Software Requirement Specification (SRS)

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

Group Number: 11

Date: November 30, 2025

Contents

Version Description

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

1 Introduction

1.1 Purpose

The main purpose of a Software Requirement Specification document is to serve as a detailed, structured guide that describes a software system's purpose, features, and requirements, acting as a blueprint for development and a contract between stakeholders. It will minimize misunderstandings by providing a single source of truth for what the software must do, and it will also ensure the final product meets business and user needs.

1.2 Intended Audience

The primary audiences for an SRS document are diverse, including technical team members such as developers and testers, project management staff, and non-technical stakeholders such as clients, end users, and leadership. For this document to reach this broad audience, the document needs to be clear, well-structured, and to define terms appropriately so that everyone can understand the project's requirements.

1.3 Overview

The software "Project Mirage" is a command-line utility designed for organizing, processing, and culling photos and videos. The software organizes photos and videos from specified directories and their sub-directories. It supports multiple file formats, including JPEG, PNG, HEIC/HEIF, MP4, and MOV.

2 External Interface Requirements

2.1 User Interface

Users will interact with the Project Mirage software via a CLI. The software is run as a Python utility, where users execute specific commands and provide arguments to control its operations.

- The users will launch the program from a terminal or command prompt, and it will invoke different functionalities such as organizing files, detecting duplicates, or performing aesthetic assessment.
- Users specify input directories, desired output configurations (like sorting by date), and other operational parameters using various flags and arguments within the command.
- A key interactive feature is the ability to perform a "dry run". This allows users to preview the exact changes the software would make to their files and directories without actually implementing them, enabling confirmation before committing to any permanent modifications.
- The system provides feedback in the console, informing the user about the files processed, duplicates found, and the actions taken (or proposed, during a dry run).

2.2 Software Interfaces

Anya communicates with external platforms such as SharePoint and Smartsheet through secure API connections to ensure reliable data transfer. The Data Services Layer manages these integrations by acting as an interface between Anya and its external data sources, including SharePoint Lists and Smartsheet. The system relies on services such as the Microsoft API, Smartsheet API, and LDAP for authentication and data access. Only approved SBPD personnel are permitted to use Anya within Microsoft Teams.

System limitations include licensing restrictions imposed by PowerApps and Microsoft 365, which may reduce the availability of certain integrations. Application performance may vary depending on network speed and device capability. Custom development is constrained by the connectors and components supported by the PowerApps platform.

External tools such as Smartsheet are integrated using modular Power Automate workflows. Additionally, a separation layer exists between Copilot Studio's intent recognition and the application logic, making the system easier to modify or extend without requiring major code changes.

3 Legal and Ethical Considerations

3.1 Data Storage and Privacy

This section explains the lower-level components that operate underneath the higher-level modules described in Section 6. It focuses on key classes, scripts, and data structures that enable functionality across the system. Each component is described in terms of its role, processing behavior, interface characteristics, and internal structure, including any constraints or design considerations that apply.

The Adaptive Card Renderer functions as a front-end user interface component within Microsoft Teams through Copilot Studio. Its purpose is to generate and display structured Adaptive Cards in response to system actions. These cards are used to present information such as support ticket details, administrative requests, or file search results to users in a consistent and readable format. The renderer does not expose a direct interface or internal processing logic, as it relies on predefined templates managed by Copilot Studio.

The Power Automate Flow Handler serves as the automation and integration layer within the system and is implemented using Power Automate. It is responsible for managing business logic, coordinating interactions between system modules, and handling communication with external data sources. Actions within this component are triggered by instructions sent from Copilot Studio. The design is script-based rather than object-oriented, meaning it follows a flat structure without hierarchy. A limitation of this component is that it cannot process long-duration tasks or perform parallel operations.

The SharePoint Ticket List is a structured SharePoint list that serves as Anya's primary data store for support tickets. It contains fields such as ticket ID, category, user information, status, and submission date. This list acts as the backend system for the TicketManager module. Access control is enforced using Azure Active Directory roles, ensuring that only authorized users can view or modify ticket data. Since it uses a flat list model, it does not implement class hierarchies.

The Session Logger, implemented as a logging utility within PowerApps and SharePoint, is responsible for capturing audit information about user interactions. It records details such as user ID, action type, timestamp, and execution status. The logger provides confirmation messages when logging is successful and error records if issues occur. It operates as a single function without inheritance and must execute within the active application session. Logging is treated as a low-priority background task and runs asynchronously to avoid affecting application performance.

Together, these components form the technical foundation that supports Anya's core functionality. They provide essential services such as user interface rendering, workflow automation, secure data storage, and activity logging. Their design promotes easier maintenance, improved testability, and future scalability for continued development within the Microsoft Power Platform environment.

3.2 Ethical Issues

Project Mirage will handle and manage information only when appropriate authorization has been granted and in full compliance with all relevant privacy regulations and organizational standards. During initial login, all users are required to formally accept the system's terms of use, which clearly describe how information is handled.

The system processes specific categories of data, including case-related details, system usage records, uploaded files, and internal communications. This information is utilized for operational purposes such as managing cases, automating routine tasks, maintaining records, and supporting reporting or analysis functions. All data is maintained within secure local environments, encrypted storage systems, or trusted cloud platforms approved by the organization. Access to this information is strictly limited to approved personnel, system administrators, and authorized services. Records of user consent are stored electronically in a secure environment to support compliance and auditing activities.

When handling client-related information, *Mirage* enforces the following safeguards:

- Required permissions are secured before storing or transmitting client data electronically.

- Confidentiality obligations, including attorney-client privilege, are upheld throughout all system operations.
- Client information is not disclosed to any external party unless supported by lawful authority or formal agreements.

Users are also entitled to the following rights:

- The ability to view records that contain their personal information.
- The ability to request updates or corrections to incomplete or incorrect data.
- The right to request removal or limitation of their data, subject to legal constraints.
- Notification if their data is involved in a security or privacy incident.

Project Mirage recognizes that automated and AI-assisted systems may introduce unintended bias, particularly within legal decision-support contexts where fairness is critical. To reduce this risk, the system is designed so that no automated function can replace or overrule professional legal judgment. Any recommendations or system-generated insights are strictly informational and intended only to support, not determine, legal decisions. If artificial intelligence is implemented, the sources of training data will be identified and routinely evaluated to ensure reliability and fairness. In addition, testing for bias will be conducted during development, deployment, and subsequent updates to identify and correct any potential issues. All outputs that may influence client outcomes must be subject to review and approval by a qualified human user.

Mirage also includes protections to guard against discriminatory behavior. The system is structured to avoid generating recommendations that are influenced by personal characteristics such as race, ethnicity, gender, or economic background. It further ensures that automated processes do not unfairly disadvantage vulnerable populations or rely on sensitive attributes for analysis or prediction. If any trend or behavior suggesting discrimination is observed, it will be documented, investigated, and corrected through appropriate mitigation measures.

To maintain transparency, Mirage clearly informs users whenever an automated or algorithmic process contributes to system output. When applicable, the system will provide information about the types of data used to generate results and will disclose any known limitations or uncertainties associated with its recommendations. This approach promotes accountability and helps users understand the context and reliability of system-generated information.

4 Glossary

Acronym	**Definition**
SRS	Software Requirement Specification
JPEG	Joint Photographic Experts Group
PNG	Portable Network Graphics
HEIC	High Efficiency Image Container
HEIF	High Efficiency Image File Format
MP4	MPEG-4 Part 14
MOV	QuickTime Multimedia
CLI	Command-line Interface
AI	Artificial Intelligence
LDAP	Lightweight Directory Access Protocol