

Software Design Document (SDD)

Team Name: Agustin, Matthew, Damian, Miguel, and Charles

Group Number: 11

Date: November 30, 2025

Contents

1	Introduction	3
1.1	Purpose	3
1.2	Intended Audience	3
1.3	Overview	3
2	System Architecture	3
2.1	Workflow of the System	3
2.2	Component Breakdown	4
3	User Interface	5
3.1	How to Use the System	5
3.2	Database Design	5
4	Glossary	7
5	References	7

Version Description

Version	Description	Date Added
1.0	Initial release for Snapshot 1	November 30, 2025

1 Introduction

1.1 Purpose

The purpose of a Software Design Document is to serve as a blueprint that converts high-level requirements into a technical roadmap for a development team. It will detail the software's architecture, data design, and specific technical decisions, ensuring everyone on the team has a shared understanding that reduces confusion, speeds up development, and eases onboarding for new members.

1.2 Intended Audience

The primary audiences for an SDD document are technical and business stakeholders who need a shared understanding of the system's architecture and design before implementations begin. Some of these technical audiences consist of Software Architects, Developers or Programmers, QA Engineers or Testers, and Project Managers.

1.3 Overview

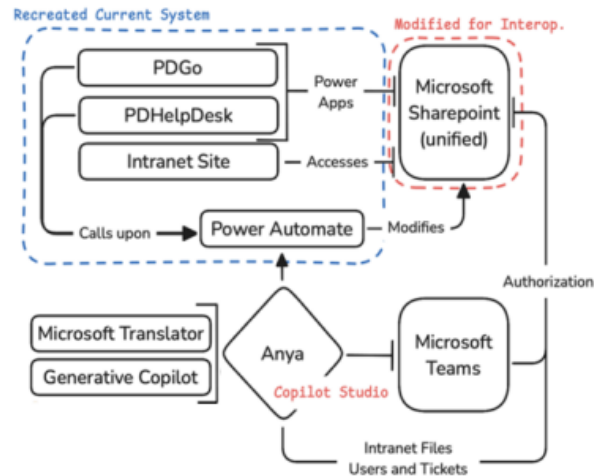
It's an AI called Anya that is operated within the Microsoft Teams environment using Microsoft Power Apps and Copilot Studio. It will allow users to submit and track IT and administrative tickets, retrieve files, retrieve contacts, translate documents, fill documents, and interact with internal systems via commands. All the data is facilitated through SharePoint and Smartsheet.

2 System Architecture

2.1 Workflow of the System

The flow of the Project Mirage system consists of the following:

- User Interface Subsystem: Manages all interaction between users and Microsoft Teams by presenting interactive components and capturing user responses.
- Conversational Engine: Built with Copilot Studio, this component interprets user input written in natural language and routes requests to the appropriate system modules.
- Session Management Subsystem: Monitors user sessions and maintains activity information, such as associating support tickets with the correct user.
- Data Services Layer: Facilitates secure communication between Anya and external platforms, including SharePoint Lists and Smartsheet, through APIs to ensure reliable and protected data transfer.



2.2 Component Breakdown

The Project Mirage software is a standalone, desktop command-line application and does not use traditional client-server architecture or an external relational database. All processing is local to the machine where the software is executed. Some architectural and Components include:

- Anya is developed using Microsoft Power Platform tools, including PowerApps and Copilot Studio. These technologies were selected due to their strong compatibility with Microsoft 365 and their support for rapid application development with minimal coding. Anya is deployed within Microsoft Teams, allowing users to access it through a platform they already use daily.
- Rather than developing new systems from the ground up, Anya builds on established tools such as PDHelpDesk and PDGo for ticket processing and administrative workflows. This method increases efficiency by avoiding duplication and ensuring smooth system integration.
- Anya communicates with users through a chat-based interface. This design makes the system easy to learn, lowers training requirements, and improves user engagement. A form-based interaction model was evaluated but ultimately not used, as chat-based interaction was more flexible and intuitive.
- The system is built using a modular architecture, where features such as ticket submission and file retrieval operate as independent components. This setup allows upgrades or modifications to be made without affecting the entire system. Additional features can be added smoothly as requirements change.
- Anya is capable of identifying issues such as invalid input, missing permissions, or broken links. The system provides clear feedback and instructions when errors occur. Users can restart sessions if problems continue. Logs are maintained to assist with troubleshooting and system maintenance.

- All system data is stored using SharePoint Lists and Smartsheet according to Santa Barbara County's security guidelines. Information is protected through encryption both during transmission and while stored. Indexing and session tracking are implemented for fast data access and reliable records management.
- A standard web-based support portal was reviewed but not selected due to poor integration with Microsoft Teams. Fully custom-built systems were also ruled out because they would require more development time and greater maintenance effort.

3 User Interface

3.1 How to Use the System

Here is a rephrased version of that section with different wording and sentence structure while keeping the original meaning:

1. From the user's perspective, Anya functions as an AI-powered virtual assistant embedded directly in Microsoft Teams that helps automate common workplace tasks. Users interact with Anya through simple, natural language messages such as "I need IT help" or "Submit a leave request." When a request is made, Anya displays interactive prompts that guide users through actions like opening support tickets, submitting administrative requests, or searching for internal files. The system is designed to be intuitive and does not require technical training, while also supporting shortcut phrases for frequently used commands.
2. Anya communicates in a conversational manner and adapts based on the user's input. If a message is unclear, the system prompts the user for clarification or suggests appropriate options. It also uses profile information to personalize responses and automatically routes tasks such as form processing or document access based on the user's department or role. To ensure accessibility, Anya supports screen readers and high-contrast modes, meeting ADA compliance standards.
3. Users receive clear visual confirmation of their actions through success messages, assigned ticket numbers, progress updates, and result summaries for file searches or contacts. Meanwhile, administrators and IT staff manage operations using PowerApps dashboards and SharePoint lists on the backend. This user-centered design allows staff to complete daily tasks efficiently without leaving the Teams platform.

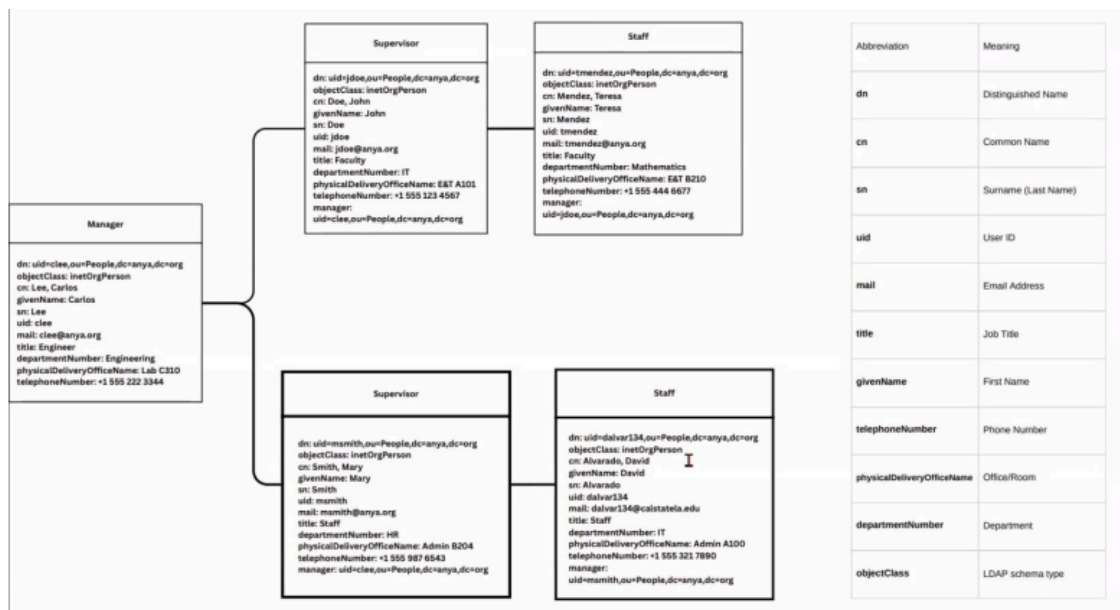
3.2 Database Design

This section provides a high-level description of the main software components in Anya's architecture and their roles within the system.

- Collects user input and displays responses within Microsoft Teams. Works only for authenticated users and supports desktop and mobile layouts via Teams. Uses Adaptive Cards and input fields to send data to the conversational engine.

- Interprets user messages using natural language processing. Routes commands to the correct module (tickets, admin requests, or file search). Uses Copilot Studio for intent recognition and error handling.
- Creates and tracks IT tickets using PDHelpDesk. Stores ticket data in SharePoint. Sends confirmation and status updates to users.
- Manages administrative tasks such as time-off requests and MCLE submissions. Works with Smartsheet and SharePoint through PDGo. Routes requests for approval and returns status updates.
- Searches for files across SharePoint and Smartsheet using keywords. Filters results based on user access level. Returns matched documents or download links.
- Tracks user activity and session details. Records timestamps and user actions in SharePoint logs. Supports system auditing and troubleshooting.
- Handles secure communication with external systems. Ensures data validation and retry logic for failed requests. Integrates with Microsoft and Smartsheet APIs and LDAP.

Overall the components are modular and loosely connected, it also improves system reliability, security, and maintainability. Finally, it supports future upgrades and feature expansion.



4 Glossary

Acronym	Definition
SDD	Software Design Document
CLI	Command-line Interface
API	Application Programming Interface
LDAP	Lightweight Directory Access Protocol

5 References

"Project Mirage", <https://ascent.cysun.org/project/project/view/220>