**Software Requirement Specification**

**April 13, 2016  
Photo Album**

1. **Introduction**

This section provides an introduction of this document. It includes sections “Purpose”; “Scope” and “Overview”.

* 1. ***Purpose***

Photo Album application is a part of the Teamwork Project Assignment for the [JavaScript Applications Course @ SoftUni](https://softuni.bg/courses/javascript-applications/).

* 1. ***Scope***

Photo Album intends to be a SPA application where users can upload, store and download photos, liked photos and commented them. This system is designed to be easy-to-use, with intuitive UI, and good usability.

* 1. ***Overview***

The next chapter, the **Overall Description** section, of this document gives an overview of the functionality of the application. It describes the informal requirements and is used to establish a context for the technical requirements specification in the next chapter. The third chapter, **Requirements Specification** section, of this document is written primarily for the developers and describes in technical terms the details of the functionality of the application. Both sections of the document describe the same software product in its entirety, but are intended for different audiences and thus use different language.

**2.0. Overall description**

***2.1. System Environment***

User panel

Admin panel

User

Admin

Photo Album

**Figure 1. System Environment**

The application has two active actors and one cooperating system. Both of them are accessing their parts from the internet. The **User** has a restricted access to the application through the UI. The **Admin** has a full access both to the front-end and the back-end of the application.

***2.2. Functional requirement specification***

***2.2.1. Use Case 1 (Accessing the system)***

The system is accessed through the internet from its initial URL. The predefined tabs for Home, Register and Login should be loaded. A side menu for Pictures, Categories and Users should be displayed. They are inaccessible unless the user is logged in.

***2.2.2. Use Case 2 (Register)***

After clicking the Register tab a register form should be displayed, where the user should input their username, email, password and repeat password. The fields for username and password are required. After successful register a page with all uploaded photos should be loaded and a “Hello, username” greetings should be displayed.

***2.2.3. Use Case 3 (Login)***

If a user has a valid registration, they can login through the Login tab. After clicking it a login form should be displayed, where the user should input their username and password. After successful login a page with all uploaded photos should be loaded and a “Hello, username” greetings should be displayed.

***2.2.4. Use Case 4 (Loading all categories)***

The link Categories is locked for non-registered/logged users. Once a Categories link is clicked, all available categories are displayed, including their name, appropriate icon and an “Add Picture” button.

***2.2.5. Use Case 5 (Loading photos by category)***

After clicking the category icon, all photos from this category should rendered, including a photo’s preview, and additional information as picture’s name, author’s username, likes and comments. The photos’ previews are displayed in four at a row; their names are above, and additional information is below the preview. The photos are ordered by likes in descending order (from most liked -> less liked). Optionally paging may be implemented – to display a user set pictures per page.

***2.2.6. Use Case 6 (Loading all photos)***

The link Pictures is locked for non-registered/logged users. Once a Pictures link is clicked all uploaded photos are rendered, including a photo’s preview, and additional information as picture’s name, author’s username, likes and comments. The photos’ previews are displayed at four at a row; their names are above, and additional information is below the preview. The photos are ordered by likes in descending order (from most liked -> less liked). Optionally paging may be implemented – to display a user set pictures per page.

***2.2.7. Use Case 7 (Display the enlarged photo)***

After a picture’s preview is clicked the enlarged photo should be rendered, including its additional information as picture’s name, author, likes and comments.

***2.2.8. Use Case 8 (Upload a photo)***

After clicking the “Add Picture” button, an input filed for picture name should be displayed and a “Choose File” button should be rendered by which a photo can be chosen to upload from user’s computer file system. After the picture’s file is open, it should be automatically uploaded to the selected category and the user should be relocated to the page with photos from this category.

***2.2.9. Use Case 9 (Like a photo)***

After photos are rendered, a heart shape button should be visible below every picture. A user can like the particular picture by clicking this button. They can like a picture only once. After liking the picture, the information with all likes should be updated immediately without reloaded the page.

***2.2.10. Use Case 10 (Load picture’s comments)***

After photos are rendered, the “Comments” text should be a link, by clicking them all comments for this picture should be displayed. They are ordered by date in descending order (from newest -> oldest). Each comment should consist of author username, date and content.

***2.2.11. Use Case 11 (Add comment)***

After comments for particular photo are loaded, the “Add Comment” button should be displayed, by which the logged user can add new comment for this picture. Clicking on this button a text area should be rendered where the user can input the comment’s content.

***2.2.12. Use Case 12 (Load all users)***

The link Users is locked for non-registered/logged users. Once Users link is clicked, all registered users are displayed, including their usernames and emails.

***2.2.13. Use Case 13 (Administration Area)***

Since the application is divided in two parts, thus an administration panel is required for the editor's purpose. The administration panel consist of the basic CRUD operations regarding the backend data.

***2.2.13.1. Use case 13.1. (Add category)***

A new category can be added. Once the editor clicks "Add Category", a form with an input field and a button “Add Icon” appear. The editor can input the name of the new category into the field and upload an appropriate icon for this category through the button.

***2.2.13.2. Use Case 13.2. (Edit category)***

The categories can be modified by "Edit Category" button. The editor can change their names and icons.

***2.2.13.3. Use Case 13.3. (Delete category)***

The existing category can be deleted by "Delete Category" button.

***2.2.13.4. Use case 13.4. (Upload photo)***

See ***2.2.8.***

***2.2.13.5. Use Case 13.5. (Delete photo)***

The existing photo can be deleted by "Delete Picture" button. The editor can delete photos if they are with inappropriate content, violence, racism, etc.

***2.2.13.6. Use case 13.6. (Add comment)***

See ***2.2.11.***

***2.2.13.7. Use Case 13.7. (Delete comment)***

The existing comment can be deleted by "Delete Comment" button. The editor can delete comments if they are with inappropriate content, violence, racism, etc.

***2.2.13.8. Use Case 13.8. (Ban user)***

A registered user can be banned by "Ban User" button. The editor can banned users if they violate the rules of the web site.

**3. Technical Requirement Specification**

***3.1. Use Case 1 (Initial access) [/]***

As a back-end will be used REST API from kinvey.com.

Accessing "/" returns the main page with tabs for Home, Register and Login and a side menu with links to Pictures, Categories and Users. The links from the side menu are inaccessible unless the user is logged in. Information about logged user is stored if headers with session token are sent. The client consumes usernames for rendering and IDs for later access.

***3.2. Use Case 2 (Register) [/register]***

POST to “<https://baas.kinvey.com/user/appId>" with a body consisting of the user meta-information, should register the user successfully. Validating username and email should be applied.

***3.3. Use Case 3 (Login) [/login]***

POST to “<https://baas.kinvey.com/user/appId>/login" with a body consisting of the user meta-information, should logged the user successfully.

***3.4. Use Case 4 (Loading all categories) [/categories]***

GET "<https://baas.kinvey.com/appdata/appId>/categories" returns all available categories. Optional should be implemented filtering and sorting by name.

***3.5. Use Case 5 (Loading photos by category) [/categories/{categoryId}]***

GET "<https://baas.kinvey.com/appdata/appId>/pictures/?query={category.\_id:{categoryId}}" returns all pictures in this category. Optional should be implemented paging, filtering and sorting by likes, names and date of upload.

***3.6. Use Case 6 (Loading all photos) [/pictures]***

GET "<https://baas.kinvey.com/appdata/appId>/pictures" returns all uploaded pictures with their additional information. Optional should be implemented paging, filtering and sorting by likes, names and date of upload.

***3.7. Use Case 7 (Display the enlarged photo) [/pictures/{pictureId}]***

GET "<https://baas.kinvey.com/appdata/appId>/pictures/{pictureId}" returns the queried picture with its additional information.

***3.8. Use Case 8 (Upload a photo) [/categories]***

POST to “<https://baas.kinvey.com/appdata/appId>/pictures" with a body consisting of the picture meta-information, should upload the picture successfully. Validating pictures size and type should be applied.

***3.9. Use Case 9 (Like a photo)***

Liking system (like / dislike) should be implemented. A user can like many pictures, but a particular picture can be liked only once. After successful like, an appropriate message should be displayed. In case of second attempt to like, an error message should be displayed.

***3.10. Use Case 10 (Load picture’s comments) [/pictures]***

GET "<https://baas.kinvey.com/appdata/appId>/comments/?query={picture.\_id:{pictureId}}" returns the comments of the queried picture with their author’s username, date and content. They should be ordered by date in descending order (from newest -> oldest).

***3.11. Use Case 11 (Add comment) [/pictures]***

POST to “<https://baas.kinvey.com/appdata/appId>/comments" with a body consisting of the comment meta-information, should add the comment successfully. Validating content length should be applied.

***3.12. Use Case 12 (Load all users) [/users]***

GET "[https://baas.kinvey.com/user/appId](https://baas.kinvey.com/appdata/appId)" returns all registered users with their additional information.

***3.13. Use Case 13 (Administration Area)***

***3.13.1. Use case 13.1. (Add category) [/admin/categories]***

POST to “<https://baas.kinvey.com/appdata/appId>/categories" with a body consisting of the category meta-information, should add a new category successfully. Validating category name length and icon size should be applied.

***3.13.2. Use Case 13.2. (Edit category) [/admin/categories/{categoryId}]***

PUT to “<https://baas.kinvey.com/appdata/appId>/categories/{categoryId}" with a body consisting of the category meta-information, should modified the category successfully. Validating category name length and icon size should be applied.

***3.13.3. Use Case 13.3. (Delete category) [/admin/categories/{categoryId}]***

DELETE to “<https://baas.kinvey.com/appdata/appId>/categories/{categoryId}" should delete the category successfully.

***3.13.4. Use case 13.4. (Upload photo)***

See ***3.8.***

***3.13.5. Use Case 13.5. (Delete photo) [/admin/pictures/{pictureId}]***

DELETE to “<https://baas.kinvey.com/appdata/appId>/pictures/{pictureId}" should delete the picture successfully.

***3.13.6. Use case 13.6. (Add comment)***

See ***3.11.***

***3.13.7. Use Case 13.7. (Delete comment) [/admin/pictures/{pictureId}]***

DELETE to “<https://baas.kinvey.com/appdata/appId>/comments/{commentId}" should delete the comment successfully.