|  |
| --- |
| cs408B Computer science project |
| Software Requirement Specification |
| Team I |
|  |
| **Date** |
| **2010-03-10** |
|  |
| **Ver. 1.0** |

|  |  |
| --- | --- |
| Sangyong Gil | 20050730 |
| Sojung Kim | 20071132 |
| Gangmin Park | 20071154 |
| Kyeonghwan Jin |  |
| Sunghyuk Im | 20071201 |

|  |
| --- |
| This document is for the computer science project class B, Team 1. |

**TABLE OF CONTENTS**

[1. INTRODUCTION 4](#_Toc256024558)

[1.1 Purpose 4](#_Toc256024559)

[1.2 Scope 4](#_Toc256024560)

[1.3 Definitions 4](#_Toc256024561)

[1.4. Overview 5](#_Toc256024562)

[2. GENERAL DESCRIPTION 6](#_Toc256024563)

[2.1 Product Perspective 6](#_Toc256024564)

[2.2 Product Functions 6](#_Toc256024565)

[2.2.1. Product Function of Personal Account Book 6](#_Toc256024566)

[2.2.2. Product Function of Budget Planning and Quick Bookkeeping Features 7](#_Toc256024567)

[2.2.3. Product Function of Account Report 7](#_Toc256024568)

[2.2.4 Product Function of General Service 7](#_Toc256024569)

[2.3 User Characteristics 7](#_Toc256024570)

[2.4 Constraints and Assumptions 8](#_Toc256024571)

[3. FUNCTIONAL DESCRIPTION 9](#_Toc256024572)

[3.1 Use Case Diagram 9](#_Toc256024573)

[3.2 Functional Requirements 9](#_Toc256024574)

[3.2.1 General Requirements 10](#_Toc256024575)

[3.2.2 Requirements for Account Book 13](#_Toc256024576)

[3.2.3 Requirement for Account Report 19](#_Toc256024577)

[3.2.4 Requirement for Planning and Quick Book-keeping 20](#_Toc256024578)

[3.3 Non-functional Requirements 24](#_Toc256024579)

[3.3.1 Process Requirements 24](#_Toc256024580)

[3.3.1.1 Management Requirements 24](#_Toc256024581)

[3.3.1.2 Implementation Requirements 24](#_Toc256024582)

[3.3.1.3 Standards Requirements 24](#_Toc256024583)

[3.3.2 Product Requirements 24](#_Toc256024584)

[3.3.2.1 Usability Requirements 24](#_Toc256024585)

[3.3.2.2 Performance Requirements 24](#_Toc256024586)

[3.3.2.3 Reliability Requirements 25](#_Toc256024587)

[3.3.2.4 Availability 25](#_Toc256024588)

[3.3.2.5 Platform Constraints 25](#_Toc256024589)

[APPENDIX A: FIGURES AND TABLES INDEX 26](#_Toc256024590)

[APPENDIX B: AUTHOURSHIPS 26](#_Toc256024591)

1. INTRODUCTION

# 1.1 Purpose

This document presents the requirements specification for an automated account book system for KAIST students. These requirements represent the understanding of the system that has been conceptualized through the team meetings based on the topic which we chose for a term project of CS408. Then the purpose of this document is following:

1. To introduce the concepts of this system, then to elicit feedback.

2. To serve as a guide during system design step.

# 1.2 Scope

This document focuses primarily on the requirements for an automated account book system. Also this document also specifies requirements for each external source that will interface with this system.

# 1.3 Definitions

**Automated account book**

An account book whose bookkeeping processes are automated by computer

**Administrator**

A person who manages the system

**Customer**

A user who uses the system except the administrator

**Action**

Customer writes a record by pressing a shortcut

**Plan**

An item which includes the number of count, description, category and amount  
User will follow plans to manage one's economy better. Plan can be connected with a shortcut by user's choice.

**Plan list**

List which represents set of plan

**Mobile device**

A device that is capable of connecting to Internet with great mobility such as smart phones

**Transaction**

An event that causes change of balance such as buying or selling, which must be kept in an account book with descriptions: date, description, and amount of money.

**Transaction List**

A list of transactions occurred in any account, such as bank account

**Category Recognizing Pattern**

1 to 1 correspondence between a description and a category: it can be manually set by user, also can be set by transaction input.

# 1.4. Overview

The remainder of this document will cover an informal overview, functional requirements, and nonfunctional requirements of the system.

2. GENERAL DESCRIPTION

The system makes customer keep a personal account book easier than current account book software. System reads the files which are bank account and credit/debit card account transaction list. Then the system has the features that automatically categorized customer’s expenses and incomes to be fixed and customer defined categories. And also the system provides the features that customer keeps writing their expenses and incomes easier. For this, the system has shortcuts pages that easily input their expenses and incomes when customer spends, or earns.

Customer makes a plan of their specific expenses before actual spending by input how much customer going to spend in fixed duration. Customer can plan these expenses by times and amount of money to spend in meals, clothes, entertainments, transportation and so on. After customer plan their expenses, the system compares with the actual expenses. Finally the system reports to customer that makes customer know their status of one’s account and know customer’s plan is well achieved.

# 2.1 Product Perspective

The system is mainly operated on the web server. The web server stores and processes customer’ data, customer can connect the system server with mobile device and PC.

# 2.2 Product Functions

Following is main functions of the system.

## 2.2.1. Product Function of Personal Account Book

The system makes customer keep a personal account book thorough Web. The system provides all features of a personal account book. The personal account book has several entries for expense and income transactions. These entries consist of date, description, amount of money, category of transaction.

Customer may manually type entries or the system automatically fills the entries by using a bank/credit card transaction list from bank/credit-card companies. Bank/credit card company provides it with files.

When fill the entry automatically, there may occur exceptions, such as ambiguous description in a bank/credit transaction list. Because of ambiguity, the system cannot decide which categories the entry should belong. These kinds of exceptions will be handled by using data learning by recognizing customer’s historical usage.

## 2.2.2. Product Function of Budget Planning and Quick Bookkeeping Features

The system provides the budget planning features that make customer easily planning their budgets. Customer can plan their budgets by defining how many times customer going to do some actions in some periods. These actions are categorized same as the personal account book. For instance, if customer plans to have 30 school meals which cost 3,000 won per meal in one month, than inputs 30 times school meals, under the meal category.

After defining the plan, the system provides simple shortcuts thorough mobile web pages. The system counts right after customer’s ‘Action’ so that customer manages their plan without forgetting and input all the expenses customer did. Customer can also check whether a defined plan is carrying correctly or not by looking a progressive report which is provided by the system.

## 2.2.3. Product Function of Account Report

The system enables Customers to get the useful information about income and expense under the certain conditions: periodical or categorical. Regarding a periodical report, Customers can compare the income and expense with other durations. Also, a categorical report shows the income and expense according to the category within in the certain duration.

## 2.2.4 Product Function of General Service

People who want to use this system, ones should sign up, sign in. After using the service, the system users signs out to exit. Moreover, Customers and administrator(s) can communicate each other through using a bulletin board.

# 2.3 User Characteristics

System user can be classified into two: customer and administrator.

In this project, we assume that customer has the following characteristics:

1. Students of KAIST

2. Want to prevent over-budget but hard to keep an account book

3. High usage of credit/debit card.

4. Prefer of doing something with PC than by hands

Administrator has following characteristics:

1. An administrator familiar with all system capabilities.

# 2.4 Constraints and Assumptions

Constraints and assumptions are following:

1. Customer should be able to access Internet thorough their PC or mobile device.

2. Customer should use the Internet banking services operated by bank/credit card company to download the bank transaction list.

3. Transaction descriptions can be mismatched by ambiguous and insufficient information in bank/credit card transaction list.

4. Bank/credit card transaction list would be sourced from a bank/credit card company as a file.

5. Assume all bank/credit card transaction list consist date, description and amount of money.

3. FUNCTIONAL DESCRIPTION

# 3.1 Use Case Diagram

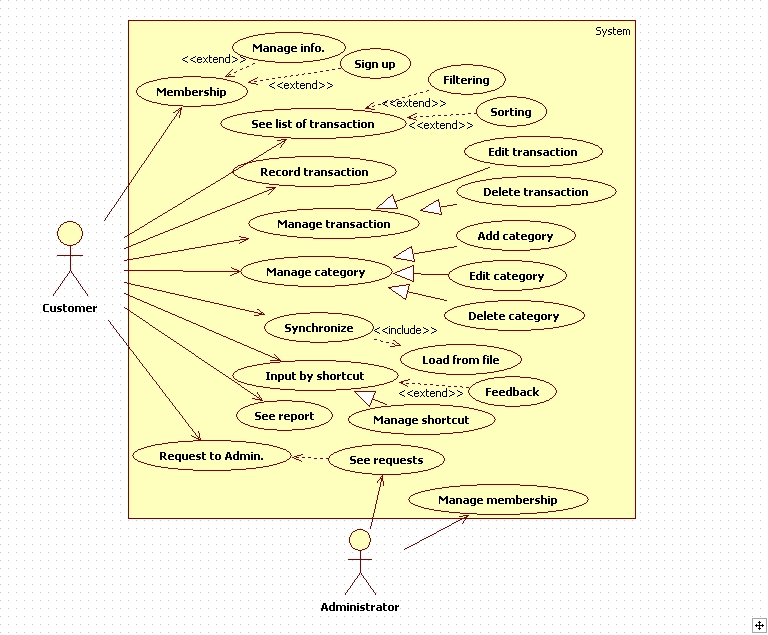


Figure : Use case diagram

# 3.2 Functional Requirements

The functional requirements are presented in the following form:

* *Description*

A description of the specific requirement

* *Input*

A description of the input(s) that the requirement need

* *Processing*

A description of what the system should do with the input(s)

* *Output*

A description of the response or new state of the system

* *Priority*

Importance of functional requirement denoted by low, medium, or high

The input, processing and output sections are only specified when needed.

## 3.2.1 General Requirements

|  |
| --- |
| FR1\_1 Sign-up  FR1-2 Sign-in  FR1-3 Manage Account Information  FR1-3-1 Edit Account Information  FR1-3-2 Delete Account Information for Customer  FR1-4 Manage Memberships  FR1-4-1 Show a status list of memberships  FR1-4-2 Delete membership  FR1-5 Manage Customer Relationships  FR1-5-1 Ask support  FR1-5-2 Response request |

Table : List of general requirements

**Functional Requirement 1\_1: Sign-up**

* *Description*
  + When new customer wants to use this software system, a customer should be a member of the system first. So the system provides the application form.
* *Input*

Customer inputs following information: ID, password, e-mail

* *Processing*

Create new Customer record.

* *Priority*

High

**Functional Requirement 1\_2: Sign-in and Sign-out**

* *Description*

When customer wants to use the system, customer signs in to the system. And customer wants to exit of this software system; customer should sign-out to the system.

* *Input*

Customer’s ID and password

* *Processing*

Verify that ID and password is correct or not.

* *Output*

If the ID and password are correct, the system displays the next page. Unless they are correct, the message is displayed to notify incorrectness.

* *Priority*

High

**Functional Requirement 1-3: Manage Account Information**

* *Description*

The system provides customer and administrator(s) to manage customer’s accounts.

**Functional Requirement 1-3-1: Edit Account Information**

* *Description*

When a customer wants to change one’s account information and then customer can edit their account information.

* *Input*

Edited information about the customer account: password, e-mail

* *Processing*

Update the customer account.

* *Priority*

Low

**Functional Requirement 1-3-2: Delete Account Information for Customer**

* *Description*

If customer wants to stop using service, the customer can withdraw from the system.

* *Input*

Customer’s ID and password

* *Processing*

Verify that ID and password is correct or not and eliminate the customer account

* *Output*

If the ID and password are correct, display the page that shows the completion of eliminating. Unless they are correct, the message to notify incorrectness will be displayed.

* *Priority*

Low

**Functional Requirement 1-4: Manage Memberships**

* *Description*

The administrator manages memberships.

**Functional Requirement 1-4-1: Show a status list of memberships**

* *Description*

The administrator wants to analysis the status of memberships; how many customer are existed

* *Input*

Administrator’s ID and password

* *Processing*

Verify that ID and password is correct or not.

* *Output*

A list consisted of the number of customer currently signed up

* *Priority*

Low

**Functional Requirement 1-4-2: Delete Membership**

* *Description*

The administrator wants to ban a customer.

* *Input*

Administrator’s ID and password, a customer’s ID

* *Processing*

Verify that ID and password is correct or not.

* *Priority*

Low

**Functional Requirement 1-5: Manage Customer Relationship**

* *Description*

Customer asks questions and claim to administrator(s) and administrator(s) answers it.

**Functional Requirement 1-5-1: Ask support**

* *Description*

Customer asks questions and claim to administrator(s).

* *Input*

Title and contents to request

* *Processing*

Save the contents and requests

* *Priority*

Low

**Functional Requirement 1-5-2: Response request**

* *Description*

Administrator(s) responses the requested support.

* *Input*

Title and contents to response

* *Processing*

Save the contents and responses

* *Priority*

Low

## 3.2.2 Requirements for Account Book

|  |
| --- |
| FR2-1: Show an Account Book Transaction List  FR2-1-1: Show the Transaction List  FR2-1-2: Filter the Transaction List  FR2-1-3: Sort the Transaction List  FR2-2: Record a Transaction  FR2-3: Manage the Transaction  FR2-3-1: Edit the Transaction  FR2-3-2: Delete the Transaction  FR2-4: Synchronize with an Bank/Credit Card Transaction List  FR2-4-1: Import with a Bank/Credit card Transaction List  FR2-4-2: View Suggestion of Category for Each Transaction  FR2-4-3: Manage Synchronizing Process  FR2-5: Export the Transaction List  FR2-6: Manage the Transaction Categories  FR2-6-1: Add New Transaction Category  FR2-6-2: Edit the Transaction Category  FR2-6-3: Delete the Transaction Category  FR2-7: Add a Category Recognizing Pattern  FR2-8: Manage Category Recognizing Patterns |

Table : List of account book functional requirements

**Functional Requirement 2-1: Show an Account Book Transaction List**

* *Description*

The system displays customer’s account book transaction list.

**Functional Requirement 2-1-1: Show the Recent Transaction List**

* *Description*

Customer wants to know current transaction list, and then the system displays accumulated transaction list.

* *Output*

Recently accumulated transaction list

* *Priority*

High

**Functional Requirement 2-1-2: Filter the Transaction List**

* *Description*

Customer wants to see the transaction list under the specific condition, and then the system provides this feature.

* *Input*

Filtering conditions; an interval of two dates or a word that customer wants to find

* *Processing*

The system fetches customer’s personal account information which corres-ponds to the input conditions.

* *Output*

Filtered transaction list

* *Priority*

Medium

**Functional Requirement 2-1-3: Sort the Transaction List**

* *Description*

Customer wants to see the sorted list, and then the system provides this list.

* *Input*

Sorting conditions; amount of money, date, description, or category

* *Processing*

The system fetches the transaction list and sorts it.

* *Output*

The sorted transaction list

* *Priority*

Low

**Functional Requirement 2-2: Record a Transaction**

* *Description*

Customer wants to input the transaction, and then the system provides adequate form to input the transaction.

* *Input*

Date, description, amount of money and category of transaction

* *Processing*

Add the new transaction to the customer’s account book

* *Priority*

High

**Functional Requirement 2-3: Manage the Transaction**

* *Description*

Customer edits or deletes transactions.

**Functional Requirement 2-3-1: Edit the Transaction**

* *Description*

Customer wants to change a transaction description, amount, date, and then customer can modify their transaction information.

* *Input*

Date, description, amount or category

* *Processing*

Update the customer’s account book

* *Priority*

High

**Functional Requirement 2-3-2: Delete the Transaction**

* *Description*

Customer deletes their transaction information.

* *Input*

The transaction which customer wants to delete

* *Processing*

Delete the customer’s account book.

* *Priority*

High

**Functional Requirement 2-4: Synchronize with a Bank/Credit Card Transaction List**

* *Description*

The system automatically classifies date, description, and amount of money in bank/credit card transaction list from the bank/credit card company. And the system determines the category of the each transaction. This process is called ‘Synchronization’. Customer can see and modify the synchronization process.

**Functional Requirement 2-4-1: Import with a Bank/Credit Card Transaction List**

* *Description*

Customer inputs the bank/credit card transaction list file which is provided by bank/credit card company, and then the system automatically classifies the data of date, transaction description, and amount of money.

* *Input*

Transaction list from bank or card company

* *Processing*

1. The system classifies the date, description, amount of money from bank/credit card transaction list.
2. The system temporarily stores these data for following process.

* *Priority*

High

**Functional Requirement 2-4-2: View Suggestion of category for each transaction**

* *Description*

In the import process, customer input the bank/credit card transaction list, and then the system temporarily stored automatically classified data. After import, user can see the suggestion of category for each transaction.

Following is detail of what system does. From stored data of import process, the system tries to find a transaction which has same description as each transaction of saved data. The system can find this correspondence not only in past transactions but also in the category recognizing patterns.

* *Input*

Stored classified data generated from import

* *Processing*

1. The system tries to find a description which is same as the description of the current transaction.
2. If found, save the description of the current transaction and the category name of the found transaction.
3. Iterate 1 and 2 until transaction remains no more.
4. Merge saved data

* *Output*

Transaction list and suggested categories of each transaction

* *Priority*

High

**Functional Requirement 2-4-3: Manage Synchronizing Process**

* *Description*

Customer handles exceptions occurred during import process manually.

* *Input*

Edited category of the suggestions

* *Processing*

The system updates the category of each transaction.

* *Output*

Updated transaction list

* *Priority*

High

**Functional Requirement 2-5: Export the Transaction List**

* *Description* Customer can get the copy of one’s account book out of the system.
* *Input*

The interval of two dates

* *Processing*

The transaction lists of given interval is transformed into a file.

* *Output*

A file which contains the records of given period

* *Priority*

Low

**Functional Requirement 2-6: Manage the Transaction Categories**

* *Description*

Customer edits the category.

**Functional Requirement 2-6-1: Add New Transaction Category**

* *Description*

Customer creates a category.

* *Input*

A name of category

* *Processing*

Add the name of category into the list of categories

* *Output*

List of categories

* *Priority*

Medium

**Functional Requirement 2-6-2: Edit the Transaction Category**

* *Description*

Customer edits the name of the category.

* *Input*

The category which customer wants to change the name, and new name

* *Processing*

Change the name of selected category

* *Output*

Updated list of categories

* *Priority*

Medium

**Functional Requirement 2-6-3: Delete the Transaction Category**

* *Description*

Customer deletes some category.

* *Input*

The category which customer wants to delete

* *Processing*

Delete the selected categories from the list of categories.

* *Output*

Updated list of categories

* *Priority*

Medium

**Functional Requirement 2-7: Add Category Recognizing Pattern**

* *Description*

Customer wants to classify a certain transaction description (from bank/credit card company) into a certain category, and then customer inputs the category and description to be matched in the system. So the description will be able to be classified automatically. In other words, customer teaches the system.

* *Input*

A description, a category

* *Processing*

1. The matched data of descriptions and a category are stored.
2. The system updates transaction list by newly matched data.

* *Priority*

Medium

**Functional Requirement 2-8: Manage Category Recognizing Patterns**

* *Description*

Customer wants to know the list of matched descriptions and categories, and then the system shows it. And then customer can delete the existing patterns.

* *Input*

A request for the list of matched descriptions and categories

* *Processing*

If customer wants to delete a pattern, the system deletes it.

* *Output*

A list of matched descriptions and categories

* *Priority*

Medium

## 3.2.3 Requirement for Account Report

|  |
| --- |
| FR3-1: Show a Periodical Total Income and Expense Report  FR3-2: Show a Categorical Total Income and Expense Repot |

Table : List of account report functional requirements

**Functional Requirement 3-1: Show a Periodical Total Income and Expense Report**

* *Description*

Customer gets the information about total income and expense within a certain interval of two dates. Also, customer can compare them with other durations.

* *Input*

 The interval of two dates

* *Output*

Customer's total incomes and expenses information within the duration as a chart and graph

* *Priority*

Medium

**Functional Requirement 3-2: Show a Categorical Total Income and Expense Report**

* *Description*

Customer gets the arranged transaction information according to the category.

* *Input*

The category, and the interval of two dates

* *Output*

 Customer’s transaction list according to the category within the duration

* *Priority*

Medium

## 3.2.4 Requirement for Planning and Quick Book-keeping

|  |
| --- |
| FR4-1: Manage Plan List  FR4-1-1: Set up a Target Budget and Duration of Plan List  FR4-4-2: Add a Plan  FR4-4-3: Edit a Plan  FR4-4-3: Delete a Plan  FR4-2: Show a Short Summary of the Plans  FR4-3: Manage the Shortcuts  FR4-3-1: Create Shortcuts  FR4-3-2: Delete Shortcuts  FR4-4: Shortcut Input and Showing a Feedback Message  FR4-5: Display a Progressive Report |

Table : List of planning and quick book-keeping requirements

**Functional Requirement 4-1: Manage Plan List**

* *Description*

Customer sets up one’s plan list and duration for that plan list. Plan list consists of a target budget and plans which are classified under the categories. The plan can be a school meal under the meal category, movie tickets under entertainment category, and so on. Customer specifies these plans and the number of plans to be done in the duration.

**Functional Requirement 4-1-1: Set up a Target Budget and Duration of Plan List**

* *Description*

Customer sets up a target budget for a plan list and duration of that plan list.

* *Input*

Target amount of expense, first date of the plan list, the last date of the plan list

* *Processing*

The system stores the target budget and the duration.

* *Output*

The newly created plan list

* *Priority*

High

**Functional Requirement 4-1-2: Add a Plan**

* *Description*

Customer adds a plan.

* *Input*

Expected plan to be spent, expected money for a plan, and expected number of times to be spent on a plan

* *Processing*

The system adds the plan into the plan list.

* *Output*

The updated plan list

* *Priority*

High

**Functional Requirement 4-1-3: Edit a Plan**

* *Description*

Customer edits a plan.

* *Input*

A selected plan, expected money for a plan, and expected number of times to be spent on the plan

* *Processing*

The system edits the plan

* *Output*

The updated plan list

* *Priority*

High

**Functional Requirement 4-1-4: Delete a Plan**

* *Description*

Customer deletes a plan.

* *Input*

A selected plan

* *Processing*

The system deletes the plan from the plan list.

* *Output*

The updated plan list

* *Priority*

High

**Functional Requirement 4-2**: **Show a Short Summary of the Plans**

* *Description*

The system displays how many times customer will be able to spend on each plan in remaining duration.

* *Processing*

Calculate remaining times of plans and the amount of money that the customer can spend on each category and plan

* *Output*

Short summary

* *Priority*

High

**Functional Requirement 4-3: Manage the Shortcuts**

* *Description*

Shortcuts help customer record a transaction easily. These shortcuts can be managed by customer; the system provides personalized shortcuts page.

**Functional Requirement 4-3-1: Create Shortcuts**

* *Description*

Customer creates shortcuts.

* *Input*

Description, amount, and category

* *Processing*

Add new shortcut into the list of shortcuts.

* *Priority*

Medium

**Functional Requirement 4-3-2: Delete Shortcuts**

* *Description*

Customer deletes shortcut.

* *Input*

Single (multiple) selection(s) from the list of shortcuts

* *Processing*

Delete the selected shortcut(s) from the list of shortcuts.

* *Priority*

Medium

**Functional Requirement 4-4: Shortcut Input and Showing a Feedback Message**

* *Description*

Customer does ‘Action’ after spending a moment ago, in customer’s account book by just selecting a shortcut of the category. Just after that, a feedback message consisting of the remaining number of the ‘Action’ shows up.

* *Input*

Selection of a shortcut of a category

* *Processing*

1. Transaction is recorded on the customer’s account book
2. Identify whether the shortcut of the Action is connected to plan.
3. If not connected, make blank feedback message.
4. If connected, the amount of money of the plan and description is known from the plan.
5. Identify the amount of money and how many times the customer spent on the plan in the current plan list duration
6. Calculate remaining times and the amount of money that the customer can spend on the plan
7. The feedback message includes The remaining times and the amount of money.

* *Output*

The feedback message

* *Priority*

Medium

**Functional Requirement 4-5: Display a Progressive Report**

* *Description*

Customer can know performances of one’s expenses compared with previous (or current) established plan list.

* *Processing*

1. Calculate the differences between plan list and the actual expenses for each plan list
2. The overspent ratios are calculated for each plan list

* *Output*

Progressive report

* *Priority*

Medium

# 3.3 Non-functional Requirements

## 3.3.1 Process Requirements

### 3.3.1.1 Management Requirements

Project schedule is following the class tentative schedule as followings

|  |  |
| --- | --- |
| Tasks to be done | Date |
| Preliminary specification | 2010/ 2/ 24/ Wed |
| Revised specification | 2010/ 3/ 3/ Wed |
| Final specification | 2010/ 3/ 10/ Wed |
| System design | 2010/ 3/ 17/ Wed |
| Revised design | 2010/ 3/ 31/ Wed |
| Preliminary test plan | 2010/ 4/ 14/ Wed |
| Final test plan | 2010/ 4/ 21/ Wed |
| System to be tested | 2010/ 4/ 28/ Wed |
| System to be tested | 2010/ 5/ 12/ Wed |
| Customer's manual | 2010/ 5/ 19/ Wed |

Table : Schedule

### 3.3.1.2 Implementation Requirements

Since this system operates on web, appropriate languages for web programming and the corresponding tools will be used to implement this system.

### 3.3.1.3 Standards Requirements

Our team will adopt the waterfall model for development process, which is a sequential process model, consists of these phases: requirements analysis, design, implementation, testing, integration, and maintenance

## 3.3.2 Product Requirements

### 3.3.2.1 Usability Requirements

The system must provide tips and the system manual with the new customer to learn how to use the system properly and easily.

### 3.3.2.2 Performance Requirements

The system must response any request within 3 seconds with 95% ratio.

The system must be able to handle 10 requests simultaneously.

The system must be able to handle 10 connected users concurrently.

### 3.3.2.3 Reliability Requirements

The system must have calculation errors less than 3% from all the data.

Customer’s data must be secured, and hidden to the other customer.

### 3.3.2.4 Availability

The system must be able to run on PCs and mobile devices.

The system must be available for more than 23 hours a day, and must be available in 7 days a week.

### 3.3.2.5 Platform Constraints

The system must be properly operated on both web-browsers and mobile web-browsers.

APPENDIX A: FIGURES AND TABLES INDEX

Figure 1: Use case diagram…………………………………………………………………………………………………………………..9

Table 1: List of general requirements…………………………………………………………………………………………..…….10

Table 2: List of account book functional requirements…………………………………………………………………..…13.

Table 3: List of account report functional requirements…………………………………………………………………...19

Table 4: List of planning and quick book-keeping requirements……………………………………………………...20

Table 5: Schedule………………………………………………………………………………………………………………………………..24

APPENDIX B: AUTHOURSHIPS

|  |  |  |
| --- | --- | --- |
| Section | Page | Author |
|  | 4 | Gangmin Park |
|  | Gangmin Park |
|  | Gangmin Park |
|  | 6~7 | Gangmin Park |
|  | Gangmin Park |
|  | 7 | Gangmin Park |
|  | 8 | Gangmin Park |
|  | 9 | Sunghyuk Im |
|  | 10~13 | Sojung Kim |
|  | 13~19 | Sunghyuk Im |
|  | 19~20 | Sojung Kim |
|  | 20~23 | Sangyong Gil |
|  | 24 | Kyeonghwan Jin |
|  | 24~25 | Kyeonghwan Jin |