DESIGN DOCUMENT STANDARDS

Contents

1. Introduction 2
1.1. Purpose 2
1.2. Scope 2
2. Document Structure 2
2.1. Title 2
2.2. Reference Table 2
2.3. Summary 3
2.4. Scenarios 3
2.5. Assumptions 3
2.6. Out of Scope 3
2.7. Logical View / Affected component view 4
2.8. Process View 4

1. Introduction

1.1. Purpose

Define the structure and content requirements for Java design documents to ensure uniformity, clarity, and completeness across all software design documentation within the organization.

1.2. Scope

Applies to all Java-based projects within the organization, including new developments and significant enhancements to existing systems.

2. Document Structure

The design document should follow a clear and logical structure, typically organized into the following sections:

2.1. Title

The title of the document should begin with the phrase “Technical Design - ” followed by a clear and concise description of the specific feature or changes being addressed.

Example:

“Technical Design - ABC Rewards"

2.2. Reference Table

This is a must have section and the reference table section should be in a tabular format and should contain the following but not limited to.

Business Requirement Reference	*< ONE PAGER Reference link>*
UX Design - Figma	*< FIGMA / UX Reference link if applicable>*
Epic(s)	*< JIRA REFERENCE>*
Product Manager	*< Product Manager >*
Designer	*< UX DESIGNER if applicable >*
References	*<ALL THE REFERENCES> Including Arch Solutions / other cross project references*
Sign Off	*<Name of the Architect>* *<Name of the Lead>* *<Name of Product Manager for Section 1,2,3,4>*
Dependencies	*< CALL DEPENDENT TEAMS / PERSONS >*
Status	*Status of the document [Draft, WIP, Signed off, etc]*

2.3. Summary

This is a must have section. Write a standard requirement of the epic and elaborate the business driver (usually copy from the one pager). This section should talk in brief about the perseverance of the requirement by the engineering team to conform on the understanding.

Example:
In the existing e-commerce platform, product details are displayed across various screens. When "Consultants" use the application and the customer is enrolled in the Rewards Program, the product details should also display any associated rewards. As reward eligibility is configured at the account level, all users associated with the eligible account should be able to see the rewards icon in the user interface (UI).

2.4. Scenarios

This is a must have section. This section could have a description of different scenario the use case is considered for implementation. This section should conform on the scenarios and use cases considered as part of the feature implementation. This could be a detail of whatever is in scope for the implementation.

Example:
Below are the impact scenarios / pages where the changes are to be done

1. Sub page 1
2. Sub page 2
    a. Alternate flow 1
    b. Alternate flow 2
3. Sub page 3

2.5. Assumptions

This is a must have section. This column should clearly talk about the major and minor assumptions the development team is making to progress on the solution. The articulation could be descriptive or bullet points

Example:

1. This feature is enabled only for the “Consultants”
2. There would be no impact to the flows which are not related to products
3. Assumption 3
4. Assumption 4

Mark as “Not applicable” if there are no assumption made for the feature delivery

2.6. Out of Scope

This is a must have and an important section. Clearly call out the out of scope items which would have a direct or indirect impact on the feature. This could be as descriptive as possible or bullet points. This section is to be signed off by the product owner.

Example:

Following points are considered as out of scope for this feature implementation

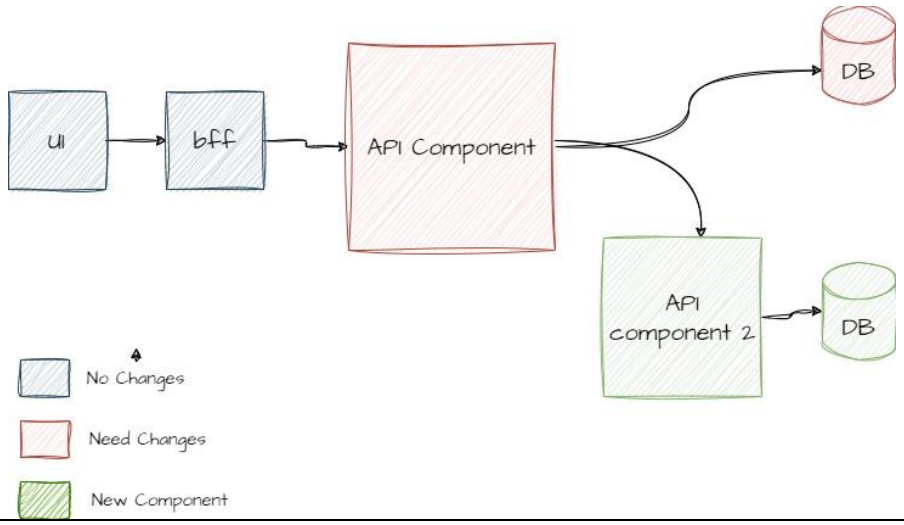
1. Out of scope item 1 description
2. Out of scope item 2 description and so on

Mark as “Not applicable” if there are no assumption made for the feature delivery

2.7. Logical View / Affected component view

This is a must have section in the document. This should have a logical view of what is being built, modified. The representation should clearly highlight the affected components and non-affected components and newly introduced components and interfaces

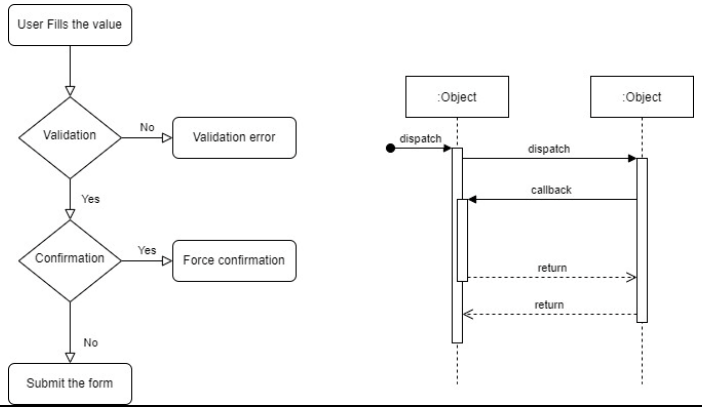
Example:



2.8. Process View

This is a good to have section. This could explain the process flow pictorially with supporting flow diagram, sequence diagrams or could cover a detailed explanation on the process that is impacted due to the changes that feature implementation is going to carry.

Example:



2.9. Detailed Design

Elaborate on each of the component impacts with Class diagram / Sequence diagram / Tabular explanation of alternatives evaluated, pros, cons and rationales. This section could be as detailed as it could be with the clear articulation of the changes involved in the component. This could talk both about technical and process changes that is delivered as part the feature establishment. It could be a custom for different project

Example

Component	Reference	Remarks
Frontend		<call out the impact of changes highlighting higlevel changes to the ui components including cache, stores and etc.,>