Software Requirements Specification
SRS

Billova

Andreas Drozd, Benjamin Lichtenstein, Sergiu Iordanescu

Document history

Version	Status	Date	Responsible person	Reason for change
1.0	Done	16.11.2024	Benjamin Lichtenstein	Document Creation.
2.0	Done	21.11.2024	Andreas Drozd	Adjustments of the document.
3.0	Done	08.12.2024	Andreas Drozd, Benjamin Lichtenstein, Sergiu-Claudiu Iordanescu	Add use cases description. Small corrections.

Abbreviations

SRS	Software Requirements Specification
OCR	Optical Character Recognition
App	Application
UI	User interface
AI	Artificial Intelligence

Andreas Drozd, Benjamin Lichtenstein, Sergiu Iordanescu

Table of contents

Andreas Drozd, Benjamin Lichtenstein, Sergiu Iordanescu

1. Introduction

1.1. Purpose

The purpose of the project is to practice Software Engineering.

1.1.1. Background and goals of the project

We want to create a website that uses OCR to scan receipts and reads their contents.

With this project our team wants to gather first experience using OCR technologies.

1.2. Product scope

1.2.1. Must-criteria

1.2.1.1. Functional

1) ID: M-F-01
   When the user presses an „add receipt" button, the system must provide the user with the possibility to scan a document via the camera or add it manually.

2) ID: M-F-02
   If a network connection is established, the system must be able to save text from a scanned receipt.

3) ID: M-F-03
   When no network connection is established or the OCR mechanism fails, the system must display an error message to the user.

4) ID: M-F-04
   When no network connection is established or the OCR mechanism fails, the system must provide the user with the ability to enter expenses and categories by themselves.

5) ID: M-F-05
   The system must provide the user with the ability to sign up / login to their own account.

6) ID: M-F-06
   The system must be able to store data in a database.

7) ID: M-F-07
   The system must provide the user with the ability to display a list of all expenses.

8) ID: M-F-08
   The system must provide the user with the ability to sort the list of all expenses by category and date.

9) ID: M-F-9
   The system must provide the user with the ability to search for expenses by date or category.

10) ID: M-F-10
    The system must provide the user with the ability to view their monthly spendings.

11) ID: M-F-11
The system must provide the user with the ability to change the currency, categories and profile details.

1.2.1.2. Non-Functional

12) ID: M-NF-01
The system must provide a responsive user interface, ensuring usability across different screen sizes, including mobile and desktop devices.

13) ID: M-NF-02
The system must display clear and concise error messages to guide the user in case of failures (e.g., OCR issues or missing internet connection).

14) ID: M-NF-03
User data, including scanned receipts and expenses, should be accessible only to the account holder.

1.2.2. Should-criteria

1.2.2.1. Functional

1. ID: S-F-01
The system should automatically generate summaries of spending patterns monthly.

2. ID: S-F-02
The system should display graphs to visualize spendings across different categories.

3. ID: S-F-03
If a set budget for a month is about to be reached, the system should provide the user with the ability to receive a notification

4. ID: S-F-04
The system should provide the user with the ability to read everything in their mother tongue.

5. ID: S-F-05
The system should provide multiple users with the ability to collaborate on tracking expenses.

6. ID: S-F-06
The system should extract more detailed information from the receipts.

7. ID: S-F-07
The system should provide the user with the ability to get personal insights based on their habit. (E.g. “You`ve spent 10% more on dining this month.”)

8. ID: S-F-08
The system should provide the user with the ability to automatically synchronize expenses from their bank account and compare receipts to match and categorize those expenses.

9. ID: S-F-09
The system must provide the user with the ability to export their expense data in a common format.

10. ID: S-F-10
When a receipt has been successfully scanned with OCR, the system must be able to identify the expense category.

1.2.2.2. Non-Functional

11. ID: S-NF-01

The system should support at least one language and display text in a readable font size and colour contrast.

1.2.3. Could-criteria

1.2.3.1. Functional

1. ID: C-F-01
   The system could provide the user with the ability to view personalized budgeting tips based on their spending behaviour.

2. ID: C-F-02
   The system could provide the user with the ability to set monthly budget limits for categories.

3. ID: C-F-03
   If there is a monthly budget limit set, the system could provide the user with the ability to send a notification when the limit is about to be reached.

4. ID: C-F-04
   If a budget limit has been met for the month, the system could provide the user with the ability to earn a reward badge.

5. ID: C-F-05
   The system could provide the user with the ability to add expenses through voice commands.

6. ID: C-F-06
   The system could tag receipts of common vendors like “Amazon” or “Starbucks”.

7. ID: C-F-07
   The system could process expenses with AI to see where expenses can be saved.

8. ID: C-F-08
   The system could store past receipts for future reference.

1.2.3.2. Non-Functional

9. ID: C-NF-01
   The system should display all content in a visually appealing, clear, and user-friendly layout. 90% of the users should be able to finish a simple interaction in the web application within 2 minutes e.g. scan a receipt or filter for certain timeframes or categories for their expenses.

1.2.4. Won’t-criteria

1. ID: W-01
   The system won’t be implemented with a native phone option.

Andreas Drozd, Benjamin Lichtenstein, Sergiu Iordanescu

2. General overview

2.1. Description of the initial situation (current state)

Users spend their money day to day. As a user I want to keep track of my expenses.

2.2. Product application

2.2.1. Areas of application

The product is a personal finance tracking tool. It can be used to get control of someone's own finances. It can be used to share expenses with multiple people so that an overview can be maintained.

2.2.2. Target groups, qualification level

People who earn and spend money, most likely adults. No qualification needed.

2.2.3. Operating conditions

- Internet connection
- Created Account

2.2.4. General restrictions

- Read only access to bank accounts.

2.2.5. Assumptions and dependencies

- Users own a phone with enough battery life to visit the website.

3. Use Case

Use Case	Receipt Scan with Camera
Description	This use case allows the user to scan a receipt with the device camera.
Stimulus	The user clicks on "Scan Receipt" and gets to the camera app, where he can scan the receipt by taking a picture.
Response	1. System shows the camera app where the user can take a picture. 2. System shows status message of the successful/unsuccessful scan.
Comments	If the step fails, the manual receipt entry screen will be displayed.

Use Case	Image Recognition
Description	Allows the system to use OCR to recognize and extract text from an image of a receipt provided.
Stimulus	1. The user uploads an image of a receipt or captures it using the camera 2. The user submits the image for OCR processing
Response	1. The system accepts the image and begins OCR processing. 2. The system extracts text from the receipt, like amount and date 3. The system displays the extracted text in a user-friendly format

Comments	If the step fails, the manual receipt entry screen will be displayed.

Use Case	**Manual Receipt Entry**
Description	This use case allows the user to add a receipt manually if the scanning is not being used.
Stimulus	1. The user clicks on “Add Receipt manually” 2. The user then enters the data in the needed fields.
Response	1. The system returns a form with the blank fields, which the user needs to fill out. 2. The system saves the receipt after the user has entered the correct data and submitted it.
Comments	Can be used in case of a failure while scanning instead or manually if the user prefers it.

Use Case	**Data Storage**
Description	A database to store all the data.
Stimulus	1. After every receipt scan the data scanned by the user gets stored in the data base. 2. User profile settings are stored in the database. 3. User profiles are stored in the database. 4. Expenses can be retrieved from the database 5. User settings can be retrieved from the database
Response	The UI shows a status message when modifications to the database where made. (e.g. “Successfully created receipt entry”.)
Comments	

Use Case	**Display Expense List**
Description	Shows all the expenses from a user.
Stimulus	The user navigates to one specific expense window.
Response	All the expenses are displayed depending on the specific type. (e.g. per month, category, ...)
Comments	

Use Case	**Sort Expenses**
Description	Allows the user to sort their list of expenses by specific criteria such as date, category, or amount.
Stimulus	1. The user opens the expense list 2. The user selects sorting criteria
Response	1. The system processes the selected sorting criteria and rearranges the list based on it 2. The system displays the sorted list to the user
Comments	The user should have the ability to toggle between ascending and descending.

Use Case	**Display Monthly Spending**
Description	Shows the monthly spending of the user.
Stimulus	The user selects the show monthly spending view.
Response	A site is displayed that shows all the monthly spending.

Comments	Depending on the must/should/could implementations, the user sees either just the expenses as text or a visual representation of the monthly spendings.

Use Case	**Change Settings**
Description	This allows the user to change certain settings, like language, currency, add self-created expense categories etc.
Stimulus	The user adapts his settings.
Response	After clicking the “Save” button, the settings are stored in the database.
Comments