[Design Document Title]

Author(s):

Created Date:

Last Updated:

Applicable Stories in the backlog:

Persona	Story	Version	Call Chain

What is my understanding of the requirements? What is the primary goal or objective?

-

What are the secondary goals/objectives?

-

What are the known design constraints?

-

What are the acceptance criteria for the solution (including non-functional)?

-

What assumptions am I making about these requirements and the system?

-

What are the unknowns? (i.e. what information is not currently available?)

-

What are the tradeoffs I am deliberately making?

-

What areas/features are most likely to change over time?

-

What are anticipated/possible failure scenarios and how should they be handled?

-

Are there any special considerations related to security?

-

What are the existing areas impacted?

-

How will the design/architecture need to change?

-

How can I encapsulate current and future change?

-

How should I test and validate the system?

-

What risks am I aware of?

-

What concerns do I have?

-

What are the specific steps to implement and what is their level of effort?

-

Guidance for Completion of Design Analysis Document

General Guidance and Purpose:

- Work through the process of analysis by writing things down
    - Use the exercise of writing down your thoughts and decisions as a way to uncover hidden assumptions and holes in the requirements
    - Structure your thoughts around the specific considerations and questions
- Time box your effort to an hour or two
- Use whatever format works best for you
    - Pictures, words, flowcharts, etc
- Some sections might not apply. Feel free to skip them.
- Final section should be a set of implementation tasks with estimates

Section Guidance:

What is my understanding of the requirements? What is the primary goal or objective?

Use this section to articulate, in your words, what the intent is of this feature/requirement and how it is intended to work. Include what the desired impact/capability will be for the intended user or beneficiary of this feature

What are the secondary goals/objectives?

List any secondary objectives or goals that are separate from the primary objective.

What are the known design constraints?

What are the limitations that have been imposed on the implementation of this feature that reduce/impact the design decision choices you can make? Constraints can come from a variety of sources: cost, time, hosting, user requirements, performance, etc. In general, design constraints are a valuable tool to reduce the scope of possible solutions to a problem so identifying these constraints early can help guide us to the best design decisions.

What are the acceptance criteria for the solution (including non-functional)?

Enumerate the list of criteria by which the success or failure of your implementation should be judged. Be sure to include non-functional criteria such as performance, quality, reliability, resource utilization, etc.

What assumptions am I making about these requirements and the system?

Often times when you are going through this process of analyzing and writing down how you are going to build something, you uncover something you had not considered that must be addressed even though it is not spelled out in the requirements. For example, if the requirement is “charge the credit card as part of a product fulfillment process” you might ask yourself, “What happens if the attempt to charge the card fails for some reason?” Another assumption might be that you will be able to test this credit card workflow without using a real credit card. This is where you can start listing your assumptions about how you should be handling some of these undocumented requirements so that other stakeholders can review and validate these assumptions.

What are the unknowns? (i.e. what information is not currently available?)

In addition to uncovering requirements that you can make assumptions about, you might also uncover something that is completely unknown that could have an impact on development once you get started. For example, maybe there is an API that you need to use but the definition of that API is not published yet. Or, maybe there is an algorithm that needs to be developed that it is not known whether said algorithm is even feasible.

What are the tradeoffs I am deliberately making?

We are always making tradeoffs. Most of the time it feels like these are implicit and not explicit. The goal of this section is to try to get the engineer to identify tradeoffs they may be making in their approach. Consistency, performance, maintainability, testability, etc?

What areas/features are most likely to change over time?

As you are analyzing these requirements it will be important for you to assess what areas of the software are likely to change as a result of testing difficult areas of the code, through user feedback, new feature development down the road, etc. This information will inform how your design should be encapsulating these volatilities.

What are anticipated/possible failure scenarios and how should they be handled?

The purpose of this section is to get you thinking about how the system might behave once you get off the "happy path" of execution. Things to think about here include (but are not limited to):

- What happens if an external dependency (service, API, etc) fails or does not respond as expected.
- What if inputs to the logic are outside your expectations?
- What if the system comes under an unexpected load?
- Where and how should exceptions be handled?
- If a multi-step process fails at some mid-point, what should be the steps to recovery, completion, and/or return to a consistent state?
- Do I need to use some sort of persisted multi-state flag to determine progress?
- Should I put in some strategic logging to give details when certain failures occur?
- Do I have any potential race conditions?

Are there any special considerations related to security?

Do I need to authenticate and/or authorize the user accessing the business logic?

Is there any special treatment of sensitive data that should be taken into account?

Am I potentially exposing sensitive data in exceptions logs, etc?

What are the existing areas impacted?

This section is intended to demonstrate identifying the modules/classes/services that are impacted by this feature along with the types of changes that will be made to these components.

This section should also be used to identify service and data contract modifications that will be required.

How will the design/architecture need to change?

Are we adding any new manages/engines/accessors/utilities?

Are we creating any new interfaces or data contracts?

How can I encapsulate current and future change?

The purpose of this section is to get you to analyze the current feature/requirement in terms of the likelihood that it will change in the future. The idea is to help you think through what the best approach is for encapsulating future change.

How should I test and validate the system?

What will be your strategy for deciding how to validate the system performs as expected? Aside from writing unit tests, are there specific integration tests that you will want to develop/modify to verify this requirement? What are some edge cases you will want to make sure you are testing?

What risks am I aware of?

Examples of risk considerations…

- Do you feel there is uncertainty in the outcome of this effort?
- Is there any risk related to the estimates that have been provided?
- Is there a level of requirements ambiguity that could result in significant re-work?
- Are you missing any information required to fully understand the requirements (e.g. UI/UX designs/workflows, algorithm specifications, etc?)
- Are we using new technologies that we are not familiar with?
- Are you concerned about the stability/robustness of technologies or third part dependencies?
- Do any of the assumptions above represent significant risk if they are not valid?

What concerns do I have?

Is there anything else that you can think of that you would want to surface in this document?

What are the specific steps to implement and what is their level of effort?

This will become the sequential plan for implementing this feature/requirement and will include the "final" estimate of the level of effort to complete each task. In addition to specific feature development tasks, other things to consider here would be:

- New/modified data contracts
- New/modified service contracts
- New/modified unit and integration tests
- Manual testing efforts