Recquirement and Analysis Document (RAD) for the Hedgehog Photo Project

Contents

Version: 1.0
Date: 2012-05-23
Author: Hedgehog Photo
This version overrides all previous versions.

1 Introduction

This section gives a brief overview of the project.

1.1 Purpose of application

The project aim is to create a photo organizer/photo handling program that's easy to use.

1.2 General characteristics of application

The application will be a standalone desktop application with a graphical user interface for the Windows/Mac/Linux platforms.

The user will be enabled to view his/her photos directly. The user will also be enabled to search after tags, part of a text that belongs to a specific photo, photo names or dates and select images based on locations from an interactive map. The applications will also feature a tag-cloud where the user can easily find pictures based on tags relevance. There will also be an area where the user can do some quick drawings.

1.3 Scope of application

This application main feature is to organise photos. This is not intended to be a photo editor, it's a photo organizer and viewer so no photo-editing capabilities will be implemented.

1.4 Objectives and success criteria of the project

For the project to be called successful we want to implement the following features:

1. tags - associate photos with tags
2. comments - add comments to pictures
3. Interactive map - to show were your pictures where taken.
4. search (tags,text,photo name) - easy to find what you're looking for
5. zoom - decide how many photos (and possibly text and tags) you want to watch in the screen

1.5 Definitions, acronyms and abbreviations

*GUI, graphical user interface.
*Java, platform independent programming language.
*JRE, Oracle's Java execution environment
*Host, a computer where the application will run
*Tag, a word associated with a picture.
*Database, an organized collection of data

2 Requirements

In this section we specify all requirements

2.1 Functional requirements

1. View photos.

2. Be able to tag photos and sort them by personal preferences.
3. Find photos by searching for tags, categories or by selecting them from the map.

2.2 Non-functional requirements

2.2.1 Usability

Usability has a high priority. The application must be effective, easy, intuitive and fun to use.

2.2.2 Reliability

The applications should not be prone to crashes.

2.2.3 Performance

Performance is not our top priority. Our applications response time is heavily dependent on the amount and size of the pictures. Any action initiated by the user, excluding importing should definitely not exceed a sec in the worst case scenario.

2.2.4 Supportability

The application will achieve modularity by following a well know strategy known as Model View Control. If implemented correctly this means that the main core of the application is separated from the View and Control making our application easily modifiable to suit other platforms, such as phones and the Webb.
There should be automated and comprehensive tests to ensure and verify our core as functional and stable. The GUI have to be tested on each platform manually as it impossible to write computer-based test for it. These human test should be recorded and included in the final documentation.

2.2.5 Implementation

The application will be platform independent so it will use the Java environment to achieve this. All host must have downloaded, installed and configured the JRE version 1.7 to be able to run the application.

2.2.6 Packaging and installation

This is a open source project so all documentation, source code, resources, etc will be included in the zip-archive which can be delivered by any medium.

2.2.7 Legal

Legal boundaries of this project may occur, if the project is to be distributed as an application. However, this will not be covered here.

2.3 Application models

2.3.1 Use case model

Please see the separated documents for UML diagram and textual descriptions of use case models.

2.3.2 Use cases priority

Basic functionality first:

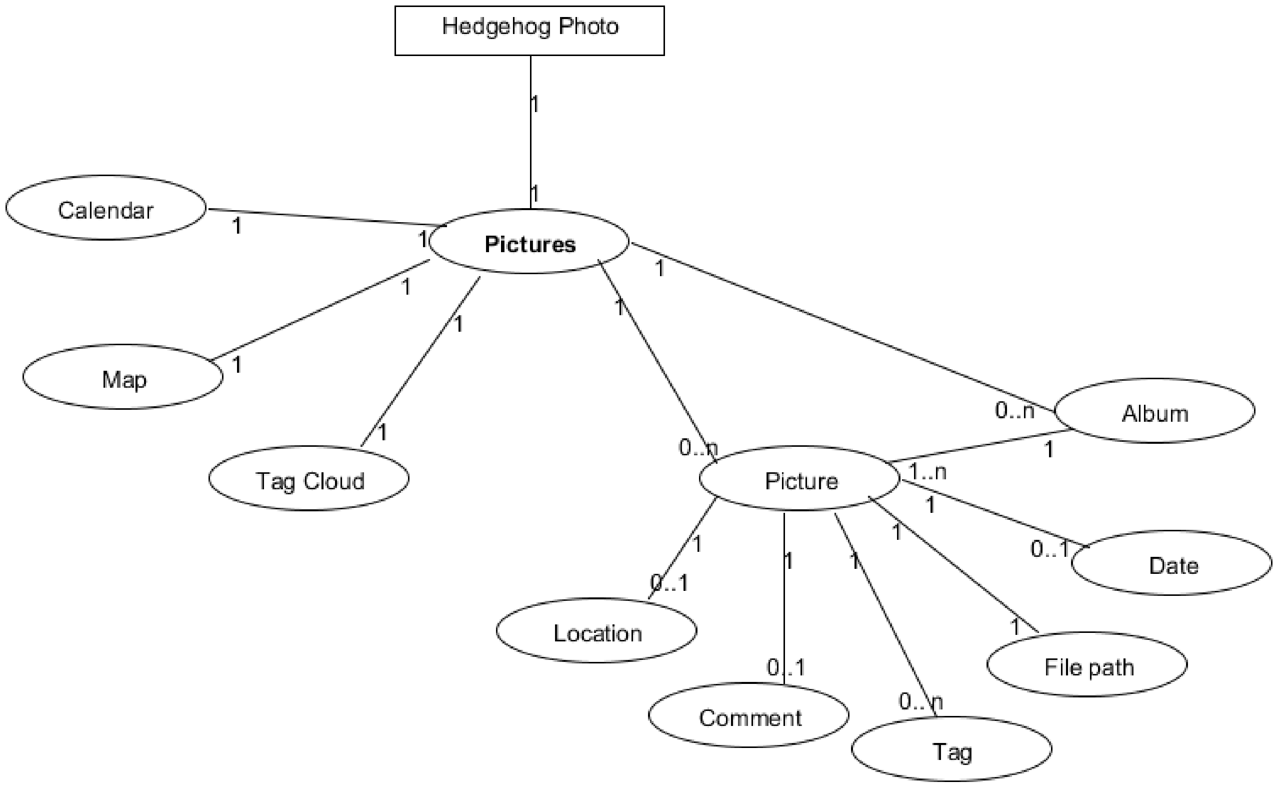
1. Exit
2. chooseSelection
3. ViewPhotos
4. ViewPhoto
5. addComment, addTag, addLocation
6. search

Unprioritised but should definitely be implemented in our application:

all the other use cases.

APPENDIX:

Domain model



Hedgehog Photo
1
1
Calendar
1
1
Pictures
1
1
Map
1
1
1
Tag Cloud
1
1
1
0..n
Album
0..n
Picture
1
1..n
1
1
0..1
Date
1
1
1
0..1
Location
1
File path
0..1
Comment
0..n
Tag