**Software Requirements Specification**

**For**

**ERAU Eagle Wallet**

Team Members:

* Samantha Balistreri
* Tyler Wise
* Joseph Moran
* Michael Fornito
* William Edwards
* Daniela Regueira

Table of Contents

[**1. Introduction**](#_z61y4x1qdolt) **3**

[1.1 System to be Produced](#_h31ybe6jp5pn) 3

[1.2 Applicable Standards](#_xrtgtui3tm6x) 3

[1.3 Definitions, Acronyms, and Abbreviations](#_pzodhv4kc2ex) 3

[**2. Product Overview**](#_xu1rshw3xzjm) **3**

[2.1 Assumptions](#_v2vvcdeed4xe) 3

[2.2 Stakeholders](#_nhwsdzsm97ar) 3

[2.3 Event Table](#_tlvvvvsq9lug) 4

[2.4 Use Case Diagram](#_857mr7saez5j) 5

[2.5 Use Case Descriptions](#_l0kuzpeojvq1) 5

[**3. Specific Requirements**](#_x18wax9tluy5) **6**

[3.1 Functional Requirements](#_hiwtdbcsidte) 6

[3.2 Interface Requirements](#_ojtbmqdezaba) 12

[3.3 User and Human Factors Requirements](#_24xogj971hg5) 13

[3.4 Documentation Requirements](#_csea7zcp85ly) 21

[3.5 Data Requirements](#_jtffgwn70b74) 24

[3.6 Resource Requirements](#_j7ewxmrm8p8o) 27

[3.7 Security Requirements](#_tbbq39jc7i0) 27

[3.8 Quality Assurance Requirements](#_x87x3b2gd4m0) 27

[**4. Supporting Material**](#_upvy14ljl8si) **27**

# 1. Introduction

## 1.1 System to be Produced

* The product to be produced will be an application that allows Embry-Riddle faculty, staff, and students to access their Eagle Card and dining account through their phone. The application allows the user to make purchases through their phone using the NFC reader.

## 1.2 Applicable Standards

* This project should be held to industry standards in both software and cyber security engineering.
* This project follows all U.S. banking regulations.
* NIST password standards.

## 1.3 Definitions, Acronyms, and Abbreviations

* IDE - Integrated Development Environment
* Android Studio - IDE where the application will be constructed
* NFC - Near-Field Communication
  + A set of communication protocols for communication between two electronic devices over a distance of 4 cm (1​1⁄2 in)
* Android Versions
  + Android 8 - Oreo, minimum required android version for the application
  + Android 9 - Pie
* Emulator - hardware or software that permits programs written for one computer to be run on another computer.
* API - a software intermediary that allows two applications to talk to each other
* Widget - a component of an interface, that enables that enables a user to perform a function

# 2. Product Overview

## 2.1 Assumptions

* Assume that the user is an Embry-Riddle faculty, staff, or student
* The only payment methods available are through Google Pay or Paypal (whatever we implement)
* Assume the user has an android device with Android 8.0 or higher

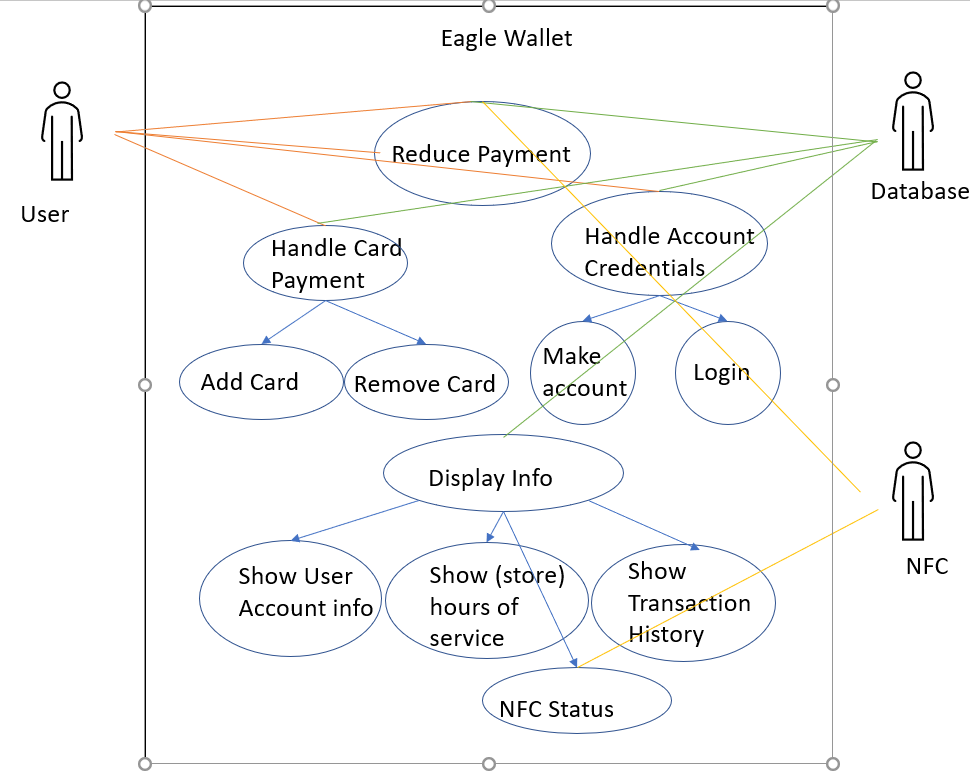
## 2.2 Stakeholders

* Customer - Dr. Jafer
  + Identifies requirements and is invested in the final product
* Product Owner - Fatimah
  + Assisting the scrum team in prioritization and evaluating project progress
* Scrum Team Members - Samantha, Tyler, Joseph, Michael, Will, and Daniela
  + Completion of the project

## 2.3 Event Table

|  |  |  |  |
| --- | --- | --- | --- |
| **Event Name** | **External Stimuli** | **External Responses** | **Internal data and state** |
| Login Successful | User successfully enters the username and password pair for their account | Application opens to the home page | Database cross references login information on file and loads account |
| Login Failed | User fails to successfully enter the username and password pair for their account | Application prompts user to attempt logging in again | Database cross references login information on file and returns an error |
| Account Created | User enters username, password and personal information | Application opens to home page | Database creates a user entry |
| Card Added | User enters card information | Application indicates to user that card has been successfully added and it will now show up in the list of payment options | Database securely stores payment information |
| Card Not Added | User enters incorrect card information | Application indicates to users that the entered card information is incorrect and prompts user to enter information again | Has no effect on internal data |
| Card Removed | User clicks the delete icon next to the card | Application indicates to user that card has been successfully removed and it will not show up in the list of payment options | Database wipes payment information |
| Process Payment | User chooses preferred payment method NFC is enabled | Application prompts user to place their phone within 4 cm of the scanner | Upon successful scan, payment entity will handle payment |
| Logout Successful | User clicks logout button | Application displays logout successful notification and sends user to login page | Database unloads user information off of local device |
| Application Timeout | User is inactive for more than 5 minutes | Application will automatically log out and return to login page | User information is secured/ system goes through logout protocol |

## 2.4 Use Case Diagram



## 2.5 Use Case Descriptions

1. Reduce Payment - The user will use the NFC take reduce the payment amount (dinning, eagle dollars, meal plans, or Sodexo Bux) and have it update within the database
2. Handle Card Payment - Deals with managing and showing the payment method for the user while connected to the database
3. Add Card - The user will add a card/payment method to their account
4. Remove Card - The user will remove a card/payment method from their account
5. Handle Account Credentials - The database has all the user’s account info (card, location, name, email, number, amount of funds, student ID)
6. Make Account - The user shall make an account and save it in the database
7. Login - The user shall login into their account
8. Display Info - Shall handle all the display information for the user
9. Show User Account Info - Shall show the users account credentials such as settings (name, location, number, email, student ID)
10. Show Store Hours - Shall display to the user all the stores open or closed on campus
11. Show Transaction History - Shall display all the user payments from the NFC, gathered from the database
12. NFC Status - Shall allow the user to manage and display the NFC status

# 3. Specific Requirements

## 3.1 Functional Requirements

|  |
| --- |
| No: F1 |
| Statement: User will be able to create an account with a unique username and password that meets NIST password standards. |
| Source: Basic functions |
| Dependency: None |
| Conflicts: Password does not meet requirement.  Username is not unique. |
| Evaluation Method:   1. Open the application and select create account 2. Fill in the required fields 3. Login into account using credentials |
| Revision History:   1. Daniela Regueira - 02/10/2021 - Version 1 |

|  |
| --- |
| No: F2 |
| Statement: The program shall show the account balance in the homepage. |
| Source: Basic functions |
| Dependency: F1 |
| Conflicts: None |
| Evaluation Method:   1. Login to app 2. Go to homepage |
| Revision History:   1. Daniela Regueira - 02/10/2020 - Version 1 |

## 

|  |
| --- |
| No: F3 |
| Statement: The program shall allow the user to enter their credit card information or add their Google Pay. |
| Source: Basic functions |
| Dependency: F1 |
| Conflicts: None |
| Evaluation Method: |
| Revision History: Daniela Regueira - 02/10/2021 - Version 1 |

## 

|  |
| --- |
| No: F4 |
| Statement: The program shall allow users to use a saved payment method to purchase additional dining dollars and eagle dollars. |
| Source: Basic Functions |
| Dependency: None |
| Conflicts: None |
| Evaluation Method: A test transaction shall be performed and proper responses verified. |
| Revision History: Tyler Wise - 2/10/2021 - Version 1 |

|  |
| --- |
| No: F5 |
| Statement: The program shall allow the user to view their balance of sodexo bucks, dining dollars, eagle dollars, and meal plans. |
| Source: Basic Functions |
| Dependency:None |
| Conflicts: None |
| Evaluation Method: A test account shall be made and balance info shall be verified. |
| Revision History: Tyler Wise - 2/10/2021 - Version 1 |

## 

|  |
| --- |
| No: F6 |
| Statement: The system shall refill all active meal plans to maximum capacity at 00:01 every Sunday. |
| Source: Basic functions |
| Dependency: None |
| Conflicts: None |
| Evaluation Method: |
| Revision History: Tyler Wise - 2/10/2021 - Version 1 |

|  |
| --- |
| No:F7 |
| Statement: The System shall allow the user to login using biometrics once their account has been created |
| Dependency: None |
| Conflicts: None |
| Supporting Materials:None |
| Evaluation Method: |
| Revision History:   1. Joseph Moran 2/11/2021 |

|  |
| --- |
| No:F8 |
| Statement: The System shall decrease the used amount from the appropriate account when a purchase is made |
| Dependency: None |
| Conflicts: None |
| Supporting Materials:None |
| Evaluation Method:   1. User makes a payment 2. User checks account balance |
| Revision History:   1. Joseph Moran 2/11/2021 |

## 3.3 User and Human Factors Requirements

* The following are user interface requirements and system interface requirements that will allow the user open and navigate the app

|  |
| --- |
| No: #HF1 |
| Statement: The user shall make an account |
| Source: Basic Functions |
| Dependency: None |
| Conflicts: None |
| Supporting Materials: None |
| Evaluation Method:   1. User clicks on make an account and is taken to the make account set-up page 2. User shall enter their credentials into the system 3. Database shall be populated |
| Revision History:   1. Michael Fornito 9/10/21 |

|  |
| --- |
| No: #HF2 |
| Statement: User shall login |
| Dependency: HF#1 |
| Conflicts: None |
| Supporting Materials: None |
| Evaluation Method:   1. User signs in using the password and username or fingerprint 2. Data credentials are retrieved from the database 3. User is taken to its home page upon successful data retrieval |
| Revision History:   1. Michael Fornito 9/10/21 |

|  |
| --- |
| No: #HF3 |
| Statement: User shall add payment |
| Dependency: HF #1, HF #2 |
| Conflicts: None |
| Supporting Materials: None |
| Evaluation Method:   1. User uses google pay or adds their credit card information 2. Database is populated 3. GUI shows that the payment is added |
| Revision History:   1. Michael Fornito 9/10/21 |

|  |
| --- |
| No: #HF4 |
| Statement: User can remove a payment method |
| Dependency: HF #1, HF #2 |
| Conflicts: None |
| Supporting Materials:None |
| Evaluation Method:   1. User goes to settings 2. User chooses which payment to remove 3. Database has been populated 4. GUI shows the payment gone |
| Revision History:   1. Michael Fornito 9/10/21 |

|  |
| --- |
| No: #HF5 |
| Statement: User can view its account balance |
| Dependency: HF #1, HF #2 |
| Conflicts: None |
| Supporting Materials: None |
| Evaluation Method:   1. User clicks on the home page icon 2. User is shown all its balances (dining dollars, sodexo, eagle dollars, and meal plans) |
| Revision History:   1. Michael Fornito 9/10/21 |

|  |
| --- |
| No: #HF6 |
| Statement: The user can view account transactions |
| Dependency: HF #1, HF #2 |
| Conflicts: None |
| Supporting Materials:None |
| Evaluation Method:   1. The user clicks on the $ icon 2. The user is shown a list of payment transactions |
| Revision History:   1. Michael Fornito 9/10/21 |

|  |
| --- |
| No: #HF7 |
| Statement: The user shall turn on NFC |
| Dependency:HF #1, HF #2 |
| Conflicts: None |
| Supporting Materials:None |
| Evaluation Method:   1. The NFC Icon is red on homepage 2. The user clicks the NFC Icon 3. The NFC icon is on (green) |
| Revision History:   1. Michael Fornito 9/10/21 |

|  |
| --- |
| No: #HF8 |
| Statement: The user shall turn off NFC |
| Dependency:HF #1, HF #2 |
| Conflicts: None |
| Supporting Materials:None |
| Evaluation Method:   1. The NFC icon is green 2. The user clicks on the Icon 3. The NFC icon is red (off) |
| Revision History:   1. Michael Fornito 9/10/21 |

|  |
| --- |
| No: #HF9 |
| Statement: The user shall make a successful payment |
| Dependency:HF #1, HF #2, HF #3, HF #7 |
| Conflicts: None |
| Supporting Materials:None |
| Evaluation Method:   1. The NFC icon is green 2. There is enough funds 3. The user places the phone on top of the NFC 4. Funds have been deducted 5. Displays successful payment |
| Revision History:   1. Michael Fornito 9/10/21 |

|  |
| --- |
| No: #HF10 |
| Statement: The user shall make a failing payment |
| Dependency:HF #1, HF #2, HF #3, HF #8 |
| Conflicts: None |
| Supporting Materials:None |
| Evaluation Method:   1. The NFC icon is red 2. There is not enough funds 3. The user places the phone on top of the NFC 4. Funds have not been deducted 5. Displays failing payment |
| Revision History:   1. Michael Fornito 9/10/21 |

|  |
| --- |
| No: #HF11 |
| Statement: The user shall go to settings |
| Dependency:HF #1, HF #2 |
| Conflicts: None |
| Supporting Materials:None |
| Evaluation Method:   1. The user clicks the gear icon 2. User is taken to settings 3. All credentials from database show (email, location, number, name, cards on file) |
| Revision History:   1. Michael Fornito 9/10/21 |

|  |
| --- |
| No: #HF12 |
| Statement: The user shall logout |
| Dependency:HF #1, HF #2 |
| Conflicts: None |
| Supporting Materials:None |
| Evaluation Method:   1. User clicks on logout button 2. User is taken to login page |
| Revision History:   1. Michael Fornito 9/10/21 |

|  |
| --- |
| No: #HF13 |
| Statement: The user shall upload a picture to their profile |
| Dependency: |
| Conflicts: None |
| Supporting Materials:None |
| Evaluation Method:   1. User clicks on gear icon 2. User is taken to settings 3. Profile picture is displayed above credentials |
| Revision History:   1. Joseph Moran 2/11/2021 |

## 3.4 Documentation Requirements

|  |
| --- |
| No: #DOC1 |
| Statement: The API for the project shall be documented through a Swagger contract. |
| Dependency: None |
| Conflicts: None |
| Supporting Materials: None |
| Evaluation Method: The documentation will be accessed through the route $BASE\_ROUTE/swagger/index.html |
| Revision History: Tyler Wise - 2/10/2021 - Version 1 |

## 3.5 Data Requirements

|  |
| --- |
| No: #D1 |
| Statement: An account shall store a username, hashed and salted password, email, first and last name, and Embry-Riddle student id number. |
| Dependency: None |
| Conflicts: None |
| Supporting Materials: None |
| Evaluation Method: |
| Revision History: Tyler Wise - 2/10/2021 - Version 1 |

|  |
| --- |
| No: #D2 |
| Statement: Each account shall have a one-to-many relationship with stored credit cards. |
| Dependency: None |
| Conflicts: None |
| Supporting Materials: None |
| Evaluation Method: |
| Revision History: Tyler Wise - 2/10/2021 - Version 1 |

|  |
| --- |
| No: #D3 |
| Statement: Each account shall have a one-to-one relationship with stored Google Pay accounts. |
| Dependency: None |
| Conflicts: None |
| Supporting Materials: None |
| Evaluation Method: |
| Revision History: Tyler Wise - 2/10/2021 - Version 1 |

|  |
| --- |
| No: #D4 |
| Statement: Each account shall have a one-to-many relationship with meal plans that consist of a current balance, maximum balance, meal plan begin date, and meal plan end date. |
| Dependency: None |
| Conflicts: None |
| Supporting Materials: None |
| Evaluation Method: |
| Revision History: Tyler Wise - 2/10/2021 - Version 1 |

## 3.6 Resource Requirements

* Windows 10 platform, 64-bit
* Android Studio
* Three sprints
  + Sprint 1 ends 2/25/2020
  + Sprint 2 ends 3/25/2020
  + Sprint 3 ends 4/29/2020

## 3.7 Security Requirements

|  |
| --- |
| No: #SEC1 |
| Statement: Each user will have their own account, entirely separate from other users. |
| Dependency: None |
| Conflicts: None |
| Supporting Materials: None |
| Evaluation Method: |

|  |
| --- |
| No: #SEC2 |
| Statement: User data will be protected with login/biometrics. |
| Dependency: None |
| Conflicts: None |
| Supporting Materials: None |
| Evaluation Method: |

|  |
| --- |
| No: #SEC3 |
| Statement: Sensitive user information will not be displayed unless the user is actively accessing or editing it. |
| Dependency: None |
| Conflicts: None |
| Supporting Materials: None |
| Evaluation Method: |

|  |
| --- |
| No: #SEC4 |
| Statement: System shall allow the user to disable NFC capability in the app, if desired. |
| Dependency: None |
| Conflicts: None |
| Supporting Materials: None |
| Evaluation Method: |

* must access to the system or information be controlled?
* must one user's data be isolated from others?
* how will user programs be isolated from other programs and from the operating system?
* how often will the system be backed up?
* must the backup copies be stored at a different location?
* should precautions be taken against fire, water damage, theft, ...?
* what are the recovery requirements?

## 3.8 Quality Assurance Requirements

|  |
| --- |
| No: #QA1 |
| Statement: The system shall notify the user when payment fails |
| Dependency: None |
| Conflicts: None |
| Supporting Materials: None |
| Evaluation Method: |

|  |
| --- |
| No: #QA2 |
| Statement: The system shall allow the user to access all of the different accounts available for purchasing food on campus. |
| Dependency: None |
| Conflicts: None |
| Supporting Materials: None |
| Evaluation Method: |

* What are the requirements for reliability, availability, maintainability, security, portability ...?
* How must these quality attributes be demonstrated?
* Must the system detect and isolate faults? If so, what types of faults?
* Is there a prescribed mean time between failures?
* Is there a prescribed time the system must be available?
* Is there a maximum time allowed for restarting the system after a failure?
* What are the requirements for resource usage and response times?