

Software Architecture Document

Version 4.0

for

SOEN 341 Team Project

Prepared by

Ait Ouaziz El Hassan	26791573	aitouazizelhassan@gmail.com
Claudia Guerreiro	26951929	claudiaguerreiro@hotmail.com
Jaskaran Kukreja	40071638	jaskaran310193@gmail.com
Himalaya Yadav	40070312	himalayayadav444@gmail.com
Nayana Raj Cheluvaraju	40071318	nayanaraj.237@gmail.com
Seif Elderbi	40029692	seif.derby@yahoo.com
Sri Akhil Varma Alluri	40082333	allurisriakhil@gmail.com

Instructor: Dr. C. Constantinides

Course: SOEN341

Date: 09-11-2018

SOEN 341 Team Project	Version:	4.0
Software Architecture Document	Date:	23-11-2018

Document history

Date	Version	Description	Author
12-10-2018	1.0	Iteration 2	Team1
26-10-2018	2.0	Iteration 3	Team1
09-11-2018	3.0	Iteration 4	Team1
23-11-2018	4.0	Iteration 5	Team1

SOEN 341 Team Project	Version:	4.0
Software Architecture Document	Date:	23-11-2018

Table of contents

Introduction	5
Purpose	5
Scope	5
Definitions, acronyms, and abbreviations	5
Architectural representation	6
Architectural requirements: goals and constraints	9
Functional requirements (Use case view)	9
Non-functional requirements	10
Use case view (Scenarios)	11
Logical view	11
Layers, tiers etc.	11
Subsystems	12
Use case realizations	13
Architecturally significant design packages	15
Development (Implementation) view	18
Reuse of components and frameworks	18
Process view	18
Quality	21

SOEN 341 Team Project	Version:	4.0
Software Architecture Document	Date:	23-11-2018

List of Figures

Fig 1: View Model	5
Fig 2: 3-tier Layer	10
Fig 3: UML interaction Diagram	11
Fig 4: Interaction Diagram for Login	12
Fig 5: Interaction Diagram for Search	12
Fig 6: Interaction Diagram to Create Item	13
Fig 7: Interaction Diagram for Modify Item	13
Fig 8: interaction Diagram for Delete item	14
Fig 9: Core Package	14
Fig 10: Client Package	15
Fig 11: Inventory Package	15
Fig 12: Library Item Package	16
Fig 13: E/R Diagram of the Database	17
Flg14: Horizontal mapping	17
Fig 15: Horizontal mapping	18
Fig 16: Filtered mapping	18

SOEN 341 Team Project	Version:	4.0
Software Architecture Document	Date:	23-11-2018

Introduction

The introduction of the Software Architecture Document provides an overview of the entire project.

This Software Architecture Document (SAD) document is for the online library web application. In this document, a complete description of the web application will be provided. This web application will allow the administrator to create, modify, search and delete records into the database whereas clients can search the items from the library catalogue. Administrator has the privileges to view clients registry. Initially, at least one administrator must be registered in the system.

Purpose

A complete description of all the functions and constraints of the web application is provided through this software Architecture Document (SAD). It contains a full description of the actions that are meant to be performed by the team to get towards the best implementation and reach better solutions.

Scope

In this document, a specification requirement gathering from the problem description will be performed to get design decisions, from software patterns and architectural styles to implementation.

Definitions, acronyms, and abbreviations

UML:Unified Modelling Language

UC:Use case

SOEN 341 Team Project	Version:	4.0
Software Architecture Document	Date:	23-11-2018

1. Architectural representation

The Architecture representation of the library web application is presented with a series of views. Use case, logical and development views are explained in this document, based on views of an underlying Unified Modelling Language(UML) model and is modelled using the 4+1 view(Figure 1).No separate implementation for process or physical view described in this document.

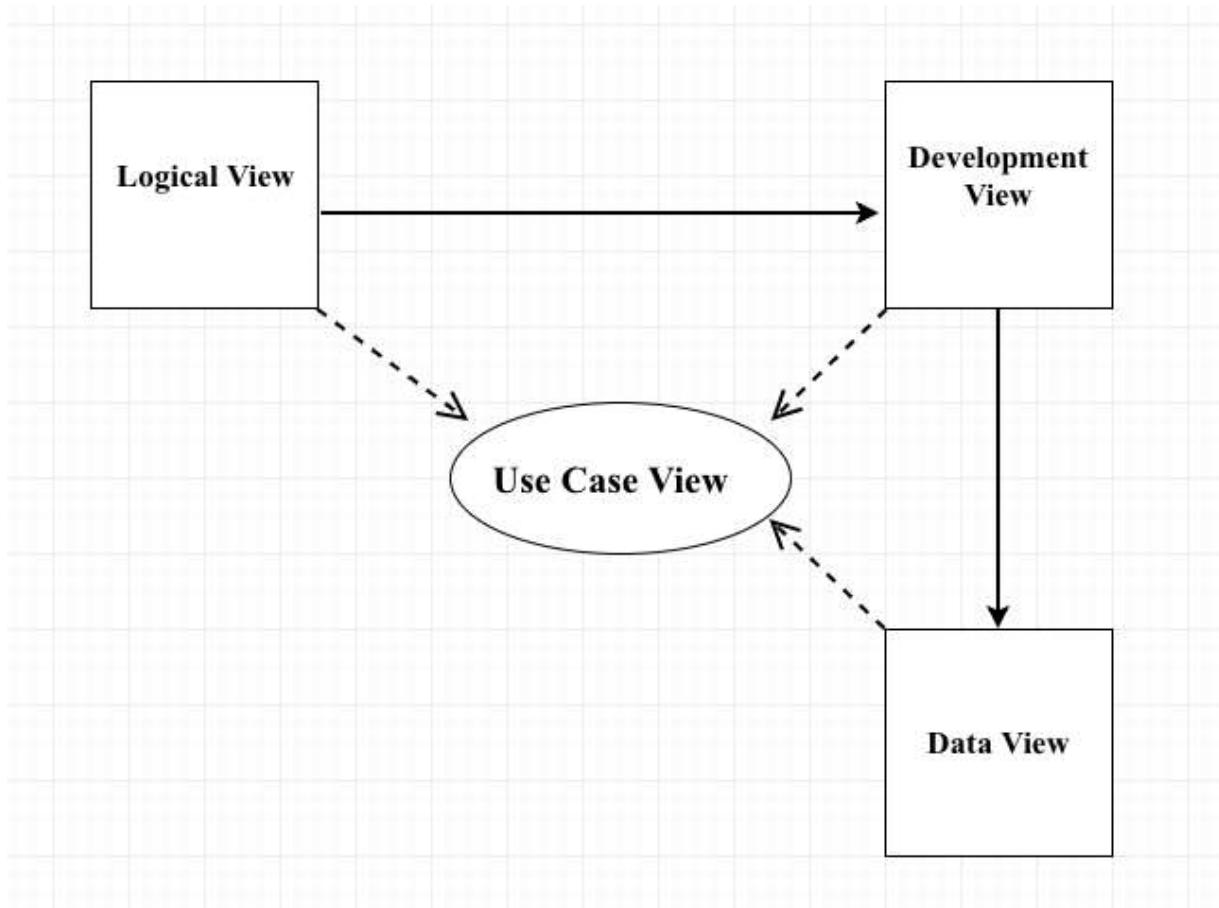


Fig 1: View Model.

SOEN 341 Team Project	Version:	4.0
Software Architecture Document	Date:	23-11-2018

1. Logical view :

Logical view provides the details about the functionalities or services of the system that will be delivered to the end users. Logical view can be best described with the help of design model which provides the functional behavior and analysis model which provides abstract description of the system. UML diagrams used to represent logical view includes the **class diagram and interaction diagram (Communication and sequence diagram)**.

2. Development view :

The development view also called as **implementation view** is the presentation of the system from the programmer's' perspective, and it's primary concern is the software management as it uses UML Component diagram to describe system components and implementation. Component diagram and Package diagram can be used to represent the development view, the UML diagram to represent process view include **Package diagram**.

3. Process view :

The process view deals with the system processes, their interaction and gives attention to run time behaviour of the system.The center of interest is an concurrency,synchronization, performance, scalability, throughput(rate of completed task) etc.The UML diagram to represent process view include **Activity diagram**.

4. Physical view :

Physical view helps to understand the physical location of the software, deployment and installation of the system which is a main concern for designers and system administrators. Physical view represents interaction of software with respect to hardware. UML diagrams are used to represent physical view which include **deployment diagrams**.

SOEN 341 Team Project	Version:	4.0
Software Architecture Document	Date:	23-11-2018

5. Use case view :

The Use case view encompasses the use cases that describe the behaviour of the system as seen by its end users and other stakeholders. This view shows the validation and illustration of system completeness. This view is redundant with other views. UML(Unified Modelling Language) diagrams that can be used in this view are **Use Case Diagrams,User Stories**. However, all other views would not be possible without use case view. It basically details high level requirements of the system. The other views details how these requirements are realized.

6. Data view: Data view is handled by Database specialist and data analysts. In Database, a view is the result of stored query. It provides various view of the data.

SOEN 341 Team Project	Version:	4.0
Software Architecture Document	Date:	23-11-2018

7. Architectural requirements: goals and constraints

Functional requirements (Use case view)

Actor goal List:

Refer to the SRS document for the Actor Goal List

Overview of the Architecturally Relevant Use-Cases:

Source	Name	Architectural relevance	Addressed in:
UC1	View Registry	This use case is relevant for Administrator who wants to see the logged-in User's;	Section 3.2 of SRS
UC2	Modify Items	This use case is relevant to Administrator when they want to modify an item of the library (either add, modify or delete an item);	Section 3.2 of SRS
UC3	Return Items	This use case is relevant to Client, who wants to return a loaned item back to library;	Section 3.2 of SRS
UC4	Loan Items	This use case is relevant to Client,	Section 3.2 of SRS

SOEN 341 Team Project	Version:	4.0
Software Architecture Document	Date:	23-11-2018

		who wants to Loan an item from the library;	
UC5	Login/Logout	This use case is the first step for each user (Administrators or Clients), to have access to the catalog;	Section 3.2 of SRS

Non-functional requirements

Name	Architectural relevance	Addressed in:
Performance Efficiency		Section 3.3 of SRS
Compatibility		Section 3.3 of SRS
Usability		Section 3.3 of SRS
Reliability		Section 3.3 of SRS
Security		Section 3.3 of SRS
Maintainability		Section 3.3 of SRS
Portability		Section 3.3 of SRS
Design Constraints		Section 3.3 of SRS
User Documentation / Help		Section 3.3 of SRS
Purchased Components		Section 3.3 of SRS
Licensing Requirements		Section 3.3 of SRS
Legal, Copyright and other notices		Section 3.3 of SRS

SOEN 341 Team Project	Version:	4.0
Software Architecture Document	Date:	23-11-2018

8. Use case view (Scenarios)

Refer to SRS document for Use Case View.

9. Logical view

The logical view is the description of the software in a way that it gives emphasis to the functionalities of the system for the user. In general, the logical view divides the whole system not only in subsystem but also in tiers and layers. Furthermore, this view will show the realisation of the use cases and the mechanisms used in order to implement the system.

Layers, tiers etc.

The web library application will be developed using the 3 layer architecture. This architecture was used because it helps to keep the system safe and at a good performance. Furthermore, maintenance and scalability will be more straightforward since every logical part is coded apart.

Table Data Gateway(TDG):

Table data gateway acts as an interface between domain layer and data source layer. With the help of TDG we can settle the table data, delete, insert and update. Domain objects send the message to the TDG, which will be sent to the database by mapping its input parameters into SQL call.

Data Mapper: -

Data Mapper helps in establishing communication (transferring the data) between two independent objects like domain objects and database while keeping them independent of each other and the data mapper itself. Thus domain objects don't know the presence of database. Since data mapper has the logic of both application and domain layer it can be placed anywhere on the domain and data source layers.

SOEN 341 Team Project	Version:	4.0
Software Architecture Document	Date:	23-11-2018

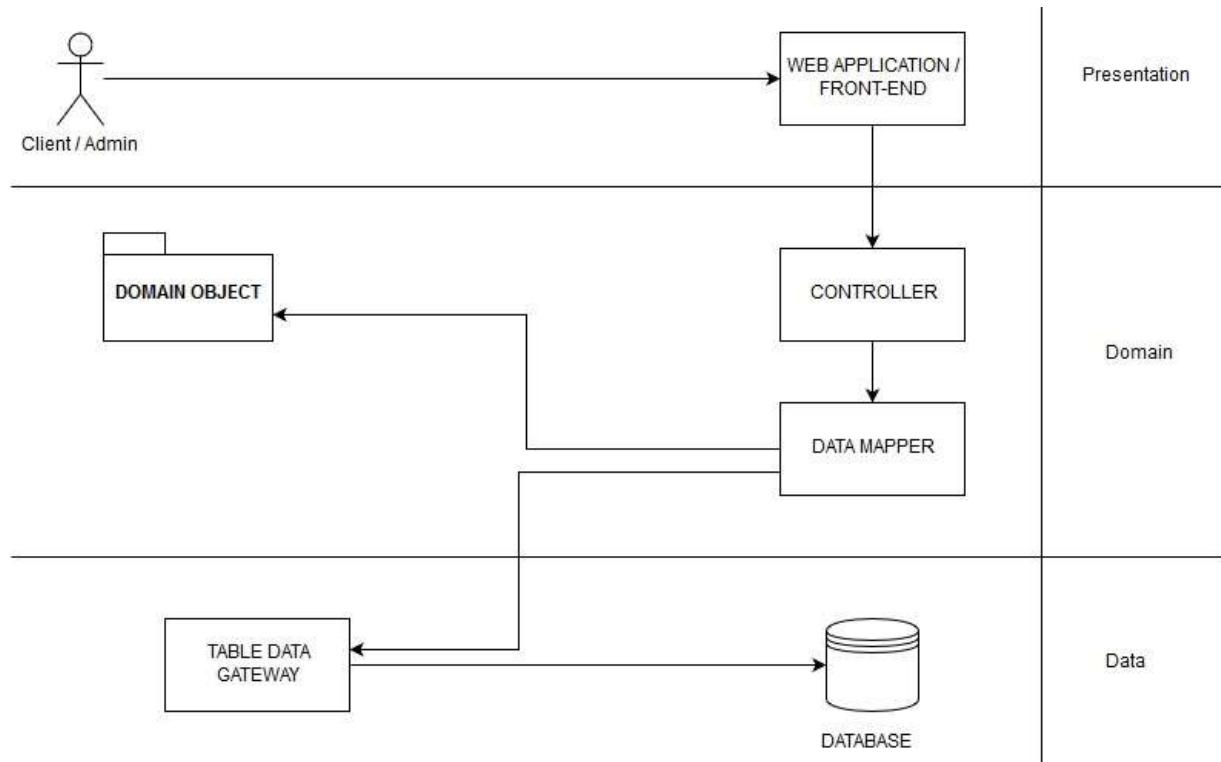


Fig 2: 3-tier Layer

SOEN 341 Team Project	Version:	4.0
Software Architecture Document	Date:	23-11-2018

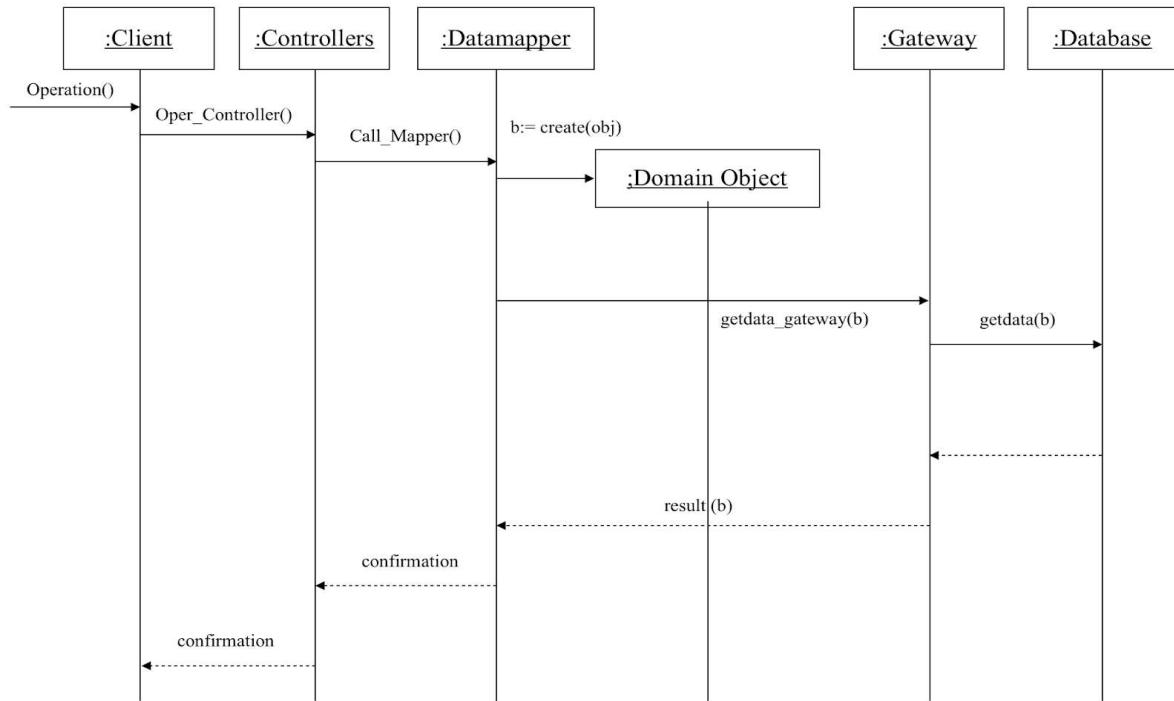


Fig 3: UML interaction Diagram

SOEN 341 Team Project	Version:	4.0
Software Architecture Document	Date:	23-11-2018

Use case realizations

Creator : The creator is responsible for assigning who should create new instance for certain class.In this web application class Administrator have the privilege of creating new instance.

Controller: In this web application, controller will receive inputs from web application and passes information to Data Mapper which in turn distributes it to the appropriate class.



Fig 4: Interaction Diagram for Login

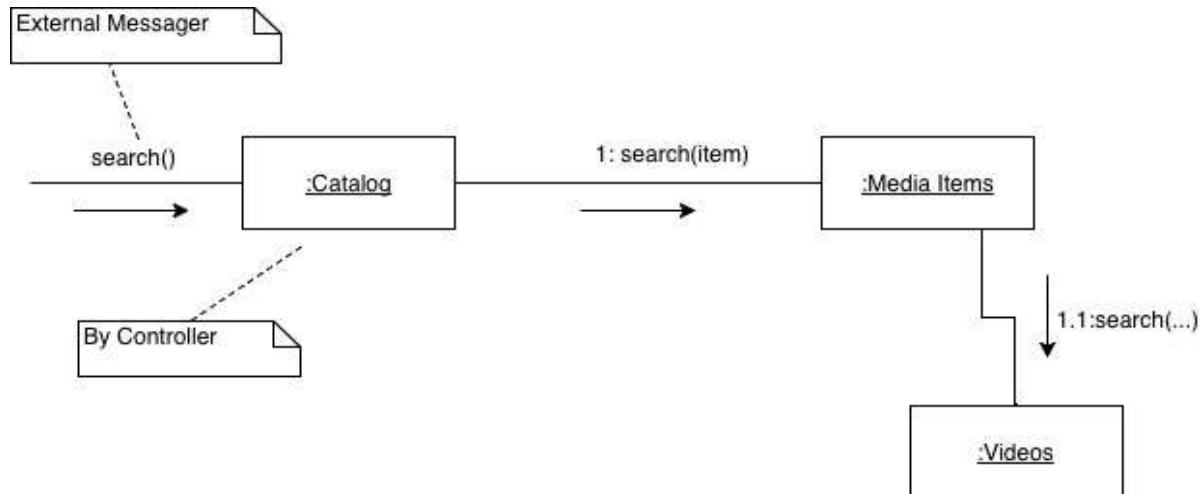


Fig 5: Interaction Diagram for Search

SOEN 341 Team Project	Version:	4.0
Software Architecture Document	Date:	23-11-2018

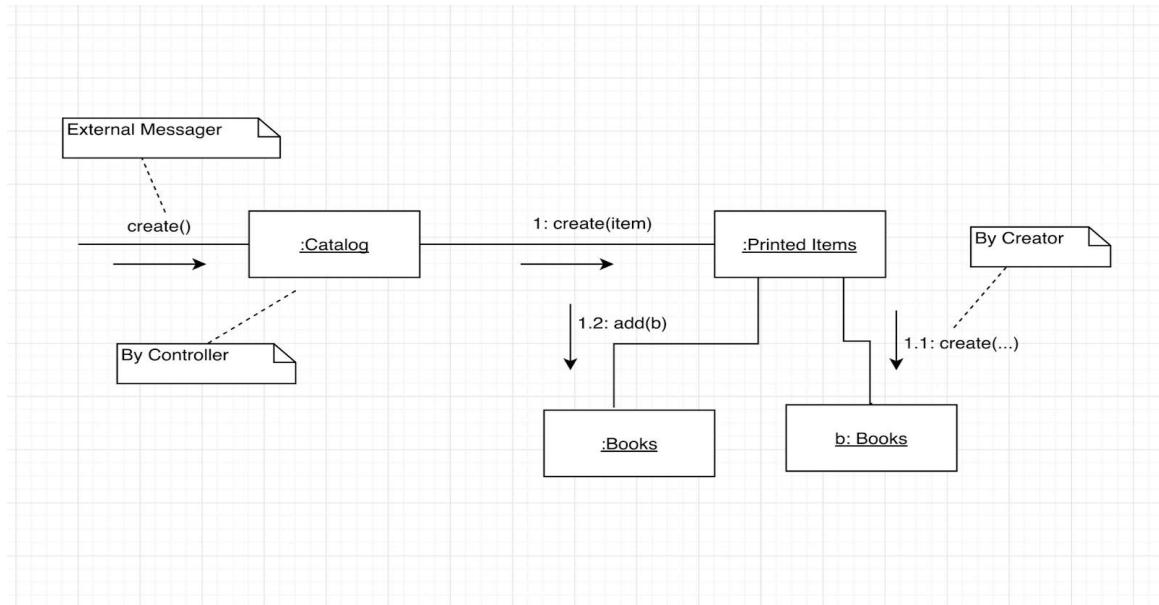


Fig 6: Interaction Diagram to Create Item

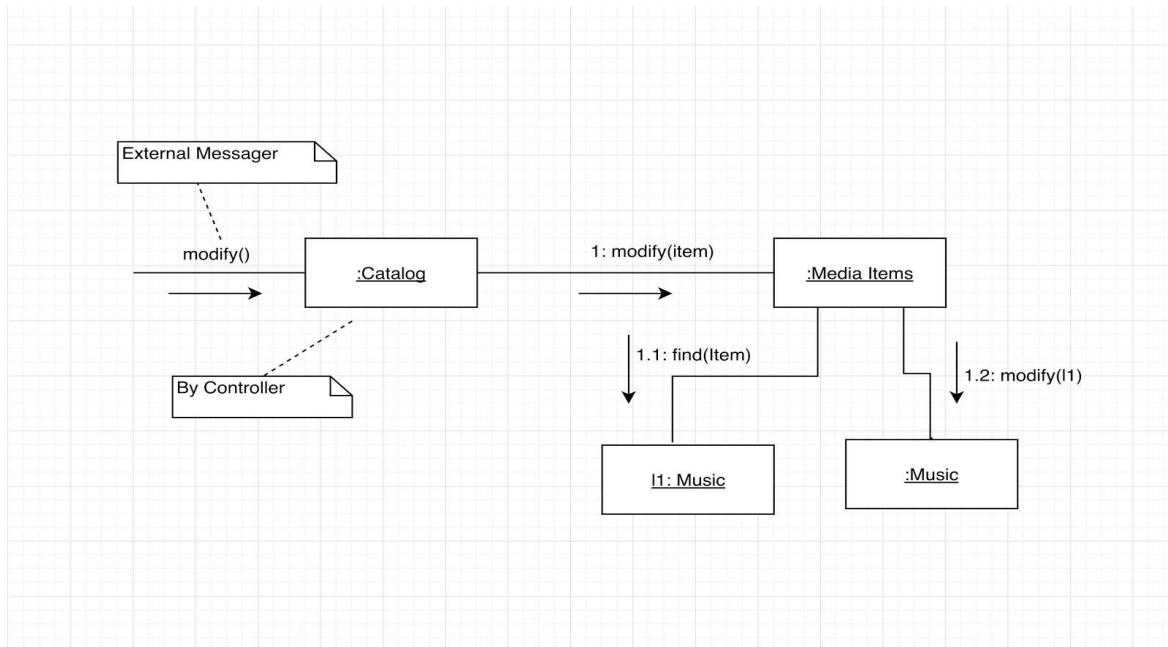


Fig 7: Interaction diagram for ModifyItem

SOEN 341 Team Project	Version:	4.0
Software Architecture Document	Date:	23-11-2018

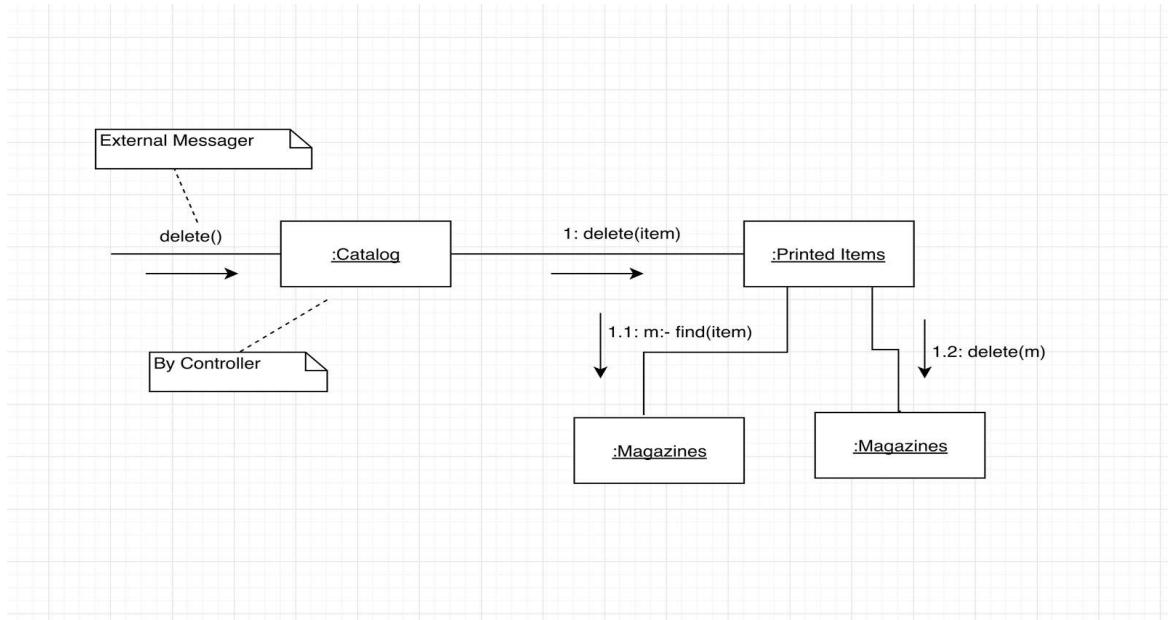


Fig 8: Interaction diagram for DeleteItem

Architecturally significant design packages

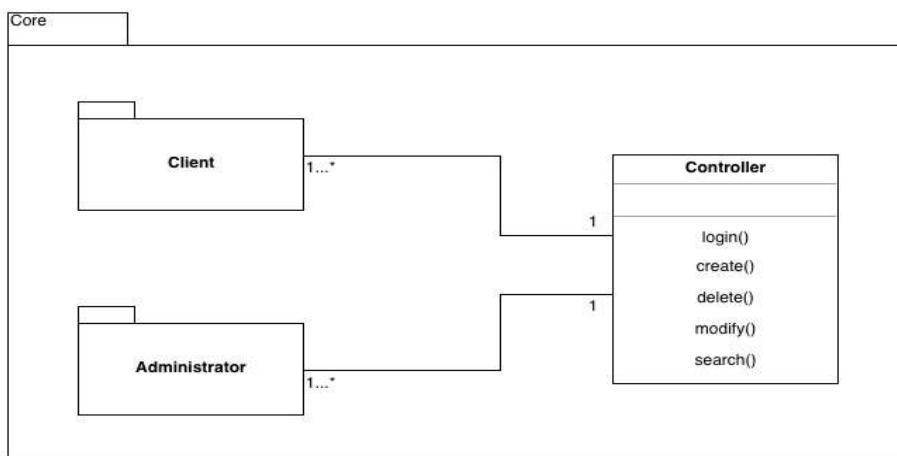


Fig 9: Core Package

SOEN 341 Team Project	Version:	4.0
Software Architecture Document	Date:	23-11-2018

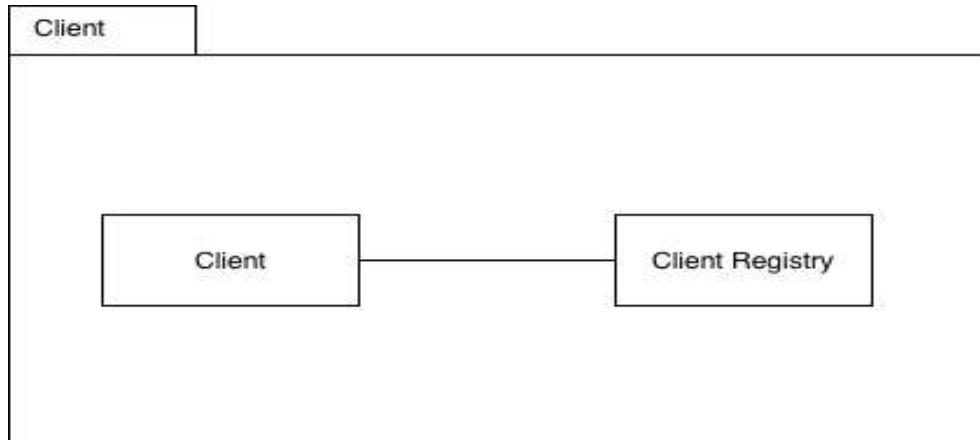


Fig 10: Client Package

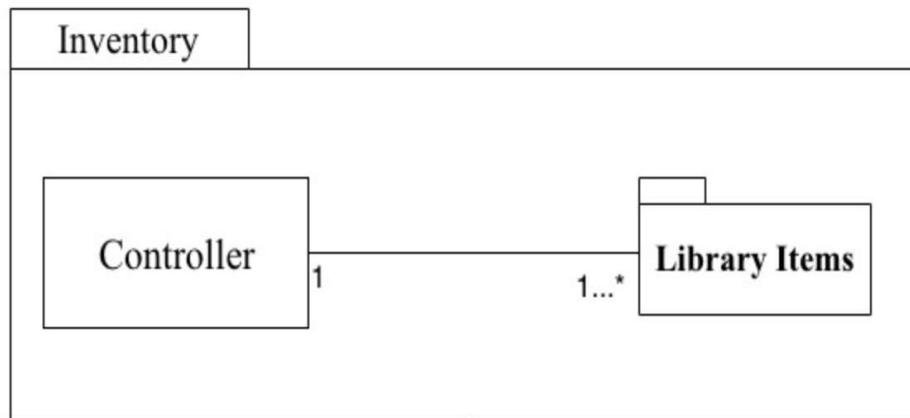


Fig 11: Inventory Package

SOEN 341 Team Project	Version:	4.0
Software Architecture Document	Date:	23-11-2018

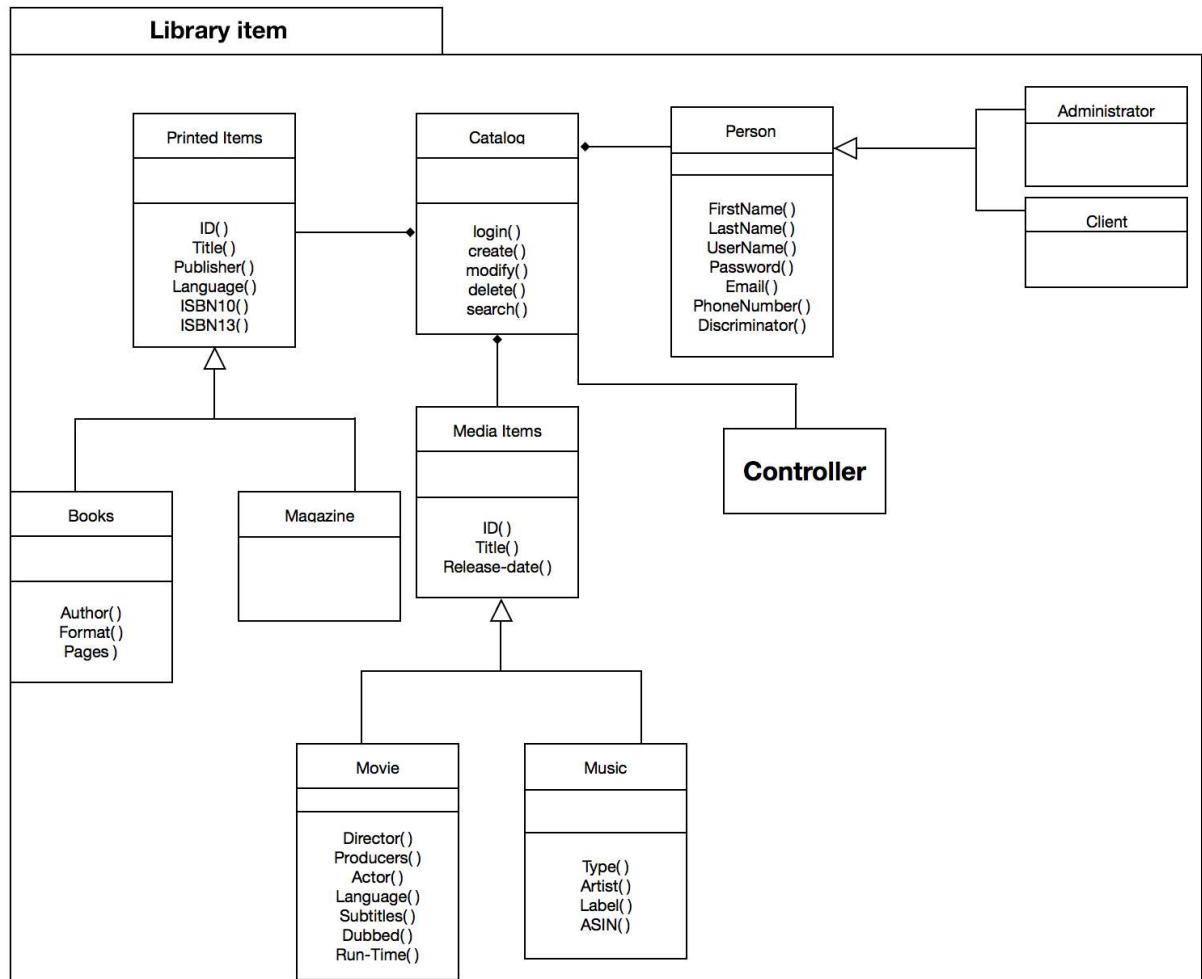


Fig 12: Library Item Package

10. Development (Implementation) view

The Session covers UML in terms of package diagram, the same is shown in above section *Architecturally significant design packages*.

11. Data View

The library web application data will be divided into filter mapping for the administrator and client, meaning that both classe will have only one table "Person". For the catalog items, we will be using a horizontal mapping, which will be mape each logical class to a table in the database.

SOEN 341 Team Project	Version:	4.0
Software Architecture Document	Date:	23-11-2018

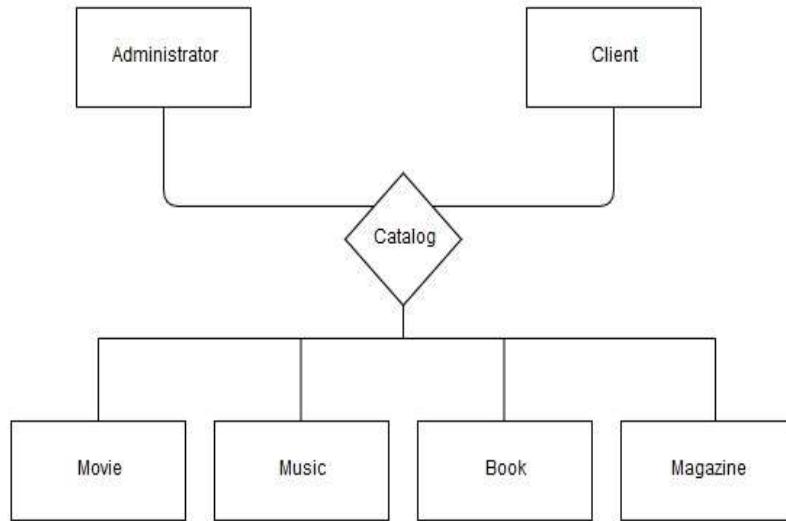


Fig 13: E/R Diagram of the Database

The below are the tables of horizontal mapping for media items.

Movie	Music
id <<PK>>	id <<PK>>
Title	Type
Director	Title
Producers	Artist
Actors	Label
Language	Release_Date
Subtitles	ASIN
Dubbed	
Release_Date	
Run_Time	

Fig 14: Horizontal mapping

SOEN 341 Team Project	Version:	4.0
Software Architecture Document	Date:	23-11-2018

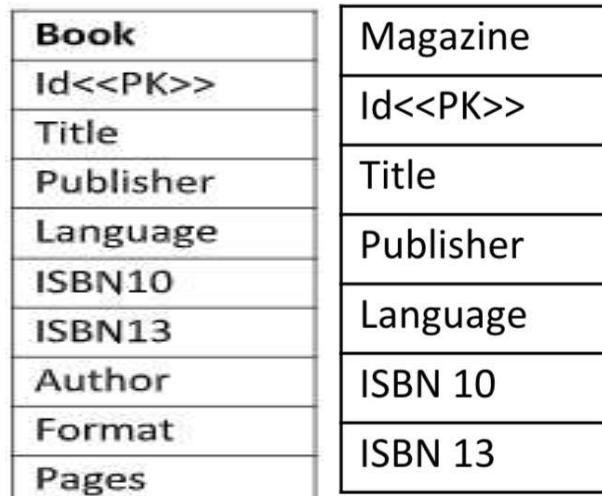


Fig 15: Horizontal mapping

Person <<Table>>
Id <<PK>>
UserName
LastName
User
Password
Email
PhoneNumber
Discriminator

Fig 16: Filtered mapping

12. Quality

Scalability:

- Description: The library web application response to high volume of clients.

SOEN 341 Team Project	Version:	4.0
Software Architecture Document	Date:	23-11-2018

- Solution: As Nodejs is an asynchronous code it won't get stuck if there are waiting time from a request or response, hence it is easily scalable.

Reliability:

- Problem: Reliability is a quality component associated with capacity of a certain system to deliver what is expected of it.
- Solution: Javascript is known to be a reliable language, specially for small applications.

Portability:

- Description: Portability is a characteristic attributed to a computer program if it can be used in any operating systems without requiring major rework.
- Solution: Since we are using javascript and .json to deploy the application it is easily accessible by many devices.

Security:

- Description: Security of web application refers to the process of protecting confidential data stored in the database from unauthorised users.
- Solution: Only clients and administrators with valid credentials can access the data from the web application and thereby protecting the confidential information from getting misused.