CIS400/401 Final Report: TrackIt
A Cash Tracking and Budgeting Application
Dept. of CIS - Senior Design 2014 - 2015 *

Shaun Ayrton and Raul Zablah
{sayrton, razablah} @wharton.upenn.edu
University of Pennsylvania
Philadelphia, PA

Abstract

Personal financial management is a critical step to help alleviate poverty in developing countries. Unfortunately, there are few applications currently available that cater to this need in a specialized manner. TrackIt is a mobile application that integrates speech-to-text and natural language processing technologies to help users track cash transactions in an easy, fast, and unobtrusive manner by providing accurate offline speech to text and seamless functionality in offline environments. TrackIt is the first application that uses these technologies to solve this market need. The application itself adopts a minimalist interface that is both simple to use and very efficient. The major technical challenges including accurate offline speech to text recognition, seamless synchronization in online/offline environments and minimizing device power consumption are overcome through design choices in the system architecture as well as the unique choice of tools and APIs used. According to user studies undertaken in the development and evaluation of the application, TrackIt exceeds the stringent technical and user adoption criteria that were initially laid out. The application can be further improved in three ways: integrating financial budgeting and investment platforms, deploying on the Windows phone platform and increasing customization. While there is potential to further improve the application, TrackIt currently has the potential to revolutionize financial tracking and literacy in the developing world.

1 INTRODUCTION

In many developing countries in the world today, the majority of the population falls in the low to middle income segment. Research studies into the spending habits of this segment of the population have revealed that cash is used as the predominant tender for transactions and that this segment does not have access to banking services[12]. As a result, there is a lack of basic financial tracking of expenditures and budgeting, which prevents this segment of the population from raising their socio-economic situation. Fortunately, the decreasing price point of Android smart phones has led to a significant growth in sales and adoption within this segment of the population. Currently, there are few incumbents that cater to this need for financial tracking and literacy in a unique way.

TrackIt is a mobile application that integrates speech-to-text and natural language processing technologies to help users track cash transactions in an easy, fast, and unobtrusive manner by providing accurate offline speech to text and seamless functionality in offline environments. TrackIt is the first application that uses these technologies to solve this market need. The application itself adopts a minimalist interface that is both simple to use and very efficient. The major technical challenges including accurate offline speech to text recognition, seamless synchronization in online/offline environments and minimizing device power consumption are overcome through design choices in the system architecture as well as the unique choice of tools and APIs used. According to user studies undertaken in the development and evaluation of the application, TrackIt exceeds the stringent technical and user adoption criteria that were initially laid out. The application can be further improved in three ways: integrating financial budgeting and investment platforms, deploying on the Windows phone platform and increasing customization. While there is potential to further improve the application, TrackIt currently has the potential to revolutionize financial tracking and literacy in the developing world.

2 BACKGROUND

The vast majority of the world’s population falls in the low to middle income segment in developing countries. Countries in South America, Africa and Asia are experiencing rapid annual growth, both in terms of GDP and population. Within this segment of the population, the vast majority of daily transactions are solely cash-based. In fact, a recent study has shown that only 7% of the population in developing countries has credit cards and only 41% of the population utilizes any form of banking service [12]. While research has shown that there are a multitude of reasons for the lack of adoption of banking services, the primary reasons are the fundamental lack of trust towards banking institutions and the vast prevalence of the ‘shadow economies’, also known as informal economies[15]. Therefore, despite contrary belief, the number and magnitude of cash-based transactions have been steadily increasing.
Research has shown that an aversion for banking services is often linked with limited educational background [16]. As a result, families in this socio-economic category in developing countries are unlikely to have any current processes for budgeting. In fact, there is almost no evidence of even basic tracking of monetary inflows and outflows within each family. This lack of visibility into the timing and magnitude of disbursement of family income renders this segment of the population unable to conduct even elementary cash planning [13]. This reveals a significant need in low-middle income families in the developing markets to understand cash inflows and outflows in order to effectively budget for the future.

While a tangible need above has been identified, the question remains about how to effectively cater to this audience, given their natural inclination to use physical cash. Fortunately, there has been a recent boom in smartphone sales in emerging markets over the last 18 months. According to a report published by IDC in March 2014, total smartphone sales will reach 1.2 billion units before the end of the year, a 23% increase over 2013 [14]. Unsurprisingly, the vast majority of this growth can be attributed to increased sales in emerging markets, predominantly due to the decreasing price point, increased functionality and increased proliferation of 3G/4G networks.

The widespread nature of the need and the opportunistic adoption of smartphones within this segment of the market presents a unique opportunity to develop and use current mobile technology to empower the low-middle income families in emerging markets in order to effectively track and manage their spending habits, and effectively budget for the future.

3 RELATED WORK

Personal finances are defined as all financial decisions and activities of an individual, which include budgeting, insurance, savings, investing, debt servicing, mortgages, and more [1]. Before digital technologies were as widespread as they are today, personal finances were tracked through paper-and-pencil techniques. Regardless of whether it was one single individual keeping track of the home expenditures or a company keeping track of all accounting practices, it was necessary to write in paper the transactions. All updating, correcting, or removing of entries had to also be carried out manually, making this necessary task extremely impractical and inefficient. With the development of digital technologies, there was a transition to digitizing all this information. As a direct consequence, tasks were easier to perform and with accessible data to work with, it became feasible for anyone with access to a computer to keep track and understand their personal finances.

The technological developments led to the diversification and specialization of tasks that tools and applications use for personal finances management. The existing technologies regarding personal finances and budgeting can be broken down into five major groups. Each group is largely characterized by the core functionality the application or tool provides to the user, usually highly correlated to the type of personal activity or tracking procedures the user of the application is most likely to utilize. The following are the five major subcategories [8], [7], [2]:

**Bank Applications** In-house developed applications for financial institutions. These require the user to have an account with the financial institution to have access to the application and its functionality.

**Loan Calculators** Applications that facilitate the finding, usage, and payment of leverage products. These do not require bank accounts, but are subject to leverage-only products.

**Budgeting Applications** Applications that help the user keep track of incomes and expenditures, as well as other budgeting activities. These do not require the user to have an account with a financial institution, just an account for the application.

**Financial Aggregators** Aggregates user’s accounts from multiple financial institutions and other income/expenditure accounts, providing a single portal through which the user can handle the accounts. These inherently require the user to hold a bank account to have access to the application and its functionality.

**Other Financial Applications** These applications are of miscellaneous nature, but the predominant one is peer-to-peer payment applications. These applications require the user to have an account with a financial institution.

Given the selected market and the problem at hand, the focus of the Related Work is predominantly on Budgeting Applications. It is important to place emphasis on the inherent difference between the Budgeting Applications and the Financial Aggregators and Banking Applications. The latter two act as a portal for the user’s already existing accounts of financial institutions of any kind. The application itself is useless unless there is an account to link and carry out the operations on. Budgeting applications on the other hand, do not need an existing account of any sort and the application itself is useful even if the user does not have access to some other financial institution [10].

After careful review of the existing Budgeting Applications, there are two resulting conclusions. First, the state-of-the-art has very few incumbents whose purpose is to track cash expenditures. The great majority of the existing applications focus on providing users estimations of

---

1 see Appendix A for a more detailed description of each
expenditures and income flows, in order to make further estimations on how much money can be spent in the future. The largest shortcoming of these applications is that their core functionality is not the tracking of the expenditures, but rather estimating future purchasing abilities, given rough estimations of income and spending habits. The second conclusion is that there is currently no other application or tool that integrates the speech-to-text technology into its functionality. All of the existing budgeting applications rely on the user manually inputting all the income/expenditure information. This becomes a serious limitation for the incumbents, because it becomes highly inefficient and obtrusive to use the application as a tracking device.

Due to the lack of incumbents in the field that integrate the speech-to-text technology into the budgeting applications, the product will have to successfully integrate the two technologies in order to make a contribution to the field. Most of the existing budgeting applications also do not provide a user-friendly and intuitive interface, so it will be of extreme importance for the proposed project to provide these to the users in order to exceed the existing performance levels within the field.

4 SYSTEM MODEL

In order to meet the market need described previously, it is essential that cash tracking and budgeting is easy, fast, and unobtrusive. This can be effectively achieved through the combination of the following technical features: speech-to-text functionality, natural language processing, data aggregation, and user-intuitive interfaces. The value proposition of the project lies in the intersection of these technical features. The application allows users to speak into the phone stating the total of the expenditure and the item to which the expenditure is related to, once it is carried out. Due to the lack of incumbents who do this, and research that shows that speech systems provide a 20% to 40% efficiency increase compared with keyboard input, the product is effectively creating value for the users [3].

The next step involves the natural language processing of the resulting text of the user’s speech input and then aggregating the data according to the type of expenditure carried out. Lastly, through the user-friendly and intuitive interface, it is possible for users of the application to visualize their expenditures, identify trends, and set up notifications to adopt new expenditure patterns. The benefits of the application to users are two-fold. One, the application provides an intuitive manner of keeping track of cash expenditures - without having to revert to the tedious and inefficient process of manually inputting data. Two, by aggregating the expenditure data that otherwise goes untracked or just estimated, users can see, understand, and evaluate their spending habits in order to improve them.

5 SYSTEM IMPLEMENTATION

5.1 System Architecture

The development of the application was divided into three distinct stages, each with its own objective. The attainment of objectives is detailed in the Evaluation section later in the paper. Separating the development into these specific stages facilitated a modular and test driven development strategy. The system architecture (Figure 3) provides a graphical representation of the developmental stages and should be used as a visual guide when reading the paragraphs below:

The first stage of development was recording and interpreting the ‘voice input’. The objective of this stage was to convert the speech to an accurate text representation in both offline and online settings. Most current speech to text and NLP engines require online connectivity where the input is sent across multiple servers on which natural language processing techniques are applied. However, given that the target population will not have continuous connectivity, it is essential that this step worked effectively offline. In addition to offline functionality, the most important design objective of this stage was an accuracy. Based on initial survey tests, anything above 91% accuracy rates was considered acceptable by users. Most users understand that speech to text is a nascent technology, which explains the threshold. Internally, we set the accuracy threshold to 95% as a conservative objective. This means that out of every 100 speech inputs, 95 of them would be converted accurately into the appropriate integer. After substantial research into the currently available speech to text technology, we found an engine that runs locally on the Android device itself and thus does not require online connectivity. The particular speech to text engine that was used is detailed in the Technical Specification section of the paper.
The second stage of development is a broad category that contains the entire processing of the numerical input. The complexity in this stage arises from the fact that the application could be online or offline at any given moment, but this should not compromise the user experience. As a result, this stage can be divided into two distinct parts. The first part handles the storage of the input and the second part handles the data that is ‘pulled’ and displayed to the user on the Profile page.

The implementation of the first part of the processing stage is reliant on whether the device is connected to the internet. If the device is online, the input is directly sent to the cloud database and the user’s profile is updated. There are two important aspects to note here. First, by storing the input directly in the database, the application is space efficient as there is not trace of the input left on the device. Second, if any adverse event occurs and the user is unable to use his device, his/her information can easily be retrieved by downloading the application and logging onto his user profile in the new device.

If the original device is offline at the time of input, the text is stored in a local ‘Speech to Text Cache’. As the device only stores strings, this is very space efficient by design. This size of the cache is designed to be flexible in order to accommodate high frequency usage when the device is offline. When the device regains internet connectivity, the data that is temporarily stored in the ‘Speech to Text Cache’ is sent to the cloud based database and subsequently removed from the cache.

The second part of the processing stage is less dependent on whether the device is offline or online. It is important to note that whenever the device was last online, it stores a copy of the most recent entries within that month in a second ‘local cache’ on the device. This is not a busy looping protocol. Instead a callback is issued both when the device regains connectivity and when a new speech input is received and added to the database. When the user navigates to the profile page, the ‘aggregator’ orders the data stored in the ”speech to text cache” and the ‘local cache’ to provide the most recent entries of the current month. The sorting algorithm used in the aggregator is simple and computationally efficient as data in the speech to text cache must automatically be more recent compared to the data in the ‘local cache’ by design.

The primary objective when designing the processing stage was efficiency. User group interviews prior to the development of the application revealed that if the entire processing took over three seconds, the delay would be perceivable and users would stop using the application. As a result, the internal target processing time for our application was under two seconds. The application performance against this target is detailed in the Results section.

The third and final step in the system architecture is the visualization. In particular, this stage refers to the profile page. The two primary objectives in this stage were seamless functionality and to develop an intuitive user experience. Seamless functionality refers to the fact that the distinction between offline and online functionality should be abstracted away from the user experience and indistinguishable to the user. User group interviews in the target market population during the initial phases of development revealed that a graphical representation of the data was essential for ease of comprehension and to obtain fast insights. As a result, the profile page offers both a graphical and tabular representation of the data including the granularity of different budget categories (food, rent, bills).

5.2 Application Functionality
The following is a detailed walk-through of how the user interacts with the application and what the user should expect from it.

Figure 2: Application Homepage

When the application is first opened, it opens up to the Home Screen (Figure 1). In this screen, the user will first see a message in the center of the screen that asks the user for a specific way to provide the speech input.
Underneath this message, the user will see three buttons, each with a broad category of expenditure. These buttons are titled: Food, Rent and Bills. It is important to note that these categories are not intended to be exhaustive. The user would be able to select and label up to ten customized categories in the next version release of the application. In order for a user to record a message, the user will press on the appropriate button to start the recording and while keeping the button pressed, speak into the phone. After finishing the speech message, the user will release the button to end the recording. Upon completion of the recording, the offline speech to text recognition engine will automatically produce and store the numerical input as text. In the case of invalid or misunderstood input, a pop-up will display asking the user to re-enter speech input. The user will also have the option to manually enter numbers in the case of misinterpretation to preserve accuracy of data.

In the right corner of the screen there is another button that says Menu. If the user presses the Menu button, a sliding menu will show on the left side of the screen. This menu will have three options to choose from. The first option is Home which takes you to (Figure 1). The second option is Profile (Figure 2) - if the user presses on this option then the application redirects the user to a new screen. The third option is that of a User Manual - if the user presses this, then a pop-up appears, and the user can scroll through it to find more information about the application and how it works. The fourth option is a Log Out - enabling the user to log out of their account.

If the user presses Profile then the redirected screen is a screen where the user’s expenditure history is shown. The data is shown in both a tabular and graphical format. The tabular format will show the user the five most recent expenditures, while the graph will display each expenditure and an aggregation for each category. This provides the user an easy way to monitor recent expenditures across budget categories in an intuitive manner. It is important to note that the data is always accurate and up to date, even if the device is offline or online. In the next version release of the application, the user will be able to modify the parameters of the graphical display including time, size of transactions, frequency of transactions, and type of transactions.

The user can easily go back to the Home Screen by pressing the Home Screen button.

Overall the design of the application and user interface is based on simplicity: easy to understand, minimum number of clicks and minimum number of screens. The reason driving this design choice is to cater more effectively to the population being targeted, since it is imperative to keep the application as simple as possible for the users to have a seamless and unobtrusive experience.

5.3 Technical Specifications

This section highlights the frameworks, tools and APIs that were used in the development of TrackIt:

- Development environment: native Android (Lollipop) development for mobile platforms.
- Speech-to-text: Google Now online/offline speech recognition API (Speech Recognizer class)
- Local Storage: Android Developer Guides
- Database: Parse Android SDK
- Graphical API: Android Graph View

5.4 Technical Challenges

As the first application to integrate offline speech to text functionality and seamless offline online functionality into a financial literacy application, the development of TrackIt was filled with technical challenges. Below are the key challenges that were faced and how they were systematically overcome to produce a successful end product:
• **Offline speech-to-text functionality**: This was one of the biggest challenges of the project, both in time and effort. Due to the ‘infant’ state of the offline speech-to-text technology, the best accuracy rates for complex inputs ranged between 30% and 40%.[11] As discovered in our user surveys, this was significantly below the acceptable threshold of 91%. To overcome this challenge, the user interface was designed in a manner that restricted the complexity of the input to simple whole number numerical inputs. In the Application Homepage (Figure 1), the user simply presses on the relevant category and says the amount that he/she spent. Through this design, most of the complex linguistic aspects of speech to text processing was removed from the input and the chosen engine simply needed to process numerical inputs. This design boosted accuracy rates and overcame the hurdle of accurate offline speech to text.

• **Seamless online and offline functionality and synchronization**: Due to the offline/online nature of the application, it was essential to develop customized protocol for synchronization of data. After significant research, the Parse Android SDK provided the optimal mix of basic synchronization and database access. While the Parse SDK enabled the use of a compatible database, the design of the entire application needed to provide a seamless user experience, while ensuring efficient processing. This led to the development of the Processing stage (Figure 3) that is detailed in the Application Functionality section. The two cache design of the system allowed the development of the application in a manner that satisfied both objectives.

• **Device power consumption**: Due to the current battery technology on mobile platforms, battery life is a significant issue. Therefore, the application must actively minimize its use of battery life. Minimizing local power consumption was achieved by using a system of callbacks for updating caches instead of ‘busy looping’ within the actually development of the application.

• **User-friendly interface**: Developing an intuitive, unobtrusive, and easy to use interface is one of the biggest shortcomings of the existing budgeting applications. It was necessary to first conduct extensive due diligence in order to understand what are the key aspects of the application that users were interested the most in. This was achieved through the use of surveys to the target audience including conjoint analysis techniques where users choose their favorite design out of a collection of similar designs with minor yet instrumental differences. This enabled us to identify that a minimalist interface scored both the highest in aesthetic appeal and comprehension of data. Additionally, the findings also revealed that an interface schema provides the minimal number of clicks and/or swipes for the user would be essential. More details about the actual design and APIs can be found in the System Architecture and Technical Specifications sections.

Despite the numerous technical challenges, TrackIt offers an essential service to the target demographic. The integration of the speech-to-text technology with seamless offline and online functionality provides a more user-friendly and intuitive way to keep track of cash expenditures. As described in the introduction, this is especially relevant for our target market, in which most cash expenditures are untracked and financial literacy rates are very low. The aggregation of individual cash flow data provides insight into user spending behaviors and allows for future versions to provide recommendations to reduce wasteful spending habits of users. In the long run, TrackIt has the potential to increase the low levels of financial education among individuals in developing, cash-based economies and thus help alleviate poverty.

6 RESULTS

With the detailed implementation of the application outlined above, it is essential to evaluate how TrackIt performs against the target objectives. There are two distinct categories of evaluation that empirically characterize the performance of the application. The first category refers to technical performance with regard to the objectives defined in the System Implementation. The second category refers to the user interest and adoption criteria and goals.

6.1 Technical Evaluation

The three criteria that are tested here are directly related to the design goals in the System Architecture (Figure 3). This ensured that TrackIt was optimized and performed well under the criteria that users deemed acceptable:

• **Accuracy of Speech to Text**: As was established earlier in the paper, user surveys led to the finding that a minimum accuracy threshold of 91% was required for persistent use and acceptability. The internal design target was set to 95% to ensure that the user criteria was met. According to both Google [9] and Apple [11], the Google Now offline speech recognition rates have increased to 98% for basic numerical speech to text. In addition, we conducted our own user group interviews and found that under regular conditions, the Google Now speech to text engine was 98% accurate as well. To reach this statistic, the application was tested for 100 different basic numerical inputs from ten different users with different English accents. It is important to note that we only tested English as
this is the first version of the application. Other languages such as Spanish and Mandarin, which have slightly lower accuracy rates, are rapidly catching up to the same detection standards as English and are already integrated in Google Now’s API. The final point to note is that the accuracy may decrease in excessively loud environments (above 90 decibels) [5].

- **Processing time:** The major finding from user surveys prior to the development of the application revealed that the total processing needed to be less than three seconds. The internal design target was set to under two seconds to ensure that the user criteria was met. In order to assess this, the testing was divided into two phases. First, it was important to test the time taken from the completion of speech input to updating the database in online setting. This was tested by comparing the local time stamp on the device with the time stamp registered on Parse database. After over 100 inputs, the average time difference was 0.76 seconds which was well below the criteria. The second important aspect to test was the time taken to populate the Profile page (Figure 2) after a user just inputs new data, as this is the worst case performance in terms of processing time. The internal application API of the Android Lollipop allowed us to determine the time between the execution of instructions and found that the average time taken for this stage was 0.73 seconds, which was also considerably below the evaluation criteria. After extensive user tests, the application was commonly described as ‘immediate’ and working in ‘real-time’, which proved that the worst case processing time was imperceptible to the user. This is because it would take the user over 0.7 seconds to click on the Profile tab after entering the voice input.

- **Seamless offline/online integration:** This aspect of the application was achieved through the design of the system architecture that abstracted the processing stage away from the end user (Figure 3). As a result, the only effective way to test this is through qualitative responses of whether the user noticed any changes between the offline and online functionality. All of the 60 beta users responded to the post-trial survey that they were unable to detect any difference.

Overall, the results of first version of the TrackIt application were able to not only meet, but to exceed the technical design objectives that were created based on initial user survey results. While this was validation of a strong product, it was important to test post development user response.

### 6.2 User Evaluation

It is important to note that users were included throughout the development of the application from creating the relevant design criteria to the various stages of testing. There are two ways that we tested potential user demand after completion of the application, which provided invaluable insight in terms of the functionality improvement and future work.

- **Survey to university students:** Although students at the university of Pennsylvania were not our target population, we conducted a user demand test on them for three reasons. First, many students were on student loans and tight budgets so would be in a similar financial situation as our target market. Second, there is a tendency to spend cash on campus in Philadelphia due to the popularity of food-carts that only accept cash. The third reason is the ease of access to this population and the ability to rapidly test a large and diverse population. One hundred and fifty students responded to the survey and the results were resoundingly positive, with 60% choosing the option, ‘very likely to use TrackIt’. Additionally, there was an opportunity to best sign up for a demo and become a beta user during select times. Through this process, we were able to conduct demos with 20 students and gained feedback on the user interface such as integrating a navigation bar instead of a button. The current version of TrackIt has already integrated this feedback into the product.

- **Focus group interviews with target population:** While student interest was a good proxy of market demand for the reasons outlined above, it was essential to test demand within our core target population. Fifteen focus group interviews were conducted with people in India, El Salvador, Costa Rica and Zimbabwe. These interviews were conducted before and after the development of the application. The key findings before the development guided the design objectives described in the system architecture. The initial release of the application simply contained a tabular representation of the data. However, findings from these focus group interviews led to the inclusion of a line graph as the optimal visual way to interpret the results. This too has been integrated within the current version of the application (Figure 2).

Overall, the application far surpassed the initial technical and user adoption objectives in the initial proposal. Despite these promising results, it is important to note that the application is still in the first version of the developmental cycle. The following section will outline a unique ethical dilemma that the application poses.
The predominant motivation behind the project is a social one: to help alleviate poverty in low-middle income households in developing countries by enabling financial tracking and literacy. However, it is natural to consider the monetization and profit potential of an application during its development. It was in this situation where the team faced an ethical dilemma. Many large scale retailers such as Amazon who offer significant price discounts are unable to access markets in developing countries due to lack of brand awareness and lack of understanding of purchasing habits of the local populace. This directly overlaps with our target demographic of users. If our application becomes widespread within this population, there is a significant monetization opportunity by anonymizing transaction data and selling the information to these large retailers. Due to our social mission, the intention was to invest this money within these local communities to build socially and environmentally sustainable projects, infrastructure and local educational institutes. However, the observation was made that if a large retail company were to start selling to these micro communities, it would cause significant unemployment of the local tradesman and the dissolution of local markets. The team therefore believed that the costs of monetization through this path exceeded the potential benefits, so decided against pursuing this trajectory.

8 FUTURE WORK

While the initial results of TrackIt were promising, there are three fundamental improvements that would be essential to help the application scale and effectively cater to the needs of the target population:

- **Integrate Financial Investment Recommendations:** As the first application that integrates speech to text and seamless offline/online functionality to financial application, it effectively caters to the target demographic in a unique way. However, it is important to note that this is just the first step towards achieving financial literacy. The current version of TrackIt provides visibility into their spending habits. The next step would be to move from awareness to tangible improvements in their financial situation. The first improvement would be to incorporate interactive software that provides users suggestions for budgeting. After this, it would be beneficial to partner with micro-finance institutions and other other investment portals to create APIs that allow this population to effectively invest surplus income.

- **Deploy TrackIt on the Windows platform:** As our target population is low to middle income populations, it is important to deploy the application on all devices that are used by members of this population. There are numerous countries in Africa that meet our target demographic. The Microsoft windows phone sales are expected to exponentially grow in Africa over the next decade [4] so it would be best to preemptively launch TrackIt on that platform as well.

- **Increased functionality:** In subsequent versions of the application, it is important to add features such as a family sharing option with administrator privileges which would enable a parent to keep track of their children’s expenditure while teaching them strong financial budgeting practices. Moreover, it would be important to provide additional customization within the application to improve the user experience. This would include customizable budget categories.

9 CONCLUSION

TrackIt is a novel, user-friendly application that was designed with the goal to empower the low-income segment with a tool to track and manage their finances with ease. It is the first application to seamlessly combine speech-to-text technology with financial literacy. It is a mobile application that works online as well as offline and assists the user with keeping track of expenditures and budgeting accordingly. Potential users were consulted throughout the development phase from creating design criteria to final testing. Stringent technical evaluation criteria were employed and extensive demand testing among target users was conducted. The results of these interviews guided improvements within the application and were incorporated into the final product. Feedback from post-developmental surveys was resoundingly positive and indicated a strong market demand for the product. To propagate this success, three areas of potential improvement were highlighted in the future work section. TrackIt therefore has the potential to revolutionize financial tracking and literacy in the developing world by empowering users and thus helping to alleviate world poverty.

A Appendix A: Types of Applications

Bank Applications Financial institutions provide in-house developed applications that are linked to the users’ bank accounts. The usual core functionality of these products revolves around balance reports and inquiries, branch-locators and ATMs, accessing bills, and paying bills. The account balance reported is highly dependent on how often the data is updated and when transactions are cleared. The best applications in this subcategory are recognized by their intuitive budgeting interface that keeps the usage at
a very easy level for the user. Some of the best reviewed applications of this kind include: HSBC Mobile Banking, Bank of America Mobile Bank, TD Bank Mobile Banking, Scotiabank Mobile, Santander Personal Banking, Citizens Bank Mobile, and many others.

**Loan Calculators** The main functionality of the applications of this nature is to provide product matching, product diagnosis, payment schedules, and other calculations relevant to different types of loans, insurance, shop financing, and other related activities. It usually involves searching for quotes, calculating product-specific details - when given a set of variables, and comparing different products for the customer. Some of the best reviewed applications of this kind include: Simple Loan Calculator, Loan Calculator, Interest Calculator, Loan Calculation, and many others.

**Budgeting Applications** These applications are similar to the Bank Applications. The core functionality is to provide the user a way to keep track of income, expenses, and other budgeting activities on the mobile phone. Due to the potential of existing lag in the Bank Applications, the Budgeting Applications provide an advantage to the user - a user who is usually characterized by the diligent nature of keeping track of receipts and inputting information on the spot to keep his/her finances up-to-date. Some of the best reviewed applications of this kind include: Goodbudget - Expense & Budget, Expense IQ - Expense Manager, Spending Tracker, My Budget, MoneyWise, Wallet pig, Keeper - Budgets & Expenses, Crunch - Simple Budgets, Money Lover Expense Manager, Speed Budget - Expense Tracker, Budget, and many others.

**Financial Aggregators** The design of Financial Aggregator applications pulls the customers' personal information from several financial institutions and other income/expenditure accounts, using the customer’s username and password. All of this information is aggregated in the application and it is then utilized to act like a personal assistant - tracking and providing assistance with expenditures and other financial operations the user carries out. Some of the best reviewed applications of this kind include: Mint.com, BillGuard, LearnVest, Check, CheckBook, BUDGT, Spendee, Expense Manager, One Touch Expenser, Toshl Finance, Money by Jumsoft, and many more.

**Other Finance Applications** Applications within this subcategory are of miscellaneous nature, usually related to debt management and tracking, branch or customer service points for financial institutions, store credit and coupon tracking. Usually these applications are very specific to one finance-oriented customer need.

**References**

1. [Definition of personal finance.](http://www.investopedia.com/terms/p/personalfinance.asp)
9. Lauren Goode. Google voice search talks me through a house of cards weekend. [http://recode.net/2015/03/03/google-voice-search-talks-me-through-a-house-of-cards-weekend/](http://recode.net/2015/03/03/google-voice-search-talks-me-through-a-house-of-cards-weekend/).

The best applications per subcategory are selected based on number of downloads and user reviews in the Google Play store, since the scope of the project is within the Android environment.


