PERSONALIZED BUDGET PLANNING SYSTEM

BY

ANG ZI WEI

A REPORT

SUBMITTED TO

Universiti Tunku Abdul Rahman

in partial fulfillment of the requirements

for the degree of

BACHELOR OF INFORMATION SYSTEMS (HONS) INFORMATION SYSTEMS ENGINEERING

Faculty of Information and Communication Technology

(Perak Campus)

MAY 2018

UNIVERSITI TUNKU ABDUL RAHMAN

REPORT STATUS DECLARATION FORM

Title: Personalized Budget Planning System

Academic Session: Year 3 Trimester 3

I ANG ZI WEI

(CAPITAL LETTER)

declare that I allow this Final Year Project Report to be kept in

Universiti Tunku Abdul Rahman Library subject to the regulations as follows:

1. The dissertation is a property of the Library.
2. The Library is allowed to make copies of this dissertation for academic purposes.

Verified by,

_________________________ _________________________

(Author's signature) (Supervisor's signature)

Address:

No 02-23d blok c jalan wira 1

Taman Tan Sri yaacob

81300 skudai johor

Chan Lee Kwun

Supervisor's name

Date: ____________________ **Date**: ____________________

PERSONALIZED BUDGET PLANNING SYSTEM

BY

ANG ZI WEI

A REPORT

SUBMITTED TO

Universiti Tunku Abdul Rahman

in partial fulfillment of the requirements

for the degree of

BACHELOR OF INFORMATION SYSTEMS (HONS) INFORMATION SYSTEMS ENGINEERING

Faculty of Information and Communication Technology

(Perak Campus)

MAY 2018

DECLARATION OF ORIGINALITY

I declare that this report entitled "**PERSONALIZED BUDGET PLANNING SYSTEM**" is my own work except as cited in the references. The report has not been accepted for any degree and is not being submitted concurrently in candidature for any degree or other award.

Signature : _________________________

Name : _______ANG ZI WEI_______

Date : _________________________

ACKNOWLEDGEMENT

I would like to express my sincere thanks and appreciation to my supervisor, Ms. Chan Lee Kwun who provided various valuable advices to the project. I also appreciate the effort and time taken for my moderator Dr. Mobashar Rehman to evaluate and give valuable advices to my project during the first phase of the project.

I would also like to thanks to my friends for giving various advices during the development of the project. Finally, I would like to thanks my parents for supporting me during the course.

ABSTRACT

This project is a web application project for fulfilling academic purposes. The project proposed involves creating a web application for managing budget. A budget management system is a system that allows user to manage budget by supplying data such as account balance, budget, and transactions. There are different types of transaction including expense, income, and transfer. A user is required to register for an account and login before they are allowed to use the budget planning system. Most budget planning system provides a dashboard that displays information about the user's budget information. The dashboard displays components using a predefined component flow such as specific graphs and recent transactions. Some users might prefer to use one chart than another. This project aims to provide a solution by allowing users to personalize the dashboard of the application. By allowing users to select the chart to use for the component on the dashboard, the improves the experience of the user. This project also provides keyboard shortcuts to access different parts of the application. The project aims to provide base functionalities that were provided by various existing systems and adding the personalize module to the system. The user can also select theme to change the appearance of the application. Tools to be used to develop the application including Asp.NET Core 2.0 with C#, HTML, CSS JavaScript, etc.

TABLE OF CONTENTS

LIST OF TABLES

Table Number **Title** **Page**

LIST OF FIGURES

Figure Number **Title** **Page**

LIST OF SYMBOLS

LIST OF ABBREVIATIONS

IDE	Integrated Development Environment
HTML	Hypertext Markup Language
CSS	Cascading Style Sheets

Chapter 1 Introduction

1.1 Background and Motivation

This project aims to provide a solution for budget planning. The goal of this project is to deliver a web application that provide several convenient functionalities that do not exist in other reviewed budget planning systems. These functionalities including using keyboard shortcuts to access different parts of the application, select a theme to change the appearance of the user interface, exporting report directly to third party cloud storages such as Dropbox and Google Drive, and enabling 2 factor authentications. The project allows users to create transaction records such as expenses, incomes and transfers and specify a budget so the users can plan for amount spent based on preferred activities. The applications that were selected for reviews were Mobills, GoodBudget, and BudgetBakers.

1.2 Problem Statements

1.2.1 Not able to export report directly to third party cloud storages

Problem	If the users want to save their report (e.g. balance overtime report) to third party cloud storage such as Google Drive or Dropbox. The user will need to manually save the report to their computer and manually upload the report to the preferred cloud storage. Although the user can directly save the report to Google Drive using the print functionality of Google Chrome, not every user uses Google Chrome and some users might be using other cloud storages such as Dropbox.
Potential Solution	Allowing users to save their report directly on third party cloud storages directly. The users can also preview the report or download the report directly onto their machine.

Table 1-1 Not able to export report directly to third party cloud storages

1.2.2 Lack of keyboard shortcuts

Problem	The budget planning systems mentioned do not provide keyboard shortcuts for the users. Keyboard shortcuts may not be necessary for a system but it can increase the speed for users to perform operations using the system once the users become familiar with the shortcuts provided.

Potential Solution	Provide keyboard shortcuts to access different parts of the application, create a new transaction, as well as go to the delete page and edit page of a record.

Table 1-2 Lack of keyboard shortcuts

1.2.3 No 2-factor authentication

Problem	2-factor authentication was not provided in the reviewed systems. Some user may prefer systems that provide 2-factor authentication to decrease the chance for other users to impersonate as the user and login as the user's account.
Potential Solution	Provide 2-factor authentication in the proposed system. The user can optionally enable it in their profile page. The user will need to do some setup before they can enable 2-factor authentication.

Table 1-3 No 2-factor authentication

1.2.4 No theme personalization

Problem	The users are unable to change the appearance of the applications.
Potential Solution	The users will be able to change the appearance of the application by selecting their preferred theme.

Table 1-4 No theme personalization

1.3 Project Scope

This project develops a web application that can be used for managing budget, the user can personalize the components to display on the dashboard. These are the functionalities covered.

1.3.1 Identity Management Module

Users can register an account by entering their email and password or sign up an account using Google. In the future, the user can either login using email and password combination, or sign in using their Google account. The users can also update their information such as email and password. 2-factor authentication can be enabled; If 2-factor authentication is enabled, when the user attempts to sign in with email and password combination, the user will also need to enter the code associated with the application shown on Google Authenticator.

1.3.2 Transaction Module

The user can create accounts that can be represented as real world accounts. These accounts can then be used to create expenses, incomes, and transfers to modify balance from the account. Expenses will deduct balance from the account that was used to create the expense while income will increase the balance of the account that was used to create the income. Transfer will transfer some amount of balance from provider account to receiver account. (E.g. create a transaction for paying electricity bill will deduct the total balance from account as well as the budget planned for electricity).

1.3.3 Budget Module

Users are allowed to specify the budget for different categories for the month. (E.g. a budget for foods while another budget for education). The budget information will be represented with a page containing pie charts. The pie charts will be composed of 2 parts. The first part will be the expenses made on the category, and the second part will be remaining amount that can be spent for the category.

1.3.4 Graph and Report Module

The dashboard of the application will contain various components to display the budget information of the user. Some components simply display numbers while some components use charts to visualize specific information. Users can generate reports based on the current budget information such as expected expenses, actual expenses, and current account balance. (E.g. create a graph to display the amount spent on different categories). The user can also export the report to Google Drive or Dropbox, preview the report, or download the report directly.

1.3.5 Personalize Theme Module

User can change the theme of the system to change the look and feel of the system. This can be useful if the user as different users have different preference for the appearance of user interfaces.

1.4 Project Objectives

This project aims to allow users to manage their budget by specifying budgets and creating transactions, while allowing users to select the theme to change the appearance of the application. The application also provides a convenient functionality which is to allow users to export report directly to third party cloud storage such as Google Drive or Dropbox. This system also intends to introduce a way to reduce the potential risk of account being accessed by malicious users if the password was unfortunately leaked by enabling 2 factor authentications. This project does not create a mobile application such as an Android application.

1.5 Impact, significance and contribution

This project will benefit the users who would like to use a web application to manage budget while having the flexibility to change the appearance of the application by selecting the theme they prefer; keyboard shortcuts will be provided so the user can perform some operations in a more convenient way. 2-factor authentication can be enabled to reduce the chance of the account being login by malicious user (in case the malicious user was able to guess the user's password by any mean). Reports generated by the system can be exported directly to third party cloud storages including Google Drive and Dropbox which can be convenient for the users.

1.6 Background Information

1.6.1 Application

Applications were developed to solve problems (Global, n.d.) such as managing employee information, communicate using the Internet, automate repetitive tasks, etc. Applications can be developed in multiple types including web applications, mobile applications, desktop applications, etc. Each type has its advantages and disadvantages. Applications are also known as system, program, software, etc.

1.6.2 Web application

Web applications are applications that can be accessed using browsers such as Google Chrome, Firefox, Microsoft Edge, and etc. Unlike desktop applications that reside in the operating systems of users, web applications are served over web servers (TechTerms, 2014). To access the application, the users are required to make requests to the server, then the server will create appropriate respond based on the request provided (Ndegwa, 2016). For example, the user can enter www.google.com to request the Google search page, then the web servers from Google will return the response back to the client, in such case, the response will be a HTML page containing a search box. There were numerous advantages of web applications such as eliminating the need to develop multiple applications to serve different operating systems as a web application that works on a web browser such as Google Chrome will work on multiple operating systems (e.g. Windows, MacOS, Linux). There were different types of web applications that were developed to achieve different objectives, for instance, users can use YouTube to watch videos, Gmail for managing emails, Slack for communication among team members, etc.

1.6.3 Authentication

Authentication is a process to determine if a user is actually the user he/she claiming to be (Rouse, 2015). For example, a user attempts to log in to Google using example@gmail.com. To ensure that the user actually owns this account, Google requires user to enter the correct password. The user is only allowed to access the account internal information such as incoming emails if the email and password combination was correct.

1.6.4 Authorization

Authorization is a process to determine if a user is allowed to perform some operations on some resources (Anderson, 2016). For example, in Gmail, a user is only allowed to view and delete emails that belong to the user instead of emails of other users.

1.6.5 Budget Planning System

A budget planning system is a system that allows user to manage their budgets by creating fictional financial balance such as wallet, primary bank account, etc, specifying budgets based on category, and creating transactions, a transaction can be categorised into expense or income. An expense will deduct balance from the account that is used for the expense while the income will increase the balance of the selected account to receive the income. The budget planning system will display budget information of the users on the dashboard, the dashboard may contain information such as recent transactions, graphs of transactions based on category, etc. The user can also display specific reports and graphs by providing different options and filters. Most budget planning system, like typical dynamic web applications provide features for user identity management such as register and login. The users can also update their credentials such as name, email, password after logging into the application. Each user will have their own budget information. For example, a transaction created by the user “Bob” will not be displayed on the dashboard of the user “Alice”.

Chapter 2 Literature Review

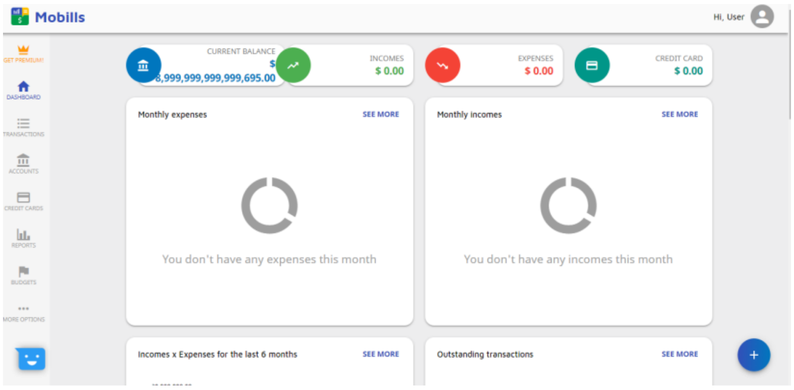
2.1 Literature Review

2.1.1 The creation of budget planning system

People who would like to keep track of their financial expense and income may attempt to track down their transaction histories using a software known as a budget planning system. These applications were developed for solving the problem and each application attempts to solve problem with some differences. The systems that are selected for reviews including Mobills, GoodBudget, and BudgetBakers.

2.1.2 Mobills

Mobills, which could be accessed via https://web.mobillsapp.com was a web application that helps user to keep track of their expenses and incomes. To use the application, the user needs to sign up an account either using email and password combination, or using third party providers including Google and Facebook. The user will be greeted with a dashboard that contains two pie charts, a calendar that displays the transactions made, a pending transactions components, a favourite transactions component, and a Mobills pro component. The first pie chart consists of expenses sliced into different categories while the second pie chart consists of incomes sliced into different categories. The user can create multiple financial accounts such as wallet, primary bank account, etc. The users can create expense or income record using the application. The users can create budgets with a particular category (e.g. shopping, food, cloth, etc) for the month. The user can then keep track of the remaining budget for the particular category since an expense record consist of that category will deduct the remaining budget of the category. A mobile application was also available for Android and IOS.



CURRENT BALANCE $ 8,999,999,999,999,695.00

INCOMES $ 0.00

EXPENSES $ 0.00

CREDIT CARD $ 0.00

Monthly expenses

You don't have any expenses this month

Monthly incomes

You don't have any incomes this month

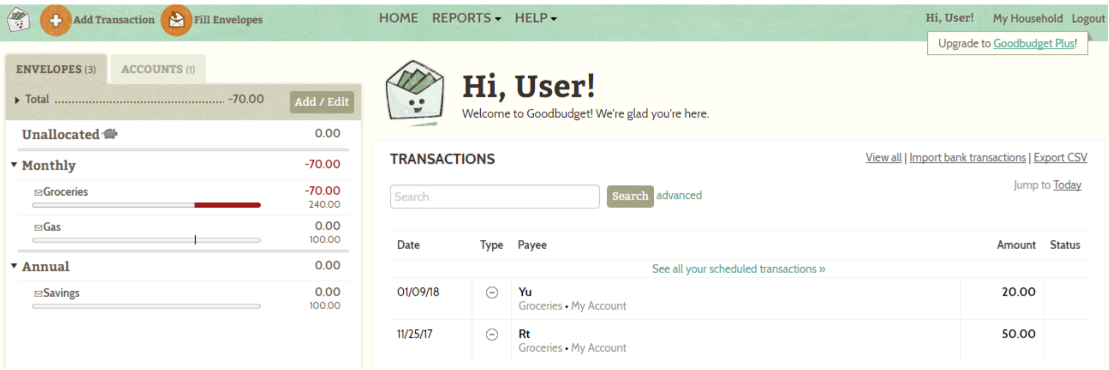
Incomes x Expenses for the last 6 months

Outstanding transactions

Figure 2-1 Mobills Dashboard (Mobills, n.d.)

2.1.3 Goodbudget

Goodbudget, is another budget management system that can be accessed via https://goodbudget.com. The user could only register an account using email and password combination, the options for registering an account via third party providers such as Google and Facebook are not available. The dashboard of the application contains a sidebar and a main component. The sidebar contains two tabs, the first tab displays the envelop information of the user while the second tab display the financial account information of the user. The main component contains the transactions that have been made by the user. The users can create multiple financial accounts, make transactions that can be categorised as expense, transfer, and income, create envelopes for a particular category, and viewing reports. The application also provides tutorials to help the user to understand the system. A forum of the application is also provided so the users can seek help in the forum by asking questions. The application manages user's budget by requiring user to create envelopes and fill the envelopes, these envelopes are treated as reserved money that can be used for a particular category. The user can then create expenses that will deduct the remaining balance from the selected envelop. Both Android application and IOS application were available as well.



Hi, User!

Welcome to Goodbudget! We're glad you're here.

Total	-70.00
Unallocated	0.00
Monthly	-70.00
Groceries	-70.00 240.00
Gas	0.00 100.00
Annual	0.00
Savings	0.00 100.00

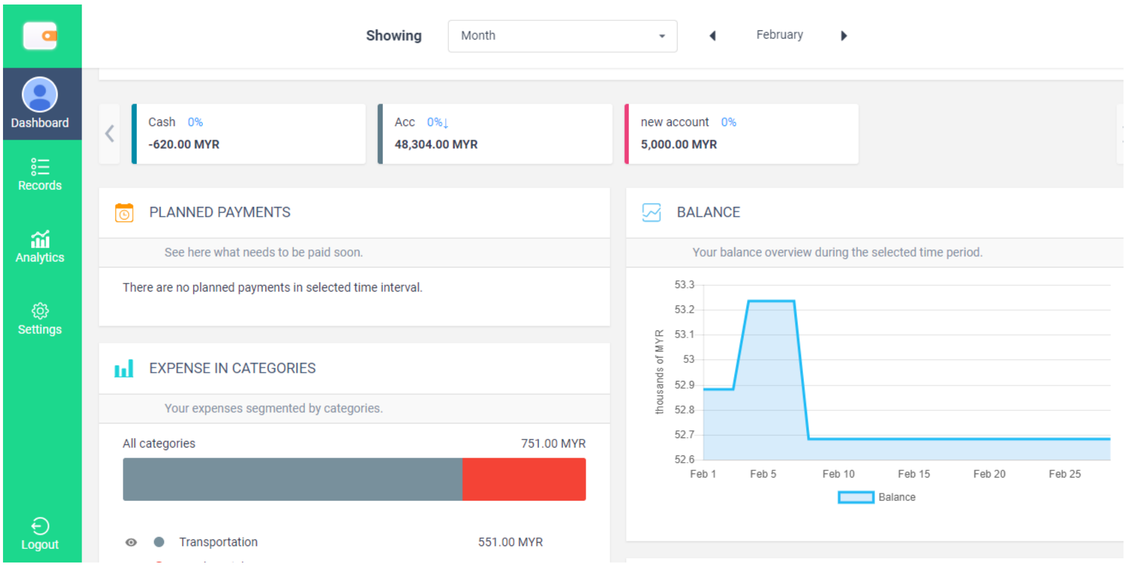
TRANSACTIONS

Date	Type	Payee	Amount	Status
01/09/18	⊖	Yu Groceries • My Account	20.00	
11/25/17	⊖	Rt Groceries • My Account	50.00	

Figure 2-2 GoodBudget Dashboard (GoodBudget, n.d.)

2.1.4 Budgetbakers

Budgetbakers, can be accessed via https://web.budgetbakers.com is another budget management system. Although there was a web version of the application, the users would need to sign up their account via the mobile applications such as the Android application or the IOS application. The user could sign up either using email and password combination or using third party providers including Google and Facebook. The application contains five tabs, which were dashboard, records, analytics, settings, and logout. The dashboard of the application contains various components including balance, cash flow, expenses, and incomes for the month, list of accounts of the user, planned payment component, expenses in category, a graph that displays the balance over time, a graph about cumulative expense, and the recent records made. The user can click on the record tab to add record such as income, expense, and transfer. They can also filter the records to view by the record type and category. The user can view their budget information in either line chart or bar chart format by clicking on the analytic tab, several options are provided for the user to produce the graph by supplying the period, record type, category and etc. The users can go to the settings tab to change the website language, update their password, rename the categories, adding new currencies, adding new accounts, etc.



Showing: Month ◂ February ▸

Cash 0% -620.00 MYR

Acc 0%↓ 48,304.00 MYR

new account 0% 5,000.00 MYR

PLANNED PAYMENTS

See here what needs to be paid soon.

There are no planned payments in selected time interval.

EXPENSE IN CATEGORIES

Your expenses segmented by categories.

All categories 751.00 MYR

Transportation 551.00 MYR

BALANCE

Your balance overview during the selected time period.

Figure 2-3 Budgetbakers Dashboard (budgetbakers, n.d.)

2.2 Critical Remarks of previous works

2.2.1 Systems selected for comparison

System/Application Name	URL
Mobills	https://web.mobillsapp.com
Goodbudget	https://goodbudget.com/home
BudgetBakers	https://web.budgetbakers.com

Table 2-1 Systems selected for comparisons

2.2.2 Comparison of selected applications with proposed solution

	Mobills	Goodbudget	BudgetBakers	Proposed system (Personalized Budget Planning System)
Users identity management	Allows user to sign up an account using email and password or external provider such as Google	Allows user to sign up an account using email and password.	Allows user to sign up an account using email and password. Unable to sign up using the system on the browser, must sign up an account using native app that can be installed on mobile devices.	Allows user to sign up an account using email and password or external provider such as Google
Tutorials to use the system	Does not provide a tutorial on how	Provides tutorials on how to use the	Does not provide tutorials on	Does not provide tutorials on

	to use the system	system by providing various articles. (Goodbudget, n.d.)	how to use the system	how to use the system
Ways to plan budget	Users enter goals for different categories, monthly income and expenses. The system will display the balance based on the information provided. (Mobills, n.d.)	Users create envelopes that represent the budgets for different categories. Then the user creates transactions to deduct the amount in the budget. (GoodBudget, n.d.)	User specifies income and expenses by adding records categorised as income, expenses, and transfer. Then the system will provide various graphical representation based on the information provided. (budgetbakers, n.d.)	Users enter goals for different categories, monthly income and expenses. The system will display the balance based on the information provided.
Reports	Unable to export report directly to third party cloud storages	Unable to export report directly to third party cloud storages	Unable to export report directly to third party cloud storages	Able to export report directly to third party cloud storages report directly to cloud storages such as Google

				Drive and Dropbox.
Responsive user interface on different screen size	Responsive. Provides native applications that fits well on the screen which can be installed on different operating system (E.g. Android)	Partially responsive. But provides native applications that fits well on the screen which can be installed on different operating system (E.g. Android)	Not responsive. But provides native applications that fits well on the screen which can be installed on different operating system (E.g. Android)	Responsive. Does not provide native applications.
Multi-currency support	Provides multi-currency support	Does not provide multi-currency support	Provides multi-currency support	Provides multi-currency support
Keyboard shortcuts	Keyboard shortcuts not provided.	Keyboard shortcuts not provided.	Keyboard shortcuts not provided.	Keyboard shortcuts provided.
2-factor authentication	No 2-factor authentication	No 2-factor authentication	No 2-factor authentication	Has 2-factor authentication
Theme selection	Unable to change theme	Unable to change theme	Unable to change theme	Able to change theme

Table 2-2 Comparison between selected systems

Chapter 3 System Design

3.1 Use case diagram

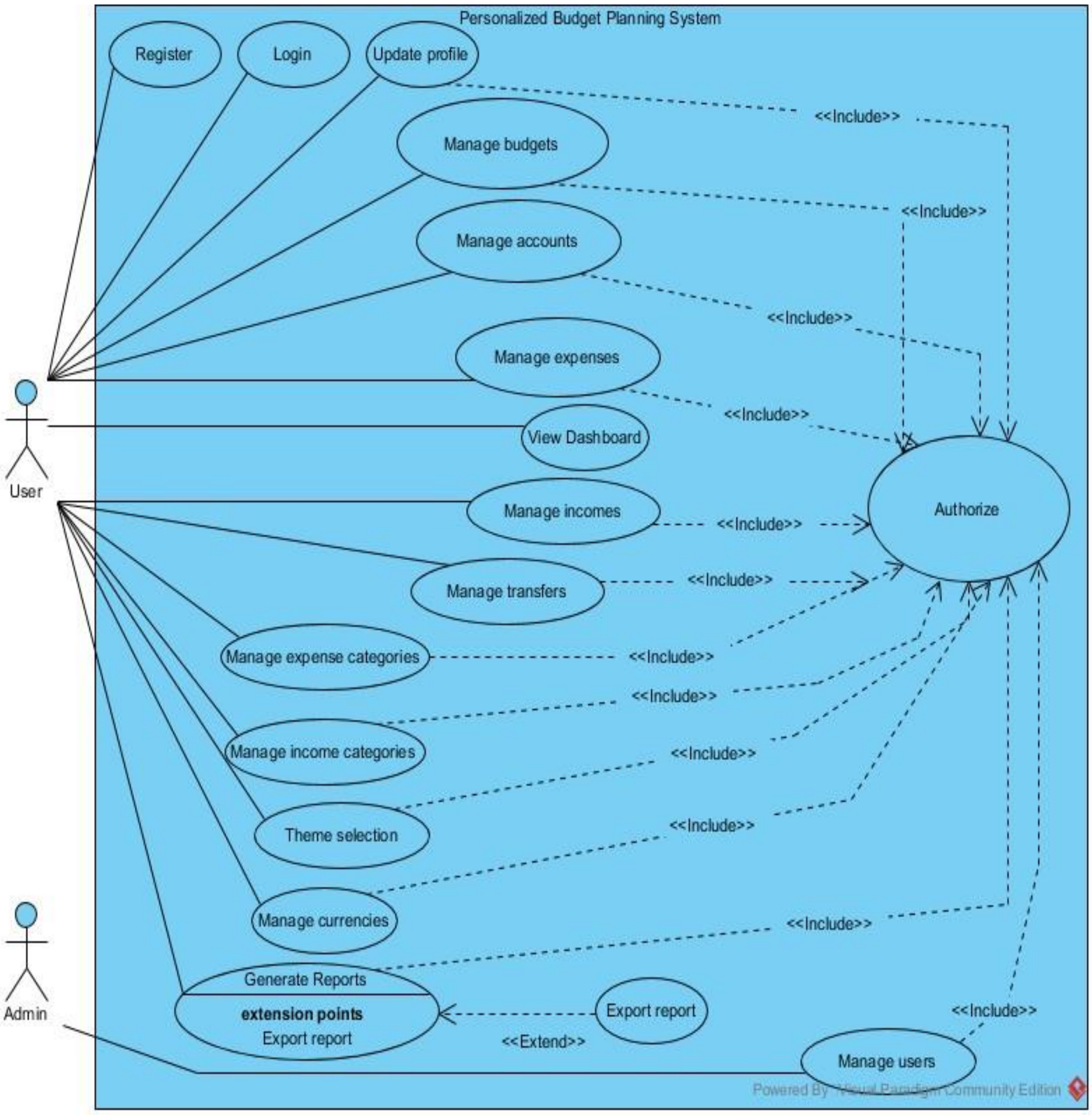


Figure 3-1 Use Case Diagram

3.1.1 Register

Users will be able to register an account using their email and password combination or using their Google+ identity. If there were validation errors such as password and confirm password do not match, the register page will display appropriate error messages to the user.

3.1.2 Login

Users will be able to login to the system given that they provided the correct email and password combination or using their Google+ identity. If 2-factor authentication was enabled and the user was using email and password combination to login to the system, the user will need to enter the code that can be accessed via Google Authenticator that was installed on their device.

3.1.3 Update profile

Users will be able to update their profile information after they have login to the system.

3.1.4 Manage expenses

Users can perform create, delete, update, and read operations on expense records. Expense records will deduct the amount of balance of the account selected by the user. The fields of an expense record including amount, date, currency, and account.

3.1.5 Manage incomes

Users can perform create, delete, update, and read operations on income records. Income records will increase the amount of balance of the account selected by the user. The fields of an income record including amount, date, currency, and account.

3.1.6 Manage transfers

Users can perform create, delete, update, and read operations on transfer records. Transfer records transfer the specified balance of one account to another account.

The fields of a transfer record including amount, date, account, and receiving account.

3.1.7 Manage accounts

Users can perform create, delete, update, and read operations on their accounts. Accounts are used to create transactions. The fields of an account record including amount and currency.

3.1.8 Manage budgets

Users can specify how much they want to spend on different expense categories. The budget is represented using a pie chart, the pie chart is composed of two parts; The first part represents the amount deducted from the budget while the second part represents the remaining of the budget. The fields of a budget record including budget amount for a specific month and default budget amount for every month.

3.1.9 Manage expense categories

Users can perform create, delete, update, and read operations on expense categories. Expense category records are used when the users want to create expense records. Expense category only has one editable field which is name.

3.1.10 Manage income categories

Users can perform create, delete, update, and read operations on income categories. Income category records are used when the users want to create income records. Income category only has one editable field which is name.

3.1.11 Theme selection

User can select a different theme to change the appearance of the user interface. The themes can be selected via a dropdown list.

3.1.12 Manage currencies

Users can perform create, delete, update, and read operations on currency records. Currencies are used when the users want to create expense, income, or transfer records. The user can also select a main currency. The main currency is used when the user views the dashboard of the application or attempts to generate reports. The fields of a currency record including the conversion rate to main currency and whether the currency is a main currency to the user.

3.1.13 Authorization

Users are only allowed to delete, update, and display the records if they are authorised. For example, user A will not be able to delete, update, and display the expense records of user B. However, an admin will be able to delete, update, and display the records of other users. Additionally, an admin can display users of the system, delete users, and update role of other users.

3.1.14 Manage users

The capability to display, delete, and update the role of other users. Only the admin is authorized to access this functionality.

3.1.15 View dashboard

The dashboard contains various components and charts including balance of all accounts, total expenses, total incomes, cash flow, expenses structure, incomes structure, cumulative expenses, cumulative incomes, balance overtime, and latest transactions to visualize the budget information of the user.

3.1.16 Generate report

The user can generate report by specifying the type of report and the chart to use for the report. For example, the expenses and incomes structure are available in two charts, pie chart and bar chart.

3.1.17 Export report

The user can export the report by preview it in the browser, download it, or export to a third-party cloud storage such as Google Drive or Dropbox. The report will be exported as pdf format.

3.2 Activity Diagram

3.2.1 Register

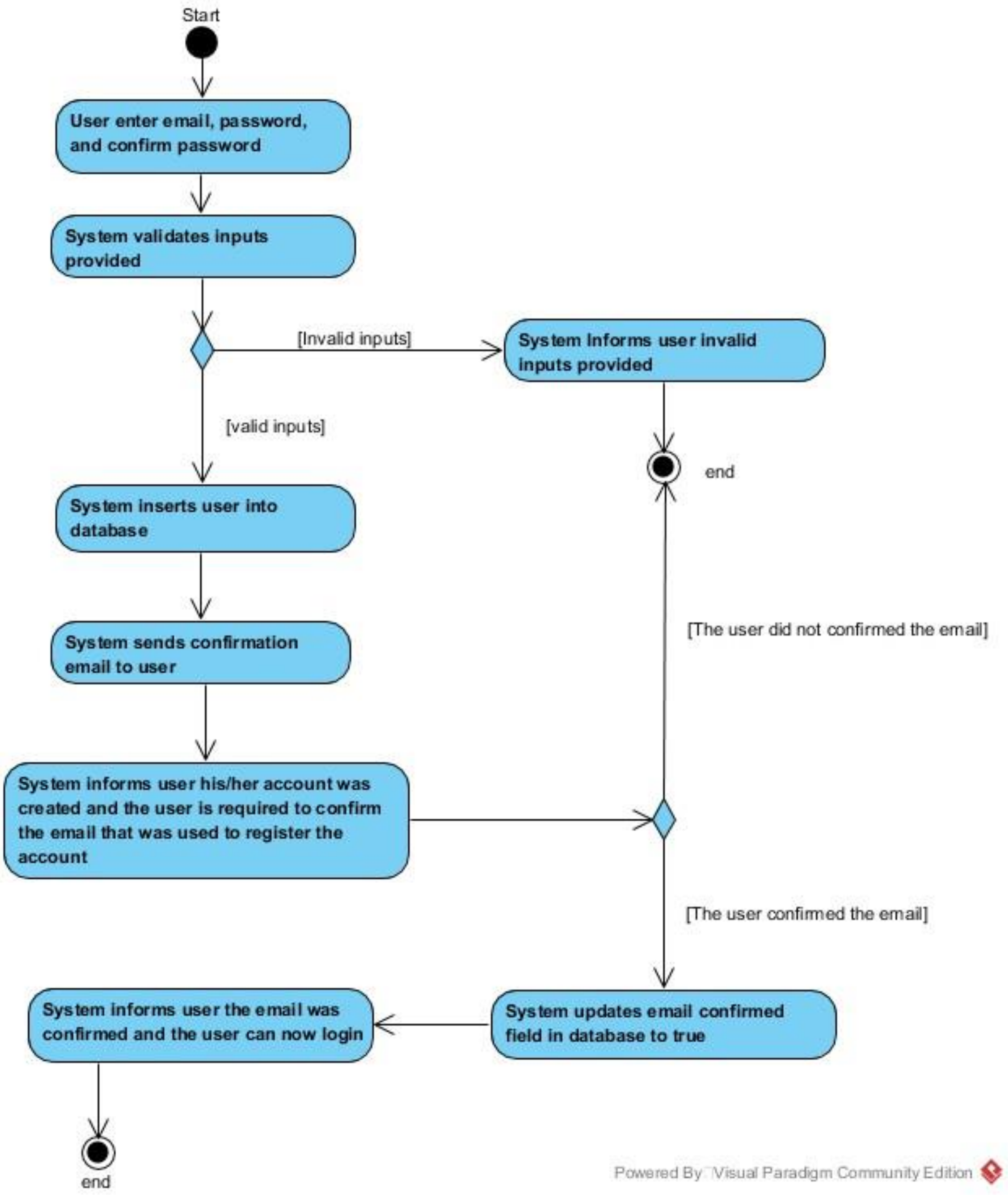

Start
User enter email, password, and confirm password
System validates inputs provided
[Invalid inputs]
System Informs user invalid inputs provided
[valid inputs]
end
System inserts user into database
System sends confirmation email to user
[The user did not confirmed the email]
System informs user his/her account was created and the user is required to confirm the email that was used to register the account
[The user confirmed the email]
System informs user the email was confirmed and the user can now login
System updates email confirmed field in database to true
end
Powered By Visual Paradigm Community Edition

Figure 3-2 Activity Diagram for Register

The user can register an account on the register page. The user will need to enter their email, password, and confirm password using the register form provided. If the inputs provided are valid, an account will be created for the user and a confirmation email will be sent to the user. The users will need to confirm their email first before they are allowed to login to the system.

3.2.2 Login

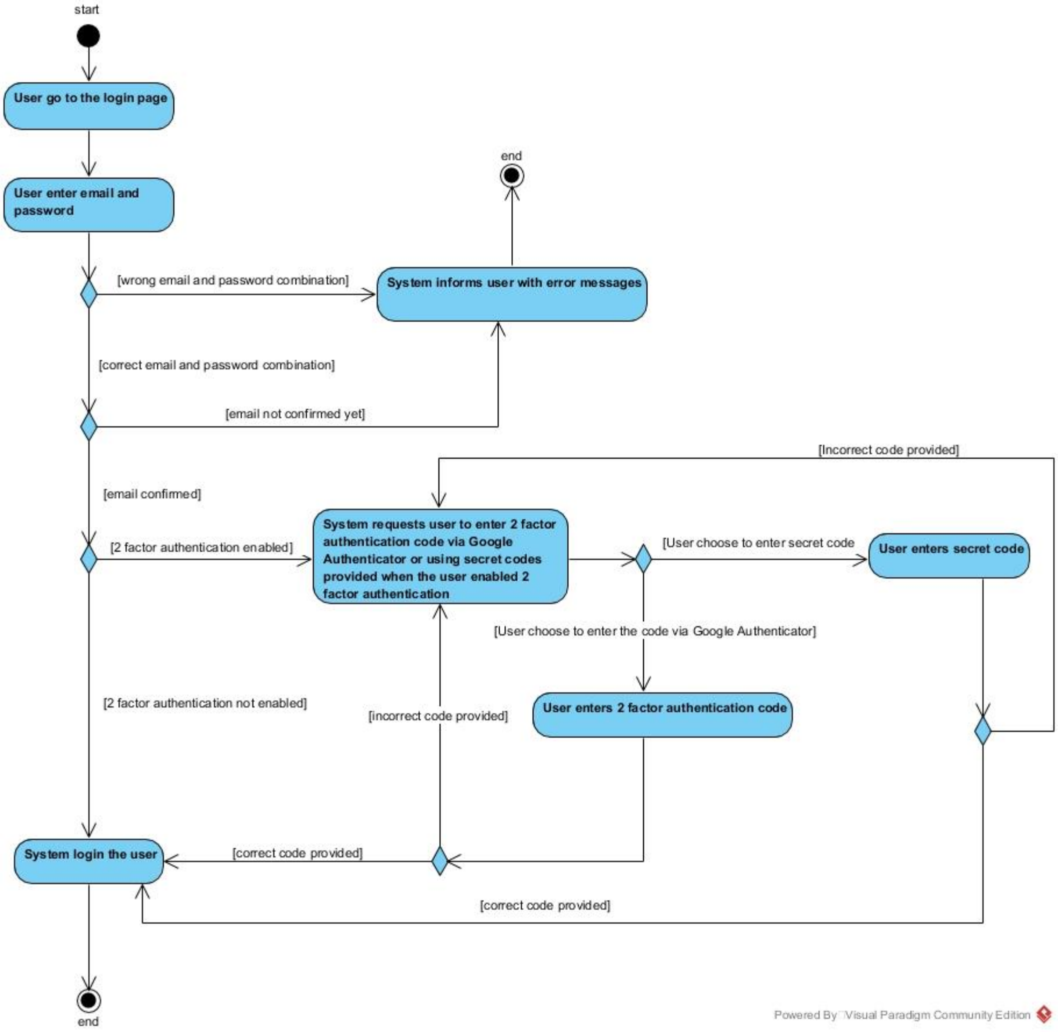

start
User go to the login page
User enter email and password
[wrong email and password combination]
System informs user with error messages
end
[correct email and password combination]
[email not confirmed yet]
[Incorrect code provided]
[email confirmed]
[2 factor authentication enabled]
System requests user to enter 2 factor authentication code via Google Authenticator or using secret codes provided when the user enabled 2 factor authentication
[User choose to enter secret code
User enters secret code
[User choose to enter the code via Google Authenticator]
[2 factor authentication not enabled]
[incorrect code provided]
User enters 2 factor authentication code
System login the user
[correct code provided]
[correct code provided]
end
Powered By Visual Paradigm Community Edition

Figure 3-3 Activity Diagram for Login

The user can login to their account by providing the correct email and password combination. If the user enters a wrong email and password combination or the email was not confirmed yet, the system will display appropriate error messages. If the email and password combination was correct and 2-factor authentication was not enabled, the user will be login to the system. If 2-factor authentication was enabled, the user will need to either enter the code that is available via Google Authenticator or using the secret codes provided when the user setup 2-factor authentication to login to the system.

3.2.3 Register Via Google

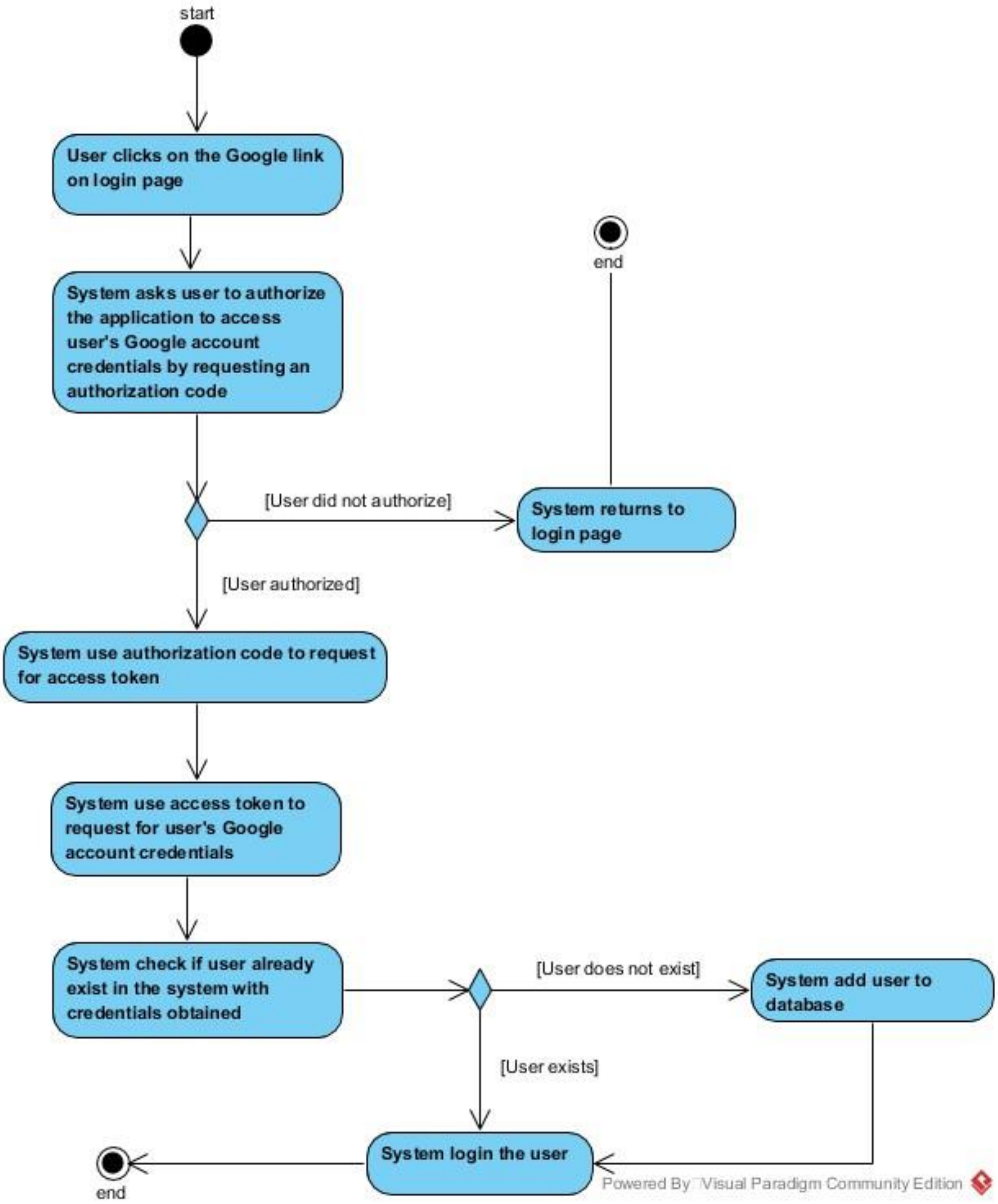

start
User clicks on the Google link on login page
System asks user to authorize the application to access user's Google account credentials by requesting an authorization code
[User did not authorize]
System returns to login page
end
[User authorized]
System use authorization code to request for access token
System use access token to request for user's Google account credentials
System check if user already exist in the system with credentials obtained
[User does not exist]
System add user to database
[User exists]
System login the user
end
Powered By Visual Paradigm Community Edition

Figure 3-4 Activity Diagram for Register via Google

The users can register their account via Google or try to login with Google if they had already created an account with Google. The system will save the Google login information if the user attempts to login via Google the first time.

3.2.4 Reset Password

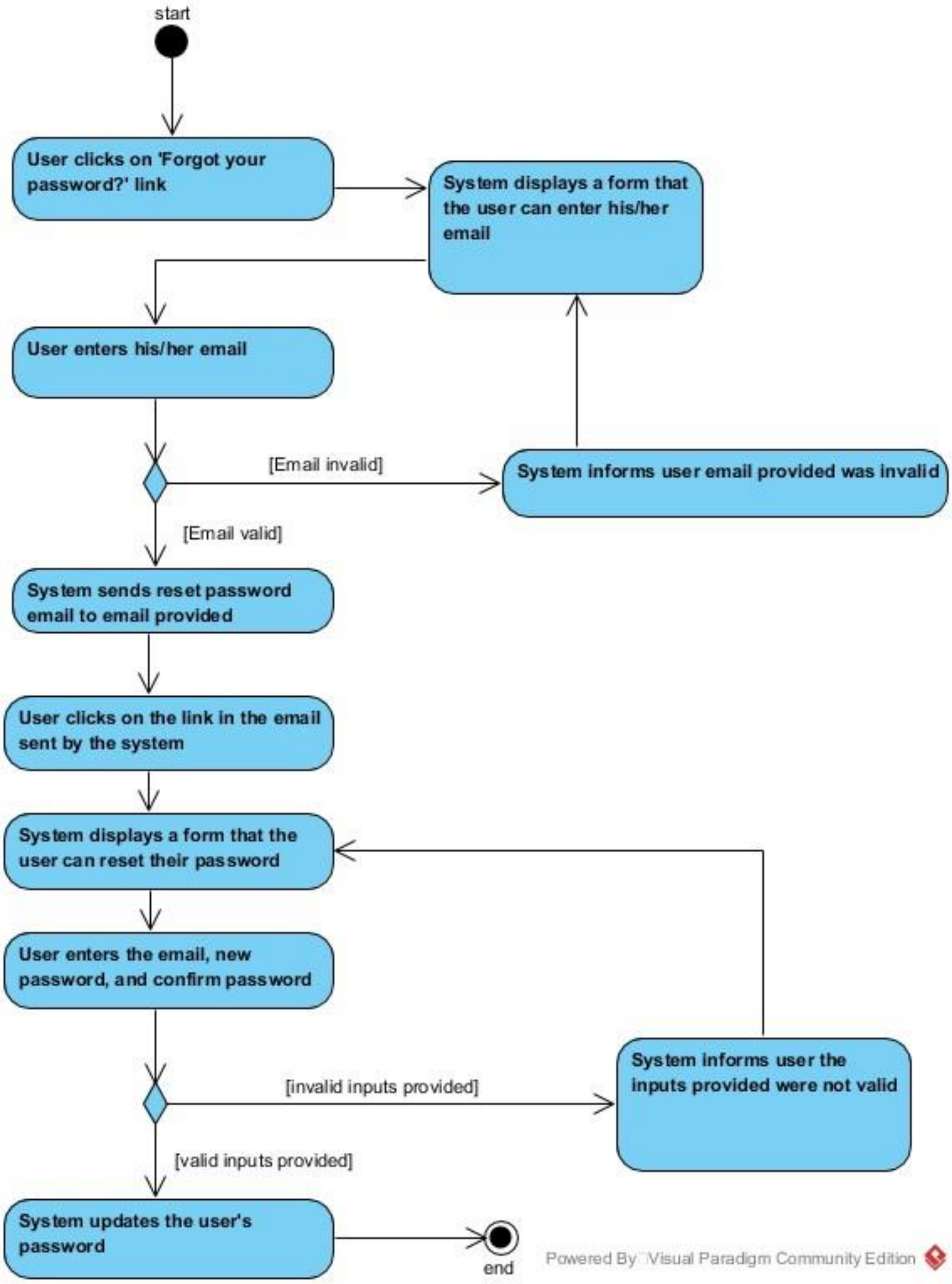

start
User clicks on 'Forgot your password?' link
System displays a form that the user can enter his/her email
User enters his/her email
[Email invalid]
System informs user email provided was invalid
[Email valid]
System sends reset password email to email provided
User clicks on the link in the email sent by the system
System displays a form that the user can reset their password
User enters the email, new password, and confirm password
[invalid inputs provided]
System informs user the inputs provided were not valid
[valid inputs provided]
System updates the user's password
end
Powered By Visual Paradigm Community Edition

Figure 3-5 Activity Diagram for Reset Password

If the users forgot their password, the user can go to the reset password page and enter the email that was used to register the account. A reset password email will be sent to the user and the user can click on the link to go to the reset password page. The users can then enter their email, new password, and confirm password to reset their password.

3.2.5 Add Expense

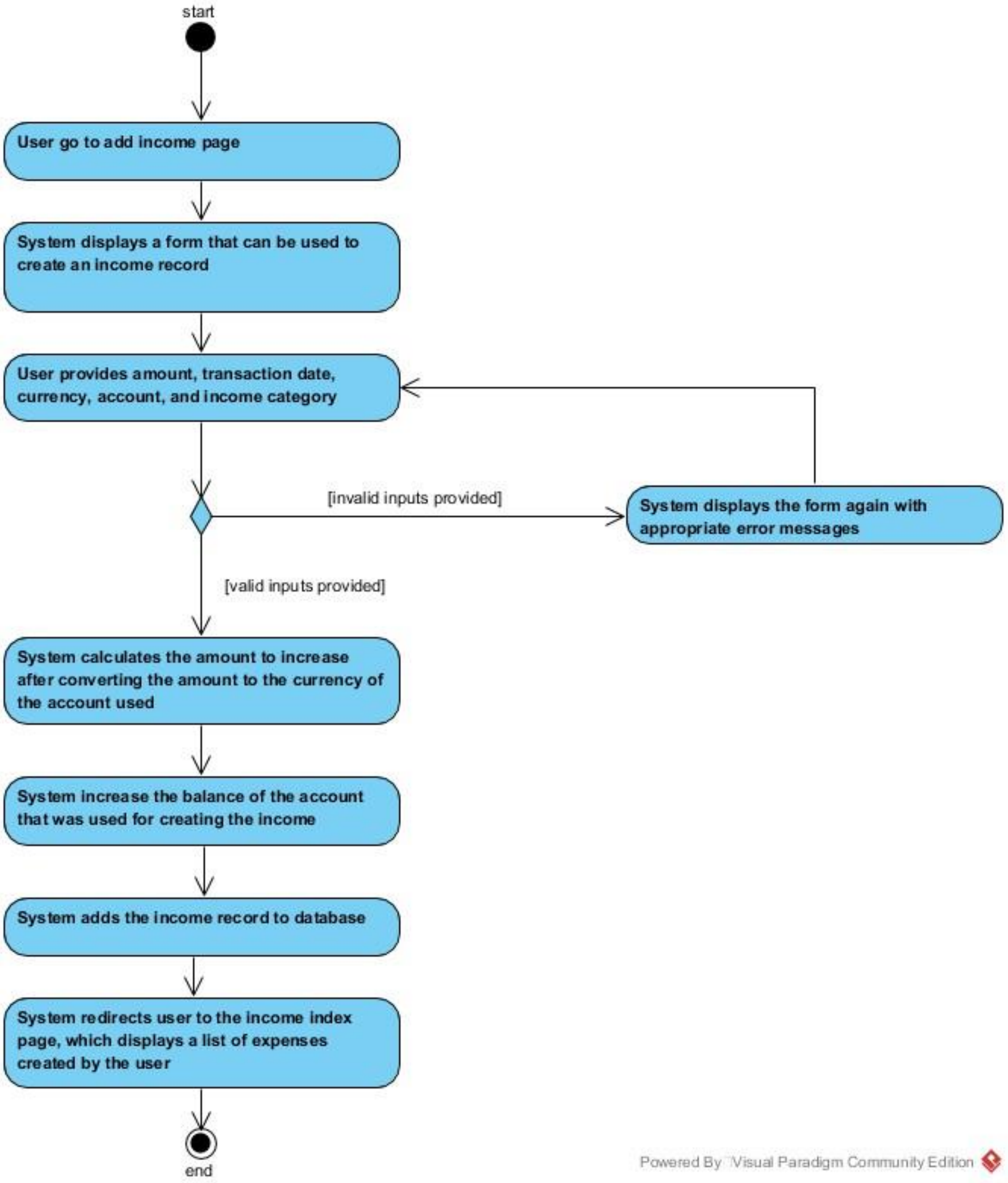

start
User go to add income page
System displays a form that can be used to create an income record
User provides amount, transaction date, currency, account, and income category
[invalid inputs provided]
System displays the form again with appropriate error messages
[valid inputs provided]
System calculates the amount to increase after converting the amount to the currency of the account used
System increase the balance of the account that was used for creating the income
System adds the income record to database
System redirects user to the income index page, which displays a list of expenses created by the user
end
Powered By Visual Paradigm Community Edition

Figure 3-6 Activity Diagram for Add Expense

The user can add an expense record by providing amount, transaction date, currency, account, and expense category. If the inputs provided are valid, the system will update the balance of the account that was used for the expense and add the expense record to the database. After that, the system redirects the user to the index page of expenses, which displays a list of expenses record created by the user.

3.2.6 Delete Expense

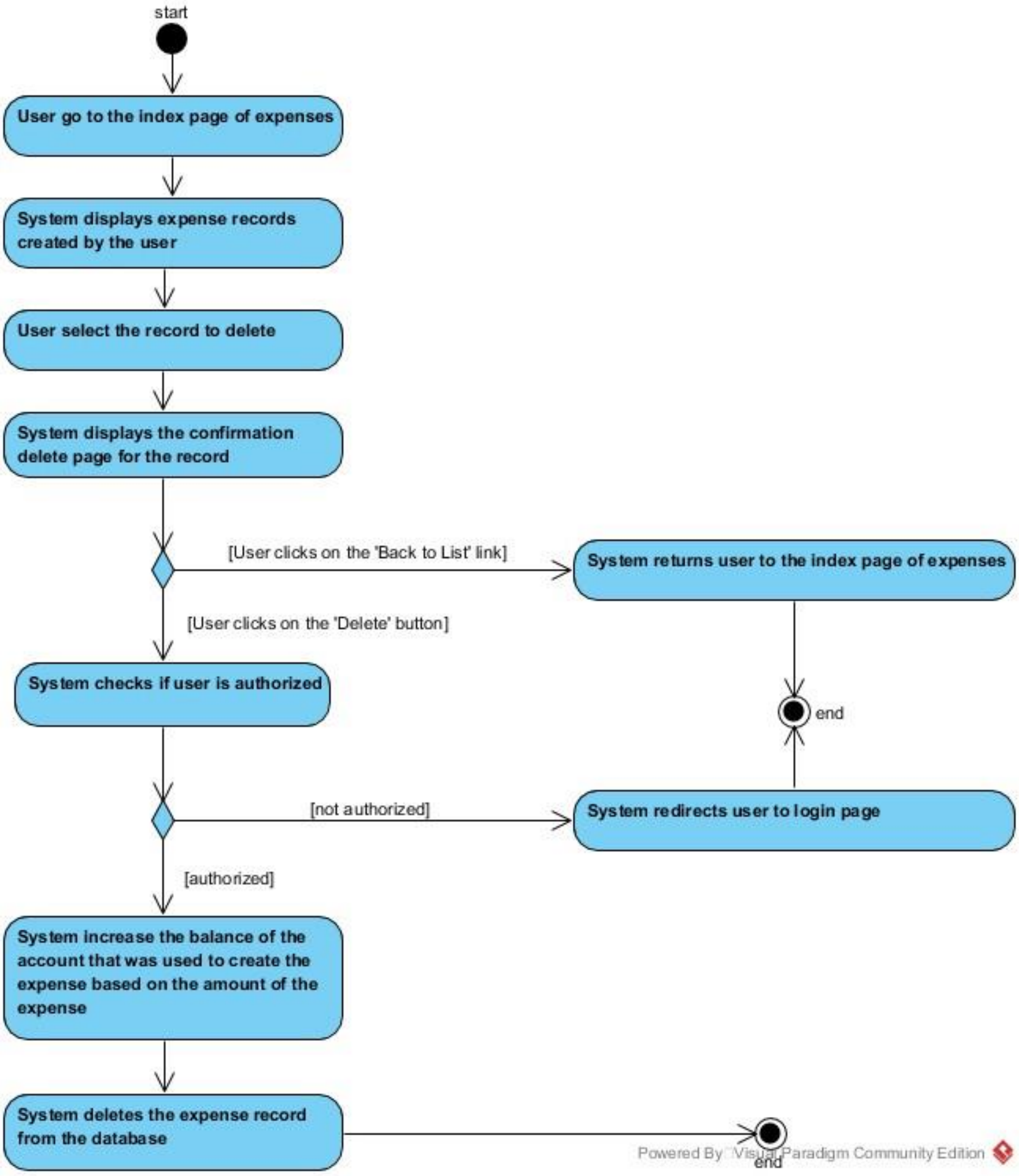

start
User go to the index page of expenses
System displays expense records created by the user
User select the record to delete
System displays the confirmation delete page for the record
[User clicks on the 'Back to List' link]
System returns user to the index page of expenses
[User clicks on the 'Delete' button]
System checks if user is authorized
end
[not authorized]
System redirects user to login page
[authorized]
System increase the balance of the account that was used to create the expense based on the amount of the expense
System deletes the expense record from the database
end
Powered By Visual Paradigm Community Edition

Figure 3-7 Activity Diagram for Delete Expense

The user can delete an expense record by going to the index page of the expenses and select the record to delete. After selecting the record to delete, the system will display a confirmation page to ensure the user actually wants to delete the record. If the delete button was clicked on the delete confirmation page, the system will check if the user is authorized to delete the record. If the user is authorized, the system will update the balance of the account that was used to create the expense and delete the expense record from the database.

3.2.7 Update Expense

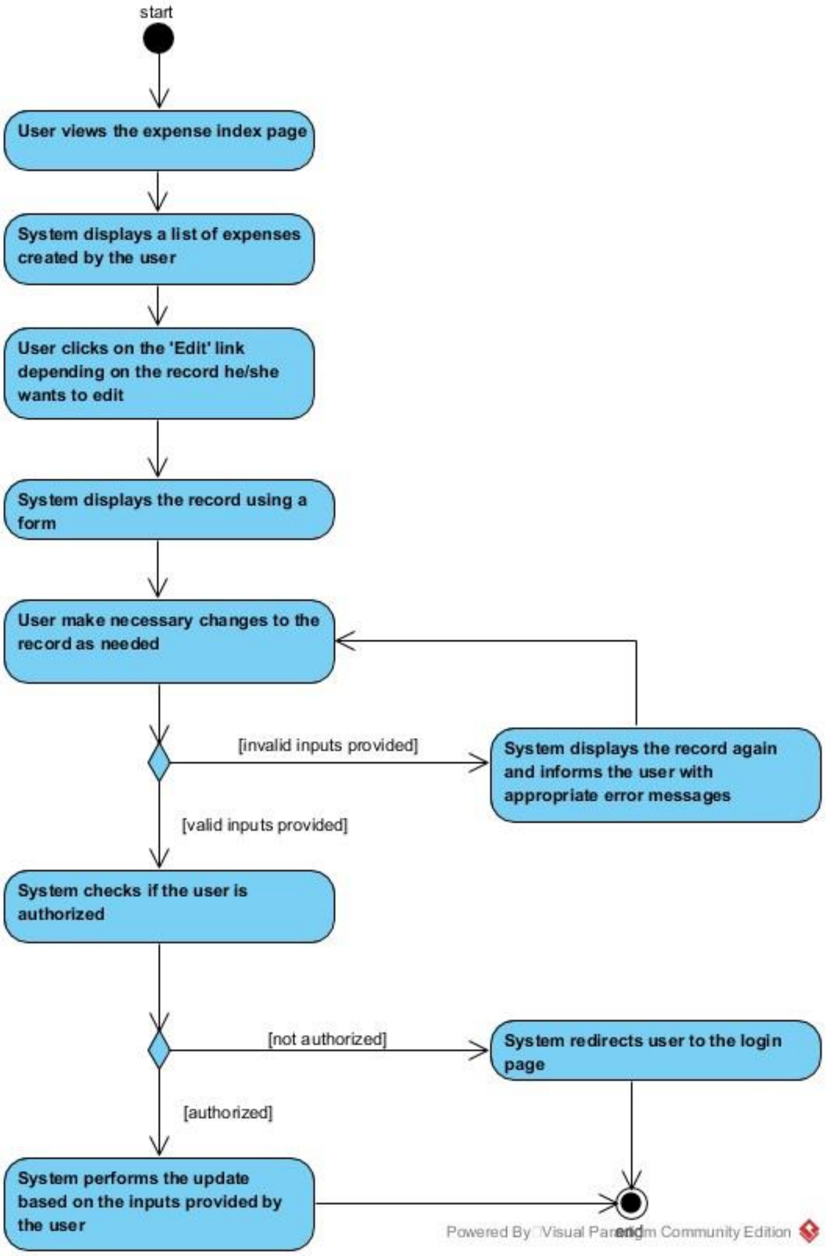

start
User views the expense index page
System displays a list of expenses created by the user
User clicks on the 'Edit' link depending on the record he/she wants to edit
System displays the record using a form
User make necessary changes to the record as needed
[invalid inputs provided]
System displays the record again and informs the user with appropriate error messages
[valid inputs provided]
System checks if the user is authorized
[not authorized]
System redirects user to the login page
[authorized]
System performs the update based on the inputs provided by the user
end
Powered By Visual Paradigm Community Edition

Figure 3-8 Activity Diagram for Update Expense

The user can update an existing expense by going to the index page of expenses, then select to record to edit. The user makes necessary changes to the record and click on the save button. The system will validate the inputs provided by the user and check if the user is authorized. The system will perform the update if inputs provided were valid and the user was authorized to update the record.

3.2.8 Add Income

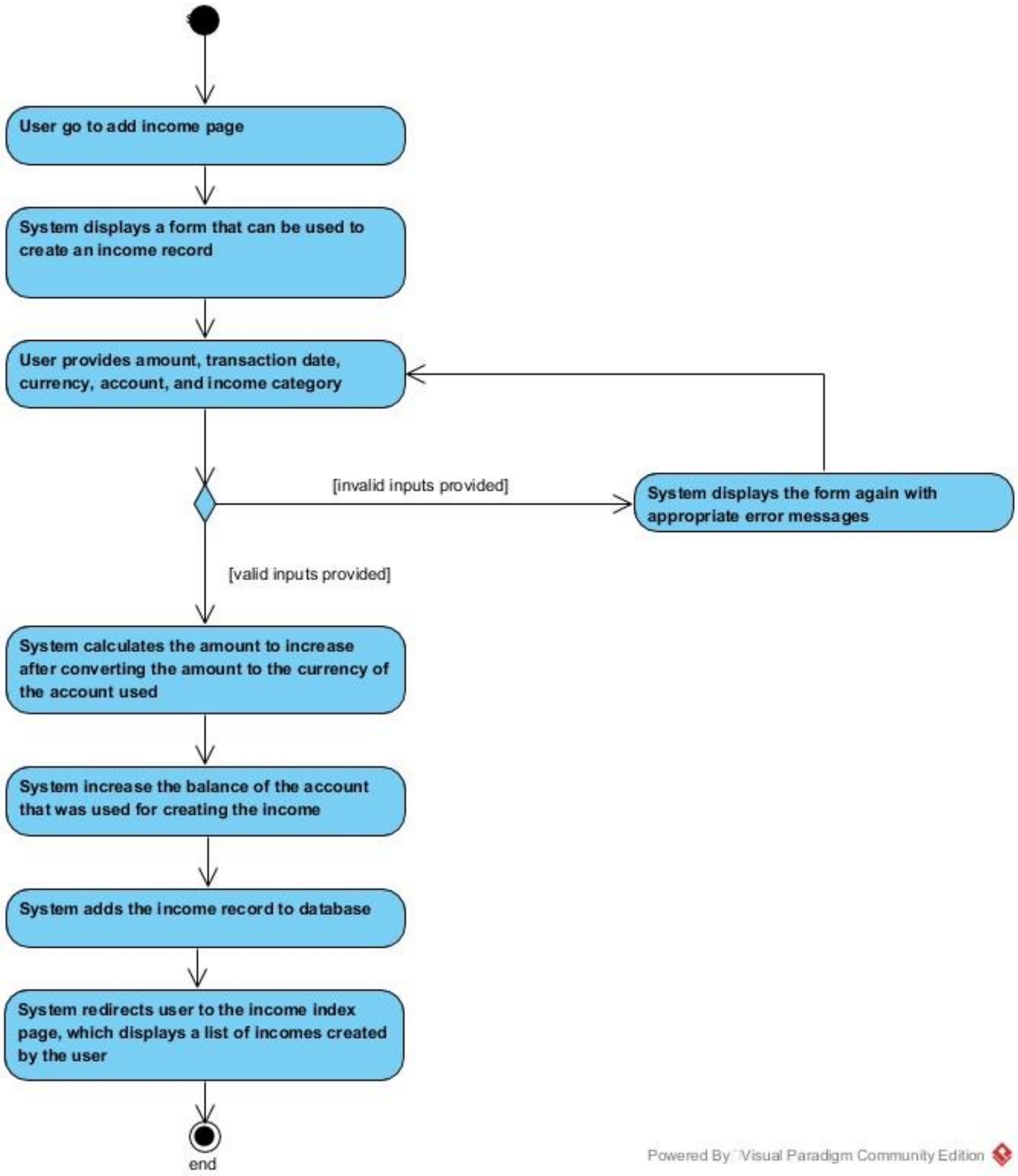

User go to add income page
System displays a form that can be used to create an income record
User provides amount, transaction date, currency, account, and income category
[invalid inputs provided]
System displays the form again with appropriate error messages
[valid inputs provided]
System calculates the amount to increase after converting the amount to the currency of the account used
System increase the balance of the account that was used for creating the income
System adds the income record to database
System redirects user to the income index page, which displays a list of incomes created by the user
end
Powered By Visual Paradigm Community Edition

Figure 3-9 Activity Diagram for Add Income

The user can add an income record by providing amount, transaction date, currency, account, and income category. If the inputs provided are valid, the system will update the balance of the account that was used for the income and add the income record to the database. After that, the system redirects the user to the index page of incomes, which displays a list of incomes record created by the user.

3.2.9 Delete Income

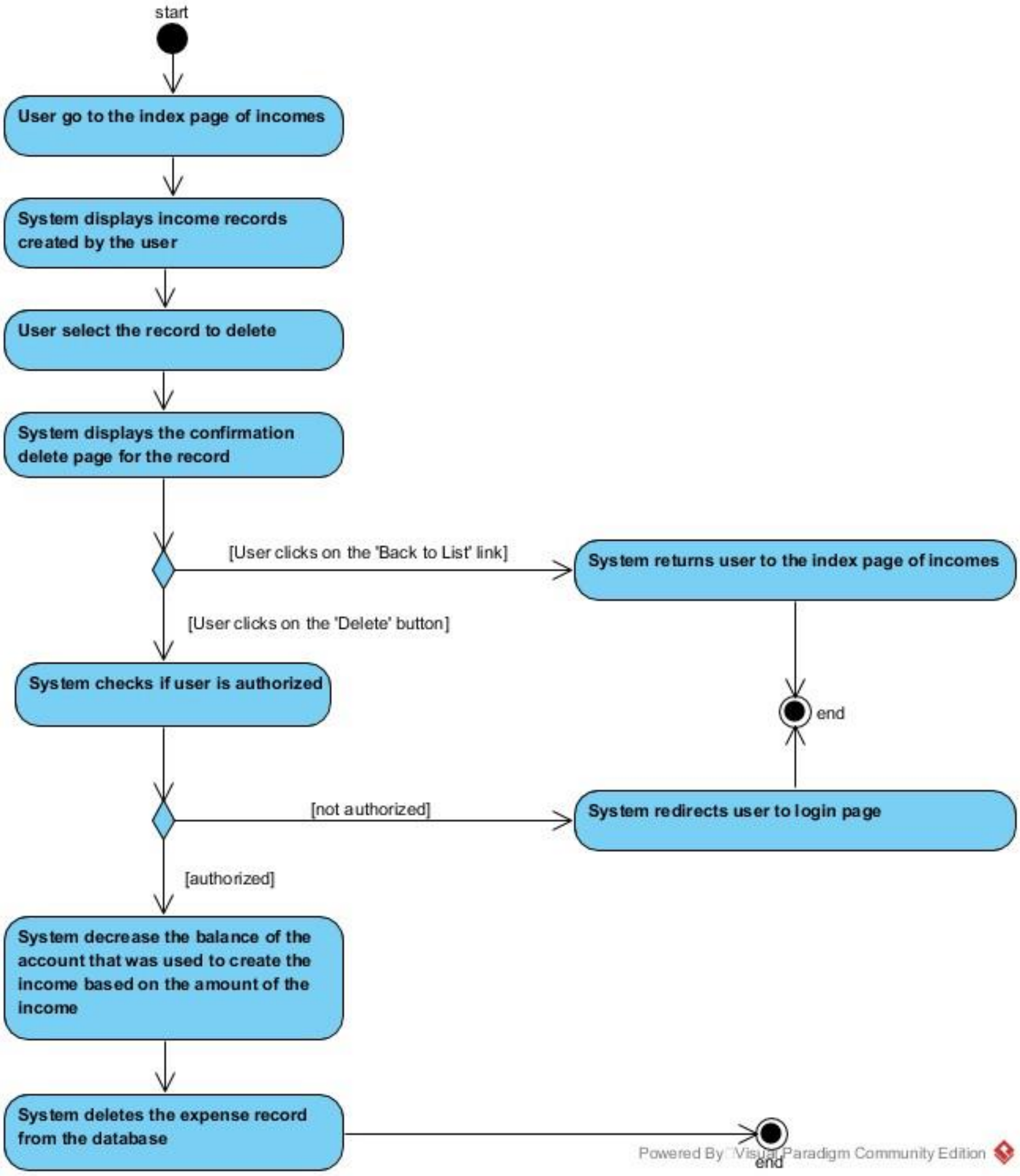

start
User go to the index page of incomes
System displays income records created by the user
User select the record to delete
System displays the confirmation delete page for the record
[User clicks on the 'Back to List' link]
System returns user to the index page of incomes
[User clicks on the 'Delete' button]
System checks if user is authorized
end
[not authorized]
System redirects user to login page
[authorized]
System decrease the balance of the account that was used to create the income based on the amount of the income
System deletes the expense record from the database
end
Powered By Visual Paradigm Community Edition

Figure 3-10 Activity Diagram for Delete Income

The user can delete an income record by going to the index page of the incomes and select the record to delete. After selecting the record to delete, the system will display a confirmation page to ensure the user actually wants to delete the record. If the delete button was clicked on the delete confirmation page, the system will check if the user is authorized to delete the record. If the user is authorized, the system will update the balance of the account that was used to create the income and delete the income record from the database.

3.2.10 Update Income

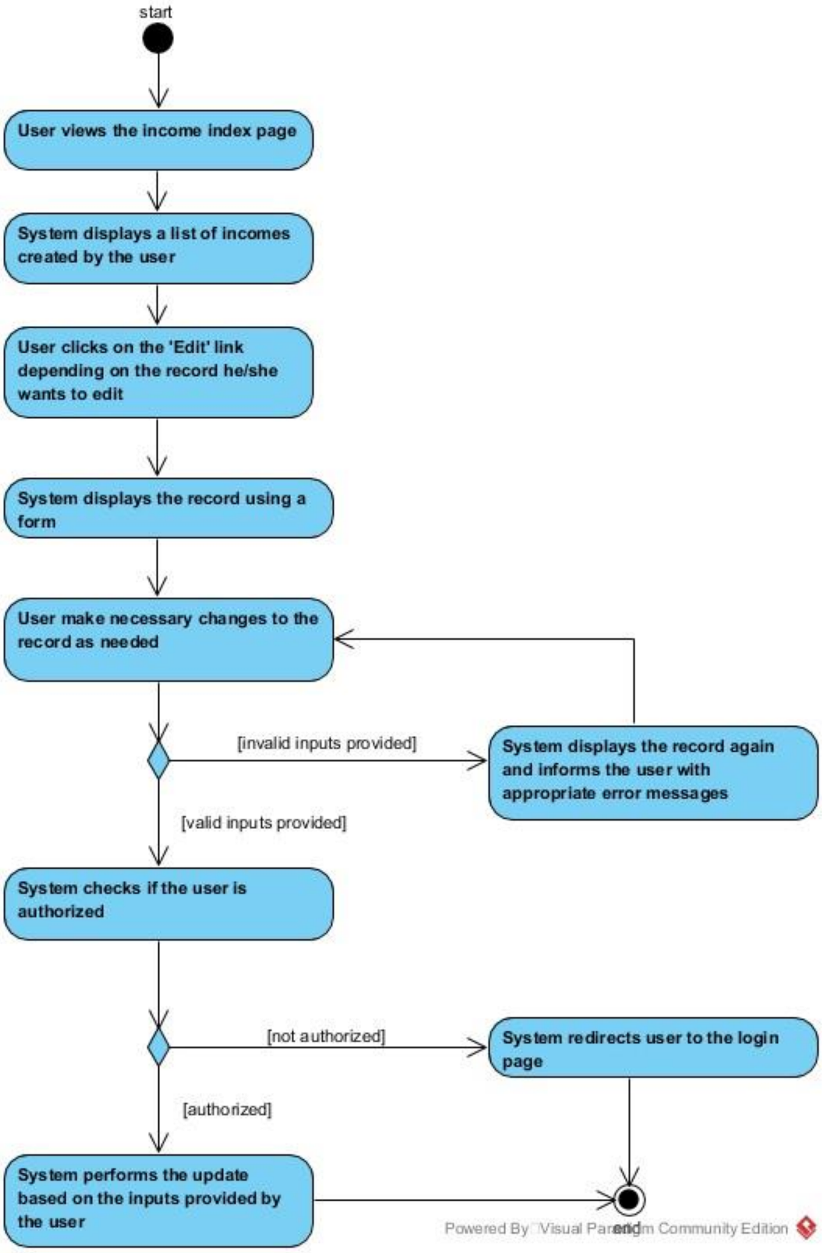

start
User views the income index page
System displays a list of incomes created by the user
User clicks on the 'Edit' link depending on the record he/she wants to edit
System displays the record using a form
User make necessary changes to the record as needed
[invalid inputs provided]
System displays the record again and informs the user with appropriate error messages
[valid inputs provided]
System checks if the user is authorized
[not authorized]
System redirects user to the login page
[authorized]
System performs the update based on the inputs provided by the user
end
Powered By Visual Paradigm Community Edition

Figure 3-11 Activity Diagram for Update Income

The user can update an existing income by going to the index page of incomes, then select to record to edit. The user makes necessary changes to the record and click on the save button. The system will validate the inputs provided by the user and check if the user is authorized. The system will perform the update if inputs provided were valid and the user was authorized to update the record.

3.2.11 Add Transfer

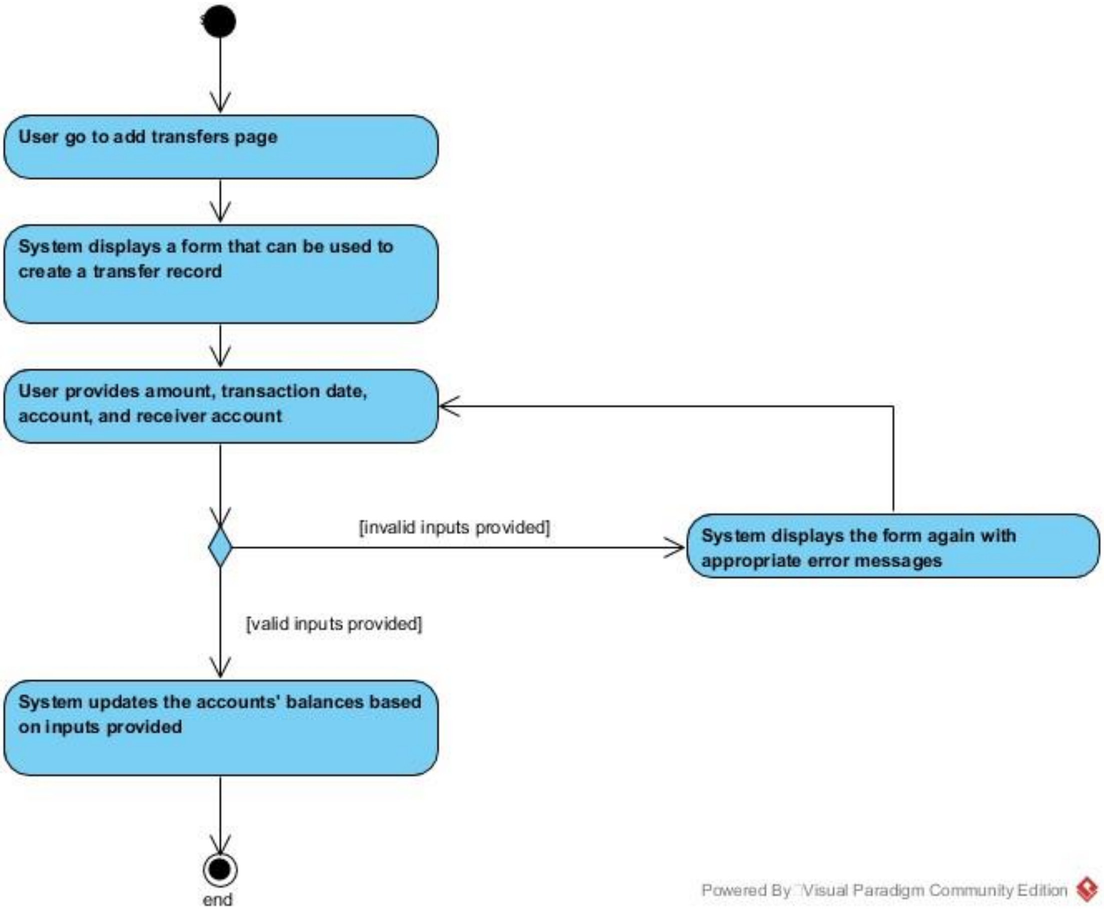

User go to add transfers page
System displays a form that can be used to create a transfer record
User provides amount, transaction date, account, and receiver account
[invalid inputs provided]
System displays the form again with appropriate error messages
[valid inputs provided]
System updates the accounts' balances based on inputs provided
end
Powered By Visual Paradigm Community Edition

Figure 3-12 Activity Diagram for Add Transfer

The user can add transfer record by providing amount, transaction date, account, and receiver account. If the inputs provided are valid, the system will deduct balance of account and increase balance of receiver account, then the system will add the transfer record to the database. After that, the system redirects the user to the index page of transfer, which displays a list of transfer records created by the user.

3.2.12 Delete Transfer

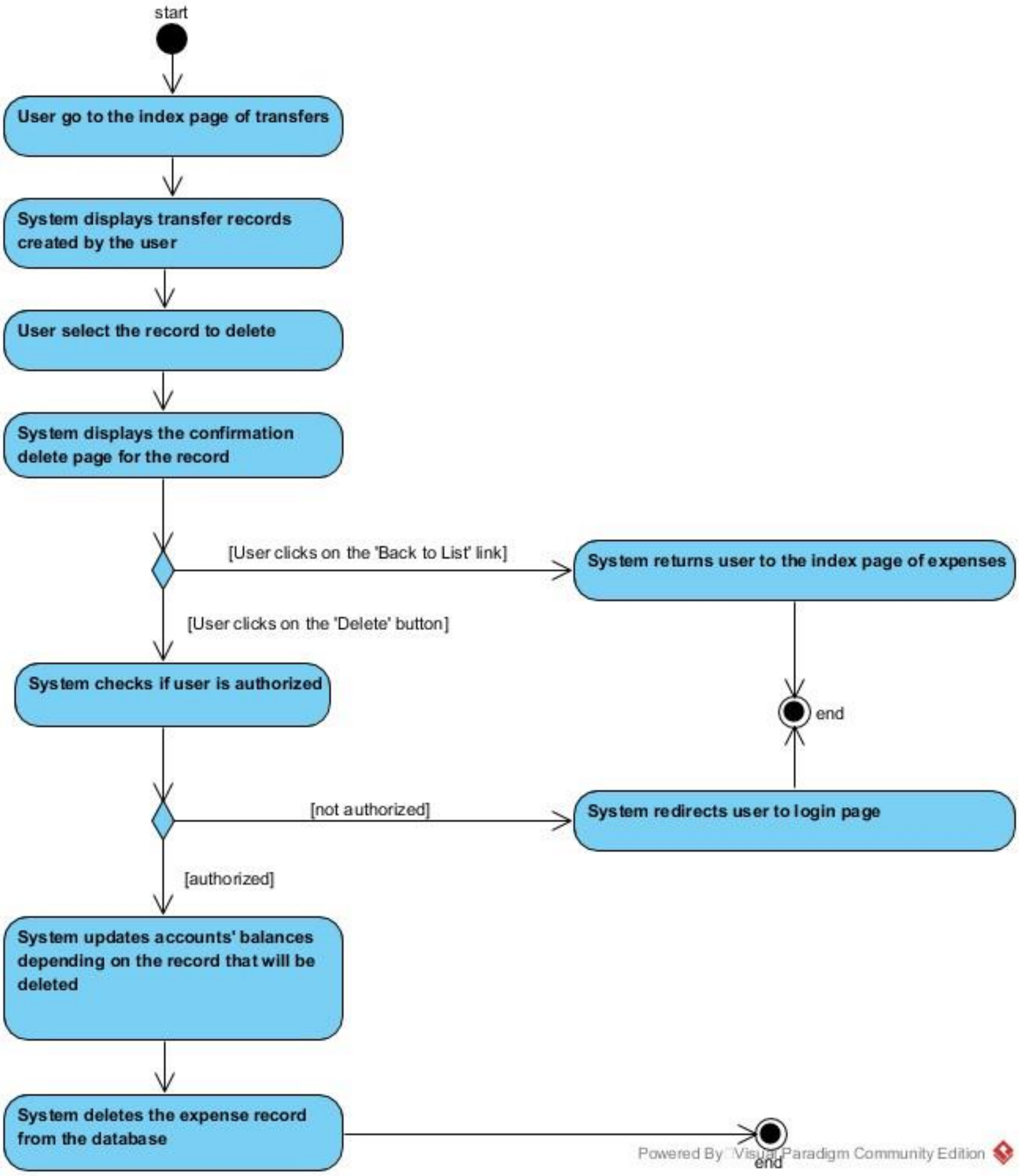

start
User go to the index page of transfers
System displays transfer records created by the user
User select the record to delete
System displays the confirmation delete page for the record
[User clicks on the 'Back to List' link]
System returns user to the index page of expenses
[User clicks on the 'Delete' button]
System checks if user is authorized
end
[not authorized]
System redirects user to login page
[authorized]
System updates accounts' balances depending on the record that will be deleted
System deletes the expense record from the database
end
Powered By Visual Paradigm Community Edition

Figure 3-13 Activity Diagram for Delete Transfer

The user can delete a transfer record by going to the index page of the transfers and select the record to delete. After selecting the record to delete, the system will display a confirmation page to ensure the user actually wants to delete the record. If the delete button was clicked on the delete confirmation page, the system will check if the user is authorized to delete the record. If the user is authorized, the system will update the balance of the account and receiver account that was used to create the transfer and delete the transfer record from the database.

3.2.13 Update Transfer

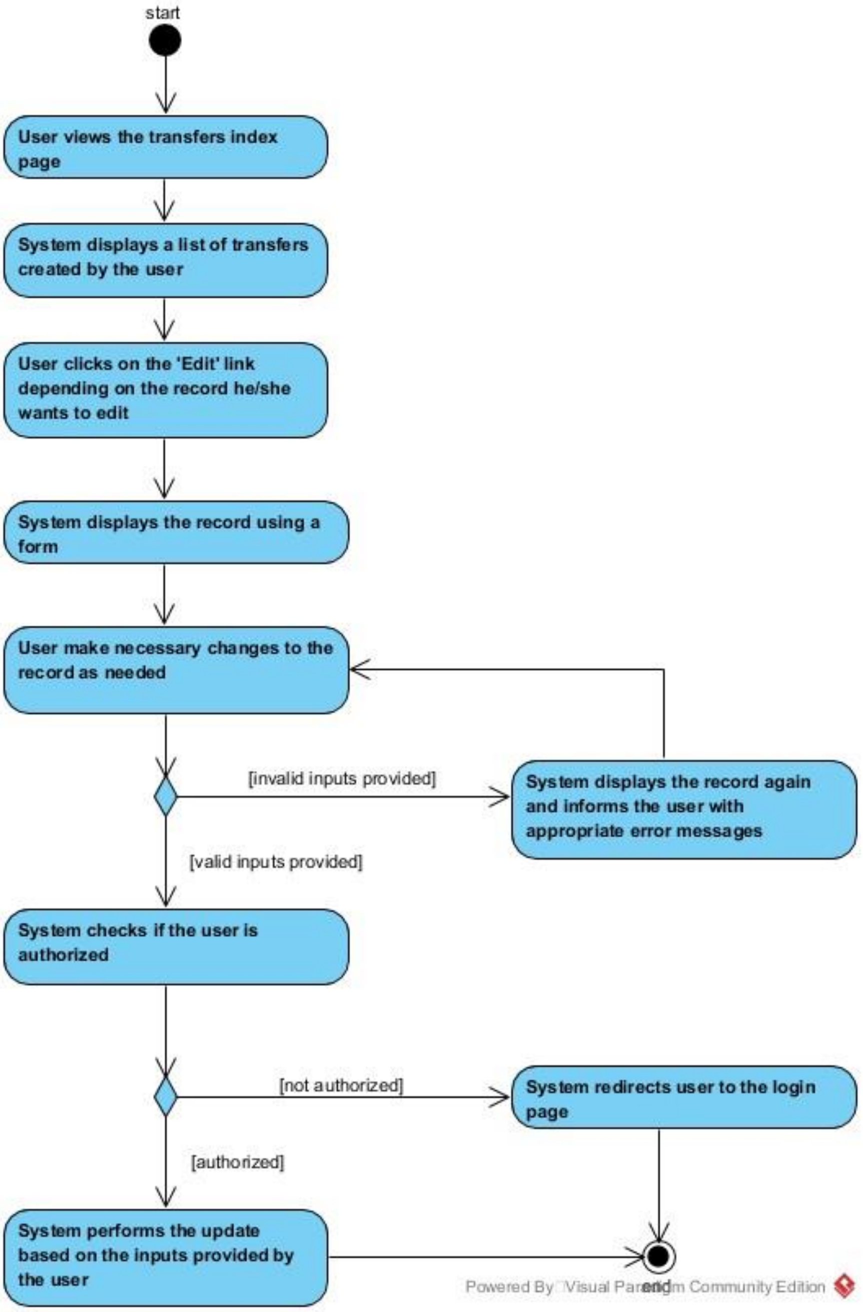

start
User views the transfers index page
System displays a list of transfers created by the user
User clicks on the 'Edit' link depending on the record he/she wants to edit
System displays the record using a form
User make necessary changes to the record as needed
[invalid inputs provided]
System displays the record again and informs the user with appropriate error messages
[valid inputs provided]
System checks if the user is authorized
[not authorized]
System redirects user to the login page
[authorized]
System performs the update based on the inputs provided by the user
end
Powered By Visual Paradigm Community Edition

Figure 3-14 Activity Diagram for Update Transfer

The user can update an existing transfer record by going to the index page of transfers, then select to record to edit. The user makes necessary changes to the record and click on the save button. The system will validate the inputs provided by the user and check if the user is authorized. The system will perform the update if inputs provided were valid and the user was authorized to update the record.

3.2.14 Update Budget

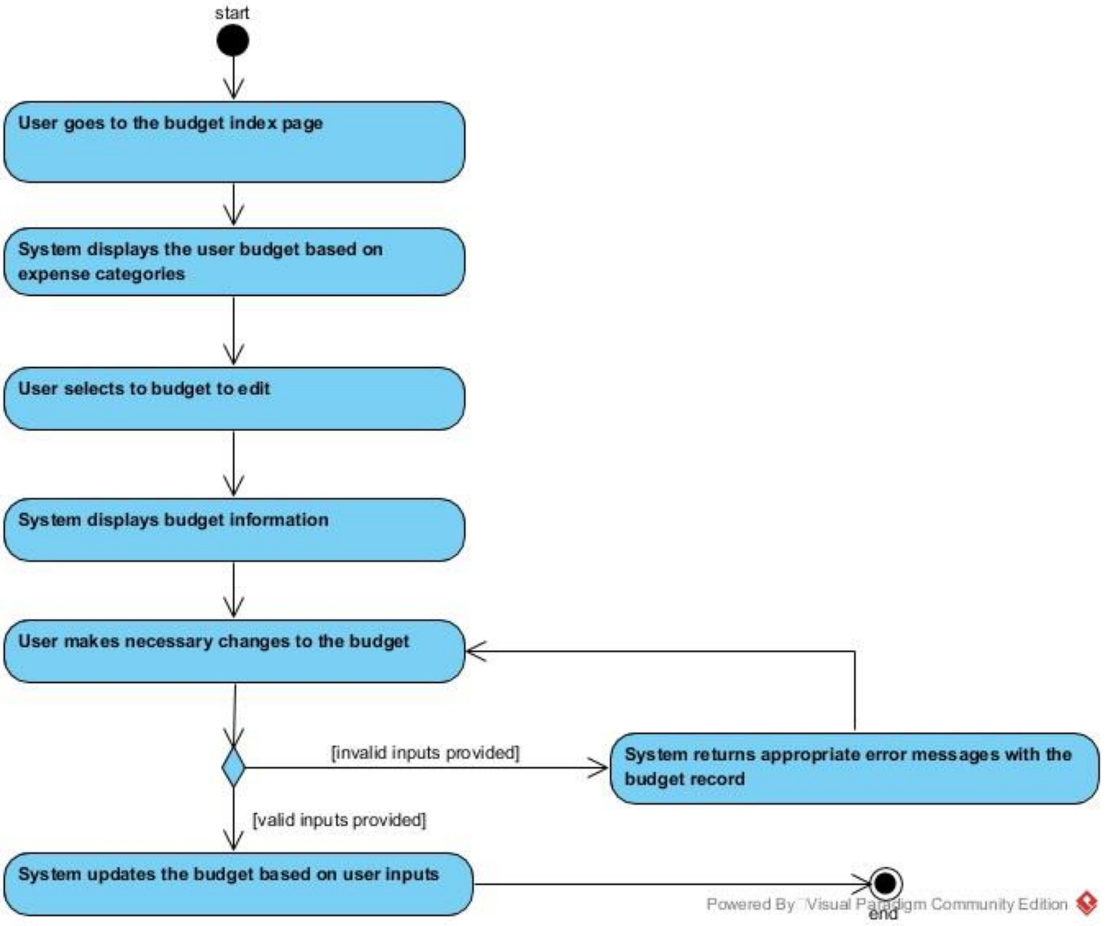

start
User goes to the budget index page
System displays the user budget based on expense categories
User selects to budget to edit
System displays budget information
User makes necessary changes to the budget
[invalid inputs provided]
System returns appropriate error messages with the budget record
[valid inputs provided]
System updates the budget based on user inputs
end
Powered By Visual Paradigm Community Edition

Figure 3-15 Activity Diagram for Update Budget

The user can update an existing budget record by going to the index page of budgets, then select to record to edit. The user makes necessary changes to the record and click on the save button. The system will validate the inputs provided by the user and check if the user is authorized. The system will perform the update if inputs provided were valid and the user was authorized to update the record.

3.2.15 Add Account

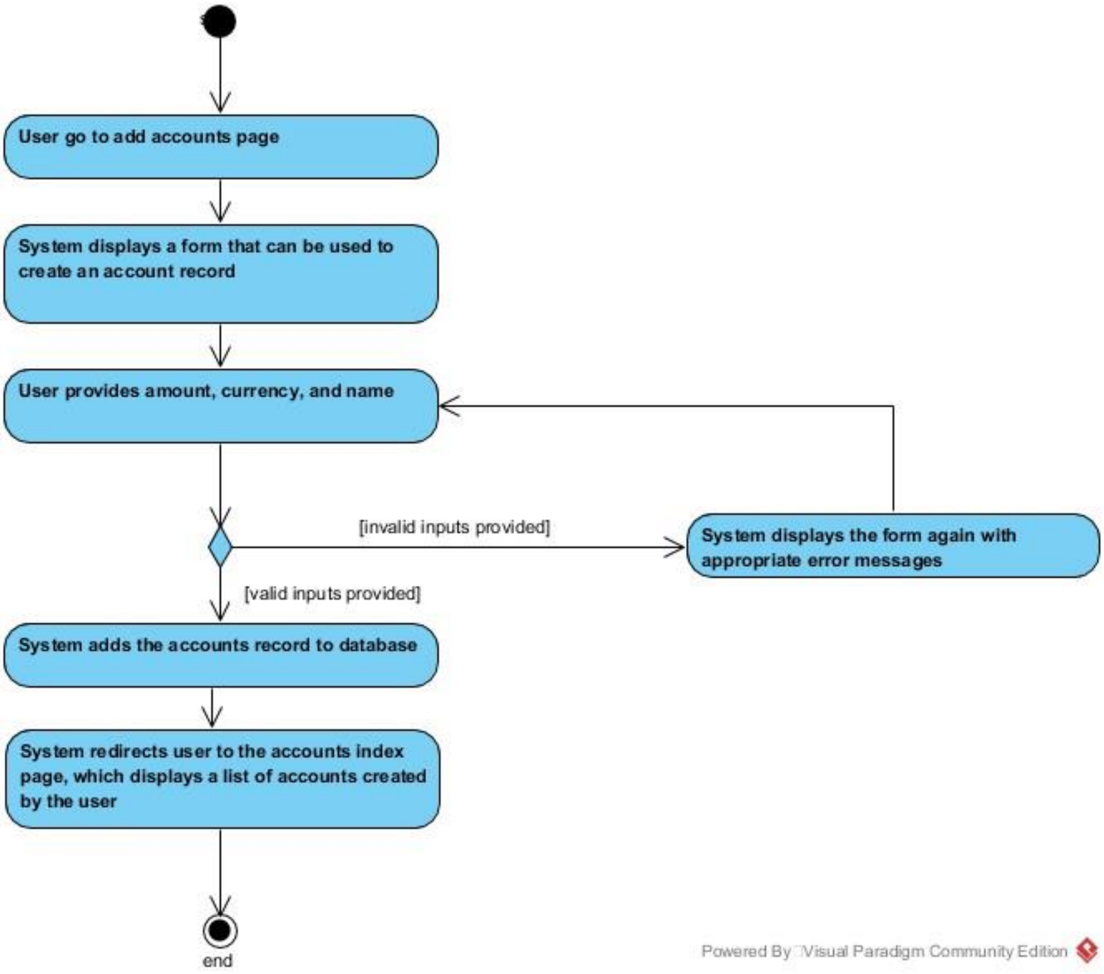

User go to add accounts page
System displays a form that can be used to create an account record
User provides amount, currency, and name
[invalid inputs provided]
System displays the form again with appropriate error messages
[valid inputs provided]
System adds the accounts record to database
System redirects user to the accounts index page, which displays a list of accounts created by the user
end
Powered By Visual Paradigm Community Edition

Figure 3-16 Activity Diagram for Add Account

The user can add an account record by providing amount, currency, and name. If the inputs provided are valid, the system will add the account to the database. After that, the system redirects the user to the index page of accounts, which displays a list of accounts created by the user.

3.2.16 Delete Account

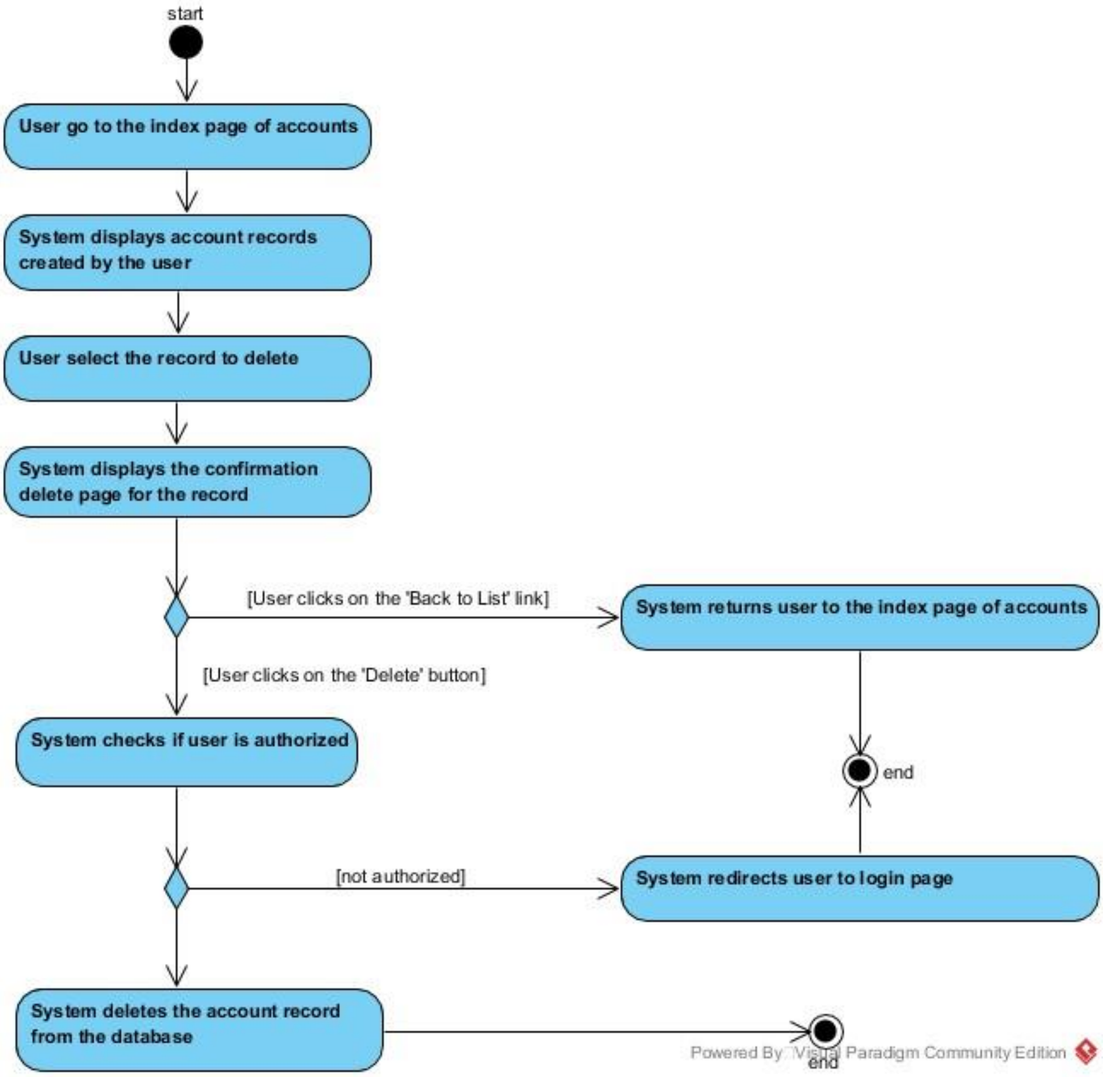

start
User go to the index page of accounts
System displays account records created by the user
User select the record to delete
System displays the confirmation delete page for the record
[User clicks on the 'Back to List' link]
System returns user to the index page of accounts
[User clicks on the 'Delete' button]
System checks if user is authorized
end
[not authorized]
System redirects user to login page
System deletes the account record from the database
end
Powered By Visual Paradigm Community Edition

Figure 3-17 Activity Diagram for Delete Account

The user can delete an account record by going to the index page of the accounts and select the record to delete. After selecting the record to delete, the system will display a confirmation page to ensure the user actually wants to delete the record. If the delete button was clicked on the delete confirmation page, the system will check if the user is authorized to delete the record. If the user is authorized, the system will remove the account from the database.

3.2.17 Update Account

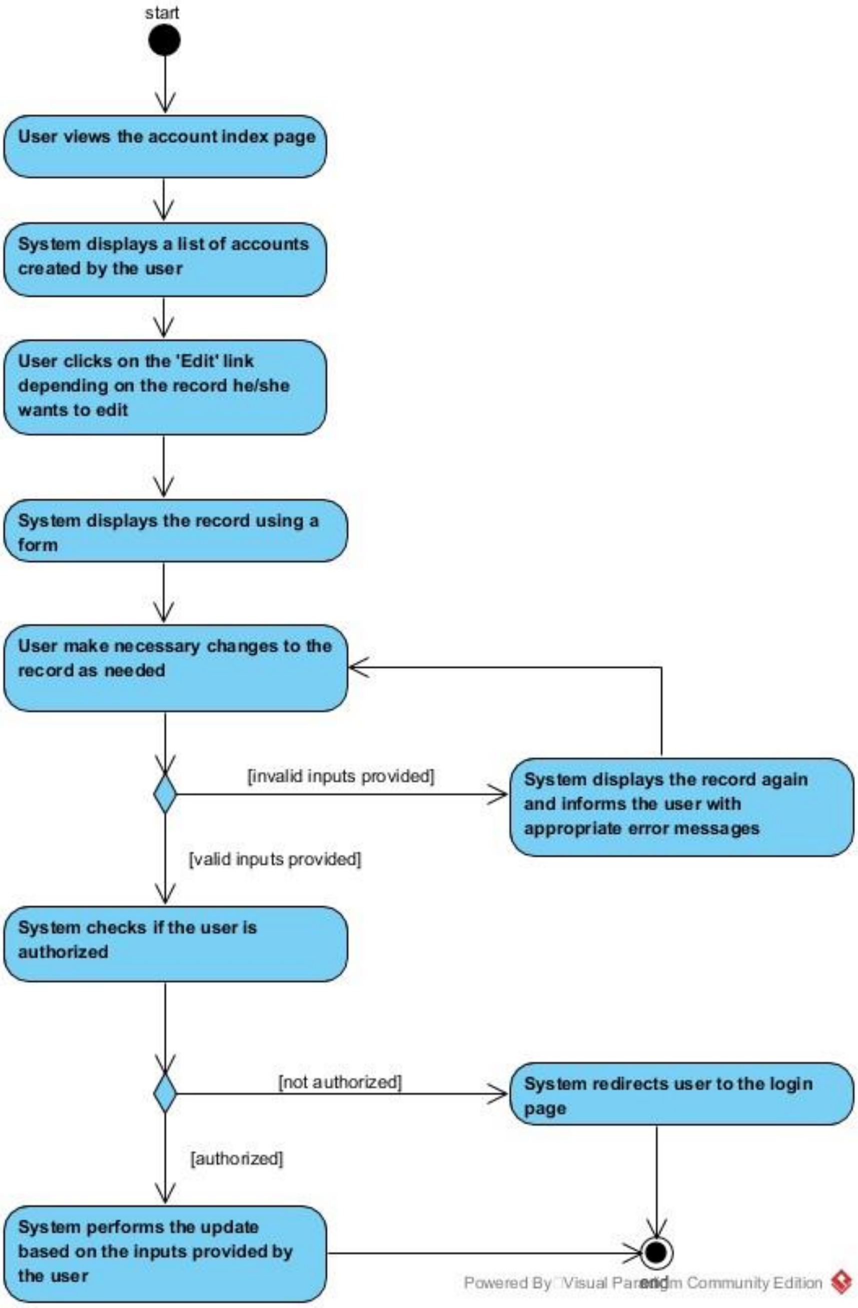

start
User views the account index page
System displays a list of accounts created by the user
User clicks on the 'Edit' link depending on the record he/she wants to edit
System displays the record using a form
User make necessary changes to the record as needed
[invalid inputs provided]
System displays the record again and informs the user with appropriate error messages
[valid inputs provided]
System checks if the user is authorized
[not authorized]
System redirects user to the login page
[authorized]
System performs the update based on the inputs provided by the user
end
Powered By Visual Paradigm Community Edition

Figure 3-18 Activity Diagram for Update Account

The user can update an existing account record by going to the index page of accounts, then select to record to edit. The user makes necessary changes to the record and click on the save button. The system will validate the inputs provided by the user and check if the user is authorized. The system will perform the update if inputs provided were valid and the user was authorized to update the record.

3.2.18 Add Currency

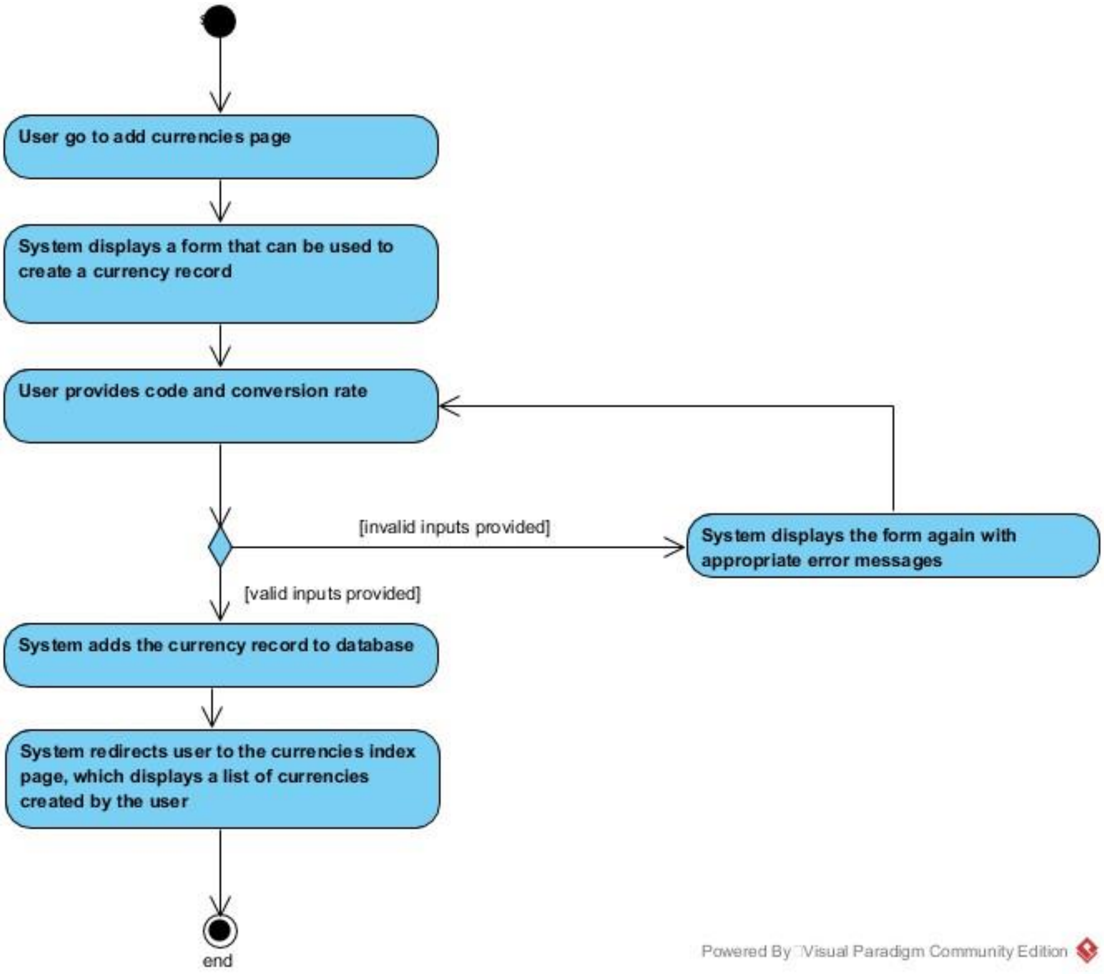

User go to add currencies page
System displays a form that can be used to create a currency record
User provides code and conversion rate
[invalid inputs provided]
System displays the form again with appropriate error messages
[valid inputs provided]
System adds the currency record to database
System redirects user to the currencies index page, which displays a list of currencies created by the user
end
Powered By Visual Paradigm Community Edition

Figure 3-19 Activity Diagram for Add Currency

The user can add a preferred currency by providing code and conversion rate. If the inputs provided are valid, the system will add the currency to the database. After that, the system redirects the user to the index page of currencies, which displays a list of currencies created by the user.

3.2.19 Update Currency

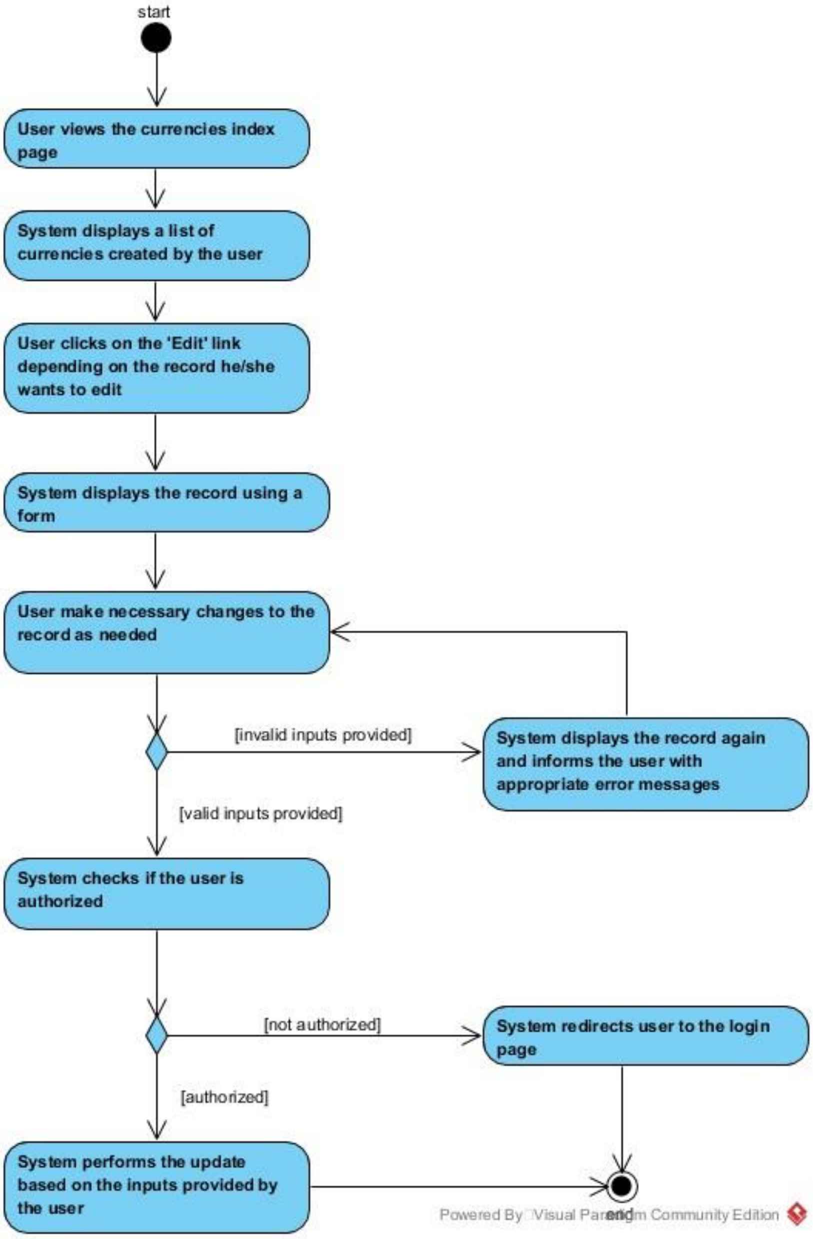

start
User views the currencies index page
System displays a list of currencies created by the user
User clicks on the 'Edit' link depending on the record he/she wants to edit
System displays the record using a form
User make necessary changes to the record as needed
[invalid inputs provided]
System displays the record again and informs the user with appropriate error messages
[valid inputs provided]
System checks if the user is authorized
[not authorized]
System redirects user to the login page
[authorized]
System performs the update based on the inputs provided by the user
end
Powered By Visual Paradigm Community Edition

Figure 3-20 Activity Diagram for Update Currency

The user can update an existing currency record by going to the index page of currencies, then select to record to edit. The user makes necessary changes to the record and click on the save button. The system will validate the inputs provided by the user and check if the user is authorized. The system will perform the update if inputs provided were valid and the user was authorized to update the record.

3.2.20 Delete Currency

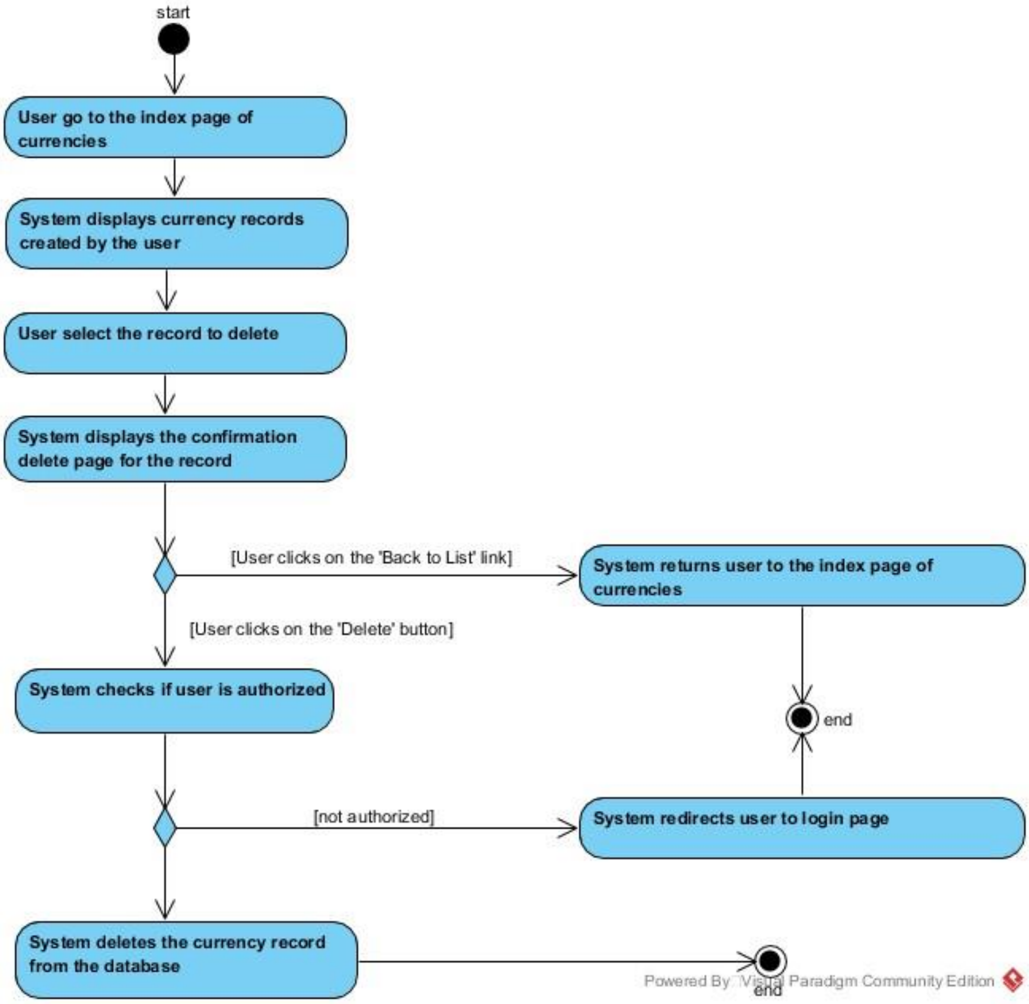

start
User go to the index page of currencies
System displays currency records created by the user
User select the record to delete
System displays the confirmation delete page for the record
[User clicks on the 'Back to List' link]
System returns user to the index page of currencies
[User clicks on the 'Delete' button]
System checks if user is authorized
end
[not authorized]
System redirects user to login page
System deletes the currency record from the database
end
Powered By Visual Paradigm Community Edition

Figure 3-21 Activity Diagram for Delete Currency

The user can delete a currency record by going to the index page of the currencies and select the record to delete. After selecting the record to delete, the system will display a confirmation page to ensure the user actually wants to delete the record. If the delete button was clicked on the delete confirmation page, the system will check if the user is authorized to delete the record. If the user is authorized, the system will remove the currency from the database.

3.2.21 Profile

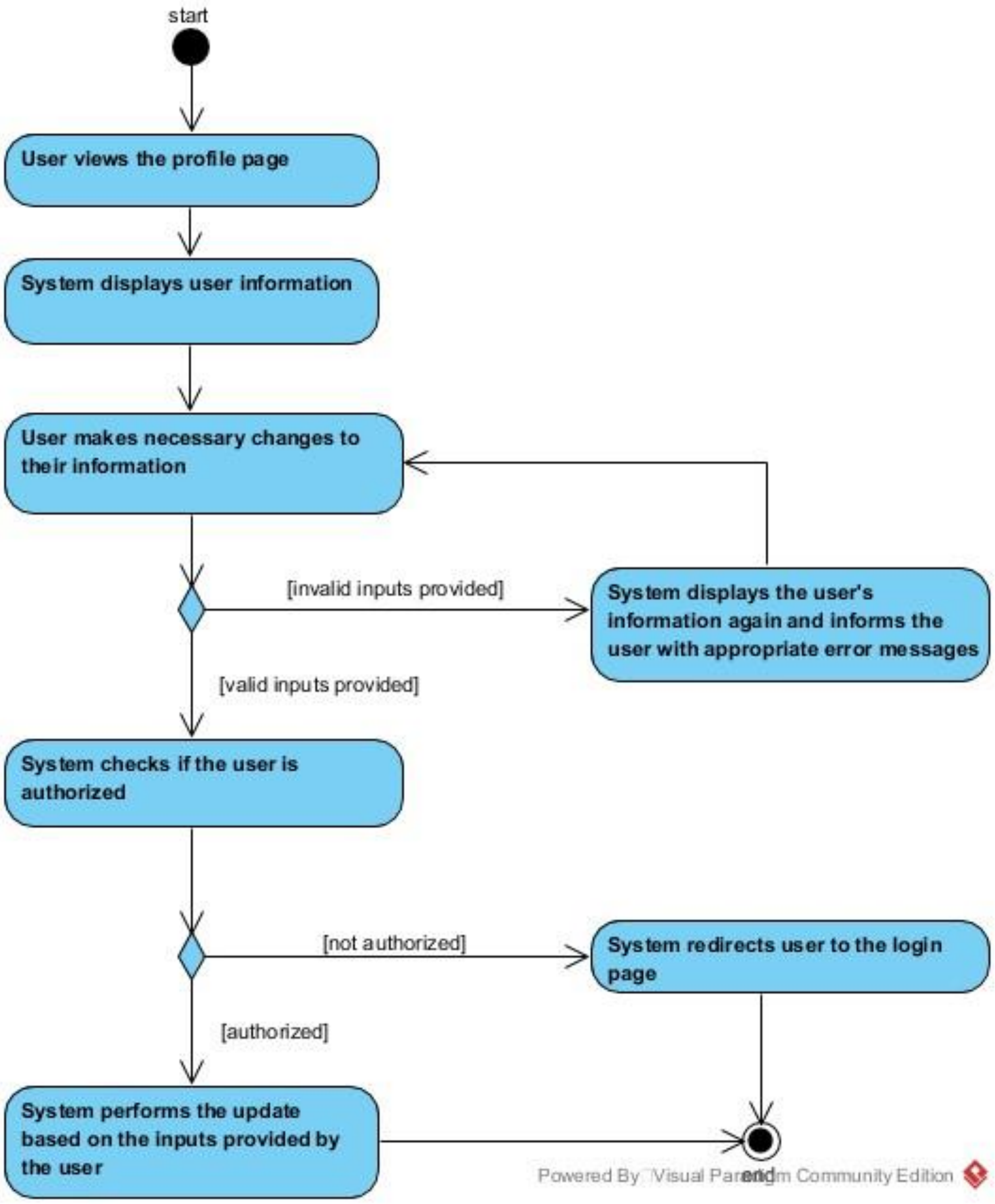

start
User views the profile page
System displays user information
User makes necessary changes to their information
[invalid inputs provided]
System displays the user's information again and informs the user with appropriate error messages
[valid inputs provided]
System checks if the user is authorized
[not authorized]
System redirects user to the login page
[authorized]
System performs the update based on the inputs provided by the user
end
Powered By Visual Paradigm Community Edition

Figure 3-22 Activity Diagram for Profile

The user can go to the profile page to update his/her information. If the user wants to update his/her information, the user will need to make necessary changes on the form and click on the save button. The system will validate if the inputs provided were valid, if the inputs were valid, the system will update the user’s information.

3.2.22 Theme

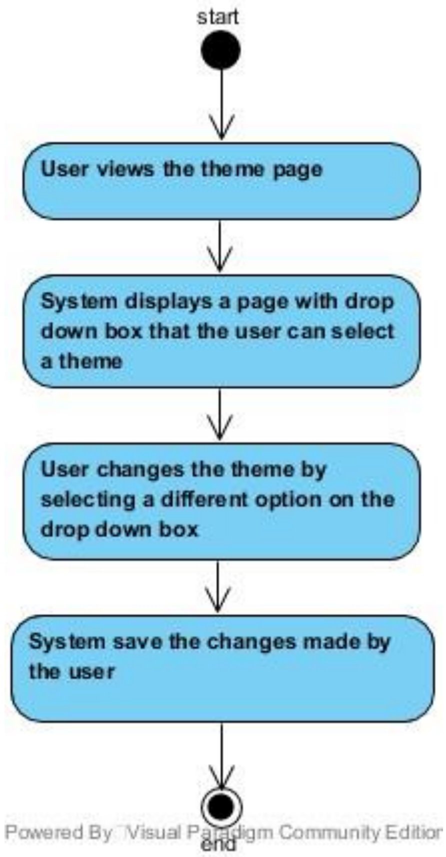

start
User views the theme page
System displays a page with drop down box that the user can select a theme
User changes the theme by selecting a different option on the drop down box
System save the changes made by the user
end
Powered By Visual Paradigm Community Edition

Figure 3-23 Activity Diagram for Theme

The user can go to the theme page and update the theme to change the appearance of the application. The user can click on the dropdown box the select the theme, the theme styles will be applied as a preview of the theme before the user actually saves the theme setting. The user will need to click on the save button if the user wants to save the theme setting.

3.2.23 Update Password

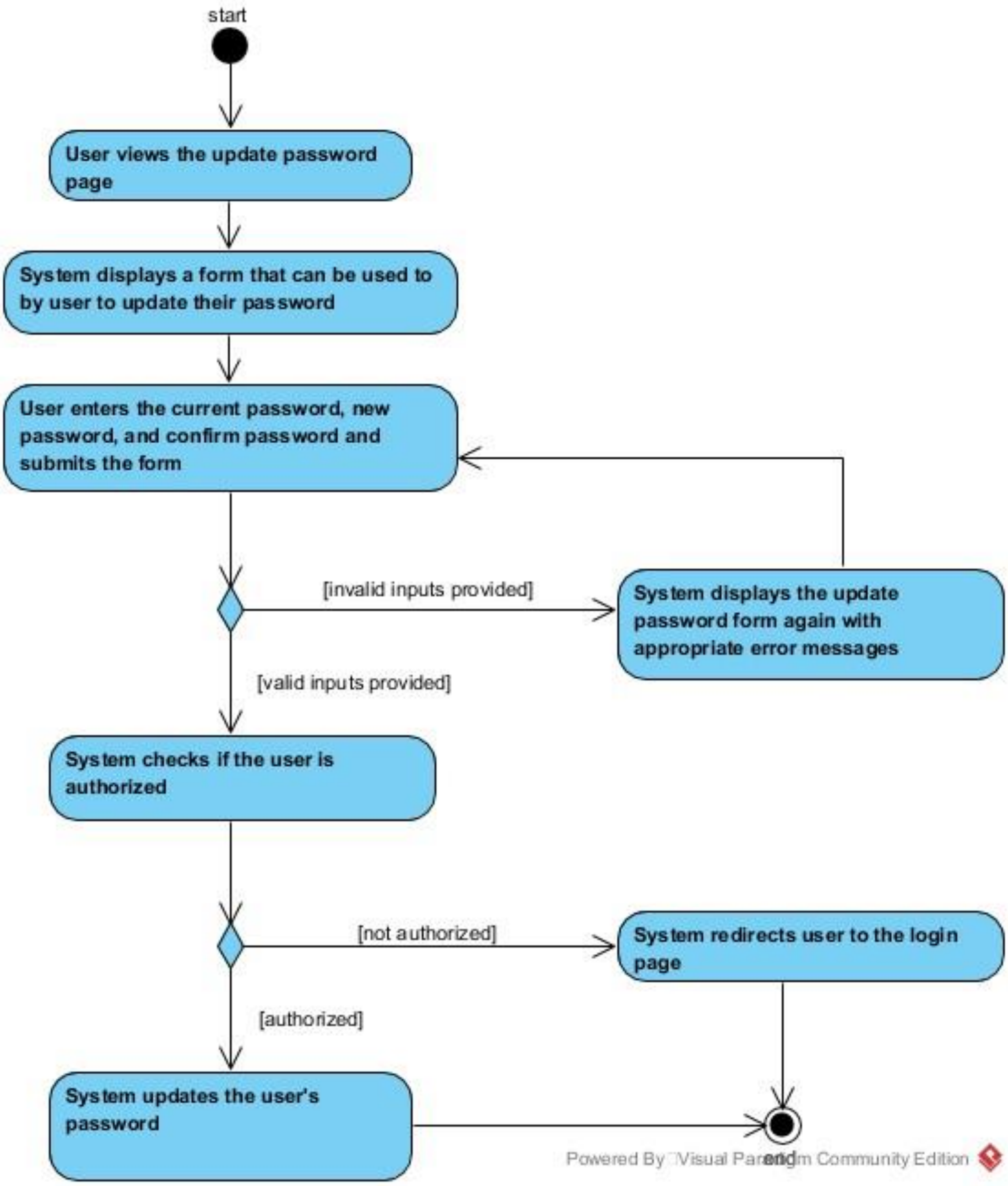

start
User views the update password page
System displays a form that can be used to by user to update their password
User enters the current password, new password, and confirm password and submits the form
[invalid inputs provided]
System displays the update password form again with appropriate error messages
[valid inputs provided]
System checks if the user is authorized
[not authorized]
System redirects user to the login page
[authorized]
System updates the user's password
end
Powered By Visual Paradigm Community Edition

Figure 3-24 Activity Diagram for Update Password

The user can update the account’s password by providing the existing password, new password, and confirm password. If the inputs provided were valid, the system will update the user’s password.

3.2.24 Enable 2-Factor Authentication

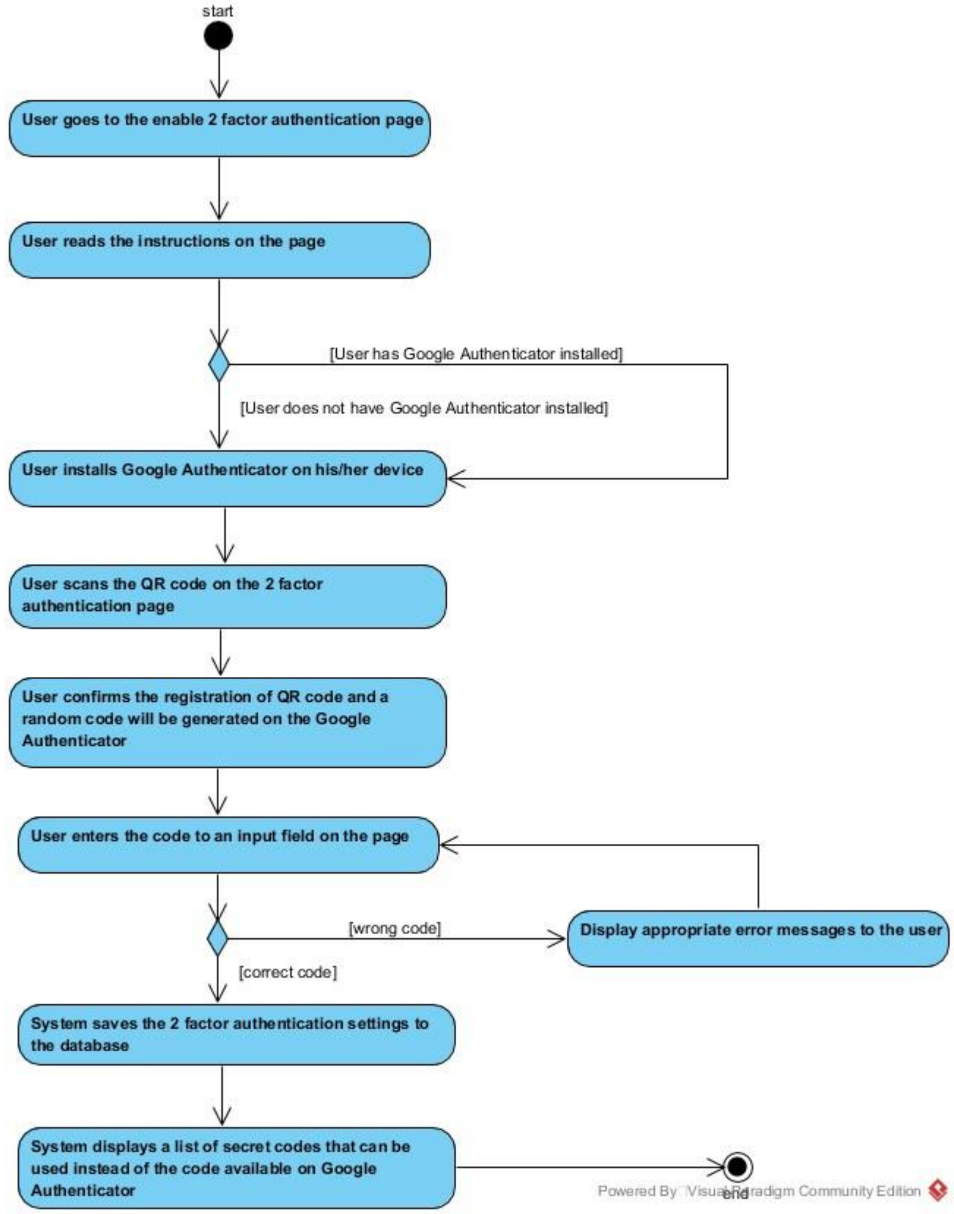

start
User goes to the enable 2 factor authentication page
User reads the instructions on the page
[User has Google Authenticator installed]
[User does not have Google Authenticator installed]
User installs Google Authenticator on his/her device
User scans the QR code on the 2 factor authentication page
User confirms the registration of QR code and a random code will be generated on the Google Authenticator
User enters the code to an input field on the page
[wrong code]
Display appropriate error messages to the user
[correct code]
System saves the 2 factor authentication settings to the database
System displays a list of secret codes that can be used instead of the code available on Google Authenticator
end
Powered By Visual Paradigm Community Edition

Figure 3-25 Activity Diagram for Enable 2-Factor Authentication

The user can enable 2-factor authentication by going to the enable 2-factor authentication page. The users will need to have Google Authenticator installed on their device, then use the Google Authenticator app to scan the QR Code on the page. After that, the user will need to enter a number that is available on the Google Authenticator App. If the number provided was correct, the system will save the 2-factor authentication settings and provide a list of secret codes that can be used as an alternate way to login to the system while having 2-factor authentication enabled.

3.2.25 Report

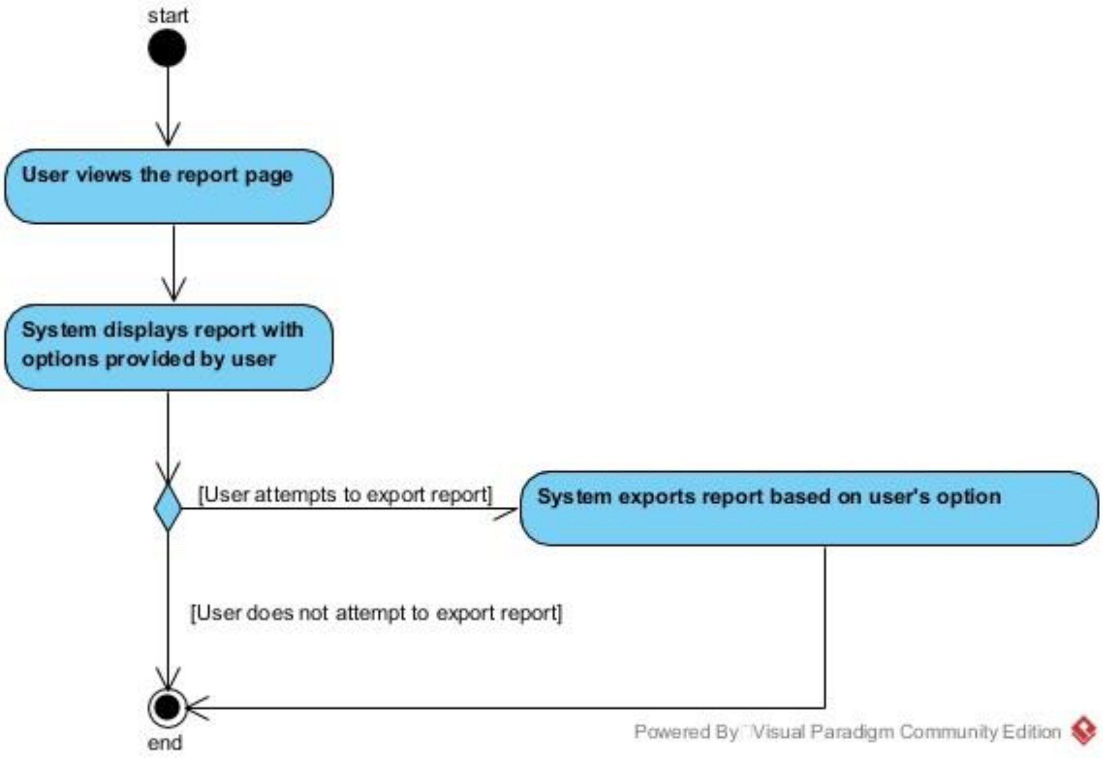

start
User views the report page
System displays report with options provided by user
[User attempts to export report]
System exports report based on user's option
[User does not attempt to export report]
end
Powered By Visual Paradigm Community Edition

Figure 3-26 Activity Diagram for Report

The user can view different reports based on the inputs provided using the filter. After viewing the report, the user can export the report to either third party cloud storages including Google Drive and Dropbox, download the report, or preview the report in PDF format.

3.3 Database Diagram – Object-Oriented Data Model

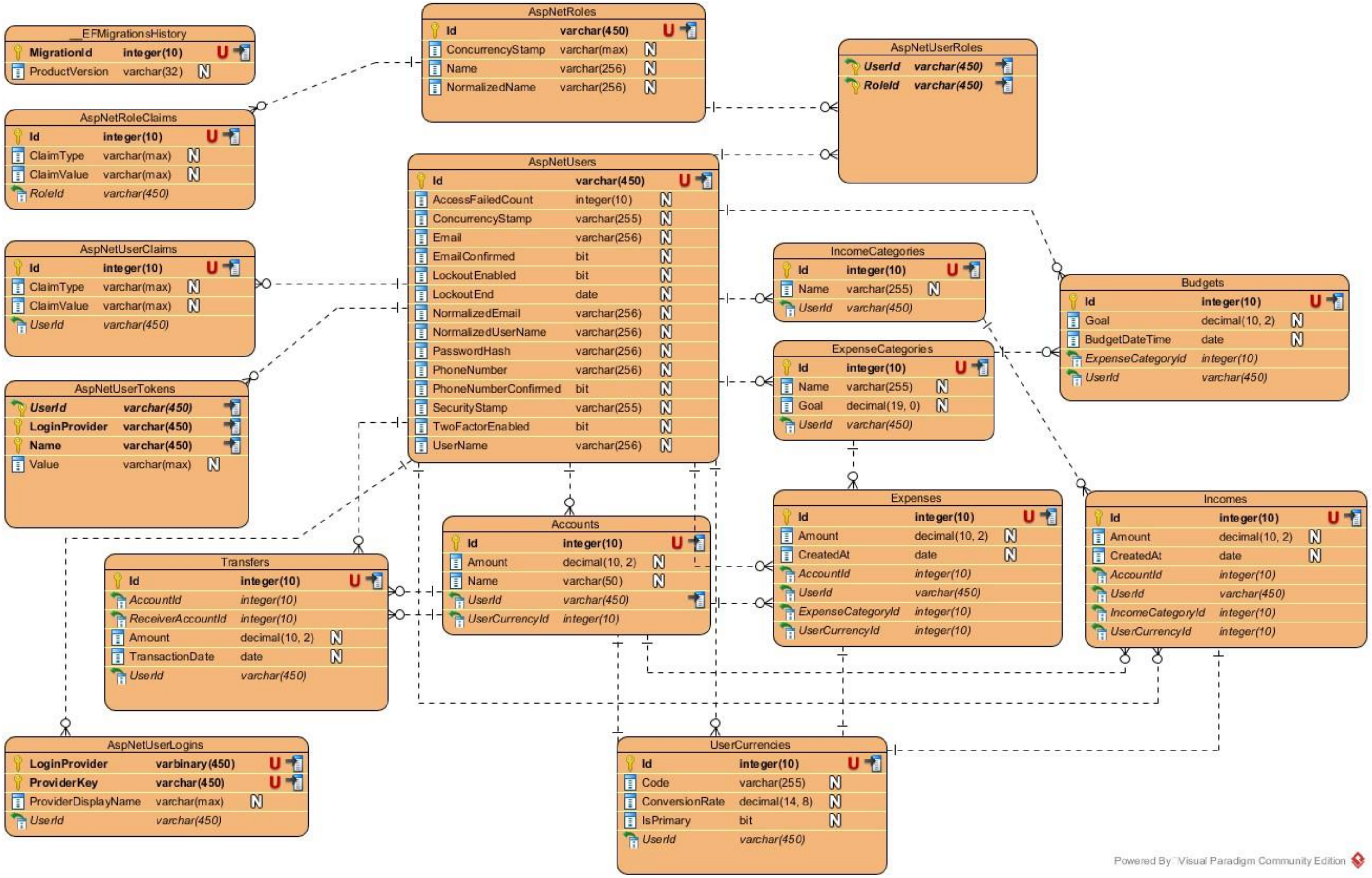
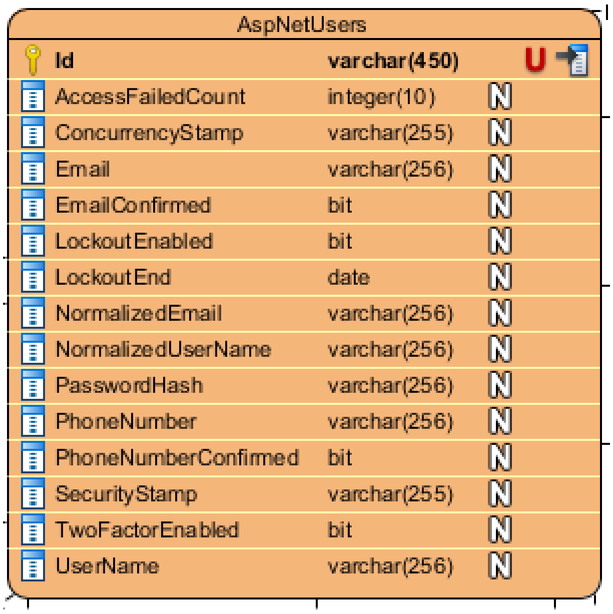


Figure 3-27 Entity Relationship Diagram

3.3.1 AspNetUsers

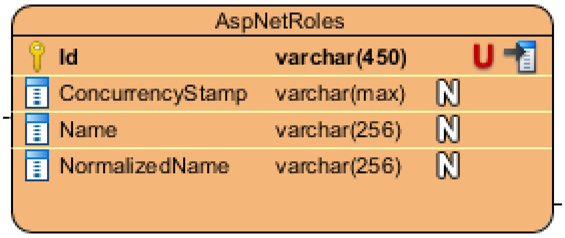


AspNetUsers		
Id	varchar(450)	U
AccessFailedCount	integer(10)	N
ConcurrencyStamp	varchar(255)	N
Email	varchar(256)	N
EmailConfirmed	bit	N
LockoutEnabled	bit	N
LockoutEnd	date	N
NormalizedEmail	varchar(256)	N
NormalizedUserName	varchar(256)	N
PasswordHash	varchar(256)	N
PhoneNumber	varchar(256)	N
PhoneNumberConfirmed	bit	N
SecurityStamp	varchar(255)	N
TwoFactorEnabled	bit	N
UserName	varchar(256)	N

Figure 3-28 AspNetUsers Table

This table represents the users of the system. The primary key of this table is Id. The fields that will be used frequently by the system are Id, Email, EmailConfirmed, PasswordHash, TwoFactorEnabled, and Username. Other fields of this table may not be used throughout the lifetime of the system. These fields are included because this table is generated by the ASP.NET framework and removing these fields might cause potential issues to the system.

3.3.2 AspNetRoles

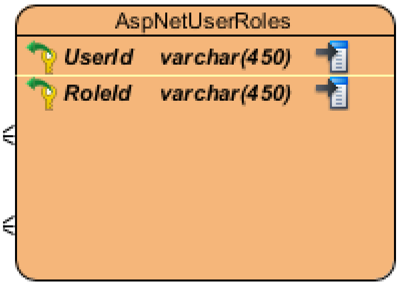


AspNetRoles		
Id	varchar(450)	U
ConcurrencyStamp	varchar(max)	N
Name	varchar(256)	N
NormalizedName	varchar(256)	N

Figure 3-29 AspNetRoles Table

This table represents the roles of the system. The primary key of this table is Id. This table will later be used with AspNetUserRoles to associate user with roles.

3.3.3 AspNetUserRoles



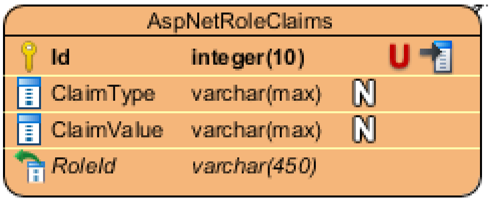
AspNetUserRoles

UserId	varchar(450)
RoleId	varchar(450)

Figure 3-30 AspNetUserRoles Table

This table is used to associate user with roles. Both UserId and RoleId are the primary and foreign key of this table.

3.3.4 AspNetRoleClaims



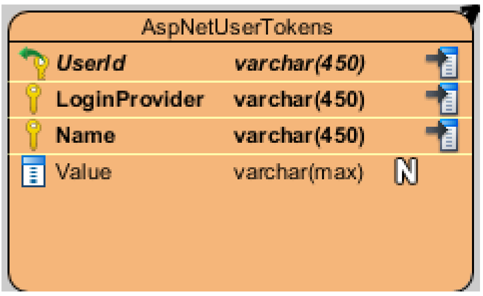
AspNetRoleClaims

Id	integer(10)	U
ClaimType	varchar(max)	N
ClaimValue	varchar(max)	N
RoleId	varchar(450)	

Figure 3-31 AspNetRoleClaims Table

This table was created to enable claim-based authorization. The primary key of this table is Id. This table was not used throughout the application. The RoleId is a foreign key associated with the Id of the AspNetRoles table.

3.3.5 AspNetUserTokens



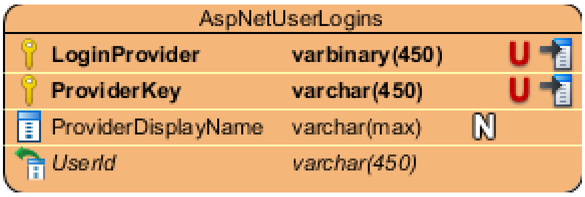
AspNetUserTokens

UserId	varchar(450)	
LoginProvider	varchar(450)	
Name	varchar(450)	
Value	varchar(max)	N

Figure 3-32 AspNetUserTokens Table

This table was created to store 2-factor authentication credentials. The primary key of this table is composed of UserId, LoginProvider, and Name. The UserId is also a foreign key associated with the Id of the AspNetUsers table.

3.3.6 AspNetUserLogins



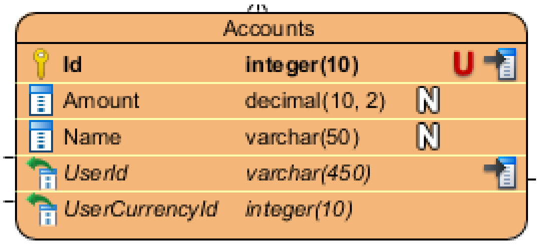
AspNetUserLogins

LoginProvider	**varbinary(450)**	U
ProviderKey	**varchar(450)**	U
ProviderDisplayName	varchar(max)	N
UserId	*varchar(450)*	

Figure 3-33 AspNetUserLogins Table

This table was created to store third-party provider credentials if the user signs up his or her account using a third-party provider such as Google. The primary key of this table is composed of LoginProvider and ProviderKey. The UserId is a foreign key associated with the Id of the AspNetUsers table.

3.3.7 Accounts



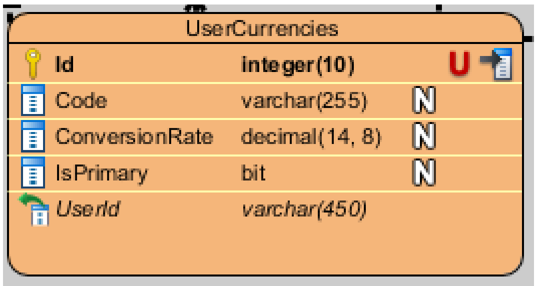
Accounts

Id	**integer(10)**	U
Amount	decimal(10, 2)	N
Name	varchar(50)	N
UserId	*varchar(450)*	
UserCurrencyId	*integer(10)*	

Figure 3-34 Accounts Table

This table was created to represent the accounts of the users. The primary key of this table is Id. The accounts created can later be used by the user to create transactions (expenses, incomes, and transfers). This table has two foreign keys, the first foreign key UserId was used to associate the account with the Id of the AspNetUsers table and the second foreign key UserCurrencyId was used to connect to the Id of the UserCurrencies table.

3.3.8 UserCurrencies

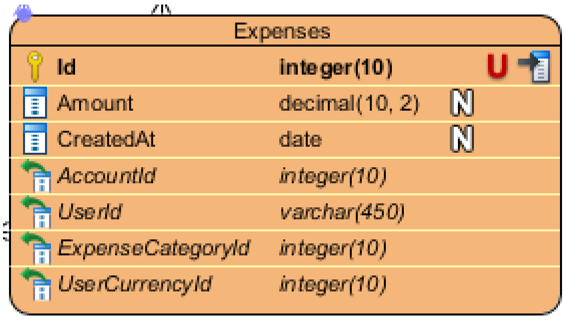


UserCurrencies		
Id	**integer(10)**	U
Code	varchar(255)	N
ConversionRate	decimal(14, 8)	N
IsPrimary	bit	N
UserId	*varchar(450)*	

Figure 3-35 UserCurrencies Table

This table was created to represent the preferred currencies of the user. The primary key of this table is Id. Code represents the currency code of the record, conversion rate indicates the conversion from this currency to the main currency. IsPrimary indicates that whether this is the primary currency of the user. The UserId is a foreign key to associate this table with AspNetUsers table.

3.3.9 Expenses

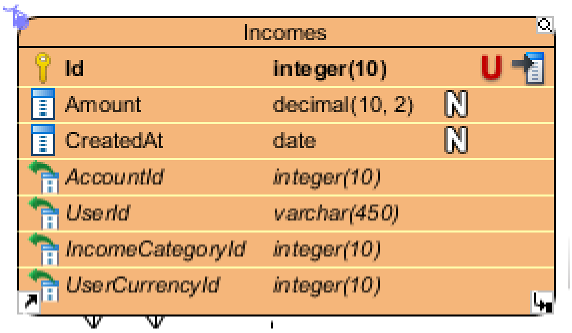


Expenses		
Id	**integer(10)**	U
Amount	decimal(10, 2)	N
CreatedAt	date	N
AccountId	*integer(10)*	
UserId	*varchar(450)*	
ExpenseCategoryId	*integer(10)*	
UserCurrencyId	*integer(10)*	

Figure 3-36 Expenses Table

This table was created to represent the expense records of the users. The primary key of this table is Id. Amount indicates the amount for the expense, and CreatedAt indicates when the expense was created. AccountId is a foreign key to the Accounts table, this field indicates the account that was used to create this expense. UserId is a foreign key associated with the AspNetUsers table. ExpenseCategoryId is a foreign key that was used to associate this table with the ExpenseCategories table. The UserCurrencyId is a foreign key that is used to associate with the UserCurrencies table.

3.3.10 Incomes



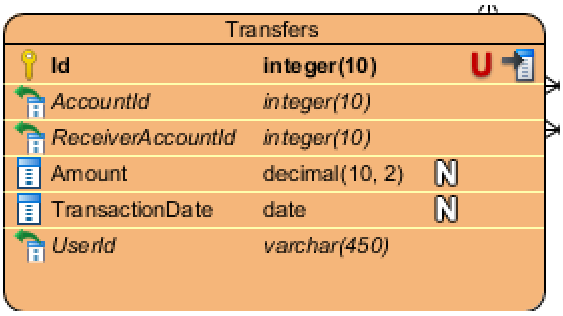
Incomes

Id	integer(10)	U
Amount	decimal(10, 2)	N
CreatedAt	date	N
AccountId	*integer(10)*	
UserId	*varchar(450)*	
IncomeCategoryId	*integer(10)*	
UserCurrencyId	*integer(10)*	

Figure 3-37 Incomes Table

This table was created to represent the income records of the users. The primary key of this table is Id. Amount indicates the amount for the income, and CreatedAt indicates when the income was created. AccountId is a foreign key to the Accounts table, this field indicates the account that was used to create this income. UserId is a foreign key associated with the AspNetUsers table. IncomeCategoryId is a foreign key that was used to associate this table with the IncomeCategories table. The UserCurrencyId is a foreign key that is used to associate with the UserCurrencies table.

3.3.11 Transfers



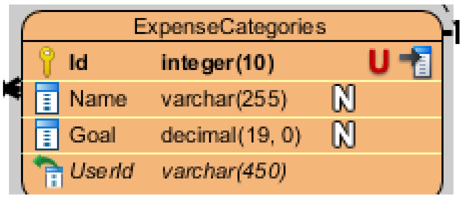
Transfers

Id	integer(10)	U
AccountId	*integer(10)*	
ReceiverAccountId	*integer(10)*	
Amount	decimal(10, 2)	N
TransactionDate	date	N
UserId	*varchar(450)*	

Figure 3-38 Transfers Table

This table was created to represent the transfer records of the users. The primary key of this table is Id. Amount indicates the amount to transfer from the selected account to the receiver account. TransactionDate indicates the date that the transaction occurs. AccountId represent the account id of the account to transfer from while ReceiverAccountId represents the account id of the account that will receive the transfer. UserId is a foreign key that will connect with the user table.

3.3.12 ExpenseCategories



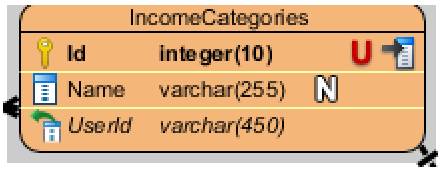
ExpenseCategories

Id	integer(10)	U
Name	varchar(255)	N
Goal	decimal(19, 0)	N
UserId	*varchar(450)*	

Figure 3-39 ExpenseCategories Table

This table was created to represent the expense categories of the users. Expense categories will be used when the user attempts to create an expense record. The primary key of this table is Id. Name represents the name of the expense category while Goal indicates the default monthly budget for the category. If Goal was not specified, it is considered a budget was not set for this category. UserId is a foreign key connected to the Id of the user table.

3.3.13 IncomeCategories



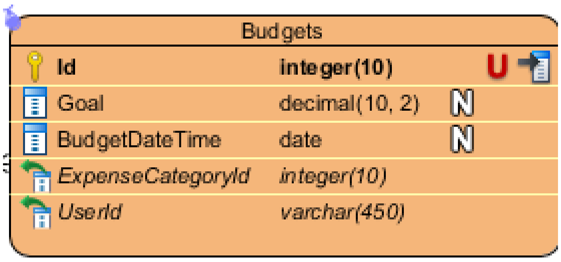
IncomeCategories

Id	integer(10)	U
Name	varchar(255)	N
UserId	*varchar(450)*	

Figure 3-40 IncomeCategories Table

This table was created to represent the income categories of the users. Income categories will be used when the user attempts to create an income record. The primary key of this table is Id. Name represents the name of the income category. UserId is a foreign key connected to the Id of the user table.

3.3.14 Budgets



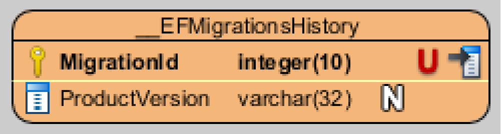
Budgets

Id	integer(10)	U
Goal	decimal(10, 2)	N
BudgetDateTime	date	N
ExpenseCategoryId	*integer(10)*	
UserId	*varchar(450)*	

Figure 3-41 Budgets Table

This table was created to represent the budgets of users. Budget is connected to the ExpenseCategories table to indicate the budget for the connected category of a specific month. The BudgetDateTime field is used to store the month the budget was specified while the Goal field represent the budget for the category. UserId is a foreign key connected to the Id of the user table.

3.3.15 _EFMigrationsHistory



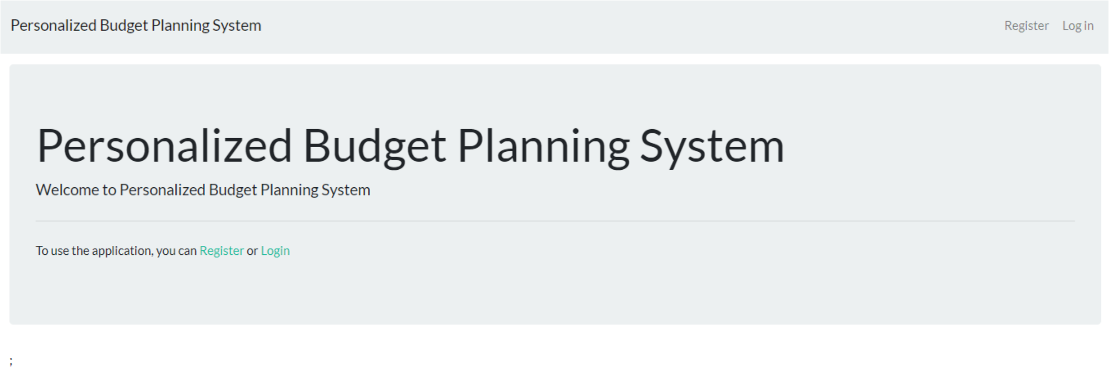
__EFMigrationsHistory		
MigrationId	**integer(10)**	U
ProductVersion	varchar(32)	N

Figure 3-42 _EFMigrationsHistory Table

This table was created to store the migration histories during the development of the application. This table is not directly related to the application itself, instead it is a table that is used for development purpose as Entity Framework was used.

3.4 User Interface Design

3.4.1 Home



Personalized Budget Planning System

Personalized Budget Planning System

Welcome to Personalized Budget Planning System

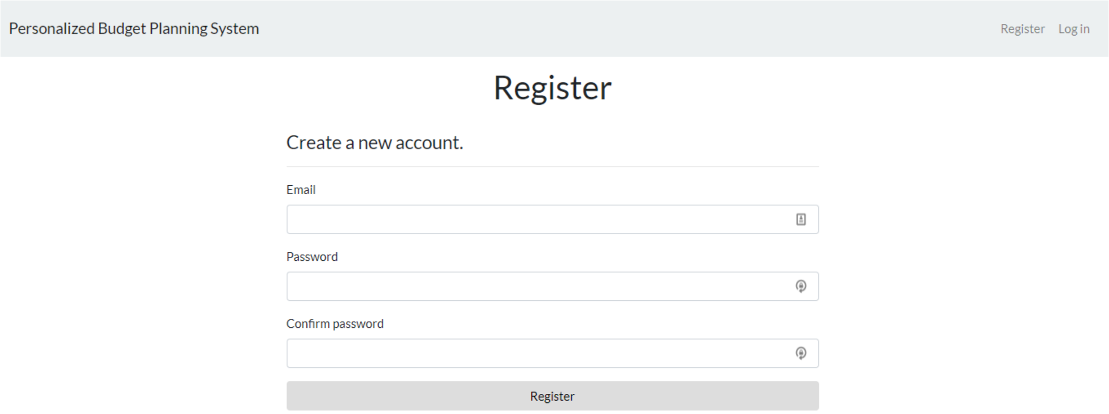
To use the application, you can Register or Login

;

Figure 3-43 Home Page

This is the home page of the application. The user can click on the register link to register an account or the login link to login to the system.

3.4.2 Register



Personalized Budget Planning System

Register

Create a new account.

Email

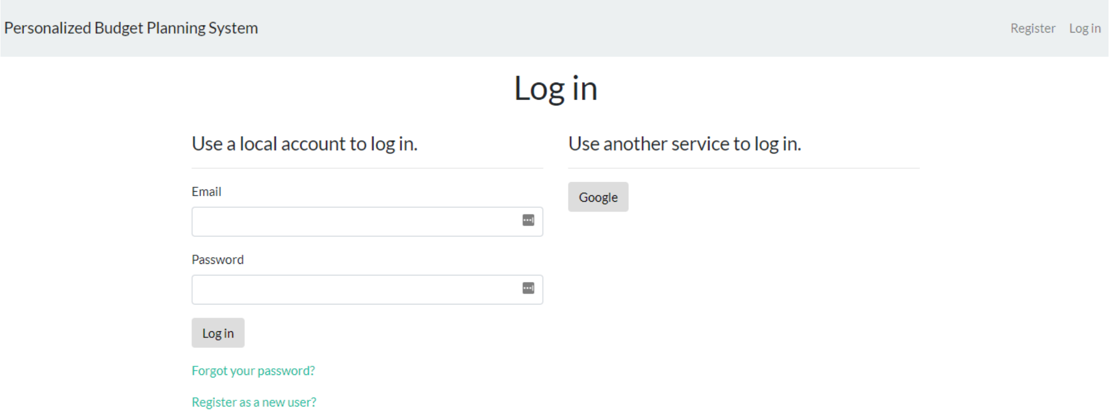
Password

Confirm password

Figure 3-44 Register Page

The user can register an account by providing the necessary credentials which are email, password, and confirm password. After the user successfully registered an account. A confirmation email will be sent to the user, the user can then click on the link provided in the confirmation email to verify his or her email. After that, the user can then login to the system with the email and password provided earlier.

3.4.3 Login



Personalized Budget Planning System

Log in

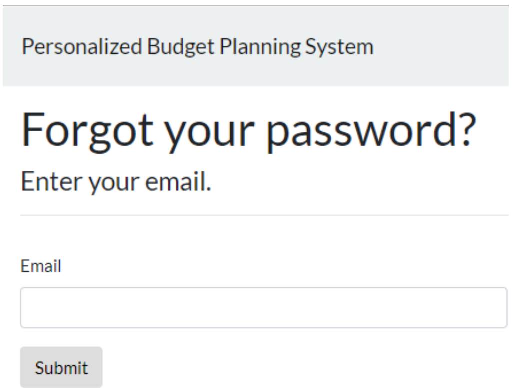
Use a local account to log in.

Use another service to log in.

Figure 3-45 Login Page

This is the login page of the system. The user can either login using their email and password combination or by using their Google Plus identity. If the user decided to use their Google Plus identity for the first time, it is assumed that the user attempts to register an account using their Google Plus identity and no confirmation email will be sent.

3.4.4 Forgot Password

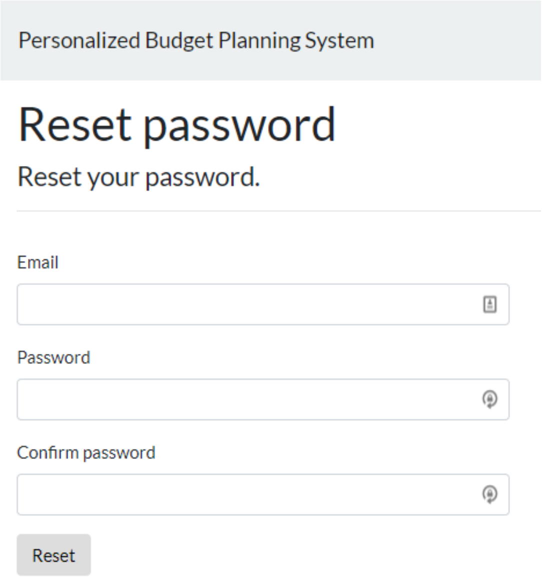


Personalized Budget Planning System

Forgot your password?

Enter your email.

Figure 3-46 Forgot Password Page



Personalized Budget Planning System

Reset password

Reset your password.

Email

Password

Confirm password

Figure 3-47 Reset Password Page

The users can enter their email to reset their password. After clicking the submit button, an email will be sent to the user’s email and user can then click on the link to reset their password.

3.4.5 General components

These are the components that are used frequently. Some components might exist at all time regardless of which page the user is viewing. For example, the navbar will exist regardless of whether the user is viewing the dashboard page or the expenses page.

Navbar

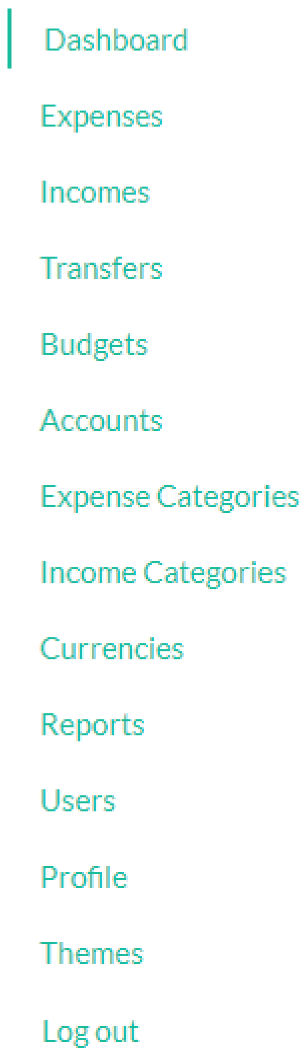


Figure 3-48 Navigation Bar

This component can be used to navigate around the application. The left border of the link indicates the page that the user is viewing. For example, if the user is viewing the dashboard page, the left border of the dashboard link will appear.

Header

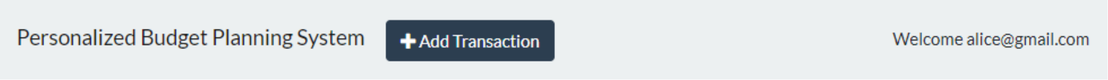
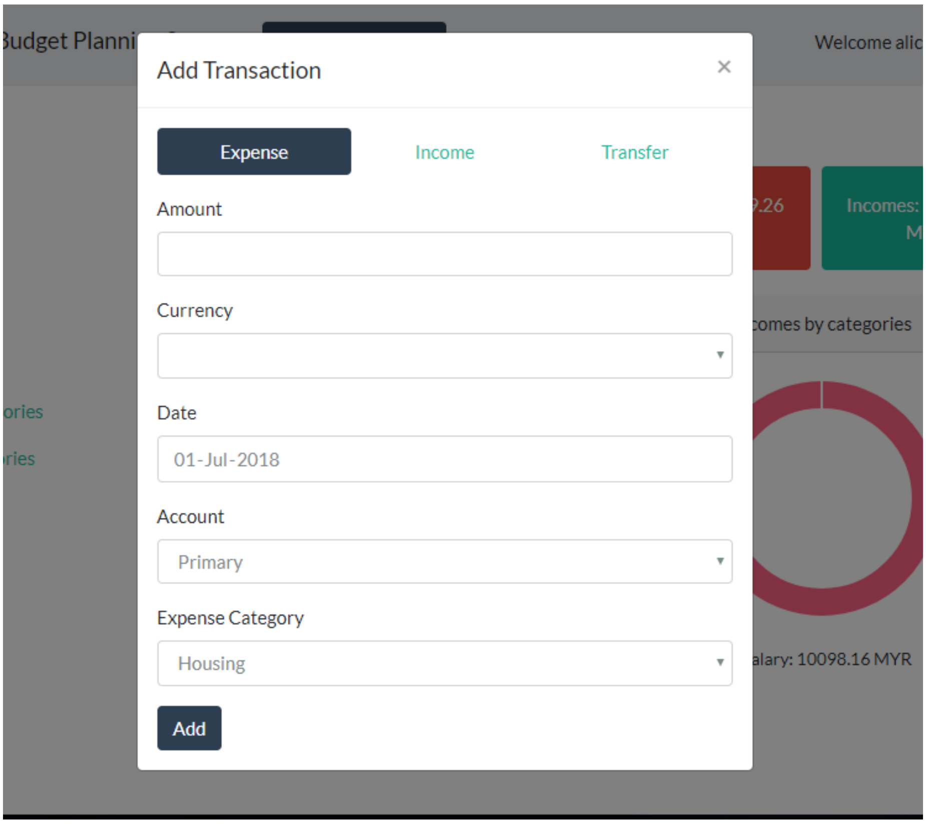


Figure 3-49 Header

Header displays the application name, a button that will open the add transaction modal, and the user’s email.

Add Transaction Modal

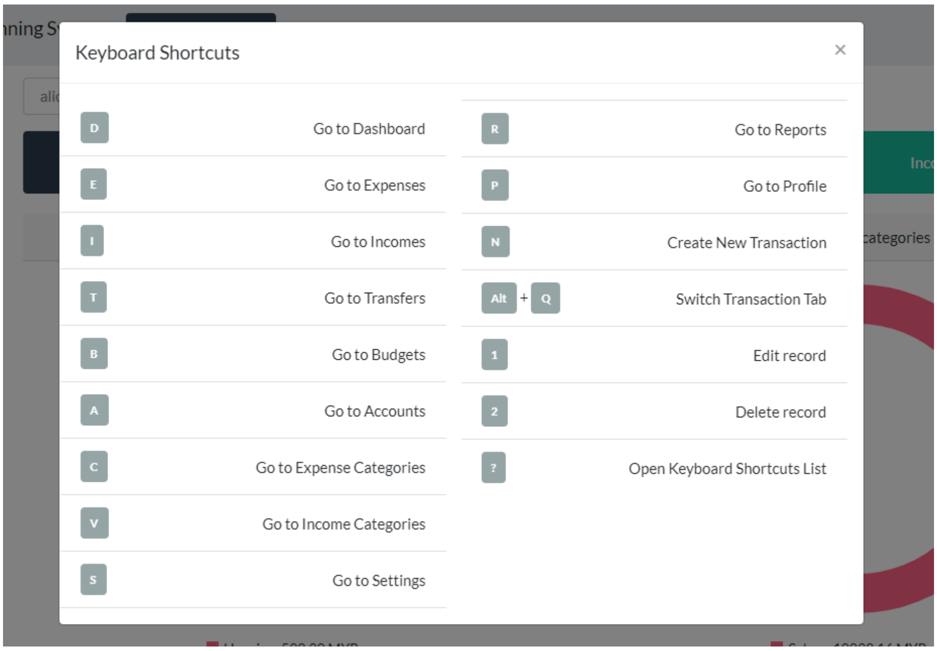


Date: 01-Jul-2018
Account: Primary
Expense Category: Housing

Figure 3-50 Add Transaction Modal

This modal can be used to create a transaction, the user can switch between the type of transaction to create by clicking on different tabs. The transaction modal can also be opened by hitting the ‘N’ key. The user can also change between different tabs with ‘ALT + Q’ key.

Keyboard Shortcuts Modal



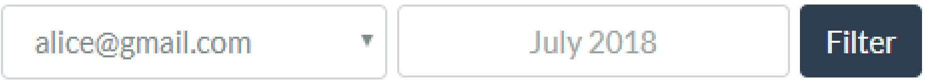
Keyboard Shortcuts ×

D	Go to Dashboard	R	Go to Reports
E	Go to Expenses	P	Go to Profile
I	Go to Incomes	N	Create New Transaction
T	Go to Transfers	Alt + Q	Switch Transaction Tab
B	Go to Budgets	1	Edit record
A	Go to Accounts	2	Delete record
C	Go to Expense Categories	?	Open Keyboard Shortcuts List
V	Go to Income Categories		
S	Go to Settings		

Figure 3-51 Keyboard Shortcuts Modal

The keyboard shortcuts modal displays a list of available shortcuts in the application. This modal can be triggered by pressing the '?' key.

Filter

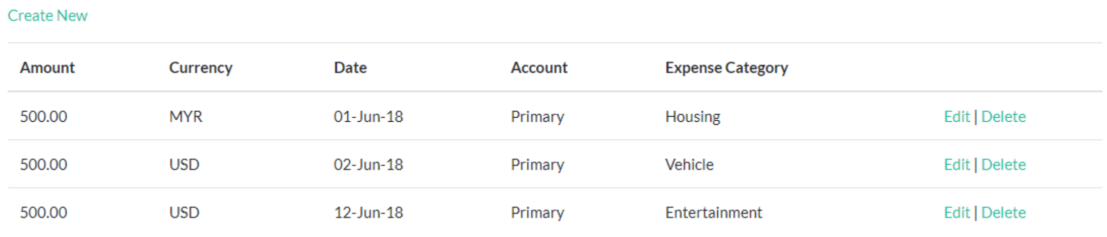


alice@gmail.com ▾ July 2018 Filter

Figure 3-52 Filters

The filter component can be used to change the data based on the criteria provided. Filters will not exist on every page since not every page need a filter. Filter on different pages might have different fields since not every page will be using the same criteria for filtering. For example, the filter component on the expenses page will be different from the report page.

List of records



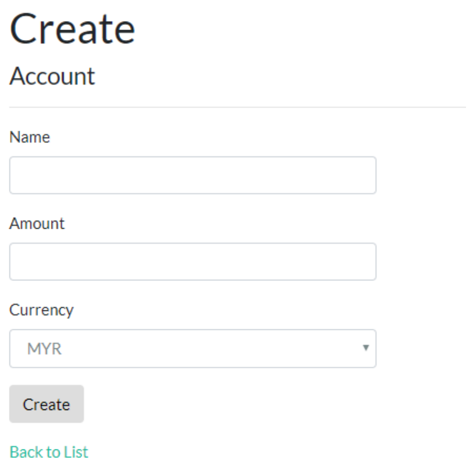
Create New

Amount	Currency	Date	Account	Expense Category	
500.00	MYR	01-Jun-18	Primary	Housing	Edit | Delete
500.00	USD	02-Jun-18	Primary	Vehicle	Edit | Delete
500.00	USD	12-Jun-18	Primary	Entertainment	Edit | Delete

Figure 3-53 List of Records

A table is used to display list of records for expenses, incomes, transfers, accounts, expense categories, income categories, currencies, and users. The table will have different columns depending on the records that are being displayed. For example, if the table was used to display the expense records, the columns of the table will include amount, currency, date, account, and expense category. The user can click on the 'Create New' link to create a new record, the 'Edit' link to edit the record, or the 'Delete' link to delete the record. The user can also press '1' on the keyboard to edit the record and '2' on the keyboard to delete the record.

Create Page



Create

Account

Name

Amount

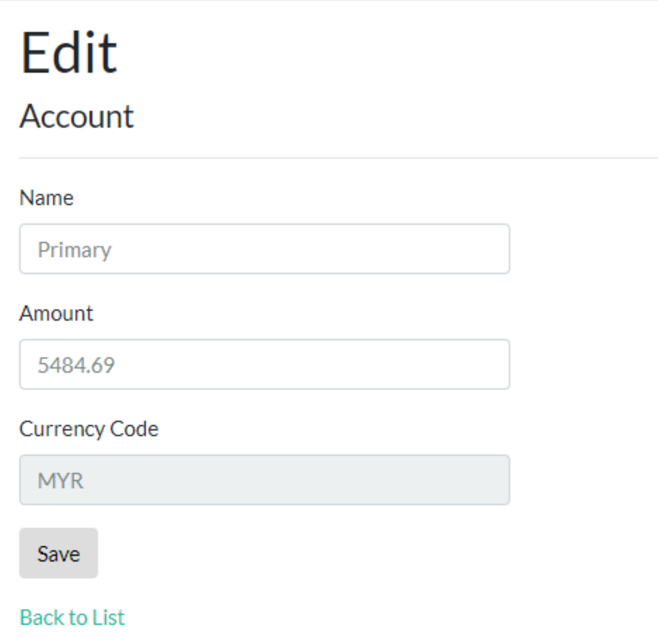
Currency

MYR

Figure 3-54 Create Page

A create page contains a form that is used to create a record. The create page will contain different fields based on the properties of the record. For example, if the user wants to create an account record, the user will need to supply the name, the amount, and the currency of the account.

Edit Page



Edit

Account

Name: Primary

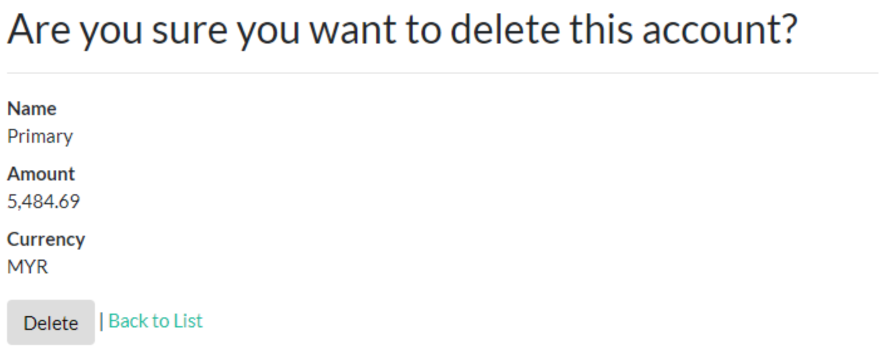
Amount: 5484.69

Currency Code: MYR

Figure 3-55 Edit Page

The edit page contains a form that is used to edit an existing record. The form will be populated with the existing record data, then the user can modify the record as necessary and click on the ‘Save’ button to save the changes. The edit form will contain different fields depending on the record that was modified. For example, if the user wants to edit an account record, the name, amount, and currency code will be populated.

Delete Page



Are you sure you want to delete this account?

Name: Primary

Amount: 5,484.69

Currency: MYR

Figure 3-56 Delete Page

The delete page is a delete confirmation page in which the user will have to make the final decision on whether to delete the record or not.

3.4.6 Dashboard

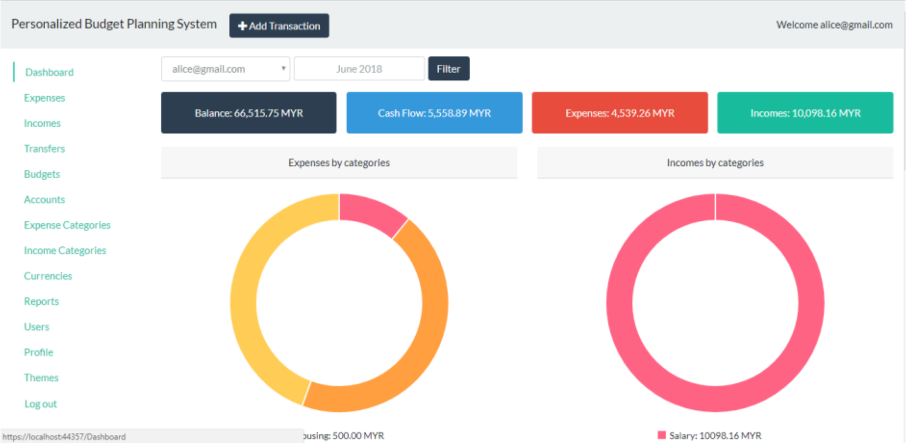


Figure 3-57 Dashboard Part 1

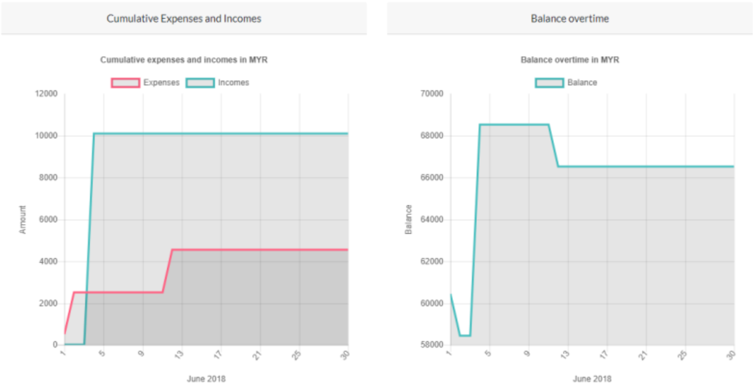
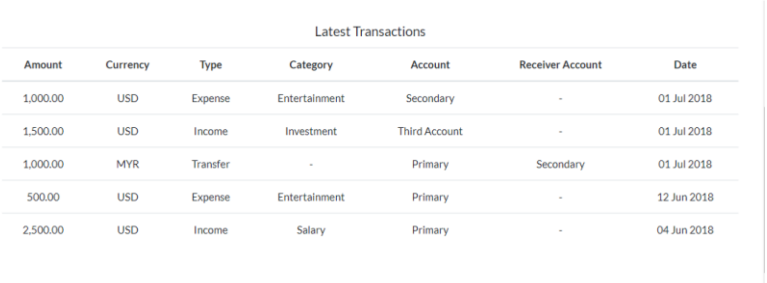


Figure 3-58 Dashboard Part 2



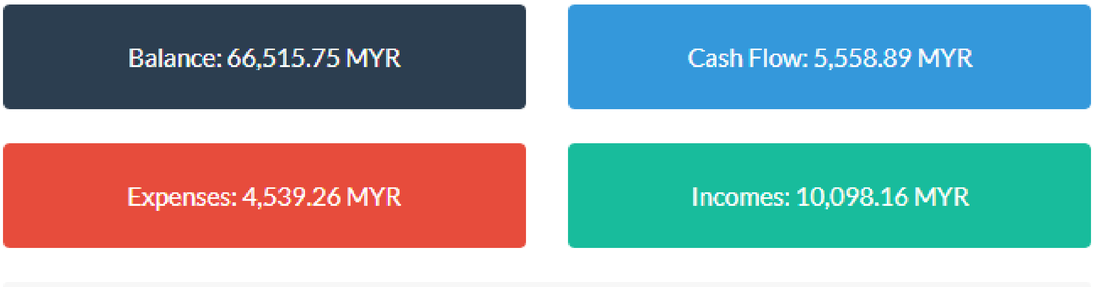
Latest Transactions

Amount	Currency	Type	Category	Account	Receiver Account	Date
1,000.00	USD	Expense	Entertainment	Secondary	-	01 Jul 2018
1,500.00	USD	Income	Investment	Third Account	-	01 Jul 2018
1,000.00	MYR	Transfer	-	Primary	Secondary	01 Jul 2018
500.00	USD	Expense	Entertainment	Primary	-	12 Jun 2018
2,500.00	USD	Income	Salary	Primary	-	04 Jun 2018

Figure 3-59 Dashboard Part 3

The dashboard contains various components and charts to display the general information about the user records. Each component will be described in the following pages.

Overview



Balance: 66,515.75 MYR

Cash Flow: 5,558.89 MYR

Expenses: 4,539.26 MYR

Incomes: 10,098.16 MYR

Figure 3-60 Overview

Balance	The amount of all account's balance combined. For example, if the user has two accounts, the balance of account A is 50,000 MYR and account B is 10,000 MYR. The combined balance will become 60,000 MYR
Expenses	The total expenses for a specific month that can be determined by the filter. The default month will be this month.
Incomes	The total incomes for a specific month that can be determined by the filter. The default month will be this month.
Cashflow	Cashflow represent expenses minus incomes.

Table 3-1 Overview Explanation

Expenses Structure

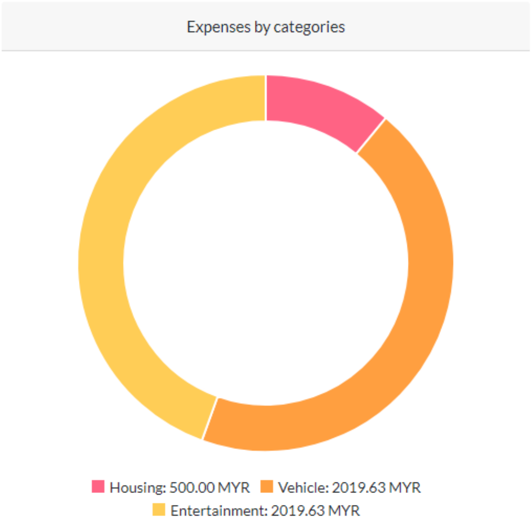


Figure 3-61 Expenses Structure

This is a pie chart that represents the user expenses structure. Each expense category will occupy a piece of the pie chart.

Incomes Structure

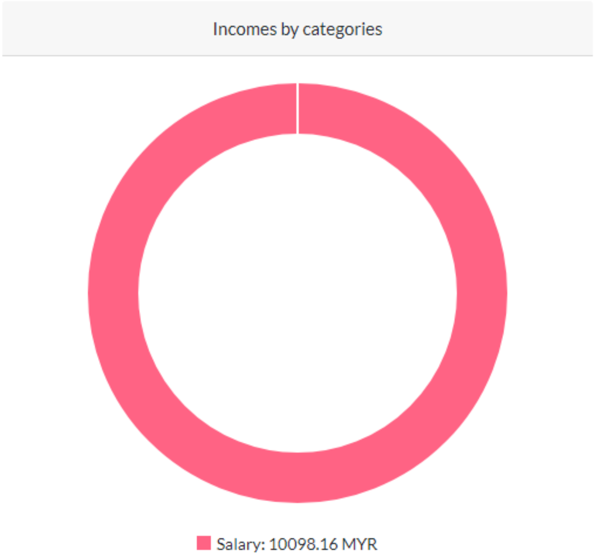


Figure 3-62 Incomes Structure

This is a pie chart that represents the user incomes structure. Each income category will occupy a piece of the pie chart.

Cumulative Expenses and Incomes

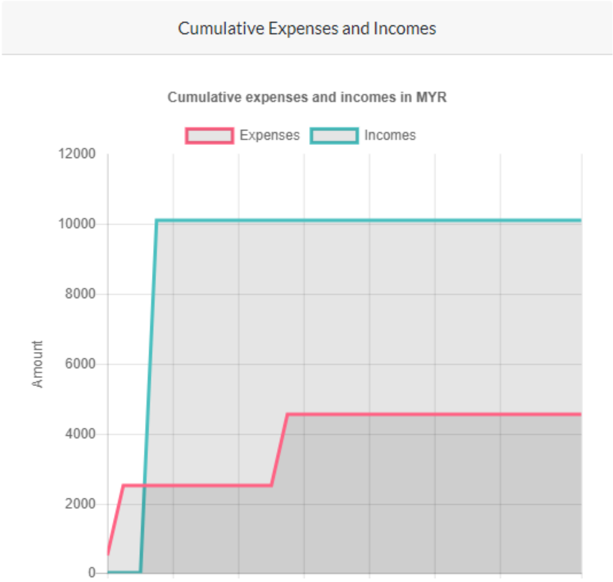


Figure 3-63 Cumulative Expenses and Incomes

This chart shows the cumulative expenses and incomes of the user from the start of the specified until the end of the specified month.

Balance Overtime

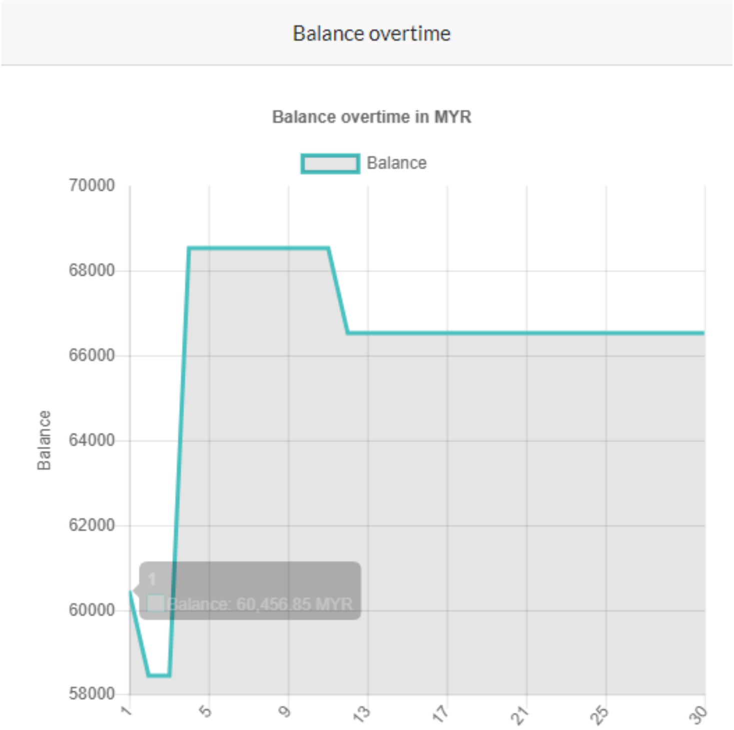
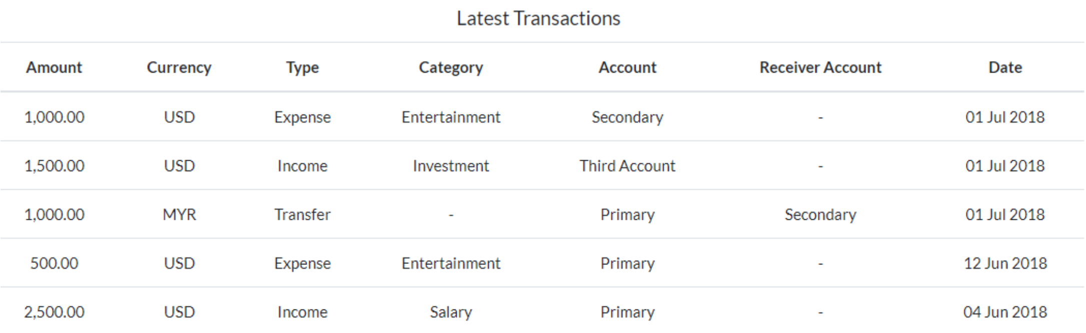


Figure 3-64 Balance Overtime

This chart displays the balance overtime of the user from the start of the specified month until the end of the specified month.

Latest Transactions



Latest Transactions

Amount	Currency	Type	Category	Account	Receiver Account	Date
1,000.00	USD	Expense	Entertainment	Secondary	-	01 Jul 2018
1,500.00	USD	Income	Investment	Third Account	-	01 Jul 2018
1,000.00	MYR	Transfer	-	Primary	Secondary	01 Jul 2018
500.00	USD	Expense	Entertainment	Primary	-	12 Jun 2018
2,500.00	USD	Income	Salary	Primary	-	04 Jun 2018

Figure 3-65 Latest Transactions

This component displays the last 5 transactions (expenses, incomes, or transfers) of the user.

3.4.7 Budgets

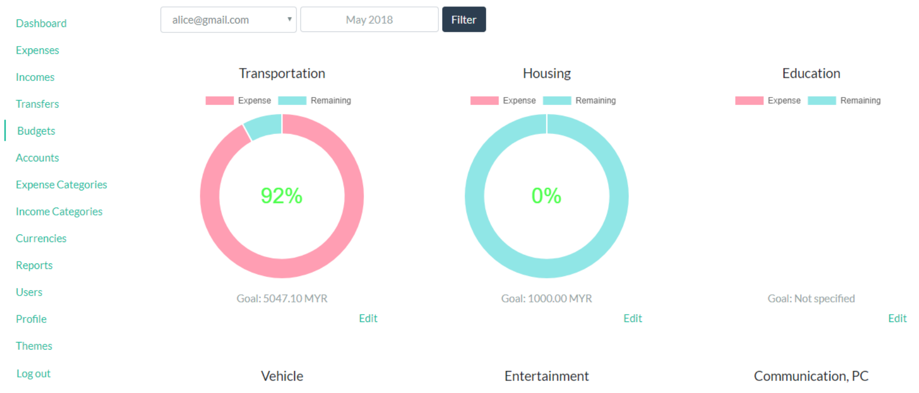
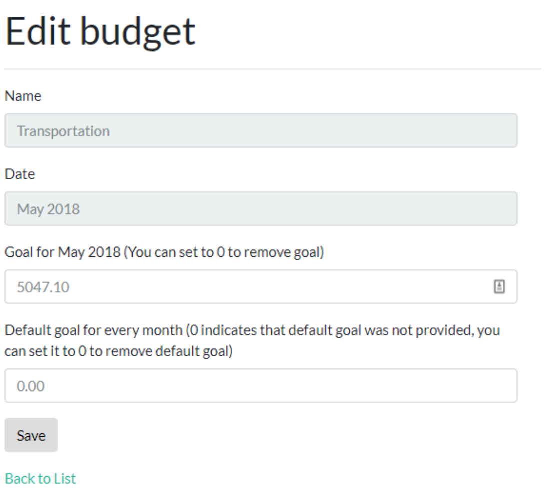

alice@gmail.com
May 2018

Figure 3-66 Budgets Page

The budgets page displays all budgets specified by the user with a list of pie charts. A pie chart will only be displayed if the user actually specified the budget for the specific expense category. Each pie chart contains two pieces, the first piece (red colour) indicates how much the user has spent for that expense category, the second piece (blue colour) indicates the remaining budget for that expense category. The percentage inside the pie charts indicates the how much the user has already spent compare to the budget. For example, if the user set the budget to 10,000 MYR on housing and had already spent 6,000 MYR on housing, then the percentage in the pie chart will become 60%. The user can click on the edit link to edit the budget.


Edit budget

Name

Transportation

Date

May 2018

Goal for May 2018 (You can set to 0 to remove goal)

5047.10

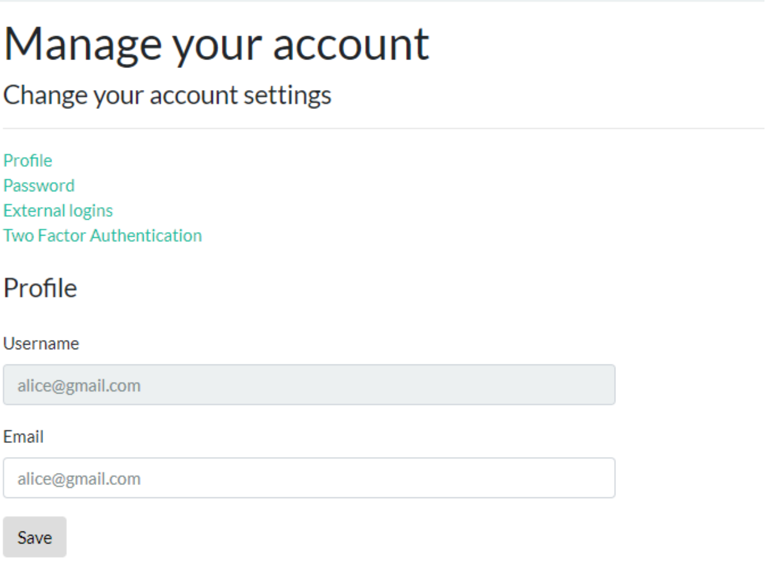
Default goal for every month (0 indicates that default goal was not provided, you can set it to 0 to remove default goal)

0.00

Figure 3-67 Edit Budget Page

The edit budget page allows user to edit the budget of a specific expense category for a specific month or specify a default for every month.

3.4.8 Profile



Manage your account

Change your account settings

Profile

Username: alice@gmail.com

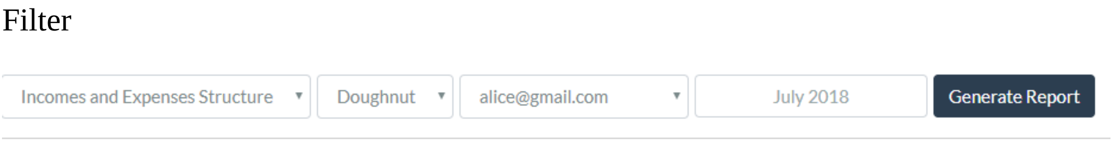
Email: alice@gmail.com

Figure 3-68 Manage Account Page

The profile page is a page that allows user to update their information such as email, password, external logins (e.g. Google), and to setup two factor authentication.

3.4.9 Report

The report page allows user to generate different type of report by specifying the value using the filter. There are 3 types of report. Expenses and Incomes Structure, balance overtime, and cumulative expenses and incomes. The user can also export the report by clicking on the export button. The format of the exported report will be in PDF format.



Filter

Incomes and Expenses Structure ▾ | Doughnut ▾ | alice@gmail.com ▾ | July 2018

Figure 3-69 Filters on Report Page

The filter component on the report contain four fields. The first field indicates the report to generate, the second field indicates the chart to use for the report, the available charts will vary depending on the type of report. The third field is the user's email, only an admin will be able to see and use this field. The fourth field is the month.

Export

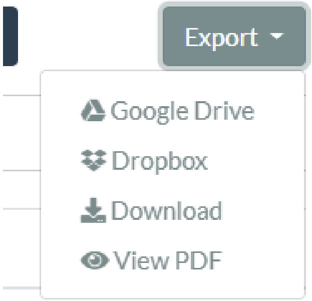
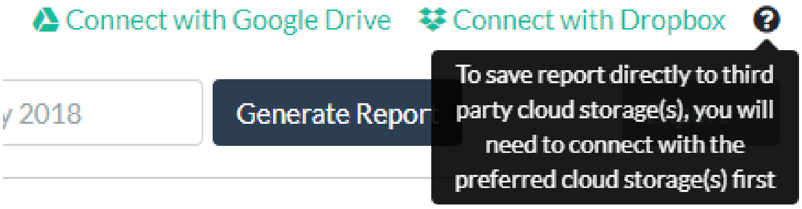


Figure 3-70 Export Buttons

The export button (on the right side of the page) can be used to export the report, the user can either download the report directly, preview the pdf, or export the report to third party cloud storages such as Google Drive and Dropbox. The user will need to authenticate with the preferred cloud storages before they can export their reports to these cloud storages.

Authenticate with Third Party Cloud Storage Links

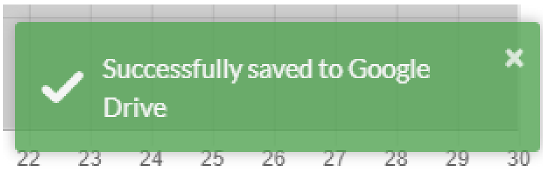


To save report directly to third party cloud storage(s), you will need to connect with the preferred cloud storage(s) first

Figure 3-71 Authentication with third party cloud storages

The user can connect to third party cloud storages by clicking on the links. When the user hover over the question mark icon, additional information will be provided about the links.

Toast Message

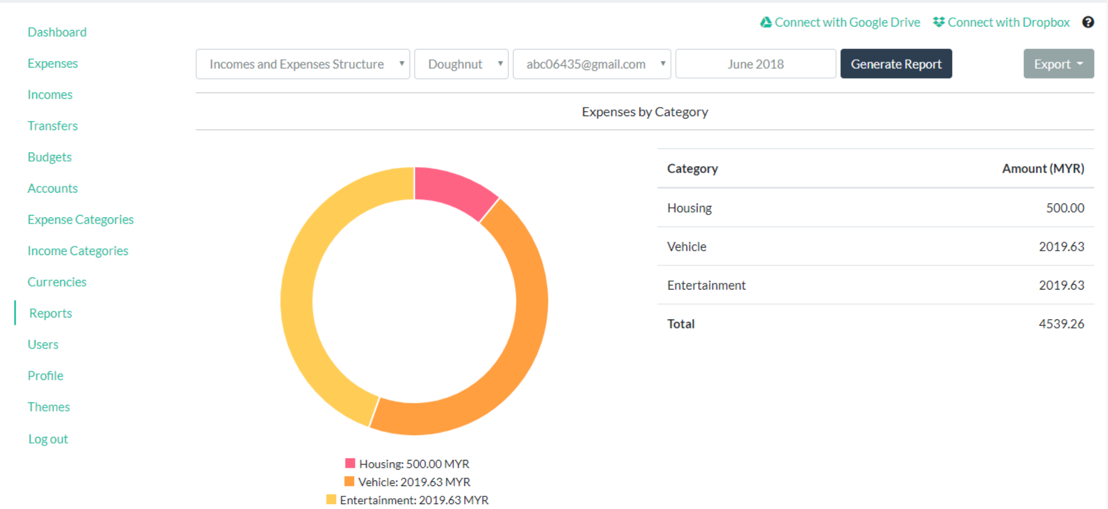


Successfully saved to Google Drive

Figure 3-72 Toastr Sample

A toast message will appear at the bottom right of the page when the user downloads the report, previews the report, or export the report to third party cloud storages. Different toast messages will be used depending on the actions performed.

Expenses and Incomes Structure Report

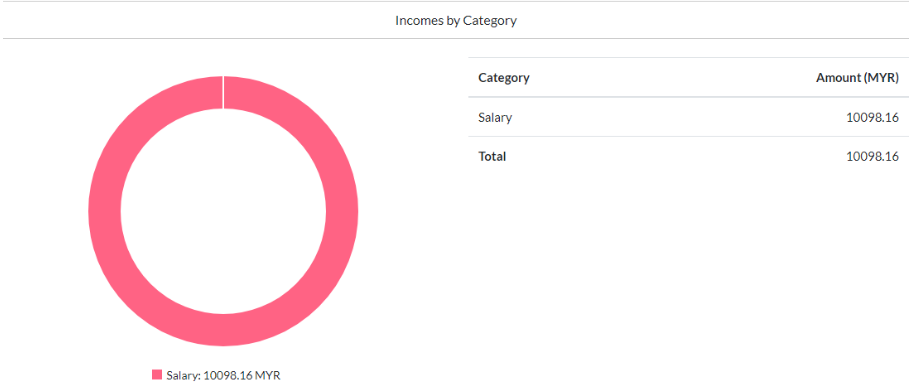


Incomes and Expenses Structure | Doughnut | abc06435@gmail.com | June 2018

Expenses by Category

Category	Amount (MYR)
Housing	500.00
Vehicle	2019.63
Entertainment	2019.63
Total	4539.26

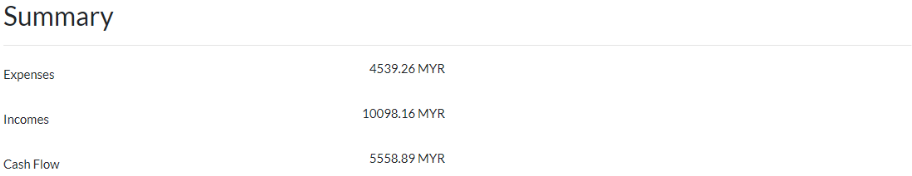
Figure 3-73 Expenses and Incomes Structure Report Part 1



Incomes by Category

Category	Amount (MYR)
Salary	10098.16
Total	10098.16

Figure 3-74 Expenses and Incomes Structure Report Part 2



Summary

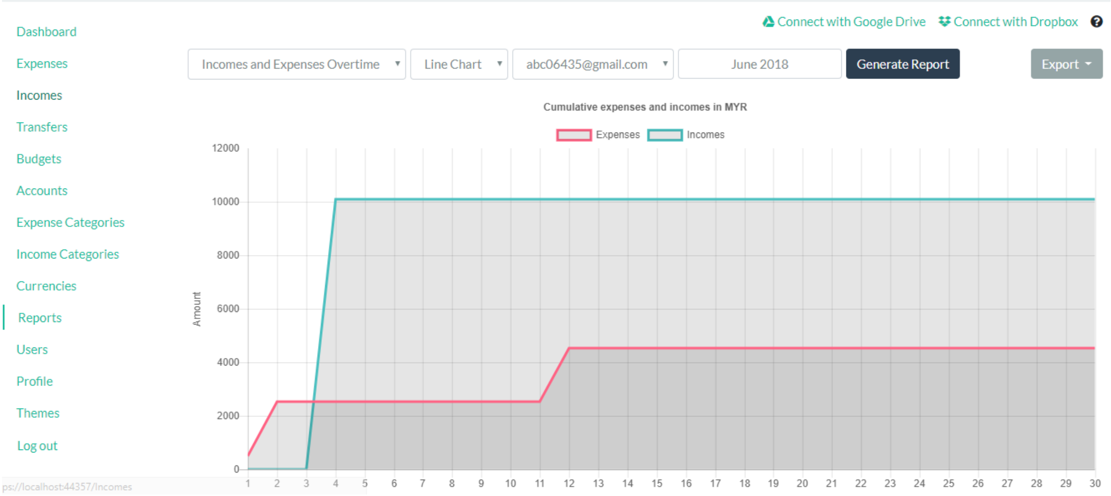
Expenses	4539.26 MYR
Incomes	10098.16 MYR
Cash Flow	5558.89 MYR

Figure 3-75 Expenses and Incomes Structure Report Part 3

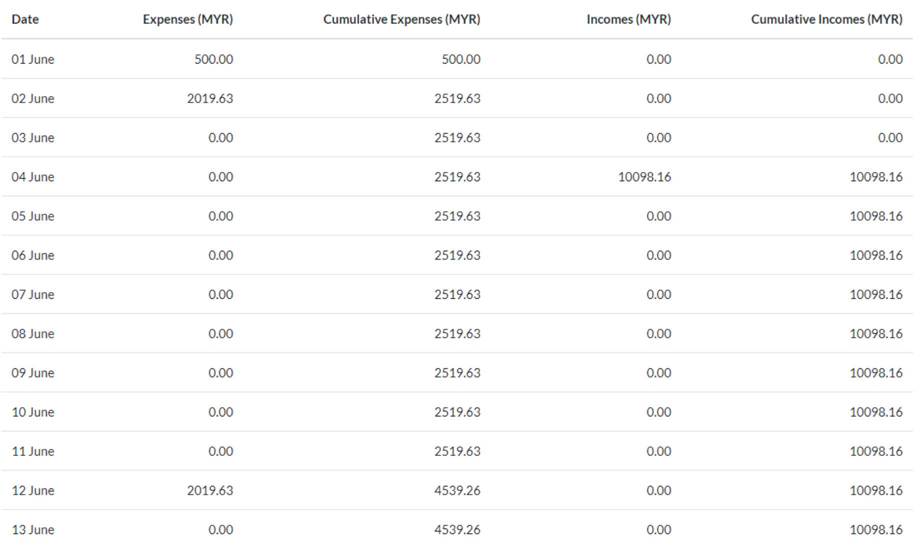
This report displays the expenses and incomes structure of the user. The data is being display in two formats, chart and table. The charts that are available for this report will be doughnut (technically a pie chart) and bar chart. At the end of the report is a summary section that displays the cash flow by using incomes to subtract expenses.

Cumulative Expenses and Incomes Report

Figure 3-76 Cumulative Expenses and Incomes Report Part 1



Incomes and Expenses Overtime | Line Chart | abc06435@gmail.com | June 2018

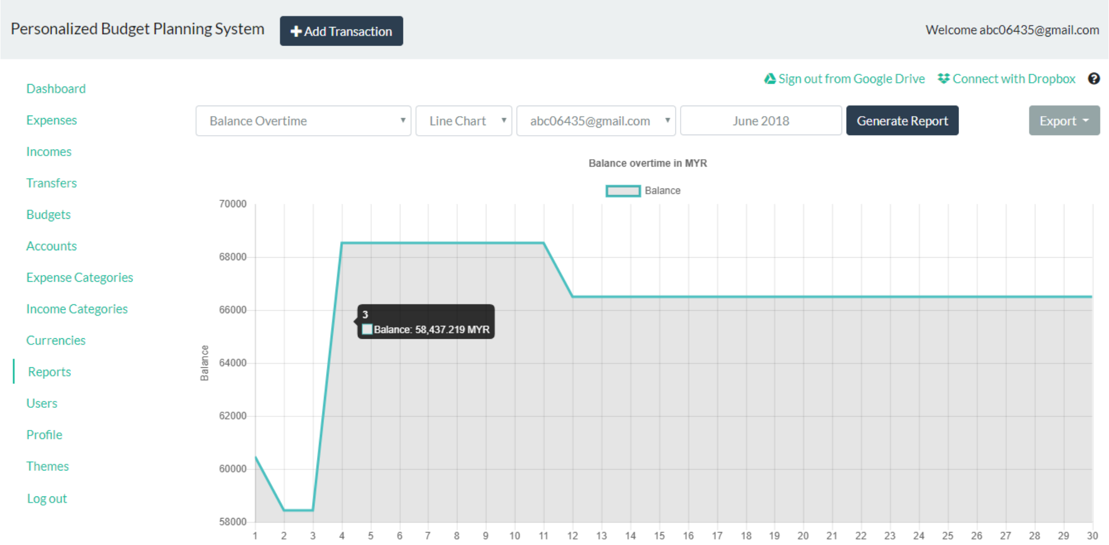


Date	Expenses (MYR)	Cumulative Expenses (MYR)	Incomes (MYR)	Cumulative Incomes (MYR)
01 June	500.00	500.00	0.00	0.00
02 June	2019.63	2519.63	0.00	0.00
03 June	0.00	2519.63	0.00	0.00
04 June	0.00	2519.63	10098.16	10098.16
05 June	0.00	2519.63	0.00	10098.16
06 June	0.00	2519.63	0.00	10098.16
07 June	0.00	2519.63	0.00	10098.16
08 June	0.00	2519.63	0.00	10098.16
09 June	0.00	2519.63	0.00	10098.16
10 June	0.00	2519.63	0.00	10098.16
11 June	0.00	2519.63	0.00	10098.16
12 June	2019.63	4539.26	0.00	10098.16
13 June	0.00	4539.26	0.00	10098.16

Figure 3-77 Cumulative Expenses and Incomes Report Part 2

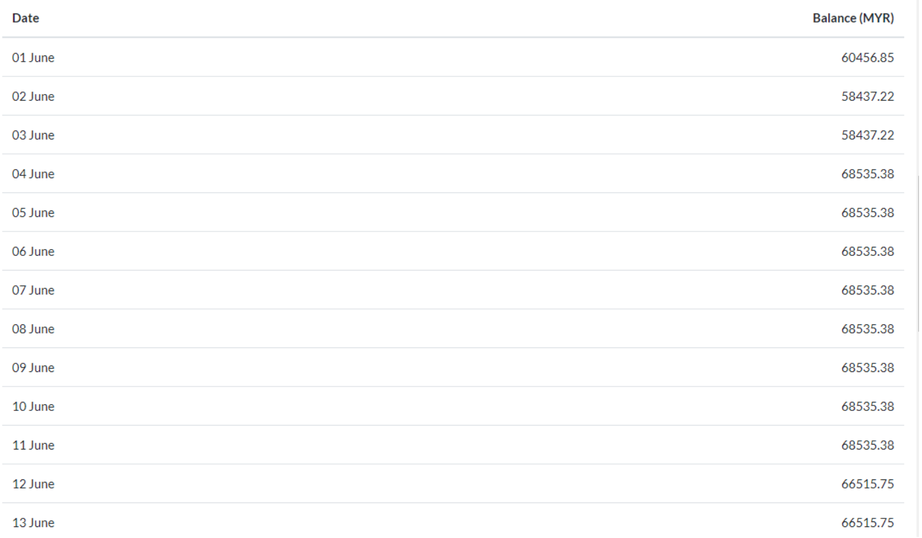
This report contains a chart and a table. The chart visualizes the user's cumulative expenses and incomes overtime using a line chart or a bar chart. The line or bar having the blue colour will be the expenses overtime while the line or bar with the red colour will be the incomes overtime. The table will display the user's cumulative expenses and incomes with a list of rows beginning with the first day of the specified month until the end of the specified month.

Balance Overtime Report



Balance Overtime | Line Chart | abc06435@gmail.com | June 2018

Figure 3-78 Balance Overtime Report Part 1

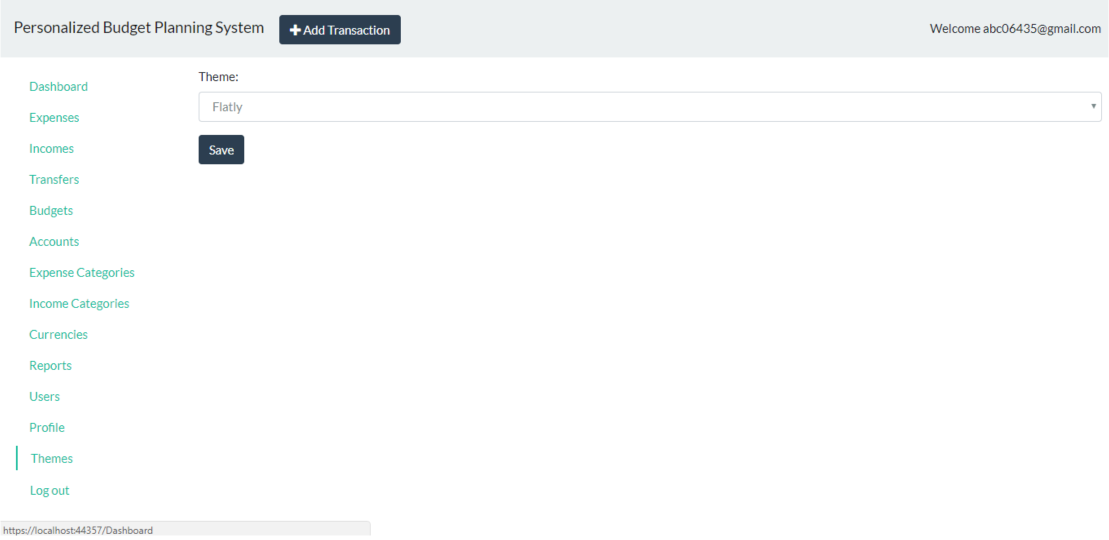


Date	Balance (MYR)
01 June	60456.85
02 June	58437.22
03 June	58437.22
04 June	68535.38
05 June	68535.38
06 June	68535.38
07 June	68535.38
08 June	68535.38
09 June	68535.38
10 June	68535.38
11 June	68535.38
12 June	66515.75
13 June	66515.75

Figure 3-79 Balance Overtime Report Part 2

This report displays the user balance overtime from the start of the specified month until the end of the specified month. The data is displayed both in chart form and table form. Two charts will be available for this report which are line chart and bar chart.

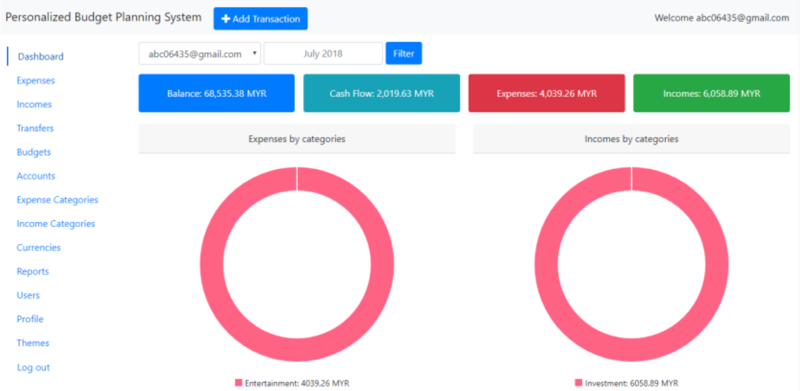
3.4.10 Theme Selection



Theme: Flatly

Figure 3-80 Theme Selection

The user can select a theme to change the appearance of the application. By default, the theme ‘Default’ will be used. Below is a comparison of ‘Default’ theme and ‘Flatly’ theme.



abc06435@gmail.com July 2018

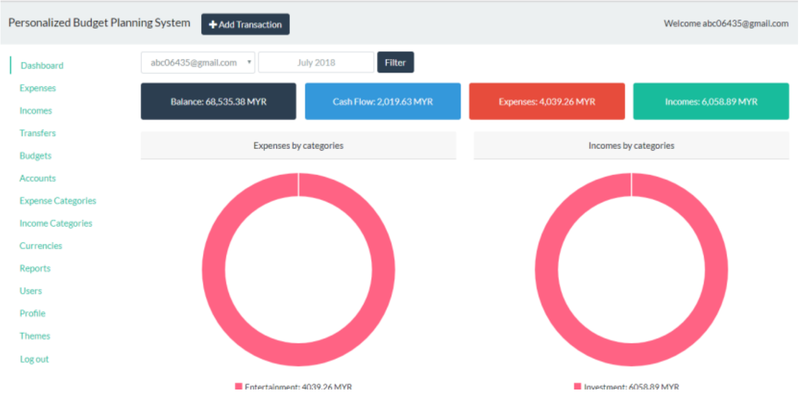
Balance: 68,535.38 MYR

Cash Flow: 2,019.63 MYR

Expenses: 4,039.26 MYR

Incomes: 6,058.89 MYR

Figure 3-81 Default Theme



abc06435@gmail.com July 2018

Balance: 68,535.38 MYR

Cash Flow: 2,019.63 MYR

Expenses: 4,039.26 MYR

Incomes: 6,058.89 MYR

Figure 3-82 Flatly Theme

Chapter 4 Methodology and Tools

4.1 Methodology

The methodology to be adopted for developing the application is the Prototype modelling methodology. The prototype modelling methodology focuses on developing a product that satisfies the minimum requirements and delivers the prototype to the customers. The customers then observe if the prototype matches their requirements while providing feedbacks (Ghahrai, 2008) that can potentially reduce the risks of developing a product that does not satisfy the customers. The prototype continued to be developed and delivered to the customer, each iteration modifies the prototype based on the customer feedback for improving the quality of the prototype.

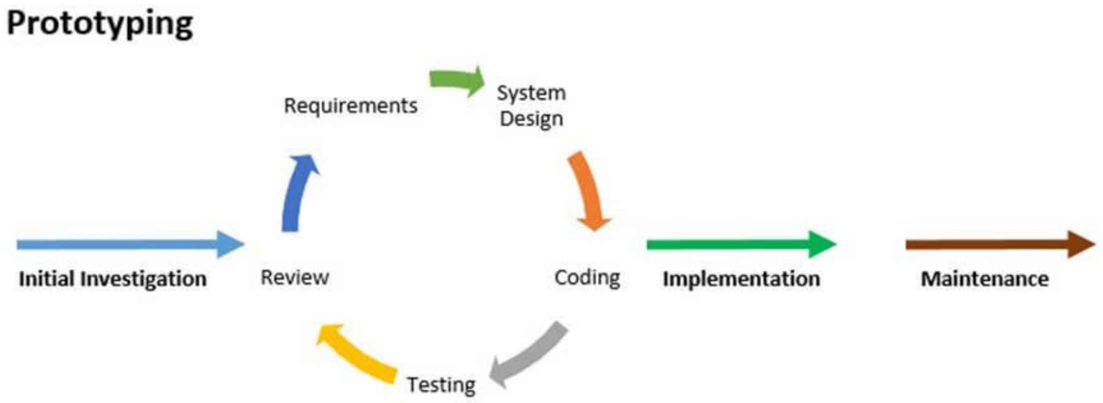


Figure 4-1 Prototyping Methodology (Ghahrai, 2008)

4.1.1 Initial Investigation

The initial investigation involves identify the problem that needs to be solved. Some requirements can be identified on this stage. This phase moves into the life cycle of prototyping that goes in the cycle of requirements -> system design -> coding -> testing -> review -> requirements.

4.1.2 Requirements

Involves a deeper analysis of the requirements of the application compare to initial investigation phase. Information were gathered on this stage to ensure the project was feasible. The requirements may be refined if the requirements phase was entered from the review phase. The requirements of the project will most likely change over time because of the iteration.

4.1.3 System Design

Some of the existing systems that provide similar functionalities of the intended application to build were investigated and analysed. The design of the application involves the technology to be used, how these technologies interact with each other, the potential technology that can replace the other technology, etc. The system design could change over time based on the requirements of the application. The system was designed as allowing users to access their budget information if they are authenticated. An authenticated user was only able to access his/her own information (e.g. User "Bob" will not be able to delete the income record of the user "Alice"). The operations the users were allowed to perform including manage financial accounts (e.g. Bank account 1, Wallet, etc), budgets, expense categories, expenses, income categories, incomes, transfers, profiles. If the user does not have an account, they can register one by providing their email and password. In the future, the user can login to the application by providing their email and password.

4.1.4 Coding

The coding phase was dependent on the previous phase (system design phase) and will change to synchronize with the system design. The purpose of coding was to implement workable functionalities based on system design.

4.1.5 Testing

The testing phase was conducted to ensure the functionalities implemented in the coding phase were working as intended. The testing phase find potential bugs and issues to reduce the number of bugs of the prototype. The testing phase followed a test plan to test different components of the application.

4.1.6 Review

The customers reviewed the prototype and provide feedbacks. The feedbacks provided were valuable information that can be used to refine the requirements of the application if necessary and thus enhance the prototype. This process reduces the risk of delivering application that does not satisfy the customers while reducing the gap of customer expectation and software functionalities. If the prototype does not satisfy the customers, the iteration continues by moving into the requirements phase. If the customers were satisfied on the prototype, the development phase moved into the implementation phase.

4.1.7 Implementation

Implementation phase involves converting the prototype into actual application that can be used in production. In case of running the application on just local server, the user is required to install necessary tools to run the application including Asp.NET Core 2.0 SDK, Visual Studio Community Edition 2017 on the local environment.

4.1.8 Maintenance

Maintenance involves ensuring the application was able to serve customers in the long run such as ensuring minimum downtime so the users can access their budget information. There are multiple factors that can affect the maintainability of the application including the web servers, databases, bugs that were not found during testing, etc. During maintenance the users may have ideas on adding features that can enhance the application as well.

4.2 Tools to use

4.2.1 Asp.NET Core using C#

Asp.NET Core is a technology from Microsoft that can be used to develop web applications (Daniel Roth, Rick Anderson, Shaun Luttin, 2017). The developers can either use an IDE known as Visual Studio or a text editor such as Visual Studio Code, Atom, Sublime Text 3, and etc to develop applications using this technology. To use Asp.NET Core, the Asp.NET Core 2.0 SDK is required to be installed into the machine, the Asp.NET Core 2.0 SDK can be downloaded at https://www.microsoft.com/net/download/windows.

4.2.2 HTML, CSS, and JavaScript

These technologies are required to build the user interface of the application. HTML was used to provide the mark up for the layout, and providing basic user interface without concerning the styling of the user interface. CSS was used to modify the appearance of the user interface while JavaScript was used to provide functionalities on the user interface such as handling mouse click event and opening a modal.

4.2.3 jQuery

jQuery is a JavaScript library that provide several utilities such as providing a CSS like way to select elements on the DOM. jQuery was also an optional dependency for the Bootstrap framework if the developer intends to use some of the Bootstrap utilities such as creating a modal or dropdown list.

4.2.4 Bootstrap

Bootstrap is a web framework that can be used to provide some default styling to the application as well as providing some frequently used user interface utilities (e.g. creating a modal, creating a dropdown menu, etc). There are various versions of the framework available currently, version 4 was used for this project.

4.2.5 ChartJs

ChartJs is a JavaScript library that is used to generate various charts such as pie chart, line chart, bar chart, and etc. This library was primarily used in the dashboard section of the application to generate various charts such as a doughnut to visualize user's income structure well as the report section of the application.

4.2.6 Node Package Manager

Node Package Manager is a tool that is used to download JavaScript dependencies to the project. This tool was used to download various JavaScript libraries such as Webpack, Gulp, PdfMake, Html2Canvas, and etc. When a dependency is successfully downloaded, the dependency will be recorded in a file named "package.json". This tool is not required to run the project as it is simply used for managing JavaScript dependencies. The application will run just fine without having this tool being installed on the machine. This tool can be downloaded via https://nodejs.org/en/.

4.2.7 Webpack

Webpack is used as a bundler to bundle different JavaScript files together. This can reduce HTTP request for different JavaScript files and allowing JavaScript modules to be imported and exported.

4.2.8 Gulp

Gulp is a task runner that can be used to automate certain tasks during development such as compiling PostCSS files into a single CSS file and automatically re bundle the JavaScript files when it detects the user changes the code in JavaScript files.

4.2.9 PdfMake

PdfMake is a JavaScript library that is used to generate PDFs based on the data provided. This library was used when the user wants to export reports.

4.2.10 Html2Canvas

This is a library that is used to screenshot the web page. This library is used to screenshot charts on the report page when the user wants to export their report.

4.2.11 Toastr

Toastr is a library that is used to display a toast message similar to a toast message on a mobile application. This library was used when the user attempts to download, preview, or export report to third party cloud storage.

4.2.12 Google Drive SDK

The Google Drive SDK is used when the user wants to export report to Google Drive. This SDK can be used to connect the user to Google Drive and export report to the user’s Google Drive given that the user was connected.

4.2.13 Dropbox SDK

The Dropbox SDK is used when the user wants to export report to Dropbox. This SDK can be used to connect the user to Dropbox and export report to the user’s Dropbox given that the user was connected.

4.2.14 SQL Server Express LocalDB

SQL Server Express LocalDB is a database management system that can used for developing applications (Anderson, 2017). It is a lightweight version of SQL Server Express Database Engine.

4.3 Requirements

4.3.1 Visual Studio Community 2017

Visual Studio Community 2017 is an Integrated Development Environment for developing software using various technologies including Asp.NET Core. There are older versions of Visual Studio available but it is recommended to install the version stated (2017) to minimize potential compatibility issues.

4.3.2 Microsoft SQL Server Management Studio 17

Microsoft SQL Server Management Studio provides a user interface for integrating with SQL Server Express LocalDB. This tool is optional and is not required to run the program.

4.3.3 Laptop

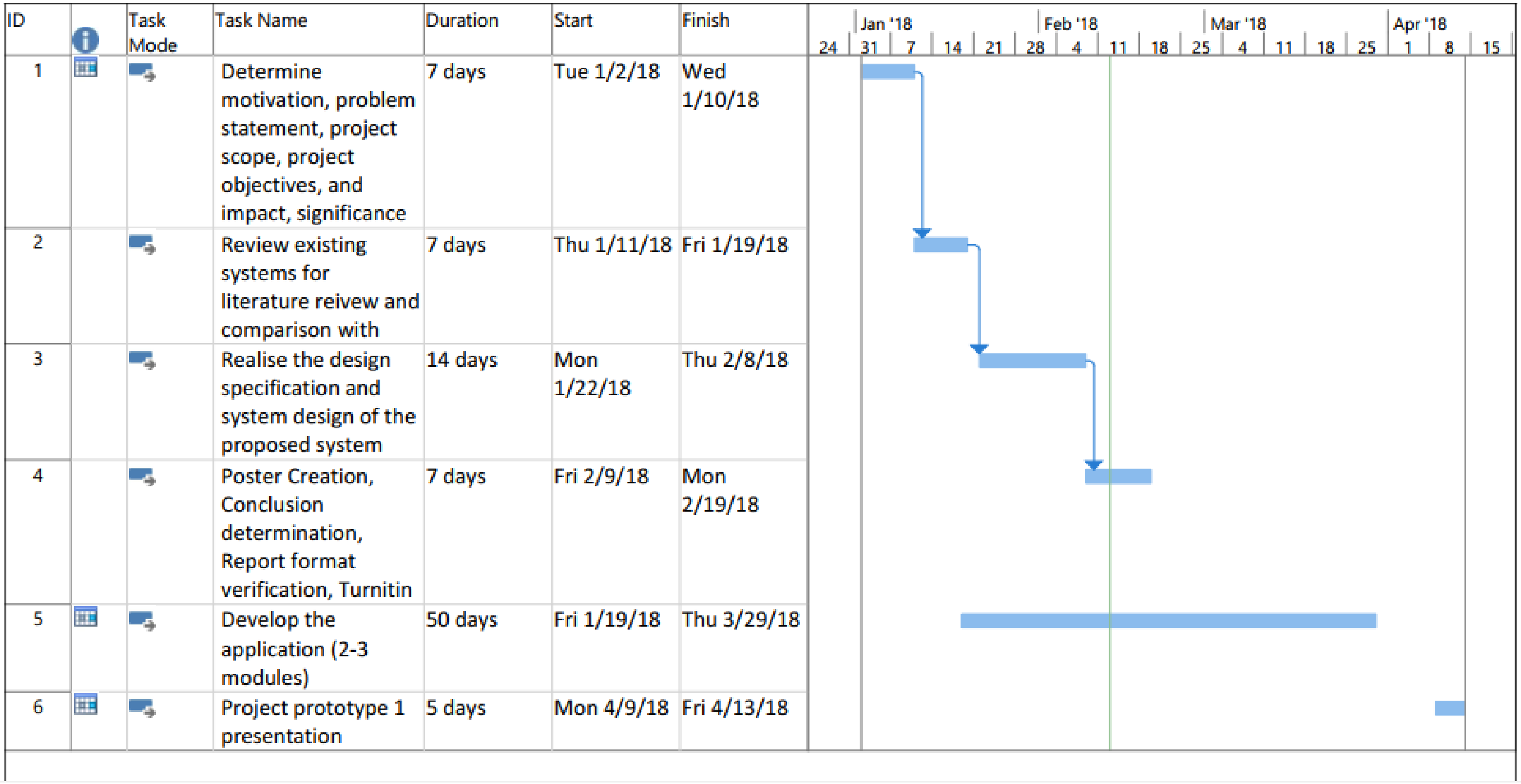
Operating System: Windows 10
Processor: Intel® Core™ i5-6200U CPU @ 2.30GHz
Installed memory (RAM): 8.00 GB (7.84 GB usable)

4.4 Timeline

4.4.1 Overview

The project begun in 2 January 2018 and the first prototype containing two to three completed modules (Project 1) is expected to be completed before 4 March 2018. During the first two weeks of January 2018 the primary focuses are information gathering, review of existing systems and feasibility studies. After the information gathering phase, the program will be developed based on the information gathered. The second prototype is expected to be completed before 10 August 2018. A review of the previous prototype will be conducted first before continuing to develop the program.

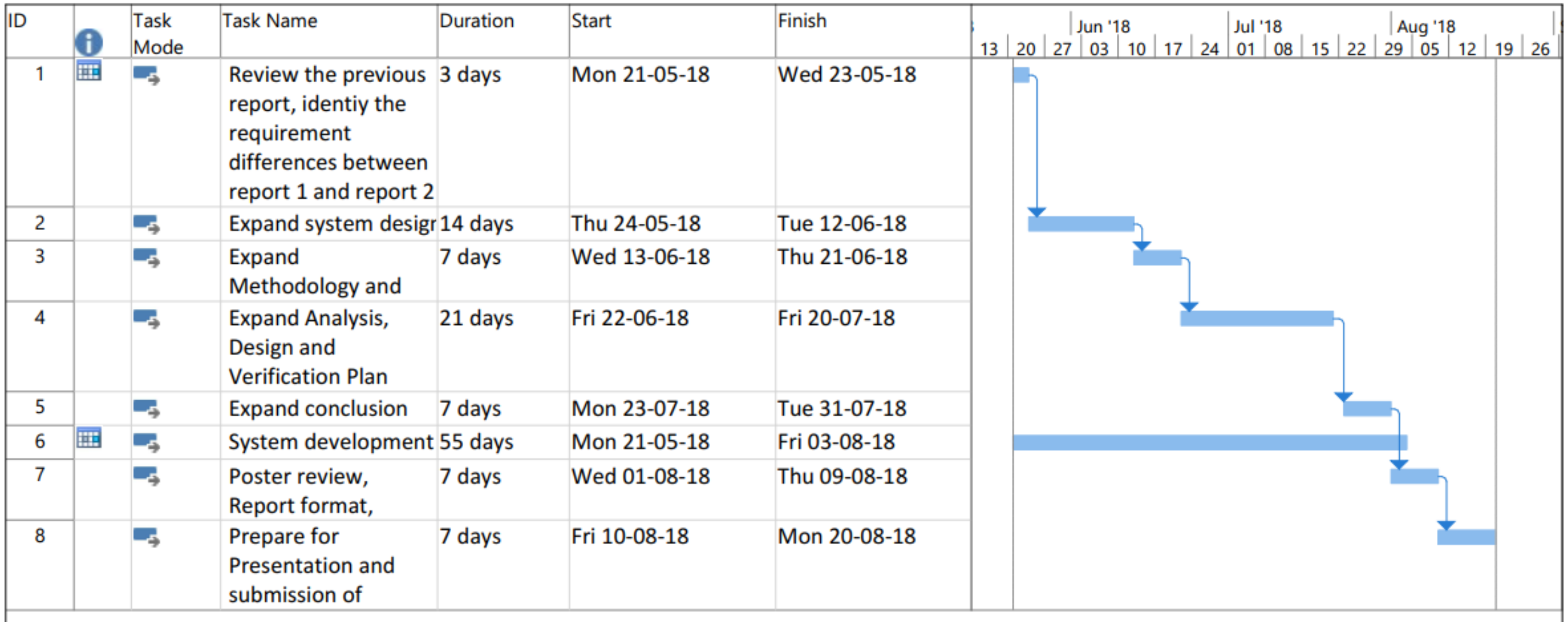
4.4.2 Gantt Chart for Current Semester



ID	ⓘ	Task Mode	Task Name	Duration	Start	Finish
1	▦	➔	Determine motivation, problem statement, project scope, project objectives, and impact, significance	7 days	Tue 1/2/18	Wed 1/10/18
2		➔	Review existing systems for literature reivew and comparison with	7 days	Thu 1/11/18	Fri 1/19/18
3		➔	Realise the design specification and system design of the proposed system	14 days	Mon 1/22/18	Thu 2/8/18
4		➔	Poster Creation, Conclusion determination, Report format verification, Turnitin	7 days	Fri 2/9/18	Mon 2/19/18
5	▦	➔	Develop the application (2-3 modules)	50 days	Fri 1/19/18	Thu 3/29/18
6	▦	➔	Project prototype 1 presentation	5 days	Mon 4/9/18	Fri 4/13/18

Figure 4-2 Gantt Chart Part 1

4.4.3 Gantt Chart for Next Semester



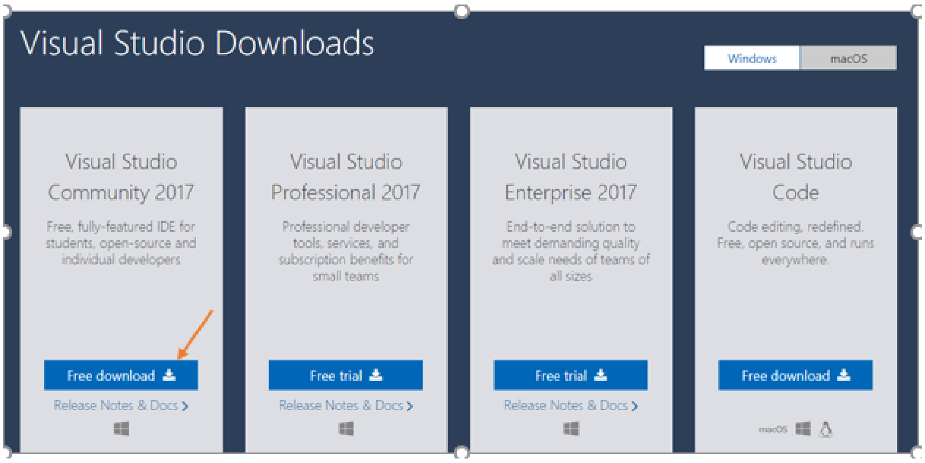
ID	ℹ	Task Mode	Task Name	Duration	Start	Finish
1	▦		Review the previous report, identiy the requirement differences between report 1 and report 2	3 days	Mon 21-05-18	Wed 23-05-18
2			Expand system desigr	14 days	Thu 24-05-18	Tue 12-06-18
3			Expand Methodology and	7 days	Wed 13-06-18	Thu 21-06-18
4			Expand Analysis, Design and Verification Plan	21 days	Fri 22-06-18	Fri 20-07-18
5			Expand conclusion	7 days	Mon 23-07-18	Tue 31-07-18
6	▦		System development	55 days	Mon 21-05-18	Fri 03-08-18
7			Poster review, Report format,	7 days	Wed 01-08-18	Thu 09-08-18
8			Prepare for Presentation and submission of	7 days	Fri 10-08-18	Mon 20-08-18

Figure 4-3 Gantt Chart Part 2

Chapter 5 Operating Manual

5.1 Install required tools

1. Go to https://www.visualstudio.com/downloads/
2. Click on free download for Visual Studio Community 2017



Visual Studio Downloads

Visual Studio Community 2017

Free, fully-featured IDE for students, open-source and individual developers

Visual Studio Professional 2017

Professional developer tools, services, and subscription benefits for small teams

Visual Studio Enterprise 2017

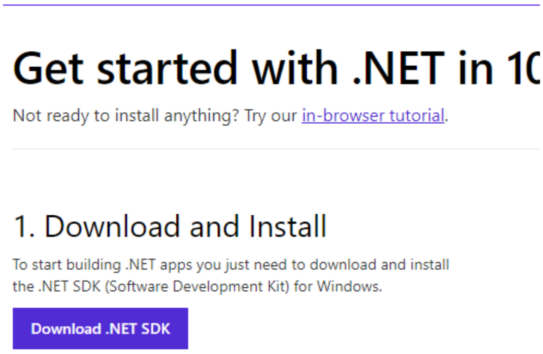
End-to-end solution to meet demanding quality and scale needs of teams of all sizes

Visual Studio Code

Code editing, redefined. Free, open source, and runs everywhere.

Figure 5-1 Visual Studio 2017 Download Page

3. Run the installer that was downloaded
4. Ensure ASP.NET web development is ticked
5. Install the selected modules (Restart might be required, after restarting, run the installer again, the installer will provide an option for launching Visual Studio Community 2017)
6. Go to https://www.microsoft.com/net/learn/get-started/windows, click on download .NET SDK



Get started with .NET in 1(

Not ready to install anything? Try our in-browser tutorial.

1. Download and Install

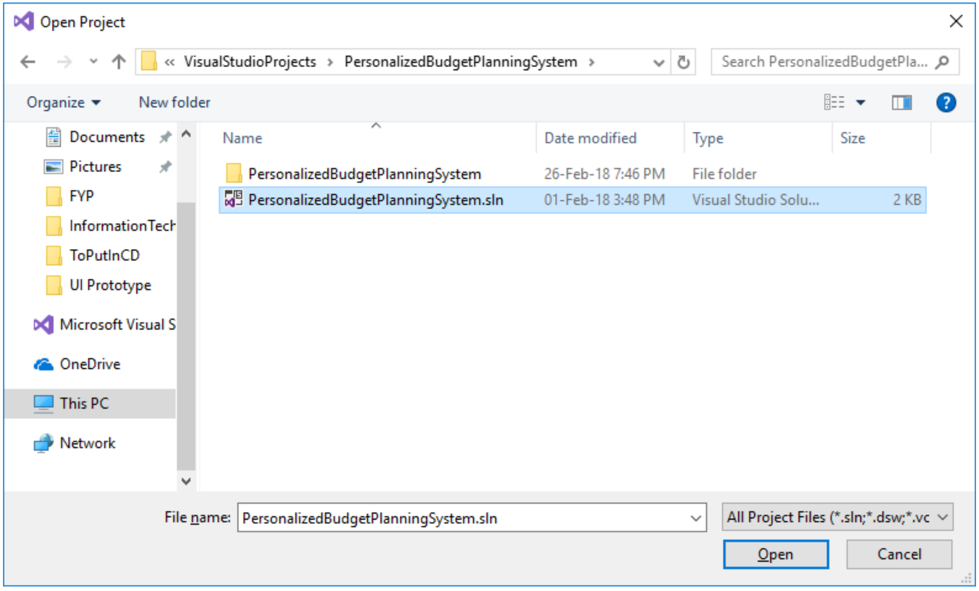
To start building .NET apps you just need to download and install the .NET SDK (Software Development Kit) for Windows.

Figure 5-2 .NET SDK Download Page

7. Run the downloaded installer

5.2 Running the project

1. Open visual studio, in visual studio, open solution by file > open
2. Select the solution file, click open



File name: PersonalizedBudgetPlanningSystem.sln

Figure 5-3 Open Project

3. Open the package manager console by going to Tools > Nuget Package Manager > Package Manager Console.

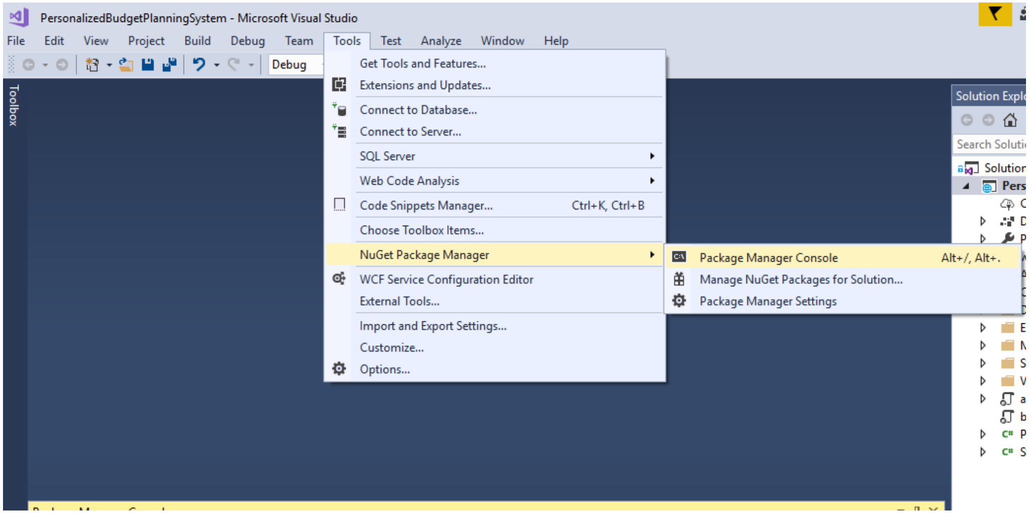
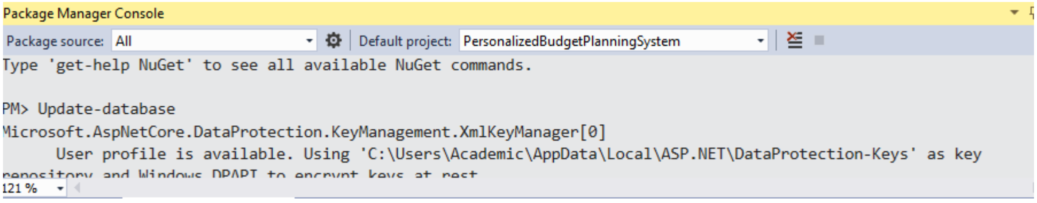


Figure 5-4 Open Package Manager Console

4. Inside the package manager console, enter “Update-database”. The purpose of this command is to create the database on the C drive based on the migration

files.

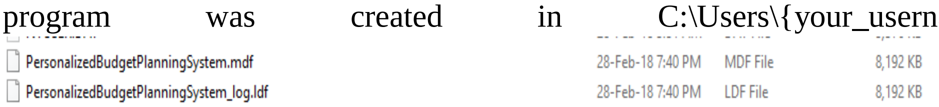


```
Type 'get-help NuGet' to see all available NuGet commands.

PM> Update-database
Microsoft.AspNetCore.DataProtection.KeyManagement.XmlKeyManager[0]
      User profile is available. Using 'C:\Users\Academic\AppData\Local\ASP.NET\DataProtection-Keys' as key
repository and Windows DPAPI to encrypt keys at rest
```

Figure 5-5 Update-Database Command

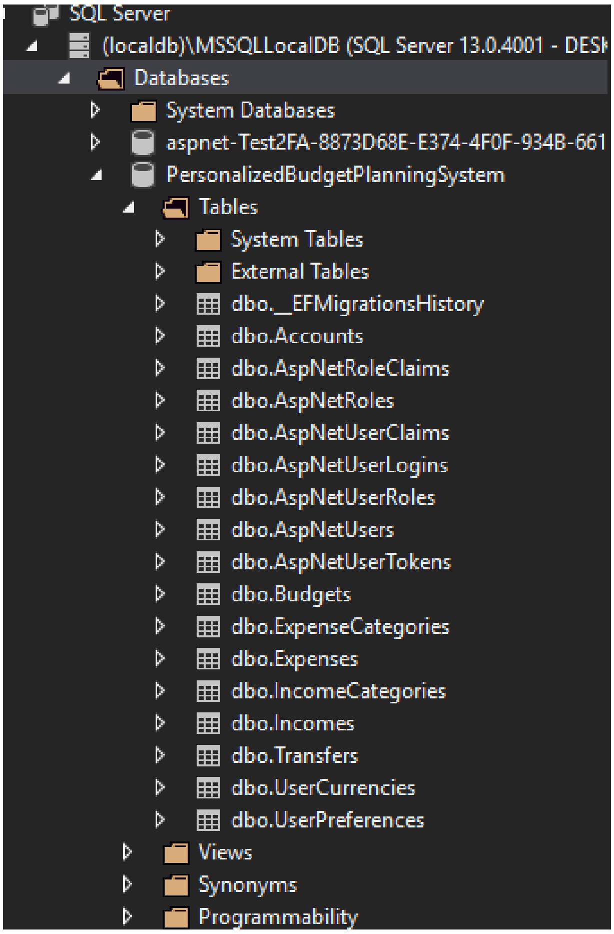
5. Verify database was created first check: verify that the database files for the program was created in C:\Users\{your_username}.



PersonalizedBudgetPlanningSystem.mdf	28-Feb-18 7:40 PM	MDF File	8,192 KB
PersonalizedBudgetPlanningSystem_log.ldf	28-Feb-18 7:40 PM	LDF File	8,192 KB

Figure 5-6 Verify Database Created Part 1

6. Verify database was created second check: Verify that the database is created by opening View (located at the top toolbar) > SQL Server Object Explorer



```
SQL Server
  (localdb)\MSSQLLocalDB (SQL Server 13.0.4001 - DESK
    Databases
      System Databases
      aspnet-Test2FA-8873D68E-E374-4F0F-934B-661
      PersonalizedBudgetPlanningSystem
        Tables
          System Tables
          External Tables
          dbo.__EFMigrationsHistory
          dbo.Accounts
          dbo.AspNetRoleClaims
          dbo.AspNetRoles
          dbo.AspNetUserClaims
          dbo.AspNetUserLogins
          dbo.AspNetUserRoles
          dbo.AspNetUsers
          dbo.AspNetUserTokens
          dbo.Budgets
          dbo.ExpenseCategories
          dbo.Expenses
          dbo.IncomeCategories
          dbo.Incomes
          dbo.Transfers
          dbo.UserCurrencies
          dbo.UserPreferences
        Views
        Synonyms
        Programmability
```

Figure 5-7 Verify Database Created Part 2

7. Open the project directory with windows explorer
8. Open a file named “UserSecrets.json” in the folder.
9. Copy all content in the “UserSecrets.json” file.
10. Right click the project and click manage user secrets.

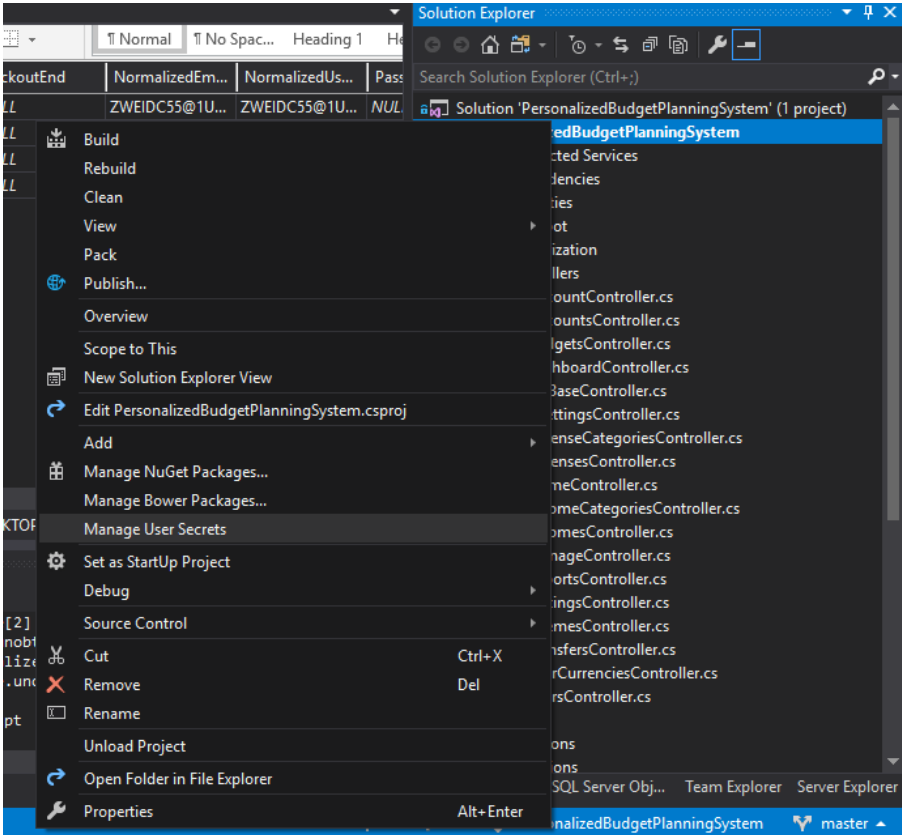
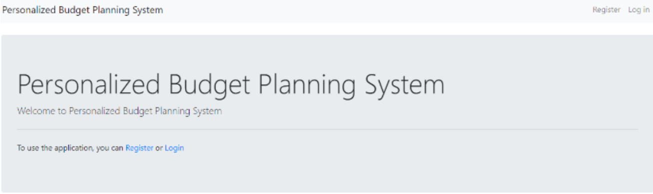


Figure 5-8 Manage User Secrets

11. Paste in the content that was copied from “UserSecrets.json”.
12. Save the file by using “CTRL + S”.
13. Explanation of user secrets (Optional): The user secrets are required for enabling Google login as well as sending emails using Sendgrid. If the user secrets were not provided, the system will not work.
14. Use “CTRL” + “F5” to run the project. (The project will run at https://localhost:44357/)
15. The application is now running, the next step will be populating some records. Currently, the database is empty and there are no available users to be logged n.


Personalized Budget Planning System

Register Log in

Personalized Budget Planning System

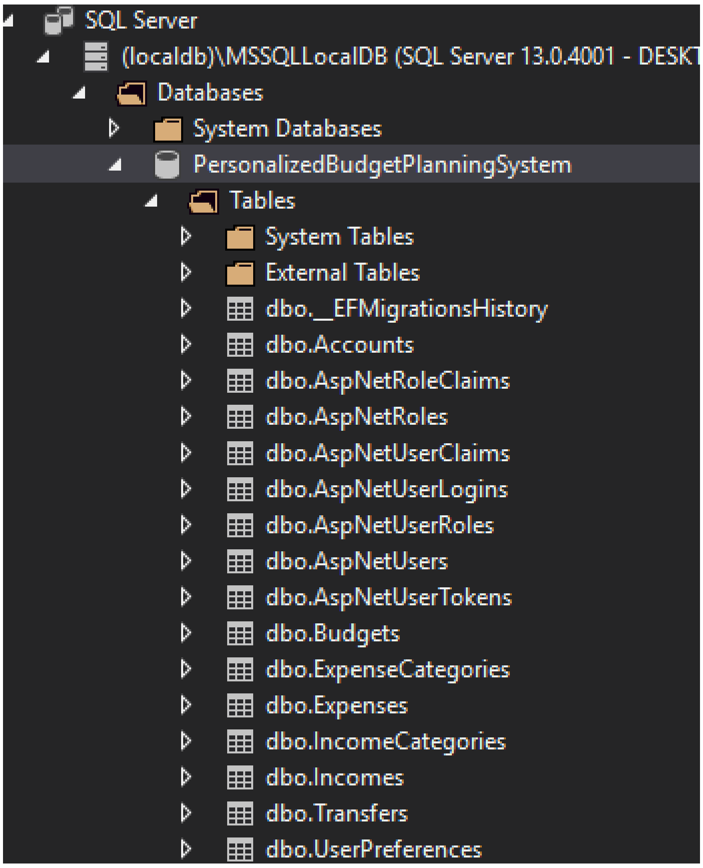
Welcome to Personalized Budget Planning System

To use the application, you can Register or Login

Figure 5-9 Application Home Page

5.3 Populating records

1. In visual studio, on the top links, click view > SQL Server Object Explorer
2. Expand the nodes with this fashion
   a. SQL Server
   b. (localdb)\MSSQLLocalDB (SQL Server 13.0.4001….) (The actual name might be different depend on the machine)
   c. Databases
   d. PersonalizedBudgetPlanningSystem

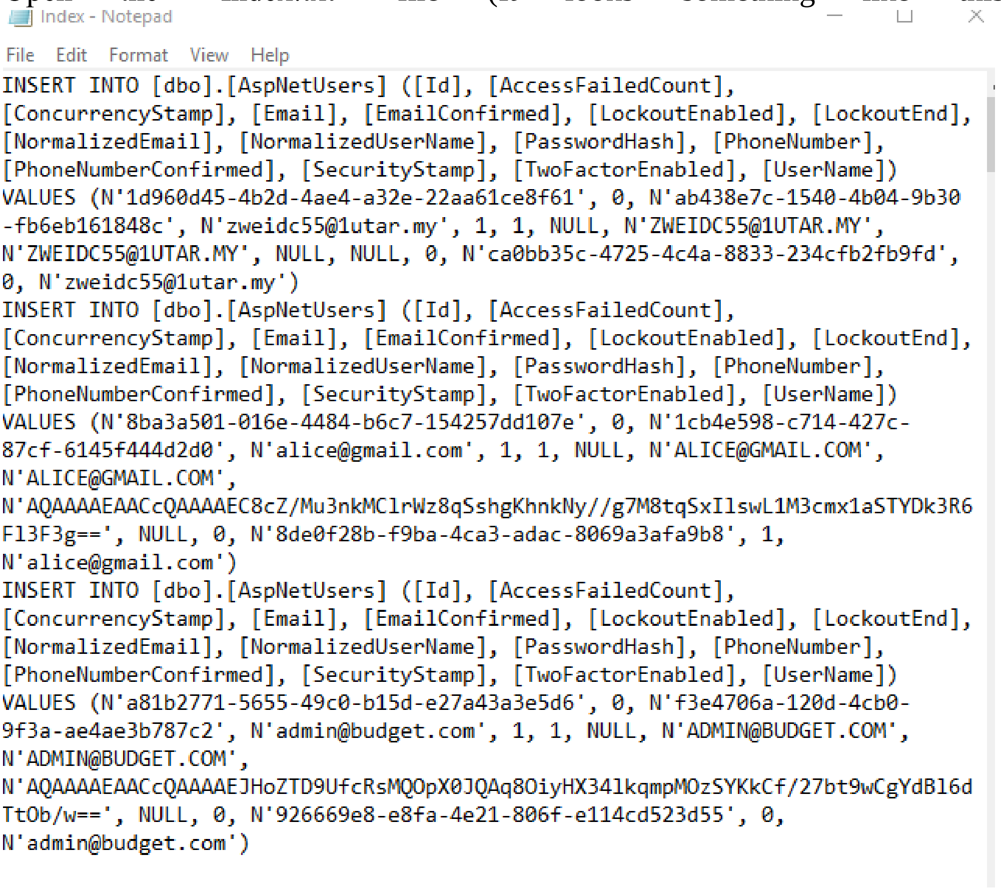


```
SQL Server
  (localdb)\MSSQLLocalDB (SQL Server 13.0.4001 - DESKT
    Databases
      System Databases
      PersonalizedBudgetPlanningSystem
        Tables
          System Tables
          External Tables
          dbo.__EFMigrationsHistory
          dbo.Accounts
          dbo.AspNetRoleClaims
          dbo.AspNetRoles
          dbo.AspNetUserClaims
          dbo.AspNetUserLogins
          dbo.AspNetUserRoles
          dbo.AspNetUsers
          dbo.AspNetUserTokens
          dbo.Budgets
          dbo.ExpenseCategories
          dbo.Expenses
          dbo.IncomeCategories
          dbo.Incomes
          dbo.Transfers
          dbo.UserPreferences
```

Figure 5-10 Database Nodes

3. Right click on PersonalizedBudgetPlanningSystem, select “New Query…”
4. Open the project directory with windows explorer
5. Open the “SQL Seeders” folder

6. Open the “Index.txt” file (It looks something like this)



```
INSERT INTO [dbo].[AspNetUsers] ([Id], [AccessFailedCount],
[ConcurrencyStamp], [Email], [EmailConfirmed], [LockoutEnabled], [LockoutEnd],
[NormalizedEmail], [NormalizedUserName], [PasswordHash], [PhoneNumber],
[PhoneNumberConfirmed], [SecurityStamp], [TwoFactorEnabled], [UserName])
VALUES (N'1d960d45-4b2d-4ae4-a32e-22aa61ce8f61', 0, N'ab438e7c-1540-4b04-9b30
-fb6eb161848c', N'zweidc55@1utar.my', 1, 1, NULL, N'ZWEIDC55@1UTAR.MY',
N'ZWEIDC55@1UTAR.MY', NULL, NULL, 0, N'ca0bb35c-4725-4c4a-8833-234cfb2fb9fd',
0, N'zweidc55@1utar.my')
INSERT INTO [dbo].[AspNetUsers] ([Id], [AccessFailedCount],
[ConcurrencyStamp], [Email], [EmailConfirmed], [LockoutEnabled], [LockoutEnd],
[NormalizedEmail], [NormalizedUserName], [PasswordHash], [PhoneNumber],
[PhoneNumberConfirmed], [SecurityStamp], [TwoFactorEnabled], [UserName])
VALUES (N'8ba3a501-016e-4484-b6c7-154257dd107e', 0, N'1cb4e598-c714-427c-
87cf-6145f444d2d0', N'alice@gmail.com', 1, 1, NULL, N'ALICE@GMAIL.COM',
N'ALICE@GMAIL.COM',
N'AQAAAAEAACcQAAAAEC8cZ/Mu3nkMClrWz8qSshgKhnkNy//g7M8tqSxIlswL1M3cmx1aSTYDk3R6
Fl3F3g==', NULL, 0, N'8de0f28b-f9ba-4ca3-adac-8069a3afa9b8', 1,
N'alice@gmail.com')
INSERT INTO [dbo].[AspNetUsers] ([Id], [AccessFailedCount],
[ConcurrencyStamp], [Email], [EmailConfirmed], [LockoutEnabled], [LockoutEnd],
[NormalizedEmail], [NormalizedUserName], [PasswordHash], [PhoneNumber],
[PhoneNumberConfirmed], [SecurityStamp], [TwoFactorEnabled], [UserName])
VALUES (N'a81b2771-5655-49c0-b15d-e27a43a3e5d6', 0, N'f3e4706a-120d-4cb0-
9f3a-ae4ae3b787c2', N'admin@budget.com', 1, 1, NULL, N'ADMIN@BUDGET.COM',
N'ADMIN@BUDGET.COM',
N'AQAAAAEAACcQAAAAEJHoZTD9UfcRsMQOpX0JQAq8OiyHX34lkqmpMOzSYKkCf/27bt9wCgYdBl6d
TtOb/w==', NULL, 0, N'926669e8-e8fa-4e21-806f-e114cd523d55', 0,
N'admin@budget.com')
```

Figure 5-11 Index.txt Content (SQL Statements)

7. Copy all of the content within the file with CTRL + A, then CTRL + C
8. Go back to visual studio, paste the content that was copied to the new query tab
9. Click the execute button (the green button on the top left) to execute the queries

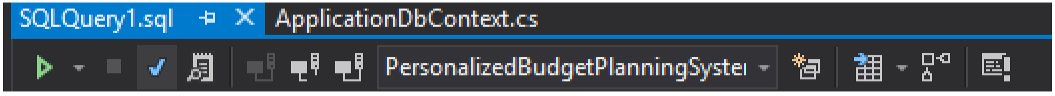


Figure 5-12 Execute Button

10. The database should now be populated with some records.

5.4 Record Purposes

Record	Purpose
Expense	This record is used to keep track of expenses created by the user. Creating this record will deduct balance of the

	account that was selected to make this expense.
Income	This record is used to keep track of incomes created by the user. Creating this record will increase balance of the account that was selected to make this expense.
Transfer	This record keeps track of transactions between two accounts. The receiver account’s balance will be increase while the provide account’s balance will be decreased upon the creation of this record.
Budget	This record keeps tracks of the user budgets. By default, no budget will be specified and the user will need to edit them to specify an amount for a particular expense category.
Account	This record is required to create expenses, incomes, and transfers.
Expense Category	This is required to create an expense. When the user creates an expense, the user will select an expense category using a dropdown list. This record also serves as a basis for specifying budget.
Income Category	This is required to create an income. When the user creates an income, the user will select an income category using a dropdown list.
Currencies	This record is required to create expenses, incomes, and accounts. When the user creates an expense, income, or

	account, the user will need to specify the currency.

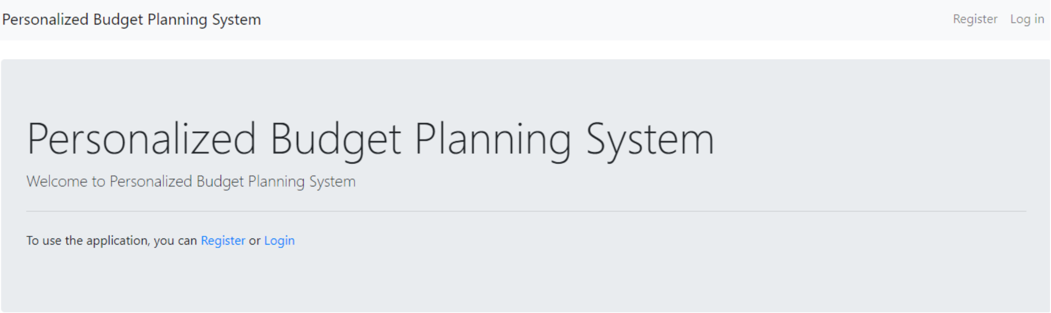
Table 5-1 Record Purposes

5.5 Using the program

5.4.1 Identity Management Module

Login as existing User

1. First, click on the login link on the page.



Personalized Budget Planning System

Register Log in

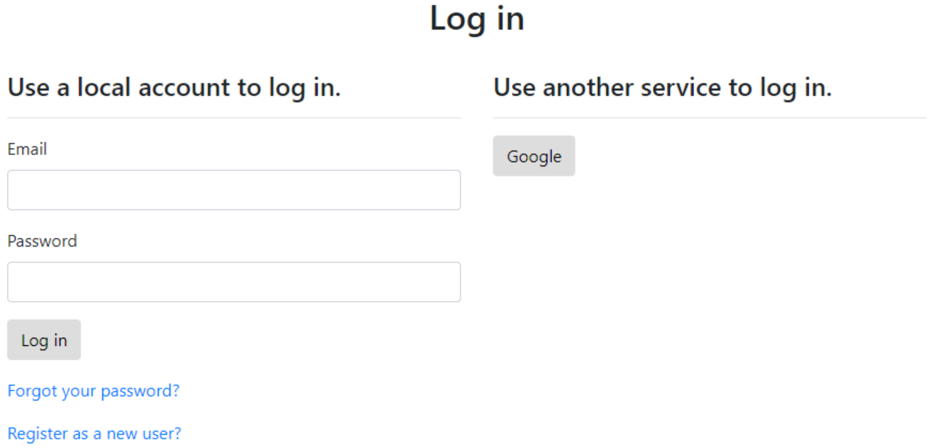
Personalized Budget Planning System

Welcome to Personalized Budget Planning System

To use the application, you can Register or Login

Figure 5-13 Application Home Page

2. Login by providing email and password, there are two users available. The first user is an admin while the second user is not an admin.



Log in

Use a local account to log in.

Email

Password

Log in

Forgot your password?

Register as a new user?

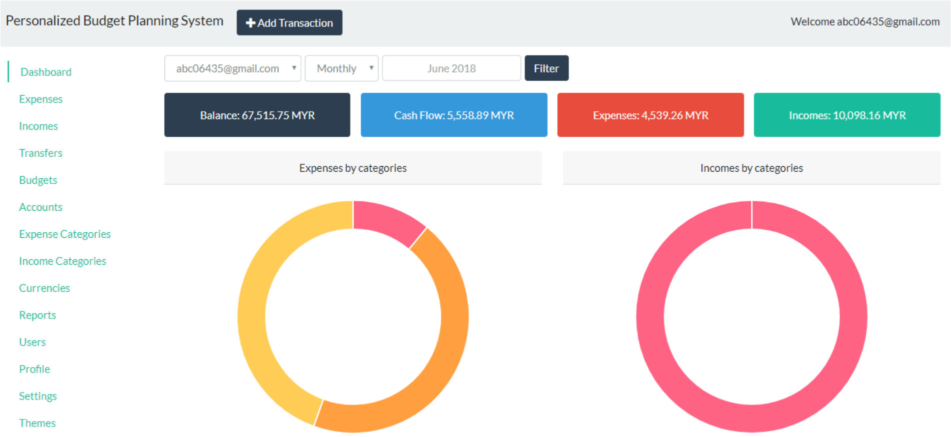
Use another service to log in.

Google

Figure 5-14 Application Login Page

Emails	Passwords	Role
Test1@budget.com	Password123$	Admin
Test2@budget.com	Password123$	User

3. The dashboard page should be displayed.



abc06435@gmail.com | Monthly | June 2018

Balance: 67,515.75 MYR

Cash Flow: 5,558.89 MYR

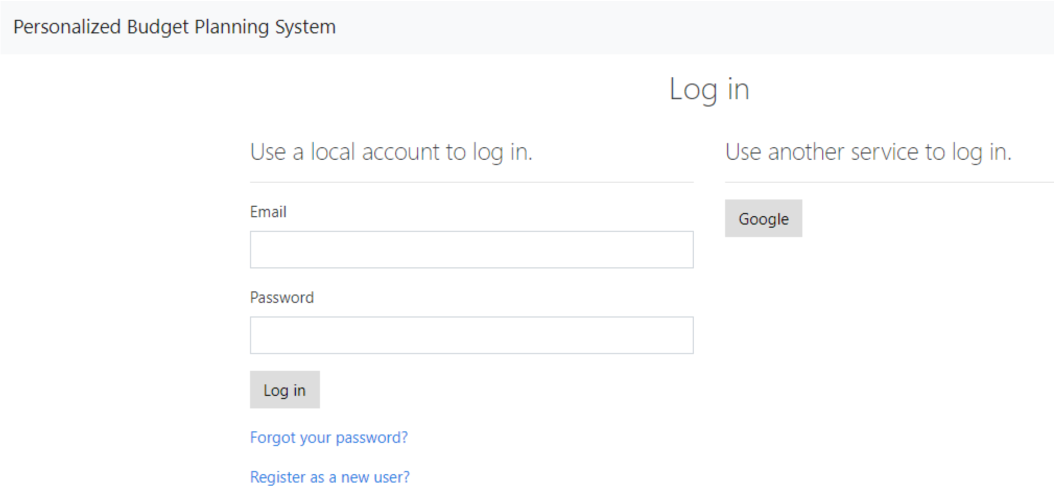
Expenses: 4,539.26 MYR

Incomes: 10,098.16 MYR

Figure 5-15 Dashboard Page

Register

You can register an account by going to the register page. You will need to fill up email, password, and confirm password, then click the register button. You can also register an account via Google. If you registered an account via email and password combination, you will need to activate it by clicking on the confirmation link on the email that was sent to the email used to register the account.



Log in

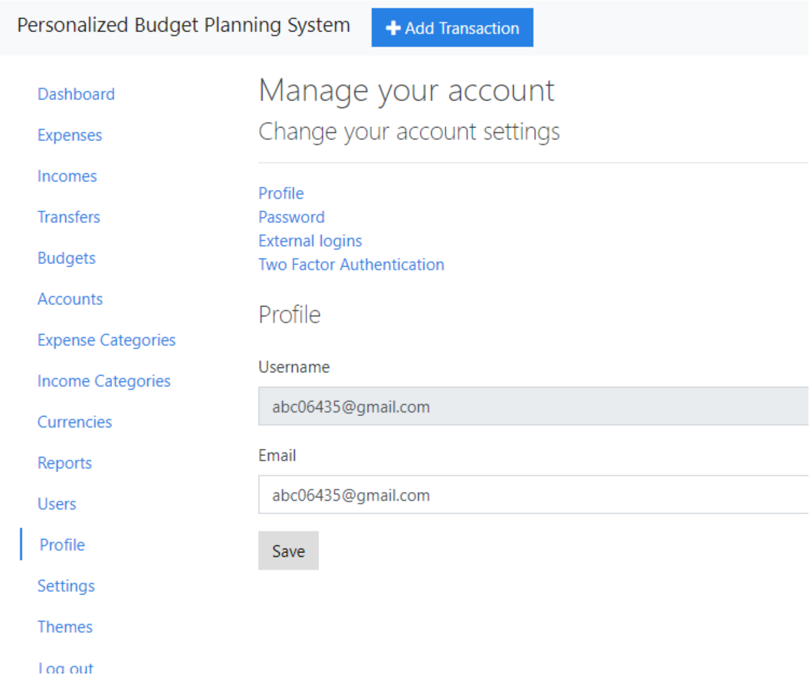
Use a local account to log in.

Use another service to log in.

Figure 5-16 Login

Profile

You can go to the profile page to update your email. After you update your email, a confirmation email will be sent to the new email.



Manage your account

Change your account settings

Profile

Username

abc06435@gmail.com

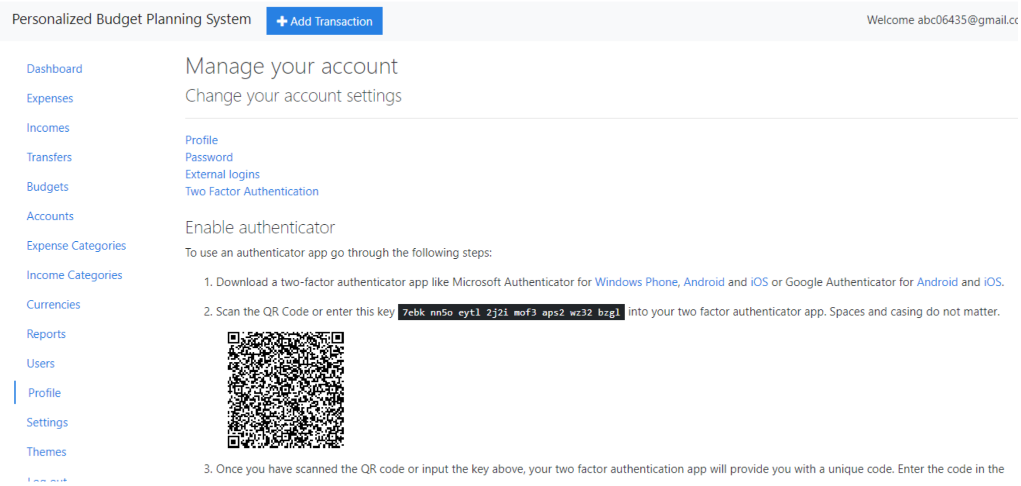
Email

abc06435@gmail.com

Figure 5-17 Profile Page

Enable Two Factor Authentication

1. Go to the profile page
2. Click on “Two factor Authentication” link
3. Click on “Configure authenticator app” link
4. You will see a page like this.



Manage your account

Change your account settings

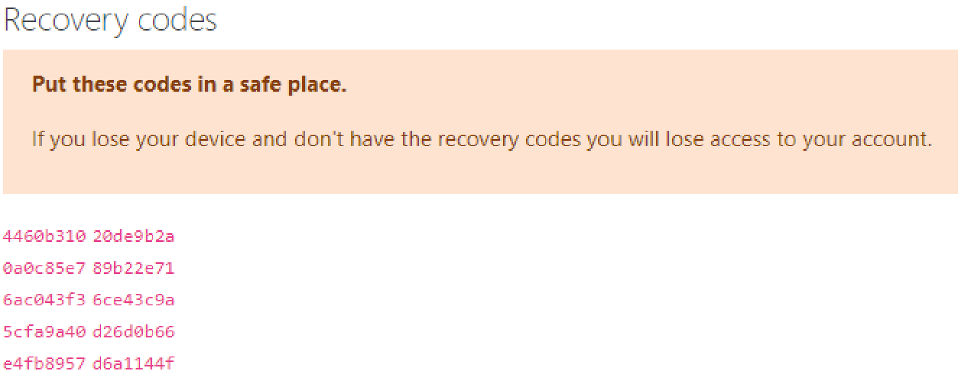
Enable authenticator

To use an authenticator app go through the following steps:

1. Download a two-factor authenticator app like Microsoft Authenticator for Windows Phone, Android and iOS or Google Authenticator for Android and iOS.
2. Scan the QR Code or enter this key `7ebk nn5o eytl 2j2i mof3 aps2 wz32 bzgl` into your two factor authenticator app. Spaces and casing do not matter.
3. Once you have scanned the QR code or input the key above, your two factor authentication app will provide you with a unique code. Enter the code in the

Figure 5-18 2FA Instructions Page

5. If you do not have Google Authenticator installed, download Google Authenticator on your mobile device.
6. Open Google Authenticator and scan the QR Code or enter the key provided on the page.
7. After scanning, Google Authenticator will provide a code. Enter the code provided on the input field and click verify.
8. After verifying, the page will update your two factor authentication details and provide you with some secret codes. These secret codes can be used as an alternative from the code provided in Google Authenticator. Your secret codes will most likely be different from the screenshot.



Recovery codes

Put these codes in a safe place.

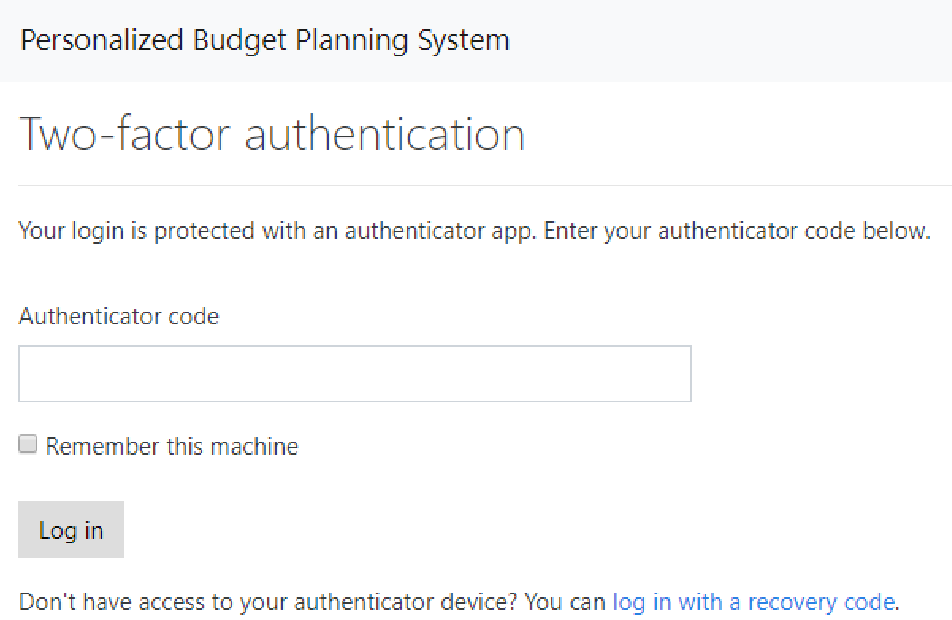
If you lose your device and don't have the recovery codes you will lose access to your account.

```
4460b310 20de9b2a
0a0c85e7 89b22e71
6ac043f3 6ce43c9a
5cfa9a40 d26d0b66
e4fb8957 d6a1144f
```

Figure 5-19 Secret Codes

9. Next time you try to login with correct email and password you will be prompted with this screen. Enter the code available via Google Authenticator to

login.



Personalized Budget Planning System

Two-factor authentication

Your login is protected with an authenticator app. Enter your authenticator code below.

Authenticator code

☐ Remember this machine

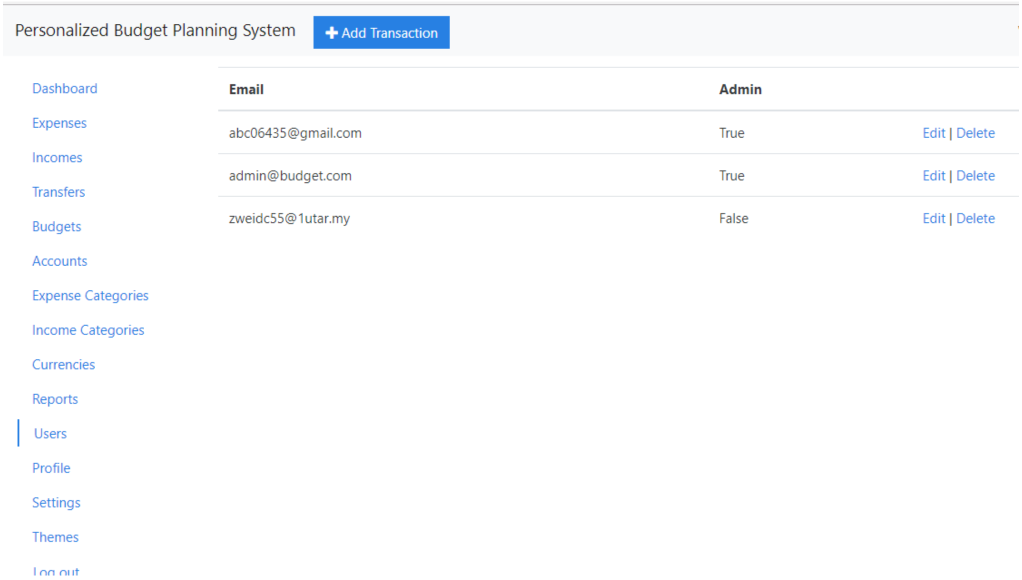
Log in

Don't have access to your authenticator device? You can log in with a recovery code.

Figure 5-20 2FA Login

Users

You can manage users by going to the users index page. Only an admin can view this page. The admin can delete a user or update a user's role. Note that deleting a user will also delete all of the records created by that particular user.



Email	Admin	
abc06435@gmail.com	True	Edit | Delete
admin@budget.com	True	Edit | Delete
zweidc55@1utar.my	False	Edit | Delete

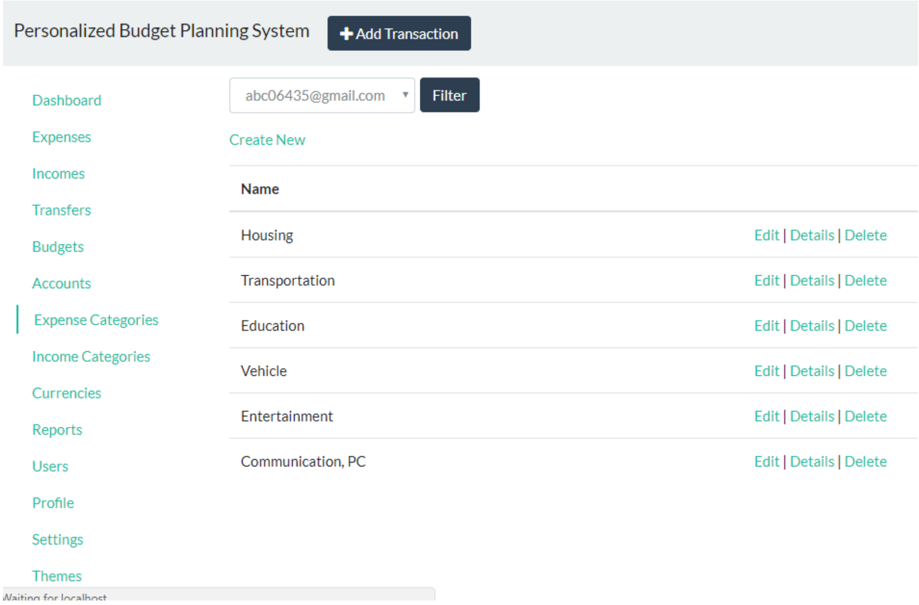
Figure 5-21 Users Page

5.4.2 Transaction Module

Create, Read, Update, Delete

Create

1. To create a record (e.g. expense, income, transfer, and etc.), the user is required to go the index page of the type of record intending to create. For example, if you want to create an expense record, you will need go to the index page of expenses by clicking on the expense link on the left navigation bar. By default, a list of expense categories and income categories will be created when the user registers an account.



Name	
Housing	Edit | Details | Delete
Transportation	Edit | Details | Delete
Education	Edit | Details | Delete
Vehicle	Edit | Details | Delete
Entertainment	Edit | Details | Delete
Communication, PC	Edit | Details | Delete

Figure 5-22 Index Page

2. To create a record. The user will need to click on the "Create new" link to go to the create page for the particular type of record.

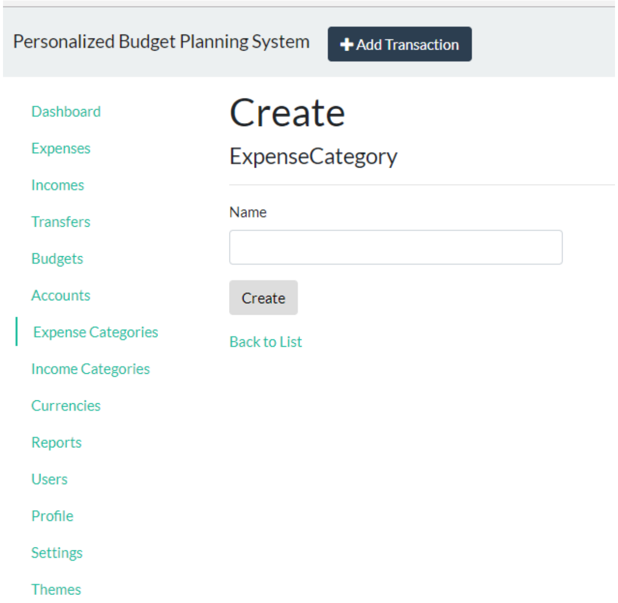


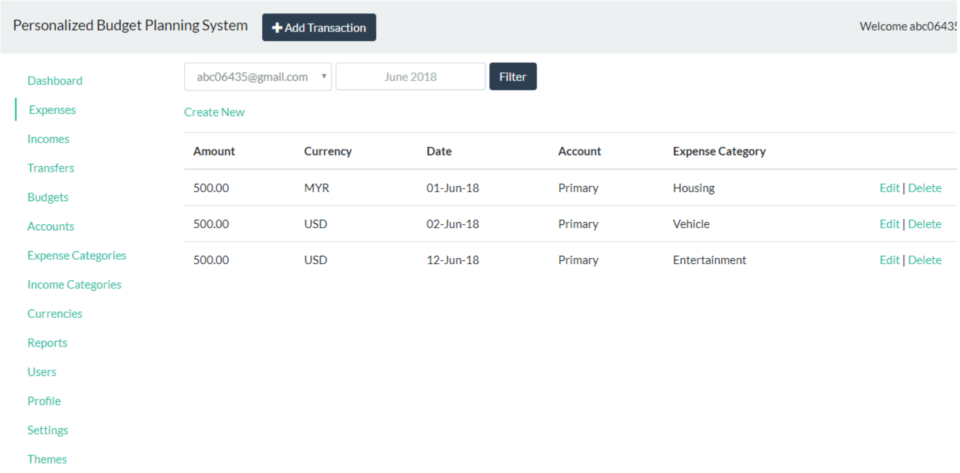
Figure 5-23 Create Page

3. After the user supplies necessary information on the form, the user can click the create button to create the record. If the record was created successfully, the system will redirect the user to the index page of a particular record type based on the record that was created. Some records will affect certain values of other records. For example, if an expense record was created with account A. The balance of account A will be reduced by the amount specified in the record.

Read

1. To read expenses (or any other record), click on the respective tab. For example, to view list of expenses, click on the expense tab. A list of expenses created by the user will be displayed.

2. It is possible to filter the records by selecting the month using the date picker above.

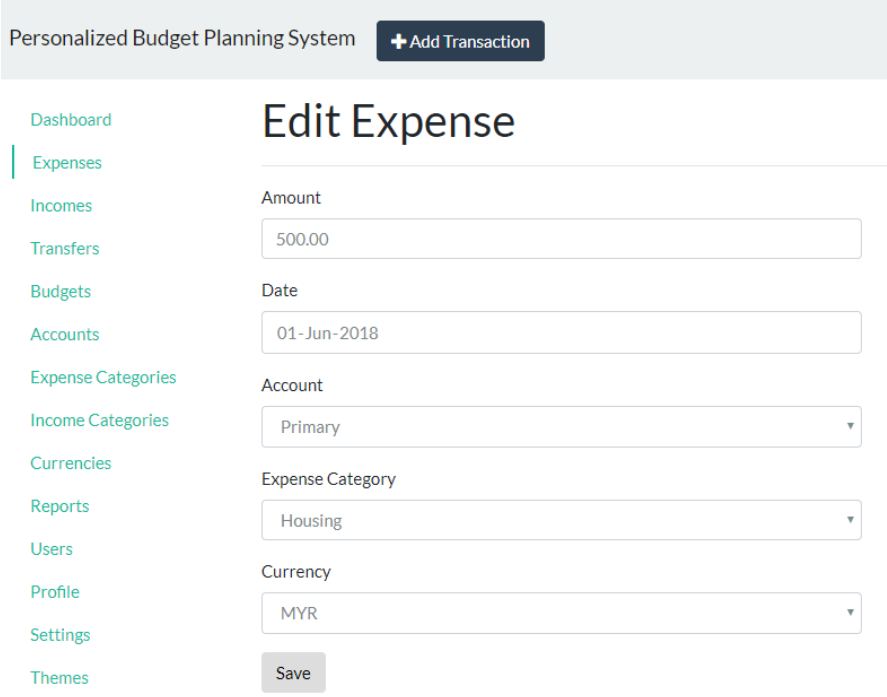


Amount	Currency	Date	Account	Expense Category	
500.00	MYR	01-Jun-18	Primary	Housing	Edit | Delete
500.00	USD	02-Jun-18	Primary	Vehicle	Edit | Delete
500.00	USD	12-Jun-18	Primary	Entertainment	Edit | Delete

Figure 5-24 List of Records Page

Update

1. To update a record, go to the list with the type of record you want to update (e.g. to update an expense record, click on the expenses link).
2. Click the edit link on the record you would like to update.
3. The update screen should be displayed to the user.



Edit Expense

Amount: 500.00

Date: 01-Jun-2018

Account: Primary ▾

Expense Category: Housing ▾

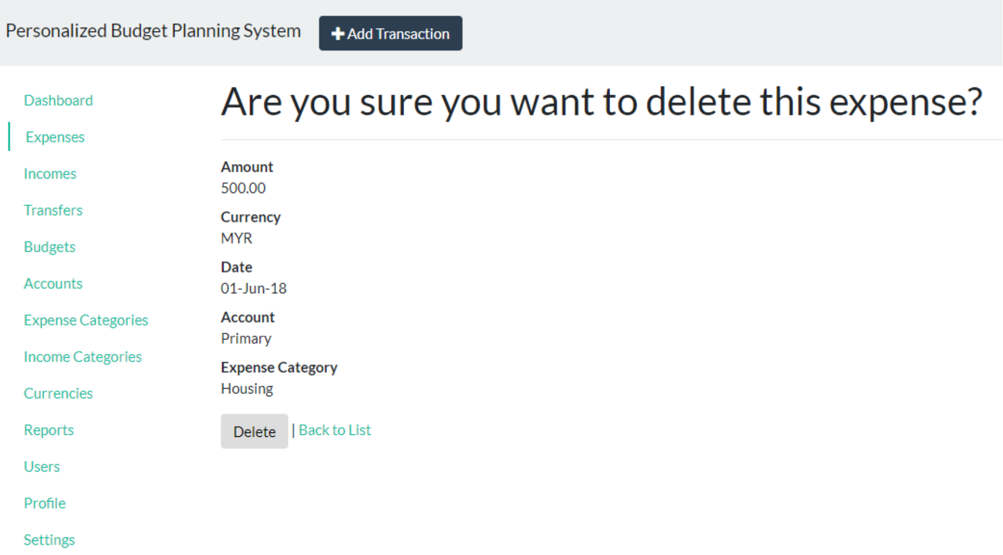
Currency: MYR ▾

Figure 5-25 Edit Expense Page

4. After making necessary changes, click the save button to save changes made.

Delete

1. To delete a record, go to the index page and click on the delete link on the record.
2. A delete confirmation page will be displayed.



Are you sure you want to delete this expense?

Amount
500.00

Currency
MYR

Date
01-Jun-18

Account
Primary

Expense Category
Housing

Figure 5-26 Delete Page

3. Click on the delete button to delete the record

5.4.3 Budget Module

1. The user can go to the budget page to set budget for an expense category.

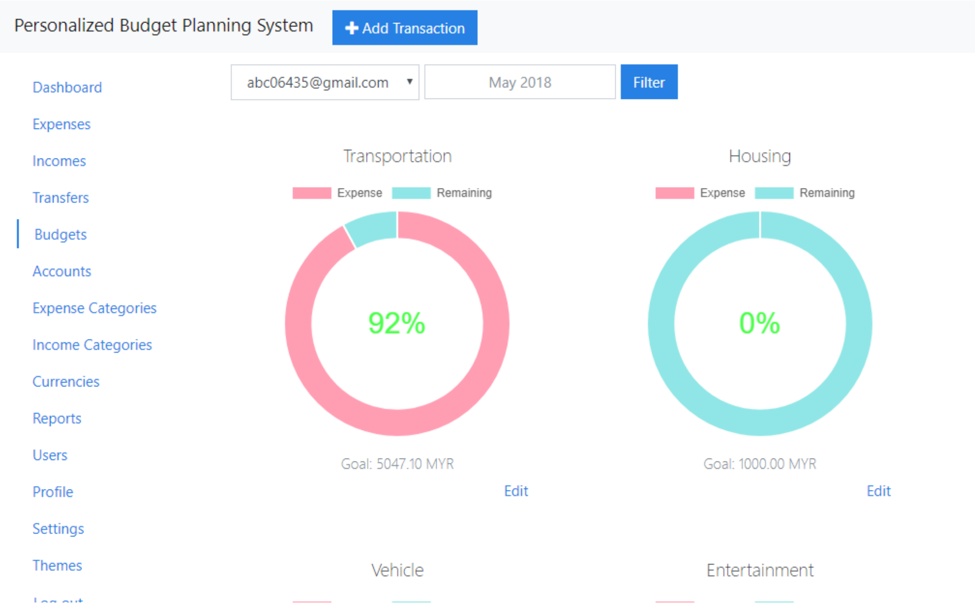
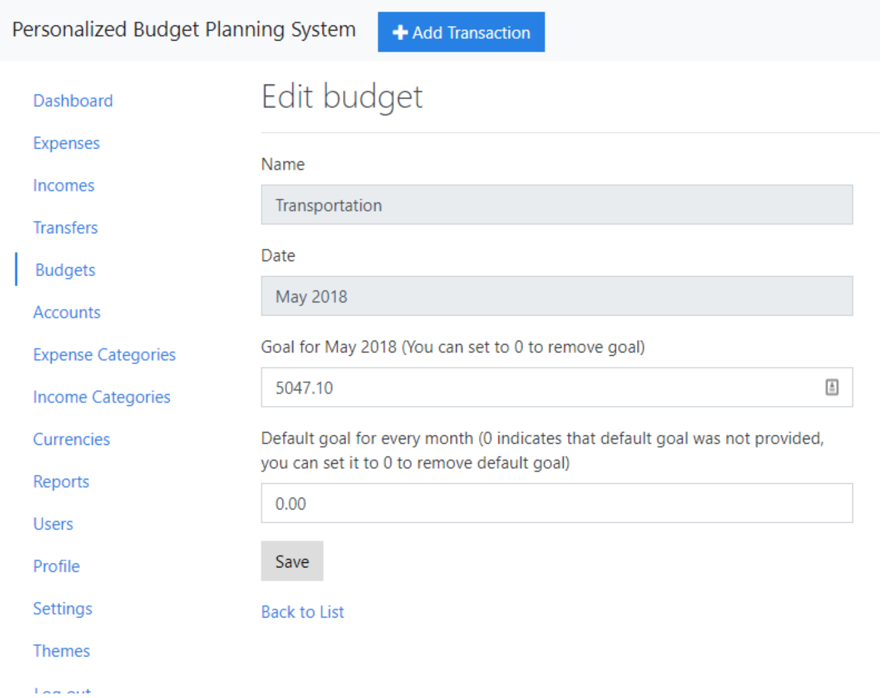


Figure 5-27 Budgets Page

2. The user can edit the budget by clicking on the edit link.
3. The edit budget page should be displayed.



Edit budget

Name

Transportation

Date

May 2018

Goal for May 2018 (You can set to 0 to remove goal)

5047.10

Default goal for every month (0 indicates that default goal was not provided, you can set it to 0 to remove default goal)

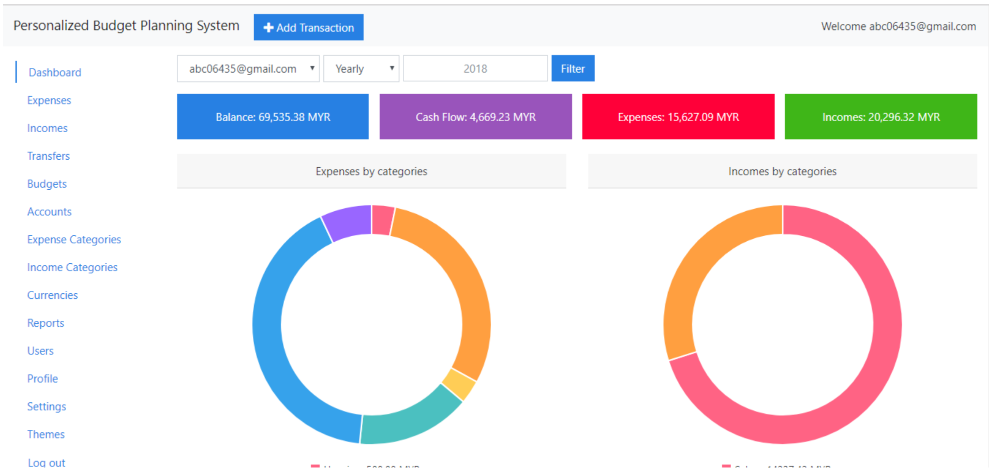
0.00

Figure 5-28 Edit Budget Page

4. After making necessary changes, click save to save the changes made.

5.4.4 Graph and Report Module

Dashboard



abc06435@gmail.com ▾ Yearly ▾ 2018

Balance: 69,535.38 MYR

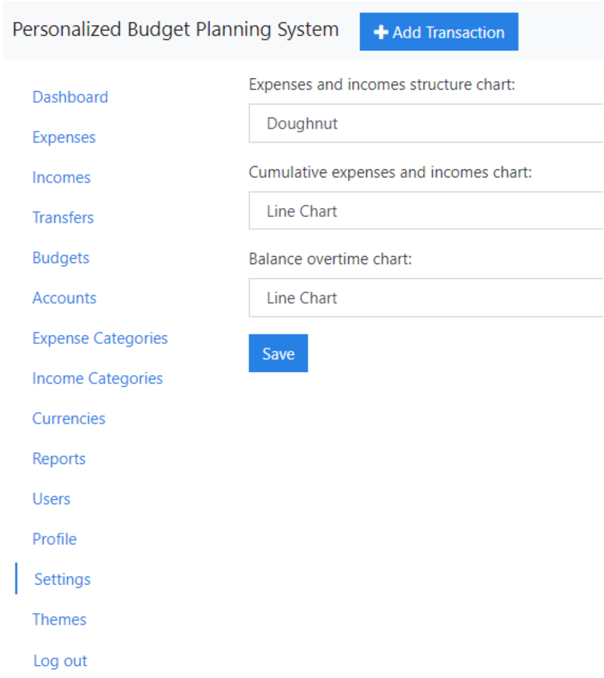
Cash Flow: 4,669.23 MYR

Expenses: 15,627.09 MYR

Incomes: 20,296.32 MYR

Figure 5-29 Dashboard Page

The dashboard is populated with various components. The user can change the dashboard chart settings by going to the settings page. The user can change the chart for expense and income structure from doughnut to bar chart, cumulative expenses and incomes from line chart to bar chart, and balance overtime from line chart to bar chart. The user can also supply different filters such as month/year, and user using the filter component. Only an admin can use user as a filter.



Expenses and incomes structure chart: Doughnut

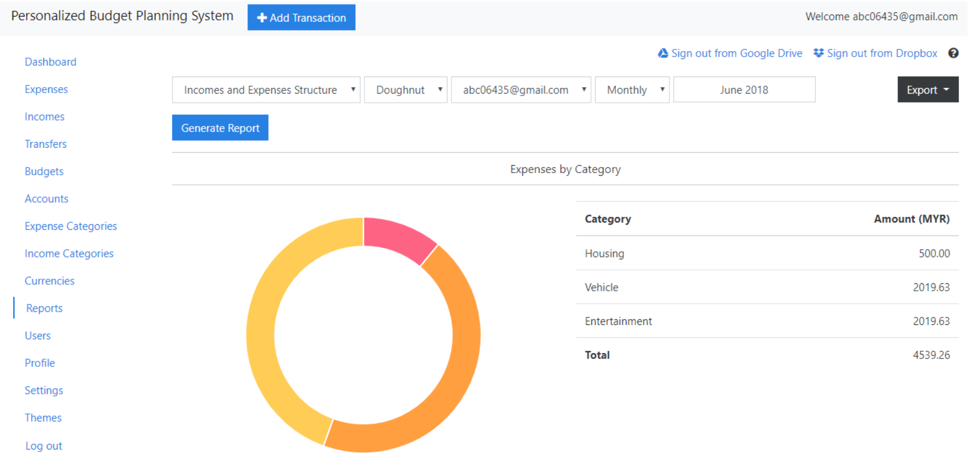
Cumulative expenses and incomes chart: Line Chart

Balance overtime chart: Line Chart

Figure 5-30 Dashboard Settings Pag

Report

You can generate report by going to the report page. You can use the first dropdown to select the report to generate, you can also select the chart to use for the report. Different types of report will have different charts available. You can click on the export button to select the approach you would like to export the report. It is possible to export report directly to third party cloud storage including Google Drive and Dropbox. Before you can export it to third party cloud storage, you will need to connect to the preferred cloud storage first. It is also possible to download the report directly or preview it.



Incomes and Expenses Structure ▾ | Doughnut ▾ | abc06435@gmail.com ▾ | Monthly ▾ | June 2018

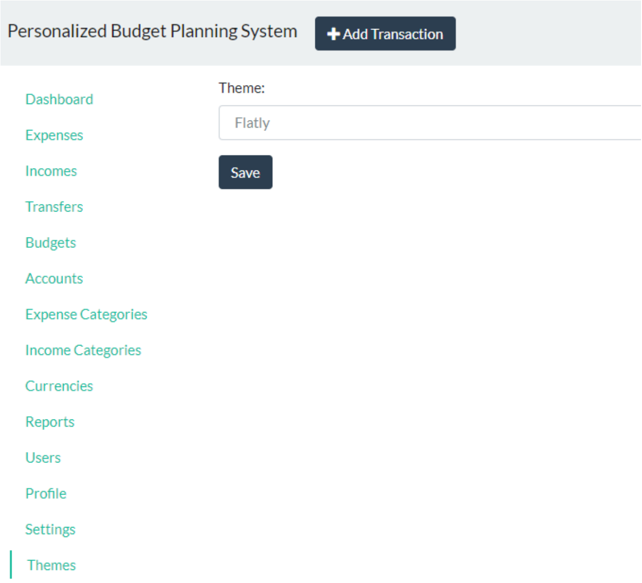
Expenses by Category

Category	Amount (MYR)
Housing	500.00
Vehicle	2019.63
Entertainment	2019.63
Total	4539.26

Figure 5-31 Report Page

5.4.5 Personalize Theme Module

You can change the theme of the application by going to the theme page. You can then select different theme to preview the theme and click on the save button to save your preference.



Theme: Flatly

Figure 5-32 Theme Selection Page

Chapter 6 Testing

Testing is an important part of system development to ensure the functionalities are working as expected. Various tests were conducted to make sure the system produces the correct output based on actions taken and to capture potential bugs. If there are bugs found when testing the system, the bugs will be fixed and the test case will be reconducted.

6.1 Register and Login Tests

Test Case	Action	Expected Result	Actual Result
1	Register by providing valid email address, password, and confirm password	The system adds a new user account to the database and sends a confirmation email to the user.	Same as expected result
2	Register by providing an email that already exist in the database and valid password and confirm password	The system informs the user that the email is already taken	Same as expected result
3	Register by proving a valid email address, but password and confirm password do not match	The system informs the user that the password and confirm password do not match	Same as expected result
4	Register by providing a password that do not fulfil the requirements of having at least one non-alphanumeric character, at least one lowercase ('a'-'z'), and at least one uppercase ('A'-'Z').	The system informs the user that the password does not meet requirements by providing password requirements information to the user.	Same as expected result
5	Login by providing the correct email and password combination and the email is already confirmed.	The system logins the user and redirect the user to the dashboard page.	Same as expected result

6	Login by providing the correct email and password but the email is yet confirmed.	The system does not login the user and returns error messages.	Same as expected result
7	Login by providing the wrong email and password combination.	The system does not login the user and returns error messages.	Same as expected result
8	User attempts to register via Google.	The system adds the user's Google identity to the database and logins the user directly.	Same as expected result
9	User attempts to login via Google	The system logins the user with their Google account.	Same as expected result
10	If user have 2 factor authentications enabled, the user will need to enter code available via Google Authenticator. User enters the code provided.	The system logins the user	Same as expected result
11	If user have 2 factor authentications enabled, the user will need to enter code available via Google Authenticator. User enters incorrect code.	The system informs the user that the code provided was incorrect	Same as expected result
12	If user have 2 factor authentications enabled, the user will need to enter code available via Google Authenticator. User choose to use alternative secret codes and provided valid secret code.	The system logins the user.	Same as expected result

13	If user have 2 factor authentications enabled, the user will need to enter code available via Google Authenticator. User provided incorrect secret code	The system informs the user the secret code was incorrect	Same as expected result

Table 6-1 Register and Login Tests

6.2 Accounts Tests

Test Case	Action	Expected Result	Actual Result
1	User tries to create an account with valid inputs.	The system adds a new account to the database. This account is now available when the user creates expenses, incomes, or transfers.	Same as expected result
2	User tries to create an account with invalid inputs.	The system informs the user the inputs provided were invalid.	Same as expected result
3	User tries to delete an account that belongs to the user.	The system deletes the account.	Same as expected result
4	User tries to delete an account that does not belong to the user.	The system does not delete the account.	Same as expected result
5	User list the accounts belong to the user.	The system displays a list of accounts belong to the user.	Same as expected result
6	User tries to update an account with valid inputs and the user is authorized to do so.	The system updates the account	Same as expected result
7	User tries to update an account with invalid inputs and the user is authorized to do so.	The system informs the user invalid inputs were provided	Same as expected result

8	User tries to update an account with valid inputs but is not authorized to do so.	The system redirects the user to the login page.	Same as expected result

Table 6-2 Accounts Tests

6.3 Currencies Tests

Test Case	Action	Expected Result	Actual Result
1	User tries to create a currency with valid inputs.	The system adds a new currency to the database. This currency is now available when the user wants to create accounts, expenses, incomes, and transfesrs.	Same as expected result
2	User tries to create a currency with invalid inputs.	The system informs the user the inputs provided were invalid.	Same as expected result
3	User tries to delete a currency that belongs to the user.	The system deletes the currency.	Same as expected result
4	User tries to delete a currency that does not belong to the user.	The system does not delete the currency.	Same as expected result
5	User list the currency belong to the user.	The system displays a list of currencies belong to the user.	Same as expected result
6	User tries to update a currency with valid inputs and the user is authorized to do so.	The system updates the currency.	Same as expected result
7	User tries to update a currency with invalid inputs and the user is authorized to do so.	The system informs the user invalid inputs were provided	Same as expected result

8	User tries to update a currency with valid inputs but is not authorized to do so.	The system redirects the user to the login page.	Same as expected result
9	User requested for conversion rate by clicking on the link.	The system returns the conversion rate via an API.	Same as expected result
10	User updates main currency from MYR to USD.	The system updates the user main currency to USD. Everything displayed on the dashboard will be in USD now.	Same as expected result

Table 6-3 Currencies Tests

6.4 Expenses Tests

Test Case	Action	Expected Result	Actual Result
1	User tries to create an expense with valid inputs.	The system adds a new expense to the database. The account that was selected had its balance deducted.	Same as expected result
2	User tries to create an expense with invalid inputs.	The system informs the user the inputs provided were invalid.	Same as expected result
3	User tries to delete an expense that belongs to the user.	The system deletes the expense and the balance of the account that was used to create this expense will be increased based on the expense amount.	Same as expected result
4	User tries to delete an expense that does not belong to the user.	The system does not delete the expense.	Same as expected result
5	User list the expenses belong to the user.	The system displays a list of expenses belong to the user.	Same as expected result

6	User tries to update an expense with valid inputs and the user is authorized to do so.	The system updates the expense. This will undo the action of deducting balance from previous account and update the balance of new account.	Same as expected result
7	User tries to update an expense with invalid inputs and the user is authorized to do so.	The system informs the user invalid inputs were provided	Same as expected result
8	User tries to update an expense with valid inputs but is not authorized to do so.	The system redirects the user to the login page.	Same as expected result

Table 6-4 Expenses Tests

6.5 Incomes Tests

Test Case	Action	Expected Result	Actual Result
1	User tries to create an income with valid inputs.	The system adds a new income to the database. The account that was selected had its balance increased.	Same as expected result
2	User tries to create an income with invalid inputs.	The system informs the user the inputs provided were invalid.	Same as expected result
3	User tries to delete an income that belongs to the user.	The system deletes the income and the balance of the account that was used to create this expense will be deducted based on the expense amount.	Same as expected result
4	User tries to delete an income that does not belong to the user.	The system does not delete the income.	Same as expected result

5	User list the incomes belong to the user.	The system displays a list of incomes belong to the user.	Same as expected result
6	User tries to update an income with valid inputs and the user is authorized to do so.	The system updates the income. This will undo the action of increasing balance from previous account and update the balance of new account.	Same as expected result
7	User tries to update an income with invalid inputs and the user is authorized to do so.	The system informs the user invalid inputs were provided	Same as expected result
8	User tries to update an income with valid inputs but is not authorized to do so.	The system redirects the user to the login page.	Same as expected result

6.6 Expense Categories Tests

Test Case	Action	Expected Result	Actual Result
1	User tries to create an expense category with valid inputs.	The system adds a new expense category to the database. This expense category will be available when the user wants to create expenses.	Same as expected result
2	User tries to create an expense category with invalid inputs.	The system informs the user the inputs provided were invalid.	Same as expected result
3	User tries to delete an expense category that belongs to the user.	The system deletes the expense.	Same as expected result

4	User tries to delete an expense category that does not belong to the user.	The system does not delete the expense category.	Same as expected result
5	User list the expense categories belong to the user.	The system displays a list of expense categories belong to the user.	Same as expected result
6	User tries to update an expense category with valid inputs and the user is authorized to do so.	The system updates the expense category.	Same as expected result
7	User tries to update an expense category with invalid inputs and the user is authorized to do so.	The system informs the user invalid inputs were provided	Same as expected result
8	User tries to update an expense category with valid inputs but is not authorized to do so.	The system redirects the user to the login page.	Same as expected result

Table 6-5 Expense Categories Tests

6.7 Income Categories Tests

Test Case	Action	Expected Result	Actual Result
1	User tries to create an income category with valid inputs.	The system adds a new income category to the database. This income category will be available when the user wants to create incomes.	Same as expected result
2	User tries to create an income category with invalid inputs.	The system informs the user the inputs provided were invalid.	Same as expected result
3	User tries to delete an income category that belongs to the user.	The system deletes the income category.	Same as expected result

4	User tries to delete an income category that does not belong to the user.	The system does not delete the income category.	Same as expected result
5	User list the income categories belong to the user.	The system displays a list of income categories belong to the user.	Same as expected result
6	User tries to update an income category with valid inputs and the user is authorized to do so.	The system updates the income category.	Same as expected result
7	User tries to update an income category with invalid inputs and the user is authorized to do so.	The system informs the user invalid inputs were provided	Same as expected result
8	User tries to update an income with valid inputs but is not authorized to do so.	The system redirects the user to the login page.	Same as expected result

Table 6-6 Income Categories Tests

6.8 Transfers Tests

Test Case	Action	Expected Result	Actual Result
1	User tries to create a transfer with valid inputs.	The system adds a new transfer to the database. The receiver account's balance is increased and the provider account's balance is decreased.	Same as expected result
2	User tries to create a transfer with invalid inputs.	The system informs the user the inputs provided were invalid.	Same as expected result
3	User tries to delete a transfer that belongs to the user.	The system deletes the transfer and balance of both accounts involved will be	Same as expected result

		updated to the state before the creation of the transfer.	
4	User tries to delete a transfer that does not belong to the user.	The system does not delete the transfer.	Same as expected result
5	User list the transfers belong to the user.	The system displays a list of transfers belong to the user.	Same as expected result
6	User tries to update a transfer with valid inputs and the user is authorized to do so.	The system updates the transfer. The system removes the side effect of the previous transfer and reapply the process of updating the accounts involved.	Same as expected result
7	User tries to update a transfer with invalid inputs and the user is authorized to do so.	The system informs the user invalid inputs were provided.	Same as expected result
8	User tries to update a transfer with valid inputs but is not authorized to do so.	The system redirects the user to the login page.	Same as expected result

Table 6-7 Transfers Tests

6.9 Budgets Tests

Test Case	Action	Expected Result	Actual Result
1	User tries to edit an existing budget by providing valid inputs and is authorised to do so. (Budgets can only be updated since budgets are based on expense categories)	The system updates the budget.	Same as expected result
2	User tries to edit an existing budget by providing invalid inputs and is authorised to do so.	The system informs the user the inputs provided were invalid.	Same as expected result

3	User tries to edit an existing budget by providing valid inputs and is not authorised to do so.	The system redirects the user to the login page.	Same as expected result

Table 6-8 Budget Tests

6.10 Theme Tests

Test Case	Action	Expected Result	Actual Result
1	User selects the theme via a dropdown list to preview the theme.	The system updates the user interface as the user selects the theme.	Same as expected result
2	User saves the theme.	The system updates the theme and use the theme selected on other pages as well.	Same as expected result

Table 6-9 Theme Tests

6.11 Report Tests

Test Case	Action	Expected Result	Actual Result
1	User select the report type, date range, and chart to use to generate the report.	The system generates the report based on criteria provided by the user.	Same as expected result
2	User exports the report to Google Drive when connected to Google Drive.	The system exports the report to Google Drive.	Same as expected result
3	User exports the report to Google Drive without being connected to Google Drive.	The system informs the user to connect with Google Drive first.	Same as expected result
4	User exports the report to Dropbox when connected to Dropbox.	The system exports the report to Dropbox.	Same as expected result
5	User exports the report to Dropbox when not connected to Dropbox.	The system informs the user to connect with Dropbo first.	Same as expected result
6	User previews the report in PDF version.	The system displays the report in PDF version with a new browser tab	Same as expected result

7	User downloads the report.	The system generates the PDF and instruct the browser to download the PDF to the user’s machine.	Same as expected result

Table 6-10 Report Tests

6.12 Dashboard Settings Tests

Test Case	Action	Expected Result	Actual Result
1	User updates the dashboard settings.	The system updates the dashboard settings. The dashboard display components based on the updated settings.	Same as expected result

Table 6-11 Dashboard Settings Tests

6.13 Users Tests

Test Case	Action	Expected Result	Actual Result same as Expected Result
1	User that is authorised displays a list of users.	The system displays a list of users to the admin.	Same as expected result
2	User that is not authorised displays a list of users.	The system informs the user is not authorised to perform this action.	Same as expected result
3	User who is authorised deletes a user.	The system deletes all records associated with the user and deletes the user	Same as expected result
4	User who is not authorised deletes a user.	The system does not delete the user and informs the user he/she is not authorised.	Same as expected result

5	Authorised user updates a user’s role	The system updates the user’s role	Same as expected result
6	Not authorised user updates a user’s role	The system does not update the user’s role and inform the user that he/she is not authorised.	Same as expected result

Table 6-12 Users Tests

6.14 Profile Tests

Test Case	Action	Expected Result	Actual Result
1	User updates his/her email.	The system updates the user’s email and sent confirmation email to new email.	Same as expected result
2	User updates the password by providing valid inputs	The system updates the user’s password	Same as expected result
3	User enables 2 factor authentications by providing the correct code.	The system enables 2 factor authentications.	Same as expected result
4	User enables 2 factor authentications by providing incorrect code.	The system informs the user that the code provided was wrong.	Same as expected result
5	User disables 2 factor authentications.	The system disables 2 factor authentications. When the user logins they are no longer required to enter 2 factor authentication code	Same as expected result

Table 6-13 Profile Tests

Chapter 7 Conclusion

In conclusion, the project was created as a web application and the mentioned functionalities were achieved. Some features that were mentioned in problem statements such as 2 factor authentications, theme personalization, exporting report to third party cloud storage, and implementation of keyboard shortcuts were completed as well. There are some drawbacks of the system including the lack of a calendar like component to display the transactions amount made which was achieved by Mobills. This project also does not provide a mobile application which is more convenient for performing common tasks, the user will not be able to sign up using their Facebook account as well. This project was mainly created for managing personal budgets, thus functionalities such as managing transactions between users were not implemented.

References

Anderson, R., 2016. *Introduction.*
Available from:
<https://docs.microsoft.com/en-us/aspnet/core/security/authorization/introduction>.
[14 February 2018].

Anderson, R., 2017. *Working with SQL Server LocalDB.*
Available from:
<https://docs.microsoft.com/en-us/aspnet/core/tutorials/first-mvc-app/working-with-sql?tabs=aspnetcore2x>. [14 February 2018].

budgetbakers, n.d. *budgetbakers.*
Available from:
<https://budgetbakers.com/>. [25 November 2017].

Daniel Roth, Rick Anderson, Shaun Luttin, 2017. *ASP.NET Core.*
Available from:
<https://docs.microsoft.com/en-us/aspnet/core/>. [14 February 2018].

Ghahrai, A., 2008. *Prototyping.* [Online]
Available from: https://www.testingexcellence.com/prototyping-model-software-development/
[18 February 2018].

Global, G., n.d. *What is an application?.*
Available from:
<https://www.gcflearnfree.org/computerbasics/understanding-applications/1/>. [14 February 2018].

Goodbudget, n.d. *Goodbudget Help Center*.
Available from:
<https://help.goodbudget.com/customer/en/portal/topics/470675-getting-started-guide/articles>. [25 November 2017].

GoodBudget, n.d. *Home | GoodBudget.*
Available from:
<https://goodbudget.com/home>. [25 November 2017].

Guru99, 2018. *MIS Development Process with SDLC & Agile.*
Available from: https://www.guru99.com/mis-development-process.html
[18 February 2018].

Mobills, n.d. *mobillsapp.*
Available from:
<https://www.mobillsapp.com/>. [25 November 2017].

Ndegwa, A., 2016. *What is a Web Application?.*
Available from:
<https://www.maxcdn.com/one/visual-glossary/web-application/>. [February 14 2018].

Rouse, M., 2015. *authentication.*
Available from:
<http://searchsecurity.techtarget.com/definition/authentication>. [14 February 2018].

TechTerms, 2014. *Web Application.*
Available from:
<https://techterms.com/definition/web_application>. [14 February 2018].

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: Year 3 Trimester 3	**Study week no.: 1**
Student Name & ID: ANG ZI WEI, 15ACB07926	
Supervisor: Ms Chan Lee Kwun	
Project Title: Personalized Budget Planning System	

1. WORK DONE

Programming part:

-Updated multi currencies support

- The user can now select different currencies for different accounts (e.g. Account A is in MYR, Account B is in USD)

2. WORK TO BE DONE

Programming part:

-Testing multi currencies part (ensure account amount is correct after transactions CRUD)

Report part:

-Review previous report and identify requirements for fyp 2 report

3. PROBLEMS ENCOUNTERED

No problem at the current state.

4. SELF EVALUATION OF THE PROGRESS

Doing fine.

_________________________ _________________________

Supervisor's signature Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: Year 3 Trimester 3	**Study week no.: 2**
Student Name & ID: ANG ZI WEI, 15ACB07926	
Supervisor: Ms Chan Lee Kwun	
Project Title: Personalized Budget Planning System	

1. WORK DONE

Programming part:

-Updated multi currencies support

- The user can now create preferred currencies based on a list of available currencies. (e.g. When creating transactions, the dropdown will only be populated with preferred currencies instead of all available currencies)
- The user can now create transactions with different currencies (e.g. Create an expense that is of 50 USD)
- Updated database structure (expenses and incomes table will now have a foreign key connected to the currencies created by the user)

2. WORK TO BE DONE

Programming part:

-Implement keyboard shortcuts to the system

-Designing the generate report module

Report part:

-Review previous report and identify requirements for fyp 2 report

3. PROBLEMS ENCOUNTERED

No problem at the current state.

4. SELF EVALUATION OF THE PROGRESS

Doing fine.

Supervisor's signature

Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: Year 3 Trimester 3	**Study week no.: 3**
Student Name & ID: ANG ZI WEI, 15ACB07926	
Supervisor: Ms Chan Lee Kwun	
Project Title: Personalized Budget Planning System	

1. WORK DONE

- Keyboard shortcuts added (The user can now travel between pages using keyboard shortcuts, open a modal to create transaction with the 'N' key)
- The report generate module is partially completed, the user can generate report in pdf format and download it.

2. WORK TO BE DONE

- Add other charts to report module (e.g. The user can view the expenses and incomes structures using either pie chart or bar chart, currently, the user can only view the expenses structure in pie chart)
- Report module export to Google Drive and Dropbox

3. PROBLEMS ENCOUNTERED

No problem at the current state.

4. SELF EVALUATION OF THE PROGRESS

Doing fine.

_________________________ _________________________

Supervisor's signature Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: Year 3 Trimester 3	**Study week no.: 4**
Student Name & ID: ANG ZI WEI, 15ACB07926	
Supervisor: Ms Chan Lee Kwun	
Project Title: Personalized Budget Planning System	

1. WORK DONE

Programming Part

- The user can now export the report to third party cloud storage (e.g. export to Google Drive and Dropbox)

Report Part

- Reviewed and completed chapter 1 and chapter 2

2. WORK TO BE DONE

Programming Part

- Allowing user to change the theme of the application
- Implement 2 factor authentication (with Google Authenticator)

Report Part

- Complete the rest of the report (Chapter 3, 4, etc.)

3. PROBLEMS ENCOUNTERED

No problem at the current state.

4. SELF EVALUATION OF THE PROGRESS

Doing fine.

Supervisor’s signature

Student’s signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: Year 3 Trimester 3	**Study week no.: 5**
Student Name & ID: ANG ZI WEI, 15ACB07926	
Supervisor: Ms Chan Lee Kwun	
Project Title: Personalized Budget Planning System	

1. WORK DONE

Programming Part

- The user can now display different charts using a dropdown box on the report module. (e.g. For expenses and incomes structure, the user can either select doughnut or bar chart as the chart for the report)
- Added new keyboard shortcuts for tables, the user can now press the '1' key to edit the record and the '2' key to delete the record.

2. WORK TO BE DONE

Programming Part

- Allowing user to change the theme of the application
- Implement 2 factor authentication (with Google Authenticator)

Report Part

- Complete the rest of the report (Chapter 3, 4, etc.)

3. PROBLEMS ENCOUNTERE

No problem at the current state.

4. SELF EVALUATION OF THE PROGRESS

Doing fine.

Supervisor's signature

Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: Year 3 Trimester 3	**Study week no.: 6**
Student Name & ID: ANG ZI WEI, 15ACB07926	
Supervisor: Ms Chan Lee Kwun	
Project Title: Personalized Budget Planning System	

1. WORK DONE

Programming Part

- The user can now change the theme of the application.

Report Part

- Reviewed and updated use case diagram

2. WORK TO BE DONE

Programming Part

- Implement 2 factor authentication (with Google Authenticator)

Report Part

- Complete the rest of the report (Chapter 3, 4, etc.)

3. PROBLEMS ENCOUNTERED

No problem at the current state.

4. SELF EVALUATION OF THE PROGRESS

Doing fine.

__________________________ __________________________

Supervisor’s signature Student’s signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: Year 3 Trimester 3	**Study week no.: 7**
Student Name & ID: ANG ZI WEI, 15ACB07926	
Supervisor: Ms Chan Lee Kwun	
Project Title: Personalized Budget Planning System	

1. WORK DONE

Programming Part

- Resolved a bug in which the dashboard actually not displaying data accurately since not every record are in the same currency now. Updated the dashboard to ensure all records retrieved are converted to the main currency before other calculations (the previous dashboard state was no longer accurate since the database structure changed and not every record will be in the same currency)

Report Part

- Reviewed and updated database diagram (ERD)

2. WORK TO BE DONE

Programming Part

- Implement 2 factor authentication

Report Part

- Complete the rest of the report (Chapter 3, 4, etc.)

3. PROBLEMS ENCOUNTERED

No problem at the current state.

4. SELF EVALUATION OF THE PROGRESS

Doing fine.

__________________________ __________________________

Supervisor’s signature Student’s signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: Year 3 Trimester 3	**Study week no.: 8**
Student Name & ID: ANG ZI WEI, 15ACB07926	
Supervisor: Ms Chan Lee Kwun	
Project Title: Personalized Budget Planning System	

1. WORK DONE

Programming Part

- In the report section, the user can generate report using either monthly or yearly date range (previously the user can only select monthly as the date range)
- Implemented 2 factor authentication

Report Part

- Reviewed previous user manual (FYP 1)

2. WORK TO BE DONE

Programming Part

- Allow user to select monthly or yearly as date range on dashboard page

Report Part

- Complete the rest of the report (Chapter 3, 4, etc.)

3. PROBLEMS ENCOUNTERED

No problem at the current state.

4. SELF EVALUATION OF THE PROGRESS

Doing fine.

Supervisor's signature

Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: Year 3 Trimester 3	**Study week no.: 9**
Student Name & ID: ANG ZI WEI, 15ACB07926	
Supervisor: Ms Chan Lee Kwun	
Project Title: Personalized Budget Planning System	

1. WORK DONE

Programming Part

- User can now select monthly or yearly as date range on dashboard page

Report Part

- Updated Chapter 3 (System Design) with various diagrams (ERD, Use cases)

2. WORK TO BE DONE

Programming Part

- Testing various functionalities and write test results on report

Report Part

- Complete the rest of the report (Chapter 4, etc.)

3. PROBLEMS ENCOUNTERED

No problem at the current state.

4. SELF EVALUATION OF THE PROGRESS

Doing fine.

Supervisor's signature

Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: Year 3 Trimester 3	**Study week no.: 10**
Student Name & ID: ANG ZI WEI, 15ACB07926	
Supervisor: Ms Chan Lee Kwun	
Project Title: Personalized Budget Planning System	

1. WORK DONE

Programming Part

- Tested various functionalities and provided the test results on report

Report Part

- Draft report completed

2. WORK TO BE DONE

Programming Part

- Wrap up the program for presentation

Report Part

- Check if changes are required
- Upload to Turnitin

3. PROBLEMS ENCOUNTERED

No problem at the current state.

4. SELF EVALUATION OF THE PROGRESS

Doing fine.

________________________ ________________________

Supervisor’s signature Student’s signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: Year 3 Trimester 3	**Study week no.: 11**
Student Name & ID: ANG ZI WEI, 15ACB07926	
Supervisor: Ms Chan Lee Kwun	
Project Title: Personalized Budget Planning System	

1. WORK DONE

Programming Part

- Wrap up the program for presentation

Report Part

- Making changes to the report according to comments
- Upload to Turnitin

2. WORK TO BE DONE

Programming Part

- Populate sample data to the system

Report Part

- Upload to Turnitin
- Make changes to the report as required

3. PROBLEMS ENCOUNTERED

No problem at the current state.

4. SELF EVALUATION OF THE PROGRESS

Doing fine.

Supervisor's signature

Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: Year 3 Trimester 3	**Study week no.: 12**
Student Name & ID: ANG ZI WEI, 15ACB07926	
Supervisor: Ms Chan Lee Kwun	
Project Title: Personalized Budget Planning System	

1. WORK DONE

Programming Part

- Prepare data for initial presentation

Report Part

- Submit draft report soft copy

2. WORK TO BE DONE

Report Part

- Prepare hard copies and CDs

3. PROBLEMS ENCOUNTERED

No problem at the current state.

4. SELF EVALUATION OF THE PROGRESS

Doing fine.

_________________________ _________________________

Supervisor’s signature Student’s signature

Personalized Budget Planning System

Project Background

A web application that can be used to manage budget information of the users. The user can manage their budget information after registering an account. The system allows user to personalize the theme to use as well as providing keyboard shortcuts. The users can also enable 2 factor authentications to reduce potential risk for malicious users to login to their account.

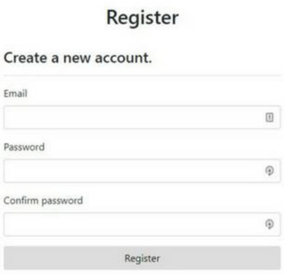
Problem statement

. The users were unable to save report directly to third party cloud storage such as Google Drive and Dropbox. The users were unable to select theme to change the appearance of the application. 2 Factor Authentications service were not provided for the systems. Keyboard shortcuts for performing common tasks were not provided

Methodology

The back end of the application was developed using Asp.NET Core 2.0 MVC using C#. The user interfaces are build using HTML, CSS, and JavaScript. Prototype modelling methodology was used. Gantt chart was used to ensure the application can be developed within specified period.

Results



Register

Create a new account.

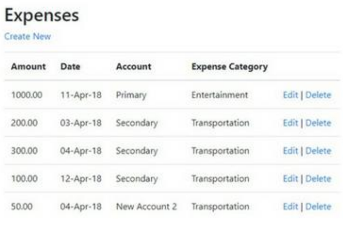
Email

Password

Confirm password

Identity Management

User can register, login, logout, and update their profile information



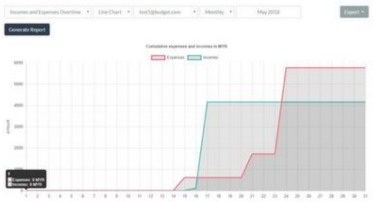
Expenses

Create New

Amount	Date	Account	Expense Category	
1000.00	11-Apr-18	Primary	Entertainment	Edit | Delete
200.00	03-Apr-18	Secondary	Transportation	Edit | Delete
300.00	04-Apr-18	Secondary	Transportation	Edit | Delete
100.00	12-Apr-18	Secondary	Transportation	Edit | Delete
50.00	04-Apr-18	New Account 2	Transportation	Edit | Delete

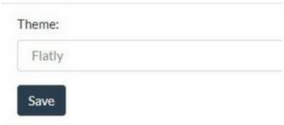
Transactions

User can manage expense, income, expense category, income category, transfer records



Graphs and Reports

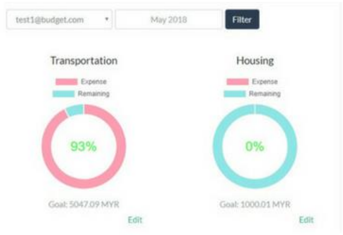
The system can display various charts to visualize the user's budget information.



Theme: Flatly

Personalization

The user can select different theme to change the appearance of the application. The user can also determine the chart to use on dashboard

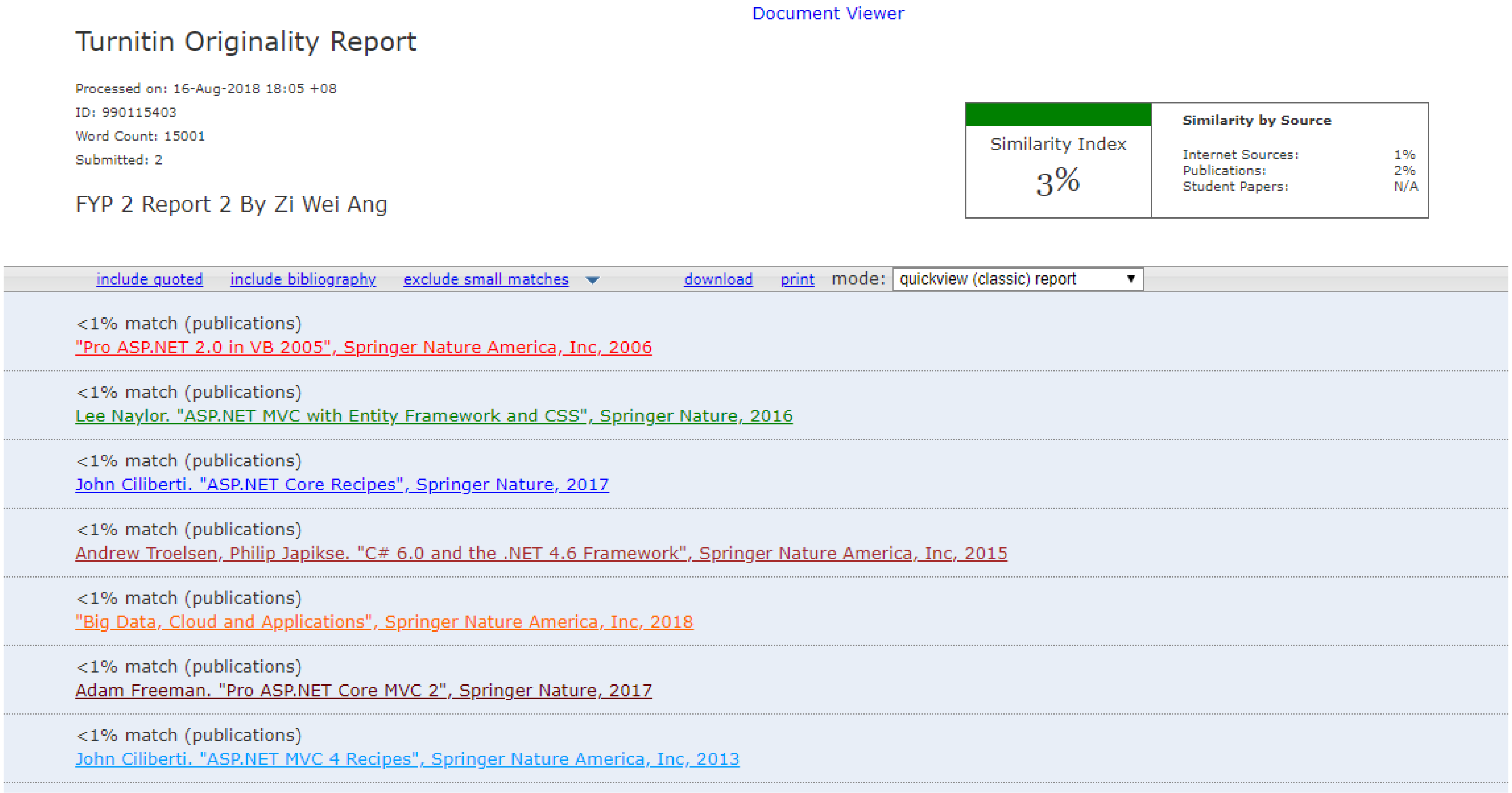


test1@budget.com May 2018

Budgets

User can create budget for a specific expense category. They can also update and delete the budget in the future.

PLAGIARISM CHECK RESULT



Document Viewer

Turnitin Originality Report

Processed on: 16-Aug-2018 18:05 +08

ID: 990115403

Word Count: 15001

Submitted: 2

FYP 2 Report 2 By Zi Wei Ang

Similarity Index

3%

Similarity by Source

Internet Sources:	1%
Publications:	2%
Student Papers:	N/A

mode: quickview (classic) report

<1% match (publications)
"Pro ASP.NET 2.0 in VB 2005", Springer Nature America, Inc, 2006

<1% match (publications)
Lee Naylor. "ASP.NET MVC with Entity Framework and CSS", Springer Nature, 2016

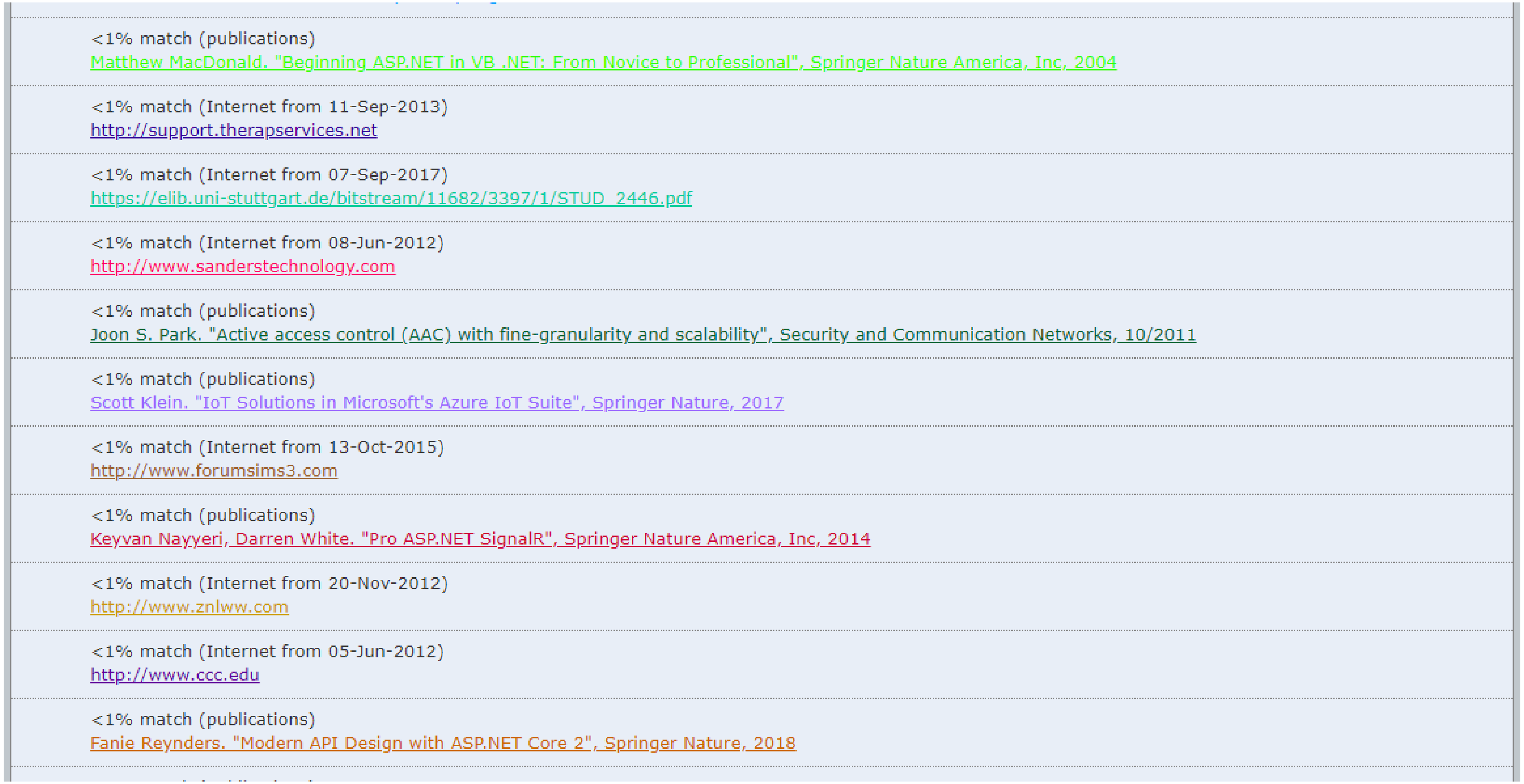
<1% match (publications)
John Ciliberti. "ASP.NET Core Recipes", Springer Nature, 2017

<1% match (publications)
Andrew Troelsen, Philip Japikse. "C# 6.0 and the .NET 4.6 Framework", Springer Nature America, Inc, 2015

<1% match (publications)
"Big Data, Cloud and Applications", Springer Nature America, Inc, 2018

<1% match (publications)
Adam Freeman. "Pro ASP.NET Core MVC 2", Springer Nature, 2017

<1% match (publications)
John Ciliberti. "ASP.NET MVC 4 Recipes", Springer Nature America, Inc, 2013



<1% match (publications)
Matthew MacDonald. "Beginning ASP.NET in VB .NET: From Novice to Professional", Springer Nature America, Inc, 2004

<1% match (Internet from 11-Sep-2013)
http://support.therapservices.net

<1% match (Internet from 07-Sep-2017)
https://elib.uni-stuttgart.de/bitstream/11682/3397/1/STUD_2446.pdf

<1% match (Internet from 08-Jun-2012)
http://www.sanderstechnology.com

<1% match (publications)
Joon S. Park. "Active access control (AAC) with fine-granularity and scalability", Security and Communication Networks, 10/2011

<1% match (publications)
Scott Klein. "IoT Solutions in Microsoft's Azure IoT Suite", Springer Nature, 2017

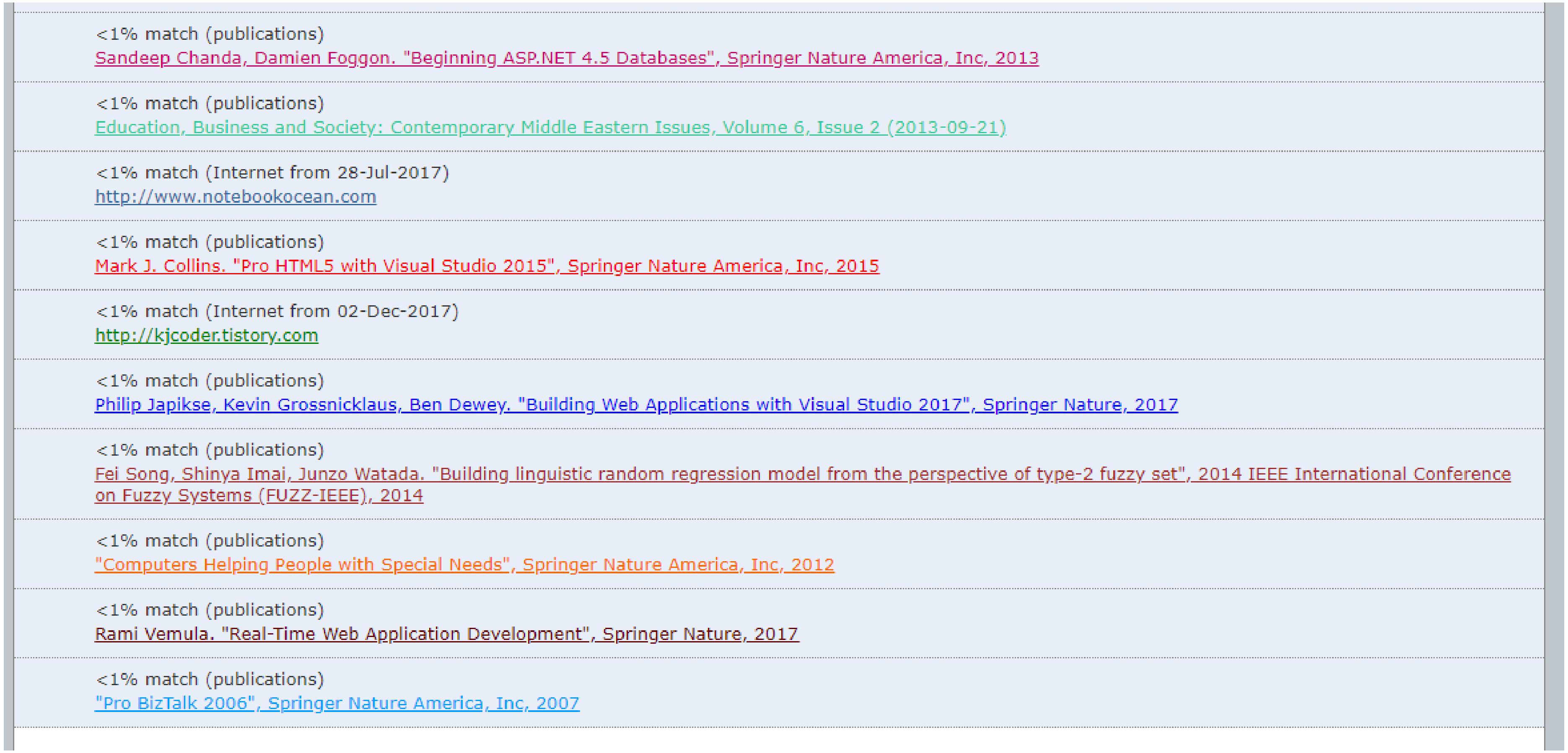
<1% match (Internet from 13-Oct-2015)
http://www.forumsims3.com

<1% match (publications)
Keyvan Nayyeri, Darren White. "Pro ASP.NET SignalR", Springer Nature America, Inc, 2014

<1% match (Internet from 20-Nov-2012)
http://www.znlww.com

<1% match (Internet from 05-Jun-2012)
http://www.ccc.edu

<1% match (publications)
Fanie Reynders. "Modern API Design with ASP.NET Core 2", Springer Nature, 2018



<1% match (publications)
Sandeep Chanda, Damien Foggon. "Beginning ASP.NET 4.5 Databases", Springer Nature America, Inc, 2013

<1% match (publications)
Education, Business and Society: Contemporary Middle Eastern Issues, Volume 6, Issue 2 (2013-09-21)

<1% match (Internet from 28-Jul-2017)
http://www.notebookocean.com

<1% match (publications)
Mark J. Collins. "Pro HTML5 with Visual Studio 2015", Springer Nature America, Inc, 2015

<1% match (Internet from 02-Dec-2017)
http://kjcoder.tistory.com

<1% match (publications)
Philip Japikse, Kevin Grossnicklaus, Ben Dewey. "Building Web Applications with Visual Studio 2017", Springer Nature, 2017

<1% match (publications)
Fei Song, Shinya Imai, Junzo Watada. "Building linguistic random regression model from the perspective of type-2 fuzzy set", 2014 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE), 2014

<1% match (publications)
"Computers Helping People with Special Needs", Springer Nature America, Inc, 2012

<1% match (publications)
Rami Vemula. "Real-Time Web Application Development", Springer Nature, 2017

<1% match (publications)
"Pro BizTalk 2006", Springer Nature America, Inc, 2007

Universiti Tunku Abdul Rahman			
Form Title : Supervisor's Comments on Originality Report Generated by Turnitin for Submission of Final Year Project Report (for Undergraduate Programmes)			
Form Number: FM-IAD-005	Rev No.: 0	Effective Date: 01/10/2013	Page No.: 1of 1

FACULTY OF INFORMATION AND COMMUNICATION TECHNOLOGY

Full Name(s) of Candidate(s)	ANG ZI WEI
ID Number(s)	15ACB07926
Programme / Course	IA
Title of Final Year Project	Personalized Budget Planning System

Similarity	**Supervisor's Comments** **(Compulsory if parameters of originality exceeds the limits approved by UTAR)**
Overall similarity index: 3 **%** **Similarity by source** Internet Sources: 1 % Publications: 2 % Student Papers: N/A %	
Number of individual sources listed of more than 3% similarity: 0	
Parameters of originality required and limits approved by UTAR are as follows: **(i) Overall similarity index is 20% and below, and** **(ii) Matching of individual sources listed must be less than 3% each, and** **(iii) Matching texts in continuous block must not exceed 8 words** ***Note: Parameters (i) – (ii) shall exclude quotes, bibliography and text matches which are less***	

Note Supervisor/Candidate(s) is/are required to provide softcopy of full set of the originality report to Faculty/Institute

Based on the above results, I hereby declare that I am satisfied with the originality of the Final Year Project Report submitted by my student(s) as named above

____
Signature of Supervisor
Name: ______________________

Date: ______________________

____
Signature of Co-Supervisor
Name: ______________________

Date: ______________________

UNIVERSITI TUNKU ABDUL RAHMAN

FACULTY OF INFORMATION & COMMUNICATION TECHNOLOGY (KAMPAR CAMPUS)

CHECKLIST FOR FYP2 THESIS SUBMISSION

Student Id	15ACB07926
Student Name	ANG ZI WEI
Supervisor Name	Ms. CHAN LEE KWUN

TICK (√)	**DOCUMENT ITEMS** Your report must include all the items below. Put a tick on the left column after you have checked your report with respect to the corresponding item.
	Front Cover
	Signed Report Status Declaration Form
	Title Page
	Signed form of the Declaration of Originality
	Acknowledgement
	Abstract
	Table of Contents
	List of Figures (if applicable)
	List of Tables (if applicable)
	List of Symbols (if applicable)
	List of Abbreviations (if applicable)
	Chapters / Content
	Bibliography (or References)
	All references in bibliography are cited in the thesis, especially in the chapter of literature review
	Appendices (if applicable)
	Poster
	Signed Turnitin Report (Plagiarism Check Result - Form Number: FM-IAD-005)

*Include this form (checklist) in the thesis (Bind together as the last page)

I, the author, have checked and confirmed all the items listed in the table are included in my report.

____

(Signature of Student)
Date:

Supervisor verification. Report with incorrect format can get 5 mark (1 grade) reduction.

____

(Signature of Supervisor)
Date: