Structure of PSPD (Product Specification & Prototype Design)

Chapter 01: Introduction

Note: This has already been done as part of the Project Proposal. PP was made up of 2 parts,

1. Problem
2. Methodologies.

So, the introduction is same as the Problem in Project Proposal

* 1. Chapter Overview – TTT (Tell what you’re going to tell, Tell, Tell what you told): 1 paragraph – 6 to 7 lines
  2. Problem Domain/ Background
  3. Problem Definition
     1. Problem Statement
  4. Aims and Objectives
     1. Aims
     2. Research Objectives

|  |  |  |  |
| --- | --- | --- | --- |
| Research Objectives | Description | Learning Outcome | Research Question |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

* 1. Novelty of the Research
     1. Problem Novelty
     2. Solution Novelty
  2. Research Gap
  3. Contribution to the Body of Knowledge
     1. Contribution to the Problem Domain
     2. Contribution to the Research Domain
  4. Research Challenge
  5. Chapter Summary

Chapter 02: SRS

Links for previous years recordings about “How to write the SRS”:

[drive.google.com/file/d/1UQGcV130J4mvY35ba7ajHc\_em9qhpLkr/view?usp=sharing](https://drive.google.com/file/d/1UQGcV130J4mvY35ba7ajHc_em9qhpLkr/view?usp=sharing), [drive.google.com/file/d/1Zn6\_\_am51GDafFSEVupUMiJ6l-MLmBVz/view?usp=sharing](https://drive.google.com/file/d/1Zn6__am51GDafFSEVupUMiJ6l-MLmBVz/view?usp=sharing)

Common problems with SRS:

<https://westminster.cloud.panopto.eu/Panopto/Pages/Viewer.aspx?id=16dacb29-adef-4913-8d93-af870022fd1e>

* 1. Chapter Overview

The goal of this chapter is to identify stakeholders who may be impacted by the project by examining all potential points of interaction with the system using a rich picture diagram. By gathering information about their perspectives, we will be able to develop an understanding of the expected use cases, functional and non-functional requirements of the prototype.

* 1. Rich Picture Diagram (<https://www.slideshare.net/BSBEtalk/rich-pictures-40784261>)
  2. Stake Holder Analysis

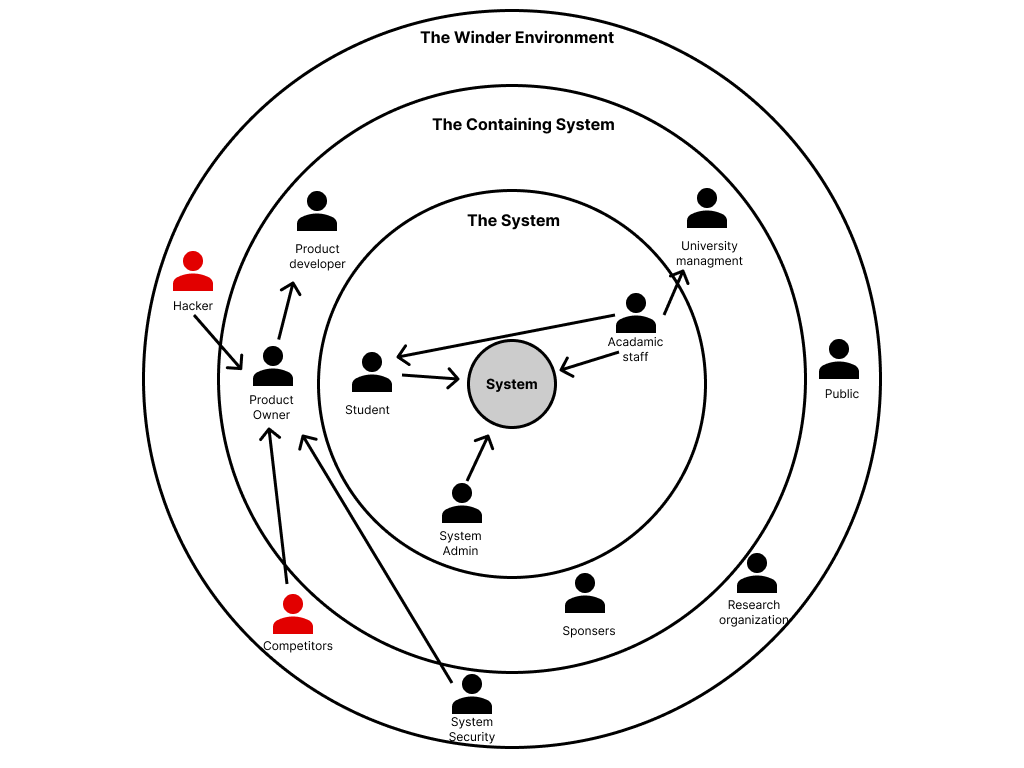
The Stakeholder Onion Model is a visual representation that illustrates the stakeholders that have been identified as being associated with the system. It also provides an explanation of each stakeholder's level of involvement in the system, as seen from the perspective of the stakeholders.

* + 1. Stake holder onion model (<https://www.slideshare.net/tharindugb/onion-diagram>

The book: <https://drive.google.com/drive/folders/155GAJoHsyc6QuGjJJgFeXzY4RZYGyiA5?usp=share_link> – page 29: A typical onion model

Page 43: Example for Influence Flow

Read the entire chapter 2)



* + 1. Stake holder Viewpoints

|  |  |  |
| --- | --- | --- |
| **Stakeholder** | **Role** | **Benefits/ Role Description** |
| Product developer |  |  |
| Product owner |  |  |
| System security |  |  |
| System admin |  |  |
| Student |  |  |
| Academic staff |  |  |
| University Management |  |  |
| Research organization |  |  |
| Sponsor |  |  |
| Hacker |  |  |
| Competitors |  |  |
| Public |  |  |

* 1. Selection of Requirement Elicitation Methodologies (you write the requirement elicitation methodologies/ techniques you use, why you use it over the others – no advantages/ disadvantages)

There were different requirement elicitation methodologies that were used to collect the necessary information for the research project. These techniques included: reviewing relevant literature, administering surveys, and creating prototypes. The choice of these specific methods for requirement elicitation is explained in more detail below.

|  |
| --- |
| **Method 1: Literature Review** |
| At the start of the project, the author conducted a comprehensive review of existing literature to identify areas of research that have not been fully explored in the chosen field of study and domain of interest. This literature review helped to identify gaps in the technologies that can be applied to the existing systems, and to understand the related technologies that could potentially be applied to the systems discussed in the literature. |
| **Method 2: Survey** |
| To gather information and gather requirements from potential users of the proposed system, a questionnaire was used as a tool for conducting a survey. This method will help the author to understand the current problems faced by users in that specific area and expectations of the users for the prototype. It also allows the author to determine whether the proposed solution will be beneficial to the intended users. |
| **Method 3: Brainstorming** |
| Several brainstorming sessions were conducted to discuss various aspects of the project. These sessions involved participation from the project supervisor, senior members from previous batches, and colleagues. These sessions proved to be beneficial in addressing issues related to system design, problem-solving, algorithm development, and project flow. The contributions made during these brainstorming sessions helped in understanding the project from different perspectives and led to the introduction of new features and development methods. Overall, brainstorming was a valuable tool for the success of the project. |
| **Method 4: Prototyping** |
| Since the Agile Software Development Lifecycle was chosen for this project, prototyping was used to iteratively test and evaluate different potential implementations, in order to identify areas for improvement. This method allows for a flexible and responsive approach to development, allowing the author to continually refine the prototype and make adjustments as needed. |

* 1. Discussion of Findings
     1. For each methodology, discuss the findings (ex. For LR)

|  |  |
| --- | --- |
| Finding | Citation |
|  |  |
|  |  |

Ex for brainstorming/ self-evaluation/ observation

|  |  |
| --- | --- |
| Criteria | Finding |
|  |  |
|  |  |

Note: Present the evidence that you have carried out brainstorming/ self-evaluation/ observation as part of the appendix

Ex for Interviews

Table 1:

|  |  |
| --- | --- |
| Codes | Themes |
|  |  |
|  |  |

Table 2:

|  |  |
| --- | --- |
| Themes | Conclusion |
|  |  |
|  |  |

No problem in merging these 2 tables (one table => codes, themes, conclusion)

Ex for Survey

Ref to Thiloshon’s report. However, if you have open-ended questions in your survey, you must carry out a thematic analysis. (For close ended => descriptive statistics)

* + 1. Summary of Findings

|  |  |  |  |
| --- | --- | --- | --- |
| Findings | Methodology 1 | Methodology 2 | Methodology 3 |
|  |  |  |  |
|  |  |  |  |

* 1. Context Diagram

Level 0 Data Flow Diagram => where system is represented as a bubble, and the various end users will be represented as a rectangle

You will not show any developers here. Only whoever uses the system.

* What users give to the system and what they get in return.
  1. Use case diagram

The users in the use case diagram and the users in the context diagram should match.

* 1. Use case descriptions (put only the main ones here, the rest goes in the appendix)
  2. Requirements

Note: Requirements must be prioritized using MoSCoW principle.

M => must have (mandatory)

S => Should have (important)

C => Could have (desirable)

W => Will not have (out of scope)

For Feb submission – Must have (M) must be covered

For April submission – Must have & Should have (M & S) must be covered with better accuracy. C (desirable) could be implemented or made as future enhancements

* + 1. Functional Requirements
    2. Non-Functional Requirements

Both functional and non-functional requirements must be prioritized using MoSCoW principle.

* 1. Chapter Summary

Refer recording for Design and Implementation chapters: <https://drive.google.com/file/d/1hJFOhiX5VhEzLSTYIfwk5O9y_K4u9fMw/view?usp=share_link>

Chapter 03: Design

3.1 Chapter overview

This chapter covers the design choices made to create an appropriate architecture for the implementation, based on the requirements that were gathered. It includes the use of high-level and low-level design, design diagrams, and UI wireframes to explain how the design objectives will be achieved. It also includes an explanation of the reasoning behind the chosen design decisions.

3.2 Design Goals (quality attributes that are important to decide the architecture)

|  |  |
| --- | --- |
| **Design Goal** | **Description** |
| Performance |  |
| Correctness |  |
| Usability |  |
| Scalability |  |
| Adaptability |  |

3.3 High level Design/ System Architecture Design

3.3.1 Architecture Diagram

- Can be drawn using tiered architecture or layered architecture

3.3.2 Discussion of tiers/ layers of the Architecture

3.4 Low-level Design/ System Design

3.4.1 Choice of design paradigm

Note: possible design paradigms => OOADM(object oriented analysis and design methodology / SSADM(structured systems analysis and design methodology)

[Depends of the programming paradigm you use.

Ex. Java -> OOADM

Structure programming -> SSADM]

3.5 Design Diagrams

3.5.1 Component Diagram (for both SSADM & OOADM)

Note: Possible design diagrams:

OOADM =>

Class diagram

Sequence Diagram

Other UML diagrams as applicable

SSADM =>

Data Flow Diagrams (Level 1 & Level 2 – only for the most important process identified in the level 1 DFD)

3.5.X1 Algorithmic Design (if you have come up with a novel algorithm- optional)

3.5.X2 Algorithmic Analysis

3.5.Y1 System Process Flow Chart (How the system will work form end to end. Use either a flow chart or an activity diagram (preferred) [object-oriented flow chart])

3.5.Z1 User Interface Design

1. low level fidelity wireframe diagram

2. high level fidelity prototype

Note: Any diagram that’s relevant to your project. At the same time don’t try to overdo by putting all possible diagrams

3.6 Chapter summary

Chapter 04: Initial Implementation

4.1 Chapter Overview

4.2 Technology Selection

You need to tell what your frontend is, your programing language, IDE, and all other things relevant to implementation

4.2.1 Technology Stack (frontend, middle tier, backend technologies)

4.2.2 Data-set Selection (only if you’re doing a data science project)

4.2.3 Development Frameworks – What you’ve chosen and why (can be justified in a tabular format)

4.2.4 Programming Languages – What you’ve chosen and why (tabular format is OK)

4.2.5 Libraries – What and why (tabular format is OK)

4.2.6 IDE – What and why (tabular format is OK)

4.2.7 Summary of Technology Selection (in a tabular form)

4.3 Implementation of the Core Functionality

- Take each functionality and put the code in image format

4.4 User Interface (either design in the design chapter or actual in implementation chapter. This is optional since prototype level; you’re not expected to have an UI.)

4.5 Chapter Summary

**Conclusion** (should contain but not limited to discussion of deviations from project proposal, initial test results, reflection on your own work that includes current status of the project against the initial plan that you submitted as part of the proposal, improvements that you need to do for the final submission of the minimum viable product when comparing to current status of the prototype (use the SRS and test result) and finally deviations from initial project plan and how you are intent to complete the rest of work on or before the end of april deadline)

Video Demo

Upload video to Youtube as unlisted video and provide Link to video demo

Chapter 05: Conclusion

5.1 Chapter Overview

5.2 Deviations

5.2.1 Scope related deviations (changes in the scope from the proposal)

5.2.2 Schedule related deviations (changes in the schedule from the proposal => according to the Gantt chart. What is planned vs actual, using a roadmap [things you promised vs what you actually achieved along with how much more to achieve and how you will achieve the balance by the end of April])

5.3 Initial Test Results (e.g. For machine learning project: confusion matrix, AUC/ROC curve)

5.4 Required Improvements (what improvements are required for your MVP, due on the end of April; and how you intend to do it)

5.5 Demo of the Prototype

Upload an unlisted video to YouTube of a demonstration of the prototype. (must be a YouTube link).

Duration: 10 minutes

Problem: 3 minutes (max)[like a pitch]

Solution: 5 minutes (explain your solution with a demonstration)

Improvements you can do: 2 minutes (a reflection of your solution)

Also provide a link to the code.

(You must run your software at this point of time. If you can’t run, people will conclude you don’t have a prototype)

5.6 Chapter Summary

PSPD mark scheme: [docs.google.com/spreadsheets/d/1sS5nITVGDdvgWUGmrCmMz6a3jUN8N2oQ/edit?usp=sharing&ouid=108287781972032294435&rtpof=true&sd=true](https://docs.google.com/spreadsheets/d/1sS5nITVGDdvgWUGmrCmMz6a3jUN8N2oQ/edit?usp=sharing&ouid=108287781972032294435&rtpof=true&sd=true)

|  |  |  |
| --- | --- | --- |
| Project Proposal | PSPD | Thesis |
| Problem | Chapter 01: Introduction (Remove the existing work and add the novelty) | Chapter 01: Introduction (Problem as it is) |
| — | – | Chapter 02: LR |
| Methodology | – | Chapter 03: Methodology |
|  | Chapter 02: SRS | Chapter 04: SRS |
|  |  | Chapter 05: SLEP (Social, Legal, Ethical and Professional Issues Faced in the Project) - https://drive.google.com/drive/folders/1yF9VZF6kElyICKwAWL3iURHk7IZJ-gqm?usp=share\_link |
|  | Chapter 03: Design | Chapter 06: Design |
|  | Prototype | You will improve the prototype to become Minimum Viable Product (MVP) |
|  | Chapter 04: Initial Implementation (based on the prototype) | Chapter 07: Implementation (based on MVP) |
|  | Chapter 05: Conclusion | – |
|  |  | Chapter 08: Testing (Quantitative Evaluation) |
|  |  | Chapter 09: Evaluation (Qualitative Evaluation) |
|  |  | Chapter 10: Conclusion |