**Software Requirements Specification**

**for**

Keyword Search System for text based images

**Version 1.0 approved**

**Prepared by,**

**Adarsh V - 1RV20CS008**

**Abhinandan - 1RV20CS003**

**Department of Computer Science Engineering**

**RV College of Engineering, Bengaluru - 560059**

**14-01-2023**

**Table of Contents**

**Table of Contents ii**

**Revision History ii**

**1. Introduction 1**

1.1 Purpose 1

1.2 Document Conventions 1

1.3 Product Scope 1

1.4 References 1

**2. Overall Description 2**

2.1 Product Perspective 2

2.2 Product Functions 2

2.3 User Classes and Characteristics 2

2.4 Operating Environment 2

2.5 Design and Implementation Constraints 2

2.6 User Documentation 2

2.7 Assumptions and Dependencies 3

**3. External Interface Requirements 3**

3.1 User Interfaces 3

3.2 Hardware Interfaces 3

3.3 Software Interfaces 3

3.4 Communications Interfaces 3

**4. System Features 4**

4.1 System Feature 1 4

4.2 System Feature 2 (and so on) 4

**5. Other Nonfunctional Requirements 4**

5.1 Performance Requirements 4

5.2 Safety Requirements 5

5.3 Security Requirements 5

5.4 Software Quality Attributes 5

5.5 Business Rules 5

**6. Other Requirements 5**

**Appendix A: Glossary 5**

**Appendix B: Analysis Models 5**

**Appendix C: To Be Determined List 6**

**Revision History**

| **Name** | **Date** | **Reason For Changes** | **Version** |
| --- | --- | --- | --- |
|  |  |  |  |
|  |  |  |  |

# **Introduction**

## **Purpose**

The purpose of a project related to a keyword search system on text-based images would likely be to enable users to search for and retrieve specific information contained within the text of images, such as documents or receipts. This system on text-based images can be used to quickly locate specific information within a large collection of images.

## **Document Conventions**

This Document was created based on the IEEE template for System Requirement Specification Documents.

## **Product Scope**

A keyword search system on text-based images can be related to corporate goals and business strategies with the purpose of giving a knowledge graph to users in several ways:

**1.Discoverability:** By using keyword search to classify and organize images, a company can make it easier for users to discover relevant information. A knowledge graph can help users to navigate the data and understand relationships between different concepts and ideas, which can support corporate goals of customer engagement and retention.

**2.Search experience:** By using keyword search to create a knowledge graph of images, a company can improve the search experience for users. This can support corporate goals of customer satisfaction and loyalty by providing users with a powerful tool for finding and understanding relevant information.

**3.Analysis and insights:** By using keyword search to classify and organize images and create a knowledge graph, a company can gain insights into user behavior and preferences. This can support corporate goals of data-driven decision-making and market research.

**4.Contextual information:** By using keyword search to classify and organize images in a knowledge graph, a company can make it easier for users to find and understand the context of images. This can support corporate goals of customer engagement and satisfaction by providing users with more meaningful and relevant information.

In general, a keyword search system on text-based images can support business strategies with the purpose of giving a knowledge graph to users by providing a powerful tool for discovering, understanding and analyzing images and their relationships, which can lead to improve user engagement, search experience, insights and contextual understanding. These are all aligned with corporate goals of providing a knowledge graph to users.

## **References**

*<List any other documents or Web addresses to which this SRS refers. These may include user interface style guides, contracts, standards, system requirements specifications, use case documents, or a vision and scope document. Provide enough information so that the reader could access a copy of each reference, including title, author, version number, date, and source or location.>*

# **Overall Description**

## **Product Perspective**

The product perspective of a keyword search on a text-based images system that is already present would be focused on the user's ability to quickly and easily search for specific text within images that have already been uploaded to the system. This would include features such as a user-friendly interface for inputting keywords, efficient image scanning and text recognition capabilities, and relevant search results that are easily accessible to the user. Additionally, the system would likely include tools for organizing and categorizing images based on their content, as well as the ability to share or export search results for further use. Overall, the goal of the product would be to make it as easy as possible for users to find and utilize the information contained within text-based images.

### **Product Functions**

* The ability to search for keywords within text-based images.
* The ability to search for keywords within text that has been OCR-processed (optical character recognition).
* The ability to display the related images of the searched keyword.
* The ability to highlight or annotate search results within the text-based image.

## 2.2 **User Classes and Characteristic**s

* Researchers: Researchers who use the keyword search on text-based images system to search for specific information or images in their field of study.

* Archivists: Archivists who use the system to search for and catalog images of historical documents or artifacts.

* Journalists: Journalists who use the system to search for images that are relevant to their news stories.

* Educators: Educators who use the system to search for images that can be used in their lessons or presentations.

* Social media managers: Social media managers who use the system to search for images that can be used in social media posts.

* Marketing professionals: Marketing professionals who use the system to search for images that can be used in marketing campaigns.

* Librarians: Librarians who use the system to search for images that can be added to library collections.

## **Operating Environment**

**1. Windows**

● **Processor**

Intel Pentium 4, Intel Centrino, Intel Xeon, or Intel Core Duo (or compatible)

1.8 GHz minimum

● **Memory**

1GB of RAM (4 GB recommended)

● **Disk Space**

256 MB of free disk space

**2. MacOs**

● **Processor**

Dual-Core Intel, PowerPC G5

● **Memory**

1GB of RAM (4 GB recommended)

● **Disk Space**

256 MB of free disk space

**3. Linux**

● **Processor**

Intel Pentium 4, Intel Centrino, Intel Xeon, or Intel Core Duo (or compatible)

1.8 GHz minimum

● **Memory**

1GB of RAM (4 GB recommended)

● **Disk Space**

256 MB of free disk space

**Software Requirements (Prerequisite Packages):**

**1. Internet Browser:** Firefox, Chrome or Safari (v11 or later)

**2. Database:**

● MongoDB

● Neo4j

**3. NodeJS and Express Web Framework**

## **Design and Implementation Constraints**

*<Describe any items or issues that will limit the options available to the developers. hardware limitations (timing requirements, memory requirements); interfaces to other applications; specific technologies, tools, and databases to be used; parallel operations; language requirements; communications protocols; security considerations; design conventions or programming standards*

* **Processing power limitations**: The keyword search on text-based images may require a significant amount of processing power to analyze and search through large amounts of data. This could limit the options available to developers in terms of the algorithms and techniques they can use to optimize the system's performance.

* **Memory limitations**: The system may require a large amount of memory to store the text-based images and the indexing data used for searching. This could limit the options available to developers in terms of the number of images that can be processed at one time and the number of keywords that can be searched for simultaneously.

* **Timing requirements**: The system may have strict timing requirements for performing keyword searches and returning results. This could limit the options available to developers in terms of the algorithms and techniques they can use to optimize the system's performance and ensure that it meets the timing requirements.

* **Data format limitations**: The system may only be able to process text-based images in a specific format, such as JPEG or PNG. This could limit the options available to developers in terms of the types of images that can be processed and the tools they can use to manipulate and analyze the images.

* **Language support:** The system may only support a limited number of languages, which could limit the options available to developers in terms of the types of text-based images that can be processed and the keywords that can be searched for.

## **User Documentation**

*<List the user documentation components (such as user manuals, on-line help, and tutorials) that will be delivered along with the software. Identify any known user documentation delivery formats or standards.>*

## **Assumptions and Dependencies**

*<List any assumed factors (as opposed to known facts) that could affect the requirements stated in the SRS. These could include third-party or commercial components that you plan to use, issues around the development or operating environment, or constraints. The project could be affected if these assumptions are incorrect, are not shared, or change.*

* **Language of the text in the images:** The system may not be able to accurately recognize and search text if it is written in a language that is not supported by the system.

* **Quality of the images**: The system may have difficulty recognizing text if the images are of poor quality or have been distorted in some way.

* **Format of the images:** The system may be designed to only recognize text in certain image formats, such as JPEG or PNG.

* **Size of the images:** The system may not be able to accurately recognize text in images that are too small or too large.

* **Complexity of the text:** The system may have difficulty recognizing text that is written in a complex font or style.

* **Number of images:** The system may be designed to only handle a certain number of images at a time, and may not be able to handle large volumes of images.

# **External Interface Requirements**

## **User Interfaces**

*<Describe the logical characteristics of each interface between the software product and the users. This may include sample screen images, any GUI standards or product family style guides that are to be followed, screen layout constraints, standard buttons and functions (e.g., help) that will appear on every screen, keyboard shortcuts, error message display standards, and so on. Define the software components for which a user interface is needed.*

**Login page**

The login page is an essential part of any website. It contains a header with the name of the website, a logo that represents the brand, and a form for users to enter their email or username and password. There is also a submit button for users to log in and a forgotten password link for those who need it.

**Home page**

The home page of a keyword search system for text-based images is designed to provide a user-friendly and efficient experience for users looking for specific information. The page includes a search bar where users can input keywords to search for, and a filter bar that allows them to narrow down the results based on date, type, and other parameters. The results area displays the search results, including thumbnails of the images and their associated text. The images can be previewed by hovering over the thumbnails, and pagination controls allow users to navigate through the search results, including buttons for going to the next or previous page and drop-down menus for jumping to specific pages. This user interface provides an intuitive and streamlined experience for users who need to quickly find the information they need.

**Upload images page**

The upload images user interface page for a keyword search system for text-based images is a crucial component of the system, allowing users to add new images and their associated text to the database. The page includes the following elements:

* **File upload button**: A button that allows users to select the image files they want to upload from their device.

* **Text input fields**: Input fields for users to enter the text associated with each image, including information such as the title, description, and keywords.

* **Preview area**: An area that displays a preview of the uploaded images and their associated text, giving users the opportunity to verify the information before submitting.

* **Submit button**: A submit button that users can click to upload the images and their associated text to the database.

* **Error handling**: Error handling mechanisms to ensure that users are notified if there are any issues with the upload process, such as missing information or file type restrictions.

The upload images user interface page provides an intuitive and straightforward process for users to add new images and their associated text to the keyword search system, making it easy to keep the database up-to-date and accurate.

**Graph page**

The graph page user interface, which uses the knowledge graph technology, is an essential component of the keyword search system for text-based images. It displays the relationships between the images and their associated text in a visual and interactive format, allowing users to explore and understand the data in a more intuitive way. The following elements are typically included:

* **Graph visualization**: A visual representation of the data, such as a node-link diagram, that displays the relationships between the images and their associated text.

* **Node information**: Information about each node in the graph, including details such as the image and its associated text, as well as any other relevant information.

* **Node interaction**: The ability for users to interact with the nodes in the graph, such as zooming in and out, panning, and selecting individual nodes to view more information.

* **Query functionality**: The ability for users to submit a query to the graph, such as searching for a specific image or keyword, and view the results in the graph visualization.

The graph page user interface provides a powerful and intuitive way for users to explore and understand the relationships between the images and their associated text, making the keyword search system for text-based images much more accessible and valuable.

**Catalog Page**

The catalog page renders the details of a user searched node and details. This catalog would display the search related info of that keyword in history that will be queried from the document database. It would also incorporate features of documenting the related keywords and user messages involved during graph updation.

## **Software Interfaces**

* Bcrypt is a password-hashing function used in the BcryptJS npm module based on the Blowfish cipher. This module is used for password encryption before storing it in the database. After each login the hashed and salted password is compared with the password in the database.
* JSON Web Token (JWT) is an open standard (RFC 7519) that defines a compact and self-contained way for securely transmitting information between parties as a JSON object. We use HMAC SHA256 algorithm for generating the signature. The token is generated only on successful login and a new token is generated on every successful login. These tokens are validated before providing any functionality to the user.

# **System Features**

The following features are required to be implemented in the application.It can be divided into the following components:

## Graph creation and updation from text based images

The system features for Graph creation and updation for text-based images using the knowledge graph technique would include the ability to extract textual information from images and create nodes in a Neo4j database. Additionally, the system should be able to identify relationships between nodes and update the graph accordingly.

**4.1.1 Description and Priority**

The description of the project would focus on the development of a system that can extract information from text-based images and represent that information in a Neo4j graph. The priority of the project would be to develop a system that can accurately extract information and create a meaningful representation in the graph.This is a MEDIUM priority feature to be implemented.

**4.1.2 Stimulus/Response Sequences**

The stimulus for the system would be an image with text, and the response would be the creation or update of nodes and relationships in the Neo4j graph. The system should be able to identify the text in the image, extract relevant information, and use that information to update the graph.

**4.1.3 Functional Requirements**

Functional requirements for the project would include

REQ-1: Ability to accurately extract information from text-based images (OCR)

REQ-2: Create nodes in a Neo4j database

REQ-3: Identify relationships between nodes

REQ-4: Updating the graph with related custom keywords

REQ-5: Real-time updation of graph

## Search for keywords and catalog search

**4.2.1 Description and Priority**

The system should allow users to search for keywords in text-based images, using a simple and intuitive user interface.The search results should be displayed in a clear and organized manner, with relevant information about each image.The system should be able to recognize and suggest related keywords, based on the user's search history and the relationships between keywords in the knowledge graph.The functionality of the search keyword functionality user interface should be the top priority, as it is the main feature that users will interact with.

**4.2.2 Stimulus/Response Sequences**

The user enters a keyword into the search field.The system processes the keyword and displays relevant images.The user is given the keyword and related keywords graph. The user clicks on a node to view more information.

**4.2.3 Functional Requirements**

**REQ-1** The system should be able to search for keywords in text-based images, using the knowledge graph technique.

**REQ-2** The system should be able to store and retrieve keyword information in a Neo4j and MongoDB database.

# **Other Nonfunctional Requirements(include which is applicable for the project)**

## **Performance Requirements**

The keyword search on text-based images may require a significant amount of processing power to analyze and search through large amounts of data. This could limit the options available to developers in terms of the algorithms and techniques they can use to optimize the system's performance. The system may require a large amount of memory to store the text-based images and the indexing data used for searching. The system may have strict timing requirements for performing keyword searches and returning results. The system may only be able to process text-based images in a specific format, such as JPEG or PNG. This could limit the options available to developers in terms of the types of images that can be processed and the tools they can use to manipulate and analyze the images.

## **Safety Requirements**

**Input Validation:** Ensure that all user input is validated and sanitized to prevent malicious attacks such as cross-site scripting (XSS) and SQL injection.

**Session Management:** Implement secure session management to prevent session hijacking, where an attacker gains unauthorized access to a user’s session.

**Password Management:** Store passwords securely using strong encryption algorithms, such as bcrypt, to prevent password cracking and unauthorized access to sensitive data.

**Error Management:** Handle and log errors properly to prevent information leakage and prevent attackers from exploiting the errors for malicious purposes.

## **Security Requirements**

**Authentication:** Implement strong authentication mechanisms, such as two-factor authentication or biometric authentication, to ensure that only authorized users have access to sensitive data and systems.

**Access Control:** Implement fine-grained access control mechanisms to ensure that only users with the necessary permissions have access to sensitive data and functionality.

**Encryption:** Encrypt sensitive data, both in transit and at rest, to prevent unauthorized access and data theft.

**Network Security:** Ensure that the web application is deployed on a secure network, with appropriate firewalls and intrusion detection systems in place to prevent unauthorized access and attacks.

**Regular Updates and Patches:** Regularly update the web application and its components to address security vulnerabilities and prevent attacks. Regularly perform security audits and penetration testing to identify and fix security weaknesses.

## **Software Quality Attributes**

*<Specify any additional quality characteristics for the product that will be important to either the customers or the developers. Some to consider are: adaptability, availability, correctness, flexibility, interoperability, maintainability, portability, reliability, reusability, robustness, testability, and usability. Write these to be specific, quantitative, and verifiable when possible. At the least, clarify the relative preferences for various attributes, such as ease of use over ease of learning.>*

**5.4.1 Flexibility:**

**Modular Design**: Ensure that the web application is designed using modular components, allowing for easy modification and replacement of individual components without affecting the entire application.

**Configuration Management:** Implement a configuration management system to manage different environments, such as development, testing, and production, and to allow for easy configuration of the web application.

**Scalability:** Ensure that the web application is designed to be scalable, allowing for easy increase in capacity as the number of users and data grows.

**Interoperability:** Ensure that the web application is designed to be interoperable with other systems and technologies, allowing for easy integration with other systems and technologies.

**5.4.2 Maintainability:**

**Code Readability:** Ensure that the code is written in a readable and understandable format, making it easier to maintain and update the code.

**Documented Code:** Ensure that the code is well-documented, making it easier for developers to understand and maintain the code.

**Version Control:** Implement a version control system to manage the code, allowing for easy rollback in case of issues, and making it easier for multiple developers to collaborate on the code.

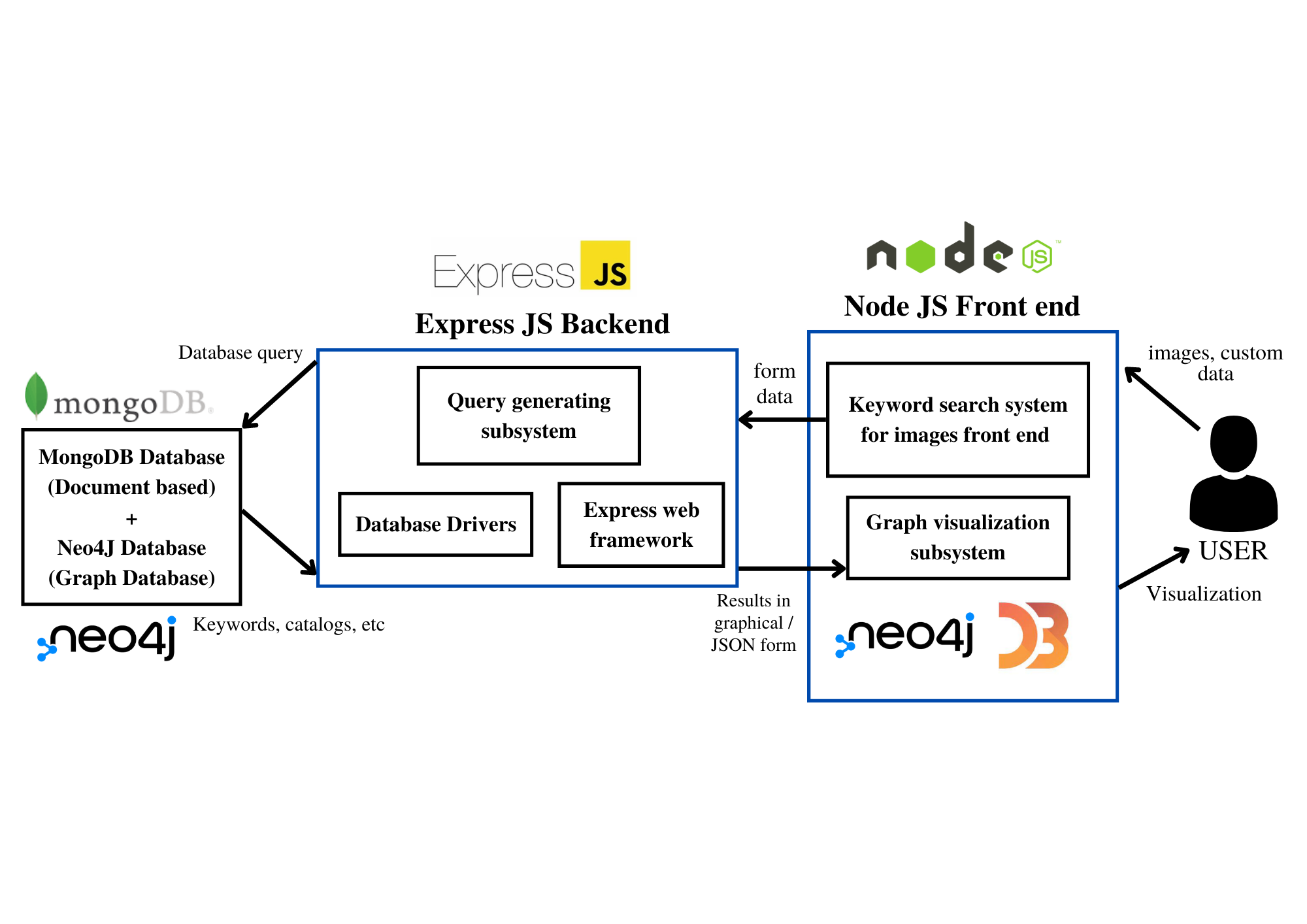
**Appendix A: Glossary**

*<Define all the terms necessary to properly interpret the SRS, including acronyms and abbreviations..>*

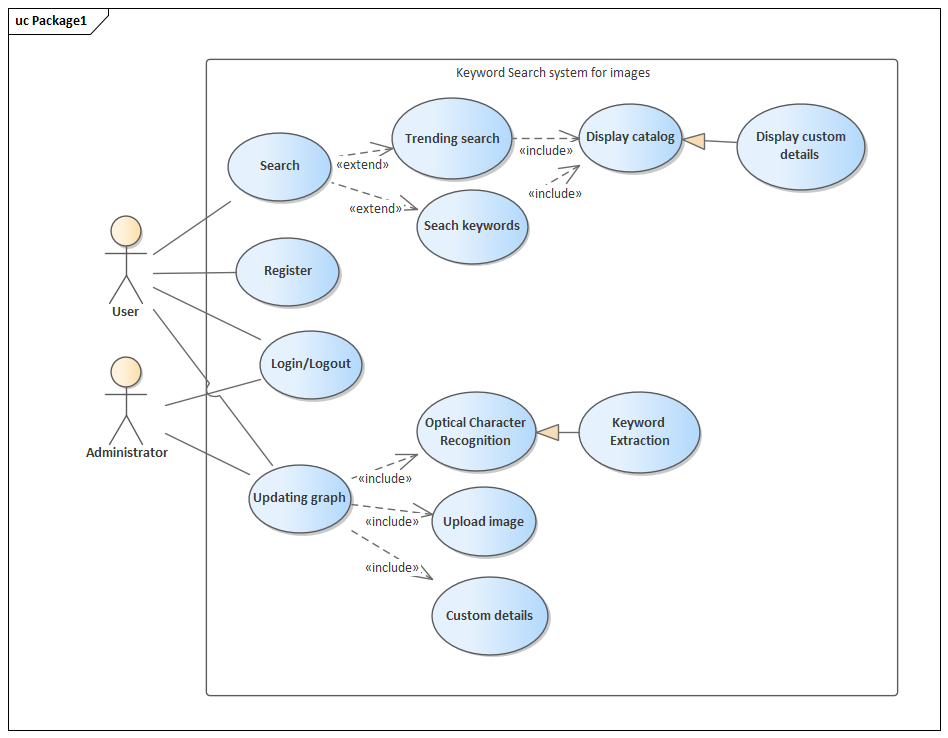
**Appendix B: Analysis Models**

*<Optionally, include any pertinent analysis models, such as data flow diagrams, class diagrams, state-transition diagrams, or entity-relationship diagrams*.>

**1.System Architecture**

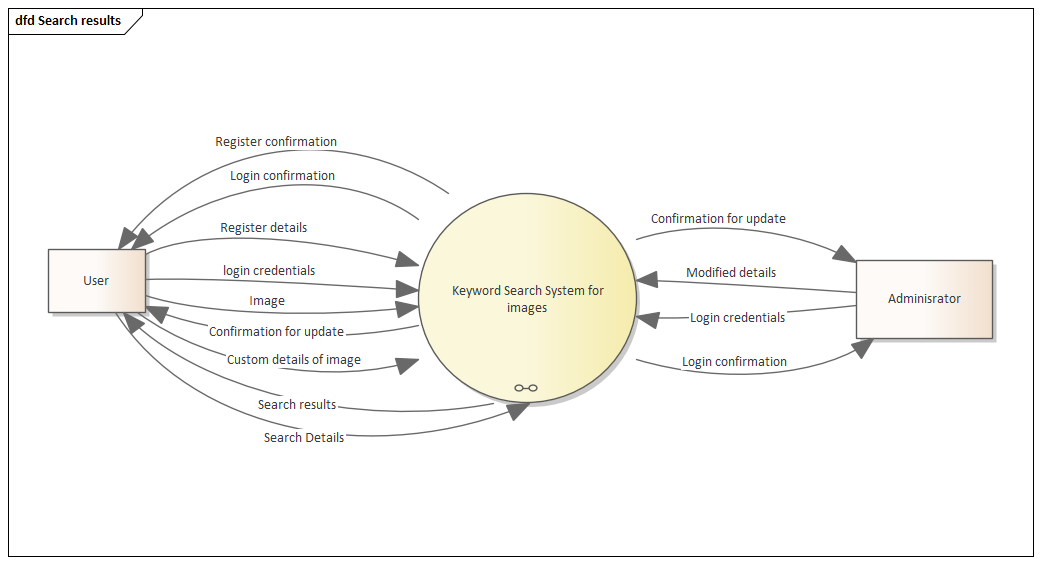
****

**2. Use Case Diagram**

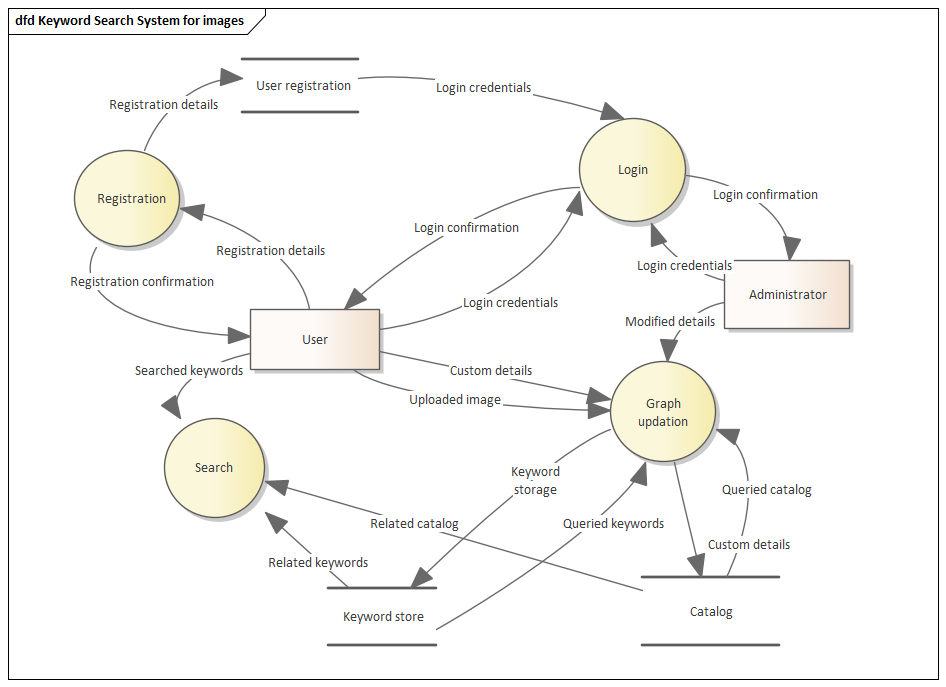
****

**3.Data Flow Diagrams**:

**3.1 DFD LEVEL 0**

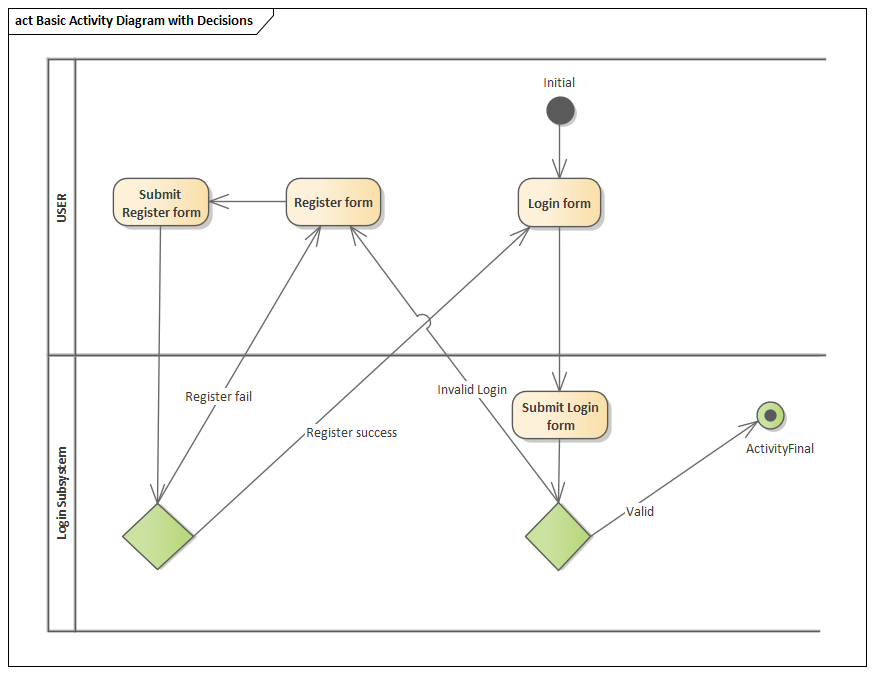
****

**3.2 DFD LEVEL 1**

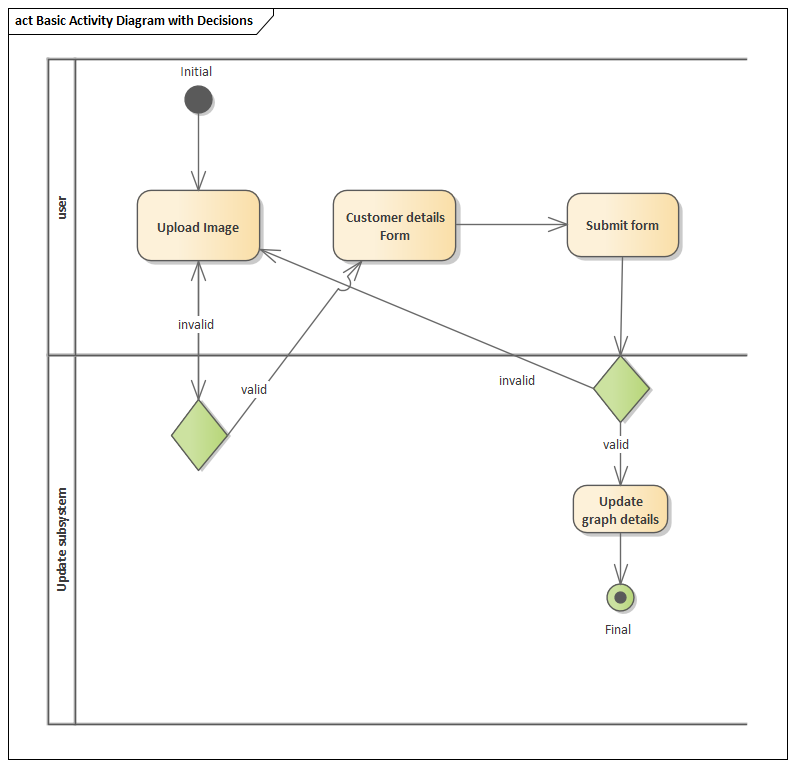
****

**4. Activity Diagrams**

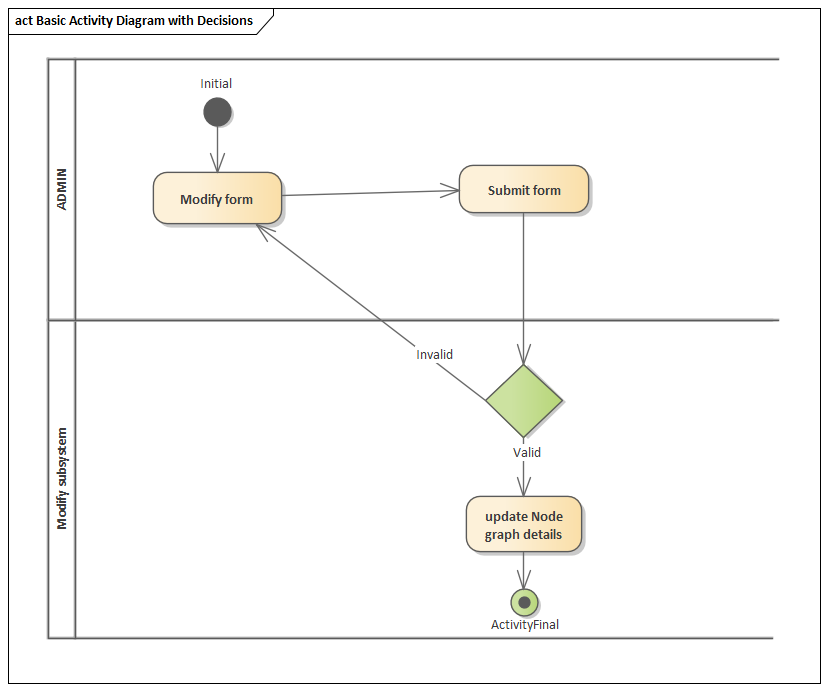
**4.1 Login Operation**

****

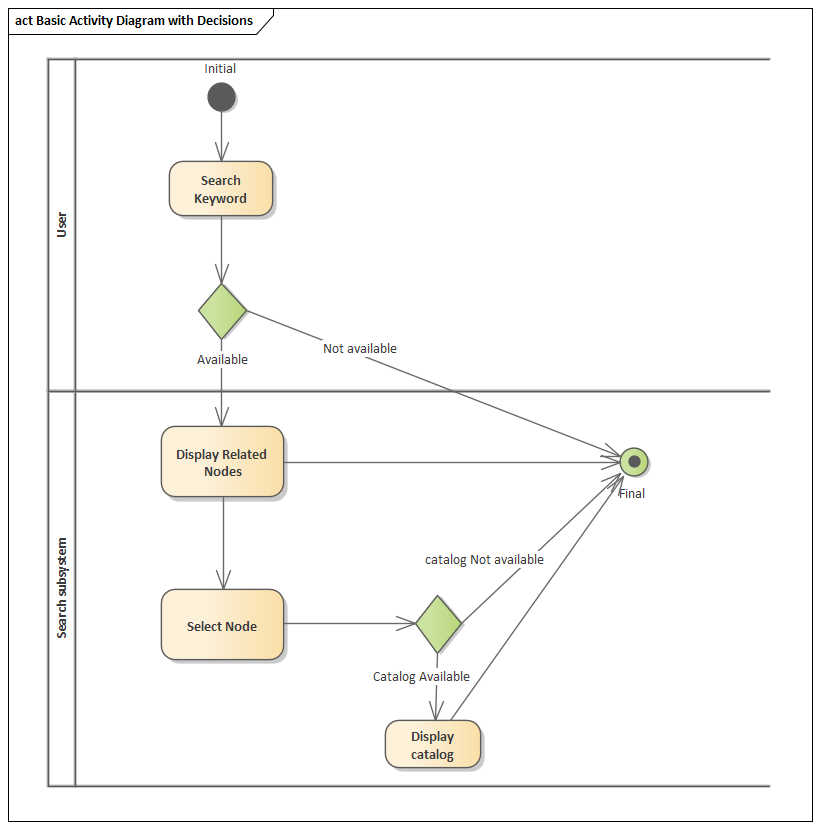
**4.2 Custom Update Operation**

****

**4.3 Modify Operation (Admin Only)**

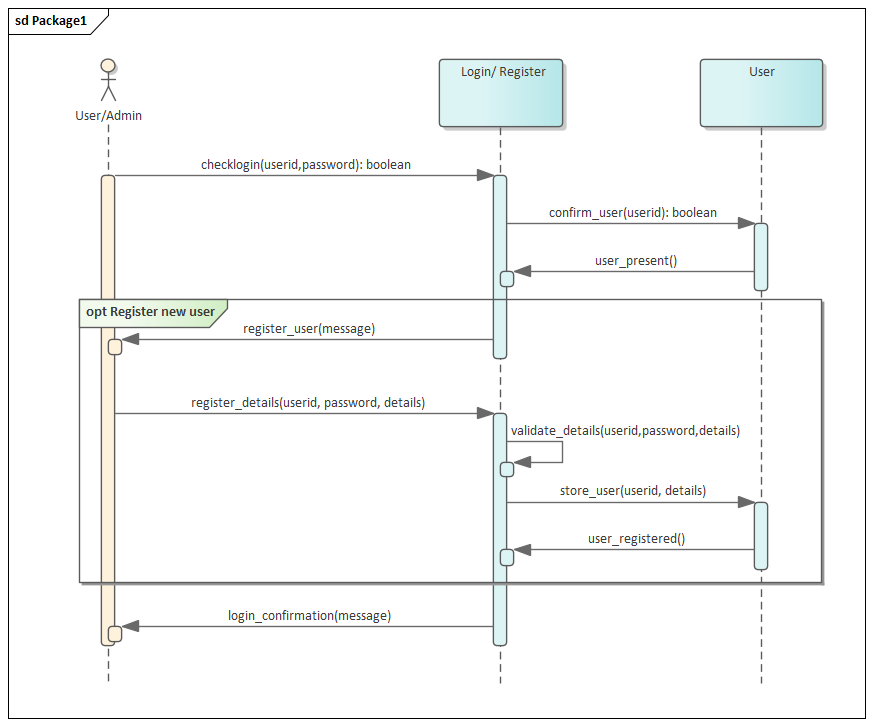
****

**4.4 Search Operation**

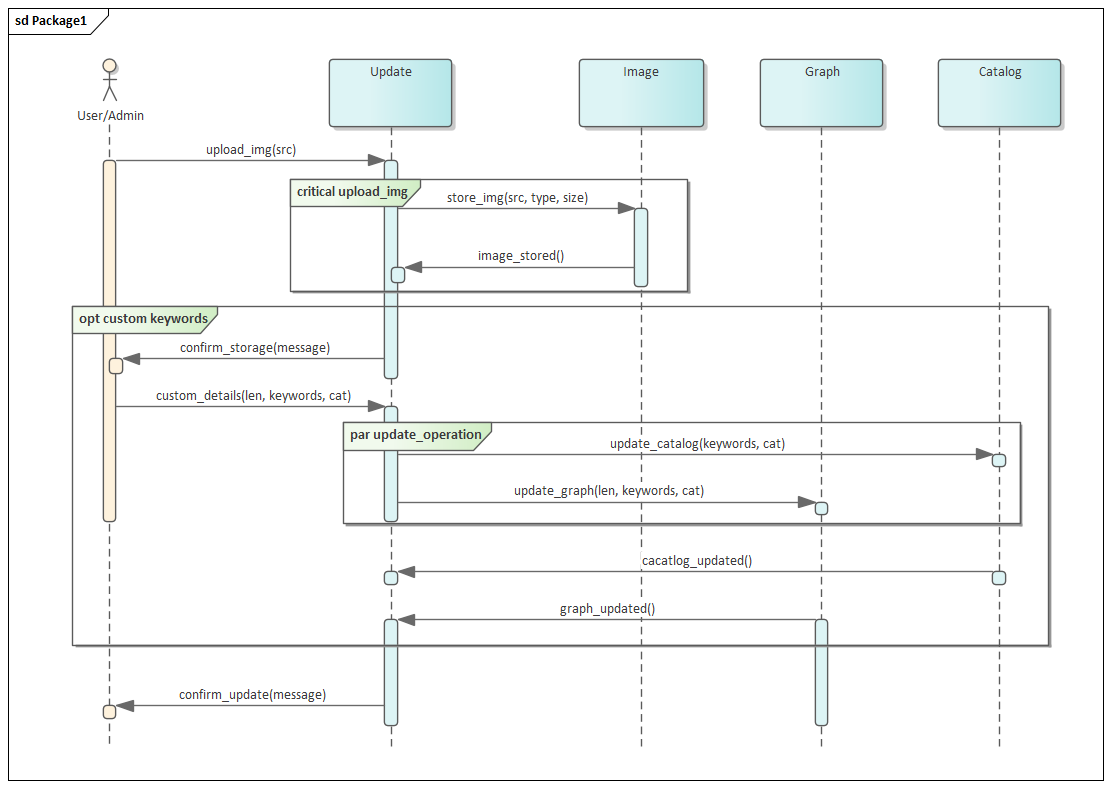
****

**5. Sequence Diagram**

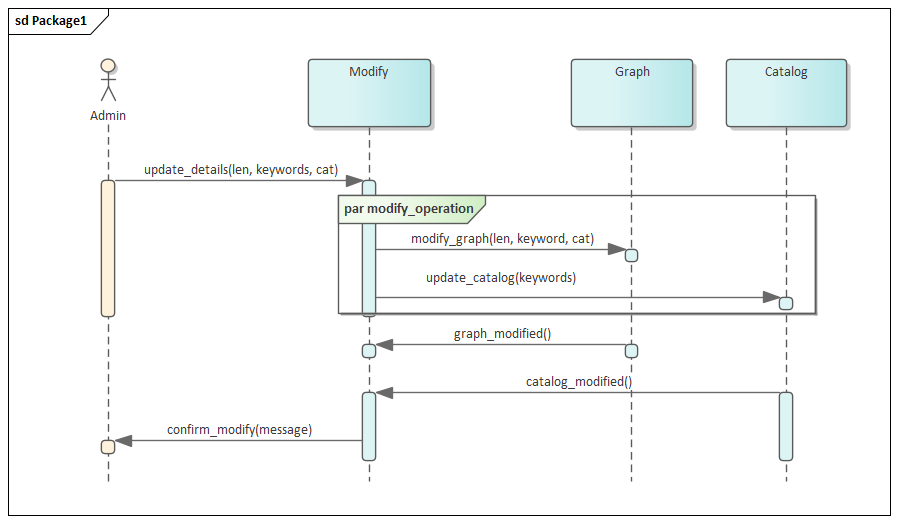
**5.1 Login/Register Operation**

****

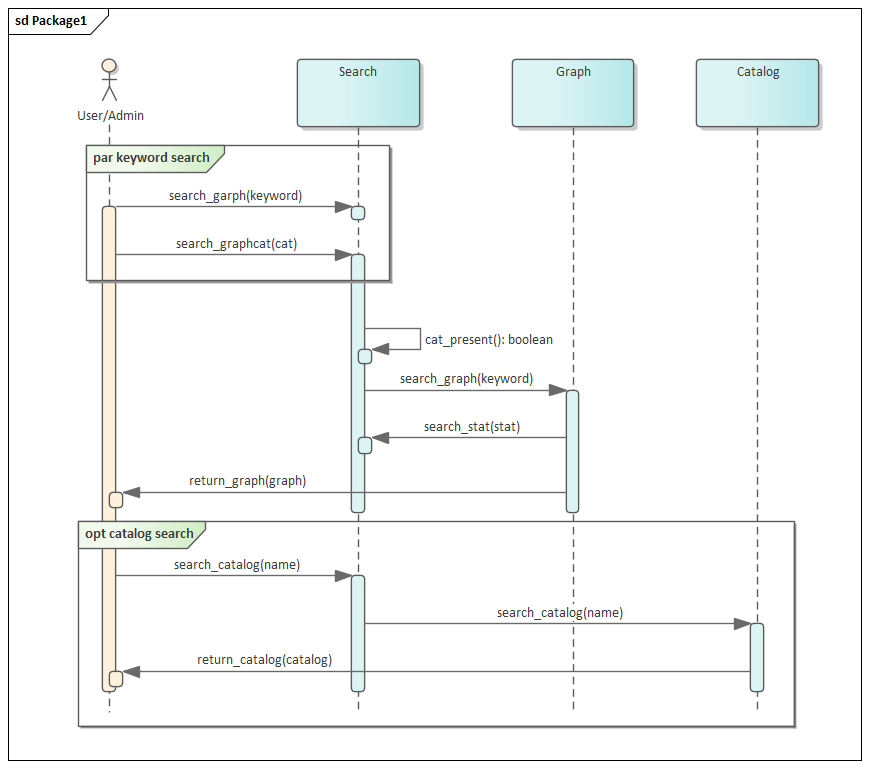
**5.2 Custom Update Operation**

****

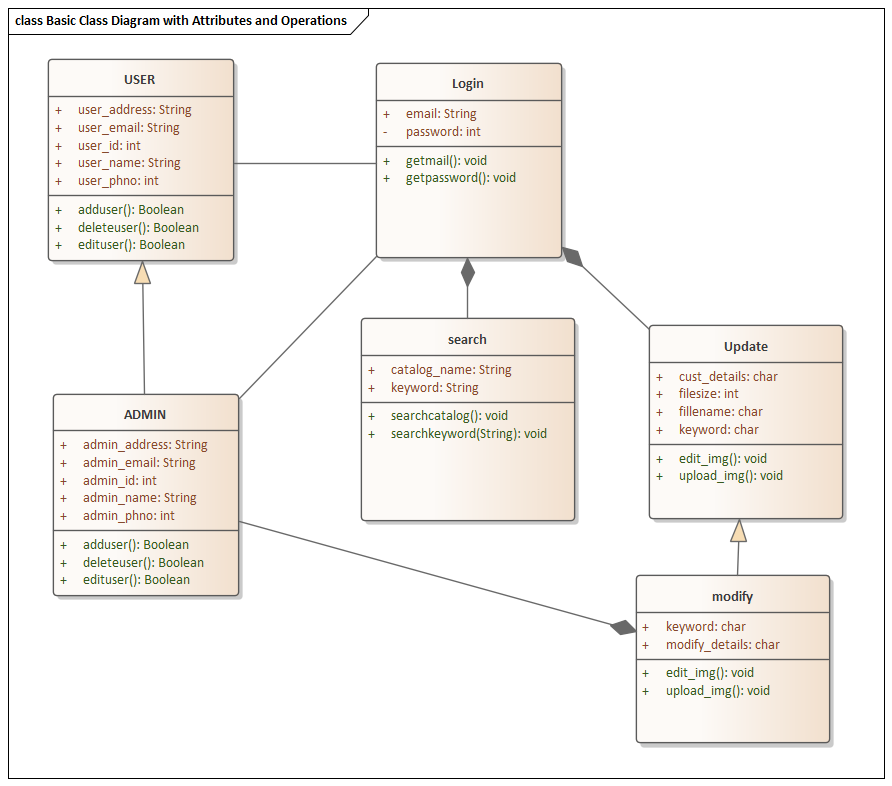
**5.3 Modify Operation**

****

**5.4 Search Operation**

****

**6. Class Diagram**

****