

Large-Scale and Multi-Structured Databases

Recap Software Engineering

Examples of System Design and Data Modelling

Prof Pietro Ducange

Exercise: Book Shop Application (I)

*Define use cases diagram and carry out the data modeling stage for a “**Book Shop Application**” (please consider also the design of a relational DB)*

Description:

The application must be used by the **employees** of a book shop, mainly for granting a complete and efficient tracing of the **books** stored in the book shop.

For each book, the following information are stored: title, author, publisher, price, number of pages, category, year of publication and number of books in stock. The application also maintains some information regarding the **author(s)** (first name, last name) and the **publisher** (name and location) to allow the salesman to query books, authors and/or publishers.

Exercise: Book Shop Application (II)

Main requirements:

The application allows the user to:

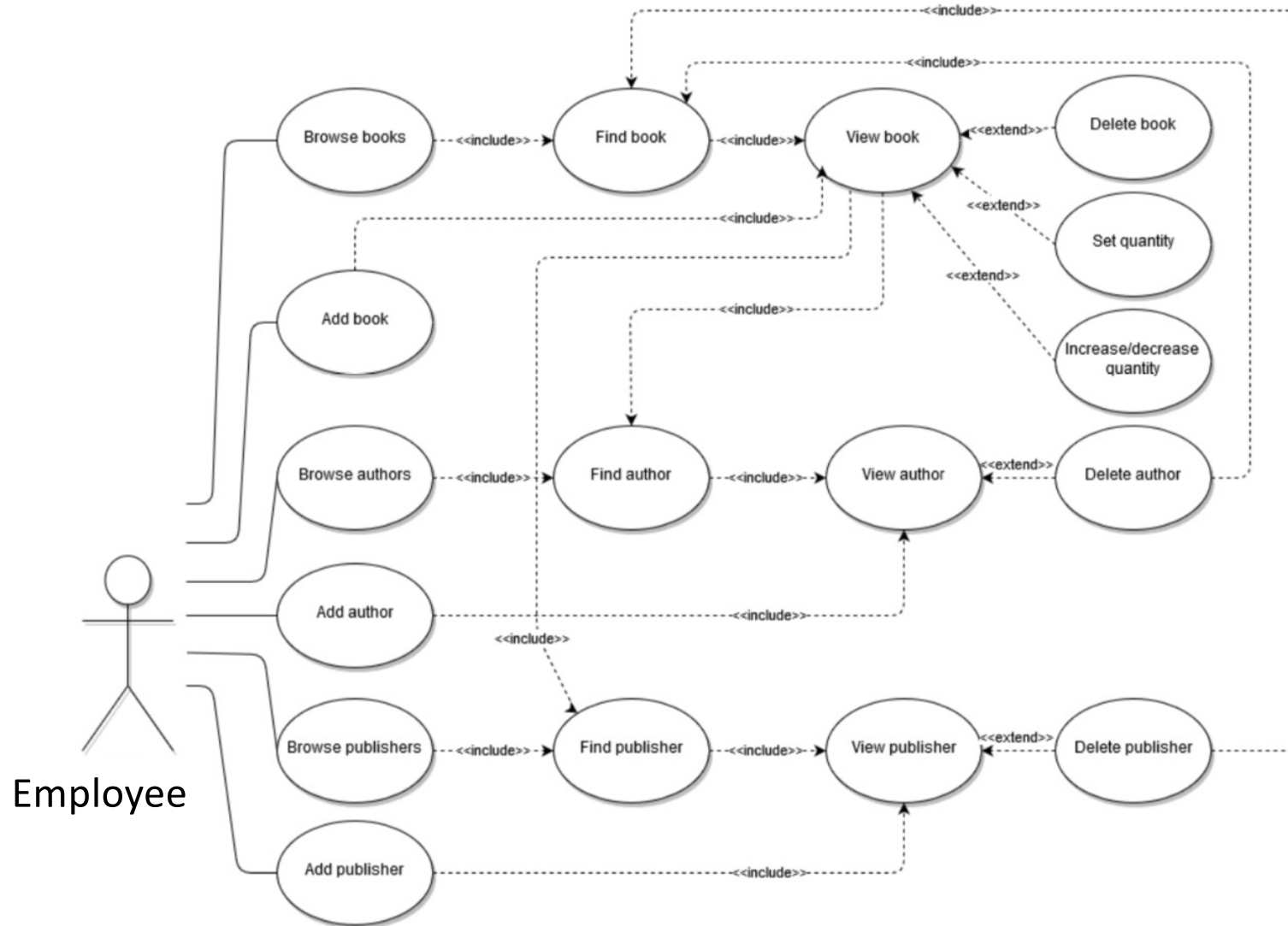
- Read the list of the books
- Insert a new book
- Remove a book
- Update the quantity of a book
- Increase/decrease the quantity of a book
- Read the list of the authors
- Insert an author
- Delete an author
- Read the list of the publishers
- Insert a publisher;
- Delete a publisher;

Some constrains:

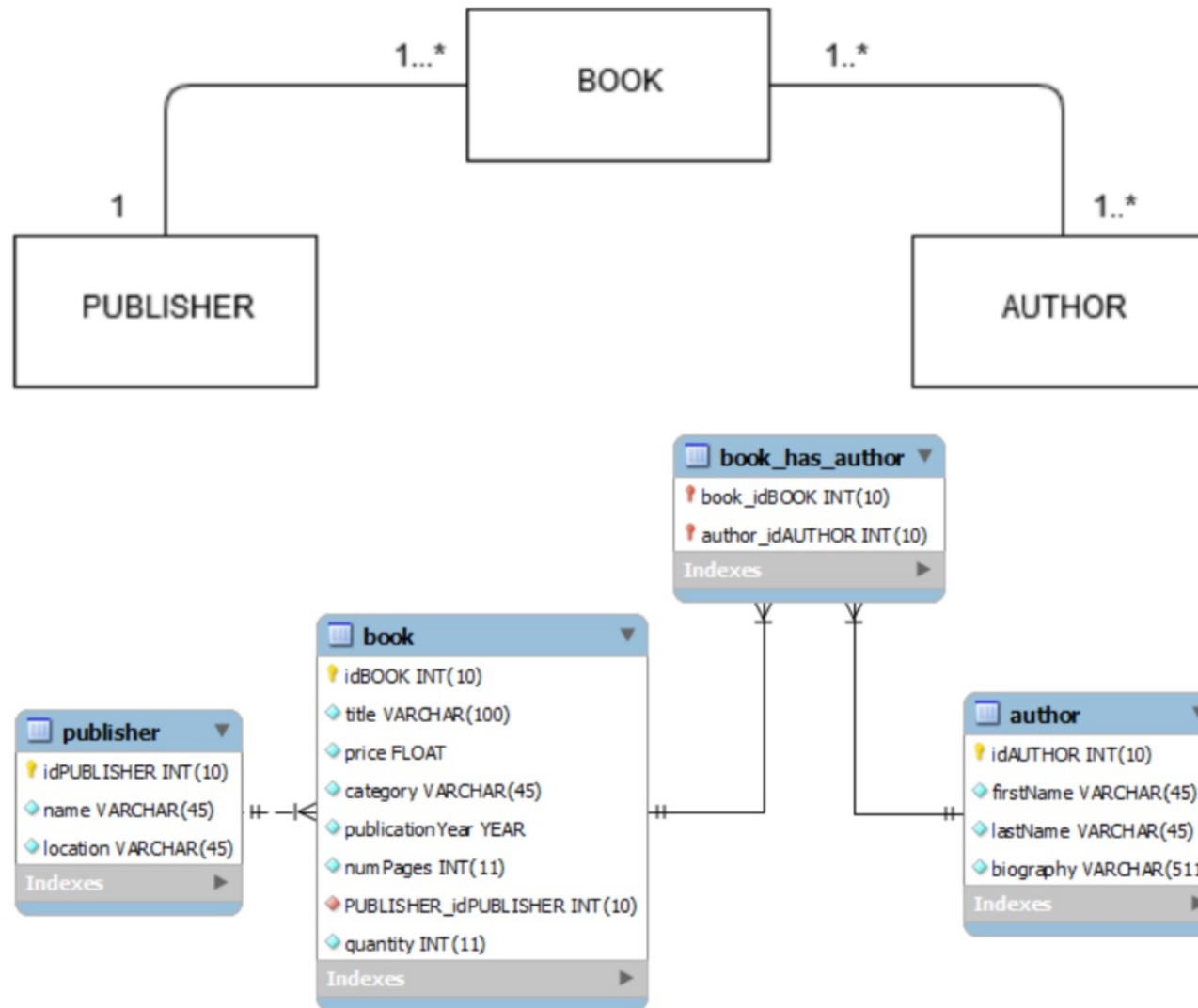
if an author, or a publisher, will be removed from the DB, also all the books associated with it must be removed.

If a book is selected (viewed) also the publisher and the author(s) must be shown.

Use Cases Diagram



UML Class Analysis and DB Scheme



Exercise II: Messaging System

Description

The application is a messaging system where registered users can create an account, exchange text messages and make groups.

A registered user can initiate a chat with another user. Moreover he/she can create a new group chat, acting as the group administrator, and send messages to the chats he/she belongs to, as well as receiving messages from those chats. He can also leave a group.

A group administrator can add and remove new users to the group. He cannot assign his/her privilege to another user in the group. A group administrator cannot leave the chat group that he/she created but can delete it.

Every time a user views a chat, all the latest messages from the chat are fetched from the server and shown to the him/her.

Exercise II: Messaging System

Main requirements:

An ***anonymous user*** must be able to:

- Register in order to become a registered user

A ***registered user*** must be able to:

- Retrieve the list of chats he/she is a member of
- Read the messages of a chat
- Send a message to a chat
- Create a new private chat
- ***Create a new group chat***
- Leave a group chat

A ***group admin*** must be able to:

- Add members to the group chat
- Remove members from the group chat
- Delete the group chat

When a user creates a new group chat, he/she automatically becomes the admin of the chat.

The ***time-based event*** must be able to update the system on regular intervals to show chat updates, i.e. new chats or new messages from the current chats, if any.

```
graph TD
    subgraph Actors
        direction TB
        AnonymousUser[Anonymous User]
        User((User))
        ChatAdmin[Chat admin]
    end

    subgraph UseCases
        direction TB
        Register((Register))
        Login((Login))
        Logout((Logout))
        BrowseChatList((Browse chat list))
        FindChats((Find chats))
        CreateChat((Create chat))
        BrowseChatMessages((Browse chat messages))
        FindMessages((Find messages))
        ViewMessage((View message))
        UpdateMessages((Update messages))
        ViewChat((View chat))
        SendMessage((Send message))
        LeaveChat((Leave chat))
        FindMembers((Find members))
        ViewMember((View member))
        DeleteChatStar((Delete chat*))
        AddMemberStar((Add member*))
        RemoveMemberStar((Remove member*))
        BrowseMemberList((Browse member list))
    end

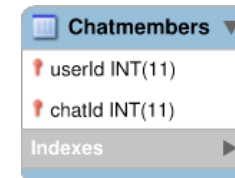
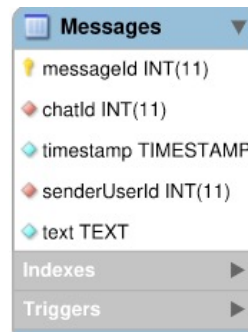
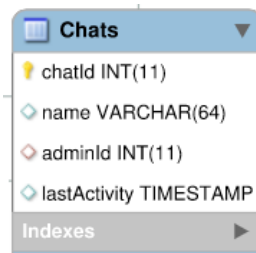
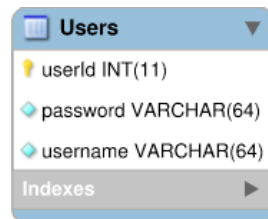
    AnonymousUser --- Register
    User --- Login
    User --- Logout
    User --- BrowseChatList
    User --- FindChats
    User --- CreateChat
    ChatAdmin --- DeleteChatStar
    ChatAdmin --- AddMemberStar
    ChatAdmin --- RemoveMemberStar

    FindChats -.->|<<include>>| ViewChat
    CreateChat -.->|<<include>>| ViewChat
    BrowseChatList -.->|<<include>>| FindChats
    BrowseChatMessages -.->|<<include>>| FindMessages
    FindMessages -.->|<<include>>| ViewMessage
    UpdateMessages -.->|<<extend>>| ViewChat
    SendMessage -.->|<<extend>>| ViewChat
    LeaveChat -.->|<<extend>>| ViewChat
    FindMembers -.->|<<include>>| ViewMember
    ViewMember -.->|<<include>>| ViewChat
    BrowseMemberList -.->|<<include>>| FindMembers
    AddMemberStar -.->|<<include>>| ViewMember
    RemoveMemberStar -.->|<<extend>>| ViewMember
    DeleteChatStar -.->|<<extend>>| ViewChat
    BrowseChatMessages -.->|<<extend>>| ViewChat
    TimerEvent[Timer event] -.->|<<extend>>| FindChats
```


UML Class Analysis and DB Scheme



DB Tables



Acknowledgements

Credits to:

- Davide Coccomini
- Luigi Gjoni
- Marilisa Lippini
- Lorenzo Nelli
- Federico Fregosi
- Mirko Laruina
- Riccardo Mancini
- Gianmarco Petrelli

Students of 2019-2020 Academic Year.

They designed and developed the two applications discussed in the proposed exercises for their TASK1.