FIN: A Money Tracker Mobile Application With Built-In Mobile Decision Support System

BSc (Hons) in Computing
Final Year Project
Final Report
B8RS100

by
Mary Louise Reyes
Student Number : 10298996
10298996@mydbs.ie
Project Supervisor : Rory O'Donnell
Submitted on: April 10, 2019

Abstract

In this project, a cross-platform mobile application which serves as a tool for financial tracking and budgeting is developed. The application enables users to easily document their finances with their mobile phones. The entries are stored in a personal database on each device. The application also has the capability to suggest whether a user should make a purchase or not. This tool is developed with machine learning. A custom dataset is collected for this project and the model was trained with multiple algorithms in order to find the best possible results. Finally, the model is deployed and embedded into the application. With this tool, the application serves as a mobile decision support system for personal purchases.

Acknowledgements

Rory O'Donnell, Damien Kettle, Terri Hoare, Paul Laird, Derek Mizak and Dr. Shahram Azizi Sazi for their incredible support and assistance in completing this project and for the duration of the final year of this degree.

Integrated Facilities Solutions; George Harold, Kieran Beggan, Alan O'Neill, and Philip Spain for providing me a wonderful internship where I have learned critical skills needed to complete the degree and this project.

Assumption Secondary School, Mr. Alan Morgan, and Dublin Business School for awarding myself the Adele Farrell Memorial Scholarship in 2015. I am deeply indebted to you for this opportunity to attend college and graduate with this degree.

My parents and my grandparents, thank you for teaching me the importance of education, hard-work and perseverance ever since I was young. The values and love that you have taught me is what I carry with me always.

My family, my motivation. Thank you for putting up with me in difficult moments where I felt exhausted and for persuading me on to follow my dream of getting this degree. This would not have been possible without your unwavering and unselfish love and support.

God, thank You for giving me strength, Your Divine Providence, Grace, and Love.
Your blessings are what made me who I am today.

Psalm 46:5
God is within her, she will not fall;
God will help her at break of day.

Table of Contents

Chapter 1: Introduction

While there are a lot of useful budgeting applications out there such as; Wallet, Monefy, Spendee, and Money Lover, and banks coming out with their mobile applications such as; AIB and Bank of Ireland, along with the rise of mobile banks like Revolut and N26; banking and tracking finances is easier than ever. However, in a world where new products are coming out nearly every day, and advertisements are almost everywhere; the temptation to spend on things that one does not need - is high, it is hard to resist spending and it is very easy to overspend. A budget plan is very important, may it be a physical list, a document or a spreadsheet. It aids the proper allocation of one's finances, helping them to not live beyond their means, save for a rainy day, and avoid debt. Also, according to Constante et al., (2017), "Nowadays to make the right decision about where and how to request or buy a service, a user is often supported by a mobile device that offers more than simple descriptive data."

This is where the inspiration for this project comes from. The developer of the FIN application is a working student, and while doing her best in her studies sometimes struggles to make ends meet. Often asking herself, "should I buy this?" or "can I afford this?". After searching high and low on Google and the Google Play Store for an application that can make recommendations and to no avail found nothing. The developer then thought of building a mobile application to solve this problem.

This project will focus on the development of a money tracker mobile application with a built-in mobile decision support system. The application aims to be a financial planner and budgeting tool. It will help end-users track where their money is coming from and where they are spending their hard-earned money. Another key feature of this application is the suggestion tool. The application will give a recommendation whether the user should make a purchase based on their details and form input. This application will be designed to be used as a tool for budgeting, and not to handle genuine money.

Chapter 2: Background

Overview

FIN Application is designed to help users document and analyse their finances. Users can store a detailed entry of transactions that they have made. Transaction entries can either be an income or expense. They can be sorted out in different categories as well. The data is stored locally in the device using a SQLite database, therefore there is no need for authentication. The application is coded in C#, and the user interface in XAML using Xamarin.Forms.
One of the most interesting feature of this application is its ability to provide users suggestions about spending. The application can process user input from a questionnaire about the product/service then analyse the users' data and will able to make recommendation whether a he/she should/should not spend on a certain item/service. The dataset for this suggestion model was collected through a Google Forms survey, processed with RapidMiner and made in R. The model is exported into a PMML file and integrated into the application using Syncfusion PMML Execution Engine.

The application uses the Xamarin toolkit, thus it is compatible with both Android and iOS platform. However, this project focuses on the Android application development.

Deliverables

Materials to be produced during the course of this project:

1. Project Proposal
2. Interim Report
3. Files that compose the mobile application
    a. XAML Files – front end code
    b. C# code – back end code
    c. PMML file – contains the model that provides the recommendation tool
4. Dataset collected with Google Forms – before and after data cleaning
5. R Code – used to build the prediction model
6. Final Report on the project
7. Presentation

Advantages

Efficiency

When deployed on the Google Play Store or Apple App Store, the app will be readily available to users. FIN is a mobile application and is local to the device, it is easily accessible for users. The application will help users save their time from physically listing and calculating their expenses and incomes. FIN does not depend on any APIs or any external applications from the internet, therefore connectivity to the internet is not required when using the app.

Organisation

Financial records can be easily stored and organised within this app. The cashflow can be digitally presented with the use of the transaction history. The transactions list can be filtered into different categories so users can quickly sort and identify their entries.

Proper money management

By having an interface that displays the incomes and expenses, and can be separated into categories; finances can be easily monitored. The app will guide the user to identify the categories where they are spending their money. This can expose poor spending habits and promote proper money management.

Target Users

People with bad spending habits

Unnecessary spending can lead to debt. By having the application, the user can view their transactions and identify the amount of money that they spend on the things they want versus the things they need. They can then make sure that they have enough money for their necessities and do not spend money that they do not have.

Savers

It will be beneficial for people who are trying to save for a particular thing or event i.e. a brand new car, holidays, concert tickets etc. to have their cashflow displayed in front of them. They can monitor where their money is going and which categories of expenses they can cut back on. The application will serve as a motivation to reach the users' savings goals.

Frugal People

The application would be a great asset to users who are frugal and wise with their money. It will be a great convenient and portable platform to document their income and expenses. They can be relieved of avoidable stress of physical budget lists and will have a better understanding of where they are in their financial goals, allowing them to improve their financial plans. The suggestion feature of the application will also help them to make smart spending decisions.

Objectives

During the course of this project; the author aims to acquire knowledge of the following:

Xamarin.Forms: The biggest complexity of this project comes from learning how to build a mobile application. At the beginning, the author knew little about mobile application development, but due to the project, the author should have better knowledge in this field.

Financial Management: As this project is aimed for managing finances, the author should have taken keen interest in this field to have a better understanding of the required features of the application.

Dataset: With every machine learning algorithm, a dataset is required in-order to train and build the prediction model. The author self-obtained the dataset using Google Forms. The dataset is tailored for the recommendation system.

Machine Learning: Since this project requires a recommendation system to be produced. The author must learn machine learning algorithms. The author must also prepare and clean the data for modelling.

SQLite: The system should have a way to store entries the users wishes to keep, whether its locally or online – or even both. The author have prior knowledge with SQL, but not with SQLite. Further research is needed to integrate SQLite in the mobile application.

R: The prediction model in the app is to be made in R. Since the author does not have a lot of prior experience working with this language, the author should learn how to use the language to achieve objectives and overcome challenges while doing so.

C# and .NET: Since the author does not have prior knowledge of building mobile applications but have prior knowledge of C# and .NET, the author aims to improve and be able to use the language to build real-world impactful projects.

Software and Hardware Resources

Software

- Microsoft Word – documentation of the project.
- Google Forms – collection of data through a survey.
- Microsoft Excel – data cleaning and preparation.
- QEMU Emulator – testing on different android devices and versions.
- XCode iOS simulator – testing application performance on iOS.
- Draw.io – creating diagrams for this report.
- RapidMiner Studio – dataset feature selection.
- R Studio – building machine learning models.
- Microsoft Visual Studio Community Edition will be used as the IDE of choice.

Hardware

- The project will be made using a MacBook Pro 15" 2018 Edition.
- The application was mainly tested using a physical android phone – OnePlus 6t, running OxygenOS version 9.0.11.

Chapter 3: Literature Review

Financial Literacy

A study in 1987 by Danes and Hira (p. 11), shows that financial record keeping in college students are positively correlated with financial management. In 2018, a study in Australia shows that adults are ill-equipped to understand basic financial management concepts and that financial management and literacy skills should be taught at an age before credit cards, personal loans and home loans become a possibility (Guest and Brimble, 2018, p. 4).

Financial literacy has been defined in numerous ways by previous studies.
The Australian Securities and Investments Commission (ASIC) defines it as 'a combination of financial knowledge, skills, attitudes and behaviours necessary to make sound financial decisions, based on personal circumstances, to improve financial wellbeing' (cited in Guest and Brimble, 2018, p. 4). While, Gale and Levine (2010, p. 2) define financial literacy as a person's capability to performs knowledgeable and practical judgements about using, organising, and managing wealth and money.

Selvakumar et al. (2018), implies that financial literacy affects the potential growth rate of the economy. Financial literacy plays a significant part in influencing financial institutions as it involves peoples' spending and investment decisions. Therefore, it affects how economic resources are allocated.

Additionally, in Financial Literacy 101 written by Guest and Brimble (2018), they mention that financially literate people can better equip themselves with knowledge and skills so they can make informed financial decisions, be less vulnerable with scams, and avoid accumulating unmanageable amounts of debt.

A study by the Aspen Institute (2018) reports that 16% of suicides in the US occurs in response to a financial problem. According to Selvakumar et al. (2018), financial literacy is connected to the well-being of a person. Financial literacy is crucial for preventing and dealing with financial problems. Thus, it is "vital to living a prosperous, healthy and happy life".

Mobile Decision Support Systems

Technology users are no longer constrained to a desktop computer, mobile devices now offers a much more flexible access to technology (Perez et al., 2008). People are heavily influenced by technology. In 2019, there are 2.71 billion smartphone users (statista.com). Smartphones are convenient devices to have because of its' portability, as well as it offers many features i.e. telecommunications, internet access, social networking, geolocation, etc. Therefore, it can be considered as the perfect tool for performing day-to-day personal tasks. A study by Dr. Joel Hoomans, shows that people make 35,000 decisions a day and these decisions occur in time-sensitive environments and can occur anywhere (Power, 2013 cited in Jaradat et al., 2018, p. 115). Mobile phones are the ideal tool for these as they can adapt to these time and location requirements. Decision making processes can be improved using mobile phones, since they can be carried out faster and it can be done anytime and anywhere (Perez et al., 2008).

The growth of wireless mobile services has brought a new way of using computers. It has increased the accessibility to data and therefore, influenced decision making processes for users on the move. The capabilities of decision support systems are greater than ever. Customers can now make real-time decisions that is based on real-time data (Cowie and Burstein, 2007 cited in Perez et al., 2008 p. 241). Mobile personal digital assistants are also rising in demand as the power of mobile decision making increases (Perez et al., 2008 p. 241).

While decision support systems (DSS) lack a single accepted definition, Finlay (1994, cited in Constante et al., 2017 p. 787) defines a DSS as broadly as a computer-based system that aids the process of decision making". According to Guo et al. (2013), "mobile decision support system (MDSS) are decision support systems built on mobile computing devices, which provide people with real time decision-making support capabilities". MDSS provides great portability and flexibility than DSS. With the great power of MDSS, the healthcare industry had already accepted that mobile technologies can appropriately support decision making processes in situations where desktop devices are not favourable (Burstein et al. 2008, cited in Jaradat et al., 2018, p. 115). mDSS are also implemented in many other industries i.e. mobile banking systems and mobile rescue support systems (Guo et al. p. 2).

Existing Mobile Applications for Financial Management

Application Selection

The following search terms were used in the Google Play Store to source mobile applications for review : “financial tracker”, “financial planner”, “money manager”, “budget planner”, “budget tracker”, “spending tracker”, “expense manager”, “expense tracker”.

Criteria:

Free download

Money tracking and documenting is the main focus.

Have an interface for money tracking feature.

More than 100,000 plus downloads.

Have 4 star ratings or more.

Must be in English.

Last updated no earlier than 2018.

Must be an Editors’ Choice.

Selected Applications:

Spendee - Budget and Expense Tracker & Planner - https://play.google.com/store/apps/details?id=com.cleevio.spendee

Wallet - Finance Tracker and Budget Planner - https://play.google.com/store/apps/details?id=com.droid4you.application.wallet

Monefy - Money Manager -

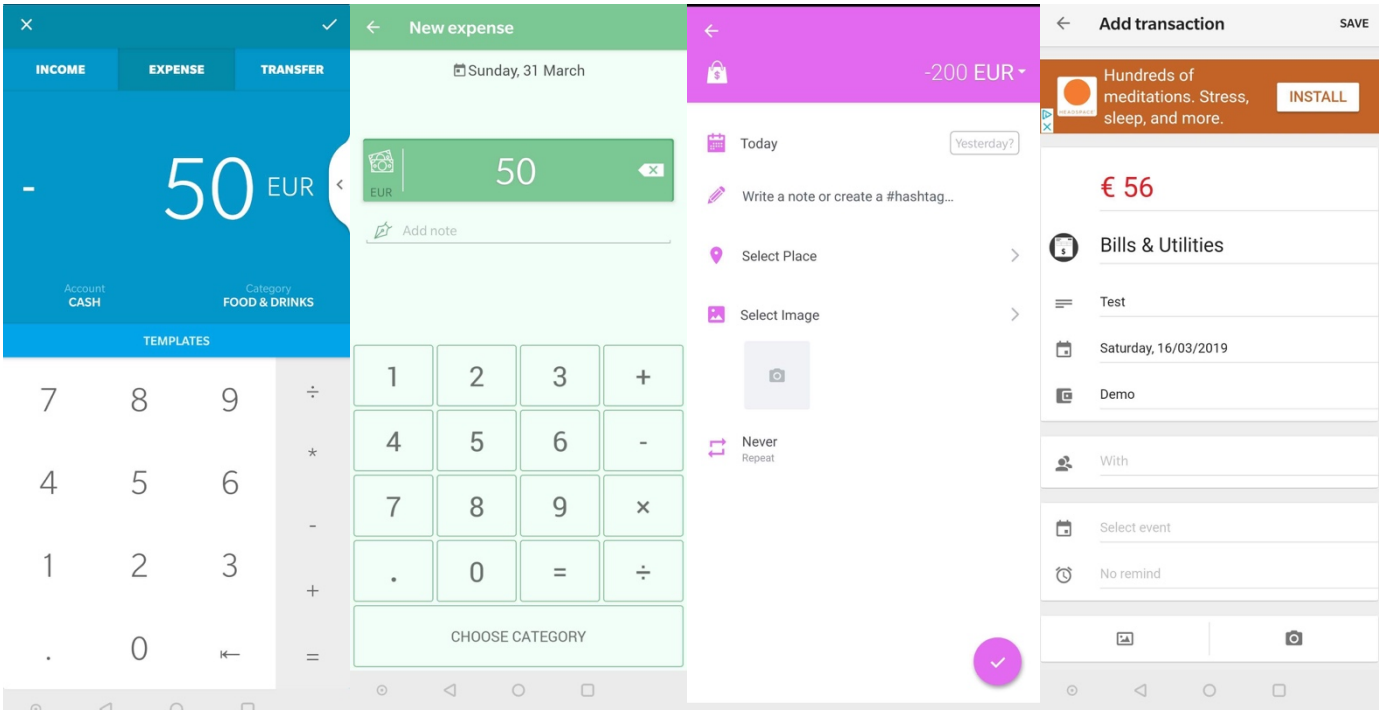
https://play.google.com/store/apps/details?id=com.monefy.app.lite

Money Lover: Expense Tracker & Budget Planner- https://play.google.com/store/apps/details?id=com.bookmark.money

Feature Analysis

The purpose of this study is to analyse finance tracking/record keeping functionality of existing mobile applications. Extra customisation features are out of scope for this study. Paid features will not be included.

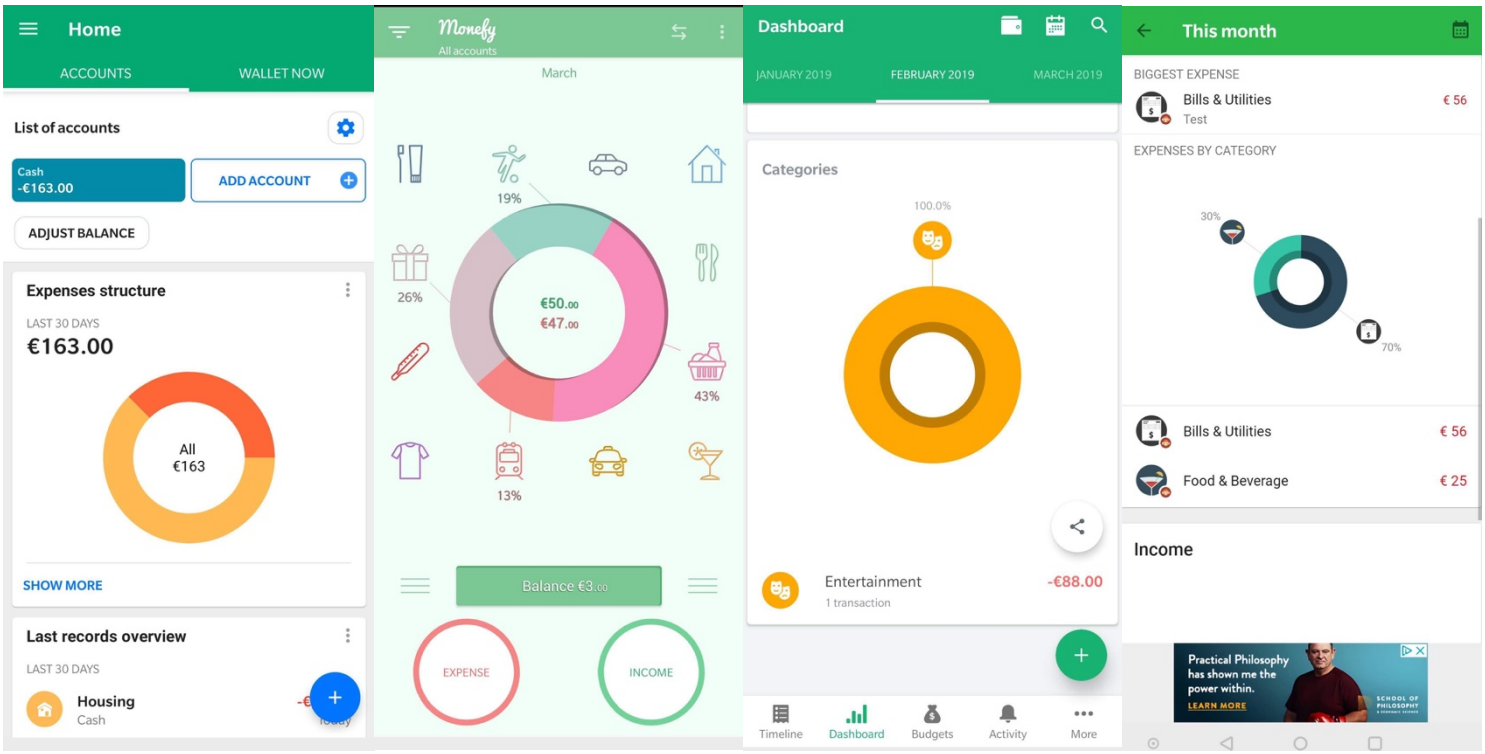
Adding an entry



Left to Right : Wallet, Monefy, Spendee, Money Lover.

All of the applications have a similar user interface for adding transaction entries. Similar to the FIN application, the above apps has the feature to categorise each expense/income transaction, add a description, and amount. A great additional feature of these applications that is not included in the FIN Application is the ability to change dates (Wallet sidebar not shown).

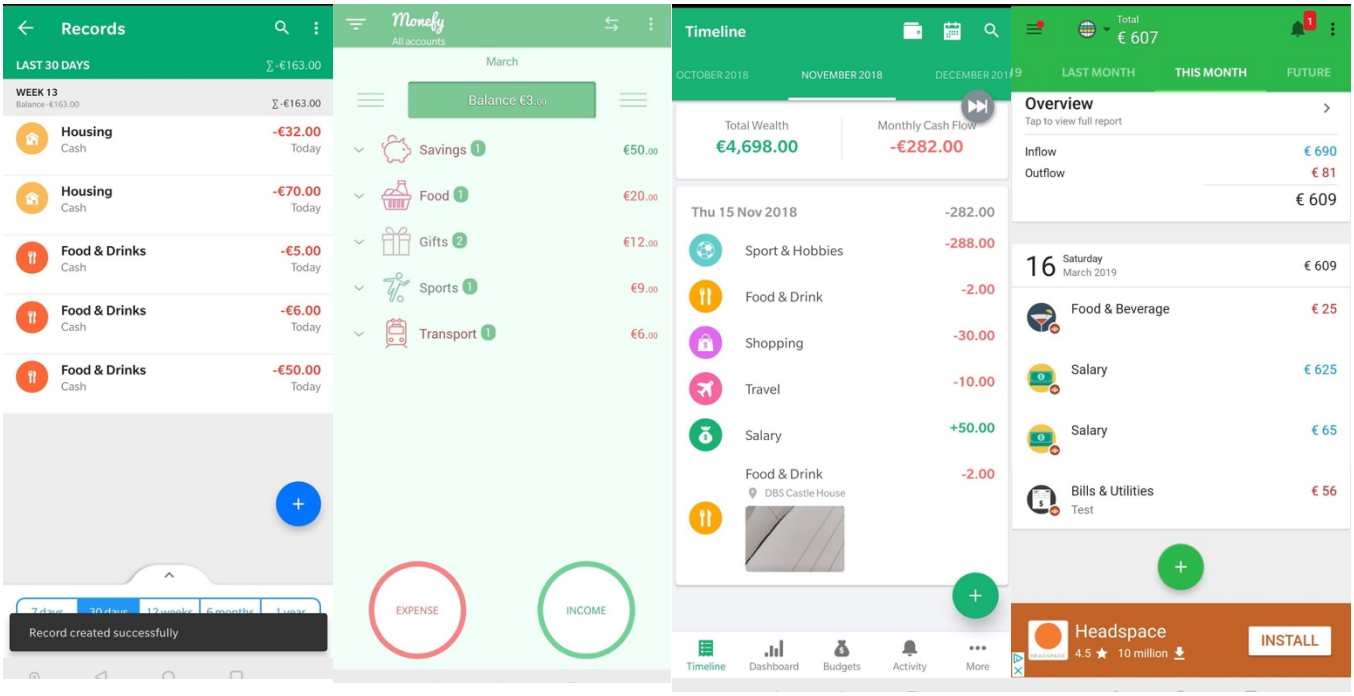
Visualisations



Left to Right : Wallet, Monefy, Spendee, Money Lover.

Here we see that all applications uses a pie-chart to visualise where the users' finances are going or coming from. We can also see that the applications has the capacity to visualise data on a monthly basis. This can be a very helpful tool as it is an quick overview of the users' financial activities. However, the pie-chart may not be an effective way to display data when categories increases as it can be difficult to distinguish between each of them.

Transaction History



Left to Right : Wallet, Monefy, Spendee, Money Lover.

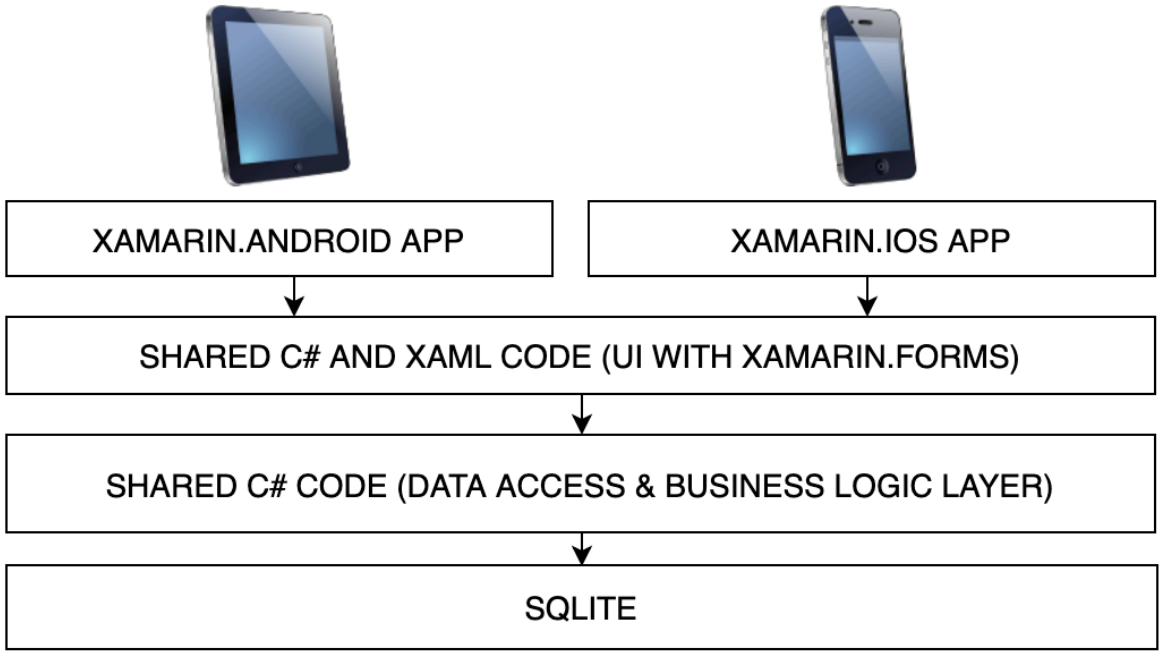
Each application also has a monthly transaction history screen. Each entry has a corresponding category icon when displayed on the list. Out of all the four applications, Monefy has a different system of listing transactions. It takes each category and adds up the amount related to that category, then it is displayed on the list. Each list entry can then be expanded to see the breakdown. This is a great way to display items on the list as it prevents multiple entries of the same name and avoids confusion.

Chapter 4: FIN Mobile Application

Building The Mobile Application

The first step taken was deciding how to make the application itself. From the project proposal and interim report; the application was set to be to be made available to android users only. Native android applications are known to be best created using Android Studio and Java. However, the author does not have any prior knowledge on both of these technologies. The author is more proficient in C# than any other programming languages. After researching, the author decided to learn how to build a mobile applications that can use C# as the back-end code. This section will talk about the mobile application built and its composing technologies.

Overview of the system



The mobile application is built using the Xamarin toolkit, which allows it to run on both Android and iOS. The back-end code is in C# and the front-end code is written using XAML using the Xamarin.Forms toolkit. The MVVM Architecture is used to make the code loosely coupled and more manageable. Data is stored on the users' local device using SQLite.

Mobile Development Approach

Native vs. Cross-Platform

Native mobile applications are specifically made to work on one single platform. It uses a specific programming language and IDE (Integrated Development Environment) that is created by the platform vendor. These applications are available through the platforms own application store i.e. Google Play Store for Android, App Store for iOS.
An advantage of the native approach is that it provides fast application performance, as well as it will gain access to the device's native features such as the camera, GPS, contact lists, etc. However, developing native applications can be very slow and expensive. Multiple teams will have to be hired and each team of developers will have to create various applications, in its' native language and architecture, in order to support all the different platforms.

Cross-platform mobile application development is another approach that has been getting more popular. This approach uses a common code that can be deployed on various different platforms. This approach offers a more affordable way to develop applications as it only requires one team of developers to create the shared code base. Development is made faster as it avoids unnecessary code repetition and code maintenance is made easier. These cross-platform applications can then be deployed on each different platforms' application stores, making it available on a wide-range of devices.

Mobile application development can be approached in many different ways. For this project, the application will be developed using the cross platform development approach as it is more efficient to build and cost effective. This is because the native approach is time consuming and would require multiple developer teams and multiple code bases to reach many end users using different platforms.

By using the cross platform development approach, the project application will be made available to vast amounts of users, while saving time and effort on development and maintenance. Using Xamarin, the product outcome is an application that has the ability to run natively.

Xamarin and Xamarin.Forms

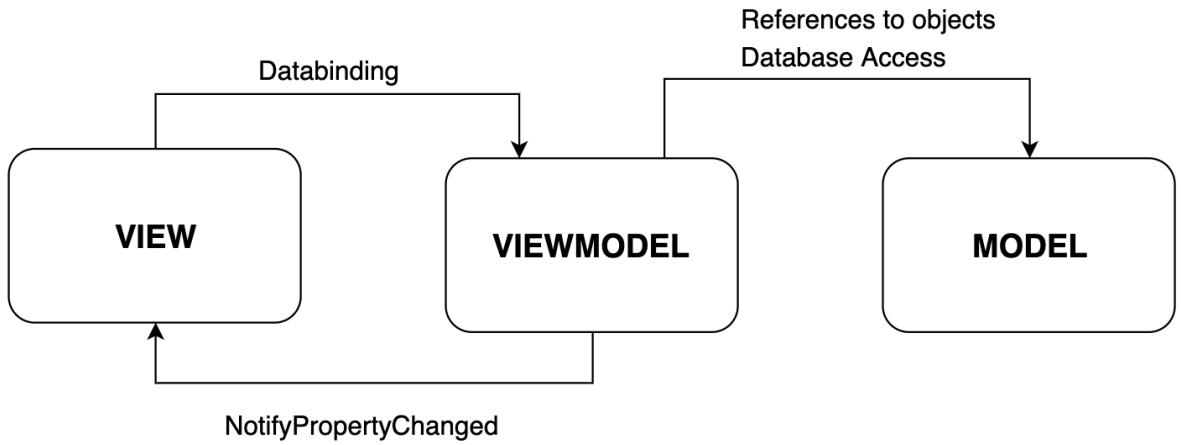
“Xamarin is unique in this space by offering a single language – C#, class library, and runtime that works across all three mobile platforms of iOS, Android, and Windows Phone (Windows Phone’s native language is already C#), while still compiling native (non-interpreted) applications that are performant enough even for demanding games.” (Microsoft, 2017). Xamarin allows the development of cross platform applications by using one C# code base that can compile natively and target multiple mobile platforms. According to Hermes (2015), “Xamarin bridges the gap between mobile operating systems and their respective development languages i.e. iOS and Objective-C, Android and Java, and Windows and C#”.

“Xamarin.Forms exposes a complete cross-platform UI toolkit for .NET developers. Build fully native Android, iOS, and Universal Windows Platform apps using C# in Visual Studio” (Microsoft, 2018). Xamarin.Forms allows developers to define the UI once for all mobile platforms, therefore developers have the ability to develop native mobile applications for different platforms at the same time.

In the FIN mobile application, a shared C# code is created. This code contains the shared business logic and data access layer, which is used in Xamarin. iOS project and Xamarin.Android project. In this way, user interface and the C# code is created once in the common project and remains the same in respect to each target platform. The code is fully shared between the Xamarin.Android and Xamarin.iOS project. When compiled, the mobile application runs natively in its’ mobile platform.

Disclaimer: Even if the cross-platform development approach is used in this project; the focus is on the development of the Android application. The Xamarin.iOS project has been included in the code solution for testing cross-platform functionality. However, this is not an area that is concentrated on during the development of the project. Therefore, the iOS application may not be working and rendering properly when tested.

MVVM Architecture



MVVM (Model-View-ViewModel) is an architectural software pattern which encourage the separation of the user interface and the business logic of the application.

Model - represents the data and the business logic of the application.
In this project, the object classes, as well as the database helper class are considered as Models.

View – represents how information is displayed to the users.
In the Xamarin project, these are the pages that contains the UI XAML code.

ViewModel – a Model that is specifically made for a View. It is a representation of what is to be displayed on the UI. It is a class which contains properties that represent data in the View, and methods that implement the logic behind the View. When the View has an event, the ViewModel responds, react and pass data back to the View. The ViewModel interacts with the Models to access the business logic and the database of the application.

The Xamarin code behind of Pages are tightly coupled with the user interface, this makes it difficult to perform unit tests. By applying MVVM, the code behind is modified and re-written on to the ViewModel class. The View can then be binded to the properties in the ViewModel.

The ViewModels and Models are portable and therefore, the user interface can simply be changed, and the code can be implemented in other applications.

DataBinding

Target UI properties in the View can bind to a data source that is in the ViewModel. Binding is automatic and it can be one-way or two-way. One-way : source to target only. Two-way: source to target and target to source. The target property must be a bindable property and the data source can be a class property or a variable.
Whenever a change in the source is made, a NotifyPropertyChanged event (which derives from the INotifyPropertyChanged interface) is implemented so the source is observable and to allow the target to be updated. UI controls, lists, texts, buttons, etc can be binded to a property or data source for example, a list or observable collection.

CommandBinding

This works almost the same as databinding. However, this is used to bind a target to a function, task, or event. Instead of creating a click handler or an event handler, elements in the View can bind to the ViewModel, which contains the command property. This property can implement NotifyPropertyChanged event to update the UI or perform tasks.

The MVVM architecture does not come with Xamarin.Form by default. Toolkits and packages can be downloaded e.g. MVVMLight. However, in this project, we manually implement the architecture.

SQLite

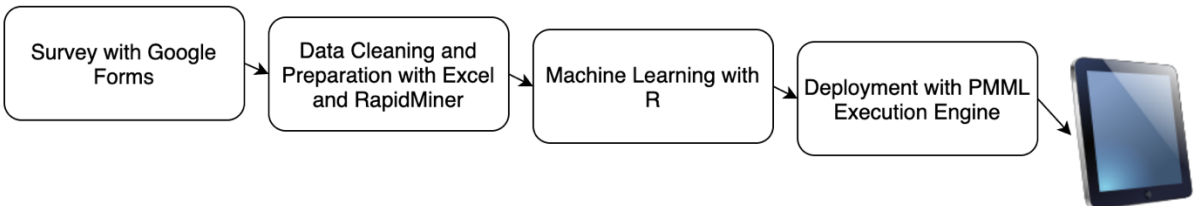
“SQLite is a C-language library that implements a small, fast, self-contained, high-reliability, full-featured, SQL database engine. SQLite is the most used database engine in the world. SQLite is built into all mobile phones and most computers and comes bundled inside countless other applications that people use every day” and “ SQLite is an in-process library that implements a self-contained, serverless, zero-configuration, transactional SQL database engine”. (SQLite, 2019)

FIN application uses SQLite for data storage since it offers a way to implement local databases into a device. MS SQL, Oracle and PostgreSQL are server-based databases and may slow down the performance of the application. SQLite is chosen because it is lightweight as it is a standalone file-based database management system which can provide small databases for small devices, we.

Xamarin.Forms can incorporate SQLite database functionalities into applications by installing the SQLite.NET PCL NuGet package created by Frank A Krueger. https://www.nuget.org/packages/sqlite-net-pcl/

Machine Learning

To create a prediction model for the application; a dataset has to be obtained and prepared. Then, the prediction model will have to be created using the data, and tested. In this section, the collection of dataset through Google forms will be discussed, as well as how it is prepared with Excel and RapidMiner. The creation and testing of the prediction model with R and deployment will be outlined.



Dataset

Initially, there was a search for available related datasets on the Internet for the project. However, after much consideration, it was decided to create a dataset tailored for the use of this project. A survey was made using Google Forms.

The survey contained a target variable which is; if the respondents has gone ahead with the purchase or not. The other survey questions were used to determine the factors that cause the respondents to make purchases.

A total of 109 responses were collected from the survey.

Data Preparation

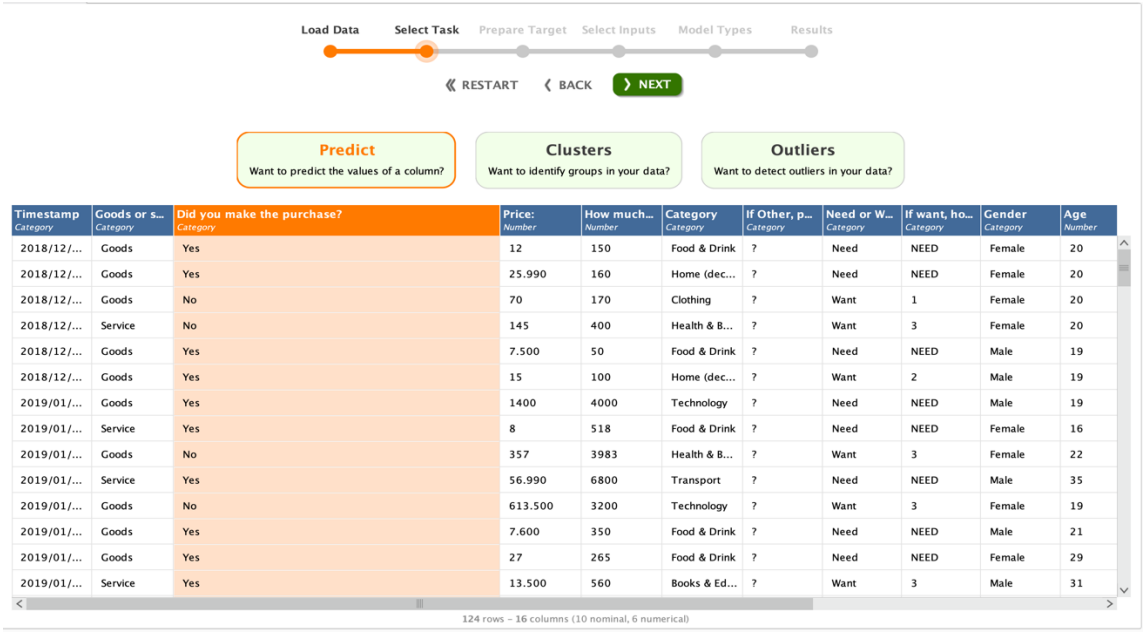
To prepare the dataset for machine learning, the following steps were taken:

1. Feature Selection
2. Data cleaning

Feature Selection

Once the dataset has been exported into a CSV file, it was uploaded to RapidMiner. The following steps were taken:

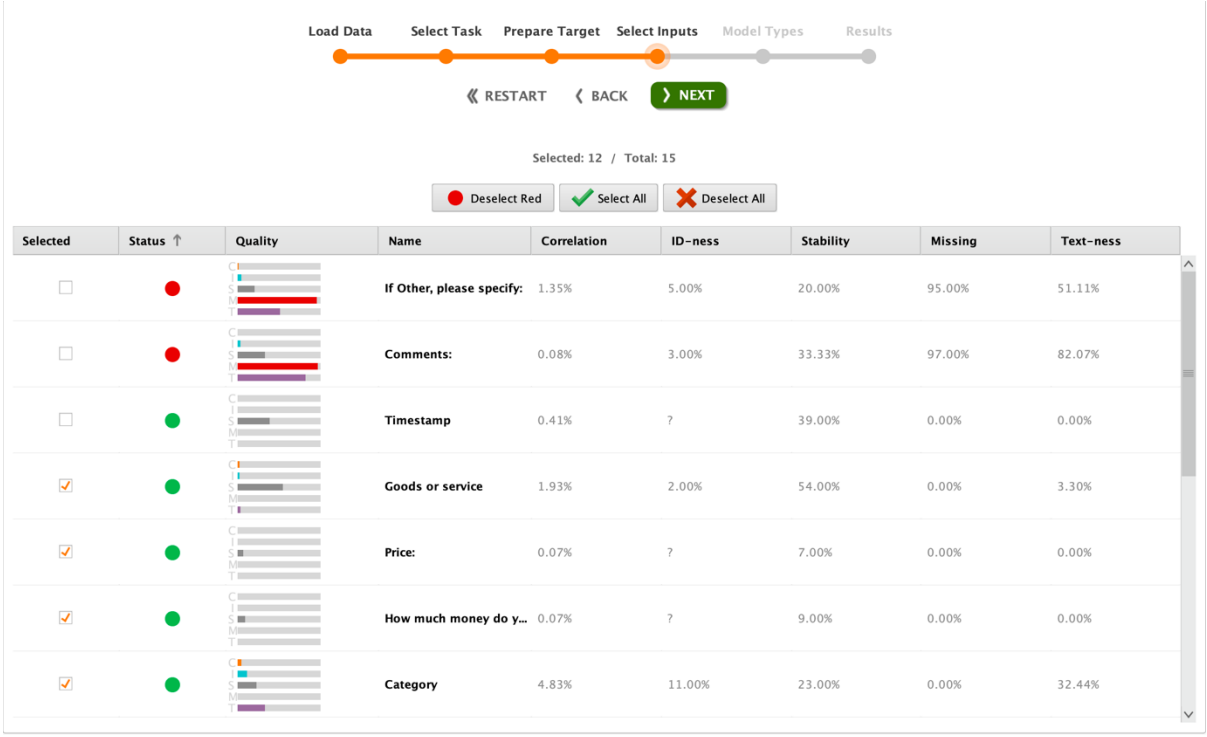
1. Choose target variable for prediction.



Timestamp *Category*	Goods or s... *Category*	Did you make the purchase? *Category*	Price: *Number*	How much... *Number*	Category *Category*	If Other, p... *Category*	Need or W... *Category*	If want, ho... *Category*	Gender *Category*	Age *Number*
2018/12/...	Goods	Yes	12	150	Food & Drink	?	Need	NEED	Female	20
2018/12/...	Goods	Yes	25.990	160	Home (dec...	?	Need	NEED	Female	20
2018/12/...	Goods	No	70	170	Clothing	?	Want	1	Female	20
2018/12/...	Service	No	145	400	Health & B...	?	Want	3	Female	20
2018/12/...	Goods	Yes	7.500	50	Food & Drink	?	Need	NEED	Male	19
2018/12/...	Goods	Yes	15	100	Home (dec...	?	Want	2	Male	19
2019/01/...	Goods	Yes	1400	4000	Technology	?	Need	NEED	Male	19
2019/01/...	Service	Yes	8	518	Food & Drink	?	Need	NEED	Female	16
2019/01/...	Goods	No	357	3983	Health & B...	?	Want	3	Female	22
2019/01/...	Service	Yes	56.990	6800	Transport	?	Need	NEED	Male	35
2019/01/...	Goods	No	613.500	3200	Technology	?	Want	3	Female	19
2019/01/...	Goods	Yes	7.600	350	Food & Drink	?	Need	NEED	Male	21
2019/01/...	Goods	Yes	27	265	Food & Drink	?	Need	NEED	Female	29
2019/01/...	Service	Yes	13.500	560	Books & Ed...	?	Want	3	Male	31

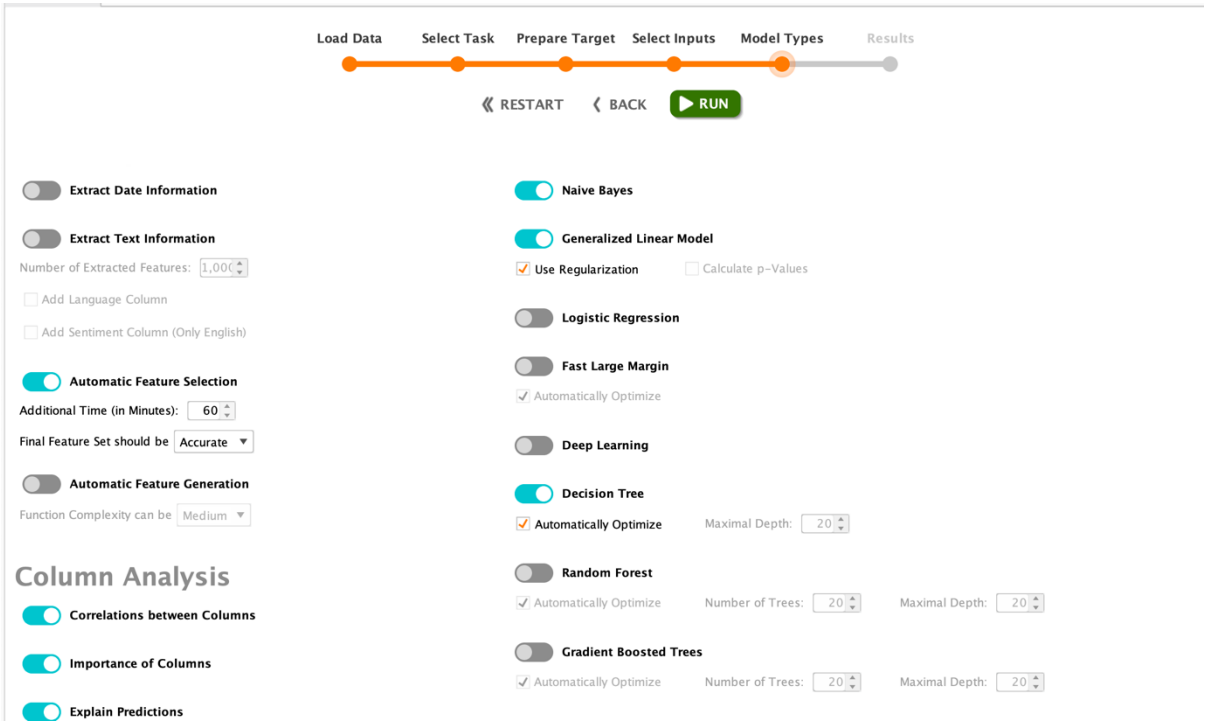
124 rows – 16 columns (10 nominal, 6 numerical)

2. RapidMiner shows a list of features who are correlated to the target variable. Disregard features that are not correlated (indicated with red). Timestamp is also deselected as it is deemed unnecessary because it is the timestamp when the survey was performed, not when the purchase was made.



Selected	Status	Quality	Name	Correlation	ID-ness	Stability	Missing	Text-ness
☐	●		If Other, please specify:	1.35%	5.00%	20.00%	95.00%	51.11%
☐	●		Comments:	0.08%	3.00%	33.33%	97.00%	82.07%
☐	●		Timestamp	0.41%	?	39.00%	0.00%	0.00%
☑	●		Goods or service	1.93%	2.00%	54.00%	0.00%	3.30%
☑	●		Price:	0.07%	?	7.00%	0.00%	0.00%
☑	●		How much money do y...	0.07%	?	9.00%	0.00%	0.00%
☑	●		Category	4.83%	11.00%	23.00%	0.00%	32.44%

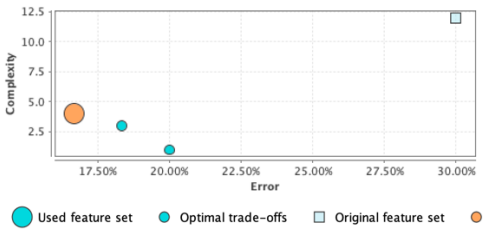
3. Choose algorithms. Here Naïve Bayes, Generalized Linear Models, and Decision Trees are chosen.



4. Take note of the results

Naive Bayes – Feature Sets

Optimal Trade-offs between Complexity and Error



Used feature set Optimal trade-offs Original feature set Shown below

Description

this and is shown in the performance tab and the model overview.

The **best** feature sets are in the bottom left corner with low complexity and low error rates. Please note that it usually is not possible to minimize the complexity without increasing the error rate at the same. However, in many cases the resulting feature spaces (round dots) are less complex and still more accurate than the original feature space (square). Using less features also means that models can be trained faster. The feature set which has been used to build the final model is shown bigger.

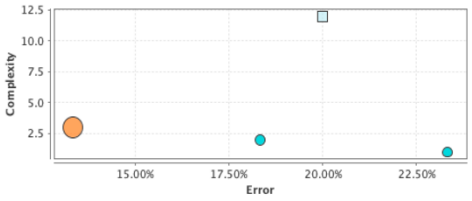
You can **click on each dot** to see the details about the feature space in the table below. This will deliver additional insights into the relative importance of features for the model and how the features interact.

Currently Selected Feature Set

Name	Expression	Complexity
If want, how much do you WANT the item? (1 being ...	[If want, how much do you WANT the item? (1 being...	1
Monthly Budget for your NEEDS	[Monthly Budget for your NEEDS]	1
Monthly Income	[Monthly Income]	1
Need or Want?	[Need or Want?]	1

Generalized Linear Model – Feature Sets

Optimal Trade-offs between Complexity and Error



Used feature set Optimal trade-offs Original feature set Shown below

Description

this and is shown in the performance tab and the model overview.

The **best** feature sets are in the bottom left corner with low complexity and low error rates. Please note that it usually is not possible to minimize the complexity without increasing the error rate at the same. However, in many cases the resulting feature spaces (round dots) are less complex and still more accurate than the original feature space (square). Using less features also means that models can be trained faster. The feature set which has been used to build the final model is shown bigger.

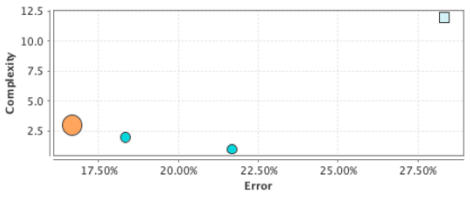
You can **click on each dot** to see the details about the feature space in the table below. This will deliver additional insights into the relative importance of features for the model and how the features interact.

Currently Selected Feature Set

Name	Expression	Complexity
Need or Want?	[Need or Want?]	1
Category	[Category]	1
Monthly Income	[Monthly Income]	1

Decision Tree – Feature Sets

Optimal Trade-offs between Complexity and Error



Used feature set Optimal trade-offs Original feature set Shown below

Description

this and is shown in the performance tab and the model overview.

The **best** feature sets are in the bottom left corner with low complexity and low error rates. Please note that it usually is not possible to minimize the complexity without increasing the error rate at the same. However, in many cases the resulting feature spaces (round dots) are less complex and still more accurate than the original feature space (square). Using less features also means that models can be trained faster. The feature set which has been used to build the final model is shown bigger.

You can **click on each dot** to see the details about the feature space in the table below. This will deliver additional insights into the relative importance of features for the model and how the features interact.

Currently Selected Feature Set

Name	Expression	Complexity
Price:	[Price:]	1
Monthly Income	[Monthly Income]	1
Age	[Age]	1

Data Cleaning

The following steps have been taken to ensure that the dataset is ready for processing:

1. **Determine necessary features.**

   The features that will be used are the combined list of feature sets from RapidMiner:

   - Did you make the purchase? (target variable)
   - If want, how much do you want the item?
   - Monthly budget for NEEDS
   - Monthly Income
   - Need or Want?
   - Category
   - Price
   - Age
   - How much money do you have in total? (even if this feature has not been produced by RapidMiner feature selection, it is included for the purpose of showing that user details in the application can be used inside for prediction)

2. **Delete features.**

   Apart from the list of selected feature sets from feature selection in RapidMiner. The remaining features were removed.. The user identity has not been implemented in the application, therefore it was decided that the Age feature is also removed for the purpose of this project.

3. **Change feature names.**

   - Did you make the purchase? – target
   - If want, how much do you want the item? – desire
   - Monthly budget for NEEDS – monthlyneeds
   - Monthly Income – monthlyincome
   - Need or Want? – needorwant
   - Category – category
   - Price – price
   - How much money do you have in total? – balance

4. **Delete irregular values.**

   The dataset obtained for the project yielded 109 results. This is a very small dataset, with lots of features, and each feature have various responses. Therefore, irregular values are most likely present . This is most especially in the category responses, e.g. very few chose home equipment, and sometimes, category is not specified.
   When the model is built and the accuracy is tested with the testing set, the model sometimes do not recognise the new categories. Therefore, these values are eliminated from the dataset.

5. **Binary target variable.**

   Revalue the target variable where, Yes = 1 and No = 0. This is done through R.

Predictive Modelling with R

Training and Testing Sets

When building the model, the dataset is randomly split into training and testing sets. The training set is used to build the model i.e. it is the data that will be learned. The testing set is the subset of the dataset that is used to determine whether the model predicts outcomes correctly.

To determine the best train and test set ratio for the dataset, ratios; 70%, 80%, and 90% are tested ten times, the accuracy from each run, for the three algorithms is stored. The average accuracy for each algorithm is then calculated. The ratio with the highest average accuracy is chosen. In this case, the algorithms are most accurate when the training set ratio is 90% of the dataset.

```
> glm_df
  ratio  accuracy
1   0.7 0.6933333
2   0.8 0.6900000
3   0.9 0.7000000
> rpart_df
  ratio  accuracy
1   0.7 0.6466667
2   0.8 0.6900000
3   0.9 0.7500000
> nb_df
  ratio  accuracy
1   0.7 0.6366667
2   0.8 0.6350000
3   0.9 0.6400000
>
```

GLM

```
> mean(glm_accu_vect)
[1] 0.71
> mean(rpart_accu_vect)
[1] 0.68
> mean(nb_accu_vect)
[1] 0.62
>
```

Three algorithms are evaluated for building the model for the application. These algorithms are: Naïve Bayes, Regression Trees, and Generalised Linear Models. The self-obtained dataset is tested ten times using each algorithm in order to find the best model for the application. For each test, the accuracy is stored on a vector. The average accuracy is then calculated and the algorithm with the highest average accuracy is chosen. Here, it is seen that Generalised Linear Model (GLM) is the most accurate.

"Generalised Linear Models (GLM) estimate regression models for outcomes following exponential distributions. In addition to the Gaussian (i.e. normal) distribution, these include Poisson, binomial, and gamma distributions." (H2o.ai, 2019)

"A binomial logistic regression (often referred to simply as logistic regression), predicts the probability that an observation falls into one of two categories of a dichotomous dependent variable based on one or more independent variables that can be either continuous or categorical." (statistics.laerd.com, 2019)

Refer to appendices : GLM Model Summary Output

The variable 'target' is a binary variable where 1 is when it is indicated that the purchase is made (Yes), and 0 for when the purchase is not made (No).

The glm() function in R is used to run a generalised linear model.

Important information is inside the parentheses. The variable 'target' is the dependent variable. '~.' implies that the rest of the variables are listed as predictor variables. Then, it is specified that the distribution/family is binomial since we are performing binomial logistic regression. Finally, it is indicated that the dataset to be used is the trainset (a subset of the whole dataset).

- Deviance Residuals : the deviance for each data point in the model is calculated. A set of residuals is produced after calculating this for every single point. The output is a description of the distribution of the residuals.
- Coefficients : the output also contains the standard errors, the z-statistic and the associated p-values. The coefficients give the change in the log odds of the outcome for a one unit increase in the predictor variable. The estimates are in logits.
- Intercept is the dependent variable.
- Independent variables : estimate is how much the dependent variable can change each time an independent variable increases. If the estimate is large, then the independent variable has a large effect on the dependent variable. Below is a brief description of the independent variables output:
    - For every one unit change in monthlyincome, the log odds of purchase being made increases by 1.347e-03.
    - For every one unit increase in balance, the log odds of purchase being made increases by -9.001e-05.
    - When the desire is rated 2 as opposed to another rating, the log odds of purchase being made increases by 6.529e+00.
    - When the category is Health and Beauty as opposed to another category, the log odds of purchase being made increases by -4.669e+00.
    - When needorwant is Want as opposed to Need, the log odds of purchase being made increases by -2.982e+00.
- Signif.codes : significance of each independent variable to the dependent variable
- Null deviance : Small values are desired. This indicate the predictive ability of model only by using the intercept.
- Residual deviance : Small values are desired. This insidcate the predictive ability of the model by using the intercept and other independent variables.
- Where there is a big difference between the Null deviance and Residual deviance, it signifies that the independent variables are significant when predicting the dependent variable.
- AIC : stands for Akaike Information Criterion. It shows an estimate of the model performance as it describes patterns in the data. Small values are also desired.
- Number of Fisher Scoring iterations : how long it took to fit the model.

Actual → **Predicted ↓**	**Negative Class (No)**	**Positive Class (Yes)**
Negative Class (No)	True Negative (TN)	False Negative (FN)
Positive Class (Yes)	False Positive (FP)	True Positive (TP)

First, the model predicts a list of values, then to determine the accuracy of the model; a confusion matrix is used by comparing the correct predictions of the model versus its actual value.

```
> glm_confusion_matrix
                     actual
glm_predictedvalues  0  1
                  0  3  3
                  1  3 18
```

The **accuracy** is the model's overall correctness of its predictions, i.e. it is the probability of making the right predictions.

- Accuracy = (TP + TN) / (TP + TN + FP + FN)
- Accuracy = (3 + 18) / (3 + 18 + 3 + 3) = 0.7778

In R, we use the following line:
glm_accuracy = sum(glm_confusion_matrix[1,1]+glm_confusion_matrix[2,2])/L

glm_confusion_matrix is a data frame.
glm_confusion_matrix[1,1] = first column and first row of the data frame = 3
glm_confusion_matrix[2,2] = second column and second row of the data frame = 18
Both values are added using the sum() function.
The sum is then divided by the length of the data frame i.e. total number of values.
glm_accuracy = sum(3 + 18)/27 = 21/27 = 0.7778

Sensitivity / True Positive Rate / Recall

Ability to predict positive values, when the actual value is positive.

Formula: TP/(TP + FN

18/(18+3) = 0.857

In this case, the model's ability to predict Yes, when the target is actually Yes is 86% of the time.

Specificity

Ability to predict negative values, when the actual value is negative.

Formula: TN/(TN+FP)

3/(3 + 3) = 0.5

In this case, the model's ability to predict No, when the target is actually No is 50% of the time.

Precision

The proportion of values that the model says is relevant, and are actually relevant.

Formula: 18/(18+3)

18/(18+3) = 0.857

In this case, when the model predicts Yes, it is correct 86% of the time.

Deployment

The machine learning model is made in R. C# is not capable of loading R models directly into the code.
The first choice considered is building a decision tree and coding it into the application using if-else statements. But, this limits the choice of algorithms and might overlook a more accurate model.

R-Shiny

As part of the project, an R-Shiny Application was created. The application is deployed on shinyapps.io. It is integrated into the application using the WebView XAML control which opens a browser window inside the application and displays the R-Shiny app.

The R-Shiny Application reads the dataset and performs supervised learning using classification algorithms such as Generalised Linear Models, Support Vector Machine, Decision Tree, and Naïve Bayes.

The user interface of the application uses the Shiny Dashboard package, to integrate colours, fonts, and most especially to integrate a dashboard view of the model and results.
The application takes in user input which are the independent variables for example, the item price, category, monthly income of user etc. Users can also choose which of the four algorithm to use. The dashboard shows the following items:

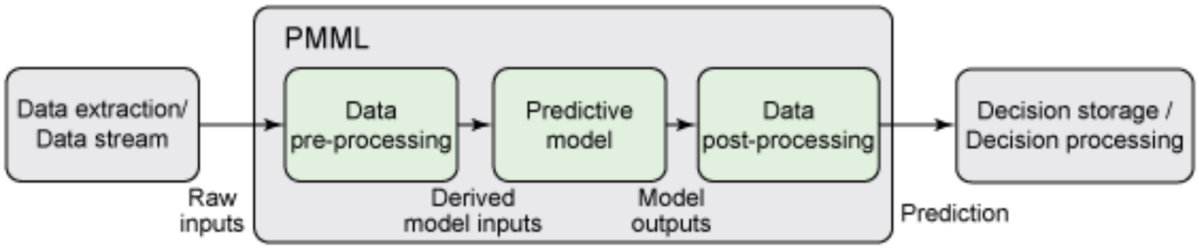
- Recommendation results – Yes or No
- Accuracy
- Model Summary
- Train and Test sets
- Data frame of predicted values versus the actual values when the model created is tested on the test sets.
- Confusion matrix

The R-Shiny Application is not implemented in the final version of the mobile application as it does not have the capacity to take values from C# properties. This is not satisfactory.

PMML

PMML stands for Predictive Model Markup Language.

According to IBM (Guazzelli, 2010), 'PMML is the de facto standard language used to represent data mining models'. It is developed by the Data Mining Group and this is the reason why leading data mining tools can export and import PMML. The models produced by machine learning algorithms can be created in one system and then saved to a PMML file. This file can be used in another system to make predictions. Structure of PMML file:



(Guazzelli, 2010)

For this project, the predictive models are built in R. Each model uses one of the three algorithms; Naïve Bayes, Generalised Linear Models, and Decision Trees. Using the R PMML package from CRAN, the models are exported into a PMML file. These files are added into the code project as an embedded resource.

To implement the model inside the mobile application, the Syncfusion PMML Execution Engine is installed. From the Syncfusion website (2019), it is described as 'a C# library developed for predicting results based on predicted modelling done in PMML over the input data'. The library can run on .NET platforms. This project uses the Syncfusion PMML .Net Core package. Thus, input data and values from C# variables and properties, which corresponds with the independent variables, can be passed from the application to the model. Using the Syncfusion PMML Execution Engine; a PMMLEvaluator instance can be created. This evaluator can consume the input data to produce a predicted value for the output variable.

In the final version of the application, only the GLM PMML file is used because it is the most accurate model based on testing. Also, the balance and monthly income are extracted from C# properties and the rest of the input are taken from a form, these are used in the evaluator to produce a prediction which suggests whether the user should go ahead with a purchase.

By integrating the predictive model into the application, a mobile decision support system is produced.

Chapter 5: Results and Evaluation

This chapter describes the testing and evaluation of the mobile application developed. We evaluate the process of project from the start. The concept testing asks the question as to whether the mobile application works with its planned features. Application and User Acceptance testing refers to the testing of the produced mobile application. We also look at the objectives and requirements of the application.

Process Assessment

Project Proposal – Submitted on October 27, 2018

The project proposal was finished on time and submitted on the due date. The author found that coming up with an idea for the project was challenging. The author knew that she wants to make a tangible application and apply machine learning into the project. The author also has an interest in financial management, and therefore decided to base the project on this.

Interim Report – Submitted on January 29, 2019

After evaluating the project proposal with the project supervisor, it was decided that the project will be simplified from the original project proposal. The original project was a banking application that will use numerous APIs to complete its processes. However, it was realised that this project is too large for the duration of the course. The project is changed to a financial management application, where users can document their finances. The recommendation tool idea was brought in the interim report as well. The interim report lays out the plans for the application to be developed.

Artefact – Created from January 29 – March 21, 2019

Collecting the dataset for building the prediction model was very challenging as the author is not an expert in this field. The survey was built with Google Forms and did not yield a lot of responses. While building the model with R, the author came across challenges with the dataset. The dataset had to be prepared for modelling i.e. change column names, remove NA values etc. After building the final model, the author built an RShiny Application to be displayed in the mobile application. However, the author was not fully satisfied with this, and the project aims to take user data collected through the application, and make recommendation using this. After further research, the model is exported to a PMML file and this was used to bring the model to the C# application.

The UI of the application is written in XAML. With prior knowledge of XAML in WPF, the author was comfortable with the language. However, there are tools and controls that are new and different when developing mobile applications. The author was able to overcome this to produce a UI with a clean design.

The application code is written in C# using Xamarin.Forms and MVVM Framework. The author is delighted to have worked with these technologies as it allowed her to improve at C# and develop a cross-platform application.

Software Development Process

The mobile application was build using the Agile Software Development Approach.

1. **PLAN**

   This phase includes the project proposal and interim report phase. This includes gathering requirements for the project such as, a tailored dataset, planned features, software and hardware resources.

2. **BUILD**

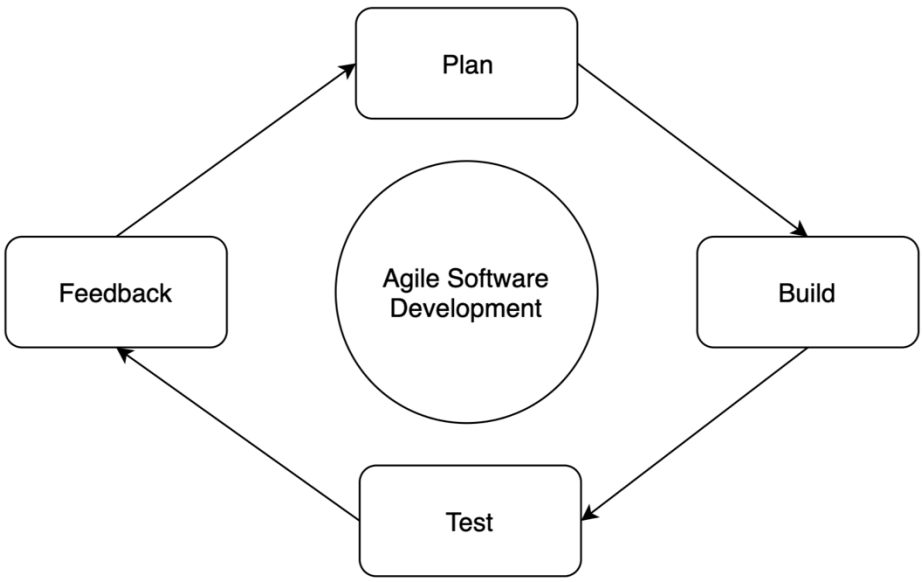
   Once the requirements and materials are in place. The mobile application and machine learning model is built. This included coding the application in C# and, designing the user interface. As well as, preparing the dataset for modelling, and building the model in R.

3. **TEST**

   When the application was built, testing is conducted on an Android device. This phase examines the ability of the application and whether it has reached its maximum potential and completed its requirements.

4. **FEEDBACK**

   This phase included handing the application to other users and showing to the project supervisor. Once, feedback was gathered, any changes or new features suggested are planned, built, and tested.



Concept Evaluation

Does the mobile application actually work?

The application developed was tested against its original requirements: documenting and tracking finances, with recommendation tool.

The application developed does meet these requirements. The mobile application is able to run on an Android and iOS emulator. The application has also been deployed to a Oneplus6T mobile phone, running OxygenOS. The application is able to take user entries and store it into a local database. The application is able to filter the entries into categories, making it easy for users to track their incomes and expenses.

Does the prediction model work?

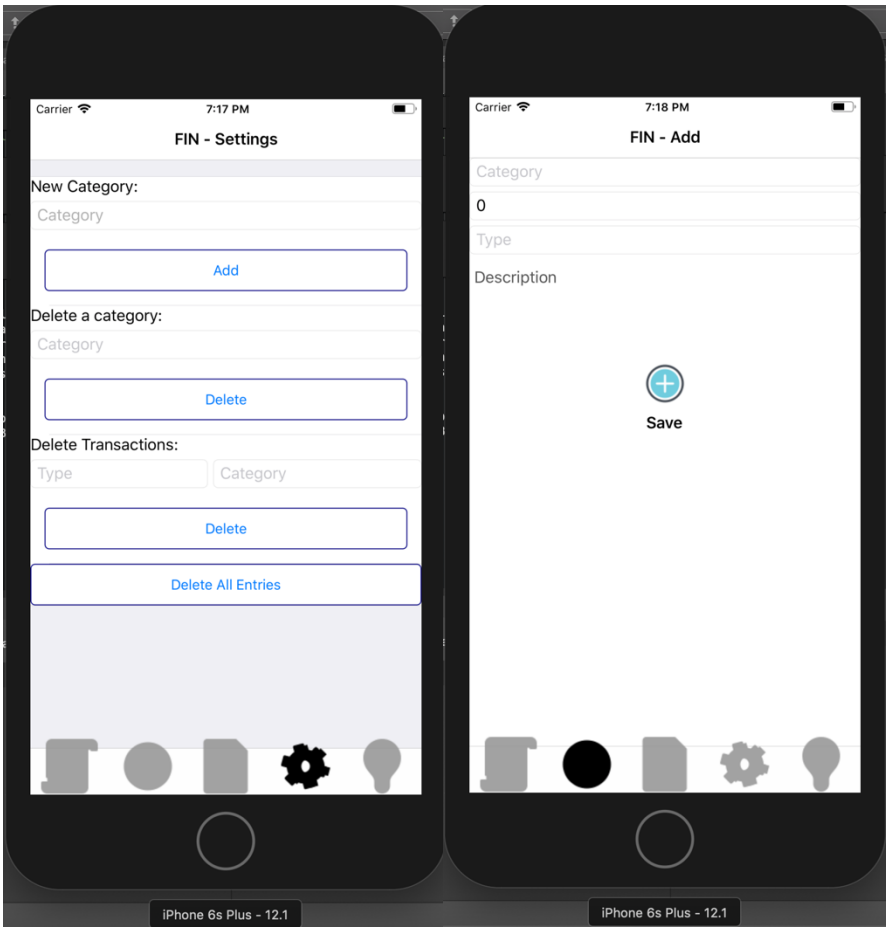
Using the dataset from Google Forms survey, the author built a prediction model to be integrated into the application. The dataset collected did not produce a lot of responses therefore the prediction model is admittedly not accurate. The model used in the application is 71% accurate when tested on a testset. The prediction model works but it has its limitations. However, from testing, it can recognise extremities and produces obvious answers, i.e. YES for Needs and NO for Want = 1.

Application Testing

An Android and iOS emulator is used to test the application's cross-platform compatibility.

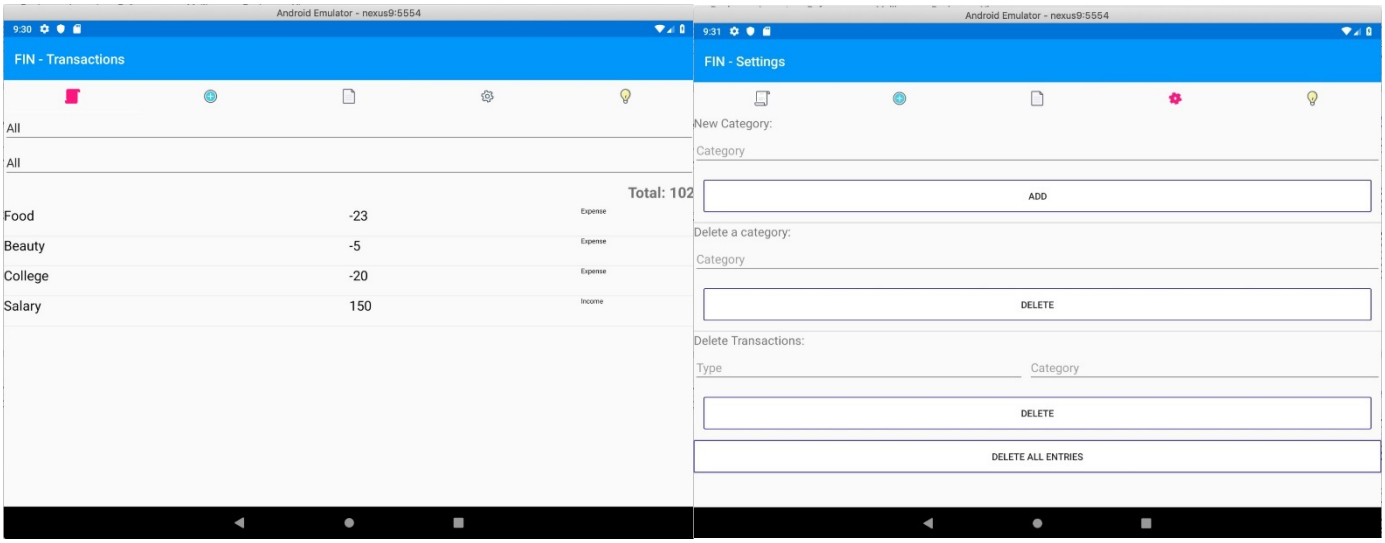
During development of the application, it is purposely made and designed for an android device. The application was continuously tested on an android device every time a UI feature or a functionality is added, even whenever a colour is changed. Therefore, colour schemes, layouts and functionalities may not work as expected on iOS even though it is cross-platform.

Here is a screenshot of the application running on an iPhone 6s Plus.



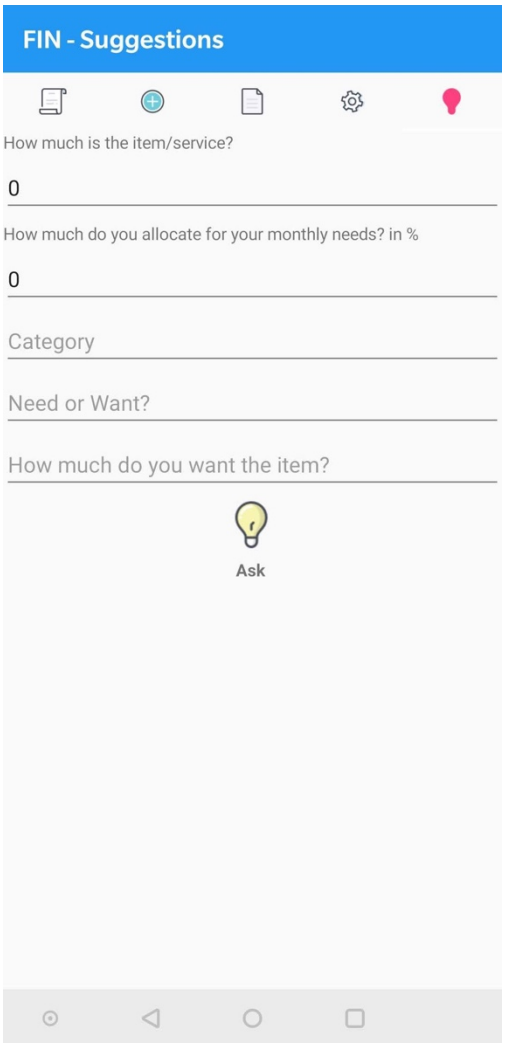
It is evident that the menu bar is not displaying like it does on Android devices. The default position of the menu bar in iOS is at the bottom. When clicking around the bottom menu, we can navigate through the pages as normal. The icons are not rendering properly. We also see that the colour scheme is different. Since the developer did not apply own style templates for the Xamarin.iOS project, the device is using its default colour theme.

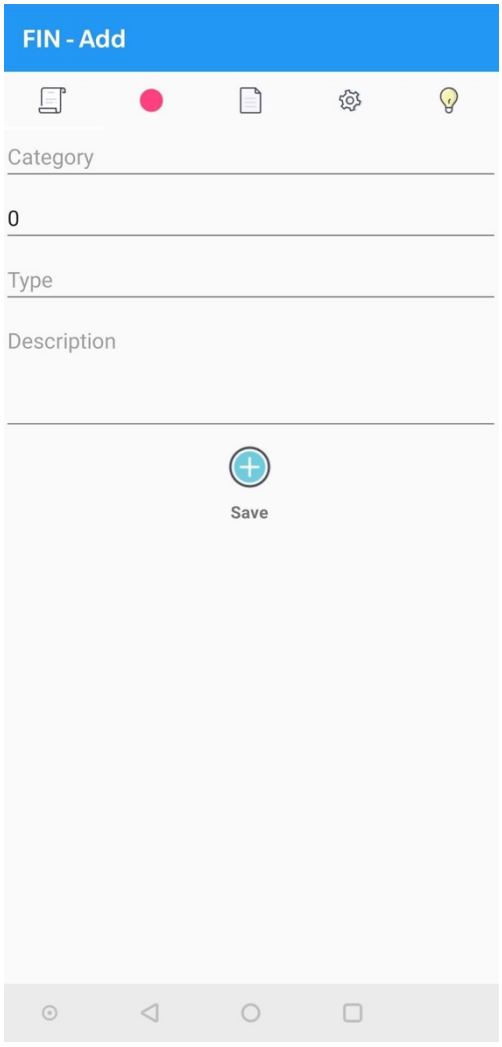
Here is a screenshot of the application running on an android tablet. (Nexus 9).



It can be seen that the user interface has expanded; buttons and text cells are wider. The colour scheme and icons are appearing as expected in this android tablet emulator.

Here are screenshots of the application running on the testing device.





User Acceptance Testing

In order to fairly evaluate the application, an acquaintance was asked to review the application.

The tester is not well-known to the developer, nor have this person seen any of the developer's previous work. This person was chosen to avoid any bias.
The tester was provided screenshots of the application with a brief description of what each functionality does. The experiment lasted 10 minutes. Below are the tester's impressions of the application and developer's evaluation. Refer to Appendices: User Guide.

Tester's thoughts will be written in *Italic* and developer's evaluation in standard writing.

User Interface Design : *'The application is overall bright, with blue coloured bars and fuchsia accent colours. The fonts are the right size, it's not too big, not too small. The menu bar at the top could be larger. But the icons are helpful and are self-explanatory.'*
The developer admits that the colour scheme for the application is not the best. However, the colours stand out from each other and elements are displayed fairly well. The menu bar is also admittedly small. However, to cater for five menu items and ensure that icons keep their quality, the developer opted for the smaller menu items. Better quality icons should have been sourced for this purpose.

Add entry: '*The Add Entry page is straight-forward. It asks for four fields and the description is optional. I noticed that users are not allowed to put in custom dates and entries are saved with the current dates.'*
The developer admits that it would be beneficial to have more details for each entry. The dates are indeed set to the current date and time the user saves the entry. This was done to simplify this component of the project and for time management. Adding more details on this form would mean more data columns, more properties, and more fields. The developer deemed this is not necessary for this part of the project, and the time could be used more effectively in developing other features.

Transaction List: '*This page is nice and clean, just as I would write it on paper. The writing is clear and clean. It shows the list and my total balance! No need for my calculator! I can also click on each entry and I can see more information about it.'*
The developer appreciates this feedback. Having a long list with too much details can be overwhelming for the end-user. The balance was included in this page for the users' convenience. The one thing that the developer would consider revising is the layout of each entry, i.e. should category come first? Price? Or type?. However, the design lists the entries effectively for the purpose of the project.

Summary Page: '*This page does not show much. It only has income, expense, and balance. I like the category filter though, it gives income and expenses for each category.'*
The developer admits not having a lot of details on this page. This is due to time constraints. However, the developer integrated a filter in this page to give it more purpose i.e. summary of income and expenses for each category. The developer hoped to integrate visualisations in this page.

Settings page: '*The settings page is clear. I appreciate that we can add personal categories, it makes the list more personal. Deleting many transactions and deleting all entries at once is also helpful, if ever I need to reset my list.'*
The settings page was designed as clean and clear as possible as the developer does not want users to be confused with the buttons to avoid accidental deletions of entries. The developer appreciates the feedback

Suggestions page: '*This suggestion tool is very handy! I relate with the developer's problem of not knowing whether she can afford something she wants. I like how I can describe the item that I want to buy in the form and the tool also takes into account my balance and income, then makes a prediction out of that.'*
The developer built the application because of this problem (refer to Chapter 1: Introduction). The form asks for five descriptors of the item/service that the user wants to purchase. However, the developer aims for the model to be smarter and ask less descriptors from the user, so the model will use more data from the application when making predictions.

Chapter 6: Conclusions and Future Work

Modifications

In this section, planned features discussed in the Interim Report which has not been included in the final artefact and the reasons for exclusion will be discussed.

- Visualisation – XAML has limited visualisation tools. The tools that the developer think would be the best for the data to be displayed are subscriptions that need to be paid for. Hence, this will not be in included in this project.
- Acquire a copy of their history of transactions. Generate a PDF file of transactions and send it to the account holder's e-mail address – where user can export the visualisations to a PDF for printing. However, visualisations are not implemented. Also, a mail server will have to be set up and that is out of scope of this project.
- Sync their personal banking account transactions with the application – out of scope of this project. Application security has not been enforced therefore it is not a great idea to connect real financial accounts to the application.
- FIN will be able to support other currencies – since genuine money is not involved, currencies are not implemented in the artefact.
- Notifications and savings wallets – due to time constraints, the app developer decided that these features do not need to be implemented as part of this project.
- Upload a file associated to an entry – since application files are stored in a local database in the device, uploading files i.e. images or documents may affect the performance of the application.
- Download the FIN mobile application from the Google Play Store for free – further testing and modifications are required before deploying the application in Google Play Store.
- Register for an account using their personal details i.e. Full name, email address, contact number etc. – since the application uses SQLite, the data is stored locally on the device. For that reason, the developer considers user registration not required.

Reflection

By doing this project, I have honed my software development skills. After studying the MVVM Framework, it has greatly assisted me in creating multi-functional, reusable, and maintainable code. I have also greatly improved my ability to program in C#. Whereby the longer I have been doing the project, the less I find myself seeking help and instructions from the internet. I think that this is a great skill to have in the industry. I am grateful to have learned how to produce a cross-platform mobile application using Xamarin.Forms in a world where mobile devices are continuously making impact on everyone's lives.

Furthermore, I improved my understanding of data analytics. I am pleased with the concept of my project which shows my ability to model data from start to finish. I collected data using Google Forms, cleaned and prepared it for modelling with Excel. I used different algorithms to identify the most important features of the dataset with RapidMiner. Then, I made predictive models using R. I then exported the model into a PMML file which is embedded in the mobile application. This serves as a mobile decision support system inside the app.

I, the author, developer, and learner for this project, is satisfied with the final product and knowledge gained during the process. At the start of the project, I had no prior experience in mobile application development and very little understanding of machine learning. I am pleased with what I have achieved with this project. I have accomplished a great deal of new knowledge and skills, along with producing a tangible mobile application which I will present to future employers.

Future Work

Many different adaptations, tests, and experiments have been left for the future due to time constraints. Below is a list of advices that can be made to improve the application

- Better dataset
    - The survey can be improved and extended. By researching more about consumer purchasing decisions and consumer spending habits; more effective questions can be asked.
    - By advertising the survey, more responses can be collected.
- More algorithms for predictive model
    - During the course of the project, we use the algorithms: Naïve Bayes, Decision Trees, and Generalised Linear Models. More algorithms can be tested to create the model. Other algorithms may work better and perform predictions more accurately.
- Extending compatibility to iOS and Windows Phone
    - There is already an iOS project in the C# code solutions. But this has not yet been optimised. The theme can be improved to match the platform.
    - Xamarin.Forms also offers the capacity to create Windows Phone applications. This can be easily added to the project solution
- For additional future work please refer to ‘Modifications’ section, where there is a list of features that were mentioned in the interim report but have not been implemented into the final version of the application.

Conclusion

The project produced an application that is able to document user entries that describes incoming and outgoing transactions. It provides the necessary template for listing and calculating transactions, which then saves users from physically writing and calculating their finances. The mobile application is written in C# using Xamarin toolkit and is compatible with Android devices, and has limited compatibility with iOS. The project also produced a predictive model which has ability to make a recommendation whether the user shall go ahead with a purchase. The mobile application stores data in a local database and the predictive model is embedded in the app. These features make the application a portable tool for end-users, so they can write transaction entries and use the suggestion tool anytime, anywhere and without the need for an internet connection. This application provides a mobile decision support system which can assist in timely decision making processes.

References

1. Statista. (2019). *Number of smartphone users worldwide 2014-2020 | Statista*. [online] Available at: https://www.statista.com/statistics/330695/number-of-smartphone-users-worldwide/ [Accessed 30 Mar. 2019].

2. Hoomans, D. (2015). *35,000 Decisions: The Great Choices of Strategic Leaders*. [online] Go.roberts.edu. Available at: https://go.roberts.edu/leadingedge/the-great-choices-of-strategic-leaders [Accessed 27 Mar. 2019].

3. Pérez, I. J., Alonso, S., Cabrerizo, F. J., and Herrera-Viedma, E. (2008). A decision support system based on mobile internet. *In XIV Congreso Español sobre Tecnologias y Lógica*.

4. Jaradat, M., Imlawi, J. and Mashaqba, A. (2018). Investigating the Moderating Effects of Self-Efficacy, Age and Gender in the Context of Nursing Mobile Decision Support Systems Adoption: A Developing Country Perspective. *International Journal of Interactive Mobile Technologies (iJIM)*, 12(2).

5. Guo, X. and Díaz López, A. (2013). Mobile Decision Support System Usage in Organizations. *Proceedings of the Nineteenth Americas Conference on Information Systems, Chicago, Illinois,*.

6. Gale, W. and Levine, R. (2009). Financial Literacy: What Works? How Could it be More Effective?. *SSRN Electronic Journal*. [online] Available at: https://www.rand.org/content/dam/rand/www/external/events/2010/11/18/financial-literacy-what-works-presentation.pdf [Accessed 20 Mar. 2019].

7. The Aspen Institute. (2018). *The Burden of Debt on Mental and Physical Health*. [online] Available at: https://www.aspeninstitute.org/blog-posts/hidden-costs-of-consumer-debt/ [Accessed 21 Mar. 2019].

8. Constant, F., Guevara, J., Silva, C., Gonçalves, D. and Marcelino, L. (2017). Decision Support System using Mobile App Statistics. *Proceedings of the 12th Iberian Conference on Information Systems and Technologies*.
Selvakumar, M., Anbuchezhienkamaraj, P., Sathyalakshmi, V. and Mohammed Arubakkar Siddique, R. (2018). Role of Commercial Banks in Financial Literacy. *Productivity*, 59(2).

9. Guest, R. and Brimble, M. (2018). FINANCIAL LITERACY 101. *Policy*, 34(1).

10. Danes, S. and Hira, T. (1987). 19. *Money Management Knowledge of College Students*, 17(1).

11. Play.google.com. (2019). *Spendee - Budget and Expense Tracker & Planner*. [online] Available at: https://play.google.com/store/apps/details?id=com.cleevio.spendee [Accessed 30 Mar. 2019].

12. Play.google.com. (2019). *Wallet - Finance Tracker and Budget Planner*. [online] Available at: https://play.google.com/store/apps/details?id=com.droid4you.application.wallet [Accessed 30 Mar. 2019].

13. Play.google.com. (2019). *Monefy - Money Manager* -. [online] Available at: https://play.google.com/store/apps/details?id=com.monefy.app.lite [Accessed 30 Mar. 2019].

14. Play.google.com. (2019). *Money Lover: Expense Tracker & Budget Planner*. [online] Available at: https://play.google.com/store/apps/details?id=com.bookmark.money [Accessed 30 Mar. 2019].

15. Docs.microsoft.com. (2019). *Xamarin.Forms - Xamarin*. [online] Available at: https://docs.microsoft.com/en-us/xamarin/xamarin-forms/ [Accessed 2 Apr. 2019].

16. Docs.microsoft.com. (2017). *Introduction to Mobile Development - Xamarin*. [online] Available at: https://docs.microsoft.com/en-us/xamarin/cross-platform/get-started/introduction-to-mobile-development [Accessed 2 Apr. 2019].

17. Sqlite.org. (2019). *SQLite Home Page*. [online] Available at: https://www.sqlite.org/index.html [Accessed 2 Apr. 2019].

18. Statistics.laerd.com. (2019). *How to perform a Binomial Logistic Regression in SPSS Statistics | Laerd Statistics*. [online] Available at: https://statistics.laerd.com/spss-tutorials/binomial-logistic-regression-using-spss-statistics.php [Accessed 2 Apr. 2019].

19. Docs.h2o.ai. (2019). *Generalized Linear Model (GLM) — H2O 3.24.0.1 documentation*. [online] Available at: http://docs.h2o.ai/h2o/latest-stable/h2o-docs/data-science/glm.html [Accessed 2 Apr. 2019].

20. Guazzelli, A. (2010). *What is PMML?*. [online] Ibm.com. Available at: https://www.ibm.com/developerworks/library/ba-ind-PMML1/index.html [Accessed 2 Apr. 2019].

21. Help.syncfusion.com. (2019). *Overview | PMML Execution Engine | Predictive Analytics | Syncfusion*. [online] Available at: https://help.syncfusion.com/predictive-analytics/pmml-execution-engine/overview [Accessed 2 Apr. 2019].

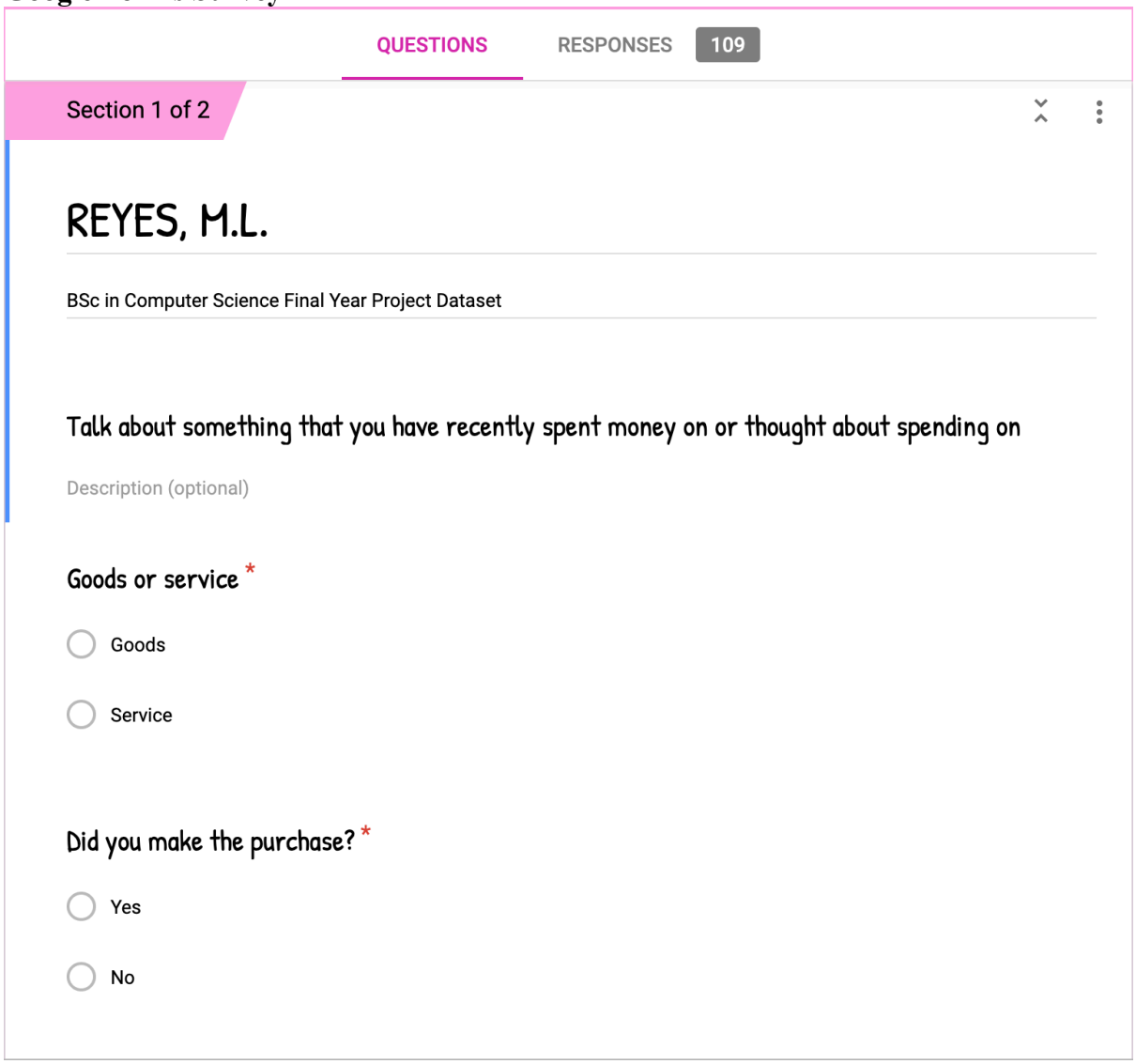
Appendices

See Syncfusion website for a list of machine learning models and .NET platforms the execution engine supports : https://help.syncfusion.com/predictive-analytics/pmml-execution-engine/overview

GLM Model Summary Output

```
Call:
glm(formula = target ~ ., family = binomial, data = trainset)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.3220  -0.1474   0.1680   0.5017   2.6330

Coefficients:
                             Estimate Std. Error z value Pr(>|z|)
(Intercept)                -1.337e+00  2.550e+00  -0.524  0.60006
price                      -7.453e-04  1.441e-03  -0.517  0.60505
balance                    -9.001e-05  3.595e-04  -0.250  0.80230
categoryClothing           -8.023e-01  2.485e+00  -0.323  0.74676
categoryFood & Drink       -2.450e+00  2.449e+00  -1.001  0.31703
categoryHealth & Beauty    -4.669e+00  2.359e+00  -1.979  0.04777 *
categoryPersonal Development -4.144e+00  2.491e+00  -1.664  0.09614 .
categoryTechnology         -3.815e+00  2.268e+00  -1.682  0.09257 .
categoryTransport          -2.988e+00  2.381e+00  -1.255  0.20942
needorwantWant             -2.982e+00  1.600e+00  -1.864  0.06234 .
desire2                     6.529e+00  2.076e+00   3.144  0.00166 **
desire3                     3.258e+00  1.553e+00   2.097  0.03595 *
desireNEED                  3.344e+00  2.018e+00   1.657  0.09760 .
monthlyincome               1.347e-03  5.668e-04   2.376  0.01750 *
monthlyneeds                4.945e-02  2.724e-02   1.815  0.06947 .
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 92.709  on 80  degrees of freedom
Residual deviance: 43.892  on 66  degrees of freedom
  (6 observations deleted due to missingness)
AIC: 73.892

Number of Fisher Scoring iterations: 7
```

Google Forms Survey



Price: *

Short-answer text

How much money do you have in total? *

only include money available for spending in your current allowance /wallet, bank account etc.. (exclude savings accounts, investments, shares etc.)

Short-answer text

Category *

1. Food & Drink
2. Health & Beauty
3. Clothing
4. Technology
5. Home (decors, household equipments, home repairs etc)
6. Books & Education
7. Transport
8. Other (please specify below)
9. Personal Development
10. Experiences

Section 2 of 2

ABOUT YOU

Description (optional)

Gender *

- ○ Male
- ○ Female

Age *

Short-answer text

Monthly Income *

Short-answer text

Monthly Budget for your NEEDS

AS A PERCENTAGE (%) OF YOUR INCOME

Short-answer text

Monthly Budget for your WANTS

AS A PERCENTAGE (%) OF YOUR INCOME

Short-answer text

Status *

1. Single
2. Married (no children)
3. Single Parent
4. Married Parent

Comments:

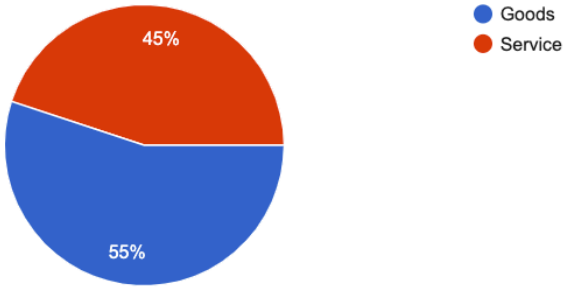
Long-answer text

Google Forms Survey Responses

QUESTIONS RESPONSES 109

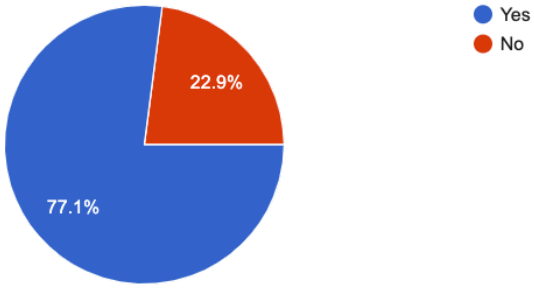
Goods or service

109 responses



Did you make the purchase?

109 responses



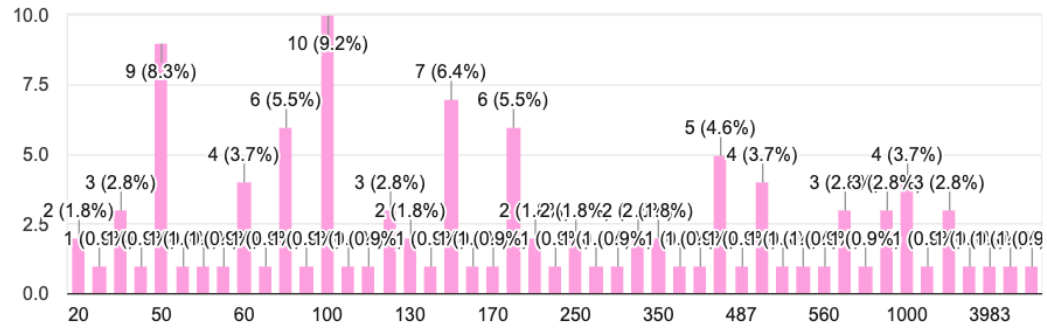
Price:

109 responses

50
5
30
20
45
15

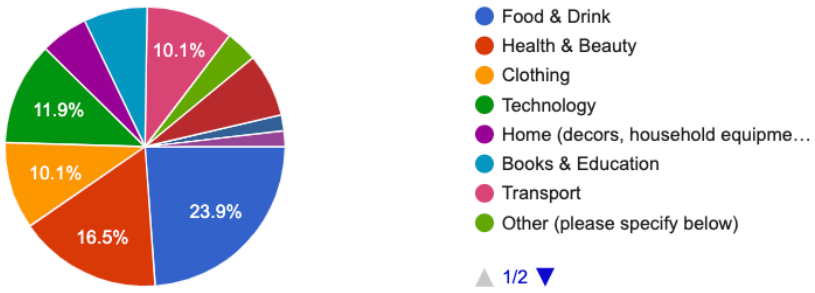
How much money do you have in total?

109 responses



Category

109 responses



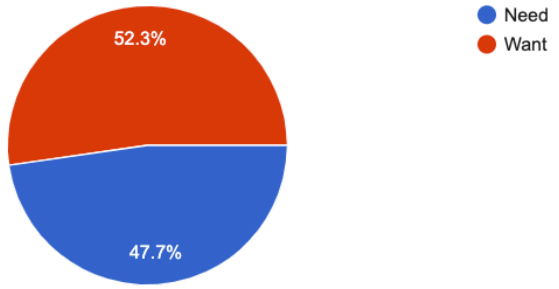
If Other, please specify:

5 responses

Car washing soap
5.00
Tattoo
hotel accomodation
tattoo

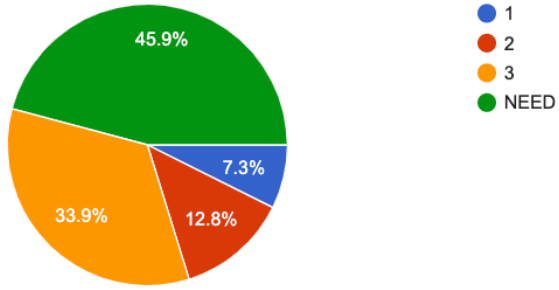
Need or Want?

109 responses



If want, how much do you WANT the item? (1 being the least, 3 being the most)

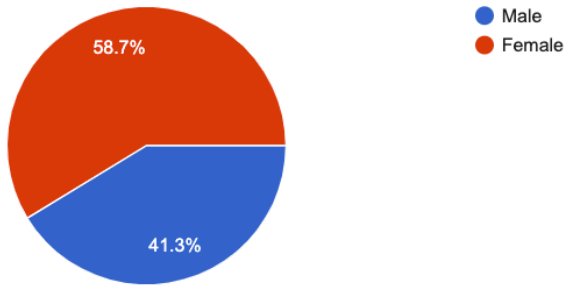
109 responses



ABOUT YOU

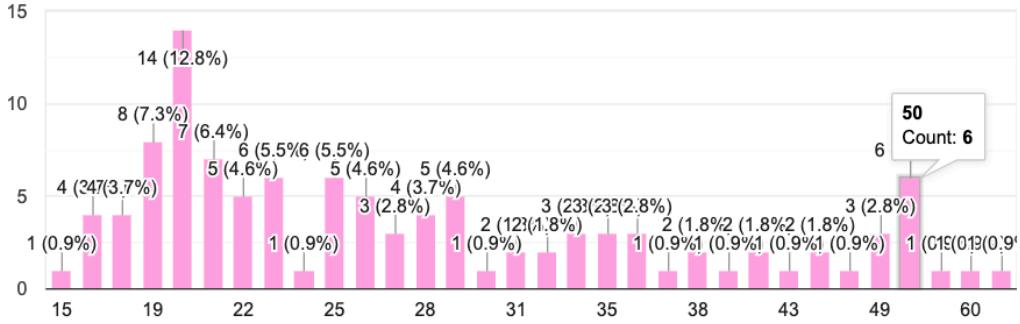
Gender

109 responses



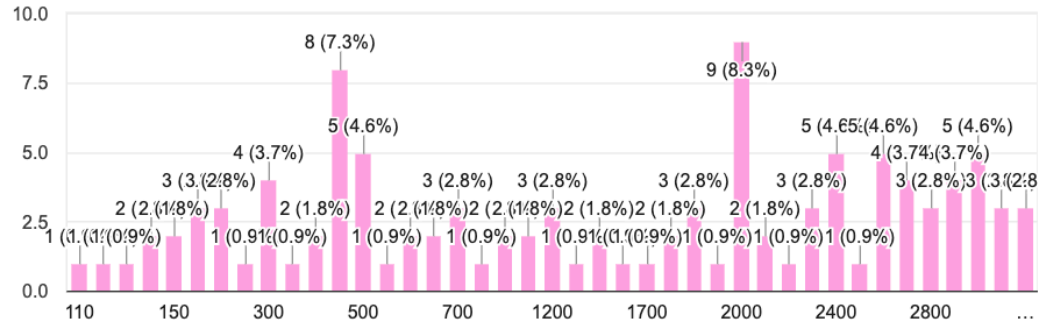
Age

109 responses



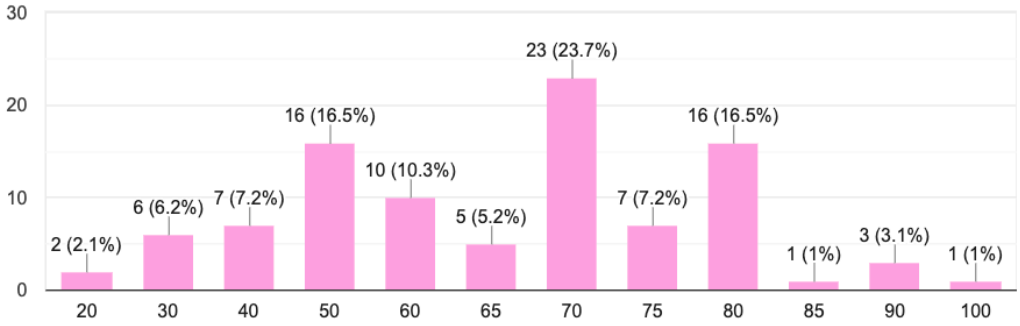
Monthly Income

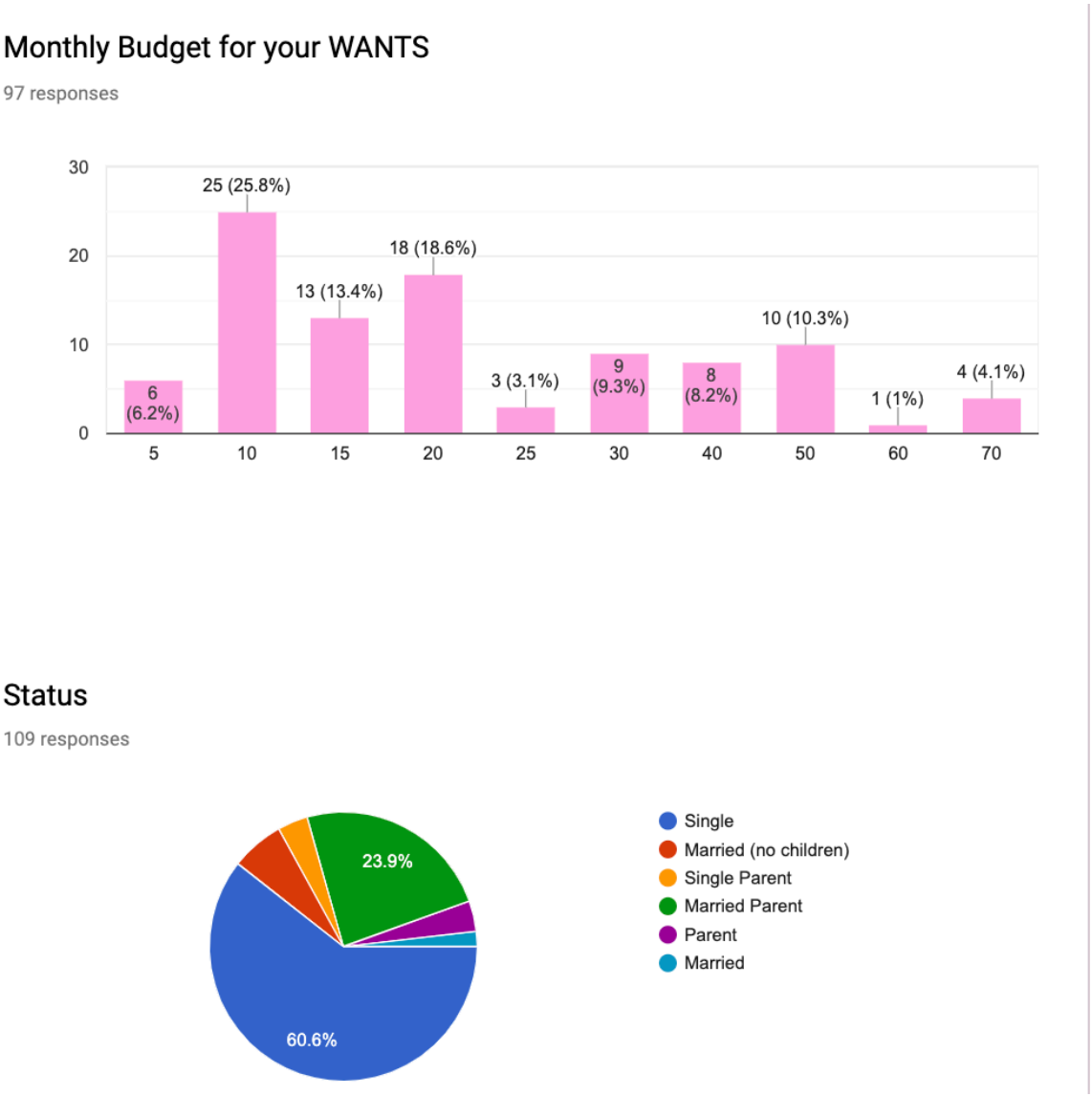
109 responses



Monthly Budget for your NEEDS

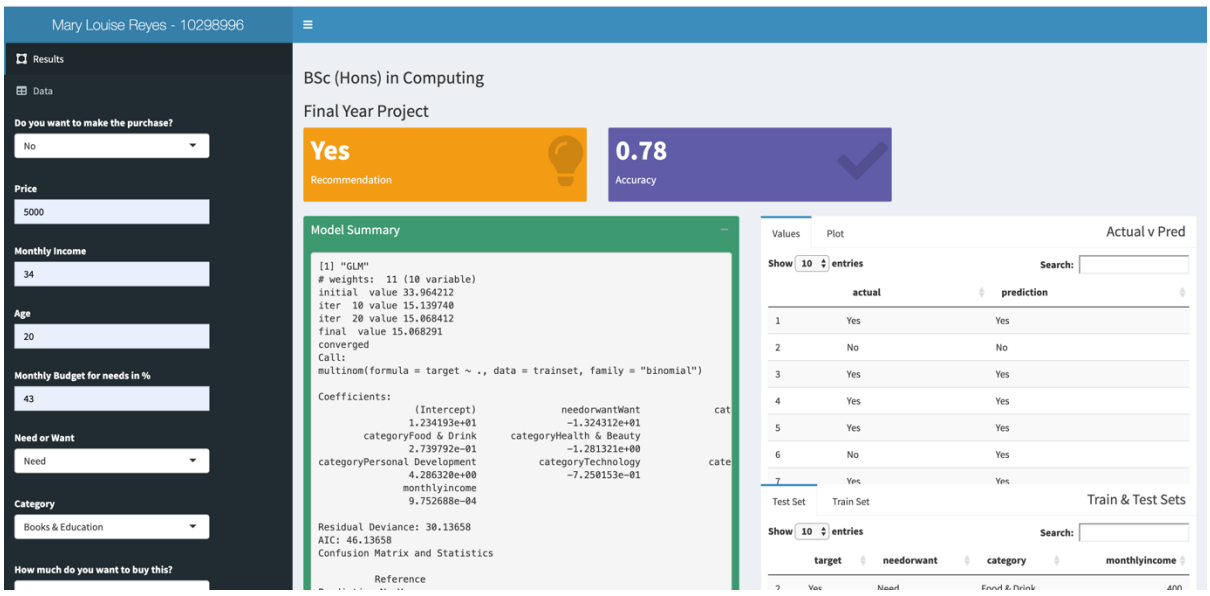
97 responses



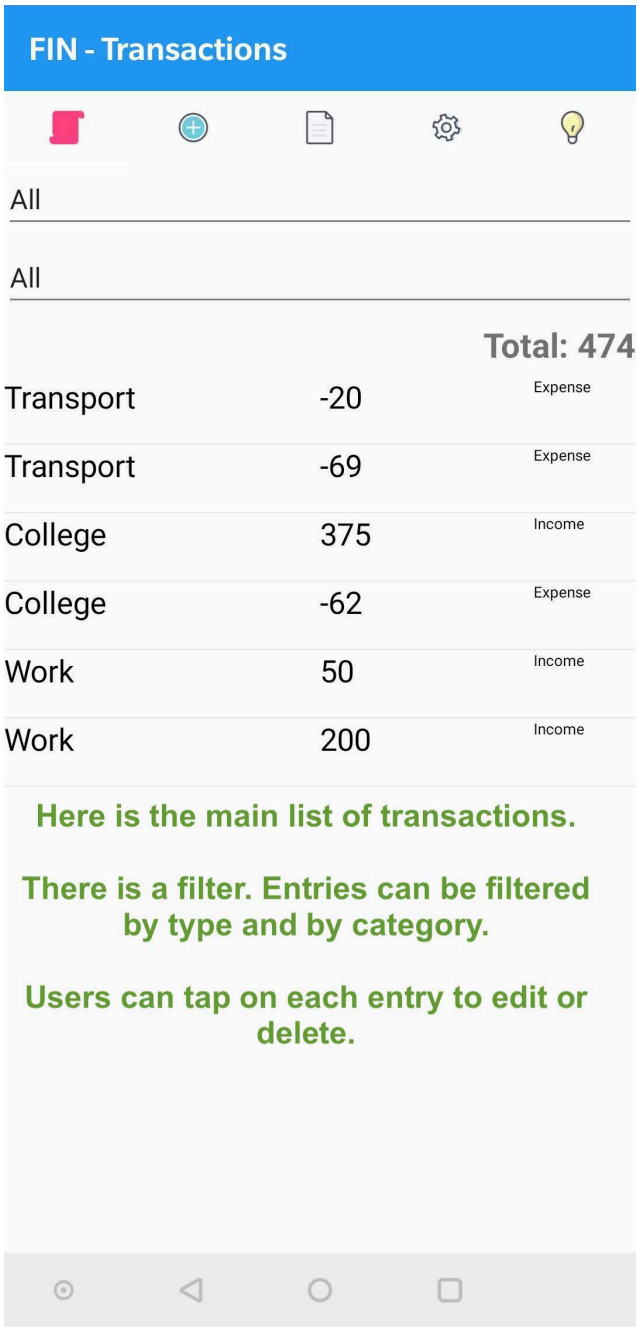


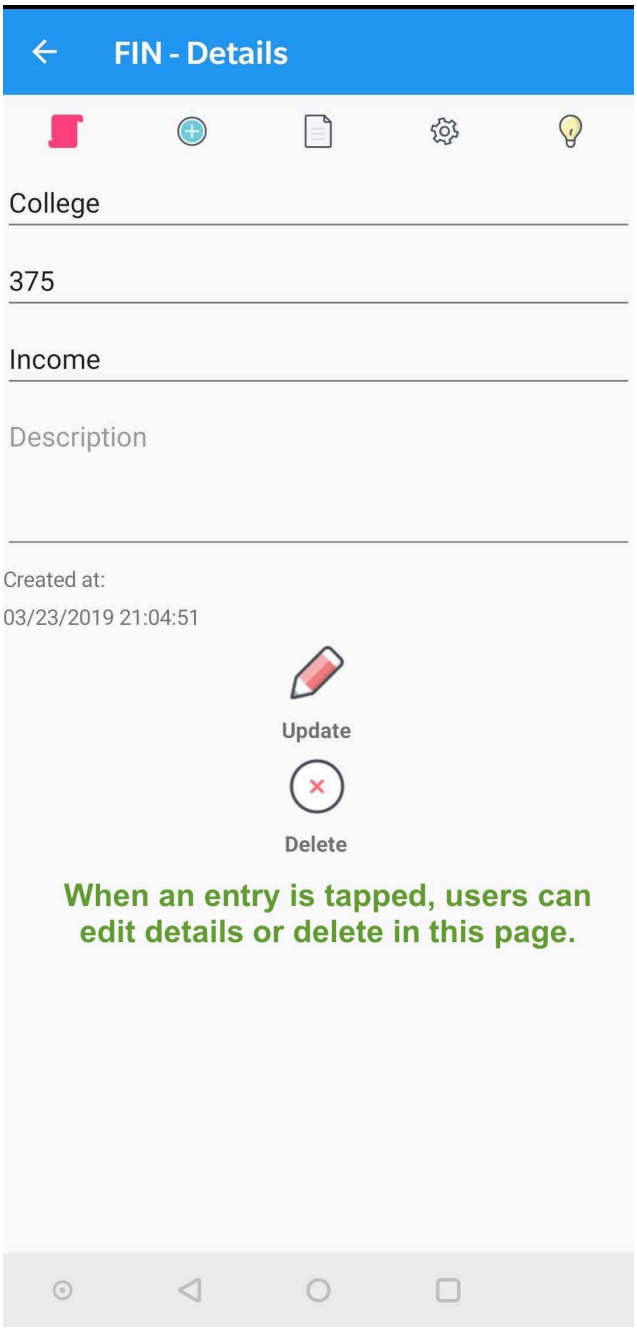
R-Shiny Application

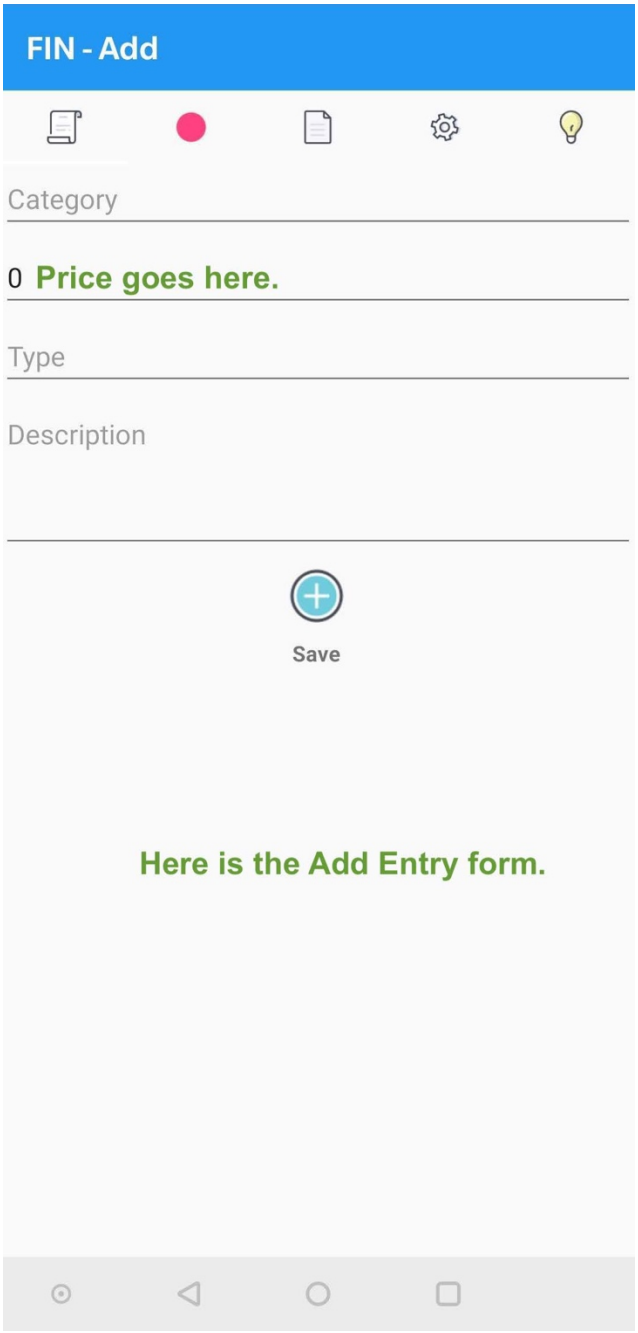
A screenshot of the application.

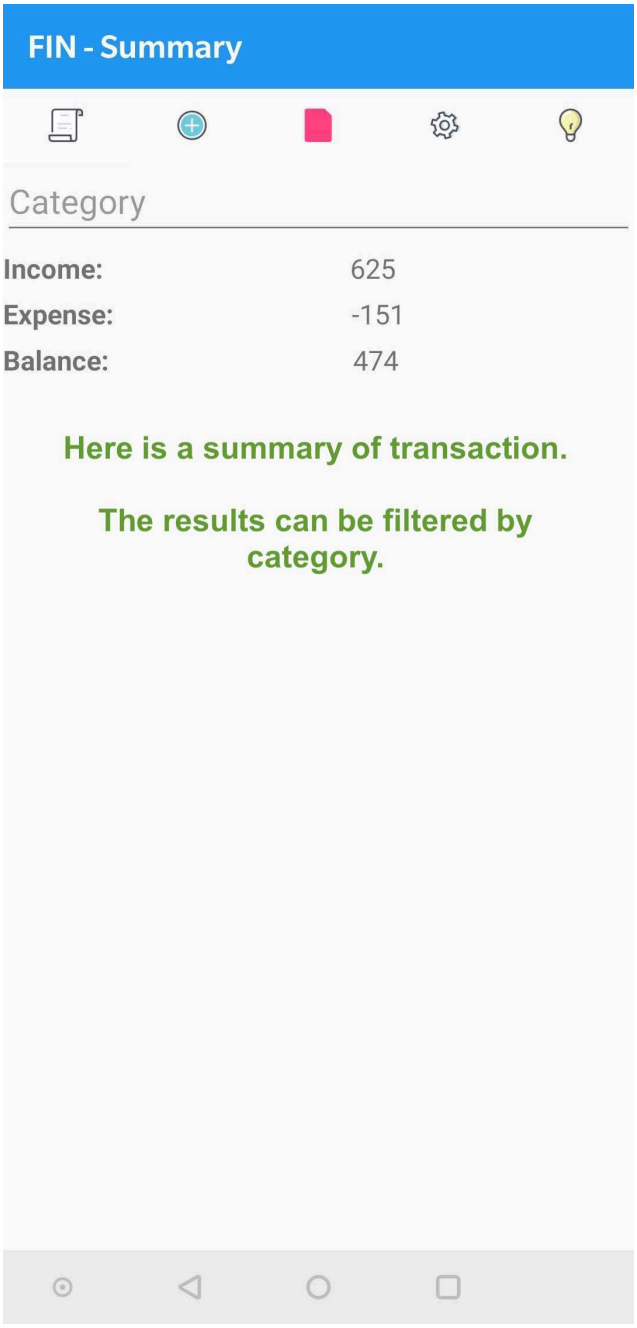


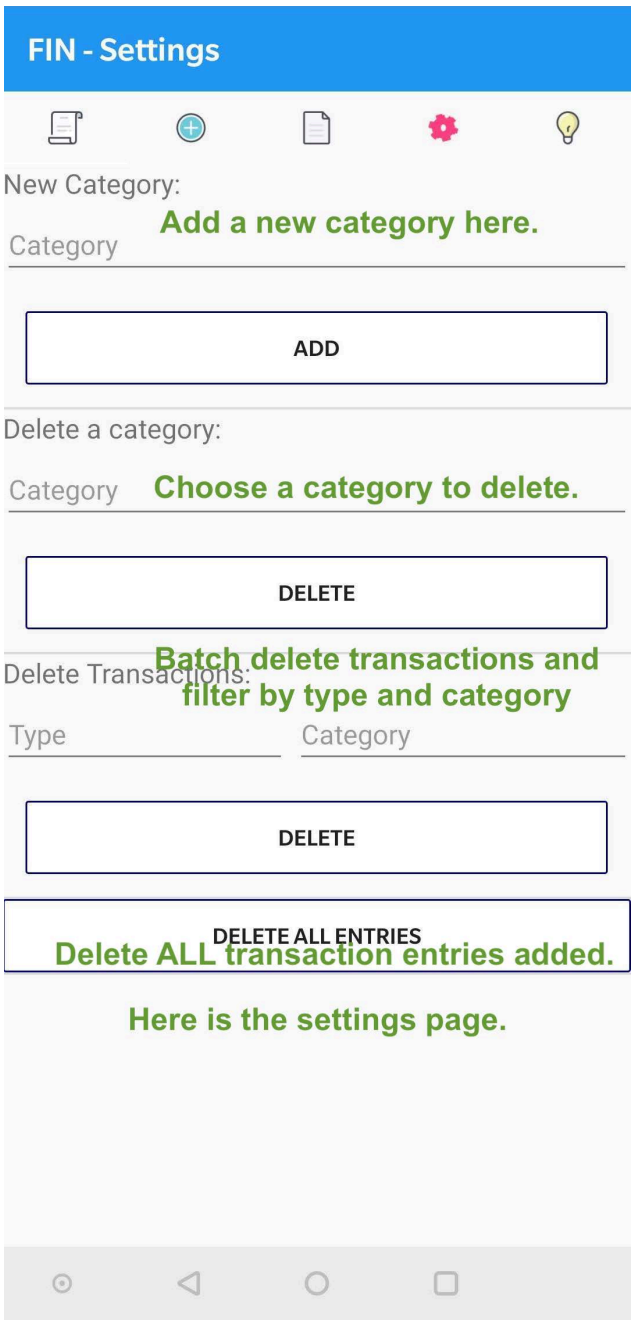
User Guide

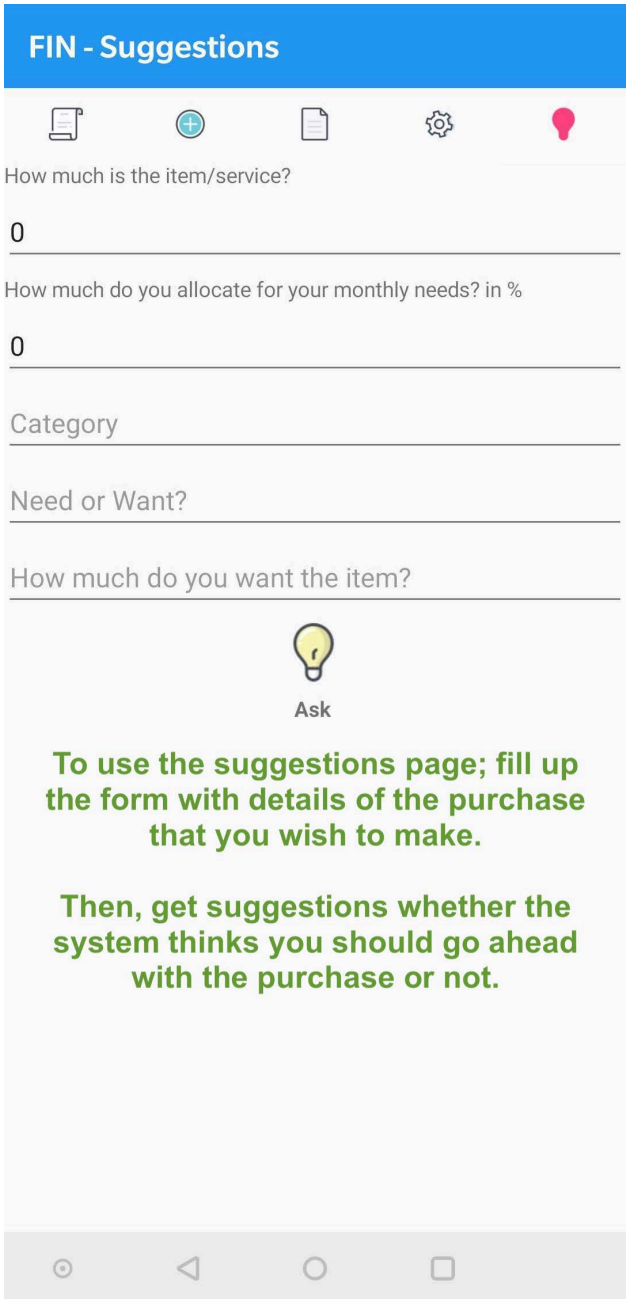

FIN - Transactions
All
All
Total: 474
Transport -20 Expense
Transport -69 Expense
College 375 Income
College -62 Expense
Work 50 Income
Work 200 Income
Here is the main list of transactions.
There is a filter. Entries can be filtered by type and by category.
Users can tap on each entry to edit or delete.

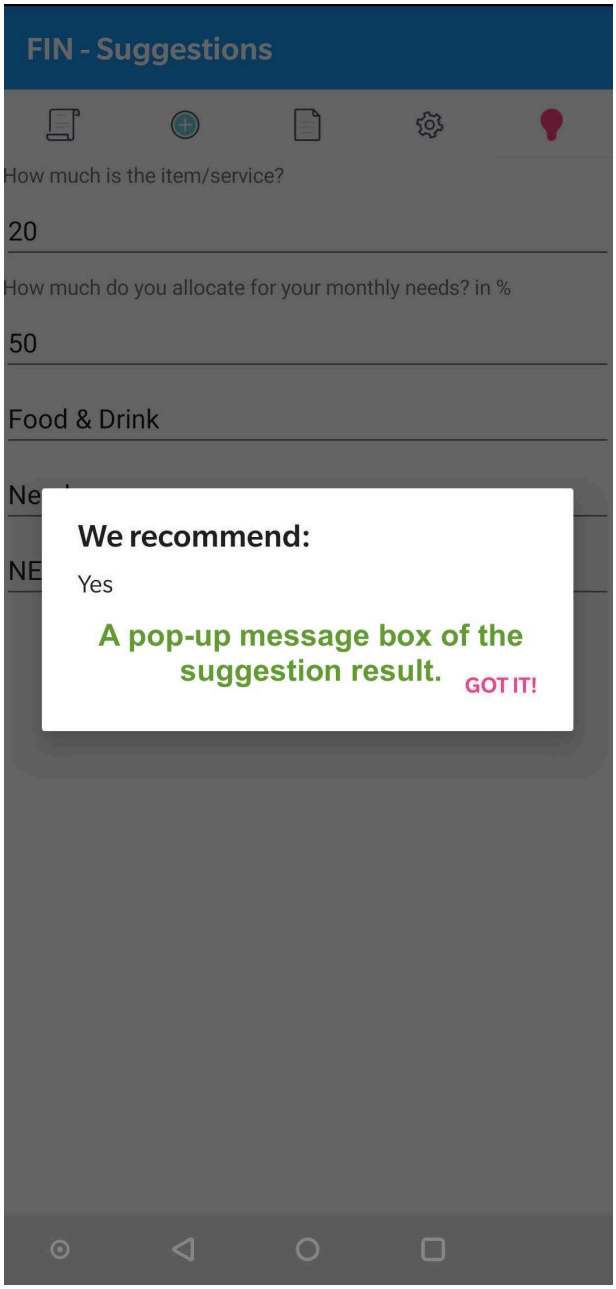

FIN - Details
College
375
Income
Description
Created at:
03/23/2019 21:04:51
Update
Delete
When an entry is tapped, users can edit details or delete in this page.


FIN - Add
Category
0 Price goes here.
Type
Description
Save
Here is the Add Entry form.


FIN - Summary
Category
Income: 625
Expense: -151
Balance: 474
Here is a summary of transaction.
The results can be filtered by category.


FIN - Settings
New Category:
Category
Add a new category here.
ADD
Delete a category:
Category
Choose a category to delete.
DELETE
Delete Transactions:
Batch delete transactions and filter by type and category
Type
Category
DELETE
DELETE ALL ENTRIES
Delete ALL transaction entries added.
Here is the settings page.


FIN - Suggestions
How much is the item/service?
0
How much do you allocate for your monthly needs? in %
0
Category
Need or Want?
How much do you want the item?
Ask
To use the suggestions page; fill up the form with details of the purchase that you wish to make.
Then, get suggestions whether the system thinks you should go ahead with the purchase or not.


FIN - Suggestions
How much is the item/service?
20
How much do you allocate for your monthly needs? in %
50
Food & Drink
We recommend:
Yes
A pop-up message box of the suggestion result.
GOT IT!