VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INFORMAČNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INFORMATION SYSTEMS

ADAPTIVE WEB PORTAL

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE ADAM PÁCHA
AUTHOR

BRNO 2010

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INFORMAČNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INFORMATION SYSTEMS

ADAPTIVNÍ WEB PORTÁL

ADAPTIVE WEB PORTAL

BAKALÁŘSKÁ PRÁCE
BACHELOR’S THESIS

AUTOR PRÁCE ADAM PÁCHA
AUTHOR

VEDOUCÍ PRÁCE Ing. RADEK BURGET, Ph.D.
SUPERVISOR

BRNO 2010

Abstrakt

Práce se zabývá dvěma hlavními tématy. V prvním zkoumáme možnosti nabízení obsahu portálu jeho navštěvníkům. Nabízení je založeno na modelování chování uživatele portálu a na systému hierarchického nabízení obsahu. Druhé téma se zabývá šířením obsahu webu do sociálních sítí a poskytováním dat a funkcionality webu aplikacím třetí strany pomocí aplikačního programového rozhraní. První kapitoly popisují použité technologie, protokoly a bezpečnostní zranitelnosti webu. Další kapitoly jsou věnované návrhu aplikace a její následné implementaci. Práce je součástí projektu na univerzitě UTAD a funkce v tomto projektu je definována v úvodu technické zprávy.

Abstract

Thesis deals with two main topics. The first topic studies possibilities of content offering to a particular user. Content offering is built on the model of portal user behavior and hierarchical system of content offering. The second topic deals with content propagation to the social networks and way how to provide data and web functionality to a third party application throught the application programming interface. The first chapters of technical report describe used technologies, protocols and web security vulnerabilities. Other chapters are devoted to the application design and subsequent implementation. Thesis is part of the project at University of Trás-os-Montes and Alto Douro. Functionality of thesis in this project is described in introduction of technical report.

Klíčová slova

HTML, CSS, PHP, SOAP, MySQL, Smarty, jQuery, bezpečnostní zranitelnosti webu, napojení sociálních síti, model chování uživatelů portálu, související obsah, web API, vrstvová architektura, geografické značky

Keywords

HTML, CSS, PHP, SOAP, MySQL, Smarty, jQuery, web security vulnerability, social network connection, model of portal user behaviour, related content, web API, layer architecture, geographical tags

Citace

Adam Pácha: Adaptive web portal, bakalářská práce, Brno, FIT VUT v Brně, 2010

Adaptive web portal

Prohlášení

I declare that I have prepared this bachelor thesis independently under the leadership of Ing. Radek Burget, Ph.D.

....
Adam Pácha
July 28, 2010

Poděkování

I would like to thank Ing. Radek Burget, Ph.D., Hugo Paredes, Ph.D. and Benjamim Fonseca, Ph.D. for technical assist and for the time devoted to consultations.

Contents

List of Figures

Chapter 1

Introduction

Today's web pages contain a lot of data and present to users data which are not searched or requiered by them. Title pages mostly contain content which is uninteresting for users. Only small part of those contents is really interesting for specific user. Adaptive web portal finds a way how to offer a content to appropriate users.

The aim of bachelor's thesis is to create system for modeling the behavior of the portal users with the aim of offering other related contents to a particular user. User's behaviour can be tracked and this information is used with the aim to create user profil. System of content offering can be based on this information. Web content from our application can be expanded outside. Social networks provide a great place where we can situated a backlinks to our content.

Current trend offers to users not only content from web pages but also across programming interface. Users or developers can use our data however they want. Adaptive web portal contains application programming interface which allows developers to create applications for sharing, uploading and downloading data from Adaptive web portal application. API is multiplatform and can be used for example by mobile phone applications etc. With social networks integration is Adaptive web portal a bridge between third-party applications and social networks.

Adaptive web portal is a user-friendly application. This requirement is observed with AJAX communication and with implemented windows system which is written in jQuery.

The application core can be used as framework for developing web pages which are constructed using php, mysql and xhtml. The core provides SQL abstraction, inserts or updates data from web forms into database automatically, language layer, directory layer, user registration, automatic script for creating MySQL tables etc.

Technical report describes used technologies, application design, implementation details and final tests.

Chapter 2

Thesis as a part of project at University of Trás-os-Montes and Alto Douro

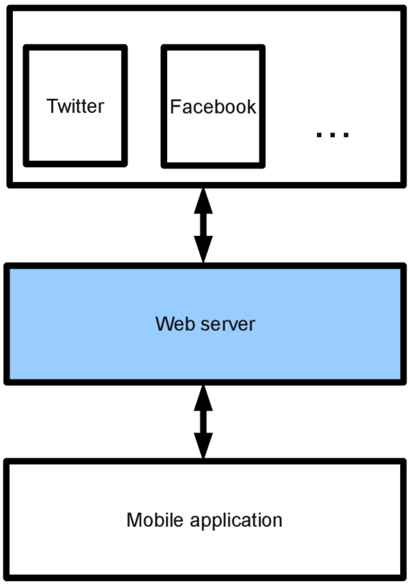

Figure 2.1: Project model (part described in thesis Adaptive web portal is highlighted)

Main idea of this project is explore content to the most viewable places. Can you image emergency situation where this type of information must be propagated among large public community. Can you image situation where we have a really nice photo or photo album and we would like say to our friends about this content. Project can automatically propagate this information to the social networks. That's not all. The content is geotagged, even this process is automatic when information is available in EXIF metadata or from GPS chip. This feature brings us possibilities of geolocation filtration.

Web server is a web application therefore we further describe technologies PHP and MySQL. Web server is not only bridge between mobile application and social networks.

Server is fully valued application which brings us a way how to presents and stores many type of content and use connections with social networks. Application uses for the purpose of presentation technologies XHTML, CSS, Smarty, jQuery and Google Analytics. Web application stores information about connections to social networks for each user. The purpose of OAuth token and openIdClaimedId are described in chapter OAuth and OpenID. Web application enables many operations with pictures thanks to WideImage PHP Library. Automatic geotagged system is based on EXIF information which describes each picture. For purpose of reading this information is used the PHP Exif Library. Web server provides wide functionality for cross-platform applications. Connection with mobile application is realised by the web services which are based on SOAP. Web server uses possibilities of ZEND framework application especially SOAP Extension. All of this technologies are described in next chapter.

Connection with social networks is based on Social Network API. This API is derived from the thesis Social Network Integration into an Information Portal[3]. Web server only stores connection information designed for access to social network through the Social Network API. Web server also provides a way how to obtain connection information through this API. In web server application is integrated system of content filtering for Social Network API. With this system of filtering we can keep a user's privacy. More information about integration of this API to the web server application is mentioned in section Connection with social networks.

The thesis Adaptive web portal does not resolve mobile application. Mobile application is an object of thesis which was written at University Trás-os-Montes and Alto Douro. Our thesis provides functionality of web server. This server provides many functions for mobile application. Mobile application can upload and download all type of content which are stored in the web server database. Downloaded content can be filtered by many parametrs including coordinates, owners, categories etc. More information about this API is mentioned in the section API under chapter Implementation.

Chapter 3

Use technologies and protocols

3.1 XHTML

XHTML (Extensible Hypertext Markup Language) is a family of XML languages that extends often used HTML. In project is used XHTML 1.0 strict which is the XML equivalent to strict HTML 4.01 which is the SGML family. If we compare HTML and XHTML we will find this main restrictions:

- elements must be properly nested
- elements must always be closed
- elements must in lowercase
- document must have only one root element
- attribute values must always be quoted
- attribute-value pairs must be written in full (does not support attribute minimization)

[36]

3.2 CSS

CSS is a style sheet language which describes semantics of documents written in HTML or XHTML. CSS is designed primarily to separate document content from document presentation. CSS allows present documents in different style according to rendering methods such as on-screen, on-print etc. CSS uses hierarchical model and inheritance where child inherits particular features from parent. Rules are interpreted in this cascade:

- extern template
- local styles of document
- local styles of element

[37]

3.3 PHP

Section is based on [1].

PHP (Hypertext PreProcesor) is a widely used scripting language that was designed for purpose to create dynamic web pages. PHP is a free software under the PHP Licence. PHP is a server-side scripting language that means that scripts are executed on the server. PHP supports many databases like MYSQL, PostgreSQL and etc. Syntax of language is influenced by several programming languages - C, Perl, Java, C++. PHP is procedural and objective language. Since version PHP5 we can use features of objects language like encapsulation, polymorphism and inheritance. For PHP exists several framewokrs for worth mentioning Zend or PEAR.

3.3.1 Zend SOAP

Zend framework is an objected-oriented web application framework implemented in PHP 5. Framework is an open-sorce project under the New BSD License. Zend framework contains many components from categories - Model-View-Controller (MVC), Tooling and Rapid Application Development, Database, Internationalization and Localization, Authentication, Authorization, Session management, Web and Web Services etc. [40]

SOAP (Simple Object Access Protocol) is a XML based protocol designed to let applications exchange information over HTTP. SOAP is a successor of XML-RPC protocol. SOAP provides a way to communicate applications independently on operating systems and programming languages. [27]

Zend Soap component is from the family Web and Web Services. Zend Soap server is intended to simplify Web Services server part development for PHP programmers. Server may be use in WSDL or non-WSDL mode.[39]

WSDL document may be auto-generated with functionality provided by Zend Soap AutoDiscovery component which keeps WSDL document actual. WSDL (Web Services Description Language) is an XML based language designed to describe Web services and their access. Zend Soap AutoDiscovery component utilizes the PHP docblocks to determine the parametrs and return types.[39, 26]

3.3.2 WideImage

WideImage is an open-source object-oriented PHP library for image manipulation. The library is focused on the most common image operations. It provides methods for loading and saving images from/to files, strings, database, supports cross-format conversion. WideImage support all formats naturally supported by GD expension which must be compiled with PHP. Library also provides a way to edit pictures and apply a many filtres (negative, greyscale, unsharp, rotate, flip etc.). [6]

3.3.3 PEL

The PHP Exif Library (PEL) provides a way how to manipulate with EXIF data. These data are placed by camera into images as meta data and describe information like coordinates, ISO value, shutter speed, focal lenght, aperture value etc. [21]

3.4 MySQL

MySQL is a database system created by MySQL AB corporation and today is owned by Sun Microsystemsm Inc. The MySQL software is under the GPL (GNU General Public License) or as well as under variety of proprietary agreements. MySQL is a relational database management system (RDBMS) and it is multi-platform. Communication is based on SQL which is a standard language for accessing and manipulating databases. MySQL Database server is very fast, reliable and easy to use. Many of high-traffic web sites (including Facebook, Wikipedia, Google, Flickr) use MySQL for data storage.

MySQL architecture is on top layer base on tools for client-server communication. The second layer contains a large part of the MySQL brain including code for parsing, analysis and optimalization. Layer also includes all functionality which is provided by multiple storage engines. Third layer contains storage engines. These engines are responsible for obtaining and storing all the data stored in MySQL. The most popular storage engine is a nontransactional MyISAM and transactional InnoDB. The other storage engines are Merge, Memory, BDB, EXAMPLE, FEDERATED, ARCHIVE, CSV, BLACKHOLE.

Since version 5.0 we can use cursors, stored procedures, triggers, views and XA transactions.

MySQL connection manager associate each client connection with a dedicated thread to it. This tread handle authentication and all requests for this connection.[24, 25]

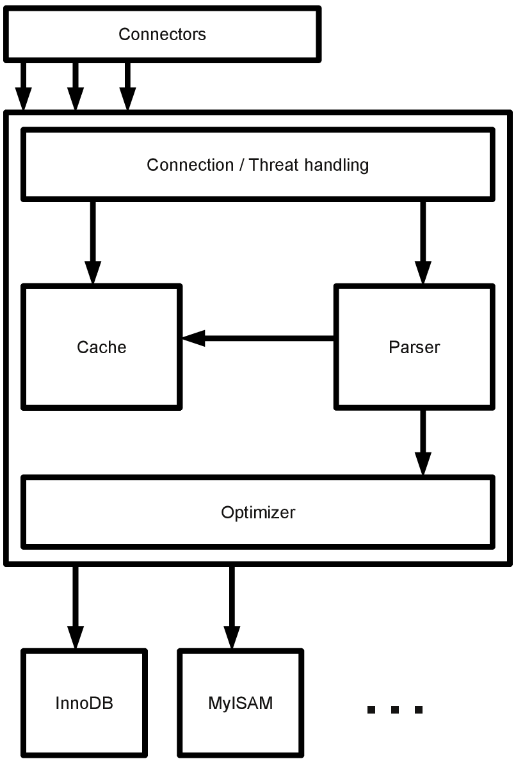

Figure 3.1: MySQL server architecture

3.5 Smarty

Smarty is a web templat system under the LGPL License. Smarty generates web content by replacement of special Smarty tags by web content. These tags can be directive for variables, functions, loops or logical statements. Smarty seperates PHP code by the presentation of the web page. It simply allows developers to create new web templates without knowledge in PHP. Smarty provides debugging option. Smarty engine is optimised for performance and allows to use cache system. [22]

3.6 jQuery

Section is based on [33, 17].

jQuery is a cross-browser JavaScript library that simlifies event handling, animatimating, selecting DOM elements and Ajax iteractions. The original autor is John Resigfrom BarCamp NYC in 2006. jQuery provides capabilities to create plugins on the top of the Javascript library.

Today we can use many plugins which extend standart jQuery library. We can mention jQuery UI which is a low level abstraction for low-level interaction and animation, advanced effects and high-level, themeable widgets. jQuery UI with jQuery Core, QUnit (project for testing jQuery code and plugins) and Sizzle (CSS selector engine) is a part of the jQuery Project. jQuery UI includes CSS Framework which allows to developers work with CSS more effectively.

Programming with jQuery is exactly as the motto of jQuery - „Write less, do more“.

3.7 Google Analytics

Section is based on [8, 7].

Google analytics is product by Google that generates detail statistics about the visitors on our web pages. This solution is free and offers to user 5 milion pageviews per month (that is approximately 2 pageviews per second). If amount is insufficient we have to link our site to an AdWords campaning.

GA (Google analytics) can track from where visitors come from, how long they have been connected, their geografical possition, bounce rate. But that is not all. GA provides more advanced features like visitors segmentation. Developers can also integrate Google AdWords or trace their marketing aims. Reports are not limited with options included on dashboard. User can creates custom reports and add it on the dashboard.

For data export, creating of integrations and developing client application we can use Data Export API. Customize tracking with Tracking API.

With Analystic Intelligence we can set up automatical alerts.

GA provides wide options for tracking your web visitors.

3.8 OAuth

Section is based on [4, 9].

The OAuth (Open Authorization) protocol enables websites or applications to share private users resources (photos, contact lists etc.) stored on one site with another having to hand out their username and password. An example is allowing our web application (the

Consumers) to access private photos stored on Facebook (the Service Provider) without requesting User to provide their Facebook credentials to our application.

The service Provider enables the Consumer to estabilish a Consumer Key and a Consumer Secret. The Consumer must estabilish a Consumer Key and Consumer Secret with the Service Provider.

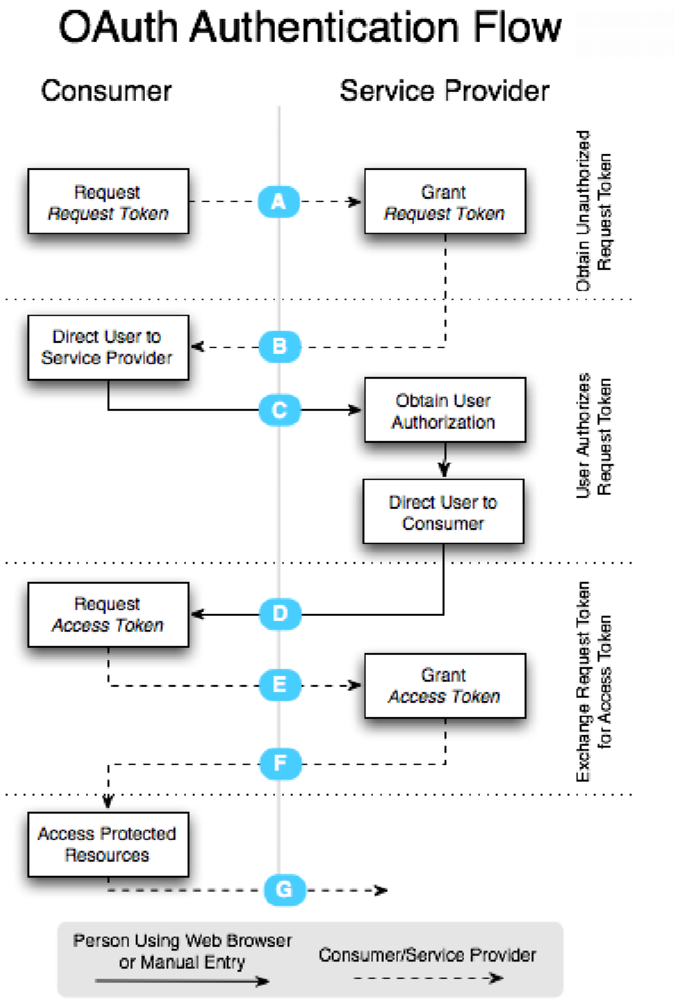

Figure 3.2: OAuth authentication flow (source http://oauth.net/core/1.0/)

The step B and D send token defined for token verification. The step C and E confirm token and the step F sends final Access Token.

The purpose of Request Token is to receive User approval.It must be used only to obtain an Access Token. The consumer exchanges Request Token for an Access Token to reveice access to Protected Resources.

OAuth can be used as a user verifying method for any API type (SOAP, REST etc.) beyond not only with web applications.

3.9 OpenID

Section is based on [23].

OpenID allows users to use existing account and sign up to multiple webpages and it does without the Relying Party needing access to end user credential such as a password or other sensitive information. You can also associate information such as name or email address with your OpenID. You can control which of this information will be shared with the websites you visit. Password is given to OpenID provider and we do not need to have any doubts with our password. No website ever sees your password. OpenID is decentralized and not own by anyone. Everyone can use OpenID or become an OpenID provider.

An OpenID is in the form of URL (Uniform Resource Locator) or XRI (Extensible Resource Identifier). OpenID providers offer the service of registering OpenID URLs or XRIs and provides OpenID authentication. Examples of OpenID provider URL formating:

- MyOpenId: http://username.myopenid.com
- Yahoo: http://yahoo.com/
- Google: https://www.google.com/accounts/o8/id
- Google profile: http://google.com/profiles/username
- Verisign: http://username.pip.verisignlabs.com

Chapter 4

Security vulnerability

Chapter is based on [15].

4.1 Cross site scripting (XXS)

Section is based on [13, 29].

XXS is a special case of code injection. In XXS, attacker tries to insert his own code prepared in the security context. The code can be HTML but the biggest security risk is JavaScript which can be also loaded from the extern third-party web application. According to that kind of loading technique is XXS named „Cross site scripting“.

We can classify XXS on non-persistent(reflected) and persistent(stored) or traditional and DOM-based vulnerabilities. Persistent vulnerabilities occur when data provided by attacker are on the server side otherwise when data are processed by the client side it is called non-persistent. Tradition vulnerabilities can occur on the server side code which is responsible for HTML response beyond DOM-base on content performed by the client (content of Document Object Model).

Protection against XXS is simple. We have to treat all function returns, in PHP with function htmlspecialchars() or disanable JavaScript on client side.

4.2 SQL injection

Section is based on [10, 32].

SQL injection is a code injection technique which occuring on database layer. The security vulnerability can occur when the user input is incorrectly filtered for escape characters and then passed into a SQL statement. This vulnerability gives to the end user of the application permission perform statements on the database layer.

A way how to prevent injection is to escape all characters that have a specific meaning in SQL. For exampe, in PHP we can use function mysql_real_escape_string() before sending all of the SQL queries.

4.3 Cross-Site Request Forgery

Section is based on [12, 28]

CSRF (Cross-Site Request Forgery) is also known as a one-click attack. The attack works by submitting unauthorized commands toward webpage where user has been already signed in. User will send the command which execute any action unconsciously.

For example Adam is using his internet bank account and also chatting with Peter. Peter will slip to Adam link bank.php?withdraw=1000?account=adam&to=Peter. If Adam click on this link Peter will receive requested money.

For protection we can check the HTTP Refer header. This header includes the address (URI) of the resource from which the request was obtained. Better technique how to protect our web is to use authenticition token for each operation with user account.

Good practices how to protect our application are use POST requests instead of GET and log off user after several minutes of his inactivity.

4.4 Clickjacking

Section is based on [14, 31].

Clickjacking is a malicious technique which asks users to perform unconsidered actions by clicking on a hidden links. Attacker loads pages hidden in transparent layer over pages which offers to user dummy buttons. Buttons from different layers are on a screen in the same coordinates and user unconsciously pushes button other than he wanted.

Prevention by this attack is use header X-Frame-Options:deny. This option reject the using of our application in frame.

4.5 Session hijack

Section is based on [11, 38].

4.5.1 Session fixation

This section is based beyond on [30].

Session fixation is a malicious technique which exploits vulnerabilitity of system. Vulnerability allows third-party user to set SID (session identifier) to another person. Whenever victim signs into application with this dummy SID attacker can use the SID for his own use.

Protection in PHP is not difficult. We can use function session_regenerate_id before every unsafe operation like a user loggin.

4.5.2 Session sidejacking

Session sidejacking is a security vulnerability in which attacker uses packet sniffing to read traffic between two parties. From that traffic he can steal session cookie.

A way how to protect the application is user's IP address control or header information User-agent. This protection technique is not holeproof. The best way of protection is to use data encryption passed between parties.

Chapter 5

Design

5.1 The way of modeling behavior of portal users

With the aim to create profile of user behavior dynamically we can collect data from user's browsing in our content gallery.

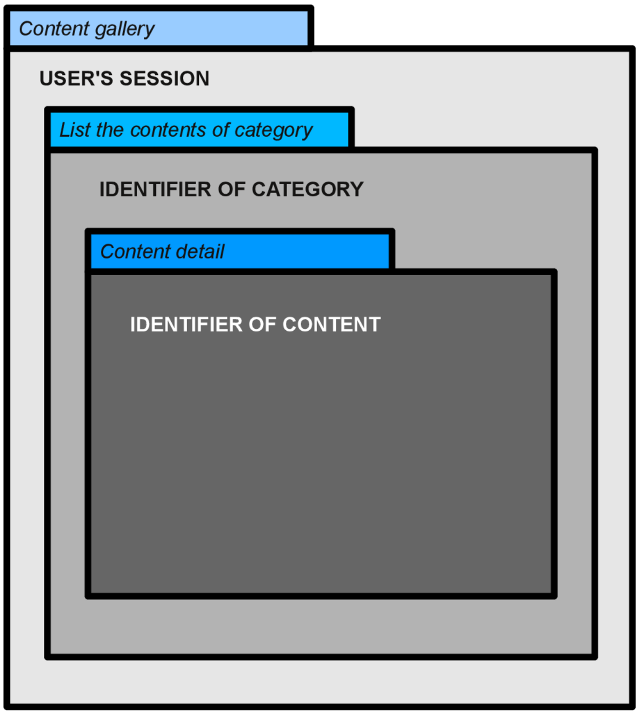

Figure 5.1: Model of content gallery

This model describes the most significant information which can be collected in content gallery. Every user who browses in content gallery is recognised by his session which is unique for each user. Deeper towards into the model we can specify user behavior easier. If content gallery includes logical category system and content is strictly divided into these categories we can use next two levels of the model. Second level inherits significant data from the first level. We can discover which of our content category is the most browsed by user. The third level of the model adds the last information which specify visited content and also inherits all last features. This layer can identify which content was visited by user. Combination of the second and third layer we will receive data for model of behavior. Thanks to the first model layer we can separate these data for each user.

5.2 The way how to offer related contents to a particular user

Each content is described with many information. We can use geotag, tag of user in content, division of contents into categories or timestamp as a content meta information. From the user point of view each of this meta information influence content differently.

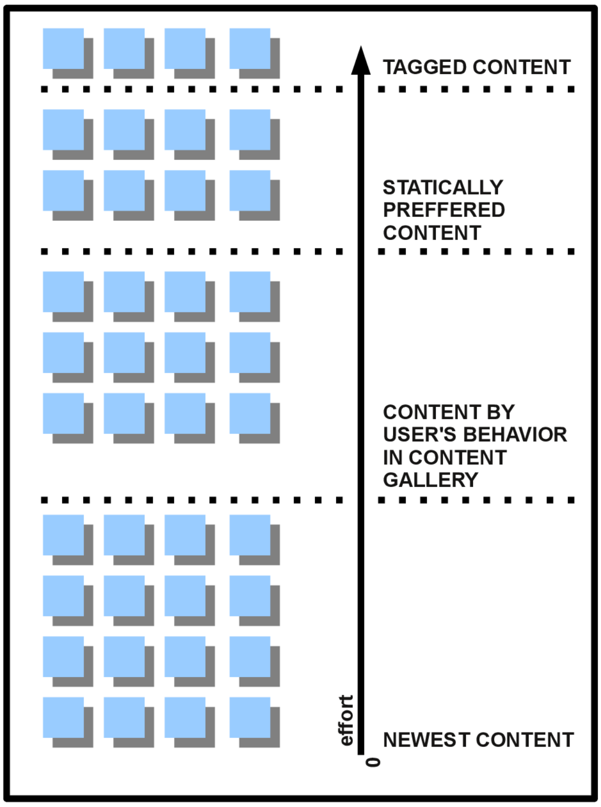
TAGGED CONTENT
STATICALLY PREFFERED CONTENT
CONTENT BY USER'S BEHAVIOR IN CONTENT GALLERY
NEWEST CONTENT
effort
0

Figure 5.2: Model of offering related contents to particular user

User has to spare different effort to receives content from each layer of the model. The simplest way how to receive content is to use timestamp information. If we want to specify content offering we can use user profile obtained from model of his portal behavior. This model is created automatically during user's browsing in content gallery. The third level of offering is based on statical category preferences. These preferences are set by user for each category what he wants. The best way how to offer a content to the user is to tag this user on the content. After that the content is especially offered to the particular user. This technique is used by some of the social network to specify the content offering for users.

This is the best way how to offer content to the particular user. The biggest effort is spared in tagging process.

The model presents measure of content specification for the particular user. The blue squares presents amount of filtered content from the same content group. The result of Model of offering related content is that tagged content is designated especially for the particular user. The problem is small amount of this content and great effort to create this type of offering. With the aim of compensation this disadvantege we should use every level of this model. More information is in chapter Implementation.

5.3 Use case diagram

System defines four main user types.

- visitor - permission type with narrow options (only to create account, to access account or to sign in)
- user - main type of permission
- API user - this user extends permission for API web services
- admin - extend functionality of API user for public category management

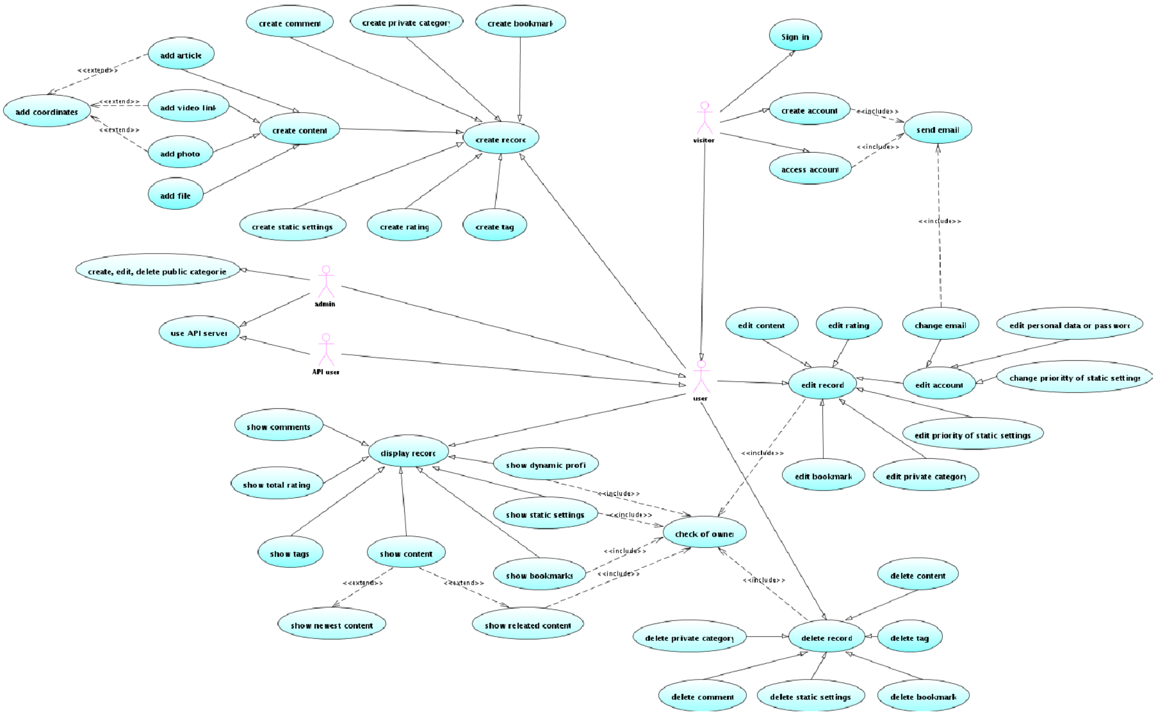

Figure 5.3: Use case diagram

5.4 Database design

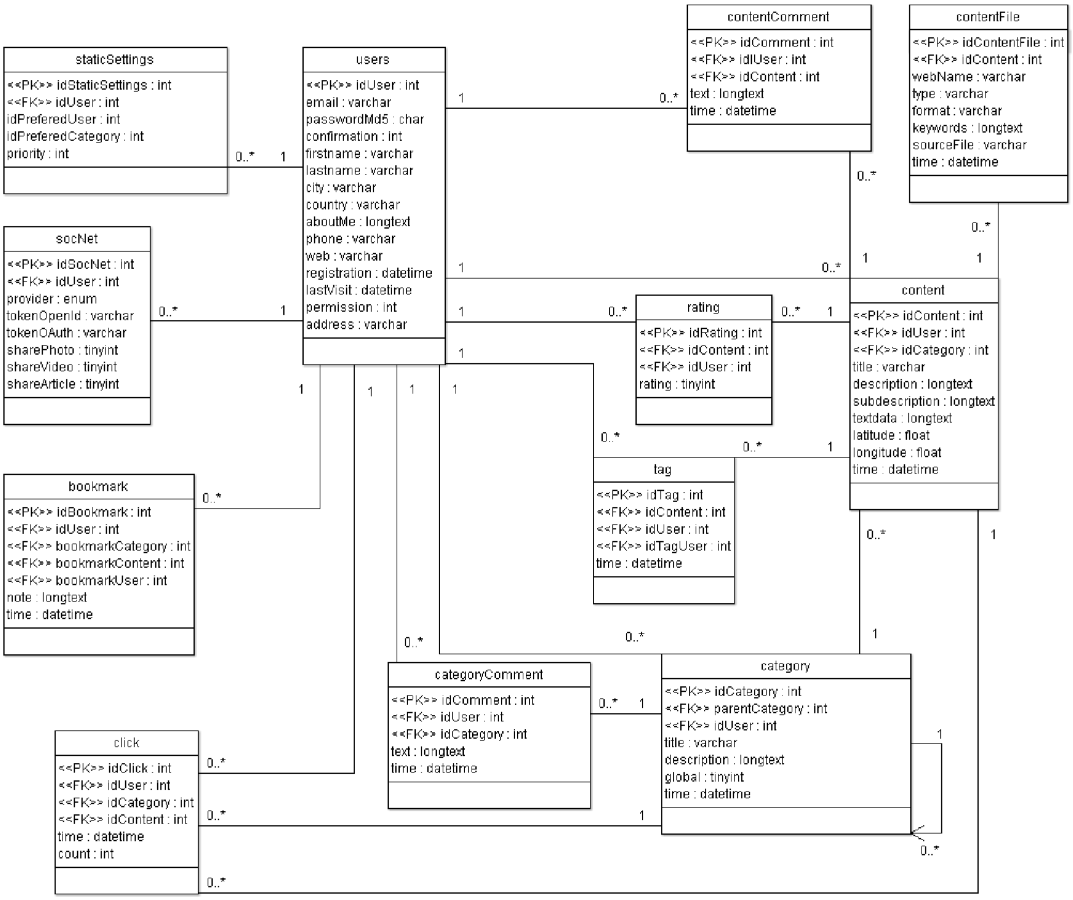

Figure 5.4: E-R diagram

Table users describe user profil and also include the same columns for account management. Column confirmation represents a flag that indicates whether user's account was verified or not. Column permission gives to users different permissions.

Table click executes main pillar for generating dynamic profil information about registered users.

Another data intended for system of offering related content are stored in staticSettings table. In this table are stored information for static priority offering.

Table category and categoryComment provide system of categories with unlimeted depth of subcategories. Every category can be commented.

Web content and its options are holding in tables called content, rating, tag, contentComment and contentFile. ContentFile does not contain binary data. It contains only path to files which are stored on server. With the aim to make searching in content much easier the best way is to add a column keyword to the contentFile table which describe every file. Format column specifies file extension, type describes type of content (pdf, pack etc.), webName divides files with the same type and format. For example, jpg pictures are uploaded three times in different sizes therefore webname has values range small, medium, large.

Bookmark data are stored in table bookmark.

Web is connected with another social networks. Access information for everyone are stored in socNet table. Privacy settings for each social network and each user are saved in this table.

5.5 Class diagram

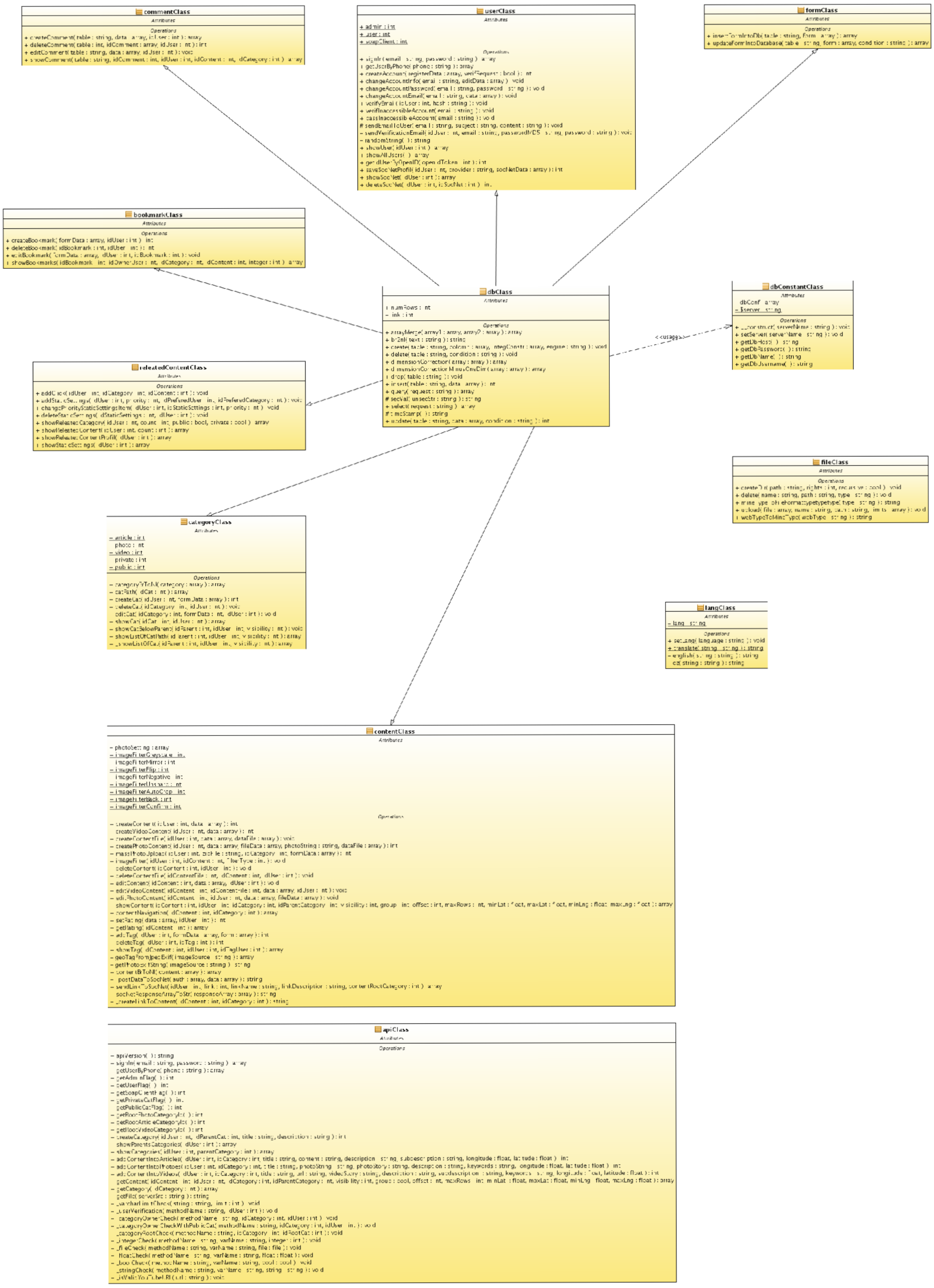
commentClass
Attributes
Operations
+ createComment(table : string, data : array, idUser : int) : array
+ deleteComment(table : int, idComment : array, idUser : int) : int
+ editComment(table : string, data : array, idUser : int) : void
+ showComment(table : string, idComment : int, idUser : int, idContent : int, idCategory : int) : array
userClass
Attributes
+ admin : int
+ user : int
+ soapClient : int
Operations
+ signIn(email : string, password : string) : array
+ getUserByPhone(phone : string) : array
+ createAccount(registerData : array, verifRequest : bool) : int
+ changeAccountInfo(email : string, editData : array) : void
+ changeAccountPassword(email : string, password : string) : void
+ changeAccountEmail(email : string, data : array) : void
+ verifyEmail(idUser : int, hash : string) : void
+ verifInaccessibleAccount(email : string) : void
+ passInaccessibleAccount(email : string) : void
# sendEmailToUser(email : string, subject : string, content : string) : void
– sendVerificationEmail(idUser : int, email : string, passwordMD5 : string, password : string) : void
– randomString( ) : string
+ showUser(idUser : int) : array
+ showAllUsers( ) : array
+ getIdUserByOpenID(openIdToken : int) : int
+ saveSocNetProfil(idUser : int, provider : string, socNetData : array) : int
+ showSocNet(idUser : int) : array
+ deleteSocNet(idUser : int, idSocNet : int) : int
formClass
Attributes
Operations
+ insertFormIntoDb(table : string, form : array) : array
+ updateFormIntoDatabase(table : string, form : array, condition : string) : array
bookmarkClass
Attributes
Operations
+ createBookmark(formData : array, idUser : int) : int
+ deleteBookmark(idBookmark : int, idUser : int) : int
+ editBookmark(formData : array, idUser : int, idBookmark : int) : void
+ showBookmarks(idBookmark : int, idOwnerUser : int, idCategory : int, idContent : int, integer : int) : array
dbClass
Attributes
+ numRows : int
– link : int
Operations
+ arrayMerge(array1 : array, array2 : array) : array
+ br2nl(text : string) : string
+ create(table : string, column : array, integConstr : array, engine : string) : void
+ delete(table : string, condition : string) : void
+ dimensionCorrection(array : array) : array
+ dimensionCorrectionMinusOneDim(array : array) : array
+ drop(table : string) : void
+ insert(table : string, data : array) : int
+ query(request : string) : array
# secVal(unsecStr : string) : string
+ select(request : string) : array
# timeStamp( ) : string
+ update(table : string, data : array, condition : string) : int
<<usage>>
dbConstantClass
Attributes
dbConf : array
– $server : string
Operations
+ __construct(serverName : string) : void
+ setServer(serverName : string) : void
+ getDbHost( ) : string
+ getDbPassword( ) : string
+ getDbName( ) : string
+ getDbUsername( ) : string
releatedContentClass
Attributes
Operations
+ addClick(idUser : int, idCategory : int, idContent : int) : void
+ addStaticSettings(idUser : int, priority : int, idPreferedUser : int, idPreferedCategory : int) : void
+ changePriorityStaticSettingsItem(idUser : int, idStaticSettings : int, priority : int) : void
+ deleteStaticSettings(idStaticSettings : int, idUser : int) : void
+ showReleatedCategory(idUser : int, count : int, public : bool, private : bool) : array
+ showReleatedContent(idUser : int, count : int) : array
+ showReleatedContentProfil(idUser : int) : array
+ showStaticSettings(idUser : int) : array
fileClass
Attributes
Operations
+ createDir(path : string, rights : int, recursive : bool) : void
+ delete(name : string, path : string, type : string) : void
+ mimeTypeOfFileFormattypetypetype(type : string) : string
+ upload(file : array, name : string, path : string, limits : array) : void
+ webTypeToMimeType(webType : string) : string
categoryClass
Attributes
– article : int
photo : int
– video : int
private : int
– public : int
Operations
– categoryTrToNl(category : array) : array
– catPath(idCat : int) : array
– createCat(idUser : int, formData : array) : int
– deleteCat(idCategory : int, idUser : int) : void
editCat(idCategory : int, formData : int, idUser : int) : void
– showCat(idCat : int, idUser : int) : array
– showCatBelowParent(idParent : int, idUser : int, visibility : int) : void
– showListOfCatPath(idParent : int, idUser : int, visibility : int) : array
– _showListOfCat(idParent : int, idUser : int, visibility : int) : array
langClass
Attributes
– lang : string
Operations
+ setLang(language : string) : void
+ translate(string : string) : string
– english(string : string) : string
cz(string : string) : string
contentClass
Attributes
– photoSetting : array
– imageFilterGreyscale : int
imageFilterMirror : int
– imageFilterFlip : int
imageFilterNegative : int
– imageFilterUnsharp : int
– imageFilterAutoCrop : int
– imageFilterBack : int
– imageFilterConfirm : int
Operations
– createContent(idUser : int, data : array) : int
createVideoContent(idUser : int, data : array) : int
– createContentFile(idUser : int, data : array, dataFile : array) : void
– createPhotoContent(idUser : int, data : array, fileData : array, photoString : string, dataFile : array) : int
– massPhotoUpload(idUser : int, zipFile : string, idCategory : int, formData : array) : int
– imageFilter(idUser : int, idContent : int, filterType : int) : void
deleteContent(idContent : int, idUser : int) : void
– deleteContentFile(idContentFile : int, idContent : int, idUser : int) : void
– editContent(idContent : int, data : array, idUser : int) : void
– editVideoContent(idContent : int, idContentFile : int, data : array, idUser : int) : void
– editPhotoContent(idContent : int, idUser : int, data : array, fileData : array) : void
showContent(idContent : int, idUser : int, idCategory : int, idParentCategory : int, visibility : int, group : int, offset : int, maxRows : int, minLat : float, maxLat : float, minLng : float, maxLng : float) : array
– contentNavigation(idContent : int, idCategory : int) : array
– setRating(data : array, idUser : int) : int
– getRating(idContent : int) : array
– addTag(idUser : int, formData : array, form : array) : int
deleteTag(idUser : int, idTag : int) : int
– showTag(idContent : int, idUser : int, idTagUser : int) : array
– geoTagFromJpegExif(imageSource : string) : array
– getPhotoExifString(imageSource : string) : string
– contentBrToNl(content : array) : array
– _postDataToSocNet(auth : array, data : array) : string
– sendLinkToSocNet(idUser : int, link : int, linkName : string, linkDescription : string, contentRootCategory : int) : array
socNetResponseArrayToStr(responseArray : array) : string
– _createLinkToContent(idContent : int, idCategory : int) : string
apiClass
Attributes
Operations
– apiVersion( ) : string
– signIn(email : string, password : string) : array
getUserByPhone(phone : string) : array
– getAdminFlag( ) : int
– getUserFlag( ) : int
– getSoapClientFlag( ) : int
– getPrivateCatFlag( ) : int
getPublicCatFlag( ) : int
– getRootPhotoCategoryId( ) : int
– getRootArticleCategoryId( ) : int
– getRootVideoCategoryId( ) : int
– createCategory(idUser : int, idParentCat : int, title : string, description : string) : int
showParentsCategories(idUser : int) : array
– showCategories(idUser : int, parentCategory : int) : array
– addContentIntoArticles(idUser : int, idCategory : int, title : string, content : string, description : string, subdescription : string, longitude : float, latitude : float) : int
– addContentIntoPhotoes(idUser : int, idCategory : int, title : string, photoString : string, photoStory : string, description : string, keywords : string, longitude : float, latitude : float) : int
– addContentIntoVideos(idUser : int, idCategory : int, title : string, url : string, videoStory : string, description : string, subdescription : string, keywords : string, longitude : float, latitude : float) : int
getContent(idContent : int, idUser : int, idCategory : int, idParentCategory : int, visibility : int, group : bool, offset : int, maxRows : int, minLat : float, maxLat : float, minLng : float, maxLng : float) : array
– getCategory(idCategory : int) : array
getFile(serverSrc : string) : string
– _varcharLimitCheck(string : string, limit : int) : void
– _userVerification(methodName : string, idUser : int) : void
– _categoryOwnerCheck(methodName : string, idCategory : int, idUser : int) : void
– _categoryOwnerCheckWithPublicCat(methodName : string, idCategory : int, idUser : int) : void
_categoryRootCheck(methodName : string, idCategory : int, idRootCat : int) : void
– _integerCheck(methodName : string, varName : string, integer : int) : void
– _fileCheck(methodName : string, varName : string, file : file) : void
– _floatCheck(methodName : string, varName : string, float : float) : void
– _boolCheck(methodName : string, varName : string, bool : bool) : void
_stringCheck(methodName : string, varName : string, string : string) : void
– _isValidYouTubeURL(url : string) : void

Figure 5.5: Class diagram

Class diagram describes classes which are used in application. Diagram also includes name of class methods and inheritance between classes. The purpose of inheritence is an availibility of all methods which handles database queries (from database layer) for methods in web core layer.

5.6 Model of web communication

Web application will be connected with social network through social network API. Application core will allow to present data in web application and also will provide data for mobile application throught multiplatform web API.

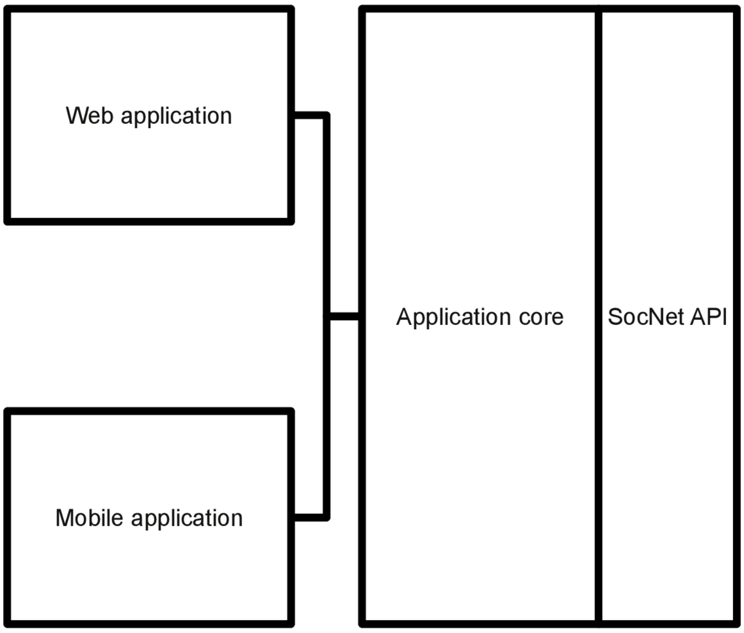

Figure 5.6: Main communication scheme

5.7 Design layers

Application is sliced into undermentioned layers. Each layer represents one package of web functionality.

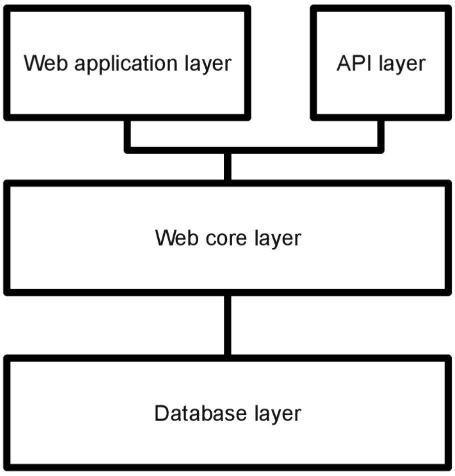

Figure 5.7: Application logical layers

5.8 System for related contents offering

System for related contents offering is inspired by routing protocols. We can define statically preferred categories or we can use our dynamic profil. This system is same as a main idea of static and dynamic routing. Static settings has bigger priority than settings according to user dynamic profil. Static settings can be divided into cascade by the priority value. Dynamic profil is created by the tracking of user's click on the contents or categories. It is the way how we can offer the most interesting categories for user.

Just now we know which content categories are the most interesting for particular user. If we are interested in related content we can extract the content of these categories. By combination of user tagging system, tagging of visited and unvisited contents and content timestamps we can obtain the most interesting content for particular user profil.

Chapter 6

Implementation

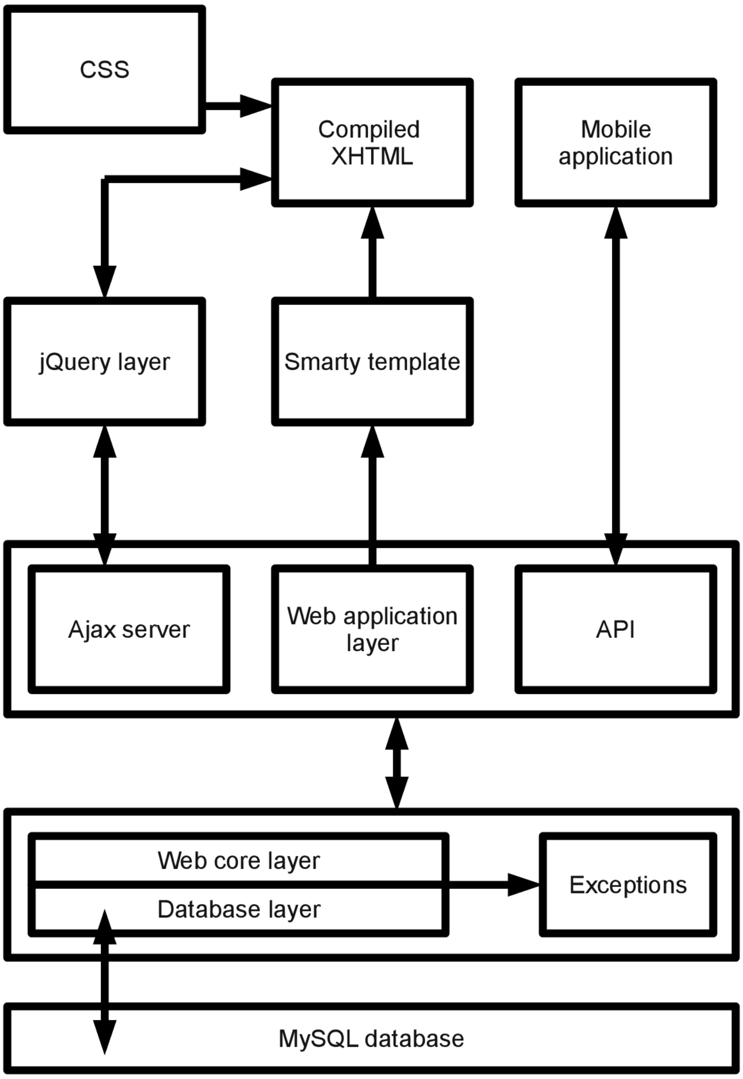

Figure 6.1: Application model

6.1 Install script

Install script assists with web application installation process. This process is divided into a few steps. Installation is realised with install guide and steps can be arbitrarily skiped. Basic installation process includes these steps - drop database tables, create database tables, insert admin account and main categories, finish installation. Database design situated in database layer is closely connected with intall script and therefore all operations are automatically executed by the database design.

Third installation step is important for the Content offering system. Three main categories are photos, videos and articles. These categories are further divided on many subcategories which specify narrower range of parent categories. For example, photos are split into Abstraction, Advertising, Humor, Nature & Architecture, People and etc. These subcategories also includes many subcategories. Result of this hierarchical system is a way how to recognise favourite content for users.

Install script also includes option to display actual database design. This extract contains information about which MySQL engine is used, integrity constractions and design of columns as well as default values and data type specification.

Moreover the script enables to delete system logs throught one click option.

```
[socNet]  => Array
    (
        [column] => Array
            (
                [idSocNet] => INT NOT NULL AUTO_INCREMENT
                [idUser] => INT NOT NULL
                [provider] => ENUM('google', 'myspace',
                   'facebook', 'twitter') NOT NULL
                [tokenOpenId] => VARCHAR(255) DEFAULT NULL
                [tokenOAuth] => VARCHAR(255) DEFAULT NULL
                [sharePhoto] => TINYINT(1) DEFAULT 0
                [shareVideo] => TINYINT(1) DEFAULT 0
                [shareArticle] => TINYINT(1) DEFAULT 0
            )

        [integrityConstraints] => Array
            (
                [0] => PRIMARY KEY (idSocNet)
                [1] => FOREIGN KEY (idUser) REFERENCES users
                   (idUser) ON DELETE CASCADE
                [2] => UNIQUE KEY (tokenOpenId)
                [3] => UNIQUE KEY (tokenOAuth)
            )

        [engine] => INNODB
    )
```

Code 6.1: Extract of database design with install script

6.2 MySQL database

Database runs on MySQL engine InnoDB which supports integrity constraints. These constraints keep database still in a consistent state.

MySQL database does not contain any binary data. All files are stored on server and are connected with database using relative paths. File is named by primary key of database record. Record includes needed information like filename extension, description keywords etc. For more information look at database design.

6.3 Database layer

The database layer handles all database queries like insert, update, delete, select, create, drop, connect or disconnect. Layer provides abstraction for all MySQL queries. There is only one way how we can access database in our application named the Database layer.

Database layer is services with two main classes. dbConstClass provides shadowing for MySQL server connection data. The class can contain more than one description of connection. We can store localhost settings and remote server settings. Later we specify which of these settings will be used for actual connection. dbClass is a backbone class designed for handling all SQL queries. dbClass includes design of database tables. Operations above database are thanks to this feature really simple. Database connetion is automatically done when instance of dbClass is created and disconnected while dbClass instance is destructed.

Database layer treats security issues arising from security attacks Cross-site scripting and SQL injection.

Standard PHP function mysql_query is extended by exceptions and all database queries can use exceptions. By combination of database layer and exception layer (dbException class) we will get powerful tool designed to error handling, debugging or system logs.

6.4 Webcore layer

Webcore layer is the layer which encapsulate web functionality. This layer also provides necessary classes for handling data flow targeted to database or server. Layer is designed to limits user permissions and make application core more secure. Combination of dbLayer functionality and webcore layer slave classes can be used as Framework for communication between database and html forms, and core reports translation.

Classes which handle main web functionality are userClass, contentClass, categoryClass, relatedContentClass, bookmarkClass and commentClass. Slave classes servising core are formClass, langClass and fileClass.

The userClass contains functionality for user account. Methods service functionality for signing in, account creation and other functions necessary for management of user profil. The methods resolve problems with user email verification and other security tasks connected with user registration. After registration the email request which includes random password and verificatin link, is send to user email. There is no way how to use another email address with the aim to create a new user account than address which is owned by user.

The contentClass is the backbone class for contents handling. User can create, edit, delete and present contents stored in database, tag users on contents, evaluate content throught this class. The class contains wide functionality for pictures. Photo methods use

PEL. PHP Exif Library(PEL) is integrated into core and allows to read EXIF meta data. Data describing camera setting (focal lenght, exposure time, fnumber etc.) and coordinates are derivated from this information. Automatically geo tagged will be a picture which includes coordinates in the EXIF meta data. We can also use picture editing method which apply selected filter on picture. Available filtres are unsharp, greyscale, autocrop, flip, mirror and negative. All filters are powered by WideImage library which is encapsulated into core. Library also covered loading, resizing and saving picture as well as cross-format conversion. The class resolve access permission to content. Each user can administrate only his own content. MassPhotoUpload function is an easier way how to upload more pictures per one click. This function uses PHP Zip Functions and enables to upload pictures packed in the zip file. Web picture will have the same name as a picture file packed in the zip file.

The relatedContentClass covers a functionality for related contents offering. Offering principle is based on the multilayer model. Firstly, unvisited data where user is tagged are presented to him. Secondly, unvisited data are offered to user according to user static setting options which are ordered according to priority. Dynamic profil which contains the most visited categories is used at the end and therefore profile is offered unvisited content from these categories. The relatedContentClass handles all methods which are necessary for static setting management or for dynamic profil generating and dynamic profil displaying. User behavior monitoring is also encapsulated in this class. Methods for content offering are constructed on SQL queries therefore are powerful.

The bookmarkClass allows to user manage bookmark system of his profil. Bookmark system is created for storing quick links on the user favourite contents, categories or users.

By commentClass user can comments every content or category. Class is designed universally. By using this class system can create, edit and delete comments independently on name of database columns.

The fromClass is from a slave category. Methods from this class serves all transfers between database and web forms. Class allows data transfer from web forms automatically to database. There is only one requiremet that the name of each variables must be same as database column name otherwise data are dropped. We can insert data into database or update data in database. If we want to update data we have to specify by conditions which rows we would like to update.

We can use another slave class. The langClass provides possibility to translate messages, errors or warnings from web core. The web core includes standartly english and czech dictionary. If we want to use another language package we will have to add new language package into langClass. English is the language for each dictionary and target language have to be defined. Translate method is static. Method is called independently on instance of langClass. Selected target language is stored in static private variable. Therefore language selection is valid during all life of langClass instance or until any changes of this variable. Variable is private that is why we can set up value only with public function setLang.

The fileClass is not handling main web functionality but likewise langClass is only a slave class. Database does not contain blob data type. Therefore all files are stored in server directories. That is saving size of MySQL database. Class enables to uploads files on server, deletes files from server, creates directories and also handles directory permissions. The class includes methods for transferring file mine type into filename extension and back.

6.4.1 How relatedContentClass works

As was mentioned in chapter before relatedContentClass executes functionality designed with purpose of content offering. The biggest part of this functionality is covered by method showRelatedContent. Method further uses method showRelatedCategory. Thereinafter flowchart describes how this method works.

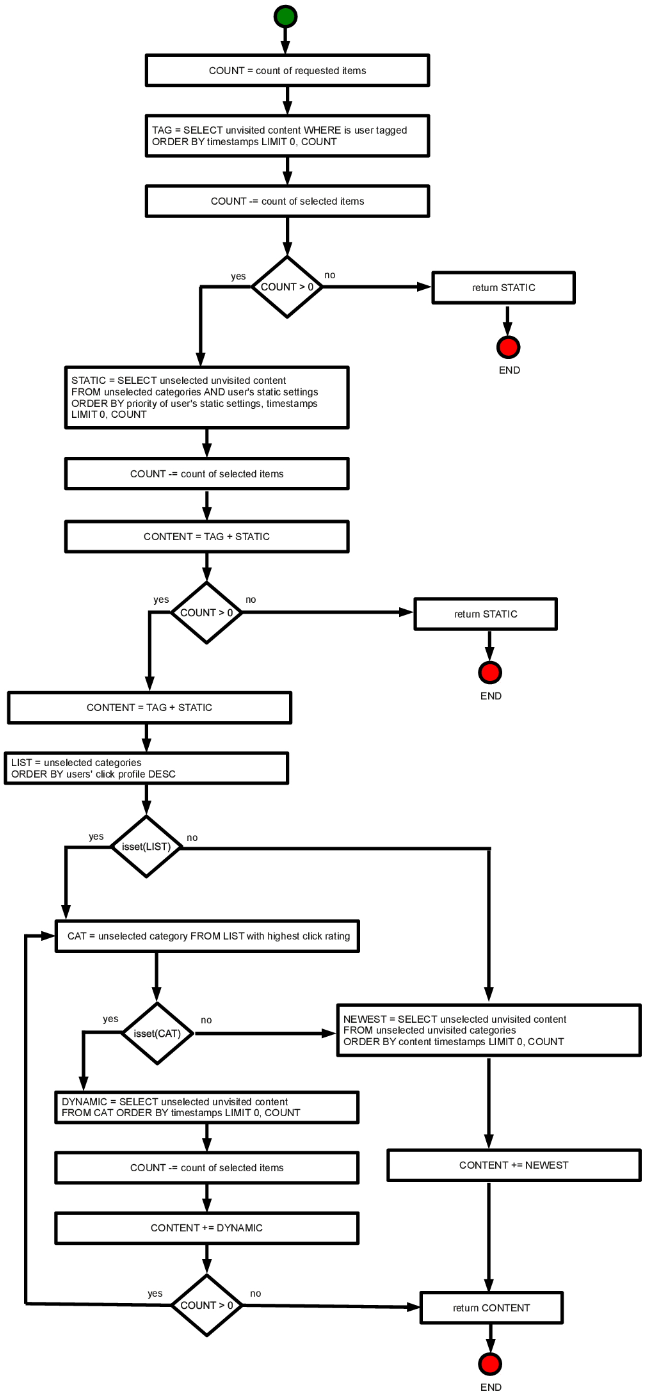
COUNT = count of requested items
TAG = SELECT unvisited content WHERE is user tagged
ORDER BY timestamps LIMIT 0, COUNT
COUNT -= count of selected items
yes
COUNT > 0
no
return STATIC
END
STATIC = SELECT unselected unvisited content
FROM unselected categories AND user's static settings
ORDER BY priority of user's static settings, timestamps
LIMIT 0, COUNT
COUNT -= count of selected items
CONTENT = TAG + STATIC
yes
COUNT > 0
no
return STATIC
END
CONTENT = TAG + STATIC
LIST = unselected categories
ORDER BY users' click profile DESC
yes
isset(LIST)
no
CAT = unselected category FROM LIST with highest click rating
yes
isset(CAT)
no
NEWEST = SELECT unselected unvisited content
FROM unselected unvisited categories
ORDER BY content timestamps LIMIT 0, COUNT
DYNAMIC = SELECT unselected unvisited content
FROM CAT ORDER BY timestamps LIMIT 0, COUNT
COUNT -= count of selected items
CONTENT += NEWEST
CONTENT += DYNAMIC
yes
COUNT > 0
no
return CONTENT
END

Figure 6.2: Flowchart for the method showRelatedContent

Moreover class provides a way how to offer categories by the user's static preferences and by the model of user's behavior. This functionality is implemented in method showRelatedCategory. Thereinafter flowchart describes how the method works.

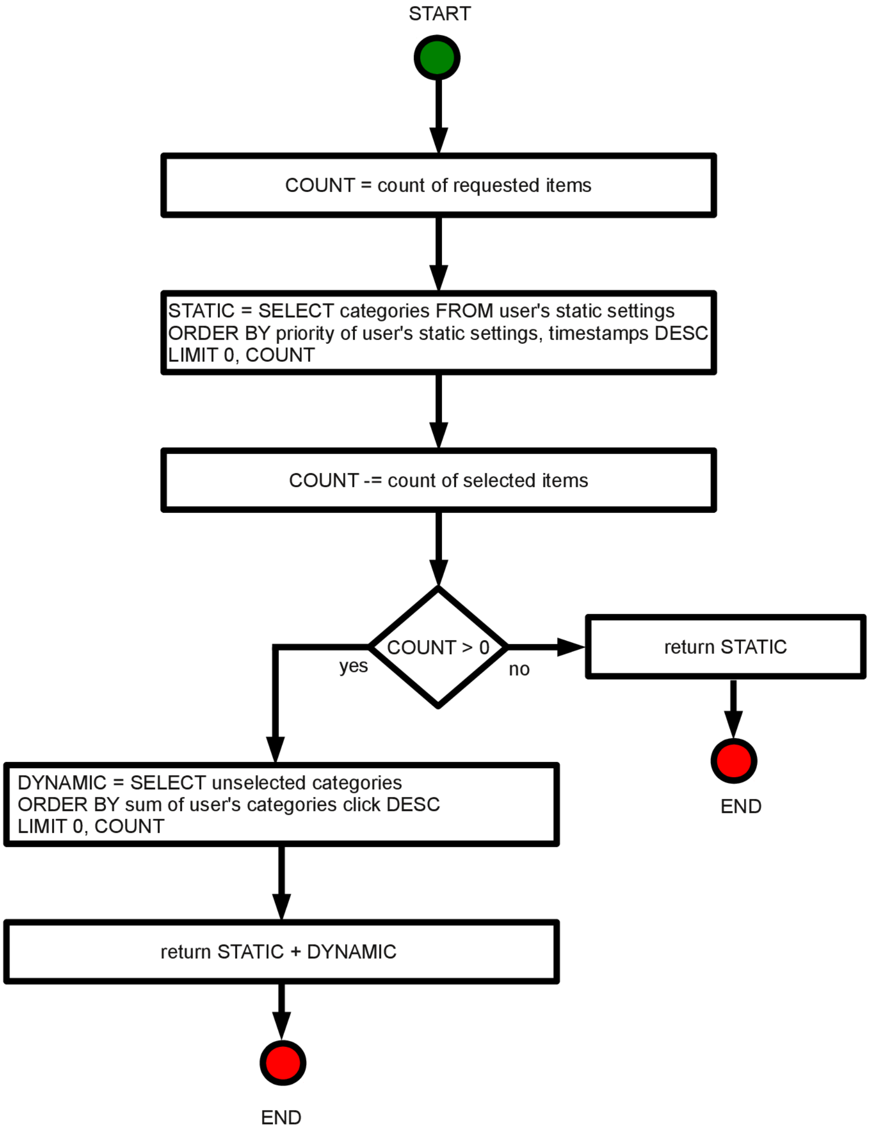
START
COUNT = count of requested items
STATIC = SELECT categories FROM user's static settings
ORDER BY priority of user's static settings, timestamps DESC
LIMIT 0, COUNT
COUNT -= count of selected items
COUNT > 0
yes
no
return STATIC
END
DYNAMIC = SELECT unselected categories
ORDER BY sum of user's categories click DESC
LIMIT 0, COUNT
return STATIC + DYNAMIC
END

Figure 6.3: Flowchart for the method showRelatedCategory

6.5 Exceptions

Exceptions are really powerful tool. Database layer, Web core layer and API are constructed on Exceptions. For each of these program units is defined separate exception class. These exception classes inherit all method from Exception class which is written standardly in

PHP5 and extend this class for a several new methods. Methods handling logs and extensible error messages. Developer allows to use debugging system for every program unit separately as well as system of logs.

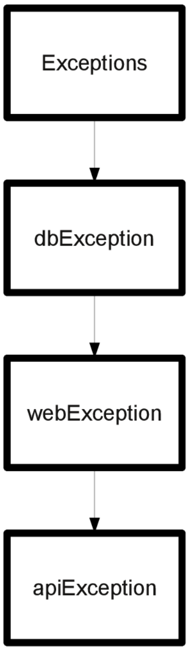

Figure 6.4: Inheritance of exceptions

6.6 API (Application programming interface)

API provides web services. Accessible methods are encapsulated in apiClass. The class includes methods for data uploading and dowloanding. Class resolve problem with mine file type transfer through web service. These data are transfered as 64base encoded binary data. Web content can be filtered with many parametrs like geo coordinates, categories, owner, listing etc. API server is secured by HTTP authentication and API is accessible only for admin or API user permissions. API server use SOAP (Simple Object Access Protocol). Description of operations offered by the service is written in the WSDL (Web Services Description Language). This document is available online and link is situated in a web footer. Developer can use API documentation which describes method calls, description how estabilish communication and examples of API calls in PHP. The documentation is available in html or pdf format. Link is also situated in the web footer.

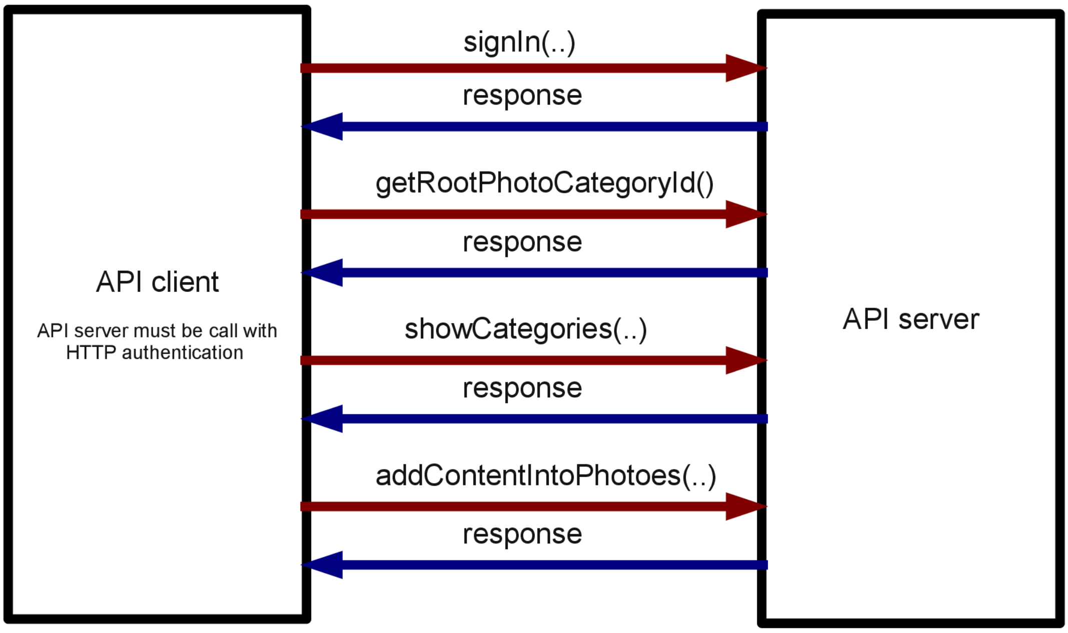

Figure 6.5: Example of API call with the aim to insert a picture

6.7 Web application layer

The web application layer is responsible for data transfers from XHTML forms to webcore layer and also responsible for assign webcore data to templates which are written in Smarty. Data transfers to webcore are realised with HTTP requests or AJAX requests. Direction from web core is executed with Smarty function assign.

The majority part of application logic requested by web core layer is situated in web application layer otherwise in templates.

Every file which handles individual web pages includes common part. This file is called common.php. This file contains Smarty settings, login and logout management, create instances of needed objects, define constants, handle warning and errors messages sourced from web core layer and furthermore assign related menu for every user individually.

File publicCategories.php, usersCategories.php and wall.php are designed for content presentation but each of them for different content. Therefore was designed file called contentGallery.php. The file encapsulate application logic and moreover with a template contentGallery.tpl solve presentation for any content. Before we include file contentGallery we have to set up variable categoryType and optionally variable idPresentedUser. We can hereby use content gallery for any content.

Messages from web application layer can be translated with langClass which make this layer multilanguage.

From the security vulnerability point of view this layer is still security threat. The layer protects web application against security vulnerabilities called Clickjacking, session fixation and partially CSRF. These topics are discussed in chapter Seccurity vulnerability.

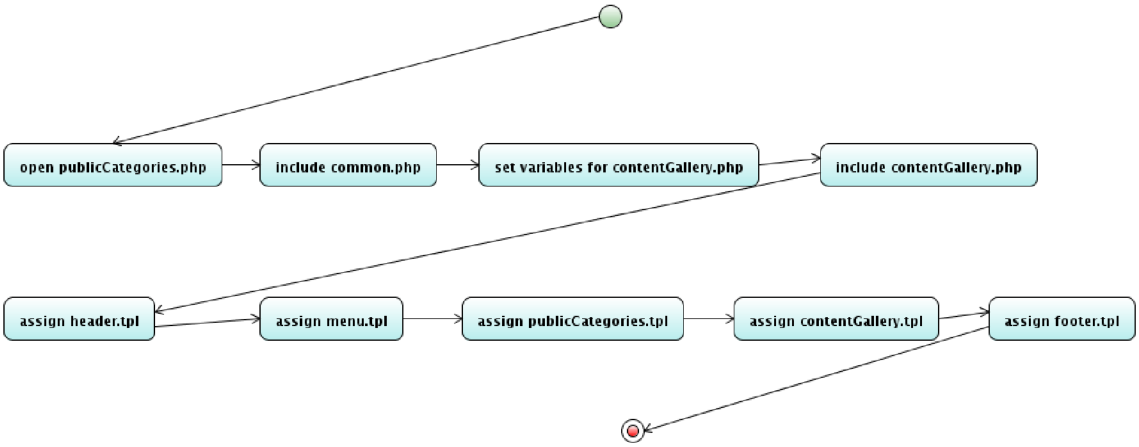
open publicCategories.php
include common.php
set variables for contentGallery.php
include contentGallery.php
assign header.tpl
assign menu.tpl
assign publicCategories.tpl
assign contentGallery.tpl
assign footer.tpl

Figure 6.6: publicCategories.php compilation and process how are files included

6.8 Ajax server

Server is designed to handle all requests from ajax client which is written in jQuery (client will be discussed in chapter jQuery layer). Ajax server receives data in POST requests. On success, server returns JSON encoded string otherwise on fault returns JSON encoded string including error message under associative index 'response'.

During request processing the request is switched into appropriate handler by the name of submit button in XHTML.

6.9 jQuery layer

jQuery layer is built on jQuery framework and represents really powerfull instrument for data representation, form validation and moreover Ajax services. Layer is created instead of simply JavaScript layer thus we can use higher level of abstraction above simple JavaScript code. Therefore web uses many plugins which provide wide functionality. Web uses these plugins: Superfish [16], Galleriffic [34], Form plugin [5], gMaps [2], jYoutube [35], Validation plugin [18], jQuery UI [20].

6.9.1 Windows system

System of windows is built on jQuery UI. System provides abstraction above jQuery UI dialog widget.

User does not have to refresh a page because window is hidden on the page and waiting for open call. Windows are displayed above actual page without wasting contents on actual page. As a result is a simple and user-friendly navigation system.

Windows which contain form can be combine with Ajax communication with the aim to create user-friendly web application.

Window is a HTML DIV element that belongs to class window and the window name is specified in a div identificator. By default is the window closed. The window can by opened with button which belongs to class showWindow and every individual window is pointed through the button value. The value must be same as a identificator of the div element encapsulating window. Window title is same as a value of the button element. Windows can be closed by button which belongs to class hideWindow. For developers there is also prepared a way how to call window containing form with initial values for this form. Button with action show window is packed into a form element. This form also includes hidden input elements where name is same as name of requested input element in pop-up window. Initial value is defined by the value of the hidden input element. This form must be caught by JavaScript code. This code must call a function showWindowWithParam with argument this.

6.9.2 Ajax client

Client is built with jQuery form plugin. Client handle all requests heading to Ajax sever as well as all his responses. Developer can display all client requests from HTML forms to AJAX server. We can use this feature for application debugging.

Ajax client contains functions which can create, edit or delete elements from HTML document in real time without reloading. Due to this gadget is final application more user-friendly.

6.9.3 gMaps

Into default gMaps jQuery plugin was integrated function for getting coordinates. Only one click on google maps will return latitude and longitude value from place where user clicked. This gadget and geolocation from EXIF meta data create system of content geolocation.

The plugin is also used for presentation of content coordinates.

6.9.4 jYouTube

In database is stored only a link to video. Plugin jYouTube returns video screenshot which is under the link.

6.10 Smarty template

Smarty is not used only as a template system which divides presentation and application logic. In smarty templates there is also situated part of application logic. For example smarty functions allow us to present edit button on place where it can be used by user with view to his permissions.

Smarty divides web into four main templates which are compiled together.

First file is a header template. The header contains common part of web pages like includes of extern CSS and JavaScript files, doctype information, main google analytics JavaScript code and div for handling messages or errors.

Second file is a menu template. This template contains web navigation and distinguishes user permissions for menu items.

Third template carries every time different presentation. This is not only one template but it is a collection of templates which are very diverse.

Fourth template file is a footer. Template which is common for every page. There are included close elements for elements open in header template. There is also mentioned common web footer.

6.11 CSS

Cascading Style Sheets (CSS) is situated in extern files. Some of these files are designed for formating jQuery plugin outputs but other determine main web design. Web application use jQuery UI CSS framework [20]. As a result is an application where theme changes are unbelievably simple. For application was created custom theme with ThemeRoller tool [20]. If developer wants to change web theme he has to only rewrite web files which were generated by ThemeRoller. Web application layout is situated into two files. One of them is designed for print media and second is for screen media.

6.12 Documentation

All core files which holding web functionality are commented with DocBlock comments. With PHPDocumentor [19] are generated two documentations. First includes design of all application core and second is designed only for API. The API documentation contains useful information for developers. In documentations there are also mentioned code samples. Documentations are available in docs folder and API documentation is also accessible under the link situated in a web footer.

6.13 Google Analytics

Google Analytics is connected with our web domain. For every web domain is generated unique account id. Onto every page on our web is added Google Analytics Tracking Code (GATC) which is a hidden JavaScript code. This code collects private visitor data and send it back to Google data collection servers for processing. Variable _setAccount must be set on our account id. This is enough for supporting standard Google Analytics functions.

Our web application uses advanced functions. This functionality is built on Custom Variables. Custom Variables provide us the power to collect the unique site data. Custom variable are divided into three levels of interaction by interaction model used in Google Analytics. Interaction model:

- Visitor - the client who visits the site
- Session - the period of the time during which is the visitor active on site
- Page - activity which sends request to the Analytic server(a pageview, an event)

Custom Variable can be created with method _setCustomVar(index, name, value, opt_scope). Argument index is the slot for the custom variable. This is a number whose value can range 1-5. Argument name is the name for the custom variable. Value is an argument including the value of the custom variable. Last argument is the scope for the custom variable. The scope defines the level of user engagement with our site. Possible values are 1 (visitor-level), 2 (session-level), 3 (page-level).

Our web application uses visitor level scope variable for distinguishing unique users and page level to trace user action in content galleries. With Google Analytics Custom Variables we can verify our system for offering related content.

Data from custom variables will apear in Google Analytics with delay a few hours.

6.14 Application logs

By the system of logs is web application recevering from fault more simply. Each of the main application layer (database, web core and api layer) has own log file. Every log record has a debugging information and information about user. User is described by IP address and by type of web browser. Logs records are signed by the time of fault. Hierarchical logs system makes fault tracking much more efficient. Logs are situated in logs folder.

6.15 Connection with social networks

Connection with social network is based on SocialNetworksAPI which is a part of bachelor's thesis Social Network Integration into an Information Portal[3]. Thesis provides SocialNetworksAPI with class CommonAPI. All of the Adaptive web portal communication with social networks is processed through this API.

Data required by communication between our application and social network are obtained through methods requireSignIn, finishSignIn, requireAccess, getAccess. Methods requireSignIn and finishSignIn are used for third party login. Adaptive web portal has a functionality for synchronisation of your account and third party login services. Result of this synchronisation is an account which can be accessible by many ways - with your application account which is in many cases created automatically, OpenID account, Twitter account etc. List of accessible social networks depends on SocialNetworkAPI.

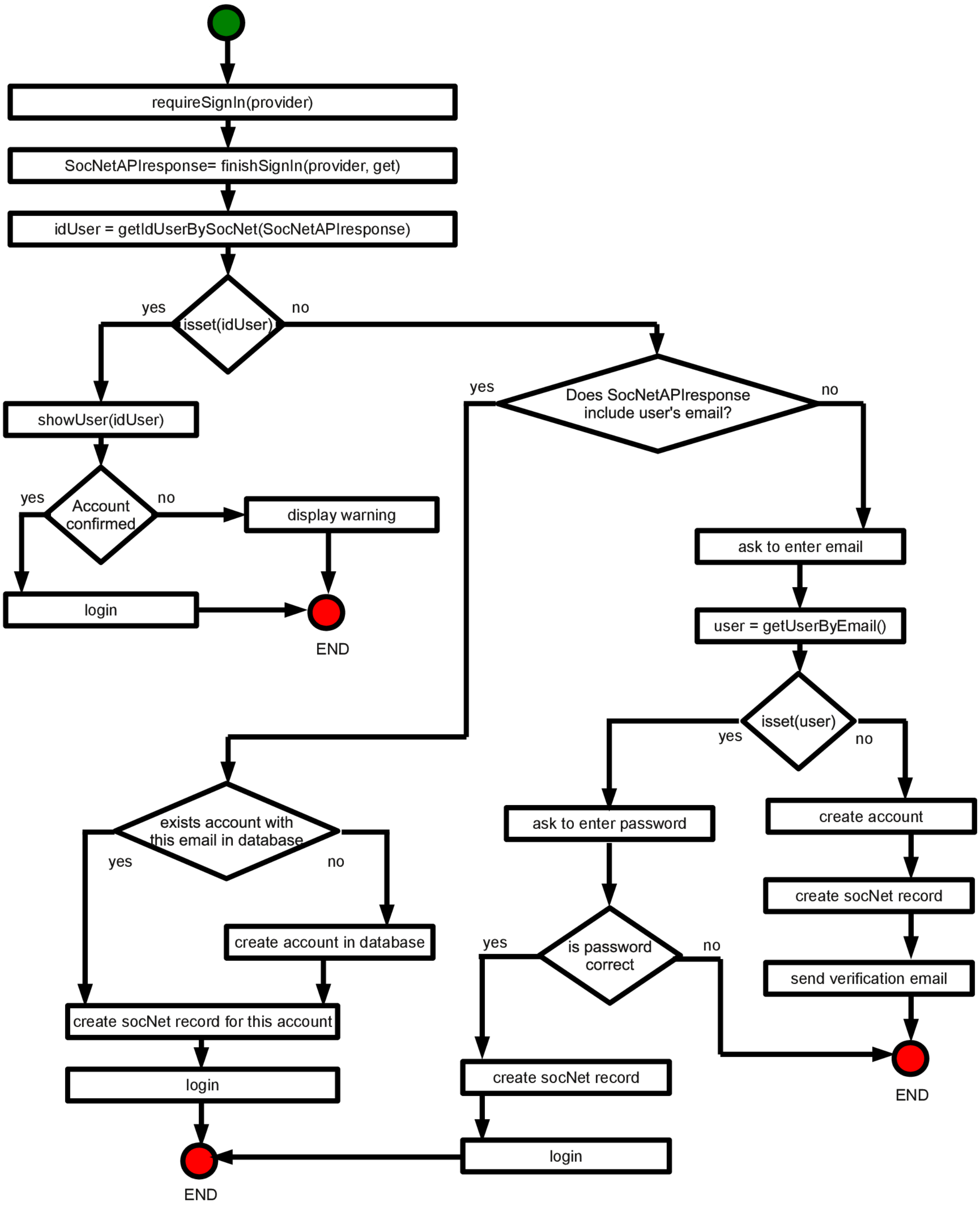

Figure 6.7: Flowchart for a third party login process

Data can be sent to the socialNetworks through postData method from CommonAPI class. Adaptive web portal provides functionality to expands backlinks to user's contents automatically during uploading. Portal keeps user's privacy. Therefore propagation of backlink is off by default. User has rights to set a filter for every main kind of portal contents and for every his social network connection. Therefore his backlinks are expanded under control (by default expand nothing). Backlinks to the content can be also expanded through the button in content gallery. In this case are the posts expanded to social networks which were registered by the user into his account without filtration.

Automatic content propagation to your social networks is included in web core layer. Therefore Adaptive web portal API can be used as a tool which propagates information about contents which was uploaded through mobil application (example of application which uses Adaptive web portal API) to the social networks.

Chapter 7

Tests

With the aim to verify model of the portal user behavior there was created account in Google Analytic (GA). This service was mentioned in chapter Implementation. GA provides to us a way how to simply verify model of behavior for portal users. Monitoring of user behavior is based on storing user's clicks in content gallery. Hereby we can select the most favourite category for particular portal user. In content gallery is situated JavaScript code which sends data to GA. This data describes every user. With GA filter we can get information which describes in which categories user spend the most time or the less time.

Google Analytic script sends data independently on our monitoring system. In comparison with information from GA and from our monitoring system we can verify way of modeling the behaviour of portal user.

Adaptive web portal was presented to the public community. After a few weeks there was collected data which provides us possibilities of comparision between data received from GA and from Adaptive web portal. Order of preferred categories was for almost all of portal users same in both of these systems. Small differences were registrated in older accounts. Reason of these differences was unsimultaneous start of GA and Adaptive web portal monitoring systems.

Conclusion of these tests is that our application offers to users content by the correct model of behaviour.

Example of printout is mentioned in appendix Comparison of Google Analytics profile and dynamic user profile generated by Adaptive Web Portal.

Chapter 8

Conclusion

8.1 Own contribution

Our application enables to work with many types of content. Thesis submission was extended for many type of contents instead of one selected type. Inbound content can be uploaded in many ways. System for content offering was independenlty verified with tools provided by Google Analytis. Result of this test was mentioned in chapter Test.

The submission of thesis was complied in full. Moreover, it was extended for application programming interface, web was connected with social networks, gMaps, youTube and with Google Analytis. Thesis is a part of project at UTAD University. The project enables users to distribute their content into many social networks uniformly through mobile or web platform. Main idea of this project was to create system of emergency messaging based on geolocation. Web provides this functionality and adds features of Adaptive web portal application.

8.2 Future expansion

For the future expansion, I propose that for the API authorization is better to utilise OAuth protocol instead of HTTP authentication. To inovate type of API for modern REST architecture and use existent user account as an OpenID account. Therefore, we use Adaptive web portal application as an OpenId provider.

Proposals of improving system for related content offering is use to public user information provided by social networks to improve his dynamic profil. In the future, the offering system can be closelly connected with the tagging system. I prefer to connect geotagging data encapsulated in content with W3C geolocation API with the aim to offer to user content from his geolocation area. This option can be also implemented by third-party application because the Adaptive web portal API supports this type of content filtration.

Bibliography

[1] Derick Rethans Andi Gutmans, Stig Saether Bakken. *Mistrovství v PHP 5*. Computer Press, 04 2007. ISBN 978-80-251-1519-0.

[2] Cedric Kastner. gmap - google maps plugin for jquery. `http://gmap.nurtext.de/documentation.html`, 2009-06-11 [cit. 2010-06-25].

[3] Drahomíra Herrmannová. Social network integration into an information portal. Master's thesis, FIT VUT v Brne, 2010.

[4] Ed. E. Hammer-Lahav. The oauth 1.0 protocol. `http://tools.ietf.org/html/rfc5849`, 2010-04 [cit. 2010-06-28].

[5] Form Plugin community. jquery form plugin. `http://jquery.malsup.com/form/`, 2010-03-12 [cit. 2010-06-25].

[6] Gašper Kozak. Wideimage - an open-source php library for image manipulation. `http://wideimage.sourceforge.net/documentation/`, 2010-02-08 [cit. 2010-06-16].

[7] Google. Developer's guide - google analytics - google code. `http://code.google.com/apis/analytics/docs/gdata/gdataDeveloperGuide.html`, 2010 [cit. 2010-06-29].

[8] Google. Google analytics — official website. `http://www.google.com/analytics/features.html`, 2010 [cit. 2010-06-29].

[9] Eran Hammer-Lahav. The authoritative guide to oauth 1.0. `http://hueniverse.com/oauth/guide/`, 2009-12 [cit. 2010-06-28].

[10] Jakub Vrána. Obrana proti sql injection. `http://php.vrana.cz/obrana-proti-sql-injection.php`, 2005-03-02 [cit. 2010-06-21].

[11] Jakub Vrána. Zabezpečení session proměnných. `http://php.vrana.cz/zabezpeceni-session-promennych.php`, 2005-07-01 [cit. 2010-06-21].

[12] Jakub Vrána. Cross-site request forgery. `http://php.vrana.cz/cross-site-request-forgery.php`, 2006-04-24 [cit. 2010-06-21].

[13] Jakub Vrána. Cross site scripting. `http://php.vrana.cz/cross-site-scripting.php`, 2009-03-06 [cit. 2010-06-21].

[14] Jakub Vrána. Clickjacking. http://php.vrana.cz/clickjacking.php, 2009-10-07 [cit. 2010-06-21].

[15] Jeremiah Grossman. Whitehat website security statistics report. Technical report, White Hat SECURITY, 10 2010 [cit. 2010-06-29].

[16] Joel Birch. jquery menu plugin - superfish. http://users.tpg.com.au/j_birch/plugins/superfish/, 2010-03-24 [cit. 2010-06-25].

[17] Karl Swedberg Jonathan Chaffer. *Learning jQuery 1.3.* Packt Publishing, 02 2009. ISBN 978-1-847196-70-5.

[18] Jorn Zaefferer. jquery validation plugin. http://docs.jquery.com/Plugins/Validation, 2010-09-03 [cit. 2010-06-25].

[19] Joshua Eichorn. phpdocumentor guide to creating fantastic documentation. http://manual.phpdoc.org/, 2008-03-31 [cit. 2010-06-25].

[20] jQuery project and the jQuery UI team. jquery ui. http://jqueryui.com/, 2010-03-06 [cit. 2010-06-25].

[21] Martin Geisler. Pel: Php exif library. http://pel.sourceforge.net/doc/, 2008-06-08 [cit. 2010-06-16].

[22] Monte Ohrt, Andrei Zmievski. Smarty - the compiling php template engine. http://www.smarty.net/manual/en/, 2007-09-17 [cit. 2010-06-18].

[23] OpenID Foundation. Openid foundation website. http://openid.net/, [cit. 2010-06-28].

[24] Oracle Corporation and/or its affiliates. About mysql. http://www.mysql.com/about/, 2010 [cit. 2010-06-18].

[25] Oracle Corporation and/or its affiliates. Mysql 5.1 reference manual. http://dev.mysql.com/doc/refman/5.1/en/, 2010 [cit. 2010-06-18].

[26] Refsnes Data. Introduction to wsdl. http://www.w3schools.com/wsdl/wsdl_intro.asp, [cit. 2010-06-18].

[27] Refsnes Data. Soap introduction. http://www.w3schools.com/soap/soap_intro.asp, [cit. 2010-06-18].

[28] Robert Auger. Cross site request forgery. http://projects.webappsec.org/Cross-Site-Request-Forgery, 2009-12-30 [cit. 2010-06-21].

[29] Robert Auger. Php cross site scripting. http://projects.webappsec.org/Cross-Site-Scripting, 2009-12-30 [cit. 2010-06-21].

[30] Robert Auger. Session fixation. http://projects.webappsec.org/Session-Fixation, 2009-12-30 [cit. 2010-06-21].

[31] Robert Hansen. Clickjacking. http://www.sectheory.com/clickjacking.htm, 2008-12-09 [cit. 2010-06-21].

[32] Ryan Barnett. Sql injection. http://projects.webappsec.org/SQL-Injection, 2010-06-14 [cit. 2010-06-21].

[33] The jQuery project. jquery: The write less, do more, javascript library. http://docs.jquery.com/Main_Page, 2010-03-24 [cit. 2010-06-25].

[34] Trent Foley. Galleriffic — a jquery plugin for rendering fast-performing photo galleries. http://www.twospy.com/galleriffic/, 2009-12-19 [cit. 2010-06-25].

[35] UZBEKJON. jquery youtube plugin. http://plugins.jquery.com/project/jyoutube, 2009-03-15 [cit. 2010-06-25].

[36] W3C HTML Working Group. Xhtml 1.0: The extensible hypertext markup language (second edition). http://www.w3.org/TR/xhtml1/, 2002-08-01 [cit. 2010-07-16].

[37] W3C HTML Working Group. Cascading style sheets level 2 revision 1 (css 2.1) specification. http://www.w3.org/TR/CSS21/, 2009-09-08 [cit. 2010-07-16].

[38] Wikimedia Foundation, Inc. Session hijacking. http://en.wikipedia.org/wiki/Session_hijacking, [cit. 2010-06-21].

[39] Zend Framework community. Programmer's reference guide 1.10, zend soap. http://framework.zend.com/manual/en/zend.soap.html, [cit. 2010-06-16].

[40] Zend Technologies Ltd. About zend framework. http://framework.zend.com/about/, [cit. 2010-06-16].

Appendix A

Content of CD

- Source code of Adaptive web portal (includes all linked libraries)
  - /portal
- Documentation generated by phpDocumentor
  - /portal/docs/all/html/index.html
  - /portal/docs/all/pdf/documentation.pdf
  - /portal/docs/api/html/index.html
  - /portal/docs/api/pdf/documentation.pdf
- Technical report (includes TEX source code)
  - /technicalReport/thesis.pdf
  - /technicalReport/thesisSourceCode.tar.gz

Appendix B

Web installation and configuration

B.1 Installation

Installation process is guided by install script. Before the script is used you have to configure a few settings.

1. Upload source file to server. Server must comply with the web requirements. Requirements are found on the web wiki or in section bellow.
2. Add your database settings into variable dbConf situated in dbConstClass
3. Set up dbClass for your new setting. Rewrite in constructor of dbClass parameter called by constructor of dbConstClass. Another option is call method setServer from dbConstClass before function mysql_connect. As a parameter must be used name of your new setting collection.
4. Run script install.php and follow the instructions.
5. Run web application.

B.2 Optional installation options

- To set web core language you must rewrite parametr on line $language-¿setLang('en') in file common.php. Available dictionaries are included in langClass.
- To change image size on the output of image resizing process you have to edit variable $photoSetting in contentClass.
- To switch over debugging information or system logs you must edit variables $debug and $logs in dbException.

B.3 Web requirements

- PHP 5.3.1 with calendar enable, soap server enable, zip enable
- PHP $<$ 5.3.1 with calendar enable, soap server enable, zip enable, save_mode off
- MySQL with engine InnoDB

- filer grayscale, negative etc available only if PHP is compiled with the bundled version of the GD library. (phpinfo() GD Version bundled)

Appendix C

Examples of API calls

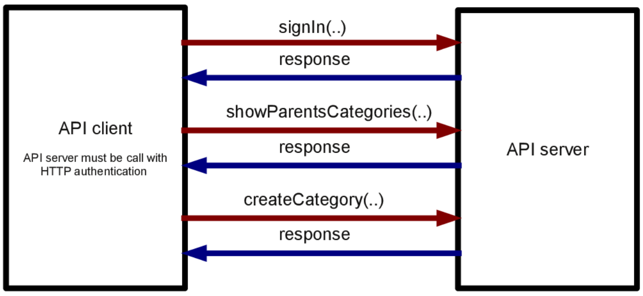
signIn(..)
response
showParentsCategories(..)
response
createCategory(..)
response
API client
API server must be call with HTTP authentication
API server

Figure C.1: Example of API call with the aim to create subcategory in my own category tree

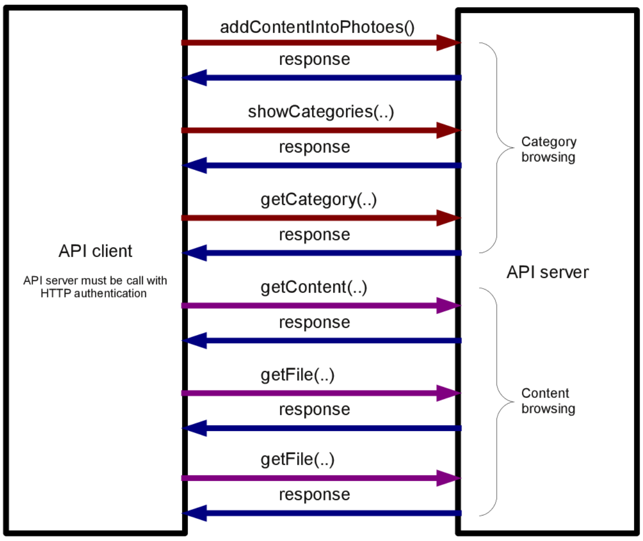

Figure C.2: Example of API call with the aim to browse in public or own categories and content

Appendix D

Comparison of Google Analytics profile and dynamic user profile generated by Adaptive Web Portal

Hereby we can compare both profiles. This is an example of printout for account pacha.adam@gmail.com:

D.1 Google analytics profile

category (category Pageviews)

- Photo - Adaptive web portal (50)
- Photo (18)
- Photo - Abstraction (17)
- Photo - Nature & Architecture - Panorama (15)
- Photo - People - Fashion/Style (10)
- Article - Test for developing (9)

D.2 Adaptive web portal profile

category (category rating)

- Photo - Adaptive web portal (9,87%)
- Photo (3,96%)
- Photo - Abstraction (2,64%)
- Photo - Nature & Architecture - Panorama (2,08%)
- Photo - People - Fashion/Style (0,69%)
- Article - Test for developing (0,28%)