# SOFTWARE REQUIREMENT SPECIFICATION

# Smart Transaction Sorter and Analytics System

## REVISION HISTORY

| Version | Description | Date |
|---|---|---|
| 1.0 | This covers the major SRS documents. | 30-10-25 |
| 1.1 | Updated as per the Given Template | 04-12-2025 |

# Table of Contents

**1. Introduction** ..... **1**

1.2 Scope ..... 1

1.3 Definitions, Acronyms, and Abbreviations ..... 4

1.4 Overview ..... 4

**2. Functional Requirements** ..... **5**

2.1 Security Management ..... 5

2.1.1 Process Sign Up ..... 5

2.1.2 Process Sign In / Logout ..... 5

2.2 Category Management ..... 6

2.2.1 Create Spending Category ..... 6

2.2.2 Manage Spending Categories ..... 7

2.3 Data Ingestion Management ..... 7

2.3.1 Upload Transaction File (CSV) ..... 7

2.3.2 File Validation ..... 8

2.4 Intelligent Sorting Management ..... 8

2.4.1 Process Transactions (AI Sorting) ..... 8

2.4.2 Data Persistence & Saving ..... 9

2.5 Analytics Management ..... 9

2.5.1 View Dashboard ..... 9

2.5.2 View Spending Breakdown ..... 9

2.6 Administrator Management ..... 10

2.6.1 Admin Panel Access ..... 10

2.6.2 User Account Management ..... 10

**3. Non-Functional Requirements** ..... **11**

3.1 Security ..... 11

3.2 Usability ..... 11

3.3 Reliability.....11
3.4 Performance.....11
3.5 Design Constraints.....12
3.6 Data Ownership & Business Rules.....12
**4. External Interface Requirements.....12**
4.1 User Interfaces.....12
4.2 Hardware Interfaces.....12
4.3 Software Interfaces.....13

# 1. Introduction

The rise of the digital economy in Pakistan has been driven by the widespread adoption of fintech platforms such as Easypaisa, SadaPay, and JazzCash. These services have empowered millions of users by simplifying payments and providing instant access to their complete transaction history. This readily available data holds the potential to offer valuable insights into personal spending habits. However, for most users, this potential remains untapped as the data is typically provided in raw formats (e..g., CSV files) without sophisticated analytical tools.

This proposal introduces the Smart Transaction Sorter and Analytics system, a web-based applica- tion designed to unlock the value hidden within this raw financial data. The project's academic objective is to develop a functional, standalone tool that serves as a research-based feasibility study. The resulting application will be architected as a powerful blueprint for an intelligent analytics module. The long-term vision, while outside the immediate scope of this project, is that such a module could be pitched for integration into the very fintech platforms it complements, thereby closing the significant analytical gap in the current market

## 1.1 Purpose

The purpose of the Smart Transaction Sorter and Analytics System is to provide users with an intelligent, automated way to analyze their personal financial spending. This web application uses Artificial Intelligence (AI) to categorize raw transaction data (from CSV files) and instantly convert it into clear, visual charts, eliminating the need for hours of frustrating manual work.

## 1.2 Scope

The scope of this project is a standalone, web-based tool that accepts a financial history file (CSV), uses AI to categorize every line based on user-defined labels, and presents the results on a BI Dashboard. The system includes User authentication, Category Management, CSV Upload/Validation, AI Sorting Logic, Data Persistence, and a Visualization Dashboard.

- **In Scope:** User login, Category creation, CSV parsing, AI classification, Dashboard charts.
- **Out of Scope:** Direct integration (API) with bank/wallet accounts, payment processing, or real-time transaction tracking.

### 1.3 Definitions, Acronyms, and Abbreviations

| Term/Abbreviation | Definition |
|---|---|
| **STS** | Smart Transaction Sorter (The name of this application). |
| **AI** | Artificial Intelligence (The component that performs automatic classification). |
| **CSV** | Comma-Separated Values (The standard file format for data input/upload). |
| **BI** | Business Intelligence (Refers to the analytical dashboard and visualization tools). |
| **DB** | Database (The system storage where user and financial data are saved). |

### 1.4 Overview

This document describes the requirements for the Smart Transaction Sorter and Analytics System (STS). It serves as the single reference point for what the system must do and how well it

must perform. The STS acts as a complimentary Business Intelligence layer for existing digital wallets and bank applications that currently only offer raw data exports.

## 2. Functional Requirements

### 2.1 Security Management

#### 2.1.1 Process Sign Up

| ID | Requirement Description |
|---|---|
| **SRS-1** | The system shall allow new users to securely register by providing a username, email, and password. |
| **SRS-2** | The system must validate that the email address is unique and not already registered. |
| **SRS-3** | The system shall store user passwords using a strong one-way hash (never in plain text). |
| **SRS-4** | Upon successful registration, the system shall create a dedicated, isolated database scope for the user. |

#### 2.1.2 Process Sign In / Logout

| ID | Requirement Description |
|---|---|
| **SRS-5** | The system shall allow registered users to log in securely using their credentials. |

| | |
|---|---|
| **SRS-6** | The system shall use HTTPS/SSL for all data transfer between the client and server to prevent interception. |
| **SRS-7** | The system shall allow users to securely log out, terminating their current session. |

### 2.2 Category Management

#### 2.2.1 Create Spending Category

| ID | Requirement Description |
|---|---|
| **SRS-8** | The system shall allow users to create new, personalized spending category labels (e.g., "Groceries," "Utilities"). |
| **SRS-9** | The system shall validate that the category name is not a duplicate within the user's profile. |
| **SRS-10** | The system shall ensure all created categories are immediately available for the AI Sorting Engine to use. |

### 2.2.2 Manage Spending Categories

| ID | Requirement Description |
|---|---|
| **SRS-11** | The system shall allow users to view a list of all their existing categories. |
| **SRS-12** | The system shall allow users to edit the name of an existing category. |
| **SRS-13** | The system shall allow users to delete a category (and optionally reassign its associated transactions). |

## 2.3 Data Ingestion Management

### 2.3.1 Upload Transaction File (CSV)

| ID | Requirement Description |
|---|---|
| **SRS-14** | The system shall provide a simple interface for the user to upload a transaction history file. |
| **SRS-15** | The system is constrained to accepting only **CSV (Comma-Separated Values)** file formats. |
| **SRS-16** | The system shall provide a progress indicator or status feedback during the upload process. |

### 2.3.2 File Validation

| ID | Requirement Description |
|---|---|
| **SRS-17** | The system shall check the uploaded file for necessary columns (specifically: Date, Description, and Amount). |
| **SRS-18** | The system shall provide clear, user-friendly error messages if the file structure is invalid or missing data. |
| **SRS-19** | The system shall reject files that exceed a predefined size limit to maintain performance. |

## 2.4 Intelligent Sorting Management

### 2.4.1 Process Transactions (AI Sorting)

| ID | Requirement Description |
|---|---|
| **SRS-20** | The system shall use a smart classification model (AI) to read transaction descriptions from the uploaded file. |
| **SRS-21** | The system shall automatically assign the most relevant custom category to each transaction. |
| **SRS-22** | The system shall utilize the **Hugging Face Transformers** library for the classification logic. |

### 2.4.2 Data Persistence & Saving

| ID | Requirement Description |
|---|---|
| **SRS-23** | The system shall securely save the processed transaction data, including the new category label, to the user's database. |
| **SRS-24** | The system shall ensure that once categorized, the financial history persists until explicitly deleted by the user. |

## 2.5 Analytics Management

### 2.5.1 View Dashboard

| ID | Requirement Description |
|---|---|
| **SRS-25** | The system shall display a primary BI (Business Intelligence) dashboard view after processing is complete. |
| **SRS-26** | The dashboard shall be interactive, allowing users to hover over elements for more details. |

### 2.5.2 View Spending Breakdown

| ID | Requirement Description |
|---|---|

| | |
|---|---|
| **SRS-27** | The system shall display clear charts (e.g., pie charts) visualizing total spending broken down by custom categories. |
| **SRS-28** | The system shall calculate and display total expenditure for the uploaded period. |

### 2.6 Administrator Management

#### 2.6.1 Admin Panel Access

| ID | Requirement Description |
|---|---|
| **SRS-29** | The System Administrator must use distinct credentials to log in to the dedicated administrative interface. |
| **SRS-30** | The Admin Panel shall be secured and inaccessible to standard users. |

#### 2.6.2 User Account Management

| ID | Requirement Description |
|---|---|
| **SRS-31** | The Admin shall have the ability to View and Search for any registered user account. |

| | |
|---|---|
| **SRS-32** | The Admin shall have the ability to Create, Edit, or Delete user accounts if necessary. |
| **SRS-33** | The Admin shall have read-only access to view user-created Categories and processed Transactions for auditing purposes. |

## 3. Non-Functional Requirements

### 3.1 Security

- **Account Protection:** User passwords must be stored using a strong one-way hash.
- **Session Security:** The system must use HTTPS for all communication to prevent data interception.
- **User Isolation:** A user's data must be completely isolated and inaccessible to any other user.

### 3.2 Usability

- The interface must be intuitive, clean, and fully mobile-responsive.
- Training time for a normal user to understand the upload and sort process should be minimal (intuitive design).

### 3.3 Reliability

- The system must be available and function correctly 99.5% of the time.
- In any error scenario, the system must provide an appropriate message and retain the state of the user's data (no silent failures).

### 3.4 Performance

- **File Processing Time:** A standard file (up to 500 transactions) must be processed, sorted by the AI, and ready for viewing within 60 seconds.

- The system dashboard should load visualization results in under 3 seconds after processing is complete.

### 3.5 Design Constraints

- **Technology Stack:** Must use Python (Django Framework) for the backend logic.
- **Database:** Designed for use with PostgreSQL for scalability.
- **AI Library:** The system must utilize the Hugging Face Transformers library.
- **Architecture:** The code must follow the MVT (Model-View-Template) architectural pattern.

### 3.6 Data Ownership & Business Rules

- **Data Ownership:** All uploaded data remains the intellectual property of the User.
- **No Third-Party Sharing:** User financial data may not be shared, sold, or exposed to any third party.
- **Persistence:** Processed data remains available until the user explicitly decides to delete it.

## 4. External Interface Requirements

### 4.1 User Interfaces

The user interface will be entirely web-based, optimized for responsive viewing on desktop and mobile devices. Key interfaces include:

- Login/Registration Forms.
- Custom Category Management Form (CRUD interface).
- CSV File Upload Interface with progress/status feedback.
- Interactive BI Dashboard (using modern charting libraries like Plotly.js or Chart.js).
- A secure Admin Panel (for System Administrator use only).

### 4.2 Hardware Interfaces

No specialized hardware is required. The system will interface with standard internet-supported client devices (PC, laptop, mobile, tablet).

### 4.3 Software Interfaces

The system interfaces with the following software components:

- **Python / Django:** Primary backend web application and API interface.
- **PostgreSQL:** Database management system for data persistence.
- **Pandas Library:** Used for data manipulation, CSV parsing, and validation.
- **Hugging Face Transformers:** The core library used for the AI classification model.
- **Plotly.js / Chart.js:** Frontend JavaScript libraries for rendering visualizations.