SYSTEM DEVELOPMENT REPORT

Integrated Information System for Dongguk University's Item rental service by using web service.

Jiwon Seo, Byeonghwa Kim, Jan Bednar, Snizhana Liashevska, Taeseong Park

Industrial and Systems Engineering, Dongguk University-Seoul

06. 16. 2022

Information System Integration and Practice

Summary

The background and purpose of the Team 1 project are to improve the existing inconvenience Dongguk University's item rental service. To develop a system in this project, we will define problems at the Planning phase and conduct an identifying business value through Technical Feasibility, Economic Feasibility, and Organizable Feasibility in Feasibility analysis. Also, we will do planning the project. In the Analysis phase, Use Case will be used, and Functional Requirements, Data Requirements, Performance Requirements, Maintenance and Supportability Requirements will be analyzed. In the Design phase, we will design the architecture of the new system and will do User Interface Design using Java Swing and SWT tool. And we will also create simple functions through Object Design. In the implementation phase, we will implement an actual web service using AWS, Django, MySQL, and SOAP. Through these processes, we will develop efficient web services.

Key words: Web Service, Information System, Database, Date Store, Interface.

Table of Contents

List of Tables

List of Figures

List of Equations

System Development Report Template

Proposed by Dr. Sojung Kim

1. Project Description

1.1. Purpose of the Project

Topic: Create a system that allows students to easily make reservations and check inventory quantity when lending Items at Dongguk University CS Center, Bachelor's Office of College, and Student Council of College.

Purpose: As described in the background, it is possible to reduce the trouble of walking in vain by identifying the inventory quantity in advance when renting items, and to make a reservation for rental items in advance when it is an important item that must be used. Also, from the point of view of the rental center managing the goods, it is to reduce the cost of loss and to facilitate the management of the items.

There are many places to rent necessary items at Dongguk University. There is the CS center of the main building, the bachelor's office of each college, and the student council room of each college (“Dongguk University CS Center”, n.d.). The project is to create a system and service that allows users to easily make reservations with smartphones and determine the number of items that can be rented when renting items at Dongguk University's item rental center. This idea originated from the library information system. The library information system is a web service that allows users to identify the number of books that can be borrowed from a library web page and make reservations for borrowing book. Like this service, it is possible to identify the place where necessary items can be rented and the number of items, thereby reducing the cost of loss and facilitating the management of goods even when the center manages the items. It will increase the convenience of students by increasing the accessibility of commodity rental services and will lead to a decrease in the loss rate of rental items.

1.2. Scope of the Work

Background: The Dongguk university’s item rental service is really useful service for students, but there are some inconveniences. When we try to rent an item, we can't check whether the quantity of the rental item is in advance, so we may go in vain. Usually then we will go to another rental center, but the other center may not have the item we are looking for in stock. Also, each rental center has different types of items, which may cause confusion. For example, we can borrow a smartphone charger from the CS Center, but not from the College of Engineering student council room.

Even in the position of an item rental center, managing items is not easy. Currently, rental information is written by hand, paper-based, but there are students who do not fill out rental information, so information may be omitted. As a result, it is not possible to accurately determine the quantity of the item. Also, when students lose an item, it is not known who lost the item, and the cost of the loss is not recoverable.

Many other students felt the same discomfort about this problem, and the Office of Academic Affairs and Student Support at Dongguk University recognized that improvement was necessary. Therefore, Dongguk University's Office of Academic Affairs & Student Support requested Team 1 to develop a system that can improve these problems. So, we decided to think of a way to solve this problem.

1.3. System Requirement

The project sponsor is Team Leader of Dongguk University's School Affairs Student Support Office. The business needs are to improve item rental service’s environment. This will eventually lead to reduction of labor cost and improvement image and customer service.

Sponsors not just want to make to show quantity of rental items but also, they want this system to have more function for customers: user-friendly interface, types of items, expected duration of rental is available, location of item rental center, reservations for rental of items, mobile payment for an item lost charge and detailed information on rental items

For user-friendly interface, top menus are made for each university rental center building, and sections of type of items are shown as sub-menus (e.g., tools, sport equipment). The submenu provides a user interface that shows the items menu. “What kinds of rental items are there?” can show which items are available and can be rented. “How long does the user have to wait to rent item?” may indicate the estimated time it takes to rent an item. If a student wants to borrow an item on a certain day, he or she can make a reservation in advance. In the conventional method, when a user loses a rental item, the lost charge cannot be recovered, so there was a considerable cost loss. In the new system, a payment method for lost costs is connected, so if the user loses or does not return the item, the lost charge is automatically charged. Detailed information on rental items indicates information on rental items that the user using the item needs to know (e.g., port type of smartphone charger.).

First, for calculating the loss cost amount of the existing system, this project check and set the average price of rental items at Dongguk University CS Center, Bachelor's Office of College, and Student Council of College. Second, assume that the total number of goods is 1000. Third, the product collection rate per year is set at 45% (Ryu, 2020), so the lost late of rental items is 55%. The process up to the third can be found in Table 1

Table 1. Price of rental Items

N = 1,000	Sport equipments	Outdoor tent	Handcart	Vacuum cleaner	Charger	Umbrella	Average
Price of rental items (Unit: ₩)	20,000	100,000	20,000	50,000	5,000	6,000	33,500
Lost rate of rental items	55%						

The total number of rental items was 1,000, the average price was ₩33,500 and the collection rate was 45% so the lost late of rental items is 55%. The reduced cost of loss could be calculated as follows.

$$1000(total\ num\ of\ rental\ items) \times ₩33{,}500 \times 55\%(lost\ late\ of\ rental\ items)\ =\ ₩18{,}425{,}000 \quad (1)$$

Finally, Dongguk University has a total of 13 places to rent goods including CS Center, Bachelor's Office of College, and Student Council of College. We expected that it could be reduced by one person at each place. The annual salary for each administrative staff at Dongguk University is ₩25,300,000 (Park, 2020). Based on this, the labor cost that can be reduced for one year is as follows.

$$13(num\ of\ reduced\ employees) \times ₩25{,}300{,}000(labor\ cost\ of\ one\ employee)\ =\ ₩354{,}200{,}000 \quad (2)$$

We estimated that we could save total of ₩372,625,000 by adding up these values. Table 2 is a system request that describes these contents.

Table 2. System Request

Project Sponsor	Dongguk University's School Affairs & Student Support Office.
Business Need	It was inconvenient because students could not check if there was any stock left at the rental center. And in many cases, information was omitted by writing rental information in paper-based records. This caused the problem of poor product management, so improvement is needed.

Business Requirements	Using the Web or smartphone application, users will be able to search for item, Check the quantity and reservation the rental item. System has to include these services for the customers. **• What is the inventory quantity of rental items?** **• User-friendly interface** **• What kinds of rental items are there?** **• How long does the user have to wait to rent item?** **• In which building are the items rental centers located?** **• Reservations for rental of items** **• Mobile payment for an item lost charge** **• Detailed information on rental items**
Business Value	In Business Value, we estimated from three perspectives. **• Reduced labor costs** **Estimation of reduced labor costs per year** = 13(num of reduced employees) × ₩25,300,000(labor cost per one employee) = **₩354,200,000** **• 100% of lost costs can be recovered by automatic billing** **Estimation of reduced cost of loss** = 1000(Total num of rental items) × ₩33,500 × 0.55(lost late of rental items) = **₩18,425,000** **• As rental information is computerized, there is no missing information, increasing the recovery rate of rental items**
Special Issues or Constraints	• Even if the reservation and rental system is switched from paper-based records to smartphone web service, it will be better to maintain one existing paper-based note for the unexpected system error. • Cooperation with the university information team will be necessary to provide a stable system

2. Feasibility Analysis

2.1. Technical Feasibility

The pre-booking and inventory identification service (e.g., Library book rental service) is widely used service. Most libraries, even Dongguk University, are providing services to check and rent books. Therefore, the system of this project is expected to be a very familiar function and application to many people who design, develop, and install it.

This project will use information that is added from Dongguk University's CS Center, Bachelor's Office of College, and Student Council of College. However, the information is already existing, it can be stored in its own database and integrated into one. So, in this project, there is no need to gather additional information. The students, professors, and officers' information are from U-DRIMS database. This means that the expected risk of the project is not that high.

To use pre-booking and inventory identification services, information must initially be stored in a database using SQL. When the initial work is finished, employees can add or reduce the quantity of goods each time users rent. Therefore, it is judged that there is no significant difference from the previous work.

2.2. Economic Feasibility

For this project, identified costs and benefits are as follow.

• **Development cost:** development team salaries, software development, equipment, and data.

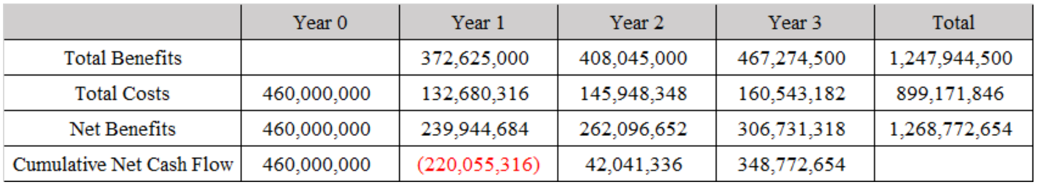
• **Operational cost:** software upgrades, operational team salaries, maintenance database costs, ingredients cost, and communications charge.

• **Tangible benefits:** reduced labor costs, eliminate the cost of lost items.

• **Intangible benefits:** improved user service, and data management.

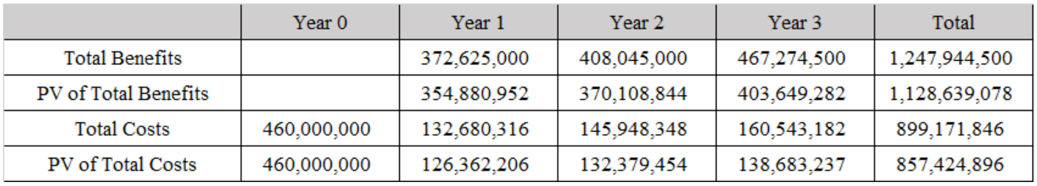
This project will develop database for pre-booking and inventory identification service. Therefore, expected development cost is ₩460,000,000 (KOSIS, 2020), and this will be one of the Year 0 costs. Expected database management cost is ₩132,680,316 ("Pricing Examples", n.d.). So, in Year 0, cost is ₩460,000,000 and after Year 0, cost is ₩132,680,316. Expected reduced labor costs' benefit is ₩354,200,000 and eliminate the cost of lost items' benefit is ₩18,425,000. So, total cost benefit per year is ₩372,625,000. And this benefit will increase by up to 10% every year. The expectation of cash flows according to the above results can be seen in Table 3, and the prediction of discount cash flows with an interest rate of 5% is shown in Table 4.

Table 3. Expected Cash Flow. (Unit: ₩)



	Year 0	Year 1	Year 2	Year 3	Total
Total Benefits		372,625,000	408,045,000	467,274,500	1,247,944,500
Total Costs	460,000,000	132,680,316	145,948,348	160,543,182	899,171,846
Net Benefits	460,000,000	239,944,684	262,096,652	306,731,318	1,268,772,654
Cumulative Net Cash Flow	460,000,000	(220,055,316)	42,041,336	348,772,654	

Table 4. Discounted Cash Flow. (Unit: ₩, Interest rate: 5%)



	Year 0	Year 1	Year 2	Year 3	Total
Total Benefits		372,625,000	408,045,000	467,274,500	1,247,944,500
PV of Total Benefits		354,880,952	370,108,844	403,649,282	1,128,639,078
Total Costs	460,000,000	132,680,316	145,948,348	160,543,182	899,171,846
PV of Total Costs	460,000,000	126,362,206	132,379,454	138,683,237	857,424,896

- **Return On Investment (ROI):**

$$\frac{\text{Total Benefits} - \text{Total Costs}}{\text{Total Costs}} = 38.788\% \tag{3}$$

- **Break-Even Point (BEP):**

$$\text{Number of years of negative cash flow} + \frac{\text{That year's Net Cash Flow} - \text{That year's Cumulative Cash Flow}}{\text{That year's Net Cash Flow}} = 1.84 \text{ years} \tag{4}$$

- **Net Present Value (NPV):**

$$\sum PV\ of\ Total\ Benefits - \sum PV\ of\ Total\ Costs = ₩271{,}214{,}182 \tag{5}$$

2.3. Organizational Feasibility

University authorities will get more benefit from reduced labor costs and eliminate the cost of lost items through using database system. Satisfaction of system users such as students, professors and officers will be greater because there is no need to go directly to the facility to check the inventory. And this project will be able to realize automation by automatically payments when the due date is exceeded or lost.

Data management is simple because when department changes the information, the information in the main database changes automatically. And if database link these technologies with U-DRIMS which store information of all students and professors, it will not be difficult to implement.

3. Project Planning

3.1. Estimation of Project Time Frame

To estimate the size of the project and the duration of the project, our team used function point analysis (Jeffery et al., 1993).

First, we need to Estimate System Size. The number and types of inputs, outputs, queries, files, and program interfaces of our system are as follows:

Inputs (5): Information on the rental center that the student chose., Item information selected by a student., The lost charge payment method chosen by the student., The time that student reservation for rent an item, Student ID and password for log in

Outputs (4): Information of payment is completed., The duration that student going to wait, Notification message when the item is available to rent, Notification message when return due date.

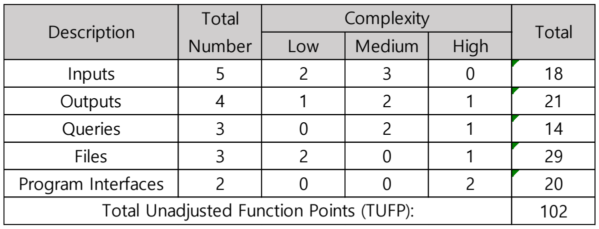
Queries (3): The item that student rent., Current stock quantity of the item., Estimated time to rent an item.

Files (3): Item information description, The item table, A document describing precautions when renting.

Program Interfaces (2): User-friendly interface of item to easily access the system, The hierarchical relationship between the upper interface menu and the lower interface menu

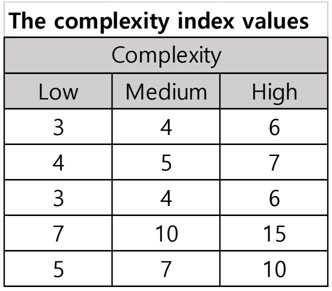
We calculated TUFP using Excel. Table 5 is a table that calculates TUFP by reflecting weight. And Table 6 is the Complexity index values that summarize the weight required to calculate the TUFP. As a result of calculating using the SUMPRODUCT function of Excel, **TUFP is 102.**

Table 5. Calculation results of TUFP.



Description	Total Number	Complexity			Total
		Low	Medium	High	
Inputs	5	2	3	0	18
Outputs	4	1	2	1	21
Queries	3	0	2	1	14
Files	3	2	0	1	29
Program Interfaces	2	0	0	2	20
Total Unadjusted Function Points (TUFP):					102

Table 6. Complexity index values.

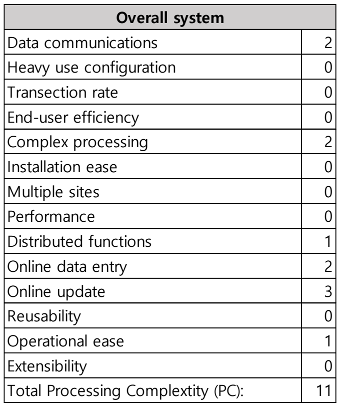


The complexity index values

Complexity		
Low	Medium	High
3	4	6
4	5	7
3	4	6
7	10	15
5	7	10

Total Processing Complexity (PC) was determined as shown in the Table 7. So, PC is 11. We thought of this system as a system that was quite complex due it needs to integration of information systems. So, we use 1.35 point to calculate APC. (0.65 = simple, 1.00 = average, 1.35 = complex)

Table 7. Calculation results of PC.



Overall system	
Data communications	2
Heavy use configuration	0
Transection rate	0
End-user efficiency	0
Complex processing	2
Installation ease	0
Multiple sites	0
Performance	0
Distributed functions	1
Online data entry	2
Online update	3
Reusability	0
Operational ease	1
Extensibility	0
Total Processing Complextity (PC):	11

Calculation of APC: 1.35 + (0.01 * 11(PC)) = **1.46 points**

Calculation of TAFP: 1.46(APC) * 102(TUFP) = **148.92 points**

We plan to use SQL to write the code. This is because SQL is most widely used in relational databases, and SQL is convenient for create, retrieve, update, delete manipulation process of data. The number of code lines required when using SQL is as follows:

The total lines of codes: 40(1TAFP per SQL code) * 148.92(TAFP) = **5956.8 lines of SQL code**

Step 2: Estimate Effort Required.

Second, we need to Estimate Required Effort. The following formula is a formula for calculating effort.

effort (in person-months) = 1.4 X thousands of lines of code

Using this formula, the efforts of our system are calculated as follows:

Effort (in person-months): 1.4 * 5956.8/1000 = 8.340 person-months

Step 3: Estimate Time Required.

Third, we need to Estimate Required Time. The following formula is a formula for calculating schedule time.

schedule time (months) = 3.0 X person-months$^{1/3}$

Using this formula, the schedule time of our project are calculated as follows:

Schedule time for project (in months): 3.0 * 8.340$^{1/3}$ = 6.084 months

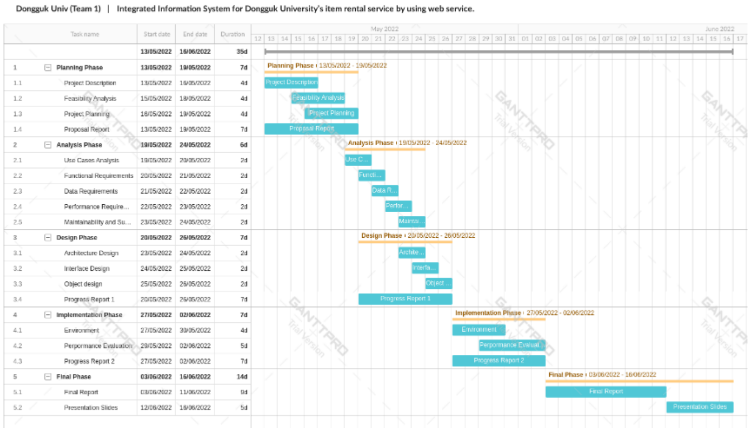
So, if we hire only one, it will take about 6.08 month (Approximately 182 days.). But our team has five members. Therefore, it can be calculated as follows:

6.08(months) / 5(team members) = **1.21 months (Approximately 36 days)**

Since the function point approach is the period calculation of implementation, the period of implementation phase is approximately 36 days as shown in the above calculation result.

3.2. Project Plan

Our team used the Gantt chart (Kumar, 2005), to manage the project plan. This is because the Gantt chart is easy to see at a glance, shows the amount of work done in certain periods of time in relation to the amount planned for those periods. Figure 1 is a Gantt chart of the Planning Phase, Analysis Phase, Design Phase, Implementation Phase and Final Phase of our project.



Dongguk Univ (Team 1) | Integrated Information System for Dongguk University's item rental service by using web service.

	Task name	Start date	End date	Duration
		13/05/2022	**16/06/2022**	**35d**
1	⊟ **Planning Phase**	**13/05/2022**	**19/05/2022**	**7d**
1.1	Project Description	13/05/2022	16/05/2022	4d
1.2	Feasibility Analysis	15/05/2022	18/05/2022	4d
1.3	Project Planning	16/05/2022	19/05/2022	4d
1.4	Proposal Report	13/05/2022	19/05/2022	7d
2	⊟ **Analysis Phase**	**19/05/2022**	**24/05/2022**	**6d**
2.1	Use Cases Analysis	19/05/2022	20/05/2022	2d
2.2	Functional Requirements	20/05/2022	21/05/2022	2d
2.3	Data Requirements	21/05/2022	22/05/2022	2d
2.4	Performance Require...	22/05/2022	23/05/2022	2d
2.5	Maintainability and Su...	23/05/2022	24/05/2022	2d
3	⊟ **Design Phase**	**20/05/2022**	**26/05/2022**	**7d**
3.1	Architecture Design	23/05/2022	24/05/2022	2d
3.2	Interface Design	24/05/2022	25/05/2022	2d
3.3	Object design	25/05/2022	26/05/2022	2d
3.4	Progress Report 1	20/05/2022	26/05/2022	7d
4	⊟ **Implementation Phase**	**27/05/2022**	**02/06/2022**	**7d**
4.1	Environment	27/05/2022	30/05/2022	4d
4.2	Perpormance Evaluation	29/05/2022	02/06/2022	5d
4.3	Progress Report 2	27/05/2022	02/06/2022	7d
5	⊟ **Final Phase**	**03/06/2022**	**16/06/2022**	**14d**
5.1	Final Report	03/06/2022	11/06/2022	9d
5.2	Presentation Slides	12/06/2022	16/06/2022	5d

Figure 1: Gantt chart of the Team 1 Project

4. Analysis of Requirements

4.1. Use Cases

Use Case 1. Rental business operation

Table 8 shows the first use cases of the university cs center, university academic practice, and university student council interaction with the system. The use case describes a series of activities performed by the user and describes how external users trigger events that the system must respond to (Dennis et al., 2015). The main procedure is to manage welfare rental goods and the overall rental business.

Table 8. Rental business operation Use Case

Use Case Name: Rental business operation	ID: UC-1	Priority: High
Actor: Dongguk University School CS Center, College Office, College Student Council Welfare Officer		
Description: This use case explains how the university welfare manager checks and manages the overall rental status of school welfare products.		
Trigger: University cs centers, university academic offices, and student council welfare officers manage or coordinate rental items through the management website. Type: (O) External ( ) Temporal		
Preconditions: Web sites are available. The rental property management database is online.		
Normal Course: a1.0 Search and inquire the management status of welfare items. a1. The system displays a default home page or a custom page. a2. The school cs center, university bachelor's office, and university student council welfare officer browse the page link or enter the account user name and password. a3. The welfare officer wants to create this account. Create an account use case. a4. The person in charge of the welfare project shall enter the status inquiry of rental goods. a5. The system displays the items you are currently renting, the borrower, the duration of the loan, and the data. a6. The person in charge of welfare projects selects the data of the borrowed goods. a7. The person in charge of the welfare project selects the current status data of the borrower. a8. The person in charge of the welfare project selects data on the lease period for each borrower. a9. The person in charge of the welfare project can check the status of the welfare products booked in advance. a10. In the event of a late payment, you can check the overdue item information, and late payment status. a11. The welfare representative selects the data to be deleted interface..		Information for Steps: User name/password Welfare item management status Save Account Current status of goods on loan Current status of the lender Rental period status by borrower Status of pre-booked welfare items Modifying the Interface
Alternative Courses: a1.1 The person in charge of welfare project is a visitor (step 1) 1. Use the interface that was changed in the previous visit to display a customized page for visitors. a1.2 An account has been created by a Welfare Service Representative (step 2) 1. Show a welcome message to the account holder. 2. The page is customized as the default interface for account holders.		Modified interface database

Postconditions: aP1. One or more status is added to the interface. aP2. Account holder interface list may be modified.			
Exceptions: aE1 : Account is not valid (Occurs at Step 2) 1. System displays incorrect user name/password 2. The system re-enter the user name/password or contact the customer service for assistance from the person in charge of welfare business.			
Summary Inputs	Source	Outputs	Destination
Username/Password Rental information	Person in charge of welfare service Admin Status Database Saved Interface Settings Database	Rental item details	rental database

In the case of Rental business operation use cases, the actor is the person in charge of the welfare project of the school cs center, university academic office, and university student council, and this use case helps to check the overall status of the welfare goods rental business. The rental goods situation and the rental business process change frequently, so the prerequisite for this use case is that the database must be online. You must create your own account, and you can use it whenever you enter this site. You can identify current rental items between your rental businesses, rental items, pre-booked items, borrower information, and overdue data. You can also modify the interface of the settings.

Use Case 2. Welfare item use

Table 9 shows the last use cases performed by students using university welfare programs. The main interests of students are to select and check rental items to be used, and to book rental items in advance if necessary. Table 6 Use Cases are organized around this information.

Table 9. Welfare item use Use Case

Use Case Name: Welfare item use	ID: UC-2	Priority: High
Actor: Dongguk University student		
Description:This use case describes the situation in which a student at Dongguk University is renting welfare items at the center.		
Trigger: The goods lender can check the status of the goods available for rent and the rental period and make a reservation. Type: (O) External ( ) Temporal		
Preconditions: Web site is available. The welfare item management database is online.		
Normal Course: b1.0 Rental Inquiry and Operation b1. The system displays a default home page or a custom page. b2. Students enter their account username and password. b3. Students want to create accounts: create account use cases. b4. During account creation, your ID, card account to register your password, and simple personal information are required. b5. Students can check what items they want to rent and how long they want to rent. b6. Select the items that the student will borrow. b7. Students check the rental status, rental period, and return date of the item. b8. Once the student has selected what he or she wants to borrow, make a		Information for Steps: User name/password Saving Rental Data Rental goods status Rental date/period status Booking service

<table>
<tr><td colspan="3">reservation on the date/time he or she wants to borrow.
b9. Send a return notification 2 days before the return date.
b10. Return the items lent by the student.
b11. If the student does not return the item even after the return date has passed, the registered card will automatically transfer the money.
b12. Students select data to delete from the interface if they wish to leave the service.</td><td>Modified interface</td></tr>
<tr><td colspan="3">Alternative Courses:
b1.1 Students are visitors (step 1)
1. Use the interface that was changed in the previous visit to display a customized page for visitors.
b1.2 Students create accounts (step 2)
1. Show a welcome message to the account holder.
2. The page is customized as the default interface for account holders</td><td>Modified interface database</td></tr>
<tr><td colspan="4">Postconditions:
bP1. One or more states are added to the interface.
bP2. The list of interfaces of the account holder may be modified.
bP3. Record student rental information
bP4. Record the modifications to the student's rental status.</td></tr>
<tr><td colspan="4">Exceptions:
bE1: Account is not valid.
1. System displays incorrect user name/password
2. The system asks students to re-enter their username/password or ask for help.
bE2: Your return date has passed.
1. The system will display a message that needs to be returned.</td></tr>
<tr><td>SummaryInputs</td><td>Source</td><td>Outputs</td><td>Destination</td></tr>
<tr><td>User name/password
Rental information
Student rental information</td><td>Student
Student's lending status database
Saved Interface Settings.</td><td>Modified interface
Modify Account</td><td>Account database
Rental item status database</td></tr>
</table>

In the case of using welfare rental items, the actor is selected as a student attending Dongguk University. The use case is based on the premise that students rent rental goods and reserve rental goods in advance according to the situation. Therefore, there is a prerequisite that the rental database must be online with the website. After the student proceeds with the rental, the rental information must be provided. The system also accepts the user if the student wants to create or log in with an existing account.ID and password. If a student fails to return the goods within the rental period, a late fee will be added and the late fee will be automatically paid from the card account registered in advance. The database may provide the student's personal information on the selected interface setting.

4.2. Functional Requirements

Functional requirements were created based on the Normal Course of each Use Case. Labeling starts with character b (e.g., b.3) is a functional requirement based on the Normal Course of Table 9. Rental Business Operation Case. Labeling starts with character b (e.g., b.5) is a functional requirement based on the Normal Course of Table 9. Rental Service Use Case.

b.3 The system shall allow students to log-in with U-DRIMS ID and password.

b.4 The system is set to the same conditions as the existing constraints.

b.5 The systems shall show images of all rental items.

b.6 The system shall allow users to know the number of items left, loan period, and deposits when they click the icon beside the image of rental items.

b.7 The system shall allow users to choose the menu what they want.

b.8 The system will send a notification to smartphone of users when two days before the specified due date.

b.9 The student returns the item.

b.10 The system will automatically transfer money as much as a deposit from account that was linked in advance when the student did not return the item.

b.11 The system will automatically delete student's information from database when students leave the service.

Our team used the Data Flow Diagram (DeMarco, 1979) to indicate the above functional requirements and processes. Figure 2 is a Data Flow Diagram based on the above description.

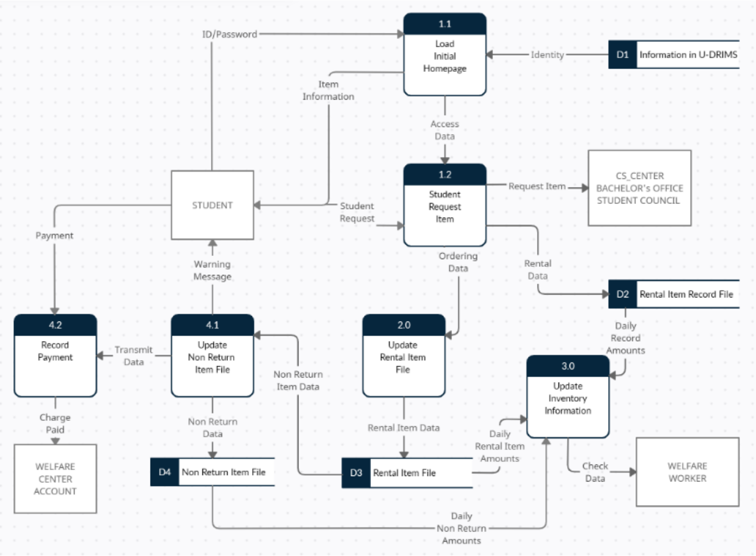


Figure 2: Data Flow Diagram of Team 1 Project.

4.3. Data Requirements

Our team used Entity Relationship Diagram (Chen, 1976) addressed in class assignment to specify relationships in our database. Figure 3 is an Entity Relationship Diagram of our team project's database. The basic sequence of system is like this. Student signs up for membership and log-in on the homepage. And then, student chooses item which they want to rent from the homepage using his/her smartphone. Each welfare center worker lend the item which is chosen by student. Each categories includes information that is necessary when the system process is progressing. Overall data is saved in below 'Database' to manage the current condition. Also, these data are used for developing and designing better system. Most of connection is handled with many to many (M:N) relationship.

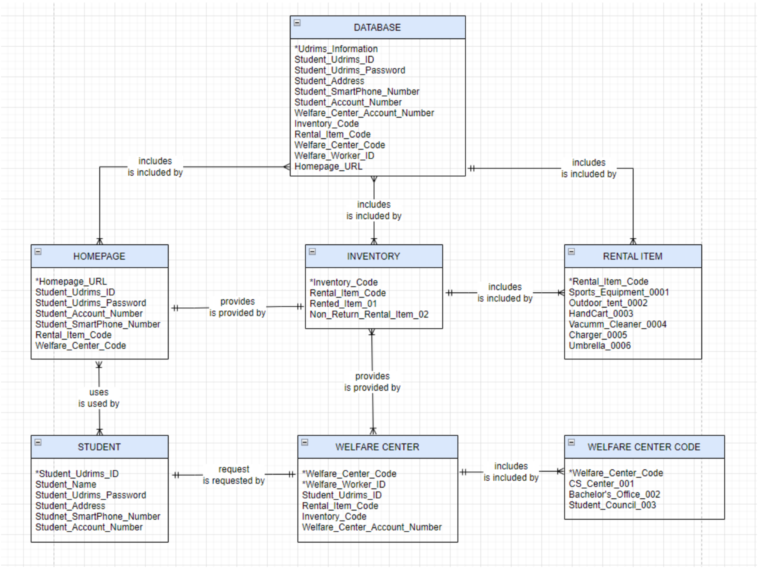


Figure 3: Entity Relationship Diagram of Team 1 Project.

4.4. Performance Requirements

We considered of performance requirements in three aspects: Speed, Capacity, and Compatibility. These three are the core performance requirements that users feel the most, and as analysts define the performance requirements for the system, the testability of the requirements is a key issue (Dennis et al., 2012). Details of each of the three aspects of performance requirements are as follows.

Speed

- Response time must be less than 2 seconds.
- Calculating time must be less than 5 seconds.
- Transmission of data must be instant and simultaneous.

Capacity

- There will be a sufficient size of server which is able to stand the maximum of 1000 users at peak times.
- System will get enough storage space that correspond with huge data to store it.

Compatibility

- System must run possibly in any operation system like Android and IOS.
- When paying for item loss charge, system will link with various payment services such as Kakao Pay and Toss.

4.5. Maintainability and Supportability Requirements

We considered maintainability and supportability requirements in three aspects: Refinement, Security, and Budget. These three factors are very important factors in Non-Functional Requirements in Software or System Engineering (Chung et al., 2012). The preceding contents were mainly considered in terms of the functionality of the system, but it is also very important to maintain and support the system to work well. Details of each of the three aspects of maintainability and supportability requirements are as follows.

Refinement

- The algorithm and code of system will be trimmed continuously to be simple.
- The system must keep consistency and not to be complicated.
- The system must be improved constantly by reflecting the user's feedback.

Security

- The system must be backed up in another file when the system is updated.
- System must prepare for several viruses.
- Users can access their accounts with username and password.

Budget

- The system needs proper financial support by university authorities.
- The profit from applying the system will be used for system managing.

5. Design

5.1. Architecture Design

In the case of our welfare goods rental management system, we use n-tired architecture as an architectural design.

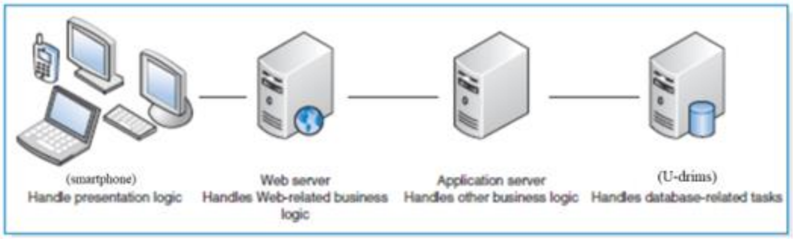

(smartphone)
Handle presentation logic
Web server
Handles Web-related business logic
Application server
Handles other business logic
(U-drims)
Handles database-related tasks

Figure 4: N-Tired Server Architecture (Dennis et al., 2012)

All student information of Dongguk University's colleges will be stored in U-DRIMS, and our team will be able to retrieve the necessary login information for the service using this U-DRIMS as a data store with student information and we need to configure an application server that handles the rental process and rental reservation. In addition, a late fee is charged when the rental period passes, so a system that allows the late fee to be automatically paid by registering a card account in advance is also required. These days, most students use smartphones, so I think it will provide good performance for implementing web services and payment systems using smartphones as clients. Therefore, our team thought it would be better to use n-tiered(four-tiered) architecture. Figure 4 is a figure of the architecture of the Team 1 project system.

5.2. Interface Design

Our team used a dialog diagram (Sutcliffe, 1988) to represent the basic design principles and hierarchical structure of the interface. When we enter the site page, we see the login page. When we log in from the login page, a menu appears in which we can select the building in which we want to rent the item we want. After that, we see a page with a list of all the things and we can rent the thing we want on it. Figure 5 shows the Dialogue Diagram for this content. And

Figure 6, Figure 7 and Figure 8 are drawings that simply implement the menu interface.

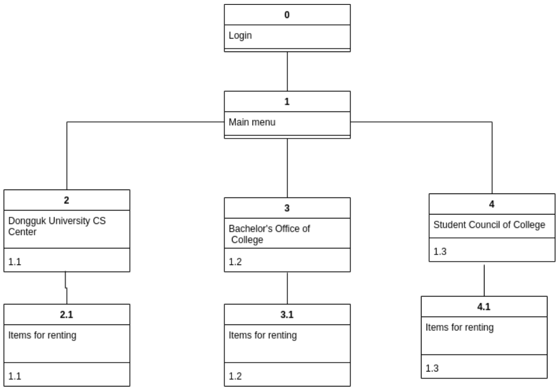


Figure 5: Interface Structure Design (Dialogue Diagram).

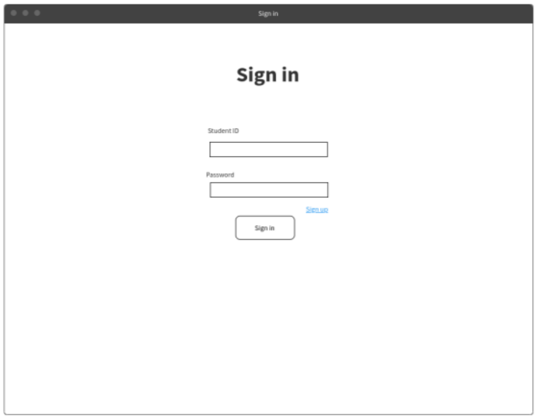


Figure 6: User Interface for Login.

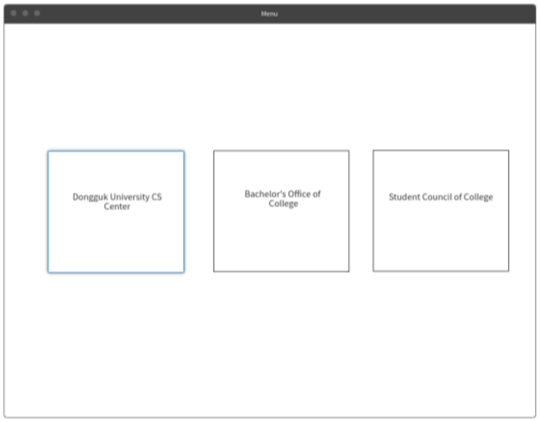

Menu
Dongguk University CS Center
Bachelor's Office of College
Student Council of College

Figure 7: User interface for choosing buildings for item rental center.

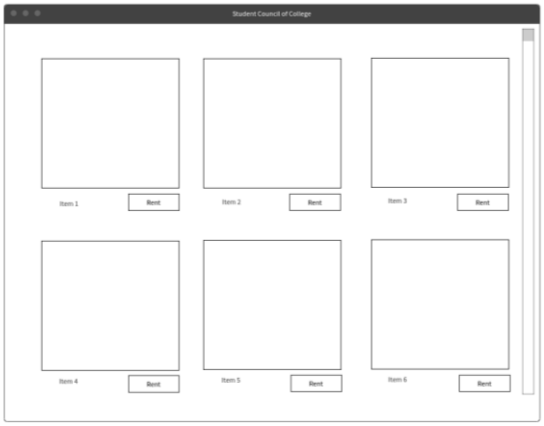


Figure 8: User interface for choosing for rental items.

5.3. Object Design

Our team used the class diagram (Booch, 2005) to represent the object relations of the class. Student information will be inherited from the StuLogin class to implement the login function. The student information is stored in U-DRIMS. The RentItem class also inherits the student information and implements the rent() function. Figure 9 is a class diagram showing the relationship between the classes and the content described above. This class diagram was made using drawio-app.com/.

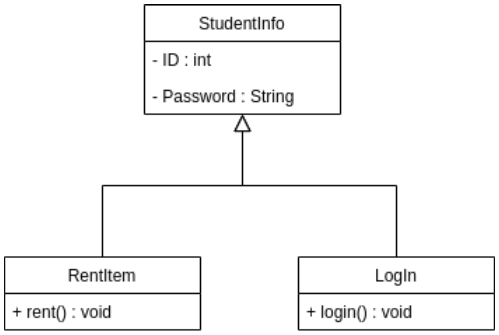
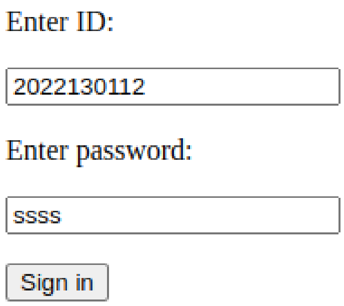


Figure 9: Class Diagram of Login and Ordering food.



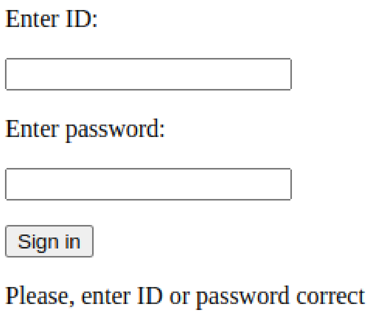
Enter ID:

2022130112

Enter password:

ssss

Figure 10: Student ID and password are incorrectly filled out.

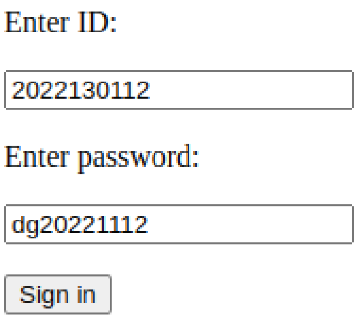


Enter ID:

Enter password:

Please, enter ID or password correct

Figure 11: Result of when the student ID and password are incorrectly filled out.



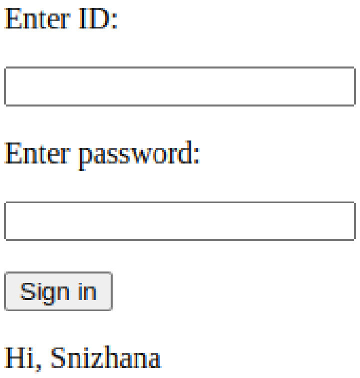
Enter ID:

2022130112

Enter password:

dg20221112

Figure 12: Student ID and password are correctly filled out.



Hi, Snizhana

Figure 13: Result of when the student ID and password are correctly filled out.

6. Implementation

6.1. Environment

The main programming language that we use is Python (version 3.10). The system application backend is based on Django framework which makes it easy to write reusable, readable and effective code. Framework also provides built-in ORM to simplify access to database, components for user form handling and other utilities for web applications. Django provides various security mechanisms that prevent making the common security mistakes.

For frontend we use bootstrap 5 CSS framework which comes with unified style for various UI components such as tables, lists, forms or containers. It also provides a huge set of JavaScript libraries. For JavaScript the JQuery framework is being used.

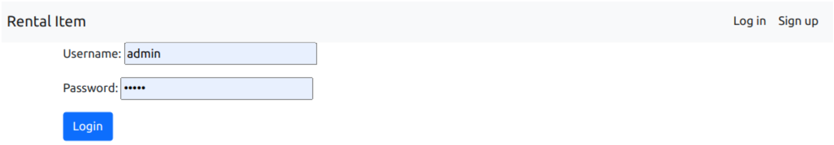
The system database backend is handled by MariaDB which is MySQL alternative. It differs in higher amount of storage engines as well as performance where it often performs better than MySQL.

For quick and simple deployment, we use Docker engine which allows to run the whole system into isolated Linux containers. Each container is based on pre-configured system image with possible additional configuration. It supports volume mounting, port forwarding, various network configurations. The main benefit here is that Docker eliminates issues caused by differences in environment the system is being ran in which is achieved by complete isolation from hosts OS.

The system is then being uploaded to AWS thanks to its docker integration, that allows creating AWS resources from configured docker containers directly. AWS is cloud computing platform from Amazon which provides virtual computer to host for instance web service applications.

6.2. Performance Evaluation

As shown in Figure 14, the first page of our site is the login page. An unauthorized user will not have access to the rest of the site.



Username: admin

Password: •••••

Figure 14: Implementation of login page

Figure 15 shows the main interface that be able to select the desired building to rent an item. We tested this interface and Figure 16 is the result of testing. It gives fast response and has been rated 93 points.

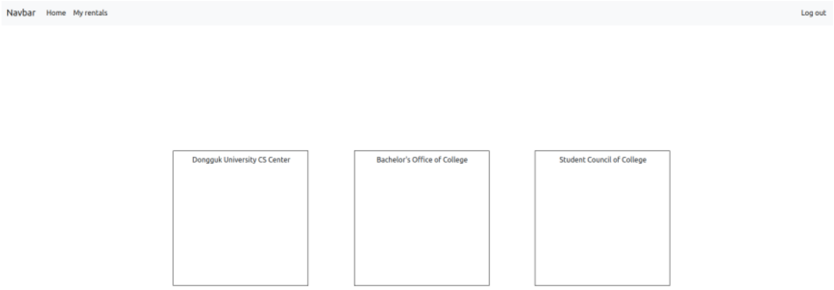
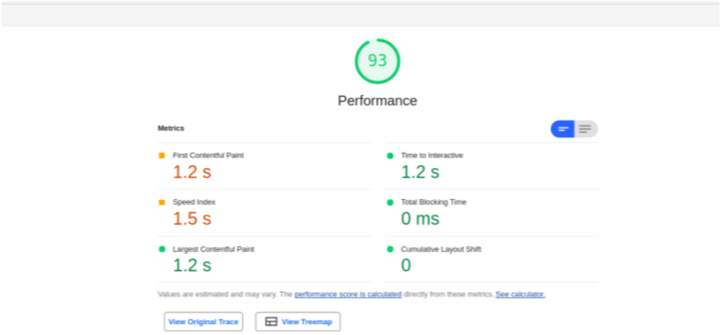

Navbar
Home
My rentals
Log out
Dongguk University CS Center
Bachelor's Office of College
Student Council of College

Figure 15: Implementation of interface to select rental center building



93

Performance

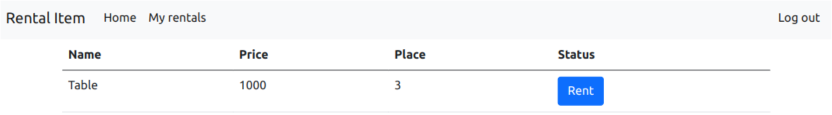
Metrics

First Contentful Paint 1.2 s	Time to Interactive 1.2 s
Speed Index 1.5 s	Total Blocking Time 0 ms
Largest Contentful Paint 1.2 s	Cumulative Layout Shift 0

Values are estimated and may vary. The performance score is calculated directly from these metrics. See calculator.

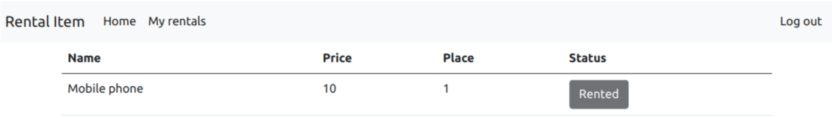
Figure 16: The result of testing on interface

After the user choose for the rental center in previous interface, user can see the table about rental information about item. Figure 17 is table of rental item and Figure 18 is table of when the item has been already rented.



Name	Price	Place	Status
Table	1000	3	Rent

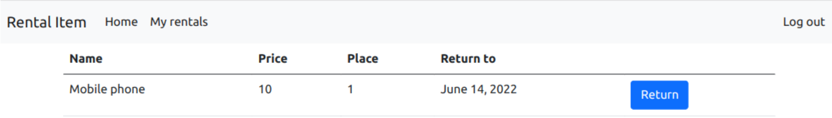
Figure 17: Table of rental item



Name	Price	Place	Status
Mobile phone	10	1	Rented

Figure 18: Table of when the item has been already rented

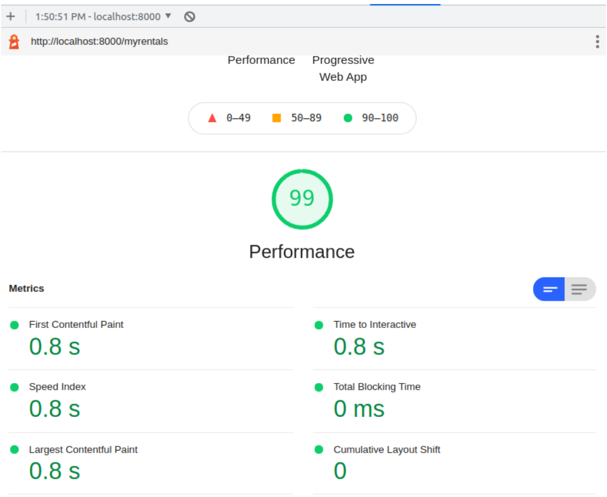
Figure 19 is the page of user's rented items, and it has an option to return the item.



Name	Price	Place	Return to	
Mobile phone	10	1	June 14, 2022	Return

Figure 19: The page of user's rented items

Figure 20 is testing about item rental table pages and the result is 99 points.



Metrics

● First Contentful Paint 0.8 s

● Time to Interactive 0.8 s

● Speed Index 0.8 s

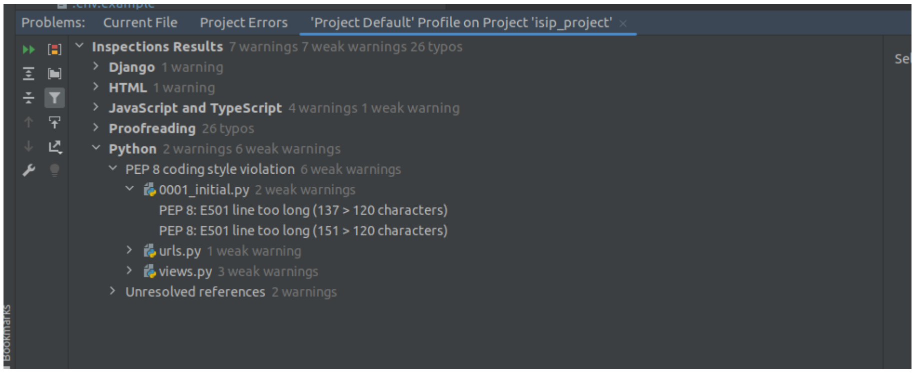
● Total Blocking Time 0 ms

● Largest Contentful Paint 0.8 s

● Cumulative Layout Shift 0

Figure 20: Testing about item rental table pages

Figure 21 is the code analysis for errors and warnings of Team 1's entire webservice implementation.



```
Inspections Results 7 warnings 7 weak warnings 26 typos
  Django 1 warning
  HTML 1 warning
  JavaScript and TypeScript 4 warnings 1 weak warning
  Proofreading 26 typos
  Python 2 warnings 6 weak warnings
    PEP 8 coding style violation 6 weak warnings
      0001_initial.py 2 weak warnings
        PEP 8: E501 line too long (137 > 120 characters)
        PEP 8: E501 line too long (151 > 120 characters)
      urls.py 1 weak warning
      views.py 3 weak warnings
    Unresolved references 2 warnings
```

Figure 21: Code analysis for errors and warnings

7. Conclusion

This project is for increasing user convenience by developing the Information system for item rental service. Based on system requirements, feasibility analysis was conducted. In particular, ROI, BEP, and NPV were calculated by predicting the cash flow in the economic feasibility analyzation. This project includes planning, analysis, and design phase and planed actual project schedule by using Gantt chart. For implementation phase plan, this project estimated time frame by using function point approach. By using Use Case and ERD, and analyzing requirements, our team prepared to make system. For design the system, architecture design and Dialogue Diagram were used. Implementation Phase of the system was implemented in AWS, Django, and MYSQL environments.

The system developed by this project will be useful system for rental users. Also, the project sponsor will be satisfied, because this system can reduce the number of lost items. Then, it will lead to reduce cost of loss. The total project time is only 35 days. Due to this time constraint, the implementation phase did not proceed perfectly as enough to hold many users in Dongguk University. But we may be able to do it with future work if we have more time and access to U-DRIMS.

References

Dongguk University CS center. (n.d.). Dongguk University. https://www.dongguk.edu/page/218#

Ryu, Y. (2020). *Conscience Yangsan Rental Business Recovery Rate 45%*. No Cut News. https://www.nocutnews.co.kr/news/5402875

Park, D. (2022). Dongguk University Student CS Center Staff Recruitment Announcement. Dongguk University. https://www.dongguk.edu/article/EMPLHIRE/detail/26744577

KOSIS. (2020). DB Development Completion Time and Deployment Cost. https://kosis.kr/statHtml/statHtml.do?orgId=127&tblId=DT_127004N_316

Pricing Examples. (n.d.). Google Cloud SQL. https://cloud.google.com/sql/docs/sqlserver/pricing-examples?hl=ko

Jeffery, D. R., Low, G. C., & Barnes, M. (1993). A comparison of function point counting techniques. *IEEE*

Transactions on Software Engineering, *19*(5), 529-532.

Kumar, P. P. (2005). Effective use of Gantt chart for managing large scale projects. *Cost engineering*, *47*(7), 14.

DeMarco, T. (1979). Structure analysis and system specification. In Pioneers and Their Contributions to Software Engineering (pp. 255-288). Springer, Berlin, Heidelberg.

Chen, P. P. S. (1976). The entity-relationship model—toward a unified view of data. ACM transactions on database systems (TODS), 1(1), 9-36.

Dennis, A., Roth, R. M., & Wixom, B. H. (2012). System analysis and design, Fifth Edition. John Wiley & Sons.

Chung, L., Nixon, B. A., Yu, E., & Mylopoulos, J. (2012). Non-functional requirements in software engineering (Vol. 5). Springer Science & Business Media.

Sutcliffe, A. (1988). Dialogue Design. In Human-Computer Interface Design (pp. 97-107). Palgrave, London.

Booch, G. (2005). The unified modeling language user guide. Pearson Education India.