**Object-Oriented Programming (SE271)**

**Team Project Proposal**

1. Title of the Project

Our team’s title of the project is <School Life Management System>.

1. Team Members

* Hyeonwoo Kwak / khw5218@dgist.ac.kr / 201911013
* Shinwook Park / bnb2011@dgist.ac.kr / 201911063
* Jaehyeon Park / jhpahk1441@dgist.ac.kr / 201911067
* Howoo Jang / janghowoo@dgist.ac.kr / 201911144

1. Motivation

Schedule, finance, and nutrients management are very important for DGIST students who live in busy daily lives. Although there are various programs that manage them on the market, they tend to be difficult to achieve multiple objectives with one program. This led to the creation of a customized student management program to efficiently manage the three items. For this program, it informs students of their schedules (deadlines, assignment submission dates, etc.) and allows them to reduce overspending in finance (pocket money management). In addition, the diet is recommended for balanced nutritional supplementation by managing nutrition (diet).

1. Implementation Notes

Below are the class and member functions we will implement.

* **Class** Nutrient

*Inpu*t: User inputs the menu that eats today as Data. This data stored as string type array.

*Recommend*: Based on input data (Menu), This program recommends a non-overlapping menu for next meal.

* **Class** Account

*Add*: Receive objects in a Date class and stored in an array.

*Remove:* Remove the Date received as an argument from the list of dates stored in the Account.

*Getsum:* Return a list of total expenditures from a to b days.

*Getall:* Return a sum of all the spending records during this month.

*Getavailable:* Return how much money is available afterwards based on the account so far.

* **Class** Date

*Add:* Get information about expenditures during the day through while statement. And the expenditure date is classified by category.

*Remove:* Remove the expenditure date that you want to erase from Data class object.

*Show:* Return the expenditure data for one day classified by category.

*Getall:* Return all the spending you spent in one day.

* **Class** Work

*Make :* Make a Work instance.

*Edit:* Edit attributes of the instance.

* **Class** Schedule

*Add:* Add schedule.

*Delete:* Delete schedule.

*Edit\_schedule:* Edit schedule.

*Show\_schedule:* Show saved schedule, ordered by deadline.

*Show\_important\_schedule:* Show saved schedule, ordered by importance.

1. Experiments / Results Expected

We will proceed with the test assuming that actual students use this program.

By using this program, it makes the schedule more efficient to manage using the data entered by the user and informs the future consumption life by analyzing consumption trends in finance. Finally, we recommend a menu without overlapping to provide nutrition management.

1. Metrics of Success in Detail

We will set up some evaluation criteria to quantify how well the program works. Below are some expected criteria to ensure that the program works fine. These criteria will be updated further as soon as we implement above class and member functions more specifically.

* Are member functions work well?
* Is the expenditure list well divided by category?
* Does the menu recommendation program not recommend duplicate menus?
* Are the remaining class implementations work well without errors?
* Are there any typo errors?

1. Plans of Final Project

We chose the inconveniences that come from schedule, finance, and diet management and then think about the solutions. In the final project, select friends (approximately 5 people) around us to use this program for about a week. Based on feedback from friends, in the final project, we will remove unnecessary parts from existing functions and add necessary parts that were not in the existing functions to create a more advanced program than the original demo version.