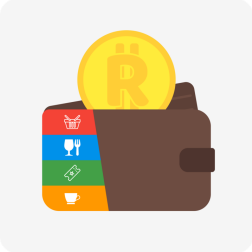

RewardWallet

School of Engineering Science | Burnaby, BC • V5A 1S6
https://github.com/RewardWallet

May 20, 2018
Dr. Andrew Rawicz
School of Engineering Science
Simon Fraser University
Burnaby, British Columbia
V5A 1S6

Re: ENSC 440 Design Optimizations for a Customizable Rewards Allocation System

Dear Dr. Rawicz:

Attached you will find our ENSC 440 Design Optimizations for a *Customizable Rewards Allocation System*. This project is designed to add reward allocation to any business by integrating their POS system into our cloud system. This easy and affordable solution is poised to give small Canadian businesses the tools needed to propel their businesses into the digital era.

These design optimizations primarily focus on how we will improve and/or optimize the original design proposed in the *Design Specification for a Customizable Rewards Allocation System* document. In addition, the engineering standards use in this project will be thoroughly examined.

RewardWallet is made up of three undergraduate engineering students: Molly Bin, Nathan Tannar, Mandy Xiao. Should you have any questions or concerns, please feel free to reach out to myself at (604) 355-6292 or ntannar@sfu.ca

Sincerely,

Nathan Tannar
President and CEO RewardWallet

Enclosure: *Design Optimizations for a Customizable Rewards Allocation Syste*

Design Optimizations for a Customizable Rewards Allocation System

Project Team:
Molly Bin
Nathan Tannar
Mandy Xiao

Contact Person:
Nathan Tannar
ntannar@sfu.ca

Submitted to:
Dr. Andrew Rawicz - ENSC 440
School of Engineering Science
Simon Fraser University

Issued date:
May 20, 2018

Revision:
1.0

Table of Contents

1. Introduction

In today's business world, companies are utilizing the latest technology to better service their customers and growth their businesses. Most franchises create their own mobile apps that allow their customers to collect rewards and/or redeem coupons. Collecting a reward for purchasing an item is another example of instant gratification and draws customers to revisit or prefer a particular business. However, this puts smaller businesses in a disadvantage because they do not have the resources to build their own reward system. Our customizable product is the system that small businesses need. By providing a low-cost solution, businesses can be empowered with a system that is customizable to their needs and deliver instant gratification to their customers.

"All good businesses are personal. The best businesses are very personal. " - Mark Cuban

RewardWallet is a system that keeps track of transactions users make at a business to allocate rewards to a customer's profile based on how the business customizes their distribution model. Each customer will have a unique digital card for each business that stores reward points. By creating the hardware needed by a business, a mobile app for the user and a cloud system to bring it all together, RewardWallet gives business owners an end-to-end solution for adding a customizable digital loyalty system to their store. The design specification for RewardWallet describes how the systems requirements will be fulfilled through technical design. This document is an extension of the Design Specifications that addresses how areas of the design can be optimized or improved.

2. Primary Design Improvements

2.1 NFC Transaction Terminal Microcontroller Circuit Optimizations

In the 405 implementation of the NFC Transaction Terminal, the RewardWallet Engineering team utilized the Raspberry Pi 3 as the bridge between the Cloud Server and iOS Application. The Raspberry Pi 3 was chosen due to its ability to perform network requests using the onboard BCM43438 wireless LAN [15]. Then, the PN532 NFC Controller Breakout Board was connected to the Raspberry Pi 3 for added NFC technology.

Optimizations of the aforementioned components not only eliminates the over-engineering features, but also significantly shrink the size of the NFC Transaction Terminal Microcontroller by 70%. This optimized version will consist of Raspberry Pi Zero W, which is the bare minimum device needed for onboard wireless LAN, and the RF430CL330H Dynamic Dual Interface NFC Transponder. This is a core NFC chip that only supports NFC Tag Type 4. Instead of using the predefined circuits for the wire SPI and I2C interface in the PN532, the RF430CL330H is used to program the necessary interface for the RewardWallet system.

The primary optimization for the NFC Transaction Terminal is defined to be the change of microcontroller from Raspberry Pi 3 to Zero W. Python will be used to program the necessary network communications between subsystems. Further optimization is the change from PN532 NFC Controller Breakout Board to RF430CL330H Dynamic Dual Interface NFC Transponder. C and Python will be used to program the interface, encoding and decoding with the interacting iOS devices.

2.2 QR Code Based Transactions

Instead of only using NFC Transaction, the RewardWallet Engineering team introduced QR Code, which is going to both speed up the e-commerce transactions and make it more secure. For those customers who does not have a NFC function in their smartphone, QR code is another way to enable reward transactions. Having a mobile app and using QR codes with it can place a positive first impression of our business in the consumer's mind. The purpose of this optimization is to allow consumers to collect and redeem rewards through a QR code.

The primary optimization for the QR code based transactions is defined to be the change of the API. The optimized version will contain consumer-presented QR codes, which generates by the RewardWallet mobile app. Customer then uses the transaction-specific QR code to collect and redeem rewards in store. Besides offering loyalty points, this method also can capture additional detail about each purchase.

2.3 Merchant Control App Usability

In the ENSC 405W proof of concept design, RewardWallet only implemented digital wallet iOS app for customers to collect and redeem rewards. By developing a merchant control app, RewardWallet allows local business to configure businesses reward distribution model and perform a better customer service. The benefits of using an iPad merchant control app are the cost savings and ease of employee training. RewardWallet dedicates to provide a platform that can help businesses improve customer retention and promote loyalty.

The design consists various features including scanning QR code presented by digital wallet iOS app, rewarding and redeeming points for customers, and sending requests to the transaction terminal. Merchants can edit and publish their business's profile to the digital wallet iOS app so customers can find them as they explore a given city. The dashboard in the iPad app provides a glance of basic sales information and recent transaction history, including the number of points allocated to the customer, date, and payment amount. By implementing the feature of sending notifications to existing and potential customers, it lets merchants attract new customers, works as an amazing loyalty program, has incredible target marketing, and includes a big social media presence. Additionally, the app will allow businesses to configure their own reward distribution model. The app must follow the engineering standards to prevent any safety

and risks issues related to data losing and app crashing. The risks and safety issues introduced by the changes in the iPad app design will be handled by the development team before releasing the merchant control app.

2.3 Digital Wallet App Usability

In the proof of concept design created in ENSC 405W the iOS app was only able to collect reward points with some predefined setup. The main focus of that design was to get the iPhone's NFC chip communicating with the Raspberry Pi. This left for a design with very little usability and serviceability for a product launch. Our plan is to deliver a robust MVP (minimum viable product) of the app that will give early adopters of RewardWallet all of the features they need to help businesses enrolled into RewardWallet thrive.

To improve the user experience and functionality of the RewardWallet app we will continue to develop the codebase. By adding map views and search functionality users can discover new businesses that reward customers with RewardWallet points. Each business will have a page which they can customize to include relevant business information such as opening hours, contact information, photos and more. All if this information will be stored and send from the cloud server. These enhancements target getting more customers into store to support the business which in turn, supports RewardWallet. Additionally, app will also feature improvements to collecting or redeeming points. The user interface design of the app will be improved to enforce built in learnability, a key concept learned in CMPT 275: Requirements Engineering. Also, an option to collect points with a QR code will be added thus to eliminate the problem of some users phones not having NFC support. To address risk management analytics will be added to the app so our development team can be informed if the app crashes, so they can fix the problem to improve the safety of the app. We will continue to uphold Engineering Standard E1 (found in Appendix I) as it improves usability if designed in a way users are currently used to; and Engineering Standard E4 to ensure consistency across the codebase.

3. Secondary Design Improvements

3.1 Improve Backend API Cloud System

In ENSC 405W initially designed our backend system to be very versatile such that we could easily expand its functionality. We focused primarily on security and the design of the database for storing the required data models. These security designs were to ensure compliance with the IEEE 802.15.5 standards for Telecommunications and information exchange between systems [13]. While these designs were completed they are missing end-to-end testing to ensure reliability in the system. Tests need to be designed for each use case. Because each subsystem, mobile apps and microcontroller, relies on the cloud system for data storage and retrieval the cloud system is a single point of failure. By developing test cases we can reduce the risk associated with the single point of failure and help to ensure that any future changes to

the system do not compromise a previous functionality. In addition to developing tests, due to the other primary design improvements, new functionality on the cloud system will need to be added to enable such features. These changes to the system will continue to fall in-line with Google's Cloud Platform Engineering Standards [12]. As a whole, these changes will continue to reduce risk in the system, make it more serviceable for future changes and uphold Google's leading Engineering Standards.

3.2 NFC Transaction Terminal Microcontroller Casing

A smaller design improvement we will be making is coming up with casing for the microcontroller. In our first iteration there was no casing, leaving the electrical components exposed. This was not safe for the user or the microcontroller as damage would be inevitable. Another design flaw we noticed while demoing the device was that users were unsure of where to tap their phone. For these reasons we will be looking to create a 3D printed casing that will safely enclose the microcontroller while featuring prominent RewardWallet branding that helps to indicate where a user should tap their phone. This will reduce risk of potential damage to the device and improve usability.

4. Conclusion

The three main subsystems of the RewardWallet system are the Embedded System, Cloud Server and Mobile Application. Each of these subsystems are comprised of their component-specific design specifications. Designs of these components are optimized for better safety, more compact in size and efficient.

In the Primary Design Specifications, engineers specified the technicalities in the design optimization of each subsystem. The Embedded System is equipped with the capabilities to perform network requests for initiating transaction and transmitting records through NFC. The NFC microcontroller unit is optimized to eliminate over-engineering, thus resulting in smaller size and increase efficiency. Adding the QR code based transactions will enable users to have an alternative to the NFC technology. The Mobile Application is developed in XCode IDE using Swift 4.0 in order to provide users with the best and newest mobile application technology. By implementing a map view and other detailed designs to the Mobile Application will better the usability of the application. Lastly, the Merchant Control Application enables improve usability on the business endpoints. Business owners are able to collect and analyze statistical information in order to create tailored distribution models for their businesses.

The Secondary Design Specifications focuses on improvement in testing of the backend. Methods of End-to-End testing and fault safety procedures will take place to ensure maximal safety of the system. Furthermore, microcontroller casing is designed to protect electrical components from environmental damage. In conclusion, the design of the RewardWallet system closely follows all related Engineering Standards. The RewardWallet team is confident in optimizing the products system designs for the upcoming prototype.

Appendix I - Engineering Standards

E1. The app must follow the guiding principle of the App Store to provide a safe experience for users to use. User interface of the app shall follow iOS Human interface guidelines [4].

E2. The Embedded System will comply with the IEEE Std 1003.13 standards for portable Realtime and Embedded Applications [10].

E3. The Embedded System and mobile device will comply with the IEEE 1625 standards for Batteries in Mobile Computing Devices [11]. The standards ensure safety of the system devices under environments.

E4. The iOS application will follow the commonly used Ray Wenderlich Swift coding style guideline [13]

E5. The Cloud Server will comply with the IEEE Std 2410 standards for Biometric Open Protocol [12]. The standards ensure security regarding cloud communication with devices.

E6. The Cloud Servers API will follow the industry leading Google Javascript coding style guideline [13]

E7. The Cloud Server, Embedded System and mobile application will comply with the IEEE 802.15.5 standards for Telecommunications and information exchange between systems [13]. The standards ensure security of component communications using wireless networks.

E8. The Cloud Server will comply with the Google Cloud Platforms API Design Guide [12]. The standards ensure security proper coding conventions.

References

[1] N. Tanner, M. Bin and M. Xiao, "Functional Specification for the Customizable Rewards Allocation System", 2018.

[2] “Integrate POS. Discover Apps.,” *Omnivore - Universal POS API*. [Online]. Available: http://omnivore.io/. [Accessed: 19-May-2018].

[3] Developer.apple.com. (2018). App Store Review Guidelines - Apple Developer. [online] Available at: https://developer.apple.com/app-store/review/guidelines/#safety [Accessed 20 Feb. 2018].

[4] Developer.apple.com. (2018). App Review - App Store - Apple Developer. [online] Available at: https://developer.apple.com/app-store/review/ [Accessed 20 Feb. 2018].

[5] Shaw, H. (2017). *Canadians just can’t seem to quit loyalty cards, despite all of the data breaches and PR headaches*. [online] Financial Post. Available at: http://business.financialpost.com/news/retail-marketing/canadians-just-cant-seem-to-quit-loyalty-cards-despite-all-of-the-data-breaches-and-pr-headaches [Accessed 20 Feb 2018].

[6] “API Design Guide | Cloud APIs | Google Cloud,” Google. [Online]. Available: https://cloud.google.com/apis/design/. [Accessed: 19-May-2018].

[7] “Uptime and downtime with 99.9 % SLA,” SLA & Uptime calculator: How much downtime corresponds to 99.9 % uptime. [Online]. Available: https://uptime.is/. [Accessed: 19-May-2018].

[8] *Apple.com*, 2018. [Online]. Available: https://www.apple.com/business/docs/iOS_Security_Guide.pdf. [Accessed: 19- May- 2018].

[9] “Core NFC,” *Purchase and Activation - Support - Apple Developer*. [Online]. Available: https://developer.apple.com/documentation/corenfc. [Accessed: 19-May-2018].

[10] “Host-based card emulation overview | Android Developers,” *Android Developers*. [Online]. Available: https://developer.android.com/guide/topics/connectivity/nfc/hce.html. [Accessed: 11-May-2018].

[11] “Adafruit PN532 RFID/NFC Breakout and Shield,” *Power Usage | Adafruit Motor Shield | Adafruit Learning System*. [Online]. Available: https://learn.adafruit.com/adafruit-pn532-rfid-nfc/ndef. [Accessed: 19-May-2018].

[12] “Core NFC,” *Purchase and Activation - Support - Apple Developer*. [Online]. Available: https://developer.apple.com/documentation/corenfc. [Accessed: 19-May-2018].

[13] Raywenderlich, “raywenderlich/swift-style-guide,” *GitHub*, 24-Jan-2018. [Online]. Available: https://github.com/raywenderlich/swift-style-guide. [Accessed: 19-May-2018].

[14] “Google JavaScript Style Guide,” *Google HTML/CSS Style Guide*. [Online]. Available: https://google.github.io/styleguide/jsguide.html. [Accessed: 19-May-2018].

[15] “Raspberry Pi 3 Model B,” *Raspberry Pi*. [Online]. Available: https://www.raspberrypi.org/products/raspberry-pi-3-model-b/. [Accessed: 19-May-2018].