

Cairo University

Faculty of Computers and Artificial Intelligent

CS251

Software Engineering I

Project Name

Software Design

Team Names

Month & Year

CS251: Phase 1 – <Team Name> Software Design Specification

Contents

| | |
|-------------------------------------|---|
| Instructions [To be removed] | 3 |
| Team | 3 |
| Document Purpose and Audience | 3 |
| System Models | 3 |
| I. Class diagrams | 3 |
| Important Algorithm..... | 4 |
| II. Sequence diagrams..... | 5 |
| Class - Sequence Usage Table..... | 6 |
| Ownership Report | 6 |
| Policy Regarding Plagiarism:..... | 7 |

CS251: Phase 1 – <Team Name> Software Design Specification

Instructions [To be removed]

- **IMPORTANT. Rename this document** according to the naming style stated in the project description.
- Remove the following notes and any red notes
- This document is the template document for your Software Design.
- For further guidelines and information, READ project details document (C251-Project Description-ParkingGarage-v1.0).

Team

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

Document Purpose and Audience

- Any document should tell the reader 2 things: What is this document? 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...?)

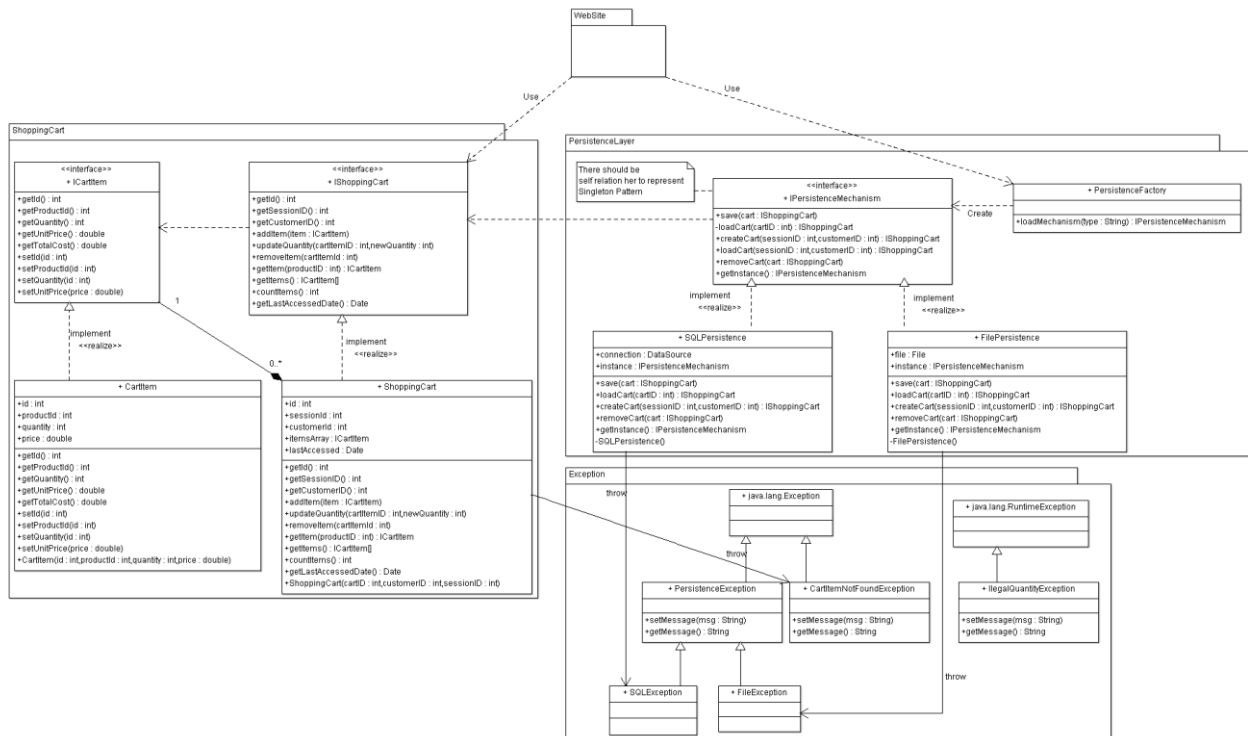
System Models

I. Class diagrams

- You should provide your class diagram. In case on 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.
- Put functions & Put parameters.
- Put data types of each attributes and the parameters.

CS251: Phase 1 – <Team Name> Software Design Specification

- Make sure to include all domain (entity), boundary and control classes needed to implement the system.
- Highly perferred: Each class has a corresponding interface
 - Let all objects parameters and returns be of interface type.
 - See Shopping Cart Case Study
- Following is Shopping Cart Component class diagram.



List down your classes and describe them

| Class ID | Class Name | Description & Responsibility |
|----------|------------|------------------------------|
| | | |

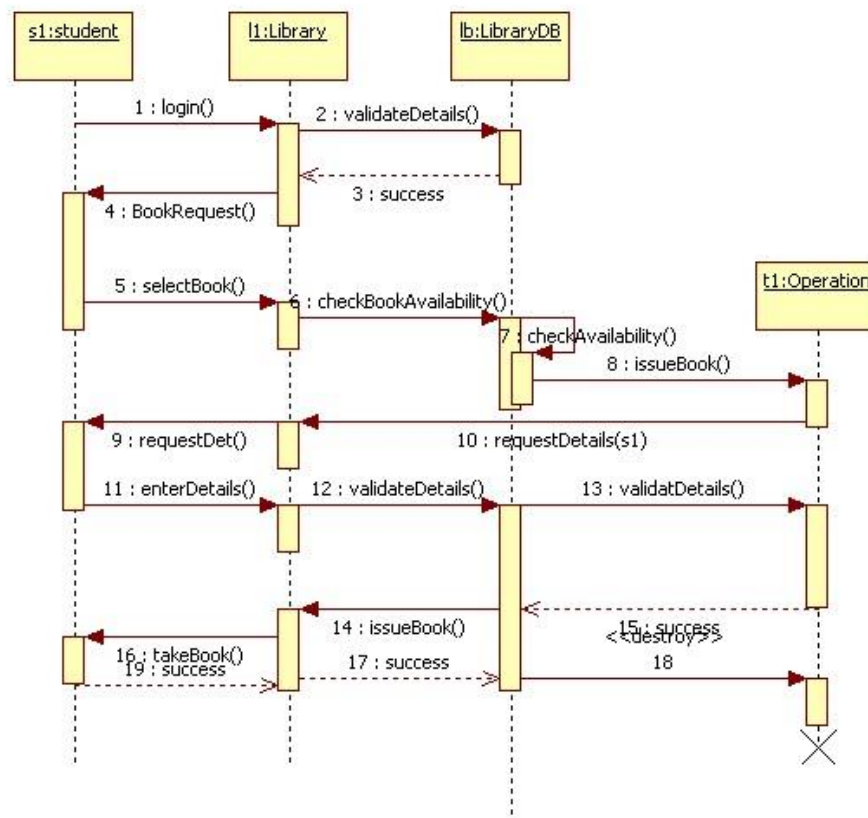
Important Algorithm

- If any method in a class is implementing an algorithm (complex enough), then you should describe it here.

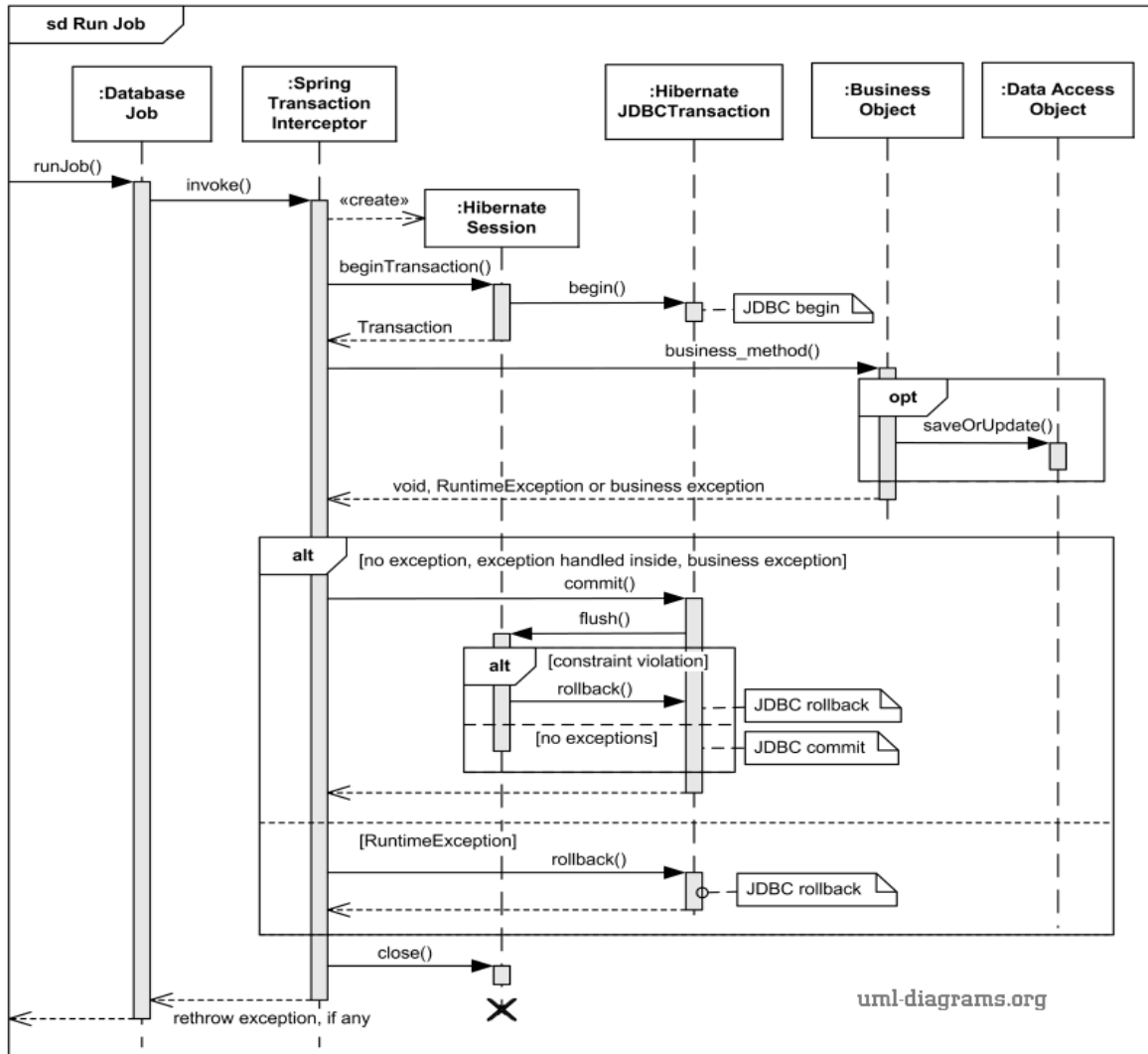
CS251: Phase 1 – <Team Name> Software Design Specification

II. Sequence diagrams

- List Sequence diagrams for all requirements. Provide for each Sequence an ID.
- Usually each use case is represented by a sequence diagram or more.
- Overall, all the diagrams should represent all requirements and possible flows.
- 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: Phase 1 – <Team Name> Software Design Specification



Class - Sequence Usage Table

- In this table, we will list EVERY class in class diagram and which sequences used this class diagram. This helps in avoiding either unused classes or extra classes appears in sequence diagrams. In "Overall used methods" section, put all functions appeared in all sequences. If this table was built in ignorance of actual class / sequence diagrams = REJECTED for whole document.

| Class Name | Sequence Diagrams | Overall used methods |
|---------------|---|----------------------|
| E.g. Employee | 1, 3, 5 (means Seq Ids 1, 3, 5 used Employee class) | Save, GetData |
| | | |

Ownership Report

CS251: Phase 1 – <Team Name> Software Design Specification

- 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 |
|------|--------|
| | |
| | |

Policy Regarding Plagiarism:

Students have collective ownership and responsibility of their project. Any violation of academic honesty will have severe consequences and punishment for ALL team members.

1. تشجع الكلية على مناقشة الأفكار و تبادل المعلومات و مناقشات الطلاب حيث يعتبر هذا جوهرية لعملية تعليمية سليمة
2. ساعد زملاءك على قدر ما تستطيع و حل لهم مشاكلهم في الكود و لكن تبادل الحلول غير مقبول و يعتبر غشا.
3. أى حل يتشابه مع أى حل آخر بدرجة تقطع بأنهما منقولان من نفس المصدر سيعتبر أن صاحبيهما قد قاما بالغش.
4. قد توجد على النت برامج مشابهة لما نكتبه هنا أى نسخ من على النت يعتبر غشا يحاسب عليه صاحبه.
5. إذا لم تكن متأكدا أن فعلا ما يعد غشا فلتسأل المعيد أو أستاذ المادة.
6. فى حالة ثبوت الغش سيأخذ الطالب سالب درجة المسألة ، و فى حالة تكرار الغش سيرسب الطالب فى المقرر.