

Lab 2 – BetterSwipe Prototype Product Specification

Team Blue

Old Dominion University

CS411W

Professor J. Brunelle

February 14, 2024

Version 1

Table of contents

1. Introduction.....3

 1.1. Purpose.....3

1.2. Scope.....3

1.3. Definitions, Acronyms, and Abbreviations.....5

1.4. References.....7

1.5. Overview.....8

2. General Description.....9

 2.1. Prototype Architecture Description..... 9

 2.2. Prototype Functional Description..... 10

 2.3. External Interfaces..... 11

 2.3.1. Hardware Interfaces..... 11

 2.3.2. Software Interfaces..... 11

 2.3.3. User Interfaces..... 11

 2.3.4. Communications Protocols and Interfaces.....11

3. Specific Requirements.....12

 3.1. Functional Requirements.....12

 3.2. Performance Requirements.....14

 3.3. Assumptions and Constraints.....14

 3.4. Non-functional Requirements.....14

1. Introduction

1.1. Purpose

- The name of the product is BetterSwipe
- BetterSwipe provides users with a simplified process to make it easier for them to find which credit card suits their needs.
- BetterSwipe will build a spending profile for the users from previous transactions in order to recommend reward cards to the user.
- BetterSwipe is for consumers looking for a new credit card, consumers who want to maximize the use of their credit card rewards, and consumers who are looking to build their credit responsibility.
- Do's: Simplifies reward card selection process, recommends rewards credit cards based on user's spending profile, provides automatic analysis of expenditures versus card benefits, and analyzes card performances of selected cards against predicted benefits
- Don'ts: Raises the user's credit score, forces users to make hasty decisions, use the application on unsecured networks, and share login information with others.

1.2. Scope

- BetterSwipe is an application designed to simplify the complex process of obtaining reward credit cards. Its core belief is that consumers should have access to a straightforward, data-driven method to identify credit card options that best suit their spending habits and financial behavior. The overarching goal of BetterSwipe is to personalize the credit card selection process, minimizing the time consumers spend comparing options and maximizing the rewards they earn based on their financial behavior.

- By analyzing a user's transaction history, BetterSwipe can provide tailored recommendations that are more accurate and beneficial to the user than generic advice. This application aims to empower users to make informed decisions, leverage rewards to their full potential, and foster financial literacy and responsibility.
- BetterSwipe includes modules for secure user login and registration, statement uploads for transaction tracking, and sophisticated algorithm that recommends credit cards by comparing the user's spending profile against a database of credit card rewards

1.3 Definitions, Acronyms, and Abbreviations

- **Artificial intelligence (AI):** development of computer systems capable of performing tasks that historically required human intelligence, such as recognizing speech, making decisions, and identifying patterns.
- **Application Programming Interface (API):** software intermediary that allows two applications to talk to each other. APIs are an accessible way to extract and share data within and across organizations.
- **Annual Percentage Rate (APR):** the cost you pay each year to borrow money, including fees, expressed as a percentage. The APR is a broader measure of the cost to you of borrowing money since it reflects not only the interest rate but also the fees you must pay to get the loan.
- **Amazon Web Services (AWS):** is a subsidiary of Amazon that provides on-demand cloud computing platforms and APIs to individuals, companies, and governments, on a metered, pay-as-you-go basis. Clients will often use this in combination with autoscaling.
- **CSS:** Cascading Style Sheets, Programming language used to determine how HTML elements are displayed.
- **GitHub:** Service that provides hosting and version control for software development projects.
- **Graphical User Interface (GUI):** a form of user interface that allows users to interact with
- electronic devices through graphical icons and audio indicators.

- **HTML:** Hyper Text Markup Language, a programming language used for creating Web pages.
- **Javascript(JS):** a scripting programming language that enables dynamic content updates.
- **Machine Learning (ML):** a branch of AI and Computer Science that focuses on the use of data and algorithms to imitate the way humans learn, gradually improving its accuracy.
- **MySQL:** Open source database management system that could be used from creating a simple shopping list to holding a large amount of company data.
- **Rewards Credit Card:** credit cards that offer you some type of “reward”—typically cash back, points, or travel miles—for every dollar you spend, sometimes up to certain limits.

1.4 References

- Black, M. L. (2022, August 15). *Survey: Credit cardholders sitting on unused rewards*. LendingTree.
<https://www.lendingtree.com/credit-cards/study/unused-rewards/>
-
- Credit Karma. (2023). *Compare credit cards & apply online instantly*. Intuit Credit Karma.
<https://www.creditkarma.com/credit-cards/search-cc>
-
- Experian. (2017). *Survey Findings: How Do Consumers Feel About Credit Cards?*
<https://www.experian.com/blogs/ask-experian/survey-findings-how-do-consumers-feel-about-credit-cards/>
-
- NerdWallet. (2023). *Compare Rewards Credit Cards for Good Credit. Rewards Credit Cards - Compare 98 Card Offers*.
<https://www.nerdwallet.com/rewards-credit-cards?amex=1>

1.5 Overview

- This product description delineates the comprehensive hardware and software framework, external interface, and the extensive capabilities and features of the BetterSwipe prototype. Detailed within the subsequent sections of this document, readers will find an elaborate outline of the hardware configuration designed to support BetterSwipe across various platforms, from personal computers to mobile devices. The software architecture, powered by a combination of robust programming languages and contemporary frameworks, establishes a secure and intuitive user environment.

2. General Description

2.1. Prototype Architecture Description

- Application Front-End: Developed with ReactJS, offering a responsive user interface that supports both user and admin roles for interaction and management
- Statement upload and OCR analysis: Users can upload their financial statements, which are then processed using Optical Character Recognition (OCR) to extract transactional data.
- Backend processing with Django: The backend, built with Django, assembles the final expense profile from OCR data and interacts with the SQL database to fetch and store user profiles and transactional information.
- SQL database: A SQL-based database securely stores user profiles, transactional data, and credit card information, ensuring data integrity and quick access for processing.
- Rewards Credit API: This component interacts with external credit card services to fetch the latest offers and updates, which are essential for generating up-to-date recommendations.
- API requests and responses: The system uses API calls to communicate between the Django backend and the Rewards Credit card API allowing for real-time data exchange.

- Output: The end products of the architecture are personalized credit card recommendations, detailed card usage reports, and expenditure profiles that are presented to the user for informed decision-making

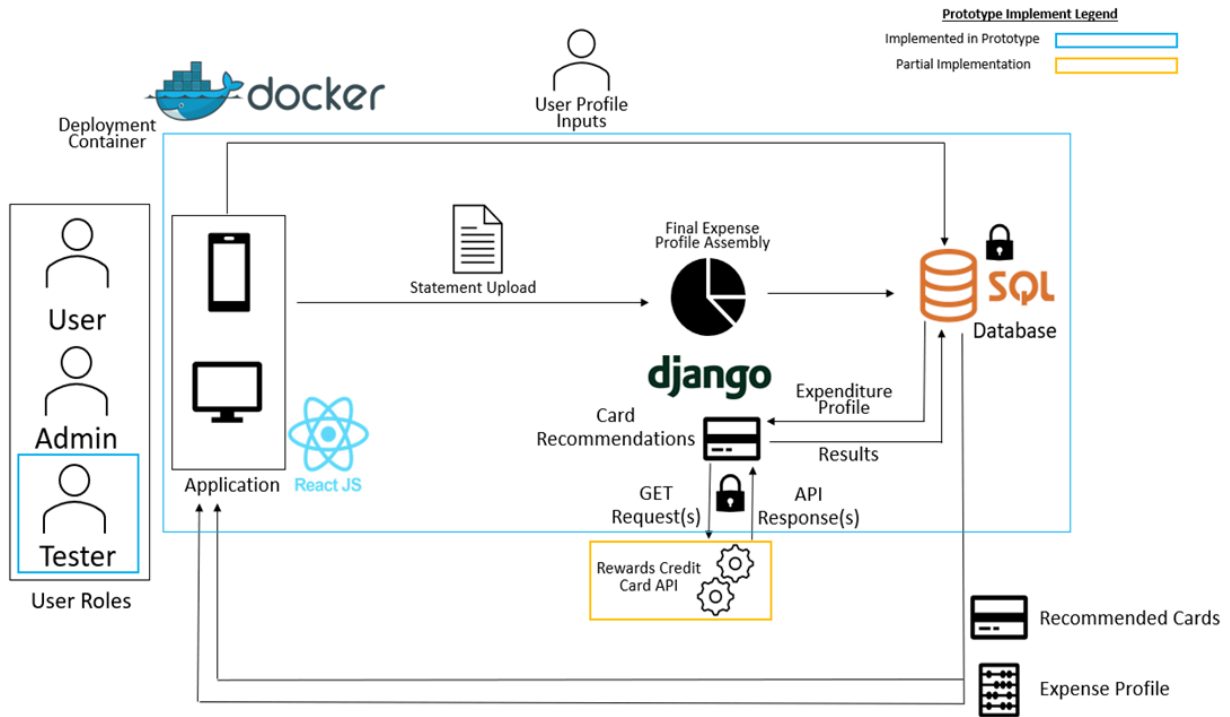


Figure 1. BetterSwipe Prototype Architecture Design MFCD

2.2. Prototype Functional Description

-

Functional Element	Prototype
General	
Web and Mobile Interfaces	Partially Implemented
Account Registration	Fully Implemented
Authentication	Fully Implemented
Profile CRUD	Fully Implemented
History of Reports	Fully Implemented
Security	
Data-at-Rest Encryption	Fully Implemented

Data-in-Transit Encryption	Partially Implemented
Statement Analysis	
Statement Upload	Fully Implemented
Expenditure categorization	Fully Implemented
Expenses Summary	Fully Implemented
Rewards Card Features	
Card Recommendations	Fully Implemented
Card Comparisons	Fully Implemented
Rewards Card Details	Fully Implemented
Continuous Monitoring	
Savings Prediction	Fully Implemented
Future Purchase Recommendations	Fully Implemented

2.3. External Interfaces

-

2.3.1. Hardware Interfaces

- Device with web connectivity and a browser.

2.3.2. Software Interfaces

- Rewards CC API
- SQL database

2.3.3. User Interfaces

- Web browsers including Edge, Firefox, Chrome, Opera, and more.
- Adaptive UI (adjusts to browser display size)

2.3.4. Communications Protocols and Interfaces

- HTTP

3. Specific Requirements

3.1. Functional Requirements

3.1.1. General

3.1.1.1. Web and Mobile Interfaces

3.1.1.1.1. The UI shall be adaptive to the screen size. The UI shall fit to the width of the screen and require no horizontal scrolling. Vertical scrolling may be used if the elements cannot fit within the screen space of the device. The UI shall have 3 layouts, and the layout chosen will be dictated by the display size:

3.1.1.1.1.1.

3.1.1.2. Account Registration (Christian & Rich)

3.1.1.2.1. As a user, I would like to have my own account.

3.1.1.2.2. GUI

3.1.1.2.2.1. BetterSwip will provide an interface through which a user will input a name for their account.

3.1.1.2.2.2. This interface will be accessible within the account settings section and features a label named "Account Name".

3.1.1.2.3. Algorithm

3.1.1.2.3.1. BetterSwipe will implement an algorithm to validate the input for the user's account name.

3.1.1.2.3.2. This algorithm will check for any limitations such as character length, prohibit certain characters (e.g. must not use %, #, @, \$), and especially prohibit any explicit language.

3.1.1.2.4. Database

3.1.1.2.4.1. BetterSwipe's database will include a field for storing an account name that is associated with each user profile.

3.1.1.2.4.2. This field must be strong enough to support strings of at least 50 characters to accommodate the account name.

3.1.1.2.5. The system shall include a password creation interface accessible to users during the account registration process.

3.1.1.2.6. Users shall be prompted to choose a password that meets specified security criteria, including minimum length,

- character complexity, and any additional requirements mandated by the system's security policy.
- 3.1.1.2.7. The system shall securely store user passwords using industry-standard encryption techniques, ensuring that passwords are not stored in plaintext or otherwise susceptible to unauthorized access.
 - 3.1.1.3. Authentication (Rich)
 - 3.1.1.3.1. The system shall automatically lock user accounts after a specified number of consecutive failed login attempts, as defined by the system's security policy.
 - 3.1.1.3.2. After successful authentication, the system shall reset the failed login attempt counter for the user's account and allow access to the account dashboard or designated areas of the application.
 - 3.1.1.3.3. Users shall be provided with clear instructions on how to unlock their account, including guidance on resetting their password or contacting support for assistance.
 - 3.1.1.4. Profile CRUD
 - 3.1.1.4.1. As a user I would like to be able to make changes to my account settings.
 - 3.1.1.5. History of Reports (John & ?)
 - 3.1.1.5.1. As a user, I would like to see my past transactions.
 - 3.1.1.5.2. As a user, I would like to see my rewards history.
 - 3.1.1.5.3. As a user, I would like to see previous card recommendations.
 - 3.1.2. Security
 - 3.1.2.1. Data-At-Rest Encryption
 - 3.1.2.1.1. As a user, I would like my personal and banking information to be secure when it is stored.
 - 3.1.2.2. Data-In-Transit Encryption
 - 3.1.2.2.1. As a user, I would like my personal and banking information to be secure when it is in transit.
 - 3.1.3. Statement Analysis
 - 3.1.3.1. Statement Upload
 - 3.1.3.1.1. As a user, I would like to be able to upload my bank statements so I don't have to type in all of my purchases by hand.
 - 3.1.3.1.2. As a user, I would like to be able to periodically upload more statements after the initial use.
 - 3.1.3.2. Expenditure Categorization (Marcelo & John)

- 3.1.3.2.1. The system shall allow users to upload their transaction history.
- 3.1.3.2.2. The system shall automatically categorize transactions into predefined categories (e.g. groceries, utilities, gas).
- 3.1.3.2.3. The system shall display the categorized transactions to the user within the application.
- 3.1.3.2.4. Algorithm
 - 3.1.3.2.4.1. The application will divide user expenses into appropriate categories.
 - 3.1.3.2.4.2. The application will accurately summarize the expense amounts.
- 3.1.3.2.5. Database
 - 3.1.3.2.5.1. BetterSwipe will store transaction summaries associated with the user profile.
- 3.1.3.2.6. GUI
 - 3.1.3.2.6.1. BetterSwipe will visually display transactions profile in a simple, easy to grasp manner.
- 3.1.3.3. Expenses Summary
 - 3.1.3.3.1. As a user, I would like to see a summary of how much I have been spending in each category.
- 3.1.4. Reward Card Features
 - 3.1.4.1. Card Recommendations (Cody)
 - 3.1.4.1.1. A ranking score shall be calculated for each card in the database that is able to be offered to the user. For each spending category, the user's spending sum in that category shall be multiplied by the card's return value in that same category to create a category score. The maximum category score between the categories shall be assigned as that card's ranking score.
 - 3.1.4.1.2. The cards shall be sorted by ranking score in descending order. The 3 cards with the highest ranking scores shall be the cards that get recommended to the user.
 - 3.1.4.1.3. As a user, I would like updated recommendations of rewards cards based on my updated bank statements.
 - 3.1.4.2. Card Comparisons
 - 3.1.4.2.1. As a user, I would like to be able to see the top 3 recommended cards compared side-by-side, so I have alternate options if I don't like the top recommendation.
 - 3.1.4.3. Rewards Card Details

- 3.1.4.3.1. As a user, I want to be able to see information about the rewards cards, such as APR and specifics about the rewards offered.
- 3.1.5. Continuous Monitoring
 - 3.1.5.1. Saving Prediction
 - 3.1.5.1.1. As a user, I would like to receive a general summary of how much I would earn if I used a selected rewards credit card.
 - 3.1.5.2. Future Purchase Recommendations
 - 3.1.5.2.1. As a user, I would like to be able to add information about future purchases so that they will be considered with savings predictions
 - 3.1.5.2.2. As a user, I would like to receive reminders of rewards that I have and where I can apply them with purchases I am already making.
- 3.2. Performance Requirements
(Numeric boundaries, should be measurable.)
 - 3.2.1.
- 3.3. Assumptions and Constraints
(Dependencies)
 - 3.3.1. The client has internet access
 - 3.3.2. RewardsCC can be accessed for rewards credit card data
- 3.4. Non-Functional Requirements (John)**
 - Security is the most important non-functional requirement for BetterSwipe
 - BetterSwipe is easily maintainable and has high availability
- 3.4.1. Data Management and Security**
 - 3.4.1.1. Protocol**
 - HTTPS will be used for requests between the front-end, back-end, and third-party APIs
 - 3.4.1.2. Database**
 - SQL database will be encrypted using AES-256.
 - 3.4.1.3. Authentication**
 - Prototype will utilize strong password requirements for users; 12 character minimum.
 - 3.4.2. Maintainability**
 - BetterSwipe will require periodic updates pertaining to updated rewards card information
 - 3.4.3. Reliability and Risk Management**

- BetterSwipe will be available 24/7, barring major maintenance or updates
- Critical functions in BetterSwipe have high availability, while non-critical functionalities downtime will not effect critical functionality
- Protection of financial information is addressed in the prototype through encryption
- Accurate card recommendations are addressed in the prototype through rigorous use-case testing