JASPER SEO

Project ID: BSIT-2024

Session: 2020-2024

University of The Punjab, Gujranwala Campus

STATEMENT OF SUBMISSION

This is to certify that following students have successfully completed the final project named as: **JASPER SEO** at The University of the Punjab, Gujranwala Campus, to fulfill the partial requirement of the degree of **Bachelor of Science in Information Technology.**

Sr. No.	Student's Roll No.	Student's Name
1	BIT20214	AZEEM ALI
2	BIT20231	DAOUD UR REHMAN
3	BIT20253	TALHA YOUNAS

Internal Supervisor
Dr. Ghulam Mustaf
PUGC.

Project Coordinator
Mr. Muhammad Younas
PUGC.

Incharge IT-Department
Dr. Naveed Jhammat
PUGC.

Director General
Prof. Dr. Muhammad Mudassar Ghaffor
PUGC.

PROOF READING CERTIFICATE

It is to certify that I have read the document thoroughly and circumspectly. I am convinced that the resultant project does not contain any spelling, punctuation, or grammatical mistakes as such. All in all, I find this document well organized and I am in no doubt that its objectives have been successfully met.

Ms. Adila Hussain
PUGC

Acknowledgments

This project is the direct result of the teamwork. We sincerely thank the instructors and students who have shared their suggestion with us over last few weeks. Many thanks go to **Dr. Ghulam Mustafa** our Project coordinator and supervisor, whose management helped us to get this project produced in timely and efficient manner. And finally, we thank **Prof. Dr. Salman Naseer and Mr. Muhammad Younas** who weren’t directly involved in this project but who have stood by us the whole way, we appreciate your encouragement and support. We are also thankful to our friends and families whose silent support led us to complete our project.

Write to us

We welcome your response to this project. If there is anything you want to mention about the improvement of this project, please let us know:

Azeem Ali	azeemali0346634@gmail.com
Daoud Ur Rehman	daoudurrehman1@gmail.com
Talha Younas	talhamughal52352@gmail.com

ABSTRACT

The objective of this project is to develop a web-based Blog Ranking SEO Tool using artificial intelligence. This tool aims to streamline the process of optimizing blog content for search engine visibility and effectiveness. Users will input specific keywords, and the tool will utilize web scraping techniques to retrieve relevant data from the internet. The retrieved data will then be processed and analyzed using AI algorithms to provide comprehensive ranking results. This tool is designed to assist content creators and digital marketers in making data-driven decisions to enhance their blog's search engine performance.

<u>For Our Beloved Parents and Teachers</u>

"Dedicated to our parents and teachers without whose wholehearted support, encouragement, and guidance it would have been impossible for us to make this project"

Table of Contents

Chapter # 1
Project Feasibility Report

1.1 Project Feasibility Report

The project seeks to address the need for a more efficient and data-driven approach to optimizing blog content for search engines. Content creators often face challenges in achieving optimal search engine visibility, and this tool aims to fill that gap by utilizing web scraping techniques and AI algorithms.

Key components typically included in a Project Feasibility Report are:

- Technical
- Operational
- Economic
- Schedule
- Specification
- Information
- Motivational
- Legal & Ethical

1.1.1 Technical Feasibility

The project requires web development skills, web scraping tools, and AI algorithms. The technical infrastructure involves cloud-based hosting for scalability, ensuring the tool can handle varying workloads.

1.1.2 Operational Feasibility

The project relies on the availability of skilled developers and aims for seamless integration with user's workflows. This approach ensures the tool is user-friendly and easily adoptable within existing content creation processes.

1.1.3 Economic Feasibility

The economic feasibility of this project is a critical aspect that evaluates its financial viability, cost-effectiveness, and potential return on investment. Economic feasibility has two parts, cost estimates and benefit estimates. In cost evaluation analysis, we have chosen most popular methodology i.e. Functional Point Analysis (FPA).

1.1.4 Schedule Feasibility

The schedule feasibility of the web-based Blog Ranking SEO Tool project is established through a well-structured timeline encompassing planning, development, testing, deployment, and ongoing maintenance phases. Key milestones, such as the completion of development and successful deployment, are identified with specific dates to track progress. Task dependencies are managed to prevent delays, and resources, including skilled developers and cloud-based infrastructure, are allocated efficiently.

1.1.5 Specification Feasibility

The specification feasibility involves evaluating the effectiveness of web scraping techniques for retrieving relevant data from the internet. The chosen methods should be efficient, accurate, and compliant with legal and ethical standards. AI algorithms for data processing must be capable of providing comprehensive ranking results.

1.1.6 Information Feasibility

This project's success depends on effective data accessibility. Web scraping will be employed to secure relevant information on search engine algorithms, keyword trends, and blog performance metrics, ensuring compliance with legal and ethical standards. The tool's AI algorithms will require diverse and representative training data for accuracy. User-friendly specifications will allow content creators to input keywords aligned with their SEO goals. Information security and privacy, real-time data retrieval, and potential API integration contribute to the tool's effectiveness. The information feasibility analysis confirms the project's capability to access and leverage relevant data efficiently.

1.1.7 Motivational Feasibility

The success of this project hinges on the collective motivation of the project team and stakeholders. Skilled developers, AI specialists, and web scraping experts must be inspired by the opportunity to innovate in the SEO tools market. Stakeholders, including investors and end-users, need a shared vision for the tool's potential to streamline blog optimization. Continuous communication and transparent goal-setting will sustain the motivation necessary for the project's success.

1.1.8 Legal & Ethical Feasibility

Compliance with data protection regulations, transparent data usage, and robust security measures are integral to this tool's legal and ethical feasibility. Prioritizing unbiased AI algorithms and user confidentiality builds trust and ensures a positive reputation. Continuous monitoring of legal and ethical standards will be crucial for adapting the tool to evolving requirements, maintaining its feasibility throughout its lifecycle.

1.2 Project Scope

The project involves developing a Web-Based Blog Ranking SEO Tool utilizing AI for targeted keyword analysis. Key features include user registration, admin management, and the ability to send reports or invoices for paid services. Users can purchase packages using various payment methods, and the tool prioritizes data accuracy, quality, and interactive mapping for an optimized user experience. The scope encompasses a secure, scalable, and user-friendly platform, catering to content creators and digital marketers seeking to enhance their blog content.

Chapter # 2
Project Costing

2. Project Costing

2.1 Project Cost Estimation by Function Point Analysis

Internal Logic File (ILF's)			
Type of Components	*DET*	*RET*	*Complexity*
Admins	4	1	Low
Rankings	9	2	Low
Keywords	12	3	Low
Blogs	8	2	Low
Users	5	1	Low

External Inputs (EI's)			
Type of Components	*DET*	*FTR*	*Complexity*
User Login	2	1	Low
Sign Up	9	1	Low
Subscription	4	1	Low
Search Keyword	2	1	Low
Write Blogs	6	3	High
User Profile Settings	5	1	Low

External Interface File (EIF's)			
Type of Components	*DET*	*FTR*	*Complexity*
Online Payment	3	1	Low

External Output (EO's)			
Type of Components	*DET*	*FTR*	*Complexity*
Keywords	4	7	Average
Blogs	12	5	High
Subscription Details	5	3	Low

External Queries (EQ's)			
Types of Components	*DET*	*FTR*	*Complexity*
Blogs	5	2	Low
Payment Method	3	1	Low
Invoices	9	1	Low
Reports	2	1	Low
Keyword Search	15	3	High

Total Unadjusted Function Points

Types of Components	*Complexity of Components*			
	Low	*Average*	*High*	*Total*
Internal Logical File	5 * 7 = 35	0 * 10 = 0	0 * 15 = 0	35
External Interface File	1 * 5 = 5	0 * 7 = 0	0 * 10 = 0	5
External Input	5 * 3 = 15	0 * 4 = 0	1 * 6 = 6	21
External Output	1 * 4 = 4	1 * 5 = 5	1 * 7 = 7	16
External Inquiries	4 * 3 = 12	0 * 4 = 0	1 * 6 = 6	18
Total Un Adjusted Function Points				95

Calculation of Value Adjustment Factor

Sr.#	*Questions*	*Scale*
1	The extent to which our data will communicate with another application?	5
2	The extent to which our data processing can be distributed	1
3	Performance of the system to fulfill client requirements	4
4	Configuration extent?	3
5	Transactional rate?	3
6	Extent to which Online data entry and control information functions are provided?	4
7	How much efficiency of the user will be increased?	4
8	The extent to which online update for the ILF is possible?	3
9	Complexity of the Task?	4
10	The extent to which application and its code can be reused in another application?	5
11	Installation eases of the Task?	4
12	Operational ease?	4
13	The extent to which application can be deployed at multiple sites?	2
14	Extent to which design of the Task is flexible?	4
Total degree of influence		50

VAF = 0.65+(0.01*TDI)
= 0.65+(0.01*50)
= 0.65+(0.5)
VAF = 1.15

Calculation of Adjusted Function Points

Adjusted Function Points = Unadjusted Function Points*VAF
= 95 * 1.15
AFP = 109.25 Function Points

Total Duration of Project:

Average Productivity = 7FP/Person-Month

Effort Month = Adjusted Function Points/Productivity
= 109.25 /7
= 15.6 Person-Month

Duration Of Project = Effort Month/No of Persons
= 16/5
= 4 Months (Approximately)

Calculation OF Total Cost:

Labor Rate = 30,000 Rs
Cost Per FP = 30,000/7
= 4,285 Rs

Project Cost = Adjusted Function Points/Cost Per FP
= 109.25*4285
= 468,136 Rs

Chapter # 3
Project Planning

3.1 CPM – Critical Path Method

Creating a Critical Path Method (CPM) for a project involves identifying the sequence of activities that must be completed on time to ensure the project's successful and timely completion. Based on the goals and objectives you provided, here's a simplified representation of the Critical Path for this project:

1. Specify the individual activities.
2. Determine the sequence of those activities.
3. Draw a network diagram.
4. Estimate the completion time for each activity.
5. Identify the critical path (longest path through the network)
6. Update the CPM diagram as the project progresses.

3.1.1 Specify the Individual Activities

Following are the individual activities involved in the project:

- Start
- Planning
- Requirement Gathering
- Specification Analysis
- Design
- Development
- Integration and Testing
- Maintenance

3.1.2 Determine the Sequence of the Activities

There are many activities that are dependent on the completion of some other activities, the dependencies of activities upon each other are below: Start → None

(A) Proposal → None

(B) Planning → Proposal

(C) Requirement Gathering → Planning

(D) Specification Analysis → Planning, Requirement Gathering

(E) Design → Specification Analysis

(F) Development → Specification Analysis, Design

(G) Integration and Testing → Design, Development

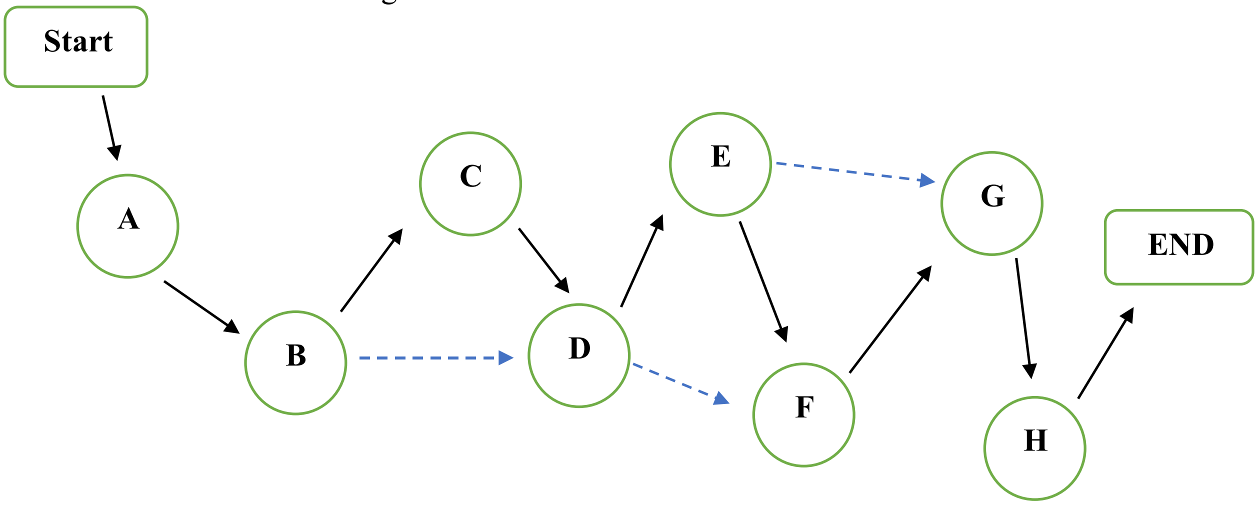
(H) Maintenance → Integration & Testing End → Maintenance

Task Dependent Table

Task ID	Task Description	Duration(days)	Dependencies
	Start	-	
A	Proposal	7	-
B	Planning	6	A
C	Requirement Gathering	14	B
D	Specification Analysis	13	B, C
E	Design	30	D
F	Development	25	D, E
G	Integration & Testing	14	F, E
H	Installation & Handover / Maintenance	20	G
	End	-	

3.1.3 Network Diagram

Network Diagram of the Activities is as under:



3.1.4 Estimate Activity Completion Time

The time required to complete each activity can be estimated using experience or the estimates of knowledgeable persons.

Task ID	Predecessors	Duration (Days)
A	-	7
B	A	6
C	B	14
D	B, C	13
E	D	30
F	D, E	25
G	F, E	14
H	G	20

3.1.5 Identify the Critical Path

Activity	Duration	ES	EF	LS	LF	TS	FS
A	7	0	7	0	7	0	0
B	6	7	13	7	13	0	0
C	14	13	27	13	27	0	0
D	13	27	40	27	40	0	0
E	30	40	70	40	70	0	0
F	25	70	95	70	95	0	0
G	14	95	109	95	109	0	0
H	20	109	129	109	129	0	0

Critical Path is: -

Start → A→ B → C→D→E→F→G→H→END

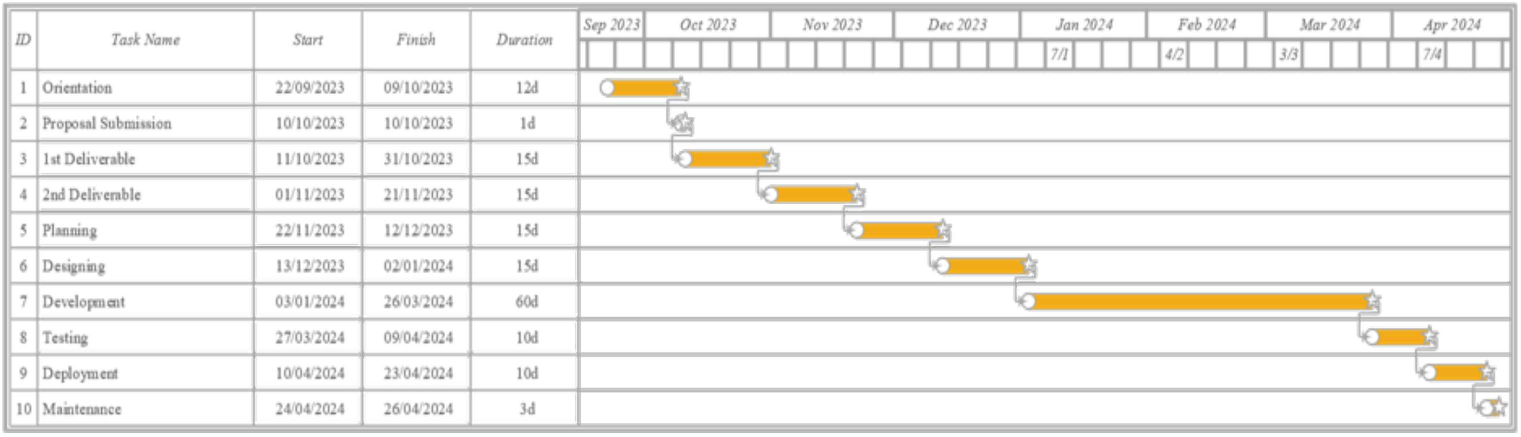
Duration of Paths: -

7+6+14+13+30+25+14+20 = 129

Total Duration of the Project: -

Duration of Critical Path (Days) = 129

3.2 Gantt Chart



ID	Task Name	Start	Finish	Duration
1	Orientation	22/09/2023	09/10/2023	12d
2	Proposal Submission	10/10/2023	10/10/2023	1d
3	1st Deliverable	11/10/2023	31/10/2023	15d
4	2nd Deliverable	01/11/2023	21/11/2023	15d
5	Planning	22/11/2023	12/12/2023	15d
6	Designing	13/12/2023	02/01/2024	15d
7	Development	03/01/2024	26/03/2024	60d
8	Testing	27/03/2024	09/04/2024	10d
9	Deployment	10/04/2024	23/04/2024	10d
10	Maintenance	24/04/2024	26/04/2024	3d

3.3 Introduction to Team member and their skill set

Following are our team Members:

- **Azeem Ali**

He is tasked with creating comprehensive project documentation, ensuring that all aspects of our development process are well-documented for clarity and future reference. He plays a pivotal role in designing the database architecture, ensuring it aligns seamlessly with the project's requirements and objectives.

- **Daoud Ur Rehman**

He is tasked with crafting the frontend of our project, ensuring a seamless and interactive user interface. He plays a pivotal role in designing the overall look and feel of the tool, emphasizing both functionality and aesthetic appeal. He is also responsible for the visual design of the tool, employing Figma to bring forth a harmonious blend of functionality.

- **Talha Younas**

As the project leader, He provides overarching guidance, ensuring cohesion and alignment with project goals. He spearheads backend development, focusing on creating a reliable and scalable infrastructure for our application. He also leads the integration of AI techniques, contributing to the generation of targeted keywords and addressing various management aspects. He is also responsible for integrating APIs, enabling the tool to interact with external services seamlessly.

3.4 Tools and Technology with reasoning

3.4.1 Tools

- **WebStorm**

  WebStorm developed by JetBrains is a powerful IDE tailored for web development. It offers advanced JavaScript support, intelligent code completion and seamless integration with version control systems like Git. With features for debugging, syntax highlighting and code refactoring. It provides a comprehensive toolkit for efficient coding on Windows, Linux, and macOS platforms.

- **PyCharm**

  PyCharm is an integrated development environment (IDE) created by JetBrains specifically designed for Python development. It offers a wide range of features including intelligent code completion, syntax highlighting, debugging support, version control integration and code refactoring. With its user-friendly interface and powerful tools, PyCharm streamlines the development process that making it a popular choice for Python programmers on Windows, Linux, and macOS platforms.

- **GIT and GitHub**

  Git is a widely used version control system that allows developers to track changes in their codebase. It provides features like branching, merging, and history tracking, enabling collaborative development and efficient management of code.

  GitHub is a web-based platform that utilizes Git for version control. It serves as a hosting service for Git repositories, allowing developers to store and manage their code online. GitHub provides additional features such as issue tracking, pull requests, and project management tools, making it a popular platform for open-source collaboration and software development projects.

- **Figma**

  Figma is a cloud-based design and prototyping tool used for creating user interfaces and interactive design prototypes. It offers a collaborative platform where multiple team members can work simultaneously on design projects that enables real-time feedback and seamless coordination. With features like vector editing, prototyping and design libraries, Figma streamlines the design process and facilitates efficient communication between designers and stakeholders. Its cloudbased nature allows for easy access and sharing of design files across different devices and platforms that making it a versatile tool for UI/UX designers.

- **MongoDB Atlas**

  MongoDB Atlas is a fully-managed cloud database that handles all the complexity of deploying, managing, and healing your deployments on the cloud service provider of your choice (AWS , Azure, and GCP). MongoDB Atlas is the best way to deploy, run, and scale MongoDB in the cloud. It is a valuable tool for developers and database administrators that offering a graphical interface to interact with NOSQL databases and perform tasks efficiently.

- **Microsoft Visio**

  Microsoft Visio is a versatile diagramming tool by Microsoft used for creating flowcharts, organizational charts and various visual representations. With a user-friendly interface and pre-defined templates. It simplifies the process of visualizing complex information for professionals across different industries.

- **Colab**

  Google Colab is a cloud-based Python development environment by Google. It offers free access to powerful computing resources that making it ideal for tasks like machine learning and data analysis. With its collaborative Jupyter notebook interface, it's a convenient choice for coding projects without the need for extensive local resources.

3.4.2 Technology

- **Next.js**

  Next.js is a powerful React framework that facilitates the building of server-rendered React applications with ease. It offers features like static site generation and server-side rendering, allowing for efficient SEO optimization and enhanced performance. With its intuitive file-based routing system and support for JavaScript. Next.js provides a robust foundation for developing modern highperformance web applications. It is a popular choice among developers for its seamless integration with React and efficient rendering capabilities that making it well-suited for projects requiring dynamic, SEO-friendly web pages.

- **Express.js**

  Express.js is a powerful and flexible web application framework for Node.js. It provides a robust set of features for building web applications. With its minimalist and flexible design, Express.js simplifies the process of creating APIs and handling routes. It also supports various middleware that making it easy to integrate additional functionality. Express.js is a popular choice for backend development due to its scalability and ease of use which making it well-suited for projects requiring efficient and customizable server-side solutions.

- **MongoDB**

  MongoDB is a source-available, cross-platform, document-oriented database program. Classified as a NoSQL database product, MongoDB utilizes JSON-like documents with optional schemas. MongoDB is developed by MongoDB Inc. and current versions are licensed under the Server-Side Public License. It offers scalability and high performance that making it a popular choice for a wide range of projects. MongoDB is known for its stability, security and ease of integration with various programming languages and frameworks which makes it a versatile and trusted database solution for many developers.

- **Python (Sentence-Transformers Model)**

  Sentence-Transformers Model is a state-of-the-art natural language processing (NLP) tool that specializes in transforming sentences and text paragraphs into numerical representations. It employs advanced techniques like Siamese and triplet network architectures along with transfer learning from largescale datasets. This enables it to generate highly meaningful and contextually rich embeddings for text. In our project, the Sentence-Transformers model plays a crucial role in tasks like semantic similarity assessment, text clustering and information retrieval. Its capabilities make it an invaluable tool for advanced NLP applications that enhancing the project's ability to extract and process information from text data efficiently and accurately.

3.5 Vision Document

Project Mission

Achieve a high level of precision in SEO recommendations, ensuring relevance to user-input keywords. Retrieve data from internet with optimal speed for efficient results. Adhere to industry-standard security practices and complete the project milestones on schedule.

Users

- Digital Marketers and SEO Professionals
- Content Creators and Bloggers
- Marketing Students and Enthusiasts
- Freelance Writers

Key Factor to judge the Quality

- Users can register to avail services.
- Implement modern technology.
- Use latest version of Node JS and latest versions of built-in libraries.
- Capable to integrate further module as per users requirement.

Financial

- Development Cost is Minimum.

3.6 Risk List

- Technology may change during the project
- Lack of resources
- Any member may leave the group
- Project time may increase than of given time
- Unexpected project scope expansions
- Ignorance of non-functional requirements
- Lack of cooperation of team members
- Inadequate estimation of required resources
- Integration copyrights
- Our system may get slow in case of heavy traffic

Chapter # 4
Requirements Engineering

4.1 Summary of Requirements

This tool will facilitate users in optimizing their blog content for search engine visibility and effectiveness. The initial requirements encompass a user-friendly interface allowing users to input specific keywords relevant to their blog content securely. To gather pertinent data, the tool will employ web scraping techniques to retrieve information from the internet. Advanced AI algorithms will process and analyze this data, considering metrics such as keyword relevance, search volume, competition, and trend analysis. The tool will generate comprehensive SEO insights and provide recommendations for enhancing content optimization. Real-time updates on ranking results, automation features, and a secure user authentication system will be implemented to streamline operations and ensure data confidentiality. The web-based interface will be designed for easy navigation and accessibility. The system will also include a user support system for user assistance, feedback gathering, and thorough documentation, encompassing user guides, technical documentation, and FAQs. Scalability will be a key consideration, accommodating potential increases in the user base and data. Thorough testing, including functional, usability, and security testing, will be conducted to ensure the reliability of the tool. Additionally, functionalities such as content creation, support for black-hat SEO practices, and in-depth website analytics will be explicitly outlined as out of scope. Future considerations will include placeholders for potential enhancements based on user feedback and emerging industry trends.

4.2 Identifying External Entities

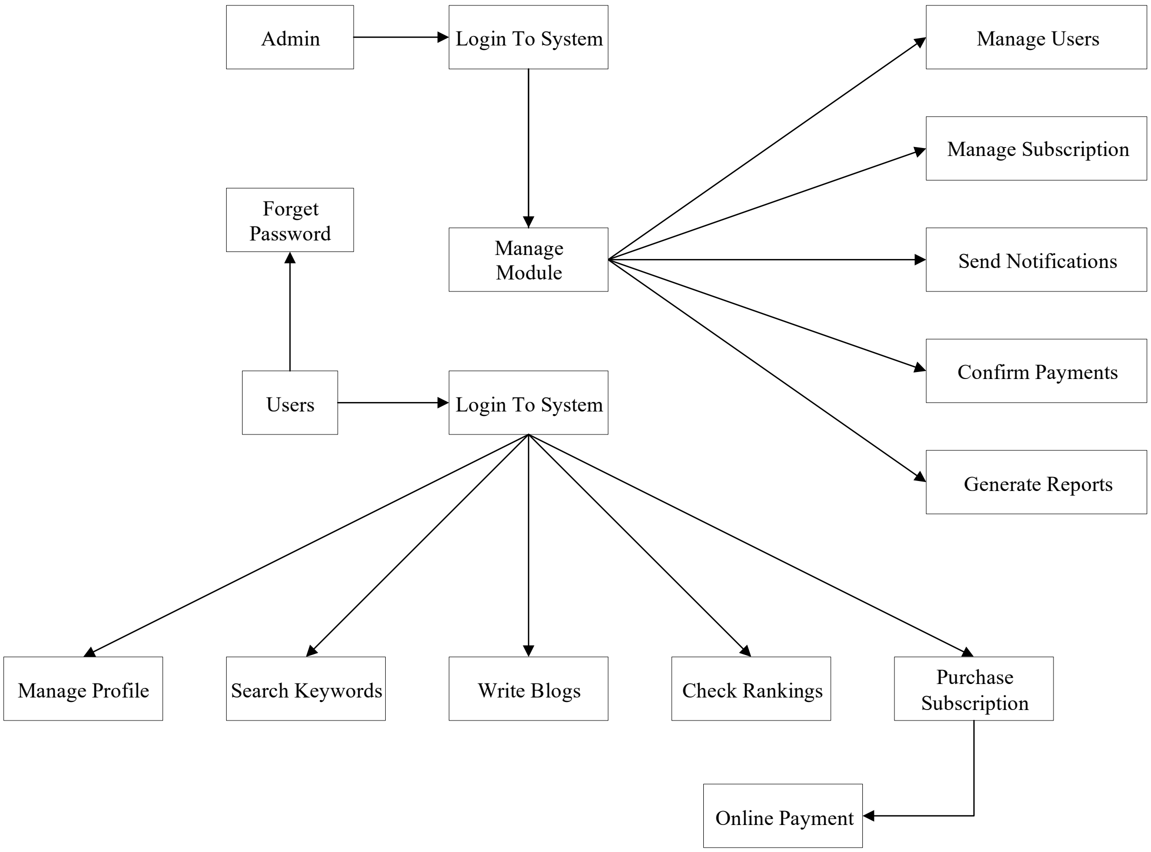
i. Over Specify Entities from Abstract

- Content Creators
- Digital Marketers
- Search Engines
- Web Data Sources
- AI and Data Science Services
- Support Services

ii. Perform Refinement

- Security Services
- Regulatory Bodies
- Competitor Tools
- API Providers
- Web Hosting Service
- Feedback Platforms

4.3 Context Level Data Flow Diagram



4.4 Capture "Shall" Statement

Para#	*External Entity*	*Initial Requirements*
1.5	Admin	Admin "shall" login to the system.
1.5	Admin	Admin "shall" be able to Delete & block the users.
1.5	Admin	Admin "shall" manage prices for subscription.
1.5	Admin	Admin "shall" be able to confirm payment for subscription and assign subscription featues to the users.
1.5	Admin	Admin "shall" be able to send notification to the user for expiration of subscription and send invoices to the user.
1.5	Admin	Admin "shall" be able to check searched Keyword.
1.5	Admin	Admin "shall" be able to see all generated keyword of specific searched keyword
1.5	Admin	Admin "shall" be able to generate reports.
1.5	Admin	Admin "shall" logout from the system.
1.5	User	User "shall" login to his/her account.
1.5	User	User "shall" view its keyword.
1.5	User	User "shall" make payment through online.
1.5	User	User "shall" view its blogs ranking.
1.5	User	User "shall" make blogs.
1.5	User	User "shall" view his subscription details.
1.5	User	User "shall" able to search keywords.
1.5	User	User "shall" able to recover his password.
1.5	User	User "shall" able to change his account settings.
1.5	User	User "shall" be able to logout from system.

4.5 Allocate Requirements

Para#	*Initial Requirements*	*Use Case Name*
1.4	Admin “shall” login to the system.	UC Login
1.4	Admin “shall” be able to Add, Delete and Update the subscription Packages.	UC_Maintain_Subscription
1.4	Admin “shall” manage prices of subscription.	UC_ Subscription _Price
1.4	Admin “shall” be able to check subscription.	UC_ Check_Subscription
1.4	Admin “shall” be able to see all generated keywords of specific keyword.	UC_Specific_Keywords
1.4	Admin “shall” logout from the system.	UC_Logout_Admin
1.4	User “shall” login to his/her account.	UC_Login_User
1.4	User “shall” view keyword.	UC_View_Keyword
1.4	User “shall” make payment through online.	UC_Payment
1.4	User “shall” view keyword detail.	UC_ Keyword _Detail
1.4	User “shall” make blogs.	UC_Blogs
1.4	User “shall” view his blogs details.	UC_View_ Blogs
1.4	User “shall” able to search keyword.	UC_Search_Keyword
1.4	User “shall” able to recover his password.	UC_Recover_Password
1.4	User “shall” able to change his account settings.	UC_Account_Settings
1.4	User “shall” be able to logout from system.	UC_Logout_User

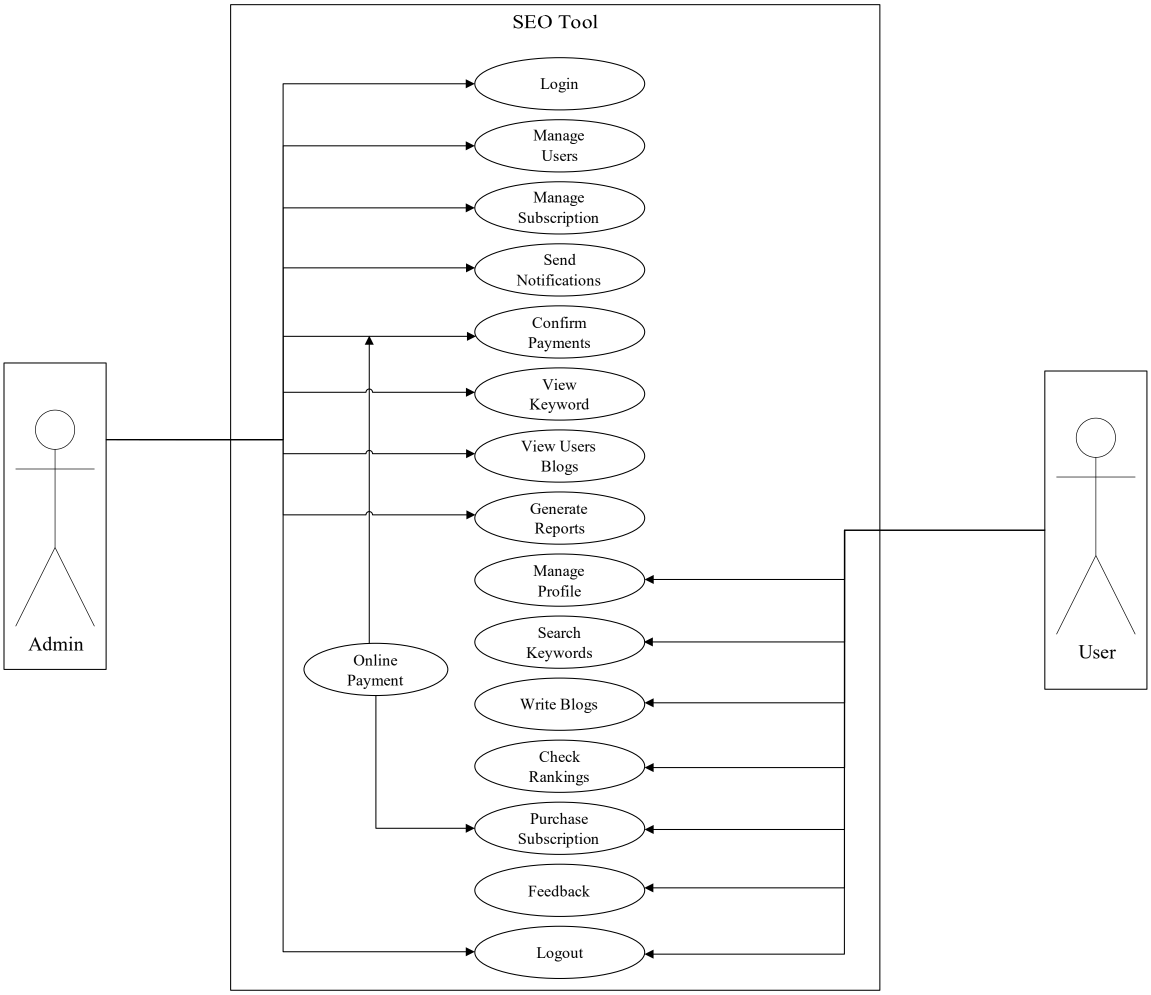
4.6 Prioritize Requirements

Para#	*Initial Requirements*	*Rank*	*Use Case ID*	*Use Case Name*
1.4	Admin “shall” login to the system.	Highest	UC_1	UC_Login
1.4	Admin “shall” be able to Add, Delete and Update the subscription Packages.	Highest	UC_2	UC_Maintain_Subscription
1.4	Admin “shall” manage prices of subscription.	Highest	UC_3	UC_ Subscription _Price
1.4	Admin “shall” be able to check subscription.	Medium	UC_4	UC_ Check_Subscription
1.4	Admin “shall” be able to see all generated keywords of specific keyword.	Medium	UC_5	UC_Specific_Keywords
1.4	Admin “shall” logout from the system.	Highest	UC_6	UC_Logout_Admin
1.4	User “shall” login to his/her account.	Highest	UC_7	UC_Login_User
1.4	User “shall” view keywords.	Medium	UC_8	UC_View_Keyword
1.4	User “shall” make payment through online.	Highest	UC_9	UC_Payment
1.4	User “shall” view keyword detail.	Medium	UC_10	UC_ Keyword _Detail
1.4	User “shall” make blogs.	Highest	UC_11	UC_Blogs
1.4	User “shall” view his blogs details.	Highest	UC_12	UC_View_ Blogs
1.4	User “shall” able to search keyword.	Medium	UC_13	UC_Search_Keyword
1.4	User “shall” able to recover his password.	Highest	UC_14	UC_Recover_Password
1.4	User “shall” able to change his account settings.	Highest	UC_15	UC_Account_Settings
1.4	User “shall” be able to logout from system.	Highest	UC_18	UC_Logout_ User

4.7 Requirement Traceability Matrix

SR#	***Para#***	***System Specification Text***	***Build***	***Use Case Name***	***Category***
1	1.3	Login credentials will be required to login to the system.	B1	UC_Login	Online
2	1.3	Admin shall have the information about the subscription Packages to update Packages.	B1	UC_Maintain_Packages	Online
3	1.3	User “shall” make payment through online.	B1	UC_Payment	Online
4	1.3	Admin “shall” be able to see all generated keywords of specific keyword.	B1	UC_Specific_Keywords	Online
5	1.3	User shall require some information to make blog	B1	UC_Blog	Online
6	1.3	User “shall” able to recover his password.	B1	UC_Recover_Password	Online
7	1.3	User “shall” able to search keywords.	B1	UC_Search_Keywords	Online

4.8 High Level Use Case Diagram



Chapter # 5
Use Case Description with Diagram

5.1 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:

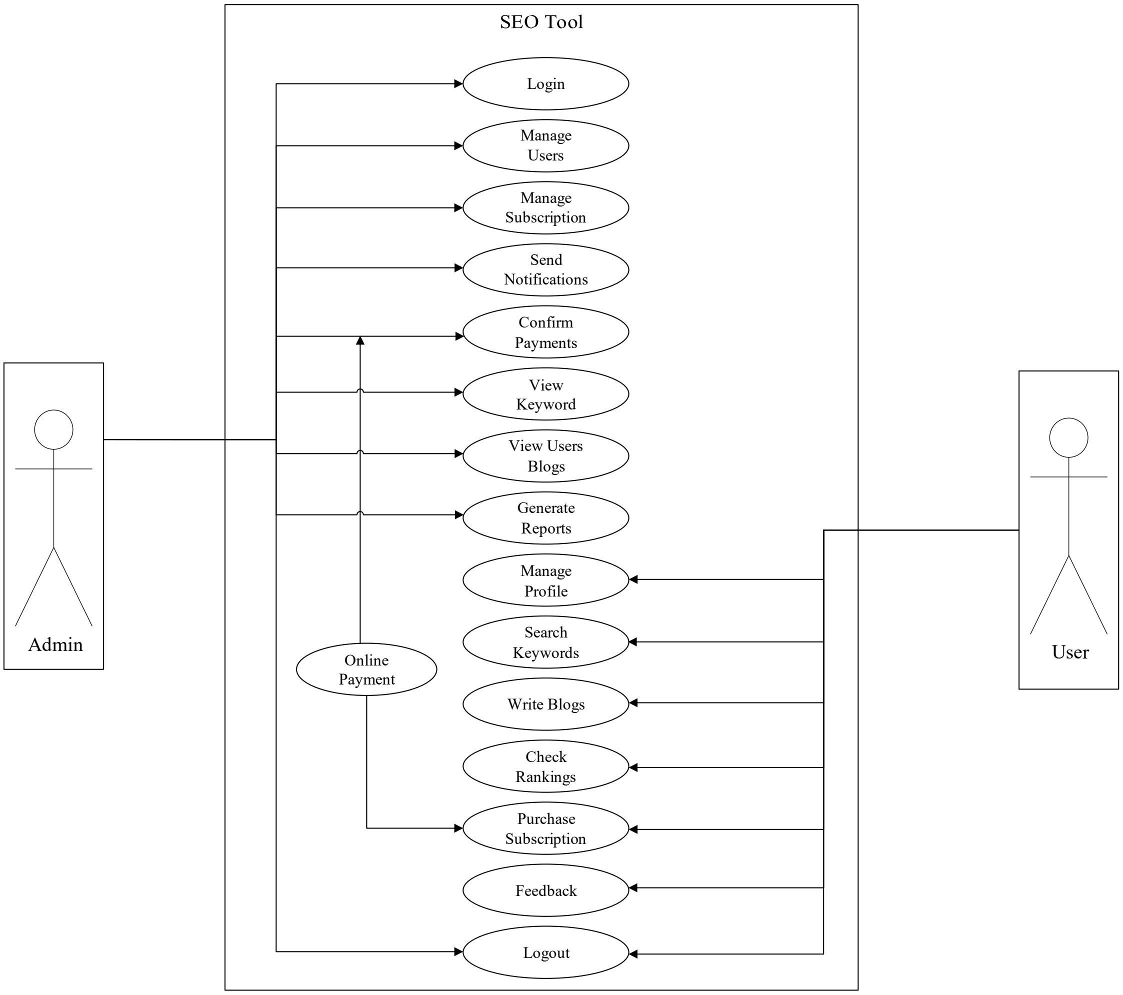
- If the provided email address or username is not found in the system during password recovery, the system displays an error message prompting the user to verify the information and try again.
- If the user attempts to save invalid profile information (e.g., invalid email format) during profile updating,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

5.2 Use case Diagram (refined and updated)



Chapter # 6
Domain Model

6. *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.

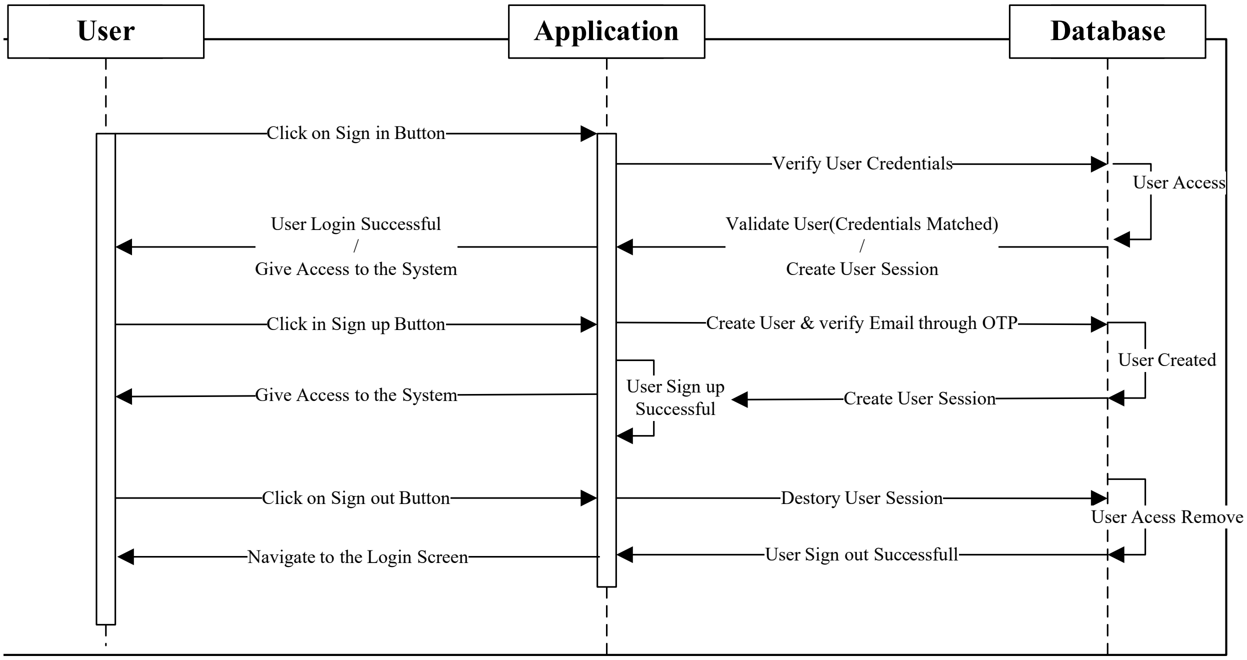
6. **Audit:**
    - Attributes: AuditID, URL, Issues, Recommendations
    - Relationships:
        - Audits identify on-page SEO issues and provide recommendations for optimization.
7. **Link:**
    - Attributes: LinkID, Source URL, Destination URL, Anchor Text
    - Relationships:
        - Links are 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.

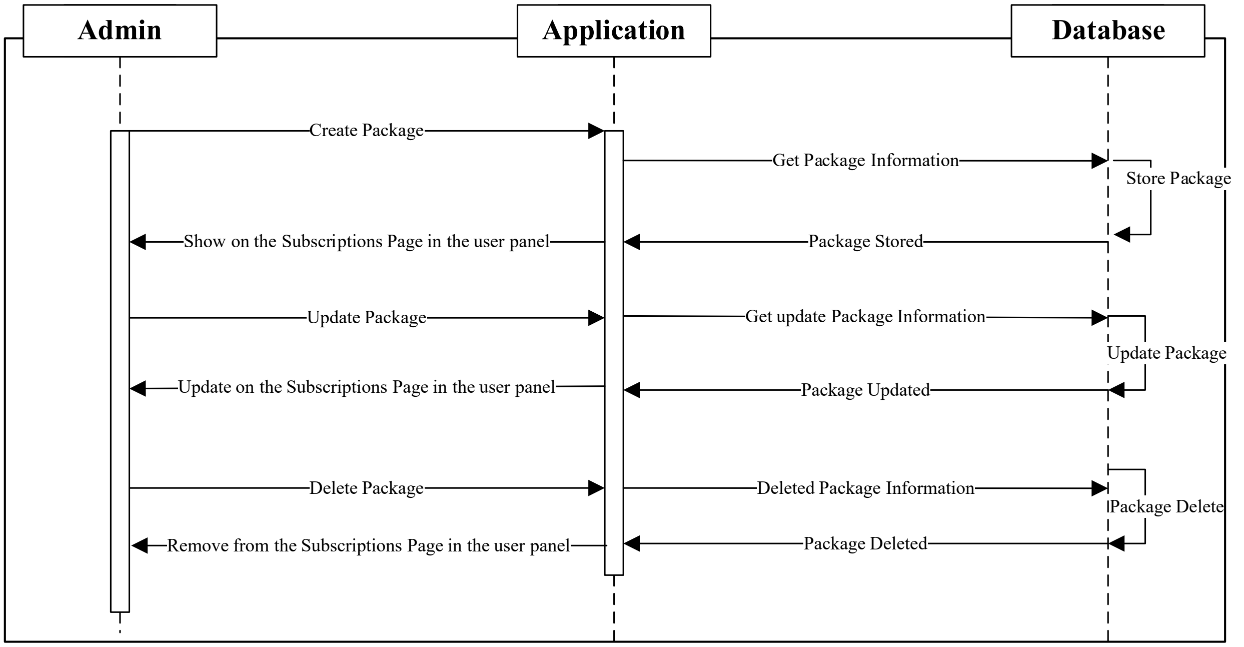
Chapter # 7
Sequence Diagrams

7. Sequence Diagram

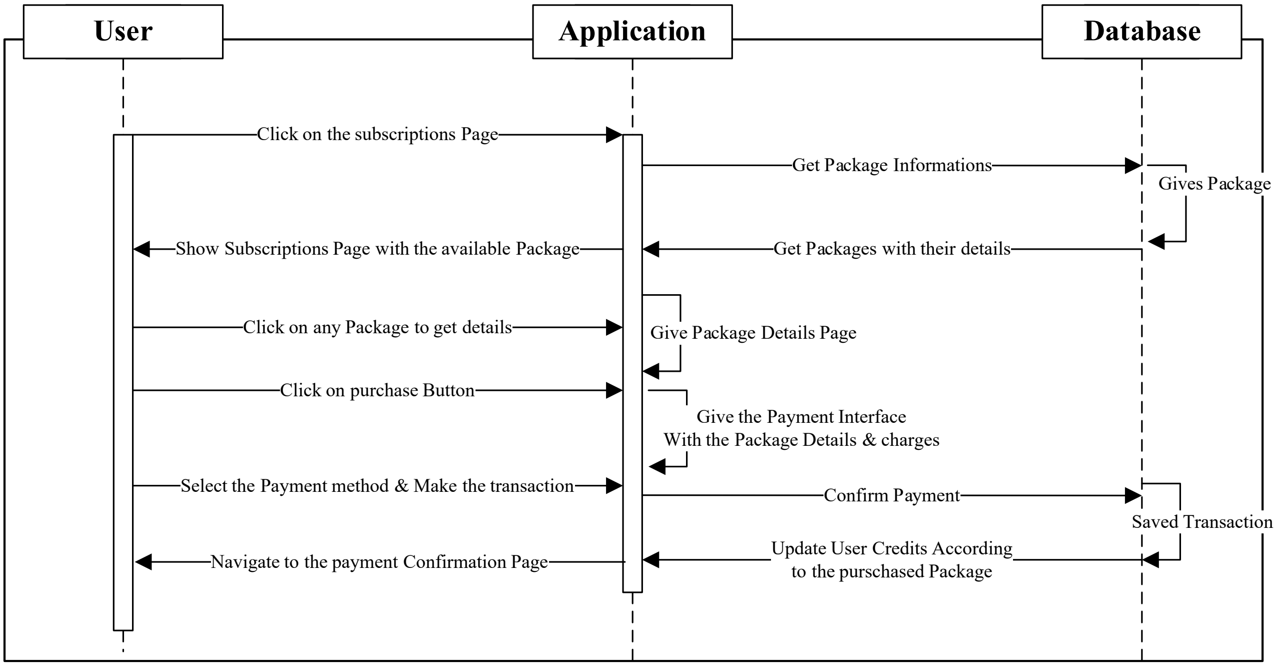
1. SD_Login



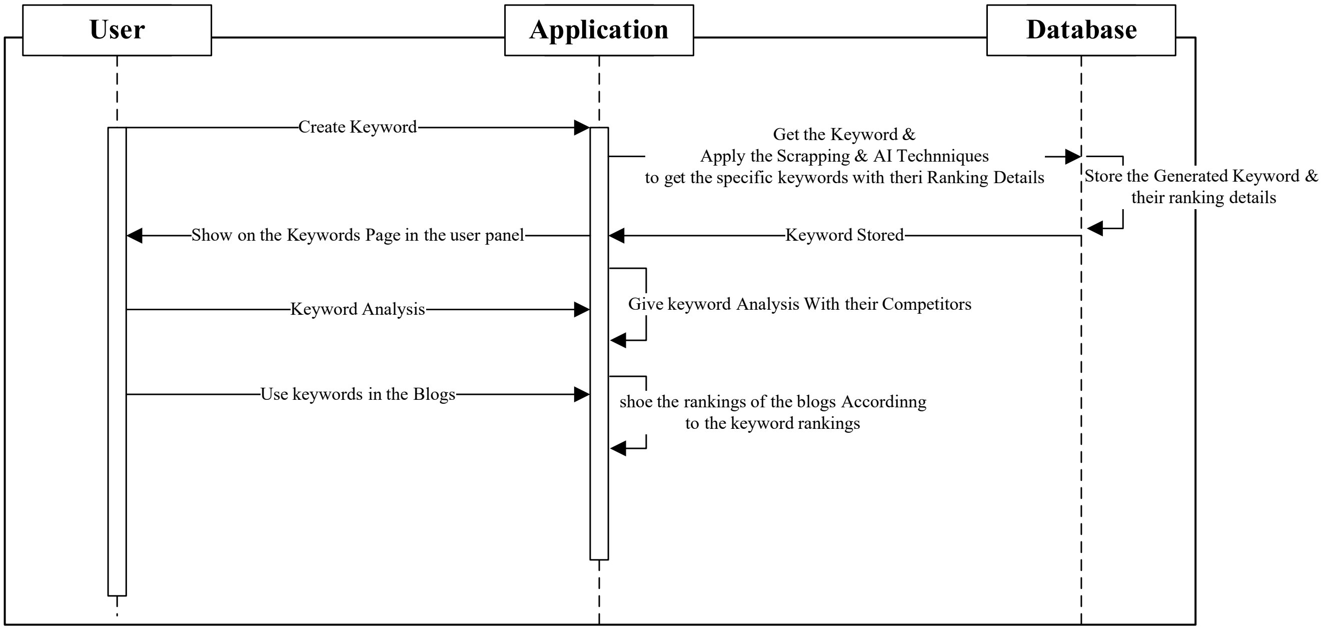
2. SD_Maintain_Packages


Admin
Application
Database
Create Package
Get Package Information
Store Package
Show on the Subscriptions Page in the user panel
Package Stored
Update Package
Get update Package Information
Update Package
Update on the Subscriptions Page in the user panel
Package Updated
Delete Package
Deleted Package Information
Package Delete
Remove from the Subscriptions Page in the user panel
Package Deleted

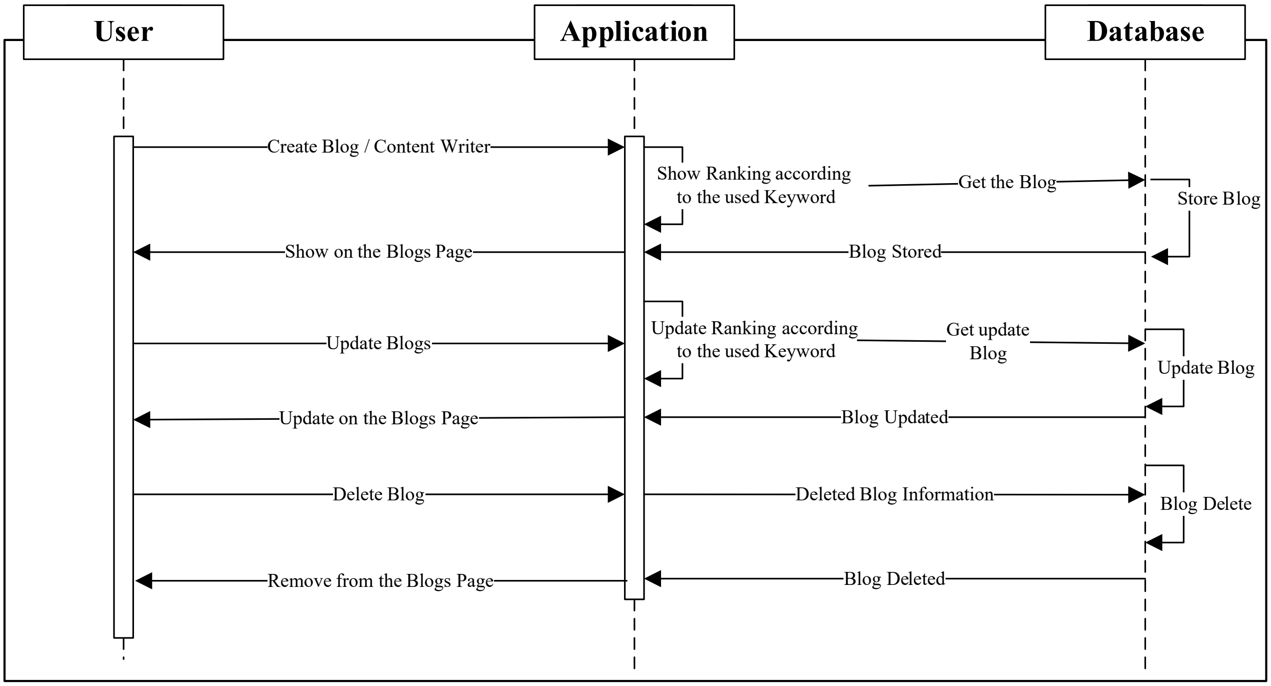
3. SD_Payment


User
Application
Database
Click on the subscriptions Page
Get Package Informations
Gives Package
Show Subscriptions Page with the available Package
Get Packages with their details
Click on any Package to get details
Give Package Details Page
Click on purchase Button
Give the Payment Interface
With the Package Details & charges
Select the Payment method & Make the transaction
Confirm Payment
Saved Transaction
Navigate to the payment Confirmation Page
Update User Credits According
to the purschased Package

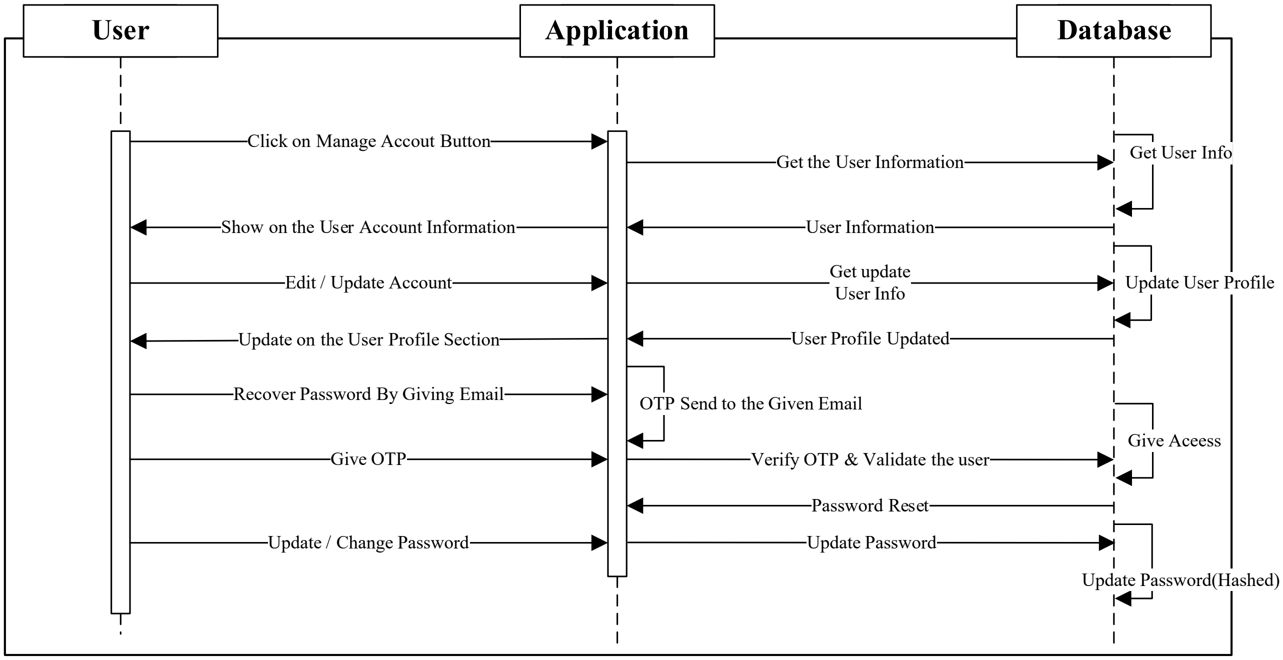
4. SD_Generate_Keywords


User
Application
Database
Create Keyword
Get the Keyword &
Apply the Scrapping & AI Technniques
to get the specific keywords with theri Ranking Details
Store the Generated Keyword &
their ranking details
Show on the Keywords Page in the user panel
Keyword Stored
Keyword Analysis
Give keyword Analysis With their Competitors
Use keywords in the Blogs
shoe the rankings of the blogs Accordinng
to the keyword rankings

5. SD_Blog



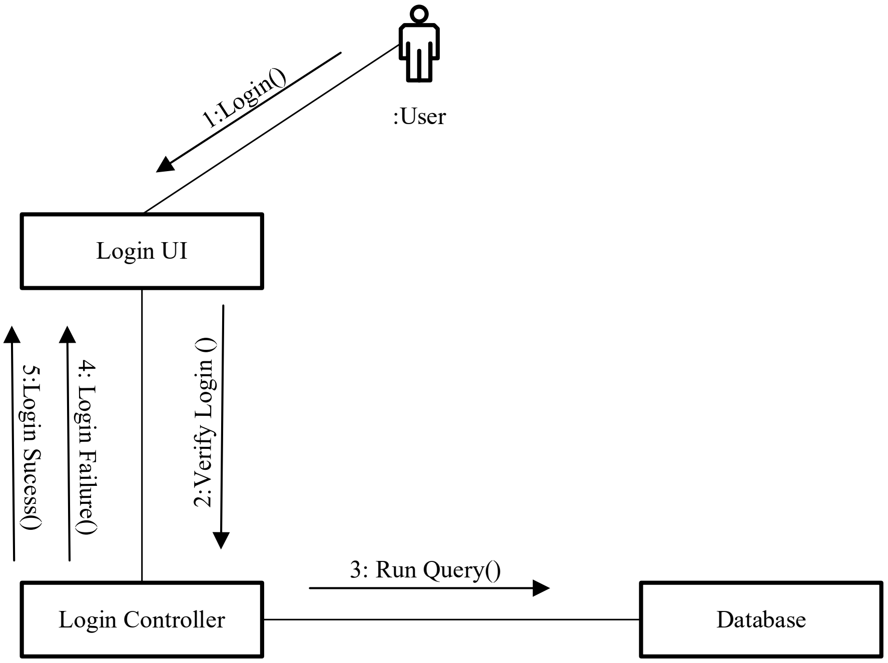
6. SD_Recover_Password _&_Update_Profile



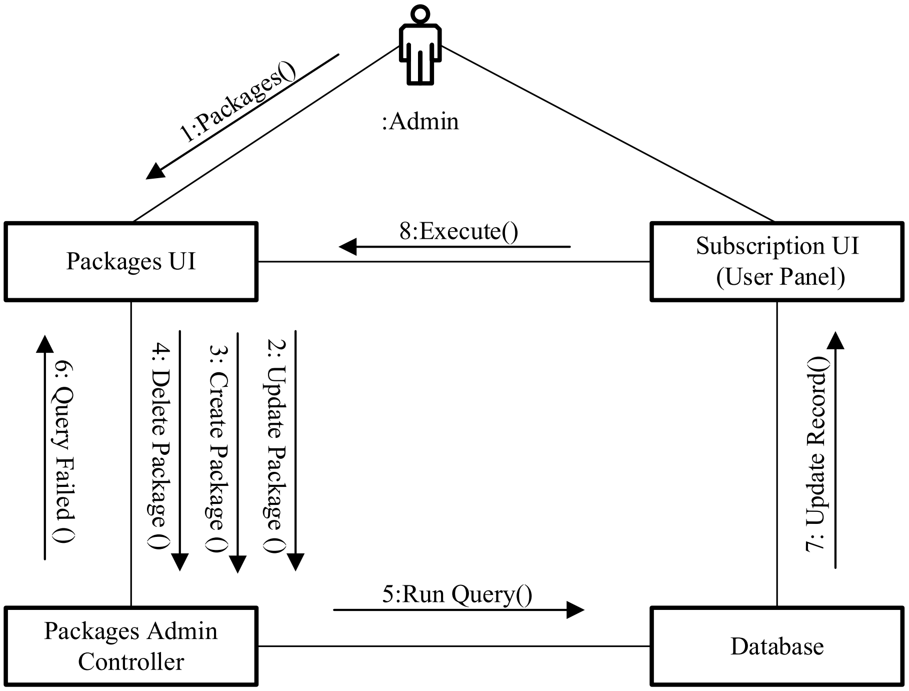
Chapter # 8
Collaboration Diagrams

8. *Collaboration Diagram*

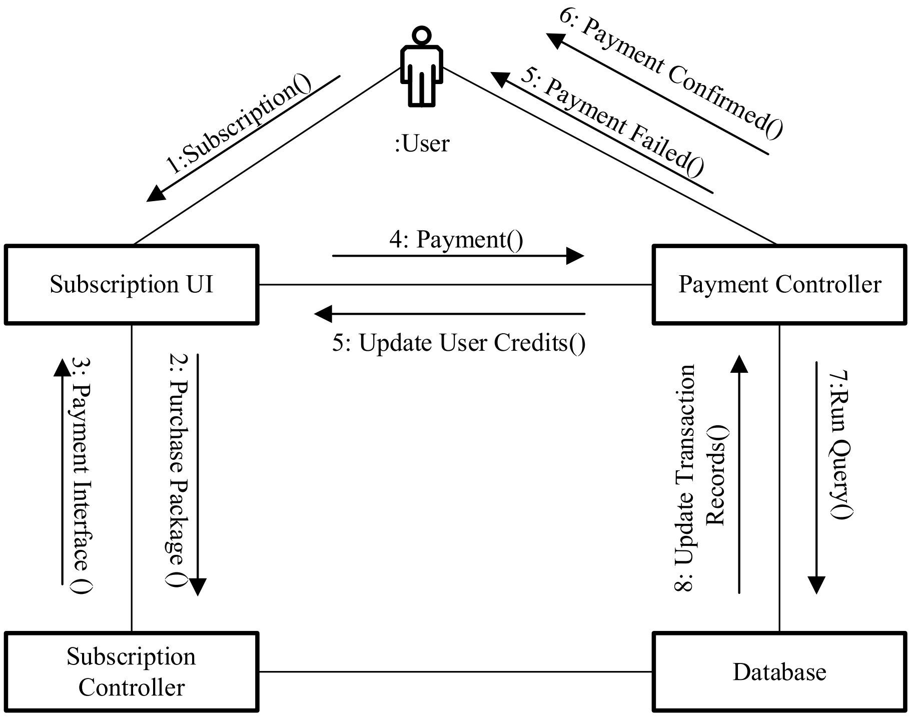
1. CD_Login



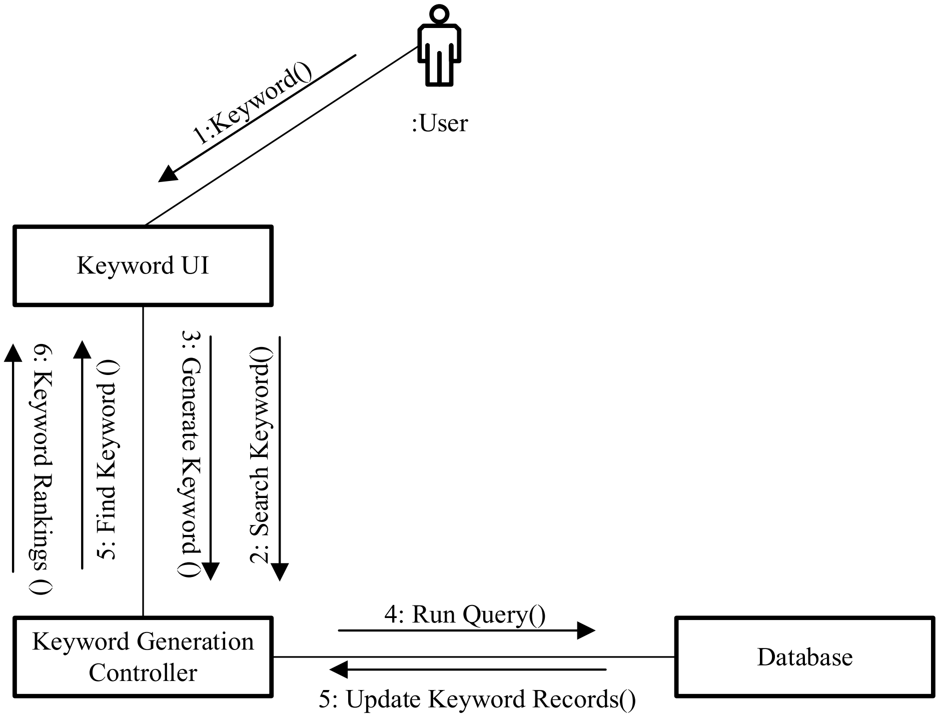
2. CD_Maintain_Packages



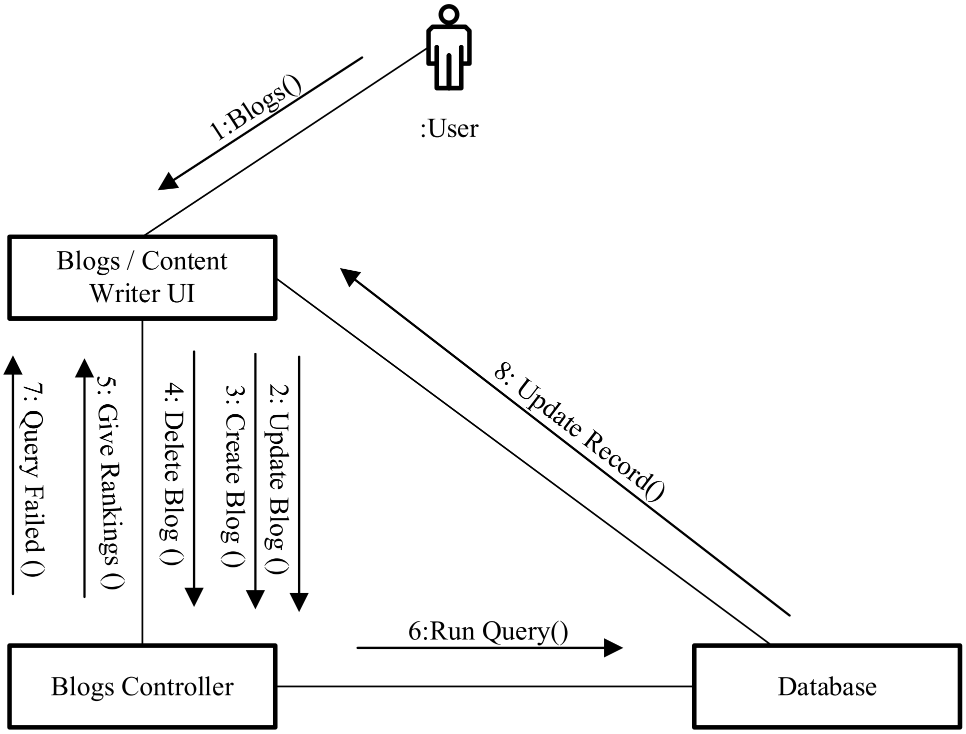
3. **CD_Payment**



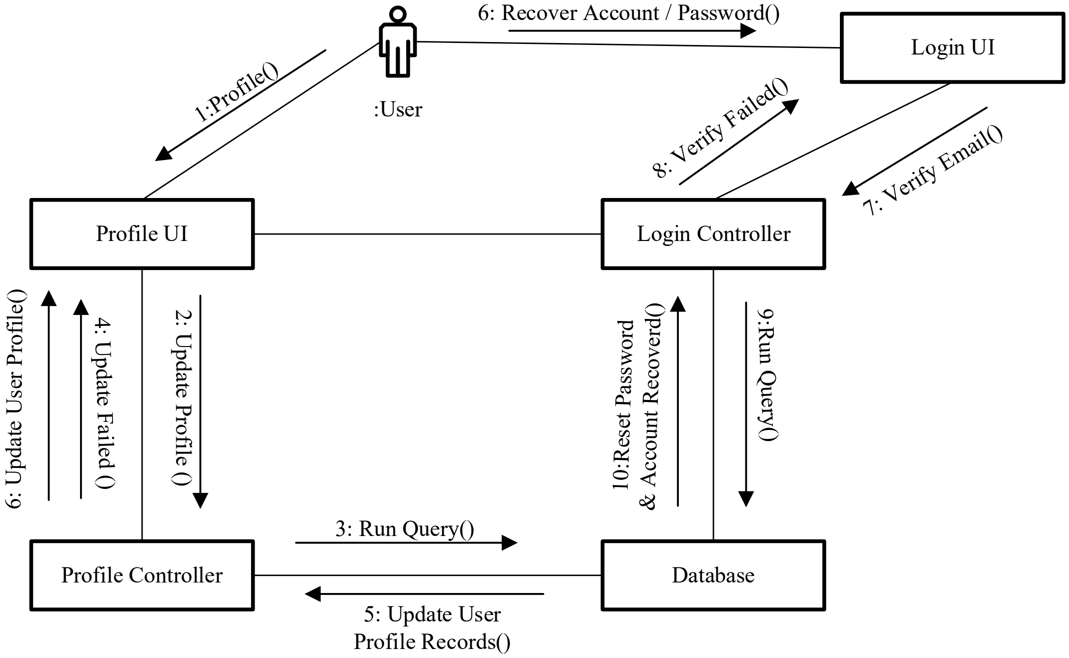
4. **CD_ Generate _Keywords**



5. CD_Blog



6. CD_Recover_Password _&_Update_Profile



Chapter # 9
Operation Contracts

9. 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.

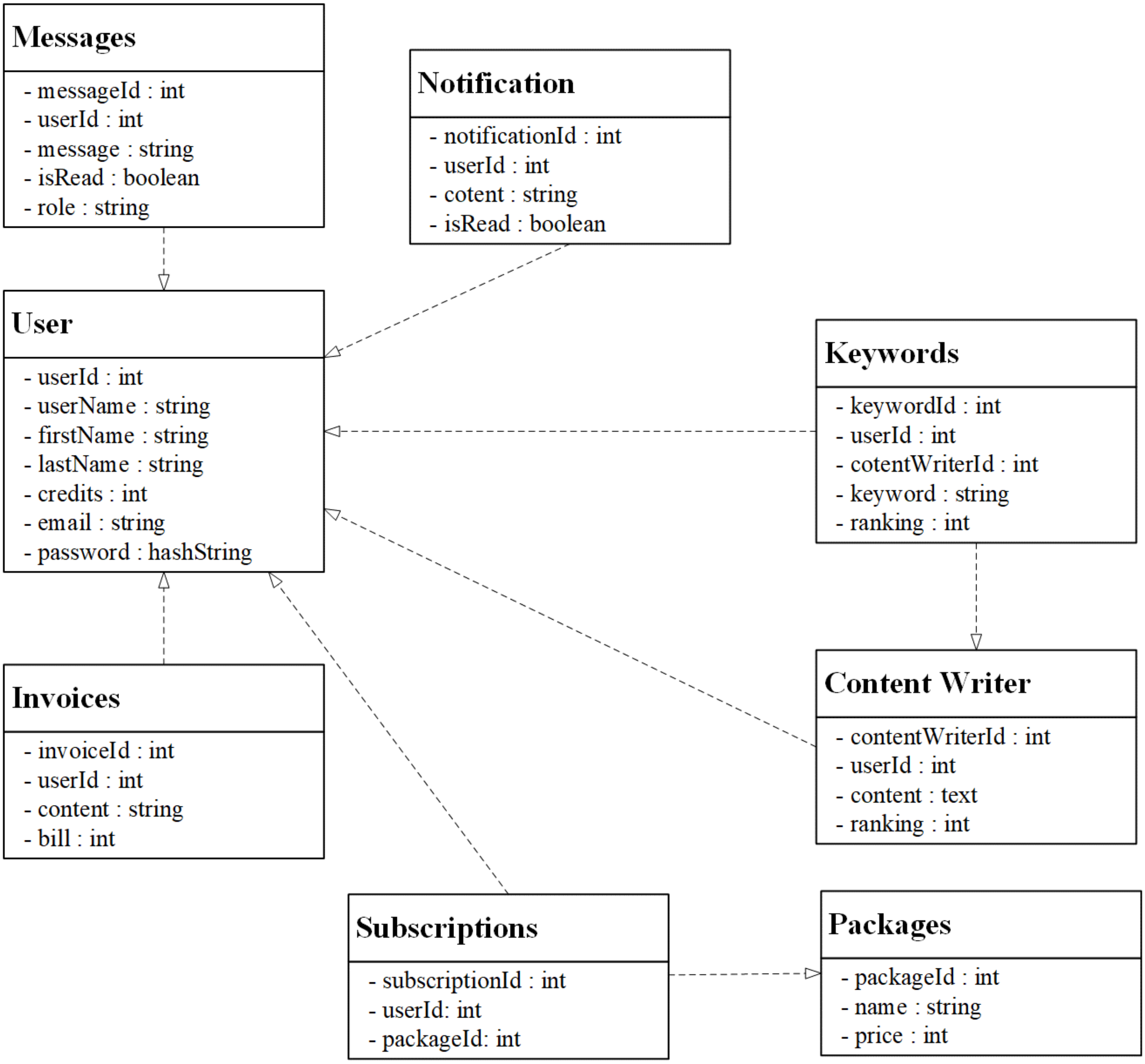
5. **OP_Blog**
    - **Operation:** Write Blog
    - **Responsibilities:** User first have some AI generated keywords for their content and write blogs and gets their ranking.
    - **Cross References:** Use Case: UC_ Blogs
    - **Exceptions:** User cannot write blogs.
    - **Preconditions:** User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

6. **OP_ Recover_Password _&_Update_Profile**

    - **Operation:** Recover Password & Update Profile.
    - **Responsibilities:** User can update its profile and change password.
    - **Cross References:** Use Case: UC_ Recover_password_&_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.

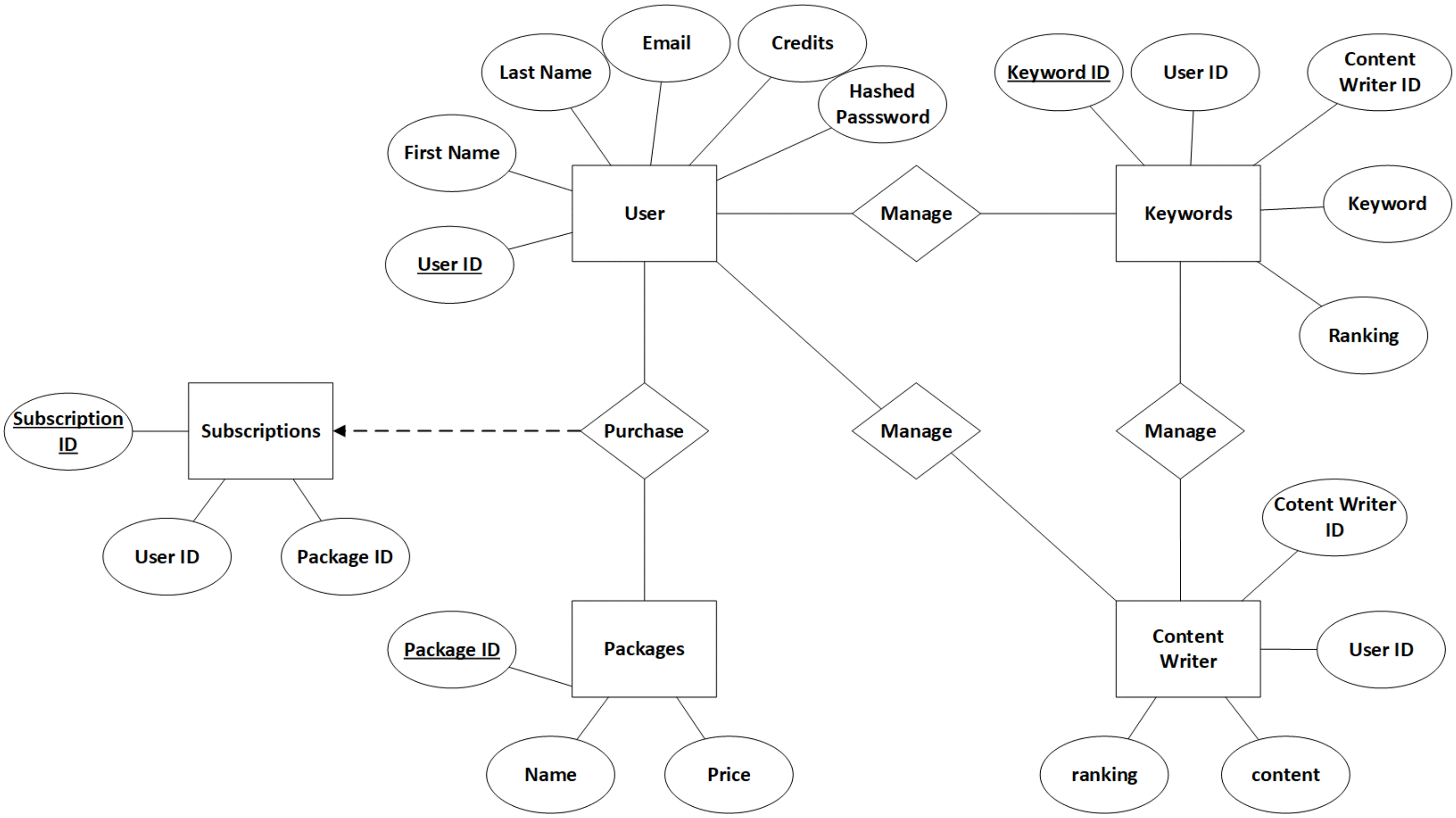
Chapter # 10
Class Diagram

10. Class Diagram



Chapter # 11
Data Model

11. Data Model


Email
Credits
Last Name
Hashed Passsword
First Name
User
User ID
Manage
Keyword ID
User ID
Content Writer ID
Keywords
Keyword
Ranking
Subscription ID
Subscriptions
Purchase
Manage
Manage
User ID
Package ID
Cotent Writer ID
Package ID
Packages
Content Writer
User ID
Name
Price
ranking
content