JUST ANOTHER PM

PRODUCT REQUIREMENT DOC (PRD)

Create a new module on the Just Another PM website

> 💡 Typically, a PRD has multiple sets of technical stakeholders – like PM, engineers, tech leads, designers, etc. As a result, it makes sense to have a clear matrix that helps readers know the people involved in creating, reviewing, and approving to the PRD

OWNERSHIP/APPROVAL MATRIX AND EDIT LOG

OWNERS		SIGN OFF REQUIRED	REVIEWERS
PM: EM: Designer:			
DATE CREATED	LAST UPDATED ON	DOC VERSION	FEATURE RELEASE DATE
5 May 2023	9 May 2023	1.0	-

Business Problem

> 💡 I like to include the business problem and the business solution in the PRD, even though it is not necessarily required (as it is detailed in the Product One Pager)

Business context

Just Another PM (JAPM) is a dynamic media company, established in 2014 by Siddharth Arora, with the primary objective of creating a thriving community for product managers (PMs) to enhance their skills and knowledge. JAPM offers an array of useful products such as practical guides, mini courses, and cohorts for new PMs, an insightful newsletter, as well as a top-notch PM mentorship program. With a global reach, JAPM has successfully amassed a sizable community of over 45,000 subscribers from all corners of the world.

We did a short survey with JAPM users to understand what resources would be helpful for them in their product management journey. A total of 500 responses were received

- 45% said that they would like more detailed guides on core PM skills
- 27% said that they want reliable PM salary benchmarks
- 15% said that they want to hear about real-world scenarios and challenges
- 13% had other multiple feedback

Given the above feedback, we want to launch a new ‘Salary’ module on the JAPM website. The module includes a salary database created by UGC. In the long run, we plan to monetise this new module to help us increase revenue.

User Persona

Functional Requirements

In this section include the functional requirements of the feature you are developing. Make this section as detailed and comprehensive as possible.

The best way to do that is to think of the user flow, and cover all steps of the flow in this section.

For example, if this PRD is for building a new login screen for your SaaS app, you should think of all use cases:

1. User lands on the login page and is already logged in
2. User lands on the login page and is not logged in
3. User enters the wrong login details
4. User enters the right login details
5. And so on.

Typically, this section covers 70-80% of the document.

Some things to keep in mind:

1. Aim to have no ambiguity.
2. Include process diagrams, mocks, extra data/research. (A picture/diagram is worth a thousand words) Visual elements make it easy for everyone to understand the requirements.
3. Always think of good and bad flows. For ex: a user clicks on submit button on a form
   a. Good flow: if the information is correct, redirect the user to the thank you page.
   b. Bad flow: if the information is incorrect, show the relevant error message

Non functional requirements (NFR)

NFRs are not very obvious and hence easy to forget while creating PRDs.

An NFR for the login flow PRD could be: “the response time for the authentication API should not exceed XXX milliseconds.”

If you aren't familiar with what NFRs are, here is a short description.

Non-functional requirements (NFRs) are a set of requirements that describe the performance, usability, reliability, and other characteristics that a product must meet.

While functional requirements define what a product should do, NFRs define how well it should do it.

In a PRD, it is important to include NFRs to ensure that the product is not only functional but also reliable, secure, and meets any other relevant quality criteria. NFRs help to ensure that the product meets the expectations of its users and stakeholders, and can help prevent costly rework and delays later in the development process.

Testing plan and acceptance criteria

Please do not skip/miss to create a testing plan with acceptance criteria. Either add it to this doc directly, or link it if you’ve created it in a separate doc/sheet.

Treat this section as a list of test cases that have to be passed before this feature is released to users.

Depending on the kind of feature you’re detailing here, you could have two types of tests: business tests and technical tests.

Usually, the technical test cases are created by engineers, and business test cases by PMs

Finally, once you have a list of all the test cases, remember to include a clear acceptance criteria for each case

Example:

Business test case: “Click on submit button with the incorrect username.”
Acceptance criteria: “Clicking on the submit button with the incorrect username should show the relevant error message.”

Open questions

There will always be aspects that need more clarity or concrete answers, which you don’t have at the time of writing.

These questions could come from engineers, designers, or other stakeholders. Documenting all such questions in one place makes it easy to keep referring to them.

At the same time, you should try to close them before the dev start date.

Future plans (optional)

If you’re working on a feature that is a part of a larger roadmap, there is a high chance that there will be more PRDs and releases on the same topic.

In such cases, ensure you add details of what you plan to include in the future versions of the product.

There might also be instances where you had planned a certain aspect in the current PRD, but for whatever reason you decide to remove that aspect from the current version and move it to the next one. Such items fit perfectly in this section.