

# "Freestyle" Project - Planning and Requirements Document

---

See: [The "Freestyle" Project](#)

Prerequisites:

- ["Data Flow Diagramming" Exercise](#)
- ["Process Diagramming" Exercise](#)

This is an optional alternative part of the Freestyle Project which may or may not be assigned, depending on the individual class. If you're not sure whether or not this is required for you, ask an instructor.

## Learning Objectives

- Practice translating project ideas into a written format to communicate them to other people and inform your development efforts.
- Memorialize descriptions of user needs and system requirements, to help you stay on-track and on-scope during development.
- Decompose system functionality into component features, then prioritize them to help you plan your development efforts.

## Instructions

Prepare a written document (in PDF format) which describes the results of your project planning, analysis, and design efforts.

The document should include an analysis of the problem, a description of user needs, and a description of the functional objectives of your proposed solution. It should convey a thorough understanding of the problem, and a detailed plan to guide future development / implementation efforts.

If the document were to be shared with clients or potential investors, it should reinforce their confidence in your capabilities and planning efforts. If the document were to be shared with a third-party development team, it should convey enough information to instruct the development team what to build.

The document should incorporate embedded illustrations and diagrams as necessary. These may be low-fidelity hand-drawn images, but prefer to deliver higher-fidelity versions produced via a [digital diagramming tool](#).

A reasonable length for this document, including embedded images, is between three and eight pages, with a maximum of around 11 pages.

If you get stuck, try thinking about responses for each of the sections outlined below:

- Problem Statement
  - User Needs

- Current State Processes
- Proposed Solution
  - System Objectives
  - Future State Processes
  - Information Requirements
  - Functionality Requirements / Features
  - Interface Requirements
  - Technology Requirements
- Development Plan

## Problem Statement

What problem would you like to solve, or what opportunity will you address? Why is it a problem? What is the impact of the problem? Provide an executive summary.

## User Needs

Who are the individuals affected by the problem? How are they impacted? How could their experience be improved? What are their needs? Please include language like: **"[who] needs [what] so that they can [why]"**.

These statements of user needs should focus more on the problem than any potential solutions to the problem (e.g. **"Office workers need a way to get to work quickly and affordably, so they can save time and money to be spent on other more important things"**, NOT: "Office workers need the city to implement a subway system, so they can get to work").

## Current State Processes

Describe the current state of the problematic process, and identify any pain points as applicable.

Include a diagram of the process (i.e. "Current State Process Diagram").

## Proposed Solution

Which solution are you proposing to implement, and why? Provide an executive summary.

## System Objectives

What are the goals or objectives of your proposed system? Please briefly describe the system's desired functionality at the highest possible level (e.g. "The system should send me an email every morning to share information about the day's weather forecast...").

## Future State Processes

Describe the future state of the process to be implemented by your proposed solution, and identify any areas of simplification and automation.

Include a diagram of the process (i.e. "Future State Process Diagram").

## Functionality Requirements

After describing the functionality at a high level, also decompose this functionality into smaller logical sub-components, and describe each in more detail. Prefer to use a hierarchical outline with a section for each of the sub-components.

Describe functionality in terms of desired user experience, as well as underlying system responsibilities. When describing system responsibilities, think in terms of information inputs and outputs (see "Information Requirements" section). When describing functionality in terms of user experience, feel free to include interface mockups and illustrations (see "Interface Requirements" section).

## Information Requirements

Describe the system's information requirements, in terms of information inputs and outputs. What information inputs does the system require in order to achieve its desired functionality? Where do these inputs come from? What is the data format of these inputs? What information outputs does the system produce in the process of achieving its desired functionality? What will be the data format of these outputs?

Include a Data Flow Diagram (DFD) to help illustrate these information requirements.

## Interface Requirements

Describe the mechanisms in which your users will interact with the system.

Include visual representations in the form of mockups, wireframes, flow-charts, storyboards, etc.

## Technology Requirements

Which APIs or third-party services, if any, does your system require in order to produce its desired functionality? What functionality do these APIs or services provide? Why are they necessary or helpful?

Which third-party Python packages, if any, does your system require in order to produce its desired functionality? What functionality do these packages provide? Why are they necessary or helpful?

Will your software be deployed to a remote server (i.e. web app or scheduled service)? Will your system need to integrate with other hardware like scanners and sensors? Describe these hardware integrations as necessary.

## Development Plan

How will you prioritize which parts of the system to work on first? If working with other individuals, how will you split-up the work? Include any supporting charts, tables, or diagrams.

## Submission Instructions

Upload a PDF version of your planning and requirements document using the designated link.

Optionally also include a copy of this document and any supporting artifacts in your project repository in a subdirectory called something like "planning" or "design".

## Evaluation

Implementations will be evaluated based on the criteria below:

Category	Weight
Problem Statement	30%
Proposed Solution	50%
Thoroughness and clarity of analysis	20%

This rubric is tentative, and may be subject to adjustments during the grading process.