Picture Publisher
Group 15

Fredrik Lundberg flundbe@kth.se
Jakob Nordlander jakobno@kth.se
Dan Wester dwester@kth.se
Per Wiklander pwik@kth.se

Requirements Document

Preface

Version history

12/20/06	Version 1.0	The first version

Expected readership of the Requirements Document

The Requirements Document is written to be read by the stakeholders of the project.

This includes:

- The project team members
- The customer (i.e. at this moment ourselves but if anyone decides to purchase the system they become the customer)
- The project supervisor

To understand the content of this document the reader should have knowledge in the following areas:

- General computer terms (i.e. not the words in the glossary)
- Basic software engineering

Table of contents

Introduction

The system's functions

The functions of the system include:

- **A client for publishing pictures**
  The client allows the user to publish one or more pictures to the system's server. It will be possible to download exact copies of these pictures and this means that the system can be used as an alternative storage and backup as well as for cataloguing of pictures. The pictures can be given keywords for more effective searches.

- **The ability to view pictures given an Internet connection and a browser**
  The system manages user accounts that are required for viewing pictures with access restrictions. Having an account also allows users to attach keywords to other user's pictures.

The need for the system

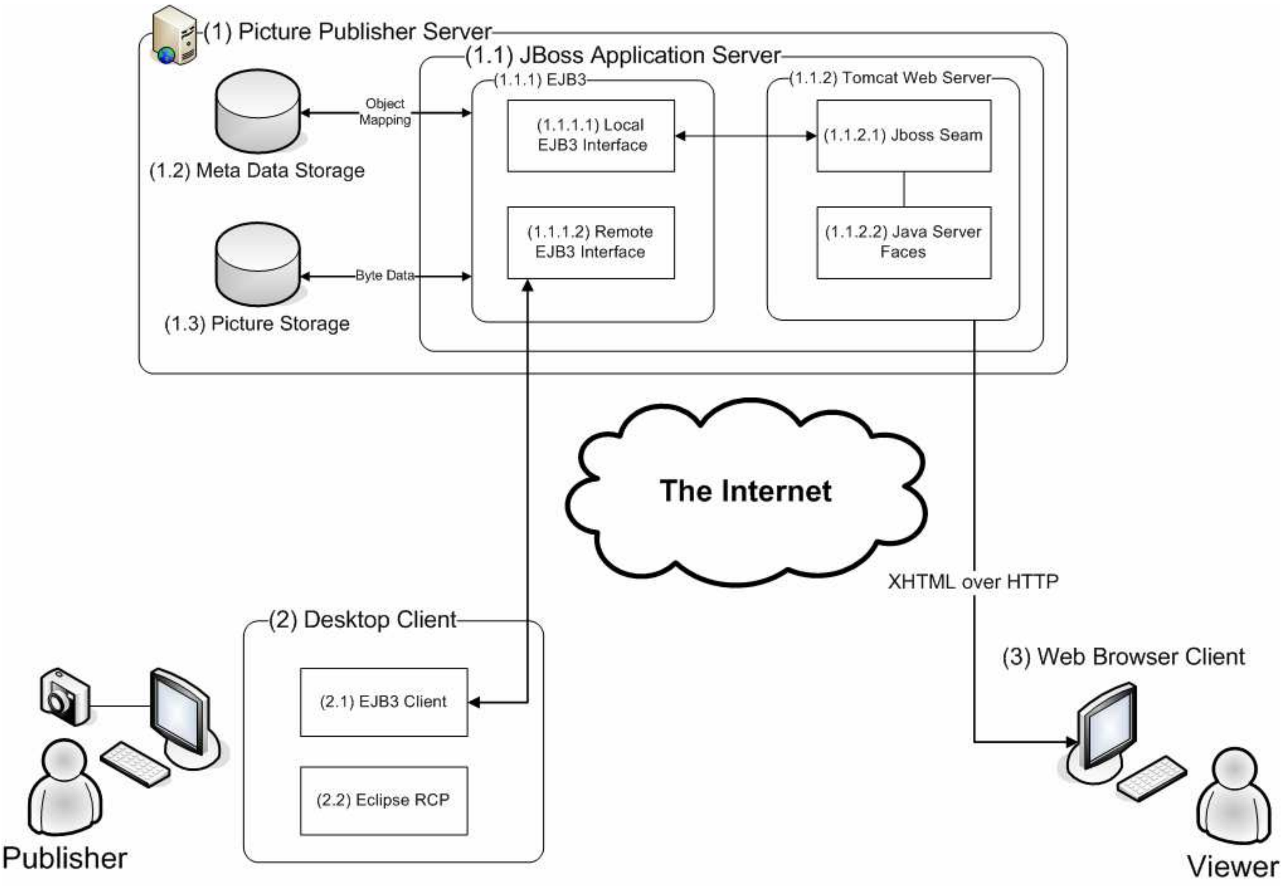
We saw a need for an easy-to-use system to share pictures with your friends. It is possible to publish more than one picture at a time and it is easy to categorize pictures for more effective searches. It is also easy to save the metadata you find appropriate for each picture and change it at a later time if needed. The system can also be useful for people with little or no space left on their hard drives can use our system to store their pictures. Our system makes it very easy to view the pictures someone has published. All you need is a standard web browser and some basic computer knowledge.

Business Objectives

The business objectives for the project are:

- The Picture Publisher should still be running a year after market introduction.
- To learn how a software project is performed.
- For personal use of picture storage.
- To pass the 2D1363 Mjukvarukonstruktion course.

System Architecture



1. Picture Publisher Server

This is the heart of the Picture Publisher system. This is where all picture storage and processing is conducted.

1.1. JBoss Application Server (AS)

“Generally, an application server is a software engine that delivers applications to client computers. Moreover, an application server should handle most, if not all, of the business logic and data access of the application” (Wikipedia).

The reason for us to choose to use JBoss AS instead of one of the other well known application servers like WebLogic (BEA), WebSphere (IBM) or Sun Java System Application Server is, other than the fact that it is open sourced and completely free to use, that it is well supported with a large user community and that the developers at JBoss tend to be on top of the latest and most interesting technologies.

1.1.1. EJB3

“Enterprise Java Bean (EJB) is a managed, server-sided component for modular construction of enterprise applications. The Enterprise JavaBeans specification is one of the several Java APIs in the Java Platform, Enterprise Edition. EJB is a server-side component that encapsulates the business logic of an application. The EJB specification was originally developed in 1997 by IBM and later adopted by Sun Microsystems (EJB 1.0 and 1.1) and enhanced under the Java

Community Process as JSR 19 (EJB 2.0), JSR 153 (EJB 2.1) and JSR 220 (EJB 3.0).” (Wikipedia)

1.1.1.1. Local EJB Interface

Local interfaces are used when EJBs are accessed by other components running in the same AS.

1.1.1.2. Remote EJB Interface

Remote interfaces are used when EJBs are accessed by remote clients running outside the AS. These clients can be run on the same physical computer or on another system anywhere on the internet.

A big difference between local and remote EJB access is that remote method calls pass parameters by reference while remote calls pass by value. This is important to think about since it has a large impact on how objects are treated when they are passed around between client and server.

1.1.2. Tomcat Web Server

"Apache Tomcat is a web container developed at the Apache Software Foundation (ASF). Tomcat implements the servlet and the Java Server Pages (JSP) specifications from Sun Microsystems, providing an environment for Java code to run in cooperation with a web server." (Wikipedia)

1.1.2.1. JBoss Seam

Seam works as a bridge (or seam) between the view layer (provided by JSF) and the business layer (provided by EJB) to simplify the development of web applications. While it is possible to use plain JSF for the view and data bindings, it tends to get quite complicated and cumbersome when the complexity of a web application grows. With a mature framework like Seam, developers can concentrate on the important things like the business logic of the application.

1.1.2.2. Java Server Faces (JSF)

"Java Server Faces (JSF) is a Java-based Web application framework that simplifies the development of user interfaces for Java EE applications. Out of the box, JSF uses Java Server Pages for its display technology." (Wikipedia)

1.2. Metadata Storage

Picture metadata is stored in a data base (PostgreSQL) for easy search and retrieval.

1.3. Picture Storage

The actual pictures are stored as files on disk. Two reasons for this are shorter access times and easier backup.

Functions in this module

Access restrictions
Account creation
Searching
Keeping publisher and viewer keywords separated
Storing pictures
Storing metadata
Removing accounts
Removing a picture
Download restrictions
Upload quota
Personal information access

2. Desktop Client

2.1. EJB Client

The EJB client code connects to a remote EJB interface on the server and handles data and business transactions between the server and the GUI.

2.2. Eclipse Rich Client Platform (RCP)

Eclipse might be most well known as a code editor or integrated development environment (IDE). The objective of the RCP is to enable Eclipse to be used in a wide range of end-user applications that are not IDEs.

Functions in this module

Rotating pictures
Downloading exact copies
Adding metadata
Sorting pictures into albums
Optimization of image quality

3. Web Browser Client

The point with using a web browser as client platform for viewing pictures is to make it as easy as possible for new users to take part of the Picture Publisher experience. Grandma should not have to install any fancy software to watch the holiday snap shots.

Functions in this module

Viewing pictures through a standard web browser

Components that are reused

Since the Picture Publisher is the first software project for this group it will be developed from scratch. However, several tools for communication and connections are used (see further the System Architecture). We will also reuse open source components as much as possible as long as we understand how the code works and feel that we would be able to implement the same functionality ourselves if we really had to.

Functional Requirements

Picture Storage

The system shall store the pictures on the server, it shall store exact copies of the pictures published. This shall be done through the client.

Rationale: This will make it possible for the user to use our system as alternative file storage for his pictures.

Picture Downloading

The user shall be able to download exact copies of the pictures previously published. This shall be done through the client

Rationale: To be able to download exact copies of the pictures will make it possible for the publisher to use our system as a way of storing their pictures on our server instead of using their own hard drive.

Rotating pictures

The user shall be able to **rotate (clockwise or counter clockwise) the pictures** inside the client before publishing them.

Rationale: All pictures are not rotated correctly by default and this function will help the publisher to rotate the pictures correctly so that the viewers do not have to tilt their heads.

Adding Keywords to Pictures

Any user shall be able to **attach keywords to each picture**.

Rationale: The user uses keywords to describe the pictures so that they are easier to find.

Sorting Pictures

The publisher shall be able to **sort the pictures into albums**.

Rationale: Keeping pictures in albums helps the publisher to organize the pictures.

Access Restrictions

The publisher shall be able to **restrict** who will be allowed to watch different pictures.

Rationale: All pictures are not for public access and the publisher must have the option to restrict who the viewer is. This keeps the publishers personal integrity intact.

Account Creation

The user shall be able to **create an account** that can be used to associate him with the pictures he publishes.

Rationale: This will help us to associate publishers with pictures. This will also help us to keep track of the amount of users and to regulate account sizes.

Viewing Pictures

The user shall be able to view **pictures on a website using a standard web browser**.

Rationale: One of the main reasons for a publisher to put pictures in our server is to share them with friends, by using a standard web browser we make it possible for anyone to view the pictures.

Picture searching

The user shall be able to **search for pictures**.

Rationale: If you don't know where to find the picture but you still know the content, you may search by the keyword or metadata associated with the picture.

System behavior in Picture Searching

The system shall search for pictures using the information stored in the keywords and metadata.

Rationale: This will make the search more accurate.

User input of metadata

The publisher shall be able to **input metadata** for each picture. If the user wants to the system can try to find as much information as possible from the image file.

Rationale: The metadata can be interesting when viewing the picture at a later time. It also helps when searching for pictures with certain properties.

Separation of keywords

Keywords attached by a different user than the publisher shall be **kept separate from the publisher's keywords**.

Rationale: This makes it possible for users to include only the publisher's keywords in search results since those keywords could be more relevant.

Optimization of image quality

The picture displayed on the website shall be **optimized for viewing in a web browser**.

Rationale: Saves bandwidth and makes it easier to view the picture.

Removing accounts

The users shall be able to **remove their own accounts**. **The administrator** shall be able to **remove any account** from the system.

Rationale: The users may not want us to store their information any more. The administrator must be able to remove the accounts of users that misbehave.

Removing a picture

The publishers shall be able to **remove the pictures** they have previously published.

Rationale: The publisher might not want others to view his pictures anymore.

Download restrictions

The publisher shall be able to **choose whether or not the original picture shall be available for downloading**.

Rationale: A publisher might for various reasons not want to make it possible for other people to download the original picture.

Upload quota

The administrator shall be able to **decide how many megabytes a publisher is allowed to upload**.

Rationale: This must be possible to regulate since otherwise one publisher could fill the server with his pictures.

Personal information access

A user shall be able to **decide if he wants other people to be able to view his personal information**.

Rationale: This is a matter of privacy; the users who want to be anonymous shall have that option.

Adding another account to use in the client

A publisher shall be able to save **more than one accounts information in the client**.

Rationale: If you would like to let your friend use your client you shouldn't have to remove your own account information.

Non-functional requirements

Usability Requirements

"How to" document

Both the client and the web page shall provide a “how to” document that helps the users to use the program.

Rationale: A “how to” document helps the user to more effectively use either the client or the web page depending on what the user wants help with.

Easy to install and publish pictures

The system should allow novice users to install the client without problems and publish pictures without training or without having to use the “how to” documentation

Rationale: A novice user should without problems be able to install the client software and publish pictures without having to use a guide describing how to carry out the publishing (e.g. the “how to” documentation).

Publishing more than one picture at once

The publisher shall have the possibility to publish more than one picture at once.

Rationale: This will make the Picture Publisher more useful for the publishers that publish a lot of pictures.

Performance Requirements

Picture access time for viewer

During optimal connection conditions the Picture Publisher software should make it possible for a viewer to view a picture 5 seconds after the picture has been uploaded to the server.

Rationale: A service that makes the Picture Publisher more useful for both publishers and viewers.

Web page download times

During optimal connection condition the Picture Publisher software shall not extend the download time for the web page and pictures to the web page.

Rationale: So that the users feel that the web page isn't slower than other web pages.

Space Requirements

Specified amount of space

The total amount of storage space which the publishers can use to publish pictures shall not exceed the Picture Publishers total storage capability on the server.

Rationale: If this happens the publishers might not be able to publish pictures.

Reliability Requirements

Availability rate

The Picture Publisher software shall have an availability rate at 99.5%, except downtime caused by other factors than the Picture Publisher software.

Rationale: To ensure the users that the Picture Publisher is a stable system from which they can access their pictures at almost any time.

Correct picture return

The system shall return the correct picture to 99.9% of the requests to the server.

Rationale: If the Picture Publisher does not return a correct picture on request the customer will be unsatisfied

Java installed

The publisher must have Java installed on their computer for the Picture Publisher client to work.

Rationale: The developers have chosen to develop the client in Java. This is a side effect of that.

Web browser specifications

The web page shall work as expected in at least Mozilla Firefox 1.5 or later and Microsoft Internet Explorer 6.0 or later.

Rationale: A viewer shall be able to view the web page without problems caused by the web page design.

Privacy Requirements

User privacy

The Picture Publisher shall be in compliance with PUL, personuppgiftslagen.

Rationale: This insures the users that their personal information will not be available to the other users, unless the user chooses to make their personal information public.

System Requirements

Picture Storage

- **Function**: Store a picture
- **Description**: Store an exact copy of the picture on the server using the client
- **Inputs**: A picture
  - o **Source**: The user selects a picture from a local storage
- **Outputs**: An exact copy of the picture
  - o **Destination**: The server
- **Action**: The client will check which pictures the publisher wants to publish. The client then prepares them for publishing and sends them to the server. The server stores them.
- **Pre-condition**: The client is installed, there is a connection to the server, the file formats of the pictures are correct
- **Post-condition**: An exact copy of the picture is stored on the server as well as downsized copies of the picture
- **Side effects**: None

Picture Downloading

- **Function**: Download picture
- **Description**: Download a picture from the server to the client.
- **Inputs**: The picture the user wants to download
  - o **Source**: The user
- **Outputs**: A picture
  - o **Destination**: The user’s computer
- **Action**: The user requests which picture to download through the client. The client sends the request to the server. The server sends the requested picture back to the client.
- **Pre-condition**: The client is installed, there is a connection to the server
- **Post-condition**: The picture is successfully downloaded to the user’s computer.
- **Side effects**: None

Rotating pictures

- **Function**: Rotation of pictures
- **Description**: Before publishing a picture it shall be possible to rotate it
- **Inputs**: A picture
  - **Source**: The client
- **Outputs**: A rotated picture
  - **Destination**: The client
- **Action**: It shall be possible to rotate the picture clockwise or counter clockwise. This shall be done 90 degrees at a time.
- **Pre-condition**: The client is installed, there exists a picture to rotate
- **Post-condition**: The picture is rotated and the picture looks like the publisher wants to
- **Side effects**: None

Adding keywords to pictures through the client

- **Function**: A keyword is attached to a picture
- **Description**: The publisher uses the client to attach a keyword
- **Inputs**: Information about the picture
  - **Source**: The publisher
- **Outputs**: A picture with keywords attached to it
  - **Destination**: The server
- **Action**: The user enters the information about the picture and saves it.
- **Pre-condition**: The client is installed, there exists a picture to attach keywords to
- **Post-condition**: The keyword is attached to the picture
- **Side effects**: None

Adding keywords to pictures through the web

- **Function**: A keyword is attached to a picture

- **Description**: The viewer uses the web client to attach a keyword

- **Inputs**: Information about the picture
    - **Source**: The viewer

- **Outputs**: A picture with keywords attached to it
    - **Destination**: The server

- **Action**: The user enters the information about the picture and submits it to the server.

- **Pre-condition**: The viewer is viewing a picture, there exists a picture to attach keywords to

- **Post-condition**: The keyword is attached to the picture

- **Side effects**: None

Sorting pictures

- **Function**: Sorting the pictures

- **Description**: The publisher shall be able to sort the picture into albums

- **Inputs**: Pictures to sort
    - **Source**: The publisher

- **Outputs**: Albums with pictures sorted into them
    - **Destination**: The server

- **Action**: The user creates albums and decides which pictures go into which album.

- **Pre-condition**: The client is installed, there exists pictures to sort

- **Post-condition**: The pictures are sorted

- **Side effects**: None

Access Restrictions

- **Function**: To restrict access to certain pictures

- **Description**: The user shall be able to restrict who can view his pictures

- **Inputs**: One or several groups or persons and one or several albums or pictures
  - **Source**: The publisher

- **Outputs**: Access restrictions for the albums and pictures
  - **Destination**: The server

- **Action**: The publisher chooses which picture or album he wants to put restriction on and decides which groups or persons are allowed to watch them.

- **Pre-condition**: The client is installed, there exists pictures to set restrictions on

- **Post-condition**: The access restrictions are set

- **Side effects**: None

Account creation

- **Function**: Create an account for the user

- **Description**: Creates an account that may for example be used to associate the publisher with the pictures he uploads

- **Inputs**: Account data
  - **Source**: The user

- **Outputs**: An account
  - **Destination**: The server

- **Action**: The user enters his desired account information and submits. The request is handled by the server who either accepts or rejects the account creation and gives the user a message telling him the outcome.

- **Pre-condition**: The user is on our account creation webpage

- **Post-condition**: The users account is created with the requested information

- **Side effects**: The account name the user has chosen is no longer available for others to choose

Viewing Pictures

- **Function**: Possibility to view pictures

- **Description**: It shall be possible to view pictures that have been published by using a standard web browser

- **Inputs**: Which picture to view
    - **Source**: The user

- **Outputs**: Display of the requested picture
    - **Destination**: The web client

- **Action**: The user enters our website and chooses a picture he wants to view. That picture is then displayed

- **Pre-condition**: Someone has published pictures that you can view, a standard web browser and a connection to our server

- **Post-condition**: The user gets to view the picture he requested

- **Side effects**: None

Picture searching

- **Function**: Finding published pictures

- **Description**: The user shall be able to search for pictures that have been published on our website

- **Inputs**: A search string
    - **Source**: The user

- **Outputs**: The result of the search
    - **Destination**: The web client

- **Action**: The user enters the search string and submits it. The server then makes the search and returns the possible matches.

- **Pre-condition**: The user is on our website, there must be pictures published that you can search for

- **Post-condition**: The server returns a result that contains either pictures or a message saying nothing was found

- **Side effects**: None

System behavior in Picture Searching

- **Function**: Better searching

- **Description**: The system shall search for pictures using the information stored in the keywords and metadata of the pictures

- **Inputs**: A search string
  - **Source**: The web client

- **Outputs**: A search result
  - **Destination**: The web client

- **Action**: The web client has received a search string from the user and now requests the information from the server. The server makes queries to the database about which picture's keywords and metadata corresponds to the search

- **Pre-condition**: A search input

- **Post-condition**: A result of the search is returned

- **Side effects**: None

User input of metadata

- **Function**: Input metadata for picture

- **Description**: The publisher can enter metadata for each picture, e.g. camera maker: Sony, the client can try to do it for him if he wants to.

- **Inputs**: A picture
  - **Source**: The publisher selects a picture from a local storage

- **Outputs**: A picture with metadata
  - **Destination**: The client

- **Action**: The publisher selects a picture and enters the metadata he wants to associate with the picture; alternatively he can ask the client to retrieve as much information as possible from the picture.

- **Pre-condition**: The client is installed, a picture

- **Post-condition**: A picture with all the information the publisher wants

- **Side effects**: None

Separation of keywords

- **Function**: To prevent viewers from destroying the search function by attaching irrelevant keywords to pictures

- **Description**: Keep track of which of the keywords attached to a picture that has been attached by the publisher and which has been attached by a viewer

- **Inputs**: A picture with keywords and a search where the viewer has requested to only view the keywords added by publishers
    - **Source**: The server

- **Outputs**: A search result
    - **Destination**: The web client

- **Action**: The viewer searches for a picture and selects that he only wants to view the pictures where the keywords added by the publisher matches his search string. The server performs the search and returns the pictures where the keywords of the publishers match the search string.

- **Pre-condition**: A picture with keywords on the server, connection to the server

- **Post-condition**: The search result only contains pictures where the keywords of the publishers match the search string

- **Side effects**: None

Optimization of picture quality

- **Function**: Optimize picture for web

- **Description**: The client optimizes the pictures being published for faster downloading in the web client

- **Inputs**: A picture being published
    - **Source**: The publisher

- **Outputs**: An optimized copy of the original image
    - **Destination**: The server

- **Action**: The publisher selects a picture to publish and the client optimizes it for faster viewing in the web client by reducing its resolution. The client makes at least two copies with different resolutions. Then the original picture and the optimized ones are published.

- **Pre-condition**: A connection to the server, a picture that can be optimized

- **Post-condition**: The server receives at least the original picture and two optimized versions

- **Side effects**: None

Removing accounts

- **Function**: Removal of account
- **Description**: It shall be possible to remove accounts
- **Inputs**: An account to be removed
    - **Source**: A user
- **Outputs**: An updated account database
    - **Destination**: The server
- **Action**: The publisher or administrator selects an account to be removed. The system updates the database and the account is removed.
- **Pre-condition**: Connection to the server, an account to remove
- **Post-condition**: The account is removed from the server
- **Side effects**: The pictures associated with this account will be removed

Removing a picture

- **Function**: Removal of a picture
- **Description**: The publisher shall be able to remove pictures that he has published
- **Inputs**: A picture to be removed
    - **Source**: The publisher
- **Outputs**: An updated server
    - **Destination**: The server
- **Action**: The publisher finds the picture he wants to remove and selects it for removal. The system removes the picture.
- **Pre-condition**: A connection to the server, a picture to remove
- **Post-condition**: The picture is removed from the server
- **Side effects**: None

Download restrictions

- **Function**: Setting download restrictions
- **Description**: The publisher shall be able to decide whether or not it shall be possible to download the original picture
- **Inputs**: A picture to restrict access for
  - **Source**: The publisher
- **Outputs**: A restriction
  - **Destination**: The server
- **Action**: The publisher selects a picture and restricts the access
- **Pre-condition**: A connection to the server, a picture to restrict download access for
- **Post-condition**: The download restriction is set
- **Side effects**: None

Upload quota

- **Function**: Setting the upload quota
- **Description**: The administrator shall be able to set how much data a publisher is allowed to upload
- **Inputs**: A quota for a specific account or groups of accounts
  - **Source**: The administrator
- **Outputs**: A quota
  - **Destination**: The server
- **Action**: The administrator chooses an account or group of accounts to set the quota for and the system is updated
- **Pre-condition**: A connection to the server, an account to set the quota for
- **Post-condition**: The quota is set
- **Side effects**: None

Personal information access

- **Function**: Limiting who can see personal information

- **Description**: The users shall be able to choose whether or not their personal information is available for public view

- **Inputs**: An account to restrict personal information viewing
    - **Source**: The user

- **Outputs**: A restriction
    - **Destination**: The server

- **Action**: The user logs into his account and chooses whether or not his personal information shall be displayed to the public.

- **Pre-condition**: Connection to the server, an account to set restrictions for

- **Post-condition**: The restriction is set

- **Side effects**: None

Adding another account to use in the client

- **Function**: To store account information for more than one account

- **Description**: It shall be possible for the publisher to store information about more than one account

- **Inputs**: An account to add
    - **Source**: The user

- **Outputs**: Another account to use in the client
    - **Destination**: The client

- **Action**: The publisher chooses to add another account. He enters the account details and chooses to add the account.

- **Pre-condition**: The client is running

- **Post-condition**: There is now two accounts to choose from

- **Side effects**: None

Use Cases

Stakeholders

- **Publisher**
  A Person using the desktop client software.

- **Viewer**
  A person using the web client.

- **User**
  A publisher or viewer where there is no difference between these two categories of users

- **Administrator**
  A person with administrative privileges (i.e. someone who can delete accounts and set system wide preferences and permissions).

- **Developers**
  The team behind the Picture Publisher system

Getting an account

Primary Actor: User
Stakeholders and Interests:

- User: Wants to be able to publish or view pictures
- Administrator: To be able to apply usage restrictions and quotas
- Developers: It's always nice that someone wants to use our system

Preconditions: The user has found the Picture Publisher web site and decided to register an account

Success Guarantee: The user successfully registers an account

Main Success Scenario:

1. User enters the registration information: name, a unique username, password and e-mail
2. The user submits the account request
3. The system confirms that the registration was successful

Extensions:

1a. The user does not want others to be able to view his personal information

1. The user chooses to not let others view his personal information
2. Continue from step 2

2a. Some of the registration information is not valid

1. The System will highlight the invalid fields
2. The user fixes the invalid parts and submits the now valid information and then continues from step 3

Installing the client and getting started

Primary Actor: Publisher
Stakeholders and Interests:

- Publisher: Wants to be able to publish pictures

Preconditions: The publisher has created an account and downloaded the client

Success Guarantee: The publisher installs the client and successfully starts the program

Main Success Scenario:

1. The publisher starts the installation
2. The publisher decides where to install the program
3. The installation is performed successfully
4. The publisher starts the program
5. The program prompts the publisher for his account details and the details for the server (e.g. address, port etc)
6. The publisher enters his username and password and accepts the preconfigured server information
7. His account name is saved and if he wants to his password is saved as well

Extensions:

–

Add an account to use in the client

Primary Actor: Publisher
Stakeholders and Interests:

- Publisher A and B: A Want to have the possibility to let his friend B publish pictures from A's client using B's account

Preconditions: A has started the client and B has an account

Success Guarantee: B is logged into his own account and can publish pictures

Main Success Scenario:

1. B brings up the "Account Settings" screen
2. B adds his existing account by entering the required details and chooses not to save his password
3. The account is added and B can use it to publish pictures
4. A later switches back to his own account

Extensions:

2a. B wants to connect to another server

1. B changes the server information and saves it
2. Continue from step 3

3a. The information B entered is invalid

1. The system will prompt for valid account information
2. Continue from step 3

Publishing pictures using the client

Primary Actor: Publisher
Stakeholders and Interests:

- Publisher: Wants to be able to publish pictures
- The viewer: Wants to be able to view the published pictures

Preconditions: The publisher is running the client

Success Guarantee: The publisher successfully publishes one or more pictures to the server

Main Success Scenario:

1. The publisher imports a directory containing some of the pictures he wants to publish. He can only see the pictures of the file types supported by the client
2. The publisher browses the pictures looking for the ones he wants to publish
3. The publisher adds the pictures to the ones being published
4. The publisher synchronizes the changes with the server
5. The system creates at least two copies of each picture with lower resolutions
6. Exact copies of the pictures and the copies of each picture is published to the server

Extensions:

–

Editing your published pictures

Primary Actor: Publisher
Stakeholders and Interests:

- Publisher: Wants to be able to publish pictures
- The viewer: Wants to be able to view the published pictures

Preconditions: The publisher is running the client

Success Guarantee: The publisher successfully edits one or more pictures.

Main Success Scenario:

1. The publisher selects a picture he wants to edit
2. He synchronizes the changes with the server

Extensions:

1a. The picture the publisher wants to edit is upside down

1. The publisher rotates the picture into the right position and then proceeds to step 1

1b. The publisher wants to add metadata to the picture

1. The publisher enters the information he finds valid for the picture and then proceeds to step 1

1c. The publisher wants to set access restrictions for a picture

1. The publisher chooses which users are allowed to view his pictures and proceeds to step 1

1d. The publisher sorts the pictures into albums

1. The publisher moves pictures into different albums according to his liking and continues from step 1

1e. The publisher removes the picture since he doesn’t want it to be available for others to view anymore

1. The publisher removes the picture and continues from step 1

1f. The publisher wants to set download restrictions

1. The publisher decides if other users should be allowed to download the original version of his pictures and continues from step 1

Downloading (exporting) pictures using the client

Primary Actor: Publisher
Stakeholders and Interests:

- Publisher: Wants to be able to retrieve exact copies of pictures previously published

Preconditions: The publisher has published some pictures to the server

Success Guarantee: The publisher successfully downloads a picture from the server

Main Success Scenario:

1. The publisher browses his previously published pictures and finds one he wants to export for other uses (e.g. edit, print or email to a friend)
2. The publisher selects the picture he wants to download
3. The system asks the publisher to choose where to save the picture
4. The publisher selects a directory and the picture is downloaded

Extensions:

–

Viewing a picture using a web browser

Primary Actor: Viewer
Stakeholders and Interests:

- Publisher: Wants the viewer to view his pictures
- Viewer: Wants to check out some pictures

Preconditions: The viewer has started a web browser and navigated to the web site containing the published pictures

Success Guarantee: The viewer found some nice pictures

Main Success Scenario:

1. The viewer selects an album containing pictures that he wants to view
2. The viewer flips through the pictures

Extensions:

1a. The viewer chooses to search for a picture

1. The viewer performs a search
2. The viewer selects one of the pictures from the search result and views it

2a. The viewer wants to add keywords to a picture

1. The viewer enters some keywords to the picture and saves it

2b. The viewer wants to download a picture

1. The viewer saves the picture to the desktop using the browsers standard save feature.

Searching

Primary Actor: Viewer
Stakeholders and Interests:

- Publisher: Wants the viewer to view his pictures
- Viewer: Wants to find a certain picture or set of pictures

Preconditions: The viewer has started a web browser and navigated to the web site containing the published pictures

Success Guarantee: The viewer found the pictures he was looking for

Main Success Scenario:

1. The viewer types in one or more words to search for
2. The system searches for pictures by trying to match the metadata of the pictures against the search string
3. The system returns the search result

Extensions:

1a. The viewer wants the search to only use the keywords inserted by a publisher

1. The viewer selects that he only wants to use keywords inserted by a publisher
2. The search continues from step 2

Removing an account (administrator)

Primary Actor: Administrator
Stakeholders and Interests:

- Administrator: To be able to remove accounts

Preconditions: The administrator is logged into the system

Success Guarantee: The account is successfully removed

Main Success Scenario:

1. The administrator chooses to remove an account
2. The administrator looks up the account he wants to remove
3. The administrator chooses to remove the account
4. The administrator is asked for confirmation that he really wants to remove the account
5. The system removes the account and all pictures associated with it

Extensions:

–

Removing an account (user)

Primary Actor: User
Stakeholders and Interests:

- User: To be able to remove his own account

Preconditions: The user is logged in to the system

Success Guarantee: The account is successfully removed

Main Success Scenario:

1. The user chooses to remove his account
2. The user is asked for confirmation that he really wants to remove his account
3. The system deletes the account and all pictures associated with it

Extensions:

–

Setting the upload quota

Primary Actor: Administrator
Stakeholders and Interests:

- User: Wants to be able to upload or view pictures
- Administrator: To be able to apply usage restrictions and quotas

Preconditions: The administrator is logged in to the system

Success Guarantee: The upload quota is changed to the desired value

Main Success Scenario:

1. The administrator opens the configuration screen
2. The administrator enters the new quota
3. The system asks him for confirmation that he wants to change the quota
4. The administrator accepts the change and the system updates the quota

Extensions:

2a. The administrator only wants to change the quota of one account

1. The administrator selects the account he want to change the quota for
2. The administrator changes the quota, continue from step 3

4a. The system tells him that if he make this change it will be possible to upload more data than the server can handle

1. The system cancels the update of the quotas
2. The administrator starts over from step 2

Product and process standards

The system handles pictures stored as image files. There is a lot of different file formats for images and the system supports these:

- .jpeg
- .tiff

The system development follows the **Java code convention** http://java.sun.com/docs/codeconv/

Trac shall be used for project management.

Glossary

A

API

"An application programming interface (API) is a source code interface that a computer system or program library provides in order to support requests for services to be made of it by a computer program." (Wikipedia)

C

Client

The application used to publish pictures.

D

Database

A collection of records stored in a computer in a systematic way, so that a computer program can consult it to answer questions.

E

EJB3 (Enterprise Java Bean 3.0)

A managed, server-sided component for modular construction of enterprise applications.

H

HTTP

"Hypertext Transfer Protocol (HTTP) is a method used to transfer or convey information on the World Wide Web. Its original purpose was to provide a way to publish and retrieve HTML pages." (Wikipedia)

J

JBoss Application Server

JBoss' implementation of a platform for delivering server-side Java applications and web services.

JSP

"Java Server? Pages (JSP) is a Java technology that allows software developers to dynamically generate HTML, XML or other types of documents in response to a Web client request. The technology allows Java code and certain pre-defined actions to be embedded into static content." (Wikipedia)

JSR

"Java Community Process or JCP, established in 1998, is a formalized process which allows interested parties to be involved in the definition of future versions and features of the Java platform. The JCP process involves the use of Java Specification Request (JSR), which are formal documents that describe proposed specifications and technologies to be added to the Java platform. Formal public reviews of JSRs are conducted before the JSR becomes final and is voted on by the JCP Executive Committee. A final JSR provides a reference implementation which provides a free implementation of the technology in source code form and a Technology Compatibility Kit to verify the API specification." (Wikipedia)

K

Key words

A type of metadata involving the association of descriptors with objects.
Here: a collection of short strings describing a picture's content
(e.g. "Eiffel Tower", "car sun beach", "John's family" etc.)

M

Metadata

Information about the content of the specified data
(e.g. the date, camera type, camera objective or subject of an image file)

P

Personuppgiftslagen (PUL)

A law designed to prevent the personal integrity from being violated through the processing of personal information.
Further reading (in Swedish): http://www.datainspektionen.se/lagar/pul.shtml

Publisher

A user who uploads pictures to his account using the client
(e.g. a photographer sharing pictures of cars from a car show)

S

Stakeholder

A person or a group involved in and interested in the success of a project
(e.g. the users of the system)

T

Tomcat Web Server

A web container that specifies a runtime environment for web components

Trac

Trac is a tool for managing software projects using a wiki, timeline, source control and a roadmap with milestones etc.

U

User

A user is anyone using the system, either a viewer or a publisher
(e.g. someone viewing his friend's holiday pictures or someone publishing those holiday pictures)

V

Viewer

A user who views the pictures uploaded by a publisher
(e.g. someone viewing his friend's holiday pictures)

Index