Software Requirements Specification

for

Student Budget Management System

Version 1.0 approved

Sufi Nukman Bin Shakimi
Muhammad Alif Luqman Bin Afandi
Nurhamizah Bt. Husin
Eman Hussein

15 March 2020

Table of Contents

1. Introduction....1
1.1 Purpose....1
1.2 Scope....1
1.3 Definitions, Acronyms, and Abbreviations....1
1.4 References....1
2. Overall Description....2
2.1 User Classes and Characteristics....2
2.2 Operating Environment....2
2.3 Design and Implementation Constraints....2
2.4 Assumptions and Dependencies....2
3. External Interface Requirements....3
3.1 User Interfaces....3
4. System Use Cases....11
4.1 Login (UC1)....12
4.2 Register (UC2)....13
4.3 Record Pocket Money (UC3)....14
4.4 View Allocation (UC4)....15
4.5 Record Expenses and Transactions (UC5)....16
4.6 View Summary (UC6)....17
5. ERD Diagram....18
6. Other Non-functional Requirements....19
6.1 Performance Requirements....19
6.2 Safety Requirements....19
6.3 Security Requirements....19
6.4 Software Quality Attributes....19

1. Introduction

1.1 Purpose

Student Budget Management (SBM) is focusing on managing budget and finance specifically for students. This project is related to finance field among UPM students. Nowadays, students are facing problem in managing their money in 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 improve the disabilities in current existing system.

1.2 Scope

The SBM focuses in managing UPM students finance 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 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 to 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.

2.2 Operating Environment

This application can be used in UPM environment. SBM is built in web application.

2.3 Design and Implementation Constraints

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

2.4 Assumptions and Dependencies

The assumptions of this application is when the user not connect the apps with the Internet.

3. External Interface Requirements

3.1 User Interfaces

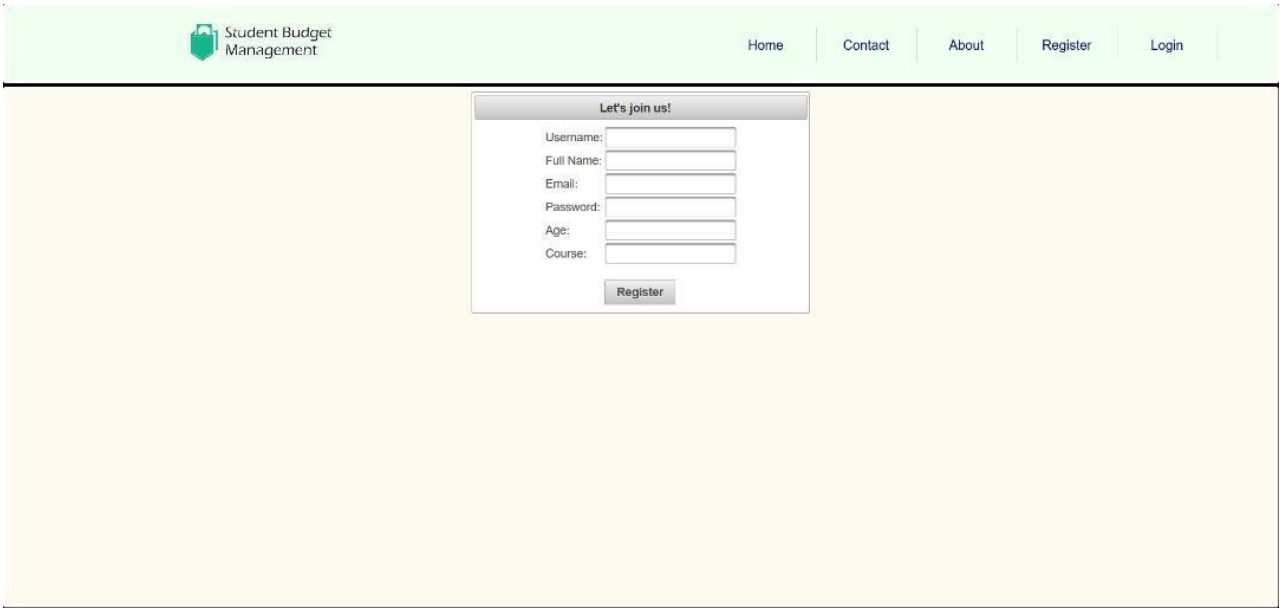

Figure 1

The above diagram is the interface of ‘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 to store the user’s information.

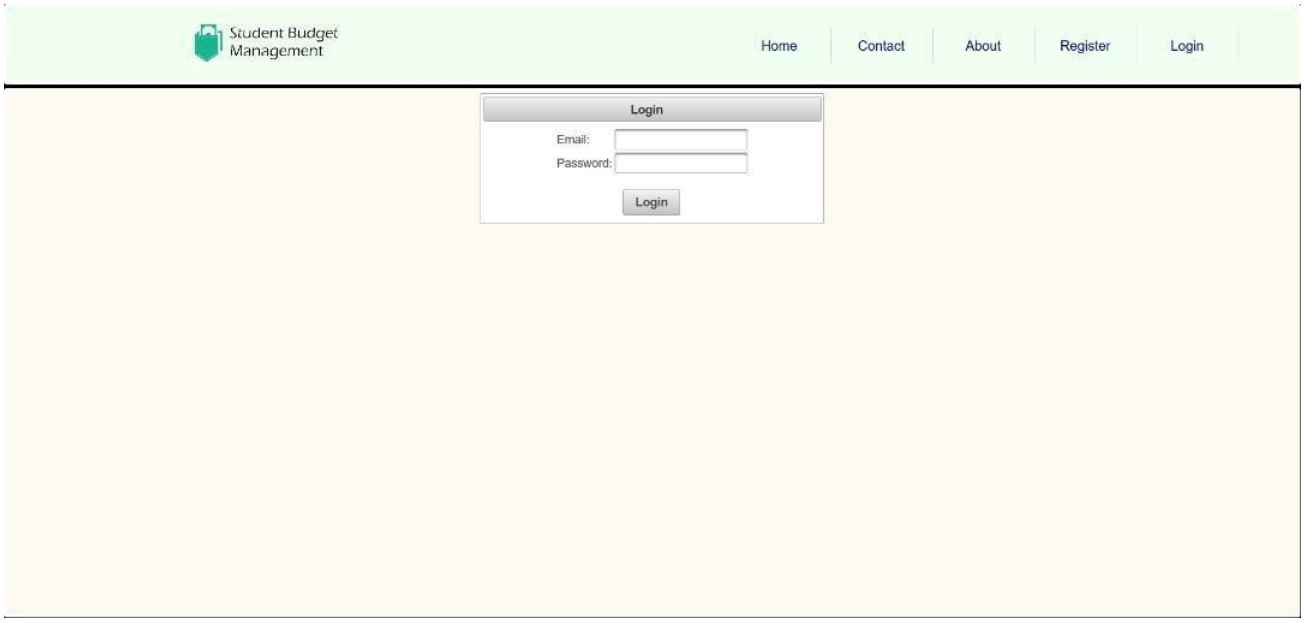

Figure 2

The above diagram is the interface of ‘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 to verify and allow user to sign in.

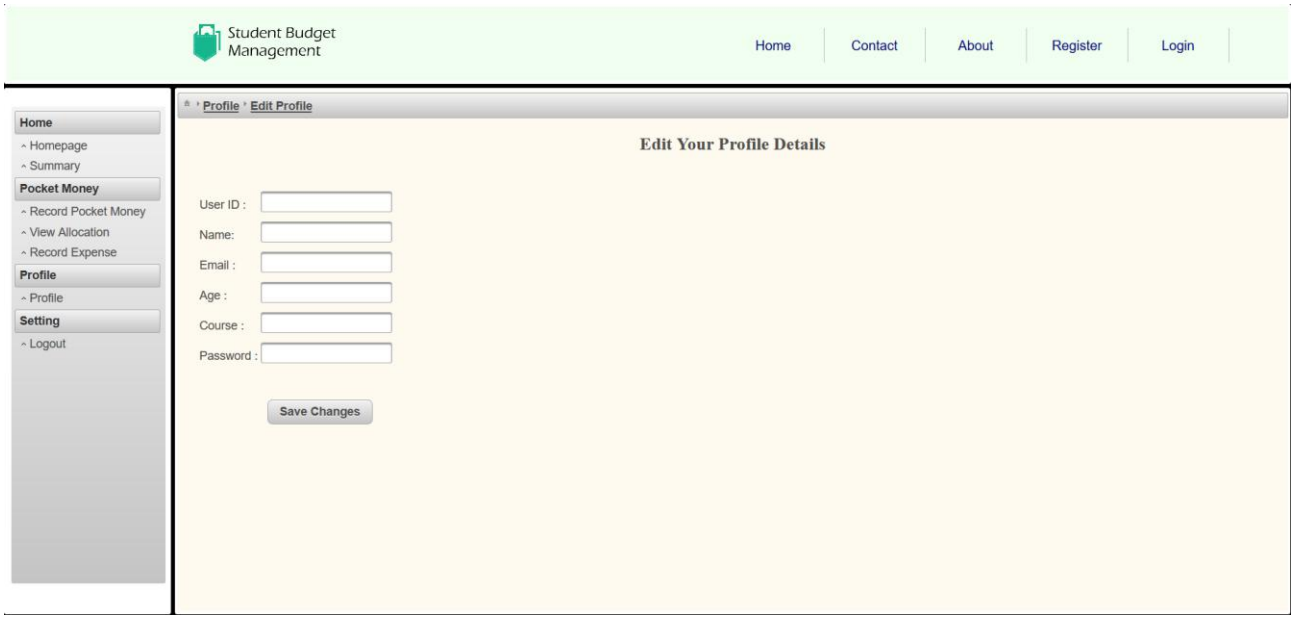

Figure 3

The above diagram is the interface for editing the user's profile. It shows the editable user's profile information. User 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.

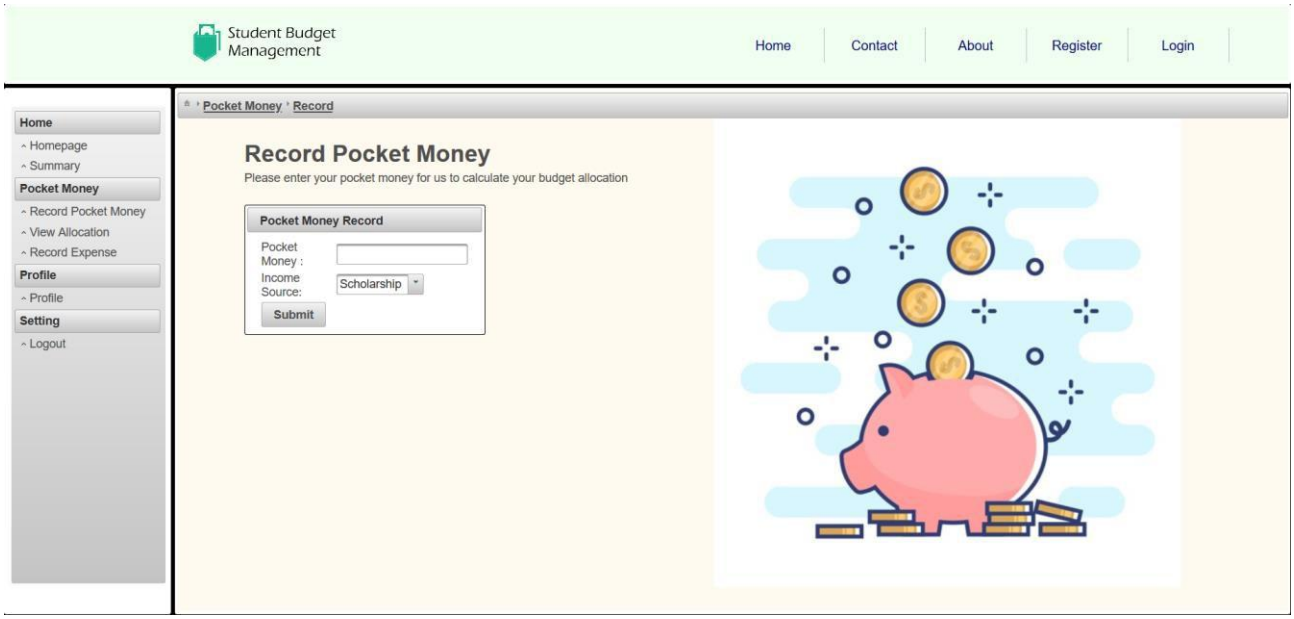

Figure 4

The above diagram is 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.

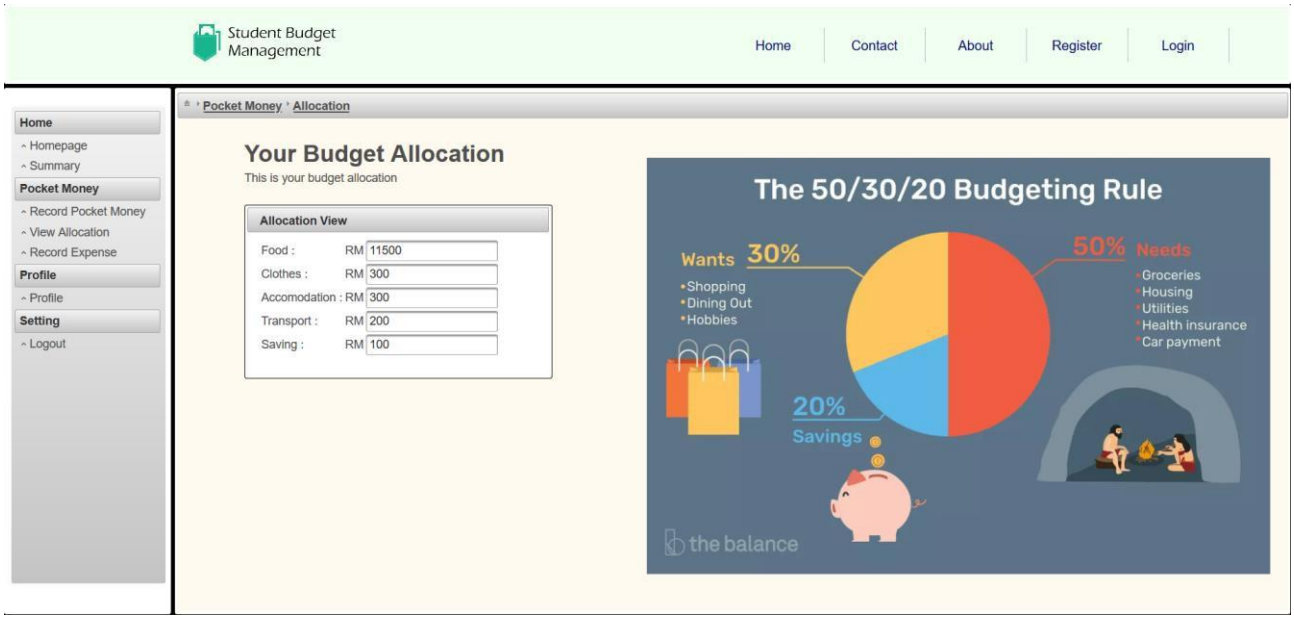

Figure 5

The above diagram is the interface of ‘View Allocation’ use case. It shows the allocations of the pocket money into 5 categories: food, clothes, accommodation, transport and saving.

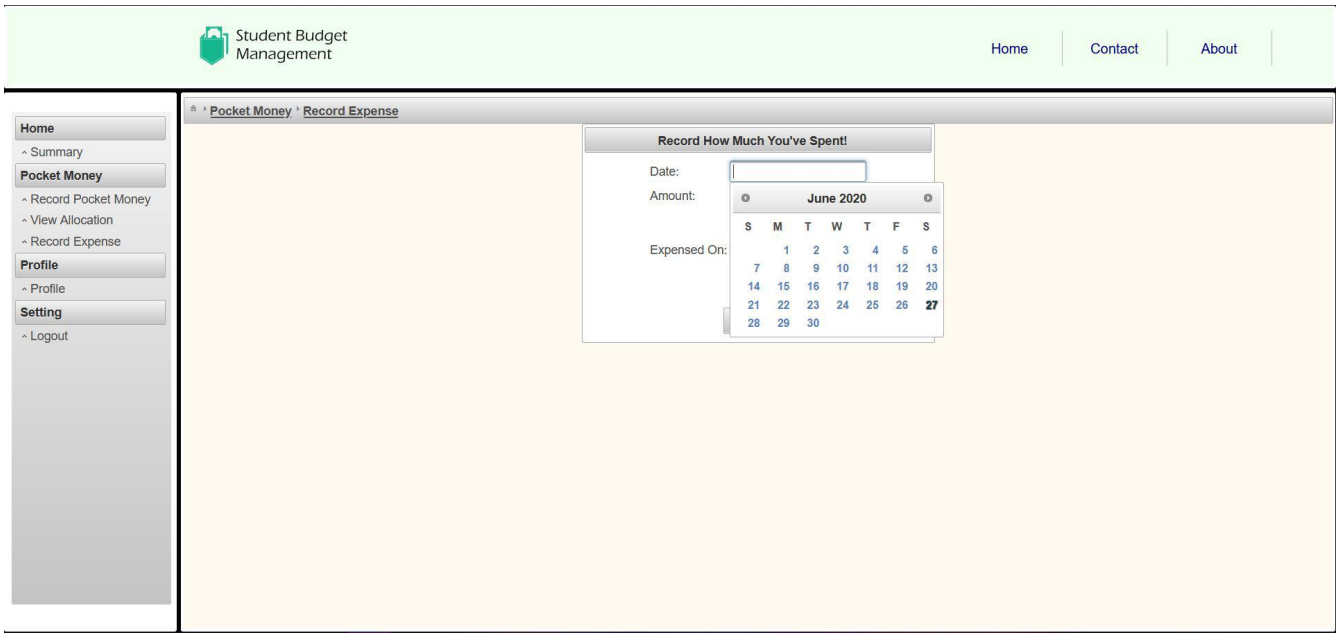

Figure 6

The above diagram is the interface of ‘Record Expense’ use case. It shows a date picker for the user to enter the date and a text field for the amount he/she has spent. Besides, there are radio buttons for the categories that he/she already spent.

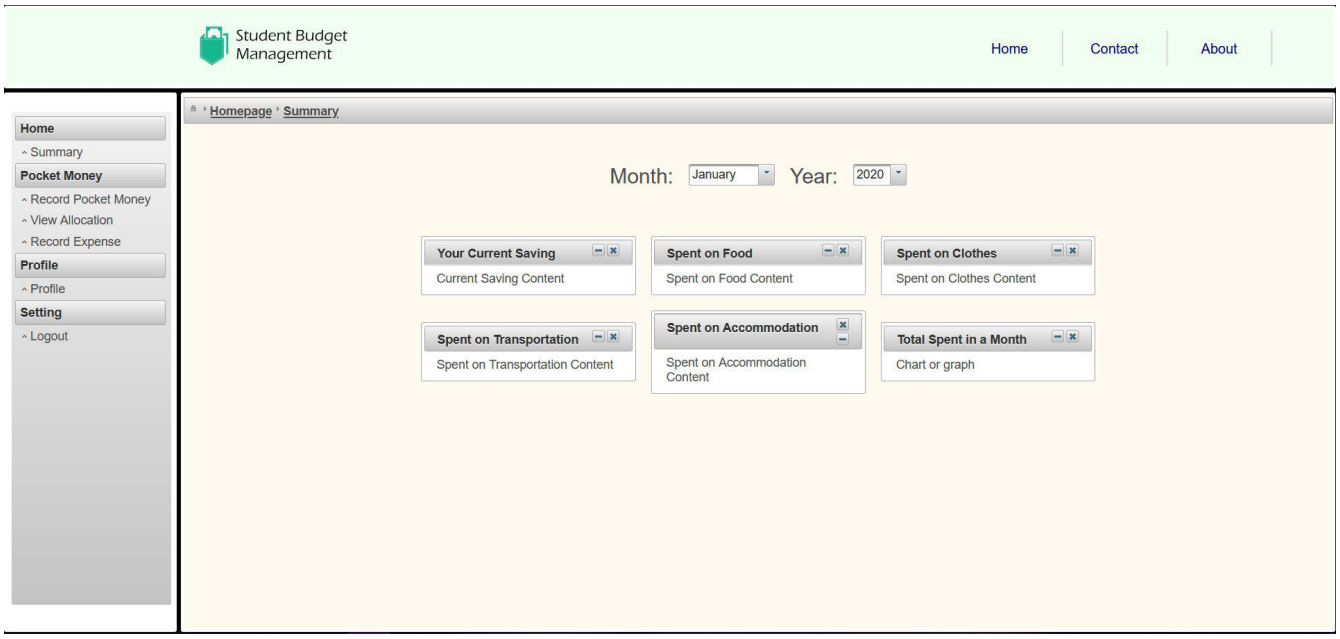

Figure 7

The above diagram is the interface of 'View Summary ' use case. It shows a month and year picker for user to choose and view 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.

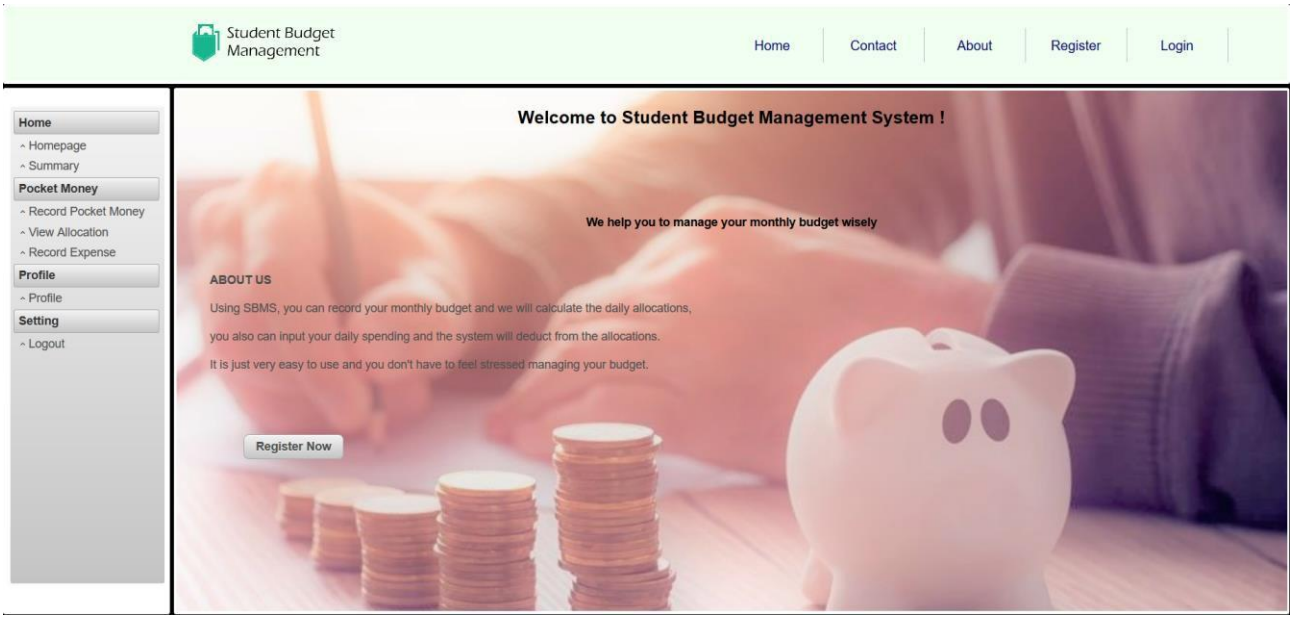

Figure 8

Figure 8 shows the interface of the homepage of SBM. It gives a little bit description of SBM to the new user.

4. System Use Cases

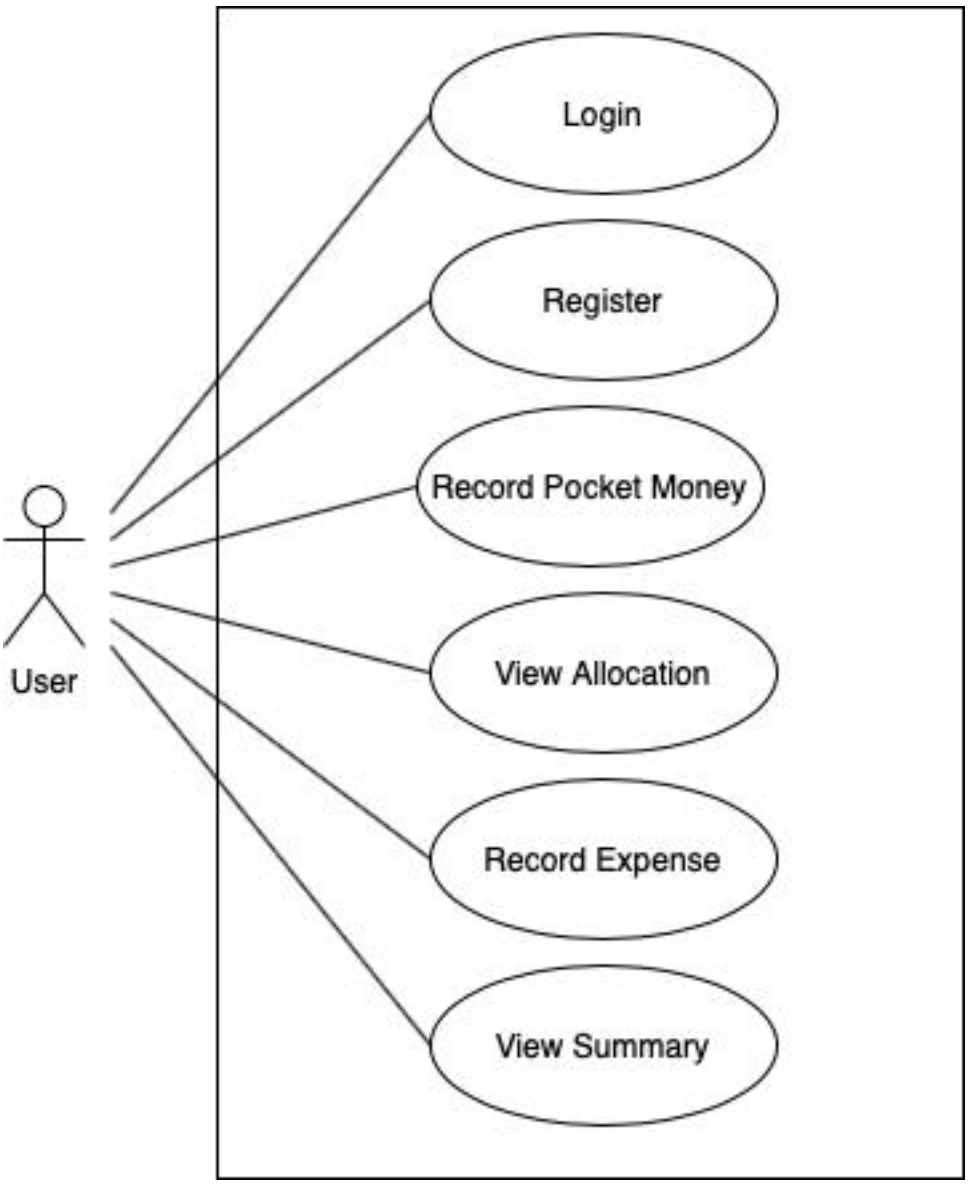

Figure 9

Figure 9 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 user.
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 renter 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 ask 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 transaction 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 have 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 student have 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

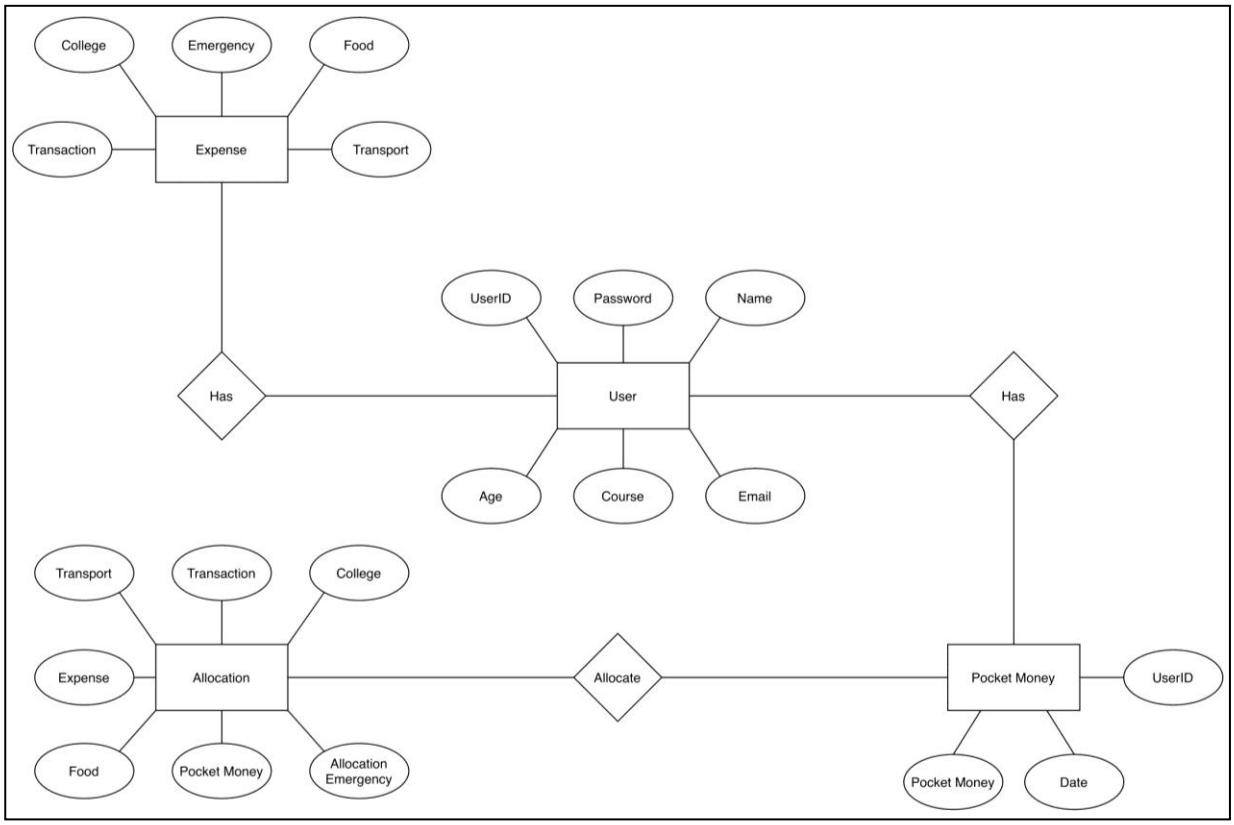

Figure 10

Figure 10 shows the ERD diagram which consists of four entities: User, Expenses, Pocket Money and Allocation with their own attributes

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 press button “Start” in order to display 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 click one of the options, the system will display the interface of the options that the user choose in 2 seconds.

6.2 Safety Requirements

The username and password are required to ensure the safety of the data from loss.

6.3 Security Requirements

Student may request a temporary password if they forgot their password or username and shall receive a link sent to their primary email address.

6.4 Software Quality Attributes

6.4.1 Availability

The system is available 100% for UPM students. It can be used 24 hours.

6.4.2 Maintainability

Every budget change can be made by the user

6.4.3 Correctness

The alert must be appeared if the user use the money exceeds the budget.