 2013d) (Google, 2013a) (Google 2013g).

#### 2.1.1 Application blocks

Applications are divided into application components in operating system Android. Application component is standalone part of application, which can be utilized independently. There is big

advantage of this approach, because such applications are very reusable. Whole application components can be immediately incorporated in totally different applications and used like libraries without a transcript. We can use e-mail form from build-in e-mail client, which is already present in the device. This hidden logic is invisible for user and it seems like part of our application. Android has four kinds of application components according base class from which component is inherited.

## **Activity**

*Activity* is part of program, which has defined one user interface for whole time. Activity should have one primary task. In case of e-mail client, we may consider one from for sending e-mail as one activity. Activity are implemented through class `Activity` (Google, 2013a).

Application consists of several activities, which are connected to one another. One activity is marked as main and it is different because it is called directly after click on program icon in launcher. However, there is also possibility that program will be firstly called through other activity, but this can be done only by other application, no user.

If activity calls new activity, it pauses itself so that new activity could go to foreground. New activity is also pushed on the stack top. This stack is called back stack, because it stores all activities in order as they have been started and it is used in case user click on back button. When user click on this button, new activity is pulled out from the top of stack and activity from current top stack is started. This is the activity which caused start of previous activity.

Activity can operate in several states:

- **running** - running activity is on foreground and it is fully available for user. Activity is on top stack and user can see it and interact with it;
- **paused** - this is intermediate state between running and stopped activity. Activity is either before start or before stop. In this state, user cannot interact with activity, but it is still visible on screen;
- **stopped** - this is activity fully covered by some other, so it is not visible on screen, but it is still storing some status information;

- **destroyed** - activity is not storing information. In this state activity should use external user-defined rules for recovering previous state. Programmer has to ensure that all needed information will be permanently stored before this state. When the activity go to foreground, these are only information it can use.

Next figure2.1 comprehensively shows life cycle of activity. Grey rectangles represents *callback methods*. Callback methods are simple methods, which implements transition from one state to another. Color ovals are main states, which activity can be in and which have been described before.

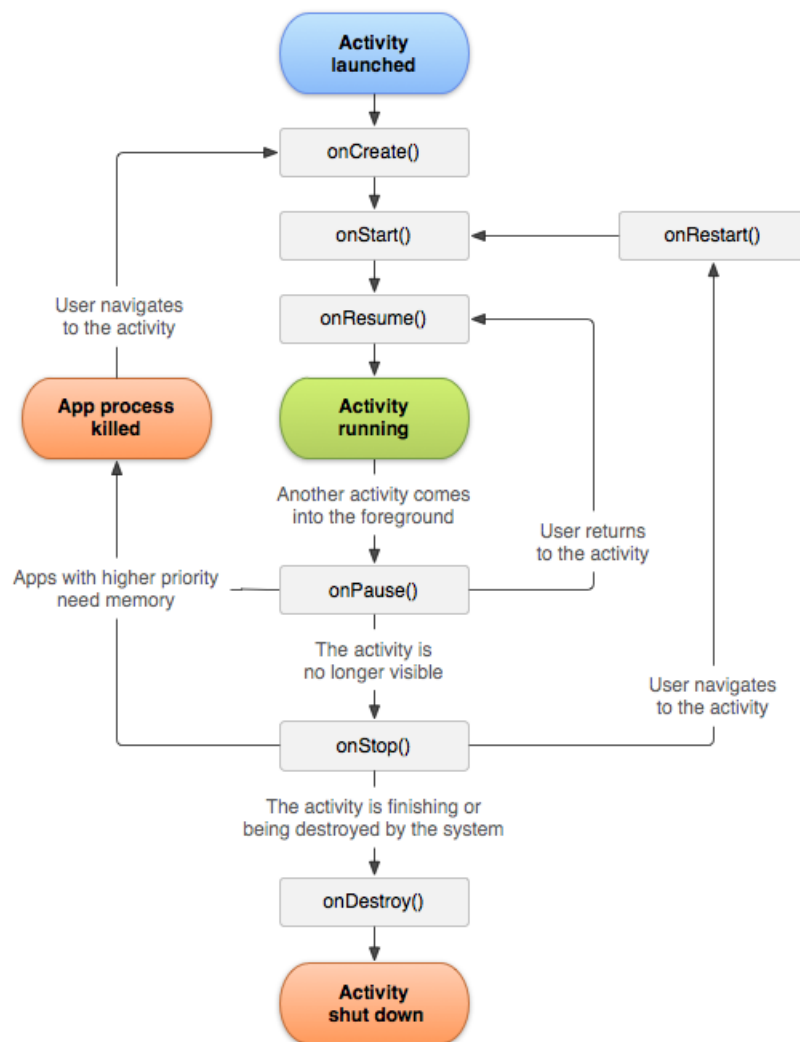


Figure 2.1: Activity life cycle

There are three loops in this figure which are necessary for activity life. Every loop describes on life cycle of some activity attribute. Every activity has these life cycles:

- **existence** - this life cycle starts with calling function `onCreate()`, for creation of activity and ends with calling function `onDestroy()` to destroy activity;
- **visibility** - life cycle starts calling function `onStart()`, when activity starts processing its task but user cannot interact and ends with calling function `onStop()` which causes application is not visible;
- **interactivity** - activity after calling `onResume()` goes to foreground and user starts interact with it, after calling method `onPause()` activity cannot react on user commands.

Other informations about activity and his life cycle are available on official site of Android operating system for Android developers (activities).

## **Service**

*Service* is part of program, which have no user interface. It has one specific task as well as like activity. This task cannot be parameterised easily. For this purpose there is class `Handler` which is aimed for communication between two and more asynchronous tasks. There can be used interprocess communication, when there is communication between two different applications. Service is asynchronous, so that do not block actually running activity. This functionality is implementd in class `Services` (Google, 2013g).

## **Content provider**

*Content provider* is part of program, which represents layer above static data of application. It manages this data and provides uniform access mainly for external applications. External applications must have access right for this action. Example of this application component is application which manages phone contacts. This applications contains content provider class `ContentProvider` (Google, 2013f), in order to other applications can access to phone book and find out telephone number of specific persons. Android, however, allows storing persistent data using SQLite database, programmers can also use Http Apache client and upload files on

web server, files can be stored in application private or public binary or text files as well. Easiest way for storing data is usage of Android embedded mechanism for storing user settings. This mechanism includes also user interface. Other mechanism is using Bundle or Extras classes which are part of Android API and uses file system of device, but the work is more comfortable than direct writing and reading from file.

### **Broadcast receiver**

Application component *Broadcast receiver* reacts on system broadcast. There are lot of notifications and messages coming directly from system. It is case of message that device screen has been turned off or the battery has reached critical state. This messages do not have to come only from system. Also applications can send broadcast for own purposes. Broadcast receiver has no user interface, but it can show notification in notification bar to alert user. To implement broad receiver there is prepared eponymous class `BroadcastReceiver` (Google, 2013e) in Android API.

### **Connection of application components**

Application components *activity*, *service* and *broadcast receiver* are activated by asynchronous message encapsulated in *Intent* class. Message Intent connects one component with each other during the program running. In this message is defined action, which has to be carried out like sending e-mail, and system itself chooses suitable component and respectively class. This applies to activity and service. Other possibility is direct determination of class which has to be invoked. The second option is aimed for calling between one application, first is used mainly between different activities.

### **2.1.2 Specific components of application package**

Application in android is simple ZIP archive with extension APK. This archive has specific structure and hierarchy and contains some specific files and folders with exact name. There are not only Java source files like in common Java application (Zakhour, 2013), but Android package contains also so-called resource files and auto-generated files and even others. This

structure enables more robust and more flexible programming. There is more files, but programming is clear and more resistant to errors of inattention. Very often used file format in these file structure is XML format. It is flexible and configurable format for storing any desired static data like design of user interface. Advantage of resource files is that application logic may be separated from static resources, which can be easily replaced and adapted in order to location settings and device configuration like screen resolution and display rotation. One of specific resources which are not in APK file however is SQLite database. Android enables running database applications through direct support of this database, which is part of Android platform (Wei, 2012).

### **Application resources**

*Application Resources* are files packaged with other application files in APK file and providing multimedia content. Multimedia content is meant by image, sound, video as well as XML files. XML files are text ASCII coded files, which contains structured data. Description of XML files is in book *Java a XML* of author McLaughlin (2006). Resource are grouped in specific file in root directory of installation application file - `res/`.

There are kind of XML files for storing various labels of applications. This are strings which can be replaced during localization of application to other language. Due to this mechanism, application works correctly without interference with application logic, which is represented by Java source files. Other usage of XML files is in design of graphic elements in device screen. Specific type of sources which uses XML files are Preferences. Preferences describe graphic user interface, which is forced for application configuration. Difference between common XML resource file is that in this case there is auto-generated additional application logic for storing global settings.

Resources are considered only as static element of application. This is similar with Java code, which is also static and determined only for reading. Images which are created during application running and activity are not considered as application resources. Main reason of separation resources and functional code is ability to provide alternative resources for different configurations of device. Logic remains retained, but it uses different resources for its purposes by different environment.

## **SQLite database**

SQLite database is favourite database for mobile platforms. It contains clear SQL interface and is very memory-efficient as well as reasonable fast. This is public domain, which everyone can use. Implicitly, it does not provide some features of bigger databases. It does not allow type control of fields. There are only types for interpretation of stored values. There can be inserted values of other types as well. There is also no foreign key integrity constraint, but some problems can be treated more precious application logic or even more comfortable - using triggers. This database is part of runtime environment in system Android and every application is authorized to use it.

## **Android Manifest file**

Basis for each application for Android is Android Manifest file. It is file `AndroidManifest.xml` in the root of application installation file (Murphy, 2011) (Google, 2013c). This file is determined for declaration of application content. In manifest is declared minimum version of Android SDK and access rights, which application requires. This requirements are due to safety viewed during installation of application. In case application requires inadequate rights, user can cancel installation and prevent security threat and suspicious applications. In manifest file is declared application components and classes responsible for its activity as well.

Content of this file is predominantly generated automatically by translation tools of Android SDK. It is automatically created when the project is created and edited when new application component is added. In other special cases programmer should also intervene directly into this file.

### **2.1.3 Alarm Manager**

Alarm Manager is class of Android API and provides access to the system alarm services which are running all the time and ensures invoking intents at specified time (Google, 2013b). In this case, application each day needs to randomly set a specific time when it call service for starting Daily Challenge as described in the chapter about Android platform. Android provides access



to a alarm manager, set and cancel the alarm. This is shown in the sample code (1).

### **Sample code 1.**

```
//access to alarm manager
AlarmManger am = (AlarmManager) getSystemService(ALARM_SERVICE);

//set one-time alarm;
set(AlarmManager.RTC, timeAtMillis, action);
//set repeating alarm
am.setRepeating(AlarmManager.RTC, timeAtMillis, interval, action);

//cancel alarm
am.cancel(action);
```

where alarm is get by `getSystemService()` method using global `ALARM_SERVICE` variable. New alarm can be set through `set()` or `setRepeating` for setting one-time or respectively repeating alarm. In function is addressed type of alarm, which is usually `AlarmManager.RTC` constant for real time clock alarm. Next parameter is time when the action, which is last parameter, will be performed. In `setRepeating` method there is parameter `interval`, which uses also constans of `AlarmManager` class like `AlarmManager.INTERVAL_DAY`. Cancel of alarm is performed through `action` as well. `Action` is `PendingActivity` where is defined which application component should be invoked when the time is gone.

## **2.2 Web technologies**

Web technologies are technologies that underpin popular service world wide web. This is most famous service of the internet, although there are also many others that are with it very closely linked. Namely, it is FTP service for sending files and e-mail protocol for sending electronic mails. This project however does not deal with this services. This project deals only with HTTP protocol, which is essential for web pages and modern technologies like cascade styles and PHP scripting language.

The basis of the web as we know it is web page. Web pages are text files written in Hyper-Text Markup Language (HTML) which internet browsers interpret and display to user. Nowadays, purpose of HTML is only structure text document from semantic side (World Wide Web Consortium, 2013). For describing the presentation semantics like formatting, positioning and overall appearance of webpage document are used Cascade style sheets or CSS, which almost all today's browsers support. To be complete, nowadays web pages often use dynamic content, which is carried out by two technologies. One is performed on client side, second is server technology. Client side is user computer or mobile device which wants to be served, and server is computer equipment which can work in parallel and handles client requests, whereas clients do not communicate with each other.

Representative of client side technology is JavaScript. As the name suggests, it is a scripting language for the browser. Thus, scripts are executed on the client side. JavaScript is suitable for simple dynamic actions or the browser which are connected with interpretation of results and viewing simple dynamic elements. JavaScript can dynamic change content of webpage and make calculations in response to mouse, keyboard or other various events. JavaScript should not produce any kind of data and should not execute massive calculations, because client side is slow and user wants primarily user-friendly interface and fast reactions of system.

For this purpose there is web server, which is designed to perform parallel task in order to allow simultaneously to answer more of parallel queries. To work with server hardware and generate difficult database queries, there is server technology such as PHP. PHP is scripting language, which generates HTML code according to parameters just on server, so it is safer, calculation is faster and saves lots of bits transmitted over the network. PHP simplifies access and communicating with database and can be used for general work on the server like creating files, running programs on server and so on. PHP has many functions and there are many frameworks for even easier work with it (Achour, 2013).

Separate category are database systems, which are widely used in web applications and will also be described in more detail in this category.

### **2.2.1 Physical layer**

Data are transferred over the network using landline broadband (over coaxial cable, fiber optic or copper wires), Wi-Fi, satellite or 3G/4G. Communication happens between webserver and client. Webserver is computer or set of computers which has access to web data mentioned before. Server has record of its IP address and domain name stored on Domain Name System (Mockapetris, p. 1987). Client can access web pages using URI, which is converted to IP address, necessary for real addressing through network (Network Working Group, 1981). Clients are all devices with connection to internet like smartphones, tablet and notebooks. Unlike client, server must be powerful system that can handle all client requests. Important features of web-hosting services are high availability, support of SQL database like MySQL, high bandwidth of FTP transfer, sufficient amount of disc space.

### **2.2.2 Software platform**

On server side there is a required software webserver installed on machine. This webserver represents network, transport and application layer. The most widely used web servers are like Apache, IIS or nginx (Netcraft, 2012). The primary function of this piece of software is respond to the request of clients using the Hypertext Transfer Protocol (Network Working Group, 1999). Many generic web servers also support server-side scripting using Active Server Pages (ASP), PHP, or other scripting languages. Client only need to have internet browser like Mozilla Firefox, Google Chrome or Internet Explorer installed on his operating system. After connecting into site, server and client start communicating using various protocols.

### **2.2.3 HTTP protocol**

HTTP is abbreviation for hypertext transfer protocol. It is stateless protocol used for transfer of hypertext documents written mainly in HTML language. HTTP is on application layer and uses network layer for transfer. Communication is between client and server. When user wants to load web page in browser, browser send HTTP request according rules of this protocol which are defined in relevant RFC document and server according this request respond again using protocol rules and returns required web page which browser can interpret and show to user.

HTTP protocol is therefore aimed for communication between server and client. Usually, loading one page leads to more HTTP requests. These requests are sent recursively. When client gets web page, it suddenly sends another according to content of this page to load content like images and nested frames. All objects which are components of web page have to be identifiable by URI, which is used by HTTP protocol for obtaining determined object (Network Working Group, 2005).

According to RFC document published by Network Working Group (1999) HTTP defines two types of messages: request and response. Request is message from client to server and contains URI of requested object. There are three main types of method of request: GET - to get content of requested object, POST - to get content of requested object, which is dynamic according to parameters posted in this request and HEAD - to get head of requested object. Newer version of protocol - HTTP/1.1 defines additional methods like PUT - to upload content of file on server, DELETE - to remove object or block of objects, and others like LINK, UNLINK as well as PATH.

If user wants to upload form data to server, there are two ways to do it with HTTP protocol. Through GET method or through POST method. POST method is directly intended for such intent and data are here transferred directly in the body of message. In GET method, there is used URI address, which has specific additional content starting with question mark and continuing with name-value pair separated by equal sign from each other and '&' - sign from other pairs. Final URI looks like this:

```
http://eu.mvanco/index.php/action/get_friends?id=54521&from_
letter=a.
```

Uploaded data are collected on the page through global variables '\$ \_GET' or '\$ \_POST' according to used method.

Response is message from server to client which contains response code defined in RFC as well.

The most common codes are:

- **200 OK** - request succeeded, request object is later in this message;
- **301 Moved Permanently** - requested object moved, new location specified later in this message after "Location:" string;

- **400 Bad Request** - request message not understood by server;
- **404 Not Found** - requested document not found on this server;
- **505 HTTP Version Not Supported** (Network Working Group, 1999).

As the HTTP is stateless, programmer must use other mechanisms for implementation of state functionality. There are two possible ways to do it. First option is to use cookies. Cookies are physically small files stored on client side by browser, but maintained on server side. There is also database on server side for maintenance. The values are always stored on the client side. When the server wants to identify and mark user, it send cookie head in HTTP request with other data. Client than reads head and according to protocol save the name and value of cookie to private storage on user computer with URL of Web page which causes storing this cookie and expiration time until which the value is valid. Every time, client want load intended page, it inserts cookie to head of request message in order server could identify the client.

Other option is keeping session through forwarding to any other page. It is achieved so, that every hyperlink on responded message is marked with identificator and after click on this hyperlink this identificator is send through previously mentioned GET or POST method. Identificator is obtained in several ways, all can be correct. But common aspect is that it is always some kind of output from hash function. Parameters of hash functions is usually string concatenated from nickname, password and current date and time. This is very old technique, but sometimes for simple special applications, it can be the right choice. This is the case for example if the user is a mobile application that generates HTTP requests only because of hidden communication with the server and uploading the files and does not expect an answer. In this case there is possible to generate session identificator as request of login of user and each other requests can be authorized through this session ID. Server has temporarily stored session ID of users and can verify, whether there is expected registered user.

#### **2.2.4 HTML and structure of web**

HTML or Hypertext Markup Language is type of SGML document. Unlike SGML (Standard Generalized Markup Language) in HTML, there is assigned semantics of hypertext document in Web environment to each mark. HTML language is defined by syntactic DTD definition

which can be found on site of World Wide Web Consortium (1991). HTML contains outside its own definition of markup language as application of SGML lots of other formal languages. According to the book of Shklar (2009), the largest are:

- script languages and mainly client JavaScript;
- cascade styles of CSS documents;
- less extensive like are coding (e.g. languages).

Body of HTML document has one `<HTML>` root element and his terminating equivalent `</HTML>` at the beginning respectively at the end of document. Inside this tag are head section bounded by `<HEAD>` tag and body section bounded by `<BODY>` tag. In head section there is invisible information about HTML document like title, description and keywords of document for search engines. Body document contains content of document. In body element there are block tags and inline tags. We can nest inline elements to block ones, but not vice versa. The most well-known block elements are `<p>` tags for paragraph on page and `<h1>`, `<h2>`, `<h3>`, . . . tags for various levels of headings. Well-known inline elements are `<b>`, `<i>`, `<u>` elements for highlighting text and tag `<a>` which is tag indicating hyperlink, which is essence of each HTML document.

Web is therefore a cluster of interconnected semantically-related documents, which creates the most extensive knowledge base in today's world.

### **2.2.5 PHP language**

PHP (Personal Home Page, later Personal Hypertext Preprocessor) has been developed as small toolkit for easier work with simple web applications, where is no need to use of sophisticated and rich application environments. It is tool for dynamic generated web pages according to content of database and other data. PHP has been massively expanded due to availability for all key platforms for free.

PHP is preprocessor, which process HTML document before sending to client. It replaces all the places bounded with `<?php`, `<?` and `?>` tags. In this place, there is often used retrieving data from the database or counting content of web page dynamically according to the given parameters.

## 2.2.6 Database system

Database system is server which stores structured data and allow to communication with it by SQL language. Communication is performed due to insertion, deletion or modification stored data as well as selection and acquisition. Data are stored in relations, where one field can link to other related. The base unit of data is data table. Each table contains columns - attributes and rows - entities. Each row is unique and can be identified and selected by primary key, which is always unique for one row. Relationships between tables are created by foreign key integrity constraint. This constraint ensures by database management system that in determined attribute will be only value from primary key of some other table. Through this relationship tables can be joined and can be detected answers to very complex queries.

SQL (Structured Query Language) is a special-purpose programming language designed for managing data held in a relational database system. SQL is divided into DDL - Data Definition Language for managing table and index structure and DML - Data Manipulation Language used to add, update and delete table data.

## 2.3 Pseudorandom number generator

### 2.3.1 Introduction

Real generators of random numbers are very computationally intensive. In most applications that do not require truly random values are used pseudogenerators therefore. All pseudogenerators work with the properties of prime number which represent multiplier and use the modulo operation in order to generated number has not been increased, but randomly varied in a given interval. Random number generators are parametrized by the distribution function. Distribution function defines probability density for values within given range. This section is benefiting from the book of Gentle (2004) named *Random Number Generation and Monte Carlo Methods*.

#### **Poisson distribution**

Poisson distribution is distribution which is natural for queueing management systems. Queueing systems are systems which contains operating part which serves clients named transactions.

Service takes time and transactions come randomly, which creates queues of waiting transactions in front of queue. This models are used for simulation of complex parallel systems which need to be optimized and operate more efficiently. Transaction arrivals in these schemes are most often implemented by poisson distribution. It defines how many customers enter the shop per unit of time. Generally it is number of any actions per time unit.

### **Uniform distribution**

Uniform distribution is basis distribution for pseudorandom generators. It is used when we need random generator in other distribution, because there are various method for conversion from uniform distribution to any other distribution which can be either defined by mathematical function or can be divided to parts, which can be explained by mathematical function. Many of pseudorandom generators use normalized version of uniform distribution. Normalized distribution is distribution function within interval  $(0, 1)$

### **Exponential distribution**

Exponential distribution is inversely proportional distribution to Poisson function. While the Poisson distribution function represents number of actions per time block, exponential distribution is time between two consecutive actions. Example of this distribution can be time intervals between failures.

### **Gauss distribution**

Gauss distribution uses randomness, which often occurs in nature. Distribution function looks like anthill shape in nature. Distribution has the highest probability in specific value and the more distant value, the smaller the probability is. Good example of this this distribution is probability of hit the dartboard.

## **2.3.2 Linear congruential generator**

The basis is a good generator of random numbers with previously mentioned uniform distribution. Requested distribution is acquired by transformation. Equation (2.1) shows the simple



formula for congruent generator.

$$x_{n+1} = (ax_n + b) \bmod m \quad (2.1)$$

where  $a$ ,  $b$  and  $m$  are constants and are intended to have suitable values. Constant  $b$  is some big prime number and represent multiplicative module and ensures random overflow over the scope. Value  $b$  is additive module, which is usually small number like 1 and is used for gradually varying of values. And value  $m$  is value for performing modulo operation and ensures the value fall into specific range. The example of simple random generator is shown in the sample code (2).

#### **Sample code 2.**

```
static unsigned ix = seed;

double Random(void) {
    ix = ix * 69069L + 1;
    return ix / ((double)ULONG_MAX + 1);
}
```

### **2.3.3 Transformation to other distributions**

Only a few applications require only uniform distribution. For all other cases, there is need to transform uniform distribution obtained from pseudorandom generator. In this project, there is a need to prohibit certain intervals of values from whole spectrum. This will lead to distribution of interval to several smaller which a different scale. Fortunately, there are more methods for such transformation, which are described in the following subsections.

#### **Inverse transformation**

This transformation has three steps. First step is obtaining distribution function and transforming into inverse function. Second step is generation random number with uniform distribution

and scope defined by domain of inverse function. Third step is obtaining function value at a given random point. The result of function is random number in new density distribution.

### **Excretory transformation**

Excretory transformation is the most intuitive method especially for sophisticated multidimensional generators. Principle can be well illustrated on two dimensional function which divides square into 2 parts. There are generated two random coordinates within this square with uniform distribution. Coordinate  $y$  determines if the value  $x$  is valid random number in defined distribution or not. If  $f(x)$  is higher than  $y$  value, the  $x$  value is not valid and the selection of random number is tried again with different random coordinates. This method is very computationally intensive, since there are generated two random numbers for each try and if there is small area of valid values, the transformation algorithm will have to repeat experiments very often.

### **Compositional transformation**

Principle of compositional transformation is fact, that we can divide complex distribution function into several more simple and for each interval can be used the most satisfactory and effective method.

# Chapter 3

## Software requirements specification and analysis

### 3.1 Brief description

The project contains of server and client side. The client side is an Android application, which runs on background the most of the time and once a day randomly alerts the user and allow him to take the picture with the phone camera during 5 minutes and add some actual thought. The server side is web server, where these photos and comments are gathered with coordinates of current location where the pictures have been captured. The users are able to view and comment the images of their friends. Photos are synchronized with server at the time the user have an Internet Access. Special emphasis will be placed on security issues, so it would not be possible to send pictures to the server in any other way than through determined Android application and at random unspecified time and guarantee that it will be possible only once per day for one user (Misra, 2013).

### 3.2 Technical requirements

The first part of the project includes critical evaluation of existing similar applications, according to which the project should be adapted. For the development of the application logic will be used Java programming language and Android API. There will be utilized application compo-

nents like Activity and Content Provider but mostly Service class for performing long-running operations in the background which does not provide a user interface.

The project should be developed by the Eclipse IDE, because it is the standard for the Android platform and is directly supported by Android. As the Android platform allows advanced programming using the Resources, for the UI design of each application screen and storing text strings will be used XML files, which are linked to the program by automatically generated java files. There should be widely use of repositories of Android open-source projects. Sharing code should be guaranteed by Git (for Android application) and SVN (for Web server) version control systems, specifically for easier revision of supervisor and more thorough backup.

On the server side, there should be used technologies for dynamic web pages like HTML, PHP and CSS as well as CodeIgniter framework for faster programming. For this purpose should be used development environment of Eclipse with Aptana Studio plugin which ensures FTP support and basic syntax highlighting. Communication over a network should be guaranteed through sockets and higher Android API and using HTTP protocol.

### **3.3 Progress of work**

First part of project will be analysis and design of technical solution of both client and server from upper view. There will be designed which technologies are intended to use and which platforms and developer tools will be prepared. The task of the project will be also finding a suitable free web hosting service that will provide storage space on the web server and where will be stored server part of the project. There will be compared existing solutions and evaluated in order to be applied for current problem. This will be followed by design of user interface both client screens and php server and designed specific appearance like used colors, graphic and others. Development and implementation on hardware will be followed by testing of functionality and the last part of the project will be evaluation of current developed client-server solution and treatment of critical mainly security bugs and shortcomings. The final phase of the project will be assessment and evaluation of possible further development and evaluation of knowledge acquired.

## **3.4 Client analysis**

As mentioned in the introduction, client is an Android application, which runs on background the most of the time and once a day randomly alerts the user and allow him to take the picture with the phone camera. This functionality does not serve to convince the larger group of users about the use. But it is based on the study objectives and their practical implementation in a particular solution. Implementation requires knowledge of randomness and its transformation, thrifty running on the background and also ability to use various external libraries that need to link to the program. In addition, there must be exploited default phone application like camera application, alarm manager or notification manager for user friendliness and ease of implementation. Further subsections describe the main identified parts in detail.

### **3.4.1 Run on background**

Whereas the big problem of today's mobile devices is battery life, application should be well designed and must be power safe. Since the application must alert the user randomly during the day, there must be a possibility to trigger the event in scheduled time. There should be used support of the Android system and its Alarm Manager as described in literature review. Program would not have to run the whole time and check whether the time is right to initiate action, this would be very power consuming.

### **3.4.2 Daily challenge**

The main aspect of the application is ability to once a day randomly alerts the user and allow him to take the picture with the camera. This is called Daily challenge for purposes of further explaining. Notifications should graphically display the progress and remaining time of daily challenges. The user should be able to refuse or accept the challenge. The taken picture should be stored in device for later upload to server. The camera functionality should be provided by default camera application implicitly installed on device.

Daily challenge would include adding comment on the picture and computing GPS coordinates according to current location. This service will be available only if device has GPS support and it is currently on. Pictures should be stored in a safe place where user can not accidentally

delete. On the other hand, it would be useful if they were accessible from the gallery as well. Comment and GPS location should be stored in private storage as well and application should remember which photos were and which were not yet uploaded.

### **3.4.3 Settings**

There has to be allowed to set sleeping time and unlimited number of presets when the user is not available. Presets would include setting of day of the week which the rule applies. The application should enable setting of vibration and setting of notification sound from default sounds of device.

## **3.5 Server analysis**

Server should allow the user registration and login. For each registered user should be held nickname, password, first name and surname as well as other additional information. After login, server should manage the correct photo upload. In addition, there should be possible to keep the relationships between users and allow the user to add new friend. The main function of the server should be browsing uploaded photos. Server should display the latest pictures of each friend with relevant date and time. There should be possible to browse older pictures of each friend as well and display detailed picture with comments. In detailed picture should be possible to also add own comments and determine location coordinates obtained from GPS service if available.

## **3.6 Comparison with existing applications**

The application is trying to differentiate itself from others. On the one hand, it is quite different. Primary unusual aspect of the application is to use pseudorandomness which moves it to game category. It is also different due to simplicity of use as user can immediately see the latest photos. Next unique feature is that the application does not require internet access while running like similar applications. It requires it only if the user wants to upload or view some uploaded

photos. This application also does not bother the user often while running on the background. It takes only 5 minutes per day.

On the other hand, there are many applications available on the Android market which are similar at least. During the research were found two similar applications. Their icons and screenshots are shown in the figure 3.1 . Each application was tested, found its functionality and found similarities and differences.

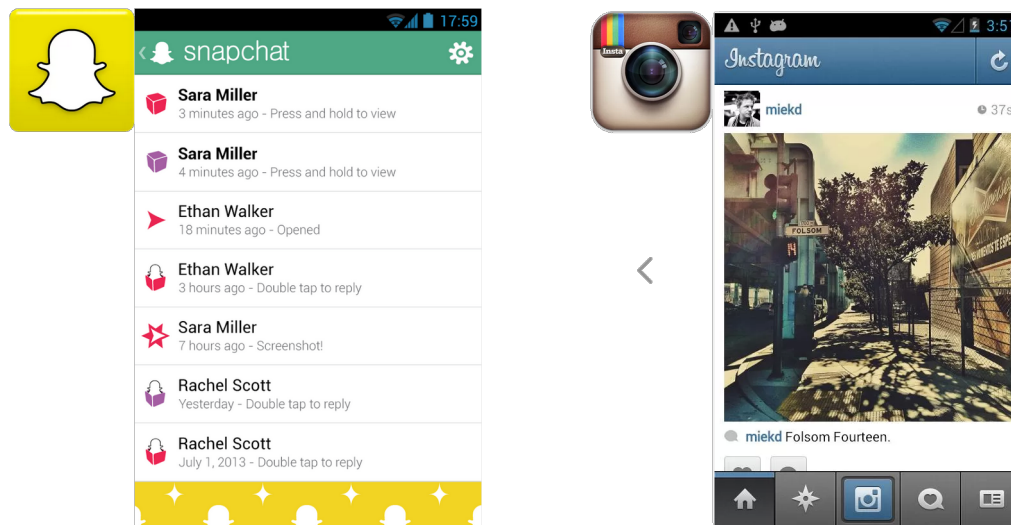


Figure 3.1: Icons and screenshots of applications Snapchat and Instagram (from left) downloaded from Android market (Google Play, 2013b) (Google Play, 2013a)

### 3.6.1 Snapchat

Snapchat is very popular social application for sending photos and short videos to friends. It is a multi-platform application that is also available for the iOS platform. In this application, there can be simply send photos or short videos by long hold of button and it can be sent even to multiple users. In addition, there may be drawn something on the photo. The main originality of this application is idea that photos or videos are viewed only for a while and then the snap disappears from the screen. If user want to save received picture, he has to take a snapshot as well and within the time limit, which is set by sender.

This is based on the philosophy, which has three pillars or motives:

- authentic moments with friends are not all about great opportunities.

- focus on close friends who easily put a smile on the face,
- there is value in the ephemeral (Google Play, 2013b).

Snapchat can also automatically add friends to the application from phone book. Application profits from simplicity and motivates users to socialization and sharing the moment. It is as simple as was this description and does not include anything extra. Therefore, there is no user manual, which would also be pointless.

At the moment, Snapchat allows also using for children which has limited functionality, does not require registration and does not send any pictures. It has only camera function with drawing. It is worth mentioning that with popularization of this application had come to Android market many clones.

## **Comparison**

Application works with pictures as well as application which is being developed. While in this project pictures are uploaded on server, in Snapchat there is only exchange between selected users. This allows to take more personal pictures and appeal to a wider group of users.

Since this is older commercial application, there has been introduced adaptations from feedbacks of users. Therefore, application is redesigned and offers an more attractive user interface. The application is focused on socialization and snapshots. Snapchat has not the purpose to take the most beautiful pictures, but rather be the funniest and most original. This is quite similar with Lila, but in Lila is certainly more time for taking pictures and picture is not too snapshot. On the other hand, Lila tries to be interesting because of uncertain location of the user as well as random time during the day. There may also arise situational pictures from this randomness. Both applications are creative, try to excel, and are focused on simplicity with having single purpose as well as focus on a narrow group of users. Snapchat is however more user friendly and is intended to impresses a wider group of users. Main reason can be better idea, more personal targeting and treated code for better performance. However, both applications share the similar philosophy and notice the uniqueness of ordinary moments of life. This is also the reason why this application is in this comparison.



### **3.6.2 Instagram**

Instagram is more famous application than Snapchat (Google Play, 2013a). It is for artistically sentient people. It provides a wonderful, artistic quality images. Prefers quality over the value of moment. This application is more complex and has more extensive functionality and also more users. The main feature of the application is uploading photos to server. This is not all, however. This application contains a sophisticated and intuitive photo editor. There are many automatic image adjustment to adjust the contrast, brightness and color and add texture and borders. Images can be rotated and there can be used a blur function and set of contrast. The list of features is great, but everything is designed for quick adjustment and not as a professional editor. For example, the contrast can not be set to a specific value, but only to turn into two states - on and off. But for the purpose of this application it is appropriate. Edited images can be shared by multiple social networks with one click. The application supports all famous social networks like Facebook, Twitter, Flickr and Foursquare. Compared with the application Snapchat, there can be edited also saved photos and the Instagram allows more camera settings as well. Very complex is viewing photos. All functions can not be listed, but a very interesting feature is display own photos on the map according to geographical coordinates assigned to captured photos. As in Lila, there may be added comments to each picture, in Instagram in addition, there is also possibility to label it as favourite.

#### **Comparison**

Instagram is in principle more similar to Lila application than Snapchat. It also works on the principle of uploaded photos. Instagram, because of its complexity, however, stands out. It is designed for a broad user base. And it may be used by almost every social network user like Facebook for easier photo uploads. Instagram is just a general tool and has no philosophical motives. Common feature of both applications is displaying uploaded photos with comments and their geographic location. The application provides a very sophisticated search of friends from phone book as well as from the social network Facebook. This feature in Lila application completely absent, making it quite a disadvantage. GPS location tracking is in the Instagram on the excellent level and the data are well utilized and displayed. In Lila, GPS coordinates are almost not used. It is only displayed without the addition of semantic values?? or showing on

the map.

### **3.6.3 Conclusion**

Comparison of the application did not end in disappointment. We should mention that previously mentioned applications are professional applications that exist for a long time on Android Market. They have the financial support and a number of people in their developers' team. Lila is different enough to take users and is a combination of compared applications. From Snapchat takes its philosophy and appreciates the ordinary moments of life, from Instagram takes the fundamental base of functionality like uploading pictures on server and commenting. On the other hand, Lila lags in support for automatic addition of friends from phone book and social networks, which is feature available on both compared applications. It also does not contain the option to edit captured photos. Nevertheless, application has a unique element of randomness and quite reasonable range of functions, when you consider that this is only a school project.

# Chapter 4

## Software design and development

During the development of application have been carried out a number of decisions about technical solutions and graphic design. The proposal is divided into client and server parts. In the client part is necessary to design the content of application screens and their interconnection. Fundamental problem is also the decision what will be the structure of application logic like. There is not addressed the implementation details, but only the general solution of design. The client application needs to solve saving options, the generation of random time by setting and using camera feature. There will also be designed solution for GPS support and many others. Server component of project needs to design application logic as well. A very important part of the server is the design of database tables and the principle of communication with the database as well as with clients.

### 4.1 Design of client application

#### 4.1.1 Application screens

##### Login Screen

Application screens were designed to be the most intuitive and natural to use as follows. Initial screen contains the title and logo of application to be clear what application was started. The first action that the user is required to do is to register. As this action is made only once, there is not necessary to always appear it on the display. However, the user must know what to do after

the first run. This was the reason why the registration button was placed on the home screen. The button does not cause discomfort to registered users and allows the user to quickly navigate into the registration screen. Registration screen will be a simple form with multiple text boxes. There will be fields to enter a nickname, password, first name and family name. Even, there were added additional field for entering birthplace and date of birth. Field for entering date of birth is the special box that allows the user to quickly select the day, month and year. It also prevents from entering invalid values unlike the conventional text field. The last elements are buttons to confirm respectively cancel registration. The user is about successful, failed, or cancelled registration informed through `Toast` messages. This is the a very nice Android user interface element for all sorts of text notifications. It is showed only for a limited time and therefore is suitable for these types of alerts. In addition to the registration button and graphic elements, there is designed login form on the main screen as this will be the most common action after starting the application. On this screen are also designed the setting options for settings stored on the client side. This options are present in the context menu as well to clarify that these are all possible settings.

### **User panel screen**

User panel is the main screen after login and provides an interface to almost all the necessary functions. The screen with the daily challenge is the second most important window. The screen shows the user name and it is therefore important feedback after login. This screen is used for communication between the user and the server. Since the application after the initial authentication is fully functional even without an internet connection, there must be possible to synchronize all the accepted and performed daily challenges with all outputs like comments and location coordinates on the server. For this purpose, there is prepared button to upload photos and its related data to the server. The screen also provides access to the personal wall as will be described in detail below.

Since the server stores all relationships of type follower-fellow, there was added the form to add a friend directly on this screen. Latest suggested button for this screen is a button to run application on the background. This event shows the home screen of the device and the user waits for a daily challenge.

## **Personal wall screens**

The wall provides a display of photos and its additional information relevant to the user. It is designed to be minimalist, simple and transparent. The user should first understand the meaning of the graphic elements and become familiar with their operation. Viewing daily challenges is designed and resolved on four levels for convenience. The first screen provides a general overview of the most recent photos for each friend. There is a great chance that the photos will be viewed in the evening, which means there should be located only random photos collected from the current day. Photos include a brown frame with a caption, time and date of collection and the name of the author. The very first contact on this screen is always logged user in order to be able to easily access and view own photos and their comments. The frame of this photo is highlighted and clearly distinguished by a different color.

When the user click on this frame there is displayed user screen. User screen represents the second level, which is already more specific and displays only photos of the determined user. There is retained the entire history of user photos taken each day. Photos are displayed chronologically from behind and therefore the user sees relevant recent photos first. They are displayed like the main wall with the caption as well as the time and date of collecting.

Another designed screen is the screen of daily challenge. The screen is triggered by clicking on the image itself as in the main screen so in the screen of the particular user. This screen displays the captured image and all the information as in the previous cases. However, in addition, it displays GPS coordinates and user comments. Comments are positioned just below the caption and are immediate followed by a form to add a new comment. Since it is on this screen many elements, it is possible to view full picture preview. For this intent is designed the last screen. The last screen contains only magnified image with all the particulars, allows zooming and horizontal and vertical scrolling.

So far, there were mentioned only the transitions to a lower more specified level. The transition to a higher, more general level is designed by back button. This hardware button, which is present on each phone, causes the loading screen, which may be one or even two levels above.

## **Daily challenge screen**

To view the daily challenges, Android notification element is used (Notifications). Notification in Android allows sound alert, showing the description of the event and allows the user to react to this event. The daily challenge duration is established on 5 minutes. During those minutes are designed three audio alerts: first, at the beginning, and the next two every minute and half. The notification includes progress bar that shows the time course and is supplemented with the information about the remaining time. Another highlights are two buttons for acceptance, respectively dismissal of the challenge. After clicking on the rejection button, notification is cancelled. In the second case, there is started the screen to capture the image. In this situation, there is displayed the camera's default application, which is followed by the screen with captured image preview. Image preview also contains two buttons for acceptance, respectively dismissal of captured picture. The last screen is shown after acceptance of image and contains as the image preview so the text field where is the place for entering the picture caption, description or actual thought about the situation that picture shows. On this screen, there is displayed the actual address and the coordinates of the current location as is designated by GPS receiver. After confirmation of the screen is the daily challenge stored in the phone memory.

## **Design for testing purposes**

Application will be prior to publication includes special tools for testing and debugging. This functionality will be hindered subsequently. This refers to the login screen as described before. There is the added action invoked by clicking on the application icon. Action displays a Toast message that reveals the scheduled time for planned daily challenge. In addition, there is also added action invoked by clicking on the application label under icon which calls the simulated background service with the task to generate the random time according to current settings. This background service is described in more detail in section *Application logic*, since it does not concern the user interface. This does not affect the actual challenge planning, but serves to store this time in order to be verified the correct operation of the event scheduler. The stored value is displayed along with the actual scheduled time by clicking on the application icon.

### 4.1.2 Application logic

Design of application logic is carried out on the basis of components like `Activity`, `Content Provider` and mainly `Service`, as well as screen design which is described in the previous section. For the development was chosen Eclipse environment with ADT (Android Development Tools), that is very flexible and can be usefully extended.

The first issue is the application settings issue. Android provides many options to save the configuration, which was in sum described in the second chapter 2 in *Content Provider* section, and it is necessary to perform a number of decisions. Another bearing component of the application is algorithm to generate the random time according to saved preferences. During the project, there had to be devised a method and format for storing images and their relevant data from daily challenges as well.

#### Settings

Application settings are resolved by easiest way for storing preferences. It is usage of Android embedded mechanism for storing user settings, which is accessible through `Preference` class. This mechanism includes also user interface. All application logic connected with settings is grouped in `eu.mvanco.lila.settings` package. The main screen is extended from `PreferenceActivity` class and uses XML file for loading settings headers. There are headers for *Sleeping time*, *Presets* as well as *Sounds*. Each header has the relevant `PreferenceFragment` assigned, which is invoked after click on relevant option. This activity uses his own list adapter `MyPrefsHeaderAdapter` extended from `ArrayAdapter<Header>`, which displays the switch button near the *Presets* header. This functionality has been largely reproduced from the existing open source project and allows to disable all configured presets by one-click. Another problem is the storing of time. Preferences implicitly support only the saving of simple values such as strings or booleans. In this case, a special preference extended from `DialogPreference` is developed and there is returned `TimePicker` in `onCreateDialogView` to display graphic component for time picking. Since there must be made the large changes to the underlying implementation classes, there must be carried out storing data as well. This is performed through `SharedPreferences` class, which is the easiest method for storing simple unstructured data.

The challenge is to allow setting unlimited number of presets. This functionality is not present in normal situations, and so the whole class `Preferences` had to be rewritten. To store presets is created variable `listOfPresets`, which is managed by `SharedPreferences` class and contains names of added presets. Since each preset allows the user to specify the time interval as well as the chosen days of the week, there is created new variable with the name of determined preset and the content specified by the proposed format `'hh:mm;hh:mm;wwwwwww'`, where 'h' is for hour, 'm' is for minute and 'w' for value 1 or 0. Semicolon divides the string into 3 parts: *from time*, *until time* and *day in week*. The first two part are quite obvious. The third part is string composed from 7 characters which represents days in weeks. The first character represents Monday, the second Thursday,...,the last Sunday. When the value is 1, the preset is active in this day, otherwise, the preset is inactive. Showing the following presets in `PreferenceFragment` is possible via creating a dynamic preferences. When the screen with presets is viewed, all the preferences are dynamically generated from the `SharedPreferences` class. The sound settings use default appearance and contains two boolean preferences for enabling/disabling sounds and vibrations. There is also preference for picking the sound which is ensured by `RingtonePreference`.

### **Pseudorandom generator for time calculation**

Pseudorandom generator is the essential of this application. To solve the problem, there must be addressed time of treshold of new day. For practical purposes it would be the best to choose a time when the value of distribution function of number of sleeping people is the maximum. It would be time between 2:00AM and 3:00AM. However, for clarity of the implementation is determined time 12:00AM. Application will invoke event handler a minute ago the treshold of new day and plan and set the alarm for new day according to the settings of user.

Midnight event handler is `BgService` class and is called by Alarm Manger as described in literature review. For this purpose is used repeated alarm. Handler extends the `Service` class and provides a calculation of the random time as well as setting the relevant notifications by Alarm Manager.

The first step is generation of random number in interval  $(0;86400)$ , where the second number is number of seconds in one day. This number is converted to corresponding hours and minutes



of day.

The second step is comparing this time with sleeping time and additional presets which is ensured by `isInSleepingTime()` and `isInQuietTime()` functions. To obtain the presets is created `Preset` class and `getPresets()` method for getting presets to a variable of type `List<Preset>` from `SharedPreferences` class.

If the random time is not within any preset or sleeping time, the time is used as valid. If there is at least one interference with the presets or sleeping time, there is generated new random time and the whole process is repeated.

### **Daily challenge**

There have been proposed saving setting and planning the daily challenge. Another activity is the capture and storage of daily challenges. Daily challenge is implemented by the `DailyChallengeReceiver` class, which is simple `Broadcast Receiver` for showing notifications with `Notification Manager`.

By clicking on notification is triggered activity `PhotoIntentActivity`. In this activity is used intent created by the following command.

```
Intent takePictureIntent =  
new Intent(MediaStore.ACTION_IMAGE_CAPTURE);
```

to obtain bitmap picture, which is returned in `onActivityResult()` function. After returning the picture is called reproduced activity to obtain GPS coordinates and even current address. This activity is rewritten to be used with `startActivityForResult()` function. Function sets the result of calculation using the `setResult()` method. This is a blocking operation, which is not too reasonable solution, but serve to demonstrate the principles of GPS service. Activity is set to invisible, but for the moment, it is still blocking the user interface of `PhotoIntent` activity.

Since the daily challenge can be triggered at a time when no internet connection is present, application stores the daily challenge data in the phone memory into the private file open for append. This application stores images and additional information separately. While the images are stored as expected in a specified folder, additional information is stored as text string with the proposed structure. The structure is following:

```
path_to_stored_image.png{comment{
received_latitude{received_longitude{address;.
```

The sections are split by a '{' sign and contains in the following order: path to stored image, comment, received latitude, received longitude and address. Each entry is terminated with ';' sign. This file serves as commit file, because it contains all the files which are waiting for upload. If the file is uploaded, it is no longer necessary and therefore deleted from the file.

### **Sending requests to server**

Sending HTTP requests is served via `HttpClient` class. To carry out the competent command, there is called `execute()` method with the request inserted as parameter. Response is returned and stored into `HttpResponse` class as shows the following sample code (3).

#### **Sample code 3.**

```
HttpClient httpClient = new DefaultHttpClient();
HttpResponse response = httpClient.execute(request);
```

GET request is wrapped in the `HttpGet` class and POST request is wrapped in the `HttpPost` class (sample code 4).

#### **Sample code 4.**

```
HttpGet getRequest = new HttpGet(url.toString());
HttpPost postRequest = new HttpPost(url.toString());
```

While the GET request has its own parameters directly in the URL string, POST request must include its parameters in the body of request through `setEntity()` method (sample code 5).

#### **Sample code 5.**

```
List<BasicNameValuePair> nameValuePairs =
    new ArrayList<BasicNameValuePair>();

nameValuePairs.add(
    new BasicNameValuePair("Picture", base64EncodedPictureString));
```

```

nameValuePairs.add(
    new BasicNameValuePair("Description", description));

postRequest.setEntity(new UrlEncodedFormEntity(nameValuePairs));

```

The HTTP response contains the body of the HTML document and the status code. The status codes is available through `getStatusCode()` method. The body of the HTML document is read through input stream obtained by `getContent()` method of HTTP entity wrapped in the `HttpEntity` class. This is shown in the sample code (6).

### **Sample code 6.**

```

int statusCode = response.getStatusLine().getStatusCode();
InputStream inputStream = response.getEntity().getContent();
Writer writer = new StringWriter();
char[] buffer = new char[1024];
try {
    Reader reader =
        new BufferedReader(
            new InputStreamReader(inputStream, "UTF-8"), 1024);
    int n;
    while ((n = reader.read(buffer)) != -1) {
        writer.write(buffer, 0, n);
    }
} finally {
    inputStream.close();
}
String htmlBodyString = writer.toString();

```

### **Personal wall**

View of personal wall is addressed through composite graphic element called `WebView`. `WebView` is considered as a small built-in browser without software control buttons. The most

important function is to display a specific web page using `loadUrl()` method (sample code 7).

#### **Sample code 7.**

```
WebView browser = (WebView) findViewById(R.id.web_view);  
browser.loadUrl(url.toString());
```

## **4.2 Server application and communication**

The server part of the project deals with the technologies for dynamic web pages like HTML, PHP and CSS. However, due to the scope of the project, there was chosen the solution using CodeIgniter framework. CodeIgniter framework simplifies the work with basic PHP and facilitates the work with the database queries. This framework also allows the clear interconnection of the web content using the MVC (Model-view-controller) architecture, which was the main reason for this design decision.

The next step is the proposal of the server user interface and the general appearance using cascading style sheets (CSS) for describing the presentation and formatting of document. These decisions about screen layout and architecture are located in chapter *Design of client application* because they are part of the client application from the user's perspective. However, to address them, there are used technologies for dynamic web pages (HTML, PHP and CSS), though it is used only in the specific client application and is not available via a common web browser.

Finally, there is performed database schema design and design of communication with client, which is described in more detail in subsequent sections.

### **4.2.1 Database schema**

Database application stores information about the user and his daily challenges according to software requirements specification. Database design is modeled via the following entity-relationship diagram (ERD) in the figure 4.1.

Each user enters the first name, family name, nickname, password, birthplace and date of birth during the registration. For this purpose, there is used the `USER` table. When registering, the user creates a new row in this table.

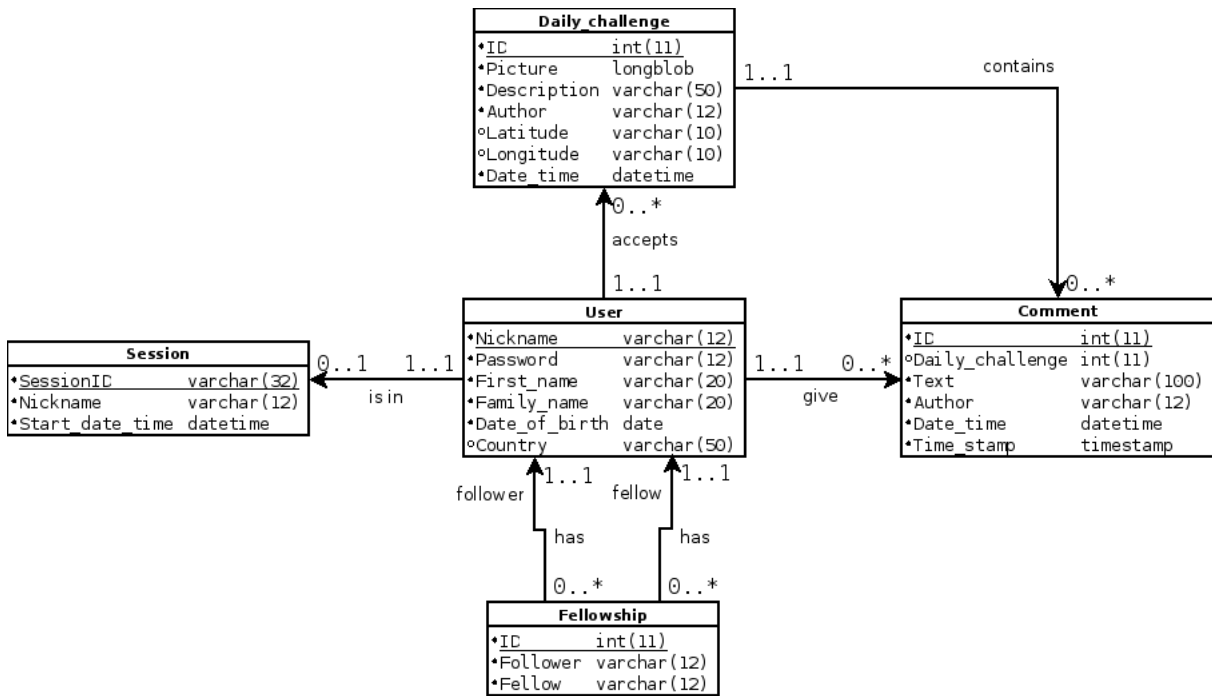


Figure 4.1: Database schema modeled using the UML class diagram

The next proposed table is the `Session` table. This table contains hash keys for the login. After the first installation of client application, startup and user login, there is generated random 32 characters long string, which is inserted into this table. The user must be authorized by this key in order to access all the other functions of server. For the further purposes is accompanied username and key stored with the date of the first logon.

Each user has a list of friends stored in the table `Fellowship`. In this table, there is a pair Follower-Fellow to model the relationship N:N.

To store the user content on the server like daily challenges and comments are intended the table `Daily_challenge` and `Comment`. As the name implies, the first table is used for storing picture, its description, latitude, longitude and date of collection. To be daily challenges well classifiable, there is information about the nickname of the author. The second table, `Comment`, is used for storing comments of daily challenge. There are columns like `Text` and `Time_stamp` for storing the content of comment and its timestamp. Additional information here is the information about the nickname of the author as well as the information about ID of the daily challenge connected with this comment.

## 4.2.2 Server communication

The communication client-server is performed through the HTTP protocol. The client sends queries through a simple GET or POST requests. There are used the GET requests in the most cases. The POST request is used only due to the posting large data such as when uploading images.

The whole form of communication is thus dependent on the correct URI. Depending on the determined CodeIgniter framework, there is chosen following URI structure:

```
http://mvanco.eu/index.php/action/<function_name>/<parameter>,
```

or alternative for functions with two parameters:

```
http://mvanco.eu/index.php/action/<function_name>/<first_  
parameter>/<second_parameter>.
```

The communication towards the client is performed through status codes described in the literature review. Where is the need to return a specific value, the value is returned as the content of an HTML document. Applied status codes are 201 (Created) and 203 (Non-Authoritative Information). The code 201 is used in the case there is added or removed item, or made any other change in the database. This refers to the successful registration, login, upload of daily challenge and others. The 203 code is used in the case there cannot be added, removed or changed the database row and thus the registration, login or other actions are not completed. Returning a specific value is used when logging as well as finding information about the registered user.

## 4.2.3 Server structure

Depending on the CodeIgniter framework, the structure of the server is divided into the functional logic placed in `application/controllers/` folder and model part, that is designed to work with information in the database, stored in `application/models/` folder. The `application/views/` folder contains the HTML code associated with any outcome of the HTML document. In this folder, there is header with the inclusion of the style definition and footer for the proper completion of html structure. There is also an HTML document for each level of personal wall as mentioned before. For this purpose are created files with names `wall.php` or `blank_wall.php` for personal wall, `daily_challenge_history.php`

for user screen, `daily_challenge.php` for specific daily challenge view as well as `show_picture.php` for detailed image preview.

There is created as model part so controller part for each server function. The model part is placed in the `/application/models/register_model.php` file and according to the CodeIgniter framework, it is nested in the `Register_model` class which extends the `CI_Model` class. The controller part of server is located in the `/controllers/action.php` file using the `Action` class which extends from the prepared `CI_Controller` class.

The first functionality of server is the registration, which is performed by the `sign_up($nickname, $password, ...)` function. For this purpose, there is used the insertion query into the `User` table. The `login($nickname, $password)` function inserts data into `Session` table. The session ID is obtained by the following command (sample code 8).

#### **Sample code 8.**

```
$sessionID =  
    md5( $nickname.$password.date('m/d/Y h:i:s a', time()) );
```

This command concatenates the nickname, password and actual date to one string and uses the `md5()` hash function for generating the 32 character long random string. This string is stored on the client side as well and is used for the authorization of all the other functions for the manipulation with the user account.

The function `getUser($sessionID)` is used in almost every subsequent function for the validation of user as well as detection and obtaining of nickname. These manipulation functions are clearly described in the following tables 4.1 and 4.2.

In the table 4.1, '<POST parameters>' are parameters send through POST request. Parameters are specifically `$_POST['Picture']`, `$_POST['Description']`, `$_POST['session_ident']`, `$_POST['Latitude']`, `$_POST['Longitude']` and `$_POST['Date_time']`.

Function name	Parameters	Description
logout()	\$sessionID	Function will delete specific row from <code>Session</code> table
add_friend()	\$sessionID, \$fellow	Function will add a new row to <code>Fellowship</code> table
get_friends()	\$sessionID	Function will get all follower friends from <code>Fellowship</code> table
uploadDailyChallenge()	<POST parameters>	Function works with POST request. According to POST parameters, there is added a new row to <code>Daily_challenge</code> table

Table 4.1: Server functions, part 1

## 4.3 Implementation

In this chapter, there are excluded all the solutions and decisions relating to design. The first section describes the environment settings. This applies to the installation, configuration, and decisions about choosing a particular development platform. The second section concerns the backup, software versioning, revision control and the collaboration during the project development. The next section refers to the administration of webhosting and the last section describes briefly the progress of the project implementation.

### 4.3.1 Preparation of the IDE

The first step of implementation or the prerequisite before a major implementation is the preparation of the development environment. In this respect, there must be performed a number of decisions as well. The first decision concerns the choice of operating system. Android supports development tools for both Linux and for windows platform. In this project is used the Linux distribution, because it is stable, operates more transparent whereas the Android uses the Linux kernel as well. Linux provides also comfortable work with the console to run utilities and console programs like `android` or `adb` from the *Android SDK* package. The selected Linux distributions, Fedora, is a Free and open source operating system, which provides the latest long-term stable release of the Linux kernel and a number of new software available directly in the *Software Management* utility. Updates are installed every day and the system is remarkably



Function name	Parameters	Description
add_comment()	\$sessionID, \$dailyChallenge, \$text	Function add a new comment to Comment table
get_comments()	\$dailyChallenge	Function gets comments for specific daily challenge from Comment table
get_wall_daily_challenges()	\$author	Function gets all daily challenges for specific logged user from Daily_challenge table
get_user_daily_challenges()	\$author	Function gets all daily challenges of specific author from Daily_challenge table
getDailyChallenge()	\$dailyChallenge	Function gets selected daily challenge from Daily_challenge table
show_wall()	\$sessionID	Function shows user wall according to get_wall_daily_challenges() function
show_user()	\$sessionID, \$user	Function shows daily challenge history for selected user according to get_user_daily_challenges() function
show_daily_challenge()	\$sessionID, \$dailyChallenge	Function shows selected daily challenge according to getDailyChallenge() function
show_picture()	\$sessionID, \$dailyChallenge	Function shows picture from selected daily challenge

Table 4.2: Server functions, part 2

stable and fast. For easier and more user-friendly work was selected the latest (KDE 4.3)*Plasma Desktop* graphical environment from the *KDE* community.

The next step is to install and configure Eclipse development environment. The chosen platform version is the latest 4 . 3 version with code name *Kepler*. This environment is enhanced by the custom plugin, called Android Development Tools (ADT) offered by Android. This allows automatic administration of Android project, creating the project structure as was described in the previous chapter as well as the compilation, debugging and the communication with the real connected hardware device. Android Development Tools requires to download the Software Development Kit (SDK) and set the correct path in Eclipse. For proper operation, it is necessary to also add it to the Linux PATH variable using set or export command. This makes the SDK commands available from any working directory as well. Another essential issue is to

download the latest Android API through the *Android SDK Manager*, which is available after installing ADT Plugin. In this project, there is downloaded Android 4.2.2 application interface. Since there is used also the command line for the communication and administration, there is installed ELT plug-in as well, which integrates command-line terminal into the Eclipse environment. This is possible only due to a suitable choice of the operating system, whereas the Windows operating system is not supported.

For proper functionality of the Eclipse environment must be installed Java Development Kit (JDK) for Java developers. *Android SDK* requires 32-bit libraries for 64-bit operating system and the USB drivers for testing hardware. Since some Android tools are only available for virtual machines, there is a need to create a new appropriate Android Virtual Device (AVD).

For testing purposes, there is prepared the mobile Android device LG Optimus L9 (P760) with the Android version 4.1.2 Jelly Bean (API level 16). This smartphone has a 4.7 inch screen with the resolution of the 540 x 960 pixels. The device has a 5 megapixel camera, supports GPS receiver and provides access to the Internet, which is characteristic of each Android device. This device, therefore, supports all the features required to run the application.

### **4.3.2 Collaboration**

Backup, software versioning, revision control and the collaboration during the project were carried out through a comprehensive Git system. Demonstration of Git history is shown in the appendix D. There is exactly reproduced content from the server, so it is possible that it contains grammatical errors. Another important note is, that it is not complete as well, because the Git repository has been reset several times. As the Eclipse has no implicit support for this distributed version control and source code management system, there is installed *Eclipse EGit* plugin, which is an Eclipse Team provider based on JGit, a pure Java implementation of the Git version control system. This GUI Git client collaborates with the *Google Code* server, which stores all versions of a project.

There are also used other source codes of open-source projects available via Git version system. These codes are used either directly, or just for inspiration.

### 4.3.3 Administration of Webhosting

The first step in the deployment is the purchase of `mvanco.eu` domain name. A domain name is purchased and held by *WebSupport.sk* company. This was followed by purchasing web hosting, which is provided by the *WEDOS Hosting* company. There were subsequently set name servers on the purchased domain to `ns.wedos.net`, `ns.wedos.eu`, `ns.wedos.cz`, `ns.wedos.com` in order to the domain is linked to the web hosting. The next step was to create a MySQL database and obtain credentials to access it. Tables were created under the previous proposal and set credentials in the client application. At first, there were uploaded the source codes of CodeIgniter framework, then it was followed by other created files under the previous proposal. Files are uploaded using the FTP protocol. As the Eclipse has no implicit support for the FTP file management, there is installed *Aptana Studio* plugin, which contains reliable FTP client.

### 4.3.4 Progress of work

After setting the Eclipse IDE and the webhosting, there is created the Android project. The project is configured to use the latest Jelly Bean version of Android (API level 16-18), because it is supported by at least one third of existing devices at present(). Minimum supported SDK version is set to API 16, since the project is not trying to target as many devices as possible but prefer to use the latest principles of Android platform. The `android:targetSdkVersion` attribute is set to API 17, since it was the latest API version at that time. Project is therefore fully compatible with the API 16-17 versions. The API level 17 is used for the compilation, while on the testing hardware device is installed Android SDK version 16. Backward compatibility ensures, that the application will be able to run on the older versions of the Android SDK as well until `android:minSdkVersion`. From the `android:minSdkVersion` until the `targetSdkVersion`, application is able to run without enabling any compatibility behaviour to maintain the app's forward-compatibility. Higher API versions may be running in the compatibility mode, especially when the major changes are proposed in later Android versions.

The first step in the actual development is to create activities that represent the application

screens. For this purpose were created activities showed in the following table.

<b>Activity</b>	<b>Description</b>
<code>MainActivity.java</code>	The activity associated with the logon screen
<code>RegistrationActivity.java</code>	The activity for registration
<code>UserPanelActivity.java</code>	The activity for showing the user panel
<code>WallActivity.java</code>	The activity for showing the personal wall
<code>camera.PhotoIntentActivity.java</code>	Activity for taking a picture and showing preview
<code>location.MainActivity.java</code>	The hidden activity to obtain GPS coordinates and address
<code>settings.MyPrefsActivity.java</code>	The activity for showing the settings extended from <code>PreferenceActivity</code>

Table 4.3: Implemented activities

In addition to the activities were created other auxiliary classes which extend from classes like `PreferenceFragment` for the setting options, `ArrayAdapter<Header>` for editing of the base `PreferenceActivity` class and `DialogPreference` for creating new type of preference item and many others.

On the other hand, there are developed other types of application components as well. There is service component `BgService`, which ensures planning a random time at midnight on the background. The next application component is `Broadcast Receiver` represented by `DailyChallengeReceiver` class. This class is used to display the notification messages about new daily challenge.

All of these classes are neatly categorized into 4 packages: `eu.mvanco.lila` for the main functionality, `eu.mvanco.lila.settings` for classes associated with the settings, `eu.mvanco.lila.location` for the activity to obtain current GPS coordinates and address and `eu.mvanco.lila.camera` for the camera-related classes.

Since in the project were also used external source codes, there were linked private libraries *The Google Play services client library* and *Support Library*. Both libraries are needed for proper operation of activity to obtain current GPS coordinates and postal address.

For each activity was designed graphical user interface through XML resource file as well as logic based on the previous proposal. Activities and their proper functioning were verified

independently. Only after was verified the correct functioning of individual activity, activity could be linked to the entire application, which was also tested. Testing was carried out on the hardware device as well as virtual device in the special cases. When testing on the hardware device, there were due to simplicity of application submitted almost all kinds of possible input. This testing was, therefore, the simplest and most effective testing, as it was the testing for the proper functioning of the graphical user interface in the most cases. For this reason, the testing was not performed by testing scripts, as is the case of console applications.

Further details are included in project logbook in the appendix C. The logbook shows the implementation on the timeline. There are formed two week periods, in which there is summarized the work that was carried out during them. There is seen what actions were performed in which order.

# Chapter 5

## Conclusions

In conclusion, this work has met all the specified objectives. There has been carried out the extensive research about the Web technologies and Android platform. The main findings of this research are listed in the literature review. In addition, there has been carried out also research on the pseudo-random number generator and related topics in the field of randomization. The information has been investigated in terms of finding the most efficient design and the most sensible implementation for actual problem. For this purpose, the existing similar application were compared to the application that was being developed. Based on this research, several solutions have been designed and many decisions about design have been made. Mainly, it was the proposal of graphical user interface and the most suitable technical solution for both parts of the project. The outcome of the whole process was the application, which was properly implemented and deployed on the web hosting as well as on the testing Android device. The application is fully functional and meets all software requirements, such as those specified in the *Software requirements specification and analysis* chapter. The last objective of this project is briefly summarized in this chapter. It involves the assessment of the project functionality in terms of technologies used and proposal of the further development on the basis of learned knowledge.

## 5.1 Contribution and benefit of the project

The contribution of this work is mainly acquired knowledge about Android platform and Web technologies. This work was designed to merge a number of technologies into a single project and therefore provides an interesting knowledge base. The application uses different hardware, such as camera and GPS receiver and proposes a specific type of communication between server and client. Communication is based on a simple sending of HTTP requests, which is considered to be an appropriate solution for the scope of the project. The work benefits from its simplicity and does not try to find the most complex and comprehensive solution. Therefore it is suitable for the study of Android platform and their basic principles. Developed project can be used to inspire aspiring developers as well. Thanks to the advanced features of the Android platform, also whole application components can be used in other projects. Example of a very exploitable part is an algorithm for generating a random time during the day. In the application are enhanced capabilities of `PreferenceActivity` class and created new dialog preferences extended from `DialogPreference` class. These are pieces of code useful for other projects.

During the project realization, there has been acquired knowledge about the principles of the Internet, Web service, HTTP protocol, structure of the HTML file, work with the MVC PHP Framework. In the Android platform, there are exploited application components `Activity`, `Service` and `Broadcast Receiver`. There are explored options for storing settings, communication with the Web server and viewing the web content directly as part of the Android application. The application also allows the use of system alarm and location services.

The importance of the work also lies in identifying security risks in the common Android applications and the assessment and comparison of similar applications. The contribution is also in the Android application itself, that contributes to a broad family of Android applications available on the Google Play. Although, the Android store contains plenty of applications, this project is specific with his unique element of randomness and quite reasonable range of functions while preserving the simplicity.

## 5.2 Assessment of functionality and security risks

Android provides a mechanism for declaring permissions that the application uses. This is ensured by inserting the corresponding tag `<uses-permission android:name="android.permission.NAME_OF_PERMISSION"/>` into the `<manifest>` tag in the `AndroidManifest.xml` file.

In this project, there are used permissions `WRITE_EXTERNAL_STORAGE`, `VIBRATE`, `ACCESS_FINE_LOCATION`, `INTERNET` and `READ_EXTERNAL_STORAGE`. These permissions identify categories of possible misuse. The main asset in this environment is the operating system itself. Linux as well as Android operating system, however, can be divided into several blocks. It is file management, process management and memory management. Since the Android is arranged to run on portable devices, there is important also power management. As assets can be considered the system memory, source of electrical energy, CPU processing time and data, which may be stored in the form of file (since in Linux almost everything is considered as a file). This paragraph and next sub-chapters are benefiting from book of Six (2011), book of Misra (2013) as well as the conference paper of Stavrou (2013).

### 5.2.1 Vulnerabilities towards the system

Vulnerabilities and the related security risks that are based on previous assets are the lack of system or physical memory, discharge of the battery as well as CPU overload and disabling other work on the phone. This is considered as a security risk because it interferes with the defined assets undesirable. It prevents the work with a mobile device, damages or changes private and critical data.

Considering the first case involving the lack of memory, Android has a pretty secure memory management. This ensures that the data of each application are separated from each other. The application, however, uses the `WRITE_EXTERNAL_STORAGE` and `READ_EXTERNAL_STORAGE` permission. External storage can be, thus, filled with unnecessary data or junk files and block access to memory for other applications. Although, there is allowed to write external storage, each file on the storage medium has assigned the access rights and, therefore, files cannot be deleted or modified without permission (Yang, 2012).



The application can also consume too much system memory resulting in the destruction of stopped applications running in the background. The system memory is massive consumed when the application loads large files, especially images, and processes them.

Next item on the battery consumption is perhaps the most important in modern Android applications, because there is a high probability that the application is spending more battery than it should be (Paul, 2010) (Paul, 2011). There are also set permissions like `VIBRATE` and `ACCESS_FINE_LOCATION` for access to GPS receiver. This additional functionality can be very energy demanding, especially if it is incorrectly used.

Last issue concerning the CPU overload can be caused by overload of the main UI thread. This may cause that the application is not responding and the user interface is blocked and unresponsive. This problem should be solved by proper use of threads.

### **5.2.2 Vulnerabilities towards the application**

Other security risks are related to assets such as proper and safe operation of the application according to software requirements specifications. This includes registration, login, secure access to the server and for example ensuring that there will be only one daily challenge every day for each user.

The first security problem is that the application allows the user login on two different mobile devices. This can be misused and can be performed two daily challenges in one day, which contradicts the rules of the game. To treat this bug, there should be prevented the logging of already logged user (the user is in the `Session` table).

The major security risk is insufficiently secured access to the server. Since the communication with the server uses only simple GET and POST HTTP request with specific URI, the packet can be easily captured and used to unauthorized interference with the server. Such intervention would be great risk with high impact, whereas in the case of decoding the communication, there could be carried out any action with the server (Wei et al., 2012). There could be uploaded unauthorized data and display sensitive data using any common web browser.

### 5.3 Review of further work

The application is, in the relation to this project, complete. However, in order to be deployed in a real environment, there is required a lot of additional work. These objectives were identified, particularly during the comparison of existing applications. The main objective for the further development is support of the social networks. The application lags in support for automatic addition of friends from these networks as well as from the phone book. Other lack is the impossibility of the confirmation and blocking of friends on the other hand of friendship.

The biggest deficiency is, however, the lack of security towards the application as well as the system itself as was mentioned in the previous section. For better marketing and presentation of the applications, there is necessary to improve and simplify the graphical user interface even more and to debug all the bugs in the program that might occur.

Furthermore, this text discusses the possible extension of functionality towards attracting more users. Based on the existing applications, application could be extended to show the map as it is in the case of *Instagram*. The application could show all the locations on the map, where were made all daily challenges of the user. The more sophisticated applications also add support for the making video and audio recordings. This can be extended by using speech recognition for adding text comments. Support for emoticons in the comments is a necessity.

Users should be more motivated for doing photos. Therefore, there should be established scoring system. Users should be able to rate and photos simply select a photo as a favorite. Based on this assessment could be calculated several statistics. The user could collect points for good photos and could be rewarded for it. There could be reflected responsible approach to performing the daily challenges and longer loyalty. Application should also extend the user profile and allow adding more information about users. Another interesting extension is the addition of topic to every day. There may be declared the color or situation, and the pictures should be related to it. This idea is inspired by series on the [theguardian.com](http://theguardian.com) site of British newspaper named *In pictures*.

The last exotic functionality could be statistics about lifestyle. This application could promote the declaration of own challenges and there could also be tracked jogging trails and evaluated the jogging statistics.

# References

first

# Appendix A

## Application screens

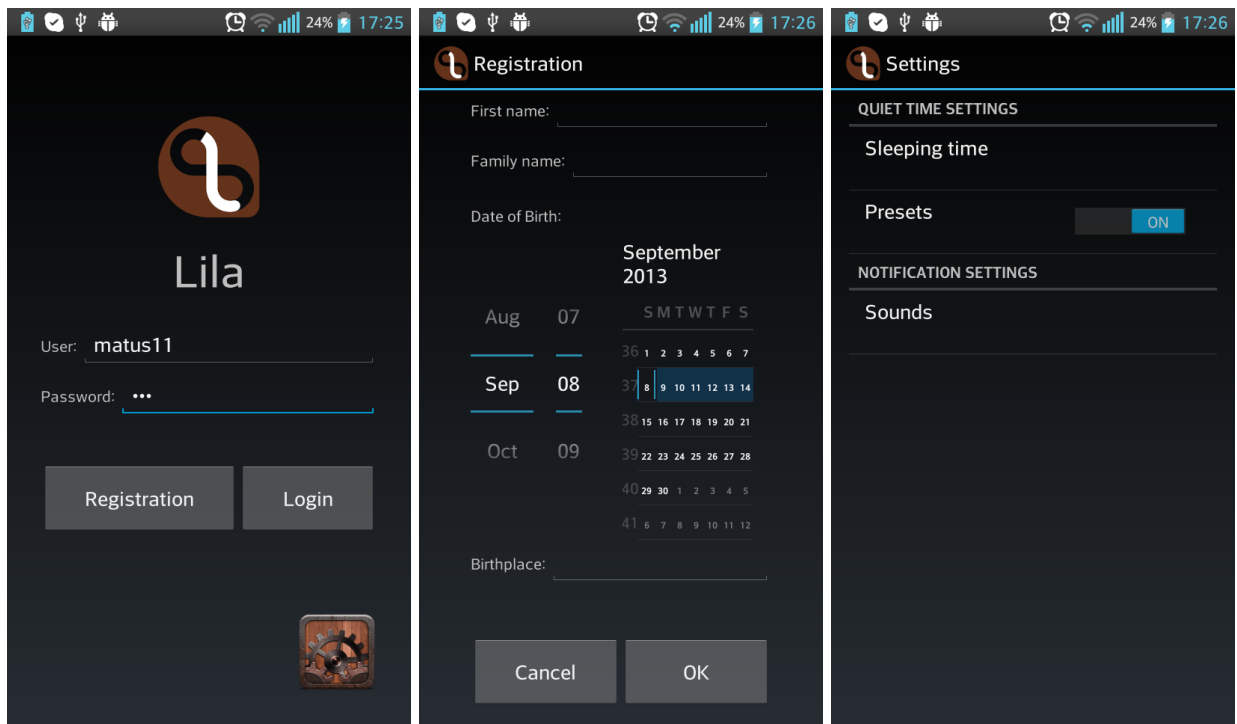


Figure A.1: Logon screen, registration screen and the main settings screen

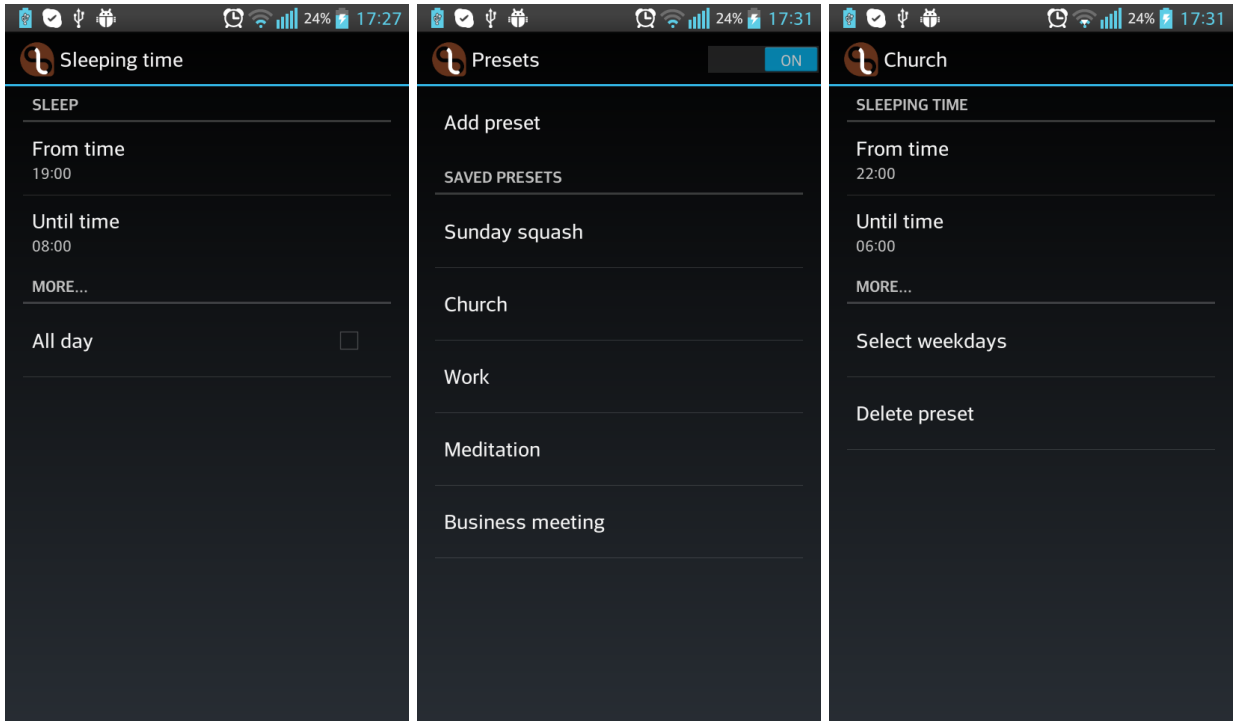


Figure A.2: Screen for setting the sleeping time, quiet time presets and the created sample preset

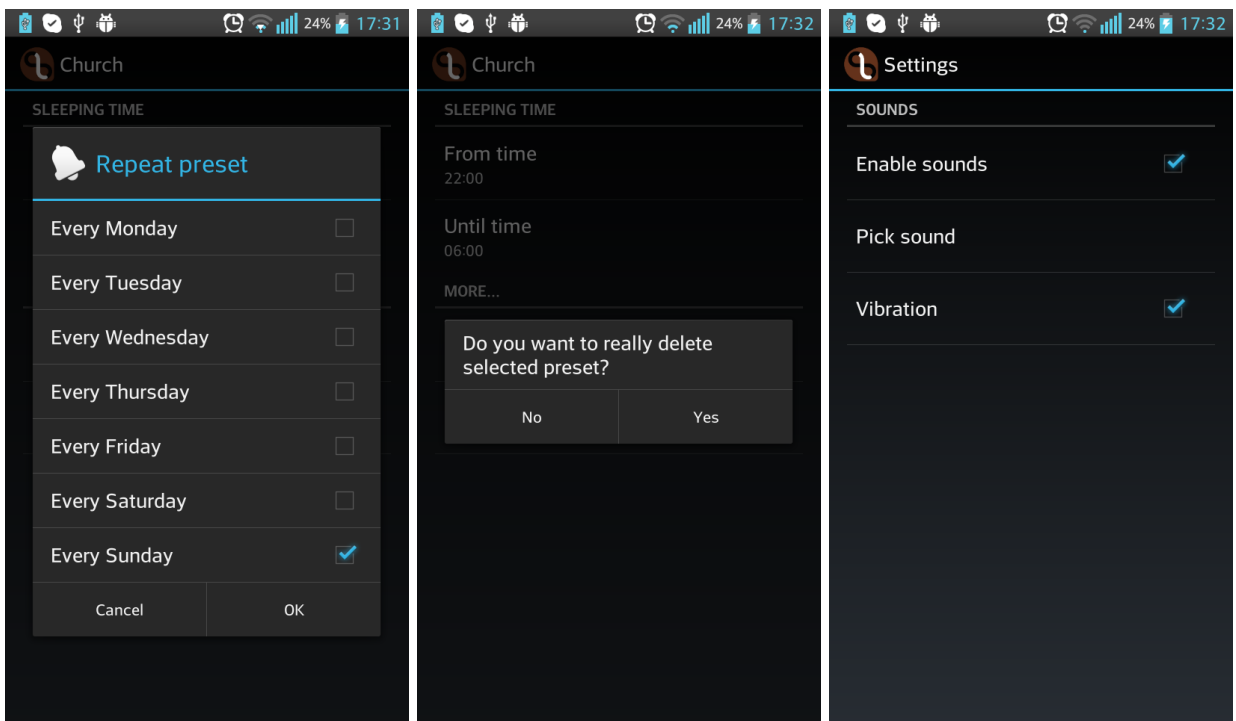


Figure A.3: Screen for setting the repeating, confirmation screen and screen for sound settings

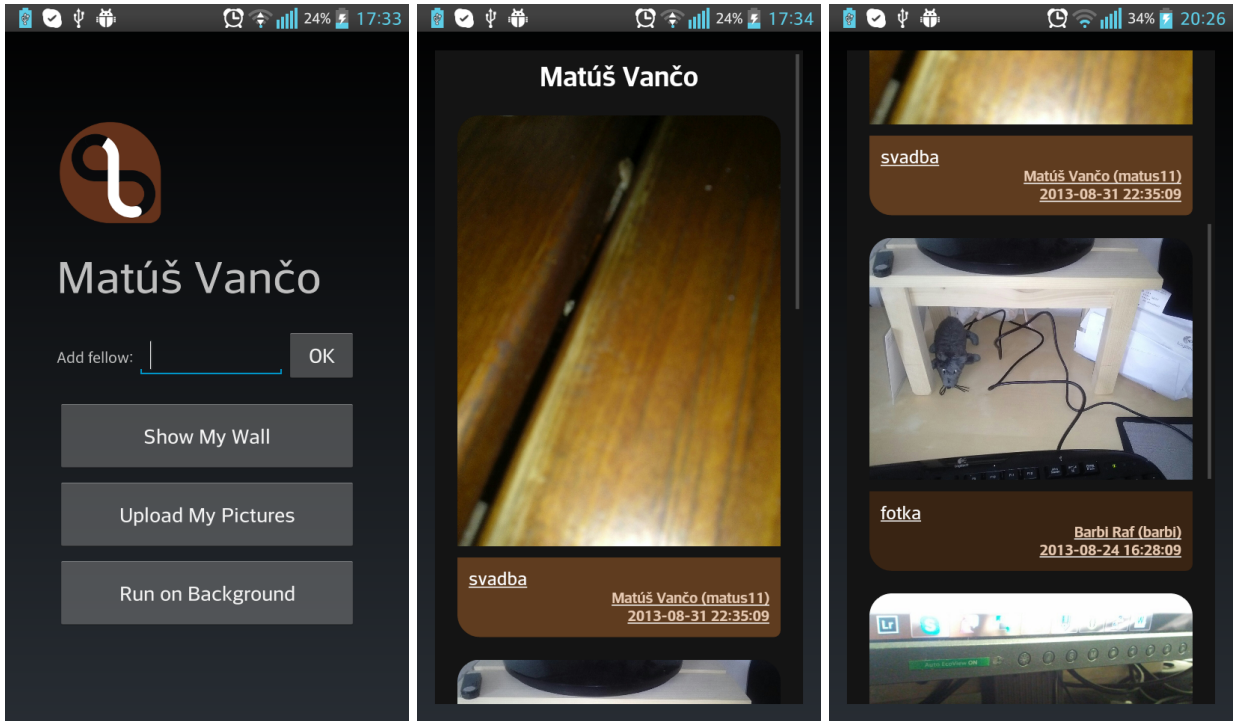


Figure A.4: User panel screen and the showing of personal wall

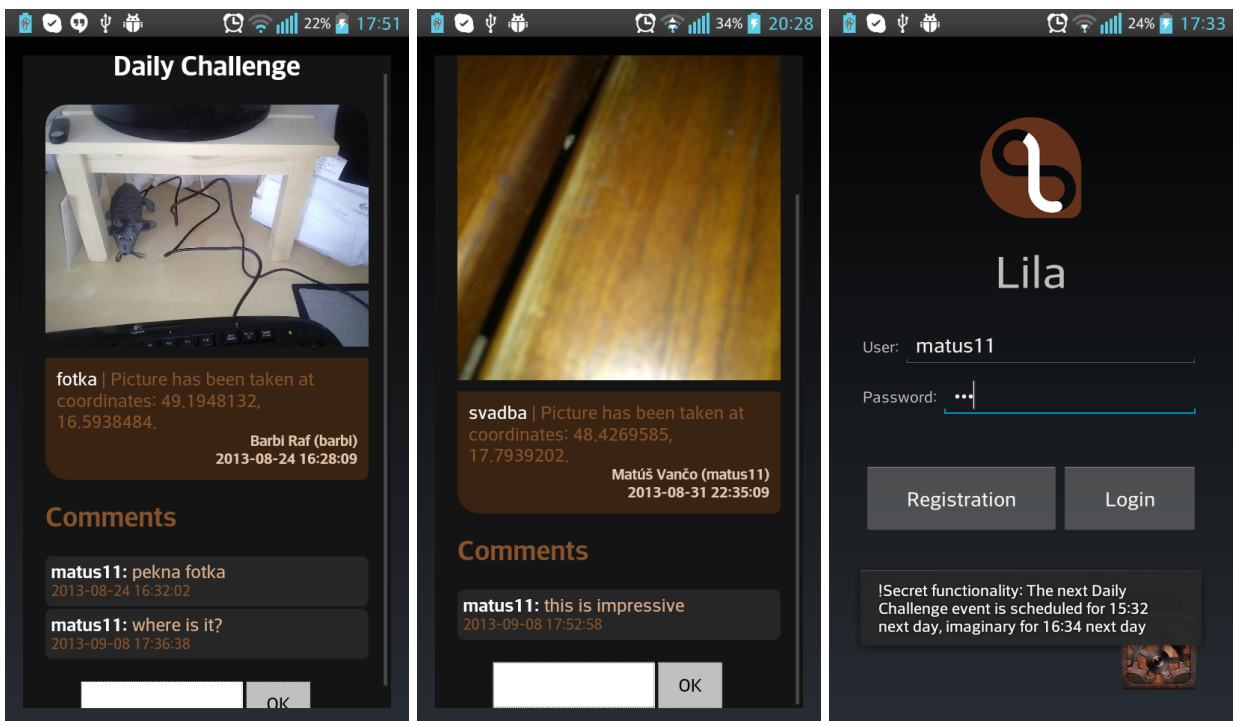


Figure A.5: Showing of uploaded daily challenge and demonstration of the hidden functionality

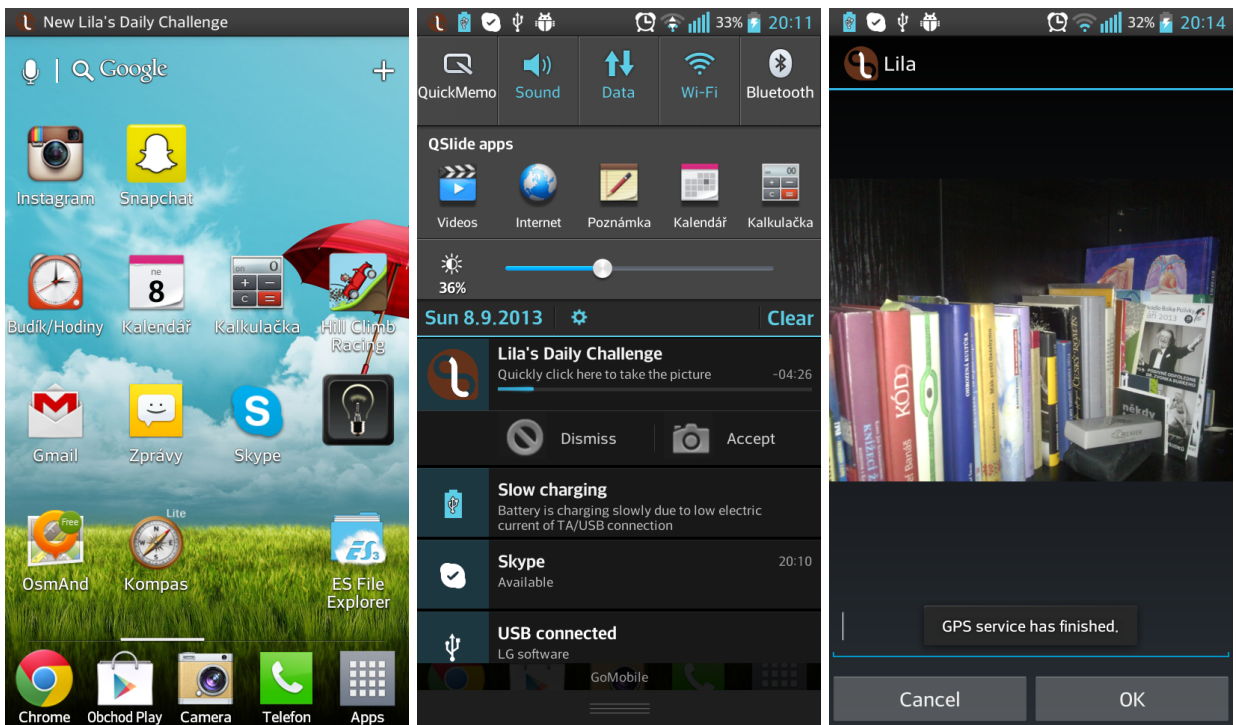


Figure A.6: Showing the notification in the notification bar and the screen to confirm the daily challenge

# Appendix B

## CD content

CD attached to this paper contains, in the root directory, these items:

- **Dissertation.pdf** – this text in the electronic format,
- **tex-sources/** – directory with the Latex source codes of this text,
- **Lila.apk** – installation file of Lila application,
- **apk-sources/** – directory with the Android source codes of the Lila application,
- **DatabaseSchema.sql** – SQL script to create database tables,
- **web-sources/** – directory with the source codes of the Web server.



# **Appendix C**

## **Project Logbook**

<b>Project title</b>	Photographic game Lila		
<b>Student name</b>	Matúš Vančo	<b>Student number</b>	12003166
<b>Serial number</b>	1	<b>Duration</b>	2 weeks
<b>Start date</b>	6.5.2013	<b>End date</b>	19.5.2013
<b>Planned to finish</b>			
The project started in advance according to the original plan. For this week, hence no task has been scheduled.			
<b>Progress, issues and comments</b>			
During this period, were evaluated project objectives and assessed motivation and background which resulted to the selection of theme. There was established communication between supervisor and student. There was chosen and established webhosting, chosen PHP framework and created and set repositories for both client and server part. There was partially set Fedora distribution and Eclipse environment.			
<b>Summary of completed work</b>			
<ol style="list-style-type: none"> <li>1. Introductory chapter written</li> <li>2. Established communication via Google+ and email</li> <li>3. Chosen <i>vlexofree.com</i> webhosting</li> <li>4. Chosen <i>CodeIgniter</i> framework</li> <li>5. Created repository for Android application (<a href="https://code.google.com/p/msc-project-lila-client/source/browse/">https://code.google.com/p/msc-project-lila-client/source/browse/</a> and Web server <a href="https://code.google.com/p/msc-project-lila-server/source/browse/">https://code.google.com/p/msc-project-lila-server/source/browse/</a> on Google Code</li> <li>6. Partially set development environment</li> </ol>			

<b>Project title</b>	Photographic game Lila		
<b>Student name</b>	Matúš Vančo	<b>Student number</b>	12003166
<b>Serial number</b>	2	<b>Duration</b>	2 weeks
<b>Start date</b>	20.5.2013	<b>End date</b>	2.6.2013
<b>Planned to finish</b>			
The project started in advance according to the original plan. For this week, hence no task has been scheduled.			
<b>Progress, issues and comments</b>			
There has been commenced detailed research about Android platform and related Android API, which includes reading technical books and internet sources like <i>Android Developers</i> site. Research was focused on issues related to the solution of the project. During these two weeks, there was started also a superficial research about Web technologies.			
<b>Summary of completed work</b>			
<ol style="list-style-type: none"> <li>1. Created Latex document structure</li> <li>2. Section <i>Basic Concepts of Android platform</i> was written</li> <li>3. Part of <i>Web technologies</i> section written</li> <li>4. Obtained important knowledge about Android platform, which will be used to design application</li> </ol>			

<b>Project title</b>	Photographic game Lila		
<b>Student name</b>	Matúš Vančo	<b>Student number</b>	12003166
<b>Serial number</b>	3	<b>Duration</b>	2 weeks
<b>Start date</b>	3.6.2013	<b>End date</b>	16.6.2013
<b>Planned to finish</b>			
<ol style="list-style-type: none"> <li>1. Evaluation of existing similar applications</li> <li>2. Initiate research about Web technologies and Android</li> </ol>			
<b>Progress, issues and comments</b>			
<p>There was scheduled and first official meeting with the supervisor. During this period, were also compiled and developed first prototype of Android application. There were determined and analyzed software requirements specification as well as compared existing similar applications. There was initiated research on the random generator.</p>			
<b>Summary of completed work</b>			
<ol style="list-style-type: none"> <li>1. Initiated first video conference via Google+</li> <li>2. Finished configuration of Fedora distribution and Eclipse development environment</li> <li>3. Developed first runnable Android application prototype</li> <li>4. <i>Software requirements specification and analysis</i> chapter written</li> <li>5. Evaluated existing similar applications</li> <li>6. Finished research about random pseudogenerators</li> </ol>			

<b>Project title</b>	Photographic game Lila		
<b>Student name</b>	Matúš Vančo	<b>Student number</b>	12003166
<b>Serial number</b>	4	<b>Duration</b>	2 weeks
<b>Start date</b>	17.6.2013	<b>End date</b>	30.6.2013
<b>Planned to finish</b>			
<ol style="list-style-type: none"> <li>1. Design of application screens</li> <li>2. Further detailed research about Android API, focused on parallel programming and background services</li> <li>3. Detailed research about Web technologies</li> <li>4. Design of database schema</li> <li>5. Design of communication between client and server</li> </ol>			
<b>Progress, issues and comments</b>			
<p>During this period, there was performed intensive research about the whole spectrum of related technologies. There was initiated detailed research about server technologies and network communication. There was also proposed graphical layout of application screen.</p>			
<b>Summary of completed work</b>			
<ol style="list-style-type: none"> <li>1. Finished research on android platform</li> <li>2. Initiated research on network communication</li> <li>3. Chapter <i>Software design and development</i> written, except for <i>Implementation</i> section</li> <li>4. Application screens were allocated and designed the layout of graphical elements</li> </ol>			

<b>Project title</b>	Photographic game Lila		
<b>Student name</b>	Matúš Vančo	<b>Student number</b>	12003166
<b>Serial number</b>	5	<b>Duration</b>	2 weeks
<b>Start date</b>	1.7.2013	<b>End date</b>	14.7.2013
<b>Planned to finish</b>			
<ol style="list-style-type: none"> <li>1. Initiate the intense development of Android application</li> <li>2. Finish the server configuration</li> <li>3. Start development of server part</li> </ol>			
<b>Progress, issues and comments</b>			
<p>During this period, was created lots of the main functionality of the client application. This includes the creation of random time generator, notification system and application preferences. Configuring the server was set up in advance, but there was revised existing webhosting solution and passed to paid web hosting. There was also proposed database schema, which was consequently tested on the server.</p>			
<b>Summary of completed work</b>			
<ol style="list-style-type: none"> <li>1. Developed main functionality of client application</li> <li>2. Detailed design of client-server communication</li> <li>3. Design of database schema</li> <li>4. Creation of database tables</li> </ol>			

<b>Project title</b>	Photographic game Lila		
<b>Student name</b>	Matúš Vančo	<b>Student number</b>	12003166
<b>Serial number</b>	6	<b>Duration</b>	2 weeks
<b>Start date</b>	15.7.2013	<b>End date</b>	28.7.2013
<b>Planned to finish</b>			
<ol style="list-style-type: none"> <li>1. Finish Android application</li> <li>2. Further development of the server and communications</li> </ol>			
<b>Progress, issues and comments</b>			
<p>There is developed the main server functionality like uploading photos from daily challenge and registration as well as logon. Subsequently, there is developed application logic to communicate with the server. It involves logging and uploading photos. Application is almost complete, but lacks some not inconsiderable goals, which contradicts with the original plan.</p>			
<b>Summary of completed work</b>			
<ol style="list-style-type: none"> <li>1. Basic logic of server developed</li> <li>2. Applications can communicate with the server</li> <li>3. <i>Implementation</i> section written and finished emphSoftware design and development chapter</li> </ol>			

<b>Project title</b>	Photographic game Lila		
<b>Student name</b>	Matúš Vančo	<b>Student number</b>	12003166
<b>Serial number</b>	7	<b>Duration</b>	2 weeks
<b>Start date</b>	29.7.2013	<b>End date</b>	11.8.2013
<b>Planned to finish</b>			
<ol style="list-style-type: none"> <li>1. Finish development of server</li> <li>2. Interconnecting the server and client part of the project</li> </ol>			
<b>Progress, issues and comments</b>			
<p>There is added functionality to synchronize the saved photos in the Android application. Also, the application contains WebKit browser for displaying photos from server. There has been proposed further improvements and was redesigned graphical user interface. There was created the program icon and the development of Android application has been completed. Server part has been tested and finalized as well.</p>			
<b>Summary of completed work</b>			
<ol style="list-style-type: none"> <li>1. Finalized the server solution</li> <li>2. Added synchronization functionality to Android application</li> <li>3. Added WebKit browser for displaying photos from server</li> <li>4. Application icon designed and graphical user interface redesigned</li> </ol>			



<b>Project title</b>	Photographic game Lila		
<b>Student name</b>	Matúš Vančo	<b>Student number</b>	12003166
<b>Serial number</b>	8	<b>Duration</b>	2 weeks
<b>Start date</b>	12.8.2013	<b>End date</b>	25.8.2013
<b>Planned to finish</b>			
<ol style="list-style-type: none"> <li>1. Evaluation of project</li> <li>2. Assessment of functionality</li> </ol>			
<b>Progress, issues and comments</b>			
<p>In this final phase of the project, there was finalized and tested the Android application. There has been evaluated the contribution and benefit of whole project as well as assessed the vulnerabilities towards the system and proper functioning of the application. The last part was to design the possible further development. During this phase was completed text of the dissertation and added attachments.</p>			
<b>Summary of completed work</b>			
<ol style="list-style-type: none"> <li>1. Finalized Android application</li> <li>2. Specified contribution of the project</li> <li>3. Assessed functionality and security risks</li> <li>4. Dissertation completed, corrected and finalized</li> </ol>			

# Appendix D

## Git history of Android Application

<b>Date</b>	<b>Commit log message</b>
<i>Aug 25, 2013</i>	FINAL COMMIT - application is fully working according requirements.
<i>Aug 5, 2013</i>	Added 2 icons for more pretty notification. Notification can now show progress bar and remaining time. The user can now enable or disable vibrations and pick his own sound for notification. Daily Challenge has 5 minutes and during this time this will be realized three times (three sound alerts or vibrations or both). App now can save comments about captured pictures to configuration file called commit, which will be basic source for synchronizing pictures with locations and comments to server. There has been treated lot of app bugs and linked (not working yet) part of open source application for getting location coordinates and address using location services. There has to be linked also google-play-services for proper working of location service.
<i>Aug 4, 2013</i>	Working camera, taking and saving pictures. UI contains EditText for adding thoughts about new photo.
<i>Aug 3, 2013</i>	Created logo icon for application and added into.
<i>Aug 2, 2013</i>	Camera functionality added to Daily Challenge.

<b>Date</b>	<b>Commit log message</b>
<i>Jul 31, 2013</i>	Test of git.
<i>Jul 31, 2013</i>	Remove useless files from repository.
<i>Jul 30, 2013</i>	Random algorithm done and tested.
<i>Jul 29, 2013</i>	Fully working Ssettings and Notifications according it (but not fully tested yet).
<i>Jul 5, 2013</i>	Management of presets to set the quiet time.
<i>Jun 28, 2013</i>	New preferences after click HW menu button to set sleeping time.
<i>Jun 25, 2013</i>	Advanced Preferences with master switch button.
<i>Jun 21, 2013</i>	Application which sets Alarm Manager for planning Daily Challenge every day. It is prepared for calling event in Notification Manager at random time (not yet implemented).