SP-115 RED

POWER BI SALES PERFORMANCE DASHBOARD

FINAL REPORT

CS 4850 – Section 04

Fall 2025

Professor Sharon Perry

December 8th, 2025

Team Members:

Annika Andersson

Collin Knowles

Neel Ranawat

Jason Thai

Alex Thomas

Website: https://sp-115-powerbi-salesdashboard.netlify.app/

GitHub: N/A

STATS & STATUS

NDA	No
Lines of Code	N/A
Components/Tools	5 - Power BI Desktop, Power Query, DAX, Dataset (Video Game Sales 2024), OneDrive
Estimated Hours	536
Actual Hours	450
Project Status	70%

Table of Contents

Introduction *4*

Requirements *5*

Data Management **5**

User Access **5**

Dashboard Visualization **6**

Reporting and Exporting **6**

System Interactions and Usability **6**

Maintenance and Support **7**

Security **7**

Capacity **7**

Usability **7**

Performance and Other **7**

Layout and Navigation **8**

Filters and Controls 8

Display and Formatting Standards 8

Accessibility and Readability 8

Hardware Interface Requirements **9**

Software Interface Requirements **9**

Analysis/Design *9*

Development *15*

Data Preparation **15**

Data Model and KPIs **16**

Dashboard Prototype **17**

Challenges **18**

Solutions **18**

Database Connection **19**

Project Setup **19**

Development Conclusion ..... Error! Bookmark not defined.

Software Test Plan and Report *20*

Test Objectives **20**

Scope of Testing **21**

In-Scope 21

Out-of-Scope 21

Test Cases **21**

Test Procedures **23**

Test Environment **25**

Hardware Environment 25

Software Environment 25

Network Environment 26

Test Data **26**

Dataset 26

KPI Data 26

Filter and Slicer Data 27

Edge Case Data 27

Software Test Report (STR) **27**

Version Control ***29***

Conclusion ***30***

Appendix ***31***

Introduction

The SP-115 Red Team's Sales Performance Dashboard project focused on transforming a raw video games sales dataset into an interactive business intelligence dashboard using Microsoft Power BI. Our goal was to give sales users a clear understanding of market trends by converting the data into key performance indicators (KPIs), visualizations, and interactive filters. Through a combination of data preparation and cleaning, data modeling, DAX calculations, and intentional dashboard design techniques, we aimed to develop a dashboard that allows users to explore and analyze sales trends across regions, platforms, publishers, and genres.

The dataset we used throughout the project was *Video Game Sales 2024*, which includes information such as game titles, publishers, genres, scoring, release years, and regional sales metrics. Before building the dashboard, our development team did data preparation and cleaning for accuracy and consistency. This involved correcting data types, standardizing columns, and cleaning duplicate/null values. This step formed the foundation for the relational data model that we built in Power BI. This data model followed a hybrid star/snowflake schema to reflect the relationships between games, publishers, developers, consoles, and sales values.

The development of this project followed the major phases of the Software Development Life Cycle (SDLC). After putting together, the requirements and specifications in our Software Requirements Specification (SRS) and Software Design Document (SDD), we moved into the data modeling and dashboard prototyping phase. Our team worked on identifying the key business questions the dashboard needed to answer, as well as creating the DAX measures to drive our KPIs: Total Market Sales, Regional Market Share, Top Titles, Top Genres, and Sales by Platform. Then, our development team created the two dashboard views: an Admin Dashboard for strategic insights and a Sales Dashboard for logistical decision-making.

Throughout the entire project, our team's focus was on usability, clarity, and performance. We encountered a few constraints related to the free Power BI license, including limited collaboration and an inability to use a data connection which would have allowed us to mirror a real-world business scenario. Another constraint we ran into was scheduling conflicts among our team of five. Despite these obstacles, our team was able to create a sales performance dashboard that meets the project goals and shows Power BI's capabilities as a tool for data-driven decision-making. This final report will summarize the project's requirements, analysis, design decisions, developmental work, testing results, and implementation of our Power BI Sales Performance Dashboard.

Requirements

For the Power BI Sales Performance Dashboard, we simulated the requirements gathering process that would have been undertaken for a video games vendor interested in gaining a better understanding of their sales data for future business strategy development. The sales dashboard project was intended to be created within Microsoft Power BI and completely integrate the loading, transformation, and visualization of video games sales data. Using this data, the dashboard would be able to provide visualizations of key performance indicators (KPI's) that can be analyzed for business insights. These are the requirements that guided this project through its design, development, and testing. Features such as AI agent integration, machine learning, predictive analysis, mobile view dashboard and other licensed features were marked as out of scope.

For the dashboard's environment constraints, we considered the deployment platform, device requirements, browser compatibility, and dataset compatibility. The end user should be able to access a modern web browser such as Firefox, Microsoft Edge, or Google Chrome that can run the Power BI web application. Alternatively, the end user should be able to install and run the Power BI desktop application on a device that runs Windows 10 or Windows 11 with a minimum of 8 gigabytes of RAM. Datasets must be uncorrupted and be provided within the range of Power BI's supported formats such as .csv or Excel files.

System constraints of the dashboard included the provided development tools, user accounts, and access control. Our team developed the proposed dashboards with all tools provided within the latest version of the Microsoft Power BI Desktop application. All users, including developers and clients, are required to login to Power BI via their own Microsoft accounts. System and dashboard access is restricted to authorized users via Microsoft's authentication protocol.

The Functional Requirements are as follows:

Data Management

FR-01: The system shall allow importing sales data from CSV or XLS files.

FR-02: The system shall verify the compatibility of datasets before importing.

FR-03: The system shall clean and structure imported data for reporting.

User Access

FR-04: The system shall require users to log in with a licensed Microsoft Power BI account.

FR-05: The system shall restrict dashboard access to authorized users only.

FR-06: The system shall provide a management/admin dashboard with access to advanced KPIs and filters.

FR-07: The system shall provide a regular sales users dashboard with more simplified KPI views.

Dashboard Visualization

FR-08: The system shall display key performance indicators (KPIs), including:

- Total global sales
- Regional sales share
- Average sales per title
- Top-selling title per region
- Sales by console
- Sales by genre
- Sales growth over time

FR-09: The system shall allow users to filter data by year of release (time/period).

FR-10: The system shall allow users to filter data by platform (console) and genre.

FR-11: The system shall allow users to filter data by sales region (e.g., North America, Europe, Japan, Other, Global).

FR-12: The system shall display visualizations such as charts, graphs, and tables for KPI reporting.

FR-13: The system shall support a drill-down function for end users to see more detailed insights into data.

Reporting and Exporting

FR-14: The system shall allow users to export reports in PDF format.

FR-15: The system shall allow users to export reports in XLS format.

System Interactions and Usability

FR-16: The system shall provide users with the ability to interact with dashboard visualizations through applying filters and changing views.

FR-17: The system shall provide a responsive user interface that adapts to different screen sizes/resolutions.

Maintenance and Support

FR-18: The system shall notify the end users when there is a dataset compatibility issue.

FR-19: The system shall allow authorized users to update/replace datasets without a reconfiguration of the entire dataset.

The Non-Functional Requirements are as follows:

Security

NFR-01: Access to the dashboard shall require a licensed Microsoft Power BI account.

NFR-02: Only authorized users with access to the workspace shall be able to view the dashboard.

Capacity

NFR-03: The system shall support datasets up to 1 GB (per dataset) without performance issues.

NFR-04: Reports exported to PDF or Excel shall include applied filters and keep data accuracy.

Usability

NFR-05: A first-time user shall be able to locate KPIs and apply basic filters with minimal assistance.

NFR-06: Users shall be able to switch between report pages and dashboard without training.

NFR-07: Visualizations shall use consistent labels, legends, and formats for easy interpretation.

NFR-08: The system shall ensure that role-based dashboards (Admin vs. Sales) keep a consistent style.

Performance and Other

NFR-09: On a standard laptop/desktop (Windows 10/11, 8 GB RAM), the dashboard shall load in less than 10 seconds for datasets with the capacity stated.

NFR-10: After filters are applied, the dashboard shall refresh visualizations within 5-10 seconds for datasets under 1 GB.

NFR-11: The dashboard shall be accessible through current versions of web browsers, including Microsoft Edge and Google Chrome.

The External Interface Requirements are as follows:

Layout and Navigation

EIR-01: The dashboard shall display KPIs, including:

- Total global sales
- Regional sales share
- Average sales per title
- Top-selling title per region
- Sales by console
- Sales by genre
- Sales growth over time

EIR-02: The dashboard shall include two interfaces/dashboards: One for management/admin users and one for regular sales users.

EIR-03: The system shall provide navigation options (tabs/buttons) for switching between pages.

EIR-04: Each page shall use consistent formatting for titles, filters, and any other visuals.

Filters and Controls

EIR-05: The dashboard shall contain filters (also referred to as slicers) for time, platform, genre, region, and publisher.

EIR-06: The system shall provide a reset option to clear filters.

Display and Formatting Standards

EIR-07: All KPI cards, charts, and graphs shall use consistent labels, legends, and unit measurements.

EIR-08: Tooltips shall be available when hovering over visualization data to provide supporting information/details.

Accessibility and Readability

EIR-09: Text, labels, and controls shall be readable on a standard laptop/desktop display.

EIR-10: All interactive elements, including filters and buttons, shall be designed with enough size and space for ease of selection using a mouse or trackpad.

Hardware Interface Requirements

The Sales Performance Dashboard will be designed to run on standard computing hardware.

EIR-11: The system shall be accessible on laptops or desktops running Windows 10 or Windows 11.

EIR-12: Devices shall have at least 8 GB of RAM to support Power BI Desktop.

EIR-13: Devices shall include a keyboard and either a trackpad or mouse.

Software Interface Requirements

The Sales Performance Dashboard will be dependent on Microsoft Power BI tools and standard file formats for data import/export.

EIR-14: Development shall be performed using the latest version of Microsoft Power BI Desktop.

EIR-15: Reports shall be deployed to the Power BI Online Service for viewing/interacting.

EIR-16: The dashboard shall allow importing datasets in CSV or XLS formats.

EIR-17: The dashboard shall allow exporting datasets in PDF or XLS formats.

EIR-18: The dashboard shall be viewable in common web browsers, such as Microsoft Edge and Google Chrome (latest versions).

Analysis/Design

Design Considerations

When considering the design, there were multiple ideas that came to mind. The team had concluded that simplicity with scalability was the best way to appeal while making the software accessible to all new users. This allows growth for the software while providing a user-friendly interface that meets the needs of the user.

Assumptions and Dependencies

The software being used to develop this dashboard is Power BI. It is a software commonly used to analyze, transform, model, and visualize datasets creating reports and dashboards. It allows for multiple users, appealing to organizational settings especially those considering analysis of sales and numbers. Assumption that all users will have

company issued laptops and logins with Power BI already installed. Company has an available dataset to import, transform, model, and visualize.

General Constraints

General constraints consist of many things such as security, data repository, and network communications. This is specifically limited to the organization that will be using the software as without a dataset, the software itself won't have any usage. Security is done through verification of an employee's or employer's identity, but there are still risks of an outside attack. This leads to network communication, which is a source of this dilemma, this is because the software is a shared platform that multiple people will have access to, increasing the possibility.

Performance, standards compliance, and validation requirements are something our team is looking to develop thoroughly to meet a set standard thereby meeting user expectations.

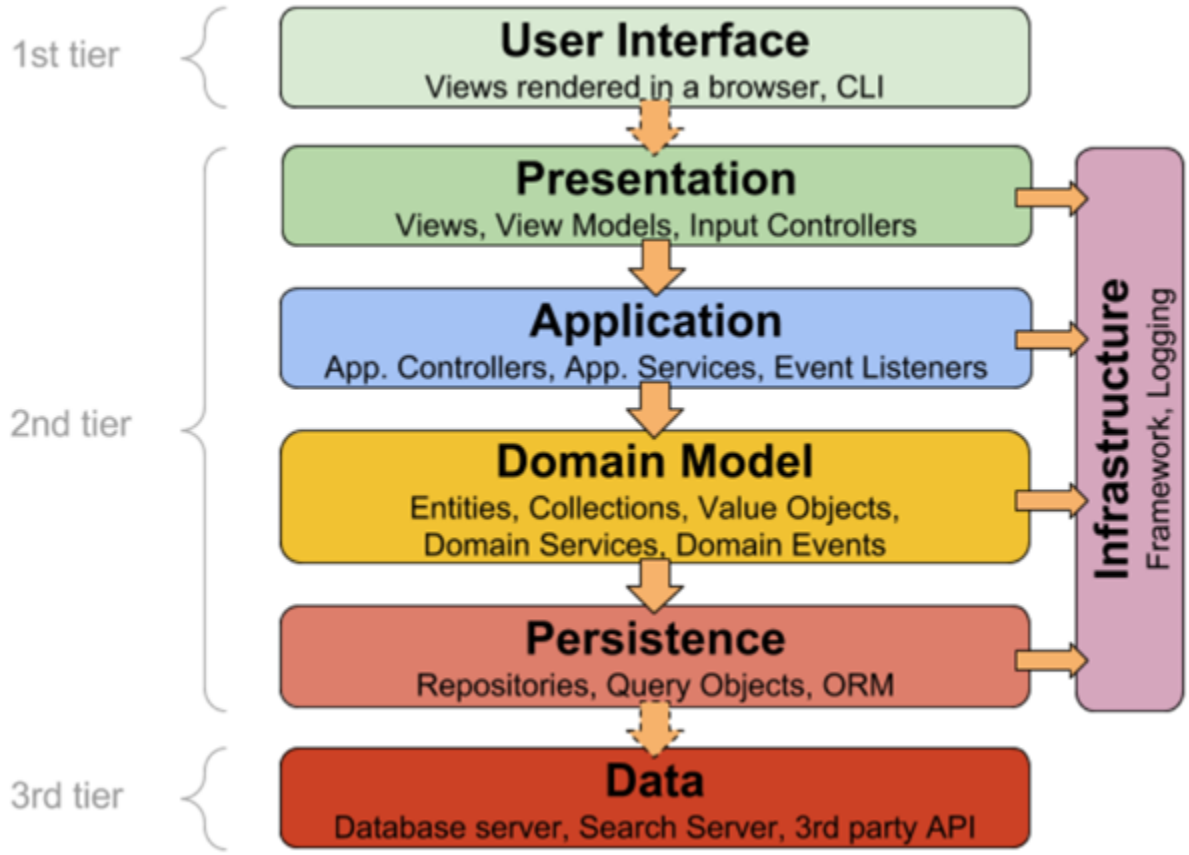
Development Methods

We plan to use the waterfall methodology which coincides with the given time. The waterfall methodology allows us to plan out in the form of documents including requirements, designs, and specifications. We enter the development phase next which includes coding and developing the software itself. Finally, testing is done before a presentation is done to further test what it'd be like for a real-world scenario. There will be a final "maintenance" which is any update or alteration done in response to the feedback of the presentation.

Architectural Strategies

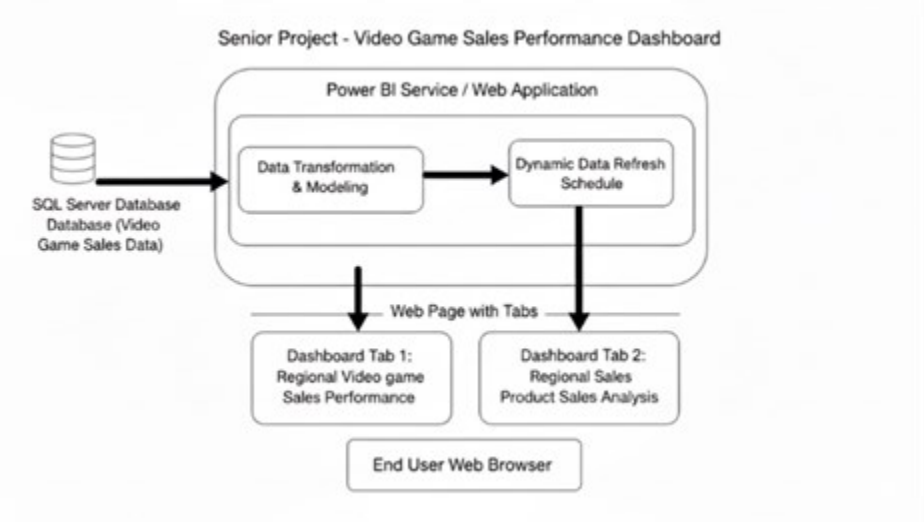
Company runs on a hierarchy or departmental based system. The software is meant to be structured so that each department has access to only what they're allowed to have access to. The strategy implemented for this project will be layered architecture. Layered architecture is a software design pattern that formats the software into multiple distinct layers allowing for specific tasks division. The reasoning for this method is because of the backend data that the program will be accessing, that data is being updated regularly and therefore should be behind. The partition of layers allows for the external database to be constantly updated, while having the infrastructure be untouched by the direct changes.

This structure has its advantages in scalability as well due to its accessibility and ease of use. Allowing the company to expand is a big part of the decision as scalability to appeal. Overall, it's not a complex structure, just a separation of user-interface, infrastructure, and the database.



We considered using a blackboard architecture, which is the shared updated knowledge of a complex problem of a data structure. In terms of Power BI and its ability to scale, transform, and model a dataset, it would be finding the answer of where the profits lie. However, we believe this model was too complex for what the program was intended to do and wasn't user-friendly to the point that a company not well versed in data science would be able to comprehend it. Simplicity is the best appeal when presenting a product that a consumer will use.

System Architecture



The sales dashboard is formulated with a multi-layered architecture, separating responsibilities related to data storage, transformation and presentation to different components of the architecture. At the system's core is a single database that is sourced for all visualizations and reporting. There is a distinct pipeline for the data to be visualized: it originates within our SQL server, is transformed and modeled within Power BI using its proprietary tooling and is shown to users as an interactive Power BI application.

The reasoning behind decoupling component responsibilities is to optimize data storage, sourcing and transforming. Relevant business information and KPIs are handled using the Power BI data model. This is to make end user logic as consistent and as easy to update as possible. Power BI also handles the services for user authorization and data distribution security.

Detailed System Design

Most components described in the System Architecture section will require a more detailed discussion. Other lower-level components and subcomponents may need to be described as well. Each subsection of this section will refer to or contain a detailed description of a system software component. The discussion provided should cover the following software component attributes:

Classification

System Architecture core component attributes:

SQL Server Database: Foundational source of data for the system. Is stored in a relational format and will provide a secure, reliable source of data for future analysis. The server will have inputs of video game transactional data such as game title, genre, region of sales, publisher, units sold, etc.

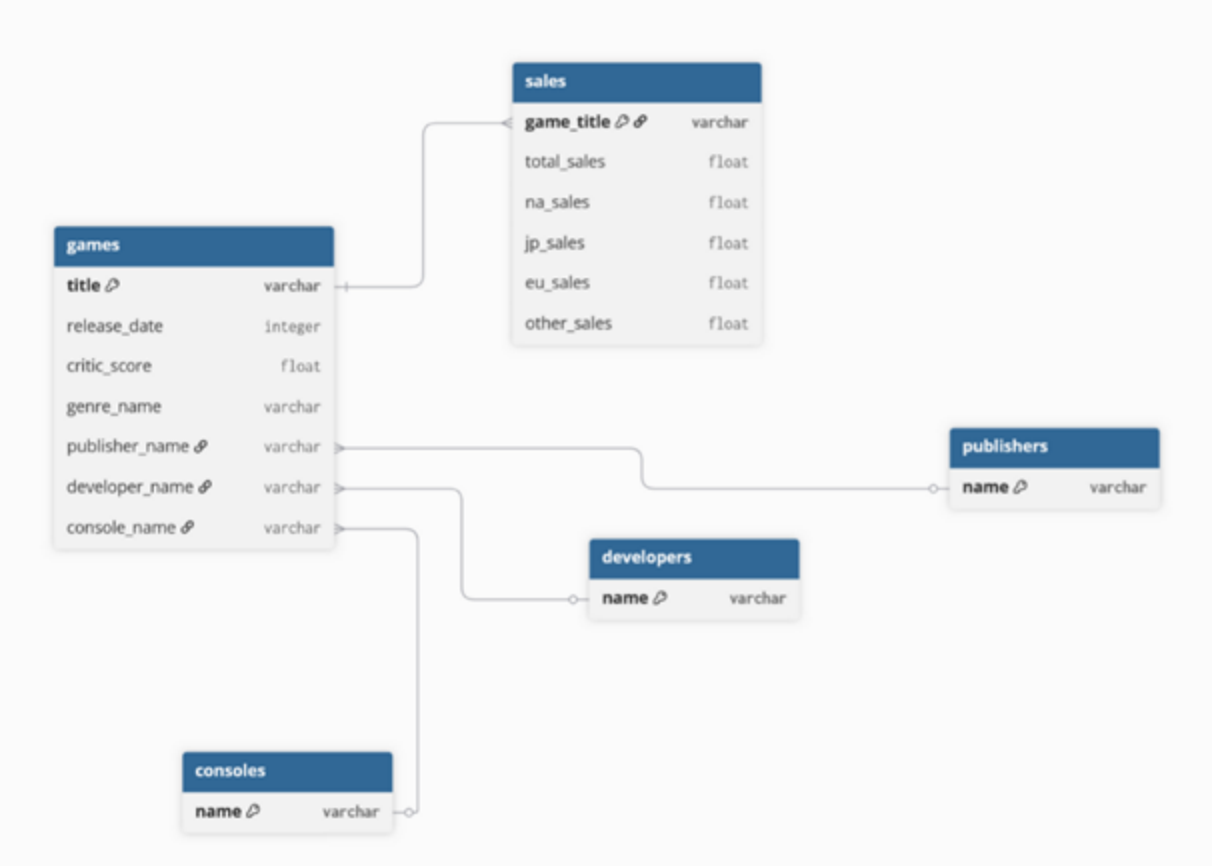
Data Transformation and Modeling: Power BI component that reengineers SQL sales data into business intelligence. This is done through cleaning the data to the point it can be properly interpreted by Power BI, applying business rules to the data, and representing logical relationships in the visualizations. Transformation occurs with the Power Query engine, whereby a sequence of ETL steps apply to it. Once the tables are transformed, they are then applied using Power Bi's data analysis expressions to create calculated columns. The output is a Power BI dataset.

Dynamic Data Refresh Schedule: A power BI service we will utilize to ensure data represented on the web page is current. This service automatically updates data, so page manual refresh is no longer required by the user with no input. This occurs at a time we designate as developers, whereby the data transformation and modeling process is repeated and refreshes the data model. A new Power BI dataset is output.

Dashboard Presentation Tabs:

Dashboard Tab 1 Regional Video Game Sales Performance: Provides a high-level view of geographical video game sales performance, with accompanying visualizations that aid sales employees in making daily determinations for inventory management and presentation. The tab will contain interactive elements such as bar and trend charts, demonstrating KPIs that are most relevant to regular game store employee duties.

Dashboard Tab 2 Regional Product Sales Analysis: Allows management to take a deeper look into video game performance by region and based on more relational factors that would be of interest to high level business decisions. This tab will also feature options for management to generate reports at the press of a button, as well as monitor some employee sales metrics.



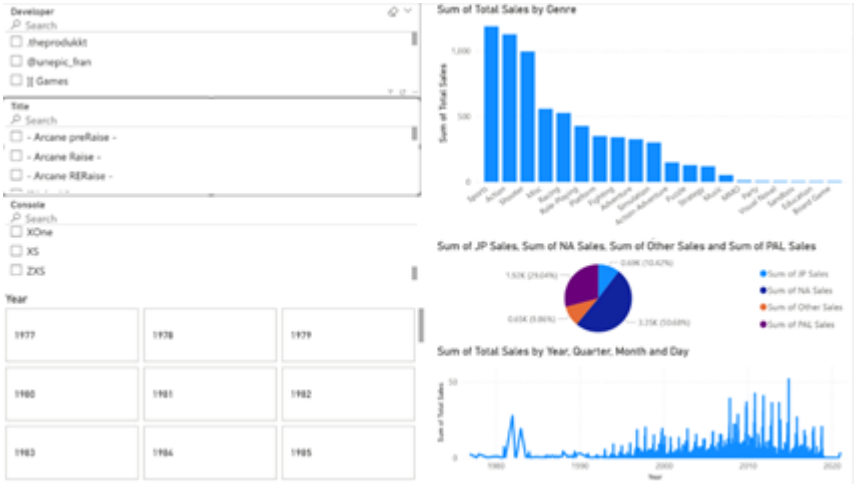
Pictured is the entity relationship diagram (ERD) of our SQL database that houses the client's video games' sales database. The database will contain tables for games, sales, consoles, developers, and publishers. Each table in the diagram labels its primary keys, foreign keys, and attributes. It displays the links between the dataset tables and the one-to-many relationships between records of different tables. The ERD provides a clean and clear blueprint for how the SQL database will organize incoming sales data and relate entries to one another.

Resources

While there will not be changes to any data using the sales performance dashboard, the application itself will need a SQL server (or equivalent database option) to pull data from. If a data pipeline is loading new data while the dashboard is refreshing, the data provided could be outdated. To fix this issue, a manual refresh of the available data can be used to provide correct data.

Interfaces/Exports

The dashboard itself will provide multiple KPI's to the user. These include the total global sales, the sales per region, the sales per title, the top-selling titles per region, sales by console, sales by genre, and sales growth over time. The figure below shows a prototype of the dashboard's interface.



As seen in the figure, the users can use filters to change the data they see. This includes a filter by developer, title, console, and year. These filters will help provide the user with a better understanding of the products based on certain criteria. For example, a user can search for the sales performance of a games made by Activision on the Xbox.

Development

This section outlines the current progress of technical development for the Power BI Sales Performance Dashboard prototype. At this stage, our focus is on documenting what has been configured and refined during early development, along with identifying upcoming steps for implementation.

Data Preparation

Dataset: *Video Game Sales 2024* (Kaggle).

The development team has completed the process of cleaning and filtering the dataset to identify primary fields needed for KPI analysis. These KPIs include total global sales, regional sales share, average sales per title, top-selling title per region, sales by console, sales by genre, and sales growth over time. This phase ensures that the dataset is consistent, accurate, and ready for model design in Power BI.

Our analysis of this data has determined that it will be used to represent global market level sales as opposed to in-store transactional sales. Hence, our dashboards will act as

strategic market analysis tools for in-store use. They ought to serve to benchmark performance, identify stocking trends, and make better, data driven business decisions.

Task Completed

- Imported all source tables into Power BI Desktop: games, publishers, developers, and consoles.
- Validated and corrected data types for primary columns.
- Established a standard for column titles to better data clarity.
- Cleaned the tables (publishers, developers, consoles) to eliminate duplicate entries.
- Unpivoted regional sales columns in the sales table into two new columns: Region and Sales. This was to normalize sales data and make filtering easier.

Upcoming Tasks

- Create a data dimension table within Power BI. This is a fundamental Power BI practice we ought to incorporate. For instance, giving attributes to the release data column such as Month, Quarter or Day of the Week.
- Create a calculated column in the games table to effectively link release_date to the new dimDate table.
- Filter out rows that contain null values.
- Remove sales records where total sales are zero.

Data Model and KPIs

The development team has loaded all data from our referenced dataset into Power BI and established relationships based on our ER Diagram. The structure we've implemented is a sort of hybrid between star and snowflake schema.

Tasks Completed

- **Data Model Design**
    - Loaded tables into the Power BI model.
    - Established relationships as defined in the ERD.
    - games[title] -> sales[game_title] (One-to-One)
    - games[publisher_name] -> publishers[name] (Many-to-One)
    - games[developer_name] -> developers[name] (Many-to-One)
    - games[console_name] -> consoles[name] (Many-to-One)
- **KPI Identification and DAX Creation**
    - Defined KPIs that would be used for real-world market analysis.
    - Established initial DAX measures for these KPIs:
    - Total Market Sales: SUM(sales[total_sales]) (or sum of the unpivoted Sales (Units)).

- Regional Market Share %: A region's sales as a percentage of Total Market Sales.
- Top N Titles by Sales: A measure to show the best-selling titles.
- Top N Genres by Sales: Identifies the most popular game categories using games[genre_name].
- Platform Market Share %: Shows the sales preference of consoles using consoles[name].
- Publisher Market Share %: Identifies dominant publishers using publishers[name].
- Year-over-Year (YoY) Sales Growth %: Tracks market trends over time (will require the new dimDate table).
- Average Critic Score: AVERAGE(games[critic_score]) to correlate score with sales.

Upcoming Tasks

- Implement Row-Level Security (RLS) to give unique views to Admin users (all sales regions) and Sales users (locked to specific regions.
- Develop Time Intelligence DAX functions to ascertain moving data (Quarter-over-Quarter Growth, Moving Averages) once the dimDate table is created.
- Create some hierarchies for drill down analysis. Ex. publishers[name] -> games[title]

Dashboard Prototype

There are two distinct user views: an Admin Dashboard for management-level insights and a Sales Dashboard for day-to-day performance tracking. The dashboard prototype focuses on usability, simplicity, and providing actionable insights to the user.

Task Completed

- Created wireframes that showcase the dashboard layout and navigation for both dashboard views.
- **Admin Dashboard**
  - Designed a high-level strategic overview page.
  - Includes KPI visualizations for Total Market Sales and YoY Sales Growth %.
  - Implemented a Regional Market Share donut chart as well as a Sales by Region map.
  - Added Sales Over Time line chart to indicate global market sales trends.
- **Sales Dashboard**
  - Designed tactical page for stocking and regional trends.
  - Implemented visuals for Top 10 Titles (by Region), Top Genres (by Region) (Tree Map), and Platform Performance (by Region) (Bar Chart).

- Added slicers for consoles[name], games[genre_name], and Year to allow sales teams to explore market niches.
- Selected and implemented a consistent color scheme and typography for clarity and readability.

Upcoming Tasks

- Refining visual design and formatting of all visuals on the dashboards.
- Implementing a navigation button panel that easily allows users to switch between Global and Regional views.
- Create custom tooltips to show detailed information when a user hovers over a visual.
- Add conditional formatting to charts where sales figures turn a certain color in relation to relative sales averages.
- Use Power BI's performance Analyzer to identify slow loading visuals or DAX functions.
- Implement accessibility features such as alternative text to visuals.
- Configure drill through pages that allow a user to right click on a specific genre and open to a hidden page containing pertinent information to that genre.
- Configure cross filtering whereby clicking on a genre filters all other visuals to that genre.

Challenges

There were several constraints and challenges that the group was faced with. The first of which was available meeting times which all five members were able to attend. Another challenge was working around the free, available options of Power BI. The school was unable to provide a license and as such, there was no way to share online between the group members about the progress of the dashboard and the work itself. This was also the first of its kind of project being done at Kennesaw State University, Power BI. As the first generation of users, we have a slight disadvantage in learning in comparison to our future successors.

Solutions

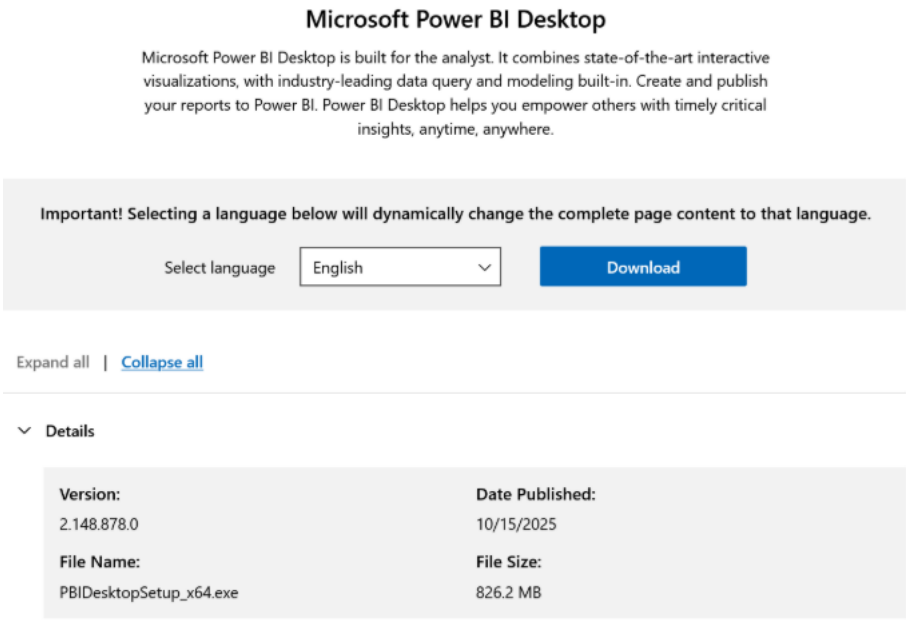
By utilizing a social media platform, known as discord, as both the site of our meetings and a way to coordinate our meetings, we killed two birds with one stone. Storing and discussing potential meeting times and being able to share with one another our thoughts made it a great tool. There was no way to get around the problem of being unable to share unless the group physically met, however, so we resorted to allowing access and copy and sharing the script itself. Finally, we worked on solutions utilizing the internet, using multiple sites as references and resources to learn, develop, and build the dashboard and this document as well.

Database Connection

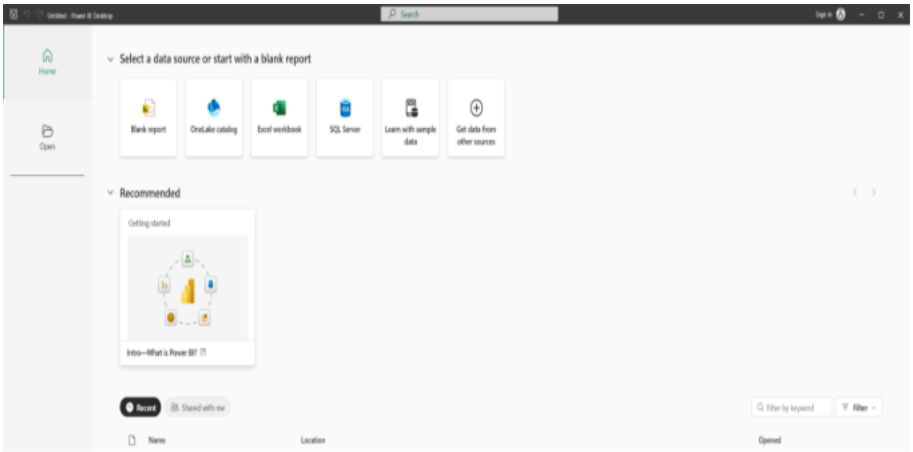
At this stage in the project, there is no database connection. The dataset is imported locally into Power BI Desktop as a CSV file. In future stages, our team is exploring connecting to an external SQL database for scalability purposes.

Project Setup

1. Navigate to the official Microsoft Power BI download page.
2. Download and install the latest Power BI Desktop application



3. Download the Video Game Sales 2024 dataset from Kaggle.
4. Verify vgchartz-2024.csv file integrity
5. Open Power BI Desktop



6. Review and clean columns from vg-chartsz-2024 via Power Query
7. Standardize column data types to a Decimal Number data type
8. Remove possible duplicate and null entries from dataset
9. Apply changes to dataset and save
10. Create data tables and establish relationships
11. Review dashboard specifications and requirements
12. Create a report with the transformed dataset
13. Define page sizes, report theme, and background
14. Visualize key categorical data, sales over time, and regional data
15. Format text, validate all the visuals and add title to all dashboard pages

Software Test Plan and Report

Test Objectives

The purpose of this Software Test Plan (STP) is to define how the Power BI Sales Performance Dashboard will be tested using our video game sales data model.

Our team's STP objectives include:

- Verify that our dashboard accurately calculates and displays the DAX KPIs including Total Market Sales, Regional Market Share %, Platform Market Share %, Publisher Market Share %, Top N Titles, Top N Genres, and YoY Sales Growth.
- Check that all filters and slicers (Region, Platform, Genre, Publisher, Year) work as expected and update visuals accurately and consistently.
- Test both dashboard views (Admin Dashboard and Sales Dashboard) to ensure their features function properly.
- Verify that visuals respond correctly to cross-filtering, highlighting, drill-down, and drill-through actions.
- Validate that unpivoted regional sales fields (NA_Sales, EU_Sales, JP_Sales, Other_Sales) update correctly when filters are applied (e.g., filtering by Genre, Year, Platform, Publisher).
- Review the dashboard's design for usability, clarity, and consistency.
- Validate the Power BI data model, including relationships built using the hybrid star and snowflake schema.
- Check and verify the data quality rules (Correct data type, no duplicates, no unexpected nulls).
- Document issues found during testing and tracking in the STR (Software Test Report) section.

The Software Test Report (STR) section will summarize the results (Pass/Fail/Severity) for each test case defined.

Scope of Testing

In-Scope

Functional Testing

DAX Measures/KPIs

- Total Market Sales
- Regional Market Share %
- Platform Market Share %
- Publisher Market Share %
- Top N Titles
- Top N Genres
- Average Critic Score
- YoY Sales Growth %

Slicers/Filters

- Year
- Region
- Platform
- Genre
- Publisher

Dashboard Features

- Admin Dashboard (potential license restriction)
- Sales Dashboard (potential license restriction)
- Cross-filter interactions between visuals
- Drill-through behavior (Genre, Platform, Title)

Non-Functional Testing

- Performance (load times for visuals using Power BI built-in Performance Analyzer)
- Dataset scalability
- Consistent formatting
- Usability (text readability, layout on laptop/desktop screens)

Out-of-Scope

- SQL database connectivity (license restriction)
- Multi-user Power BI collaboration (license restriction)
- Dashboard user security (license restriction)

Test Cases

Functional Test Cases

Test Case ID	Description	Expected Result

FTC-01	Global Sales KPI Verification	The dashboard's displayed 'Total Global Sales' must match the source dataset.
FTC-02	Region Filtering	Selecting a specific region such as JP(Japan) must not return data for external regions.
FTC-03	Data Sum Logic	Verify that [JP Sales] + [NA Sales] + [PAL Sales] + [Other Sales] equals the total sold for each title.
FTC-04	Genre Slicing	Select a certain genre such as Action and all visuals should only display Action game data.
FTC-05	Duplicate Title Sales	Validate the manner the dashboard displays the same title sold and released on separate gaming platforms.
FTC-06	Year Slicing	Select a year and ensure that any sales data outside of this range does not appear.
FTC-07	Missing Values Handling	Ensure that games with missing data fields aren't displayed at all

Non-Functional Test Cases

Test Case ID	**Description**	**Expected Result**
NFTC-01	Dataset Load Time	Dashboard should load and display all visuals within a reasonable amount of time (<4-5 seconds).
NFTC-02	Slicer Load Time	Data visualizations should automatically and quickly update upon the selection of filters on the data.
NFTC-03	Blank State Display	Ensure that selecting a combination of filters that

		contradict the visuals simply display no data and don't break.
NFTC-04	Browser Functionality	Dashboards should be fully functional and identical when displayed on different browsers such as Edge, Chrome, or Firefox.
NFTC-05	Visual Color Scheme	Categorical colors should be consistent, and dashboards should maintain a theme that is easy on the eyes.
NFTC-06	Default Data Sort	Data should be immediately loaded into default displays of every category or filter.

Test Procedures

FTC-01:

- Step 1: Open the Power BI Sales Dashboard
- Step 2: Locate 'Total Global Sales'
- Step 3: Open the source dataset
- Expected Result: 'Total Global Sales' matches the dataset

FTC-02:

- Step 1: Open the Power BI Sales Dashboard
- Step 2: Select a region
- Step 3: Cross-Reference
- Expected Result: Other regional data doesn't appear in selected region
- Regional data should be present

FTC-03:

- Step 1: Open the Power BI Sales Dashboard
- Step 2: Select a title
- Step 3: Verify sales numbers for each region
- Step 4: Manually add sum and compare to total sum
- Expected Result: Total sum should equal displayed total sum for the selected title

FTC-04:

- Step 1: Open the Power BI Sales Dashboard
- Step 2: Select a genre
- Step 3: Verify the genre contains the genre visuals

- Expected Result: Only the selected genre should be displayed

FTC-05:

- Step 1: Open the Power BI Sales Dashboard
- Step 2: Select a title
- Step 3: Select title splicer
- Step 4: Cross-verify with the dataset
- Expected Result: Matching Results with the dataset

FTC-06:

- Step 1: Open the Power BI Sales Dashboard
- Step 2: Select year; button splicer
- Step 3: Can select multiple years to see visuals from each of the years' sales
- Expected Result: Displaying only selected years' sales

FTC-07:

- Step 1: Open the Power BI Sales Dashboard
- Step 2: Check dataset for any missing values or Null values
- Step 3: Locate titles with missing values or Null values in the dashboard
- Expected Result: Title doesn't appear at all

NFTC-01:

- Step 1: Open the Power BI Sales Dashboard
- Step 2: Start a timer for 4-5 seconds
- Step 3: Observe dashboard
- Expected Result: Dashboard should appear with visuals within timer range

NFTC-02:

- Step 1: Open the Power BI Sales Dashboard
- Step 2: Select a slicer
- Step 3: Visualization should update based off selected slicer
- Expected Result: New visualization should match selected slicer and be updated within a reasonable time, expectedly within a 2 second time frame

NFTC-03:

- Step 1: Open the Power BI Sales Dashboard
- Step 2: Select a combination of slicers that will result in no matches
- Step 3: Confirm no visuals

- Expected Result: There should be a blank slate, no visuals, no errors

NFTC-04:

- Step 1: Open the Power BI Sales Dashboard on one browser
- Step 2: Open the Power BI Sales Dashboard on a different browser
- Step 3: Compare layout between the two browsers
- Expected Result: Identical layout and be fully functional

NFTC-05:

- Step 1: Open the Power BI Sales Dashboard
- Step 2: Check charts, select splicers
- Step 3: Check Power BI Sales Dashboard across other browsers
- Expected Result: Consistency in color scheme all throughout, color should be soft

NFTC-06:

- Step 1: Open the Power BI Sales Dashboard
- Step 2: Load Dashboard with dataset
- Step 3: Examine tables and charts
- Expected Result: Displayed should be Sum of total sales in general, no splicers checked

Test Environment

This section will define the hardware, software, and network environments used to execute the test procedures for the Power BI Sales Performance Dashboard.

Hardware Environment

Testing will be performed on standard desktop hardware that meets and/or exceeds Microsoft's recommended specifications for running Power BI
Desktop. The hardware environment incudes:

- Operating System: Windows 10 or Windows 11 (Required for Power BI Desktop)
- Processor: Multi-core processor
- Memory: Minimum of 2 GB RAM, 4 or more GB RAM recommended
- Storage: Local disk space allocated for dataset files, PBIX files, etc.
- Display: 1080p resolution (to verify visuals and layouts)

Software Environment

The software environment includes:

- Power BI Desktop (latest version)

    - Installed from official Microsoft download page: https://www.microsoft.com/en-us/download/details.aspx?id=58494
- Power Query Editor (included in Power BI Desktop service)
- Optional: OneDrive and/or GitHub for PBIX file version control

Power BI Desktop is the main application for both development and testing since the project scope does not currently use any cloud-based features (Power BI Service or shared workspaces).

Network Environment

The dashboard uses a locally imported CSV dataset, so the testing phase does not rely on a network environment. Power BI Desktop runs offline and there are no external databases and/or APIs connected.

Network access is only needed for the following:

- Downloading Power BI Desktop and updates
- Downloading the dataset from Kaggle
- Accessing online documentation

There are no network performance expectations since the testing phase will occur locally.

Test Data

The testing phase uses the cleaned and prepared file version of the “Video Game Sales 2024” dataset that was pulled from Kaggle and imported into Power BI Desktop. The dataset includes standard data types, removed duplicates and nulls, etc. to ensure consistency and reliability for all test cases.

Dataset

The dataset testing relies on the following tables:

- games
- sales (Regional/Sales fields)
- publishers
- developers
- consoles

These tables serve as a model for the KPI calculations and visuals.

KPI Data

Our team used Power BI’s built-in tools for the data view, table visualization, and DAX tools to manually test and verify the following KPIs:

- Total Market Share
- Sales by Region
- Sales by Platform
- Top Titles
- Top Genres
- Market Share % (Regional/Platform/Publisher)

These KPIs were calculated and manually tested directly in Power BI to ensure accuracy for each DAX measure, as well as ensure the actual dashboard outputs are correct.

Filter and Slicer Data

Specific subsets of the dataset (such as a specific Region, Platform, Genre, or Year) are used to verify the accuracy of how our slicers, filters, and drill-throughs behave. These smaller, more specific subsets make testing more efficient as our team can confirm that visuals update as intended when filters are applied, drill-through pages display the intended information, etc.

Edge Case Data

Edge case data is tested to ensure the dashboard returns expected results in scenarios that are not common or are unexpected. The test data includes the following edge case scenarios:

- Filters that return very small or no data
- Titles that have low sales.
- Titles that are apart of uncommon genres.

Testing these scenarios helps to verify that the dashboard will behave correctly even when the data is not in the typical ranges.

Software Test Report (STR)

Test Case Results

Test Case ID	Description	Result	Severity	Notes
FTC-01	Global Sales KPI	Pass	Critical	Displayed total matches source CSV exactly.
FTC-02	Region Filtering	Pass	Critical	JP selection successfully hides NA/EU/Other regional data.
FTC-03	Data Sum Logic	Pass	Critical	Regional sums equal Global sum for tested titles.
FTC-04	Genre Slicing	Pass	High	No cross contamination between visuals during filtering.

FTC-05	Duplicate Title Sales	Pass	Moderate	Multi-platform titles are aggregated correctly.
FTC-06	Year Slicing	Pass	High	Date ranges are filtered accurately as well as outside years being hidden.
FTC-07	Missing Values Handling	Pass	Moderate	Nulls are either filtered out or categorized as unknown
NFTC-01	Dataset Load Time	Pass	Moderate	Average load time is less than 4 seconds.
NFTC-02	Slicer Load Time	Pass	Moderate	Visuals update within 1 – 2 seconds maximum.
NFTC-03	Blank State Display	Pass	Low	Visuals can return blank clearly without any application ending errors.
NFTC-04	Browser Functionality	Pass	Moderate	The layout on multiple browsers (chrome, edge, Firefox) is verified to be identical.
NFTC-05	Visual Color Scheme	Pass	Low	Categorical colors remain consistent across pages.
NFTC-06	Default Data Sort	Pass	Low	Tables sort by Sales (Desc) immediately upon load.

Summary of Results

The testing phase for our application confirms that the software meets both the functional and non-functional requirements that were defined in our scope.

Key Findings:

- All DAX measures and KPIs (FTC-01 - FTC-03) resulted in 100% accuracy when compared to the source Kaggle dataset.
- Interactions involving slicers and cross-filtering (FTC-02, FTC-04, FTC-06) functioned with minimal latency and yielded a smooth user experience.
- Reports load within our stated acceptable range of 5 seconds on standard hardware. This validates performance on all benchmarks.
- Our data cleaning process was successful in removing duplicate entries and handling null values. Edge case testing yielded no calculation errors.

No significant bugs were found during our implementation of the test plan. It can be concluded that the dashboard is stable, accurate in results, and prepared for deployment.

Version Control

Because the Sales Performance Dashboard project did not involve source code files, we relied on Microsoft's OneDrive and other standard file sharing methods for version control rather than GitHub. Although GitHub was originally included in the Project Plan, we confirmed with the project advisor, Professor Sharon Perry, that it was not necessary for this specific PBIX-based project. In addition, the free Power BI licenses provided by the course did not support collaborative editing or a shared workspace, which meant the dashboard file could not be worked on at the same time. As a result, we focused on maintaining a single, up-to-date version of the PBIX file and documenting any changes in a separate document.

Due to licensing constraints, only one team member (Alex Thomas) had access to edit the PBIX file throughout development. The PBIX was stored in Alex's OneDrive, and any updates to the file were made exclusively through his Power BI Desktop environment. Alex would send the PBIX files manually for the team to review. When other team members suggested changes to the dashboard, it would be communicated to Alex through Discord for implementation. Having a single team member editing the file prevented any version of conflicts and ensured there was only one up-to-date copy of the dashboard.

All other project materials (including the Project Plan, SRS, SDD, Development Document, Prototype Presentation, STP/STR, Gantt Chart, Meeting Notes, and Final Report) were stored in a shared OneDrive folder that was accessible to all team members. This allowed us to collaborate on all written deliverables, while keeping the dashboard file separate to avoid versioning issues. OneDrive's version history was a useful tool for our deliverables as it allowed us to track changes, recover earlier draft versions, and view each document's progress.

We also relied on Discord for our day-to-day communication and updates about any file changes, task assignments, and project/deliverable adjustments. We would schedule and attend voice chat meetings which kept everyone on the same page. Discord was also a useful communication tool because we were able to have separate channels for meeting notes, questions, documents, and general discussions. Our team also used Microsoft Teams to have status meetings with Professor Perry. These meetings helped our team to align project expectations, next steps, and remain on the course schedule.

Conclusion

The SP-115 Red Team's Sales Performance Dashboard project demonstrated our ability to apply the Software Development Life Cycle to a real-world business intelligence scenario by transforming a raw video game sales dataset into an interactive and user-friendly analytics tool. Through extensive data preparation, modeling, DAX development, and iterative dashboard design, we were able to create a system that provides meaningful KPIs and visual insights for sales users. Despite constraints related to Power BI licensing, collaboration limitations, and team scheduling challenges, our group maintained an effective workflow using OneDrive for version control and Discord for communication, ensuring consistent progress and alignment. The final dashboard meets the project's planned requirements and provides clear, accessible visualizations that support data-driven decision-making. Overall, this project strengthened our understanding of Power BI, enhanced our technical and collaborative skills, and demonstrated the value of intentional design and structured development when creating business intelligence solutions.

Appendix – Project Plan

1. Project Overview / Abstract

Development of an interactive Power BI dashboard to analyze retail sales data across regions, time periods, and product categories. The primary objective is to design and implement key performance indicators (KPIs) that help identify trends and opportunities to increase profits based on historical and real-time data. The scope includes visualization, advanced data modeling, scalability, safety, and accurate information.

1.1 Extended Scope

The dashboard will feature dynamic data visualizations to track KPIs over time, enabling more informed and strategic decision-making. This can be implemented using Data Forecasting, a well-known technique using historical data sets to predict.

1.2 Platform

Microsoft Power BI is a powerful business analytics tool renowned for its interactive visualizations. Key features like data cleaning, transformation, and modeling are integral to the success of this project. Additionally, Power BI's collaborative capabilities enable users to easily share dashboards, making it a highly user-friendly platform.

2.0 Project Website

Our team will develop and maintain a project website to showcase documentation, deliverables, and our final product(s). The website linked here is a placeholder site that will be updated later in a separate project phase: https://sites.google.com/view/sp115-red-salesperformancedashboard.
The website will be accessible to the public and not require any form of authorization.

3.0 Deliverables

The following deliverables will be developed and submitted during this project:

Deliverable	Planned Date
Project Selection	August 28, 2025
Project Plan and Gantt Chart	August 31, 2025
Software Requirements Document (SRS)	September 14, 2025
Software Design Document (SDD)	September 14, 2025
Development Document	October 26, 2025
Prototype Presentation	November 6, 2025
Software Test Plan (STP)	November 30, 2025

Software Test Report (STR)	November 30, 2025
Draft Report	November 10 – December 7, 2025
Final Report Package	December 8, 2025

4.0 Group Meeting Schedule Date/Time

The team will hold weekly status meetings with the project advisor, Professor Perry, on Tuesdays from 11:15-11:30 AM. In these weekly status meetings, the team will provide updates, address any questions, and discuss feedback. Additional meetings will be scheduled on Discord or in person as needed to collaborate on deliverables and coordinate progress. A documentation team member will take notes for each meeting, and they will be stored in a shared OneNote notebook within the team's shared OneDrive folder.

5.0 Collaboration and Communication Plan

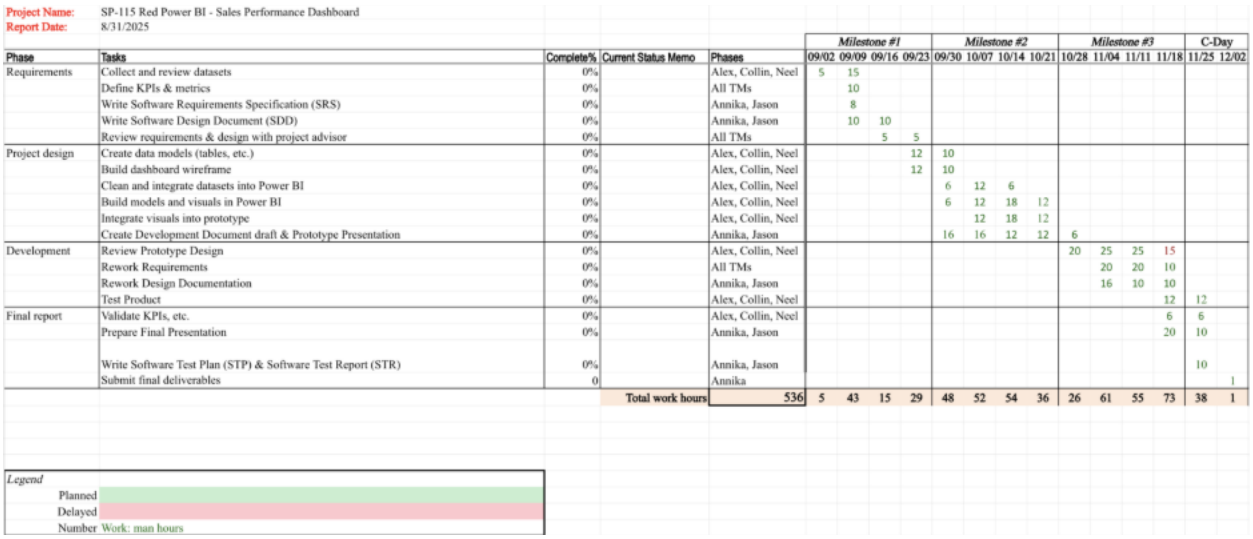
Our team will use the following collaboration/communication tools:

- Microsoft Teams: Weekly status meetings with the project advisor, Professor Perry.
- Discord: Daily communication, coordination, and virtual meetings.
- OneDrive: Document storage, collaborative editing, and version history tracking.
- Cell Phones: Urgent communication.

These tools will help our team stay organized and ensure our deliverables progress on schedule.

6.0 Project Schedule and Task Planning

The production of the Sales Performance Dashboard using Power BI will be structured around the phases of the software development lifecycle (SDLC). The phases include requirements, design, development, and testing. Throughout the project's development, our team will work through dataset collection/validation, defining KPIs, dashboard development, documentation, and final presentation preparation. The project responsibilities will be distributed across the team based on roles (team leader, documentation, and development). The project progress will be tracked and reviewed through a weekly status meeting with our project advisor/facilitator, Professor Perry.
Our team has created a Gantt chart to estimate hours for each phase of the project and visualize tasks. A separate copy of this Gantt chart will be maintained throughout the project for planning and tracking purposes.



Project Name: SP-115 Red Power BI - Sales Performance Dashboard
Report Date: 8/31/2025

Phase	Tasks	Complete%	Current Status Memo	Phases	Milestone #1 09/02	09/09	09/16	09/23	Milestone #2 09/30	10/07	10/14	10/21	Milestone #3 10/28	11/04	11/11	11/18	C-Day 11/25	12/02
Requirements	Collect and review datasets	0%		Alex, Collin, Neel	5	15												
	Define KPIs & metrics	0%		All TMs		10												
	Write Software Requirements Specification (SRS)	0%		Annika, Jason		8												
	Write Software Design Document (SDD)	0%		Annika, Jason		10	10											
	Review requirements & design with project advisor	0%		All TMs			5	5										
Project design	Create data models (tables, etc.)	0%		Alex, Collin, Neel				12	10									
	Build dashboard wireframe	0%		Alex, Collin, Neel				12	10									
	Clean and integrate datasets into Power BI	0%		Alex, Collin, Neel					6	12	6							
	Build models and visuals in Power BI	0%		Alex, Collin, Neel					6	12	18	12						
	Integrate visuals into prototype	0%		Alex, Collin, Neel						12	18	12						
	Create Development Document draft & Prototype Presentation	0%		Annika, Jason					16	16	12	12	6					
Development	Review Prototype Design	0%		Alex, Collin, Neel									20	25	25	15		
	Rework Requirements	0%		All TMs										20	20	10		
	Rework Design Documentation	0%		Annika, Jason										16	10	10		
	Test Product	0%		Alex, Collin, Neel												12	12	
Final report	Validate KPIs, etc.	0%		Alex, Collin, Neel												6	6	
	Prepare Final Presentation	0%		Annika, Jason												20	10	
	Write Software Test Plan (STP) & Software Test Report (STR)	0%		Annika, Jason													10	
	Submit final deliverables	0		Annika														1
			Total work hours	536	5	43	15	29	48	52	54	36	26	61	55	73	38	1

Legend
Planned
Delayed
Number Work: man hours

The key milestones for this project are:

Milestone	**Planned Date**	**Owner(s)**
Project Selection	August 28, 2025	All Team Members
Project Plan and Gantt Chart	August 31, 2025	Annika (Team Lead, Doc), Jason (Doc)
SRS and SDD (Requirements Specifications)	September 14, 2025	Collin Knowles (Dev), Neel Ranawat (Dev), Alex Thomas (Dev)
Integrate Datasets	September 21, 2025	Annika (Team Lead, Doc), Jason (Doc)
Development	October 21, 2025	Collin Knowles (Dev), Neel Ranawat (Dev), Alex Thomas (Dev)
Prototype Presentation	November 6, 2025	Annika (Team Lead, Doc), Jason (Doc)
Testing Phase	November 25, 2025	Collin Knowles (Dev), Neel Ranawat (Dev), Alex Thomas (Dev)
Final Report	December 8, 2025	Annika (Team Lead, Doc), Jason (Doc)

7.0 Risk Assessment

Our team has identified potentials risks that could impact our project's success, including:

- Data quality issues within the datasets.
- Scheduling conflicts with team members.
- Technical challenges when integrating datasets into Power BI.

To help mitigate risks, our team will validate the datasets used, schedule weekly status checks, and seek feedback from Professor Perry (and other necessary resources). Using these risk mitigation strategies, we will keep the project on schedule, minimize potential issues, and ensure our project's success. Using online resources and previously known data, we will create a dashboard uniquely ours, avoiding risks of the unknown.

8.0 Version Control Plan

The team will use a shared GitHub organization account to serve as a central repository for source code, Power BI project files, and any additional scripts. Our team will follow a clear branching strategy and consistent commit message format to promote traceability and organization. All project documents and submissions will be stored in a shared OneDrive folder, where version history will be used to track updates. As a result, GitHub and OneDrive will provide effective version control for our team's technical and non-technical deliverables.