Software Requirements Specification

for

Student Budget Management System

Version 2.0 approved

Sufi Nukman Bin Shakimi

Muhammad Alif Luqman Bin Afandi

Nurhamizah Bt. Husin

Eman Hussein

15 May 2020

Table of Contents

1. Introduction

1.1 Purpose

Student Budget Management (SBM) is a web platform system where it focuses on managing budget and finance specifically for students. This project is related to the finance field among UPM students. Nowadays, students are facing problems in managing their money in a proper way due to lack of management skill and ignorance of importance of management skill. That's the reason for us to choose students as our users. Therefore, we think that through this SBM project, we can help the students to manage their finance. We believe that this project can guide them the way to manage their finance throughout their studies based on their income like loans or scholarships. This project also improves the weaknesses in the current existing systems since most of them are not focusing on students' financial management. Therefore, this project is going to be more focused on students other than existing systems do.

1.2 Scope

The SBM focuses on managing UPM students' finances based on their income and pocket money every semester.

1.3 Definitions, Acronyms, and Abbreviations

Terms	Definitions
SBM	Student Budget Management
SRS	Software Requirement Specification
UPM	Universiti Putra Malaysia

1.4 References

[1] Lecture notes, Software Requirement Engineering.
[2] https://softwarekno.blogspot.my/2016/09/waterfall-model.html
[3] https://gyires.inf.unideb.hu/GyBITT/07/ch01.html

2. Overall Description

2.1 Product Perspective

Student Budget Management System (SBMS) is intended to maintain the financial challenges faced by UPM students, and it benefits them effectively by providing features that allows them to record their budget, expenses and can view the effective allocations performed by the system and follow it. SBMS is a web platform management tool that requires a client\web browser, web server and database. Firstly, a web browser is responsible to deliver the requested pages and display to users, a web server is responsible for satisfying user's requests and processes them through the Internet. Lastly, the database is the storage element of user's data. Figure 2.1 illustrates the interaction.

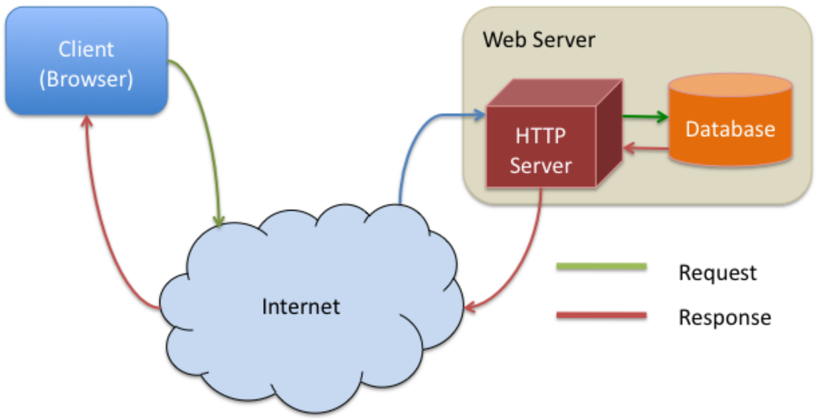


Figure 1: Product Perspective Diagram Illustration

2.2 Product Functions

In this section, the use case model is illustrated as well as described briefly. The system's use case diagram is shown in Figure 2.2.

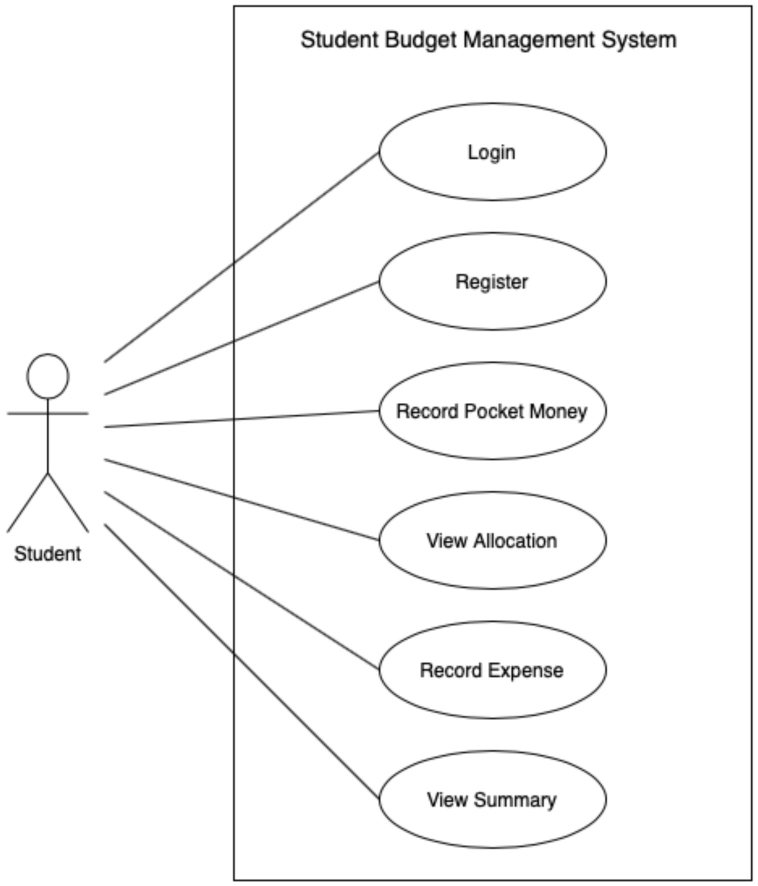


Figure 2: Use Case Diagram

Identifier	Use cases	Description
UC1	Login	This use case allows users to login and access the system features.
UC2	Register	This use case allows new users to register for an account.
UC3	Record Pocket Money	This use case allows students to input their total pocket money.
UC4	View Allocation	This use case allows students to view the allocation of each category calculated by the system.
UC5	Record Expense	This use case allows students to input their daily spent expenses.
UC6	View Summary	This use case allows students to view their summary of spending.

2.3 User Classes and Characteristics

SBM contents only involve one type of user that is student. As a result, SBM will focus only for students. They can manage their budget by inserting their pocket money for a semester. After that, the system will automatically divide their pocket money into some categories so that the user can spend their money by the system's guidance without getting over budget. The system will also tell them how much can be spent on certain things. This application can be used by UPM students.

2.4 Operating Environment

OE1 - The system shall operate in all modern browsers.
OE2 - The system shall operate in different operating systems such as Windows and iOS.

2.5 Design and Implementation Constraints

DC1 - SBM is created for university students since it is a public application.

2.6 Assumptions and Dependencies

AD1 - It is assumed that users shall have fair Internet connection to access the system.
AD2 - It is assumed that users have experience in using basic web application features.
AD3 - It is assumed that users understand English language.
AD4 - The date picker is manipulated from a third-party component library named JCalendar.
AD5 - The summary report provided by the system is manipulated from a third-party component library of graphs and charting.

3. External Interface Requirements

3.1 User Interfaces

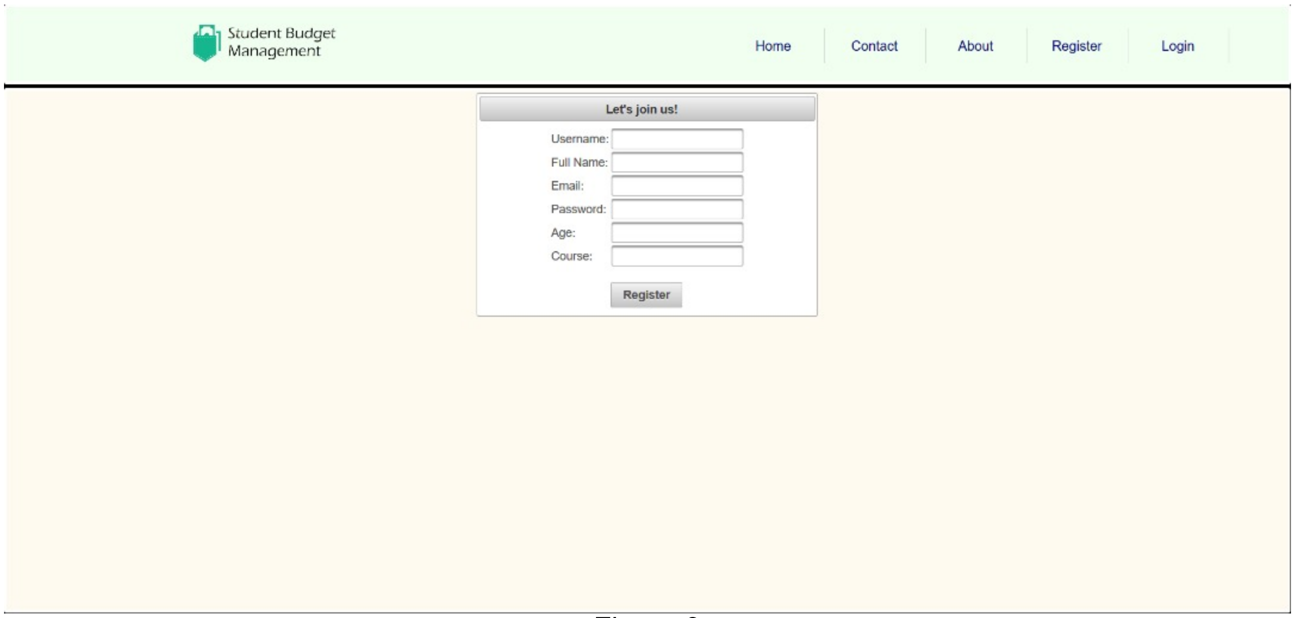


Figure 3

Figure 3 shows the interface of the ‘Register’ use case. It shows six text fields which are available for the user to enter their username, full name, email, password, age and course. The ‘Register’ button will let the system store the user’s information.

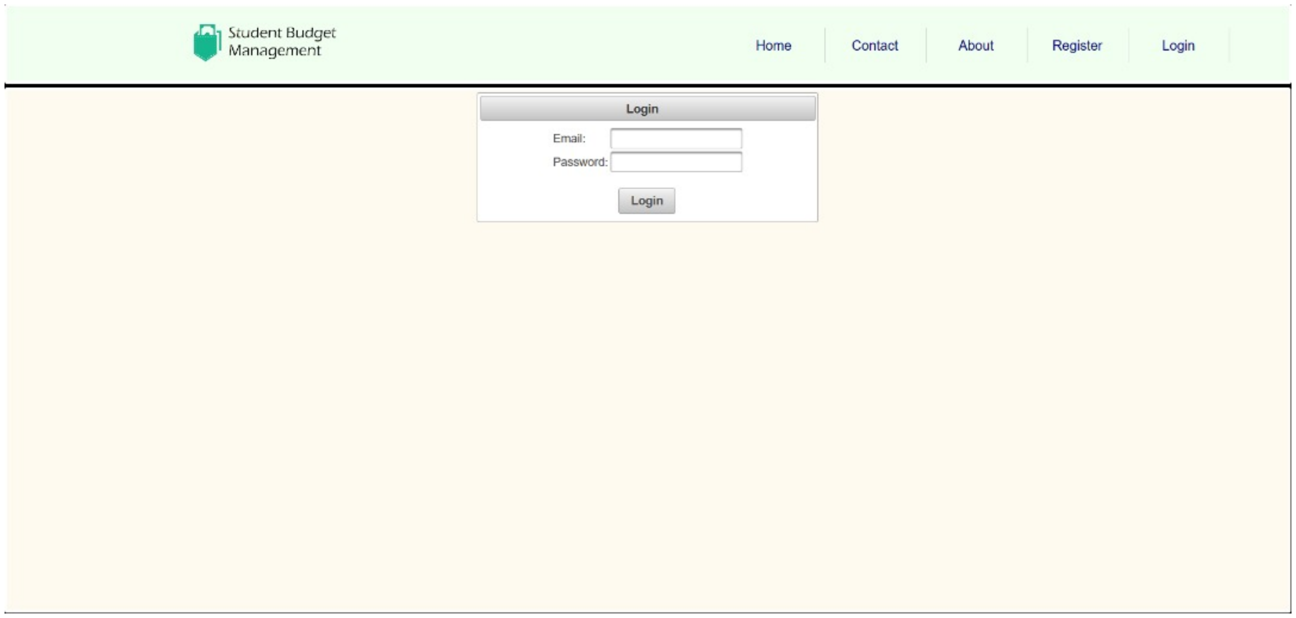
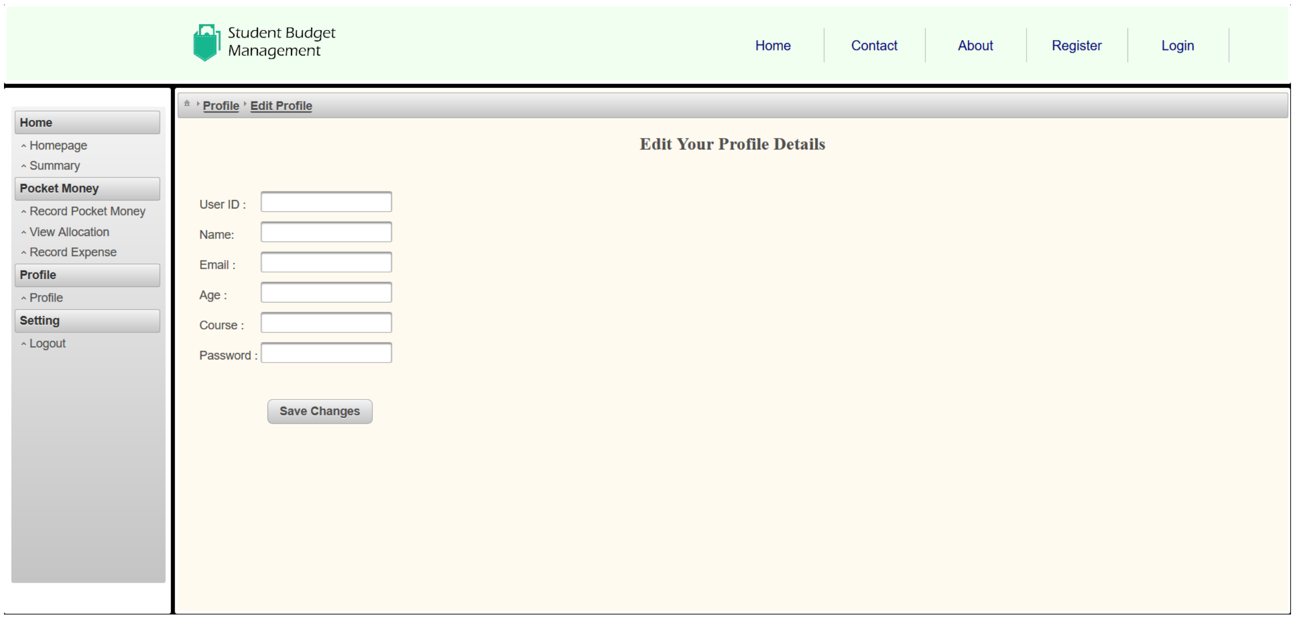


Figure 4

Figure 4 shows the interface of the ‘Login’ use case. It shows two text fields which are available for the user to sign in the SBM with email and password. The ‘Login’ button will let the system verify and allow users to sign in.

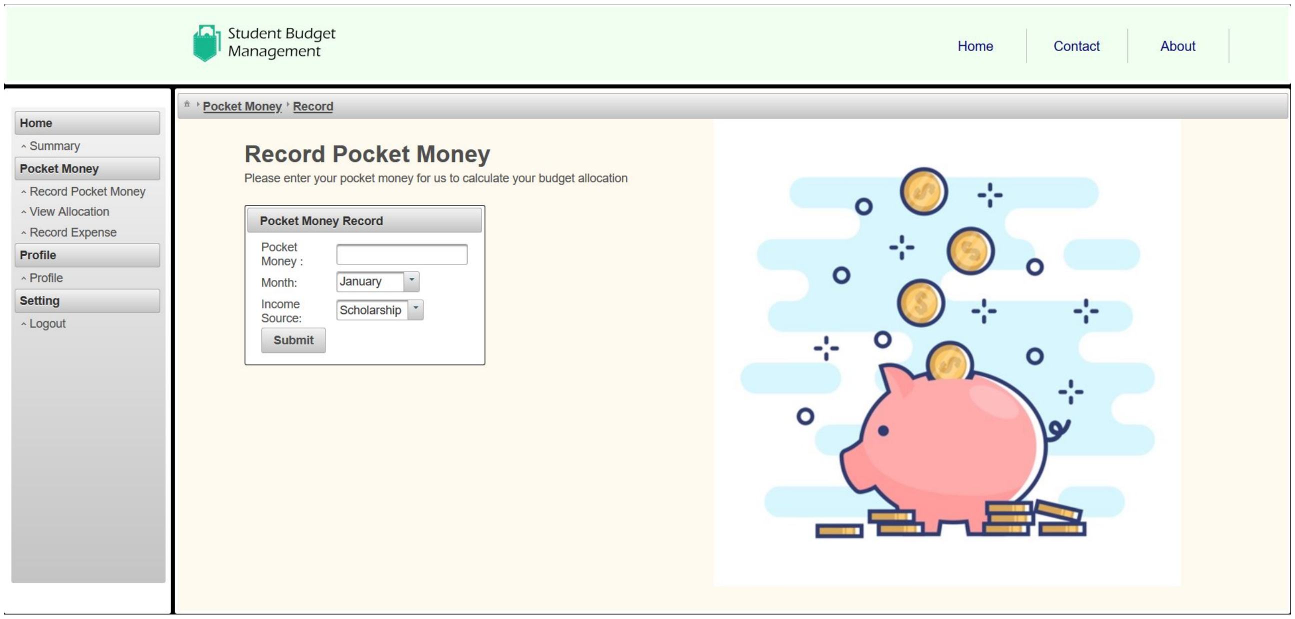


Edit Your Profile Details

User ID :	
Name:	
Email :	
Age :	
Course :	
Password :	

Figure 5

Figure 5 shows the interface for editing the user's profile. It shows the editable user's profile information. Users can update his/her profile information by editing the text fields. The 'Save Changes' will allow the system to update the user's profile information.



Record Pocket Money

Please enter your pocket money for us to calculate your budget allocation

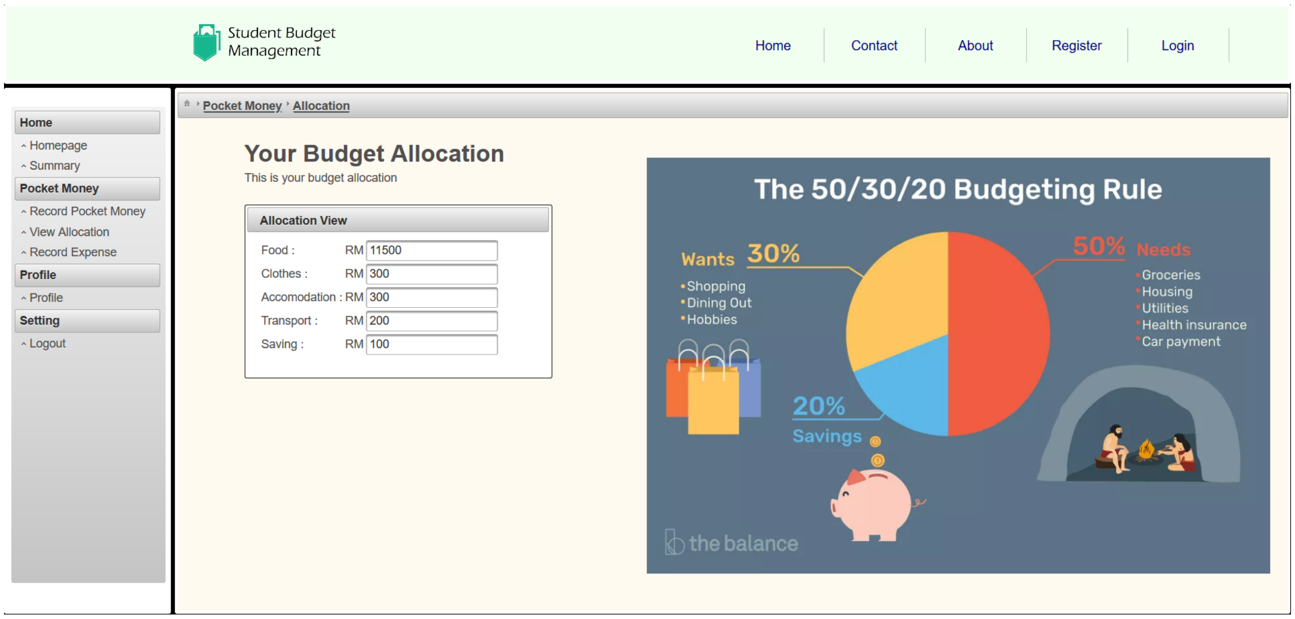
Pocket Money Record

Pocket Money :	
Month:	January
Income Source:	Scholarship

Submit

Figure 6

Figure 6 shows the interface of ‘Record pocket money’ use case. There is one text field allowing the user to enter his/her pocket money and a drop down list for the user to choose the type of the pocket money. The ‘Submit’ button triggers the system to store the amount of the pocket money.



Your Budget Allocation

This is your budget allocation

Allocation View

Food :	RM 11500
Clothes :	RM 300
Accomodation :	RM 300
Transport :	RM 200
Saving :	RM 100

Figure 7

Figure 7 shows the interface of the 'View Allocation' use case. It shows the allocations of the pocket money into 5 categories: food, clothes, accommodation, transport and saving.

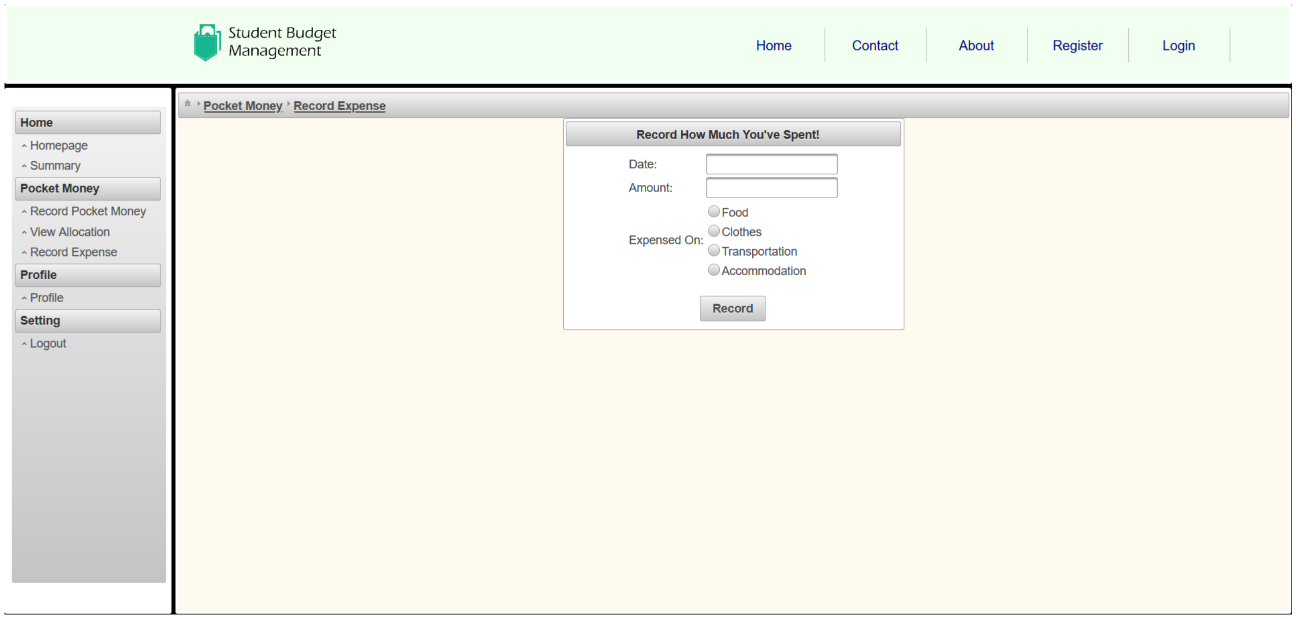


Figure 8

Figure 8 shows the interface of 'Record Expense' use case. It shows two text fields for the user to enter the date and amount he/she has spent. Besides, there are radio buttons for the categories that he/she already spent.

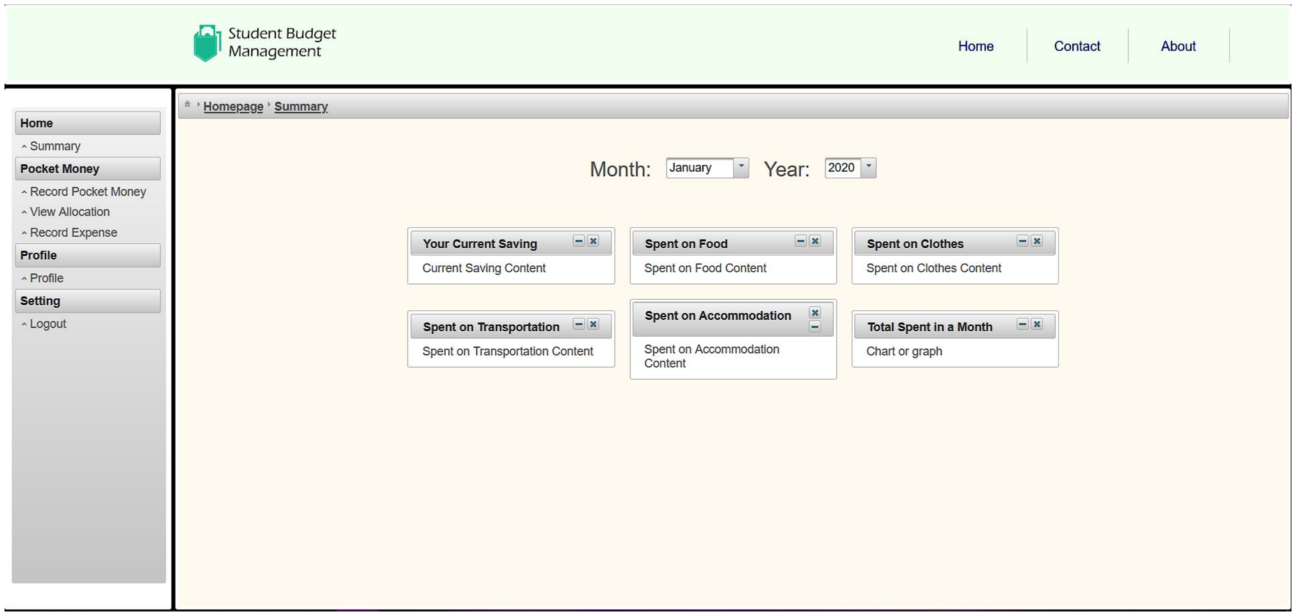
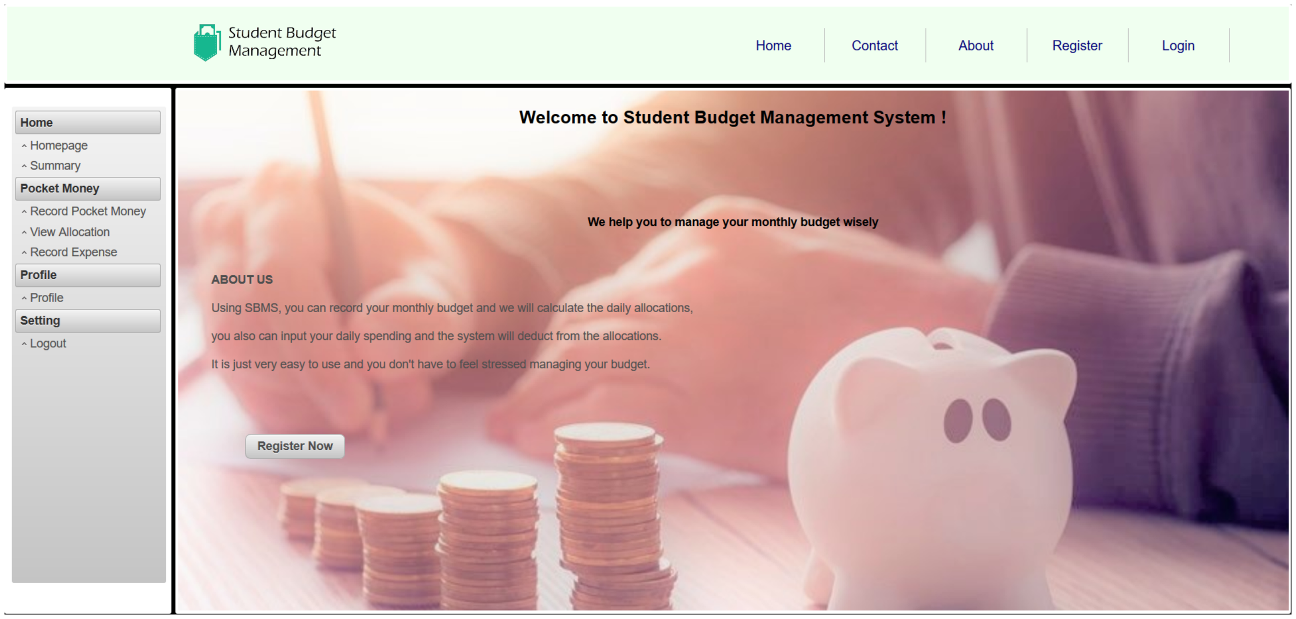


Figure 9

Figure 9 shows the interface of the 'View Summary ' use case. It shows the summary of the total money the user has saved, the total amount of money the user has spent and total spent in a month.



Welcome to Student Budget Management System !

We help you to manage your monthly budget wisely

ABOUT US

Using SBMS, you can record your monthly budget and we will calculate the daily allocations,

you also can input your daily spending and the system will deduct from the allocations.

It is just very easy to use and you don't have to feel stressed managing your budget.

Figure 10

Figure 10 shows the interface of the homepage of SBM. It gives a little bit of a description of SBM to the new user.

3.2 Hardware Interface

For Student Budget Management to run, you will need a PC (regardless of its operating system, it can be Windows or Linux) or Mac, with an operating system. A functional standard keyboard and mouse is required.

3.3 Software Interface

- Browser: Google Chrome, Mozilla Firefox, Internet Explorer, Safari and Opera.
- Server: Java Application Server
- Database: SQL Database
- Libraries: JFreeChart, JCalendar

4. System Use Cases

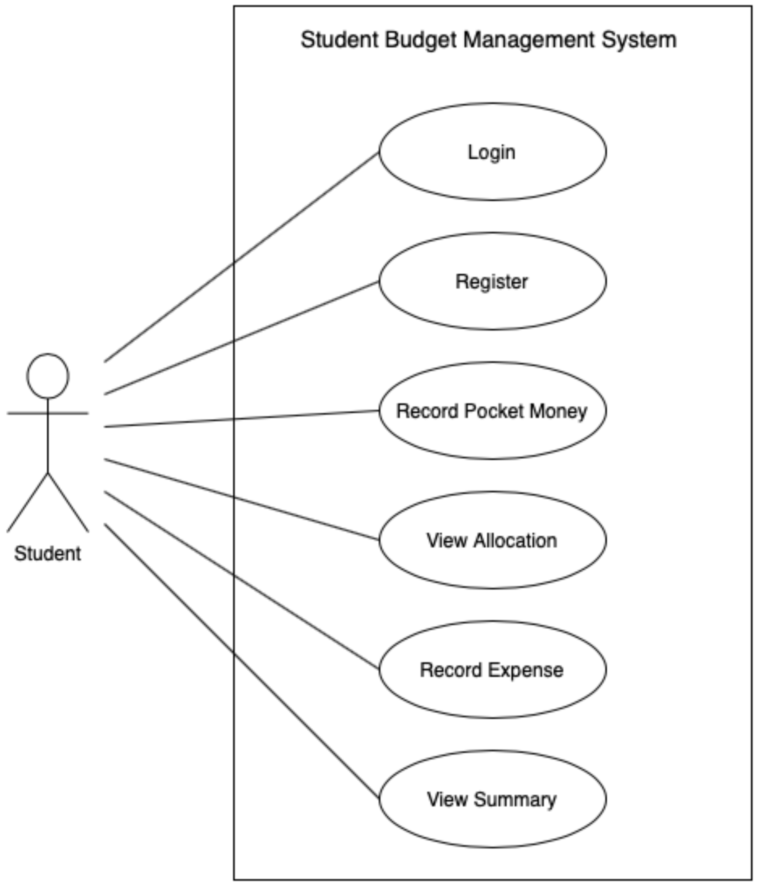


Figure 11

Figure 11 is a use case diagram. There are 6 use cases, they are Login, Register, Record Pocket Money, View Allocation, Record Expenses and View Summary.

4.1 Login (UC1)

Description	This use case allows users to login and access the system features.
Objective	To store the login data safely only for users.
Actors	Students
Basic Flow of Events	1. User has to press login button 2. System will display text field to insert ID and password 3. User has to insert ID and password 4. User has to press Login button 5. System will validate password 6. User can login successfully
Alternative Flow of Events	1. At step 2.1.5 error message is prompt and ask the user to enter again the password
Preconditions	Starts when the user opens the web application.
Postconditions	User is successfully logged into the system and able to access its features

4.2 Register (UC2)

Description	This use case allows new users to register for an account.
Objective	To store students personal information.
Actors	Students
Basic Flow of Events	1. User have to click create profile button 2. System will display empty form 3. User have to fill in the information 4. System will ask the user to confirm 5. User have to confirm the profile 6. System will store the profile information 7. System will display the profile
Alternative Flow of Events	1. At step 2.2.5 error message is prompt and asks the user to input valid information.
Preconditions	Starts when successfully logged in.
Postconditions	The user is successfully registered to the system.

4.3 Record Pocket Money (UC3)

Description	This use case allows students to input their total pocket money.
Objective	To know students pocket money in order to calculate allocation.
Actors	Students
Basic Flow of Events	1. User have to press Record Pocket Money button 2. System will display empty form 3. User have to fill in the information 4. System will save the pocket money 5. System will display the confirmation
Alternative Flow of Events	-
Preconditions	Starts when successfully logged in.
Postconditions	Student’s pocket money is successfully recorded.

4.4 View Allocation (UC4)

Description	This use case allows students to view the allocation of each category calculated by the system.
Objective	To calculate the allocation according to categories and display them.
Actors	Students
Basic Flow of Events	1. User have to select the view allocation button 2. System will calculate the money 3. System will save the allocation 4. System will display the allocation
Alternative Flow of Events	-
Preconditions	Starts when the pocket money is recorded.
Postconditions	Student's money allocation is successfully displayed.

4.5 Record Expenses and Transactions (UC5)

Description	This use case allows students to input their daily spent expenses.
Objective	To record daily expenses and transactions to deduct it from the allocation.
Actors	Students
Basic Flow of Events	1. User have to press the record transaction and Expense button 2. User have to choose the type of record 3. User have to fill in the details 4. User have to choose the type of needs 5. User have to submit 6. System will deduct the amount of money 7. System will save the allocation
Alternative Flow of Events	1. At 2.5.7 reminder message if the expense exceeds allocation after saved.
Preconditions	Starts when the user has successfully logged in.
Postconditions	The balance amount is successfully saved.

4.6 View Summary (UC6)

Description	This use case allows students to view their summary of spending .
Objective	To show the total money that the user has spent over a month.
Actors	Students
Basic Flow of Events	1. User clicks on View Summary button 2. System will display the monthly summary
Alternative Flow of Events	-
Preconditions	Starts when the user has successfully logged in and one month of expenses have been recorded.
Postconditions	The summary is successfully displayed.

5. ERD Diagram

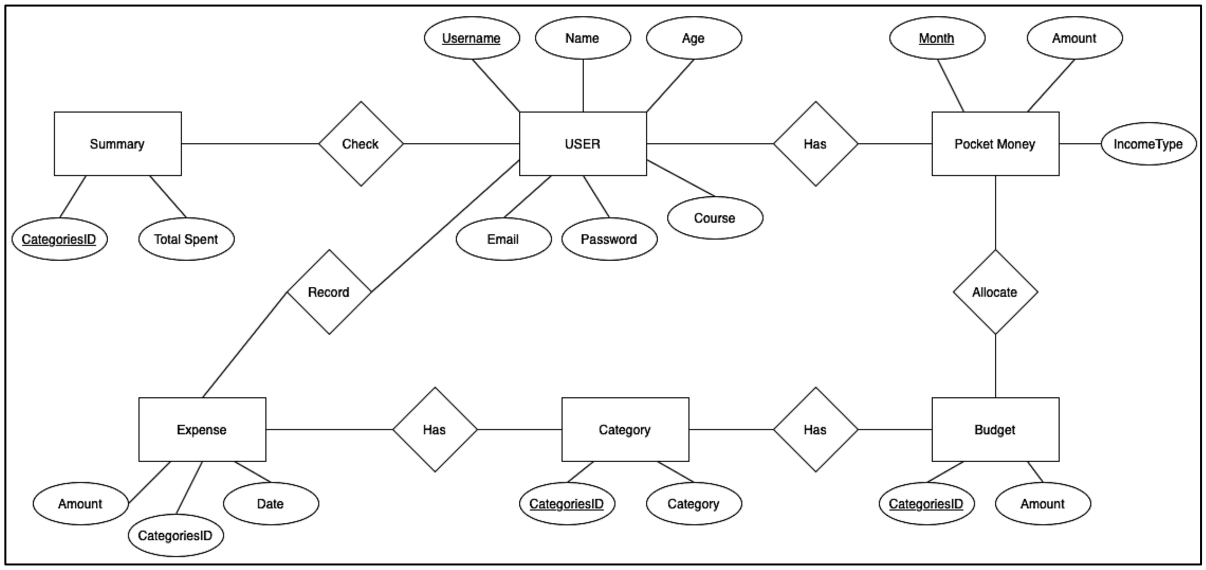


Figure 12

Figure 12 shows the ERD diagram which consists of six entities: User, Expense, Pocket Money, Category, Budget, and Summary with their own attributes. For the User entity, it has a primary key which is username and other five attributes which is name, age, email, password and course. Also for the Pocket Money entity, it contains one primary key which is month and other two attributes which is amount and incomeType. For the Expense entity, it has three attributes which are amount, categoriesID, and date. This entity does not have a primary key. For a Category entity, it has two attributes which is categoriesID and category where categoriesID is a primary key for this entity. Another entity is Budget. For this entity, it contains two attributes with one primary key which is amount and categoriesID respectively. The last one is Summary entity that contains also two attributes with one primary key which is total spent and categoriesID respectively. The user that has pocket money can allocate their budget by having or choosing which category that they will expense their money. Users also can record their expenses and check for the summary of their pocket money.

6. Other Non-functional Requirements

6.1 Performance Requirements

The system will display the Main page of the SBM application after 2 seconds when the app is opened. It requires 2 seconds when the user presses “Start” button in order to display the login page. To display the login page, it requires 10 seconds. When the user key-in their username and password. When the login is successful, the app will go to the Menu page after 2 seconds. In the Menu page, when the user clicks one of the options, the system will display the interface of the options that the user chose in 2 seconds.

6.2 Safety Requirements

The user’s data is stored in an efficiently designed database which is only accessible by developers which eliminates any data loss.

6.3 Security Requirements

The system provides an authentication facility which allows only authorized users to access the system using their login credentials. Accordingly, the login feature provided mitigates the probability of the system to be attacked.

6.4 Software Quality Attributes

6.4.1 Availability

The system is available for UPM students 24/7. In cases of the system being out of service due to any error or crash, the system will be recovered at most within half a day.

6.4.2 Maintainability

The system will be developed with high modularity which promotes low coupling between modules. As a result, this design decision enhances maintainability of the system in the future whereby modules are maintained individually when necessary.

6.4.3 Correctness

Calculation algorithms are correctly applied in the system as well as producing correct results that are efficiently interpreted by users. To illustrate that, the alert must be appeared if the user uses the money which exceeds the budget.

6.4.4 Usability

The system provides a user-friendly and simple user interface for users to effectively interact with. It is expected that users do not need any training or tutorials to use the system; moreover, the system is easily learnt from the first time used and the flow of interaction and tasks can be remembered in further visits.

6.4.5 Efficiency

Any given task in the system shall be completed by a short and fast procedure that requires few button clicks and second to accomplish the task. For example, three button clicks per task and a total of 5 seconds to get the task done efficiently.