Cairo University
Faculty of Computers and Artificial Intelligent

SCS252

Software Modeling

Project Name

Software Analysis

Team Names

Month & Year

About this document:
Prepared by Mostafa Saad and Mohammad El-Ramly
Edited by Mohamed Samir

SCS252: Phase 2 – <Team Name>
Project: <Project Name>

Software Analysis Specification

Contents

SCS252: Phase 2 – <Team Name>
Project: <Project Name>

Software Analysis Specification

Instructions [To be removed]

- **IMPORTANT. Rename this document to SCS252-TAName-LeaderID-SDDocument.docx (e.g. SCS252-MohamedSamir-20040752-SDSDocument.docx)**
- **Remove the following notes and any red notes**
- **This document is the template document for your Software Analysis.**
- **For further guidelines and information, READ project details document (SCS252 -Project Description-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

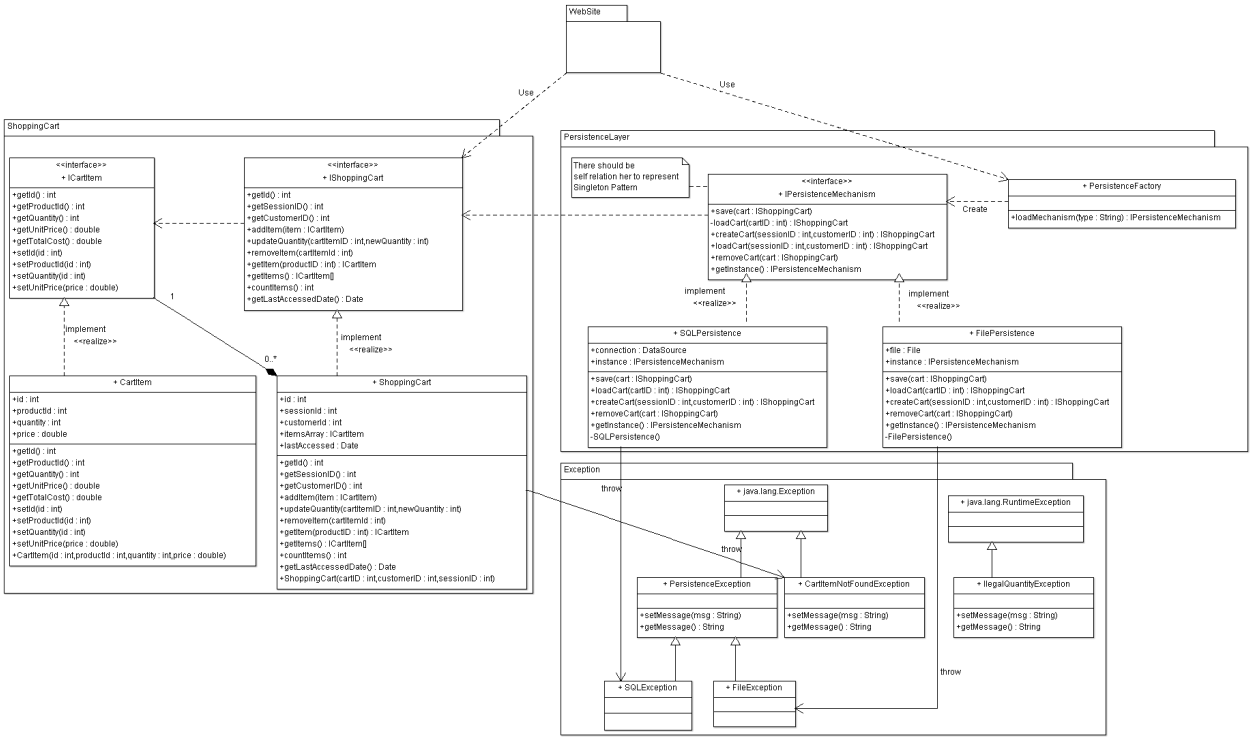
I. Class diagrams

- **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).**

SCS252: Phase 2 – <Team Name>
Project: <Project Name>

Software Analysis Specification

- **Put attributes in the classes.**
- **Put functions & Put parameters.**
- **Put data types of each attributes and the parameters.**
- **Make sure to include all domain (entity), boundary and control classes needed to implement the system.**
- **Highly perfered: 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

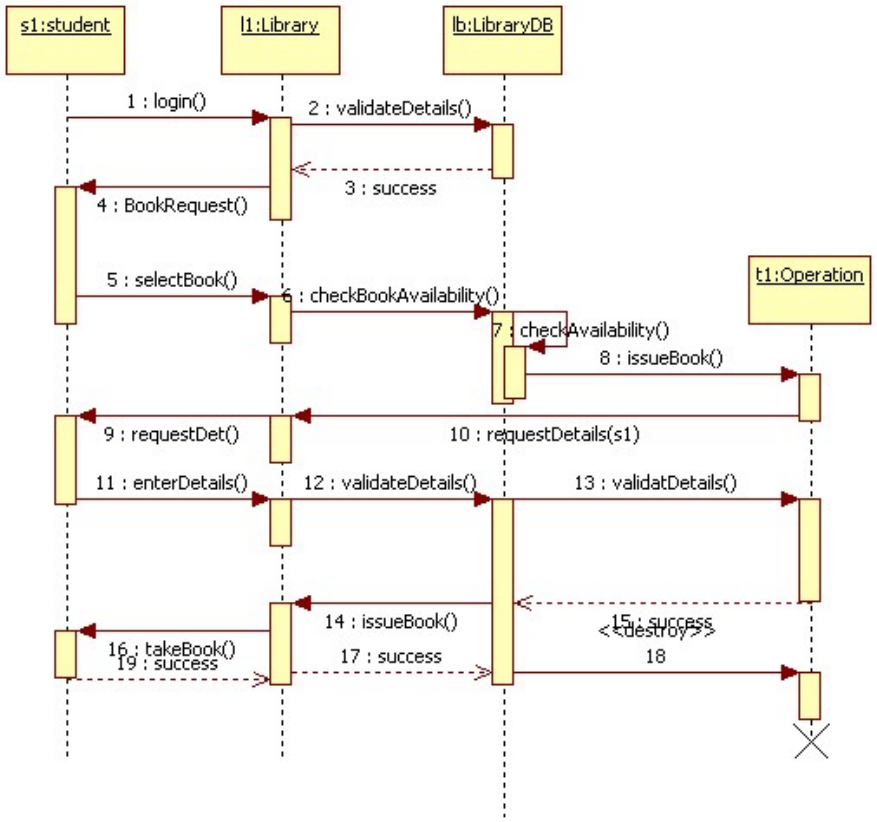
Software Analysis Specification

Important Algorithm

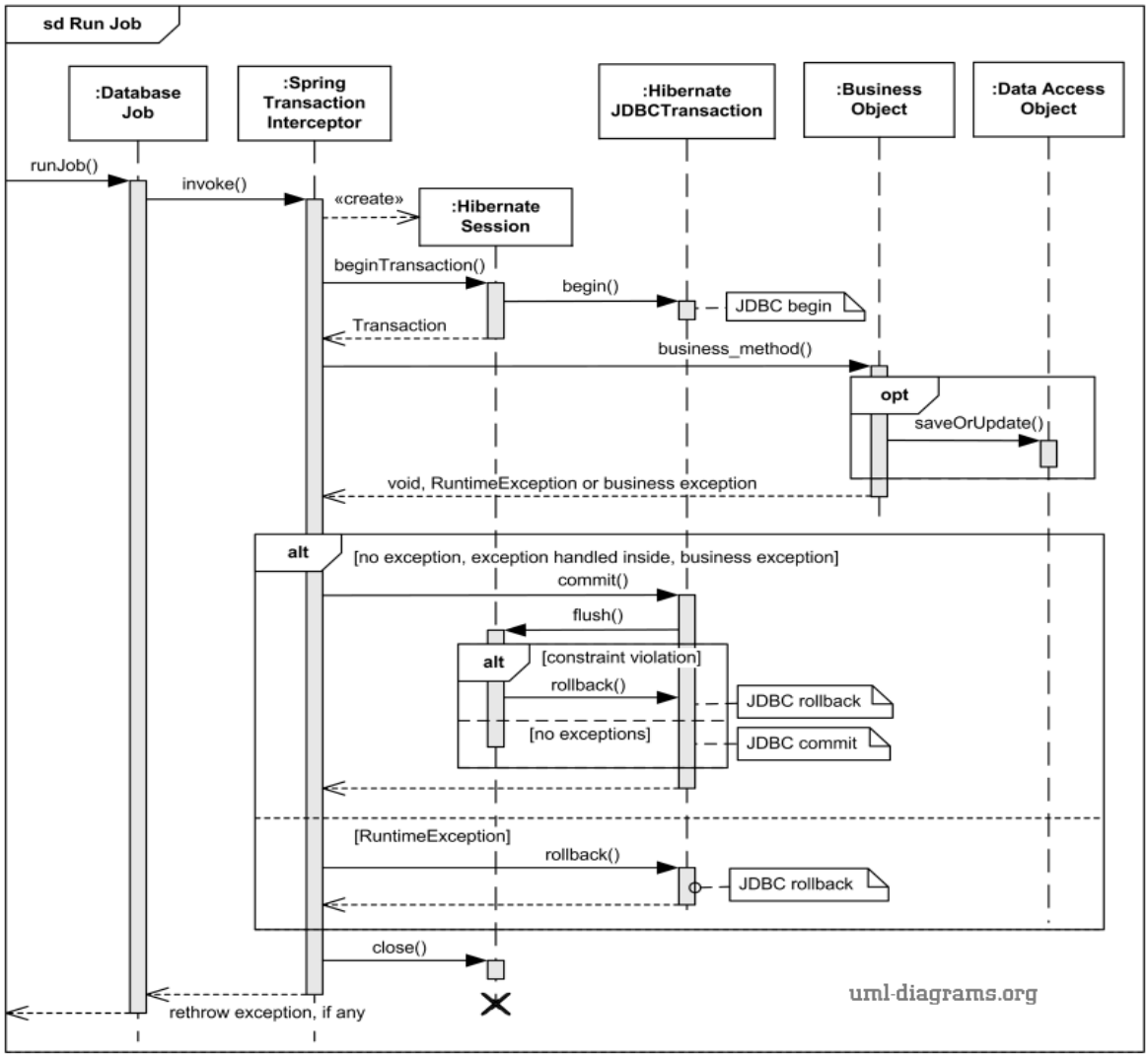
- **If any method in a class is implementing an algorithm (complex enough), then you should describe it here.**

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 / meduim 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.**



Software Analysis Specification



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

SCS252: Phase 2 – <Team Name>
Project: <Project Name>

Software Analysis Specification

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 جوهريا لعملية تعليمية سليمة
2. ساعد زملاءك على قدر ما تستطيع و حل لهم مشاكلهم فى 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.

References

- http://www.mhhe.com/engcs/compsci/pressman/graphics/Pressman5sepa/common/cs1/design.pdf

Authors

- Mostafa Saad and Mohammad El-Ramly (Edited by Mohamed Samir)