

Software Requirement Specification

for

SkyBar Photo Management System

Group ID: R01

Abhyuday Choumal-CS20B1001,
Beerelly Srinitha-CS20B1004,
Kaustubh Kesharwani-CS19B1015,
Rakesh Kumbhkar-CS20B1017,
Sikander Kathat-CS20B1020,
Yugal Garg-CS20B1027.

Instructor: Manish Singh
Course: Software Engineering (CS210)
TA's: Suryamukhi K, Ashish Kishor
Date: 20th Feb 2022

Contents

1. Introduction	2
1.1 Purpose	2
1.2 Intended Audience	2
1.3 Project Scope	2
1.4 Acronyms and Abbreviations	2
1.5 Overview	3
2. Overall Description	3
2.1 Product Perspective	3
2.2 Product features	4
2.3 User Classes and Characteristics	5
2.4 Operating Environment	5
3. System Features	6
3.1 Add Photos	6
3.2 Sort Photos	7
3.3 Manage Photos	7
3.4 Online Sharing	7
3.5 Settings	8
3.6 Downloading Options	8
4. External Interface Requirements	9
4.1 User Interfaces	9
4.2 Software Interfaces	9
4.3 Communications Interfaces	10
5. Non-Functional Requirements	11
5.1 Performance	11
5.2 Security	11
5.3 Software Quality Attributes	11

1. Introduction

1.1. Purpose

The purpose of this Software Requirement Specification (SRS) document is to provide a detailed description of the Photo Management System (PMS) product. This document contains the functional and non-functional requirements of the project. This document contains the guidelines for the developers according to the client's proposal for the system which can help in the organization and for sharing of digital photographs both within the private and public workspaces throughout their wider network of friends and colleagues.

1.2. Intended Audience

This document is aimed mainly at the designers and developers so that they can design and code as per the requirements of the clients and end-users of the described system. It will hopefully also be useful to the Inspection Team (Customers, Testers, Authors) of the project, allowing them to inspect the work-products as the development progresses that meet the criteria specified below.

1.3. Project Scope

The PMS product is intended to provide a single system to allow multiple users in the house to flexibly store and sort their photographic media, both between themselves and among others. It allows media to be stored and maintained in a single location, from which they can be shared to other locations accessible by a wider group of people.

We feel that a system that allows central management and selective distribution to such devices would be highly desirable to a wide demographic of people.

The Photo Management System is composed of components like a web-frontend application that is used by users, a server-side application that is handled by the maintainers, a database to store the pictures. The system is designed to facilitate the smooth organizing of the images over the server.

1.4. Acronyms and Abbreviations

SRS	Software Requirement Specification
PMS	Photo Management System
RQ	Requirements
XML	Extensible Markup Language

JSON	JavaScript Object Notation
-------------	-----------------------------------

1.5. Overview

The description is intended to give a specific overview of the key features that the product will require, and the minimum of operations it must perform.

2. Overall Description

2.1. Product Perspective

The Photo Management project is a new, self-contained product intended for use as a web application.

Everyone has their way of organizing their photos. This means everyone is different, which is a major frustration during photo sharing. Collections could be sorted when they are displayed to the user, allowing anyone to see them however they desire, rather than only the owner's preferred system.

Sharing photos also contains several subtasks that could be accomplished with minimal user input. Many steps are repeated while adding to the storage system and then to the sharing systems. It would be simple if any of these tasks happen automatically.

The main purpose of this system is to provide a way for users to be able to share photos in a far more unified manner and also be able to share them with people via social networking sites.

The scope of the project covers and surrounds both server and client-side functionalities so that both aspects are covered more in detail within this document.

2.2. Product Features

The system has three core functionalities which are **Add**, **Manage**, and **Share** respectively. A central server device stores the user's photos and performs tasks when requested by a client device.

Client devices could be any of the phones, cameras, laptops, digital photo frames, or any other device which is capable of a network connection and running a web application to use our product. The process begins when the client device comes into contact with the server, and new photographs are automatically moved across to the server and processed. The user can have predefined behaviours based on certain rules, allowing the system to do as much of the work as possible for them.

Whenever a collection of photos is displayed to choose from, the user can filter and sort them by any information they have entered into the system, allowing them to have them appear in whatever way they prefer.

Most importantly, they can decide to share the photos with any device they own that supports the system, and with any service that the system can interact with. If the user likes, they can have the system automatically push new photos to whichever device they like as soon as they add them.

Some more Features:

1. User Registration and Welcome

For the first time, the user should undergo the sign-up operation with our product in order to start using its features. After signing up, the user can manage his/her own dashboard according to his/her choice.

2. Photo Upload

The user can upload his photos with the help of a product application to store on our server.

3. Photo Management

All the other functionalities come under the management section which is explained later in detail in this document like Add, Arrange, Delete, and Quick-Share.

4. Settings Menu

It allows the user to customize his/her own preferences, enabling the user to modify certain features and functionalities, and can be accessed at any time with the help of the Setting Tab.

5. Help Menu

Displays a list of topics covering the different components of the SkyBar PMS. And also offers detailed information on each feature, menu, etc and can be accessed at any time via the Settings Menu.

2.3. User Classes and Characteristics

The users of the system are not expected to have a wide range of computing skills as many of the users will be home users and from different backgrounds of varying ages, therefore requiring a simple interface to use the system effectively. Consequently, the application will have little or no big learning curve, and the user interface will be as intuitive as possible. They are most likely to have a rough idea of how the system functions, as most will be familiar with interfaces for uploading and viewing photos.

The typical sub-categories of user classes within a typical household generally consist of parents and their children.

Since the device is intended for household users, security must be considered as the implementation of features such as parental control is considered necessary to prevent data loss which may occur accidentally, i.e., due to users being unable to understand the system. With parental controls, users can be assured that only they have the privilege to perform required actions.

Student households are considered more computer-literate, and it can be assumed that this user class will be having less trouble than others while using the system. In terms of the product design, there should not be as much consideration necessary for this user class in comparison to others.

2.4. Operating Environment

The main component of the project - "SkyBar PMS" is the software application, which will be limited only to the Web Browsers (Specifically

supporting HTML 5.0 or later). The application is not resource or graphics-intensive, so there are no practical hardware constraints applicable. The core of the system and the server device are intended to operate within a place and should be able to be placed in any room with power and a (wireless or wired) network connection.

There should be several software clients developed that can interact with the server. These clients should ideally be able to run on any device that can store, upload, or display photos.

As to be noted, the hardware device should use common wireless communication technologies without causing interference with other devices.

3. System Features

3.1. Add Photos

3.1.1 Description and Priority

There are two possible methods for adding photos, automatic and on request. The user can configure devices to automatically send photos to the server when they come into contact, with a specific workflow that is already specified when this happens. It should be possible to have any task that can be performed on the photos happen automatically. The user should also be able to set up a process that occurs automatically but requires little information, for example, an album name and description before it completes. In these instances, the server should submit the queries to the user, when the user connects explicitly (i.e., not automatically) the next time.

The ability to add photos to the system is one of the core features, and this is considered to have a very high priority. The ability of the process to happen automatically is one of the unique features that separate this system from others, and thus being non-essential it is having high importance.

3.1.2 Functional Requirements

RQ-1: Server must be able to receive photos from a client device, and if requested, then it should automatically carry out any applicable operation on the photos.

RQ-2: Client devices must be able to connect and send photos to the server, both automatically and when requested by the user, and either provide the server with any pre-configured work or specify a workflow saved on the server.

3.2. Sort Photos

3.2.1 Description and Priority

Whenever a selection of photos is displayed for the user to select from or to interact with, they should be able to choose whether to sort and filter the displayed photos by any information stored about them.

3.2.2 Functional Requirements

RQ-3: It should be possible to sort and filter photos instantly by any information available. It should be possible to have different sorts (i.e., by Date, Size, Name and then by Group).

3.3. Manage Photos

3.3.1 Description and Priority

Users should be able to select their photos in order to manage them. It must be possible to create an album of the selected photos, to add and edit any information about them. Users should be able to delete photos that they no longer want to keep. Users should be able to add new types of information that can be added, with a name, description, and type (number, date, text).

3.3.2 Functional Requirements

RQ-4: Must be possible to add photos to the created album and edit the respective photo's information.

RQ-5: Must be possible to create an album of selected photos, and give the album a name, description, and add any other details the user would like to provide.

RQ-6: Must be possible to add new types of information.

RQ-7: Must be possible to remove photos from the system.

3.4. Online Sharing

3.4.1 Description and Priority

Users should be able to choose from a range of different devices and services to share selected photos, including the device from which the user is making the request. The user should be informed of any information required to share the photos that are not previously available and shall be given the option to change the information that will be submitted. If the user selects to share a photo with a device, those photos should be pushed to the device and displayed as soon as the device connects to the server. For a service, the photos should be sent in whatever form the service requires its submission. There should be an easily extendable framework for adding new devices and services.

3.4.2 Functional Requirements

RQ-8: Must be possible to pick photos and share them with any available service or device.

RQ-9: Must be easy to add new services or devices to share to.

RQ-10: Client Devices must be able to display photos shared with them, and should do so automatically if appropriate.

RQ-11: Any information that will be submitted to other services should be displayed for the user to check, and any necessary information not already provided should be requested from the user.

3.5. Settings

3.5.1 Description and Priority

Should be able to add user accounts and restrict the available operations to certain accounts.

3.5.2 Functional Requirements

RQ-11: There should be a user management system that allows certain users to be limited in the tasks they can perform.

3.6. Downloading Options

3.6.1 Description and Priority

The user will be provided with all the general options (like PDF, JPG, JPEG, etc)while downloading the images so that the user will find it easy and will face no problem for conversions after the downloading process is done.

3.6.2 Functional Requirements

RQ-12: There should be an image processor which will be able to convert the image type according to the user's need.

4. External Interface Requirements

4.1. User Interfaces

Since there will be client software on multiple types of devices, it is important that all devices have a consistent interface. Many tasks (adding details for photos, sharing photos with different devices/services) will require information from the users. The client application should be able to display these lists dynamically, using a description provided by the server to inform the user what is required, and the type of information requested to display a suitable method of entry. This will allow additional options to the information needed to only require an update to the server, not the client software, and keep information consistent across all client devices.

Operations that allow users to Share and Delete photos should be clearly highlighted to prevent users from accidentally using them.

There should be a minimum of core screens, one for each major operation (Add, Manage, View, Settings) and all additional options and messages should be overlaid on top of these screens.

When users are presented with the ability to select photos, options to filter and sort the photos all available information should be clearly presented because the ability to access photos in a manner suitable to the user is the core goal of the system.

Every Menu will have a descriptive title respectively.

4.2. Software Interfaces

The most critical software interface will be server-client communication. The server should provide the client software with any information that is requested about the photos, services and options present. We recommend that such information is communicated using XML/JSON containing the appropriate data. As an established and flexible passable language, XML/JSON is perfect for this system and allows strict but extendable information to be generated.

It is also important that there is a well-documented interface to allow the addition of extra sharing services/devices to the server software so that new options can be added easily. The easy addition of new sharing methods makes the ability of the server to inform the client what information is needed for each option very significant.

The server should also use a database to store the information about existing photos in the system. The specific implementation used is not essential as long as it is fast and can cope with sorting the large databases without issue on low powered hardware.

4.3. Communication Interfaces

The communication between client and server is important so that a stateful communication interface is used, i.e., it must be known whether a message has reached its intended destination. This should help to prevent the manipulation of data and ensure that whenever issues occur, the user is not left wondering why nothing is happening. Also, when communicating between Client and Server especially from an external device via the internet the information, photos and the information about when, where and what they depict, is secure. Such information comes with critical privacy concerns.

The server should use HTTP and HTTPS to connect to online photo sharing services, and should also be able to use SMTP to share photos by email. Such services come with the same privacy concerns like internal communications between Client and Server, and all measures to secure it should be taken, but this may not be possible depending on the service.

The server exists to retrieve from the database and send it to the front-end. Furthermore, whenever a user opens the web application, a pull protocol will be used to sync all the images of that user.

5. Non-Functional Requirements

5.1 Performance

The uploading process of photos with this system should be fast, allowing the user to access their photographs instantly from any device connected to the system. The system must have a large memory to store a considerable number of photos the user requires.

5.2 Security

As the information stored on this system is personal and is related to individual privacy, there will be a need to be relatively secure. The users will be able to access and edit the information on the system only through their personal devices. An authentication system should be provided requiring a username and a password pair. Once a user has authenticated ideally all communications between the client and server should be encrypted, and at a minimum, all private data, for example, connection details for sharing services, should be protected. Ideally, the public and private key systems would be used to ensure maximum security of the potentially highly personal data being transferred.

5.3 Software Quality Attributes

This system needs to be highly reliable due to users having a high emotional attachment to the majority of their photographs. Also, it needs to be robust and easy to maintain, as a typical home user would want a simple system that will last a long time. The system must be relatively easy to use, as there will be a wide variety of user abilities that will work and access this system. A manual must be provided with step-by-step guidance on how to set up and use the system, with information on the variation of devices able to connect to the system. This manual must also contain any help links (customer care phone numbers, emails etc.) if difficulty occurs for any of the users.