Software Requirements Specification

Version 1.0

<<Annotated Version>>

April 15, 2004

Web Publishing System

Joan Teamleader

Paul Adams

Bobbie Baker

Charles Charlie

Submitted in partial fulfillment

Of the requirements of

CS 310 Software Engineering

<<Any comments inside double brackets such as these are *not* part of this SRS but are comments upon this SRS example to help the reader understand the point being made.

Refer to the SRS Template for details on the purpose and rules for each section of this document.

This work is based upon the submissions of the Spring 2004 CS 310. The students who submitted these team projects were Thomas Clay, Dustin Denney, Erjon Dervishaj, Tiffanie Dew, Blake Guice, Jonathan Medders, Marla Medders, Tammie Odom, Amro Shorbatli, Joseph Smith, Jay Snellen, Chase Tinney, and Stefanie Watts. >>

# Table of Contents

[Table of Contents i](#_Toc77487619)

[List of Figures ii](#_Toc77487620)

[1.0. Introduction 1](#_Toc77487621)

[1.1. Purpose 1](#_Toc77487622)

[1.2. Scope of Project 1](#_Toc77487623)

[1.3. Glossary 2](#_Toc77487624)

[1.4. References 2](#_Toc77487625)

[1.5. Overview of Document 2](#_Toc77487626)

[2.0. Overall Description 4](#_Toc77487627)

[2.1 System Environment 4](#_Toc77487628)

[2.2 Functional Requirements Specification 5](#_Toc77487629)

[2.2.1 Reader Use Case 5](#_Toc77487630)

[Use case: Search Article 5](#_Toc77487631)

[2.2.2 Author Use Case 6](#_Toc77487632)

[Use case: Submit Article 6](#_Toc77487633)

[2.2.3 Reviewer Use Case 7](#_Toc77487634)

[Use case: Submit Review 7](#_Toc77487635)

[2.2.4 Editor Use Cases 8](#_Toc77487636)

[Use case: Update Author 8](#_Toc77487637)

[Use case: Update Reviewer 9](#_Toc77487638)

[Use case: Update Article 9](#_Toc77487639)

[Use case: Receive Article 10](#_Toc77487640)

[Use case: Assign Reviewer 11](#_Toc77487641)

[Use case: Receive Review 11](#_Toc77487642)

[Use case: Check Status 12](#_Toc77487643)

[Use case: Send Response 12](#_Toc77487644)

[Use case: Send Copyright 13](#_Toc77487645)

[Use case: Remove Article 14](#_Toc77487646)

[Use case: Publish Article 14](#_Toc77487647)

[2.3 User Characteristics 15](#_Toc77487648)

[2.4 Non-Functional Requirements 15](#_Toc77487649)

[3.0. Requirements Specification 17](#_Toc77487650)

[3.1 External Interface Requirements 17](#_Toc77487651)

[3.2 Functional Requirements 17](#_Toc77487652)

[3.2.1 Search Article 17](#_Toc77487653)

[3.2.2 Communicate 18](#_Toc77487654)

[3.2.3 Add Author 18](#_Toc77487655)

[3.2.4 Add Reviewer 19](#_Toc77487656)

[3.2.5 Update Person 19](#_Toc77487657)

[3.2.6 Update Article Status 20](#_Toc77487658)

[3.2.7 Enter Communication 20](#_Toc77487659)

[3.2.8 Assign Reviewer 21](#_Toc77487660)

[3.2.9 Check Status 21](#_Toc77487661)

[3.2.10 Send Communication 22](#_Toc77487662)

[3.2.11 Publish Article 22](#_Toc77487663)

[3.2.12 Remove Article 23](#_Toc77487664)

[3.3 Detailed Non-Functional Requirements 23](#_Toc77487665)

[3.3.1 Logical Structure of the Data 23](#_Toc77487666)

[3.3.2 Security 25](#_Toc77487667)

[Index 26](#_Toc77487668)

# List of Figures

[Figure 1 - System Environment 4](#_Toc77487669)

[Figure 2 - Article Submission Process 6](#_Toc77487670)

[Figure 3 - Editor Use Cases 8](#_Toc77487671)

[Figure 4 - Logical Structure of the Article Manager Data 23](#_Toc77487672)

# 1.0. Introduction

## 1.1. Purpose

The purpose of this document is to present a detailed description of the Mobile web application named 'HYUⓔmini'. It will explain the purpose and features of the application, the interfaces of the application, what the application will do, the constraints under which it operates and how the application will react to external inputs. This document is intended for both the stakeholders and the developers of the application.

.

## 1.2. Scope of Project

This application will a compact version of currently using Hanyang University application. We will call this as a HYUmini. This application provides only key functions of current Hanyang University application. This application is more intuitive and lighter(uses less memory spaces and fast run time). By reducing functions, user can reduce overuse of time to search for wanted functions and that leads to remove inconvenience of using abundant application.

## 1.3. Glossary

App. : application

Univ. : university

Restaurants : Dormitory, Student, Foodcourt, Staff, Business incubator Center restaurants

DB : including student information, shuttle bus schedule information and so on

User : student

Shuttle bus : bus on circular route whole day

BBS : students can write questions, comments, and so on in the internet

## 1.4. References

IEEE. *IEEE Std 830-1998 IEEE Recommended Practice for Software Requirements Specifications.* IEEE Computer Society, 1998.

## 1.5. Overview of Document

The next chapter, the Overall Description section, of this document gives an overview of the functionality of the product. It describes the informal requirements and is used to establish a context for the technical requirements specification in the next chapter.  
The third chapter, Requirements Specification section, of this document is written primarily for the developers and describes in technical terms the details of the functionality of the product.  
Both sections of the document describe the same software product in its entirety, but are intended for different audiences and thus use different language.

# 2.0. Overall Description

## 2.1 System Environment

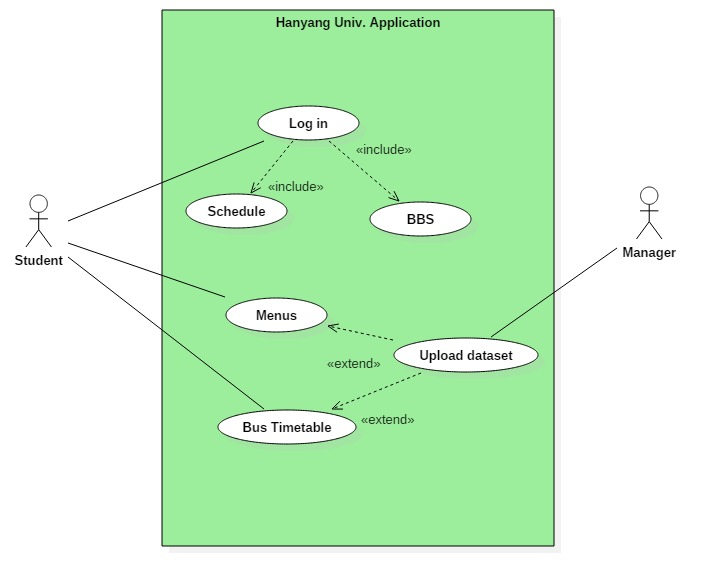


Figure - System Environment

The Hanyang app. has 2 active actors. Univ. Manager upload information in Hayang application's database. student can receive information.

.

## 2.2 User Characteristics

The users are students in the university. The students are expected to be able to use in daily life of university. The main functions of HYU Mini will have the notice board, function, schedule of lectures, restaurant menus, bus timetable and log in for mobile student card.

Every user are expected to use student number to log in. The application should facilitate the students’ school life.

The detailed look of these pages is discussed in section 3.2 below.

## 2.4 Constraints

HYU Mini is developed by students and the team consists of 10 persons. Therefore, there are few limitations. Following are the limitations :

* Lack of professionalism
* Lack of reality database

## 2.5 Non-Functional Requirements

* Click Limitations - no more than 3 clicks
* Convenient View - simple pictures of button, priority of menu button

# 3.0. Requirements Specification

## 3.1 External Interface Requirements

The application of Hanyang Univ. is going to use the browsers in operating system. Moreover, the link to an external system is Database to verify the identity of students.

## 3.2 Functional Requirements

### 3.2.1 Log-in system Table

|  |  |
| --- | --- |
| **Use Case Name** | Log in |
| **Goal** | Student who want to check personal information |
| **Trigger** | Student clicks the ‘log in’ button to sign in. |
| **Precondition** | User didn’t sign in the system. |
| **Basic Paths** | 1. Students click the ‘Sign in’ button in HYU Mini. 2. Student should input his ID and password in edit box. 3. Student clicks the ‘enter’ button for accessing system. 4. The student wants to check his individual information. 5. The student is enable to check schedule and BBS after sign in. |
| **Alternative Paths** | In Keeping the log in state |
| **Post-condition** | The account accesses to database. |

|  |  |
| --- | --- |
| **Use Case Name** | Log out |
| **Goal** | Student can log out this system. |
| **Trigger** | Student clicks the ‘log out’ button to sign out. |
| **Precondition** | Student already log in this system. |
| **Basic Paths** | 1. Student already accesses HYU application.  2. Student who wants to log out clicks the ‘log out’ button.  3. HYU application will be broken the connection. |
| **Alternative Paths** | Remain user information. |
| **Post-condition** | The account disconnects the database. |

### 3.2.2 BBS system Table

|  |  |
| --- | --- |
| **Use Case Name** | Writing contents |
| **Goal in context** | *People write a content in a bulletin board* |
| **Trigger** | Writer clicks “writing” button in a bulletin board page |
| **Precondition** | Writer logins the app and accesses into a bulletin board page |
| **Basic Path** | 1. The writer clicks “writing” button. 2. The system shows the writer edting page. 3. The writer inputs words of a title and content. 4. The writer chooses if he(she) clicks “save” button or “cancle” button. 5. If writer clicks “save”button, the system checks the validation of a title or content. 6. If the validation is passed, writing action is done. |
| **Alternative Paths** | In step 4, if the writer clicks “cancel” button, the system stops editing function and shows up a list page of a bullentin board section.  In step 6, the validation is not passed, the system displays alert for which one writer should input again. Return to step 3. |
| **Postcondition** | The system pops up alert to identify saving for the writer.  The system saves the content in a db.  The system shows up a list page of a bulletin board. |

|  |  |
| --- | --- |
| **Use Case Name** | Reading contents |
| **Goal in context** | *People reading a content in a bulletin board* |
| **Trigger** | Writer clicks one of the titles in a bulletin board list page |
| **Precondition** | Writer logins the app and accesses into a bulletin board page |
| **Basic Path** | *1.* The reader clicks the title of content. |
| **Alternative Paths** |  |
| **Postcondition** | The system display the content |

|  |  |
| --- | --- |
| **Use Case Name** | Editing content |
| **Goal in context** | *People edit the content or the title of which themselves wrote* |
| **Trigger** | Editor clicks “editing” button in a content page |
| **Precondition** | Editor logins the app  Editor has ownership of the content  Editor accesses the content page |
| **Basic Path** | *1.* The editor clicks “editing” button.  2. The system displays a editing page.  3. The editor edits the title or content.  4. The editor chooses if he(she) clicks “finish” button or “cancle” button.  5. If writer clicks “finish”button, the system checks the validation of a title or content.  6. If the validation is passed, writing action is done. |
| **Alternative Paths** | In step 4, if the editor clicks “cancel” button, the system stops editing function and shows up a list page of a bullentin board section.  In step 6, the validation is not passed, the system displays alert for which one editor should input again. Return to step 3. |
| **Postcondition** | The system pops up alert to identify updating for the editor.  The system updates the content in a db.  The system shows up a list page of a bulletin board. |

|  |  |
| --- | --- |
| **Use Case Name** | Deleting content |
| **Goal in context** | *People delete a content of which themselves wrote* |
| **Trigger** | User clicks “delete” button in a content page |
| **Precondition** | User logins the app  User has ownership of the content  User accesses the content page |
| **Basic Path** | *1.* The user clicks “delete” button in a content page.  2. The system pops up a alert for user to delete this content once again.  3. If the user clicks “yes” button in the alert, the system delete a content. |
| **Alternative Paths** | In step 3, if the user clicks “no” button in the alert, the system stops any actions and displays the content. |
| **Postcondition** | The system deletes all about the content in the db.  The system displays the list page after deleting the content. |

|  |  |
| --- | --- |
| **Use Case Name** | Replying for a content |
| **Goal in context** | *User replies for a content* |
| **Trigger** | User writes text in a reply text box. |
| **Precondition** | Editor logins the app  Editor accesses the content page |
| **Basic Path** | *1.* User writes text in a reply text box.  2. User clicks “reply” button. |
| **Alternative Paths** |  |
| **Postcondition** | The system saves the reply in a db.  The system resets for a blank in a reply text box.  The system displays the content page. |

|  |  |
| --- | --- |
| **Use Case Name** | Editing reply |
| **Goal in context** | *Editor edits the reply* |
| **Trigger** | Editor clicks “edit” button in a reply place |
| **Precondition** | Editor logins the app  Editor accesses the content page  Editor has a ownership for the reply |
| **Basic Path** | *1.* Editor clicks “edit” button in a reply place.  2. Editor system displays a editing page.  3. Editor editor edits the reply.  4. Editor editor chooses if he(she) clicks “finish” button or “cancle” button.  5. If editor clicks “finish”button, the system checks the validation of a content.  6. If the validation is passed, editing action is done. |
| **Alternative Paths** | In step 4, if the editor clicks “cancel” button, the system stops editing function and shows up a content  In step 6, the validation is not passed, the system displays alert for which one editor should input again. Return to step 3. |
| **Postcondition** | The system saves the reply in a db.  The system resets for a blank in a reply text box.  The system displays the content page. |

|  |  |
| --- | --- |
| **Use Case Name** | Deleting reply |
| **Goal in context** | *User deletes reply* |
| **Trigger** | User clicks “delete” button in a reply place |
| **Precondition** | User logins the app  User accesses the content page  User has a ownership for the reply |
| **Basic Path** | *1.* The user clicks “delete” button in a content page.  2. The system pops up a alert for user to delete this reply once again.  3. If the user clicks “yes” button in the alert, the system delete the reply. |
| **Alternative Paths** | In step 3, if the user clicks “no” button in the alert, the system stops any actions and displays the content. |
| **Postcondition** | The system deletes the reply in a db.  The system resets for a blank in a reply text box.  The system displays the content page. |

|  |  |
| --- | --- |
| **Use Case Name** | Searching content. |
| **Goal in context** | *User searchs a content which he(she) wants to* |
| **Trigger** | User accesses the list page of bulletin board section. |
| **Precondition** | User logins the app. |
| **Basic Path** | 1. The user chooses how to search the content. The choices are by Author, by keyword, and by name of a title . 2. If the search is by Author, the user selects the dropdown menu to author. 3. The user inputs the name of author on the input text box. 4. The user clicks “find” button. |
| **Alternative Paths** | In step 2, if the user selects to search by keyword, the user selects the dropdown menu to keyword.  3. The user inputs the keyword on the input text box. Return to step 4.  In step 2, if the user selects to search by keyword, the user selects the dropdown menu to a name of title.  3. The user inputs the name of title on the input text box. Return to step 4. |
| **Postcondition** | The system creates and presents a list of all contents in the db. |

### 3.2.3 ShuttleBus System Table

|  |  |
| --- | --- |
| **Use Case Name** | Show shuttle bus schedule |
| **Goal in context** | To provide shuttle bus schedule |
| **Trigger** | User clicks the ‘bus schedule’ button |
| **Precondition** | Run HYUmini application |
| **Basic Path** | Student want to check bus schedule, so he or she click the ‘bus schedule’ button in HYUmini application.  Application shows two tabs in ‘bus schedule’.  -Shuttle bus tab(displayed by default)  -Commute bus tab  ‘Shuttle bus tab’ shows shuttle bus schedule. Using current time, application calculates whether it’s during semester, session or break and decide which schedule to display.  Shuttle bus schedule is provided in GUI format. Bus on operating is showed with tracking the current location of the bus.  User can check whole shuttle bus schedule by clicking ‘Entire shuttle schedule |
| **Alternative Paths** | If current time is not operation time, the page shows the earliest shuttle bus schedule. |
| **Postcondition** | Shuttle bus schedule is provided in visualized format |

|  |  |
| --- | --- |
| **Use Case Name** | Show commute bus schedule |
| **Goal in context** | To provide commute bus schedule |
| **Trigger** | User clicks the ‘commute bus schedule’ tab |
| **Precondition** | User checked shuttle bus schedule |
| **Basic Path** | 1. User clicks commute bus schedule tab to check commute bus schedule. 2. In commute bus schedule tab, user can select which route to be displayed among 9 routes.   Application displays selected route in detail with GUI form. |
| **Alternative Paths** | If current time is not operation time, the page shows the earliest shuttle bus schedule. |
| **Postcondition** | Commute bus schedule is provided in visualized format |

### 

### 3.2.4 Lecture Schedule system table

|  |  |
| --- | --- |
| **Use Case Name** | Show schedule |
| **Use Case Format** | Schedule Button Click |
| **Goal in context** | To show student schedule that user easily check in match one weeks course time schedule |
| **Precondition** | A user did login on a web with valid id and password |
| **Trigger** | The user decide to “Schedule” the hanyang web site |
| **Basic Path** | 1. A users are observe web page in hanyang app 2. if users can search the student schedule 3. A users are click the “Schedule”button 4. Web system check the normal mode 5. Web system connects the DB server 6. Web system find server which have excellent condition stat about transferring data of schedule and DB server returns the schedule data 7. Web system converts schedule data to show the week’s schedule 8. Web system pop up the complete message 9. Web system shows the student schedule to user |
| **Alternative Paths** | Step 6 if DB server isnot connecting, Web system pop up th  e disconnet message |
| **Postcondition** | The schedule list is generated from the student schedule information |

|  |  |
| --- | --- |
| **Use Case Name** | Save Down Schedule |
| **Use Case Format** | schedule down button click |
| **Goal in context** | To show student schedule that anywhere user see |
| **Precondition** | To save the user’s week’s schedule to user’s local machine when user click the ‘Save Schedule’ button in schedule web page |
| **Trigger** | The user decide to “Download” the schedule web page |
| **Basic Path** | 1. If users can download the week’s schedule 2. A users are click the “Save Schedule”button 3. Web system make the on of week schedule file and  returns the one page of student week schedule file 4. local system save the file in local repository 5. Web system is checking the saving file to local repository 6. Web system pop up the download complete message 7. A users can see student schedule file to local repository |
| **Alternative Paths** | Step 5 if Web system is not checking the saving file to local repository Web system pop up the error message |
| **Postcondition** | Updated downloaded schedule file log  Download result pop up to the user |

|  |  |
| --- | --- |
| **Use Case Name** | Share Schedule File |
| **Goal in context** | To share student schedule with another people for project meeting and Making plan in Kakao Talk |
| **Precondition** | To share the user’s student schedule when user click the ‘share’ button in schedule web page |
| **Trigger** | The user decide to “share” in the schedule web page |
| **Basic Path** | 1.if users can share the student schedule  2.A users are click the “share”button  3.Web system connect Kakao server  4.Web system find server which have excellent condition stat about transferring data of schedule and Kakao server returns the completed results  5.Schedule is posted to share with another people in the Kakao Talk  6.Web system pop up the complete message  7.A user and another people can see the schedule int Kakao Talk |
| **Alternative Paths** | Step 4 if Kakao server is not connecting, Web system pop up th  e disconnet message |
| **Postcondition** | The schedule list is shared to post the KakaoTalk |

|  |  |
| --- | --- |
| **Use Case Name** | Highlight view mode |
| **Goal in context** | To show highlight week’s schedule that user easily check when select the Highlight view mode in setting for matching currently time and schedule time |
| **Precondition** | A user did login on a web with valid id and password  Set up the highlight view mode |
| **Trigger** | The user decide to “schedule” the hanyang web page |
| **Basic Path** | 1.if users can search the student schedule  2.A users are click the “Schedule” button  3.Web system check the highlight view mode  4.Web system connects the DB server  5.Web system find server which have excellent condition stat about transferring data of schedule and DB server returns the schedule data  6.Web system converts schedule data to show the week’s schedule  7.Web system pop up the complete message  8.Web systme is matching currently time and schedule time  9.Web system is showing the near time schedule |
| **Alternative Paths** | Step 5 if DB server is not connecting, Web system pop up th  e disconnet message |
| **Postcondition** | The near schedule list is showed to get highlight view |

|  |  |
| --- | --- |
| **Use Case Name** | Setting |
| **Goal in context** | To show highlight student schedule and normal schedule that user easily check when select the Highlight mode or normal mode |
| **Precondition** | To set up the user’s student schedule of highlight mod and normal mode when user click the ‘Setting’ button in schedule web page |
| **Trigger** | The user decide to “setting” the schedule web site |
| **Basic Path** | 1. if users can select the schedule view of highlight mode and nomal mode 2. A users are click the “Setting” button 3. User check the highlight mode or normal mode 4. Web system connects the DB server 5. Web system find server which have excellent condition stat about transferring data of schedule and DB server save the setting data 6. Web system pop up the complete message 7. Web systme update schedule view mode |
| **Alternative Paths** | Step 4 if DB server is not connecting, Web system pop up th  e disconnet message |
| **Postcondition** | User can see different view mode |

|  |  |
| --- | --- |
| **Use Case Name** | Add User Schedule |
| **Goal in context** | To add user's schedule to student schedule and display to application |
| **Precondition** | To add the user's schedule when user click the 'Add Schedule'button |
| **Trigger** | The user decide to “Add Schedule” the schedule web site |
| **Basic Path** | 1.if users can add the user's privative schedule  2.User are click the 'Add Schedule' button  3.User insert user's time information  4.User click the 'Save'  5.Application upload the time information  6.Schedule is update  7.Display the new Schedule |
| **Alternative Paths** | Step 3 if time is already use, User can not insert new time information |
| **Postcondition** | User can add Privative Schedule |

### 3.2.5 Restaurant menu System table

|  |  |
| --- | --- |
| **Use Case Name** | Selecting the day of week |
| **Goal in context:** | User see the menu of the restaurants that user can visit at the selected day of week. |
| **Trigger** | Press the button among the buttons of the day of week ‘월’, ‘화’, ‘수’, ‘목’, ‘금’, ‘토’, ‘일’ |
| **Precondition** | ‘Weekly menu’(‘금주의 식단’) page is opened. |
| **Basic Path** | 1. If user open ‘Weekly menu’(‘금주의 식단’) page, it shows the buttons of the day of week ‘월’, ‘화’, ‘수’, ‘목’, ‘금’, ‘토’, ‘일’, by default, the current day button is selected. 2. It shows the menu, user can visit, to user. 3. If user pressed other day button, it proceed step3. |
| **Alternative Paths** | In step3, the shown restaurants varies from the selected day of the week.   1. After user selects the day of week button, if user select time, the menu of several restaurant by selected time. 2. If user selects ‘토’or ‘일’ button, it shows only the menu of ‘기숙사 식당’, regardless of what time is selected. |
| **Post condition** | It shows the menu of the selected day of week that user can visit. |

### 

|  |  |
| --- | --- |
| **Use Case Name** | Dividing the menu by time. |
| **Goal in context:** | Show the menu can be ordered by time. |
| **Trigger** | 1. Press one button among ‘아침’, ‘점심’, ‘저녁’ button. 2. Swipe screen to left or right in ‘금주의 식단’ page. |
| **Precondition** | The one of the day of week button ‘월’, ‘화’, ‘수’, ‘목’, ‘금’ is selected. |
| **Basic Path** | 1. In weekday, when user visit ‘금주의 식단’ page, time button(‘아침’, ‘점심’, ‘저녁’) is shown, by default, the time button, close to the current time, is selected. 2. It shows menu of selected time. 3. When user select other time, proceed step3. |
| **Alternative Paths** | After step1, the time buttons can be not exist, according to selected day of week   * If ‘토’ or ‘일’ button is selected, it shows the menu of ‘기숙사 식당’ for all time, regardless of time. |
| **Post condition** | Display the menu user can order at time, in page. |

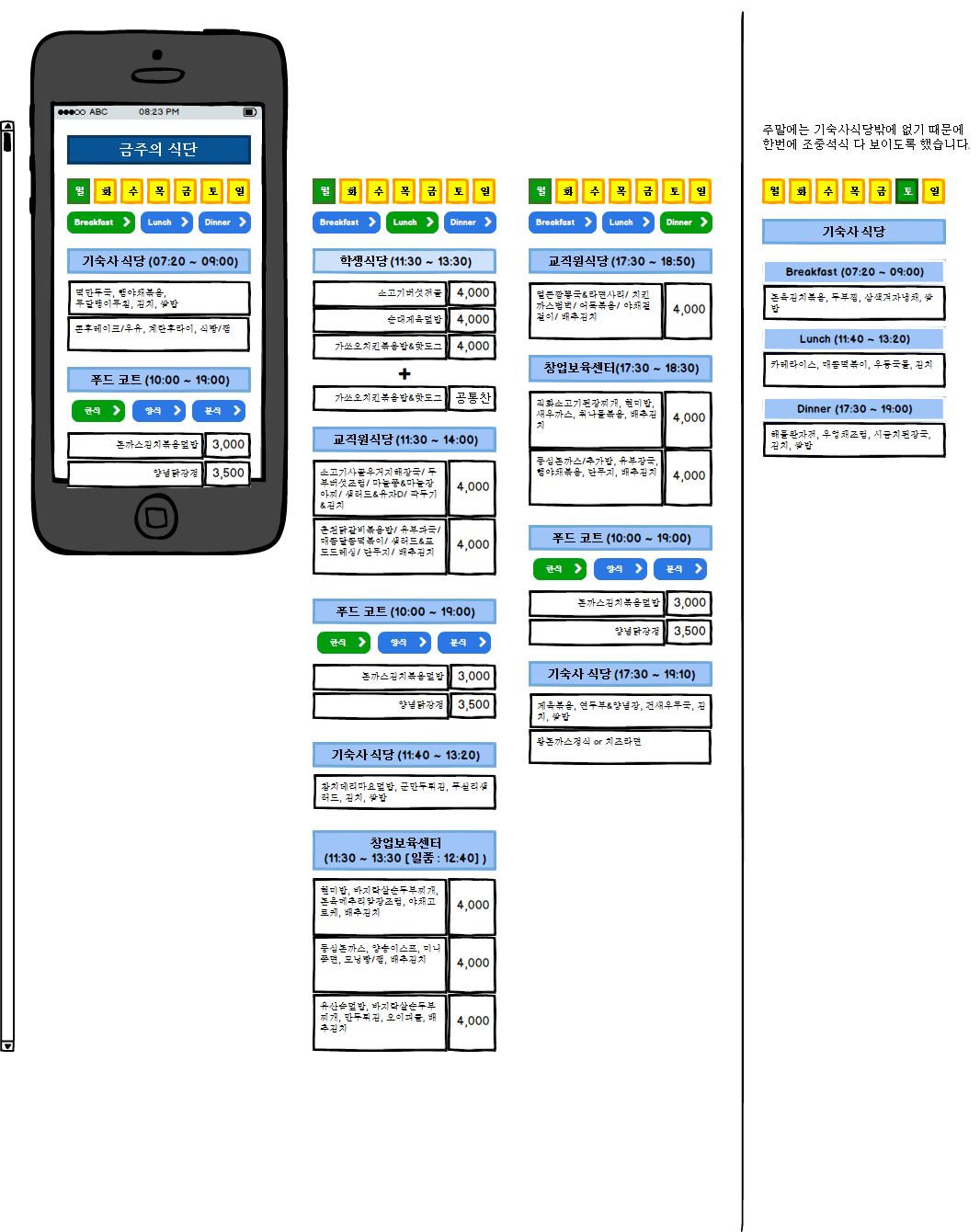
|  |  |
| --- | --- |
| **Use Case Name** | Set priority for the restaurants |
| **Goal in context:** | Show the menu by priority user set. |
| **Trigger** | User drag the panel of the restaurants. |
| **Precondition** | 1. In ‘Weekly Menu’ page, the button of the day among the ‘월’, ‘화’, ‘수’, ‘목’, ‘금’ is selected, not ‘토’, or ‘일’. 2. User logged in. |
| **Basic Path** | 1. In ‘Weekly menu’ page, press long the restaurants panel current shown. 2. The restaurants panel state is changed the draggable state, and enabled to move position by dragging up or down to other restaurants. 3. After position is moved, update user database for the priority. |
| **Alternative Paths** | 1. If user don’t login, this function is not enabled. 2. If ‘토’, or ‘일’ button is selected, this function is not enabled because that only ‘기숙사 식당’ is exists. |
| **Post condition** | 1. Update the priority changed by user to user setting database. 2. Show the menus applying priority user changed. |

Description: C:\Users\Frederica Bernkastel\AppData\Local\Microsoft\Windows\INetCache\Content.Word\usecase1 (1).png

## 3.3 Detailed Non-Functional Requirements

3.3.1 User Interface

- 3.3.1.1 Weekly Menu



### 3.3.2 Security

- Login issue

User password must be longer than 8 characters and Captcha method is implanted to keep the program from automated sign up

the system should protect itself and its sensitive data from unauthorized access, malicious, modification . if there is no reaction of client, the user will be log-out automatically. this will prevent the unauthorized access problem