



*Fitts*

내게 딱 맞는 핏을 찾다



# Fitts: Requirement Specification (Purpose: Reverse engineering)

Documented by:



**Team2**

Nguyen Khanh Binh	2019711305
Lim Gunsik	2019711699
Haziq Hamzah	2019711435
Jang Hanbin	

# Table of Contents

<b>1. Introduction .....</b>	<b>4</b>
<b>1.1. Background of Personalization Application.....</b>	<b>4</b>
<b>1.2. Background of Fitts Application.....</b>	<b>5</b>
<b>2. User Requirements Definition.....</b>	<b>5</b>
<b>2.1. Functional Requirements .....</b>	<b>5</b>
<b>2.1.1. Sign-up/Sign-in .....</b>	<b>5</b>
<b>2.1.2. User Personalization manager .....</b>	<b>6</b>
<b>2.1.3. Homepage.....</b>	<b>6</b>
<b>2.1.4. Store manager .....</b>	<b>6</b>
<b>2.1.5. Review System.....</b>	<b>7</b>
<b>2.1.6. Notification System .....</b>	<b>7</b>
<b>2.1.7. Mypage .....</b>	<b>7</b>
<b>2.1.8. Cart.....</b>	<b>7</b>
<b>2.1.9. Payment System .....</b>	<b>7</b>
<b>2.1.10. Showroom.....</b>	<b>8</b>
<b>2.1.11. Fitts Point System .....</b>	<b>8</b>
<b>2.1.12. Search.....</b>	<b>8</b>
<b>2.1.13. Log-out .....</b>	<b>8</b>
<b>2.1.14. DB manager .....</b>	<b>8</b>
<b>2.2. Non-Functional Requirements.....</b>	<b>8</b>
<b>2.2.1. Product Requirements.....</b>	<b>8</b>

2.2.2. Organizational Requirements .....	9
3. System Requirements Specification .....	10
3.1. Functional Requirements – Frontend .....	10
3.1.1. Sign-up .....	10
3.1.2. Sign-in .....	11
3.1.3. User Personalization Manager (Frontend).....	11
3.1.4. Homepage .....	12
3.1.5. Store Manager (Frontend) .....	13
3.1.6. Review System (Frontend) .....	13
3.1.7. Notification System .....	14
3.1.8. Mypage .....	15
3.1.9. Cart.....	15
3.1.10. Payment System (Frontend) .....	16
3.1.11. Showroom.....	17
3.1.12. Fitts Point System .....	17
3.1.13. Search.....	18
3.1.14. Sign-out.....	18
3.2. Functional Requirements – Backend.....	19
3.2.1. User Personalization Manager .....	19
3.2.2. Store Manager .....	20
3.2.3. Review System.....	20
3.2.4. Payment System .....	21

3.2.5. DB Manager .....	22
3.3. Non-Functional Requirements .....	22
3.3.1. Product Requirements .....	22
3.3.2. Organizational Requirements .....	22
3.3.3. External Requirements .....	22
4. System Architecture .....	24
4.1. Frontend Diagram .....	24
4.2. Backend Diagram .....	25
4.3. Use-case Diagram .....	26

# 1.Introduction

This section briefly describes the target system including the background of personalization application and about the Fitts application itself.

## 1.1. Background of Personalization Application

Nowadays, due to the enormous growth of the internet and technologies, it enables people to do many things through their mobile phone, laptop, or television. Online shopping malls appeared in the growth of both fashion companies and the internet. Mobile shopping mall application provides functions for users convenient shopping. Customers can check details of the product at their home, pay as they go, and receive the product at their chosen

place. The extent of this is when an application allowed users to personalize their user's data for better shopping experiences.

## **1.2. Background of Fitts Application**

While so many existing online shopping mall applications on the application store, they are all similar to each other. Thus, some company chooses to make a difference by implementing some minor changes. Fitts is one of those applications that is unique in the market. The key idea of Fitts's success is that they utilize the user's body information to suggest the products which follow the user's personalization. With these unique features, Fitts can increase its revenue.

## **2. User Requirements Definition**

This section covers the definitions for the user requirements of the proposed system. The requirements can be categorized into functional and non-functional requirements.

### **2.1. Functional Requirements**

The functional requirements include, but are not limited to, the following items:

#### **2.1.1. Sign-up/Sign-in**

This is a basic requirement for a user to use the Fitts application. The requirement is as such:

- At first, users must sign-up an account of the Fitts system with ID, password, and e-mail, or users can create an account via Facebook or Kakao account.
- Users must Sign-in with their ID and Password.

- When users Sign-in, the system should authenticate the users by matching the given input with the user's account information in the DB.

### **2.1.2. User Personalization manager**

#### **2.1.2.1. Frontend**

After logged-in, users must input the information about their bodies such as height, weight, body type, shoe size, and style of clothing.

#### **2.1.2.2. Backend**

The user personalization manager must store the information in system DB.

### **2.1.3. Homepage**

Show up the posted reviews of other users, which is about the clothes that they bought with a direct link to buy the product. Must include today's famous posting, daily best product deals, a trend that is right with my bodies information, and other famous product that is not related to my bodies information.

### **2.1.4. Store manager**

#### **2.1.4.1. Frontend**

Show up many products with the price available in the Fitts system and direct link to the shop that sells the products. Users can bookmark the product and the shop.

#### **2.1.4.2. Backend**

Responsible to handles user request related to store and link it with the DB manager.

### **2.1.5. Review System**

#### **2.1.5.1. Frontend**

Users can upload reviews about outfits, shoes, bags using their creativity.

#### **2.1.5.2. Backend**

Responsible to handles user request related to review and link it with the DB manager.

### **2.1.6. Notification System**

Users should receive any notice about events, coupons, best deals, payment completions, shipping information and many more.

### **2.1.7. Mypage**

Mypage provides users with information related to ordering and activity. Users can check their orders and shipping information. From mypage, users can go to their showroom to view their posted reviews. It also provides a bookmark feature, event notification feeds, and friend invitation.

### **2.1.8. Cart**

All of the products to be bought by users are listed in the cart.

### **2.1.9. Payment System**

#### **2.1.9.1. Frontend**

Payment of products is linked with various payment methods such as credit card, master card, Kakao pay, Naver pay, and so on.

#### **2.1.9.2. Backend**

Responsible to handles user request related to payment and link it with the DB manager..

#### **2.1.10. Showroom**

The reviewed product by the users must be shown here as a kind of user's private blog.

#### **2.1.11. Fitts Point System**

Fitts points can be viewed on mypage. This is the point earn from posting reviews about the product. Users can collect this point to buy coupons or products in Fitts store. Besides earn point from posting review, users can earn points by shopping, inviting peoples, and complete some events.

#### **2.1.12. Search**

Users can search for other users, shop, or product in this area.

#### **2.1.13. Log-out**

After users log-out, all application instances and services must be terminated with the user's device. Except for background service such as notification services.

#### **2.1.14. DB manager**

Responsible to update and store DB data according to the incoming request from other components.

## **2.2. Non-Functional Requirements**

The non-functional requirements can be further classified into the Product Requirements and Organizational Requirements as follows:

### **2.2.1. Product Requirements**

#### **2.2.1.1. Performance**

- It needs the same performance as other personalization applications when users are shopping for products.



- It also needs a good system to upload and download the image for products. It needs to balance the network to make it fast enough when users are using certain services.

#### **2.2.1.2. Security**

A review must be reliable and must guarantee the anonymity of the reviewer. Each personal information must be protected from external access. ID and passwords should also be protected by any third-party access.

#### **2.2.1.3. Usability**

The UI of the application should be simple to provide important information at the tip of a finger. Tutorial of using application must be given when the user uses the application for the first time.

#### **2.2.1.4. Safety**

The Fitts point must be stable and reliable throughout the application.

### **2.2.2. Organizational Requirements**

- Any updates must be conduct as a team, and must be discussed (Github or KakaoTalk).
- Documents created must follow the guidance of SE class.
- All diagrams drawn must follow the UML model.
- Attribute Driven Develepmont must be use when refining the target system

# 3.System Requirements Specification

This chapter specifies the functional and non-functional requirements. We focus on the following detail requirements: Name, description, inputs, outputs, action, requirements, pre-condition and post-condition.

## 3.1. Functional Requirements – Frontend

### 3.1.1. Sign-up

<b>Name</b>	Sign-up
<b>Description</b>	Users can create their account
<b>Inputs</b>	{ID, password, user details}
<b>Outputs</b>	The account of the user is created
<b>Action</b>	When the user first opens the application, it requires the user to Sign-in or if the user does not have an account, the user must sign-up. When signing-up, user must input their desired ID and password. They also are required to input details about them
<b>Requirements</b>	The person with no account
<b>Pre-condition</b>	Users should have email, Facebook or Kakao account
<b>Post-condition</b>	N/A

**Table 1:** Sign-up

### 3.1.2. Sign-in

<b>Name</b>	Sign-in
<b>Description</b>	User can Sign-in to their account after signing-up or already have an account
<b>Inputs</b>	{ID, password}
<b>Outputs</b>	Homepage
<b>Action</b>	The user Sign-in with the account that was signed-up previously. After user is logged in to the application, the first-page instance that the user view is homepage.
<b>Requirements</b>	User must have the credential in the application system
<b>Pre-condition</b>	N/A
<b>Post-condition</b>	N/A

**Table 2:** Sign-in

### 3.1.3. User Personalization Manager (Frontend)

<b>Name</b>	User personalization manager (frontend)
<b>Description</b>	Provide UI for the user to input the data related to user's body information
<b>Inputs</b>	{Body weight, height, shoe size, .....}

<b>Outputs</b>	User personalization manager sends the data to the DB manager for DB update
<b>Action</b>	The user opens the UI to input their body information. User input all the required inputs. The user personalization manager gets the input data and send to DB manager.
<b>Requirements</b>	User body information
<b>Pre-condition</b>	Input options must be provided by personalization manager
<b>Post-condition</b>	Input data must be sent to DB manager

**Table 3:** User Personalization Manager (Frontend)

#### 3.1.4. Homepage

<b>Name</b>	Homepage
<b>Description</b>	Provide view of available products, shops, daily deals, and trends
<b>Inputs</b>	User body information
<b>Outputs</b>	The data related to products, shops, daily deals, and trend which relates to the user body information
<b>Action</b>	When user access homepage, it will automatically provide view of products, shops, daily deals, and trends which relates to the user body information

<b>Requirements</b>	User body information
<b>Pre-condition</b>	N/A
<b>Post-condition</b>	N/A

**Table 4:** Homepage

### 3.1.5. Store Manager (Frontend)

<b>Name</b>	Store manager (frontend)
<b>Description</b>	Manage the view for products and shops
<b>Inputs</b>	User body information
<b>Outputs</b>	The data related to products and shops
<b>Action</b>	When user access the store page, store manager will automatically provide view of products and shops
<b>Requirements</b>	User body information
<b>Pre-condition</b>	N/A
<b>Post-condition</b>	N/A

**Table 5:** Store Manager (Frontend)

### 3.1.6. Review System (Frontend)

<b>Name</b>	Review system (frontend)
<b>Description</b>	Manage the page instances relates to reviewing a product
<b>Inputs</b>	Reviewed data of a product

<b>Outputs</b>	Fitts point gained from reviewing product
<b>Action</b>	When user review certain products, review system will automatically provide view of products and shops
<b>Requirements</b>	User must fill up all required data
<b>Pre-condition</b>	Check whether reviewed data is valid to the system standard
<b>Post-condition</b>	N/A

**Table 6:** Review System (Frontend)

### 3.1.7. Notification System

<b>Name</b>	Notification system
<b>Description</b>	Manage the notification of certain deals of products and some payment related notices
<b>Inputs</b>	Data to be notify
<b>Outputs</b>	User is notified with the data
<b>Action</b>	Whenever the application is running, user should be noticed with the deals about product and also must notify user for every successful payment
<b>Requirements</b>	Application must be running in the device environment
<b>Pre-condition</b>	N/A
<b>Post-condition</b>	N/A

**Table 7:** Notification system

### 3.1.8. Mypage

<b>Name</b>	Mypage
<b>Description</b>	Organizes user data and let user view it in one page. It also contains link to bookmark and showroom
<b>Inputs</b>	User related data
<b>Outputs</b>	View of organized data
<b>Action</b>	When user open the mypage, user can view their shopping history. They also can go to bookmark and showroom via mypage
<b>Requirements</b>	Data of user must be valid to that specific account
<b>Pre-condition</b>	Collect user data from DB
<b>Post-condition</b>	User data must be organized before view

**Table 8:** Mypage

### 3.1.9. Cart

<b>Name</b>	Cart
<b>Description</b>	Temporarily store the product to be bought by users
<b>Inputs</b>	Products to be bought
<b>Outputs</b>	Payment of the products
<b>Action</b>	When user is shopping, they will want to store some products in a list before buying them

<b>Requirements</b>	Cart must be empty after all product was payed
<b>Pre-condition</b>	Product that is stored in a cart depends on the availability of the product
<b>Post-condition</b>	N/A

**Table 9: Cart**

### 3.1.10.Payment System (Frontend)

<b>Name</b>	Payment system (frontend)
<b>Description</b>	Interface for user to pay for products
<b>Inputs</b>	User payment details
<b>Outputs</b>	Notification of success payment
<b>Action</b>	When user wants to pay for the product, payment system will provide interface for payment procedure. Payment system will link with notification system to notify user when payment is successful
<b>Requirements</b>	User must be notify for either succeed or failed payment
<b>Pre-condition</b>	Validate the payment details
<b>Post-condition</b>	N/A

**Table 10: Payment System (Frontend)**



### 3.1.11.Showroom

<b>Name</b>	Showroom
<b>Description</b>	This is the page that shows all of the posted review by the user
<b>Inputs</b>	History of reviewed data
<b>Outputs</b>	View for the posted reviews
<b>Action</b>	When user go into the showroom page, user can view the reviews that they posted
<b>Requirements</b>	Reviews must be only from the user
<b>Pre-condition</b>	N/A
<b>Post-condition</b>	N/A

**Table 11:** Showroom

### 3.1.12.Fitts Point System

<b>Name</b>	Fitts point system
<b>Description</b>	In app currency that can be consider as real money
<b>Inputs</b>	Token related to user's account
<b>Outputs</b>	Fitts point of the user
<b>Action</b>	Everytime user open pages that contains Fitts point, user must be able to view it
<b>Requirements</b>	Fitts point must be reliable and stable

<b>Pre-condition</b>	N/A
<b>Post-condition</b>	N/A

**Table 12:** Fitts Point System

### 3.1.13.Search

<b>Name</b>	Search
<b>Description</b>	Responsible to find the word that wants to be searched by users
<b>Inputs</b>	Word to be searched
<b>Outputs</b>	Contains related to the word
<b>Action</b>	When user search for any word (ex. Store or product name), the search result must be shown to the users
<b>Requirements</b>	Words that use for the search must only be in the app domain
<b>Pre-condition</b>	Filter words that is not in the app domain
<b>Post-condition</b>	Prioratize the search result according to the input words

**Table 13:** Search

### 3.1.14.Sign-out

<b>Name</b>	Sign-out
<b>Description</b>	End all session related to the user's account
<b>Inputs</b>	User's account data

<b>Outputs</b>	Terminte all services
<b>Action</b>	When user sign-out, the session of the user must be terminated
<b>Requirements</b>	User sessions must be terminated
<b>Pre-condition</b>	N/A
<b>Post-condition</b>	N/A

**Table 14:** Sign-out

## 3.2. Functional Requirements – Backend

### 3.2.1. User Personalization Manager

<b>Name</b>	User personalization manager (backend)
<b>Description</b>	Interfaces with DB manager, and store the user personalization data in the DB system.
<b>Inputs</b>	{Body weight, height, shoe size, .....}
<b>Outputs</b>	User personalization manager sends the data to the DB manager for DB update
<b>Action</b>	The user opens the UI to input their body information. User input all the required inputs. The user personalization manager gets the input data and send to DB manager.
<b>Requirements</b>	User body information

<b>Pre-condition</b>	Input options must be provided by personalization manager
<b>Post-condition</b>	Input data must be sent to DB manager

**Table 15:** User Personalization Manager

### 3.2.2. Store Manager

<b>Name</b>	Store manager (backend)
<b>Description</b>	Manage the interfaces between frontend views and DB manager
<b>Inputs</b>	User body information
<b>Outputs</b>	The data related to products and shops
<b>Action</b>	When user access the store page, store manager will automatically provide view of products and shops
<b>Requirements</b>	User body information
<b>Pre-condition</b>	N/A
<b>Post-condition</b>	N/A

**Table 16:** Store Manager

### 3.2.3. Review System

<b>Name</b>	Review system (backend)
<b>Description</b>	Manage the interface between frontend views with DB manager

<b>Inputs</b>	Reviewed data of a product
<b>Outputs</b>	Fitts point gained from reviewing product
<b>Action</b>	When user review certain products, review system will automatically provide view of products and shops
<b>Requirements</b>	User must fill up all required data
<b>Pre-condition</b>	Check whether reviewed data is valid to the system standard
<b>Post-condition</b>	N/A

**Table 17:** Review System

#### 3.2.4. Payment System

<b>Name</b>	Payment system (backend)
<b>Description</b>	Background validation service to ensure payment of products is done in a right way
<b>Inputs</b>	User payment details
<b>Outputs</b>	Notification of success payment
<b>Action</b>	When user wants to pay for the product, backedn payment system will provide the interface required
<b>Requirements</b>	User must be notify for either succeed or failed payment
<b>Pre-condition</b>	Validate the payment details
<b>Post-condition</b>	N/A

**Table 18:** Payment System

### 3.2.5. DB Manager

<b>Name</b>	DB manager
<b>Description</b>	Update and store DB data according to the incoming request from other components
<b>Inputs</b>	Query commands
<b>Outputs</b>	Query output
<b>Action</b>	DB manager handles all request going in and out from the DB system
<b>Requirements</b>	Query data must be validated by DB manager
<b>Pre-condition</b>	N/A
<b>Post-condition</b>	N/A

**Table 19:** DB Manager

## 3.3. Non-Functional Requirements

### 3.3.1. Product Requirements

Refer to 2.2.1.

### 3.3.2. Organizational Requirements

Refer to 2.2.2.

### 3.3.3. External Requirements

#### 3.3.3.1. Personal Information Protection

The system collects and stores user information when they sign-up. Only some basic information of users must be gathered, and

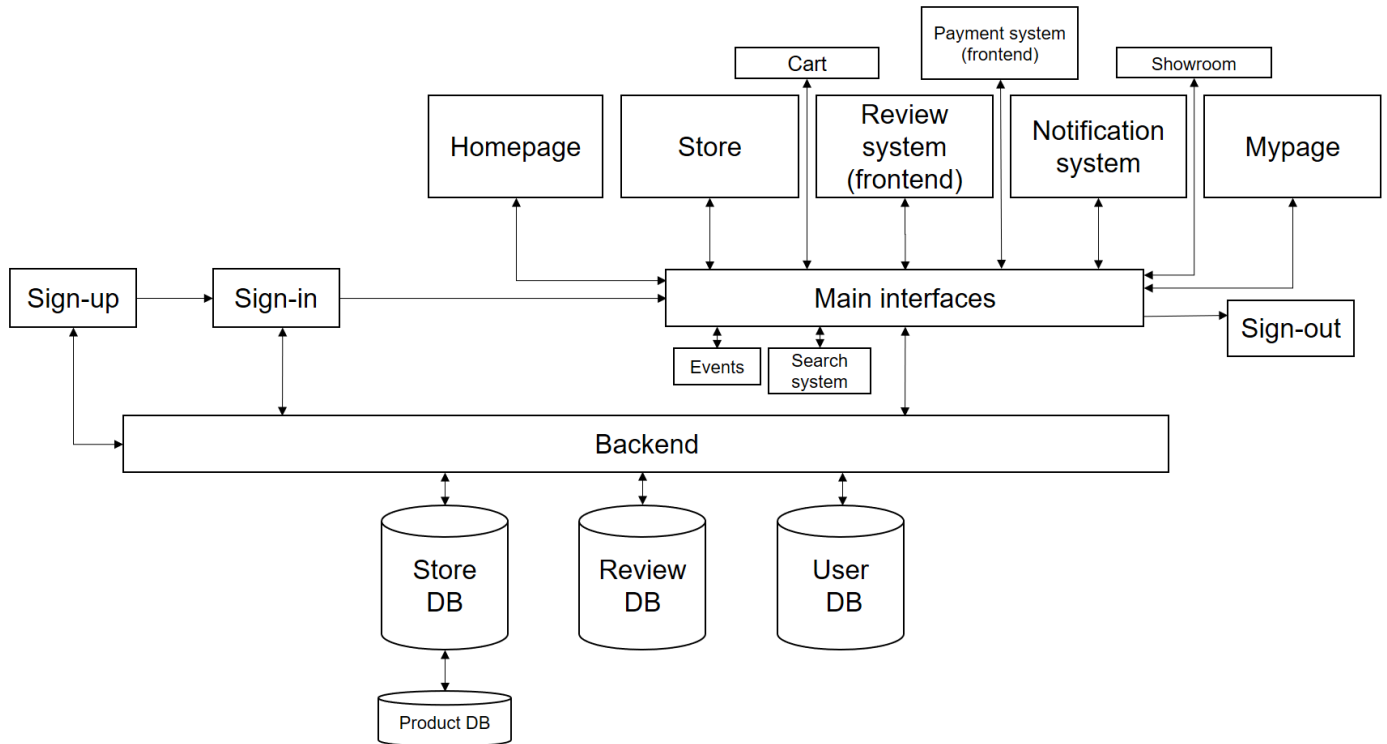
most of the personal information should be confidential from third-party viewers.

#### **3.3.3.2. External System Policy**

As this system provides information reprocessing of the review list of other systems, the system that prohibits unauthorized reproduction of data may violate the operation policy depending on the data collection method. Therefore, it is necessary to carefully check the operation policy of the information collection of the target system and collect data without violating the policy.

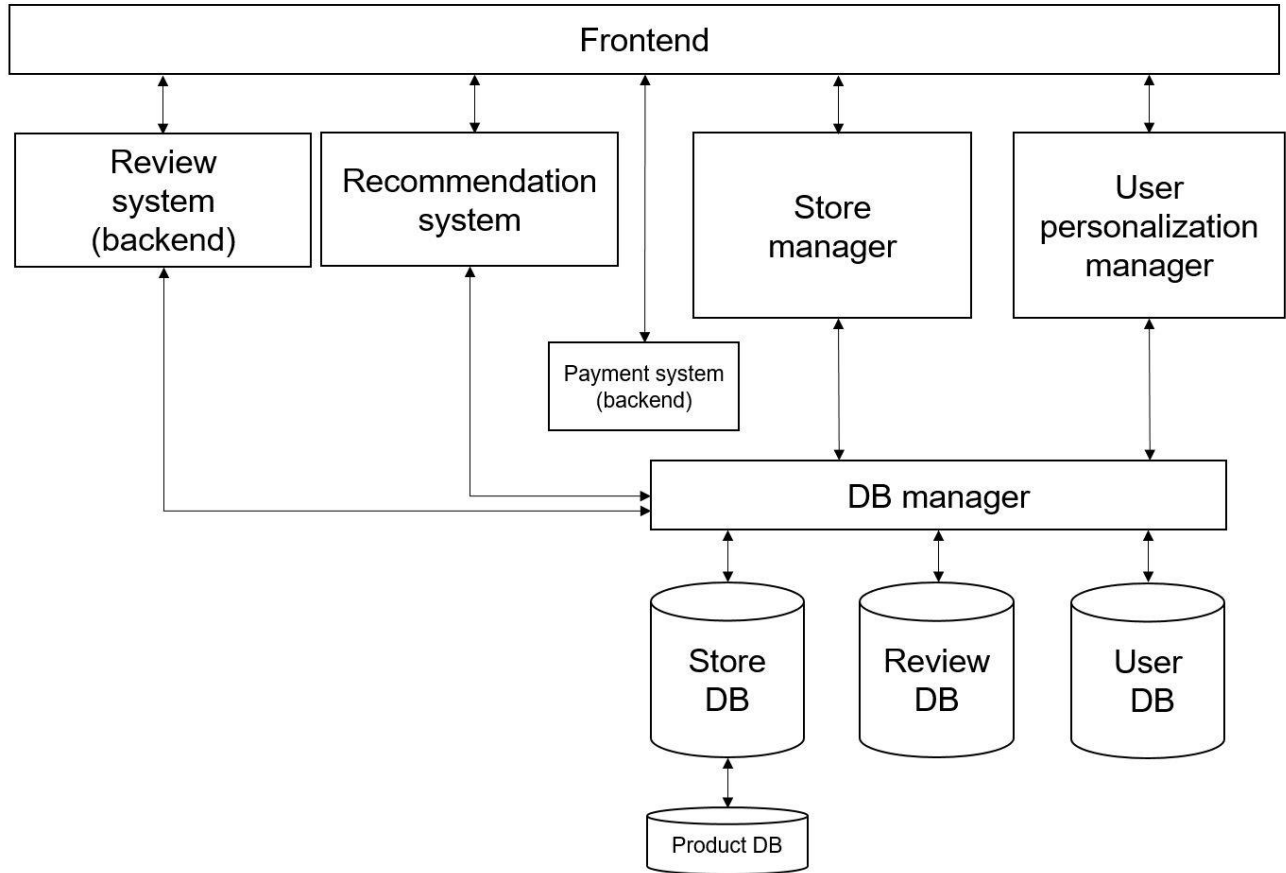
# 4. System Architecture

## 4.1. Frontend Diagram





## 4.2. Backend Diagram



## 4.3. Use-case Diagram

