Business Requirements Document (BRD)

1. Project Overview

1.1 Project Name

Steam Sale Analysis

1.2 Date of Document

May 2025

2. Business Objectives

The primary objective of this project is to develop a scalable, centralized data platform in **Microsoft Fabric** that mirrors the digital operations of a business like **Steam**. The platform will support analytics, reporting, and insights for stakeholders across product, marketing, and finance functions.

2.1 Centralize and Integrate Gaming Ecosystem Data

Objective:
Consolidate data from various source systems into a single, structured, and query-optimized environment to support holistic analysis.

Key Activities:

- Ingest six core datasets (games, users, orders, Wishlist, promotions, order items)
- Standardize schemas and relationships across domains
- Store raw and curated layers in the Lakehouse

Business KPIs:

- **Data freshness SLA:** > 95% of tables updated within SLA window (e.g., 12h)
- **Data completeness rate:** > 98% of expected records ingested per load
- **Schema validation errors:** < 2% per month

2.2 Enable Revenue and Sales Performance Analytics

Objective:

Provide robust reporting and insights on game performance, revenue generation, and purchase patterns to drive product and pricing decisions.

Key Activities:

- Model fact tables for orders, order items, and promotions
- Track total revenue, discounts, and refund rates
- Enable time-based trend analysis

Business KPIs:

- **Monthly Gross Revenue** by game and publisher
- **Promotion Conversion Rate:** % of users who purchased during a promotion
- **Users Genres favorites:** Most popular genres that a purchased by
- **Average Order Value (AOV):** Total revenue / number of orders
- **Top 10 Selling Games:** Ranked by revenue or units sold

2.3 Support User Behavior and Engagement Analysis

Objective:
Understand user preferences, demographic segmentation, and Wishlist dynamics to improve user engagement and marketing precision.

Key Activities:

- Enrich user profiles with game interaction and purchase data
- Track Wishlist additions and conversions to purchase
- Analyze demographics and device usage patterns

Business KPIs:

- **Wishlist-to-Purchase Conversion Rate**
- **New User Acquisition Rate:** Based on user registration dates
- **Churn Rate:** % of inactive users over time
- **Avg. Wishlist Size per User**

2.4 Ensure Data Governance, Security, and Scalability

Objective:
Implement a secure, governed, and scalable environment that complies with enterprise standards and supports future growth.

Key Activities:

- Mask PII fields such as email, phone, and card numbers

- Implement Row-Level Security in Power BI
- Monitor pipeline health, lineage, and schema drift

Business KPIs:

- **Security Audit Pass Rate:** 100% compliance with PII policies
- **Pipeline Success Rate:** > 98%
- **Time to Onboard New Dataset:** < 3 business days
- **Data Lineage Coverage:** 100% from source to report

2.5 Enable Developers and Publishers Performance Analytics

Objective:
Enable developers and publishers to track how their games perform in the marketplace and understand user engagement trends to guide future development, pricing, and promotion strategies.

Key Activities:

- Attribute game metrics (sales, reviews, wishlists) back to publishers and developers
- Provide self-serve dashboards with filters by game, genre, promotion, etc.
- Track game lifecycle performance (launch, peak, long-tail)

Business KPIs:

- **Game Launch Success Score:** Composite metric combining first-month sales, Wishlist adds, and ratings
- **Average Revenue per Game (ARPG):** Total revenue / number of published games per Developer
- **Review Sentiment Score:** Based on recommendation count or external rating integration
- **Number of Active Titles per Developer**
- **Total Revenue by Publisher**
- **Units Sold per Publisher Portfolio**
- **Promotion ROI per Publisher:** Incremental revenue from promoted games vs. non-promoted
- **Avg. Game Lifecycle (Active Sales Period)**

3. Project Scope

3.1 In Scope

The following components and activities are within the scope of this project:

- **Data Ingestion & Integration:**
    - Ingest six core datasets from a designated source (AWS)
    - Use Microsoft Fabric pipelines to move data into OneLake.
- **Data Transformation & Cleaning:**
    - Perform schema standardization, data validation, null/duplicate handling, and enrichment using Notebooks and Dataflows Gen2.
- **Data Storage:**
    - Store both raw and cleaned data layers in One Lake using Delta format.
    - Maintain a curated Lakehouse optimized for querying and reporting.
- **Data Modeling:**
    - Design and implement a dimensional model (snowflake schema).
    - Establish relationships between fact and dimension tables using Fabric's modeling tools.
- **Analytics & Reporting:**
    - Enable analytical KPIs such as revenue tracking, promotion performance, user engagement, and game lifecycle insights.
- **Data Governance & Security:**
    - Data masking policies
- **Performance & Scalability:**
    - Ensure pipelines, transformations, and reporting meet defined SLAs.

3.2 Out of Scope

- **Real-Time Data Streaming:**
    - No real-time ingestion or stream processing; batch updates on a weekly basis only.
- **Data Science or Predictive Modeling:**
    - No implementation of ML/AI models for recommendation systems, forecasting, or churn prediction.
- **Advanced Data Privacy Compliance:**
    - No full GDPR compliance audit or legal review; basic PII masking and classification only.

4. Requirements

4.1 Functional Requirements

These define the core data-related functionalities to be delivered:

1) **Data Ingestion & Integration**
   a) Ingest six CSV files from a designated storage location.
   b) Create ETL pipelines to process and integrate data into a centralized data warehouse or lake.
2) **Data Transformation & Cleaning**

Data Source	Column	Transformation
Games	All Columns	Remove Whitespace in column name
Games	Release Date	Convert to date format dd/mm/yyyy
Profiles	All columns	Check for nulls
Profiles	All columns	Check for duplicates
Profiles	All columns	Check for data validation (valid phone, email ...etc.)
Promotions	All Columns	Data quality is the same as profiles
Orders	All columns	Check for nulls
Orders	All columns	Check for Duplicates
Orders	All columns	Check for data validation (valid Dates, valid amount...etc.)
Orders	Taxes	Correct tax rate based on user country
Orders_items	All columns	Check for nulls
Orders _items	All columns	Check for Duplicates
Orders _items	All columns	Check for data validation (valid Dates, valid amount...etc.)

3) **Data Modeling**
   a) Store cleaned data in a **Lakehouse** or **Warehouse** (Synapse Data Warehouse in Fabric).
   b) Establish foreign key relationships between tables using Fabric's **Relational modeling UI**.
   c) Create a star or snowflake schema and model the cleaned tables accordingly
   d) Establish joins and primary keys
4) **Data Validation**
   a) Implement data quality checks using **Data Quality Monitoring** or **custom Spark notebooks**
5) **PII Handling**

a) Use **Row-Level Security (RLS)** and **Data Masking Policies**
b) Store sensitive information in encrypted formats. (Optional)

4.2 Non-Functional Requirements

These specify quality attributes and performance expectations:

1) **Performance**
    a) Leverage DirectLake or Import mode in Power BI for performance tuning.
    b) Ensure Spark jobs and transformations in Lakehouse complete within defined SLAs (e.g., <1 hour for 100k records/file).
2) **Scalability**
    a) Use Auto-Scaling Compute Pools for heavy workloads in notebooks or pipelines.
    b) Data architecture must support horizontal scaling via Fabric's distributed compute.
3) **Reliability**
    a) ETL processes must be built using **Fabric Pipelines with retry logic and failure handling**.
    b) Notification alerts for ETL failures must be configured.
4) **Security**
    a) Apply **Microsoft Entra (formerly Azure AD)** for authentication and access management. (**Optional**)
    b) Use **RLS and workspace roles** for managing data access.
5) **Compliance**
    a) For masking and visibility of data, your mentor should be able to access read some data while sensitive data such as name, email, credit cards should be masked and not all columns visible.
    b) Use Microsoft Purview for data governance, classification, and audit trails (**Optional**).
6) **Maintainability**
    a) Version control with **Git integration** in Power BI/Fabric workspace.
    b) Modular dataflows and reusable components in Notebooks or Pipelines.

4.3 Data Requirements

1) **Data Sources**
    a) All 6 Tables to be ingested into **OneLake**, Fabric's unified storage layer.
2) **Data Volume**
    a) Optimized for up to 1 million records/file; use Spark when processing large data in Notebooks.
3) **Update Frequency**
    a) Weekly file drops; configure **scheduled refresh in Data Pipelines** or **OneLake triggers**. (Optional)

4) **Data Quality Expectations**
    a) Validate data using **Notebooks with PySpark** or **Power Query validations** in Dataflows Gen2.
5) **Key Fields**
    a) Use relationships and metadata in **Warehouse/Lakehouse tables** to define schema and enforce constraints.
6) **Sensitive Data**
    a) Encrypt and classify sensitive fields using Microsoft **Information Protection labels**.

5. Data Flow

This section outlines the end-to-end data pipeline, from the source AWS database into Microsoft Fabric, including ingestion, transformation, modeling, and consumption layers.

5.1 Overview

The data flow involves the following stages:

1. **Source Database (AWS)**
2. **Ingestion into Microsoft Fabric**
3. **Staging in Lakehouse**
4. **Transformation and Data Quality**
5. **Data Modeling and Storage**
6. **Data Consumption and Reporting**

5.2 Detailed Data Flow Description

Step 1: Source Data – AWS Database

- **Source:** Relational database hosted on AWS (e.g., Amazon RDS for MySQL/PostgreSQL).
- **Access:** Secure connection using **ODBC/JDBC**, **SSL**, or **VPN/Private link**.
- **Tables Available:**
    - `steam_games`
    - `steam_profiles`
    - `promotions`
    - `wishlist`
    - `orders`
    - `order_items`

- **Frequency:** Data updated weekly; ingestion schedule will mirror this frequency via incremental load (if supported).

Step 2: Ingestion into Microsoft Fabric

- **Tool: Microsoft Fabric Data Pipelines** with support from **Azure Data Factory runtime**
- **Method:**
    - Use **Copy Data activity** or **Database connector** to pull data from AWS into Fabric.
    - Optional use of **Linked Services** with gateway or self-hosted integration runtime.
    - Load data directly into **OneLake's Lakehouse staging layer**.
- **Destination:** `Lakehouse > /Tables/staging/` or `/Files/raw/`

Step 3: Staging & Lakehouse Storage

- **Tool:** Microsoft **Lakehouse** in Fabric
- **Actions:**
    - Load full or incremental snapshots of source tables into staging tables.
    - Store in Delta Lake format for performance and schema enforcement.
    - Retain raw tables for auditing and rollback purposes.

Step 4: Data Transformation & Quality

- **Tools:**
    - **Notebooks (PySpark)** for heavy transformations and joins.
    - **Dataflows Gen2 (Power Query Online)** for business rule application and enrichment.
- **Tasks:**
    - Clean null or malformed data.
    - Split and normalize multi-value columns (e.g., `Genres`, `Categories`).
    - Create intermediate views or clean tables.
    - Validate referential integrity (e.g., ensure `AppID` in orders exists in `steam_games`).

Step 5: Data Modeling

- **Tool: Lakehouse SQL Endpoint** or **Data Warehouse in Fabric**
- **Model:**
    - Design a **star schema** or **snowflake schema**:
        - **Facts:** Orders, Order Items, Wishlist

        - **Dimensions:** Games, Users, Promotions, Genres, Publishers, Developers, Categories, Countries
    - Define relationships and constraints.
    - Implement calculated columns and derived metrics (e.g., `total_paid`, `discount_applied`).

Step 6: Data Consumption & Reporting (Optional)

- **Tool: Power BI (DirectLake or Import Mode)**
- **Action:**
    - Build dashboards and datasets on top of the curated Lakehouse tables.
    - Implement filters, KPIs, and dynamic visuals (e.g., revenue trends, top wishlisted games).
    - Enforce **Row-Level Security (RLS)** and **sensitivity labels** on reports.

5.3 Monitoring & Logging

- **Tool:** Fabric **Monitoring Hub**, integrated with **Purview** for data lineage.
- **Features:**
    - Log pipeline runs, errors, durations.
    - Track source-to-report lineage and data freshness.