

Fitts: System Design



Student Name	Student Number
--------------	----------------

빙 (Nguyen Khanh Binh)	2019711305
-----------------------	------------

임근식 (Lim Geunsik)	2019711699
-------------------	------------

하지크 (Haziq Hamzah)	2019711435
--------------------	------------

장한빈 (Jang Hanbin)	
-------------------	--

Contents

Contents.....	2
1. Preface	6
1.1. Objectives.....	7
1.2. Document Structure	8
2. Introduction	9
2.1. Objectives.....	9
2.2. Applied Diagram.....	9
2.2.1. UML.....	9
2.2.2. ER Diagram.....	10
2.2.3. Class Diagram.....	12
2.2.4. Package Diagram.....	13
2.2.5. State Diagram.....	14
2.2.6. Sequence Diagram.....	15
2.2.7. Deployment Diagram.....	16
2.3. Applied Tool.....	17
2.3.1. yUML	17
2.3.2. Visio	17
2.3.3. Creately.....	18

2.4. Project Scope	18
3. System Architecture - Overall.....	20
3.1. Objectives.....	20
3.2. System Organization.....	20
3.2.1. Frontend Application	22
3.2.2. Backend Application.....	23
4. System Architecture - Frontend.....	25
4.1. Objective.....	25
4.2. Subcomponents	26
4.2.1. MyPage.....	26
4.2.2. Item Ranking.....	29
4.2.3. Item Search.....	32
4.2.4. Item Detail	34
4.2.5. Recommendation.....	36
5. System Architecture - Backend.....	39
5.1. Objectives.....	39
5.2. Overall Architecture.....	39
5.3. Subcomponents	41
5.3.1. Server	41
5.3.2. Review Collection System	44
5.3.3. Review Analysis System	47

5.3.4. Review Point System	48
5.3.5. Item Ranking System	49
6. Database Design.....	50
6.1. Objectives.....	50
6.2. ER Diagram.....	51
6.2.1. User Entity.....	53
6.2.2. Bookmark Entity	54
6.2.3. Certification Entity.....	54
6.2.4. Item Entity.....	55
6.2.5. Recommendation Entity	55
6.2.6. Point Entity	56
6.2.7. Review Entity.....	57
6.2.8. Review Reference Entity.....	57
6.3. Relational DB Schema	58
6.3.1. Item.....	59
6.3.2. User.....	59
6.3.3. Certification	60
6.3.4. Point.....	60
6.3.5. Bookmark	61
6.3.6. Reference.....	62
6.3.7. Seller	63

Design Specification

6.3.8. Review	63
6.3.9. Recommendation.....	64
7. Protocol Design.....	66
7.1. Objectives.....	66
7.2. XML.....	66
7.3. CURL.....	67
7.4. CURL API	70
7.4.1. Item	70
7.4.2. User.....	71
7.4.3. Certification	74
8. Appendix: Terminology	76
9. Appendix GitHub Repository	77

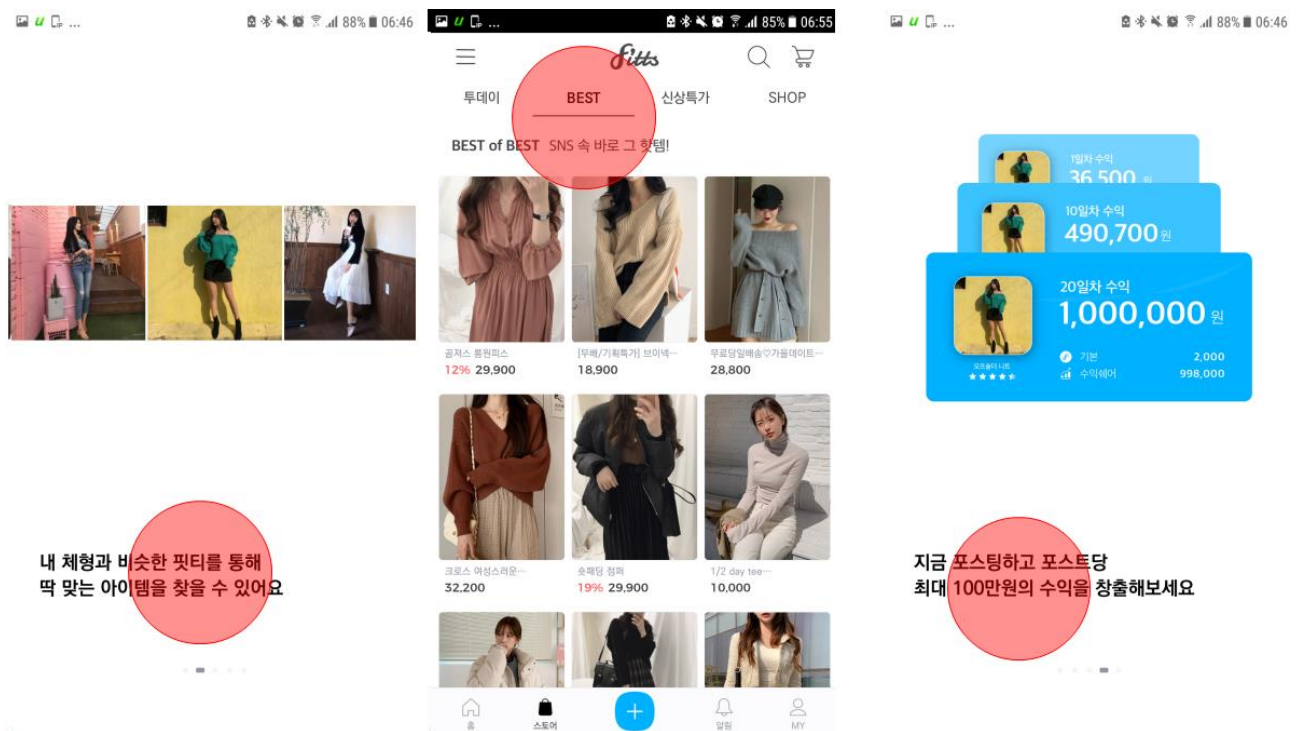
1. Preface

In this chapter, we address the entire structure of the document. And we briefly introduce the content of each chapter.



Drawing 1 Fitts's homepage

This section is intended to help readers who want to understand Fitts' system architecture. This document helps users understand how Fitts' entire system architecture is designed.



Drawing 2 The major functions (=strong points) of the Fitts system

1.1. Objectives

This document describes the system design that Fitts composes. In this chapter, we briefly describe the document structure of how the overall flow diagram, Frontend system architecture, Backend system architecture, etc. are organized to understand Fitts' system design.

1.2. Document Structure

1. **Introduction**

This document describes in detail the system design that Fitts comprises. We describe the overall flow chart, frontend system architecture, and backend system architecture to understand Fitts' system design.

2. **System Architecture**

This section describes the overall operating structure of the system. Then, we explain how to configure each subsystem and hardware environment to understand how the overall structure of the system is implemented.

3. **Protocol Design**

Fitts' system architecture consists of the Frontend System, which handles the user's interface, and the Backend System, which actually handles the user's requests. We also describe how interfaces and protocols are configured to communicate between frontend and backend systems. It also describes how the user interface is connected to the backend system.

4. **Database Design**

We describe in detail the schemes of the database based on the functional diagrams and the non-functional diagram requirements described in the system requirements document. The properties of each data entity are represented using an ER diagram. Finally, we describe the relational database schema and the SQL DDL (Data Definition Language).

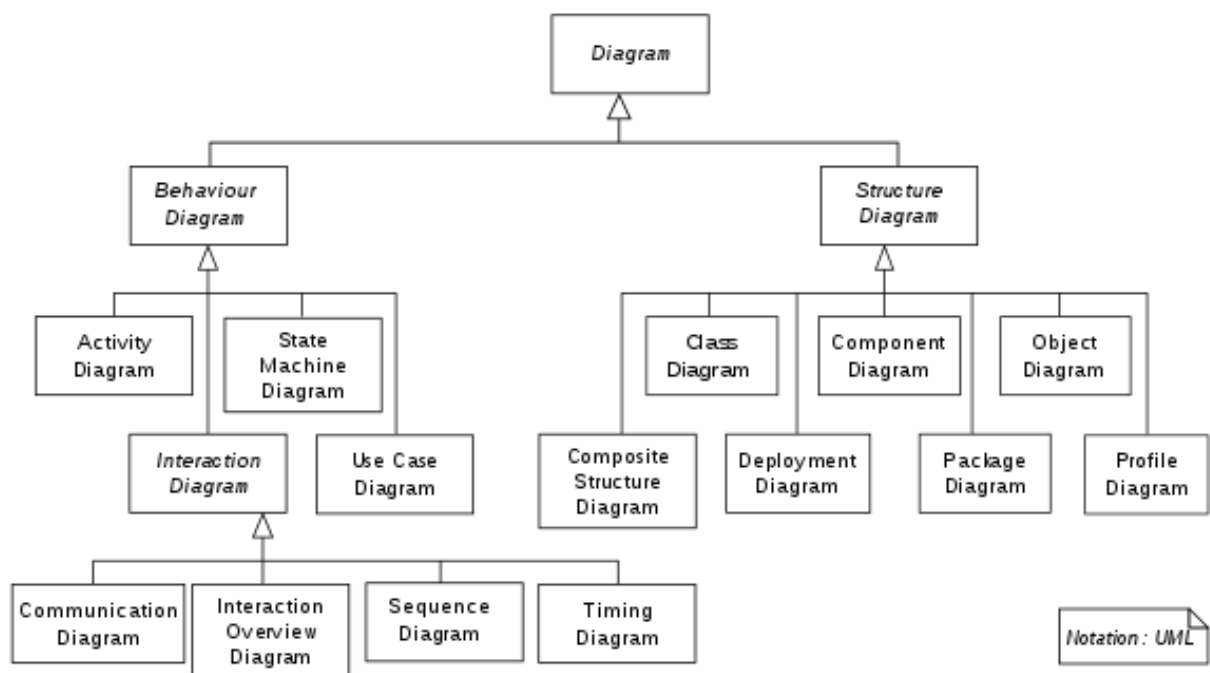
2. Introduction

2.1. Objectives

This document describes in detail the system design that Fitts comprises. We describe the overall flow chart, frontend system architecture, and backend system architecture to understand Fitts' system design.

2.2. Applied Diagram

2.2.1. UML



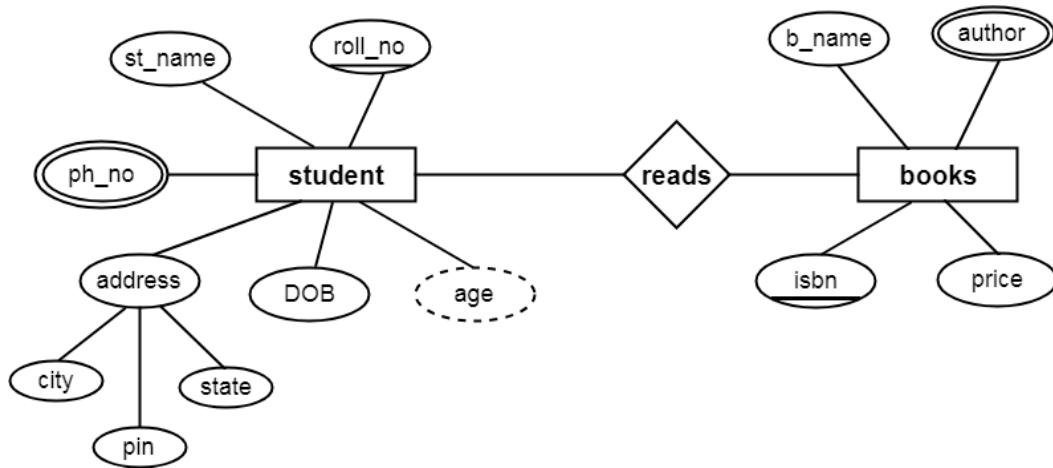
Drawing 3 Two categories of UML

The Unified Modeling Language (UML) is a general-purpose, developmental, modeling language in the field of software engineering that is intended to provide a standard way to visualize the design of a system. UML has many types of diagrams, which are divided into two categories: behavior

diagram, structure diagram. Some types represent structural information, and the rest represent general types of behavior, including a few that represent different aspects of interactions. Behavior diagrams emphasize what must happen in the system being modeled. Since behavior diagrams illustrate the behavior of a system, they are used extensively to describe the functionality of software systems. As an example, the activity diagram describes the business and operational step-by-step activities of the components in a system. Structure diagrams emphasize the things that must be present in the system being modeled. Since structure diagrams represent the structure, they are used extensively in documenting the software architecture of software systems. For example, the component diagram describes how a software system is split up into components and shows the dependencies among these components.

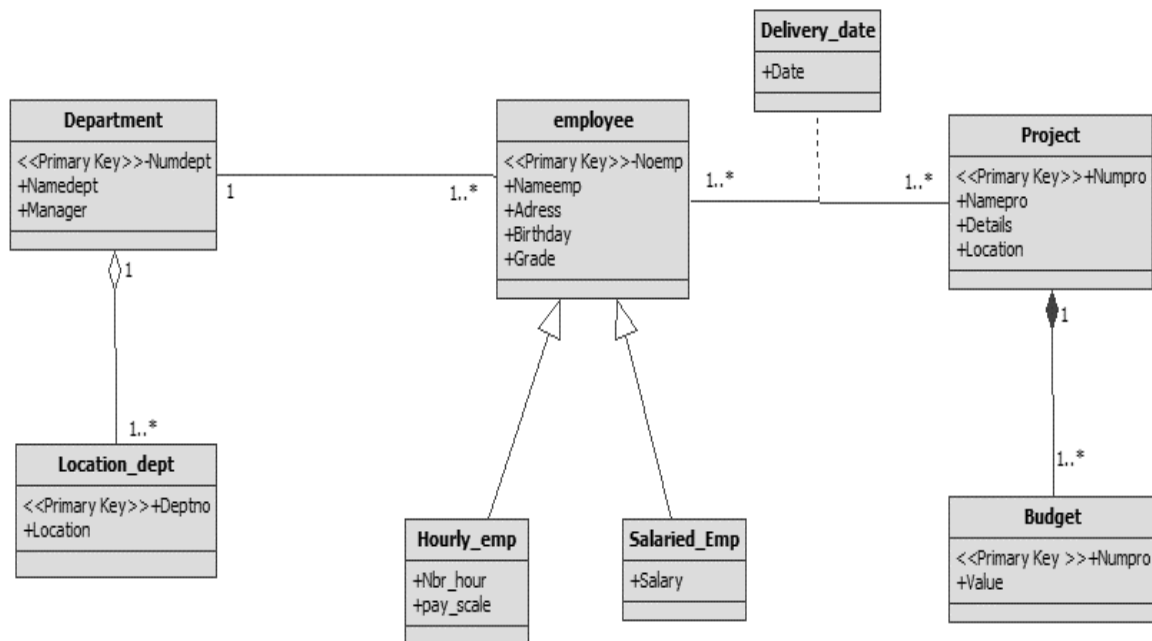
In this document, we use mostly the structure diagrams to illustrate class diagram, component diagram, package diagram, and object diagram. Also, we use the behavior diagrams such as activity diagram, user case diagram, sequence diagram.

2.2.2. ER Diagram

**Drawing 4 Example of ER Diagram**

An entity-relationship model (or ER model) describes interrelated things of interest in a specific domain of knowledge. A basic ER model is composed of entity types and specifies relationships that can exist between entities. In software engineering, an ER model is commonly formed to represent things need to perform business processes. Consequently, the ER model becomes an abstract data model that defines a data or information structure which can be implemented in a database, typically a relational database. In our case, we describe the ER Diagram to express DB scheme and DDL.

2.2.3. Class Diagram

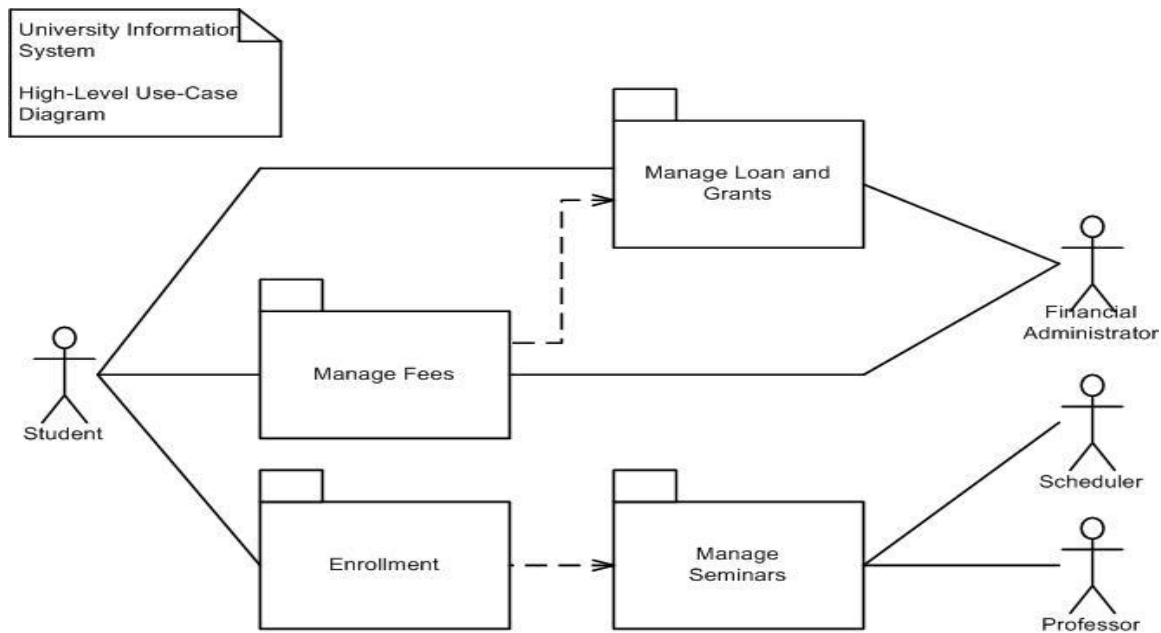


Drawing 5 Example of ER diagram

A class diagram in the Unified Modeling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among objects.

The class diagram is the primary building block of object-oriented modeling. It is used for general conceptual modeling of the structure of the application, and detailed modeling translating the models into programming code. Class diagrams can also be used for data modeling. The classes in a class diagram represent both the main elements, interactions in the application, and the classes to be programmed.

2.2.4. Package Diagram

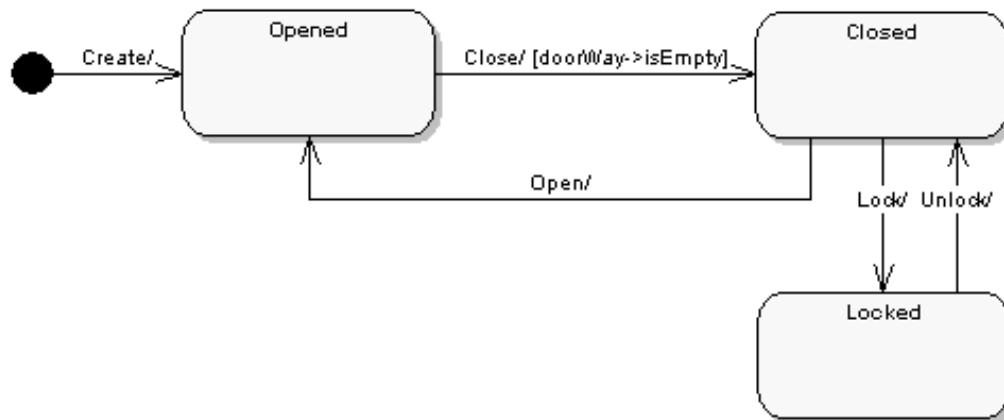


Drawing 6 Example of Package Diagram

A package diagram in the Unified Modeling Language depicts the dependencies between the packages that make up a model. Package diagrams can use packages containing use cases to illustrate the functionality of a software system.

Package diagrams can use packages that represent the different layers of a software system to illustrate the layered architecture of a software system. The dependencies between these packages can be adorned with labels and stereotypes to indicate the communication mechanism between the layers.

2.2.5. State Diagram

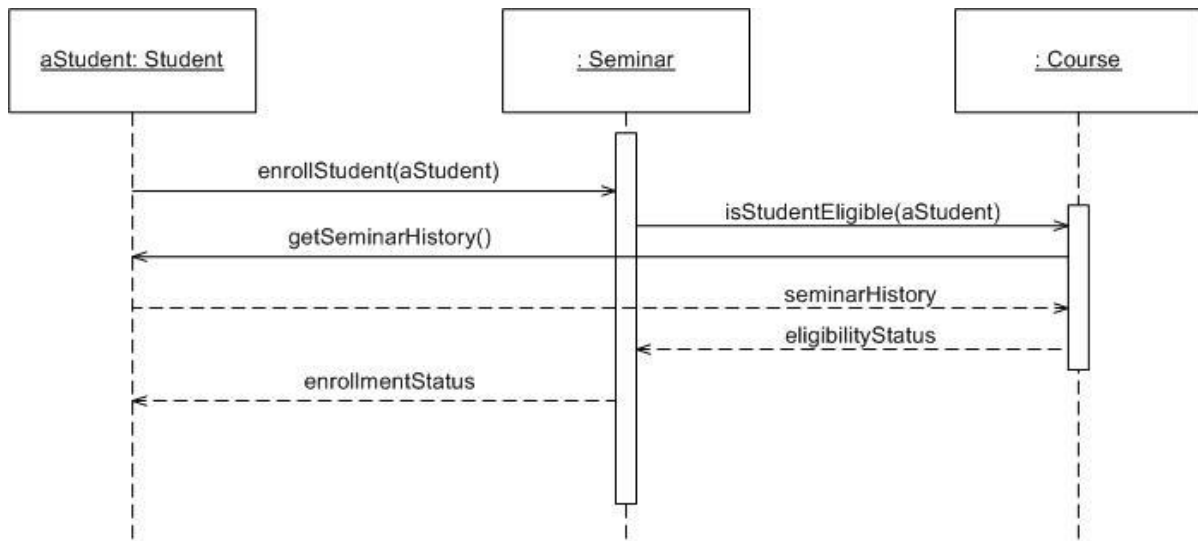


Drawing 7 Example of State Diagram

State Diagram is a technique to represent different states of a system and all possible next states based on some particular activities which trigger the change of the state.

This kind of diagram is critical to analyze different scenarios of the system as the states are represented as nodes and events as arcs, which helps in identifying the behavior of the object classes defined in the class diagram.

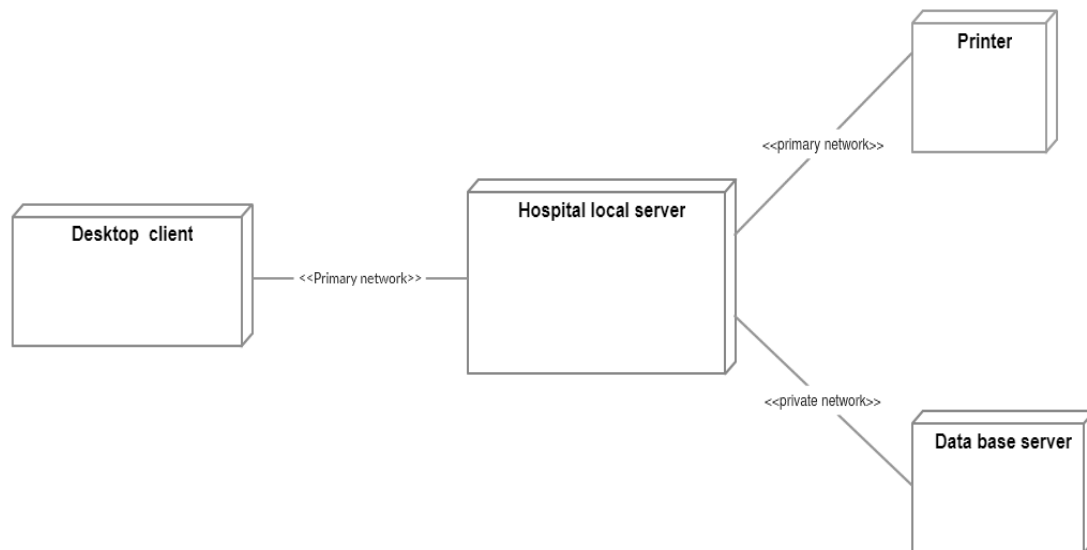
2.2.6. Sequence Diagram



Drawing 8 Example of Sequence Diagram

A sequence diagram depicts the interaction between objects in a sequential order ie the order in which these interactions take place. We can also use the terms event diagrams or event scenarios to refer to a sequence diagram. Sequence diagrams describe how and in what order the objects in a system function.

2.2.7. Deployment Diagram



Drawing 9 Example of Deployment Diagram

Deployment diagrams are used to visualize the hardware processors, nodes, devices of a system, the links of communication between them, and the placement of software files on that hardware. To describe a web site, for example, a deployment diagram would show what hardware components ("nodes") exist (e.g., a web server, an application server, and a database server), what software components ("artifacts") run on each node (e.g., web application, database), and how the different pieces are connected (e.g. JDBC, REST, RMI).

2.3. Applied Tool

2.3.1. yUML



Drawing 10 yUML Logo

yUML is an online modeling tool that provides many basic templates and shapes. Most of the diagrams used in this document were created with this tool. Also, yUML allows you to create pictures without using any visual drawing tools. You can write code to request an image programmatically using a GET or a PUT.

2.3.2. Visio



Drawing 11 Visio Logo

Visio is a powerful UML authoring tool from Microsoft. Since Visio has so many built-in geometric shape features, we used it to create complex diagrams that would be difficult to create with online-based draw.io.

2.3.3. Creately



Drawing 12 Creately Logo

Creately draws even the most complex of data designs easily with its intuitive features. Also, we used this online tool because it easily creates smart connectors, preset styling options, and a full ER diagram shape library. Creately can be easily created with a simple operation compared to draw.io when trying to create an ER Diagram, so we used Creately to generate an ER Diagram.

2.4. Project Scope

Fitts present system provides an online shopping system that helps the clothes can look good when wearing clothes users to buy clothes. Photographed wearing the clothes you buy clothes, customers buy photos and upload their own experience in the online store. Then there are other customers to see what it looks when wearing clothes. Also, customers look at the reviews online shopping company shall pay the amount of compensation equivalent. That's Fit service provides a "Fitts Store" service to receive adequate compensation by sharing experience and shoppers buy clothes. Key features of this system are to help based on a review of your product experience allowing customers to choose the best outfits for you. As a result, customers can purchase the product that best fits with your body. This chapter describes the system design of Fitts. The system architecture is composed as follows:

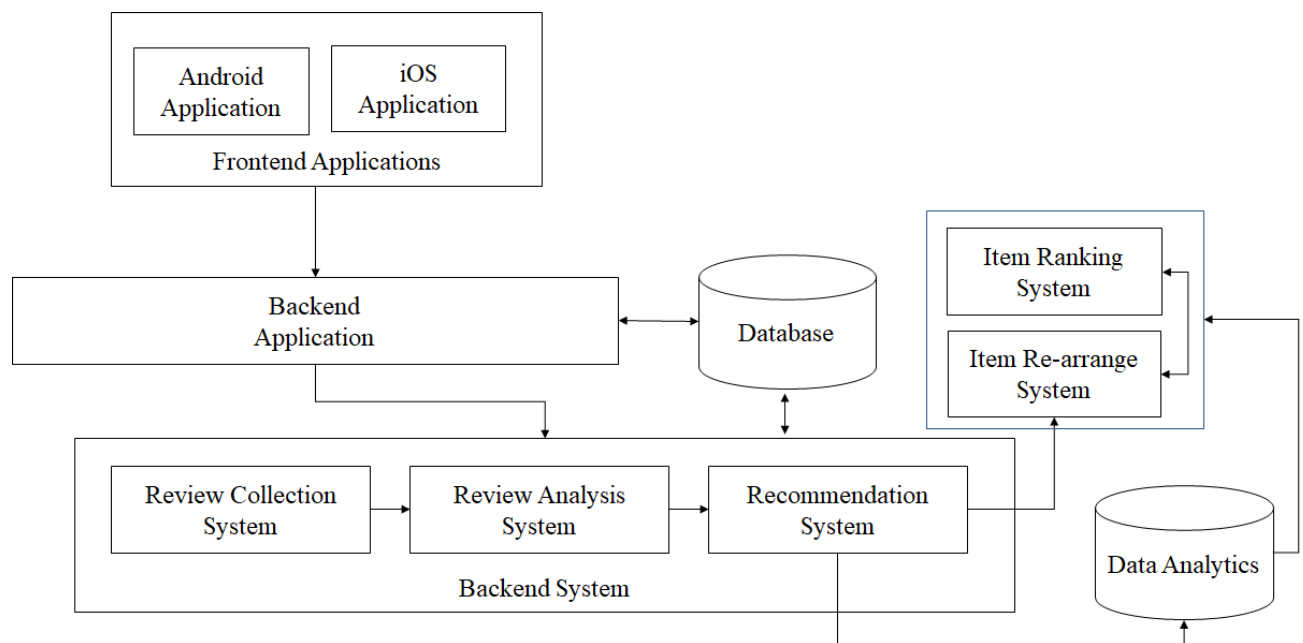
1. Frontend System: This system is responsible for interaction with the software system and the user.
2. Backend System: This system is in response to a data request from the Frontend System. Then executes the internal system such as Review Collection System, Review Analysis System, Recommendation System.
 - A. Review Collection System: Collect the reviews that you have created. Then the system into a suitable structure for the ad to show to customers reviews rearranges reports and information.
 - B. Review Analysis system: This system results in a list of review required in real-time from the Review Collection System. The System is to analyze the collected information, review and analyze the positive assessment of fitness levels and reviews of reviews. The system database stores the analysis information.
 - C. Recommendation system: This system, based on the popularity of items using the analysis information stored in the database, it is recommended the product to the user. The system updates the evaluation value and the approved categories of goods for each additional product reviews.

3. System Architecture - Overall

3.1. Objectives

This section describes the overall operating structure of the system. Then, we explain how to configure each subsystem and hardware environment to understand how the overall structure of the system is implemented.

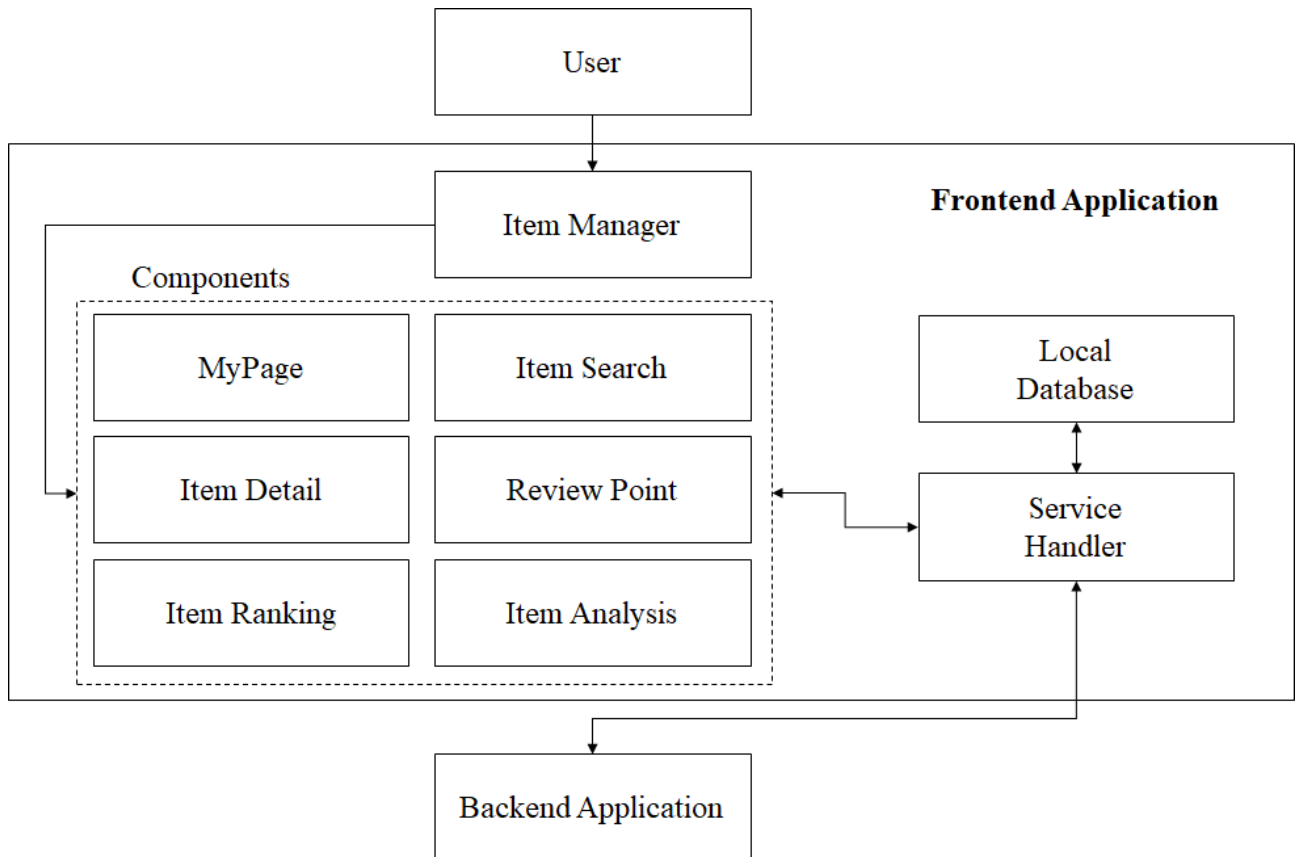
3.2. System Organization



Drawing 13 Overall System Architecture

We show overall system architecture after analyzing the operating structure of the service. Fitts was designed as a server-based model. The Frontend Application is responsible for interaction with the user. The picture shows Frontend Application and Backend Application using the XML data format, and the HTTP protocol to send and receive data with each other. At this time, the Backend Application shall process the request of Frontend Application. Data needed to process the requests of Frontend Application may obtain information from the database. Or after processing a request for Frontend Application stores, the data are processed by the database. Backend Application performs tasks such as using a SQL query for processing the data stored in the database. Backend Application is to perform functions such as Frontend whenever you request Application to manage recommendations and reviews of the product Backend System This information collection, information analysis reviews, product recommendation. The list of product reviews and information displayed in the online store are stored in a database and management. When a user writes a summary of information products, Frontend Application will be asked to save the information to review Backend Application software systems. Then Review Collection System to collect and evaluate data from the Backend Application identifies whether the correct information, review the information. Review and Analysis System will store the information organized into useful information after analyzing the information collected in the database. Then the Recommendation System is based on the score and reviews the confidence score of the product using updated review information to update the recommended categories of goods. Item Ranking System utilizes a recommendation rating, reviews data to manage the grade of the product. Item Re-arrange System will rearrange the recommended categories using the updated product rating.

3.2.1. Frontend Application



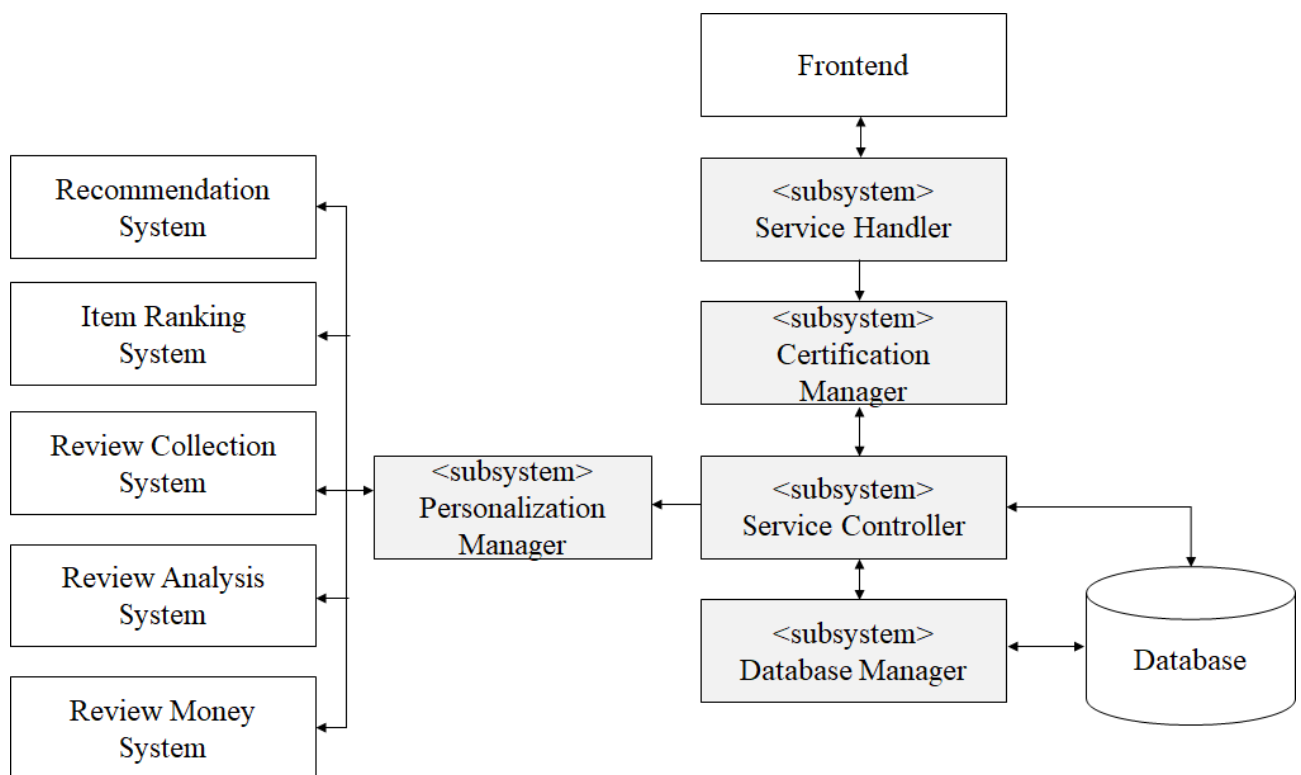
Drawing 14 System Architecture - Frontend

Frontend Application is a system that is responsible for interaction with the user. Frontend Application must manage each component via a web framework for mobile environments. The Front Application Item Manager is present in the elements, such as MyPage, Item Search, Item Detail, Point, Item Ranking Item Analysis. The Local Database system manages shared Resource sharing between components. Service Handler performs related work to ensure that each component can communicate with the Backend Application. If each part were to request data

needed by Service Handler, the data are transmitted to the Service Handler of Backend Application.

At this time, the system requires data with the communication protocol after the conversion.

3.2.2. Backend Application



Drawing 15 System Architecture - Backend Application

The Backend Application is responsible for performing the tasks requested by the Frontend Application. When Frontend Application is customized requirements from the user forwarded to the Backend Application, the Backend Application should process the job to be actually performed. The

first Certification Manager is responsible for the security processing for the authenticated user. If the user is a member of the mall, the Service Controller stores user-related log information to the local database. The Service Controller then asks the user to review the evaluation of the registered Personalization manager. Personalization manager collects, analysis, reliability calculations, and reviews reactivity calculations, each of the operations to the recommendation system, item ranking system, review collection system, review analysis system, review point system to perform such a review compensation amount calculated to review information allocates and manages.

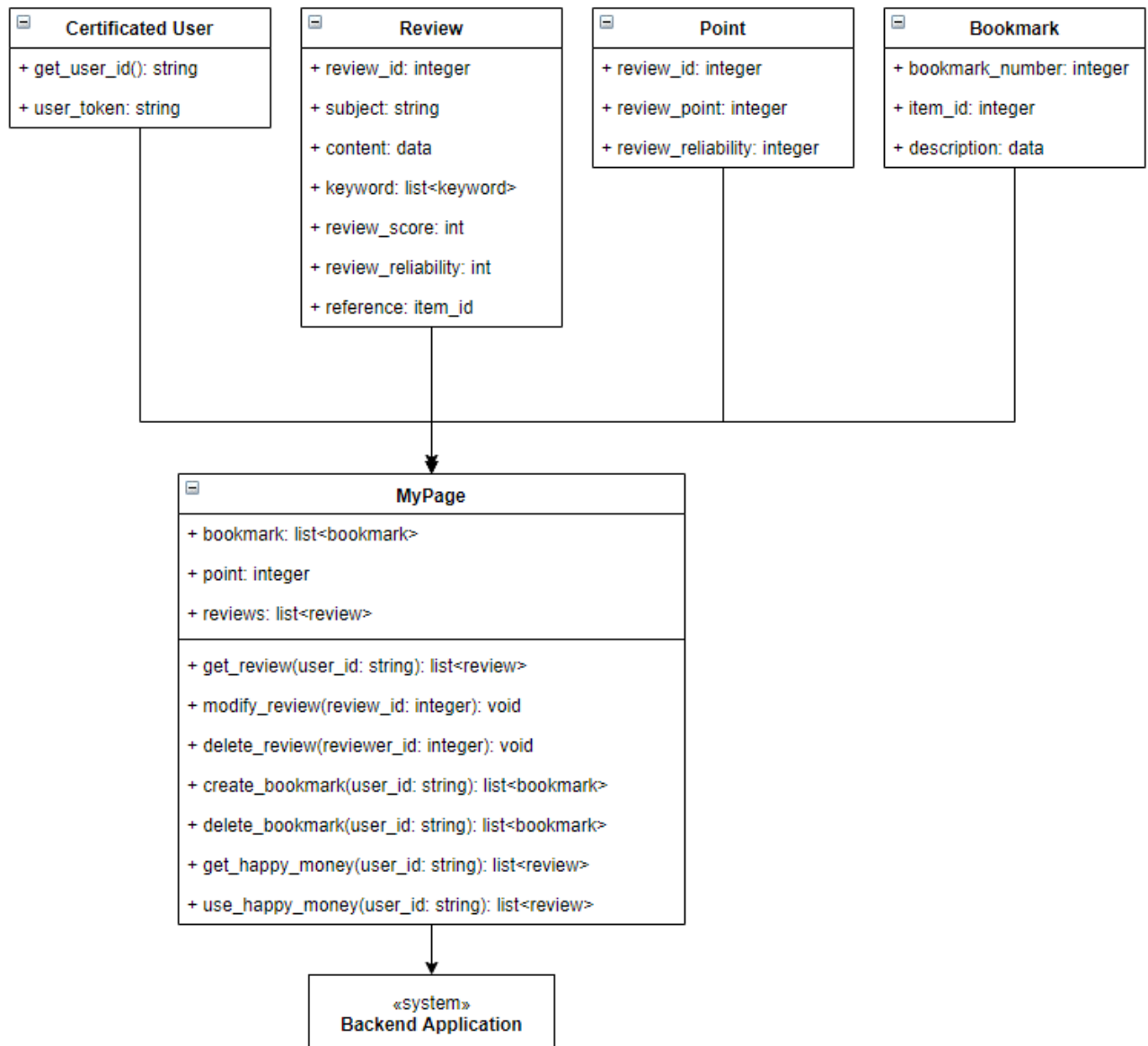
4. System Architecture - Frontend

4.1. Objective

This section describes the Frontend System responsible for the interaction with the user. And it describes the relationship between a method and component for each component that is present for the Frontend System.

4.2. Subcomponents

4.2.1. MyPage



Drawing 16 System Architecture - Frontend - MyPage

1. MyPage - The object of MyPage

A. attributes

- + Bookmark: the list of users to bookmark the item
- + Point: the user receives a payment from the online store after the company registered an electronic money product reviews
- + Reviews: Review the list that you created

B. methods

- + Get_review (user_id: String) function to import user-created reviews
- + Modify_review (review_id: Integer): function to correct a review you have written
- + Delete_review (review_id: Integer): function to delete a user-created reviews
- + Create_bookmark (user_id: String) function for users to add specific items to your bookmarks
- + Delete_bookmark (bookmark_id: Integer): function to delete a Bookmark
- + Get_point (user_id: string): function to query the electronic money payment from the user is shopping company
- + Use_point (user_id: string): function to use when you buy clothes to electronic money payments received from the Company Store

2. Certificated User: certificated user object

A. attributes

- + User_toekn (string): Token information for user authentication using the Token ID

B. method

- + Get_user_id (): function to import user ID information

3. Review: the review object of the item

A. attributes

- + Review_id (integer): review ID that is generated during the complete review
- + Subject: review title
- + Content: review description
- + Keyword (list <keyword>): keyword used to find the content review
- + Review_score (integer): Review scores
- + Review_reliability (integer): Reviews are the reliability of the information, figures
- + Reference (item_id: integer): a product ID

4. Point: the point object of the user

A. attributes

- + Review_id (integer): Reviews generation ID information when reviewing a particular product
- + Review_point (integer): Electronic money amount information received in the review of certain goods
- + Review_reliability (integer): Reviews reliability of the information, figures

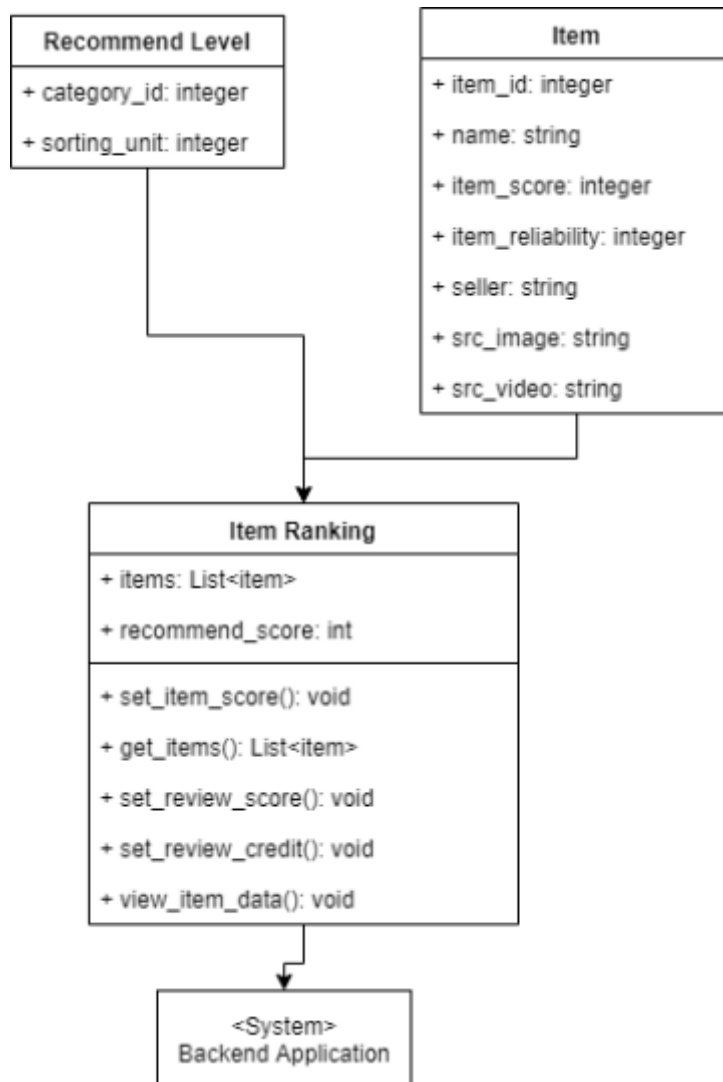
5. Bookmark: the bookmark of the item

A. attributes

- + Bookmark_number: Bookmark number
- + Item_id: item ID
- + User_id: user ID
- + Description: description of the bookmark

4.2.2. Item Ranking

1. Class Diagram



Drawing 17 System Architecture - Frontend - Ranking

1. Item Ranking - Ranking page objects

A. attributes

+ items: List of items that are displayed in the Ranking Page:

+ Recommend_score: reference point that is used to view a list of items in the Ranking page

B. methods

+ Set_item_score (criteria: score): Set the search conditions of the product.

+ Get_items (criteria: score): Bring the list of items corresponding to the search criteria in the Backend Database.

+ Set_review_score (item_id: integer): Enter the review scores of certain goods.

+ Set_review_credit (item_id: integer): Enter the reliability of the information entered into product reviews.

+ View_item_data (item_id: integer): Retrieves the contents of specific product information.

2. Recommend Level - object search conditions

A. attributes

+ Category_id: Category ID to identify an item

+ Sorting_unit: Sorting order (Ascending, Descending)

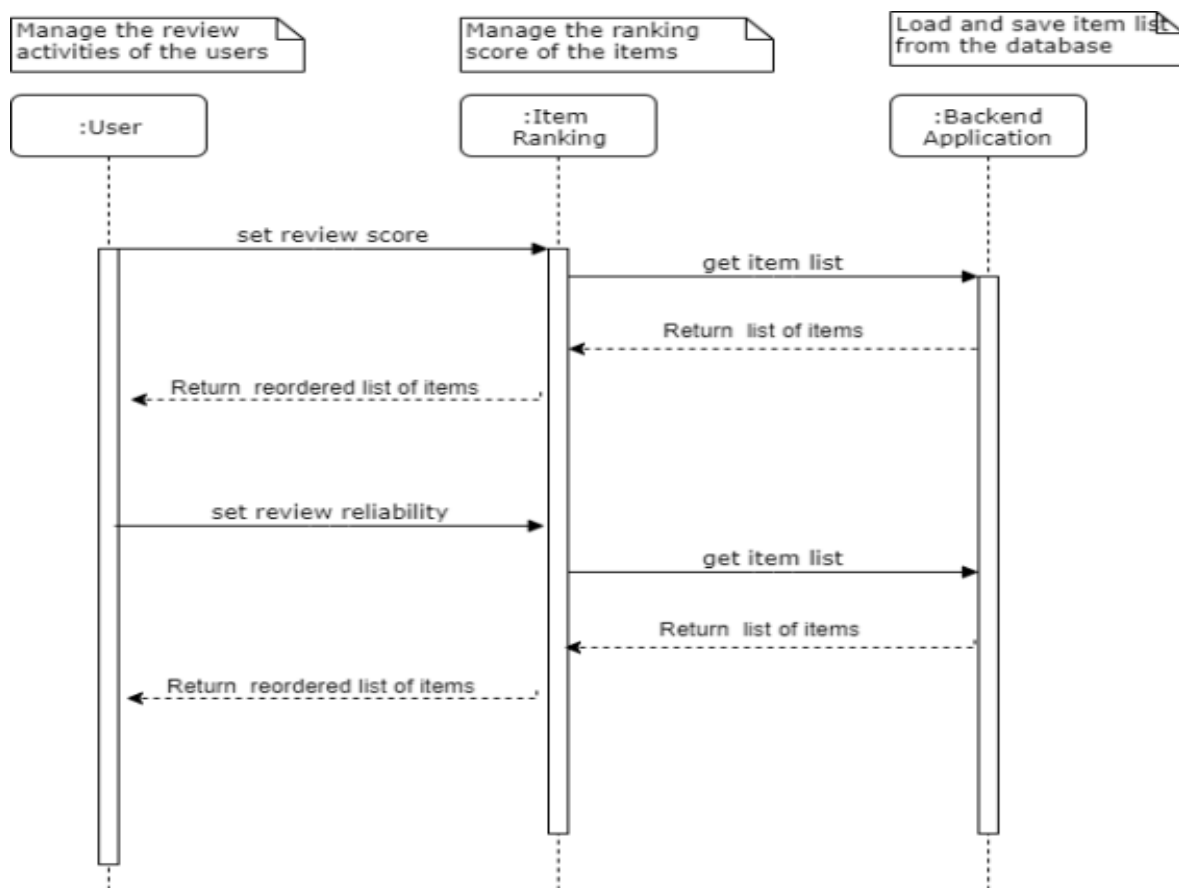
3. Item - Item Object (It refers to the unique name of the item as an object.)

A. attributes

+ Item_id: item ID

- + Name: item name
- + Item_score: recommendation score of the item
- + Item_reliability: the reliability score of the item
- + Seller: seller name (or company name)
- + Src_image: image address of the item
- + Src_video: video address of the item

2. Sequence Diagram



Drawing 18 System Architecture - Frontend - Ranking - Sequence Diagram.

The above sequence diagram shows a structure in which interaction between the user, item ranking, backend application in order. The user sequentially sets the reliability of the score information of the review and review score. This information is stored in a database via a backend application. The contents of the database are the sort order of the goods list. It updates based on the review score and review reliability.

4.2.3. Item Search



Drawing 19 System Architecture - Frontend- Search

1. SearchItem -Item object of the search menu

A. attributes

- + Current_keyword: Current search criteria
- + Search_log: the user's search history
- + Search_result: Product search results list according to the search criteria

B. methods

- + Search_item (criteria: SearchCriteria): Search for products by their search criteria.
- + Search_item (criteria: SearchCriteria, history_id: integer): Search for a specific product.
- + Add_search_history (history_id: integer): Add a specific search history.
- + Delete_search_history (history_id: integer): Delete specific search history.
- + Query_search_history (history_id string): retrieves the item of stored searches.

2. Certificated User - User Authentication object

A. methods

- + Get_user_id (): bring your ID information.

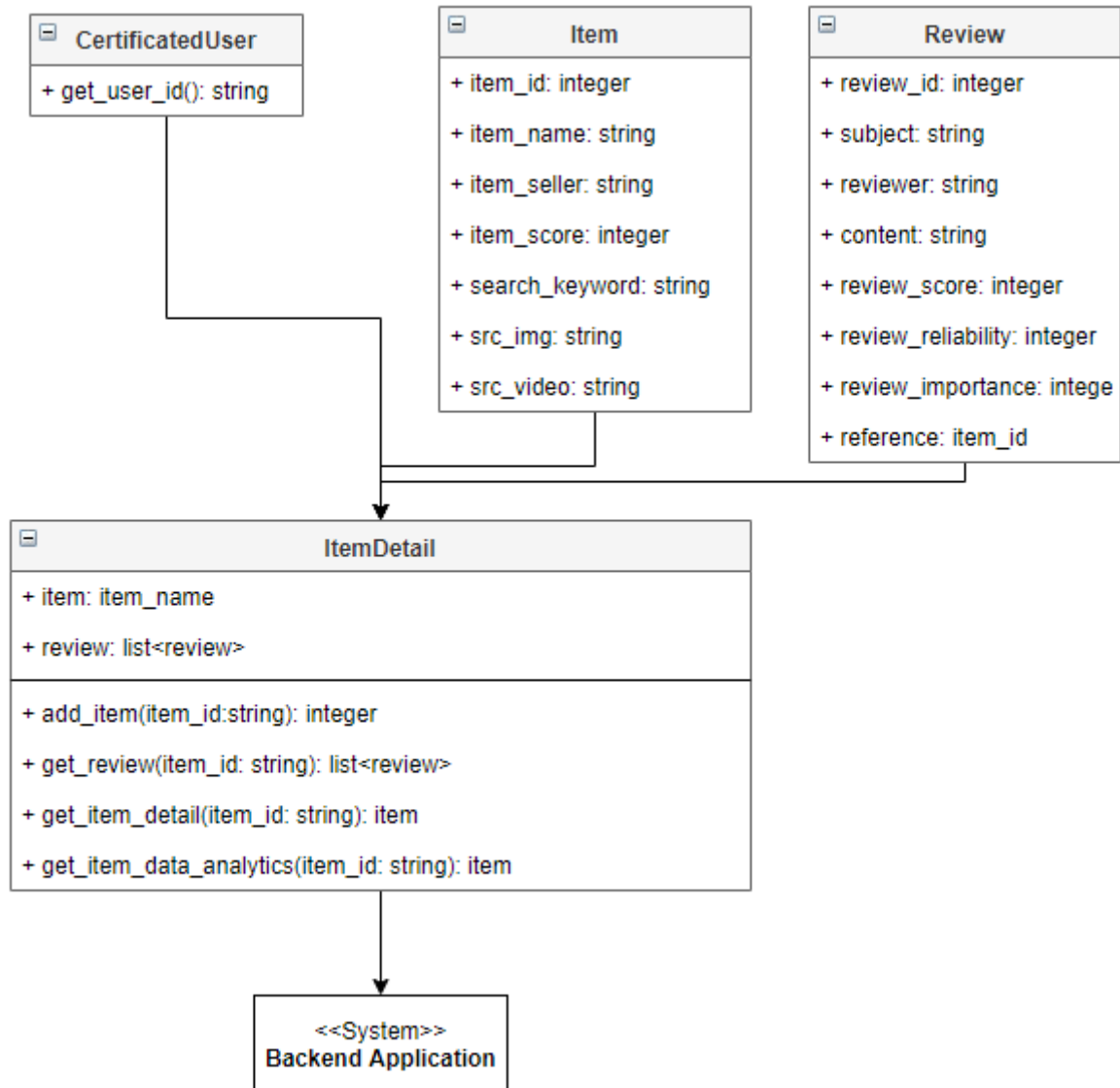
3. SearchFacility - criteria object (DTO)

A. Attributes

- + History_id: criteria for the record ID
- + History_description: Record the description of the retrieved information.
- + Search_keyword: keywords written in the search criteria
- + Category_id: Search criteria Categories

4.2.4. Item Detail

1. Class Diagram



Drawing 20 System Architecture - Frontend - Item Detail

1. ItemDetail - Detail page object to display item information

A. attributes

- + Item: item object of the current query
- + Reviews: item related review list on the current product

B. methods

- + Add_item (item_id: String): Add the item to your list.
- + Get_review (item_id: string): Retrieves the review of information products.
- + Get_item_detail (item_id: string): to display the details of this product.
- + Get_item_data_analytics (item_id: string): The query analyzing the data in the product information using Google's data analysis platform.

2. Certificated User - the object of the certificated user

A. methods

- + Get_user_id (): bring your ID information to purchase goods.

3. Item - item object

- Is the same as the definition of the product object Item Ranking.

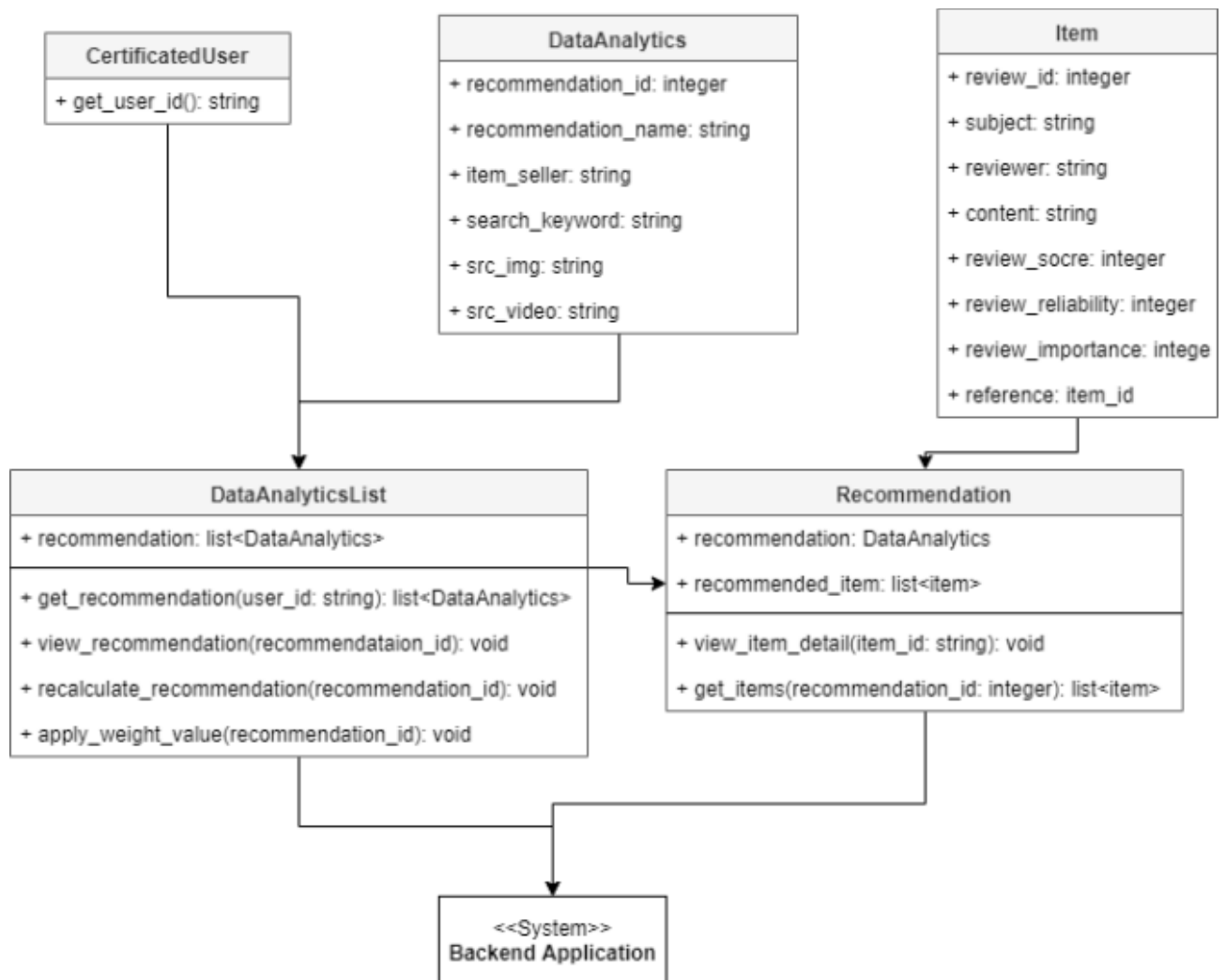
4. Review - the review object of the item

A. attributes

- + Review_id: the review ID
- + Subject: the review title
- + Reviewer: who that write the review on the item
- + Content: review content
- + Review_score: the review score on the item (1 to 100)
- + Review_reliability: the review reliability score (1 to 100)

- + Review_importance: the importance ratio of the review content (1% to 100%)
- + Reference: the item ID that is mapped by the review ID

4.2.5. Recommendation



Drawing 21 System Architecture - Frontend - Recommendation

1. DataAnalyticsList - Data Analytics List object to do item recommendation

A. attributes

- + Recommendations: the recommendation item list with google data analytics framework

B. methods

- + Get_recommendation (user_id: String): to obtain a list of recommended categories as user information-based functions
- + View_recommendation (recommendation_id: Integer): The function to check the details of a specific recommendation category. This function moves on to run Page Recommendation.
- + Recalculate_recommendation (recommendation_id): calculated each time the user enters a review rating information based on the importance of re-review.
- + Apply_weight_value (recommendation_id): Apply a better place featured a lot of action points to the user reviews.

2. DataAnalytics - The object of recommendation page according to Data Analytics framework

A. attributes

- + Recommendation_id: ID of the recommended category
- + Recommendation_name: recommended category names
- + Item_seller: item sellername
- + Search_keywords: keywords included in the recommended category
- + Src_img: recommended categories Thumbnail address
- + Src_video: Featured Video Thumbnail Category Address

3. Certificated User - The object of the certificated user

A. Attribute

- + Get_user_id (): User ID on the recommended item

4. Recommendation - The object of the recommended category

A. attributes

- + Recommendation_id: ID of the recommended category

- + Recommended_item: The recommended item list

B. methods

- + View_item_detail (item_id: Integer): the detail information after querying the item

- + Get_items (recommendation_id: Integer): query the item

5. Item - the item object

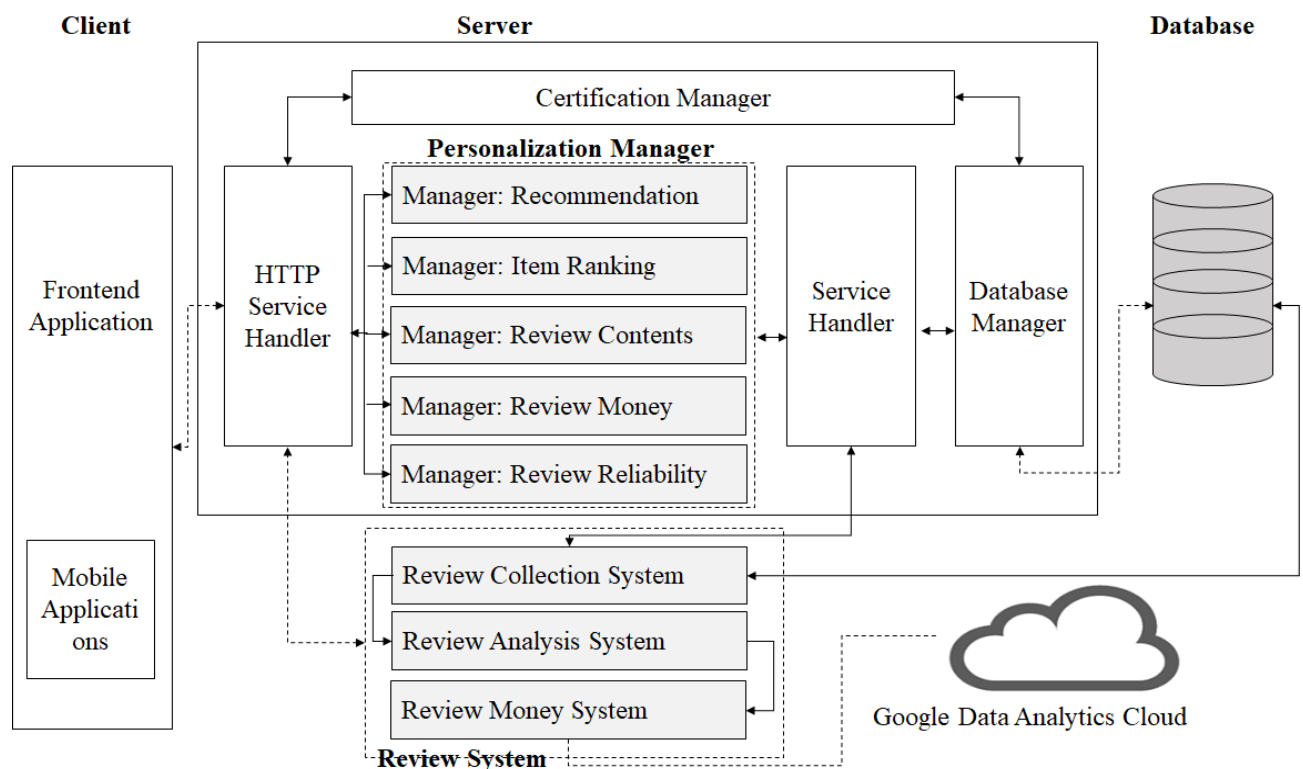
- It is equal to the definition of the item ranking object.

5. System Architecture - Backend

5.1. Objectives

In this section, we detail the Backend System that handles the request of Fronted Application. We describe the structure of the Backend System and Sub-system.

5.2. Overall Architecture

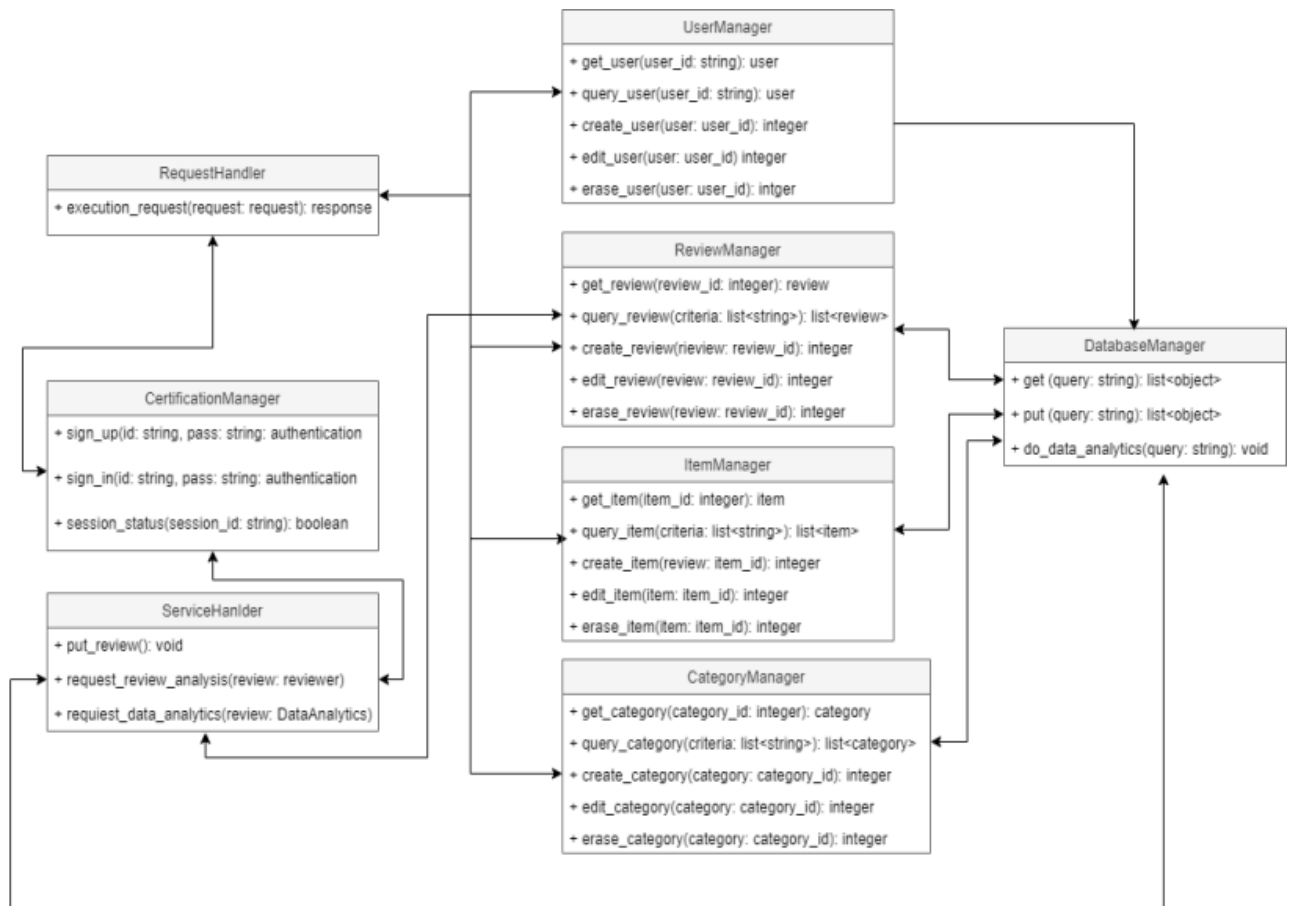


Drawing 22 System Architecture - Backend - Overall

The figure above shows the entire operating structure of the Backend System. The HTTP Service Handler is responsible for Frontend Application and direct communication. The Backend System handles and manages all requests for Frontend Application using a Personalization Manager and Review System. Personalization Manager performs such Recommendation, Item ranking, review contents, review point management, review reliability to manage a list of items. And Review System executes Review collection system, review analysis system, review point system, etc. To analyze the review information collected by using the remote Data Analytics Cloud.

5.3. Subcomponents

5.3.1. Server



Drawing 23 System Architecture - Backend - Application Server

1. Request Handler: the object to execute the request of the frontend application

A. methods

- + Execution_request (request: Request): function to distribute the incoming requests to the server to each Manager Task

2. Manager (Common activity)

A. attributes

- None

B. Methods

- + Get_entity (entity_id): to obtain a function entity
- + query_entity : Import function followed by a list of entities that match: (List <String> criteria) criteria
- + Create_entity (entity: Entity): a function of generating the entity
- + Edit_entity (entity: Entity): function to modify the entity
- + Erase_Entity (entity_id): a function for deleting the entity

3. Service Handler

A. Attributes:

- None

B. Methods

- + Put_review (): function that stores information Reviews
- + Request_review_analysis (review: reviewer): a function that runs the analysis on the review
- + Request_data_analytics (review: DataAnalytics): get the request and execution result of the review data analysis from Data Analytics Cloud function

4. Certification Manager

A. Attributes:

- None

B. Methods:

- + Sign_up (id: string, pass: String): Login to the ID / PW and the function of issuing an authentication token
- + Sign_in (token: String): from requiring authentication logic functions to determine whether the user has a valid login token
- + Session_status (session_id: String): It shows the current token-based login after loading information from the user,

5. Database Handler

A. Attributes:

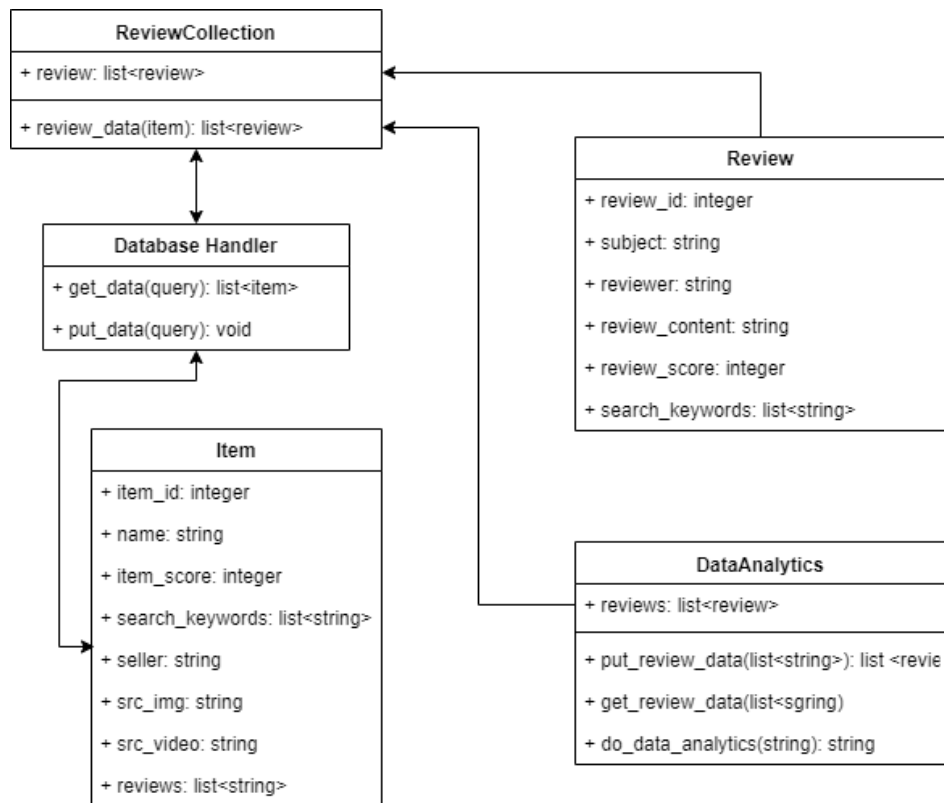
- None

B. Methods:

- + Get_data (query: String): It gets query results from the database.
- + Put_data (query: String): querying the database, enter the query string in the database.
- + Do_data_analytics (query: string): transfer the contents of a database to Google Data Analytics Cloud. It reports data analysis results using the deep learning function

5.3.2. Review Collection System

1. Class Diagram



Drawing 24 System Architecture - Backend - Review Collection System

1. Review Collection: the object to collect the review data from the reviewers
 - A. Attributes:
 - + Review: the review data
 - B. methods
 - + Review_data (item): a function that collects information about a particular item reviews

2. Database Handler: the object for database handler

A. methods

- + Get_data (query): function for importing data from a specific database
- + Put_data (query): a function for storing information in the database will review a specific item

3. item: the object of the item

A. Attributes:

- + Item_id: the item ID
- + Name: the item name
- + Item_score: the score of the item
- + Search_keywords: a keyword to search an item
- + Seller: the company that sells the item
- + Src_image: address of the image on an item
- + Src_ivideo: address of the video on an item
- + Reviews: the review list on the item

4. Review:

A. Attributes:

- + Review_id: the review id of the item
- + Subject: the title of the review
- + Reviewer: a writer that posts the review comments
- + Review_content: the description of the review
- + Review_score: the review score (1 to 100)
- + Search_keywords: a keyword to search the reviews

5. DataAnalytics:

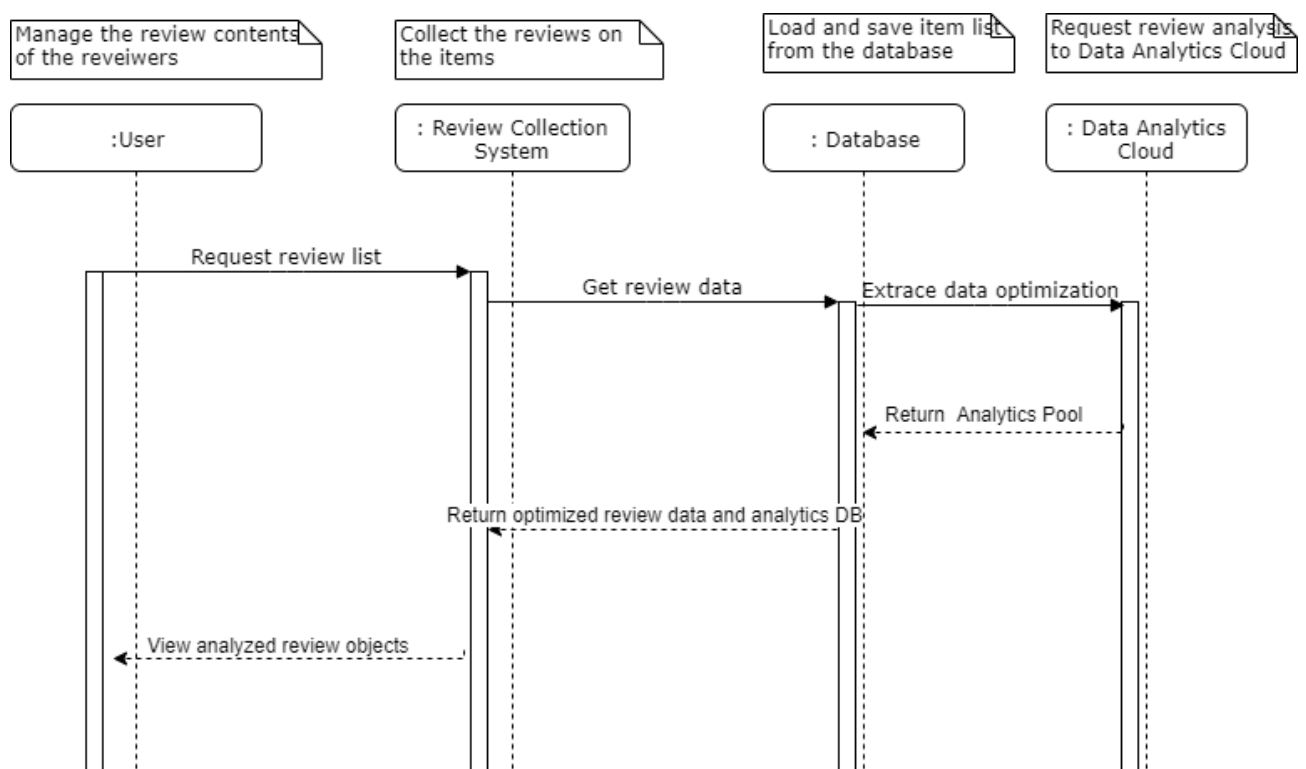
A. Attributes:

- + Reviews: the review list that is executed by Google data analytics cloud

B. Methods:

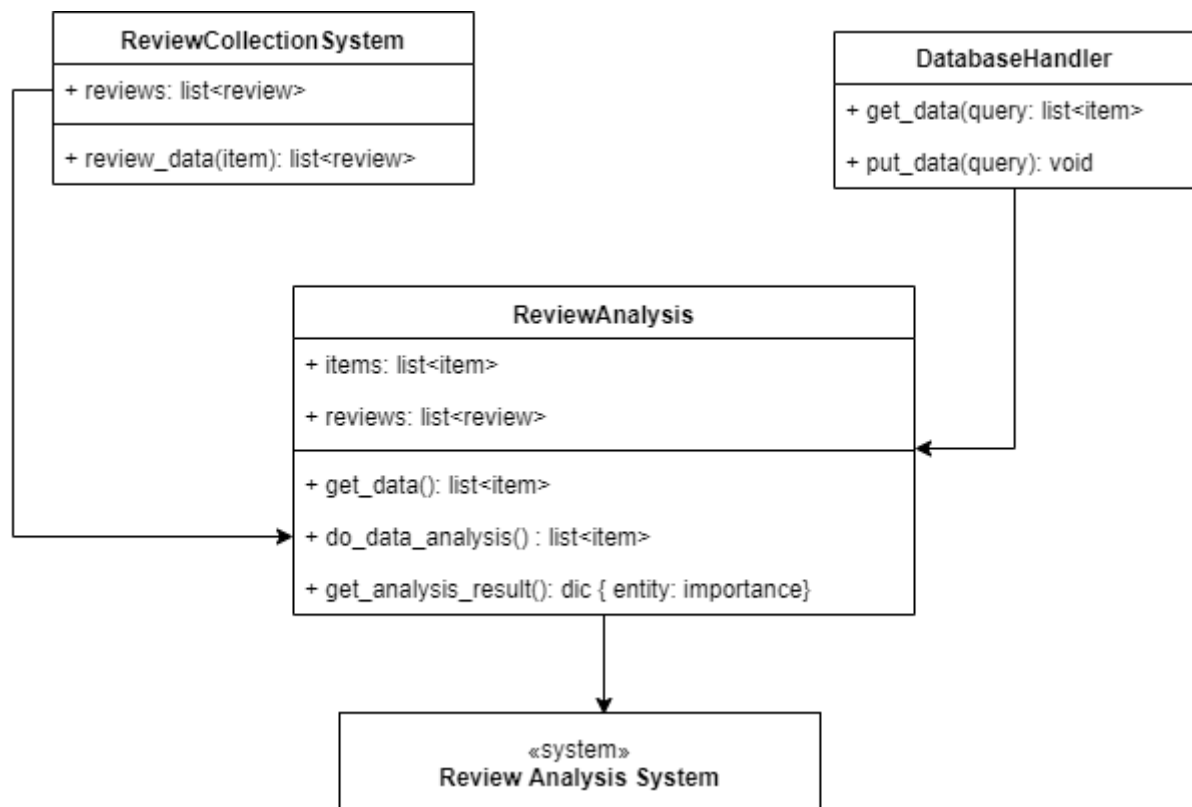
- + Put_review_data: a function for storing data analysis review
- + Get_review_data: function for importing analyzed data information Reviews
- + Do_data_analytics: function to analyze the data using the collected data. It gets the optimization result of deep learning.

2. Sequence Diagram



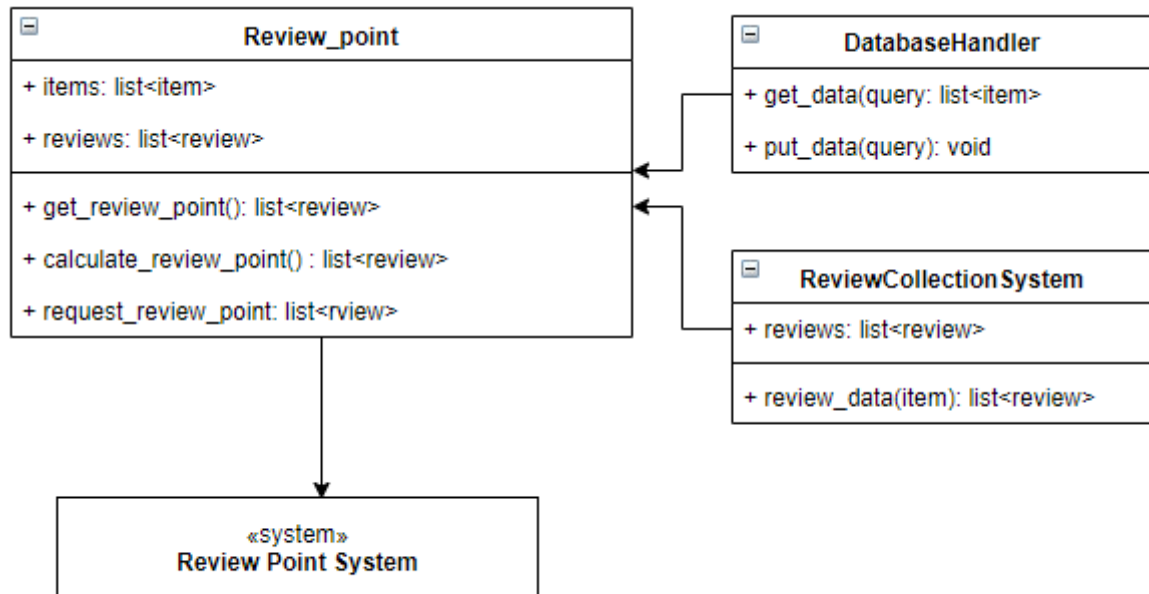
Drawing 25 System Architecture - Backend - Review Collection System - Sequence Diagram

5.3.3. Review Analysis System



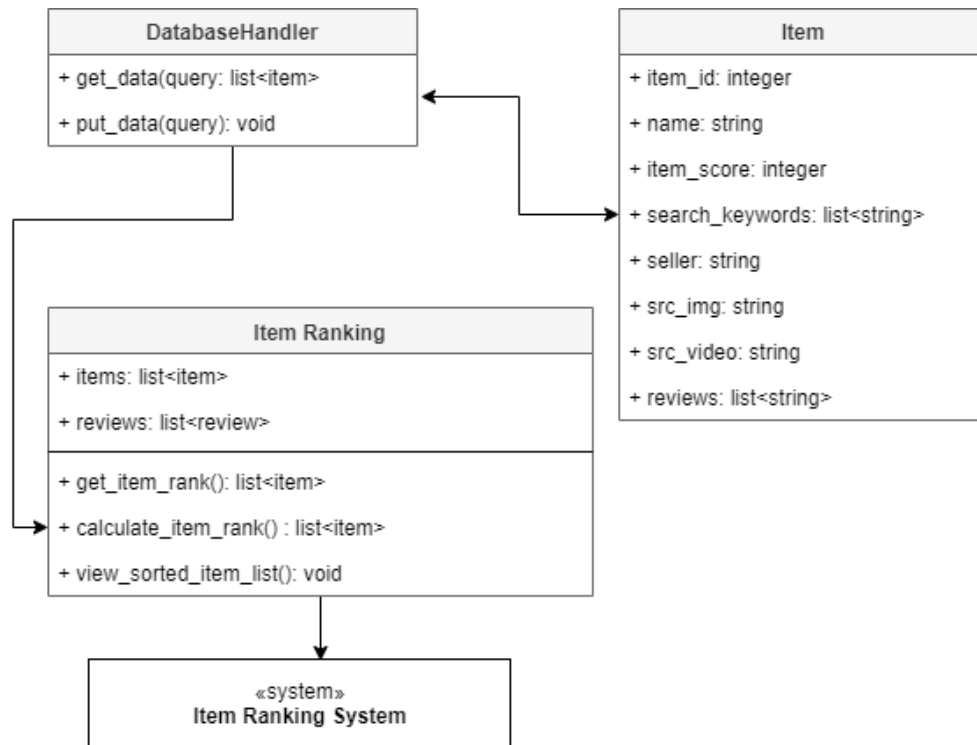
Drawing 26 System Architecture - Backend - Review Analysis System

5.3.4. Review Point System



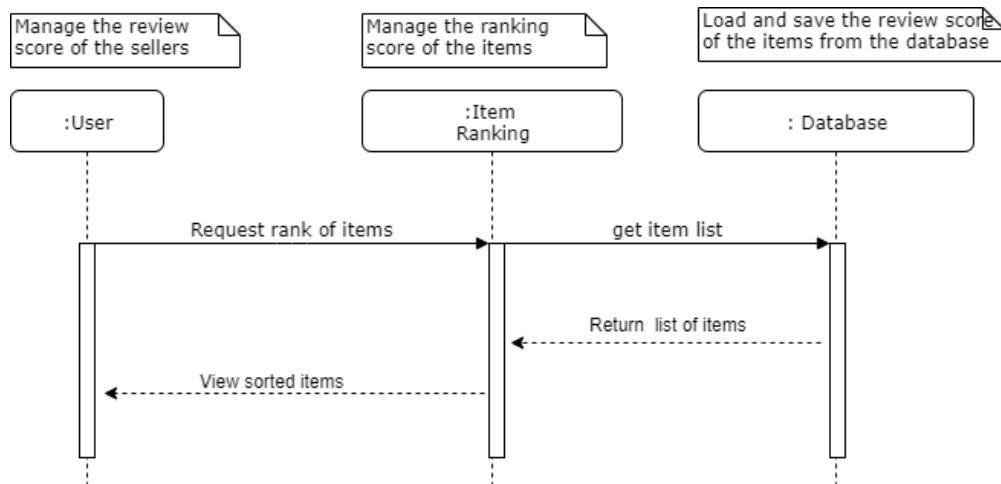
Drawing 27 System Architecture - Backend - Review Point System

5.3.5. Item Ranking System



Drawing 28 System Architecture - Backend - Item Ranking System.

Design Specification



Drawing 29 System Architecture - Backend - Item Ranking System - Sequence Diagram

6. Database Design

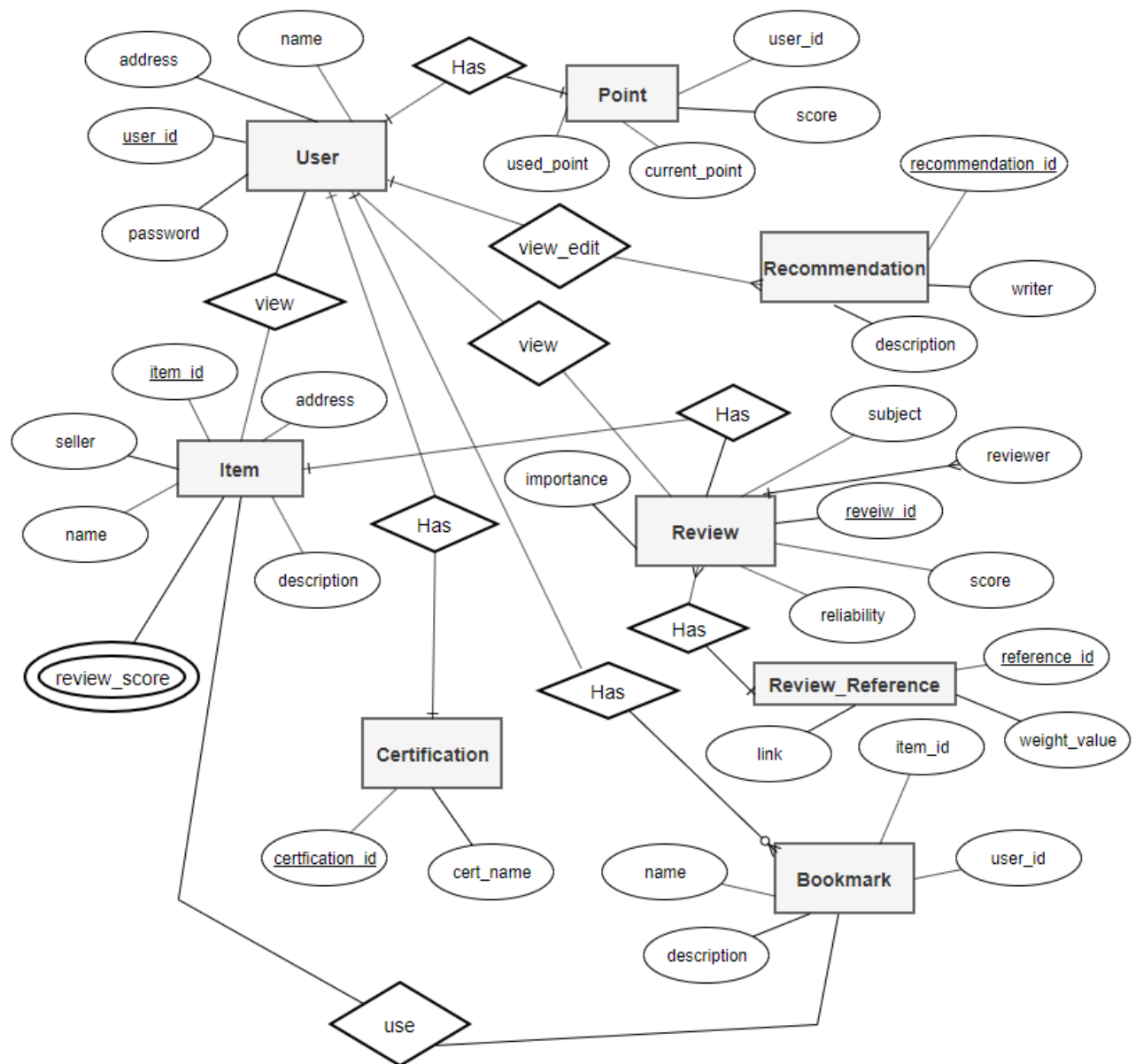
6.1. Objectives

In this section, we depict in detail the database schema for the Fitts system. The database design is based on data analyzed from the system requirements (Milestone2). We are using ER diagrams to represent the attributes of the data as described in a picture. We also describe the SQL Data Definition Language (DDL) and a relational database schema.

6.2. ER Diagram

Fitts online shopping system, there are eight entities of such User, Item, Bookmark, Certification, Recommendation, Review, Review Reference, Point, Data Analytics. We express the ER diagram in the following way.

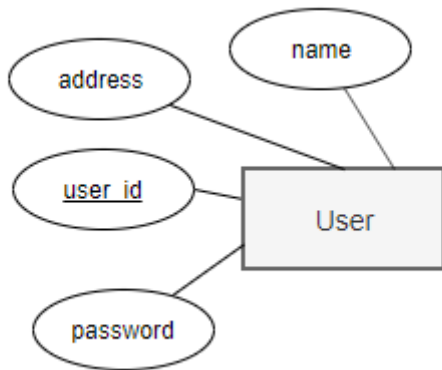
- Each entity represents a square box.
- A diamond-shape represents the relationship between the entities.
- In case a particular Entity, the plurality of relationships with other Entities is expressed by a line of the Entity Arrow shape.
- If an entity is not a particular Entity, we represent a small round mark side of the client entity.
- A unique key that identifies each Entity is described by an underlines mark.
- Besides, if the entity allows multiple attributes, It is expressed by drawing a border with a double.



Drawing 30 Overall flow of ER diagram

The attribute and the configuration of each Entity are as follows:

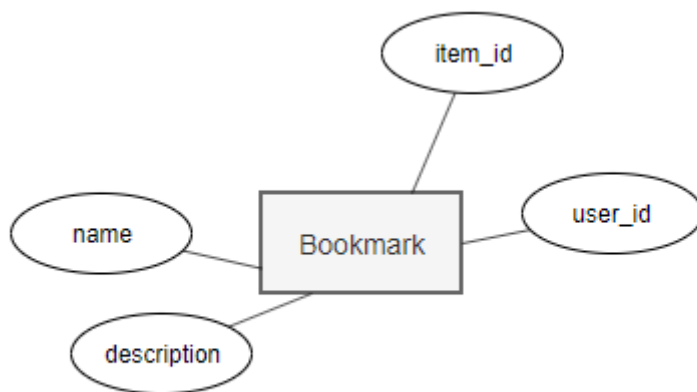
6.2.1. User Entity



Drawing 31 ER diagram of entity user.

User Entity represents your information. The user_id attribute is a primary key. And User entity has a username, password, address data.

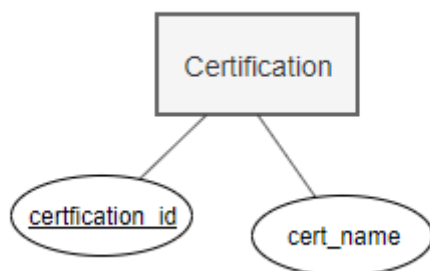
6.2.2. Bookmark Entity



Drawing 32 ER diagram of entity bookmark

Bookmark Entity represents the information that each user has set for the Item. The primary key does not exist. Bookmark Entity finds a bookmark with a combination of a composite key.

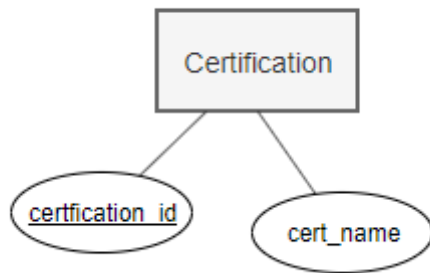
6.2.3. Certification Entity



Drawing 33 ER diagram of entity certification

Certification Entity represents a credential that each user can have. Certification Entity's primary key is the certification_id, which is the name attribute as the name of the authority.

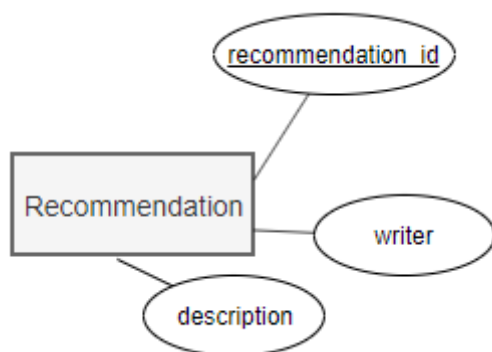
6.2.4. Item Entity



Drawing 34 ER diagram of entity item

Item Entity has information about each product. The Item Entity's primary key is the item_id. In addition to, Item Entity has properties such as name, company, specs, links, review scores. review_scores can have multiple values.

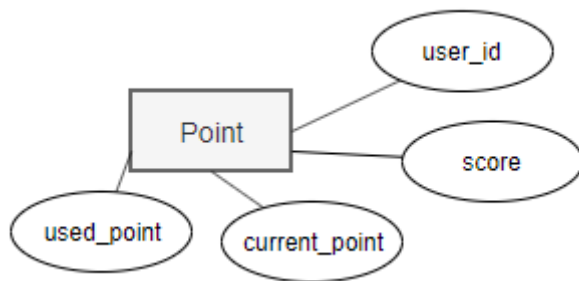
6.2.5. Recommendation Entity



Drawing 35 ER diagram of Recommendation entity

Recommendation Entity has the attribute information about the recommended categories. The primary key of Recommendation Entity is recommendation_id, and other properties to have a writer, description.

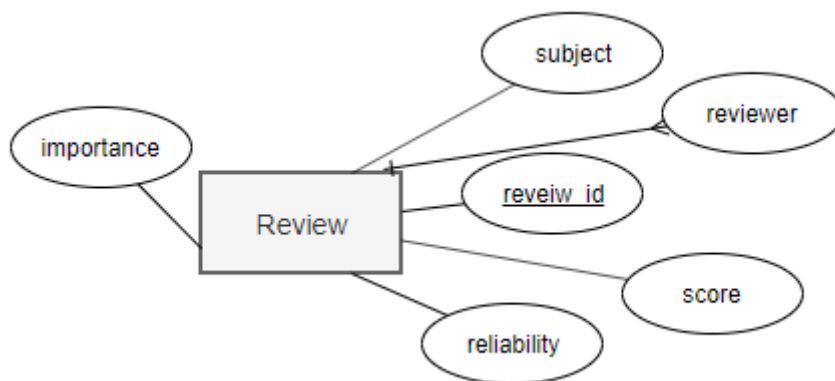
6.2.6. Point Entity



Drawing 36 ER diagram of entity recommendation

Recommendation Entity has information on keywords that describe each product. The keyword of Recommendation Entity has not only duplicates the name attribute.

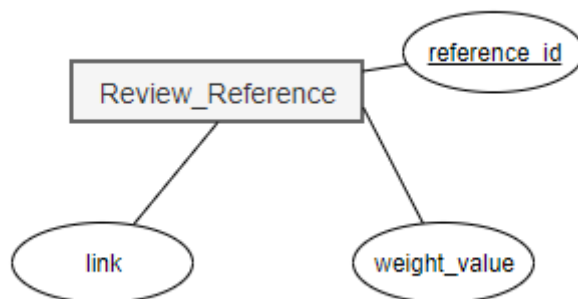
6.2.7. Review Entity



Drawing 37 ER diagram of entity review

Review Entity will present information about the review. Review Entity's the primary key is review_id. Other properties include a subject, reviewer, reference, importance, reliability, score.

6.2.8. Review Reference Entity

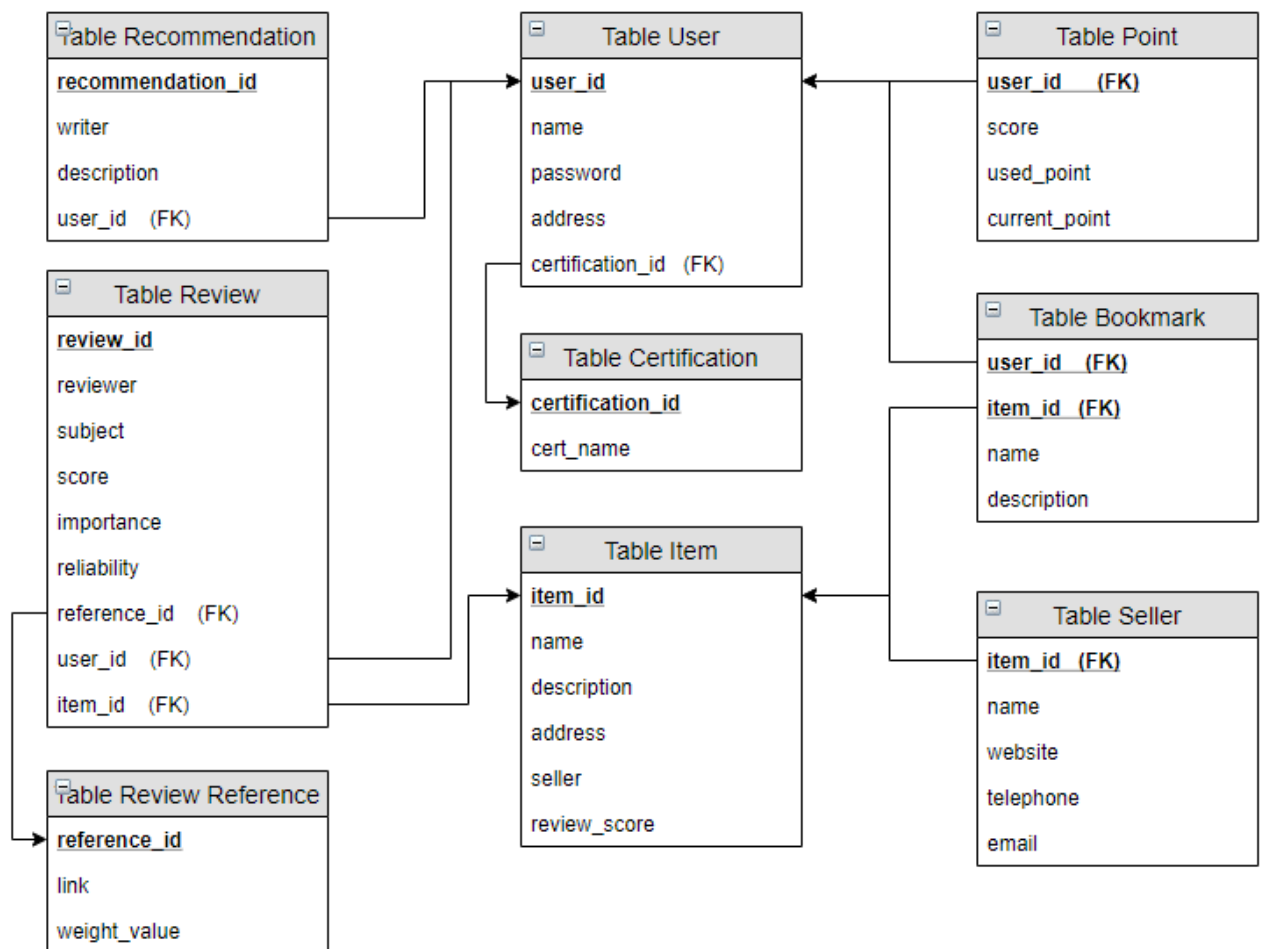


Drawing 38 ER diagram of entity review_reference

Review Reference Entity represents information about the origin of the goods written a review. The primary key of Review Reference Entity is reference_id. There are other properties such as the weight_valu and, link.

B. Relations

6.3. Relational DB Schema



Drawing 39 Relation DB Schema of Fitts system.

7.4. SQL DDL

6.3.1. Item

```
CREATE TABLE Item (  
  item_id INT NOT NULL,  
  name VARCHAR (50) NOT NULL,  
  description VARCHAR (20) NOT NULL,  
  address VARCHAR (20) NOT NULL,  
  seller VARCHAR (100) NOT NULL,  
  review_score INT NOT NULL,  
  src_image VARCHAR (100) NOT NULL,  
  src_video VARCHAR (100) NOT NULL,  
  PRIMARY KEY (item_id)  
);
```

6.3.2. User

```
CREATE TABLE User  
(  
  user_id INT NOT NULL,  
  name VARCHAR (10) NOT NULL,
```

```
password PASSWORD NOT NULL,  
address VARCHAR (40) NOT NULL,  
certification_id INT NOT NULL,  
PRIMARY KEY (user_id),  
FOREIGN KEY (certification_id) REFERENCES Authority (certification_id));
```

6.3.3. Certification

```
CREATE TABLE Certification  
(  
certification_id INT NOT NULL,  
cert_name VARCHAR (20) NOT NULL,  
PRIMARY KEY (certification_id)  
);
```

6.3.4. Point

```
CREATE TABLE Point  
(  
user_id VARCHAR (20) NOT NULL,
```

```
score INT NOT NULL,  
  
used_point INT NOT NULL,  
  
current_point INT NOT NULL,  
  
    FOREIGN KEY (user_id) REFERENCES user (user_id)  
  
);
```

6.3.5. Bookmark

```
CREATE TABLE Bookmark  
  
(  
  
    user_id INT NOT NULL,  
  
    item_id INT NOT NULL,  
  
    name VARCHAR (20) NOT NULL,  
  
  
    description VARCHAR (40) NOT NULL,  
  
    PRIMARY KEY (item_id),  
  
    PRIMARY KEY (user_id),  
  
    FOREIGN KEY (user_id) REFERENCES User (user_id),  
  
    FOREIGN KEY (item_id) REFERENCES Item (item_id)  
  
);
```

6.3.6. Reference

```
reviewer VARCHAR (20) NOT NULL,  
  
CREATE TABLE Reference  
  
(  
  
reference_id INT NOT NULL,  
  
reviewer VARCHAR (20) NOT NULL,  
  
subject VARCHAR (20) NOT NULL,  
  
score INT NOT NULL,  
  
importance INT NOT NULL,  
  
reliability INT NOT NULL,  
  
user_id VARCHAR (20) NOT NULL,  
  
item_id VARCHAR (20) NOT NULL,  
  
PRIMARY KEY (reference_id),  
  
FOREIGN KEY (user_id) REFERENCES User (user_id),  
  
FOREIGN KEY (item_id) REFERENCES Item (item_id)  
  
);
```

6.3.7. Seller

```
CREATE TABLE Seller
(
  item_id INT NOT NULL,
  name VARCHAR (20) NOT NULL,
  website VARCHAR (50) NOT NULL,
  telephone VARCHAR (20) NOT NULL,
  email VARCHAR (20) NOT NULL,

  PRIMARY KEY (item_id),
  FOREIGN KEY (item_id) REFERENCES Item (item_id)
);
```

6.3.8. Review

```
CREATE TABLE Review
(
  review_id INT NOT NULL,
  reviewer VARCHAR (50) NOT NULL,
```

```
subject VARCHAR (10) NOT NULL,  
  
score INT NOT NULL,  
  
importance INT NOT NULL,  
  
reliability INT NOT NULL,  
  
reference_id INT NOT NULL,  
  
item_id INT NOT NULL,  
  
user_id INT NOT NULL,  
  
PRIMARY KEY (review_id),  
  
FOREIGN KEY (item_id) REFERENCES Item (item_id),  
  
FOREIGN KEY (reference_id) REFERENCES Reference (reference_id),  
  
FOREIGN KEY (user_id) REFERENCES User (user_id)  
  
);
```

6.3.9. Recommendation

```
CREATE TABLE Recommendation  
  
(  
  
recommendation_id INT NOT NULL,  
  
writer VARCHAR (20) NOT NULL,  
  
description VARCHAR (200) NOT NULL,  
  
user_id INT NOT NULL,
```


PRIMARY KEY (recomendation_id),

FOREIGN KEY (user_id) REFERENCES User (user_id)

);

7. Protocol Design

7.1. Objectives

Fitts' system architecture consists of the Frontend System, which handles the user's interface, and the Backend System, which manages the user's requests. We also describe how interfaces and protocols are configured to communicate between frontend and backend systems. It also explains how the user interface is connected to the backend system.

7.2. XML



Drawing 40 XML Logo

XML is a markup language that defines a set of rules for encoding documents in a format that is both human-readable and machine-readable. The design goals of XML emphasize simplicity, generality, and usability across the Internet. It is a textual data format with strong support via

Unicode for different human languages. Although the design of XML focuses on documents, the language is widely used for the representation of arbitrary data structures, such as those used in web services.

We use XML technique to display meaningful data format by separating the data and the UI interface. It can efficiently express the list of items on the Web page.

7.3. CURL

curl is a command-line tool for transferring data and supports about 22 protocols including HTTP.

curl is a computer software project providing a library (libcurl) and a command-line tool (curl) for transferring data using various protocols. The name stands for "Client URL". The original author and lead developer is the Swedish developer Daniel Stenberg. This combination makes it an excellent ad-hoc tool for testing our REST services. curl supports over 200 command-line options. And we can have zero or more of them to accompany the URL in the command.

```

curl --user "APITest#API.User" --request GET https://secure.p03.eloqua.com/api/REST/1.0/data/contacts?count=2

{
  "elements": [
    {
      "type": "Contact",
      "id": "1",
      "createdAt": "1403034086",
      "depth": "minimal",
      "name": "george.washington@america.com",
      "updatedAt": "1410193024",
      "emailAddress": "george.washington@america.com"
    },
    {
      "type": "Contact",
      "id": "2",
      "createdAt": "1403113589",
      "depth": "minimal",
      "name": "john.a.macdonald@canada.com",
      "updatedAt": "1403113589",
      "emailAddress": "john.a.macdonald@canada.com"
    }
  ],
  "page": 1,
  "pageSize": 2,
  "total": 527
}

```

Drawing 41 CURL data example

Fitts online shopping system utilizes a web interface, HTTP for communications between Frontend and Backend, request and response format uses the CURL format. The CURL data are transferred and received in the (Representational State Transfer) REST API. This is to separate the resources that are stored on a server in the data format of CURL format means the design type of the API to send and receive data. REST API based on CURL shall be composed of resources, actions, represented as follows:

- Resources (Resource): URI, indicates the data that the server is stored, each resource has a unique URI.
- Act (Activity): HTTP Method, as requested action to the server to access a resource operation state, each operation action is expressed by the HTTP Method. (POST, GET, PUT, DELETE)
- Expression (Representation): It is mainly used by CURL format. If you use the REST API to communicate data between the client and server, you can easily access resources.

7.4. CURL API

7.4.1. Item

a. Get

- Request

Method	GET	
URI	/ Items /: id	
Parameters	-	-
Header	-	-

- Response

Success Code	200 OK	
Failure Code	404 Not Found (No Item corresponding to the id)	
Success Response Body	item	Item Object
Failure Response Body	-	-

b. Search

- Request

Design Specification

Method	GET	
URI	/ Items	
Parameters	id	Item ID
	name	Item Name
	company	Item manufacture company
	keywords	Item Keywords
	score	Item Score
Header	Authorization	User authentication token

- Response

Success Code	200 OK	
Failure Code	404 Not Found (Item no corresponding criteria)	
Success Response Body	items	Item Object List

7.4.2. User

a. Get

- Request

Method	GET	
URI	/ Users /: id	

Design Specification

Parameters	-	-
Header	Authorization	User authentication token

- Response

Success Code	200 OK	
Failure Code	404 Not Found (User does not have to correspond to the id)	
Success Response Body	user	User Object
Failure Response Body	-	-

b. Search

- Request

Method	GET	
URI	/ Users	
Parameters	id	User ID
	name	User Name
	authority	User Authority
Header	Authorization	User authentication token

- Response

Success Code	200 OK
--------------	--------

Design Specification

Failure Code	404 Not Found (User does not have to correspond to the search criteria)	
Success Response Body	users	User Object List
Failure Response Body	-	-

c. Add / Remove Bookmark

- Request

Method	POST / DELETE	
URI	/ Users /: id / bookmarks	
Parameters	item_id	Item ID
Header	Authorization	User authentication token

- Response

Success Code	200 OK	
Failure Code	400 Bad Request (Im already registered / not bookmark)	
Success Response Body	-	-
Failure Response Body	-	-

7.4.3. Certification

a. Sign-up

- Request

Method	POST	
URI	/ Certification / signup (REST exception)	
Request Body	id	User ID
	name	username
	password	User password

- Response

Success Code	200 OK	
Failure Code	400 Bad Request (if the ID duplicate)	
Success Response Body	-	-
Failure Response Body	message	reason for failure

b. Sign-in

- Request

Method	POST	
URI	/ Certification / login (REST exception)	

Design Specification

Request Body	id	User ID
	password	User password

- Response

Success Code	200 OK	
Failure Code	400 Bad Request (ID / PW, if an inconsistency)	
Success Response Body	success	true
Failure Response Body	success	false
	message	reason of failure

8. Appendix: Terminology

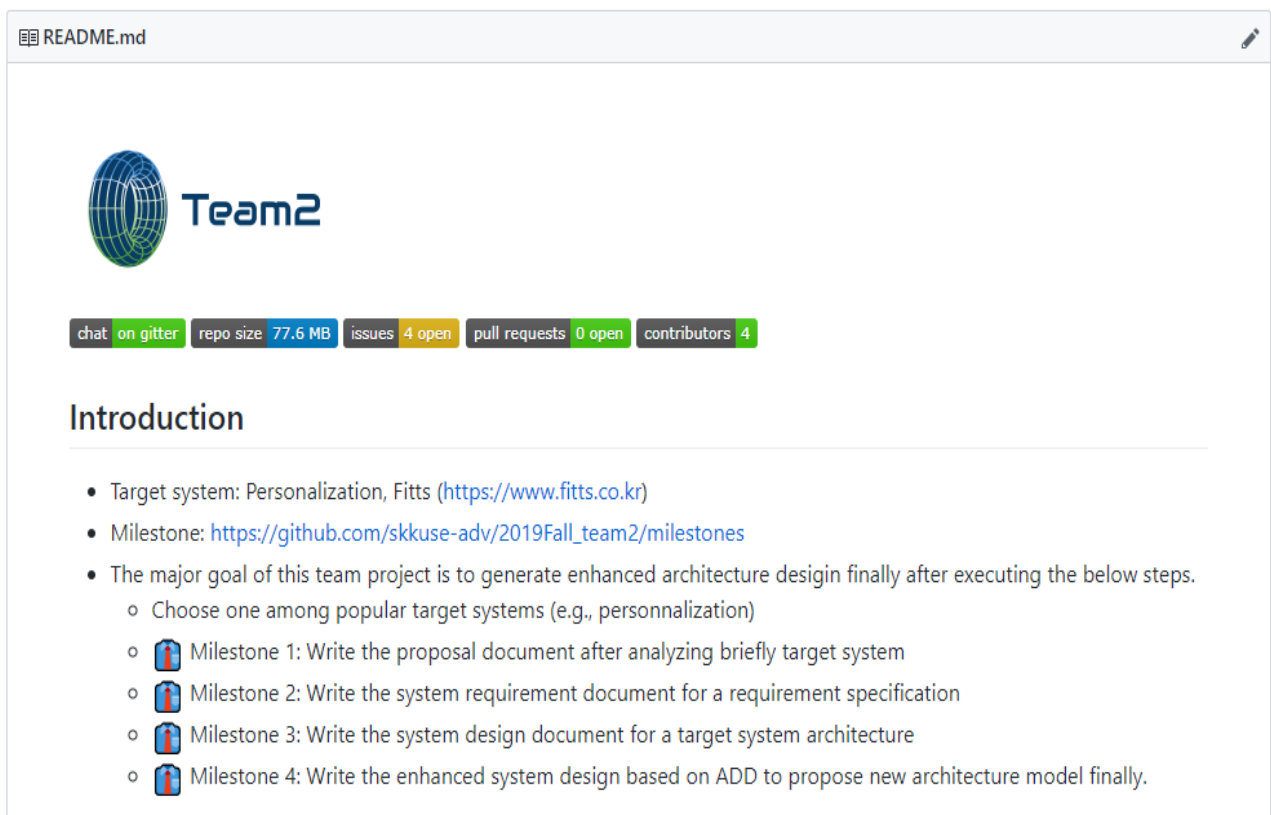
This section describes the meaning of the abbreviation words as following:

- DTO: Data Transfer Object
- DAO: Data Access Object
- ER: Entity-Relation
- XML: Extensible Markup Language
- DB: Database
- DDL: DB Definition Language

9. Appendix GitHub Repository

We archive the JAVA source code files that are decompiled by reverse-engineer approaches. All data are available at the GitHub Repository as following:

- https://github.com/skkuse-adv/2019Fall_team2



Drawing 42 Github Repository of Team2