

Cairo University

Faculty of Computers and Artificial Intelligence

CS251

**Introduction to
Software Engineering**

Project Name

Software Design Specifications

Version X.X

Team Names and Emails



CS251: <Team Name>
Project: <Project Name>

Software Design Specification

Month & Year

Contents

| | |
|-------------------------------|---|
| Instructions [To be removed] | 3 |
| Team | 3 |
| Document Purpose and Audience | 3 |
| System Models | 3 |
| I. Architecture Diagram | 3 |
| II. Class Diagram(s) | 4 |
| III. Class Descriptions | 5 |
| IV. Sequence diagrams | 6 |
| Class - Sequence Usage Table | 7 |
| V. State Diagram | 8 |
| VI. SOLID Principles | 8 |
| VII. Design Patterns | 8 |
| Tools | 8 |
| Ownership Report | 8 |



CS251: <Team Name>
Project: <Project Name>

Software Design Specification

Instructions [To be removed]

- **IMPORTANT. Rename this document to**
CS251-2025-SectionNumber-TAFirstName-Topic-IDs-DraftSDS.pdf // For draft version
CS251-2025-SectionNumber-TAFirstName-Topic-IDs-SDS.pdf
- **Include it in a zip file with the code of the project and project presentation**
- **Remove the following notes and any red notes**
- This document is the template document for your Software Design.
- For further guidelines and information, READ homework 2 document, and sample SRS.

Team

| ID | Name | Email | Mobile |
|----|-------------------------|-------|--------|
| | 1st name is team leader | | |
| | | | |
| | | | |

Document Purpose and Audience

- Any document should tell the reader 2 things: (1) What is this document? (2) Who is expected to read it?
- Write in simple notes: what this document is.
- List the target audience to read this document (e.g. CEO? Project Manager? Customer? Developers, ...?)

System Models

I. Architecture Diagram

- Decide on suitable software architecture for this system. Describe the architecture you chose and why it is suitable for the project.
- Divide your system into components or packages.
- Provide an architecture diagram showing the different components of the system and their relation to each other. Use suitable notation like C4 or arrow and box.



CS251: <Team Name>
Project: <Project Name>

Software Design Specification

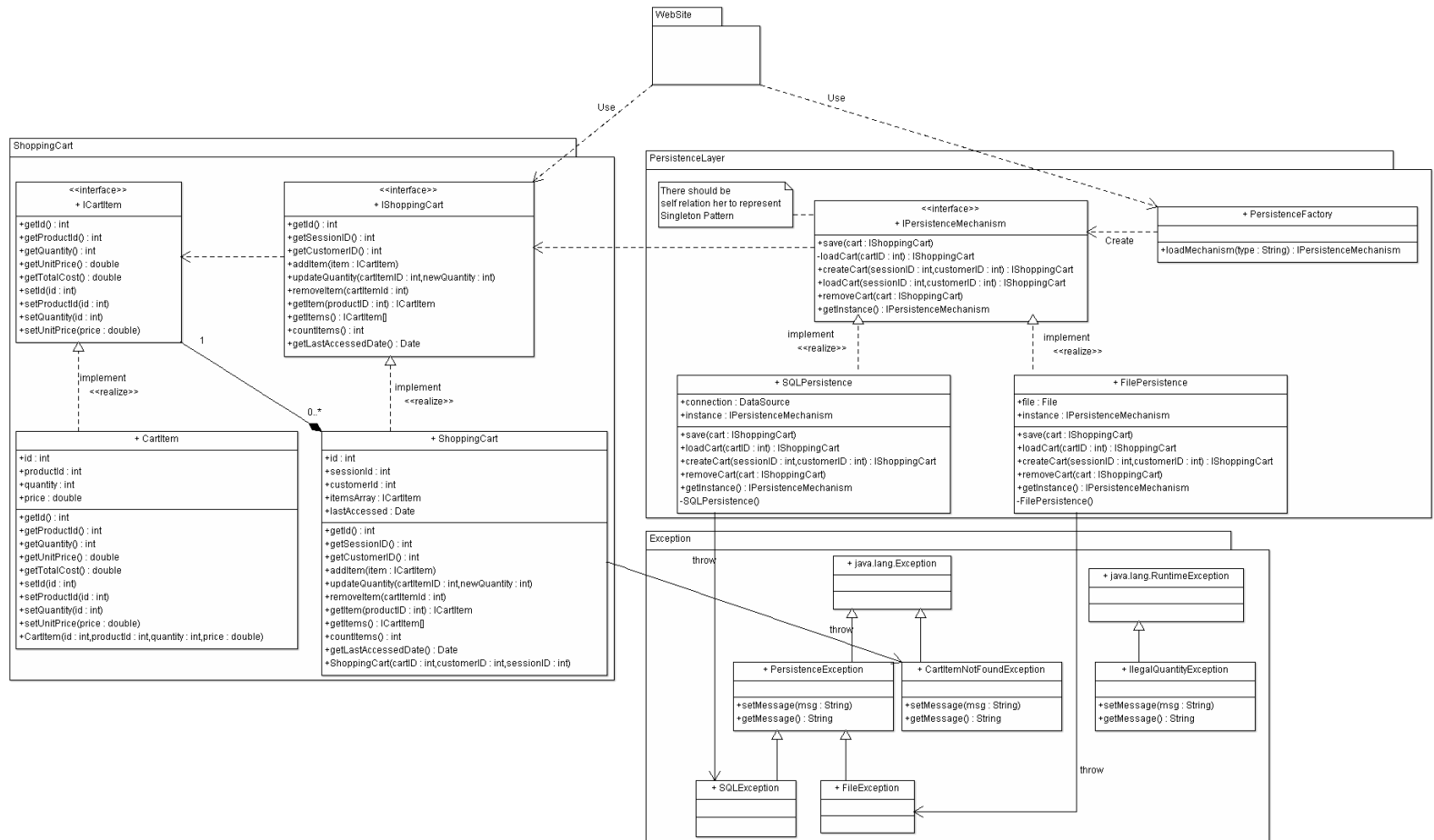
II. Class Diagram(s)

- You should provide your class diagram. Diagram should show the components and packages in your system as well as all classes, their attributes and operations and their relations.
- In case one diagram is so complex, divide it to several ones of reasonable size or draw separate ones, each for one of the components on the system decomposition diagram.
- Class diagram is a static diagram and should not represent any dynamic flow of events.
- Put stereotypes of the classes to give more information. UML predefines some stereotypes like: <<interface>>, <<type>>, <<implementationClass>>, <<enumeration>>, etc. and you create your own also.
- Put Relationships between classes and the types of the relationships.
- Put multiplicity.
- Put relationship name (e.g. faculty "offer" course).
- Put attributes in the classes and their types and visibility.
- Put functions, parameters and return types.
- Include all domain (entity), boundary and control classes needed to implement the system.
- The following is a Shopping Cart Component class diagram example.



CS251: <Team Name> Project: <Project Name>

Software Design Specification



III. Class Descriptions

- List down your classes and describe them
- A class with very small responsibility should be deleted and one with very big responsibility should be divided.

| Class ID | Class Name | Description & Responsibility |
|----------|------------|------------------------------|
| 1. | | |
| 2. | | |

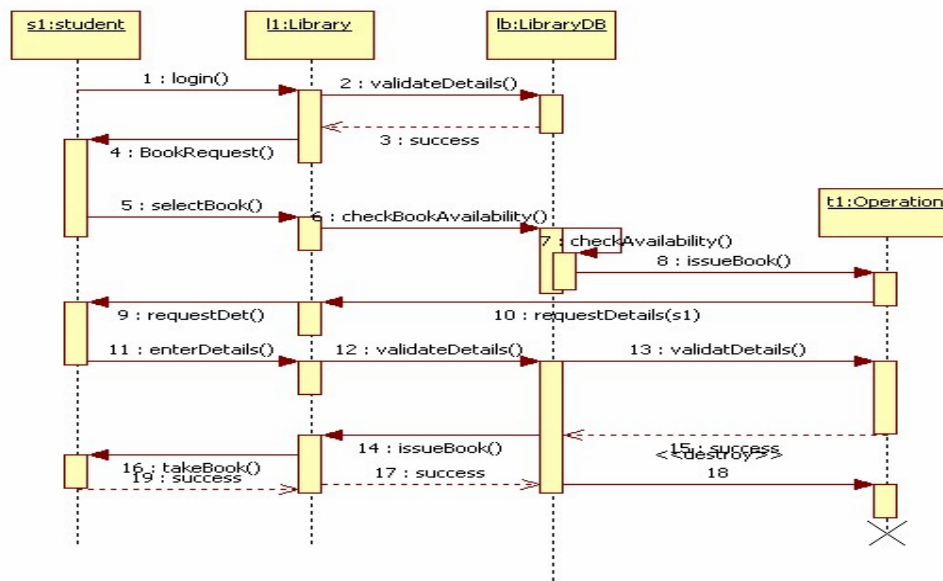


CS251: <Team Name>
Project: <Project Name>

Software Design Specification

IV. Sequence diagrams

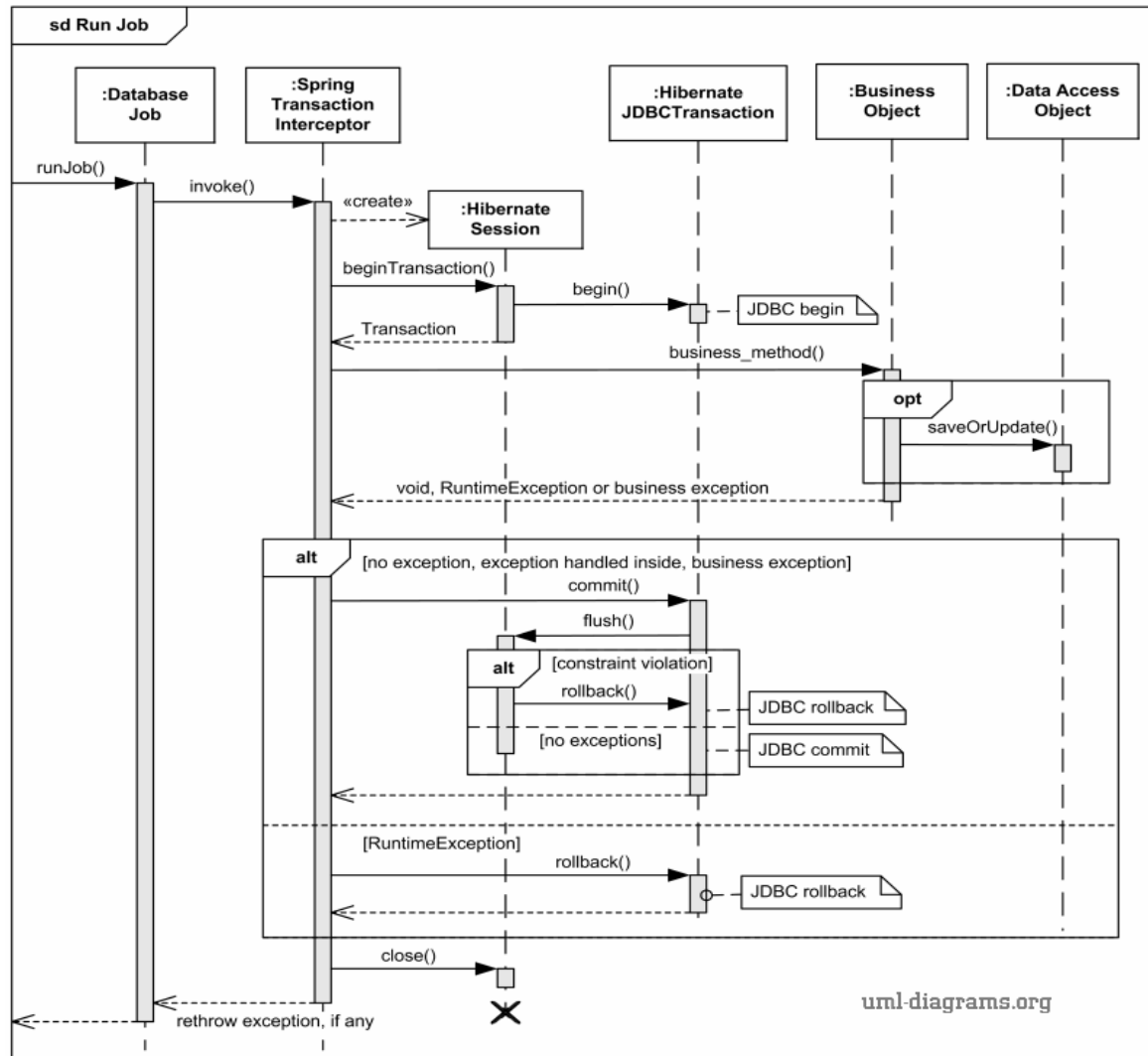
- Usually each use case is represented by a sequence diagram or more.
- Draw a sequence diagram for the most important SIX use cases (user stories) that have complex interaction.
- Overall, all the diagrams should represent all requirements and possible flows for the use case.
- Make sure that each object in the sequence diagram has a corresponding class in the class description table above. If not, it will be **REJECTED**.
- Put actual function calls with proper parameters and return types corresponding to class diagrams.
- Following are couple of examples for small / medium examples. We expect such diagrams, however there is a missing thing in them. Most of calls don't have parameters. Please always specify the parameters in the call, matching the class diagram.





CS251: <Team Name>
Project: <Project Name>

Software Design Specification



Class - Sequence Usage Table

- In this table, we will list the sequence diagrams you drew. For each one, list all the classes used in this sequence. For each class list all the methods you used in this class. Every method or object on a sequence diagram must belong to an existing class in the class diagram and be shown there. If sequence diagrams do not reflect actual classes and methods, they will be REJECTED.

| Sequence Diagram | Classes Used | All Methods Used |
|------------------|-----------------------------|-------------------------------|
| 1. Book Field | Class Field Class Player | Methods Methods |



CS251: <Team Name>
Project: <Project Name>

Software Design Specification

V. State Diagram

- For the ONE MOST IMPORTANT object, draw a state diagram to show the developer the different states it can be in. (for example it is initially created, then it can be shipped, cancelled (if cancelling is possible), ..., etc.)

VI. SOLID Principles

- Explain how you applied THREE OF THE SOLID PRINCIPLES in your design and show the part that the principles were applied in.

VII. Design Patterns

- Use at least THREE DESIGN PATTERNS, any ones from the 23 patterns, not just ones explained in lecture. Explain where you used it and what was the benefit of using it in this place.

Tools

- Write a list of all tools used to develop the design (e.g., PlanetUML, ArgoUML, Visual-Paradigm, etc.)

Ownership Report

- Remove the following notes and any red notes
- For every item in this document, write the owners. If someone is owner of something, s/he understands it 100%.
- Team leader must verify the table with the team members.

| Item | Owners |
|--------------------|--|
| Mohammad Ali Sayed | Part of class diagram and sequence diagrams 2 and 3. |
| | |
| | |