"Pocket Manager"

User Manual

- **Table of Contents (Index)**

• Introduction

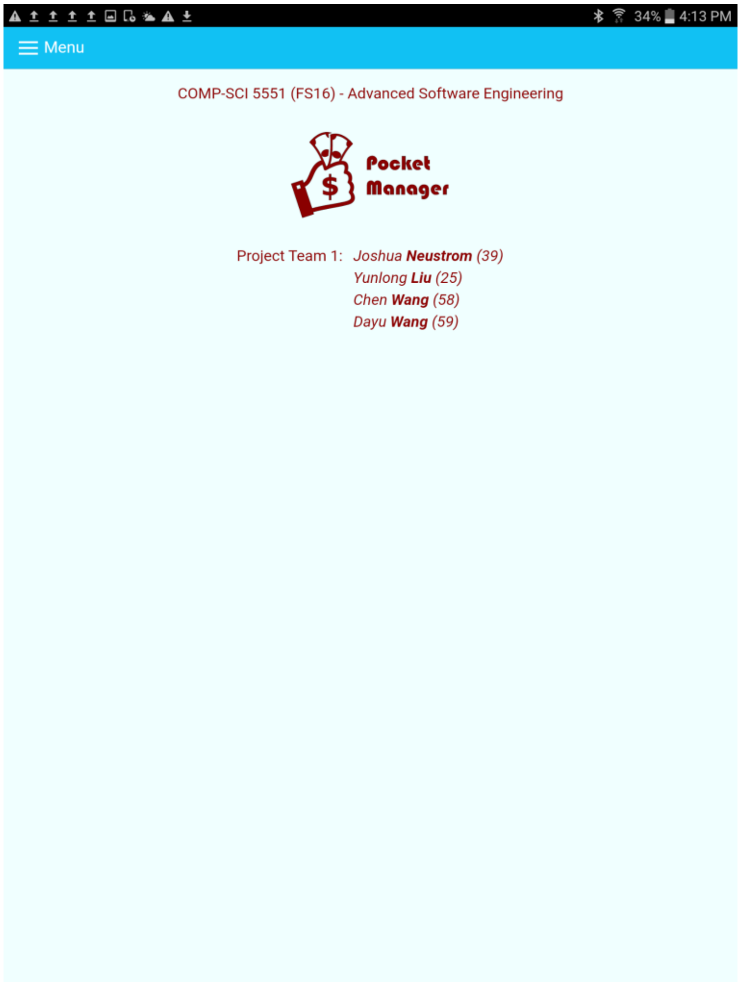
Pocket Manager is an Android application developed in order to serve college student, which can help the youth wisely manager their money in their pocket. The motivation that drove us to develop such a system is based on our own observation of the fact that most college student does not know how to manage their income. CNN says that in 2015, a college student had an average of $3,000 of credit card debt when he/she graduated from college. Even though some students have an monthly income of $1,500, they still fell financially stressed because of they are lack of skills and tips to manage their money. Targeted to some common bad habits in managing money amongst college students, *Pocket Manger* does have powerful function to help college students.

• Know about *Pocket Manager*

Pocket Manager is an Android application that is compatible with almost every size of your device. In order to correct run the application on your device, please make sure that your device fulfills the following requirements:

1. Only compatible with Android 6.0 or higher.
2. Your device must be connected to the Internet.

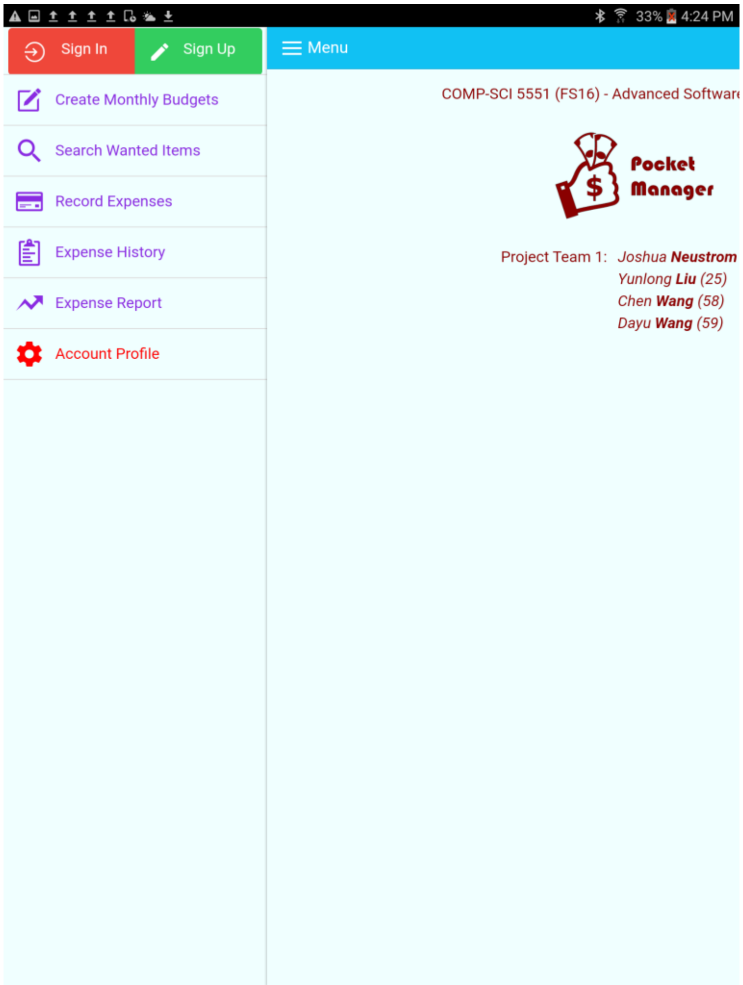
The main page of our system looks like the following.



COMP-SCI 5551 (FS16) - Advanced Software Engineering

Project Team 1: *Joshua* ***Neustrom*** *(39)*
Yunlong ***Liu*** *(25)*
Chen ***Wang*** *(58)*
Dayu ***Wang*** *(59)*

The menu button on the top-left corner opens the side menu, which you start all your work from. The side menu looks like the following:

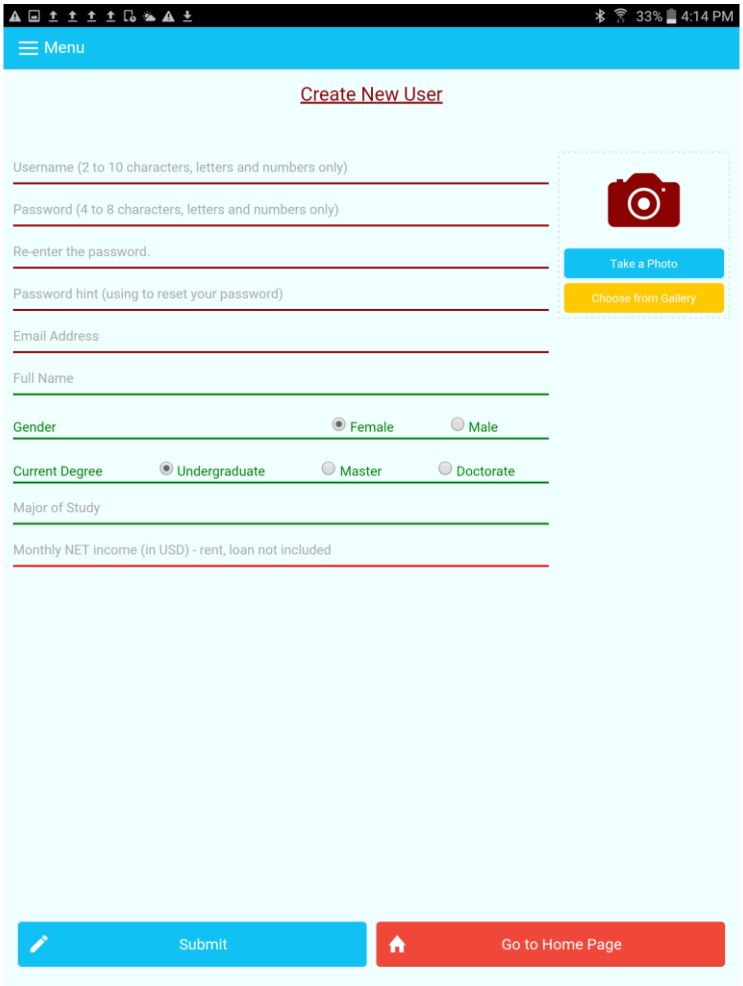


COMP-SCI 5551 (FS16) - Advanced Softwar

Project Team 1: *Joshua **Neustrom***
*Yunlong **Liu** (25)*
*Chen **Wang** (58)*
*Dayu **Wang** (59)*

- **Log-in System**

Since money is a sensitive issue for almost everyone, *Pocket Manger* does not allow the user to do anything without a log-in. User can sign in from the side menu, or register a new account. The registration page looks like the following:



Create New User

Username (2 to 10 characters, letters and numbers only)

Password (4 to 8 characters, letters and numbers only)

Re-enter the password.

Password hint (using to reset your password)

Email Address

Full Name

Gender ● Female ○ Male

Current Degree ● Undergraduate ○ Master ○ Doctorate

Major of Study

Monthly NET income (in USD) - rent, loan not included

Also, *Pocket Manger* allows the user to upload his/her photo which serves as part of his/her profile.

- **Create a Budget**

Target problem:

Because a lot of college students always spend their expense without plan, so this phenomenon is very popular for students which is they always have a lot of money and spend it as millionaire, but at the end of the month, they always lack of money and live like a pauper.

Solution:

To solve this phenomenon, the most important thing is to make a plan, so the reasonable budget is very important for solving this problem. We give a fitting curves to create reasonable budgets for college students. It separates the budget as 4 aspects: Food, Living, Entertainment and Saving. All the parts increasing with the growing of the income. And when the income is not very high, the food part is the dominant part, but with the increasing of income, the more and more expense will be used for the entertainment and saving.

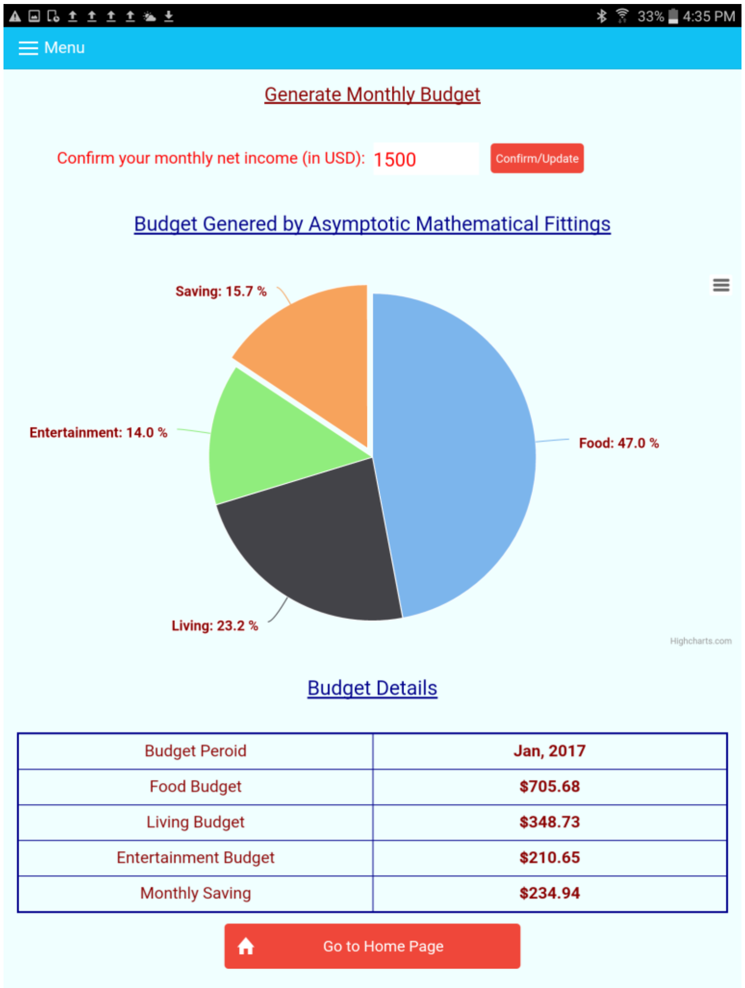
Necessary:

Create budget can help college student manage their expense more efficacious and reasonable, the budget always created based on users' income, it is reasonable and can help users to correct their expense habit.

Feature description:

With this feature, user will get a budget based on their income when they registered, and they also can input another total used for their whole month expense. Then they will get their own budget and they can cost based on its advising.

Result:



Generate Monthly Budget

Confirm your monthly net income (in USD): 1500

Budget Genered by Asymptotic Mathematical Fittings

Budget Details

Budget Peroid	Jan, 2017
Food Budget	$705.68
Living Budget	$348.73
Entertainment Budget	$210.65
Monthly Saving	$234.94

- **Separate Wants from Needs**

target problem:

Because the credit card is more and more popular in the world, the overdraft is very useful and full of temptation for the college student. A lot of student have different kind of credit card and overdraft them excessive. The money in the credit card is much easily to waste than the cash.

solution:

The key to good money management is separating needs from wants. If you aren't sure if an item is a need or a want, do without it for a period of time. If after that time you truly can't live without it, it may be a need. However, even the essentials like shelter or transportation involve a want vs. need calculation. For instance, you may have evaluated all possible transportation methods for you to get to work and determined that you need to purchase a car. Fine, but which car you buy is another choice you make.

necessary:

Separate the wants from needs can help the user to get the real thing which they are needed. Based on this the students can improve the habit of the expense of college student, the Pocket Manager can tell the student what kind of expense are useful and necessary, then the user can manage their expense more reasonable.

feature description :

With this feature, the user can search an items through the eBay Web Service, then the photo, price and details about items will be returned. The user can analyses if this item is needed or just wanted based on the information that the feature displayed. It can help the user use their money more Reasonable. It will reduce a lot of unnecessary wastes.

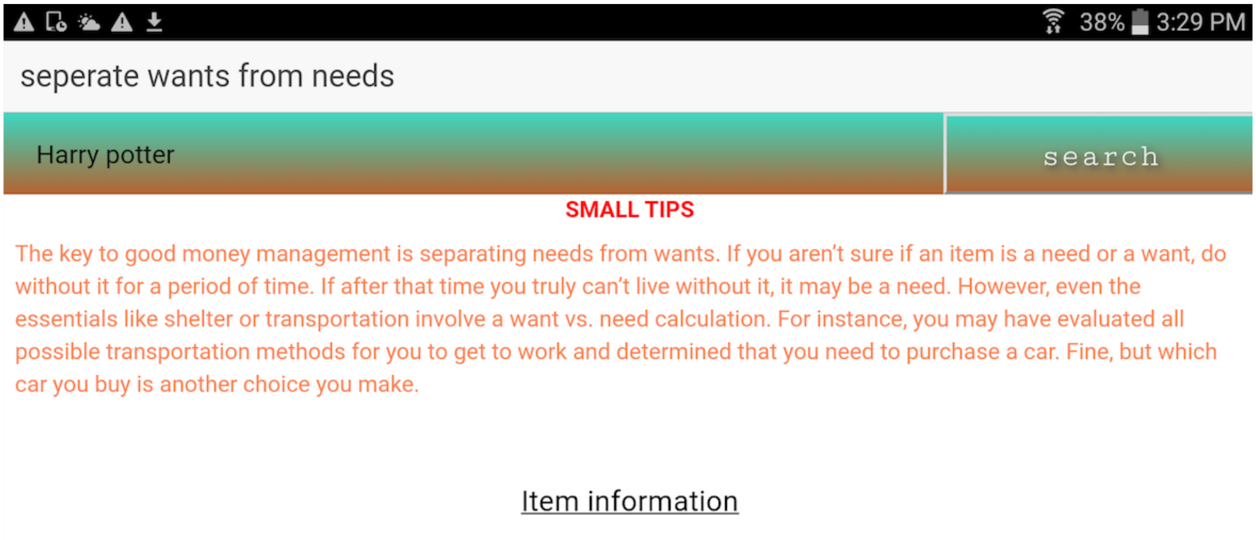
problem introduction:

during designing and implementing this part of system, I got confused by a problem, how to get a specific item information once user input a rough item name. At the very beginning, i chose to use wiki api, then I can have the full description of every items, but next step, I realized that, we can't have the most probable price of the items, then I turn to use Amazon Web Service, I have to confess that I didn't figure out how to call these api in my application. Finally, I used eBay web service to get the price and the item information and another useful thing is the direct items pages. That's the problem what I faced during implement the application.

instruction:

1. search item information — know what you want

   type in the full name of the items in the search box, and click search button, then the item information will be showed on the screen, including: full specific item name; price shows on the eBay website; and the direct time link to the eBay page.

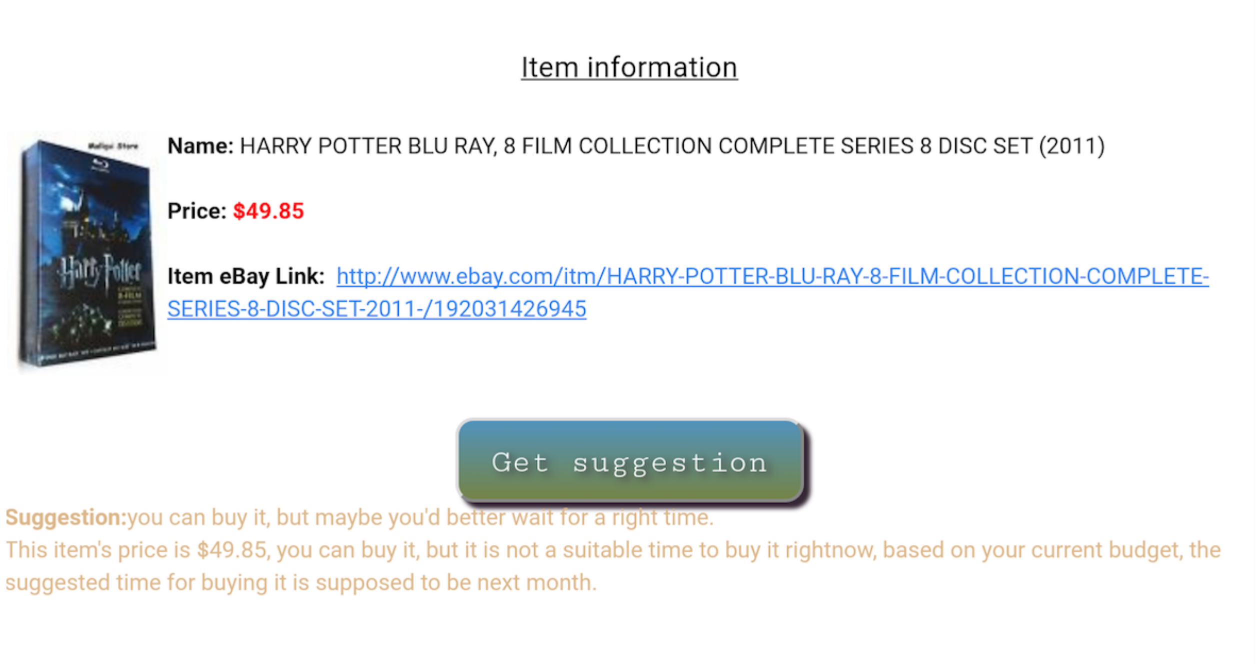


The key to good money management is separating needs from wants. If you aren't sure if an item is a need or a want, do without it for a period of time. If after that time you truly can't live without it, it may be a need. However, even the essentials like shelter or transportation involve a want vs. need calculation. For instance, you may have evaluated all possible transportation methods for you to get to work and determined that you need to purchase a car. Fine, but which car you buy is another choice you make.

Item information

2. get the suggestion — get small tips

the system will generate three suggestions for items you wants, for each item you searched, the system will provide a suggestion, and a corresponding reason why we provide you this suggestion to make user have a clear mind of wether buying it or not.



Item information

Name: HARRY POTTER BLU RAY, 8 FILM COLLECTION COMPLETE SERIES 8 DISC SET (2011)

Price: **$49.85**

Item eBay Link: http://www.ebay.com/itm/HARRY-POTTER-BLU-RAY-8-FILM-COLLECTION-COMPLETE-SERIES-8-DISC-SET-2011-/192031426945

Suggestion:you can buy it, but maybe you'd better wait for a right time.
This item's price is $49.85, you can buy it, but it is not a suitable time to buy it rightnow, based on your current budget, the suggested time for buying it is supposed to be next month.

- **How to Use the Expense Entry Feature**

1. Overview

Entering an expense brings the reality of actual spending into the user's desired budget. Furthermore, entering expenses naturally incentivizes reduced spending.

For expense recording, a user has two options including typing and photographing a receipt. The steps to accomplish both are detailed below.

2. How to Enter an Expense

a. Step 1 - As seen in **Figure 1**, a user can click on Record Expenses in the menu shown in on the left side of the screen.

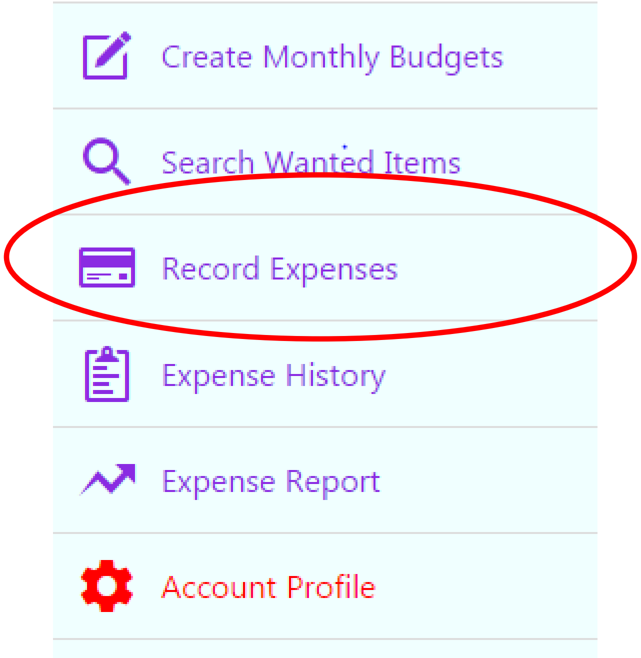
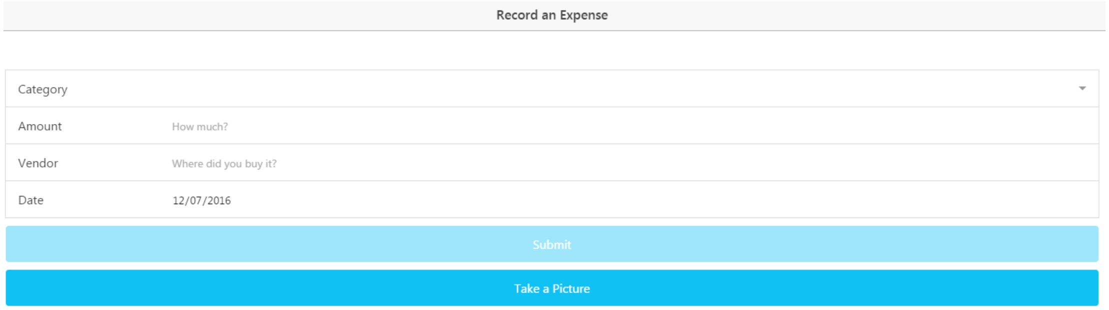


Figure 1. Application Menu

b. Step 2 – As seen in **Figure 2**, the user can type an expense manually selecting a category, typing the total spend, entering a vendor, selecting a date, and clicking submit. For added convenience, a user can take a photo of the receipt and have the information scanned automatically. When entering an amount, the user is protected from typos by being restricted from entering huge positive or negative numbers. Today’s date is also entered by default.

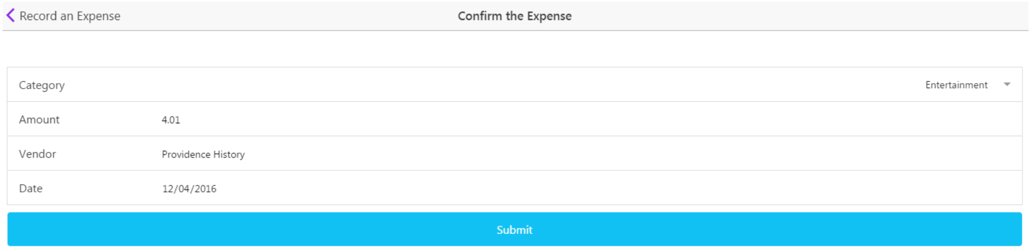


Record an Expense

Category	
Amount	How much?
Vendor	Where did you buy it?
Date	12/07/2016

Figure 2. Type Expense

c. Step 3 – As seen in **Figure 3**, the will next see a confirmation screen with the information typed or scanned from the receipt. The user gets a final chance to check the info before clicking submit to send the data to the online storage of expense history. If the user scanned a receipt, they should see information for all of the fields except category which they will still have to select manually.


Record an Expense
Confirm the Expense

Category	Entertainment
Amount	4.01
Vendor	Providence History
Date	12/04/2016

Figure 3. Confirm the Expense

d. Step 4 – Give yourself a high five for entering an expense. As seen in **Figure 4**, you can choose to enter another expense or go back to the home screen.

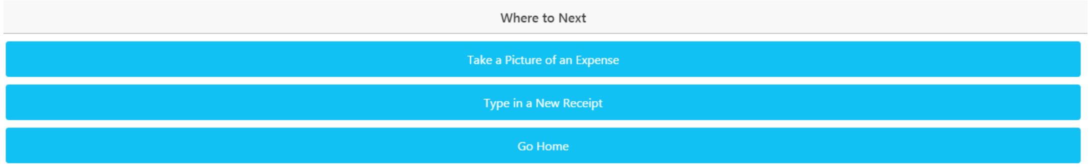

Where to Next

Figure 4. Where to Next?

- **Sample Interaction**

1. Overview

Below is a sample set of steps by the user for entering an expense along with what the user would see in the application from the system’s response.

2. Example of Entering an Expense

   a. Step 1 - As seen in **Figure 1**, a user opens the left panel and clicks Record Expense

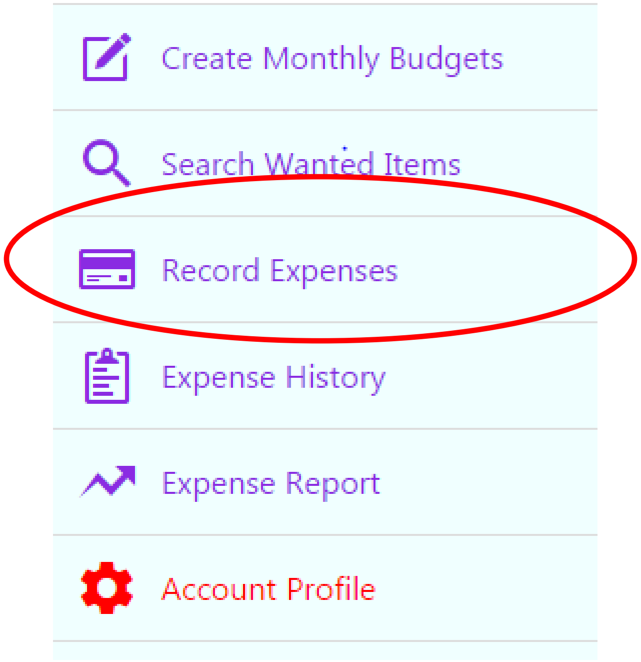
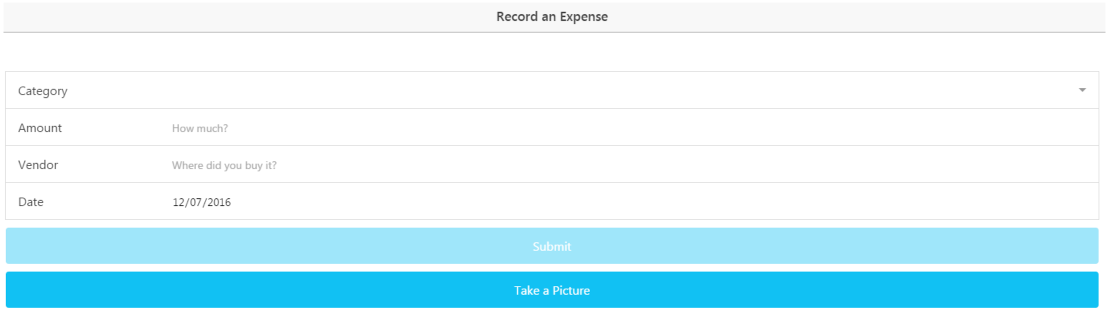


Figure 1. Application Menu

b. Step 2 – As seen in **Figure 2**, the user can type or photograph their expense. If taking a photo, the application takes the first line of text and sets it as the Vendor along with removing any "Welcome to" statements. For the date, the application looks for a set of numbers which have "/" or "-" separating them in a standard date format. Finally, the total is smartly found on the receipt using a proximity



Record an Expense

Category

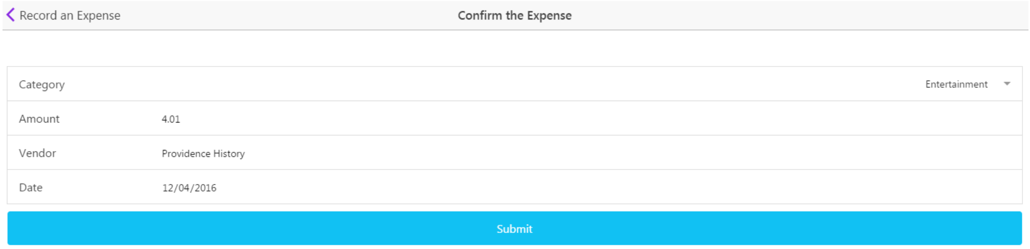
Amount How much?

Vendor Where did you buy it?

Date 12/07/2016

Figure 2. Type Expense

c. Step 3 – As seen in **Figure 3**, the user has a chance to see their data scanned or typed and corrected errors before a final submission. The scanning feature does not select a category for the user so they will have to select a category before the final submission. If the photo quality is poor, the user may have to correct the text before submission.



Category	Entertainment
Amount	4.01
Vendor	Providence History
Date	12/04/2016

Figure 3. Confirm the Expense

d. Step 4 –. As seen in **Figure 4**, the user can enter another expense or go home.

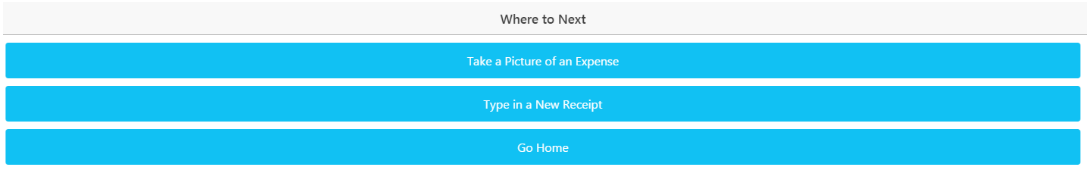


Figure 4. Where to Next?

- **Error Handling**

3. Entering Expense Errors

The expense entry portion has a number of features to help a user handle errors. As seen in **Table 1**, a list is below with the corresponding error correction.

	Error	**System Response**	**Error Code #**
1.	Incomplete Form	System Blocks Submission for an incomplete form. When a user touches a field and does not enter info, a warning is displayed below the field	
2.	Unrealistic Dates and Years	Years longer than 1 year before the current date and 3 years after are not	

		allows as input on the form. Amounts over +/- $10,000 are also blocked	
3.	Unavailable Session Storage	System checks for system storage and throws an Error #1 message. If seen, the user may need to update their operating system.	1
4.	MongoDB Connection Error	System throws Error #2. Warning about the connection error. Typically getting internet access resolves the issue	2
5.	Google Vision Connection Error	System throws Error #3. Warning about the connection error. Typically getting internet access resolves the issue	3
6.	Write File Error	System throws Error #4. Typically granting file write permissions on the phone security prompt will fix the issue	4
7.	Camera Error	System throws Error #5. If a camera exists on the phone, granting camera access on the security prompt should fix the error	5
8.	Inaccurate Scanned Text	Low photo quality can cause the text read by Google Vision to appear scrambled on the confirmation screen. The user can either correct the info manually or take another picture with better lighting and a smoothed out receipt	

Table 1. Error Handling

- **Known Issues**

4. Entering Expense Known Issues

The expense entry portion has several known issues. With the limited development time and lack of experience coding Machine Learning Algorithms, the software has several limitations on what it can accomplish. As seen in **Table 2**, a list is below of the known limitations and their potential solutions.

	Limitation	**Possible Solution**
1.	Category Not Selected Automatically	After building a large collection of expense data,

	by Receipt Scan	the information could be used to find correlations between Vendor Names and selected categories which could be used to predict future categories
2.	Vendor Name Must be in Receipts First Line	The software looks for the vendor in the first line of a receipt while also removing a welcome statement. Once a large dataset has been created, receipt names placed elsewhere could be found by referencing a large list of known vendors in the collection.
3.	Dates written as words not detected	Software only detects numerical dates delimited by “/” or “-“. Adding code to find the text of a month, the dates could be detected.
4.	Poor Quality Photo Returns Incoherent Text	Although the code can clean up some text errors, a wrinkled or low light receipt returns text with many inaccuracies. Using Machine Learning with a more robust language such as Java or C++ the software could learn from user responses to the error and predict the correct text despite the error
5.	Internet Connection Warning Incomplete	The receipt scanning was moved from being a controller to a service. Afterwards, a lack of internet connection does not throw a proper error. A solution is to check for internet connectivity and warn the user before a photo is taken

Table 2. Known Issues