# Transaction Report iOS Application - Feature Overview

This document outlines the core features, utilities, and structure of the Transaction Report iOS application developed using Swift, UIKit, and following the MVVM architecture.

---

# 1. Overview

This iOS application displays a user's transaction history in a tabular format and allows exporting the report as a well-formatted PDF. It also includes support for caching, pull-to-refresh, connectivity handling, and reusable components.

---

# 2. Key Features

## Pull to Refresh

Users can refresh the data by pulling the table view down. This action re-triggers the API call and updates the transaction list.

# API Integration with MVVM

The app fetches transaction data from a remote API. The API calling logic is encapsulated in a Generic APIService, ensuring a clean separation of concerns using the MVVM pattern.

# Data Caching

The last successful API response is cached using CacheManager. On app launch, cached data is used if it's less than 5 minutes old. This improves speed and provides offline support.

# PDF Report Generation

A downloadable PDF version of the transaction report is generated using the TransactionReportPDFGenerator. It supports:

- Proper header & table
- Multi-page support
- Clean formatting
-

Share, Save, Preview options

## Share and Save

Users can easily share the generated PDF or save it locally.

## Internet Connectivity Handling

Before making API calls, the app checks for internet availability. If there's no connection, a toast message is shown, and the API is not triggered.

## Toast Notifications

Success or failure messages are displayed using a reusable Toast utility, improving user interaction and providing clear feedback.

## Dynamic Transaction Cell UI

A custom UITableViewCell is used to display transaction category, date, and amount. The

amount is color-coded (green for credit, red for debit) and includes an arrow icon indicating credit/debit.

---

# 3. Utility Components

**TransactionReportPDFGenerator** and **Toast** are designed as reusable utilities that can be easily adapted for other parts of the app or different applications.

## Generic API Service

The APIService is fully generic, working with any Decodable response type, making it easy to scale and maintain.

---

# Thanks