



University of the Punjab
Gujranwala Campus

Second Deliverable

Project ID: CC-411-3



TABLE OF CONTENTS

1.0	Introduction	3
1.1	Use case description	3
1.2	Use case Diagram (refined and updated)	9
1.3	Domain Model	9
1.4	Sequence Diagram	11
1.5	Collaboration Diagram.....	14
1.6	Operation Contracts	17
1.7	Design Class Diagram.....	19
1.8	Data Model	20



1.0 Introduction

The second deliverable is all about the use case modeling and software design. In the previous deliverable, analysis of the system is completed. So, we understand the current situation of the problem domain. Now we are ready to strive for a solution for the problem domain by using object-oriented approach. Following artifacts must be included in this deliverable.

1. Use case description
2. Use case diagram refined
3. Domain Model
4. Sequence Diagram
5. Collaboration Diagram
6. Operation Contracts
7. Design Class Diagram
8. Data Model

Now we discuss these artifacts one by one as follows:

1.1 Use case description

1. UC_Login

Primary Actor: User

Brief Description: This use case describes the process of a user registering & logging into the system.

Pre-Conditions:

- The system is running.
- The user has a valid username and password.

Basic flow:

- The user navigates to the login page.
- The system presents the login form.
- The user enters their username and password.
- The user submits the login form.
- The system verifies the credentials.
- If the credentials are valid, the system grants access to the user's account.

Alternate flow:

- If the credentials are invalid, the system displays an error message and prompts the user to re-enter their login information.

Handler: LoginController.

**Post Conditions:**

- If the login is successful, the user gains access to their account and can proceed with their intended actions within the system.
- If the login fails, the user remains on the login page with an error message displayed.

2. UC_Maintain_Packages**Primary Actor:** Admin**Brief Description:** This use case involves the administration of subscription packages available for users to purchase.**Pre-Conditions:**

- The admin is logged into the system.
- The system is running.

Basic flow:

- The admin navigates to the package management section.
- The system presents the list of existing subscription packages.
- The admin selects a package to modify or creates a new package.
- If modifying an existing package then the administrator updates the package details such as name, duration, price, and features.
- If creating a new package then the admin fills in the details for the new package, including name, duration, price, and features.
- The admin saves the changes or creates the new package.

Alternate flow:

- If there are no existing packages, the administrator may create a new package directly without selecting an existing one.

Handler: PackageAdminController.**Post Conditions:**

- The changes made by the admin to the subscription packages are saved in the system.
- Users can view and purchase the updated or newly created subscription packages from their accounts.

3. UC_Payment**Primary Actor:** User**Brief Description:** This use case involves the process of a user making a payment to purchase a subscription package and obtain credits for further use within the system.**Pre-Conditions:**

- The user is logged into the system.
- The system is running.

**Basic flow:**

- The user navigates to the subscription purchase section.
- The system presents the available subscription packages for purchase.
- The user selects a subscription package they wish to purchase.
- The system displays the details of the selected package along with the payment options.
- The user chooses a payment method (e.g., credit card, PayPal).
- The user provides the necessary payment information.
- The system processes the payment transaction securely.
- Upon successful payment processing, the user's account is credited with the corresponding credits associated with the purchased subscription package.

Alternate flow:

- If the payment transaction fails, the system displays an error message and prompts the user to retry or choose a different payment method.

Handler: PaymentController.

Post Conditions:

- If the payment is successful, the user's account is updated with the purchased subscription package and associated credits.
- If the payment fails, no changes are made to the user's account, and the user is prompted to retry the payment.

4. UC_Generate_Keywords

Primary Actor: User

Brief Description: This use case involves the generation of optimized keywords using AI or web scraping techniques based on a given keyword.

Pre-Conditions:

- The admin is logged into the system.
- The system is running.

Basic flow:

- The user navigates to the "Generate Keywords" section within the system.
- The user enters a primary keyword for which they want to generate optimized keywords.
- The system offers options for generating keywords:
- The system utilizes AI algorithms to generate a list of optimized keywords related to the primary keyword.
- The system scrapes relevant websites or search engine results pages (SERPs) to extract keywords related to the primary keyword.
- The system processes the selected keywords and generates a list of optimized keywords.
- The user can review and customize the generated keywords if necessary.

**Alternate flow:**

- The system may require additional time for processing depending on the complexity of the keyword and the AI algorithms used.
- The system may encounter delays or errors due to fluctuations in web data availability or changes in website structures.

Handler: KeywordGenerationController.

Post Conditions:

- The user receives a list of optimized keywords based on the given keyword.
- The generated keywords can be used for SEO purposes to improve the visibility and ranking of the user's content on search engines.

5. UC_Blog

Primary Actor: User

Brief Description: This use case involves the process of a user writing a blog post using the generated optimized keywords and receiving a ranked percentage based on the usage of those keywords.

Pre-Conditions:

- The user is logged into the system.
- The system has generated optimized keywords related to the blog topic.

Basic flow:

- The user navigates to the "Blog" section within the system.
- The user initiates the creation of a new blog post.
- The user enters the title and content of the blog post.
- The user selects relevant optimized keywords generated by the the system to include in the blog post.
- The system analyzes the usage of the selected keywords within the blog post content.
- The system calculates a ranked percentage indicating how well the selected keywords are integrated into the blog post.
- The user can review the ranked percentage and make adjustments to the blog post if necessary to improve keyword integration.
- The user submits the blog post for store & publishing.

Alternate flow:

- If the user does not select any optimized keywords, the system may prompt the user to choose keywords before proceeding.

Handler: BlogsController.

**Post Conditions:**

- The blog post is saved in the system's database and is available for publishing.
- The user receives feedback on the ranked percentage indicating the effectiveness of keyword integration in the blog post.
- The system may update its keyword analysis algorithms based on user interactions and feedback for future improvements.

6. UC_Recover_Password_&_Update_Profile**Primary Actor:** User**Brief Description:** This use case involves the user's ability to recover their password and update their profile information within the system.**Pre-Conditions:**

- The user is registered and logged into the system.
- The system is running.

Basic flow:

- The user navigates to the "Account Settings" section within the system.
- The user selects the option to reset their password or update their profile.
- If resetting password then the system presents the user with a form to enter their registered email address or username. The user submits their email address or username. The system verifies the provided information. If the information is valid, the system sends a password reset link to the user's email address. The user clicks on the password reset link received in their email. The system prompts the user to enter a new password. The user enters a new password and confirms it. The system updates the user's password in the database.
- If updating profile then the system displays the user's current profile information. The user makes the desired changes to their profile information (e.g., name, email address, profile picture). The user saves the changes. The system validates the updated information. If the information is valid, the system updates the user's profile in the database.

Alternate flow:

- If the provided email address or username is not found in the system during password recovery, the system displays an error message prompting the user to verify the information and try again.
- If the user attempts to save invalid profile information (e.g., invalid email format) during profile updating, the system displays an error message and prompts the user to correct the information.

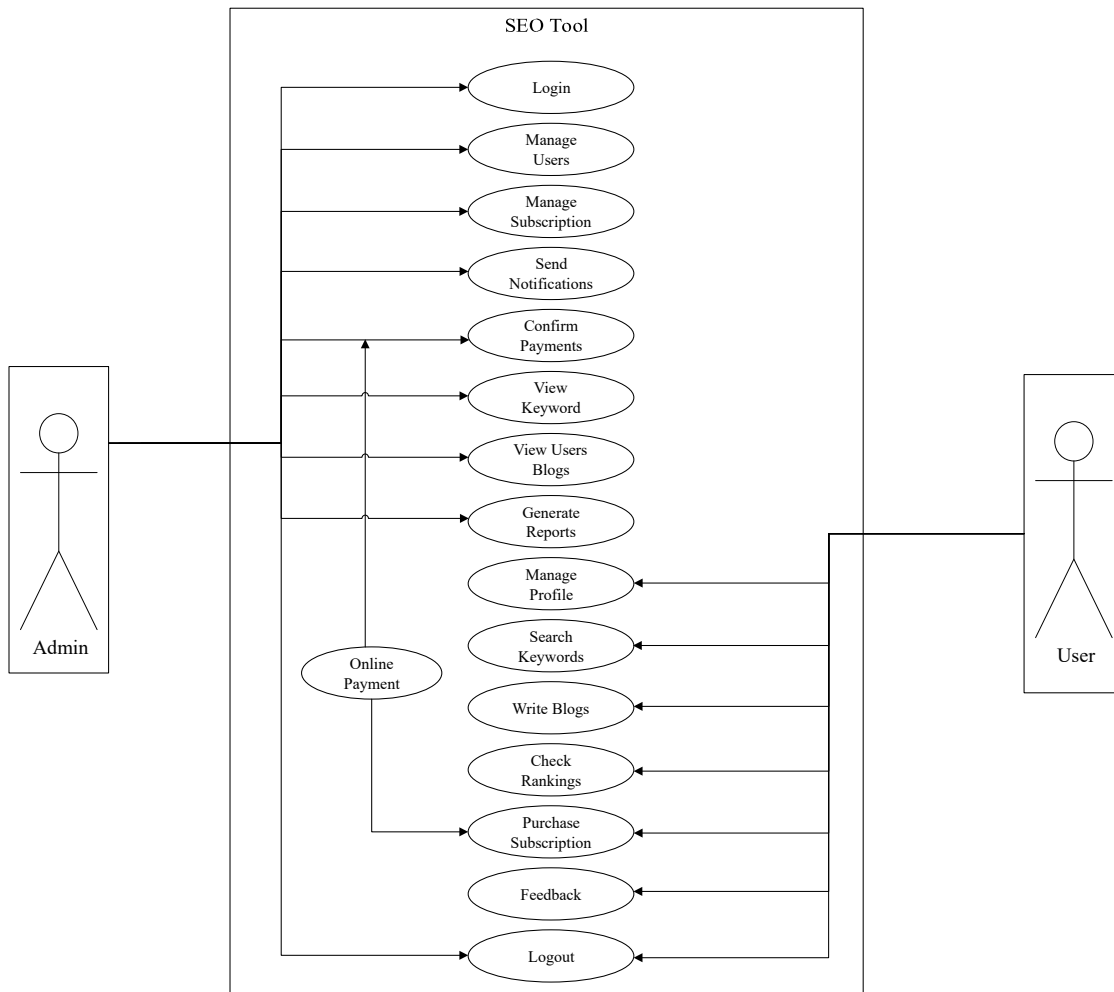
Handler: ProfileController.



Post Conditions:

- The user's password is successfully updated in the system.
- The user can log in using the new password.
- The user's profile information is successfully updated in the system.
- The updated profile information is displayed to the user and used for system interactions.

1.2 Use case Diagram (refined and updated)



1.3 Domain Model

1. User:

- Attributes: UserID, Username, Email, Password
- Relationships:
 - Users can have different roles (e.g., admin, content creator, marketer).



2. **Keyword:**

- Attributes: KeywordID, Phrase, Search Volume, Competition Level
- Relationships:
 - Keywords are used for various functionalities such as keyword research and content optimization.

3. **Content:**

- Attributes: ContentID, Title, Body, Metadata
- Relationships:
 - Content is analyzed and optimized for SEO purposes, including readability, keyword density, and metadata.

4. **Competitor:**

- Attributes: CompetitorID, Name, Website URL
- Relationships:
 - Competitors' websites are analyzed to identify keywords, backlink profiles, and on-page optimization strategies.

5. **Ranking:**

- Attributes: RankingID, KeywordID, URL, Position
- Relationships:
 - Rankings track the position of target keywords for specific URLs over time.

6. **Audit:**

- Attributes: AuditID, URL, Issues, Recommendations
- Relationships:
 - Audits identify on-page SEO issues and provide recommendations for optimization.

7. **Link:**

- Attributes: LinkID, Source URL, Destination URL, Anchor Text
- Relationships:
 - Links are analyzed for their quality and relevance, and opportunities for link building are identified.



8. Location:

- Attributes: LocationID, Name, Latitude, Longitude
- Relationships:
 - Locations are relevant for local SEO optimization, including managing citations and monitoring reviews.

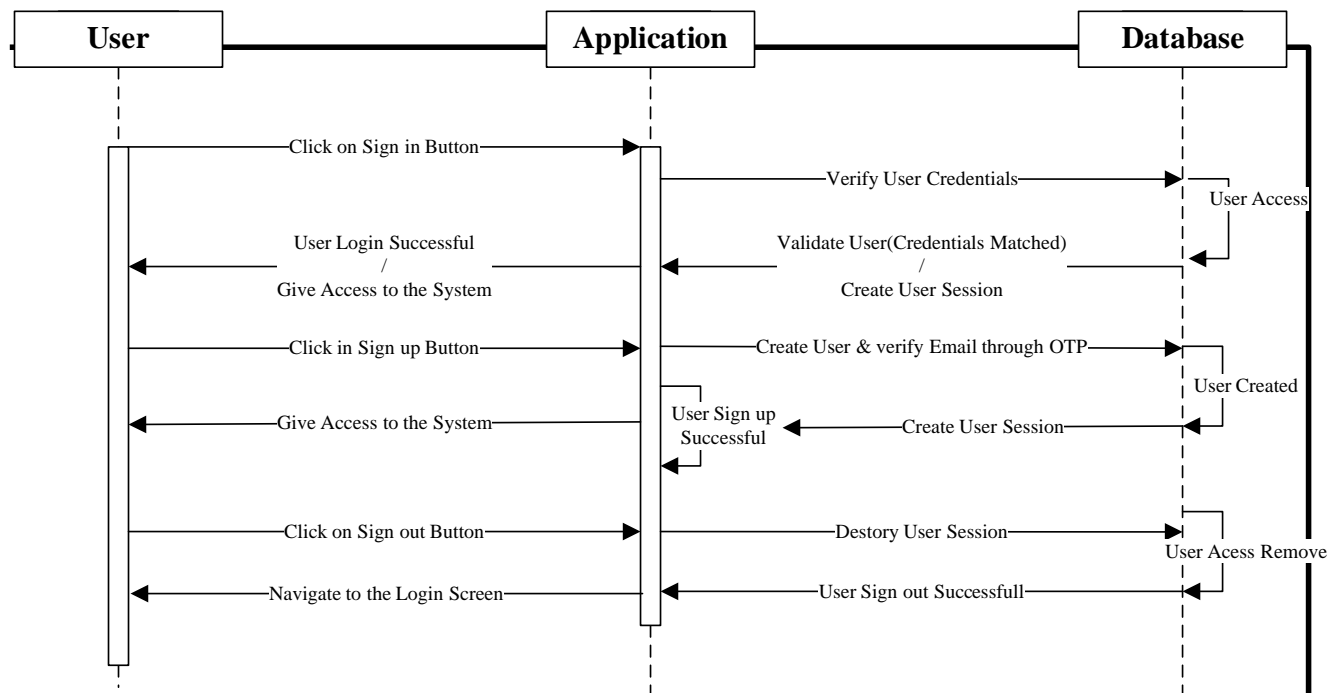
9. Report:

- Attributes: ReportID, UserID, Date, Metrics
- Relationships:
 - Reports summarize SEO performance metrics such as organic traffic, keyword rankings, and backlink growth.

This domain model captures the essential entities involved in this project and their relationships. Depending on the specific features and functionalities of our tool, we may need to expand or modify this model to accurately represent the system's domain.

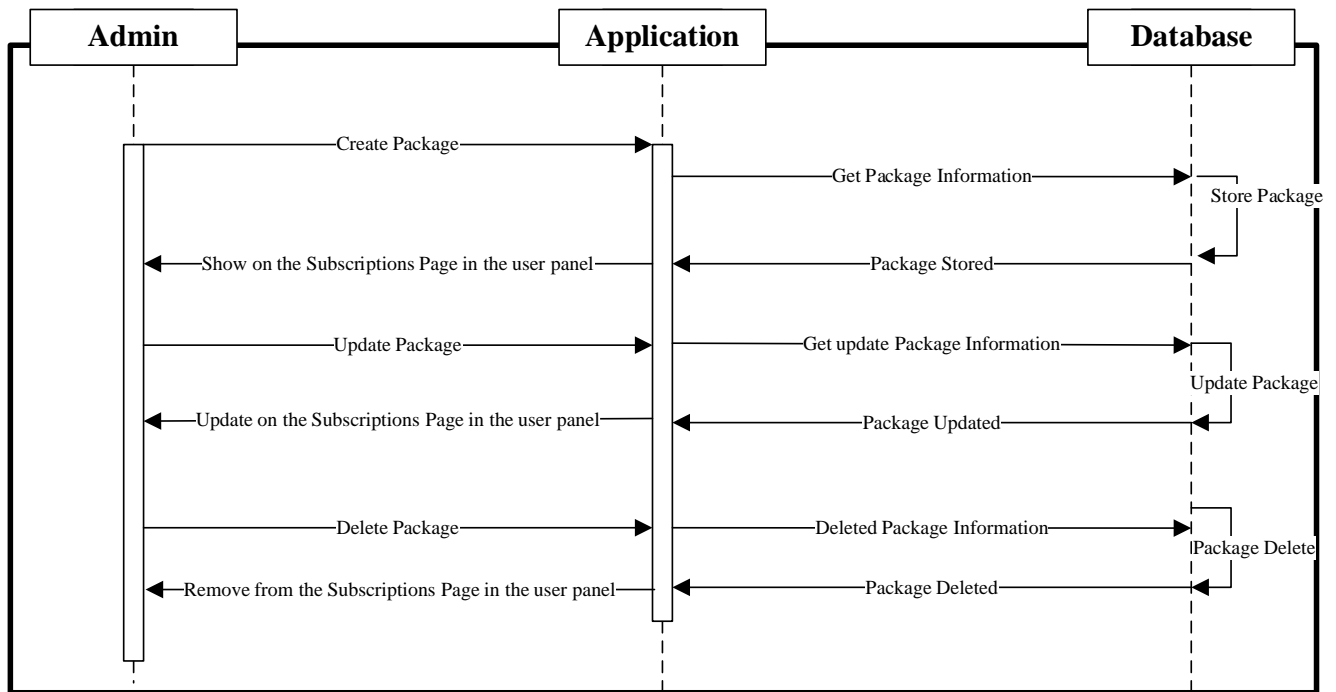
1.4 Sequence Diagram

1. SD_Login

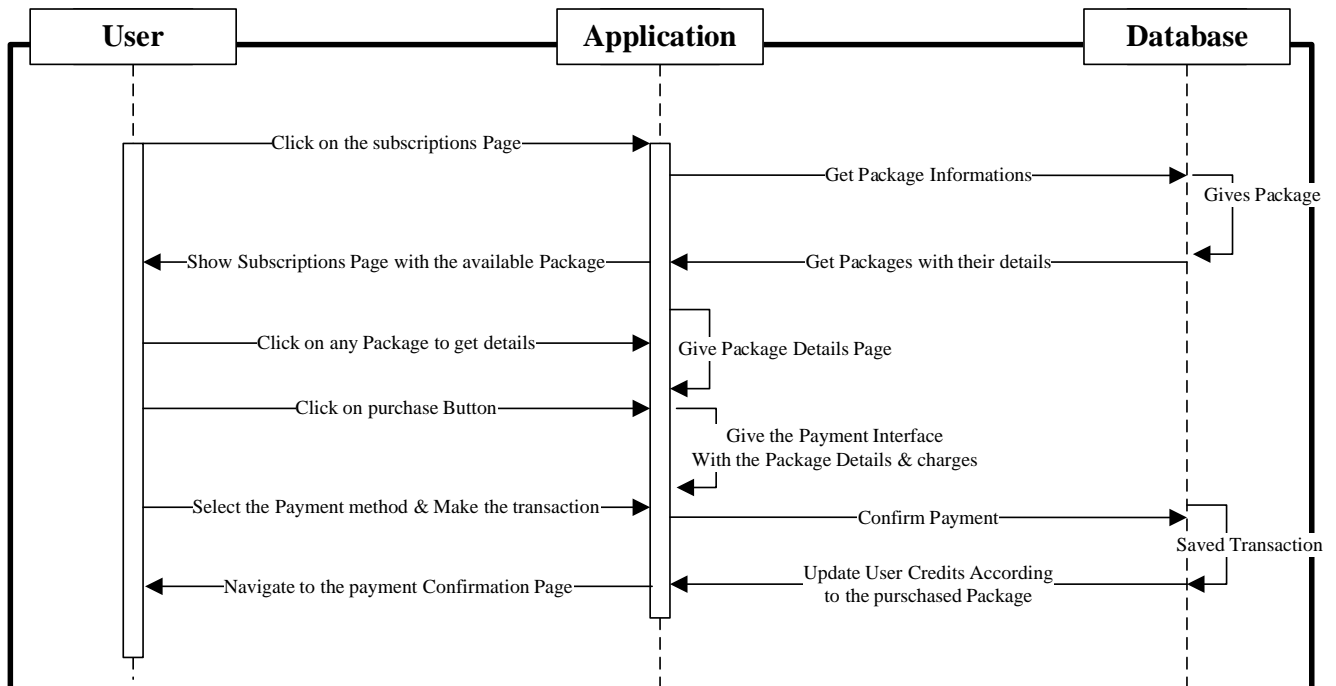




2. SD_Maintain_Packages

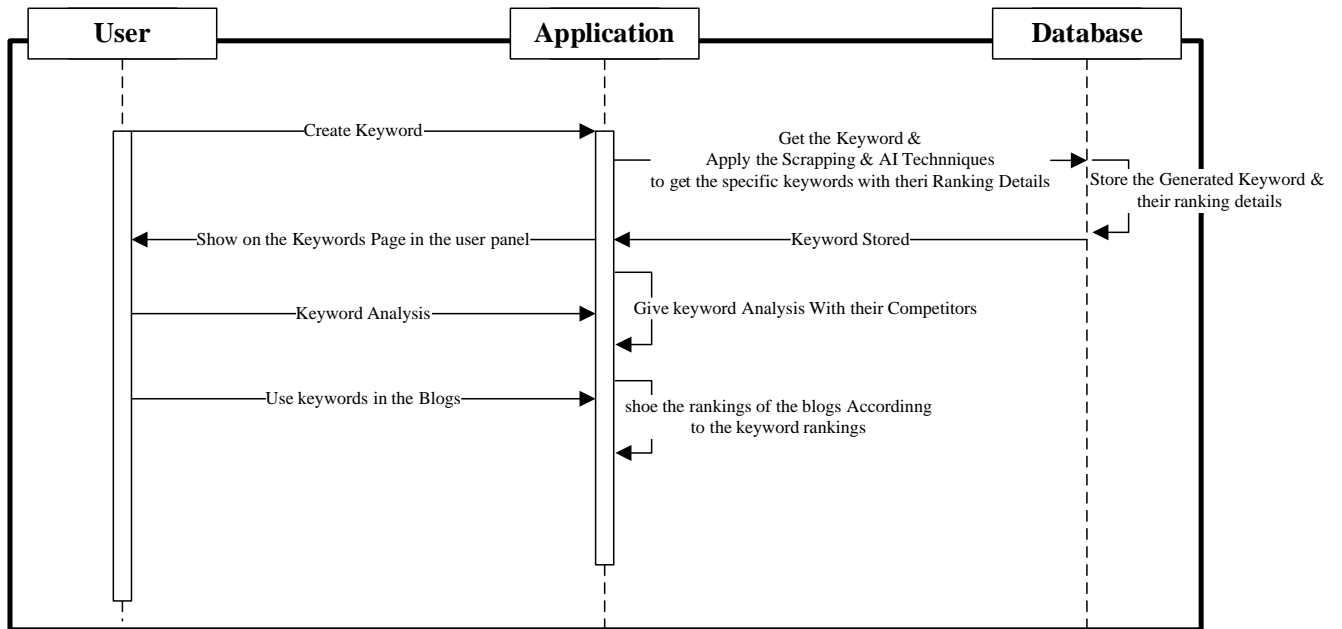


3. SD_Payment

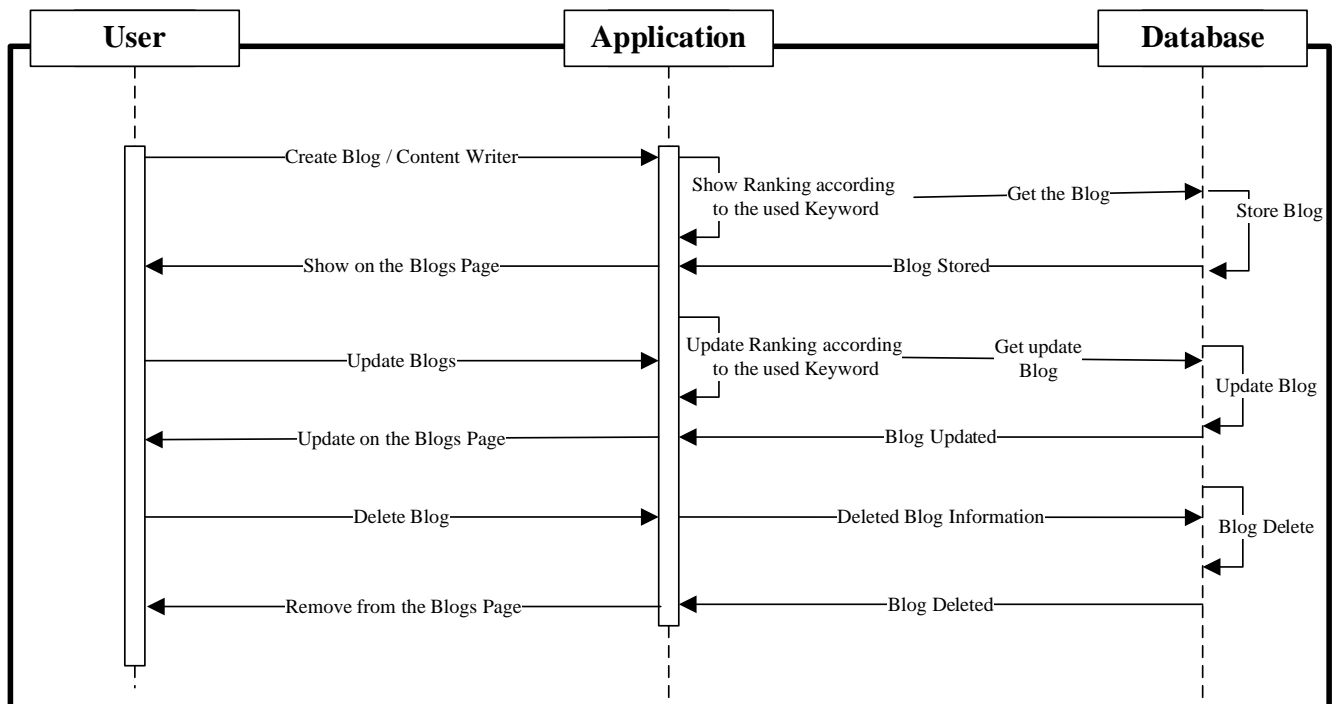




4. SD_Generate_Keywords

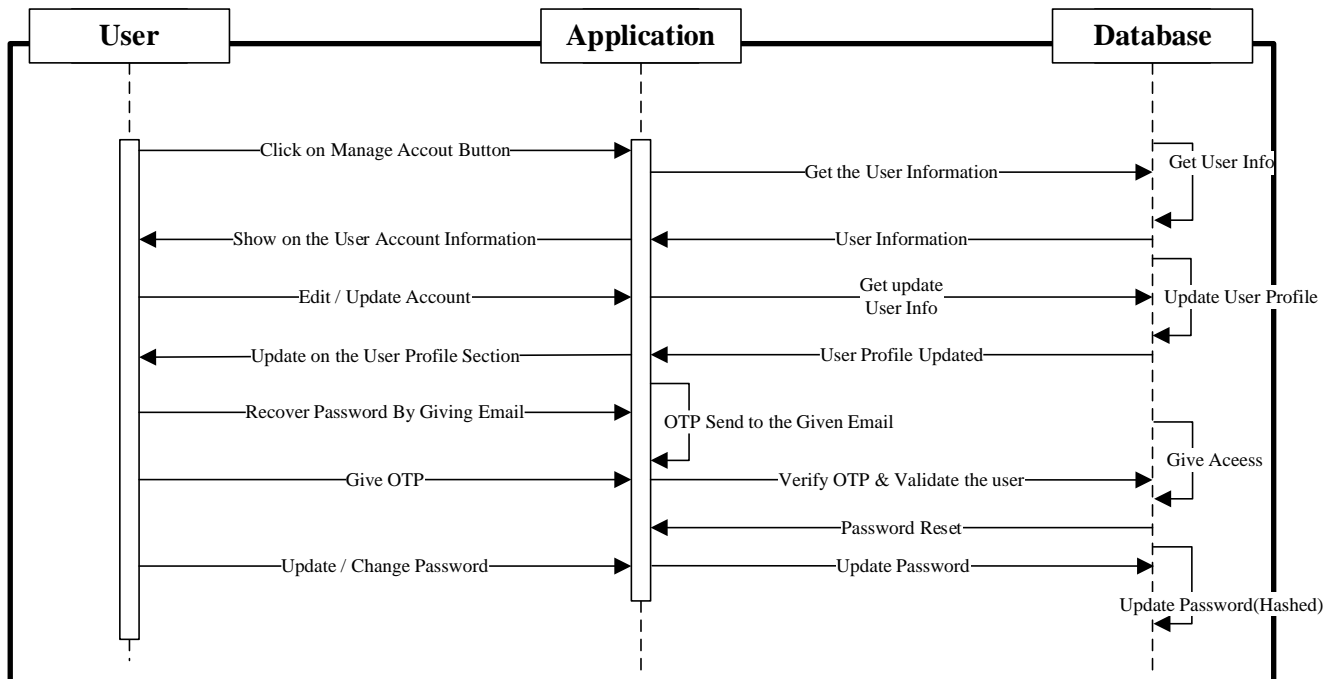


5. SD_Blog



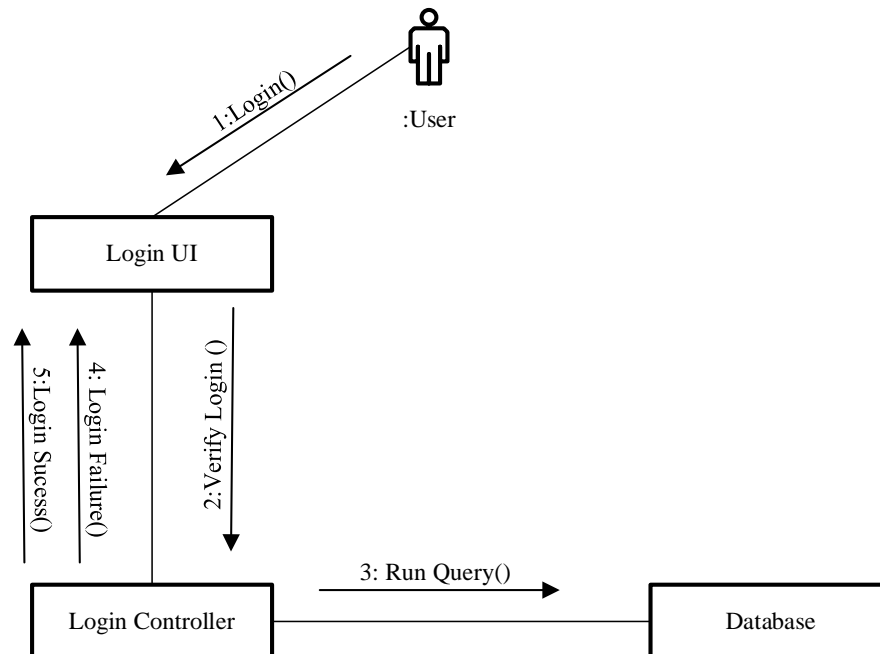


6. SD_Recover_Password &_amp;_Update_Profile



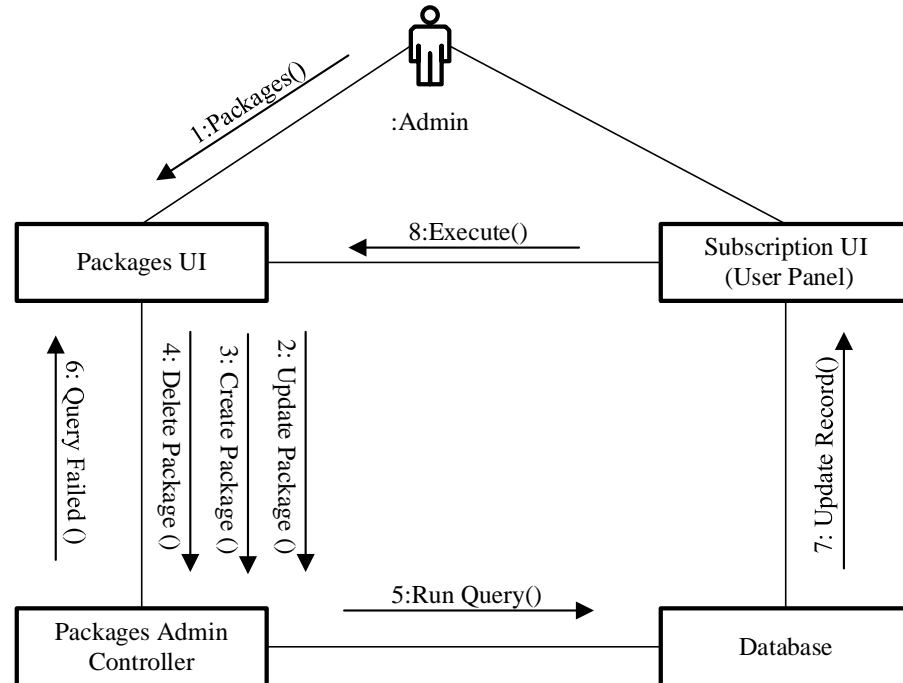
1.5 Collaboration Diagram

1. CD_Login

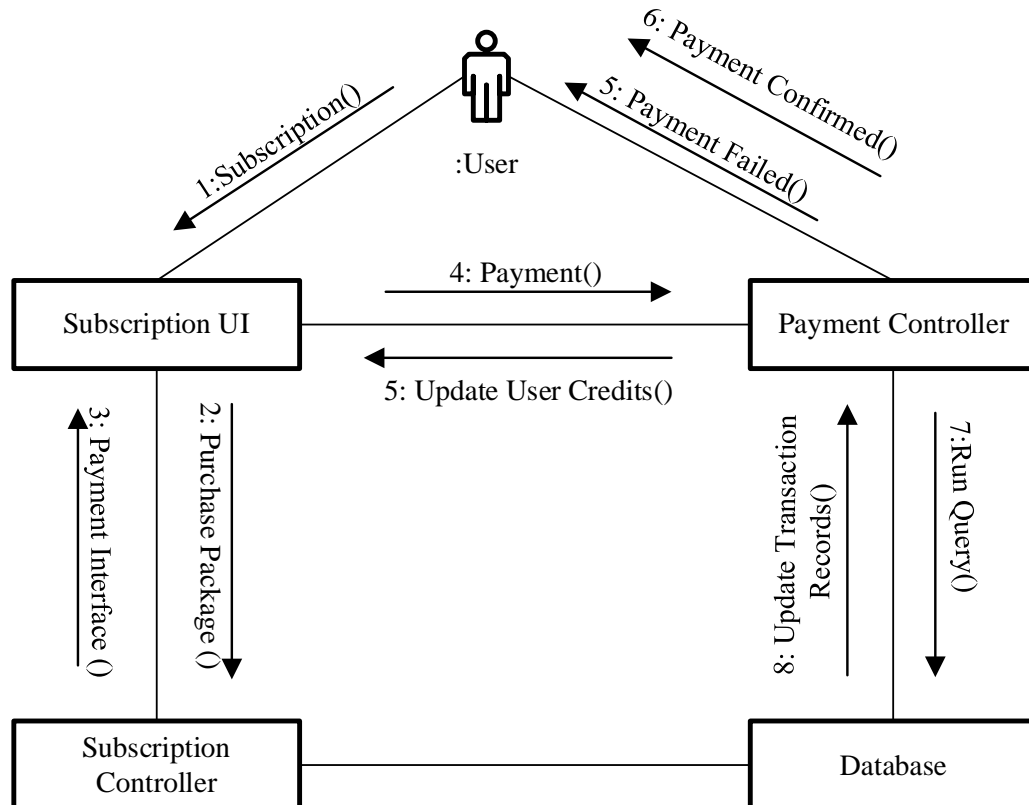




2. CD_Maintain_Packages

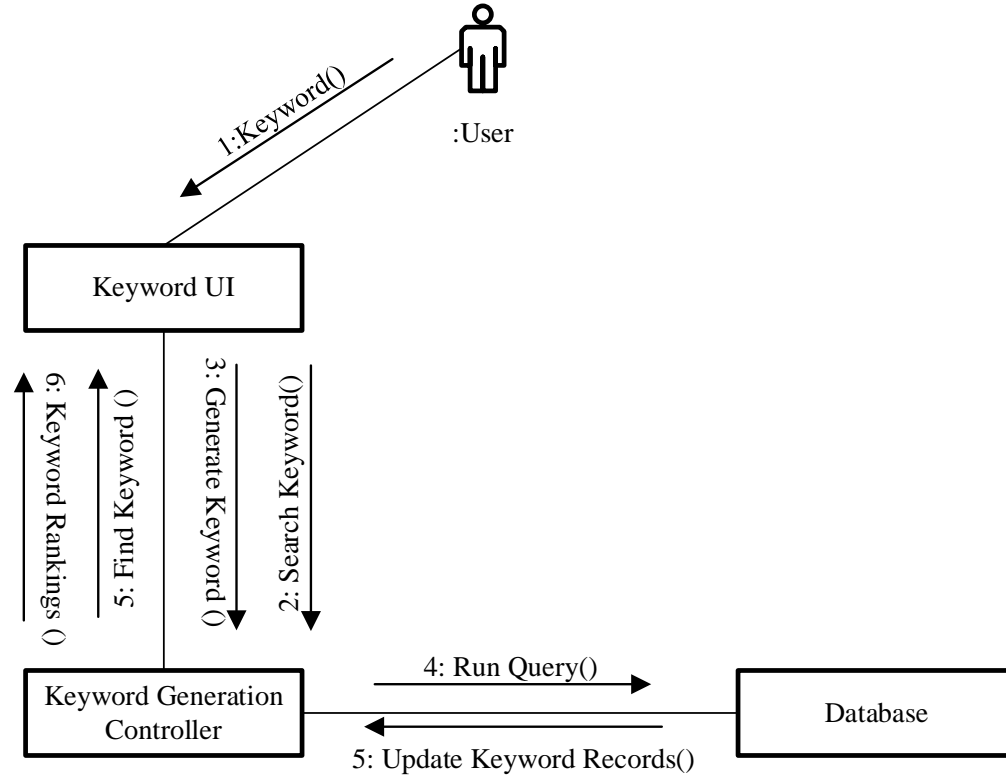


3. CD_Payment

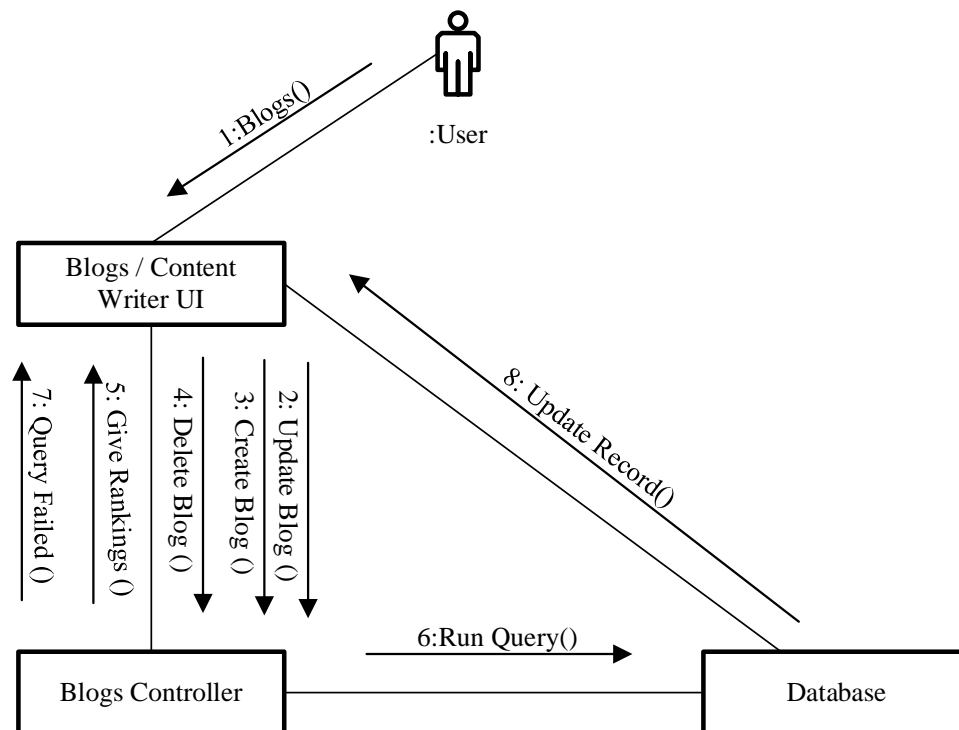




4. CD_Generate_Keywords

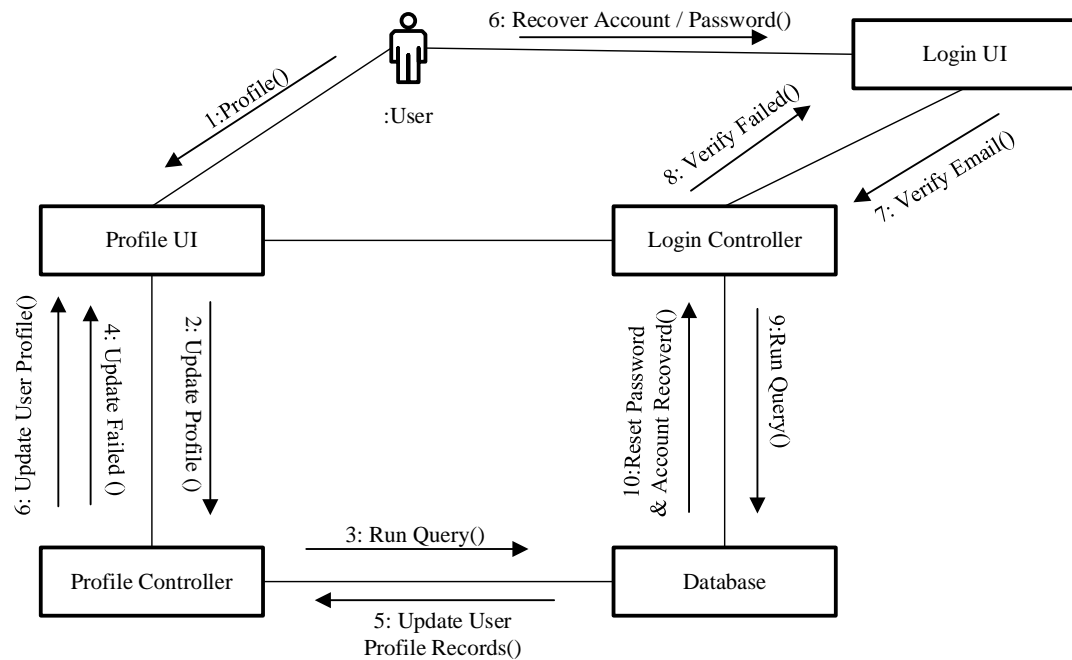


5. CD_Blog





6. CD_Recover_Password &_amp; Update_Profile



1.6 Operation Contracts

1. OP_Login

- **Operation:** User Login
- **Responsibilities:** Enable user to register itself & login to access the system.
- **Cross References:** Use Case: UC_Login
- **Expectations:** User clicks on login without entering username or other required fields.
- **Preconditions:** Server should be online and user can easily access it.
- **Postconditions:** The user has to access to the system successfully.

2. OP_Maintain_Packages

- **Operation:** Maintain Packages
- **Responsibilities:** Admin can perform all CRUD operations on packages.
- **Cross References:** Use Case: UC_Maintain_Packages
- **Exceptions:** Admin click on save button without entering required fields.
- **Preconditions:** Admin has to logged in to perform CRUD on packages.
- **Postconditions:** The packages have managed successfully.



3. OP_Payment

- **Operation:** Online Payment
- **Responsibilities:** User can purchase subscription through online payment.
- **Cross References:** Use Case: UC_Payment
- **Exceptions:** User is not registered or user is not verified or payment is not verified.
- **Preconditions:** User has to logged in to purchase subscription through online payment.
- **Postconditions:** The payment has done successfully.

4. OP_Generate_Keywords

- **Operation:** Generate Keywords
- **Responsibilities:** User can generate optimized keywords for their content and blogs.
- **Cross References:** Use Case: UC_Generate_Keywords
- **Exceptions:** User click on generate button without giving some initial word.
- **Preconditions:** User has to logged in & have credits to generate keywords.
- **Postconditions:** The keywords are generated successfully.

5. OP_Blog

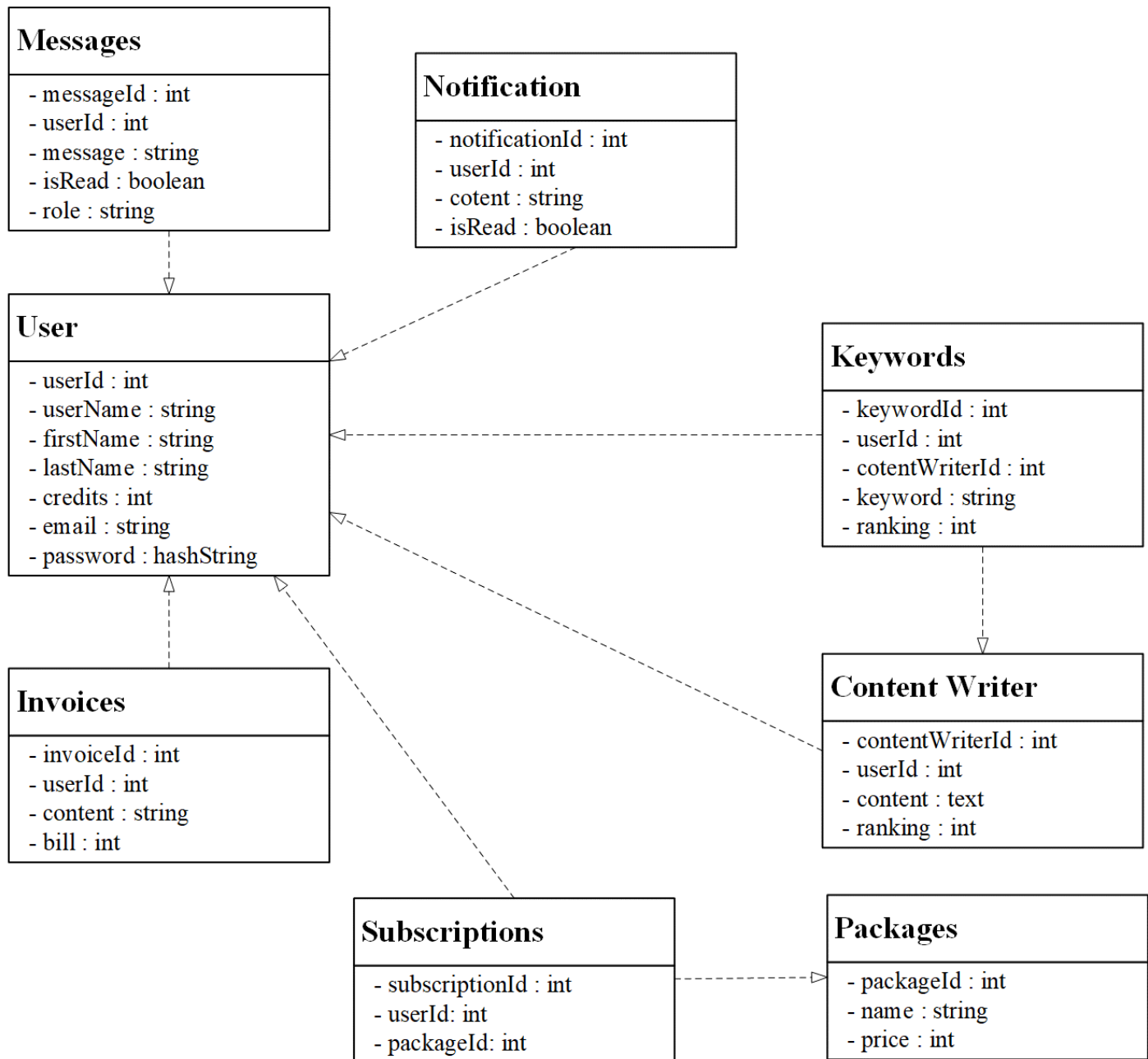
- **Operation:** Write Blog
- **Responsibilities:** User first have some AI generated keywords for their content and write blogs and gets their ranking.
- **Cross References:** Use Case: UC_Blogs
- **Exceptions:** User cannot write blogs.
- **Preconditions:** User has to logged in and some competitor analysis on keywords.
- **Postconditions:** The blogs are ranked successfully.

6. OP_Recover_Password &_amp; Update_Profile

- **Operation:** Recover Password & Update Profile.
- **Responsibilities:** User can update its profile and change password.
- **Cross References:** Use Case: UC_Recover_password &_amp; Update_profile.
- **Exceptions:** User can't login to the system or user is not verified.
- **Preconditions:** User has to logged in to update profile.
- **Postconditions:** User profile updated successfully.



1.7 Design Class Diagram





1.8 Data Model

