Requirements Specification

Software Engineering Team 9

김준태 2017314886

김지연 2016312917

양진율 2015313803

이승윤 2016313513

이재원 2015314213

Table of Contents

1. Preface

This chapter explains the readership and document structure.

1.1. Readership

1.1.1. User Requirements Readership

Users of MODU are the target readers of this document. The requirements are written from the user's perspective. Technical words are used at minimum whereas natural language and visual illustrations are mostly used.

1.1.2. System Requirements Readership

The developers of MODU are the target readers of this document. The features and constraints of the development are explained using structural language. Such systematic information is used to write official contract with the users.

1.2. Document Structure

1.2.1. Introduction

It introduces why MODU is so special and needed. The abstract structure and functionalities are given along with the possible effects in society.

1.2.2. Glossary

List of technical words is defined in tabular form. All the users, developers, and stakeholders of MODU can read this document and understand the usage of terminologies in specific context.

1.2.3. User Requirements Definition

Functional and non-functional user requirements are defined in natural language.

1.2.4. System Architecture

Structure of the entire system of MODU is illustrated. It shows how each subsystem handle each user requirements.

1.2.5. System Requirements Specification

This document is the detailed version of User Requirements Definition. Charts and diagrams are mainly used to show the entire development process.

1.2.6. System Models

The relationship among each component in MODU is shown in diagrams. The relationship between MODU and external environment is shown in diagrams as well.

1.2.7. System Evolution

It reflects the truth and limitation of developing MODU. Possible changes in the developing environment are suggested. Risks are measured and corresponding solutions are written thoroughly.

1.2.8. Appendices

Any additional external resources, hardware/software requirements and development environment are noted.

1.2.9. Index

List of diagrams, charts and figures are given in order.

2. Introduction

2.1. Needs

매년 공모전의 규모는 커지고 있다. 이전에는 신문사나 광고 기획사를 통해 광고공모전이 진행되거나, 대기업이나 공기업을 통해 논문공모전, 학술공모전이 진행되는 정도였다. 하지만 해가 지나감에 따라 공모전의 분야는 물론 이를 개최하는 주최 사 또한 다양해지고, 그 수 또한 늘어나고 있다. 현재의 공모전은 일반 기업들의 마케팅 수단이 되며, 행정부나 지자체, 공기업들의 경우 대중과 소통을 위한 수단으로 이용하는 경우가 많다. 뿐만 아니라 공모전의 수상작품을 직접 상품화하거나 수상자를 실제 실무에 참여시키기도 한다. 공모전에 참가하는 대중의 입장에서 보면, 공모전은 수상금을 얻는 것은 물론 경험과 스펙을 쌓을 수 있는 기회가 되며 경우에 따라 취업을 특전을 얻을 수 있는 장소가 되기도 한다. 때문에, 이러한 참여형 프로젝트에 관한 관심이 해가 갈 수록 늘어나고 있다.

이러한 참여형 프로젝트에 관한 관심은 비단 공모전에 관한 것 만으로 그치지는 않는다. 현대사회는 아이디어의 사회라 할 수 있을 만큼 참신한 아이디어와 이를 실현시키는 힘이 중요시되는 사회이다. 또한, 설령 공모전이라는 형태가 아니더라도 프로젝트에 참가했다는 실적은 미래를 위한 경험과 스펙이 된다. 때문에 공모전의 여부와는 관계없이 자신의 참신한 아이디어를 실현시키기 위해 개인적으로 개발 팀을 모아 프로젝트를 진행시키기도 한다. 이러한 경향은 특히 사회에 나가기 직전 경험을 쌓고 배우는 단계인 대학생들에게 두드러지게 나타난다.

하지만 높아지는 공모전과 프로젝트에 관한 열기와는 반대로, 정작 실제 공모전이나 프로젝트에 관한 참여율은 저조한 경우가 많다. 실제로 공모전이 개최되었음에도 불구하고 참가자가 모이지 않아 공모전이 취소되는 경우가 있는가 하면, 수차례 기한을 연장시키는 사례들도 있다. 때로는 최대 수상자 수 보다 참가자가 적게 모여 심사와 수상조차 제대로 이루어 지지 않는 경우도 있다.

이러한 공모전이나 프로젝트를 진행하기 위한 팀 모집은 전통적으로 지인이나 학연을 통하여 구하는 수단이 있다. 하지만 이러한 경우 구할 수 있는 인원의 폭이 좁으며, 적절한 지인이 존재하지 않는 경우도 있다. 때문에, 다양한 분야의 인원을 구하거나 지인 관계가 좁은 경우 "에브리타임"이나 "카카오톡", "캠퍼스픽"과 같은 커뮤니티 앱을 이용하고는 한다. 하지만 커뮤니티 앱을 이용하면 쉽사리 다른 글에 파묻혀져 다른 이에게 노출

되지 못한 채 잊혀져버리는 경우가 많다. 실제로 여러가지 커뮤니티 앱을 통해 프로젝트 모집과 관련된 글을 읽어보면 낮은 조회수와 함께 댓글조차 달리지 않는 경우를 많이 볼 수 있다. 프로젝트에 참가하고자 하는 개발자의 입장에서 보면, 커뮤니티 앱을 통한 모집은 사람마다 양식이 달라 필요한 정보 교환을 위한 대화 과정이 번거로우며, 상대방의 역량이나 프로젝트를 향한 진심을 확인하기 어렵기 때문에 신뢰성이 부족하다고 느끼게 된다.

따라서, 프로젝트를 위한 팀결성을 돕기 위해 프로젝트와 개발자의 소개를 위한 정리된 양식을 제공하고, 의사소통을 위한 간편한 수단을 제공하며, 상호 간의 신뢰성을 판단할 수 있는 시스템의 개발이 필요하다.

2.2. System Overview

MODU는 사전에 데이터베이스에 입력된 프로젝트에 관한 정보와 개발자에 관한 정보를 바탕으로 태그 및 필터, 순서 정렬을 통하여 사용자가 원하는 프로젝트와 개발자를 찾아주고,"참가 문의"와 "초대"기능을 통하여 팀을 결성하게 해주는 팀 매칭 시스템이다.

또한 사용자의 자격증이나 수료증, 수상경력, 재학 정보 등을 추가하고 MODU에서 진행하였던 프로젝트에 관한 이력들을 정리함으로써 사용자 스스로 간편하게 Portfolio를 관리할 수 있는 기능을 갖추고 있다.

MODU에서는 단순히 팀을 결성하는 데에 도움을 주는 것에 그치지 않고, 프로젝트 완료 후 팀원 상호 간의 역량을 평가하도록 하는 기능이 있다. 이렇게 평가된 유저의 정보는 유저 스스로가 역량 강화를 위한 지표로써 사용 가능하며, 이후 또다른 유저가 해당 유저의 프로젝트에 참가하거나 자신의 프로젝트로 초대하기 위한 신뢰도 평가의 척도로도 활용 가능하다.

2.3. Expected Effects

2.3.1. Novel Projects and Finding team members

본 시스템은 간편하고 적절한 프로젝트 제안 수단을 제공함으로써 적극적인 프로젝트 제안을 촉진시키며, 개발자의 주요 관심 분야, 과거 프로젝트 이력, 상호 평가를 통한 강점과 약점에 관한 것 등 상세한 정보를 제공함으로써 본인에게 적절한 팀원을 모집할 수

있도록 만들어 준다.

2.3.2. Student Connection

설령 프로젝트가 완료되더라도 팀원과의 관계가 끝나는 것은 아니다. 개발자 간의 상호 평가를 통하여 리뷰를 남기게 되며, 진행하였던 프로젝트에 관한 기록이 남고, 마음에 든 프로젝트나 개발자에게 관심을 표현할 수 있다. 때문에 본 시스템을 통하여 학생들은 해당 프로젝트를 수행하기 위해서 뿐만이 아닌 앞으로의 인생을 위한 관계를 구축할 수 있다.

2.3.3. Personal Reflection

기존의 프로젝트 진행 방식은 사람마다 평가 방식이 다르며 때로는 서로 간의 제대로 된 평가조차 이루어 지지 않아 피드백으로 이어지지 않는 경우가 있다. 본 시스템은 프로젝트를 완료한 후 일정한 양식의 상호 평가를 제공하며, 이러한 평가 결과가 추후의 프로젝트 팀 결성에 영향을 미치기 때문에, 유저가 본인의 약점을 파악하고 이를 개선하도록 촉구하게 된다.

3. Glossary

Table 1. Glossary

Term	Description
MODU	Our official Website Name.
MODU 모여라	A page that shows a list of Projects.
MODU 어딨니	A page that shows a list of Student Developer Portfolios.
User	The people who are in the process of signing up and who are already signed up as a member of the MODU website.
Project	A Software Engineering Project idea that consists of the Project title, purpose, expected output, roles, development status, project duration, and currently involved team members. A project would be proposed in order to gather Student Developers who are interested in working together on the project.
Project Proposer (=Idea Proposer)	A Student Developer who proposes the project and posts it on the MODU 모여라 web page.
Developer (=Student Developer)	The students who are involved in at least one of the projects of the MODU 모여라 web page.
Search	A method of using the search tab and entering keywords/hashtags in order to efficiently find Projects/Student Developers that they are interested in.
Tag(#)	A tool used in the Search tab to efficiently find Projects/Student Developers with certain advanced attributes.
Filter	A tab on the left side of both the MODU 모여라 and MODU 어딨니 page to efficiently find Projects/Student Developers by choosing among the pre-defined keywords in the Filter tab.
Favorite	A function to show interest in a particular Project, and get future notifications when the Project information is updated. A user may "Favorite" many Projects without limit.
Follow	A function to show interest in a particular Student Developer, and get future notifications when the Student Developer information is updated. A user may "Follow" many Student Developers without limit.
Comment	A function used to ask questions and add opinions about Projects in the Project Specification Page.
Participation Inquiry	This function is used when a user finds a Project that he/she is interested in. The user may send message the Project Proposer and initiate communication about joining the Project.

Invitation	This function is used when a Project Proposer/Student Developer finds another Student Developer that they are interested in working with. The Project Proposers/Student Developers may message another Student Developer and initiate communication.
System Notification	A notification alarm that shows a message when the user's Peer Assessments is updated, or when Participation Inquiries/Invitations are sent by other users.
Portfolio	A page that shows each users' name, self-introduction, LinkedIn/Github, E-Mail, current projects, past projects, Peer Assessment reviews.
My Page **(=Personal Page)**	Each users' own Portfolio page.
Portfolio Page	A page that shows all the MODU users' Portfolio information in a summarized thumbnail.
Peer Assessment	A function that is conducted after a Student Developer has completed a project with other users. This is used to rate the other users' performance as a Project team member. A 5-star rating method and review adding is used.
Waterfall Model	A sequential model that divides the software development into pre-defined pages. Each phase must be completed in prior of the next phase. The phases are Requirement Analysis, System Design, Implementation, System Testing, System Deployment, and System Maintenance. The Waterfall Model is used to build our MODU system.

4. User Requirements Definition

This chapter describes both functional and non-functional requirements. Users are the target reader of this chapter, so technical words are used at minimum. Mostly, natural languages and visual illustrations are used to describe each requirement.

4.1. Functional Requirements

4.1.1. User Authentication

User authentication is a requirement where users can identify themselves by using their email and password prior to using the website. When signing up and logging in, the users should input their email address and password. The sign-up process would require an additional email verification process. Using Firebase Authentication, verifying users would be simplified and user information would be safely protected.

4.1.2. Search

Search is a requirement that allows users to search information using specific keywords, and saves time by enabling users to find only the information with interest. In all the pages of the website, users can enter specific keywords related to both projects(MODU 모여라) and student developers(MODU 어딨니), and find projects/student developers related to the keywords of input. Tags are additionally used to search advanced keywords such as deep learning.

4.1.3. Project Information

Project Information is a requirement used when users upload their project ideas on the MODU 모여라 page. The website requires the idea proposer to input information of the project as specifically as possible by entering a certain format. The idea proposer should enter the project idea title, purpose, expected output, roles, development status, project duration, and currently involved team members. Each project thumbnail would show a summarized form of the project information so that users can look through many different projects in a short period of time and choose the project of most interest. The specific information would be shown when a user clicks a certain project thumbnail.

4.1.4. Portfolio

Portfolio shows the user's information in categories. The data is entered once the user signs up. The data can be further edited in the personal setting. Basically, the user writes a short self-introduction. If possible, the user can attach links to LinkedIn or GitHub. It is mandatory to show the email address to the public in case of any inquires related to projects. Moreover, list of currently working-on projects is shown along with a list of past projects. For each past project, the user can manage the portfolio by indicating his/her roles and writing down personal feedback of the project. For example, the user can reflect on what has turned out well and what has not. Thus, all the records can become good references in the future with all the personal know-hows.

4.1.5. Peer Assessment

Peer assessment simply means that the remaining members in the project evaluate your performance once the project is completed. The requirement is to evaluate one another in terms of ideation, development, communication, and overall input to the project. A clear rating method is used; if the member has outstanding performance, then five stars can be given. Additional comments can be written only if necessary. Name of the peer assessor is available in public.

When an idea proposer wants to access peer assessment of a developer, strengths and weaknesses are shown in text without ratings. For example, if the developer has received multiple responses with five stars for ideation and communication. The proposer is not allowed to access the detailed peer assessment.

If a developer has shown bad performance in the past three projects, then a warning will be shown in the Portfolio page.

4.1.6. System Notification

시스템에서 사용자에게 필요한 정보를 알려주는 기능이다. 사전에 사용자가 회원가입 시 혹은 Portfolio를 통하여 입력한 키워드가 포함된 프로젝트가 시스템에 등록되었을 경우, 사용자에게 메시지 알림을 보낸다. 이 외에도, 프로젝트가 완료된 후 상호평가 등록, "favorite"나 "follow" 표시, 관리자로부터의 메시지 전송 등의 경우에 사용자에게 알림으로 전해진다.

4.1.7. Favorite & Follow

사용자가 해당 프로젝트나 개발자에게 관심이 있음을 표시하는 기능이다. 사용자는 개발자나 프로젝트의 Search 페이지 (MODU 모여라, MODU 어딨니)에서 관심을 표시하고 싶은 항목의 "Favorite"를 클릭하여 표시를 활성화시키거나 다시 클릭하여 이를 비활성화시킬 수 있다. 사용자는 마이페이지를 통하여 "Favorite" 표시를 하였던 개발자나 프로젝트들의 현황을 확인할 수 있으며, 만약 개발자나 프로젝트에 변화가 생겼다면 이에 관한 알림을 받을 수 있다. "Favorite"를 통하여 관심을 받게 된 개발자나 프로젝트 게시자는 "Favorite"를 받았다는 알림을 받게 된다.

4.1.8. Participation inquiry

프로젝트 상세 페이지의 "참가 문의" 버튼을 클릭하여 프로젝트 게시자에게 참가의사를 전하는 기능이다. 사용자가 "참가 문의" 버튼을 클릭하면, 프로젝트 게시자에게 사용자와 참가 문의가 온 프로젝트에 관한 내용이 알림으로 전해진다.

4.1.9. Invitation

개발자 상세 페이지의 "초대 제의" 버튼을 클릭하여 개발자에게 프로젝트로의 초대의사를 전하는 기능이다. 사용자가 "초대 제의" 버튼을 클릭하고 초대하고 싶은 프로젝트를 선택하면, 해당 개발자에게 사용자와 초대 제의가 된 프로젝트에 관한 내용이 알림으로 전해진다.

4.2. Non-functional Requirements

4.2.1. Product Requirements

4.2.1.1. Usability

Usability is one of the important requirements of website service for recruitment. MODU should provide a very intuitive yet simple user interface (UI). The system provides such UI through minimal, but essential functions, buttons and tabs which help the user to navigate quickly between different sections without any inconvenience.

4.2.1.2. Efficiency

It is important for the members to have interactive activities between members other than MODU itself. To do this, the absolute number of members and the amount of service use by those members must increase.

4.2.1.3. Dependability

Since **MODU** requires free recruitment and interactive communication between members, it is necessary to minimize the intervention of administrator. However, it is necessary to intervene in matters that violate the internal policy, such as illegal activities or creation of false profiles.

4.2.1.4. Security

All members' profile information and contact information should be set disclosed. In addition, in order to increase the number of members and usage, members should be able to trust the policies for safe security and privacy of the system.

4.2.2. Organizational Requirements

4.2.2.1. Environmental

MODU is a PC web-based service, so should be developed using tools suitable for it. In addition, pages and services should work properly no matter what Internet browser used.

4.2.2.2. Operational

MODU must have continuous communication with the server. As it is an online service, continuous and immediate communication with the server is prerequisite. Minimization of transmission / reception of data and stabilization of services are essential. **MODU** provides continuous and stable service using docker server.

4.2.2.3. Development

In the case of **MODU**, it is important to manage documents and artifacts, and it is expected that there will be little or no change in requirements. Therefore, **MODU** is carried out using the waterfall model.

4.3. External Requirements

4.3.1. Regulatory

MODU collects the user's personal information, and if the personal information is not properly protected, there is a possibility of taking civil and criminal responsibilities. Therefore, when signing up, only minimal information necessary for user identification is provided, and sufficient efforts should be made to prevent personal information from being leaked. Objects that are not specified in the user's personal information agreement cannot access the user's personal information

4.3.2. Ethical

Mutual evaluation of team members, which is conducted for the reliability of team members, can be exploited to give low or high scores without evidence. This will reduce the reliability of the evaluation and should be restrained through regulations. In addition, the intention to promote the progress of non-profit projects by college students has been altered, and posts for job openings may be posted. This should also be prevented through a reporting policy.

5. System Architecture

This chapter describes the general architecture of the system. The overall structure of the system, the composition of each subsystem, and the relationship between the subsystems are outlined, and each structure is attached with a diagram to help understanding.

5.1. Architecture Overview

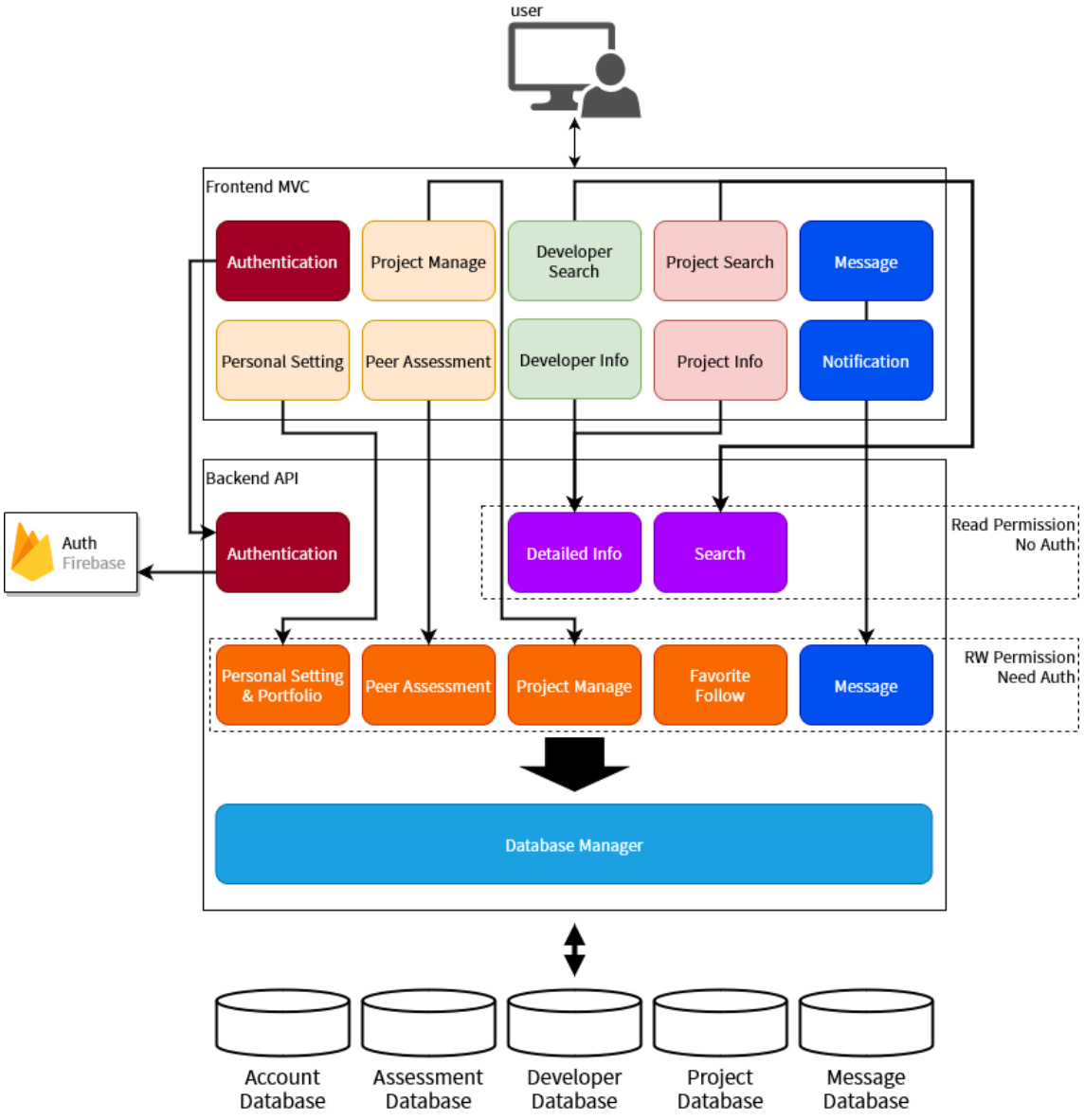


Figure 1. Architecture Overview

The overall architecture is composed with frontend and backend parts. In MODU, the frontend interface is made of authentication feature, 2 search features, 2 information view features, project

manage feature, personal setting & portfolio manage feature, peer assessment feature, and message & notification features.

In backend server, there are eight subsystems. Authentication subsystem supports authentication itself. Detailed information view subsystem supports information lookup. Search subsystem supports project and developer search features. These two subsystems do not need user account authentication information as an argument of API call. Other five subsystems need to be authenticated to work. All subsystems do their jobs as their names suggest.

5.2. Authentication System

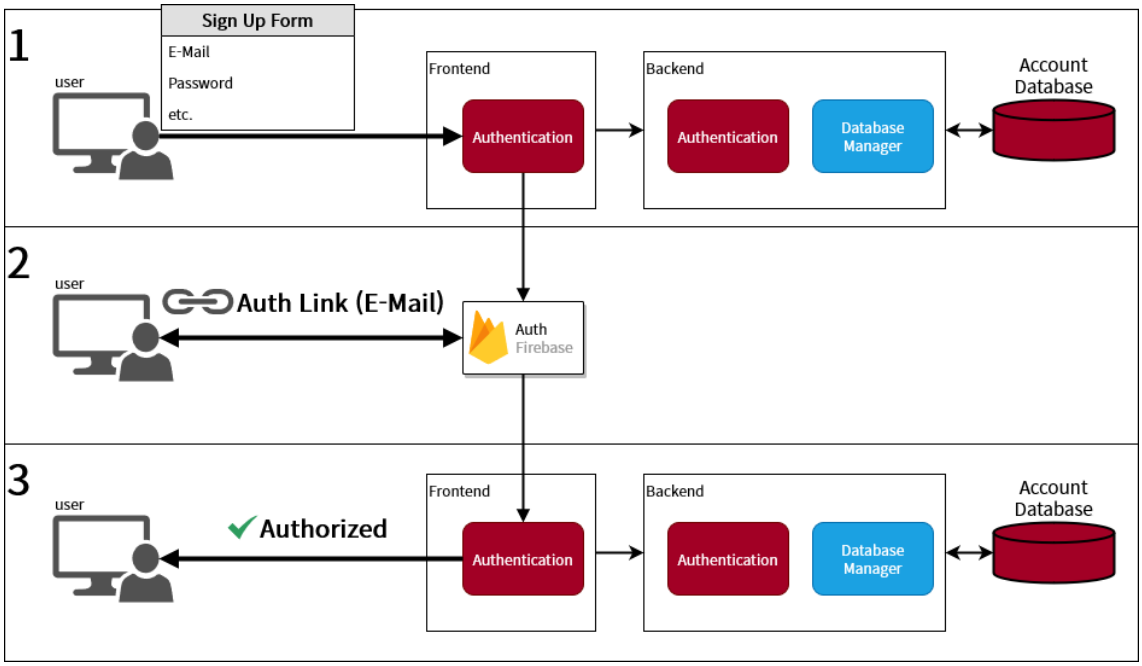


Figure 2. Sign Up

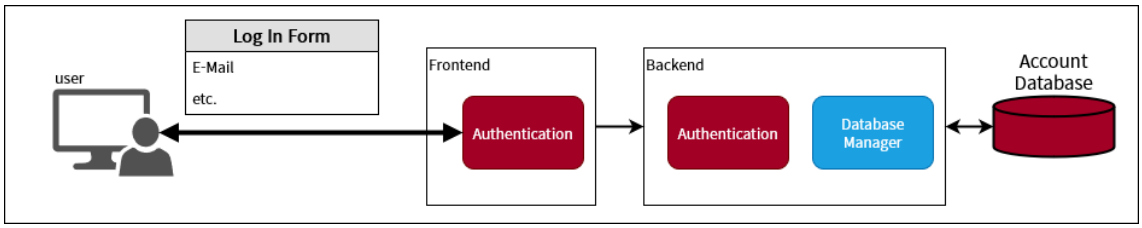


Figure 3. Log In & Out

Authentication system supports the sign-up process and log in & out processes. Sign-up utilizes external E-Mail linking based authentication system with Firebase SDK.

5.3. Project Search System

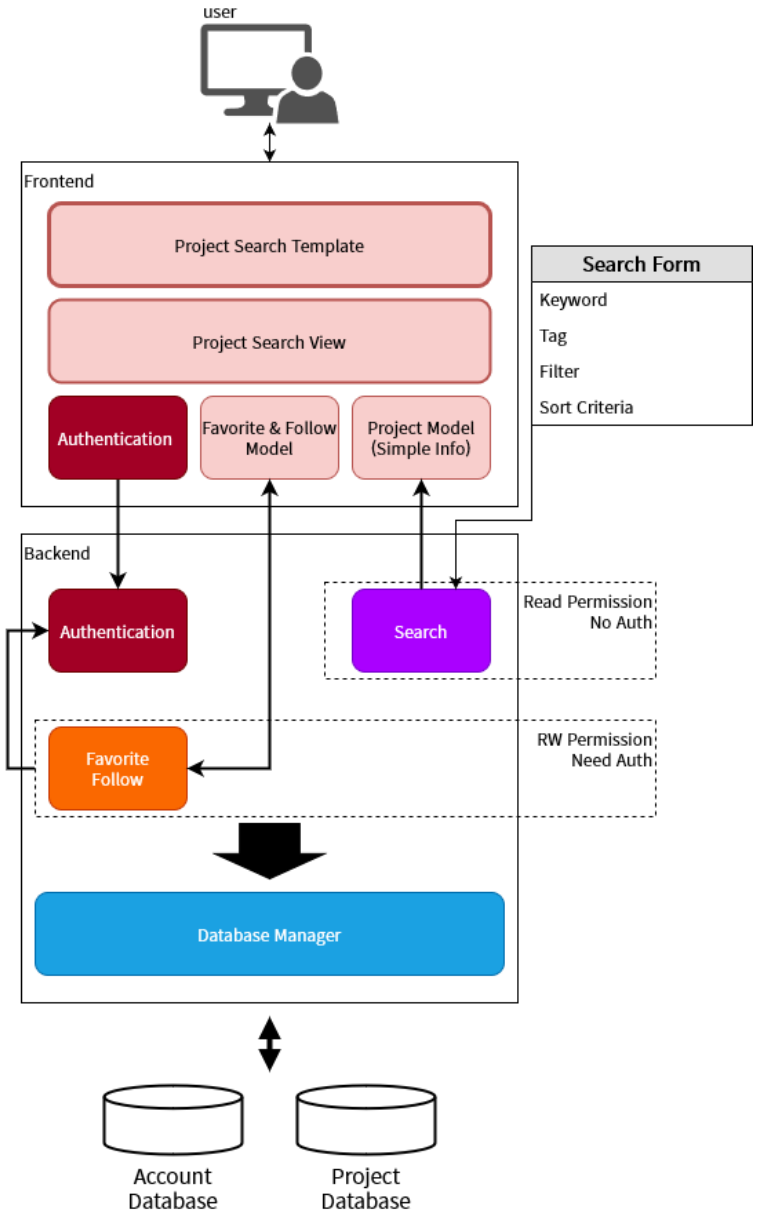


Figure 4. Project Search System

Project search system deals with project search requests. It consists of one template, one view, and two models, in terms of the Django MTV pattern, neglecting the auth subsystem. In the backend, search subsystem retrieves search result and favorite/follow relation subsystem retrieves user's "favorite" status for each project, from project database.

User can use tag, filter, and sort criteria features for search. Accessing a page of any retrieved project will activate project information view system.

5.4. Developer Search System

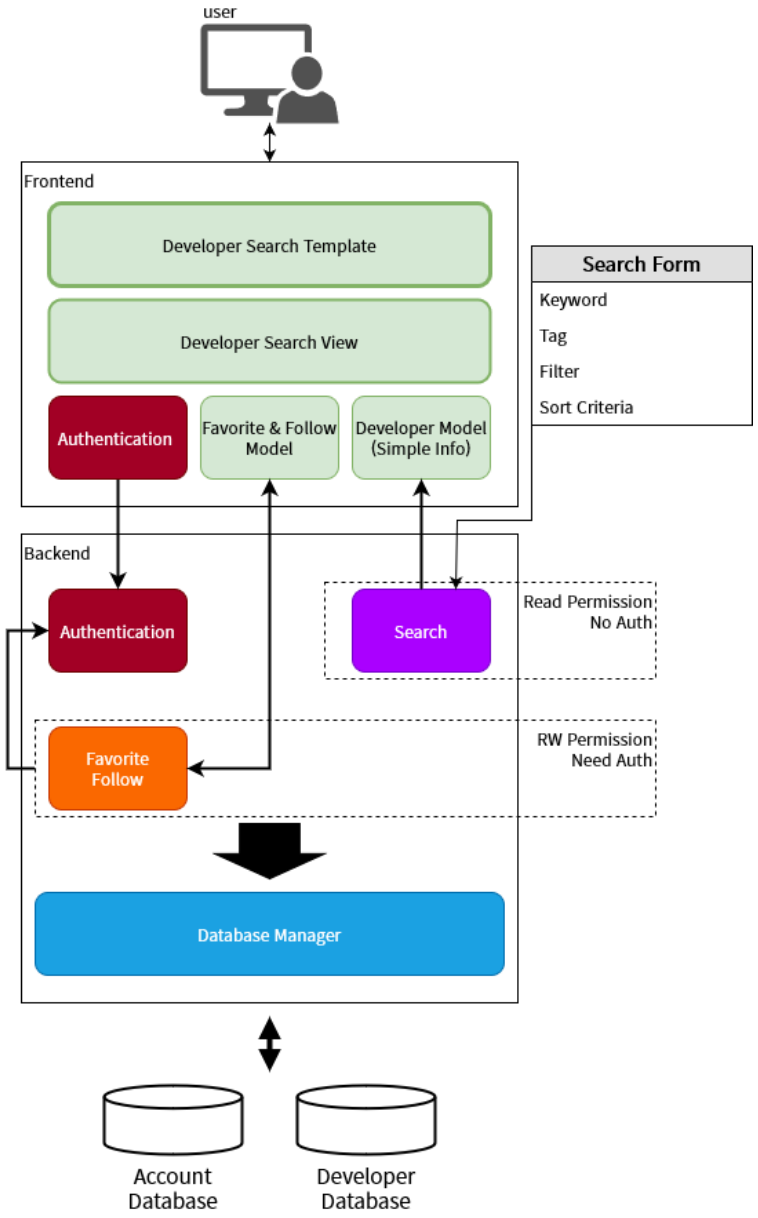


Figure 5. Developer Search System

Developer search system deals with student developer search requests. It consists of one template, one view, and two models, in terms of the Django MTV pattern, neglecting the auth subsystem. In the backend, search subsystem retrieves search result and favorite/follow relation subsystem retrieves user's 'follow' status for each developer, from developer database.

User can use tag, filter, and sort criteria features for search. Accessing a page of any retrieved project will activate project information view system.

5.5. Project Information View System

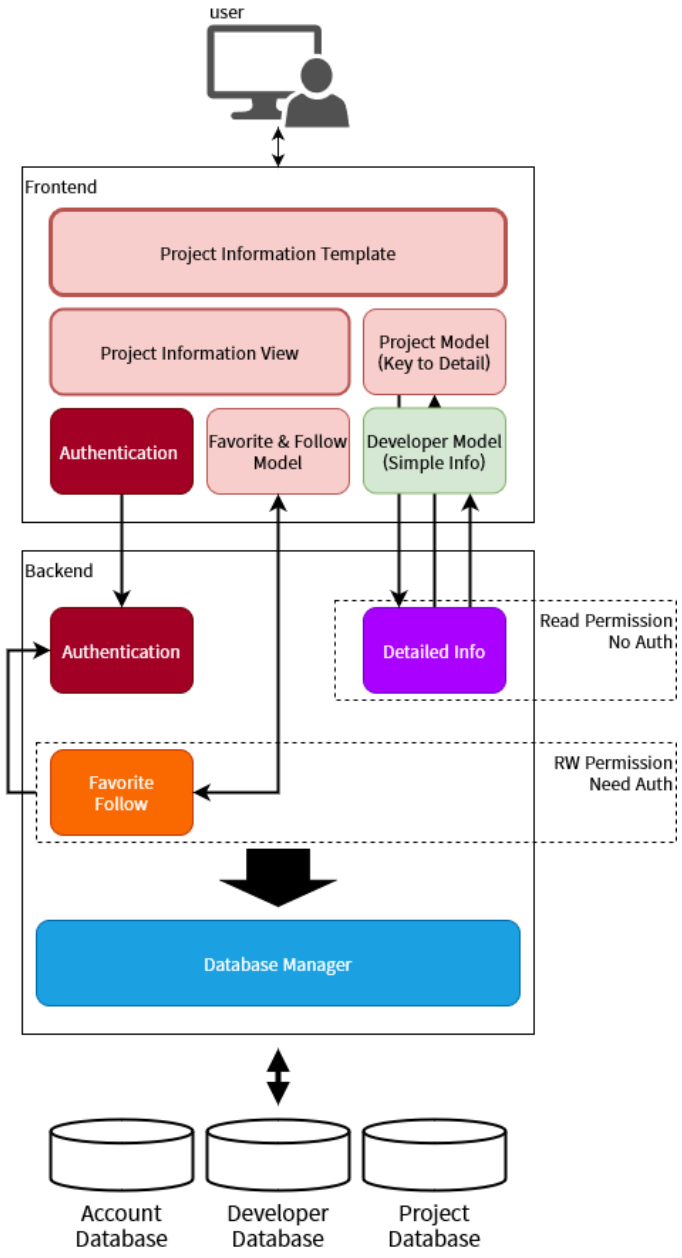


Figure 6. Project Information View System

Project information view system deals with project information page view requests. It consists of one template, one view, and three models, in terms of the Django MTV pattern, neglecting the auth subsystem. In the backend, detailed information view subsystem shows the detailed information and favorite/follow relation subsystem retrieves user's "favorite" status for the project from project database. User can "favorite" the project, add comments and inquire for participation to the project.

5.6. Developer Information View System

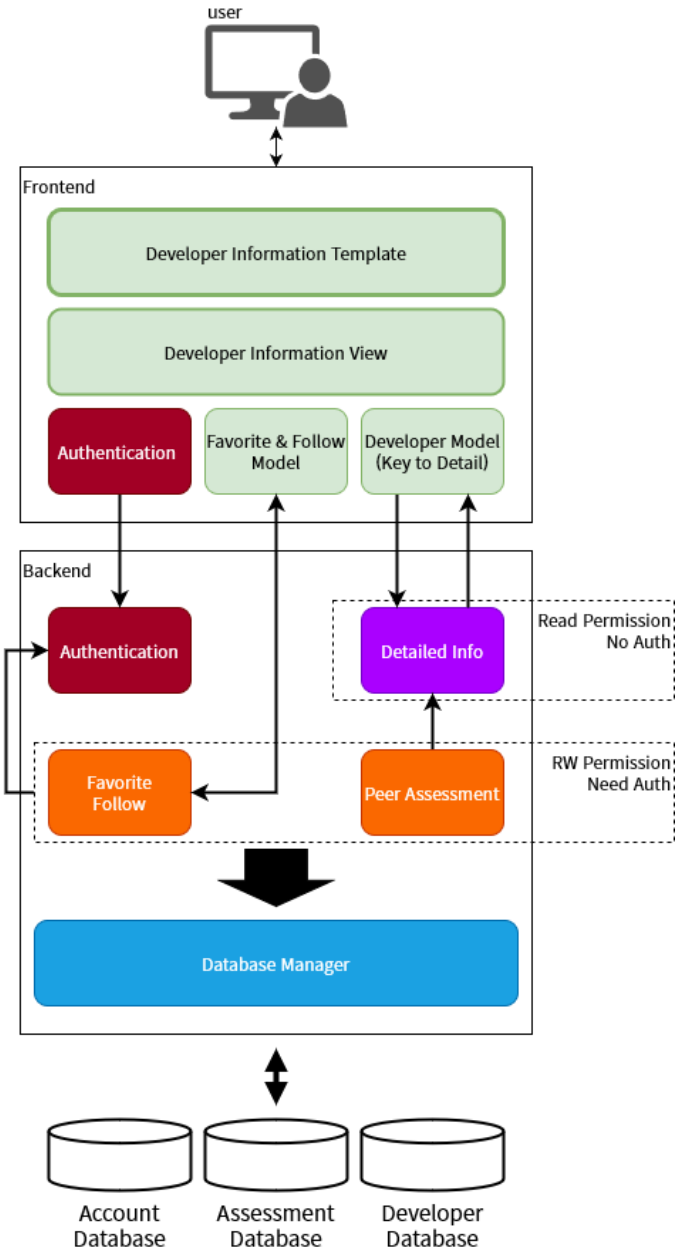


Figure 7. Developer Information View System

Developer information view system deals with developer information page view requests. In the backend, detailed information view subsystem shows the detailed information, favorite/follow relation subsystem retrieves user's 'follow' status for the developer from developer database, and peer assessment subsystem shows peer assessments about the developer. User can 'follow' the developer, see peer assessments about the developer, and invite to user's project.

5.7. Project Manage System

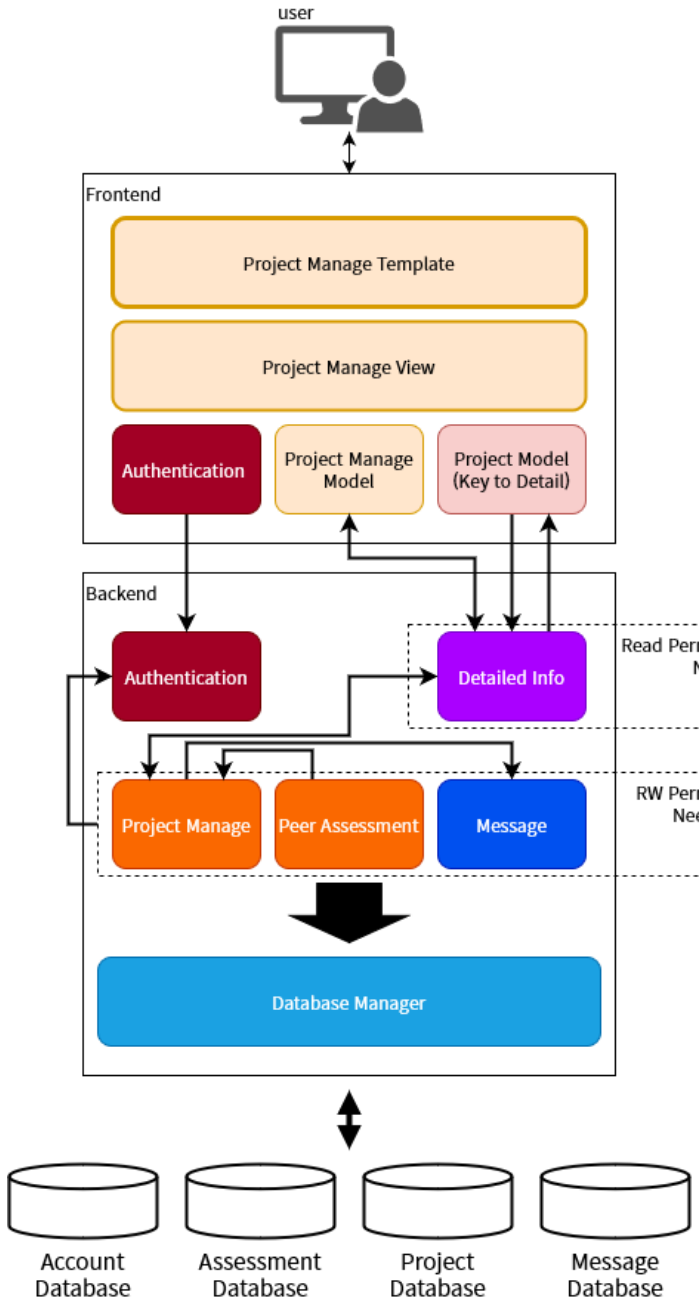


Figure 8. Project Manage System

Project manage system utilizes the management capability for the project members. If the project is completed, the link of peer assessment system for the other members would be activated. Project manage subsystem can communicate with peer assessment subsystem to achieve this feature. When a change is issued, users who 'favorited' the project get a notification via the message system.

5.8. Portfolio Manage System

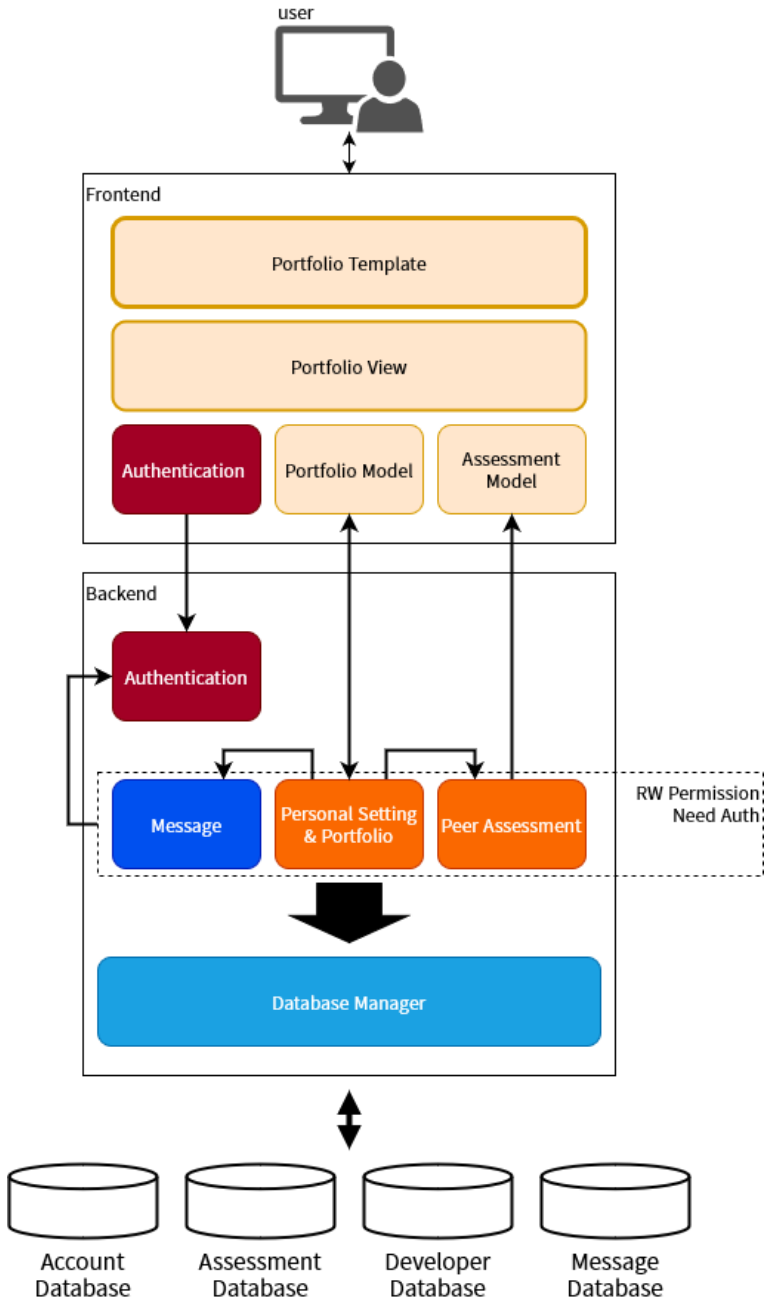


Figure 9. Portfolio Manage System

Project manage system utilizes the management capability for each user's own portfolio. In the system, users can also see their peer assessment scores and opinions, via peer assessment subsystem. With that, users can easily notice their strengths and weaknesses. When a change is issued, users who 'followed' the developer get a notification via the message system.

5.9. Peer Assessment System

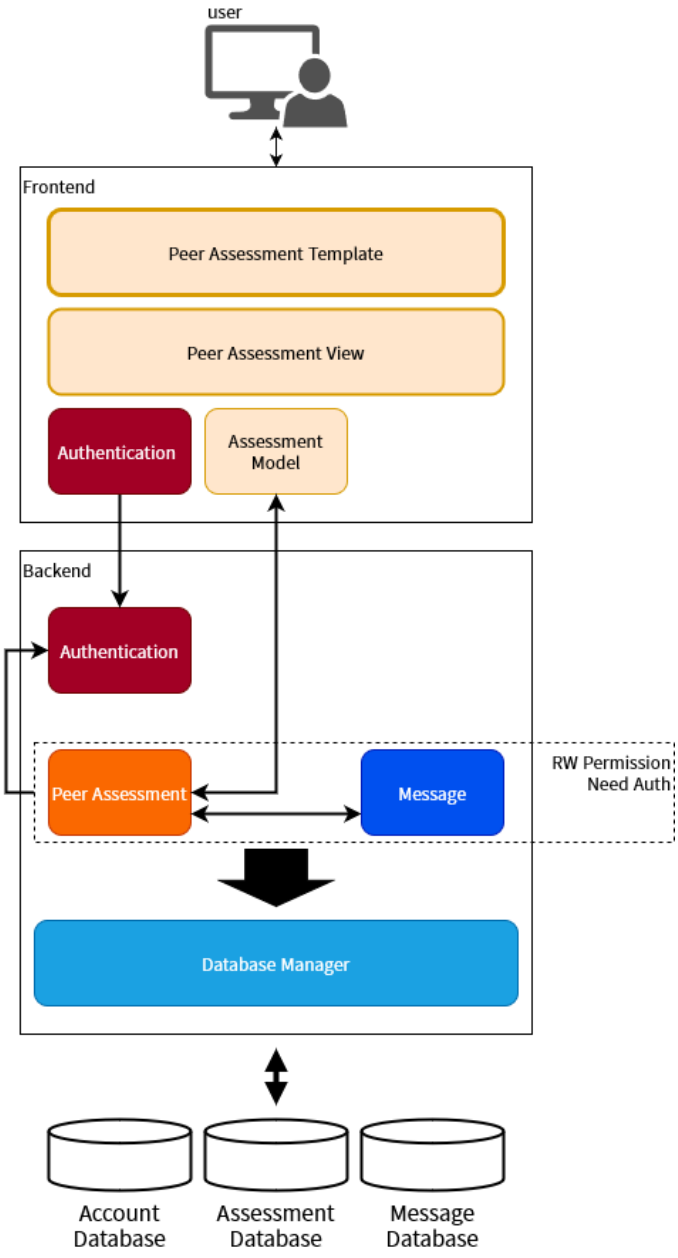


Figure 10. Peer Assessment System

Peer assessment system utilizes the assessment feature to evaluate one's development skill, communication skill, or many other aspects, with both score and opinion methods. In the system, users only can evaluate their ex-team members. Making an assessment will trigger the system to notify the peer assessment update to the subject.

5.10. Participation Inquiry System

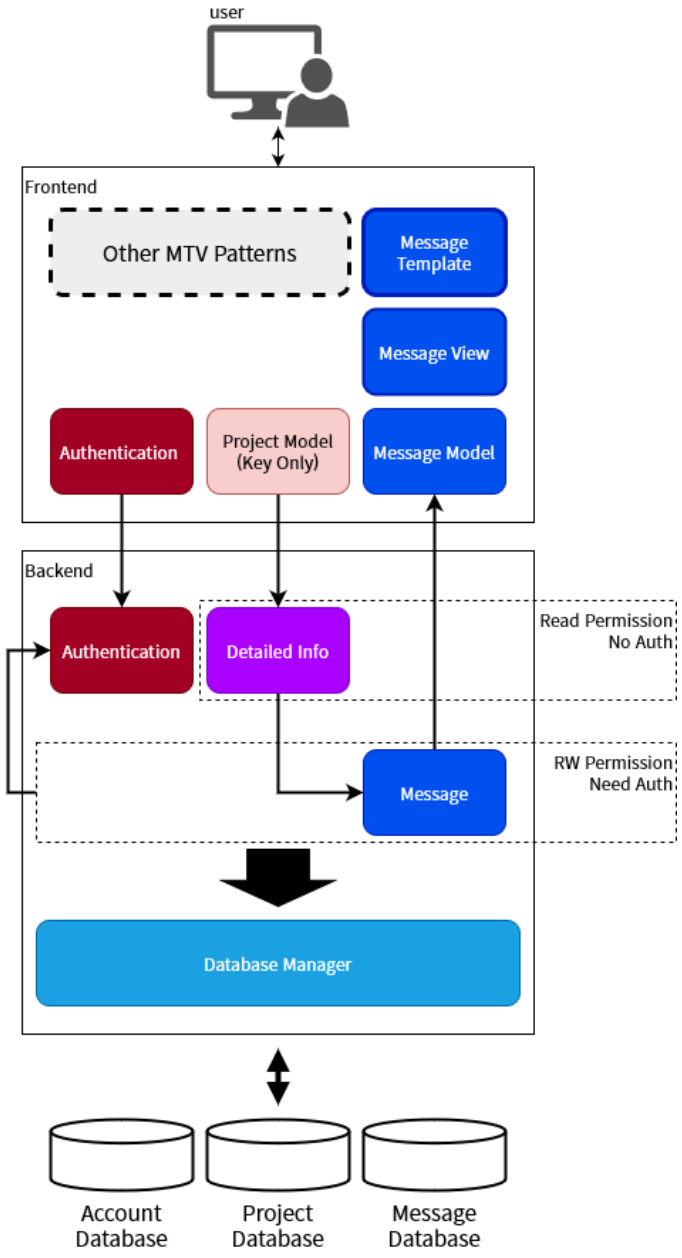


Figure 11. Participation Inquiry System

Using this system, user can send a participation inquiry to the project leader. This system uses message subsystem to send inquiries. Detailed information view subsystem will check the validity of the target project.

5.11. Invitation System

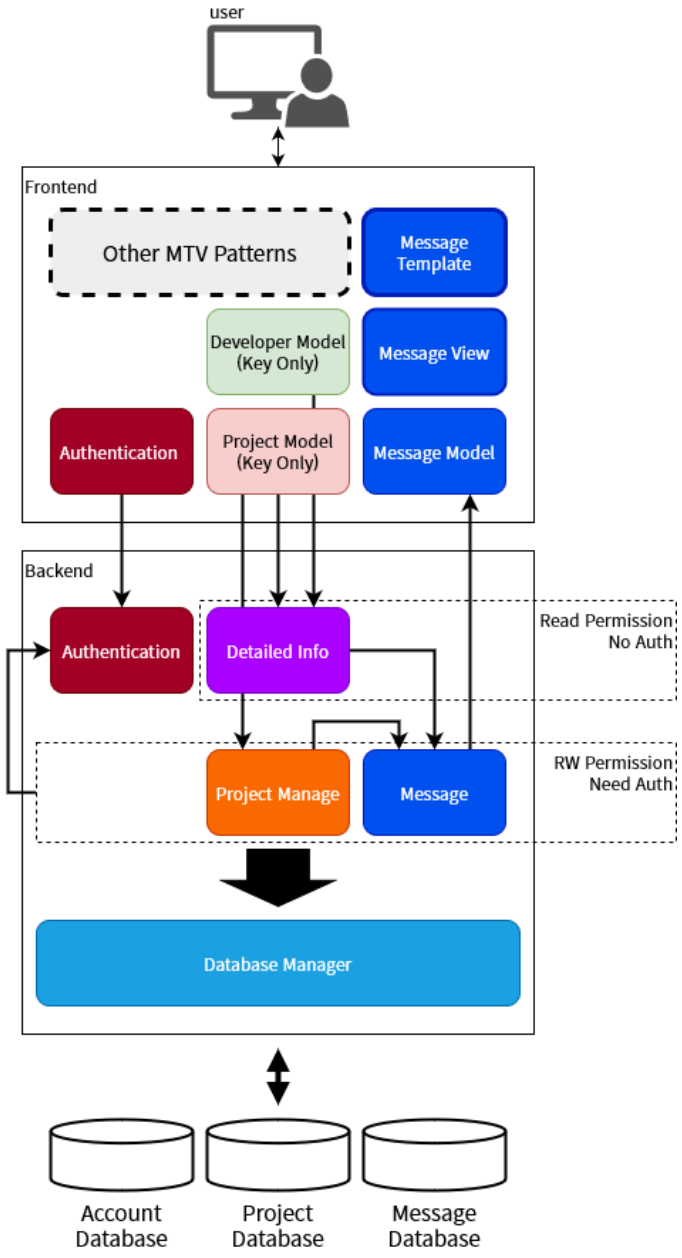


Figure 12. Invitation System

Using this system, user can send an invitation for their project to other developers. This system uses message subsystem to send invitations. To invite a developer, user must have proper authority for the project. Project manage subsystem communicates with detailed information view subsystem and authentication subsystem to check validity of the target project & invitee and the authority of inviter.

6. System Requirements Specification

6.1. Functional Requirements

6.1.1. User Authentication (Sign up and Log in)

Table 2. User Authentication Requirement

Name	**User Authentication**	
Description	A page where users can sign up and log in	
Inputs	E-Mail	E-Mail string of user
	Password	Password string of user
Outputs	The starting page that includes a personal Portfolio page would be shown.	
	The sign up process would additionally output an Email verification page.	
Action	This page allows users to identify themselves by using their email and password prior to using the website. Users can sign up/log in using their email and password. The sign up process would require an additional email verification process. When signing up, the authentication email and password would be updated in the Account DB.	
Requirements	Email and Password	
Pre-condition	New users would not have a personal Portfolio, and users who have an account but did not sign in would not be able to see their personal Portfolio page.	
Post-condition	By signing up, new users would have an account. By logging in, users would be able to access their Portfolio page.	

6.1.2. Adding/Modifying a New Project Idea

Table 3. Adding/Modifying a New Project Idea Requirement

Name	**Project Information**	
Description	A page where Project Proposers can upload new Project ideas.	
Inputs	Project information Adding Button Click	Clicking the Project information Adding Button
	Project Idea	Entering the project idea and specifications
Outputs	A new project idea	
Action	Project Information is a requirement used when users upload their project ideas on the **MODU** 모여라 page. The idea proposer should enter the project idea title, purpose, expected output, roles, development status, project duration, and currently involved team members. When a user uploads a new Project idea, the idea would be added in the Project DB.	
Requirements	The user should enter all of the required format specifications of the project.	
Pre-condition	The user would not have uploaded the project idea.	
Post-condition	The user would successfully have uploaded the project idea.	

6.1.3. Adding/Modifying User Portfolio

Table 4. Adding/Modifying User Portfolio Requirement

Name	**Portfolio**	
Description	A page that specifies user portfolio information	
Inputs	E-Mail	E-Mail string of user
	Password	Password string of user
Outputs	Portfolio information of the user.	
Action	This page shows the user's information in categories. It can be further modified in the personal setting. The user writes a short self-introduction, and it is possible to attach links to LinkedIn or GitHub. It is mandatory to show the email address to the public in case of any inquires related to projects. Moreover, list of currently working-on projects is shown along with a list of past projects. For each past project, the user can manage the portfolio by indicating his/her roles and writing down personal feedback of the project. The Developer DB would be updated when a user adds or modifies the User Portfolio.	
Requirements	Both Email and password must be specified.	
Pre-condition	-	
Post-condition	The User Portfolio would be added or modified.	

6.1.4. Searching for Projects to Join

Table 5. Searching for Projects to Join Requirement

Name	**Project Information**	
Description	A page that specifies all the Project idea information thumbnails.	
Inputs	Project information Button Click	Clicking the Project information Button
	Search Keywords, hashtags, filters, sorting criteria.	Searching Projects using keywords, hashtags, filters, and sorting criteria.
	Project Uploaded Time	When searching for Projects, the Projects would be sorted by uploaded time (default)
Outputs	Project idea thumbnails.	
Action	Each project thumbnail would show a summarized form of the project information so that users can look through many different projects in a short period of time, and choose the project of most interest. The specific information would be shown when a user clicks a certain project thumbnail.	
Requirements	-	
Pre-condition	When opening the Project Information page, the Project thumbnails would be seen in uploaded time sequence by default.	
Post-condition	-	

6.1.5. Portfolio Search for finding teammates

Table 6. Portfolio Search for finding teammates Requirement

Name	**Portfolio**	
Description	A page that shows Portfolio information of other students	
Inputs	Clicking Portfolio information Button	Clicking the Portfolio information Button
	Search Keywords, hashtags, filters, sorting criteria	Searching Portfolio using keywords, hashtags, filters, and sorting criteria.
	Portfolio Uploaded Time	When searching for Student developers, the Portfolios would be sorted by uploaded time(default)
Outputs	Portfolio thumbnails of other users.	
Action	This page shows a list of thumbnail portfolios of other users. Each thumbnail includes a name, school/department, a short self-introduction, and LinkedIn or GitHub links if the user has one. The technical skills of the user would also be specified such as fluent computer languages and projects that the user was involved in.	
Requirements	-	
Pre-condition	When opening the Portfolio page, the Portfolio thumbnails would be seen in uploaded time sequence by default.	
Post-condition	-	

6.1.6. Specific Portfolio Page

Table 7. Specific Portfolio Page Requirement

Name	**Portfolio**	
Description	A page that shows specific Portfolio information of other students	
Inputs	Clicking Portfolio thumbnail Button	Clicking the Portfolio thumbnail of interest
Outputs	Specific Portfolio information of other users.	
Action	When a user clicks a portfolio thumbnail of interest, it would show specific user information in categories. It would include a short self-introduction, and possibly a link to LinkedIn or GitHub. The email address and list of currently working-on projects is shown along with a list of past projects. For each project, specific roles and personal feedback may be included.	
Requirements	-	
Pre-condition	-	
Post-condition	-	

6.1.7. Conducting Peer Assessment of Other Members

Table 8. Conducting Peer Assessment of Other Members Requirement

Name	**Peer Assessment**	
Description	A page that shows the peer assessment rates and reviews.	
Inputs	5-Star Rates	Rating peers in a range of 1-5 stars.
	Reviews	Entering reviews of peers.
Outputs	Star rates and reviews of peers.	
Action	Peer assessment is done to project to evaluate your performance once the project is completed. The requirement is to evaluate one another in terms of ideation, development, communication, and overall input to the project. A rating method is done by giving star rates of 1-5. Additional comments can be written only if necessary. Name of the peer assessor is available in public. When a user conducts Peer Assessment of another member, the Assessment DB would be updated.	
Requirements	A user has to have conducted a project with the member that the user wishes to write a peer assessment about.	
Pre-condition	-	
Post-condition	A peer assessment would be added.	

6.1.8. Participation Inquiry/Invitation

Table 9. Participation Inquiry/Invitation Requirement

Name	**Participation Inquiry/Invitation**	
Description	A message link that enables users message project proposers to join projects or for project proposers to message student developers of interest.	
Inputs	Clicking buttons for Participation Inquiries/ Invitations	Participation Inquiry - Clicking the '참가문의' button Invitation – Clicking the 'Invite' button
	Message	Typing in the message
Outputs	A message would be sent.	
Action	Student developers may search for projects that catch interest, and send a participation inquiry. Project proposers may search for student developers who have necessary skills, and send an Invitation link. When sending a participation inquiry or an Invitation message, this would be added in the Message DB.	
Requirements	A user has to be logged in.	
Pre-condition	Student developers and Project Proposers would not be able to show interest to each other by messaging.	
Post-condition	Student developers and Project Proposers would be able to communicate through messaging.	

6.1.9. System Notification (Message)

Table 10. System Notification Requirement

Name	**System Notification**	
Description	A message that is shown in the notification box.	
Inputs	-	-
Outputs	A notification message would be seen.	
Action	When a user sends a message, or a peer assessment for the user is uploaded by another user, or when some other user follows/favorites the user, a message would be shown in the notification box.	
Requirements	A user must be logged in.	
Pre-condition	-	
Post-condition	There would be an additional message in the notification box.	

6.1.10. Favorite & Follow

Table 11. Favorite & Follow Requirement

Name	**Favorite & Follow**	
Description	A user may show interest of a project or another student developer user by "favoriting" the project/student developer	
Inputs	Click "Favorite" button	Clicking the "Favorite" button
Outputs	The "favorite" button would be clicked, and the user who got the "favorite" would get a notification.	
Action	A user can express interest to a project/ student developer by "favoriting" the project/student developer, and the user who uploaded the project idea or the developer themselves would get a notification. When a student developer is "favorited", the "favorite" information would be updated in the Developer DB. When a project is "favorited", the "favorite" information would be updated in the Project DB.	
Requirements	A user must be logged in.	
Pre-condition	-	
Post-condition	-	

6.2. Non-functional Requirements

6.2.1. Product Requirements

6.2.1.1. Usability

Usability is one of the important requirements of website service for recruitment. MODU should provide a very intuitive yet simple user interface (UI). MODU provides such UI through minimal, but essential functions, buttons and tabs which help the user to navigate quickly between different sections without any inconvenience.

6.2.1.2. Efficiency

It is important for the members of MODU to have interactive activities between members other than MODU itself. To do this, the absolute number of members and the amount of service use by those members must increase.

6.2.2. Dependability

Since MODU requires free recruitment and interactive communication among members, it is necessary to minimize the intervention of administrator. However, it is necessary to intervene in matters that violate MODU policy, such as illegal activities or creation of false profiles.

6.2.3. Security

All members' profile information and contact information should be set disclosed. In addition, in order to increase the number of members and usage, members of MODU should be able to trust the policies for safe security and privacy of the system.

6.3. Organizational Requirements

6.3.1. Environmental

MODU is a PC web-based service, so should be developed using tools suitable for it. In addition, pages and services should work properly no matter what Internet browser is used.

6.3.2. Operational

MODU must have continuous communication with the server. As it is an online service, continuous and immediate communication with the server is prerequisite. Minimization of transmission / reception of data and stabilization of services are essential. **MODU** provides continuous and stable service using docker server.

6.3.3. Development

In the case of **MODU**, it is important to manage documents and artifacts, and it is expected that there will be little or no change in requirements. Therefore, **MODU** is carried out using the waterfall model.

6.4. External Requirements

6.4.1. Regulatory

MODU collects the user's personal information, and if the personal information is not properly protected, there is a possibility of taking civil and criminal responsibilities. Therefore, when signing up, only minimal information necessary for user identification is provided, and sufficient efforts should be made to prevent personal information from being leaked. Objects that are not specified in the user's personal information agreement cannot access the user's personal information.

6.4.2. Ethical

Mutual evaluation of team members, which is conducted for the reliability of team members, can be exploited to give low or high scores without evidence. This will reduce the reliability of the evaluation and should be restrained through regulations. In addition, the intention to promote the progress of non-profit projects by college students can be altered, and posts for job openings may be posted.

6.5. Scenario Examples

6.5.1. Login Scenario

6.5.1.1. Initial Assumption

사용자는 이미 MODU에 가입되어 있으며, 앞서 회원가입시 설정한 E-Mail과 Password로 시스템에 로그인하고자 한다.

6.5.1.2. Normal flow of events

로그인 페이지를 통해 사용자가 E-Mail과 Password를 입력하고 로그인 버튼을 누르면, 시스템은 데이터 베이스와 대조하여 해당 입력 정보가 일치하는지 확인한다.

6.5.1.3. What can go wrong

입력정보가 사전에 유저가 회원가입시 등록했던 정보와 일치하지 않는다면, 올바르지 못한 정보라는 메시지를 띄우고 입력 정보를 초기화 하여 사용자가 올바른 정보를 입력하도록 유도한다.

6.5.1.4. System state on completion

모든 정보가 올바르게 입력되었다면, 사용자는 MODU에 로그인 되어 로그인 된 상태의 홈 페이지로 이동한다.

6.5.2. Project Search Scenario

6.5.2.1. Initial Assumption

사용자는 어떠한 종류의 프로젝트를 찾고자 하는지를 알고 있다.

6.5.2.2. Normal flow of events

사용자는 검색하고자 하는 프로젝트의 내용과 일치하도록 In-Campus 여부, 개발 언어, 개발 인원, 개발 기한을 필터로 설정하고 그 이외의 내용을 태그의 형태로 추가한 후 검색 버튼을 누른다. 이 때, 검색될 프로젝트들의 정렬을 최신 순, 오래된 순, 조회수 순 등의 순서로 설정할 수 있으며, 기본 설정은 최신순으로 한다. 시스템은 입력된 정보와 데이터베이스를 대조하여 일치하는 정보들을 불러온다.

6.5.2.3. What can go wrong

아무런 검색조건 없이 검색 버튼을 누르면, 데이터 베이스에 등록되어 있는 모든 프로젝트를 앞서 설정된 정렬순서에 따라 표시해준다.

데이터 베이스에 검색조건과 일치하는 프로젝트가 존재하지 않는다면, 어떠한 프로젝트도 표시하지 않고 일치하는 프로젝트가 없다는 메시지가 적혀져 있는 결과창으로 이동한다.

6.5.2.4. System state on completion

검색 조건과 일치하는 필터와 태그의 개수를 1순위, 앞서 설정된 정렬 순서를 2순위로 하여 프로젝트들을 표시한 결과창으로 이동한다.

6.5.3. Developer Search Scenario

6.5.3.1. Initial Assumption

사용자는 어떠한 개발자를 찾고자 하는지를 알고 있다.

6.5.3.2. Normal flow of events

사용자는 검색하고자 하는 개발자의 내용과 일치하도록 In-Campus 여부, 학년을 필터로 설정하고 그 이외의 내용을 태그의 형태로 추가한 후 검색 버튼을 누른다. 시스템은 입력된 정보와 데이터베이스를 대조하여 일치하는 정보들을 불러온다.

6.5.3.3. What can go wrong

아무런 검색조건 없이 검색 버튼을 누르면, 데이터 베이스에 등록되어 있는 모든 개발

자를 표시해준다.

데이터 베이스에 검색조건과 일치하는 개발자가 존재하지 않는다면, 어떠한 개발자도 표시하지 않고 일치하는 개발자가 없다는 메시지가 적혀져 있는 결과창으로 이동한다.

6.5.3.4. System state on completion

검색 조건과 일치하는 필터와 태그의 개수를 1순위, 앞서 설정된 정렬 순서를 2순위로 하여 개발자들을 표시한 결과창으로 이동한다.

6.5.4. Invitation Scenario

6.5.4.1. Initial Assumption

사용자는 MODU에 로그인 되어있으며, 완료되지 않은 상태의 1개 이상의 프로젝트를 시스템에 등록하고 있는 상태이다.

6.5.4.2. Normal flow of events

초대하고자 하는 개발자의 상세 페이지의 초대 버튼을 누른다. 그 후 사용자가 시스템에 등록한 프로젝트들이 표시되며, 사용자는 이들 중 해당 개발자를 초대하고자 하는 프로젝트를 선택하여 확인 버튼을 누른다.

6.5.4.3. What can go wrong

사용자가 MODU에 로그인 되지 않은 상태로 초대 버튼을 누른 경우, 로그인 페이지로 이동한다.

6.5.4.4. System state on completion

초대하고자 하는 개발자에게 사용자와 프로젝트의 정보를 포함한 초대 메시지 알림이 보내진다.

6.5.5. Participation Inquiry Scenario

6.5.5.1. Initial Assumption

사용자는 MODU에 로그인 되어있는 상태이며, 참가하고자 하는 프로젝트를 확인한 상태이다.

6.5.5.2. Normal flow of events

참가하고자 하는 프로젝트의 상세 페이지의 참가 문의 버튼을 누른다.

6.5.5.3. What can go wrong

사용자가 MODU에 로그인 되지 않은 상태로 참가 문의 버튼을 누른 경우, 로그인 페이지로 이동한다.

해당 프로젝트의 개발 인원이 이미 만원이거나 이미 프로젝트가 완료된 경우 등 더이상 인원 모집 상태가 아닐 경우, 참가 문의를 할 수 없다는 내용의 메시지를 표시한다.

6.5.5.4. System state on completion

참가하고자 하는 프로젝트를 MODU에 등록한 개발자에게 사용자의 정보를 포함한 참가 문의 메시지 알림이 보내진다.

6.5.6. Peer Assessment Scenario

6.5.6.1. Initial Assumption

사용자는 MODU에 로그인 된 상태이며, MODU를 통해 프로젝트를 완료한 상태이다.

6.5.6.2. Normal flow of events

사용자는 Personal page를 통하여 나의 프로젝트 현황 목록에서 상호평가가 필요한 프로젝트를 클릭하여 프로젝트 관리 페이지로 이동한다. 해당 페이지에서 상호 평가 버튼을 클릭하면, 사용자를 제외한 프로젝트 팀원의 리스트가 표시된다. 리스트에서 평가하고자 하는 팀원 선택하여 개별 평가 페이지로 이동한다. 그 후, 프로젝트를 진행한 팀원들의

이름을 클릭하면 해당 팀원의 개별 평가 페이지로 이동한다. 개별 평가 페이지에서는 아이디어, 개발능력, 커뮤니케이션 능력, 총 기여도의 4가지 항목을 최대 5점의 별표 형식으로 평가하게 된다. 마지막으로 해당 팀원에 대한 짧은 코멘트를 쓰고 모든 평가가 완료되면 평가 완료 버튼을 클릭한다.

6.5.6.3. What can go wrong

사용자가 프로젝트를 혼자서 진행하였을 경우, 상호 평가 페이지에 평가할 팀원이 없다는 내용을 표시한다.

사용자가 모든 평가를 완료하지 않고 평가 완료 버튼을 누른 경우, 평가가 완료되지 않았다는 메시지를 표시한다.

6.5.6.4. System state on completion

평가된 내용은 해당 개발자의 사용자 정보에 저장되며, 이후 해당 개발자의 상세페이지에 표시될 강점과 약점에 반영된다.

6.5.7. Portfolio Management Scenario

6.5.7.1. Initial Assumption

사용자는 MODU에 로그인 된 상태이며, MODU에 저장된 자신의 정보를 확인하고 이를 편집하고자 한다.

6.5.7.2. Normal flow of events

사용자 자신의 Portfolio 페이지에서 편집하고자 하는 항목의 수정 버튼을 클릭한다. 사용자는 자격증, 수료증, 수상 경력, 재학정보, 활동 이력, 간략 소개 등의 내용을 수정할 수 있으며, 편집이 완료되면 수정 완료 버튼을 클릭한다.

6.5.7.3. What can go wrong

필수 입력 항목의 내용들을 삭제하고 수정 완료 버튼을 누른다면, 해당 사항에 관한 내용이 채워지지 않았다는 메시지를 표시한다.

6.5.8. System state on completion

편집이 완료되면 수정된 내용을 사용자 정보에 저장하고, 사용자의 개발자 상세 페이지로 이동한다.

7. System Models

In this chapter, the entire system, the relationship between each system component, and the environment surrounding the system is described through various diagrams.

7.1. Entire System

7.1.1. Context Diagram

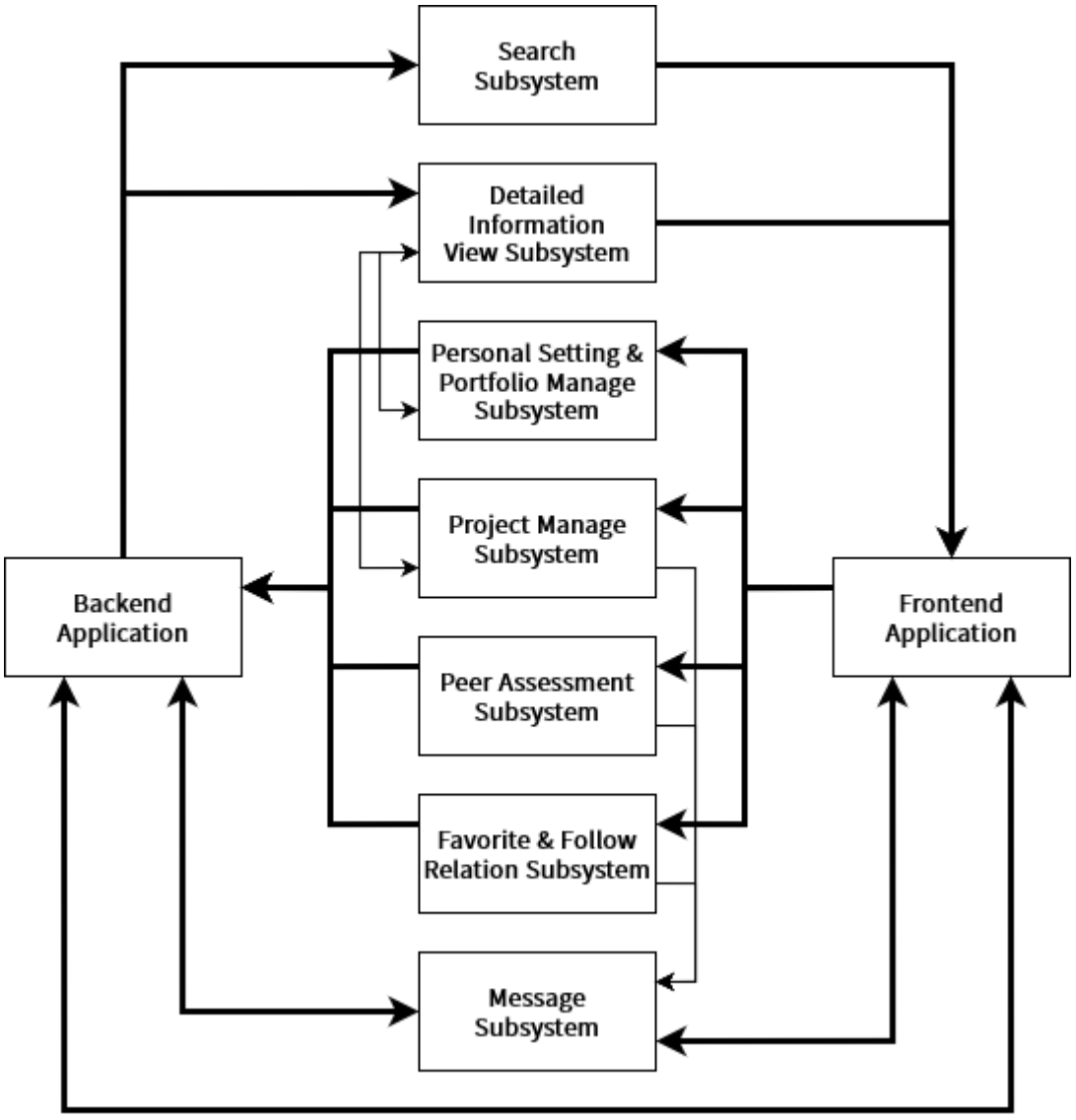


Figure 13. Context Diagram

7.1.2. Process Diagram

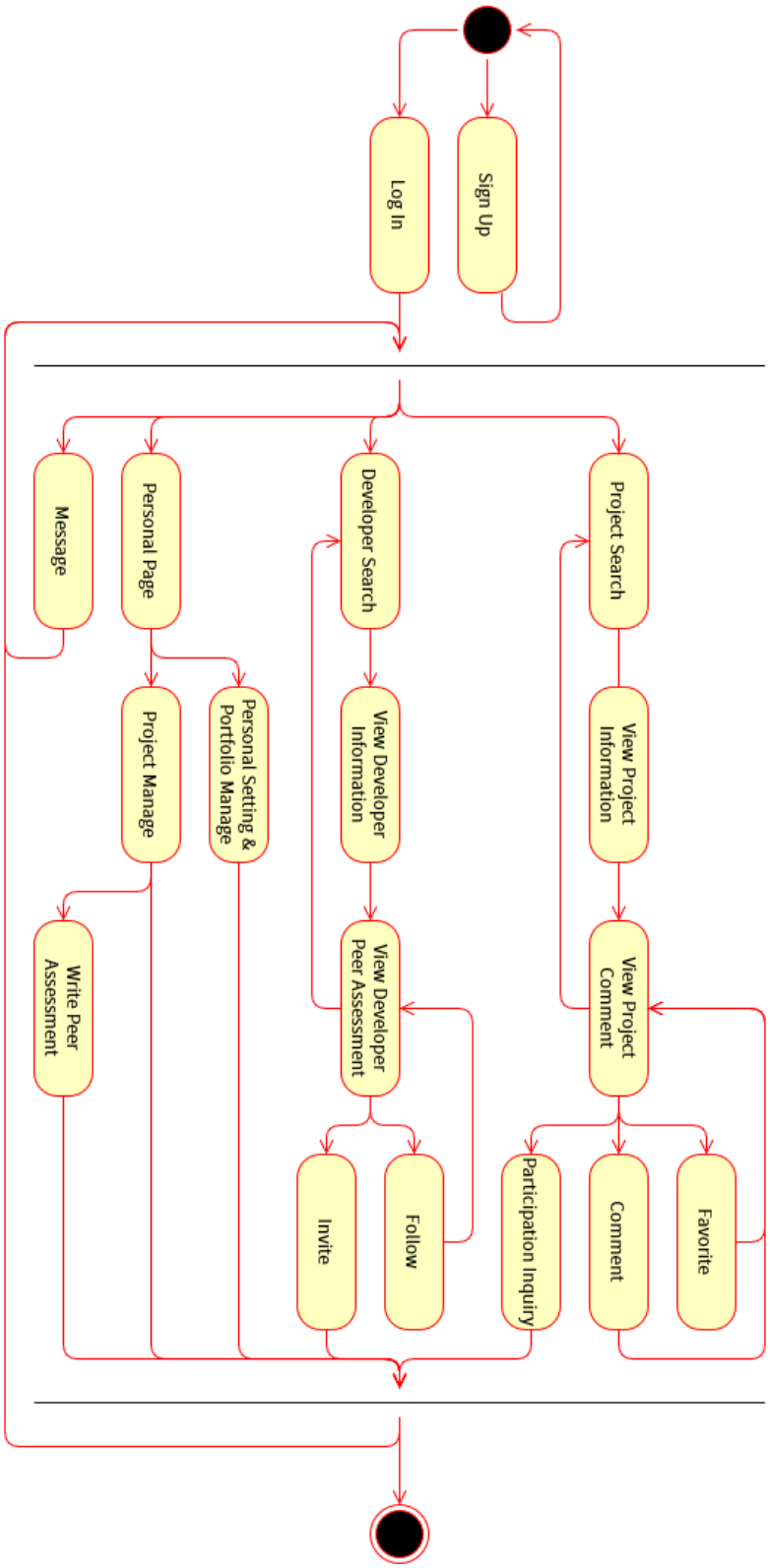


Figure 14. Process Diagram

7.1.3. Use Case Diagram

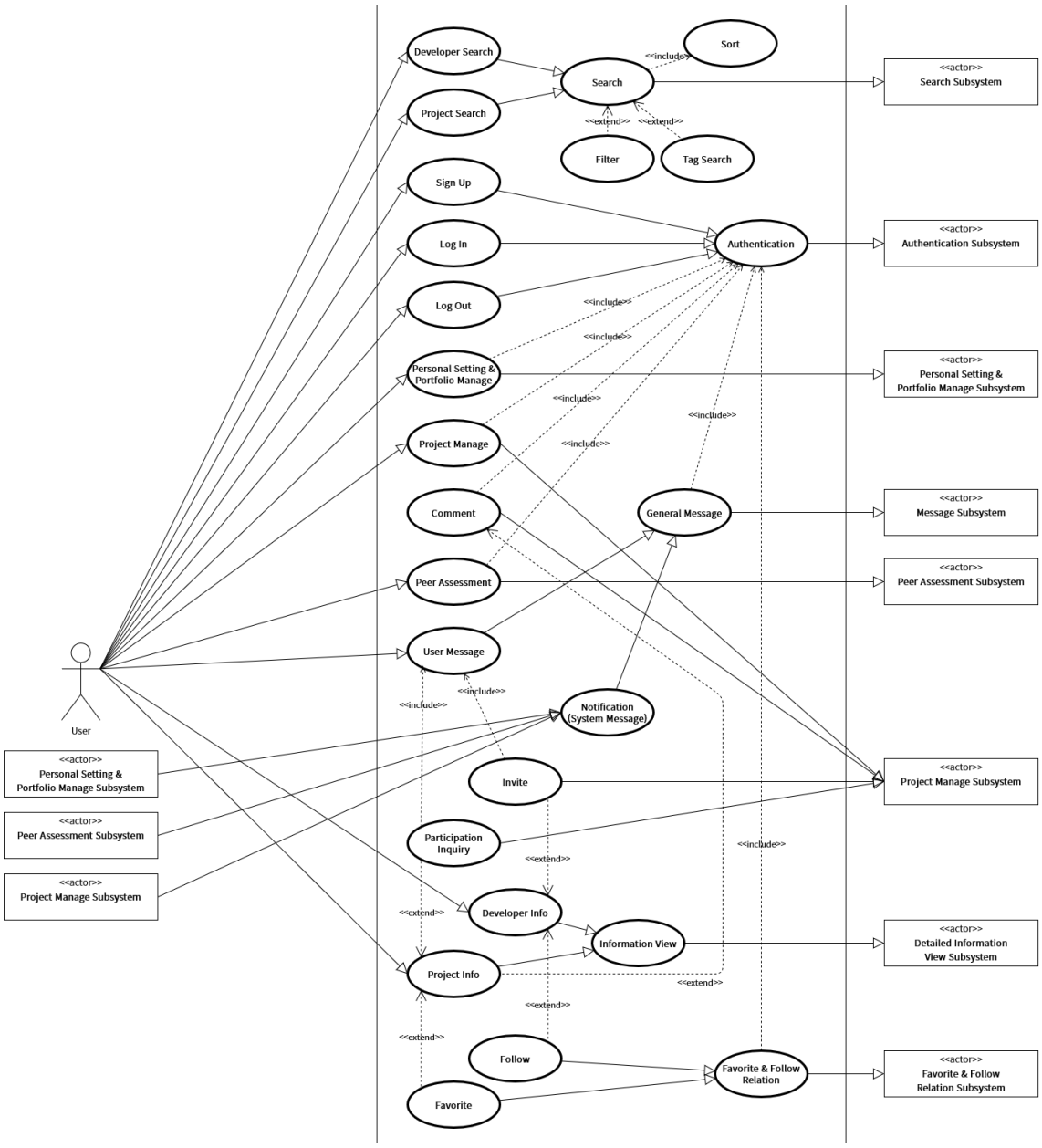


Figure 15. Use Case Diagram

7.1.4. Tabular Description of Use Case Diagram

7.1.4.1. Project Search Use Case Diagram

Table 12. Project Search

Use case	Project Search
Actor	User, Search subsystem
Description	Search the project through keywords, tags and filters for the programing language, project scale, project duration, etc.
Trigger	User enters search condition and clicks the search button.
Success Response	Transform the search condition into a query, search project DB, and print a list of matching projects.
Failure Response	If there are no projects in project DB that match the search condition, a message is displayed indicating that there are no projects corresponding to the search condition with empty items.

7.1.4.2. Developer Search Use Case Diagram

Table 13. Developer Search

Use case	Developer Search
Actor	User, Search subsystem
Description	Search the developer through keywords, tags and filters.
Trigger	User enters search condition and clicks the search button.
Success Response	Transform the search condition into a query, search Developer DB, and print a list of matching developer.
Failure Response	If there are no developers in Developer DB that match the search condition, a message is displayed indicating that there are no developers corresponding to the search condition with empty items.

7.1.4.3. Sign Up Use Case Diagram

Table 14. Sign Up

Use case	Sign Up
Actor	User, Authentication subsystem
Description	This is the process which the user updates his information in Account DB for registration.
Trigger	User enters E-Mail and Password.
Success Response	After verifying the entered email, the user information registration is activated in Account DB.
Failure Response	If the E-Mail or Password is incorrect or the E-Mail is already registered in Account DB, display an error message.

7.1.4.4. Log In Use Case Diagram

Table 15. Log In

Use case	Log In
Actor	User, Authentication subsystem
Description	This is the process of determining whether the information that the user entered when signing up matches the information in Account DB to access the system.
Trigger	User enters E-Mail and Password.
Success Response	Check whether the entered user information matches the information in Account DB and grants access permission to the user.
Failure Response	If the E-Mail format is incorrect or there is no matching information in Account DB, display an error message.

7.1.4.5. Log Out Use Case Diagram

Table 16. Log Out

Use case	Log out
Actor	User, Authentication subsystem
Description	The process of releasing the log in state to disconnect from the system.
Trigger	The user clicks the Log out button.
Success Response	After the user returns the access permission to the system, the user goes to the home page.
Failure Response	

7.1.4.6. Personal Setting & Portfolio Manage Use Case Diagram

Table 17. Personal Setting & Portfolio Manage

Use Case	Personal Setting & Portfolio Manage
Actor	User, Personal Setting & Portfolio Manage Subsystem, Authentication Subsystem
Description	Each user's own portfolio is managed, and the users can check their peer assessment scores and reviews.
Trigger	The MODU 어땠니 page would show all the previously existing Portfolios. In prior to adding/editing the Portfolio page, the user account authentication would be needed as an argument of API call. After that, the user can add/edit their Portfolio page.
Success Response	The previously existing Portfolios would be managed. Changes in the Portfolio would update the Portfolio Page and the Developer DB. If another user has "followed" the student developer, the user would get a notification, and the Message DB would add this change.
Failure Response	If a user is not authorized, or if the Portfolio information added/uploaded is inappropriate, an Error message would occur.

7.1.4.7. Project Manage Use Case Diagram

Table 18. Project Manage

Use Case	Project Manage
Actor	User, Project Manage Subsystem, Authentication Subsystem
Description	Each of the project information are managed, and the users can conduct peer assessment when a project is completed.
Trigger	The MODU 모여라 page would show all the previously existing Projects. In prior to adding a new Project, the user account authentication would be needed as an argument of API call. After that, the user is able to upload new Project titles with specific Project information according to the format. After each project is conducted, the users are able to do Peer Assessment about other student developers who they worked with.
Success Response	Project Upload- The previously existing Projects would be managed. New Projects would be uploaded and the Project DB would be updated. Peer Assessment- The previously existing Peer Assessments would be managed. New Peer Assessments would be uploaded and the Assessment DB would be updated. A user who has gotten a new Peer Assessment would get a notification, and the Message DB would be updated.
Failure Response	Project Upload - If a user is not authorized, or if the user is trying to upload a new project but did not fill all the specific information required in the Project format, an Error message would occur. Peer Assessment – If a user is trying to upload a Peer Assessment but did not previously conduct a Project with the person he/she is evaluating, an Error message would occur.

7.1.4.8. Comment Use Case Diagram

Table 19. Comment

Use Case	Comment
Actor	User, Project Manage Subsystem, Authentication Subsystem
Description	Users are able to Comment each Project when they have a question.
Trigger	In prior to adding a Comment, the user account authentication would be needed as an argument of API call. When they find a project that they are interested in, the user may ask questions by adding a Comment under each Project Information.
Success Response	A Comment would be added under the Project of interest, and the Project DB would be updated. If another user has "favorited" the Project, the user would get a notification, and the Message DB would add this change.
Failure Response	If the user is not authorized, or if the comment is inappropriate, or if the Project for which the user is trying to add a comment does not exist, an Error message would occur.

7.1.4.9. Peer Assessment Use Case Diagram

Table 20. Peer Assessment

Use Case	Peer Assessment
Actor	User, Peer Assessment Subsystem, Authentication Subsystem, Message Subsystem
Description	Peer assessment is done to project. Evaluate user's performance once the project is completed.
Trigger	User clicks peer assessment button right beside the names of each ex-teammates.
Success Response	A peer assessment would be uploaded and updated to the Assessment DB. Star rates and reviews of peers are successfully entered and recorded. Making an assessment will trigger the system to notify the peer assessment update to the subject. Thus the Message DB is updated.
Failure Response	If the review has an input in inappropriate form, then do not upload the assessment, and send error message.

7.1.4.10. User Message Use Case Diagram

Table 21. User Message

Use Case	User Message
Actor	User, Message Subsystem, Authentication Subsystem
Description	User has a message to send or receive.
Trigger	User has a message to send or receive.
Success Response	Once it succeeds, it is sync into the Message DB.
Failure Response	If the Target Developer Unique ID do not exist, it fails. The authentication fails. A failure message will be shown.

7.1.4.11. Notification Use Case Diagram

Table 22. Notification

Use Case	System Notification
Actor	Peer Assessment Subsystem, Project Manage Subsystem, Message Subsystem, Personal Setting & Portfolio Manage Subsystem, Authentication Subsystem
Description	A message is shown in the notification box from the system.
Trigger	When system sends out notification, a notification message would be shown in the notification box.
Success Response	Once it succeeds, it is uploaded into Message DB. Summary of additional message will appear in the notification box.
Failure Response	If the authentication fails, then show an error message.

7.1.4.12. Invite Use Case Diagram

Table 23. Invite

Use Case	Invite Developers to the Project
Actor	Project Manage Subsystem, Authentication Subsystem
Description	It includes User Message. It extends Developer Information.
Trigger	A Project Proposer clicks “초대” Invite button on Developer Information View Page.
Success Response	Invitation greeting message is sent to the developer. Check validation of target developer and project by their unique IDs in Account DB and Project DB, respectively. Update in the Message DB.
Failure Response	If the inviter ID does not exist, it fails. If the Target Developer Unique ID does not exist, it fails. If the Target Unique ID does not exist, it fails. Authentication fails. Show an error message.

7.1.4.13. Participation Inquiry Use Case Diagram

Table 24. Participation Inquiry

Use Case	Send Participation Inquiry to the Project Proposer
Actor	Project Manage Subsystem, Authentication Subsystem
Description	It includes User Message. It extends Project Information.
Trigger	User clicks “참가문의” Participation Inquiry button on Project Information View Page.
Success Response	Inquiry message is sent to the developer. Check validation of target project by its unique ID compared to the Project DB. If valid, update in the Message DB.
Failure Response	If the Target Unique ID does not exist, it fails. Authentication fails. Show an error message.

7.1.4.14. Developer Info Use Case Diagram

Table 25. Developer Info

Use case	Developer Info
Actor	User, Detailed Information View Subsystem
Description	User can check the detailed information of a specific developer.
Trigger	User clicks a specific developer on developer search.
Success Response	Using the clicked developer ID, information of the developer with the corresponding ID is retrieved from the developer database and assessment database
Failure Response	If the required ID does not exist in developer database, send an error message.

7.1.4.15. Project Info Use Case Diagram

Table 26. Project Info

Use case	Project Info
Actor	User, Detailed Information View Subsystem
Description	User can check the detailed information of a specific project.
Trigger	User clicks a specific project on project search.
Success Response	The project information is loaded from the project database using the corresponding project ID. Also, information of project participants is retrieved and displayed from project database and developer database.
Failure Response	If the required ID does not exist in project database, send an error message.

7.1.4.16. Favorite Use Case Diagram

Table 27. Favorite

Use case	Favorite
Actor	User, Favorite & Follow Relation Subsystem, Authentication Subsystem
Description	The user can register a specific project in the user's project list of interest.
Trigger	User clicks the interest button of a specific project.
Success Response	User ID is registered in the interested developer list of project database.
Failure Response	If the user ID or project ID do not exist in developer database and project database each, send an error message. Also, if the user's authentication is failed, do not register Favorite and send an error message.

7.1.4.17. Follow Use Case Diagram

Table 28. Follow

Use case	Follow
Actor	User, Favorite & Follow Relation Subsystem, Authentication Subsystem
Description	The user can register a specific developer in the user's developer list of interest.
Trigger	User clicks the interest button of a specific developer.
Success Response	The developer ID that the user wants to register in the interest list is retrieved from the developer database and registered in the interest list of the user. Developer database.
Failure Response	If the developer ID does not exist in developer database, send an error message. Also, if the user's authentication is failed, do not register Follow and send an error message.

7.2. Subsystems

7.2.1. Authentication System – Sign Up Data Flow Diagram

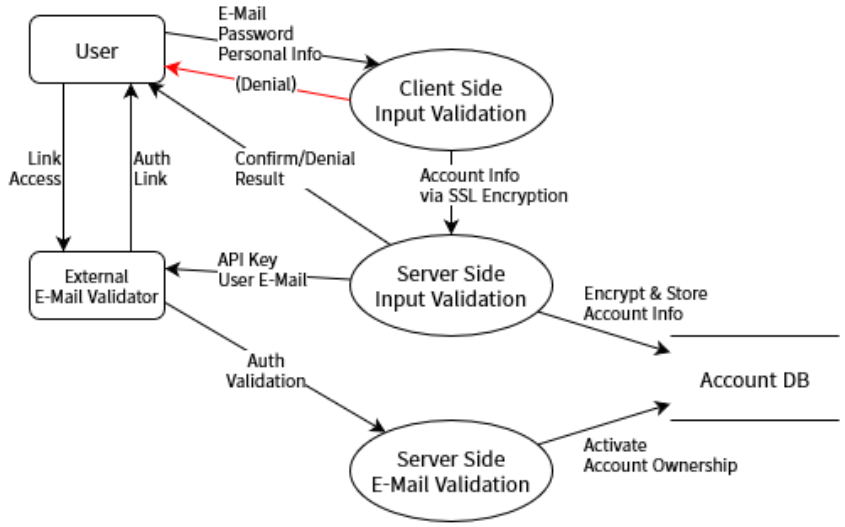


Figure 16. Authentication System DFD

7.2.2. Search System – Data Flow Diagram

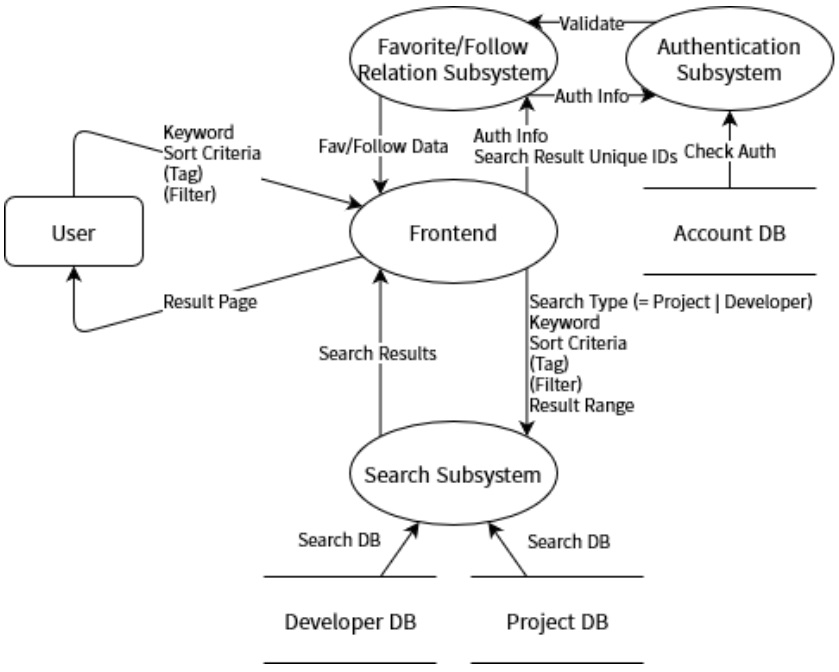


Figure 17. Search System DFD

7.2.3. Project Information View System – Data Flow Diagram

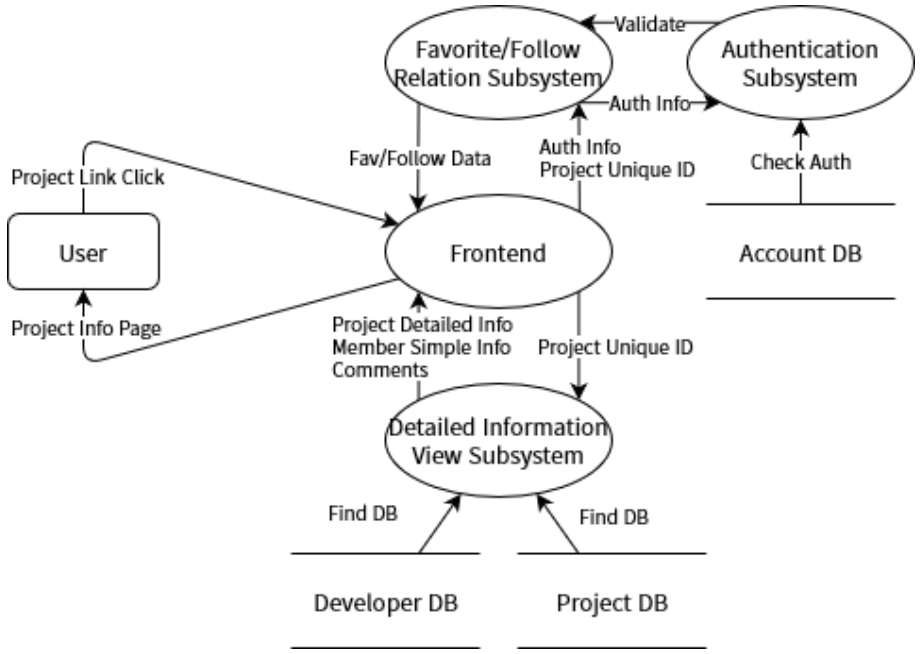


Figure 18. Project Information View System DFD

7.2.4. Developer Information View System – Data Flow Diagram

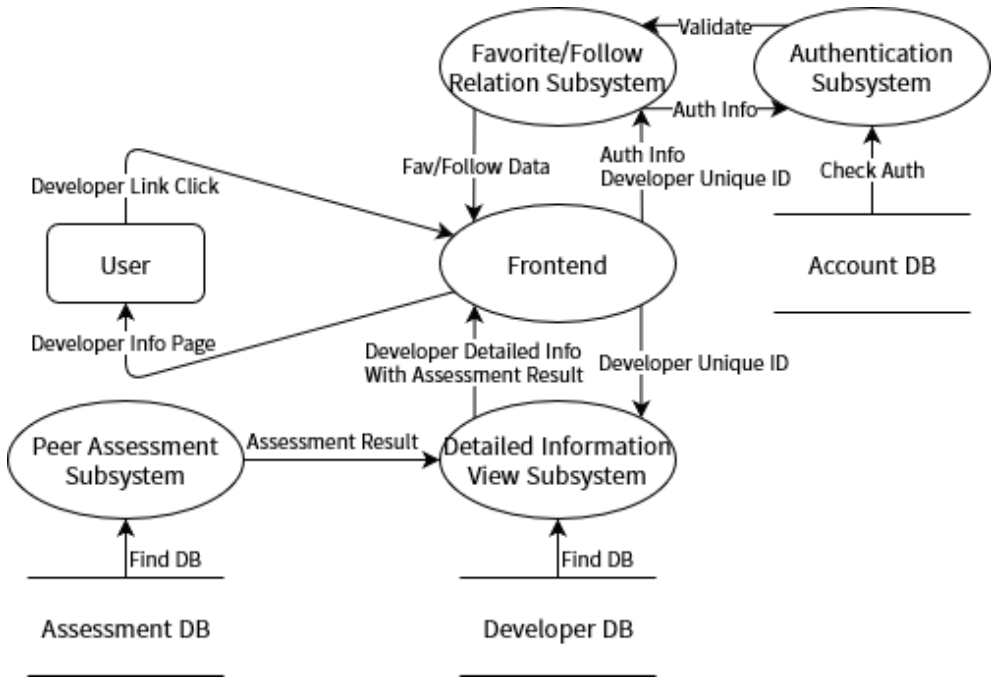


Figure 19. Developer Information View System DFD

7.2.5. Invite System – Data Flow Diagram

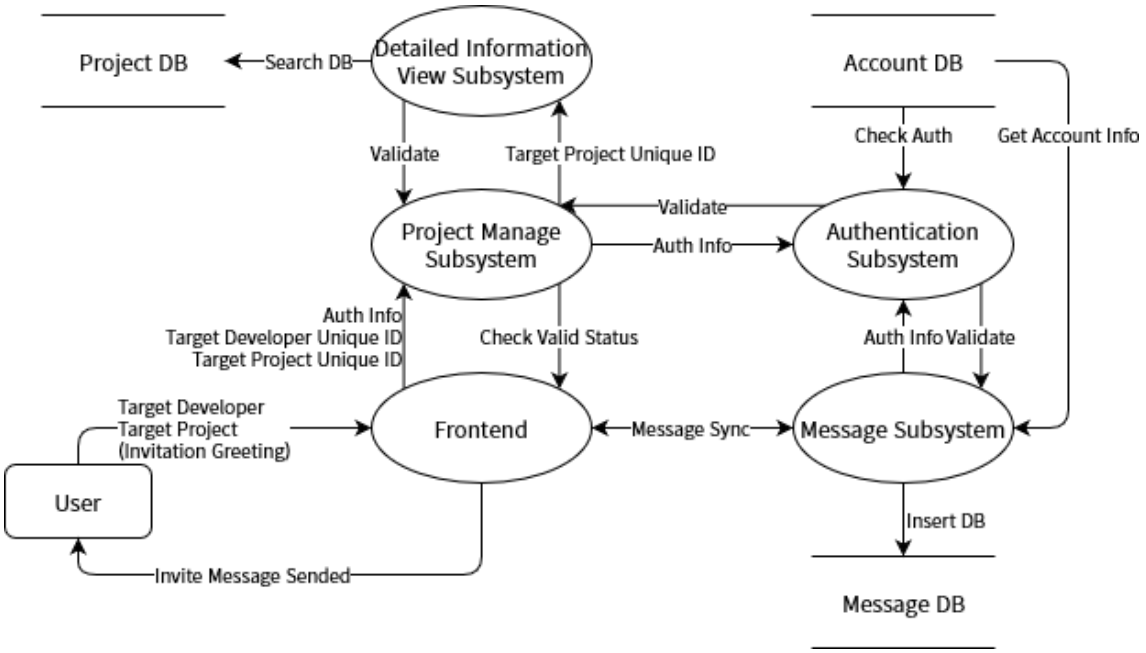


Figure 20. Invite System DFD

7.2.6. Participation Inquiry System – Data Flow Diagram

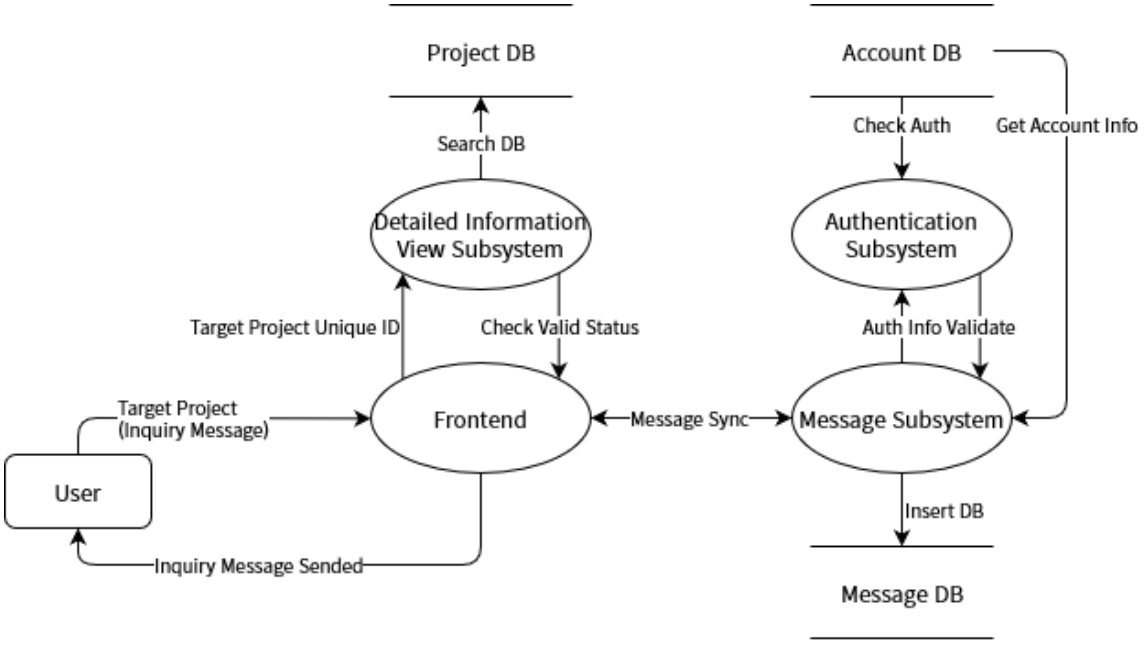


Figure 21. Participation Inquiry System DFD

7.2.7. Message System and Notification System – Sequence Diagram

7.2.7.1. Message System

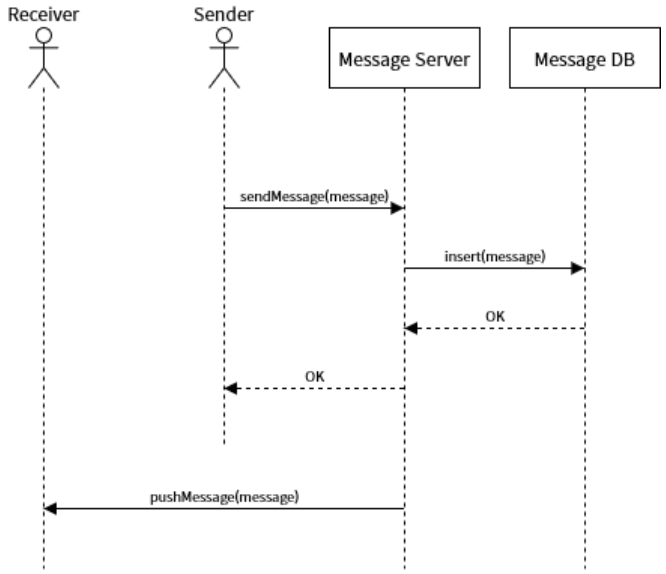


Figure 22. Message System Sequence Diagram

7.2.7.2. Notification System

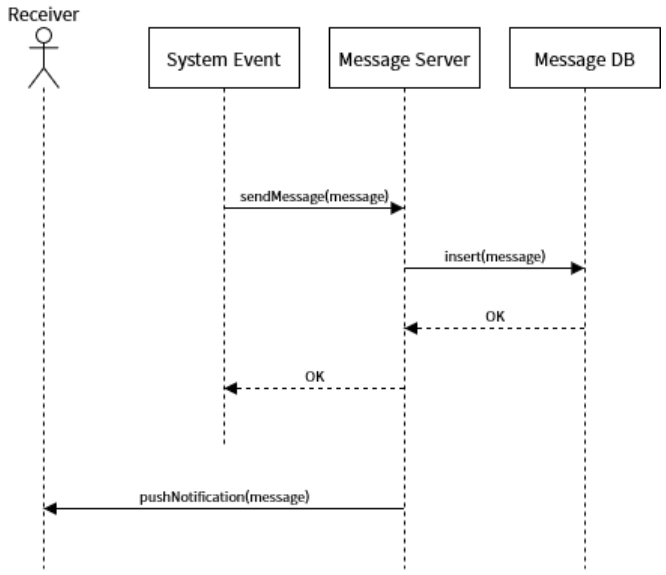


Figure 23. Notification System Sequence Diagram

Both message system and notification system have similar sequence, but the message server can distinguish internal system event such as peer assessments and comments from other users, so in case of a system event, a notification would be sent rather than a message.

8. System Evolution

This chapter describes the environment surrounding the system, how to expect the system to respond to the anticipated changes, and to anticipate the various changes surrounding the system that may occur while deploying and operating the system. This eliminates the possibility of side-by-side design changes that may occur in future modifications to the system and lowers the cost of modification.

8.1. Linkage with LinkedIn and GitHub

MODU provides its own portfolio management services. However, members who were using other existing portfolio management services may feel burdened with creating a new portfolio. Therefore, it is possible to conveniently link and manage portfolios or codes through linkage with own account in LinkedIn and GitHub, beyond the existing portfolio management. This will increase the satisfaction and convenience of MODU members, thereby increasing the use of services.

8.2. Extension to mobile applications

MODU is a PC web-based service. However, if the system is made with the more intuitive and simpler user interface, this system can also be considered extending to a mobile application with good accessibility. Since MODU members can use the service through both PC and mobile platforms, they can increase their service use and increase their satisfaction.

8.3. Report of peer assessment

Peer assessment can be an indicator of how reliable user can be to the relevant MODU members in future projects. However, if these evaluations are exploited, they may be misrepresented or given due to personal feelings or reasons not related to the project. This can lead to an overall drop in the confidence in the MODU member profile, so to prevent this false peer assessment, a reporting system can be introduced so that administrators can penalize members based on specific regulations.

8.4. Project analysis

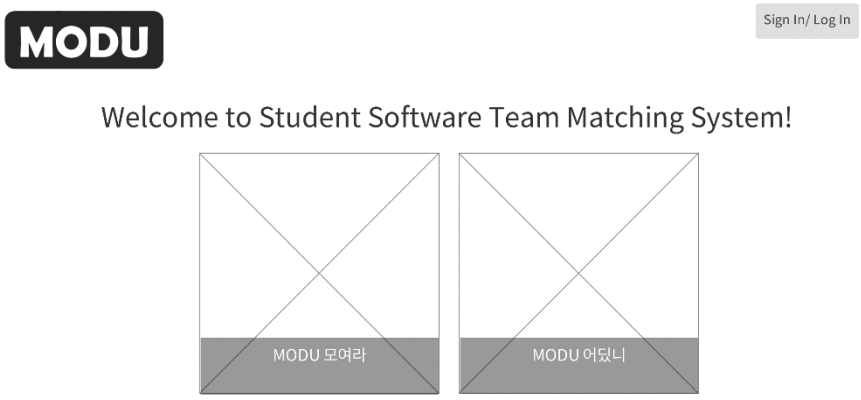
If project invitation is not going well, the project proposer may wonder why. There may be various factors for this reason. First, the project may not be exposed at the top of the search result, or the subject

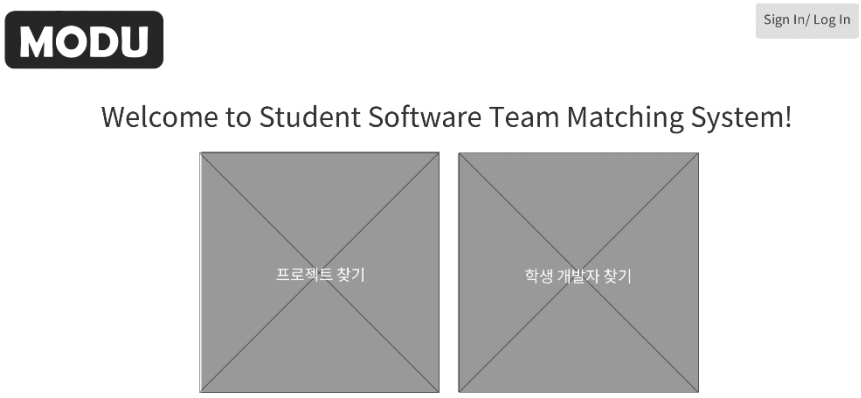
of the project may not suit the taste of users. To provide the most probable reason among these various reasons, MODU compares statistical numbers such as the number of search impressions of a project, comparison with other projects, and the number of interests of members, and provide the most probable result to the project proposer.

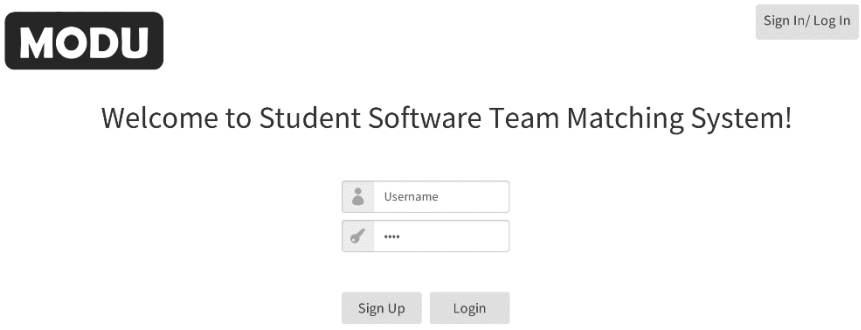
8.5. Providing competition information based on natural language processing

Sometimes members work on personal projects, but sometimes they do projects for competitions. MODU can crawl and provide information about these competitions from other sites. At this time, using natural language processing, the system will be able to automatically give more information about competitions to the MODU members.

9. Appendices



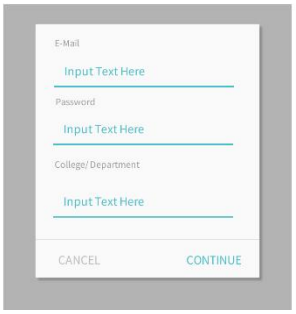




MODU

Sign In/ Log In

Welcome to Student Software Team Matching System!



MODU

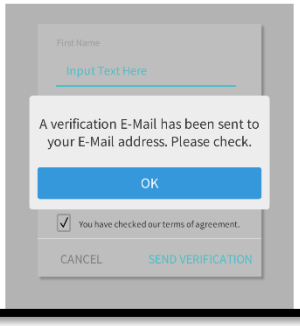
Sign In/ Log In

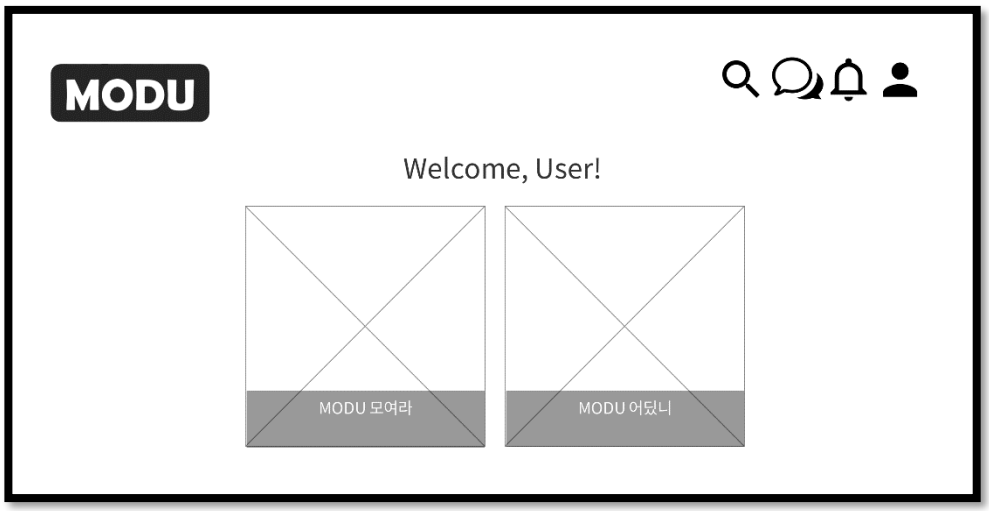
Welcome to Student Software Team Matching System!

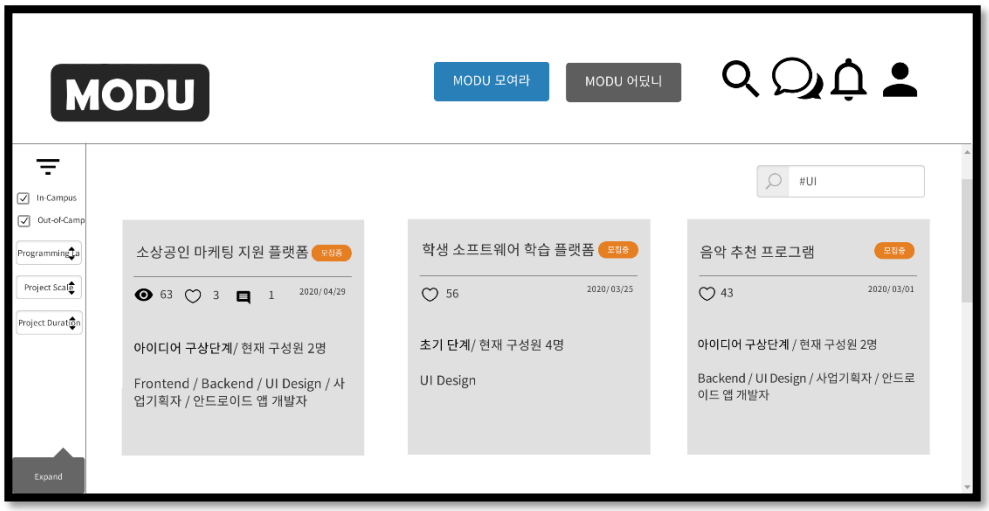
MODU

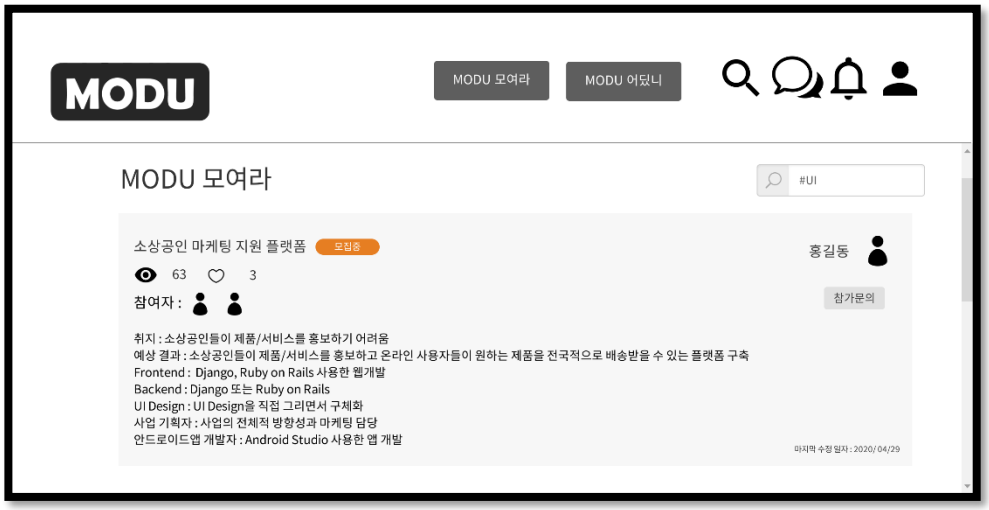
Sign In/ Log In

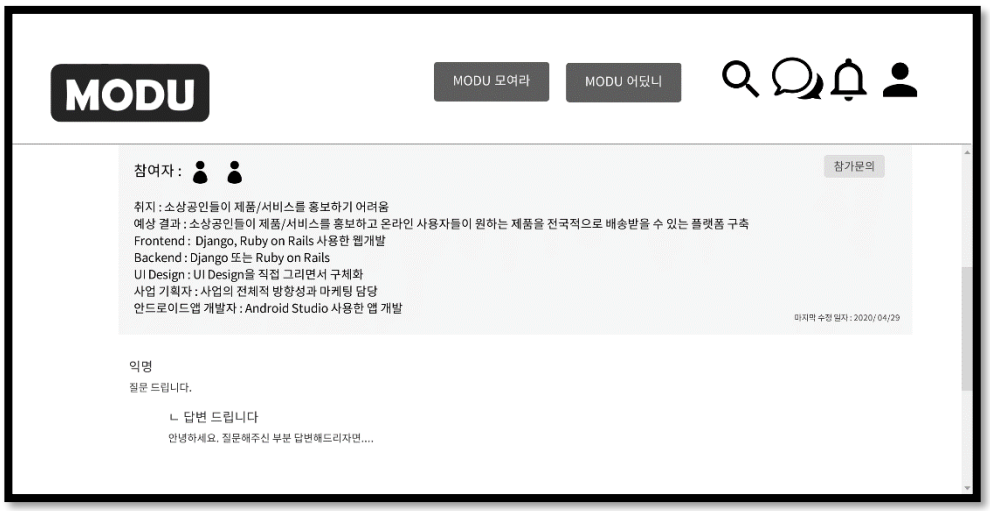
Welcome to Student Software Team Matching System!

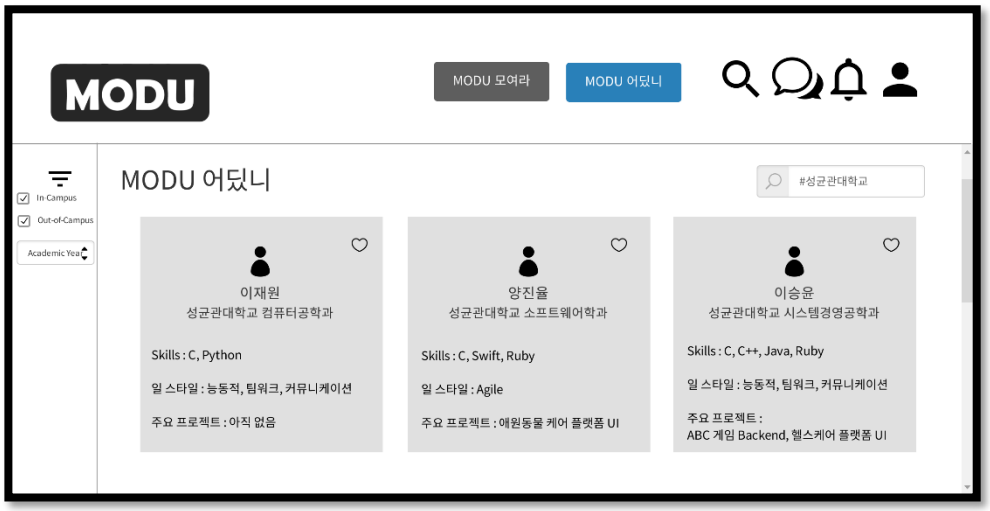


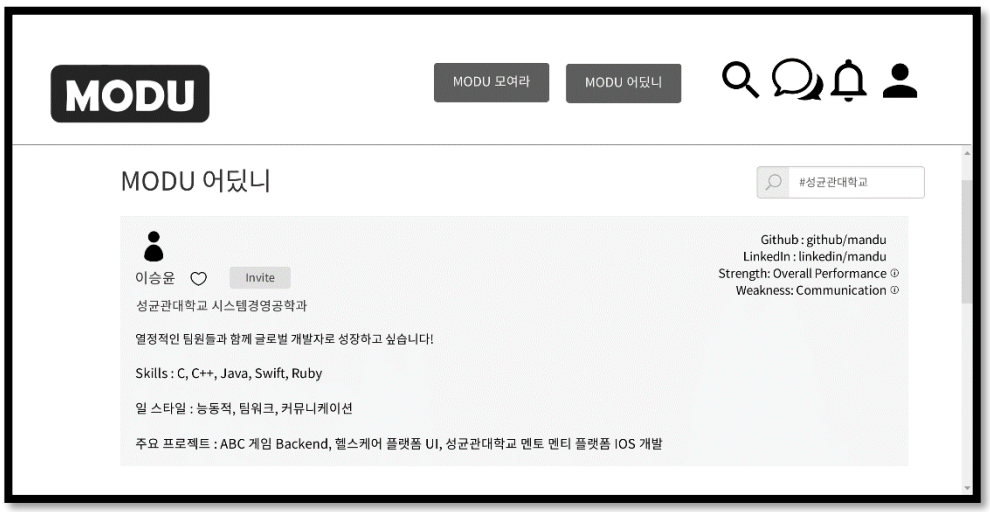

MODU
Welcome, User!
MODU 모여라
MODU 어딨니

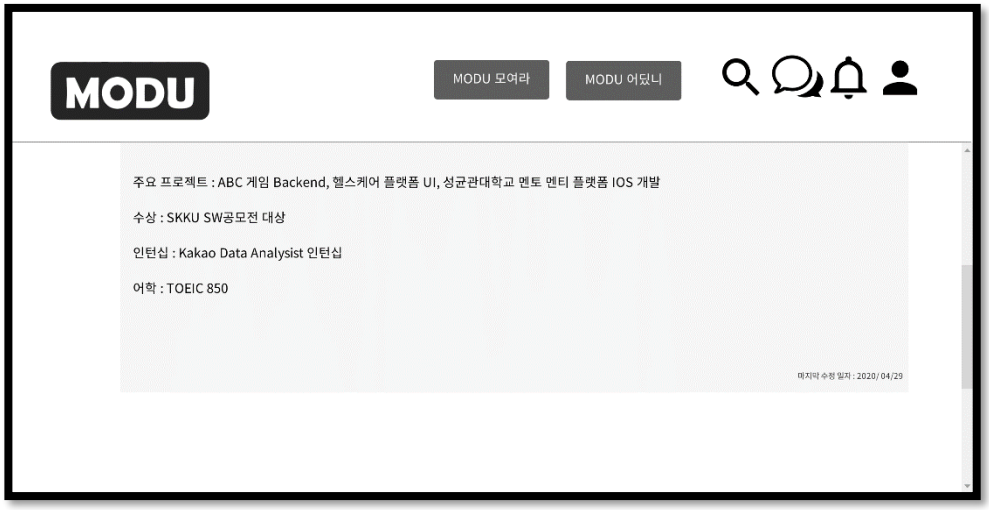

MODU
MODU 모여라
MODU 어딨니
In Campus
Out-of-Camp
Project Scal
Project Durat
Expand
#UI
소상공인 마케팅 지원 플랫폼
모집중
63
3
1
2020/04/29
아이디어 구상단계/ 현재 구성원 2명
Frontend / Backend / UI Design / 사업기획자 / 안드로이드 앱 개발자
학생 소프트웨어 학습 플랫폼
모집중
56
2020/03/25
초기 단계/ 현재 구성원 4명
UI Design
음악 추천 프로그램
모집중
43
2020/03/01
아이디어 구상단계 / 현재 구성원 2명
Backend / UI Design / 사업기획자 / 안드로이드 앱 개발자


MODU
MODU 모여라
MODU 어딨니
MODU 모여라
#UI
소상공인 마케팅 지원 플랫폼
모집중
63
3
참여자 :
홍길동
참가문의
취지 : 소상공인들이 제품/서비스를 홍보하기 어려움
예상 결과 : 소상공인들이 제품/서비스를 홍보하고 온라인 사용자들이 원하는 제품을 전국적으로 배송받을 수 있는 플랫폼 구축
Frontend : Django, Ruby on Rails 사용한 웹개발
Backend : Django 또는 Ruby on Rails
UI Design : UI Design을 직접 그리면서 구체화
사업 기획자 : 사업의 전체적 방향성과 마케팅 담당
안드로이드앱 개발자 : Android Studio 사용한 앱 개발
마지막 수정 일자 : 2020/04/29


MODU
MODU 모여라
MODU 어딨니
참여자 :
참가문의
취지 : 소상공인들이 제품/서비스를 홍보하기 어려움
예상 결과 : 소상공인들이 제품/서비스를 홍보하고 온라인 사용자들이 원하는 제품을 전국적으로 배송받을 수 있는 플랫폼 구축
Frontend : Django, Ruby on Rails 사용한 웹개발
Backend : Django 또는 Ruby on Rails
UI Design : UI Design을 직접 그리면서 구체화
사업 기획자 : 사업의 전체적 방향성과 마케팅 담당
안드로이드앱 개발자 : Android Studio 사용한 앱 개발
마지막 수정 일자 : 2020/ 04/29
익명
질문 드립니다.
ㄴ 답변 드립니다
안녕하세요. 질문해주신 부분 답변해드리자면....


MODU
MODU 모여라
MODU 어딨니
In Campus
Out-of-Campus
Academic Yea
MODU 어딨니
#성균관대학교
이재원
성균관대학교 컴퓨터공학과
Skills : C, Python
일 스타일 : 능동적, 팀워크, 커뮤니케이션
주요 프로젝트 : 아직 없음
양진율
성균관대학교 소프트웨어학과
Skills : C, Swift, Ruby
일 스타일 : Agile
주요 프로젝트 : 애원동물 케어 플랫폼 UI
이승윤
성균관대학교 시스템경영공학과
Skills : C, C++, Java, Ruby
일 스타일 : 능동적, 팀워크, 커뮤니케이션
주요 프로젝트 :
ABC 게임 Backend, 헬스케어 플랫폼 UI


MODU
MODU 모여라
MODU 어딨니
MODU 어딨니
#성균관대학교
이승윤
Invite
성균관대학교 시스템경영공학과
Github : github/mandu
LinkedIn : linkedin/mandu
Strength: Overall Performance
Weakness: Communication
열정적인 팀원들과 함께 글로벌 개발자로 성장하고 싶습니다!
Skills : C, C++, Java, Swift, Ruby
일 스타일 : 능동적, 팀워크, 커뮤니케이션
주요 프로젝트 : ABC 게임 Backend, 헬스케어 플랫폼 UI, 성균관대학교 멘토 멘티 플랫폼 IOS 개발


MODU
MODU 모여라
MODU 어딨니
주요 프로젝트 : ABC 게임 Backend, 헬스케어 플랫폼 UI, 성균관대학교 멘토 멘티 플랫폼 IOS 개발
수상 : SKKU SW공모전 대상
인턴십 : Kakao Data Analysist 인턴십
어학 : TOEIC 850
마지막 수정 일자 : 2020/ 04/29

10. Index

10.1. Figures

10.2. Tables

11. References

<톡톡 튀는 원자력 아이디어 나왔지만...참가자 반토막 난 원자력 공모전>

http://dongascience.donga.com/news/view/29793

<추락하는 학내 공모전 참가율, 해법은 없는가>

https://www.hanyang.ac.kr/web/www/hanyangnews?p_p_id=newsView_WAR_newsportlet&p_p_lifecycle=0&p_p_state=normal&p_p_mode=view&p_p_col_id=column1&p_p_col_count=1&_newsView_WAR_newsportlet_action=view_message&_newsView_WAR_newsportlet_messageId=2053

<학내 프로그램, 경쟁은 부재중?>

http://www.jbpresscenter.com/news/articleView.html?idxno=2160