**WITTY WALLET: A PERSONAL FINANCE**

**MANAGEMENT APPLICATION**

A Project Study

Presented to the Faculty of the

School of Information Technology

University of Baguio

In Partial Fulfilment

of the Requirements for the Degree

Bachelor of Science in Information Technology

By:

Lysle L. Baday

Anthony D. Ballug Jr.

Jonan V. Bie

Lyra Angelica S. Padua

Erna-kristi N. Martinez

December 2018

**APPROVAL SHEET**

This project study entitled **WITTY WALLET: A PERSONAL FINANCE MANAGEMENT APPLICATION** prepared and submitted by **LYSLE L. BADAY, ANTHONY D. BALLUG JR., JONAN V. BIE,** and **LYRA ANGELICA S. PADUA** in partial fulfilment of the requirements for the degree of **BACHELOR OF SCIENCE IN INFORMATION TECHNOLOGY**, has been examined and is recommended for acceptance and approval for oral examination.

**ERNA-KRISTI N. MARTINEZ, MIT**

Class Adviser/ Technical Adviser

**PROJECT STUDY COMMITTEE**

|  |  |
| --- | --- |
| **Cherry Ann A. Carpiso, MIT**  Member | **Engr. Elisabeth D. Calub, MSIT**  Member |
| **Erna-kristi N. Martinez, MIT**  Member | **Vanessa M. Vallejo, MIT**  Member |

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**PANEL OF EXAMINERS**

APPROVAL by the Committee on Oral Examination on \_\_\_\_\_\_\_\_\_\_\_ with a grade of \_\_\_\_\_\_\_\_\_\_.

|  |  |
| --- | --- |
| **DIVINE L. AGUILAR, BSIT**  Panel Member | **HYDI D. TOYENG, MIT**  Panel Member |
| **ENGR. ELISABETH D. CALUB, MSIT**  Panel Member | |

ACCEPTED AND APPROVED in partial fulfilment of the requirements for the degree of **BACHELOR OF SCIENCE IN**

**INFORMATION TECHNOLOGY.**

|  |  |
| --- | --- |
|  | E**NGR. ELISABETH D. CALUB**  Dean, SIT |

**Abstract**

One of the best solutions to financial problems is budgeting. Without budgets, problems, like getting into debt, running out of money, and not having money to spend in case of emergencies, may arise. Though budgeting is essential in a person's daily life, many people struggle in making one.

To assist people with their finances, the researchers developed Witty Wallet. Witty Wallet is a personal finance management application that lets the user keep track of their savings and expenses, allowing them to monitor their funds. What sets Witty Wallet apart is that it makes it possible for a user to automatically build a budget plan based on his or her previous expenditures through the use of machine learning.

The primary objective of the study was to design and develop a mobile application to aid users to manage their finances. Mainly, the objectives of the study were to identify the requirements for Witty Wallet; and to identify an appropriate supervised machine learning algorithm to be integrated into the application to suggest a budget plan.

The team utilized the rapid application development (RAD) methodology to build the mobile application. The methodology allowed the fast-paced prototyping and iterative testing of the application at an early stage.

Through the study, the developers have determined the requirements to design and develop the application which is reflected in the software requirements specification (SRS) and software design document (SDD). The researchers were also able to conclude that multiple linear regression is the most fitting supervised machine learning algorithm to apply to the given problem of suggesting a budget plan. The algorithm is best suited for the development of Witty Wallet since the application projects the user’s monthly budget based on their previous expenses, both variables having a linear relationship with one another.

**KEYWORDS**: machine learning, linear regression, finance, financial management, budgeting

**Acknowledgement**

The researchers would like to thank the almighty Father for the countless blessings, wisdom and guidance He has extended to the team all throughout the study.

The researchers would also like to express their deepest gratitude to the following who made the success of the study possible:

To Erna-kristi N. Martinez, the researcher’s technical adviser, for the never-ending support, assistance and inspiration she has given the group.

To the panel of examiners, Engr. Elisabeth D. Calub, Dr. Ellen M. Halover, Hydi D. Toyeng, and Divine L. Aguilar, for sharing their knowledge, insight, comments and recommendations that allowed the improvement of the study.

To Edgar Allan R. Macatulad, without the guidance and knowledge he has offered the researchers, Witty Wallet would not have been possible.

To Ralph Anthony N. Cecilio, for the endless support and kind gestures that contributed to the success of the study.

Lastly, the researchers would like to thank the University of Baguio’s School of Business Administration and Accountancy, particularly Sir John V. Balanquit, the testers, their friends, and every other individual who helped and supported the study.

**L. L. B**

**A. D. B. J**

**J. V. B**

**L. A. S. P**

**Table of Contents**

|  |  |
| --- | --- |
| [TITLE](#Title) PAGE 1 | |
| Approval Sheet 2 | |
| [Table of Contents](#TOC) 3 | |
| [List of Tables](#LOT) 6 | |
| [List of Figures](#LOF) 4 | |
| CHAPTER |  |
| 1 | [INTRODUCTION](#_Toc515637058)  [Background of the Study](#_Toc515637059) 10  [Objectives of the Study](#_Toc515637061) 16  [Scope and Limitations of the Study](#Scope) 16  [Significance of the Study](#SOTS) 18 |
| 2 | [REVIEW OF RELATED LITERATURE/SYSTEMS](#_Toc515637063)  20 |
| 3 | [METHODOLOGY](#_Toc515637066)  [Methodology 29](#_Toc515637067)  [Locale and Population](#Locale) 32  [Data Gathering Tools](#_Toc515637072) 33  [Data Gathering Procedures](#_Toc515637072) 34 |
| 4 | [DISCUSSION OF FINDINGS](#_Toc515637076)  38 |
| 5 | [CONCLUSSIONS AND RECOMMENDATIONS](#_Toc515637078)  [Conclusions](#_Toc515637079) 58  [Recommendations](#_Toc515637080) 60 |
| [REFERENCES](#References) 61 | |
| APPENDICES | |
| A | INTERVIEW REQUEST LETTER 72 |
| B | INTERVIEW GUIDE 73 |
| C | DATA SET SAMPLE 74 |
| D | FEEDBACK FORM 75 |
| E | SUS COMPUTATION 76 |
| F | DATA POLICY AND TERMS OF USE 77 |
| G | SYSTEM PROFILE 81 |
| H | SYSTEM REQUIREMENTS SPECIFICATION (SRS) 96 |
| I | TABLE OF FIELDS 141 |
| I | SYSTEM DESIGN DOCUMENT (SDD) 152 |
| J | TEST PLAN 188 |
| K | TEST CASES 199 |
| M | CURRICULUM VITAE 213 |

**List of Tables**

|  |  |
| --- | --- |
| **Table** |  |
| 1 | [Comparison of Selected Algorithms in the](#Tbl_1)  [Prediction of Data](#Tbl_1)  47 |
| 2 | [Functional Requirements for Witty Wallet](#Tbl_2)  49 |
| 3 | [Minimum Hardware and Software Specifications for](#Tbl_3)  [the Witty Wallet Application](#Tbl_3)  52 |
| 4 | [Witty Wallet Server Specifications](#Tbl_4)  55 |

**List of Figures**

|  |  |
| --- | --- |
| **Figure** |  |
| 1 | [Rapid Application Development](#Incremental)  29 |
| 2 | [CART](#Fig_2) 43 |
| 3 | [K-Nearest Neighbors (KNN)](#Fig_3) 44 |
| 4 | Simple [Linear Regression](#Fig_4) 45 |
| 5 | Comparison of SLR (left) and MLR (right) Model 51 |
| 6 | System Usability Test Results 56 |

**CHAPTER 1**

**INTRODUCTION**

This section covers the background, significance and the objectives and the scope of the study.

**Background of the Study**

Finance is referred to as the provision of money at the time when it is needed. Satyanarayana (2016) mentioned that finance may be defined as the art and science of managing money. It includes financial service and financial instruments.

Investopedia (n.d.) defined money management as the process of budgeting, saving, investing, spending or otherwise overseeing the capital usage of an individual or group. But it takes a lot of time, effort, risks and self-discipline to be able to manage money wisely. In a consumerist society, it is only practical to manage money well. According to Hill (2017), good financial sense, sticking to a monthly budget and living within their means is part of money management. Based on the findings from a quantitative research by Clarkson and Malton (2017), around 55% ranked managing money by the 470 respondents within their top three priorities in life at the moment. As one progress in his/her career and earn more money, understanding how to invest it wisely becomes important to reaching milestones such as having the down payment for a home.

There are tools to help an individual to manage their finances, one of those being a budget. A budget forecasts the financial results and financial position of a company for one or more future periods. At the most minimal level, a budget contains an estimated income statement for future periods (AccountingTools, 2017). In other words, a budget is an amount of money that can be spent and how it was spent. Thus, budgets should be ambitious but realistic (Harvard Business Review Staff, 2015). To prepare a budget, budgeting takes place.

BusinessDictionary (n.d.) defined budgeting as the process of expressing quantified resource requirements (amount of capital, amount of material, number of people) into time-phased goals and milestones. In the article "The Advantages of Budgeting in a Business", Siddiqui (2017) stated that a budget can be made for a person, family, group of people, business, government or a country which is used to ensure long-term financial security, planning, and foresight. An individual can keep his/her finances on track using a personal budget. Also, it is integral to establish a budget for any corporation or individual. According to Caldwell (2017), most people who do not have a budget end up overspending each month. Overspending may limit their spending power in the future as more and more of their salaries have to be applied to debt payments. As a proof, Cuaresma (2015), a BusinessMirror reporter, emphasized that Filipinos have difficulty keeping their budgets and lose a significant amount of their weekly budget to "mystery spending", a study conducted by a credit-card giant exposed. The repercussion behind this "mystery spending" that Filipinos face in general is alarming. It reflects on their nation's economy through their ways of spending.

There are many reasons why most Filipinos do not practice proper budgeting. Common reasons are awful spending habits, which often results in being financially restrained. Silang (2016), revealed that many Filipinos are familiar with the shortage of money within five or fewer days from the salary cut-off. And surprisingly, many in this situation survive it with debt or generosity of people around them. Filipinos know this as “Petsa de Peligro”. *Petsa de Peligro* translates to the day of danger, a phrase not often used in the literal sense but more to describe a period during the month when one’s cash on hand is running low and payday is still a week or so away (Silang, 2017). Likewise, based on Nielsen consumer habits study (2017), fifty-eight percent of respondents polled by Nielsen in the April to June period said they would save whatever is left after living expenses. Unfortunately, this is common to Filipinos as if it is a cultural norm. In particular, only 18 percent of Filipinos saw themselves as "thriving" financially, while the rest of the represented population said they are "struggling" or "suffering" in terms of economic security as written by Diola (2015) in the article “8 in 10 Filipinos 'struggling, suffering' financially”. This shows that it is significant to improve the Filipinos’ financial well-being.

Moreover, a survey conducted by Bangko Sentral ng Pilipinas in 2014 reveals that only 1 in 4 Filipino households have savings. The Philippines has one of the lowest savings rates in Southeast Asia (Featuredesk, 2017). Saving is an essential component of good budgeting (Ritchie, 2014). To be financially secured is through saving. Coincidentally, according to a survey done by rating agency Standard & Poor’s (2018), only one out of four Filipino adults is considered financially literate. Financial literacy is the possession of skills that allow people to make smart decisions with their money. Zucchi (2018) stated that financial literacy is crucial to help consumers save enough to provide adequate income in retirement while avoiding high levels of debt that might result in bankruptcy, defaults, and foreclosures. Being financially literate is having the ability to manage personal finances which is a valuable life skill. The place where everyone should start to build their financial literacy is through budgeting (Yardley, 2016).

To help address money management problems among Filipinos, the developers proposed a personal finance mobile application called Witty Wallet that would help its users manage their money. Witty Wallet features the necessary processes in money management such as tracking of expenses and income, setting financial goals and budgeting. A personal budget plan was created through the use of a machine learning algorithm.

Machine learning is a sub-part of Artificial Intelligence (AI) that enables programming applications to be precise in anticipating results without being programmed in detail (AppSquadz, 2018). According to Jain (2017), 39% of businesses use Artificial Intelligence and Machine Learning technologies in some form. Machine Learning enables businesses to effortlessly discover new trends and patterns from large and diverse data sets. Businesses can now automate analysis to interpret business interactions, which were traditionally done by humans, to take evidence-based actions. TechTarget (n.d.) stated that the basic premise of machine learning is to build algorithms that can receive input data and use statistical analysis to predict an output while updating outputs as new data becomes available. Considering ML as a strategic initiative can be a lucrative decision. However, deployment might carry certain business risk. It is better to approach investment decisions with utmost care (Flatworld Solutions, n.d.).

Machine Learning holds significant potential for improving financial planning, budgeting, and forecasting, particularly when combined with advanced mathematical techniques (Hagell, 2017). Faggella (2018) further supports this by stating that “Today, machine learning has come to play an integral role in many phases of the financial ecosystem, from approving loans to managing assets, to assessing risks. Machine learning has had fruitful applications in finance well before the advent of mobile banking apps, proficient chatbots or search engines”.

With the aid of Machine Learning, the developers integrated a machine learning algorithm that can suggest a budget based on the previous monthly expenses of the users. Generally, there are three different kinds of machine learning algorithms namely: supervised learning, unsupervised learning and reinforcement learning (Raschka, 2017). The researchers used the supervised learning method of machine learning. MathWorks (n.d.) defined supervised learning as a type of machine learning algorithm that uses a known dataset (called the training dataset) to make predictions. The training dataset includes input data and response values. As adaptive algorithms identify patterns in data, a computer "learns" from the observations. When exposed to more observations, the computer improves its predictive performance (MathWorks, n.d.). Sommer (2016) declares that machine learning tools will keep getting smarter. These tools will understand, with a growing amount of history and other data points, how the predictions or plans can be made even more accurate. Machine learning has several very practical applications that drive the kind of real business results – such as time and money savings – that have the potential to dramatically impact the future of the organization (Haffner, 2016). Instead of explicitly coding complex tasks to develop the application, this can be solved with the algorithm in Machine Learning.

Witty Wallet will not only help its users manage their money, but it could be a means for them to be disciplined, organized, and make wise decisions to achieve financial goals.

**Objectives of the Study**

The researchers’ objectives were to design and develop a mobile application that will provide a way to view and manage personal finances. Particularly, the objectives of the study are:

1. To identify the necessary functional and non-functional requirements for Witty Wallet; and
2. To identify an appropriate supervised machine learning algorithm to be integrated into the application to suggest a budget plan.

**Scope and Limitations of the Study**

The researchers created Witty Wallet with the intent to make the mobile application available for any user who is interested in utilizing the application’s features. Listed below is the scale to which the study is limited to.

The coverage of the study focused on users’ finances but did not include investments since more concepts are to be considered if the investment aspect of finance is to be included. Similarly, bearing in mind how the intended users live in different locations, the developers did not consider inflation rates. Incorporating inflation may require a separate research, most especially since inflation rates are not similar in various areas.

Moreover, Witty Wallet does not cater to businesses. As indicated by the title of the study, the application is meant to be for an individual’s personal use only. The application will not allow the users to connect bank accounts nor to transfer their data from other financial management or budgeting applications due to risk and security factors.

Although the application will not be built to be exclusive to the residents of Baguio City, the development team initially gathered data from the area. The researchers limited the collection of data and testing to the area due to the accessibility, time and cost constraints.

Furthermore, Witty Wallet is only available for Android-powered devices only. This is due to the limitations of deploying applications to the App Store. Witty Wallet is limited to the following functionalities: monitoring the user’s expenses and allowing the user to categorize each expense they enter into the application, providing an interface for the users to record their savings and goals as well as to allow the users to participate in saving challenges, such as “52 Week Challenge”, and “31 Day Challenge”. Additional features includes the recording of bills and reminders of due dates, and the allocation of a budget that a user may customize with the implementation of a machine learning algorithm that analyses the user’s previous expenditures in order to create a budget plan the user can customize. The application includes a supervised machine learning algorithm and did not utilize deep learning or neural networks.

**Significance of the Study**

Generally, the study aimed to provide users an application that allows them to manage their personal finances. Despite the need for budgets, people find it tedious to make budget. The study may benefit the users who struggle in making budgets for this reason. Witty Wallet provides the user with a tool that will allow them to automatically build a customizable budget plan based on their spending habits.

In line with helping users to manage their finances, Witty Wallet may also help people who tend to overspend through a feature that permits the users to list and categorize their expenses, allowing them to keep track of how much money they have spent and how much money is left, thus controlling and reducing their spending habit.

Aside from expenses, another key aspect of personal finance and budgets are savings. The study may help the user in saving for emergencies and other unexpected expenses that can make an impact in their finances by adding a function that gives the user a choice to add a certain amount to the emergency fund category each time a certain amount is left from the previous budget. To further encourage saving, the application also allows the user to add the total leftover fund from the monthly budget to other savings wallets the user can customize, aside from the emergency category.

In addition, the development team created a data policy agreement and an agreement regarding the terms of use of the application that will serve as a guide the users and would be users of the application. A full transcript of the stated agreements can be found on Appendix F.

Ultimately, the study also provides information that may be used as a foundation by future researchers in developing a similar application. The study likewise contain the essential information for researchers who wish to improve Witty Wallet. Additionally, the study benefits the researchers in expanding their knowledge and skills needed in building the team member’s future careers.

**CHAPTER 2**

**REVIEW OF RELATED LITERATURE/SYSTEMS**

This chapter presents the related systems, studies and other literature that was utilized in the development of the study.

**Related Literature/Systems**

With the help of modern technology, financial management has taken on an entirely different approach over the past few years. Various mobile applications have emerged to provide digital solutions to financial activities such as budgeting, saving, investing and much more. Presented below are examples of such applications and their features that were used as a basis in the development of Witty Wallet.

**Mint by Intuit Incorporated (2010).** Mint is a useful all-in-one resource for creating a budget through the use of hundreds of default categories that can be easily renamed or re-categorized by users, tracking spending and managing money where a user can connect bank and credit card accounts, as well as track monthly bills by providing alerts so that users can know when payments are due (Mint, n.d.). Similar to one of the stated features, Witty Wallet provides notifications to alert the user on their payment dues.

**You Need a Budget (2015).** You Need a Budget (YNAB) takes a unique approach to budgeting applications (Rosenberg, 2018). Rather than relying on traditional budgeting buckets, the application builds the budget based on a user’s income and breaks it down into categories. Witty Wallet utilizes this feature by allowing the user to budget by each expense category. Witty Wallet will also make use of YNAB's feature as a basis wherein a user can enter expenses manually that includes debt payoff and goal tracking features to help motivate them to reach their money goals.

**Oval Money by Oval Money Ltd (2015)**. Oval Money is an app that combines expense tracking, saving and investing in one application (Butcher, 2018). Oval employs machine learning that uses the random forest regression algorithm and works by analyzing the user's spending behavior, after which provides a saving plan which the user can make use of for meeting various financial goals. To make budgets less tedious, Witty Wallet integrated a machine learning algorithm that provides budget plan suggestions by analyzing the user's previous expenses.

The related systems presented above have common features which most personal finance apps have. One of those standard features which Witty Wallet adopted is the use of pre-set categories for expenses, income and savings that users can customize, re-categorize or tweak to meet their preferences and needs on using the application. All the applications also make use of notifications, so for better user experience, Witty Wallet also makes use of such feature of notifications and alerts to remind users of upcoming bills and payments. Timely and appropriate reminders are vital in keeping track of user spending and bills.

Discussed below are several studies, journals and other forms of literature that further supports the need for an application like Witty Wallet:

**The Process of Financial Planning in Personal Finance by Doda & Fortuzi (2015).** The key component of personal finance is planning, a process that requires monitoring at regular intervals. Doda & Fortuzi explained that with the help of organizing personal finances which are income and expense, a person could understand how much money is being spent and how much money is earned. By implementing personal finance budgeting concepts in the application, Witty Wallet will help individuals understand their financial situation more clearly, which is essential to be able to improve the stability of their finances.

**7 Reasons Why We Need to Start Budget Tracking Now by David Ning (2018).** To emphasize the importance of budgeting finances, Ning explained in his article that having a budget helps a person to be disciplined and organized in their finances, and that is the first step in determining their financial health.Budgets provide a structured plan that a person can follow, keeping them on track with their finances. Part of the budget tracking process is monitoring finances regularly in order to discover patterns and trends, as well as to determine areas that can still be improved. Such patterns can help a person build a plan that can maximize their income, thus avoiding the need for debt. In general, budgeting and knowing one’s financial stability can improve a person’s finances and is a necessity before progress can be made.

**Exploring the Intersection of Personal Finance and Technology – Full Podcast by Wharton University (2015).** Wharton University of Pennsylvania's interview supports the concept of crossing finance and technology with Elizabeth Diep, a partner with PWC's Asset Management Practice in New York City. Diep was asked a question on the meaning of intersection for the financial services sector (commercial banking to investment services) and technology, Diep had this to say:

"An interesting technology trend that I am sure you are all seeing is that numerous mobile apps are being developed every single day for smartphones...this is enabling individuals to manage their finances on their own without the need, in many instances, of a banker. … What this means for banks and financial advisors is that in today's technological world if the technology is not up to par, you run the risk of losing clients. And many of our clients, both when you look at investment services as well as commercial banks, know that and so they look to keep up the pace of change".

Diep’s interview answer regarding the financial sector and technology implies that mobile applications help people manage daily endeavors, she mentioned apps that manage personal finances without help from a professional. Diep also pointed that the use of technology in financial matters is an advantage, this reinforces the purpose of developing Witty Wallet. To keep up with the trend of technology on financial services, Witty Wallet will provide a platform for users to manage their finances through their mobile device.

**Budgeting Becoming Popular Thanks to Apps by Malcolm (2015).** Mobile devices, specifically smartphones have made it very convenient and much more comfortable for people to perform various tasks. Smartphones can now give people more insight into their financial habits through budgeting apps that can provide updates and relevant information regarding an individual's account balances, spending habits and progress towards financial goals. Basing from the Nielsen data survey conducted on April 2015, Malcolm asserted that of about 4000 smartphone users. Eighteen percent use smartphones to handle personal finance, and 69 percent of those agreed that budgeting apps helped changed their spending habits. The study proves that financial management has come a long way in helping users most especially now that financial management is available in a platform that is more accessible to users. Being a personal finance application for smartphones, Witty Wallet can help users in monitoring their expenses thus possibly changing their spending habits.

**Managing Your Finances with Budgeting Apps by Daily Monitor (2015).**  Daily Monitor Magazine states that one of the significant benefits of using a finance app is the ability to analyze finances and track spending patterns of an individual. Most of the budgeting apps allow users to compare income against spending, while others apps make use of categories separately. Other apps also provide a thorough analysis of the expenses with the use of charts and infographics which makes it easy for users to manage their money.

The use of personal budgeting apps helps individuals maintain a good practice of less spending and overall control of finances, Witty Wallet, being a personal finance management application, is in line with that purpose.

**The Next Phase of Machine Learning In Mobile App Development by Jain (2017).** To further enhance the user experience, Witty Wallet will utilize machine learning (ML) in the creation of budgets. Vipin Jain, CEO, and co-founder of Konstant Infosolutions validates the application of machine learning in mobile applications in his article. Jain discusses various industries that machine learning would be an essential part of, including Finance Management. Training a machine learning model that can be used to predict future trends in finance can be done by tracking previous transactions or user spending. The article states that a financial application with ML can provide suggestions on how to save and invest without involving human interaction and this is achievable through the use of machine learning.Since Witty Wallet is a mobile application for managing personal finance, the proponents would take advantage of ML technologies to perform budgeting automatically, thus saving time for the user.

**Machine Learning in Financial Services: 3 Potential Applications.** Other potential applications of machine learning in finance include customer service, personal finance, and fraud and risk management. Under personal finance, applications that are equipped with ML capabilities provide users the benefit of highly targeted financial recommendations and help them track and reduce insignificant spending. ML also provide valuable data on every aspect of spending and behavior (Frank, 2016). Applications that are equipped with machine learning algorithms lessen the manual effort from the user, one essential use of ML in finance is forecasting, and an example is trying to forecast future stock market prices using ML algorithms or models like Support Vector Machines (SVM).

Machine learning algorithms use computational methods to find and learn from existing patterns in data that can produce insight that can be used in several applications such as decision making, forecasting, and stocks (Mathworks, 2018). Machine learning and big data usually come hand in hand. Mathworks (2018) also states that ML algorithms adapt and enhance as the number of available data to learn from rises.

However, obtaining large amounts of data can be expensive and difficult to acquire (Johnson and Nguyen, 2017). Moreover, there are still situations wherein small data, datasets that contains specific sets of attributes (Kavis, 2015), can still arise.

**Learning from Little: Comparison of Classifiers by Cohen and Forman (n.d.).** The study “Learning from Little: Comparison of Classifiers Given Little Training” by Cohen and Forman compared the performance of machine learning models having small training data. The study proved that the various models excel in different situations thus giving future researchers a guide on choosing a model for a specific application. The results of the study also suggest that the Support Vector Machines (SVM) algorithm performed generally well given a small set of data in certain situations during the series of tests made in the study.

**Machine Learning for Predictive Modelling Based on Small Biomedical and Clinical Data by Shaikhina (2016).** Other studies also prove the possibility of using a smaller set of data. To conduct the study, the researchers generated a set of data from random numbers to still resemble the original data statistically in terms of mean, standard deviation and range. Data sets in the field of bioengineering are usually scarce, but the models used in the study were able to achieve an accuracy of 98.3% and 85% respectively despite the small dataset (35 bone specimens and 80 kidney transplants) used. Despite having a scarcity in the amount of data, the model was able to produce a precise outcome. Given the results of the study and the accuracy rates the model was able to yield, it suggests that it is possible for the proponents to use a smaller set of data to train the model to be used for Witty Wallet.

The studies mentioned above prove that it is not always necessary to use a large amount of data in order to implement machine learning. The studies also support the possibility of using a small amount of data in the development of Witty Wallet.

**CHAPTER 3**

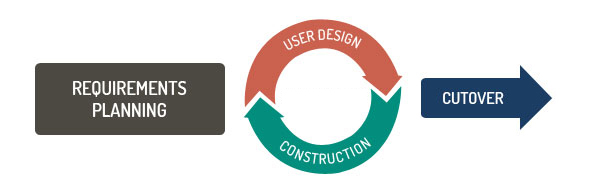
**METHODOLOGY**

This chapter presents the overall process by which the researchers conducted the proposed project.

**Methodology**

The rapid application development (RAD) methodology, as shown in Figure 1, was used for the study as the team developed and documented the application.

Rapid application development was coined in 1980 by James Martin at IBM. Rapid application development, according to TutorialsPoint (2018), is a methodology that emphasizes fast-paced prototyping. RAD allows the early and reiterative user testing of the designs. The method is ideal given the time constraints of the project since RAD is used when there is a need to create a product in a short amount of time.



*Figure 1. Rapid Application Development (TatvaSoft, 2015)*

The following phases were used to guide the developers in the creation of Witty Wallet:

**Requirements Planning**

During this stage of the methodology, the development team identified the application’s objectives and requirements. The objectives of the phase are to identify the problem to be solved, to define and finalize the requirements and the project scope.

The team conducted a research in order to identify problems related to the field of personal finance management. Research was also performed in order to identify some of the requirements needed in the development of Witty Wallet.

To assist the developers in defining some of the functional and non-functional requirements of the study, an interview was administered with the faculty of the School of Business Administration and Accountancy of the University of Baguio. Aside from a clear specification of requirements, other output for the phase include a System Requirements Specification (SRS) document, which can be found on Appendix H.

The project’s scope and requirements were agreed upon and was finalized by the developers and the panel. The developers did not involve other users during this phase of development. Users were engaged during the other phases of the development instead.

**User Design**

In this phase, the developers designed and developed prototypes of the application according to the requirements identified during the requirements gathering phase. The developers also created prototypes of the machine learning model, separate from the mobile application. The model was only integrated during later iterations.

Along with the mobile application, the developers also created prototypes of the machine learning model. Before beginning the development of the machine learning model, tests were performed to identify which algorithm to use in the development of the prototype. This was performed after the requirements definition and before creating the machine learning prototypes.

A Software Design Document (SDD), found on Appendix J, was created in this stage to accurately describe and elaborate the application’s design. The document also served as an overall guide in the development of the project. Along with the creation of the SDD, the phase also required the developers to finalize the system design before moving to construction**.**

**Construction**

The design described during the user design stage was implemented during the construction stage. The phase also required the developers to verify the system construction by testing the application in order to check the behavior of each function and ensure the quality of Witty Wallet before moving to the cutover or transition stage.

A copy of the system test plan and test cases can be found on Appendices H and I respectively.

During the iterations of the user design phases, users were asked to participate in focus group tests conducted in order to test the prototypes of the application created. The users test the prototype and provide feedback using the feedback form found on Appendix D. The comments and suggestions which the users provide were then considered during the construction of the next build.

**Cutover**

The goal of the cutover stage was to deploy the finished application in the intended deployment environment. The cutover phase also includes user training; however, the developers did not conduct user training since Witty Wallet does not have a specific stakeholder.

Discussed below are the methods, services and platforms that were used in the deployment of Witty Wallet.

**Witty Wallet Application Deployment.** The cutover phase required the developers to deploy Witty Wallet to a digital distribution platform. Witty Wallet was deployed in a cloud based file sharing platform.

**Witty Wallet Server.** The development team utilized Heroku; a cloud based platform that allows developers to deploy their applications in a real-time environment.

**Locale and Population**

Witty Wallet was created to cater to any user who is interested in utilizing the application’s features. However, due to the accessibility of the domain experts to be interviewed, the development team conducted the study within Baguio City only. Specifically, the researchers gathered data from the faculty of the School of Business Administration (SBAA) of the University of Baguio to obtain a reliable source for the information needed in the creation of Witty Wallet. Caused by the limited number of available respondents for the needed data, the development team gathered the necessary training and testing data from existing data sets that are appropriate for the study.

**Data Gathering Tools**

This part defines the tools used by the researchers to gather the necessary data needed in the creation of Witty Wallet.

* **Interview Guide.** The proponents created an interview guide to collect and verify the necessary information needed for the development of the project from the appropriate people. A copy of the interview guide can be found under Appendix A.
* **User Feedback.** Feedback was gathered from the participants involved in the focus group using a feedback form to assess the overall user experience of the application, which can be found in Appendix D. The feedback form is based on the System Usability Scale (SUS) created by John Brooke in 1986. SUS, as defined by Sauro (2016), is a tool used for administering usability tests and has been tested on hardware, software, websites, mobile phones, and so on, making it a reliable tool to gather user feedback.
* **List of Panel Suggestions**. The researchers utilized the panel's recommendations to enhance the study further.

**Data Gathering Procedures**

This section covers the processes used by the proponents in collecting the data required for the creation of Witty Wallet.

* **Interview**. The proponents conducted an interview to gather information that may be used for the creation of the application. The interview process is as follows:

1. Creation of the interview guide and policies

The researchers composed a set of questions as well as the terms and conditions to be observed during the interview. A copy of the questions and policies were then sent to the development team’s technical adviser for approval.

Listed below are terms and conditions that were considered for the interview:

* The participation of the interviewee is voluntary, and the data gathered from the interview is solely for academic purpose only, specifically for the study, Witty Wallet. The interviewee may opt out of the interview if he or she wishes to.
* The interview is recorded, and a transcript is to be produced. A copy of the transcript was sent to the interviewee for approval and or corrections before it is to be utilized by the researchers.

1. Creation, distribution, and signing of request letters

The researchers created a letter addressed to the dean of the School of Business Administration and Accountancy of the University of Baguio, requesting for permission to conduct an interview. A copy of the approved interview questions was attached to the request letter.

A copy of the interview guide and interview request letter can be found under Appendix A and Appendix B respectively

1. Interview Proper

After the approval of the interview request, an interview was immediately scheduled. The terms and policies were explained to the interviewee before the interview proper was started. A designated interviewer administered the questions listed in the interview guide while the other members recorded and transcribed the interviewee’s response (with permission from the interviewee). Follow-up questions were also administered to clarify or expound some information.

1. Approval of the transcript

A transcript was encoded based on the researcher’s notes and recordings. A copy was then sent to the interviewee for the verification. After the interviewee approves the transcript, the developers used the information gathered for the completion of the study.

* **Focus Group Discussions**. The team assembled a group composed of a minimum of 6-8 people inclusive of the developers to test the application, after each build. Aside from gathering suggestions from testers to enhance the current build, the focus group was also a means to collect the necessary information that was being used for future builds. A sample of the feedback form used to gather user rating and suggestions can be found on Appendix D.
* **Brainstorming and Group Meetings.** The team scheduled regular meetings to discuss suggestions, present problems encountered and to combine and associate ideas regarding the project. Meetings were done either face to face or through electronic means like instant messaging and SMS.
* **Consultations.** The project team set consultations with the technical adviser and other reviewers of the project to verify information and to seek advice and recommendations for the improvement of the project.
* **Panel Review**. The research was evaluated by the panel of examiners to guarantee the quality of the study and each deliverable included in the study.
* **Research**. The proponents conducted a research to find relevant information, tools and the like to establish a background for Witty Wallet and to further support the information related to the project. The research was also conducted to determine existing systems similar to the Witty Wallet, as well as their functions and features, which was used as a basis for the development of the application.

**CHAPTER 4**

**DISCUSSION OF FINDINGS**

This chapter presents the discussion of findings, a summary of the System Profile and the Software Requirements Specification of Witty Wallet.

Witty Wallet is a mobile application that can support users in their finances, more specifically in tracking their expenses, savings and budgets.

An interview was conducted with a faculty of the School of Business Administration and Accountancy of the University of Baguio. The results of the interview are discussed below:

**The Process of Creating a Personal Budget**

The first interview question asked the interviewee to walk the researchers through the process of creating a personal budget. The interviewee stated the following steps in creating a personal budget:

1. Start with identifying how much income is earned, then
2. Create a budget. Budgeting can go both ways; an individual may consider expenses immediately, or after receiving income, a portion is set aside as savings and the remaining amount is used for expenses.

The interviewee also stated another approach for budgeting called the petty cash concept. As an example, the interviewee mentioned the following:

“Example, in a given day, you have set that your expense will be P100. Once the P100 is spent, there should be no exceeding expenses. That is a possible option to make the approach simple.”

After stating an example, the interviewee also suggested that the application would notify the users if they are overspending.

A follow up question was also asked by the developers regarding the process of creating a budget. The developers asked if it was possible to approach budgeting in such a way wherein expenses and savings are recorded then the summation of both categories would be assumed to be the income instead of the traditional way of entering the income then breaking it down into expenses and savings. The interviewee affirmed that it was possible to use such approach and supported the idea further by discussing the “petty cash concept”.

Given the data gathered during the interview, the developers created an application that allows the user to create budgets for various expenses; as well as to record savings and create saving goals. The application also helps the user in monitoring their progress for each saving goal they set. Moreover, the development team integrated the interviewee’s suggestion of alerting the user if he or she has exceeded their allocated budget.

**The Elements of a Personal Monthly Budget**

From the interview, the developers learned that there are three main elements to personal budgeting. Namely, income, savings, and expense.

Savings, according to the interviewee, savings can be specified according to a person’s preferences. For example, a person may have savings for emergencies, leisure, education, retirement and so on. The interviewee specified that though savings may vary from person to person, the general concept of saving is for emergencies.

Similarly, the interviewee also states that expenses can also vary. It was mentioned during the interview that there are no specific titles for expense but a person may list their expenses according to what they usually spend on.

Given the aforementioned circumstance, the interviewee gave the following suggestion:

“Make the application as flexible as possible so users have the capability to create their own list of expenses and be able to determine what they can save.”

Based on the interview and the research conducted, the developers provided various categories under which users can classify their expenses. Since different people have different things to spend on, Witty Wallet gives the user the ability to specify a name for each wallet and transaction they create.

**Methods Used in Forecasting Budgets**

The development team also asked for methods that are used in forecasting a (personal) budget. It was ascertained that for personal budgets, the simplest approach would be the variance approach.

Using the variance approach, the user is expected to enter a budget then list their actual expenses. According to the interviewee, the difference between the budget and the actual expense is called variance. Furthermore, the interviewee states that the user can use the variance as a reference to adjust their next budget.

The developers also asked which method (of forecasting) would be best suited for the application. The interviewee answered that the variance approach should be considered instead of any complicated trend analysis which would be harder to execute with the given time constraint.

Taking the interviewee’s suggestion into account and along with the result of tests, to be discussed in the following parts of this chapter, during the research, the developers used variance as one of the independent variables in making a suggested budget plan for the user.

**Budgeting Model for Witty Wallet**

The interviewee suggested the “income minus savings is expense” method when asked to suggest a budgeting model that can be used in creating a personal finance management application. However, due to the response to the follow up question for the process of creating a budget stated above, the developers did not implement the suggested budgeting model. The developers took a different approach instead; the users are expected to create and allocate budgets for expense and saving wallets. The application does not require the users to enter their income, but in order for the users to check if they have exceeded the amount that they can budget, the application displays the total allocated budget.

**Supervised Machine Learning Algorithms**

Machine learning algorithms can be classified into three main groups, namely supervised, unsupervised and reinforced learning algorithms. Soni (2018) defines supervised machine learning algorithms as algorithms that aim to predict an output through uncovering relationships or patterns from a set of data. Supervised learning algorithms are taught by example - the algorithm is provided with training data which includes required input and output, and in turn, the algorithm determines a way to provide the desired output (Wakefield, n.d.).

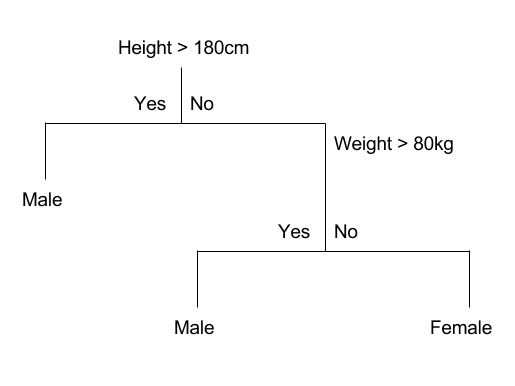
Elite Data Science (2017) identified three machine learning tasks under supervised machine learning namely, regression, classification, and clustering. The task of suggesting a budget plan, which is done by Witty Wallet, falls under regression.

Regression is a supervised machine learning task used for the prediction of continuous numerical data (Elite Data Science, 2017). Regression algorithms must understand the relationships between the variables in order to forecast or predict.

Listed in the following sections are regression algorithms which were considered for the selection of algorithms to be used for Witty Wallet.

1. **Classification and Regression Trees (CART)**

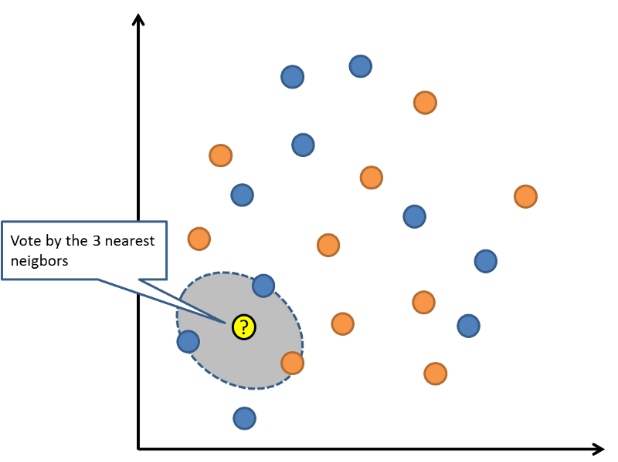
The Classification and Regression Trees (CART), also known as decision tree algorithm, creates models in a tree form from its training data in order to solve a classification or, in the case of Witty Wallet, a regression problem (Shubham, 2018).



*Figure 2. Decision Tree Example (Brownlee, 2016)*

1. **K-Nearest Neighbors (KNN)**

Predictions using the KNN algorithm are made using the entire training data. The prediction is done by searching the data set for K number of similar instances and summarizing the instances as an output for the prediction (Le, 2018).



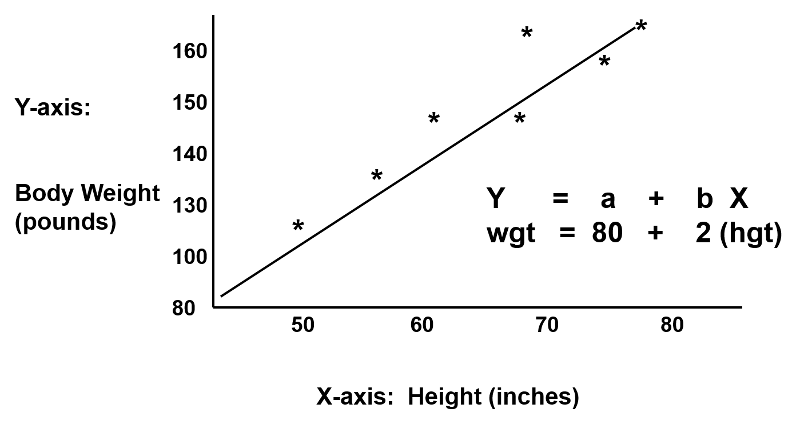
*Figure 3. K-Nearest Neighbors Example (Le, 2018)*

1. **Linear Regression**

According to Le (2018), linear regression may be the most popular and well understood statistical machine learning algorithm. Linear regression is used to predict a value (Y) from a predictor (X).

1. **Simple Linear Regression**

Simple Linear Regression is a statistical method that allows us to summarize and study relationships between two continuous quantitative variables: x and y (PennState, Eberly College of Science, n.d.). In Simple linear regression, the impact of one independent variable on the outcome are determined.



*Figure 4. Simple Linear Regression Example (Sullivan, 2016)*

1. **Multiple Linear Regression**

According to Boston University School of Public Health (2013), Multiple Linear Regression is an extension of simple linear regression analysis, used to assess the association between two or more independent variables and a single continuous dependent variable. The multiple linear regression equation is as follows:

MLR1.png,

Where Y-hat.png is the predicted or expected value of the dependent variable, X1 through Xp are p distinct independent or predictor variables, b0 is the value of Y when all of the independent variables (X1 through Xp) are equal to zero, and b1 through bp are the estimated regression coefficients.

**Selection of a Supervised Machine Learning Algorithm**

The development team conducted a test on the aforementioned algorithms to determine the most appropriate algorithm to adapt for the requirements in the creation of Witty Wallet. The results are presented in Table 1.

Table 1.

*Initial Comparison of Selected Algorithms in Predicting Data*

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Algorithm** | **Predicting Existing Data** | | | **Predicting New Data** | | |
| Input Data | Expected Output | Predicted Output | Input Data | Expected Output | Predicted Output |
| CART | 230.1 | 22.1 | 22 | 215.4 | 17.1 | 22 |
| KNN  (k = 3) | 230.1 | 22.1 | 18.5 | 215.4 | 17.1 | 18.5 |
| Linear Regression  (SLR) | 230.1 | 22.1 | 20.56… | 215.4 | 17.1 | 17.21… |
| Linear Regression  (MLR given 2 independent variables) | 230.1 | 22.1 | 22 | 215.4 | 17.1 | 17.14 |

A training data set from the http://www-bcf.usc.edu

/~gareth/ISL/Advertising.csv consisting of advertising data was used in order to conduct an initial assessment of the different algorithms.

From the initial assessment of algorithms, it was discovered that almost all the algorithms performed well in predicting data that already exists in the training data given. It was observed that the classification and regression tree (CART) performed the best during the prediction of existing data. However it was not able to predict data that is not in the data set; instead, it yielded the same results during the second test. The KNN algorithm was also able to provide a value that was close to the expected output; however, it was the multiple linear regression (MLR) model that presented the most accurate output.

The initial tests performed had a similar concept of predicting a dependent value given an independent value or values. But according to an article by Elite Data Science (2017), a theorem known as the “no free lunch theorem” states that no single algorithm works best for every problem, particularly for supervised learning problems like predictive modelling. The algorithm chosen must depend on the kind of problem being solved. In order to address this, further tests were required in order to determine how the algorithms performed given a data set related to the problem of predicting a budget based on expense.

During the second test, the algorithms were initialized with a person’s monthly budget and actual expense for food for half a year, a snippet of the data used can be found on Appendix C. The same process as the initial test was applied to the second assessment. Table 2 presents the results of the test conducted.

Table 2.

*Comparison of Selected Algorithms in Predicting Data*

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Algorithm** | **Predicting Existing Data** | | | **Predicting New Data** | | |
| Input Data | Expected Output | Predicted Output | Input Data | Expected Output | Predicted Output |
| CART | 7426 | 1619 | 1619 | 845 | 184 | 1140 |
| KNN  (k = 3) | 7426 | 1619 | 1619 | 845 | 184 | 1598 |
| Linear Regression  (SLR) | 7426 | 1619 | 1618.92… | 845 | 184 | 183.8593… |
| Linear Regression  (MLR) | 7426, 5807 | 1619 | 1619. Ō… | 845 | 184 | 183.9 |

Similar to the finding for the first test conducted, it was observed that all four algorithms were able to predict the expected output accurately. Alike the initial test, it was multiple linear regression that presented the closest value to the expected output during the prediction of data that does not exist in the data set.

From the results presented in Table 2, the algorithms were ranked from highest to lowest according to the accuracy of the data the algorithm was able to predict. The ranking is as follows: 1. Multiple Linear Regression, 2. Simple Linear Regression, 3. CART, and 4. KNN (given 3 k instances).

Based on the data presented in both Tables 1 and 2, the proponents found out that linear regression can predict both existing and new data quite well.

Thus, given the task of predicting a suggested monthly budget (the independent variable) for a user based on the user’s expenditure (the dependent variable) and the fact that linear regression performed well compared to the other algorithms, the developers selected linear regression as the most suitable algorithm to be used for the development of Witty Wallet.

Linear regression is one of the most common algorithms utilized to solve regression problems. As mentioned, linear regression predicts an output (Y), the dependent variable or target, based on a value (X), otherwise known as the independent variable or feature. The algorithm is used to model and predict variables that have a linear relationship, as the name suggests (Gandhi, 2018). The algorithm is best suited for the development of Witty Wallet since the application aims to suggest the user’s monthly budget based on their previous expenses, both variables having a linear relationship with one another.

As stated by Brownlee (2016), if there is a one independent variable (x) and one dependent variable (y), it is referred to as simple linear regression. Meanwhile if there is more than one independent variable (x), it is referred to as multiple linear regression.

To further enhance the output of the machine learning model, the developers compared simple linear regression (SLR) and multiple linear regression (MLR). The results of the aforementioned tests are shown in the figure below:



*Figure 5. Comparison of SLR (left) and MLR (right) Models*

It can be observed from the figure above that the MLR model has more accurate predictions as compared to the SLR model. It was also observed that the multiple linear regression model is more capable in adjusting when the changes in data are significantly large.

**Requirements and Design for the Development of Witty Wallet**

A purpose of the study was to determine the functional and non-functional requirements for the mobile application. The developers gathered necessary information by conducting an interview with the faculty of the School of Business Administration and Accountancy of the University of Baguio. Based from interviews and research done by the developers, the following are the requirements for Witty Wallet:

**User Requirements**

Presented below are the functional requirements for the development of the mobile application.

Table 3.

*Functional Requirements for Witty Wallet*

|  |  |
| --- | --- |
| **Use Case Name** | **Description** |
| Create Account | This allows the user to create an account. |
| View Profile | Allows the user to view his/her own nickname. |
| Edit Nickname | Allows the user to edit his/her own profile. |
| Login | This allows the user to login to the system |
| Logout | This allows the user to logout from the system |
| Forgot Password | This allows the user to recover his/her account by resetting the password |
| Change Password | This allows the user to update his/her password |
| Add Wallet | This facilitates the creation of an expense and savings wallet. |
| View Wallet List | Allows the user to view a list of his/her wallets. |
| Edit Wallet | This allows the user to update the selected wallet. |
| Delete Wallet | This allows the user to remove an existing wallet. |
| Add Transaction | This allows the user to add a transaction of an existing expense wallet. |
| Take Saving Challenge | This allows the user to participate in a saving challenge. Specifically the 52-Week Challenge and the 31-Day Challenge. |
| Deposit | This allows the user to deposit in a saving challenge. |
| Add Savings | This allows the user to add a transaction to a savings wallet. |
| Create Budget | This allows the user to create a monthly budget with the following options:   * automatic, * modified, and * manual   Both automatic and modified options utilize the machine learning model. |
| Project Budget | This allows the user to project his/her budget for the following month. |
| Budget Overview | This allows the user to view a report of the budget summary of the current month and previous months. |

Table 3 provides a brief description of the functional requirements gathered by the developers. The development team constructed Witty Wallet based on the data presented in the table above. Further discussions of the functional requirements can be found on the Software Requirements Specification, which is located in Appendix H of the document.

The minimum hardware and software requirements for the deployment environment are presented in Table 4.

Table 4.

*Minimum Hardware and Software Specification (Witty Wallet Application)*

|  |  |  |
| --- | --- | --- |
| **Device** | **Component** | **Specification** |
| **Tablet/ Smartphone** | **Operating System** | Android |
|  | **Version** | Android v 4.4 – KitKat or higher |
|  | **Free Space** | 11 MB |

Witty Wallet was created specifically for Android tablets or smartphones. As discussed in Table 4, the mobile device should be supported by Android version 4.4 (KitKat), or a higher version of the said operating system, and should have 9 megabytes of available space in order to install the application. The mobile device should also be connected to a network in order to access the Witty Wallet server.

The following table describes the Witty Wallet server specifications:

Table 5.

*Witty Wallet Server Specifications*

|  |  |
| --- | --- |
| **Component** | **Specification** |
| **CPU** | Single-core Shared |
| **Memory** | 500 MB |
| **Disk Space** | 500 MB |

The witty wallet server was deployed and hosted on Heroku. Heroku is a platform as a service (PaaS) that allows developers to build, run and operate applications on the cloud. The developers chose the platform since it can support various programming languages, including Node.js which was used to build Witty Wallet’s server.

**User Feedback for Witty Wallet**

The developers used the system usability scale (SUS) to gather user feedback for the mobile application. The feedback for the release build of Witty Wallet is discussed in this section.

Figure 6 System Usability Test Results

Figure 6 System Usability Rating for Witty Wallet

As shown in Figure 6, out of 30 respondents, ten percent (10%) of the respondents indicated that the usability performance of Witty Wallet is awful, thirty percent (30%) answered poor, none of the respondents answered okay, thirty-seven (37%) responded good, meanwhile twenty-three percent (23%) indicated that the system is excellent.

The developers calculated the SUS score for Witty Wallet, which resulted to 70.17. The general guideline on the interpretation of SUS scores indicates that Witty Wallet is good in terms of system usability. Attached in Appendix E of is the computation of the SUS score of Witty Wallet.

**CHAPTER 5**

**CONCLUSIONS AND RECOMMENDATIONS**

**Conclusions**

Managing one’s personal finance can help in tracking the flow of money. For instance, one can determine when to limit his/her expenditures if the budget is tight. Budgeting is a crucial aspect of financial management. Budgets can benefit a person in many different ways such as preventing overspending, helping a person save and prepare for an emergency. However, creating a budget can be a tedious task.

Witty Wallet is a mobile application for android devices that features primary elements of budgeting such as tracking a user’s savings and expenses, notifying users with regards to bill due dates, and other features. Witty Wallet also incorporates a machine learning algorithm that enables users to project a suggested budget plan for the following month. But given the results of the system usability assessment, it can be concluded that the system needs improvement in terms of the user experience.

From the various tests conducted by the development team, the researchers conclude that linear regression, specifically multiple linear regression, is the most appropriate supervised machine learning algorithm to be integrated into Witty Wallet to suggest a budget plan. The researchers were also able to determine the needed requirements in the development of the application, which was reflected in the system requirements specification and software design document.

Overall, Witty Wallet is a personal finance management application that can help users in managing finances. The machine learning model used in projecting budgets may benefit users who are new to budgeting. Since the integrated algorithm analyzes how much a user can spend in a month, the application is able to prevent users from over budgeting or under budgeting.

**Recommendations**

The developers of Witty Wallet recommend the following to future researchers for the development of the mobile application:

1. The exploration of other algorithms, optimization or other means to further enhance the predictions done by Witty.
2. The refinement of the application in terms of user experience and interface to enrich the overall feel of the mobile app.
3. The improvement of the application by adding image processing to scan receipts and automatically log transactions.
4. Making the app more engaging by adding more functions and features that will make Witty Wallet a more entertaining application.
5. The consideration of inflation and other factors in the prediction of a user’s budget.

**REFERENCES**

**Books**

Satyanarayana, P. S. (2016). *Financial Management*. New Delhi:

Namit Wasan.

**Electronic and Online Sources**

ABS-CBN News. (2017). Save less, shop more? Study shows how

Filipinos spend spare cash. Retrieved on June 10, 2018 from http://news.abscbn.com/business/08/31/17/save-less-shop-more-study-shows-how-filipinos-spend-spare-cash

ABS-CBN News. (2017). 5 Reasons Filipinos Don’t Save Money.

Retrieved on June 10, 2018 from http://news.abs-cbn.com/business/09/25/17/5-reasons-filipinos-don’t-save-money

AccountingTools. (2017, December 17). Budget definition. Retrieved on June 20, 2018 from https://www.accountingtools.com/articles/what-is-a- budget.html

AppSquadz. (2018, February 6). MACHINE LEARNING - THE FUTURE OF APP DEVELOPMENT TECHNOLOGY. Retrieved on June 19, 2018 from https://dev.to/appsquadz/machine-learning---the-future-of-app-development-technology- 4djh

Bell, A. (2017). 6 Reasons Why You Need a Budget. Retrieved on

June 9, 2018 from <https://www>.investopedia.com/financial-

edge/1109/6-reasons-why-you-need-a-budget.aspx

Boston University School of Public Health. (2013). Multiple

Linear Regression Analysis. Retrieved on December 11, 2018

from http://sphweb.bumc.bu.edu/otlt/MPH-

Modules/BS/BS704\_Multivariable/BS704\_Multivariable7.html

Brownlee, J. (2016). Linear Regression for Machine Learning.

Retrieved on December 7, 2018 from https://machinelearningmastery.com/linear-regression-for-machine-learning/

Brownlee, J. (2017). Difference between Classification and

Regression in Machine Learning. Retrieved on June 21, 2018 from https://machinelearningmastery.com/

classification-versus-regression-in-machine-learning/

BusinessDictionary. (n.d.). Budgeting. Retrieved on June 18, 2018

From http://www.businessdictionary.com/

definition/budgeting.html

Butcher, M. (2018). Oval Money app launches its investment

products for Millennials. Retrieved on June 21, 2018 from https://techcrunch.com/2018/06/18/oval-money-app-launches-its-investment-products-for-millennials/

Caldwell, M. (2018). Planning for Financial Emergencies.

Retrieved on June 9, 2018 from https://www.thebalance.com/

planning-for-financial-emergencies-2385813

Caldwell, M. (2017). Reasons Why You Should Budget Your Money.

Retrieved from <https://www>.thebalance.com/reasons-to-

budget-money-2385699

Caldwell, M. (2017). Reasons Why People Do Not Budget.

Retrieved on June 9, 2018 from https://pocketsense.com/

reasons-people-do-not-budget-7912296.html

Clarkson, T. & Malton, C. (2017). Young Adults and Money Management: behaviours, attitudes and useful rules of thumb. Retrieved on June 27, 2018 from http://britainthinks.com/pdfs/Young\_Adults\_and\_Money\_Man agement\_-FINAL.pdf

Cuaresma, B. (2015). Filipinos can’t stick to budget due to

‘mystery spending’–survey. Retrieved from

<https://businessmirror>.com.ph/filipinos-cant-stick-to- budget-due-to-mystery-spending-survey/

Cuckow, P. (2014). Using Machine Learning to Predict Personal Expenditure. Retrieved on June 21, 2018 from http://www.pezcuckow.com/portfolio/reports/final-year- project.pdf

Daily Monitor (2018, Apr 21). Managing Your Finances with

Budgeting Apps. Retrieved on June 26, 2018 from https://bit.ly/2KuH4J1

DF Team (2017). Real-life Applications of SVM (Support Vector

Machines). Retrieved on June 21, 2018 from https://data-

flair.training/blogs/applications-of-svm/

Diola, C. (2015). 8 in 10 Filipinos 'struggling, suffering' financially. Retrieved one June 21, 2018 from https://www.philstar.com/headlines/2015/03/25/1437430/8-10-filipinos-struggling-suffering-financially

Elite Data Science (2017). Modern Machine Learning Algorithms:

Strengths and Weaknesses. Retrieved on October 3, 2018

from https://elitedatascience.com/machine-learning-algorithms

Faggella, D. (2018). Machine Learning in Finance – Present and

Future Applications. Retrieved on June 19, 2018 from <https://www.techemergence.com/machine-> learning-

in-finance/

Featuresdesk (ICG). (2018). PSBank I Good To Know: How Financially Savvy Are You? Retrieved from http://pageone.ph/psbank-i-good-to-know-how-financially- savvy-are-you/

Flatworld Solutions. (n.d.). 8 ways businesses can benefit from machine learning. Retrieved on June 19, 2018 from <https://www.outsource2india.com/software/articles/busi> nesses-benefits-machine-learning.asp

Fumo, D. (2017). Types of Machine Learning Alogorithms You

Should Know. Retrieved on July 4, 2018 from

https://towardsdatascience.com/types-of-machine-learning-

algorithms-you-should-know-953a08248861

Gandhi, R. (2018). Introduction to Machine Learning Algorithms:

Linear Regression. Retrieved on October 3, 2018 from

https://towardsdatascience.com/introduction-to-machine-

learning-algorithms-linear-regression-14c4e325882a.

Haffner, P. (2016). What is Machine Learning – and Why is it Important?. Retrieved on July 11, 2018, from https://resources.interactions.com/machine-learning- important/

Hagell, D. (2017). Machine Learning the Financial Planning, Budgeting, and Forecasting models. Retrieved on June

28, 2018 from https://www.linkedin.com/pulse/machine- learning-financial-planning-budgeting-models-donald- hagell

Harvard Business Review Staff. (2015). The Right Way to Prepare Your Budget. Retrieved on June 27, 2018 from https://hbr.org/2015/07/the-right-way-to-prepare-your- budget

Hill, B. (2017). Importance of Money Management. Retrieved on

June 18, 2018 from https://budgeting.thenest.com/

importance-money-management-3204.html

Investopedia. (n.d.). Money Management. Retrieved on June 18,

2018 from https://www.investopedia.com/terms/m/

moneymanagement.asp

Jain, V. (2017). BCW. *The Next Phase of Machine Learning in*

*Mobile app Development.* Retrieved on June 27, 2018 from

https://bit.ly/2tLZL0n

Jain, V. (2017). The Next Phase of Machine Learning In

Mobile App Development. Retrieved on June 19, 2018

from http://www.businesscomputingworld.co.uk/

the-next- phase-of-machine-learning-in-mobile-app-development/

Kavis, M. (2015). Forget Big Data -- Small Data Is Driving The

Internet of Things. Retrieved on July 12, 2018 from https://www.forbes.com/sites/mikekavis/2015/02/25/forget-big-data-small-data-is-driving-the-internet-of-things/#3b0e2b2a5d7e.

Le, J. (2018). A Tour of the Top 10 Algorithms for Machine

Learning Newbies. Retrieved on October 3, 2018 from

https://towardsdatascience.com/a-tour-of-the-top-10-

algorithms-for-machine-learning-newbies-dde4edffae11.

Malcolm, H. (2015). USA Today. *Budgeting Becoming Popular*

*Thanks to Apps.* Retrieved on June 27, 2018 from

https://usat.ly/2yOweZF

MathWorks. (n.d.). Machine learning technique for building predictive models from known input and response data. Retrieved on June 28, 2018 from https://www.mathworks.com/discovery/supervised- learning.html

MathWorks. (n.d.). Supervised Learning Workflow and

Algorithms. Retrieved on June 28, 2018 from

https://www.mathworks.com/help/stats/supervised-learning-

machine-learning-workflow-and-algorithms.html

Mint (n.d.). How Mint Works. Retrieved on June 21, 2018 from

https://www.mint.com/how-mint-works

PennState, Eberly College of Science. (n.d.). What is Simple

Linear Regression? Retrieved on December 11, 2018 from

https://onlinecourses.science.psu.edu/stat501/node/251/

Raschka, S. (2017). 3 Different Types of Machine Learning

Algorithms. Retrieved on June 28, 2018 from https://www.kdnuggets.com/2017/11/3-different-types-machine-learning.html

Ritchi, A. (2014). Financial Literacy. Retrieved from https://www.annuity.org/financial-literacy/Siddiqui, F.

(2015). The Advantages of Budgeting in a Business.

Retrieved from https://www.linkedin.com/pulse/advantages-

budgeting-business-fareed

Rosenberg, E. (2018). The Best 8 Budgeting Apps to Download

in 2018. Retrieved on June 21, 2018 from https://www.thebalance.com/best-budgeting-apps-4159414

Shubham, J. (2018). Decision Trees in Machine Learning.

Retrieved on October 3, 2018 from https://becominghuman.

ai/decision-trees-in-machine-learning-f362b296594a

Siddiqui, F. (2015). The Advantages of Budgeting in a

Business. Retrieved on June 10, 2018 from

https://www.linkedin.com/pulse/advantages-budgeting- business-fareed/

Silang, K. (2016). Choose not to have Petsa de Peligro.

Retrieved from http://www.manilatimes.net/choose- not-to-

have-petsa-de-peligro/279124/

Silang, K. (2017). Breaking Free From Petsa De Peligro.

Retrieved from https://www.imoney.ph/articles/prevent-

survive-petsa-de-peligro/

Soni, D. (2018). Supervised vs. Unsupervised Learning.

Retrieved on October 3, 2018 from https://towardsdatascience.com/supervised-vs-unsupervised-learning-14f68e32ea8d

Sommer, B. (2016). Anticipatory budgeting, planning and forecasting – a radical future. Retrieved on June 27, 2018

from https://diginomica.com/2016/06/30/budgeting- planning- and-forecastings-radical-future/

Statistics Solution (2018). What is Logistic Regression.

Retrieved on October 3, 2018 from http://www.statisticssolutions.com/what-is-logistic-regression/

Sullivan, L. (2016). Simple Linear Regression. Retrieved on

December 11, 2018 from http://sphweb.bumc.bu.edu/otlt/MPH-

Modules/BS/BS704-EP713\_MultivariableMethods/

Sund, A. (2015). Creating the Machine Learning model.

Retrieved on July 10, 2018 from https://arnesund.com/

tag/linear-regression/

TechTarget. (n.d.). Machine learning (ML). Retrieved on

June 19, 2018 from https://searchenterpriseai.techtarget

.com/definition/machine-learning-ML

Thomas, N. (2015). How to Use the System Usability Scale (SUS)

to Evaluate the Usability of Your Website. Retrieved on

December 7, 2018 from https://usabilitygeek.com/how-to-

use-the-system-usability-scale-sus-to-evaluate-the-

usability-of-your-website/

TutorialsPoint (2018). SDLC – Rapid Application Development

Model. Retrieved on June 19, 2018 from https://www.tutorialspoint.com/adaptive\_software\_development/sdlc\_rapid\_application\_development\_model.htm

Wakefield, K. (n.d.). A Guide to Machine Learning Algorithms

and Their Applications from https://www.sas.com/en\_gb/

insights/articles/analytics/machine-learning-algorithms.html

Yardley, M.G. (2016). Financial Literacy Starts with

Budgeting. Retrieved from https://www.yardleywealth.net

/2016/01/financial-literacy-starts-with-budgeting/

Zucchi, K. (2018). Why financial literacy is so important.

Retrieved on June 9, 2018 from https://www.investopedia.com/articles/investing/100615/why-financial-literacy-and-education-so-important.asp

**Journals**

Doda, S. & Fortuzi, S. (2015). The Process of Financial

Planning in Personal Finance. International Journal of

Human Resource Studies, 5(4), 30-34.

**Unpublished Materials**

Cohen, I. & Forman, G. (n.d.) Learning from Little: Comparison

of Classifiers Given Little Training. Unpublished Manuscript, Hewlett-Packard Research Laboratories, Palo Alto, California

Johnson, M. & Nguyen, D. Q. (2017). How much data is enough?

Predicting how accuracy varies with training data size.

Macquarie University, Sydney, Australia.

|  |
| --- |
| **APPENDICES** |

**APPENDIX A**

**INTERVIEW REQUEST LETTER**

July 12, 2018

Dr. Kareen B. Leon

Dean

School of Business Administration and Accountancy

Madam:

We are 4th year Bachelor of Science in Information Technology students of the University of Baguio who are currently enrolled in PRSTDY1 (Project Study 1). As a partial requirement of the course, we would like to ask your permission to allow us to conduct an interview among your faculty members to collect the necessary information for the study “Witty Wallet: A Personal Finance Management Application”. Attached to the letter is a copy of the interview questions.

Rest assured that all information gathered would be treated with utmost confidentiality and was restricted for the purpose of this study only.

Thank you and more power.

Sincerely yours,

**Lysle L. Baday**

Group Representative

Noted by:

|  |  |
| --- | --- |
| **Erna-kristi N. Martinez**  PRSTDY1 Class Adviser | **Engr. Elisabeth D. Calub**  Dean, School of Information Technology |

**REPLY FORM**

This is to confirm that the School of Business Administration and Accountancy is allowing the group to conduct the interview which would be of help to the group in the completion of the project.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Signature over Printed Name of the Representative

Date Signed: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**APPENDIX B**

**INTERVIEW GUIDE**

**Interviewee:**

**Interview Date:**

Information that was collected through the interview was used solely for academic purposes only, specifically for the use of the study Witty Wallet. The interviewee’s participation is voluntary.

1. Can you walk us through the process of creating a personal budget?
2. What are the elements of a personal monthly budget?
   1. What are the key categories that should be included under expense
3. What are the different methods used in forecasting?
4. Which is best suited for the proposed application?
5. We are creating a personal finance mobile application, can you suggest what can be the best budgeting model that we can use? Why?

**APPENDIX C**

**SAMPLE DATA SET USED IN THE SELECTION OF A SUPERVISED MACHINE LEARNING ALGORITHM**

|  |  |
| --- | --- |
| November 2018 | |
| Category | Total Expense |
| Bills and Utilities | 2421.00 |
| Food and Beverage | 2007.00 |
| Transportation | 720.75 |
| Shopping | 479.75 |

**APPENDIX D**

**FEEDBACK FORM**



**Sytem Usability Scale for Witty Wallet**

This feedback form is based on the system usability scale created by John Brooke (1986). Rest assured that all information gathered below would be treated with utmost confidentiality and will be restricted for educational purposes only.

**Instructions:** For each of the following statements, kindly mark one box that corresponds to your response.

Strongly Agree

Strongly Disagree

|  |  |
| --- | --- |
| 1. I think that I would like to use this system frequently. |  |
| 1. I found the system unnecessarily complex. |  |
| 1. I thought the system was easy to use. |  |
| 1. I think that I would need the support of a technical person to be able to use this system. |  |
| 1. I found the various functions in this system were well integrated. |  |
| 1. I thought there was too much inconsistency in this system. |  |
| 1. I would imagine that most people would learn to use this system very quickly. |  |
| 1. I found the system very cumbersome to use. |  |
| 1. I felt very confident using the system. |  |
| 1. I needed to learn a lot of things before I could get going with this system. |  |

**Comments/ Suggestions:**

**APPENDIX E**

**System Usability Scale Computation**

The individual SUS scores was computed using the formula:

Where:

= sum of all points for all odd-numbered questions – 5

= 25 – sum of all all points for all even-numbered questions

The computed SUS scores for the thirty respondents are as follows:

|  |  |
| --- | --- |
| 1.) | 0 |
| 2.) | 42.5 |
| 3.) | 52.5 |
| 4.) | 57.5 |
| 5.) | 60 |
| 6.) | 60 |
| 7.) | 60 |
| 8.) | 60 |
| 9.) | 60 |
| 10.) | 60 |
| 11.) | 62.5 |
| 12.) | 62.5 |
| 13.) | 67.5 |
| 14.) | 70 |
| 15.) | 72.5 |
| 16.) | 75 |
| 17.) | 75 |
| 18.) | 77.5 |
| 19.) | 80 |
| 20.) | 80 |
| 21.) | 80 |
| 22.) | 80 |
| 23.) | 80 |
| 24.) | 85 |
| 25.) | 85 |
| 26.) | 85 |
| 27.) | 87.5 |
| 28.) | 87.5 |
| 29.) | 100 |
| 30.) | 100 |

The overall SUS score was computed by getting the mean of the individual scores of each respondent.

Where:

xi: the summation of the SUS scores from each user;

n: the total respondents

**APPENDIX F**

**DATA POLICY AGREEMENT AND TERMS OF USE OF WITTY WALLET**

* + 1. **Data Policy Agreement**

This policy describes information regarding the collection, use, and disclosure of user's (also "you") personal information when you use Witty Wallet's (also "us", "we", or "our") service. These policies are subject to change at any given time without notice.

**I. What Information Do We Collect?**

To provide Witty Wallet Services, we need to collect and process information about our users. This would include:

1. **Information and Content You Provide**

Witty Wallet collects the content such as account information, budgets, transactions and other information you provide when you use Witty Wallet. We collect these information for the purposes stated in section II of this document

1. **Application Usage**

We collect information about how users use our service. These information may include, but is not limited to, features you use, the time and frequency of your usage, and the last time you accessed the application

**II. How Do We Use Data We Collect?**

Witty Wallet use the information collected as described bellow

1. **Providing Content and Personaliztion**

Witty Wallet use the information users provide to personalize features and content for each user; this may include, but is not limited to, customized components in the UI, reminders, and notifications. We also use your information to provide content and features such as budget forecasts.

1. **Updates and Improvements**

We also use information from users to improve our service. Information can be used to develop new features and functions or improve existing ones.

1. **Safety and Security**

We use information you provide to verify accounts and activity, maintain the integrity of our application and promote the security of Witty Wallet's users' accounts

1. **Reaching our Users**

We use the information we have to communicate with you about our services. We also use the information you provide in order to respond to your queries when you contact us

* + 1. **Terms of Use**
       1. **Accounts**

1. **Accuracy and Completeness of Information**

When you create an account with us, you must provide us with information that is accurate, complete, and current at all times. Failure to do so constitutes a breach of the Terms, which may result in immediate termination of your account on our Service.

1. **Passwords**

You are responsible for safeguarding the password that you use to access the Service and for any activities or actions under your password, whether your password is with our Service or a third-party service. You agree not to disclose your password to any third party. You must notify us immediately upon becoming aware of any breach of security or unauthorized use of your account.

* + - 1. **Termination**
         1. We may terminate or suspend access to our Service immediately, without prior notice or liability, for any reason whatsoever, including without limitation if you breach the Terms.
         2. All provisions of the Terms which by their nature should survive termination shall survive termination, including, without limitation, ownership provisions, warranty disclaimers, indemnity and limitations of liability.
         3. We may terminate or suspend your account immediately, without prior notice or liability, for any reason whatsoever, including without limitation if you breach the Terms.
         4. Upon termination, your right to use the Service will immediately cease. If you wish to terminate your account, you may simply discontinue using the Service.
      2. **Governing Law**
         1. These Terms shall be governed and construed in accordance with the laws of the Philippines, without regard to its conflict of law provisions.
         2. Our failure to enforce any right or provision of these Terms will not be considered a waiver of those rights. If any provision of these Terms is held to be invalid or unenforceable by a court, the remaining provisions of these Terms will remain in effect.
         3. These Terms constitute the entire agreement between us regarding our Service, and supersede and replace any prior agreements we might have between us regarding the Service.
      3. **Changes**
         1. We reserve the right, at our sole discretion, to modify or replace these Terms at any time. If a revision is this material we will try to provide at least 15 days notice prior to any new terms taking effect. What constitutes a material change will be determined at our sole discretion.
         2. By continuing to access or use our Service after those revisions become effective, you agree to be bound by the revised terms. If you do not agree to the new terms, you are free to stop using the Service.

**APPENDIX G**

**THE SYSTEM PROFILE**



**Witty Wallet:**

**A Personal Finance**

**Mobile Application**

**Version 0.4**

**Prepared By:**

Baday, Lysle L.

Ballug, Anthony Jr. D.

Bie, Jonan V.

Padua, Lyra S.

December 2018

**Table of Contents**

[**ABOUT THE COMPANY 83**](#_Toc531164266)

[1.1 History 83](#_Toc531164267)

[1.2 Information System History 83](#_Toc531164268)

[1.3 Organizational Chart 84](#_Toc531164269)

[1.4 Positions 84](#_Toc531164270)

[**THE INFORMATION SYSTEM 85**](#_Toc531164271)

[2.1 Processes 85](#_Toc531164272)

[2.2 Source Documents 85](#_Toc531164276)

[2.3 Reports 86](#_Toc531164277)

[2.4 Policies 86](#_Toc531164278)

[2.5 Table of Information System Details 87](#_Toc531164279)

[2.6 Suggestions and Recommendations](#_Toc531164280) 90

[2.7 Flowchart of the existing system](#_Toc531164281) 91

[**REFERENCES**](#_Toc531164282) **95**

**1. ABOUT THE COMPANY**

**1.1 History**

Financial Management means planning, organizing, directing and controlling the financial activities such as procurement and utilization of funds of the enterprise (Management Study Guide, n.d.). It was a branch of economics until 1890 but has emerged as a distinct field of study at the turn of this century (Swarna & Tasnim, 2018). However, financial management relies heavily on economics for its theoretical concepts until now. According to Jain (n.d.), personal finance is the financial management which an individual or a family unit performs to budget, save, and spend monetary resources over time, taking into account various financial risks and future life events. Before a specialty in personal finance was developed, various disciplines which are closely related to it, such as family economics, and consumer economics were taught in various colleges as part of home economics for over 100 years (Hira, 2009).

Before 1990, mainstream economists and business faculty paid little attention to personal finance (Nandagopal & Sathyapriya, 2017). However, there is a renewed attention to personal finance education. This subject matter is currently gaining attention from various quarters of society, such as academia, government, corporations and nonprofit organizations (Hira, 2009). A variety of financial educational programs has emerged, the programs are frequently known as "financial literacy". Before the 2008 financial crisis occurred, there was no standardized curriculum for personal finance education. On January 22, 2008, the former president of the United Stated of America, George Bush, created the President’s Advisory Council on Financial Capability to advise the President and the Secretary of the Treasury on ways to improve financial literacy among all Americans (Golden, 2009).

**1.2 Information System History**

Personal Financial Management (PFM), a software that helps users manage their money started in 1983 with the founding of Intuit. PFM often lets users categorize transactions and add accounts from multiple institutions into a single view and also typically includes data visualizations such as spending trends, budgets and net worth. Scott Cook and Tom Proulx, the company’s founders, witnessed the rise of the personal computer and saw an opportunity to develop personal financial software (Hopkins, n.d.). Their flagship product, Quicken, became a standard for many households and was eventually followed by QuickBooks, which continues to help small businesses manage their finances today (GFT Group, 2015). Banks may offer PFM to their customers by creating their own PFM solutions or integrating third-party PFM solutions through Application Programming Interface (API) services.

Nowadays, personal finance can be done manually using a pen and paper, along with having the basic knowledge in financing. Thus, one can readily record his/her balance sheet and cash-flow statements. Modern technology is also a great help for personal finances with the aid of PFM applications that can be used in personal computers or mobile devices. Today’s PFM ecosystems are characterized by the following alternatives: Standalone FPM, a web-based application that allows end users to gather their financial data from different bank accounts into one place; Integrated online and mobile banking systems—based on a third party solution, this element integrates a product into an online banking platform and mobile banking app; and Custom developed solutions. Today, the integration of PFM functionality is one of the biggest sticking points within banks’ digital transformation programs (GFT Group, 2015).

**1.3 Organizational Chart**

Not applicable.

**1.4 Positions**

Not applicable.

**1.5 Services Offered**

Not applicable.

* + 1. **THE INFORMATION SYSTEM**

**2.1 Processes**

**2.1.1 Develop a financial goal**

* Analyze financial values and goals. Differentiate the needs from wants.
* Set long-term goals for the financial plan then short-term goals for the budget.
* Prioritize each goal.

**2.1.2 Manage monthly income and cash flow**

* Determine monthly income and amount.
* Determine monthly expenses. Sort them into categories.
* In each category of expense, set the initial monthly budget amount. By the end of the month, get the monthly actual amount or the actual value of expense. Get the difference of monthly budget amount to monthly actual amount.
* For the next month’s budget, make adjustments on the amount of budget or actual amount based on the calculated difference.
* To get amount of Savings, subtract total amount of expense to total amount of income. Outline how expenses measure up to the income.
* Stick to the budget and review the budget on a regular basis.

**2.1.3 Create a financial plan**

* Use budget plan to determine the current financial situation with regards to income, savings, living expenses, and debts.
* Prepare a list of current assets, debt balances and amounts spent for various items.
* Calculate the current net worth by getting the difference of current assets and liabilities.
* Organize financial records.

**2.2 Source Documents**

* **Bill -** an amount of money owed for goods supplied or services rendered, set out in a printed or written statement of charges.
* **Dividend -** a reward, cash or otherwise, that a company gives to its shareholders. Receipt is used for proof of receiving the dividend.
* **Interest –** a fee paid for the use of another party's money. Interest is stated in an agreement between two parties.
* **Loan -** an amount of money that is borrowed to be paid back with interest. The borrower signs a memorandum of agreement to the lender.
* **Payment Receipt –** a document that is a proof of payment.
* **Pay slip -** a document an employee receives either as a notice that the direct deposit transaction has gone through or is attached to the paycheck.
* **Income Tax Return -** a compulsory financial contribution imposed by a government.
* **Savings Account -** used to save money for specific expenses or for longer-term undefined goals.

**2.3 Reports**

* **Personal Budget Report –** a document that itemizes the sources of one’s income and expenditures for a future period.
* **Financial Plan report -** a document that is a comprehensive evaluation of an individual's current pay and future financial state.

**2.4 Policies**

**2.4.1 Financial goal**

* Goals should be Specific, Measurable, Attainable, Realistic, and Time-based (SMART) which is vital to financial planning.
* A long-term goal requires financial commitment for a plan to do in five or more years in the future.
* A short-term goal is smaller in scope with a definite target date for accomplishing it. The general rule is to save into cash deposits, such as saving accounts or fixed-term savings bonds.

**2.4.2 Budgeting**

* A standard rule is to divide after-tax income, spending 50% on needs and 30% on wants while allocating 20% to savings. Savings can be allocated for interest rates, loans, tax, and etc.
* Adjustments are made when the difference between budget and expense is either overestimated or underestimated.
* When setting a new monthly budget, make budgets accurate compared to last month’s budget.
* If not satisfied with the effect of spending and saving habits on the net worth, make changes in future income and expenditures.

**2.4.3 Financial Plan**

* Out of the monthly income, a certain percentage has to be saved before it is spent.
* Finances need to be personalized according to one’s risk profile, situations, etc.

**2.5 Table of Information System Details**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **User/**  **Position** | **Source Document /Fields** | **Process** | **Output Report/ Fields** | **Policies** |
| Consumer | (2.2.)  Dividend, Interest, Pay slip  (2.2)  Savings Account  (2.3)  Personal Budget Report | (2.1.1)  Analyze financial values and goals. Differentiate the needs from wants.  (2.1.1)  Set long-term goals for the financial plan then short-term goals for the budget.  Prioritize each goal.  (2.1.2)  Determine monthly income and amount.  Determine monthly expenses.  Sort them into categories.  In each category of expense, set the initial monthly budget amount.  By the end of the month, get the monthly actual amount or the actual value of expense.  Get the difference of monthly budget amount to monthly actual amount.  For the next month’s budget, make adjustments on the amount of budget or actual amount based on the calculated difference.  To get amount of Savings, subtract total amount of expense to total amount of income.  Outline how expenses measure up to the income.  Stick to the budget and review the budget on a regular basis.  (2.1.3)  Use budget plan to determine the current financial situation with regards to income, savings, living expenses, and debts.  Prepare a list of current assets, debt balances and amounts spent for various items.  Calculate the current net worth by getting the difference of current assets and liabilities.  Organize financial records. | (2.3)  Initial financial plan and Initial personal budget plan  (2.2)  Savings Account  (2.3)  Personal Budget Report  (2.3)  Financial Plan Report | (2.4.1)  Goals should be SMART.  (2.4.1)  A long-term goal requires financial commitment for a plan to do in five or more years in the future.  A short-term goal is smaller in scope with a definite target date for accomplishing it. The general rule is to save into cash deposits, such as saving accounts or fixed-term savings bonds.  (2.4.2)  A standard rule is to divide after-tax income, spending 50% on needs and 30% on wants while allocating 20% to savings.  Adjustments are made when the difference between budget and expense is either overestimated or underestimated. When setting a new monthly budget, make budgets accurate compared to last month’s budget.  If not satisfied with the effect of spending and saving habits on the net worth, make changes in future income and expenditures.  (2.4.3)  Out of the monthly income, a certain percentage has to be saved before it is spent. Finances need to be personalized according to one’s risk profile, situations, etc. |

**2.6 Suggestions and Recommendations**

* Make budgeting simple and easy to track.
* Creating a financial plan and a budget should be less tedious.

**2.7 Flowchart of the existing system**

**2.7.1 Develop a financial goal**

|  |
| --- |
| Consumer |
|  |

**2.7.2 Manage monthly income and cash flow**

|  |  |
| --- | --- |
| Consumer | |
|  | |
| Consumer | |
|  | |

**2.7.3 Create a financial plan**

|  |
| --- |
| Consumer |
|  |

**References**

Management Study Guide. (n.d.). Financial Management-Meaning,

Objectives and Functions. Retrieved on July 15, 2018

From https://www.managementstudyguide.com/financial- management.htm

Swarna, R. & Tansim, A. (2018). Definition & Origin of Financial

Management. Retrieved on July 15, 2018, from https://www.businesshaunt.com/principles-of-financial-management/definition-of-financial- management/

Jain, D. (n.d.). Personal Finance is the Financial Management

Which an Individual or a Family Unit Performs to Budget.

Retrieved from https://www.scribd.com/document/343997759/

Personal-Finance-is-the-Financial-Management-Which-an- Individual-or-a-Family-Unit-Performs-to-Budget

Hira, T. (2009). Personal Finance: Past, Present and Future.

Retrieved on July 16, 2018 from

https://www2.indstate.edu/business/NFI/leadership /briefs/2009-PB-10\_Hira.pdf

Nandagopal, R. & Sathyapriya, M. (2017). THE IMPORTANCE OF

PERSONAL FINANCE TODAY: A DISCUSSION OF CONCEPTS AND

COMPETENCE OF FINANCIAL LITERACY EDUCATION, IN INDIA.

Retrieved on July 16, 2018 from

https://scholarworks.montana.edu/xmlui/bitstream/ handle/1/2449/TschacheC0809.pdf

Golden, P. (2009). President's Advisory Council on Financial

Literacy Issues Annual Report. Retrieved on

July 16, 2018 from https://www.nefe.org/Press-

Room/News/PRESIDENTS-ADVISORY-COUNCIL-ON-FINANCIAL-

LITERACY

Hopkins, M. (n.d.). Scott Cook, Entrepreneurial Skills Article. Retrieved on July 16, 2018 from https://www.inc.com/magazine/20040401/25cook.html

GFT Group. (2015).The evolution of PFM. Retrieved on July 16,

2018 from https://slidex.tips/download/the-evolution-of-pfm

**APPENDIX H**

**SOFTWARE REQUIREMENTS SPECIFICATION (SRS)**



**Witty Wallet:**

**A Personal Finance**

**Mobile Application**

**Version 0.5**

**Prepared by:**

Lysle L. Baday

Anthony D. Ballug Jr.

Jonan V. Bie

Lyra Angelica S. Padua

December 2018

|  |  |
| --- | --- |
| **Document Sign-off** | |
| **Team Members** | **Panel Members** |
| **Lysle L. Baday**  System Developer  **Anthony D. Ballug Jr.**  System Developer  **Jonan V. Bie**  System Developer  **Lyra Angelica S. Padua**  System Developer  **Erna-Kristi N. Martinez**  Technical Adviser | **Divine Aguilar**  Panel Member  **Ellen M. Halover**  Panel Member  **Hydi D. Toyeng**  Panel Member |

**Revision History**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Date** | **Reason For Changes / Comments** | **Version** |
| Witty Wallet | 03/07/18 | Created the first draft for Chapters 1 – 3. | 0.1 |
| Witty Wallet | 10/08/18 | Added functional requirements and revised initial content. | 0.2 |
| Witty Wallet | 10/15/18 | Updated Product Functions | 0.3 |
| Witty Wallet | 11/26/18 | Updated Product Functions and Added GUI | 0.4 |
| Witty Wallet | 12/06/18 | Added Appendices and minor revisions | 0.5 |

**Table of Contents**

[**1.** **Introduction 100**](#_Toc527307640)

[1. 1 Purpose 100](#_Toc527307641)

[1. 2 Scope 100](#_Toc527307642)

[1. 3 Definitions, Acronyms, and Abbreviations 101](#_Toc527307643)

[1. 4 References 102](#_Toc527307644)

[1. 5 Overview 102](#_Toc527307645)

[**2.** **Business Description**](#_Toc527307646) **103**

[2. 1 Constraints](#_Toc527307647) 103

[2. 2 Business Description](#_Toc527307648) 103

[2. 3 Business Objectives](#_Toc527307649) 103

[2. 4 Stakeholder Profile](#_Toc527307650) 104

[**3.** **The Overall Description 105**](#_Toc527307651)

[3. 1 Product Perspective 105](#_Toc527307652)

[3. 2 Product Functions 106](#_Toc527307653)

[3. 3 User Characteristics 107](#_Toc527307654)

[3. 4 Assumptions and Dependencies 107](#_Toc527307655)

[**4.** **Specific Requirements 108**](#_Toc527307656)

[4. 1 Functional Requirements 108](#_Toc527307657)

4. 1. 1WW-UC01 (Create Account) 108

4. 1. 2WW-UC02 (Login) 109

4. 1. 3WW-UC03 (Create Budget) 110

4. 1. 4WW-UC04 (View Profile) 113

4. 1. 5WW-UC05 (Edit Nickname) 114

4. 1. 6WW-UC06 (Add Wallet) 115

4. 1. 7WW-UC07 (View Wallet List) 117

4. 1. 8WW-UC08 (Edit Wallet) 117

4. 1. 9WW-UC09 (Delete Wallet) 119

4. 1. 10WW-UC10 (Add Transaction) 120

[4. 1. 11WW-UC11 (Take Saving Challenge)](#_Toc532211106) 121

4. 1. 12WW-UC12 (Deposit) 122

4. 1. 13WW-UC13 (Add Savings) 123

4. 1. 14WW-UC14 (Budget Overview) 125

4. 1. 15WW-UC15 (Project Budget) 125

4. 1. 16WW-UC16 (Change Password) 126

4. 1. 17WW-UC17 (Forgot Password) 128

[4. 1. 18WW-UC18 (Logout)](#_Toc532211113) 129

[4. 2 Graphical User Intrerface 108](#_Toc527307657)

1. **Introduction**
2. **Purpose**

Witty Wallet will be created to support users in managing their finances, specifically to track and monitor their expenditures, savings, bills, debts, income and budgets.

The purpose of this SRS document is to contextualize the processes involved in Witty Wallet. Additionally, the document will serve as a tool to aid the team in the development of the application.

1. **Scope**

Witty Wallet is a mobile application that allows a user to manage their personal finances. The application is comprised of the following features:

* Provide an interface for the users to record their income, savings and expenses
* Categorization of expenses entered into the application,
* Offer saving challenges like *“the spare change challenge”,* etc. which the user may be interested in,
* Recording bills and debts with payment reminders on and before the due date, and the
* Creation of a personalized budget plan suggestion based on the user’s previous expenditure with a machine learning algorithm.

The mobile application will not require the user to connect their banks accounts nor to provide banking information in order for them to use the application. Witty Wallet will not provide a feature that allows the user to connect actual bank accounts nor will it consider the investment aspect of financial management. Witty Wallet will be designed for personal financial management only and is not meant for the use of businesses.

1. **Definitions, Acronyms, and Abbreviations**

**Definitions**

|  |  |
| --- | --- |
| **Term** | **Definition** |
| **Bills** | An itemized statement of money owed for products or services provided. |
| **Budget** | A budget is an approximation of expenses, income and resources over a specific period of time (Business Dictionary, 2018). |
| **Debts** | A debt is a particular amount of money borrowed by an individual from another (Investopedia, n.d.). |
| **Expense** | Expense is the cost incurred by an individual used to pay for a product or service (Debitoor.com, n.d.). |
| **Income** | A (financial) gain resulting from capital or labor (Merriam Webster Dictionary, 2018). |
| **Loans** | Loans refer to an arrangement wherein a party lends money to another party under a condition it is to be paid back, typically with interest, in the future (Quicken, n.d.). |
| **Personal Finance** | Personal finance is an activity involving expenses and savings (Kurt, 2018) |
| **Personal Financial Management** | The management of one’s assets (Batten, n.d.). |
| **Savings** | A portion of money allocated to be set aside for future use. |
| **Witty Wallet** | The name of the application that is to be developed. |

1. **References**

Batten, J. (n.d.). Personal Financial Management.

Retrieved on July 13, 2018 from

https://www.skillsyouneed.com/rhubarb/personal

-financial-management.html

Business Dictionary (n.d.) Budget. Retrieved on

July 3, 2018 from http://www.businessdictionary

.com/definition/budget.html.

Debitoor (n.d.). Expense – What is Expense?

Retrieved on July 13, 2018 from

https://debitoor.com/dictionary/expense

Income (2018). In Merriam-Webster .com Retrieved on

July 13, 2018 from https://www.merriam-

webster.com/dictionary/income

Investopedia (n.d.). Debt. Retrieved on July 13,

2018 from https://www.investopedia.com

/terms/d/debt.asp

Kurt, D. (2018). What is Finance? Retrieved on

July 13, 2018 from https://www.investopedia

.com/ask/answers/what-is-finance/

Quicken (n.d.). What are Loans?. Retrieved on

July 13, 2018 from https://www.quicken.com/what-

are-loans

* 1. **Overview**

This SRS document is subdivided into five major sections, namely the following:

Section 1 –Introduction. Provides an overview of Witty Wallet.

Section 2 –Business Description

Section 3 – The Overall Description

Section 4 –Specific Requirements

Section 5 –Appendices

1. **Business Description**
2. **Constraints**

The following are the limits relating to the development of Witty Wallet:

* Due to security purposes, the mobile application will not provide a feature that allows the user to enter bank credentials into the application.

1. **Business Description**

Investopedia (n.d.) defined money management as the process of budgeting, saving, investing, spending or otherwise overseeing the capital usage of an individual or group. But it takes a lot of time, effort, risks and self-discipline to be able to manage money wisely.

Budgeting is the process of planning on how to spend the money and it is one way of managing money. In the article “The Advantages of Budgeting in a Business”, Siddiqui (2017) stated that a budget can be made for a person, family, group of people, business, government or a country which is used to ensure long term financial security, planning and foresight. An individual can keep his/her finances on track using a personal budget. Also, it is integral to establish a budget for any corporation or individual.

1. **Business Objectives**

* **Allocation of Resources**. The management of personal finances and the budgeting process distributes assets or funds to various areas of expense, such as rent, bills and utilities, and the like.
* **Expense Tracking.** Knowing where money is being spent is essential before being able to change spending habits. Tracking expenses can show a person which areas are in need of improvement and which areas can be reduced or changed for the better.
* **Savings**. Saving allows the person to set aside a portion of their total assets for future use. One of the main objectives of saving is for a person to have an emergency fund to prepare for unexpected expenses.
* **Providing Structure.** Giving people a framework or a stable plan to manage spending in order to maximize their income.

1. **Stakeholder Profile**

|  |  |
| --- | --- |
| **Representative** | Lysle L. Baday, Anthony D. Ballug Jr., Jonan V. Bie, Lyra Angelica S. Padua |
| **Description** | The developers who are to create the application. |
| **Type** | Developers |
| **Responsibilities** | * Conduct the activities needed in the development of Witty Wallet. * Submit all the deliverables and meet deadlines. |
| **Involvement** | * Develop Witty Wallet |
| **Deliverables** | * Witty Wallet System Profile * Witty Wallet System Requirements Specification * Witty Wallet System Design Document * Witty Wallet Prototypes |
| **Comments / Issues** |  |

|  |  |
| --- | --- |
| **Representative** | Erna-kristi N. Martinez |
| **Description** | Project Study class adviser and technical adviser |
| **Type** | Project Reviewer |
| **Responsibilities** | * Gives deadlines for the submission of deliverables * Discuss the necessary information needed to successfully established the documentation |
| **Involvement** | * Monitors the project development * Check the project for deficiencies and gives recommendations/ suggestions |
| **Deliverables** | * Witty Wallet System Profile * Witty Wallet System Requirements Specification * Witty Wallet System Design Document * Witty Wallet Prototypes |
| **Comments / Issues** |  |

1. **The Overall Description**
   * 1. **Product Perspective**
2. **System Interfaces**

Witty Wallet has no required system interfaces.

1. **Hardware Interfaces**

The mobile application has no required hardware interfaces.

1. **Software Interfaces**

Witty Wallet has no required software interfaces.

1. **Communications Interfaces**

Witty Wallet will use the Hypertext Transfer Protocol to communicate between the user and the server.

1. **Operations**

Witty Wallet will be accessible to users 24/7 but certain functions may require internet connection, therefore the function will not be available until the user is connected to the internet.

1. **Site Adaptation Requirements**

The mobile application has no site adaptation requirements.

* + 1. **Product Functions**

|  |  |
| --- | --- |
| **Functionality** | **Description** |
| Create Account | This allows the user to create an account. |
| View Profile | Allows the user to view his/her own profile. |
| Edit Nickname | Allows the user to edit his/her own nickname. |
| Add Wallet | This facilitates the creation of a wallet. |
| View Wallet List | Allows the user to view Wallet list. |
| Edit Wallet | This allows the user to update selected wallet. |
| Delete Wallet | This allows the user to remove an existing wallet. |
| Add Transaction | This allows the user to add a transaction of an existing expense wallet. |
| Take Saving Challenge | This allows the user to participate in a saving challenge. |
| Deposit | This allows the user to enter an amount in a saving challenge. |
| Add Savings | This allows the user to deposit to to a savings wallet. |
| Budget Overview | Allows the user to view summarized report of current budget |
| Project Budget | This allows the user to project his/her budget for the next month. |
| Login | This allows the user to login to the system |
| Create Budget | This allows the user to create budget |
| Logout | This allows the user to logout from the system |
| Forgot Password | This allows the user to reset his/her own password |
| Change Password | This allows the user to change his/her password |

* + 1. **User Characteristics**

|  |  |  |
| --- | --- | --- |
| **User** | **Description** | **Functions** |
| **End user** | The end users of the application. | The client can manage his or her expenses, savings, bills, debts and loans. The user can also create a budget plan with the help of Witty Wallet’s machine learning algorithm. |

* + 1. **Assumptions and Dependencies**

The following are the assumptions and dependencies of Witty Wallet:

The system design and some of the defined requirements may change based from testing phase and user feedback.

1. **Specific Requirements**
2. **Functional Requirements**
3. **WW-UC01 (Create Account)**

|  |  |  |  |
| --- | --- | --- | --- |
| WW-UC01 | Create Account | | |
| **Description** | Allows the user to create an account | | |
| **Actor** | User | | |
| **Trigger** | User needs to create an account | | |
| **Complexity** | Easy | | |
| **Pre-condition** | * Sign up page is displayed | | |
| **Post condition** | * User has successfully created an account | | |
| **Basic Flow** | **Actor Action** | | **System Response** |
| **Step 1:** User fills up the sign-up form  **Step 2:** User submits form | | **Step 3:** Verifies and validates input  **Step 4:**  Saves input to the database.  **Step 5:**  Sends verification email. Goes back to login page and display success message with instructions. |
| **Alternate Flow** | **Actor Action** | | **System Response** |
| **Step 3a2:**  Goes back to Step 2 | | **Step 3a:**  Account details already exist  **Step 3a1:** Displays an error message that email is already in use |
| **Exception Flow** | **Actor Action** | **System Response** | |
|  |  | |
| **Includes** |  | | |
| **Assumptions** | * *Emergency Fund* Wallet is automatically created upon successful account registration | | |
| **Business Rules** |  | | |
| **Related Models/**  **Diagrams** |  | | |
| **Field Validations** | * Email * Password * Nickname | | |
| **Author** | Padua, Lyra | | |
| **Date** | August 16, 2018  October 4, 2018 | | |

1. **WW-UC02 (Login)**

|  |  |  |  |
| --- | --- | --- | --- |
| WW-UC02 | Login | | |
| **Description** | Allows the user to login to the system | | |
| **Actor** | User | | |
| **Trigger** | User needs to login to the system | | |
| **Complexity** | Easy | | |
| **Pre-condition** | * Login page is displayed * User account exists and is verified. | | |
| **Post condition** | * User has successfully logged in | | |
| **Basic Flow** | **Actor Action** | | **System Response** |
| **Step 1:** Fills up the form  **Step 2:** Submits form | | **Step 3:** Verifies and validates input  **Step 4:** Redirects the user to the home page |
| **Alternate Flow** | **Actor Action** | | **System Response** |
| **Step 3a2:**  Goes back to Step 1 | | **Step 3a:**  Incorrect username or password  **Step 3a1:**  Displays an error message (email or password is incorrect) |
| **Step 3a2:**  Goes back to Step 1 | | **Step 3a:**  User account does not exist.  **Step 3a1:**  Displays an error message (user not found). |
|  | | **Step 3a:**  User account is not activated.  **Step 3a1:**  Displays an error message (account not activated). |
| **Exception Flow** | **Actor Action** | **System Response** | |
|  |  | |
| **Includes** |  | | |
| **Assumptions** |  | | |
| **Business Rules** |  | | |
| **Related Models/**  **Diagrams** |  | | |
| **Field Validations** | * Email * Password | | |
| **Author** | Padua, Lyra  Ballug Jr. , Anthony | | |
| **Date** | August 16, 2018  October 4,2018  October 12, 2018  November 26, 2018 | | |

1. **WW-UC03 (Create Budget)**

|  |  |  |  |
| --- | --- | --- | --- |
| WW-UC03 | Create Budget | | |
| **Description** | Allows the user to create budget | | |
| **Actor** | User | | |
| **Trigger** | User is logged in  It’s the first day of the month | | |
| **Complexity** | Medium | | |
| **Pre-condition** | * Home page is displayed | | |
| **Post condition** | * User has successfully created budget | | |
| **Basic Flow** | **Actor Action** | | **System Response** |
| **Step 2:** Chooses Automatic option  **Step 4:**  User confirms  **Step 8:** Chooses to deposit  **Step 10:** Chooses an existing savings wallet | | **Step 1:**  Asks user to choose how budget is created  **Step 3:**  Displays brief explanation and confirmation of selected option  **Step 5:**  Automatically creates wallets(budget)  **Step 6:** Checks previous budget for extra savings  **Step 7:** Asks user to deposit extra savings  **Step 9:** prompts user to choose from savings wallet to deposit extra savings  **Step 11:** Allocates extra savings to selected savings wallet and informs user that extra savings has been successfully allocated |
| **Alternate Flow** | **Actor Action** | | **System Response** |
| **Step 2a:** Chooses Modified option  **Step 2a2:** User confirms  **Step 2a4:** User confirms and submits form | | **Step 2a1:**  Displays brief explanation and confirmation of selected option  **Step 2a3:** Redirects user to Budget creation page for editing of wallets to be kept  **Step 2a5:** Automatically creates wallets(budget) |
| **Step 2b:** Chooses Manual option  **Step 2b2:** User confirms | | **Step 2b1:** Displays brief explanation and confirmation of selected option  **Step 2b3:** Dismisses budget creation prompt, budget will be created manually |
| **Step9a:**  User declines to deposit | | **Step9a1:** dismiss budget creation modal |
| **Exception Flow** | **Actor Action** | **System Response** | |
|  |  | |
| **Includes** | * Add Wallet | | |
| **Assumptions** | * Budget can be created through an automated budget, modified by the user or manually created by the user * The Accuracy of budget prediction relies on the volume of the user’s data (the more data, the more accurate) * Allocation of extra savings is done when the period of budget has ended. * In the case where there are no other savings wallets created, ‘Emergency Fund’ will be default | | |
| **Business Rules** |  | | |
| **Related Models/**  **Diagrams** | A.12. Create Budget  A.3. Home Page | | |
| **Field Validations** | B.3. Create Budget | | |
| **Author** | Padua, Lyra,  Ballug Jr. , Anthony | | |
| **Date** | November 26, 2018  February 2, 2019 | | |

1. **WW-UC04 (View Profile)**

|  |  |  |  |
| --- | --- | --- | --- |
| WW-UC04 | View Profile | | |
| **Description** | Allows the user to view his/her own profile | | |
| **Actor** | User | | |
| **Trigger** | User needs to view his/her profile | | |
| **Complexity** | Easy | | |
| **Pre-condition** | * User is logged in | | |
| **Post condition** | * User has successfully viewed his/her profile | | |
| **Basic Flow** | **Actor Action** | | **System Response** |
| **Step 1:** Selects to view Profile | | **Step 2:**  Displays profile |
| **Alternate Flow** | **Actor Action** | | **System Response** |
|  | |  |
| **Exception Flow** | **Actor Action** | **System Response** | |
|  |  | |
| **Includes** |  | | |
| **Assumptions** |  | | |
| **Business Rules** |  | | |
| **Related Models/**  **Diagrams** |  | | |
| **Field Validations** |  | | |
| **Author** | Padua, Lyra | | |
| **Date** | August 19, 2018 | | |

1. **WW-UC05 (Edit Nickname)**

|  |  |  |  |
| --- | --- | --- | --- |
| WW-UC05 | Edit Nickname | | |
| **Description** | Allows the user to edit his/her own Nickname | | |
| **Actor** | User | | |
| **Trigger** | User needs to edit his/her nickname | | |
| **Complexity** | Easy | | |
| **Pre-condition** | * User profile is displayed | | |
| **Post condition** | * User has successfully edited his/her Nickname | | |
| **Basic Flow** | **Actor Action** | | **System Response** |
| **Step 1:** Selects to edit Nickname  **Step 3:** Edits field  **Step 4:** Submits form | | **Step 2:**  Displays edit Nickname form  **Step 5:**  Verifies and validates input  **Step 6:** Updates, saves Nickname to the database and displays message that user has successfully updated the Nickname  **Step 8:**  Dismisses edit form |
| **Alternate Flow** | **Actor Action** | | **System Response** |
| **Step 3a:**  Cancels to edit wallet | | **Step 3a1:**  Dismisses edit form |
| **Exception Flow** | **Actor Action** | **System Response** | |
| **Step 5a2:**  Goes back to Step 3 | **Step 5a:**  Field is empty  **Step 5a1:** Displays an error message | |
| **Includes** |  | | |
| **Assumptions** |  | | |
| **Business Rules** |  | | |
| **Related Models/**  **Diagrams** |  | | |
| **Field Validations** | * Name | | |
| **Author** | Padua, Lyra  Ballug Jr. , Anthony | | |
| **Date** | August 19, 2018  October 12, 2018 | | |

1. **WW-UC06 (Add Wallet)**

|  |  |  |  |
| --- | --- | --- | --- |
| WW-UC06 | Add Wallet | | |
| **Description** | Allows the user to add a Wallet | | |
| **Actor** | User | | |
| **Trigger** | User wants to create a wallet | | |
| **Complexity** | Moderate | | |
| **Pre-condition** | * Wallet list is displayed * Home page is displayed | | |
| **Post condition** | * User has successfully added a wallet | | |
| **Basic Flow** | **Actor Action** | | **System Response** |
| **Step 1:**  Selects to add Wallet  **Step 3:** Fills up the form  **Step 4:** Submits form | | **Step 2:**  Redirects user to adding wallet page  **Step 5:**  Verifies and validates input  **Step 6:** Saves input to the database and displays message that user has successfully added a new Wallet |
| **Alternate Flow** | **Actor Action** | | **System Response** |
| **Step 4a:**  Cancels to add wallet | | **Step 4a1:**  Redirects user to the default page |
| **Exception Flow** | **Actor Action** | **System Response** | |
| **Step 5a2:**  Goes back to Step 3 | **Step 5a:**  Input is invalid  **Step 5a1:** Displays an error message | |
| **Includes** |  | | |
| **Assumptions** | * The two types of Wallet are Expense and Savings * The following categories are: Food, Education, Housing, Bills & Utilities, Transportation, Health & Fitness, Entertainment, Leisure, Travel, Shopping, Debts & Loans, Gifts & Donations, Taxes, Personal Care, and Others for Expense Wallets * Amount in expense is the budget while amount in savings is the saving goal * Notifications can be set for Bills | | |
| **Business Rules** | * Only Expense type wallet can be categorized. * The *Emergency Fund* wallet cannot be modified, or deleted by the user and is a default savings wallet | | |
| **Related Models/**  **Diagrams** |  | | |
| **Field Validations** | * Name * Budget/Goal * Category | | |
| **Author** | Padua, Lyra  Ballug Jr. , Anthony | | |
| **Date** | August 16, 2018  October 4, 2018 | | |

1. **WW-UC07 (View Wallet List)**

|  |  |  |  |
| --- | --- | --- | --- |
| WW-UC07 | View Wallet List | | |
| **Description** | Allows the user to view list of wallets | | |
| **Actor** | User | | |
| **Trigger** | User needs to view wallets | | |
| **Complexity** | Easy | | |
| **Pre-condition** | * User is logged in | | |
| **Post condition** | * User has successfully viewed list of wallets | | |
| **Basic Flow** | **Actor Action** | | **System Response** |
| **Step 1:** Selects to view list of wallets | | **Step 2:**  Displays Wallet List |
| **Alternate Flow** | **Actor Action** | | **System Response** |
|  | |  |
| **Exception Flow** | **Actor Action** | **System Response** | |
|  |  | |
| **Includes** |  | | |
| **Assumptions** |  | | |
| **Business Rules** |  | | |
| **Related Models/**  **Diagrams** |  | | |
| **Field Validations** |  | | |
| **Author** | Padua, Lyra  Ballug Jr. , Anthony | | |
| **Date** | August 19, 2018  October 12, 2018 | | |

1. **WW-UC08 (Edit Wallet)**

|  |  |  |  |
| --- | --- | --- | --- |
| WW-UC08 | Edit Wallet name | | |
| **Description** | Allows the user to edit a wallet | | |
| **Actor** | User | | |
| **Trigger** | User wants to edit a wallet | | |
| **Complexity** | Easy | | |
| **Pre-condition** | * List of wallets is displayed | | |
| **Post condition** | * User has successfully edited a wallet | | |
| **Basic Flow** | **Actor Action** | | **System Response** |
| **Step 1:**  Selects a wallet to be edited  **Step 3:** Edits wallet fields and submits form | | **Step 2:**  Redirects user to edit wallet page  **Step 4:**  Verifies and validates input  **Step 5:** Updates, saves input to the database and displays message that user has successfully updated the wallet  **Step 6:**  Redirects user to the wallet page |
| **Alternate Flow** | **Actor Action** | | **System Response** |
| **Step 3a:**  Cancels to edit wallet | | **Step 3a1:**  Redirects user to the wallet list page |
| **Exception Flow** | **Actor Action** | **System Response** | |
| **Step 4a2:**  Goes back to Step 3 | **Step 4a:**  Field is empty  **Step 4a1:** Displays missing field error | |
| **Includes** |  | | |
| **Assumptions** | * Budget cannot be edited once a transaction record exists on selected wallet(Expense) | | |
| **Business Rules** |  | | |
| **Related Models/**  **Diagrams** |  | | |
| **Field Validations** | * Wallet Name * Budget/Goal * Category | | |
| **Author** | Padua, Lyra  Ballug Jr. , Anthony | | |
| **Date** | August 17, 2018  October 12, 2018 | | |

1. **WW-UC09 (Delete Wallet)**

|  |  |  |  |
| --- | --- | --- | --- |
| WW-UC09 | Delete Wallet | | |
| **Description** | Allows the user to delete a Wallet | | |
| **Actor** | User | | |
| **Trigger** | User wants to delete a wallet | | |
| **Complexity** | Easy | | |
| **Pre-condition** | * List of wallets is displayed | | |
| **Post condition** | * User has successfully deleted a wallet | | |
| **Basic Flow** | **Actor Action** | | **System Response** |
| **Step 1:**  Selects a wallet to be deleted  **Step 3:** Confirms to delete | | **Step 2:**  Asks for confirmation  **Step 4:** Deletes wallet  **Step 5:** Updates wallets list |
| **Alternate Flow** | **Actor Action** | | **System Response** |
| **Step 3a:**  Cancels to delete wallet | | **Step 3a1:**  Dismisses confirmation |
| **Exception Flow** | **Actor Action** | **System Response** | |
|  |  | |
| **Includes** |  | | |
| **Assumptions** | * Deleting a wallet includes the deletion of transactions of the wallet | | |
| **Business Rules** |  | | |
| **Related Models/**  **Diagrams** |  | | |
| **Field Validations** |  | | |
| **Author** | Padua, Lyra | | |
| **Date** | August 17, 2018 | | |

1. **WW-UC10 (Add Transaction)**

|  |  |  |  |
| --- | --- | --- | --- |
| WW-UC10 | Add Transaction | | |
| **Description** | Allows the user to add a transaction of expense | | |
| **Actor** | User | | |
| **Trigger** | User needs to add transaction | | |
| **Complexity** | Easy | | |
| **Pre-condition** | * Home page is displayed (Expense segment) | | |
| **Post condition** | * User has successfully added a transaction | | |
| **Basic Flow** | **Actor Action** | | **System Response** |
| **Step 1:** Selects an expense wallet  **Step 3:** Selects to add Transaction  **Step 5:** Fills up the form  **Step 6:** Submits form | | **Step 2:**  Displays transaction page of the expense wallet  **Step 4:**  Displays add transaction form  **Step 7:**  Verifies and validates input  **Step 8:** Saves input to the database  **Step 9:**  Displays message that user has successfully added a new transaction |
| **Alternate Flow** | **Actor Action** | | **System Response** |
| **Step 6a:**  Cancels to add transaction | | **Step 6a1:**  Dismisses add transaction form |
| **Exception Flow** | **Actor Action** | **System Response** | |
| **Step 7a2:**  Goes back to Step 5 | **Step 7a:**  Field is empty  **Step 7a1:** Displays an error message | |
| **Includes** |  | | |
| **Assumptions** |  | | |
| **Business Rules** |  | | |
| **Related Models/**  **Diagrams** |  | | |
| **Field Validations** | * Description * Amount | | |
| **Author** | Padua, Lyra | | |
| **Date** | August 16, 2018  October 6, 2018 | | |

1. **WW-UC11 (Take Saving Challenge)**

|  |  |  |  |
| --- | --- | --- | --- |
| WW-UC11 | Take Saving Challenge | | |
| **Description** | Allows the user to participate in a saving challenge | | |
| **Actor** | User | | |
| **Trigger** | User wants to take a saving challenge | | |
| **Complexity** | Easy | | |
| **Pre-condition** | * Saving Challenge page is displayed | | |
| **Post condition** | * User has successfully taken a saving challenge | | |
| **Basic Flow** | **Actor Action** | | **System Response** |
| **Step 1:** Selects a saving challenge  **Step 3:** Selects to take challenge | | **Step 2:**  Displays saving challenge description  **Step 4:**  Activates saving challenge  **Step 5:**  Displays message that user has accepted the saving challenge |
| **Alternate Flow** | **Actor Action** | | **System Response** |
| **Step 3a:**  Cancels to take challenge | | **Step 3a1:**  Redirects user to the saving challenge page |
| **Exception Flow** | **Actor Action** | **System Response** | |
|  |  | |
| **Includes** |  | | |
| **Assumptions** | The saving challenges are:   * The spare change challenge * 52-Week challenge | | |
| **Business Rules** | * The user can only take one saving challenge at a time. | | |
| **Related Models/**  **Diagrams** |  | | |
| **Field Validations** |  | | |
| **Author** | Padua, Lyra | | |
| **Date** | August 19, 2018  October 8, 2018 | | |

1. **WW-UC12 (Deposit)**

|  |  |  |  |
| --- | --- | --- | --- |
| WW-UC12 | Deposit | | |
| **Description** | Allows the user to deposit for a saving challenge | | |
| **Actor** | User | | |
| **Trigger** | User needs to deposit | | |
| **Complexity** | Easy | | |
| **Pre-condition** | * Saving Challenge page is displayed | | |
| **Post condition** | * User has successfully taken a saving challenge | | |
| **Basic Flow** | **Actor Action** | | **System Response** |
| **Step 1:**  Selects an active Saving Challenge  **Step 3:**  Deposits amount to save | | **Step 2:** Displays Saving Challenge Details  **Step 4:**  Saves input to the database  **Step 5:**  Displays message that amount has been deposited |
| **Alternate Flow** | **Actor Action** | | **System Response** |
|  | |  |
| **Exception Flow** | **Actor Action** | **System Response** | |
|  |  | |
| **Includes** |  | | |
| **Assumptions** |  | | |
| **Business Rules** |  | | |
| **Related Models/**  **Diagrams** |  | | |
| **Field Validations** |  | | |
| **Author** | Padua, Lyra | | |
| **Date** | October 8, 2018 | | |

1. **WW-UC13 (Add Savings)**

|  |  |  |  |
| --- | --- | --- | --- |
| WW-UC13 | Add Savings | | |
| **Description** | Allows the user to add a deposit to a savings wallet | | |
| **Actor** | User | | |
| **Trigger** | User needs to add a deposit | | |
| **Complexity** | Easy | | |
| **Pre-condition** | * Home page is displayed (Savings segment) | | |
| **Post condition** | * User has successfully added a deposit | | |
| **Basic Flow** | **Actor Action** | | **System Response** |
| **Step 1:** Selects a savings wallet  **Step 3:** Selects to add deposit  **Step 5:** Fills up the form  **Step 6:** Submits form | | **Step 2:**  Displays deposits page of the saving wallet  **Step 4:**  Displays add deposit form  **Step 7:**  Verifies and validates input  **Step 8:** Saves input to the database  **Step 9:**  Displays message that user has successfully added a new deposit |
| **Alternate Flow** | **Actor Action** | | **System Response** |
| **Step 6a:**  Cancels to add deposit | | **Step 6a1:**  Redirects user to the deposits page |
| **Exception Flow** | **Actor Action** | **System Response** | |
| **Step 7a2:**  Goes back to Step 5 | **Step 7a:**  Input is invalid  **Step 7a1:** Displays an error message | |
| **Includes** |  | | |
| **Assumptions** |  | | |
| **Business Rules** |  | | |
| **Related Models/**  **Diagrams** |  | | |
| **Field Validations** | * Amount | | |
| **Author** | Padua, Lyra,  Ballug Anthony | | |
| **Date** | August 16, 2018  October 6, 2018 | | |

1. **WW-UC14 (Budget Overview)**

|  |  |  |  |
| --- | --- | --- | --- |
| WW-UC14 | Generate Report | | |
| **Description** | Allows the user to view summarized report of current budget | | |
| **Actor** | User | | |
| **Trigger** | User needs to view budget summary | | |
| **Complexity** | Easy | | |
| **Pre-condition** | * Homepage is displayed | | |
| **Post condition** | * User has successfully generated budget overview report | | |
| **Basic Flow** | **Actor Action** | | **System Response** |
| **Step 1:** Selects Budget Summary tab  **Step 3:** Sets report filter  **Step 4:** Submits input | | **Step 2:**  Displays report page and generates report  **Step5:** Displays filtered report |
| **Alternate Flow** | **Actor Action** | | **System Response** |
|  | |  |
|  | |  |
| **Exception Flow** | **Actor Action** | **System Response** | |
|  |  | |
| **Includes** |  | | |
| **Assumptions** |  | | |
| **Business Rules** |  | | |
| **Related Models/**  **Diagrams** |  | | |
| **Field Validations** |  | | |
| **Author** | Padua, Lyra | | |
| **Date** | August 17, 2018 | | |

1. **WW-UC15 (Project Budget)**

|  |  |  |  |
| --- | --- | --- | --- |
| WW-UC15 | Project Budget | | |
| **Description** | Allows the user to project a suggested budget for the next month | | |
| **Actor** | User | | |
| **Trigger** | User needs to see the suggested budget | | |
| **Complexity** | Difficult | | |
| **Pre-condition** | * Budget Overview page is displayed | | |
| **Post condition** | * User has successfully projected a budget | | |
| **Basic Flow** | **Actor Action** | | **System Response** |
| **Step 1:** Selects Budget Summary | | **Step 2:**  Displays budget Summary and projects budget |
| **Alternate Flow** | **Actor Action** | | **System Response** |
|  | |  |
|  | |  |
| **Exception Flow** | **Actor Action** | **System Response** | |
|  |  | |
| **Includes** |  | | |
| **Assumptions** |  | | |
| **Business Rules** | * At least one month worth of data is needed to project a budget | | |
| **Related Models/**  **Diagrams** |  | | |
| **Field Validations** |  | | |
| **Author** | Padua, Lyra | | |
| **Date** | August 17, 2018  October 8, 2018 | | |

1. **WW-UC16 (Change Password)**

|  |  |  |  |
| --- | --- | --- | --- |
| WW-UC16 | Change Password | | |
| **Description** | Allows the user to change his/her password | | |
| **Actor** | User | | |
| **Trigger** | User needs to change the password | | |
| **Complexity** | Easy | | |
| **Pre-condition** | * User profile is displayed | | |
| **Post condition** | * User has successfully changed his/her password | | |
| **Basic Flow** | **Actor Action** | | **System Response** |
| **Step 1:** Selects to change Password  **Step 3:** Fills up the form  **Step 4:** Submits form | | **Step 2:** Displays change password form  **Step 5:**  Verifies and validates input  **Step 6:** Updates and saves input to the database  **Step 7:**  Displays message that user has successfully updated his/her password  **Step 8:**  Logs out user and redirects to login page |
| **Alternate Flow** | **Actor Action** | | **System Response** |
| **Step 4a:** Cancels to change password  **Step 5a1:** Goes back to Step 3  **Step 5b1:** Goes back to Step 3  **Step 5c1:** Goes back to Step 3 | | **Step 4a1:**  Redirects back to Profile page  **Step 5a:** Display error message that new password and confirmation of new password does not match  **Step 5b:** Display error message that current password does not match  **Step 5c:** Display error message that current password and new password are the same |
| **Exception Flow** | **Actor Action** | **System Response** | |
|  |  | |
| **Includes** | * Logout | | |
| **Assumptions** |  | | |
| **Business Rules** | * Password must be 6 to 20 characters long. | | |
| **Related Models/**  **Diagrams** |  | | |
| **Field Validations** | * Current Password * New Password * Confirm New Password | | |
| **Author** | Padua, Lyra,  Ballug Jr. , Anthony | | |
| **Date** | October 8, 2018  November 26, 2018 | | |

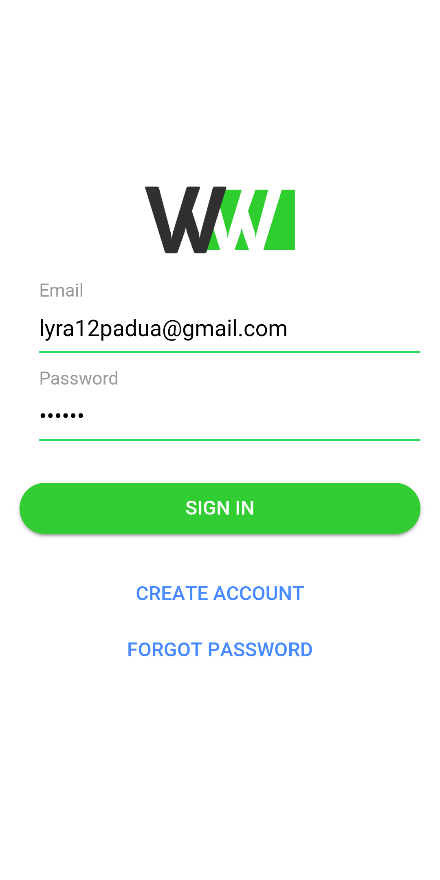
1. **WW-UC17 (Forgot Password)**

|  |  |  |  |
| --- | --- | --- | --- |
| WW-UC17 | Forgot Password | | |
| **Description** | Allows the user to reset his/her password | | |
| **Actor** | User | | |
| **Trigger** | User forgot his/her password | | |
| **Complexity** | Moderate | | |
| **Pre-condition** | * Login page is displayed | | |
| **Post condition** | * User has successfully reset his/her password | | |
| **Basic Flow** | **Actor Action** | | **System Response** |
| **Step 1:** Selects Forgot Password  **Step 3:** Enters Email  **Step 4:** Submits form | | **Step 2:** Prompts user to enter email  **Step 5:** Verifies and Validates input  **Step 6:** Sends password verification to email  **Step 7:** Informs user that a mail has been sent for password verification |
| **Alternate Flow** | **Actor Action** | | **System Response** |
| **Step 5a2:** Goes back to step 3 | | **Step 5a:** Email does not exist in the database  **Step 5a1:** Prompts user to enter a valid email |
| **Exception Flow** | **Actor Action** | **System Response** | |
|  |  | |
| **Includes** |  | | |
| **Assumptions** |  | | |
| **Business Rules** |  | | |
| **Related Models/**  **Diagrams** |  | | |
| **Field Validations** |  | | |
| **Author** | Padua, Lyra | | |
| **Date** | August 19, 2018 | | |

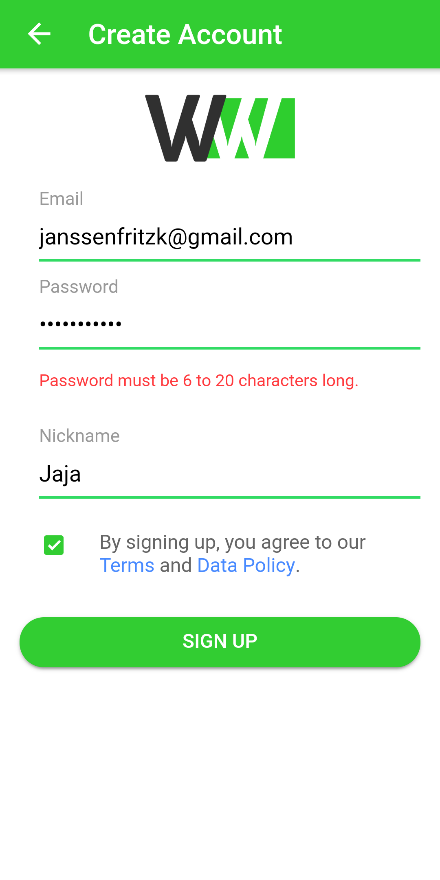
1. **WW-UC18 (Logout)**

|  |  |  |  |
| --- | --- | --- | --- |
| WW-UC18 | Logout | | |
| **Description** | Allows the user to logout from the app | | |
| **Actor** | User | | |
| **Trigger** | User needs to logout | | |
| **Complexity** | Easy | | |
| **Pre-condition** | * User is logged in | | |
| **Post condition** | * User has successfully logged out | | |
| **Basic Flow** | **Actor Action** | | **System Response** |
| **Step 1:** Selects Sign Out | | **Step 2:** Logs out user  **Step 3:** Redirects user to Login page |
| **Alternate Flow** | **Actor Action** | | **System Response** |
|  | |  |
| **Exception Flow** | **Actor Action** | **System Response** | |
|  |  | |
| **Includes** | * Login | | |
| **Assumptions** |  | | |
| **Business Rules** |  | | |
| **Related Models/**  **Diagrams** |  | | |
| **Field Validations** |  | | |
| **Author** | Padua, Lyra | | |
| **Date** | August 16, 2018 | | |

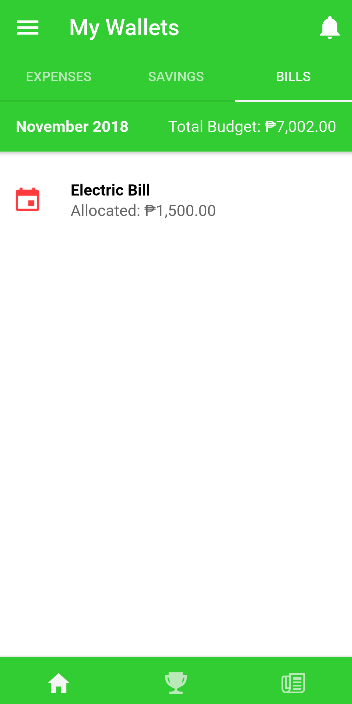
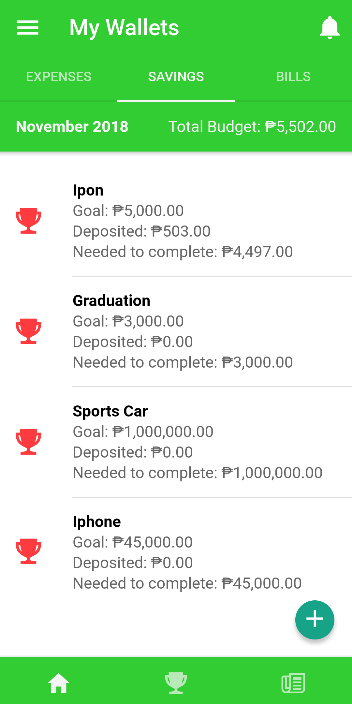
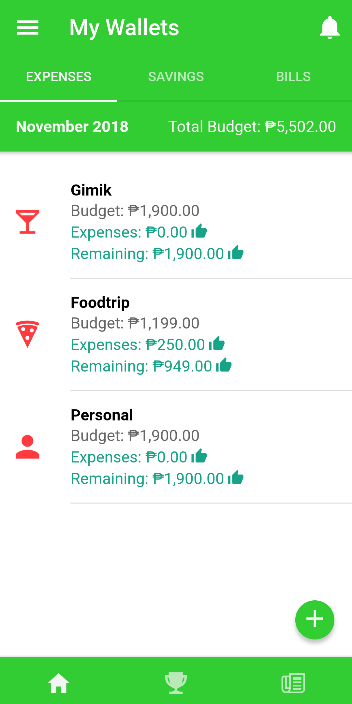
1. **Graphical User Interface (GUI)**
2. **Sign in Page**

****

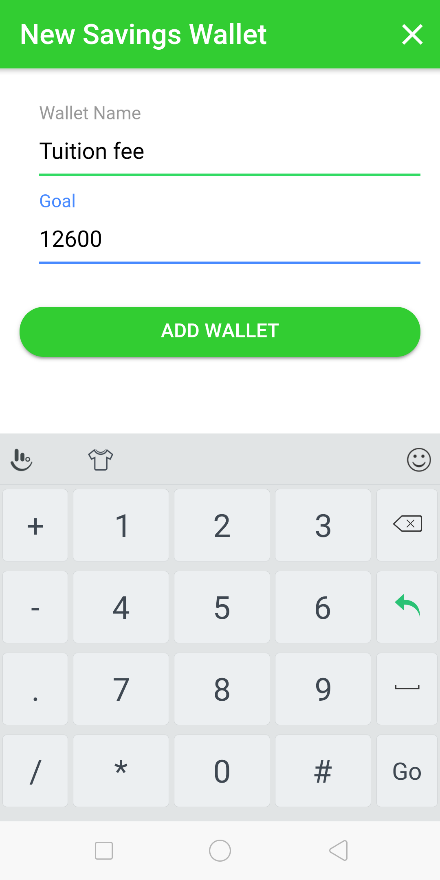
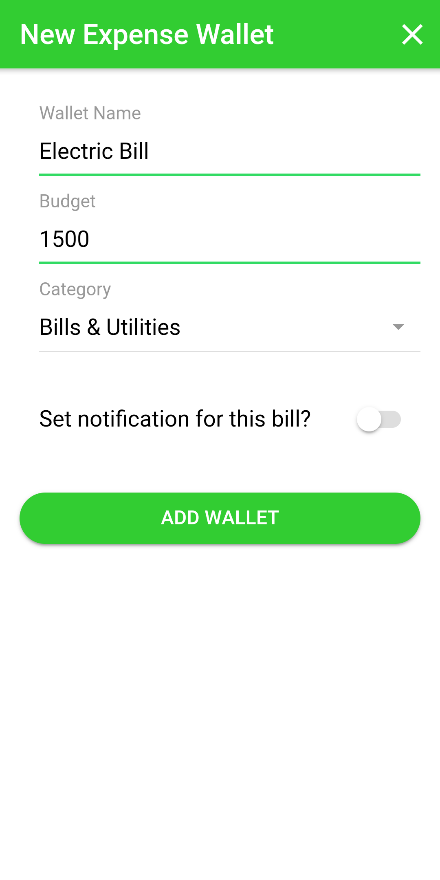
1. **Registration Page**

****

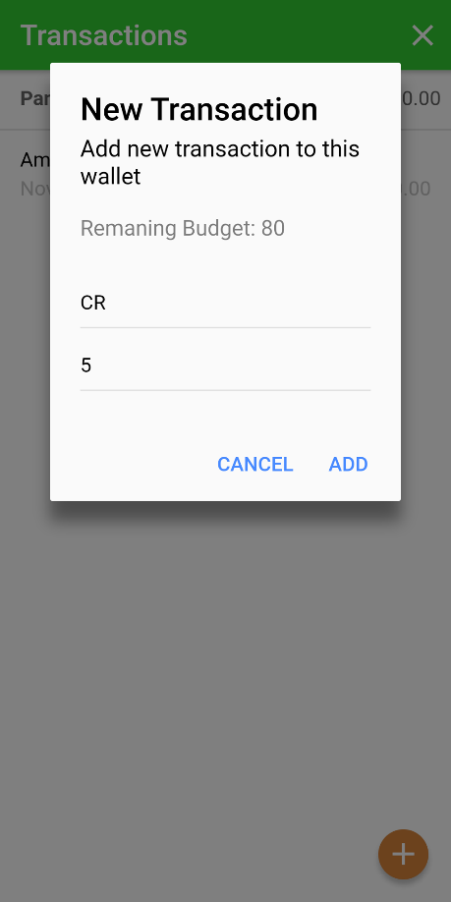
1. **Home Page**

****

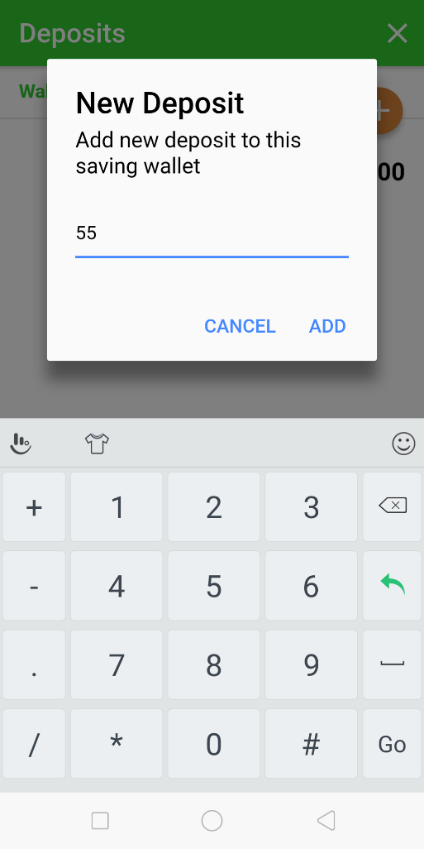
1. **Add Wallet**

****

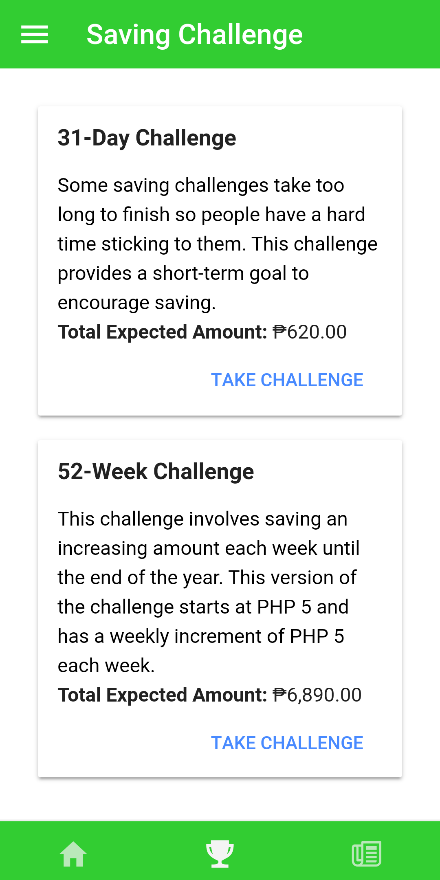
1. **Add Transaction**



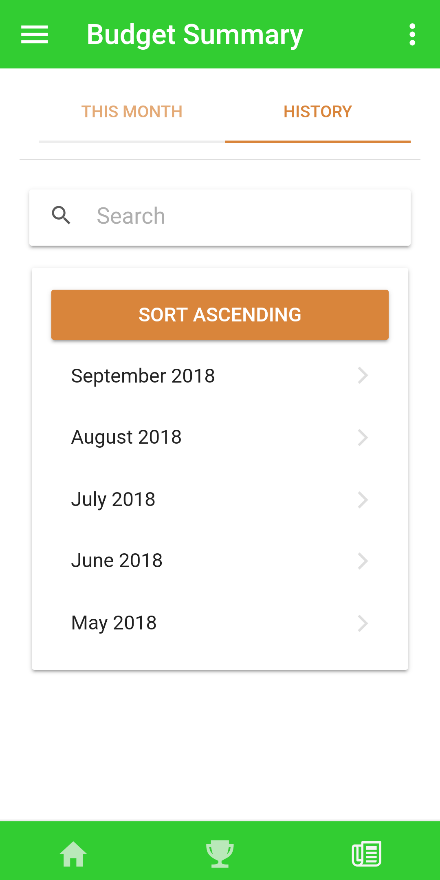
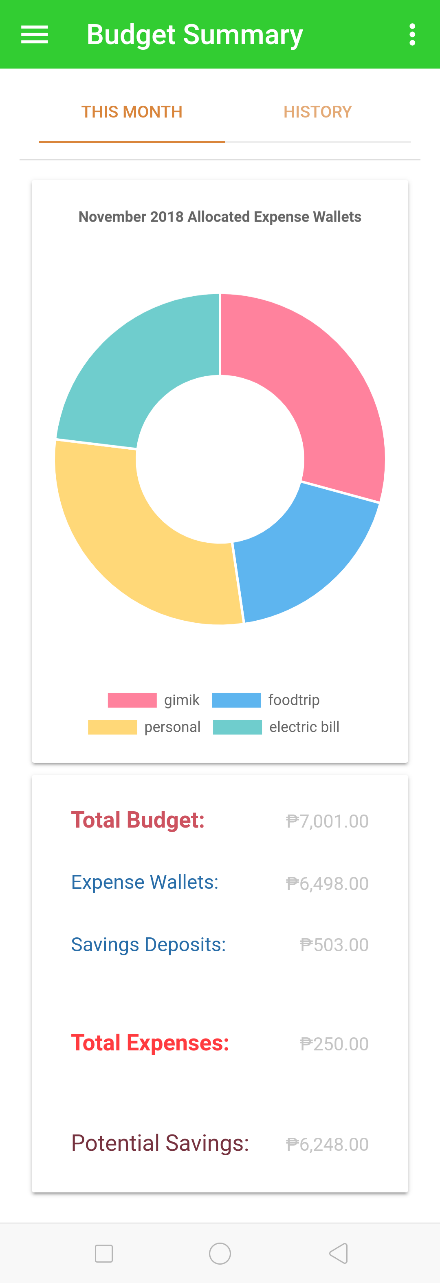
1. **Add Savings**



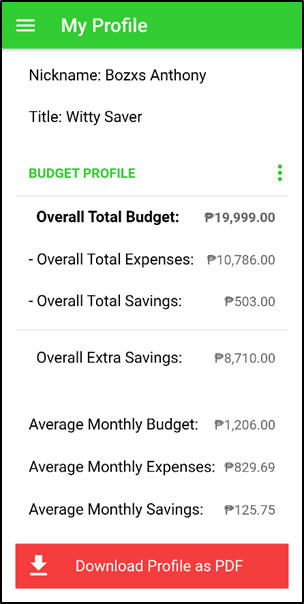
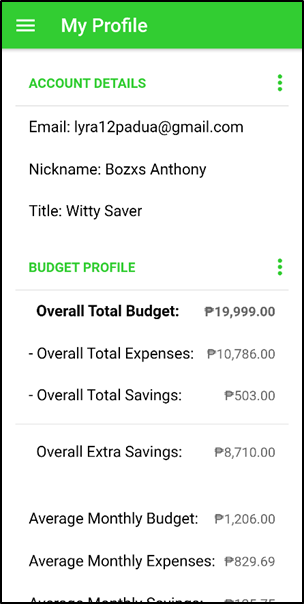
1. **Saving Challenge**

****

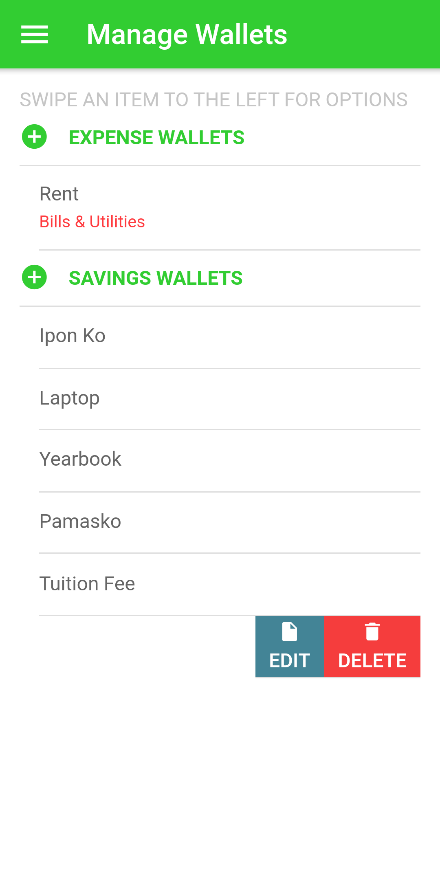
1. **Budget Overview**

****

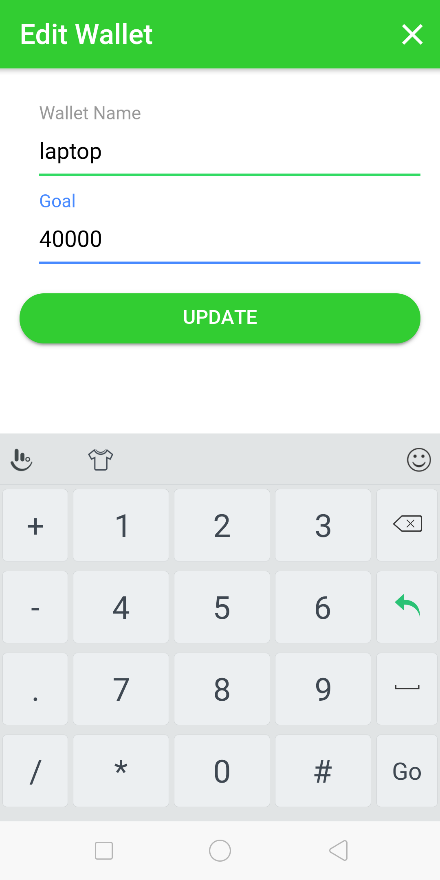
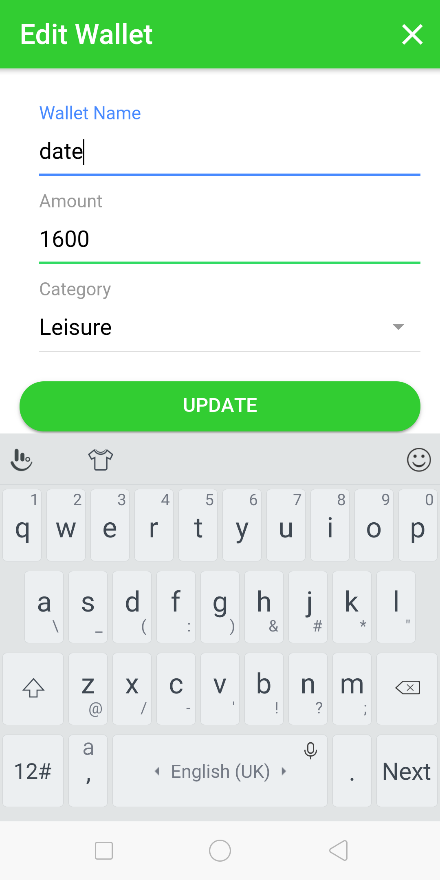
1. **View Profile**

****

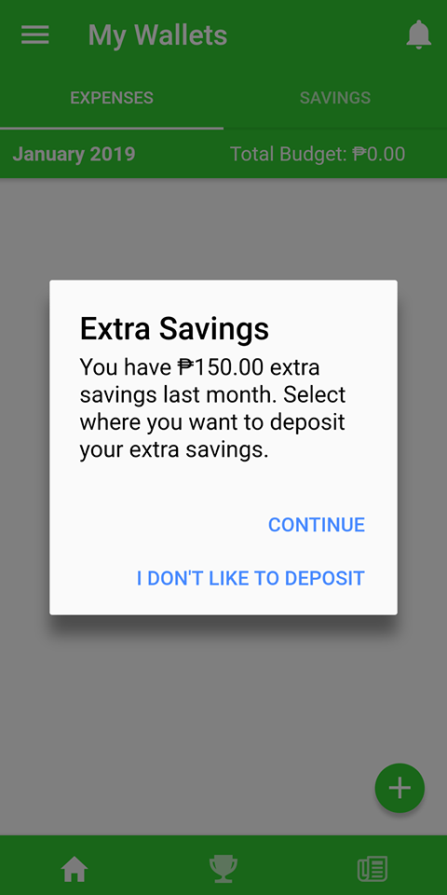
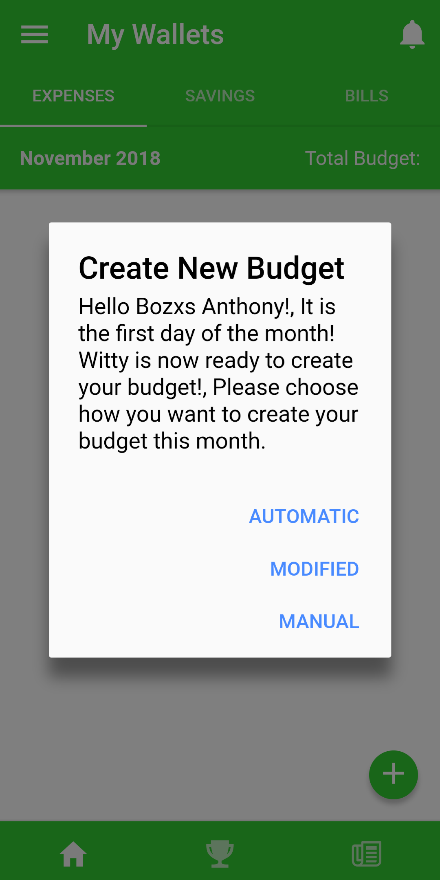
1. **Manage Wallets**

****

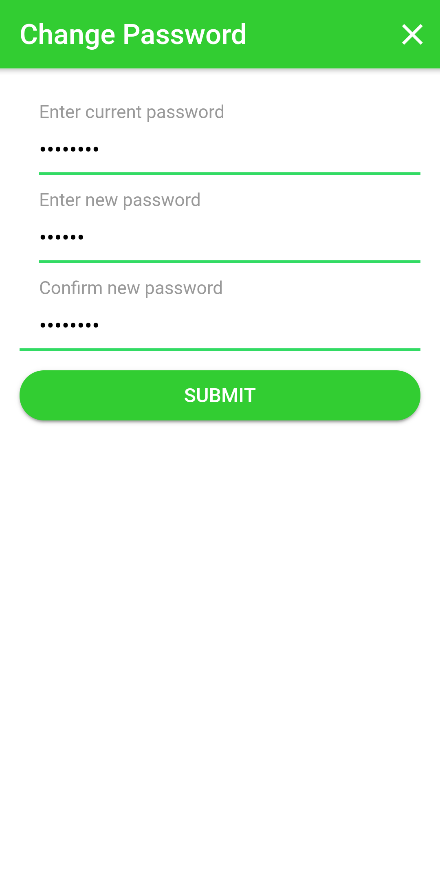
1. **Edit Wallet**

****

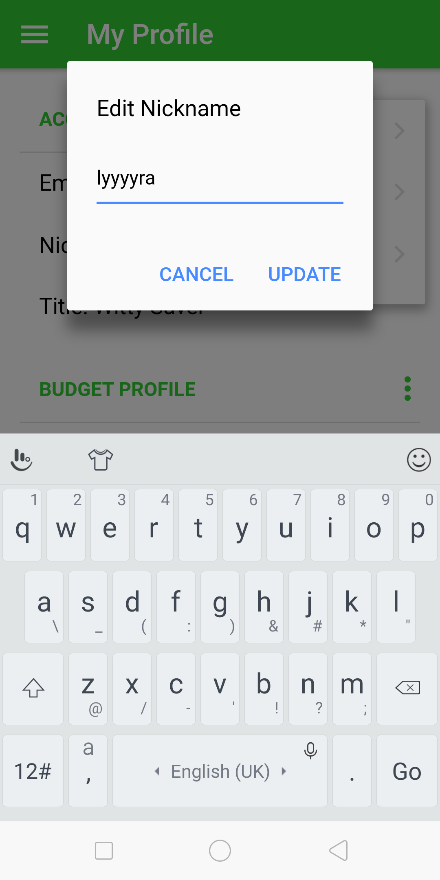
1. **Create Budget**

****

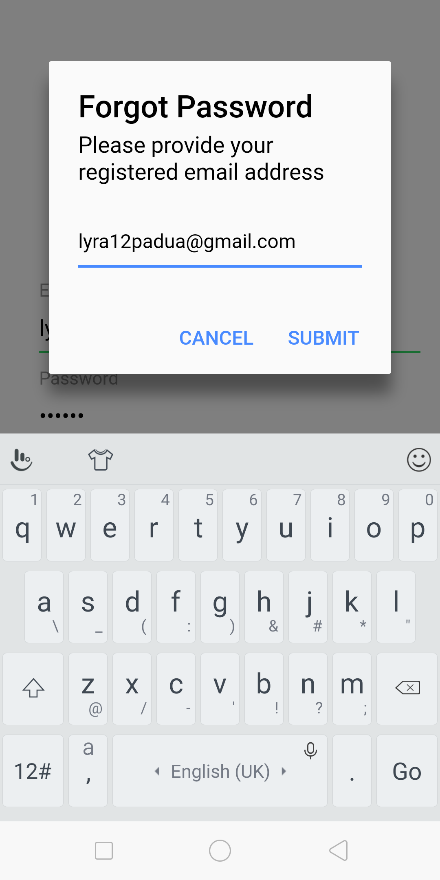
1. **Change Password**

****

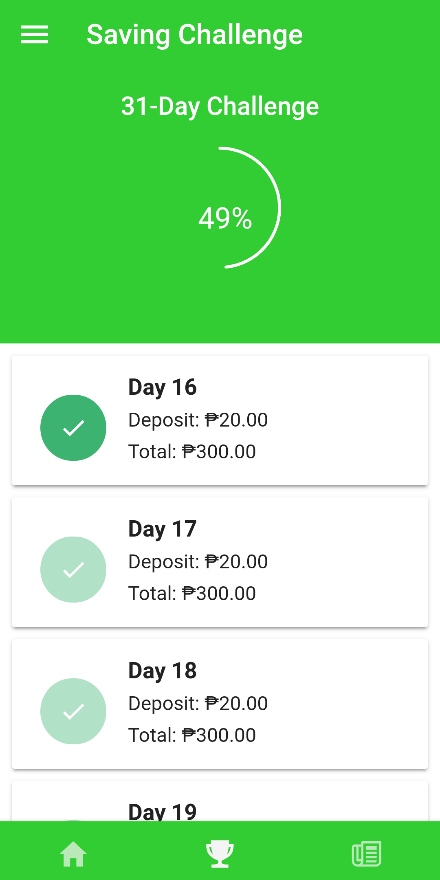
1. **Edit Nickname**

****

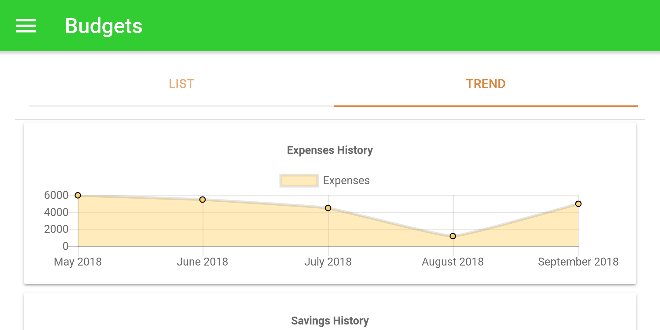
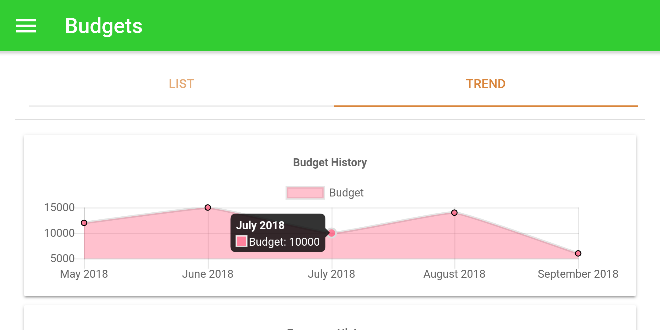
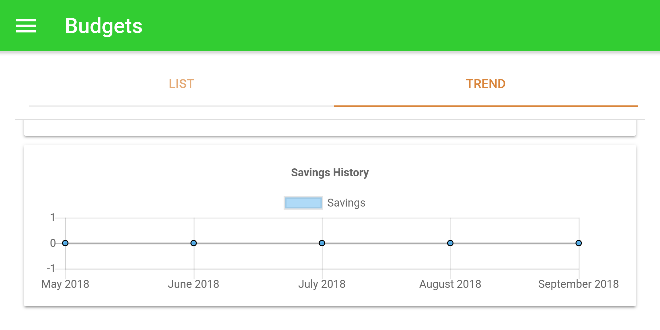
1. **Forgot Password**

****

1. **Deposit**

****

1. **Budgets**

****

**APPENDIX I**

**TABLE OF FIELDS**



**Witty Wallet:**

**A Personal Finance**

**Management Application**

**Prepared by:**

Lysle L. Baday

Anthony D. Ballug Jr.

Jonan V. Bie

Lyra Angelica S. Padua

1. **Create Account**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Fields** | **Type** | **Domain Constraints** | **Required**  **Fields** | **Other Considerations** | **User who**  **edits the content of the field** |
| User |
| User ID No. | Text (Hidden) | 0-9; a-z; | Automatically generated | Used as the  primary key | X |
| Email | Text field | A-Z; a-z;  0-9;  -; @; \_; .; min=12,  max=50  characters | Required | Email should not be already taken | X |
| Password | Passwordfield | 0-9; A-Z; a-z; -; min=6  characters | Required |  | X |
| Nickname | Text field | A-Z; a-z;  -; min=4,  max=20 characters | Required |  | X |

1. **Login**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Fields** | **Type** | **Domain Constraints** | **Required**  **Fields** | **Other**  **Considerations** | **User who**  **edits the content of the field** |
| User |
| Email | Text field | A-Z; a-z;  0-9;  -; @; \_; .; min=12,  max=50  characters | Required | Email should not be already taken | X |
| Password | Password field | 0-9; A-Z; a-z; -; min=6  characters | Required |  | X |

1. **Create Budget**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Fields** | **Type** | **Domain** **Constraints** | **Required**  **Fields** | **Other**  **Considerations** | **User who**  **edits the content of the field** |
| User |
| Wallet ID No. | Text  (Hidden) | 0-9; a-z; | Automatically generated | Used as the  primary key | X |
| Confirmation | Text Label | Hello <nickname> ! It is the first day of the month! Witty is now ready to create your budget! Please choose how you want to create your budget this month. |  | Used to confirm creation of budget | X |
| Budget | Number | 0-9; .; | Automatically generated/ Required |  | X |

1. **View Profile**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Fields** | **Type** | **Domain Constraints** | **Required**  **Fields** | **Other**  **Considerations** | **User who**  **edits the content of the field** |
| User |
| User ID No. | Text (Hidden) | 0-9; a-z; | Automatically generated | Used as the  primary key | X |
| Email | Text label | A-Z; a-z;  0-9;  -; @; \_; .; min=12,  max=50  characters |  | Email should not be already taken | X |
| Nickname | Text label | A-Z; a-z;  -; min=4,  max=20 characters |  |  | X |
| Title | Text label | A-Z; a-z; |  |  | X |
| Overall Total Budget | Number label | 0-9; .; |  |  | X |
| Overall Total Expenses | Number label | 0-9; .; |  |  | X |
| Overall Total Savings | Number label | 0-9; .; |  |  | X |
| Overall Extra Savings | Number label | 0-9; .; |  |  | X |
| Average Monthly Budget | Number label | 0-9; .; |  |  | X |
| Average Monthly Expenses | Number label | 0-9; .; |  |  | X |
| Average Monthly Savings | Number label | 0-9; .; |  |  | X |

1. **Edit Nickname**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Fields** | **Type** | **Domain** **Constraints** | **Required**  **Fields** | **Other**  **Considerations** | **User who**  **edits the content of the field** |
| User |
| User ID No. | Text (Hidden) | 0-9; a-z; | Automatically generated | Used as the  primary key | X |
| Nickname | Text field | A-Z; a-z;  -; min=4,  max=20 characters | Required |  | X |

1. **Add Wallet**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Fields** | **Type** | **Domain Constraints** | **Required**  **Fields** | **Other**  **Considerations** | **User who**  **edits the content of the field** |
| User |
| Wallet ID No. | Text (Hidden) | 0-9; a-z; | Automatically generated | Used as the  primary key | X |
| User ID No. | Text (Hidden) | 0-9; a-z; | Automatically generated | Used as the  foreign key of the user | X |
| Category ID No. | Text (Hidden) | 0-9; a-z; | Automatically generated | Used as the  foreign key of the category | X |
| Wallet Name | Text field | a-z; A-Z; min=1, max=20 characters | Required |  | X |
| Amount | Number field | 0-9; .; | Required |  | X |
| Period | Text label | ‘Month name + Year’ | Automatically generated |  | X |

1. **View Wallet List**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Fields** | **Type** | **Domain Constraints** | **Required**  **Fields** | **Other**  **Considerations** | **User who**  **edits the content of the field** |
| User |
| Wallet ID No. | Text (Hidden) | 0-9; a-z; | Automatically generated | Used as the  primary key | X |
| User ID No. | Text (Hidden) | 0-9; a-z; | Automatically  generated | Used as the  foreign key of the user | X |
| Category ID No. | Text (Hidden) | 0-9; a-z; | Automatically  generated | Used as the  foreign key of the category | X |
| Wallet Name | Text label | a-z; A-Z; min=1, max=20 characters |  |  | X |
| Amount | Number label | 0-9; .; |  |  | X |
| Remaining Budget | Text label | 0-9; .; |  |  | X |

1. **Edit Wallet**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Fields** | **Type** | **Domain Constraints** | **Required**  **Fields** | **Other**  **Considerations** | **User who**  **edits the content of the field** |
| User |
| Wallet ID No. | Text (Hidden) | 0-9; a-z; | Automatically  generated | Used as the  primary key | X |
| User ID No. | Text (Hidden) | 0-9; a-z; | Automatically  generated | Used as the  foreign key of the user | X |
| Category ID No. | Text (Hidden) | 0-9; a-z; | Automatically  generated | Used as the  foreign key of the category | X |
| Wallet Name | Text field | a-z; A-Z; min=1, max=20 characters | Required |  | X |
| Amount | Number field | 0-9; .; | Required |  | X |

1. **Delete Wallet**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Fields** | **Type** | **Domain Constraints** | **Required**  **Fields** | **Other**  **Considerations** | **User who**  **edits the content of the field** |
| User |
| Wallet ID No. | Text (Hidden) | 0-9; a-z; | Automatically  generated | Used as the  primary key | X |
| User ID No. | Text (Hidden) | 0-9; a-z; | Automatically  generated | Used as the  foreign key of the user | X |
| Category ID No. | Text (Hidden) | 0-9; a-z; | Automatically  generated | Used as the  foreign key of the category | X |
| Wallet Name | Text label | a-z; A-Z; min=1, max=20 characters | Required |  | X |
| Confirmation | Text label | Are you sure you want to delete wallet? |  | Used to confirm deletion of wallet | X |

1. **Add Transaction**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Fields** | **Type** | **Domain Constraints** | **Required**  **Fields** | **Other**  **Considerations** | **User who**  **edits the content of the field** |
| User |
| Wallet ID No. | Text (Hidden) | 0-9; a-z; | Automatically  generated | Used as the  primary key | X |
| User ID No. | Text (Hidden) | 0-9; a-z; | Automatically  generated | Used as the  foreign key of the user | X |
| Category ID No. | Text (Hidden) | 0-9; a-z; | Automatically  generated | Used as the  foreign key of the category | X |
| Description | Text field | a-z; A-Z; min=1, max=20 characters | Required |  | X |
| Amount | Number field | 0-9; .; | Required |  | X |
| Remaining Budget | Number label | 0-9; .; |  |  | X |

1. **Take Saving Challenge**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Fields** | **Type** | **Domain Constraints** | **Required**  **Fields** | **Other**  **Considerations** | **User who**  **edits the content of the field** |
| User |
| 31-Day Challenge | Text label |  |  |  | X |
| 52-Week Challenge | Text label |  |  |  | X |
| Total Expected Amount | Text label |  |  |  | X |

1. **Deposit**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Fields** | **Type** | **Domain Constraints** | **Required**  **Fields** | **Other**  **Considerations** | **User who**  **edits the content of the field** |
| User |
| Deposit | Number label |  |  |  | X |
| Total | Number label |  |  |  | X |

1. **Add Savings**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Fields** | **Type** | **Domain Constraints** | **Required**  **Fields** | **Other**  **Considerations** | **User who**  **edits the content of the field** |
| User |
| Wallet ID No. | Text (Hidden) | 0-9; a-z; | Automatically  generated | Used as the  primary key | X |
| User ID No. | Text (Hidden) | 0-9; a-z; | Automatically  generated | Used as the  foreign key of the user | X |
| Category ID No. | Text (Hidden) | 0-9; a-z; | Automatically  generated | Used as the  foreign key of the category | X |
| Amount | Number field | 0-9; .; | Required |  | X |

1. **Budget Overview**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Fields** | **Type** | **Domain Constraints** | **Required**  **Fields** | **Other**  **Considerations** | **User who**  **edits the content of the field** |
| User |
| Wallet ID No. | Text (Hidden) | 0-9; a-z; | Automatically  generated | Used as the  primary key | X |
| Wallet Name | Text label | a-z; A-Z; min=1, max=20 characters |  |  | X |
| Total Budget | Number label |  |  |  | X |
| Expense Wallet | Number label |  |  |  | X |
| Savings Deposit | Number label |  |  |  | X |
| Total Expenses | Number label |  |  |  | X |
| Potential Savings | Number label |  |  |  | X |
| Search | Text field |  |  | Used to search date | X |

1. **Project Budget**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Fields** | **Type** | **Domain Constraints** | **Required**  **Fields** | **Other**  **Considerations** | **User who**  **edits the content of the field** |
| User |
| Savings History | Text label |  |  |  | X |
| Budget History | Text label |  |  |  | X |
| Expense History | Text label |  |  |  | X |
| Period | Text label | ‘Month name + Year’ |  |  | X |

1. **Change Password**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Fields** | **Type** | **Domain Constraints** | **Required**  **Fields** | **Other**  **Considerations** | **User who**  **edits the content of the field** |
| User |
| User ID No. | Text (Hidden) | 0-9; a-z; | Automatically  generated | Used as the  primary key | X |
| Current Password | Passwordfield | 0-9; A-Z; a-z; -; min=6  characters | Required |  | X |
| New Password | Passwordfield | 0-9; A-Z; a-z; -; min=6  characters | Required |  | X |
| Confirm New Password | Passwordfield | 0-9; A-Z; a-z; -; min=6  characters | Required |  | X |

1. **Forgot Password**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Fields** | **Type** | **Domain Constraints** | **Required**  **Fields** | **Other**  **Considerations** | **User who**  **edits the content of the field** |
| User |
| User ID No. | Text (Hidden) | 0-9; a-z; | Automatically  generated | Used as the  primary key | X |
| Email | Text field | A-Z; a-z;  0-9;  -; @; \_; .; min=12,  max=50  characters | Required | Email should not be already taken and have been activated | X |

1. **Log out**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Fields** | **Type** | **Domain Constraints** | **Required**  **Fields** | **Other**  **Considerations** | **User who**  **edits the content of the field** |
| User |
| User ID No. | Text (Hidden) | 0-9; a-z; | Automatically  generated | Used as the  primary key | X |

**APPENDIX J**

**SOFTWARE DESIGN DOCUMENT (SDD)**



**Witty Wallet:**

**Personal Finance**

**Management Application**

**Version 0.5**

**Prepared by:**

Lysle L. Baday

Anthony D. Ballug Jr.

Jonan V. Bie

Lyra Angelica S. Padua

December 2018

**Revision History**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Date** | **Reason For Changes / Comments** | **Version** |
| Witty Wallet | August 17, 2018 | * Created first draft of SDD | v 0.1 |
| Witty Wallet | October 1, 2018 | * Added minor changes in Introduction * Updated Design Consideration * Updated System Environment * Updated Architecture * Added Data Design | v 0.2 |
| Witty Wallet | October 12, 2018 | * Minor changes in Design Consideration, Architecture, and Data Design | v 0.3 |
| Witty Wallet | November 26, 2018 | * Added GUI and Report Design * Minor changes | v 0.4 |

**Table of Contents**

[**1.** **Introduction** 154](#_Toc531180494)

[1.1. System Overview *15*](#_Toc531180495)*4*

[1.2. Scope *15*](#_Toc531180496)*4*

[1.3. Document Overview *155*](#_Toc531180497)

[**2.** **Design Consideration** 15](#_Toc531180498)5

[*2.1. Assumptions and Dependencies 15*](#_Toc531180499)*5*

[2. 1. 1 Assumptions 156](#_Toc531180500)

[2. 1. 2 Dependencies 15](#_Toc531180501)5

[*2.2. Drivers and Constraints 15*](#_Toc531180502)*6*

[2. 2. 1 Drivers 15](#_Toc531180503)6

[2. 2. 2 Constraints 15](#_Toc531180504)6

[*2.3. System Environment 15*](#_Toc531180505)*6*

[2.3.1 Development Environment 15](#_Toc531180506)7

[2.3.2 Deployment Environment 15](#_Toc531180507)7

[*2.4.Risk and Volatile Areas 15*](#_Toc531180508)*7*

[**3.** **Architecture** 15](#_Toc531180509)8

[*3.1.Architectural Style 15*](#_Toc531180510)*8*

[3. 1. 1 Deployment Architecture 15](#_Toc531180511)8

[*3.2. Program Structure 1*](#_Toc531180512)*59*

[3. 2. 1 Structure for the End-user 1](#_Toc531180513)59

[**4.** **Data Design** 16](#_Toc531180514)0

[4.1.Database Model 16](#_Toc531180515)0

[4.2.Database Schema 16](#_Toc531180517)4

[**5.** **Graphical User Interface** 16](#_Toc531180525)5

[5.1.Login 16](#_Toc531180526)5

[5.2.Registration 16](#_Toc531180527)5

[5.3.Home page 1](#_Toc531180528)66

[5.4.Add wallet 167](#_Toc531180529)

[5.5.Add Transaction 1](#_Toc531180531)68

[5.6.Add Savings 1](#_Toc531180532)69

[5.7.Saving Challenge 17](#_Toc531180533)0

[5.8.Side Menu 17](#_Toc531180534)1

[5.9.Profile. 17](#_Toc531180535)2

[5.10.Manage wallets page 17](#_Toc531180537)3

[5.1.Edit wallet 174](#_Toc531180539)

[5.2.Create budget 1](#_Toc531180541)75

[5.3.Change Password 176](#_Toc531180543)

[5.4.Edit nickname 177](#_Toc531180545)

[5.5.Forgot password 178](#_Toc531180547)

[5.6.Deposit 179](#_Toc531180549)

[**6.** **Report Design** 180](#_Toc531180551)

[6.1.Budgets 180](#_Toc531180552)

[6.2.Budget Summary 18](#_Toc531180554)1

1. **Introduction**
   1. **System Overview**

Witty Wallet discusses the details of the Witty Wallet system design. This document will introduce the architecture and design of the system for consideration. It includes the drivers and constraints that will be imposed upon the development of the software. Other discussions are the hardware requirements, software requirements, platform requirement, system environment, and the user interface of the system.

* 1. **Scope**

Witty Wallet is a mobile application will allow the user to manage his/her personal finances. The application will have the features that are solely for the purpose of helping the users in money management. The features are saving challenges, monthly budget suggestions, tracking and monitoring of bills, budgets, debts, expenses, and savings.

* 1. **Document Overview**

The document is divided into 7 major sections:

**Section 1: Introduction** – provides an overview of the Software Design Document and the scope of the document.

**Section 2: Design** Consideration – provides the assumptions and dependencies in the context of design; drivers and constraints; hardware and software requirements and the risk and volatile areas, if applicable.

**Section 3: Architecture** – shows the architectural style and program structure of the system.

**Section 4: Data Design** – developers shall refer to this section for database model and schema.

**Section 5: User Interfaces** – provides the user interfaces and discusses the events and actions.

**Section 6: Report Design** – provides the narrative overview of the reports that can be generated from the system and the layout of each report.

**Section 7: Definitions, Acronyms and Abbreviations** – provides definitions of all terms, acronyms and abbreviations needed for the SDD.

1. **Design Consideration**
   1. **Assumptions and Dependencies**

The following are the assumptions and dependencies of Witty Wallet:

1. **Assumptions**

* The team is meeting regularly and ensures that all agenda items are covered.
* The system will be made available to the public for personal use.

1. **Dependencies**

* Deadlines – the team has requirements that must be fulfilled before or after working on another requirement.
* Deliverables – upon the submission of deliverables the team can determine the development of the project.
* Deployment – the project deployment is done after accomplishing all the project dependencies.
* Methodology – following the RAD (Rapid Application Development) Methodology will allow the team to figure out what the project is supposed to accomplish.
* Schedule – to drive the project schedule, dependencies and constraints are set on all activities during the project scheduling.
  1. **Drivers and Constraints**

1. **Drivers**

* Quality – To ensure the quality of the system, the team will have different consultations with several advisers. The non-functional requirements are evaluated for the quality attributes.
* Strategy – The team entails a strategic plan to provide focus, direction and action to cope with changes in the project.
* Time Management – The team will have dedicated time allotted for developing the system. All schedules should be plotted in advanced.

1. **Constraints**

* Time Constraint – The team has limited time in developing the project and has to deal with conflict schedules.
* Resource Constraints – The team does not have enough resources such as private modules needed to complete the project. The team is also in the process of learning on utilizing the resources.
  1. **System Environment**

1. **Development Environment**

|  |  |
| --- | --- |
| **Hardware Specifications** | |
| **Hardware** | **Specifications** |
| Operating System | Windows 10 |
| Platform | Laptop |
| CPU | 2 GHz or faster x86 or x64 processor |
| Memory | At least 4 GB RAM (64-bit) |
| Disk Space | 50 GB or higher available hard disk space |
| **Software Specifications** | |
| **Software** | **Specifications** |
| Database | MongoDB v3.6.5 |
| Runtime | NodeJS v8.12.0, NPM 5.6.0 |
| Text Editor | Visual Studio Code v 1.25.1 |
| Programming Language | Angular v4.0 |
| Mark-up Language | HTML5, SASS |
| Framework | Ionic v3.19.0 |
| Version Control | Git v2.17.1 |

1. **Deployment Environment**

|  |  |
| --- | --- |
| **Server Specifications – Heroku Server** | |
| **Database** | **Specifications** |
| CPU | Singe-core Shared |
| Memory | 500MB |
| Disk Space | 500MB |
| **Mobile Specifications** | |
| **Software** | **Specifications** |
| OS Version | Android v4.4.0 KitKat |

**2.4 Risk and Volatile Areas**

* Certainty of producing output that are consistent with the project specifications.
* Balanced dictation of schedule, resources, and product definition.
* The developer’s training in using the development tools.

1. **Architecture**

**Architectural Style**

1. **Deployment Architecture**

|  |  |
| --- | --- |
| **Client Tier** | **Data Tier** |
| Mobile Application on | Database Server |

The software architecture of Witty Wallet is Two-Tier Architecture. The client and the database are separated into two components. The user interface runs on the mobile application making Witty Wallet understandable to the user. The application written in Angular and data are hosted on a server. The Data tier provides data storage and access to the application data using MongoDB. Data is accessed by the application tier via API calls.

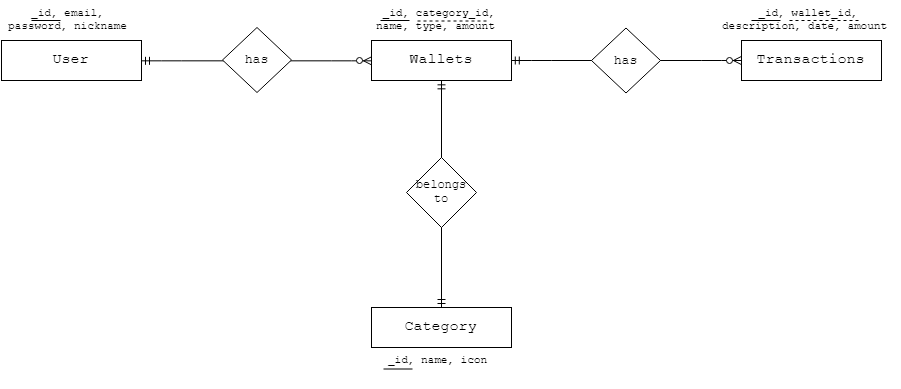
**Program Structure**

1. **Structure for the End-user**
2. **Data Design**

**Database Model****­­­**

MongoDB Document Model



ERD Model 

**Database Schema**

1. **Account**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Fields** | **Type** | **Size** | **Domain Constraints** | **Description** | **Mandatory** |
| \_id | String | 12-byte | 0-9; a-z; | This is the unique id of the user. | Yes |
| Email | String | 50 | email@emailprovider .domain | This is the user’s e-mail. | Yes |
| password | String | Min=6  Max=20 | 0-9; A-Z; a-z; -; @; | This is the user’s password to login to the system. | Yes |
| Name | String | Min=4  Max=20 | a-z; A-Z; | This is the user’s name in the system. | Yes |
| Activated | Boolean |  | True; False | This is the account status of the user | Yes |

1. **Expense**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Fields** | **Type** | **Size** | **Domain Constraints** | **Description** | **Mandatory** |
| \_id | String | 12-byte | 0-9; a-z; | This is the unique id of the wallet. | Yes |
| userId | String | 12-byte | 0-9; a-z; | This is the unique id reference of the user. | Yes |
| CategoryId | String | 12-byte | 0-9; a-z; | This is the category id reference of the category. | Yes |
| name | String | Min=1  Max=20 | a-z; A-Z; | This is the wallet name in the system. | Yes |
| amount | Number | Min=1  Max=9 | 0-9 | This is the amount set for the wallet. | Yes |
| period | String |  | ‘Month name + Year’ | This is the period by month where the wallet exists. | Yes |

1. **Savings**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Fields** | **Type** | **Size** | **Domain Constraints** | **Description** | **Mandatory** |
| \_id | String | 12-byte | 0-9; a-z; | This is the unique id of the wallet. | Yes |
| userId | String | 12-byte | 0-9; a-z; | This is the unique id reference of the user. | Yes |
| name | String | Min=1  Max=20 | a-z; A-Z; | This is the wallet name in the system. | Yes |
| goal | Number | Min=1  Max=9 | 0-9 | This is the amount set for the wallet. | Yes |

1. **Transactions (Expense)**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Fields** | **Type** | **Size** | **Domain Constraints** | **Description** | **Mandatory** |
| \_id | String | 12-byte | 0-9; a-z; | This is the unique id of the wallet. | Yes |
| walletId | String | 12-byte | 0-9; a-z; | This is the unique id reference wallet. | Yes |
| Description | String | Min=1  Max=50 | a-z; A-Z; | This is the description of the transaction | Yes |
| Amount | Number | Min=1  Max=9 | 0-9 | This is the amount of the transaction | Yes |

1. **Deposits (Savings)**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Fields** | **Type** | **Size** | **Domain Constraints** | **Description** | **Mandatory** |
| \_id | String | 12-byte | 0-9; a-z; | This is the unique id of the wallet. | Yes |
| walletId | String | 12-byte | 0-9; a-z; | This is the unique id reference wallet. | Yes |
| amount | String | Min=1  Max=20 | a-z; A-Z; | This is the amount of the deposit | Yes |
| Period | Number | Min=1  Max=9 | ‘Month name + Year’ | This is period when the deposit is added | Yes |

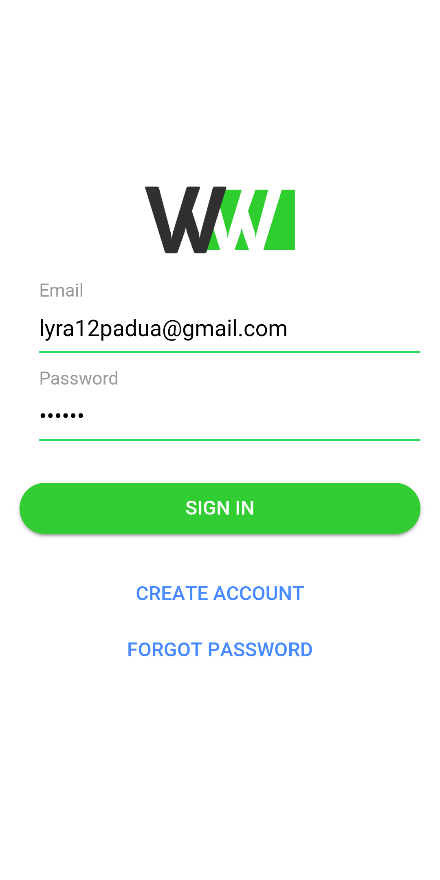
1. **Category**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Fields** | **Type** | **Size** | **Domain Constraints** | **Description** | **Mandatory** |
| \_id | String | 12-byte | 0-9; a-z; | This is the unique id of the category. | Yes |
| name | String |  | Food; education; housing; bills & utilities; transportation; health & fitness; entertainment; leisure; travel; shopping; debts & loans; gifts & donations; taxes; personal care; others; | This is the name of the category. | Yes |
| icon | String |  | a-z; | This field is the icon of the category | Yes |

1. **Archive**

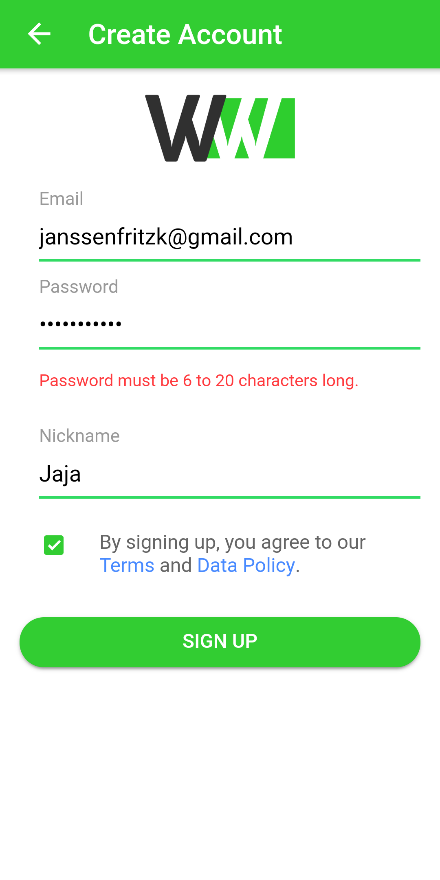
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Fields** | **Type** | **Size** | **Domain Constraints** | **Description** | **Mandatory** |
| \_id | String | 12-byte | 0-9; a-z; | This is the unique id of the archives. | Yes |
| totalBudget | Number | Min=1  Max=9 | 0-9 | This is the amount set for the total budget of the archive. | Yes |
| totalSavings | Number | Min=1  Max=9 | 0-9 | This is the amount set for total savings of the archive. | Yes |
| totalExpenses | Number | Min=1  Max=9 | 0-9 | This is the amount set for the total expenses of the archive. | Yes |
| period | String |  | ‘Month name + Year’ | This is the period by month where the archives exist. | Yes |

1. **Graphical User Interface**
2. **Login** –Allows the user to login into the app**.**



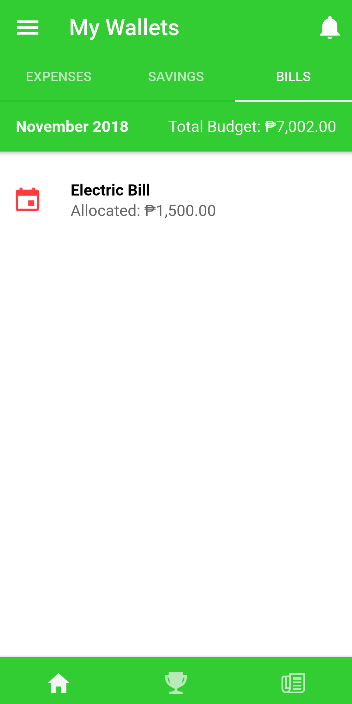
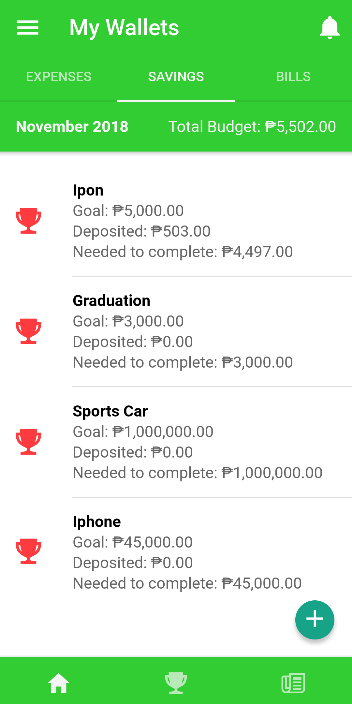
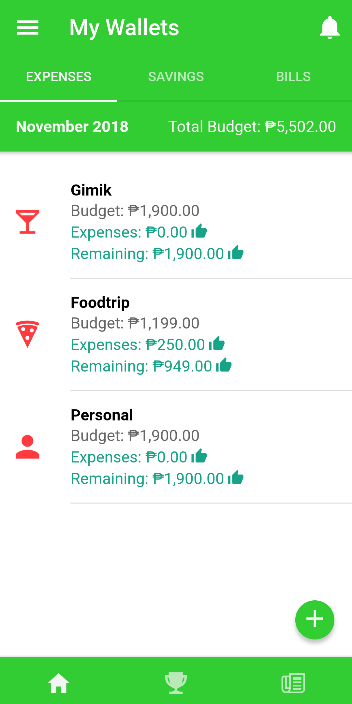
|  |  |  |  |
| --- | --- | --- | --- |
| Task | Object | Event | Event Details |
| Buttons | | | |
| User taps on the ‘Sign in’ button | Sign in | Tap | Submits login form. Redirects the user to home page |
| User taps on the ‘Create account’ link | Sign up | Tap | Redirects the user to registration page. |
| User clicks on the ‘forgot password’ link | Forgot password | Tap | Presents forgot password form. |
| Textboxes | | | |
| The user enters email | Email | Submit | Validates the user’s email |
| The user enters password | Password | Submit | validates the user’s password |

1. **Registration** –Allows the user to register an account.



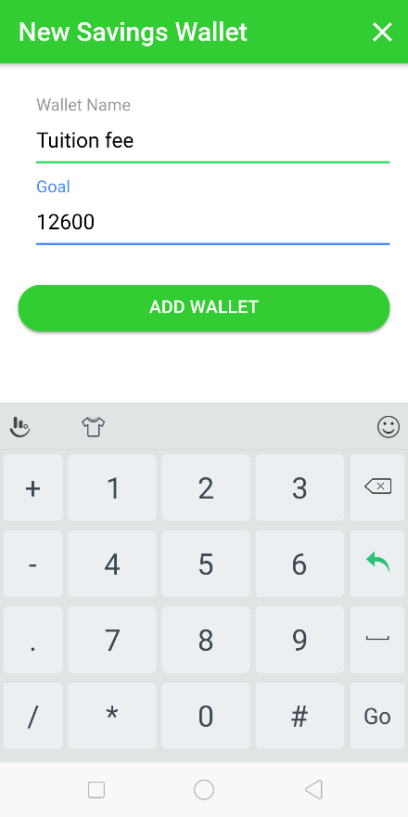
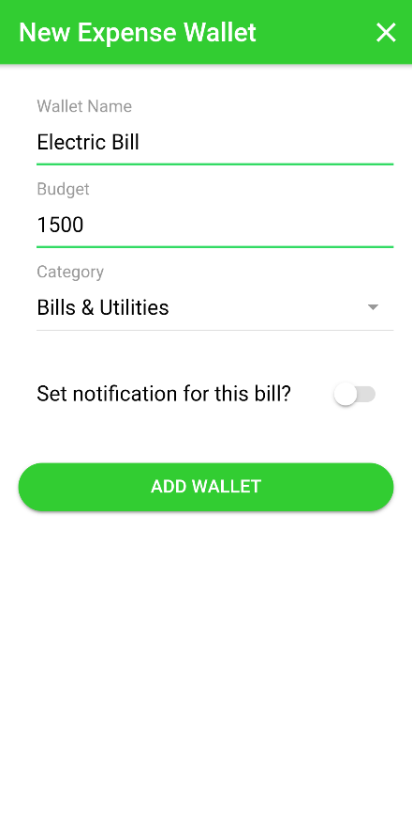
|  |  |  |  |
| --- | --- | --- | --- |
| Task | Object | Event | Event Details |
| Buttons | | | |
| User taps ‘Sign up’ button | Sign up | Tap | Submits Sign up form. Creates a new user account |
| User taps on back button | Back | Tap | Redirects user to login page |
| Toggle | | | |
| User switch toggle on | Toggle | Tap | Activates sign up button (all fields must be filled) |
| Links | | | |
| User taps on ‘Terms’ link | Terms | Tap | Shows *Witty Wallet registration Terms and Conditions* |
| User taps on ‘Data Policy link | Data Policy | Tap | Shows *Witty Wallet Data Privacy Policy* |
| Textfield | | | |
| User enters email | Email | Submit | Validates and saves submitted email |
| User enters password | Password | Submit | Saves submitted password |
| User enters nickname | Nickname | Submit | Saves submitted nickname |

1. **Home page** – The default page of the user.



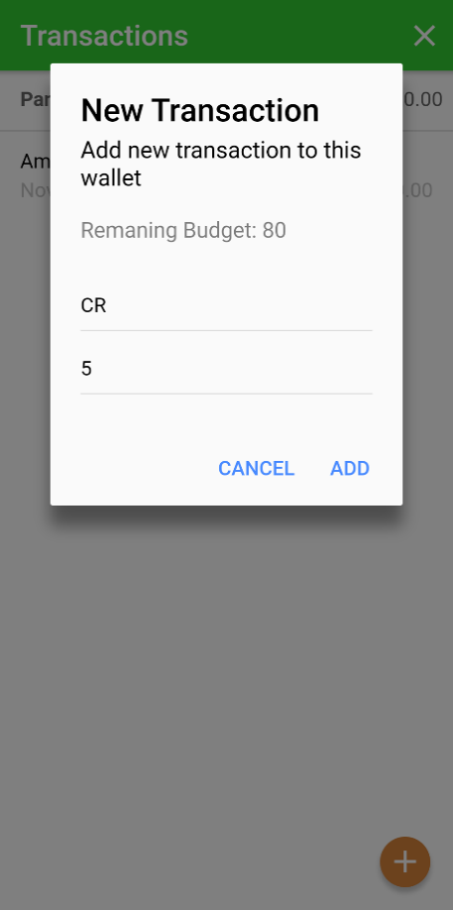
|  |  |  |  |
| --- | --- | --- | --- |
| Task | Object | Event | Event Details |
| Buttons | | | |
| The user taps on Side menu button | Side menu | Tap | Displays Side Navigation/Menu |
| The user taps on ‘Expenses’ segment button | Expenses | Tap | Displays user’s current expense wallets |
| The user taps on ‘Savings’ segment button | Savings | Tap | Displays user’s current savings wallets |
| The user taps on ‘Bills’ segment button | Bills | Tap | Displays user’s current bills wallets |
| The user taps on ‘Home’ tabs button | Home icon | Tap | Redirects user to *wallets* page |
| The user taps on ‘Challenges’ tabs button | Trophy icon | Tap | Redirects user to *saving challenge* page |
| The user taps on ‘Overview’ tabs button | Paper icon | Tap | Redirects user to *Budget Overview* page |
| The user taps on ‘Transaction’ button | Bell icon | Tap | Displays current month expenses/transactions |
| The user taps on ‘Add wallet’button | Fab button | Tap | Redirects user to *add wallet* page |
| List item | | | |
| The user taps on a wallet from the list | List item | Tap | Displays wallet’s transactions (expense)  Displays wallet’s deposits (savings) |

1. **Add wallet –** allows the user to add a new wallet.



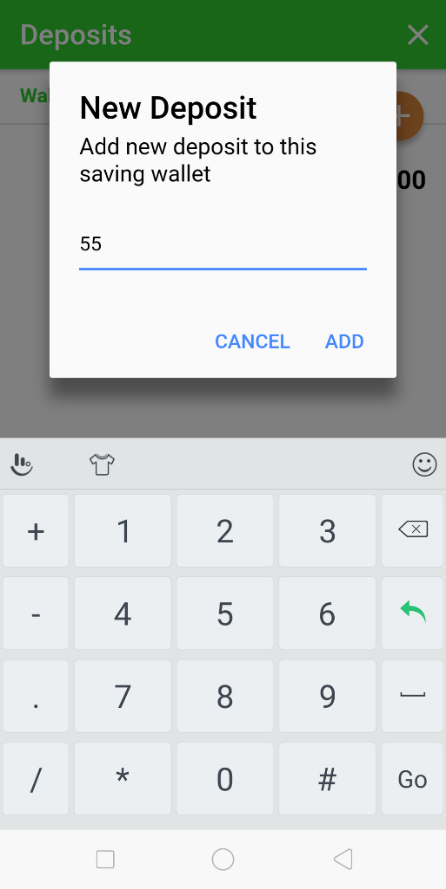
|  |  |  |  |
| --- | --- | --- | --- |
| Task | Object | Event | Event Details |
| Buttons | | | |
| The user taps on ‘Add wallet’ button | Add wallet | Tap | Submits add wallet form. Saves new wallet to the database |
| The user taps on ‘Close’ button | Close | Tap | Dismisses the *add wallet* form |
| Text Fields | | | |
| The user enters wallet name | Wallet name | Submit | Saves submitted wallet name |
| The user enters budget/goal | Budget(Expense)  Goal(Savings) | Submit | Saves submitted wallet budget(expense), goal(savings) |
| Combo box | | | |
| The user selects category (Expense) | Category | Tap | Saves selected category (expense) |
| Toggle | | | |
| The user selects to set notification for the wallet (Bills category) | Notification toggle | Tap | Displays notification form (Bills category) |

1. **Add Transaction** – Allows the user to add a transaction of expense.



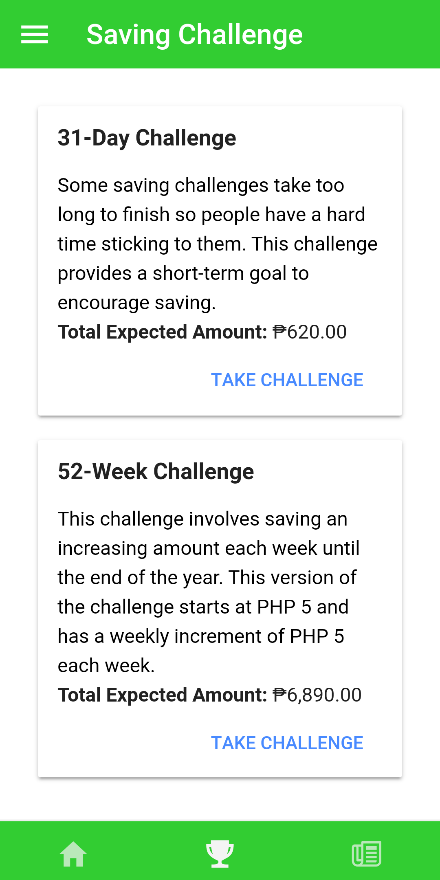
|  |  |  |  |
| --- | --- | --- | --- |
| Task | Object | Event | Event Details |
| Buttons | | | |
| The user taps on ‘Add’ button | Add Transaction | Tap | Submits new transaction form then saves new wallet transaction. |
| The user taps on ‘Cancel’ button | Close | Tap | Dismisses the *add transaction* form |
| The user taps the floating button | Fab button | Tap | Opens the add transaction form |
| Text Fields | | | |
| The user enters description | Description | Submit | Saves submitted wallet description |
| The user enters amount | Amount | Submit | Saves submitted transaction amount |

1. **Add Savings** - Allows the user to add a transaction of savings.



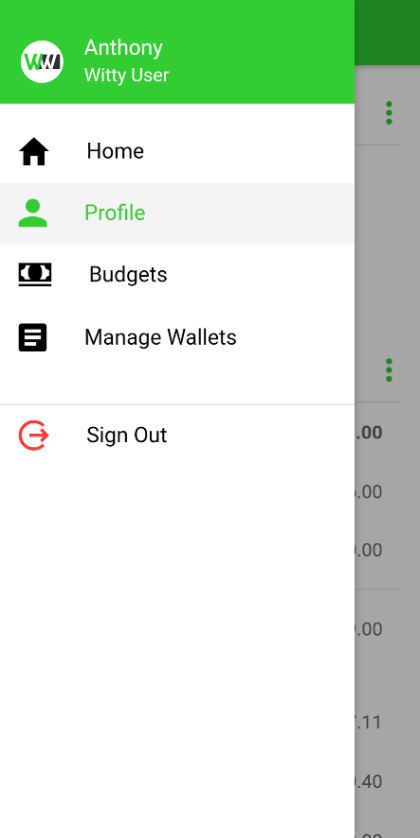
|  |  |  |  |
| --- | --- | --- | --- |
| Task | Object | Event | Event Details |
| Buttons | | | |
| The user taps on ‘Add’ button | Add Transaction | Tap | Submits add savings form. Saves new wallet deposit to the database |
| The user taps on ‘Cancel’ button | Close | Tap | Dismisses the *add transaction* form |
| The user taps the floating button | Fab button | Tap | Opens the add deposit form |
| Text Fields | | | |
| The user enters amount | Amount | Submit | Saves submitted deposit amount |

1. **Saving Challenge –** allows the user to take a saving challenge.



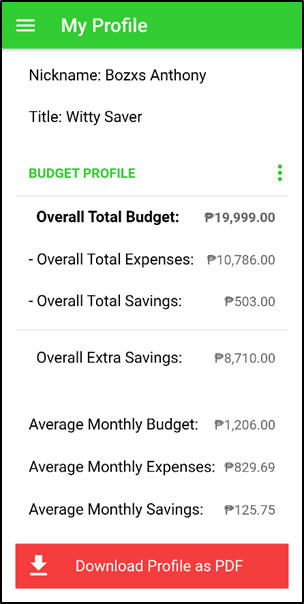
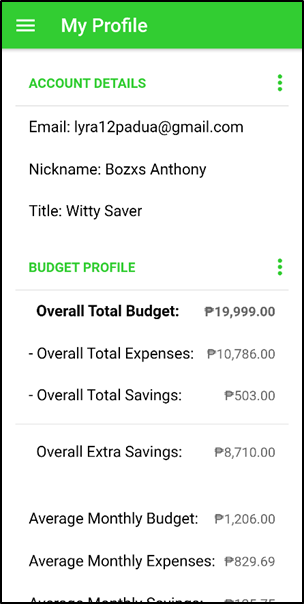
|  |  |  |  |
| --- | --- | --- | --- |
| Task | Object | Event | Event Details |
| Buttons | | | |
| The user taps on ‘Take challenge button’ button | Take challenge | Tap | Starts a saving challenge |

1. **Side Menu** – allows the user to view Side menu.



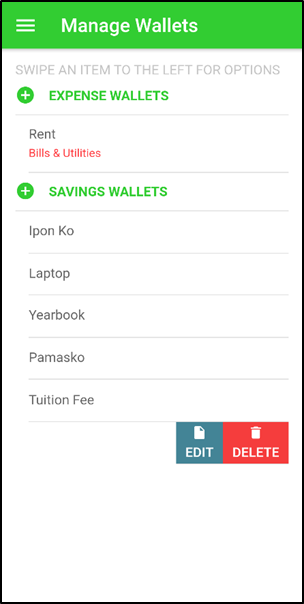
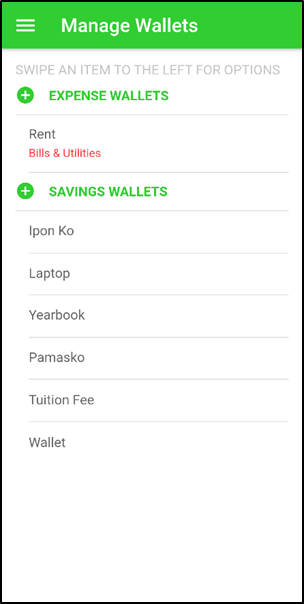
|  |  |  |  |
| --- | --- | --- | --- |
| Task | Object | Event | Event Details |
| Buttons | | | |
| The user taps on Home | Home | Tap | Redirects user to the *home* page |
| The user taps on ‘Profile’ | Profile | Tap | Redirects user to the *profile* page |
| The user taps on ‘Budgets’ | Budets | Tap | Redirects user to budgetspage (history and budget trend) |
| The user taps on ‘Manage Wallets’ | Manage Wallet | Tap | Redirects user to *manage wallets* page (list of wallets) |
| The user taps on ‘Sign Out’ tabs button | Sign Out | Tap | Logs out user and redirects to sign-in page |

1. **Profile –** allows the user to view his/her account details and budget profile.



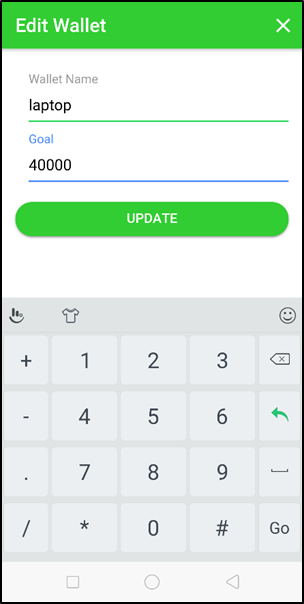
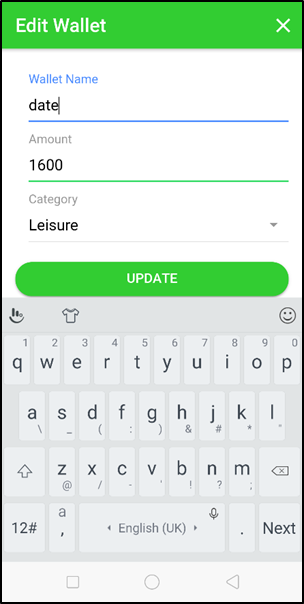
|  |  |  |  |
| --- | --- | --- | --- |
| Task | Object | Event | Event Details |
| Buttons | | | |
| The user taps on ‘Download Profile’ button | Download | Tap | Downloads the user profile as pdf and saves it in mobile device |
| The user taps on ‘More’ button (Account details) | More | Tap | Displays list of editing functions for account details (Edit nickname, change password) |
| The user taps on ‘More’ button (Budget Profile) | More | Tap | Displays a brief description of the budget profile |

1. **Manage wallets page –** allows the user to edit and delete existing wallets and add new wallets.



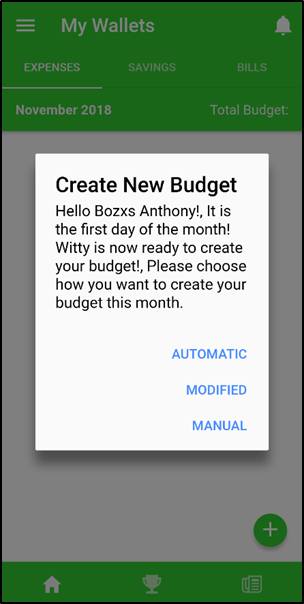
|  |  |  |  |
| --- | --- | --- | --- |
| Task | Object | Event | Event Details |
| Buttons | | | |
| The user taps on the ‘Add’ button | Add | Tap | Redirects user to *add wallet* page |
| The user taps on ‘Edit’ button | Edit | Tap | Redirects user to *edit wallet* page |
| The user taps on ‘Edit’ button | Delete | Tap | Deletes selected wallet |
| List item | | | |
| The user swipes a selected list item | Wallet | Swipe | Displays edit and delete buttons for selected wallet |

1. **Edit wallet –** allows the user to edit a selected wallet.



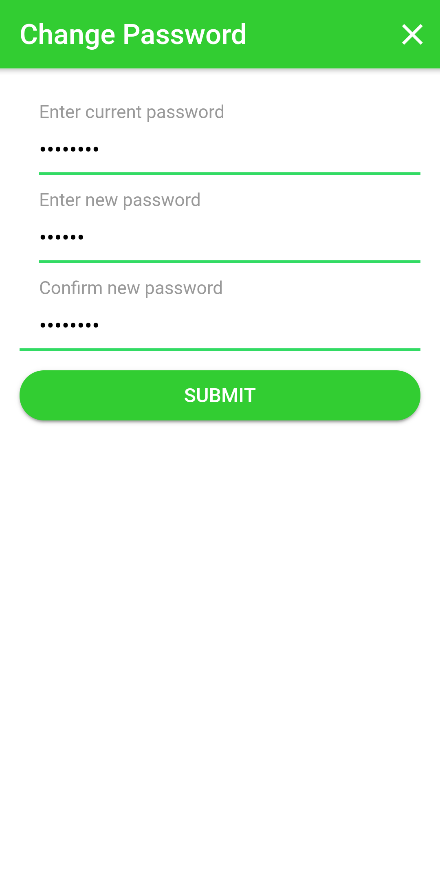
|  |  |  |  |
| --- | --- | --- | --- |
| Task | Object | Event | Event Details |
| Buttons | | | |
| The user taps on the ‘Update’ button | Update | Tap | Submits edit wallet form. Updates selecetd wallet |
| The user taps on ‘Close’ button | Close | Tap | Dismisses the edit wallet form |
| Text Fields | | | |
| The user edits the wallet name | Wallet name | Submit | Updates selected wallet’s name |
| The user edits budget/goal | Budget(Expense)  Goal(Savings) | Submit | Updates selected wallet’s budget(expense), goal(savings) |
| Combo box | | | |
| The user selects a category | Category(Expense) | Submit | Updates selected wallet’s category(expense) |

1. **Create budget –** allows the user to create a new budget.



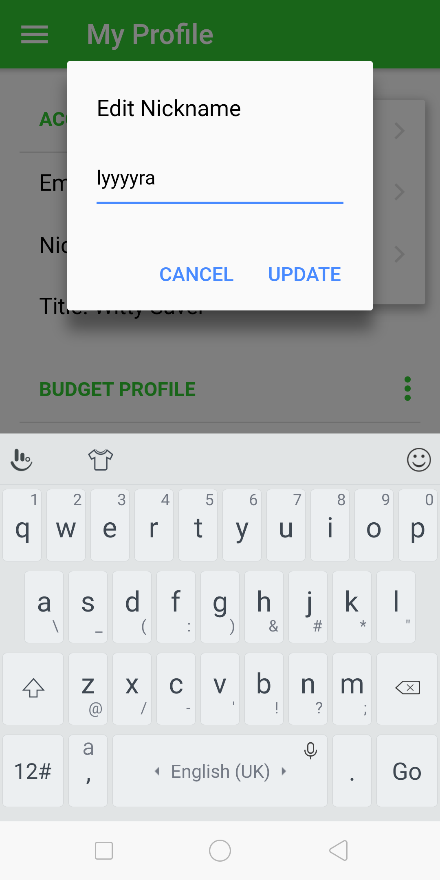
|  |  |  |  |
| --- | --- | --- | --- |
| Task | Object | Event | Event Details |
| Buttons | | | |
| The user taps on the ‘Automatic’ button | Automatic | Tap | Creates month budget automatically using the ML algorithm with the user’s historical data |
| The user taps on ‘Modified’ button | Modified | Tap | Redirects user to *create budget* page where the user can modify what wallets he/she wants to keep, afterwhich, the app will set wallet amounts using the ML algorithm with the user’s historical data |
| The user taps on ‘Manual’ button | Manual | Tap | Dismisses the prompt and lets the user start anew. (previous wallets will not be kept) |

1. **Change Password –** allows the user to change his/her password.



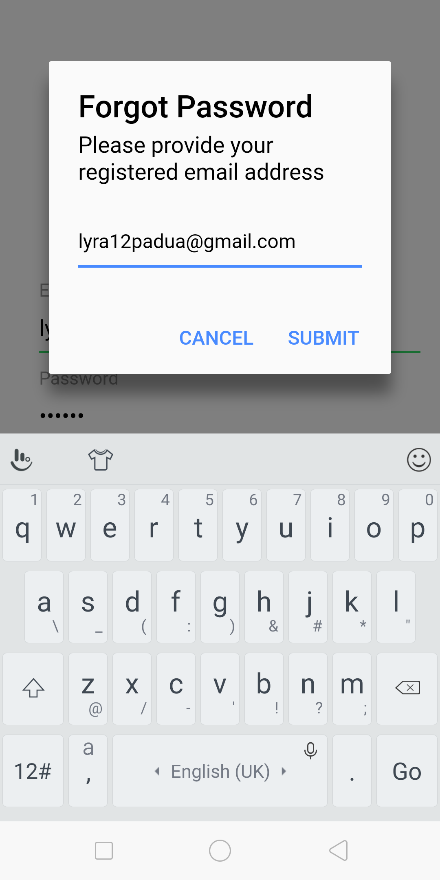
|  |  |  |  |
| --- | --- | --- | --- |
| Task | Object | Event | Event Details |
| Buttons | | | |
| The user taps on the ‘Submit’ button | Submit | Tap | Submits change password form. Updates user password |
| The user taps on ‘Close’ button | Close | Tap | Dismisses the change password form |
| Text Fields | | | |
| The user enters the current password | Current password | Submit | Validates user’s current password |
| The user enters the new password | New password | Submit | Updates user’s password |
| The user enters confirmation for the new password | Confirm new password | Submit | Validates new password entered |

1. **Edit nickname –** allows the user to edit his/her nickname.



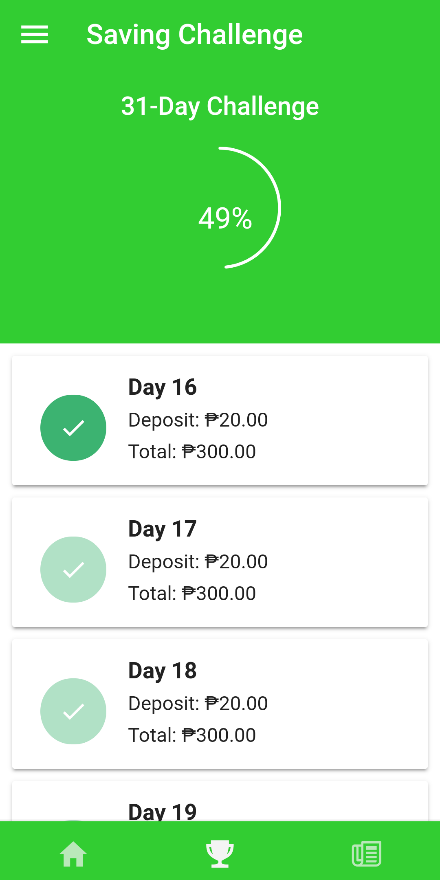
|  |  |  |  |
| --- | --- | --- | --- |
| Task | Object | Event | Event Details |
| Buttons | | | |
| The user taps on the ‘Submit’ button | Submit | Tap | Submits the forgot password request form |
| The user taps on ‘Cancel’ button | Cancel | Tap | Dismisses the forgot password form |
| Text Fields | | | |
| The user enters registered email address | Email | Submit | Validates submitted email |

1. **Forgot password –** allows the user to reset his/her password.



|  |  |  |  |
| --- | --- | --- | --- |
| Task | Object | Event | Event Details |
| Buttons | | | |
| The user taps on the ‘Submit’ button | Submit | Tap | Submits the forgot password request form |
| The user taps on ‘Cancel’ button | Cancel | Tap | Dismisses the forgot password form |
| Text Fields | | | |
| The user enters registered email address | Email | Submit | Validates submitted email |

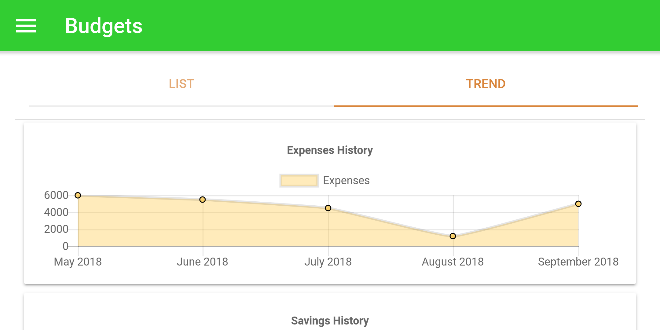
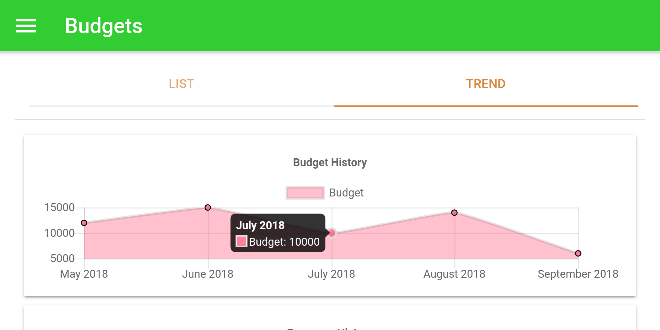
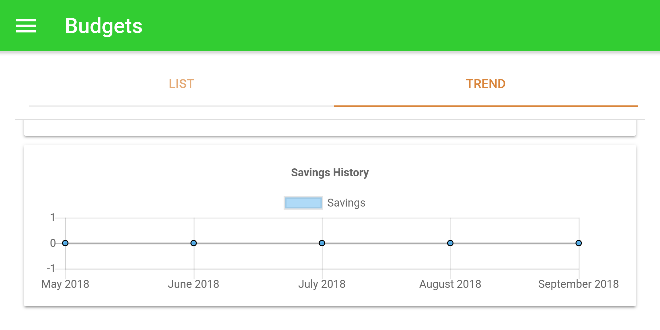
1. **Deposit –** allows the user to deposit to active saving challenge.



|  |  |  |  |
| --- | --- | --- | --- |
| Task | Object | Event | Event Details |
| Buttons | | | |
| The user taps on the ‘Deposit’ button | Deposit | Tap | Updates the challenge |

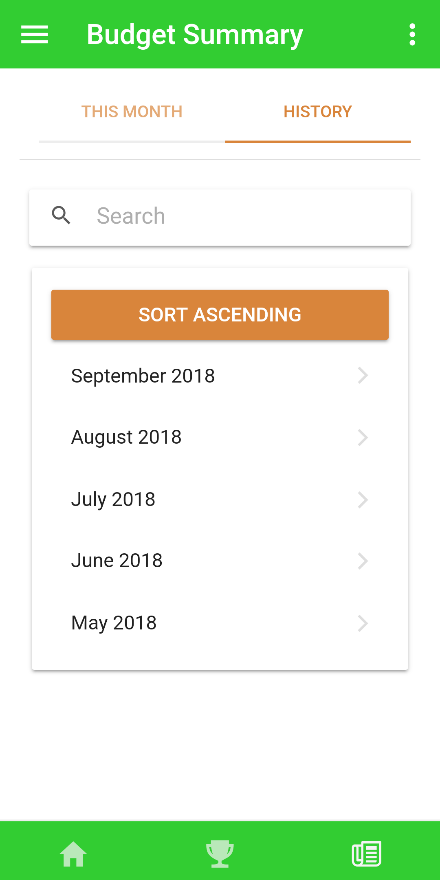
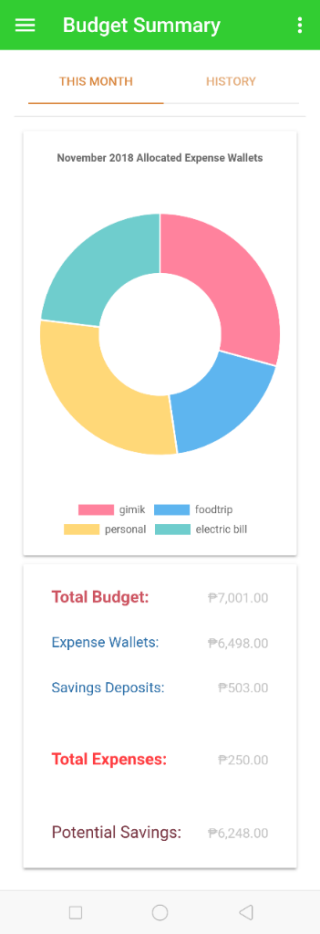
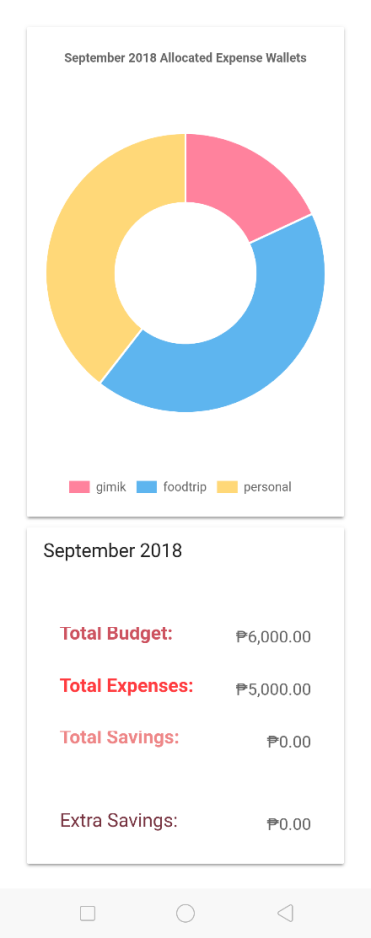
1. **Report Design**

**6.** **1 Budgets –** allows the user to view his/her previous budget predictions and budget trends.



|  |  |  |  |
| --- | --- | --- | --- |
| Task | Object | Event | Event Details |
| Buttons | | | |
| The user taps on the ‘List’ segment button | List | Tap | Displays list of previous budget predictions |
| The user taps on ‘Trend’ segment button | Trend | Tap | Displays user’s budget trend in chart form |

**6.** **2 Budget Summary –** allows the user to view the current budget and previous budgets overview.

|  |  |  |  |
| --- | --- | --- | --- |
| Task | Object | Event | Event Details |
| Buttons | | | |
| The user taps on ‘This month’ segment button | This month | Tap | Displays the current month budget overview |
| The user taps on ‘History’ segment button | History | Tap | Displays previous months list |
| List item | | | |
| The user selects a month/period from the list | List item (months) | Tap | Displays the selected month’s budget overview |

**APPENDIX K**

**SOFTWARE PROJECT TEST PLAN (SPTP)**



**Witty Wallet:  
A Personal Finance Management Application**

**Version 0.3**

**Prepared by:**

Lysle L. Baday

Anthony D. Ballug Jr.

Jonan V. Bie

Lyra Angelica S. Padua

**Revision History**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Version #** | **Implemented**  **By** | **Revision**  **Date** | **Approved**  **By** | **Approval**  **Date** | **Reason** |
| 0.1 | Lyra Padua |  |  | 10/22/18 | Test Plan draft |
| 0.2 | Lyra Padua | 11/26/18 |  | 12/01/2018 | Revision |

**Table of Contents**

[**1. Introduction 191**](#_Toc511416322)

[1.1 Purpose 191](#_Toc511416323)

[1.2 Audience 191](#_Toc511416324)

[1.3 Test Objectives 191](#_Toc511416325)

[1.4 Scope 191](#STP_Scope)

[1.5 Relationship to other documents 192](#_Toc511416332)

[**2. Test Strategy 194**](#_Toc511416335)

[**3. Pass/Fail criteria 194**](#_Toc511416336)

[3.1 Entry Criteria 194](#_Toc511416337)

[3.2 Exit Criteria 195](#_Toc511416338)

[3.3 Suspension Criteria 195](#_Toc511416339)

[3.4 Resumption Criteria 195](#_Toc511416340)

[**4. Test Methodology 196**](#_Toc511416341)

[4.1 Testing Environment 196](#_Toc511416342)

[4.2 General Test Strategy 196](#_Toc511416346)

[4.3 Test Result Recording 197](#_Toc511416347)

[4.4 Roles and Responsibilities 197](#_Toc511416348)

[**5. Additional Considerations 198**](#_Toc511416351)

[**6. Testing schedule 198**](#_Toc511416352)

* 1. [Test Schedule 198](#_Toc511416353)

1. **Introduction**
2. **Purpose**

The Test Plan document serves as a blueprint that describes how the testing activity of Witty Wallet will be performed. The Test Plan document includes test strategy, test methodology, and testing schedule.

1. **Audience**

The target audience of this document is the project team and the stakeholders (user and panel reviewers). The project team plans for the testing activities and performs tasks specified in this document. The stakeholders provide input and recommendations on this document to ensure accountability. The document is intended to accomplish its purpose only for the intended audiences.

1. **Test Objectives**

This part discusses the objectives of the tests. The goal is to validate and verify the functionalities of the system.

1. **Provide adequate coverage metrics**

* This will allow the audience to analyze, track, and measure the effectiveness of the project by checking the processes for errors and how it performs on different environments.

1. **Validate and verify system response**

* This will be implemented for use cases under varying workload conditions.

1. **Ensure stakeholder requirements for quality**

* Standards for data quality should be established such that sufficient data is provided for those making the decision to release by conforming to system benchmarks and adhering to stakeholder policies.

1. **Scope**

The scope of the tests identifies all the features and combinations of features of the project to be tested. Testing operations will be performed manually by the intended audience.

* + 1. **Features to be tested**

The features to be tested are the functions of the Witty Wallet application: saving challenges, monthly budget suggestions, tracking and monitoring of bills, budgets, debts, expenses, income, and savings.

* + 1. **Items that will not be tested**
* Infrastructure components

The internet/WIFI backbone will be used during the testing activities. However, no tests will be written and executed to directly test the communications backbone.

* 1. **Relationship to other documents**

This part discusses the relationship of the test plan to the other documents produced during the development phases.

1. **Relationships to documents**

Black box/Functionality tests relating to use cases are developed from the use case descriptions in the SRS (software requirements specification) document.

Performance tests derived from nonfunctional requirements are developed from the nonfunctional requirements in the Software Design Document (SDD).

Unit/component and Integration tests are developed from the SDD. The integration tests generally come from the overall package diagram describing the architecture of the system. The architecture is also used to help in determining the integration test approach. The test environment (hardware/software) is also derived from the SDD.

1. **Test naming scheme**

The names of test cases will indicate from where they have been derived using a system of prefixes. The following are the prefixes:

|  |  |
| --- | --- |
| **USE CASE** | **FUNCTION** |
| WW-UC01 | Create Account |
| WW-UC02 | Login |
| WW-UC03 | Create Budget |
| WW-UC04 | View Profile |
| WW-UC05 | Edit Nickname |
| WW-UC06 | Add Wallet |
| WW-UC07 | View Wallet List |
| WW-UC08 | Edit Wallet |
| WW-UC09 | Delete Wallet |
| WW-UC10 | Add Transaction |
| WW-UC11 | Take Saving Challenge |
| WW-UC12 | Deposit |
| WW-UC13 | Add Savings |
| WW-UC14 | Budget Overview |
| WW-UC15 | Project Budget |
| WW-UC16 | Change Password |
| WW-UC17 | Forgot Password |
| WW-UC18 | Logout |

1. **Test Strategy**

The test cases are divided according to User Class processes:

|  |  |  |
| --- | --- | --- |
| **REQUIREMENT ID** | **FUNCTION/USE CASE** | **RELEASE SCHEDULE** |
| **USR** | | |
| USR-WW-UC01 | Create Account | Release 1 |
| USR-WW-UC02 | Login | Release 1 |
| USR-WW-UC03 | Create Budget | Release 1 |
| USR-WW-UC04 | View Profile | Release 1 |
| USR-WW-UC05 | Edit Nickname | Release 1 |
| USR-WW-UC06 | Add Wallet | Release 1 |
| USR-WW-UC07 | View Wallet List | Release 1 |
| USR-WW-UC08 | Edit Wallet | Release 1 |
| USR-WW-UC09 | Delete Wallet | Release 1 |
| USR-WW-UC10 | Add Transaction | Release 1 |
| USR-WW-UC11 | Take Saving Challenge | Release 1 |
| USR-WW-UC12 | Deposit | Release 1 |
| USR-WW-UC13 | Add Savings | Release 1 |
| USR-WW-UC14 | Generate Report | Release 1 |
| USR-WW-UC15 | Project Budget | Release 1 |
| USR-WW-UC16 | Change Password | Release 1 |
| USR-WW-UC17 | Forgot Password | Release 1 |
| USR-WW-UC18 | Logout | Release 1 |

The testing will be concentrated on the functionalities defined in the Release 1 of Witty Wallet Application.

1. **Pass/Fail criteria**
   1. **Entry Criteria**

The following Entry Criteria are supposed to be true before any testing stage starts:

* + 1. The Test Case created by the developers and be reviewed and approved by the technical adviser
    2. The version of Witty Wallet application is built for testing
    3. The test environment has been setup in conformity with the requirements for the infrastructure, functionality and test data sets, and the intake of this environment has been successfully concluded
    4. All developed code must have undergone unit/component testing.
  1. **Exit Criteria**

The following Exit Criteria are supposed to be true before the unit test starts:

1. All major issues regarding construction and testing are resolved.
2. All expected and actual results are captured and documented with the Test Cases.
3. There are no functional defects outstanding that bring unacceptable business risks when Witty Wallet goes into production.
   1. **Suspension Criteria**

The Test Team may discontinue partial or full testing activities if any of the following happens:

* + 1. When the environment is not viable for testing.
    2. A defect is encountered that cannot allow further testing.
    3. A deficit in resources required by the testing team (e.g. Smartphones)
    4. Sudden changes in stakeholder requirements.

1. **Resumption Criteria**

The Test Team may discontinue partial or full testing activities if any of the following happens:

* + 1. The testing environment is already viable
    2. When a fix is successfully implemented and the Testing Team is notified to continue testing
    3. Complete resources are provided for
    4. Change in requirements discussed with the stakeholders and negotiated extension in delivery

1. **Test Methodology**

This part describes the supposed approach to the testing process. It discusses the reasons for the selected testing strategy.

1. **Testing Environment**

This part identifies the resources that are needed for testing.

1. **Facilities required**

WLAN Facility is needed to provide access to the application.

1. **Hardware required**

A Smartphone is used to test the function and performance of the application.

1. **Software required**

The Software requirements stated in the SDD will be used for the test execution.

1. **General Test Strategy**
2. **White Box test**

White Box testing examines the program structure and derives test data from the program logic/code. System/Functionality test will cover functionality, performance and requirements validation tests. It will be done by validating functionality based on requirements defined in the SRS. The White box test is done by the project team.

1. **Black Box Test**

Black Box Testing, either functional or non-functional, tests without reference to the internal structure of the component or system.

User Acceptance Testers will be conducting the test as they have no knowledge of the internal structure of the system. The UAT will be using the same test case template as the system testers. The testers will be looking for:

* Interface errors
* Missing functions
* Design flaws

1. **Test Result Recording**

Test results from the UA testers will be conducted by the project team. The documentation process will be handled by the test team through feedback form (Appendix D). The documentation is done at the same time as the testing.

No internal structure information is given to the UA testers from the developers all throughout the test.

1. **Roles and Responsibilities**

This part lists down the human resources needed so the testing will be conducted systematically.

* + - 1. **Description of Roles and Responsibilities**

The Test Lead is responsible for oversight of testing on the project. This person is also accountable for the processes used to ensure the quality of the deliverable. The Test Lead is also is accountable for conducting quality assurance testing and executing on the test plan.

The Developer is responsible for the development of the creation and execution of test scripts. They are also responsible being the functional testers of the project.

During the User Acceptance Test, the stakeholders will test the software to make sure it can handle required tasks in real-world scenarios, according to specifications.

1. **Resources**

Below is a table containing the names of the persons who will be taking the responsibilities stated above.

|  |  |  |
| --- | --- | --- |
| **Role** | **Resource Assigned** | **Duration** |
| Test Lead | Jonan Bie | All throughout |
| Developers | Lysle Baday, Anthony Ballug  Jr., Jonan Bie, Lyra Padua | All throughout |
| Unit Testers | Ralph Cecilio, Alice Smith | All throughout |
| System Integration Tester/Programming Lead | Anthony Ballug Jr. | All throughout |
| Requirements Analysts/System Testers | Erna-kristi Martinez, Aguilar, Divine, Ellen Halover, Hydi Toyeng,  Lysle Baday, Anthony Ballug  Jr., Jonan Bie, Lyra Padua | All throughout |

1. **Additional Considerations**

This part identifies the set of users who will be testing the functionalities of the application. They will give the approval for deployment if the functionalities exhibited by the system really fit their needs as individual users of the Witty Wallet.

1. **Testing schedule**

This part of the test plan covers the test schedule, responsibilities, staffing needs, risks and contingencies, and other additional considerations.

1. **Test Schedule**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **1st Release** |  |  |  |  |
| **Test Phase** | **Time** | **Date Started** | **Date Ended** | **Owner** |
| Test Plan Creation | 3 days | 10/14/18 | 03/16/18 | Test Lead, Developers |
| Unit Testing | 1 week | 10/16/18 | 10/22/18 | Test Lead, Developers |

**APPENDIX L**

**SYSTEM PROJECT TEST CASE**



**Witty Wallet:**

**A Personal Finance Management Application**

**Prepared by:**

Lysle L. Baday

Anthony D. Ballug Jr.

Jonan V. Bie

Lyra Angelica S. Padua

**WW-TCO1-CREATE ACCOUNT**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Case ID:** | WW-TC01 | | | |
| **Test Case Description:** | Allows the user to test creating an account | | | |
| **Positions:** | User | | | |
| **Function/Features:** | Create Account | | | |
| **Tested** **By**: | Anthony Ballug Jr. | | | |
| **Test** **Sequence**: | | | | |
| **Steps** | **Data** | **Expected** **Result** | **Actual** **Result** | **Remarks** |
| Enters valid and complete data on all the fields and submits form | Email: test@gmail.com  Nickname: test  Password: test12 | Redirects user back to login page and displays notification message: "Registration Successful!”. Display instructions to activate account. | Redirected user back to login page and displays notification message: "Registration Successful!”.  Display instructions to activate account. | Passed |
| Submits an existing email | Email: anthony@gmail.com | Displays error message that email already exist in the database | Displayed error message that email already exist in the database | Passed |

**WW-TCO2-LOGIN**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Case ID:** | WW-TC02 | | | |
| **Test Case Description:** | Allows the user to test logging in to the system | | | |
| **Positions:** | User | | | |
| **Function/Features:** | Login | | | |
| **Tested** **By**: | Lyra Padua | | | |
| **Test** **Sequence**: | | | | |
| **Steps** | **Data** | **Expected** **Result** | **Actual** **Result** | **Remarks** |
| Submits valid email and password | Email:  test@gmail.com  Password: test12 | Directs user to the home page | Directed user to the home page | Passed |
| Submits invalid email and or password | Email:  zigmahbaguio@yahoo.com  Password:  1 | Displays an error message that user is not found | Displayed an error message that user is not found | Passed |
| Submits non-existing email | Email:  zigmahbb@email.com  Password:  1 | Displays an error message that user is not found | Displayed an error message that user is not found | Passed |
| Submits a not yet activated email account | Email:  lyra@yahoo.com  Password:  1 | Displays an error message that user is not yet activated | Displayed an error message that user is not yet activated | Passed |

**WW-TCO3- CREATE BUDGET**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Case ID:** | WW-TC03 | | | |
| **Test Case Description:** | Allows the user to test creating a budget | | | |
| **Positions:** | User | | | |
| **Function/Features:** | Create Budget | | | |
| **Tested** **By**: | Lyra Padua | | | |
| **Test** **Sequence**: | | | | |
| **Steps** | **Data** | **Expected** **Result** | **Actual** **Result** | **Remarks** |
| Selects Automatic option |  | Displays brief explanation and confirmation of selected option | Displayed brief explanation and confirmation of selected option | Passed |
| User confirms Automatic option |  | Automatically creates wallets(budget) | Automatically created wallets(budget) | Passed |
| Selects Modified option |  | Displays brief explanation and confirmation of selected option | Displayed brief explanation and confirmation of selected option | Passed |
| User confirms Modified option |  | Redirects user to Budget creation page for editing of wallets to be kept | Redirected user to Budget creation page for editing of wallets to be kept | Passed |
| User modifies wallets to keep and selects next step |  | Displays wallets with predicted values | Displayed wallets with predicted values | Passed |
| User selects to save |  | Saves new wallets to the database | Saved new wallets to the database | Passed |
| Selects Manual option |  | Displays brief explanation and confirmation of selected option | Displayed brief explanation and confirmation of selected option | Passed |
| User confirms Manual option |  | Dismisses budget creation prompt | Dismisses budget creation prompt | Passed |
| User selects to deposit |  | Displays lists of wallets for allocation | Displayed lists of wallets | Passed |
| User declines to deposit |  | Dismisses Budget Creation modal | Dismissed Budget Creation modal | Passed |
| User selects wallet |  | Allocates extra savings to certain wallets | Allocated extra savings to certain wallets | Passed |

**WW-TCO4-VIEW PROFILE**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Case ID:** | WW-TC04 | | | |
| **Test Case Description:** | Allows the user to test viewing his/her profile | | | |
| **Positions:** | User | | | |
| **Function/Features:** | View Profile | | | |
| **Tested** **By**: | Anthony Ballug Jr. | | | |
| **Test** **Sequence**: | | | | |
| **Steps** | **Data** | **Expected** **Result** | **Actual** **Result** | **Remarks** |
| User selects to view profile |  | Redirects user to profile page | Redirected user to profile page | Passed |

**WW-TCO5-EDIT NICKNAME**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Case ID:** | WW-TC05 | | | |
| **Test Case Description:** | Allows the user to test editing his/her Nickname | | | |
| **Positions:** | User | | | |
| **Function/Features:** | Edit Nickname | | | |
| **Tested** **By**: | Lyra Padua | | | |
| **Test** **Sequence**: | | | | |
| **Steps** | **Data** | **Expected** **Result** | **Actual** **Result** | **Remarks** |
| Select Edit Nickname |  | Displays Edit Nickname form | Displayed Edit Nickname form | Passed |
| Enters valid data and submits form | Nick Name:  Falalayl | Updates user’s nickname in the database and displays a message that Nickname has been successfully updated | Updated user’s nickname in the database and displayed a message that Nickname has been successfully updated | Passed |
| Enters null data | Nick Name:  <null> | Displays error message that nickname must not be empty | Displayed error message that nickname must not be empty | Passed |
| Cancels editing nickname |  | Dismisses form | Dismissed form | Passed |

**WW-TCO6-ADD WALLET**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Case ID:** | WW-TC06 | | | |
| **Test Case Description:** | Allows the user to test adding a wallet | | | |
| **Positions:** | User | | | |
| **Function/Features:** | Add Wallet | | | |
| **Tested** **By**: | Anthony Ballug Jr. | | | |
| **Test** **Sequence**: | | | | |
| **Steps** | **Data** | **Expected** **Result** | **Actual** **Result** | **Remarks** |
| Selects to Add Wallet (Expense) |  | Displays add wallet modal | Redirected user to adding wallet page | Passed |
| Enters valid and complete data on all the fields and submits form | Wallet Name:  Food  Budget: 1000  Category: Food | Displays a message that an expense wallet has been successfully added | Displayed a message that an expense wallet has been successfully added | Passed |
| Cancels adding wallet |  | Dismisses add wallet modal | Dismissed add wallet modal | Passed |
| Selects to Add Wallet (Savings) |  | Displays add wallet modal | Redirected user to adding wallet page | Passed |
| Enters valid and complete data on all the fields and submits form | Wallet Name:  Food  Goal: 1000 | Displays a message that an expense wallet has been successfully added | Displayed a message that an expense wallet has been successfully added | Passed |
| Cancels adding wallet |  | Dismisses add wallet modal | Dismissed add wallet modal | Passed |

**WW-TCO7-VIEW WALLET LIST**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Case ID:** | WW-TC07 | | | |
| **Test Case Description:** | Allows the user to test viewing of Wallet list | | | |
| **Positions:** | User | | | |
| **Function/Features:** | View Wallet | | | |
| **Tested** **By**: | Lyra Padua | | | |
| **Test** **Sequence**: | | | | |
| **Steps** | **Data** | **Expected** **Result** | **Actual** **Result** | **Remarks** |
| Selects to view Manage Wallet |  | Redirects user to wallets list page | Redirected user to wallets list page | Passed |

**WW-TC08-EDIT WALLET**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Case ID:** | WW-TC08 | | | |
| **Test Case Description:** | Allows the user to test editing of wallet name | | | |
| **Positions:** | User | | | |
| **Function/Features:** | Edit Wallet | | | |
| **Tested** **By**: | Anthony Ballug Jr. | | | |
| **Test** **Sequence**: | | | | |
| **Steps** | **Data** | **Expected** **Result** | **Actual** **Result** | **Remarks** |
| Selects a wallet to be edited |  | Displays edit wallet modal | Displayed edit wallet modal | Passed |
| Enters valid data and submits form | Wallet Name: Pamasko  Amount: 4500 | Displays a message that wallet has been successfully updated | Displayed a message that wallet has been successfully updated | Passed |
| Cancels editing wallet name |  | Dismisses edit wallet modal | Dismissed edit wallet modal | Passed |

**WW-TC09-DELETE WALLET**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Case ID:** | WW-TC09 | | | |
| **Test Case Description:** | Allows the user to test deleting of wallet | | | |
| **Positions:** | User | | | |
| **Function/Features:** | Delete Wallet | | | |
| **Tested** **By**: | Lyra Padua | | | |
| **Test** **Sequence**: | | | | |
| **Steps** | **Data** | **Expected** **Result** | **Actual** **Result** | **Remarks** |
| Selects a wallet to be deleted |  | Asks for confirmation | Asked for confirmation | Passed |
| Confirms to delete wallet |  | Deletes wallet and displays a message that wallet has been successfully deleted | Deleted wallet and displays a message that wallet has been successfully deleted | Passed |
| Cancels deleting wallet |  | Dismisses confirmation | Dismisses confirmation | Passed |

**WW-TC10-ADD TRANSACTION**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Case ID:** | WW-TC10 | | | |
| **Test Case Description:** | Allows the user to test adding of transaction | | | |
| **Positions:** | User | | | |
| **Function/Features:** | Add Transaction | | | |
| **Tested** **By**: | Anthony Ballug Jr. | | | |
| **Test** **Sequence**: | | | | |
| **Steps** | **Data** | **Expected** **Result** | **Actual** **Result** | **Remarks** |
| Selects an expense wallet to add transaction |  | Displays list of transactions of the selected expense wallet | Displayed transaction page of the selected expense wallet | Passed |
| Selects to add Transaction |  | Displays add transaction form | Displays add transaction form | Passed |
| Enters valid data and submits form | Description:  Taxi  Amount: 200 | Displays a message that transaction has been successfully added. Updates transaction list | Displayed a message that transaction has been successfully added. Updated transaction list | Passed |
| Enters null data | Description:  <null>  Amount:  <null> | Display error message that fields must not be empty | Displayed error message that fields must not be empty | Passed |
| Cancels to add transaction |  | Dismisses adding form | Dismisses adding form | Passed |

**WW-TC11-TAKE SAVING CHALLENGE**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Case ID:** | WW-TC11 | | | |
| **Test Case Description:** | Allows the user to test taking of saving challenge | | | |
| **Positions:** | User | | | |
| **Function/Features:** | Saving Challenge | | | |
| **Tested** **By**: | Lyra Padua | | | |
| **Test** **Sequence**: | | | | |
| **Steps** | **Data** | **Expected** **Result** | **Actual** **Result** | **Remarks** |
| Select a saving Challenge |  | Displays saving challenge description |  |  |
| Select to take challenge |  | Activates saving challenge and displays a message that user has accepted the saving challenge |  |  |
| Cancels to take challenge |  | Redirects user to the saving challenge page |  |  |

**WW-TC12-DEPOSIT**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Case ID:** | WW-TC12 | | | |
| **Test Case Description:** | Allows the user to test depositing for a saving challenge | | | |
| **Positions:** | User | | | |
| **Function/Features:** | Deposit | | | |
| **Tested** **By**: | Lyra Padua | | | |
| **Test** **Sequence**: | | | | |
| **Steps** | **Data** | **Expected** **Result** | **Actual** **Result** | **Remarks** |
| Selects an active Saving Challenge |  | Displays Saving Challenge Details | Displayed Saving Challenge Details | Passed |
| Deposits amount to active saving challenge |  | Updates progress bar | Updated progress bar | Passed |

**WW-TC13-ADD SAVINGS**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Case ID:** | WW-TC13 | | | |
| **Test Case Description:** | Allows the user to test adding savings | | | |
| **Positions:** | User | | | |
| **Function/Features:** | Add Savings | | | |
| **Tested** **By**: | Anthony Ballug Jr. | | | |
| **Test** **Sequence**: | | | | |
| **Steps** | **Data** | **Expected** **Result** | **Actual** **Result** | **Remarks** |
| Selects a savings wallet to add savings |  | Displays list transactions of the selected savings wallet | Displays list transactions of the selected savings wallet | Passed |
| Selects to add Transaction |  | Displays an add savings form | Displayed an add savings form | Passed |
| Enters valid data and submits form | Amount: 150 | Displays a message that transaction has been successfully added. Updates transactions list | Displayed a message that transaction has been successfully added. Updated transactions list | Passed |
| Enters null data | Amount:  <null> | Displays error message that fields must not be empty | Displayed error message that fields must not be empty | Passed |
| Cancels adding savings |  | Dismisses adding form | Dismissed adding form | Passed |

**WW-TC14-BUDGET OVERVIEW**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Case ID:** | WW-TC14 | | | |
| **Test Case Description:** | Allows the user to test generating report of current month and previous months budget summary | | | |
| **Positions:** | User | | | |
| **Function/Features:** | Generate Report | | | |
| **Tested** **By**: | Anthony Ballug Jr. | | | |
| **Test** **Sequence**: | | | | |
| **Steps** | **Data** | **Expected** **Result** | **Actual** **Result** | **Remarks** |
| Selects Budget Overview tab |  | Generates and displayed budget summary report of current month | Generated and displayed budget summary report of current month | Passed |
| Selects History segment |  | Displays list of budget history | Displayed list of budget history | Passed |
| Select an item (period) from the list |  | Generates and displays budget summary of selected item | Generated and displays budget summary of selected item | Passed |

**WW-TC15-PROJECT BUDGET**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Case ID:** | WW-TC15 | | | |
| **Test Case Description:** | Allows the user to test projecting next month budget | | | |
| **Positions:** | User | | | |
| **Function/Features:** | Project Budget | | | |
| **Tested** **By**: | Lyra Padua | | | |
| **Test** **Sequence**: | | | | |
| **Steps** | **Data** | **Expected** **Result** | **Actual** **Result** | **Remarks** |
| Selects to view Budget prediction |  | Displays computed budget prediction for the next month | Displayed computed budget prediction for the next month | Passed |

**WW-TC16-CHANGE PASSWORD**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Case ID:** | WW-TC16 | | | |
| **Test Case Description:** | Allows the user to test changing his/her password | | | |
| **Positions:** | User | | | |
| **Function/Features:** | Change Password | | | |
| **Tested** **By**: | Lyra Padua | | | |
| **Test** **Sequence**: | | | | |
| **Steps** | **Data** | **Expected** **Result** | **Actual** **Result** | **Remarks** |
| Selects to change Password |  | Displays change password form | Displayed change password form | Passed |
| Enters valid and complete data on all the fields and submits form | Current Password:  tonton  New Password:  anthony  Confirm Password:  anthony | Displays a message that user has successfully changed password. Logs out user | Displayed a message that user has successfully updated password.  Logs out user | Passed |
| Enters wrong/invalid current password | Current Password:  tontonton  New Password:  Anthony  Confirm new Password:  anthony | Displays error message that current password did not match against the database | Displayed error message that current password did not match against the database | Passed |
| Enters new and confirmation of new password that does not match | Current Password:  Tonton  New Password:  anthony  Confirm new Password:  anthonyx | Displays error message that new password and new password confirmation did not match | Displayed error message that new password and new password confirmation did not match | Passed |
| Enters new password that is the same with current password | Current Password:  Tonton  New Password:  Tonton | Displays error message that new password and current password should not be the same | Displayed error message that new password and current password should not be the same | Passed |
| Cancels changing password |  | Dismisses change password form | Dismisses change password form | Passed |

**WW-TC17-FORGOT PASSWORD**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Case ID:** | WW-TC17 | | | |
| **Test Case Description:** | Allows the user to test resetting his/her password | | | |
| **Positions:** | User | | | |
| **Function/Features:** | Forgot Password | | | |
| **Tested** **By**: | Anthony Ballug Jr. | | | |
| **Test** **Sequence**: | | | | |
| **Steps** | **Data** | **Expected** **Result** | **Actual** **Result** | **Remarks** |
| Selects Forgot Password |  | Prompts user to enter email | Prompted user to enter email | Passed |
| Enters registered email and submits form | mibo@gmail.com | Informs user that a mail has been sent for password reset verification | Informed user that a mail has been sent for password reset verification | Passed |
| Enters not existing email and submits form | ton@gmail.com | Displays an error message that email does not exist in the database | Displayed an error message that email does not exist in the database | Passed |
| Confirms password reset verification |  | Updates user’s password and send the temporary password to user’s email address. Informs the user that a new email has been sent to complete the process. | Updated user’s password and send the temporary password to user’s email address. Informed the user that a new email has been sent to complete the process. | Passed |

**WW-TC18-LOGOUT**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Case ID:** | WW-TC18 | | | |
| **Test Case Description:** | Allows the user to test logging out from the system | | | |
| **Positions:** | User | | | |
| **Function/Features:** | Logout | | | |
| **Tested** **By**: | Anthony Ballug Jr. | | | |
| **Test** **Sequence**: | | | | |
| **Steps** | **Data** | **Expected** **Result** | **Actual** **Result** | **Remarks** |
| Selects to Sign out |  | Logs out user from the system and directs user to the login page | Logged out user from the system and directs user to the login page | Passed |

# **APPENDIX M**

**CURRICULUM VITAE**





