

Software Design Description for

Ahmed Mohamed, Mohamed Ahmed, Nour Ahmed, Lobna Ahmed
Supervised by: Dr. Supervisor Name, Eng. TA Name

April 30, 2023

Table 1: Document version history

Version	Date	Reason for Change
1.0	25-Jan-202x	SDD first version's description are defined.
1.1	2-Feb-202x	Added Sequence Diagram.
1.3	5-Feb-202x	Requirement Matrix updated.

GitHub:

Contents

1	Introduction	3
1.1	Document Purpose	3
1.2	Document Scope	3
1.3	Intended audience	3
1.4	Reference Material	3
1.5	Definitions and Acronyms	3
2	System Overview	3
2.1	System Scope	4
2.2	System objectives	4
2.3	Project Timeline	4
3	Design viewpoints	4
3.1	Context viewpoint	4
3.2	Composition viewpoint	7
3.2.1	Design Rationale	7
3.3	Logical viewpoint	7
3.4	Patterns use viewpoint	8
3.5	Algorithm viewpoint	8
3.6	Interaction viewpoint	8
3.7	Interface viewpoint	8
4	Data Design	8
4.1	Data Description	8
4.2	Dataset Description	8
4.3	Database design description	8
5	Human Interface Design	8
5.1	User Interface	8
5.2	Screen Images	8
5.3	Screen Objects and Actions	8
6	Requirements Matrix	8
7	APPENDICES	9
7.1	Github	9
7.2	Other appendices as appropriate	9

Abstract

Add your project abstract here. (Word Limit 150)

1 Introduction

1.1 Document Purpose

In this section identify the purpose of this software design description (SDD) document and its intended audience.

1.2 Document Scope

Provide a brief description of the document scope.

1.3 Intended audience

Identify here the stakeholders that this SDD document is intended for. Describe any security or privacy considerations associated with the use of this SDD document.

1.4 Reference Material

List any documents, including SRS and testing plan, which are used as sources of information for the SDD.

1.5 Definitions and Acronyms

Provide definitions of all terms, acronyms, and abbreviations that might exist to properly interpret the SDD. These definitions should be items used in the SDD that are most likely not known to the audience.

Term	Definition
Software Design Document (SDD)	Used as the primary medium for communicating software design information.
Design Entity	An element of a design that is structurally and functionally distinct from other elements.
Design rationale	Information capturing the reasoning of the designer that led to the system as designed, including design options, trade-offs considered, decisions made, and the justifications of those decisions. .

2 System Overview

This section shall briefly describe the system to which this document applies.

2.1 System Scope

Define The scope of your project.

2.2 System objectives

In this section; write out a short list of your project main objectives.

Avoid using generic objectives with no clear measurement, For example:

- To build an easy to use application. (bad example)

Instead you may use:

- To build a usable system that achieve a score equal to or higher than 68 in System Usability Scale (SUS) [1].

2.3 Project Timeline

This section provides the latest version of the project plan between SRS and Technical Phase, including the major tasks to be accomplished, their inter-dependencies, and their tentative start/stop dates.

3 Design viewpoints

3.1 Context viewpoint

In this section you can: The Context view is often a starting point of system design. It provides a description of System's services and users. That context is defined by reference to actors that include users and other stakeholders, which interact with the design subject in its environment. The Context viewpoint provides a "black box" perspective on the design subject.

The purpose of the Context viewpoint is to identify a design subject's offered services, its actors (users and other interacting stakeholders), to establish the system boundary.

- Write a description of the offered services and actors.
- Provide UML context diagram as in figure 1.
- Provide UML use case diagram as in figure 2.

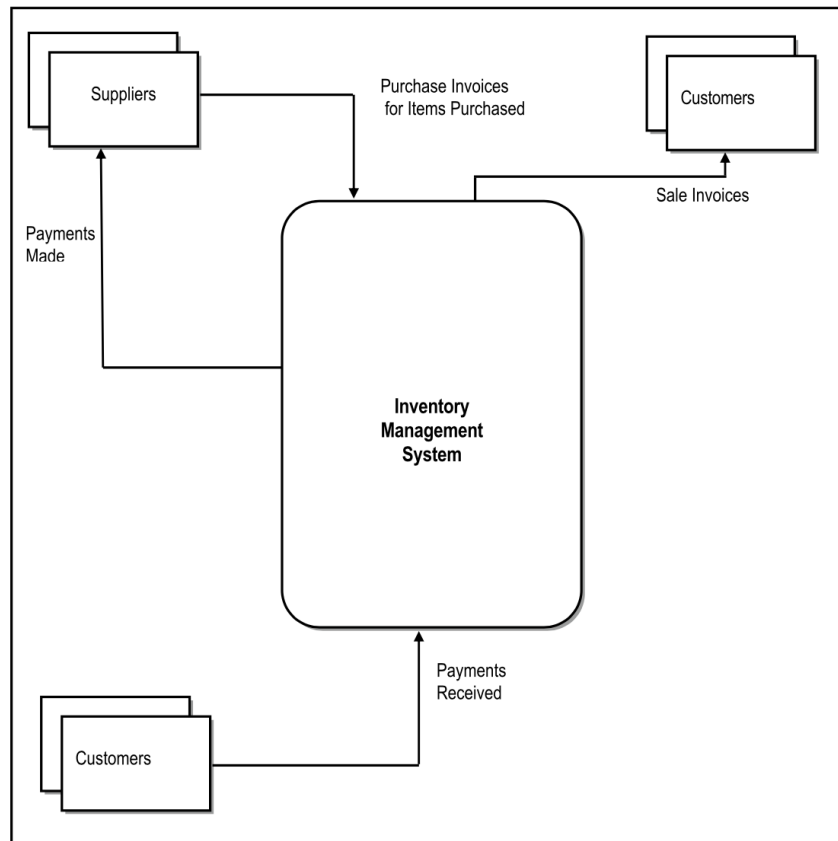


Figure 1: Context Diagram for the Inventory Management System

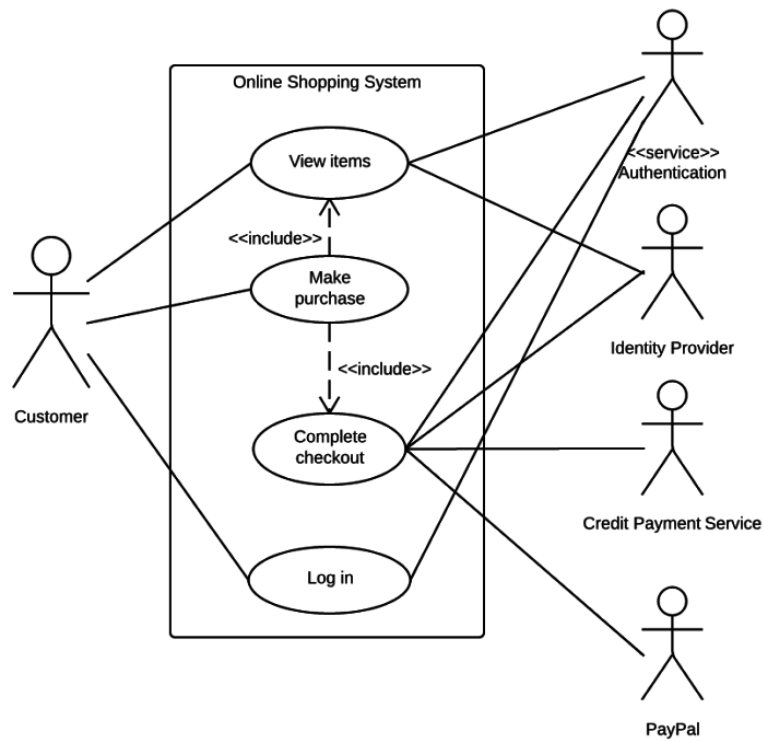


Figure 2: Use Case Diagram Example

3.2 Composition viewpoint

In this section you can: The Composition viewpoint describes the way the design subject is structured into constituent parts and establishes the roles of those parts. The main purpose is to gain a general understanding of how and why the system was decomposed, and how the individual parts work together. The Composition viewpoint supports the recording of the part-whole relationships between design entities using realization, dependency, aggregation, composition, and generalization relationships.

In this section you can:

- Provide a description of the architectural design, used architecture patterns such as MVC or layered architecture. Describe *design entities* such as: system decomposition into subsystems, components, modules; also used libraries, frameworks, software repositories.
- Write a high level overview of how responsibilities of the system were partitioned and then assigned to subsystems. Identify each high level subsystem and the roles or responsibilities assigned to it.
- Describe how the subsystems collaborate with each other in order to achieve the desired functionality.
- Use diagrams to show the major subsystems, data repositories and their interconnections such as: architectural design as in figure ?? and/or UML package diagram.

3.2.1 Design Rationale

Discuss the rationale for selecting the architecture described in section 3.2.

3.3 Logical viewpoint

In this section you can:

- Provide a UML class diagram to Illustrate static structure (classes, interfaces, and their relationships).
- Provide a table to describe each of your classes.

Table 2: ClassName

Abstract or Concrete:	xxxx
Superclasses	xxxx
Subclasses	xxxx
Purpose	xxxx
Collaborations	xxxx
Attributes	xxxx
Operations	xxxx

3.4 Patterns use viewpoint

3.5 Algorithm viewpoint

3.6 Interaction viewpoint

3.7 Interface viewpoint

4 Data Design

4.1 Data Description

4.2 Dataset Description

If your project includes the use of a dataset provide a clear description in this section.

4.3 Database design description

Describe any databases (provide database schema diagram) and/or description of other data storage items.

5 Human Interface Design

5.1 User Interface

Describe the functionality of the system from the user s perspective.

5.2 Screen Images

5.3 Screen Objects and Actions

List screen objects and actions associated with those objects.

6 Requirements Matrix

Provide a cross reference that traces components and data structures to the requirements in your SRS document.

Table 3: Requirements Ratrix

Req. ID	Req Desc	Class	Test Cases ID	Status
FR01	xxxx	class name	TC01, TC02	In Progress
FR02	xxxx	class name	TC03, TC04	Developed

7 APPENDICES

Appendices may be included, either directly or by reference, to provide supporting details that could aid in the understanding of the Software Design Document.

7.1 Github

Add screenshots from Github repository showing your project.

7.2 Other appendices as appropriate

Optional section.

References

- [1] John Brooke. “Sus: a “quick and dirty’usability”. In: *Usability evaluation in industry* 189 (1996).