Class Design Solution

for

Book Rental and Sales System

Version 1.1 approved

Prepared by Trinh Thai Linh
Le Quang Chinh
Pham Tung Duong
Do Tien Dung
Nguyen Hoang Anh Dung

Group 7

November 30, 2023

Copyright © 1999 by Karl E. Wiegers. Permission is granted to use, modify, and distribute this document.

Table of Contents

1. Introduction 1

1.1 Purpose 1

1.2 Intended Audience and Reading Suggestions 1

1.3 Product Scope 2

1.4 References 2

2. Class Design 3

2.1 With Subsystem Interface 3

2.1.1 Chat Service 3

2.1.2 Register for a Rental Plan 3

2.1.3 Read Book 4

2.1.4 Buy Book 4

2.1.5 Sell Used Book 5

2.1.6 Cancel Order 6

2.1.7 View Order 7

2.1.8 Request Refund 8

2.1.9 Search Book 9

2.2 With Persistency Mechanism 10

2.2.1 Buy Book 10

2.2.2 Cancel Order 11

2.2.3 Edit Profile 12

2.2.4 Mange Post 13

2.2.5 Request Refund 14

2.2.6 Register for a Rental Plan 15

2.2.7 Review Order 16

2.2.8 Sign Up 16

2.2.9 Manage Book 17

2.3 With Security 18
2.3.1 Sign In 18

Table of Figures

Figure 2-1: Chat Service Subsystem Class Design (with interface). 3
Figure 2-2: RentalPlan Class Design (with interface). 3
Figure 2-3: Read Book Class Design (with interface). 4
Figure 2-4: Buy Book Class Design (with interface). 5
Figure 2-5: Sell Used Book Class Diagram (with interface). 6
Figure 2-6: Cancel Order Class Diagram (with interface). 7
Figure 2-7: View Order Class Diagram (with interface). 8
Figure 2-8: Request Refund Class Diagram (with interface). 9
Figure 2-9: Search Book Class Diagram (with interface). 10
Figure 2-10: Buy Book Class Diagram (with Persistency). 11
Figure 2-11: Cancel Order Class Diagram (with Persistency). 12
Figure 2-12: Edit Profile Class Diagram (with Persistency). 13
Figure 2-13: Forum Service Class Diagram (with Persistency). 14
Figure 2-14: Request Refund Class Diagram (with Persistency). 15
Figure 2-15: Register for a Rental Plan Class Diagram (with Persistency). 16
Figure 2-16: Review Order Class Diagram (with Persistency). 16
Figure 2-17: Sign Up Class Diagram (with Persistency). 17
Figure 2-18: Update Book Class Diagram (with Persistency). 17
Figure 2-19: Sign In Class Diagram (with Security). 18

Revision History

Name	Date	Reason For Changes	Version
Trinh Thai Linh	30-Nov-23	Create Document Template	1.0
Trinh Thai Linh	05-Dec-23	Add Contents to Document	1.1
Le Quang Chinh	08-Dec-23	Add Diagrams to Document	1.2
Pham Tung Duong	08-Dec-23	Add Diagrams to Document	1.2

1. Introduction

1.1 Purpose

This is a report prepared by Group 7 (Object-oriented Analysis and Design Class 20, 22-23) to Describe the Run-time Architecture Solution for Book Rental and Sales System and is written based on the reporting format "IEEE Std 830-1998 IEEE Recommended Practice for Software Requirements Specifications".

1.2 Intended Audience and Reading Suggestions

The different types of reader that the document is intended for are:

- **Designers:** Design the classes and the subsystems architecture that satisfies the requirements specified in this SRS.
- **Developers:** Implement codes from the design and these documents. They use this document as a blueprint, and it makes sure that, firstly, their code works properly, and secondly, it is in sync with others' codes.
- **Tester:** Use this document to understand the requirements. They have to create a test plan that outlines the scope of testing, testing objectives, test environment setup, resources required, and a schedule.
- **Documentation writers:** Ensure that the documentation aligns with the software's requirements as specified in the Software Requirements Specification (SRS). Verify that the documented information is accurate and complete.

1.3 Product Scope

The software, “Book Rental and Sales System”, is designed to provide a platform for users to rent, purchase, read books in both physical and electrical form. Additionally, it includes a forum where users can discuss with other readers about the books they purchased, rented or simply ones that captured their interest. The system aims to enhance the reading experience and promote the reading community engagements.

1.4 References

[1] Form of presentation IEEE. IEEE Std 830-1998 IEEE Recommended Practice for Software Requirements Specifications. IEEE Computer Society, 1998.

[2] System Requirements Specification Content and Format Standard, which specifies the content and format of this specification.

[3] Previous Project of Object-Oriented Analysis and Design Course, The Cargo Carriage System.

2. Class Design

2.1 With Subsystem Interface

2.1.1 Chat Service

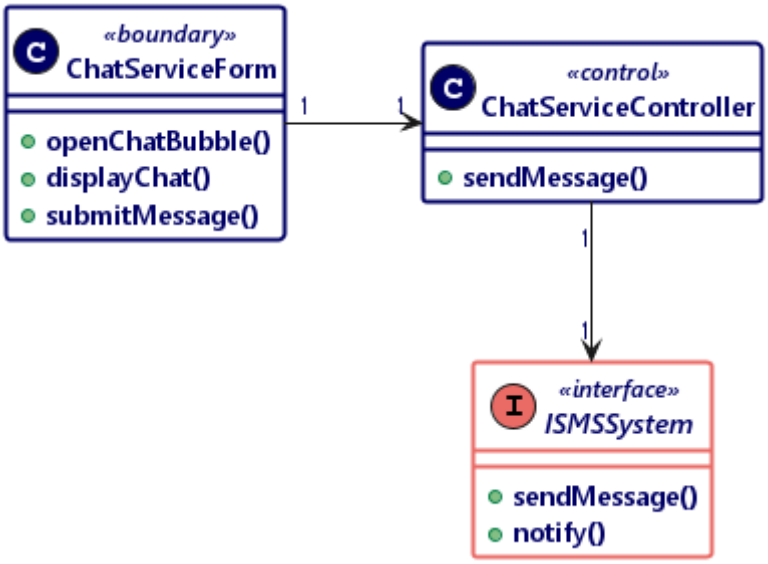

Figure 2-1: Chat Service Subsystem Class Design (with interface).

2.1.2 Register for a Rental Plan

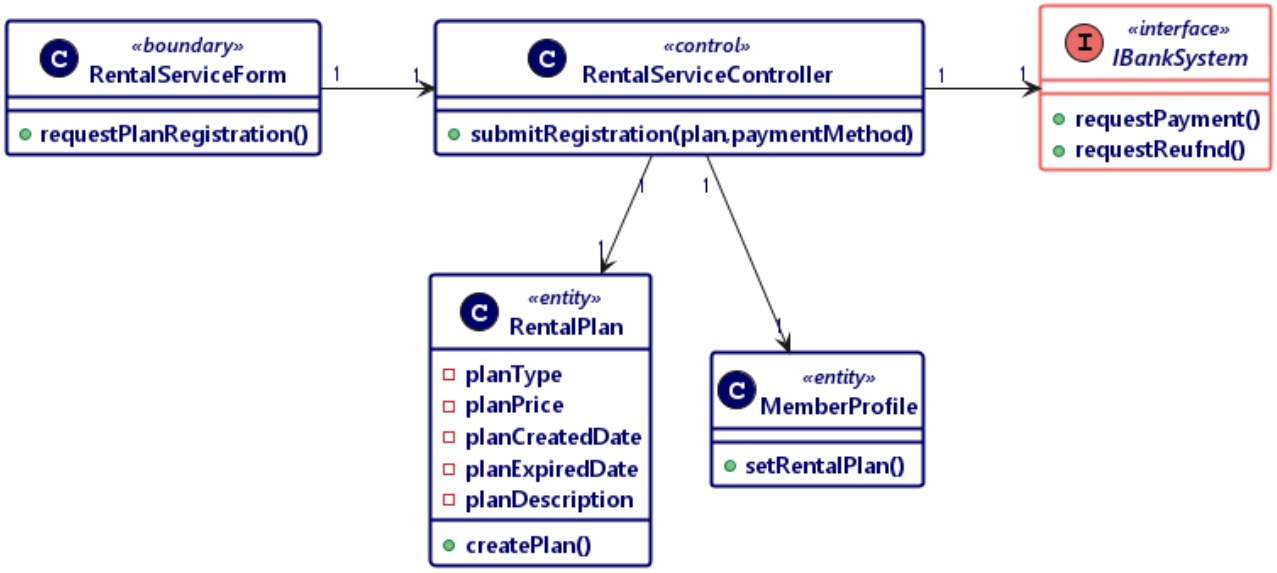

Figure 2-2: RentalPlan Class Design (with interface).

2.1.3 Read Book

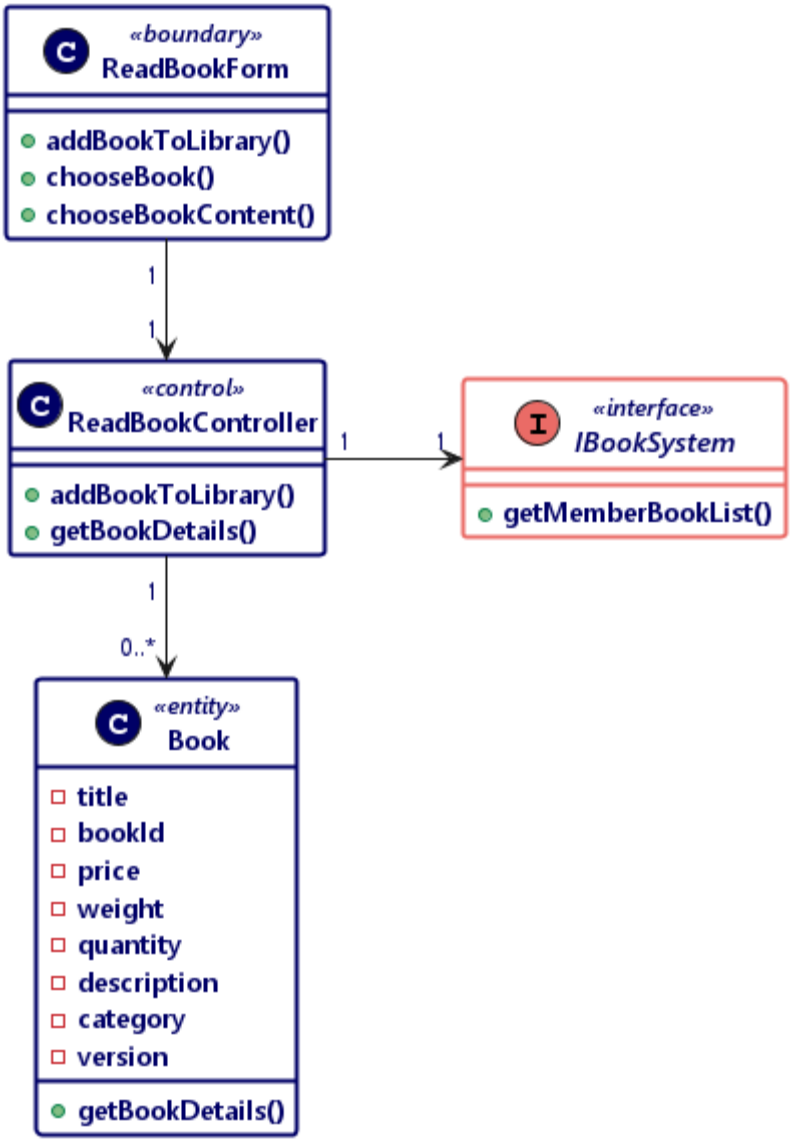

Figure 2-3: Read Book Class Design (with interface).

2.1.4 Buy Book

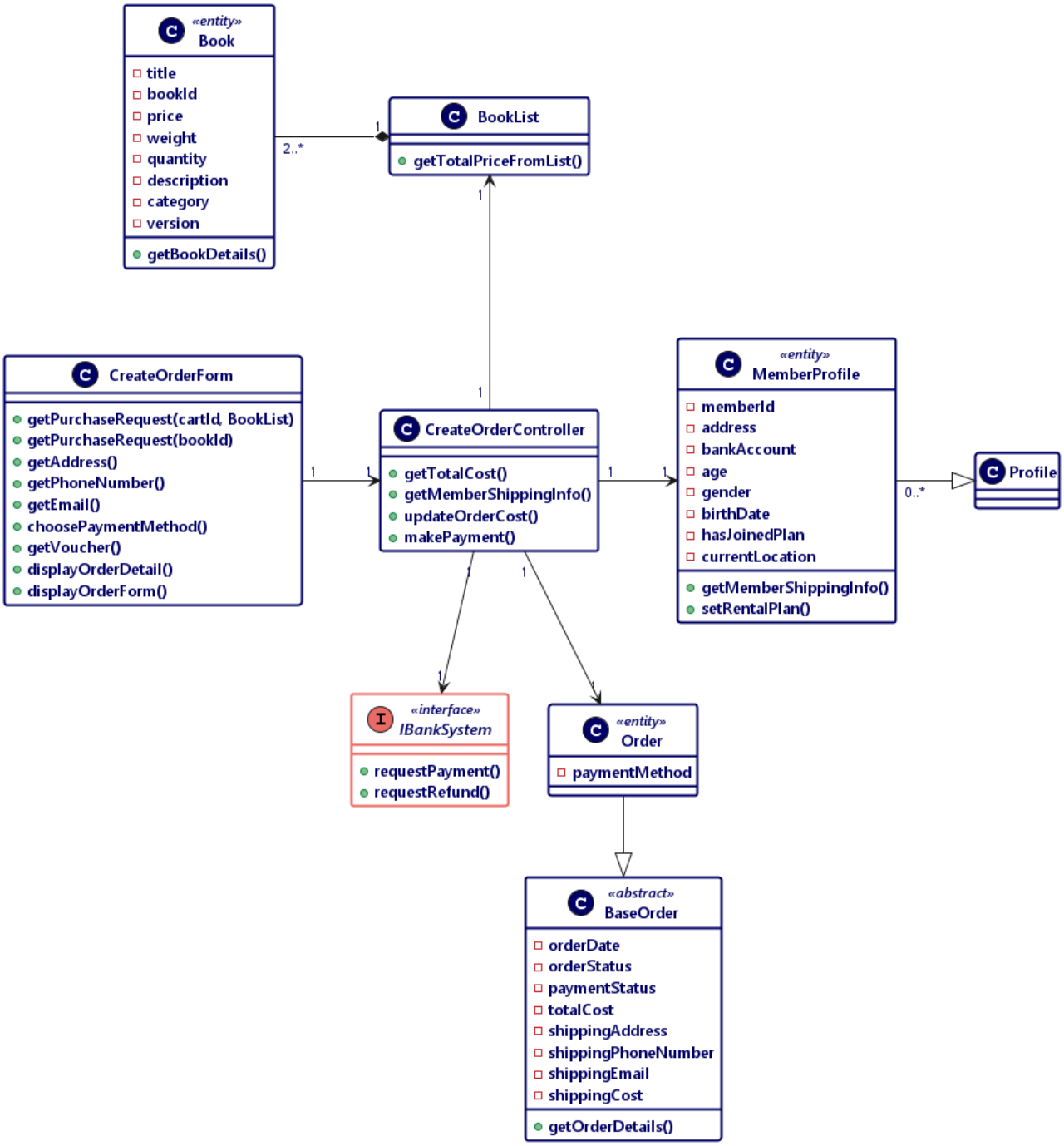

Figure 2-4: Buy Book Class Design (with interface).

2.1.5 Sell Used Book

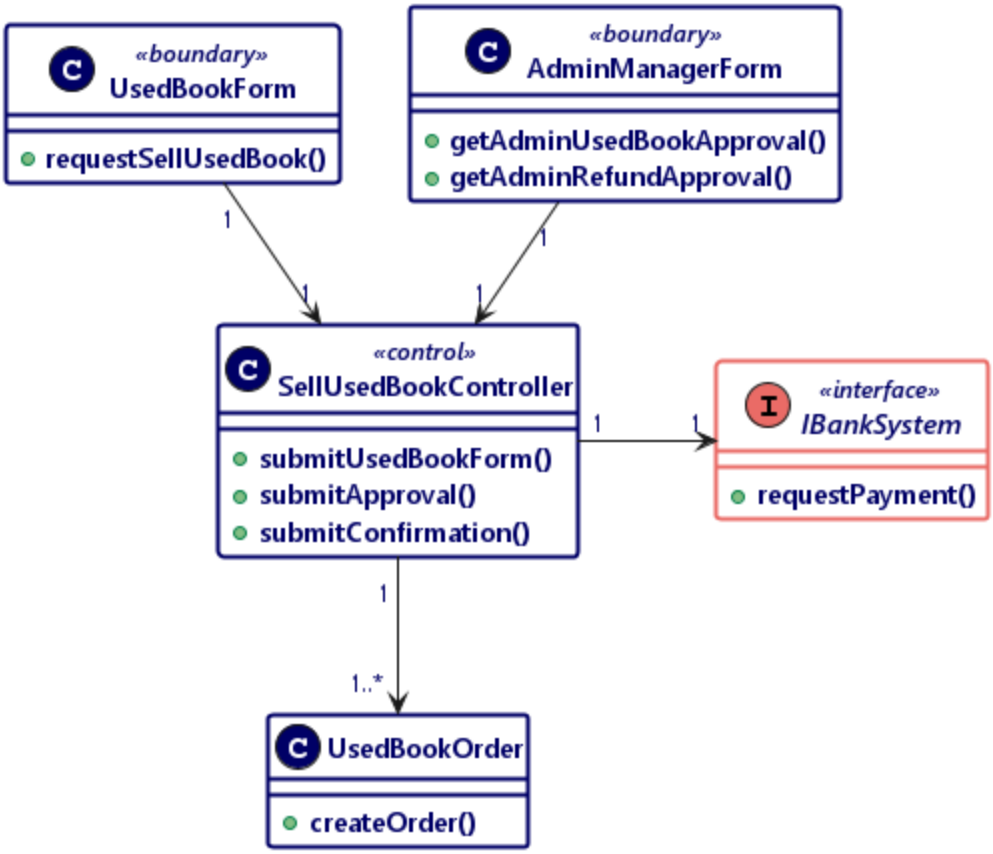

Figure 2-5: Sell Used Book Class Diagram (with interface).

2.1.6 Cancel Order

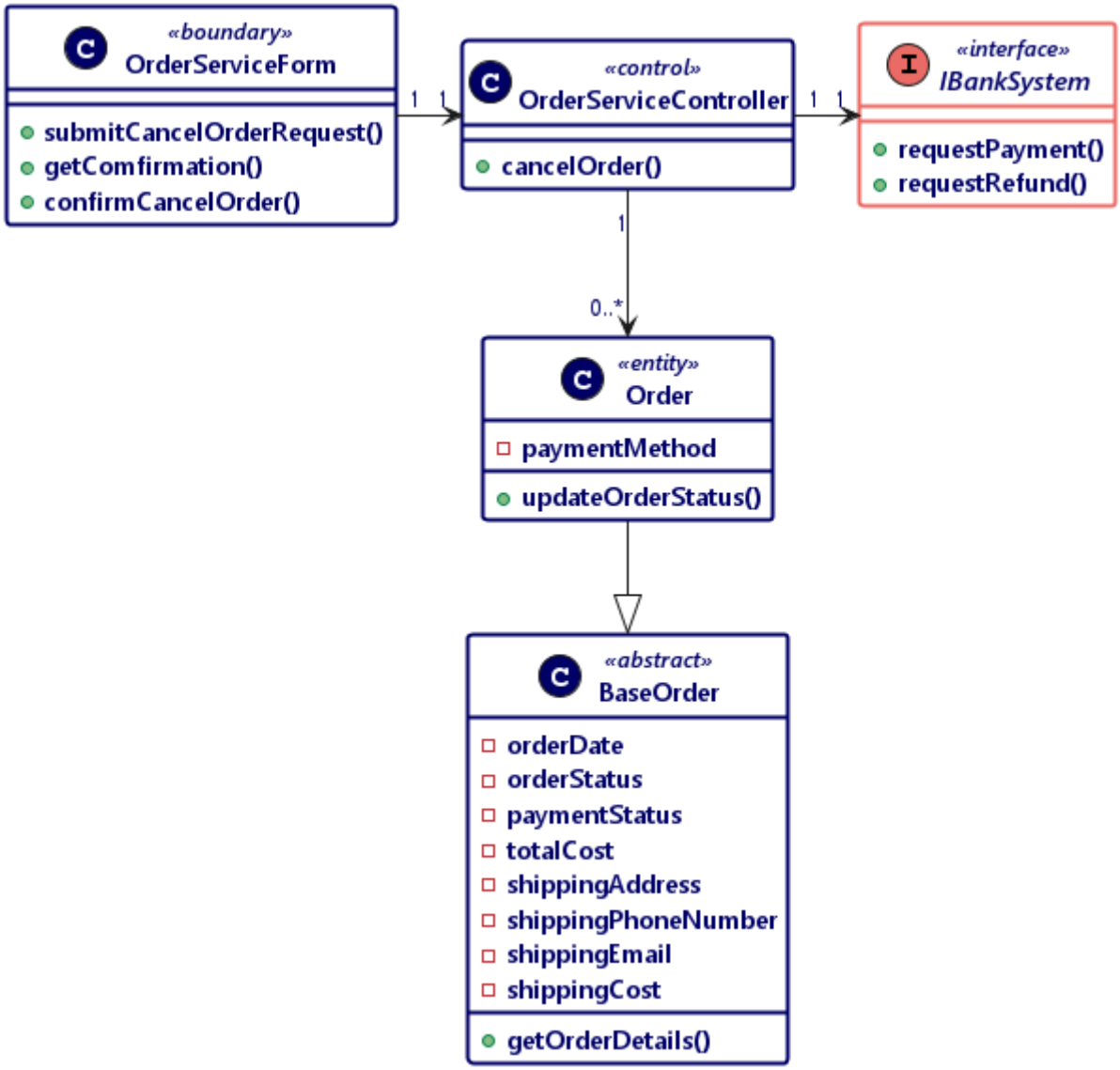

Figure 2-6: Cancel Order Class Diagram (with interface).

2.1.7 View Order

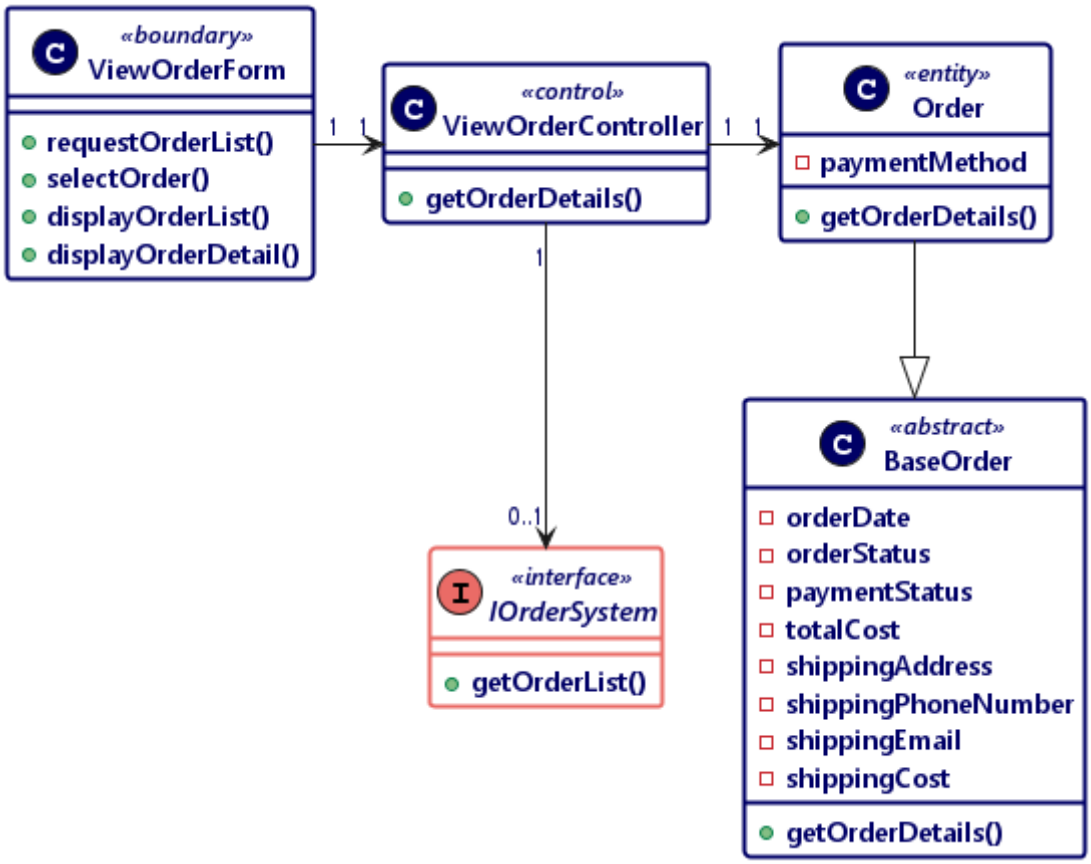

Figure 2-7: View Order Class Diagram (with interface).

2.1.8 Request Refund

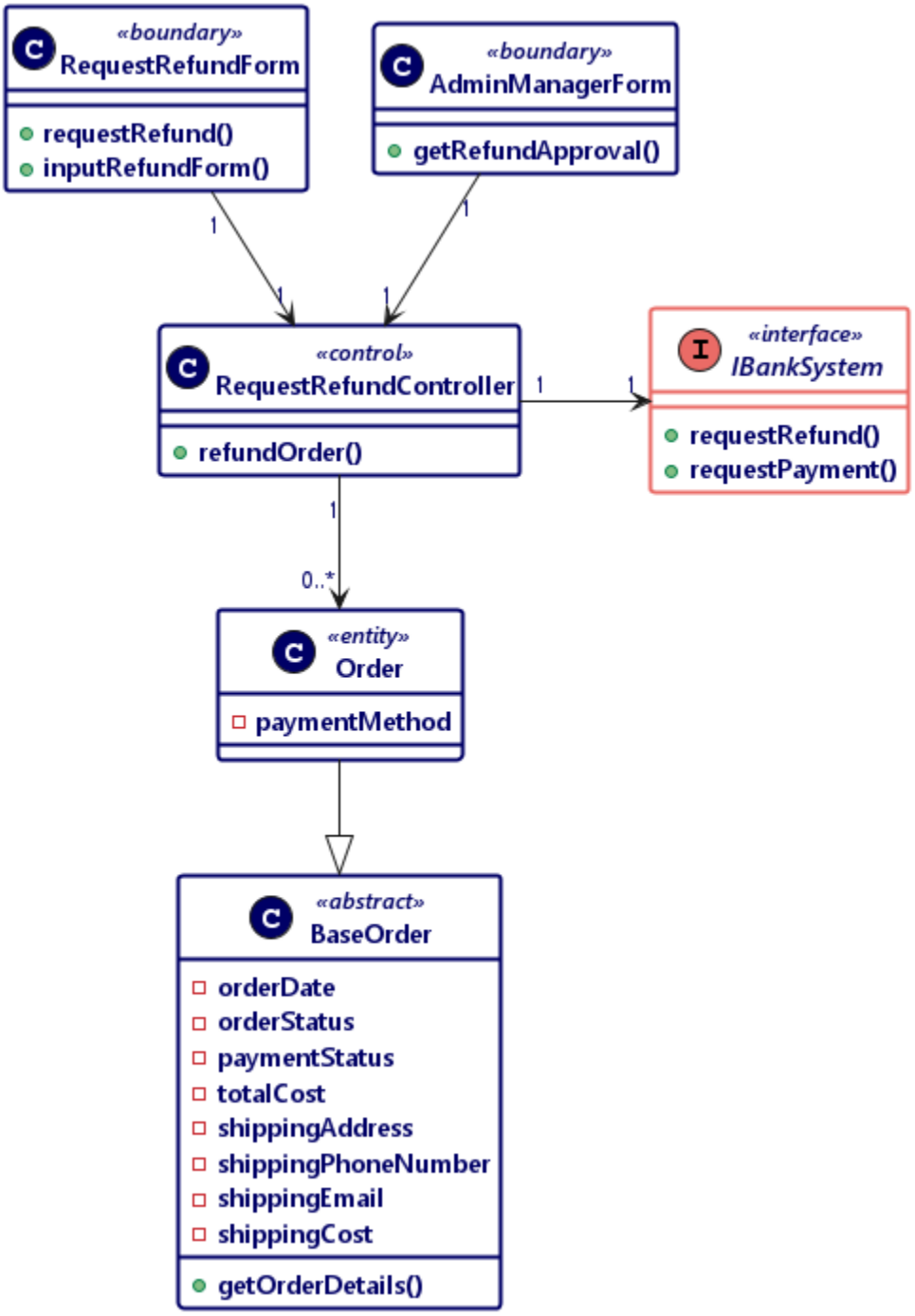

Figure 2-8: Request Refund Class Diagram (with interface).

2.1.9 Search Book

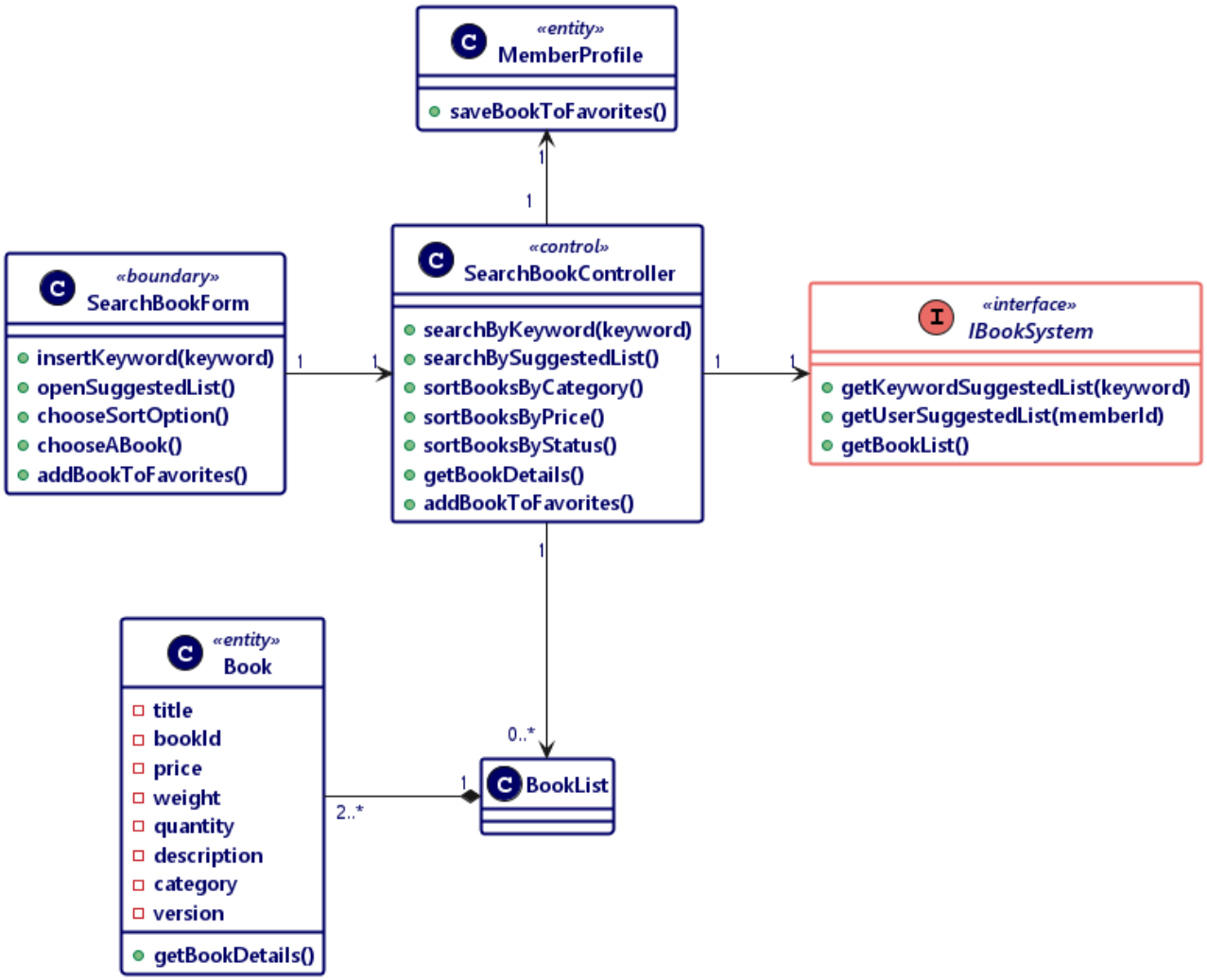

Figure 2-9: Search Book Class Diagram (with interface).

2.2 With Persistency Mechanism

2.2.1 Buy Book

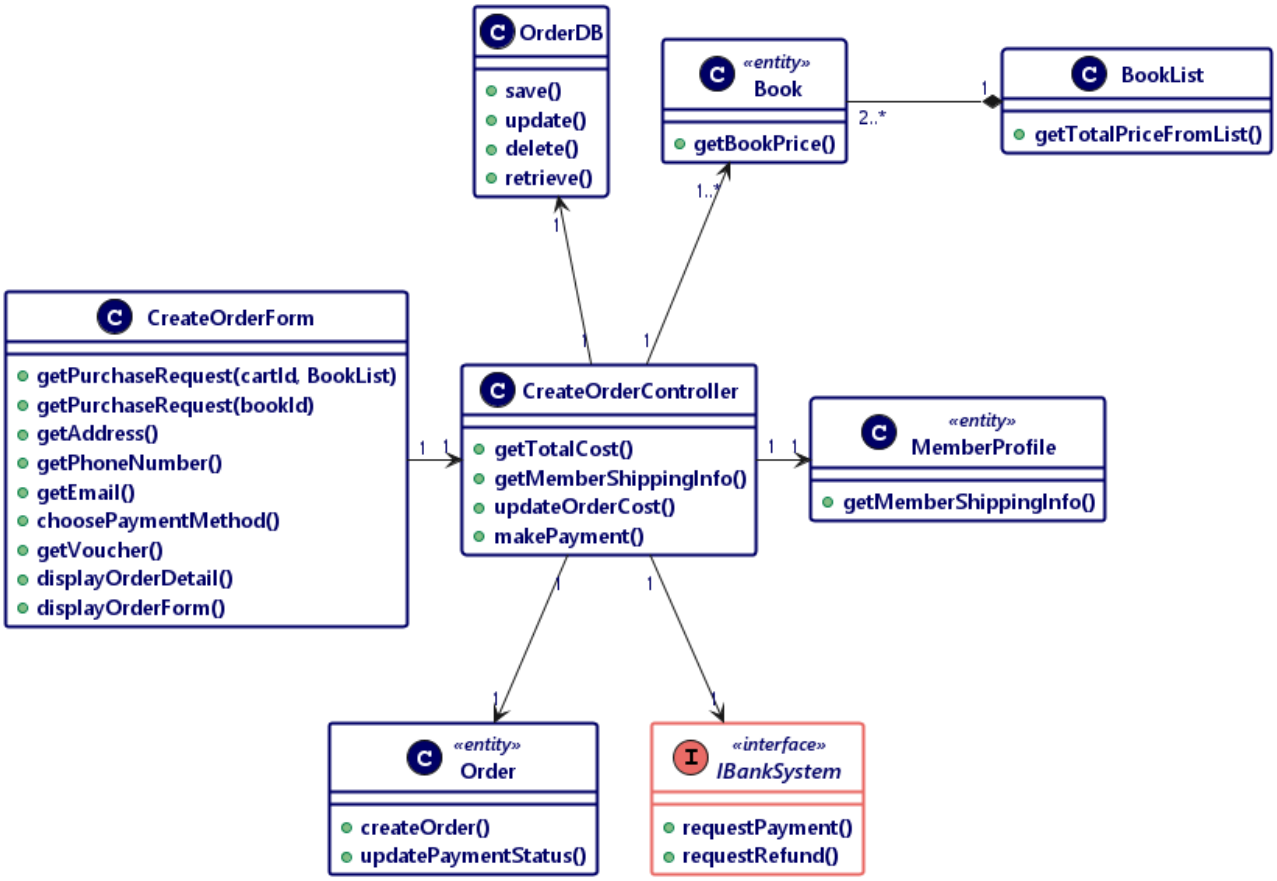

Figure 2-10: Buy Book Class Diagram (with Persistency).

2.2.2 Cancel Order

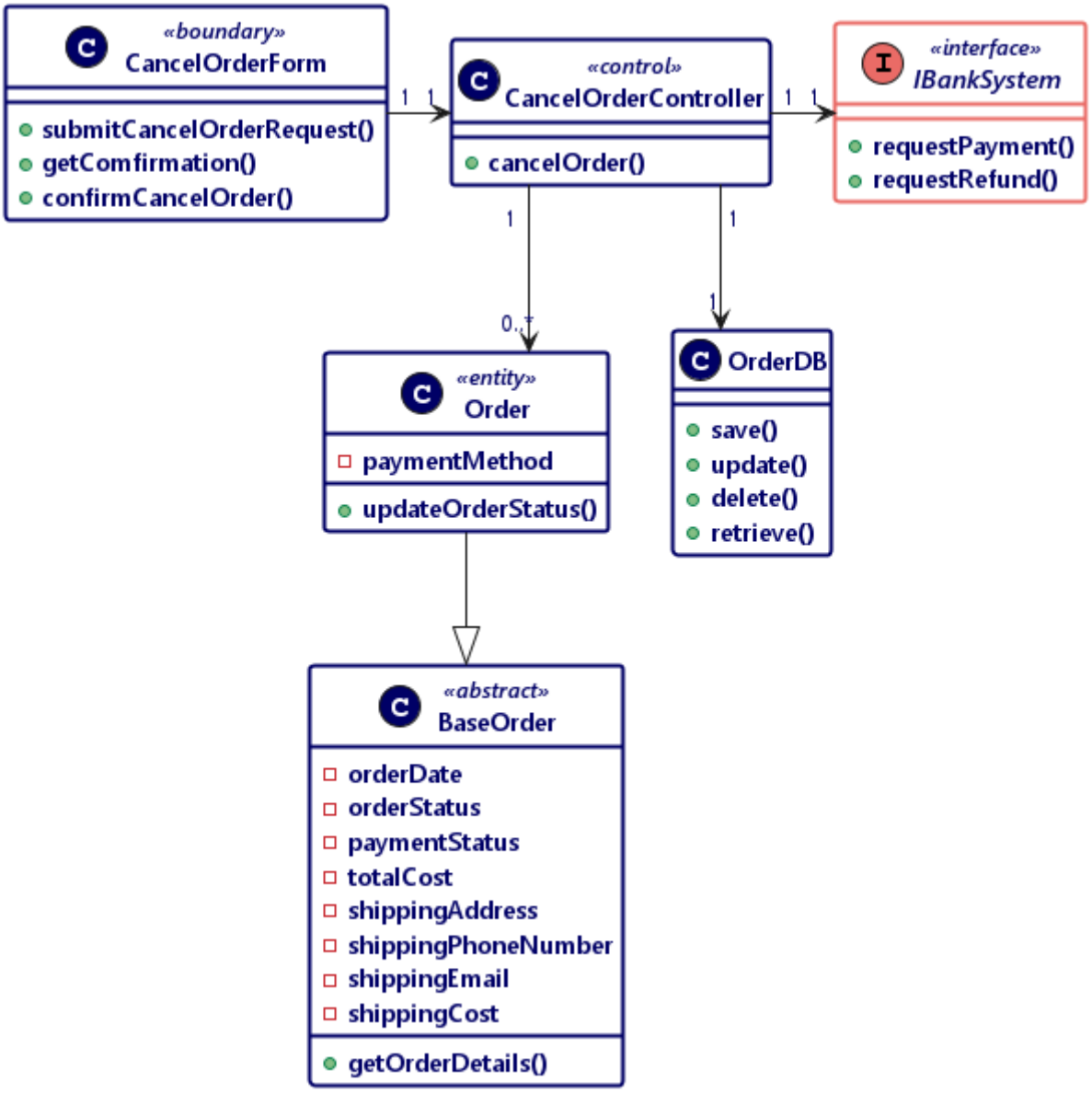

Figure 2-11: Cancel Order Class Diagram (with Persistency).

2.2.3 Edit Profile

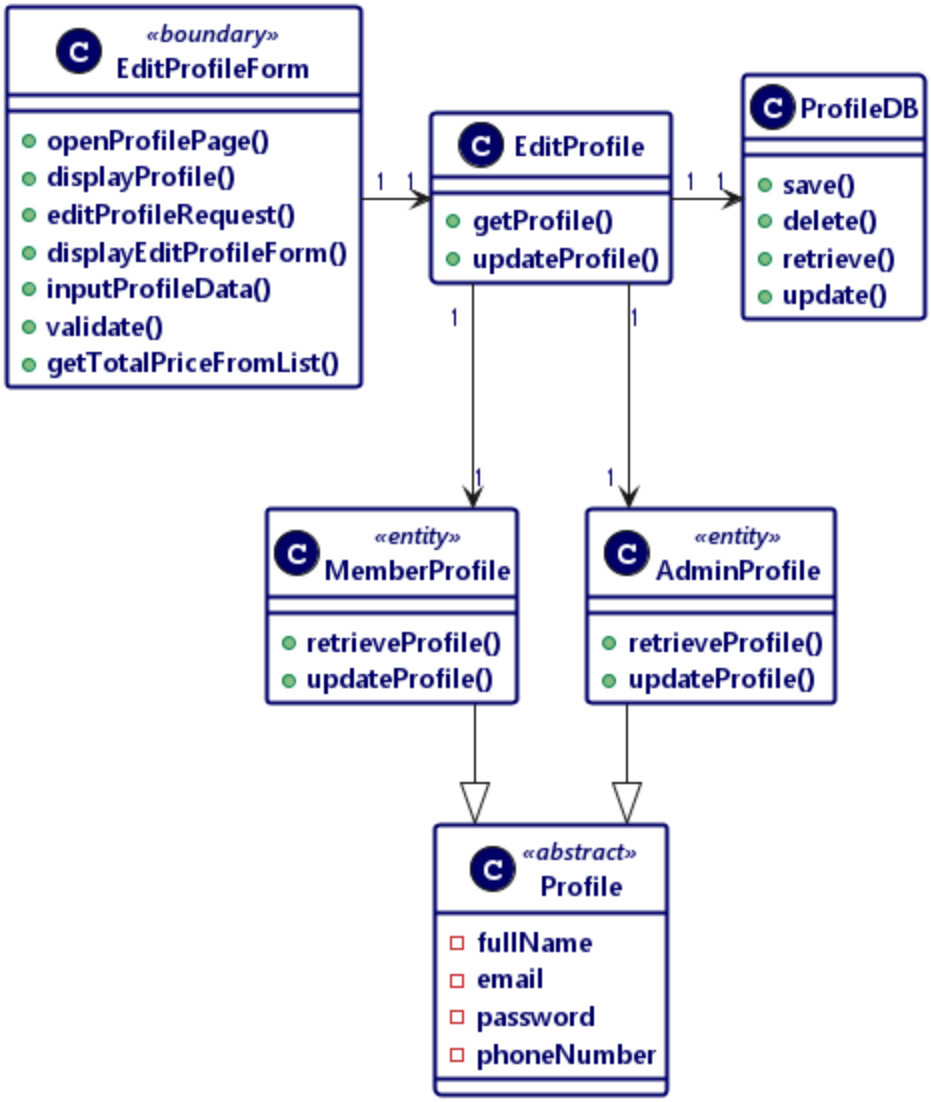

Figure 2-12: Edit Profile Class Diagram (with Persistency).

2.2.4 Mange Post

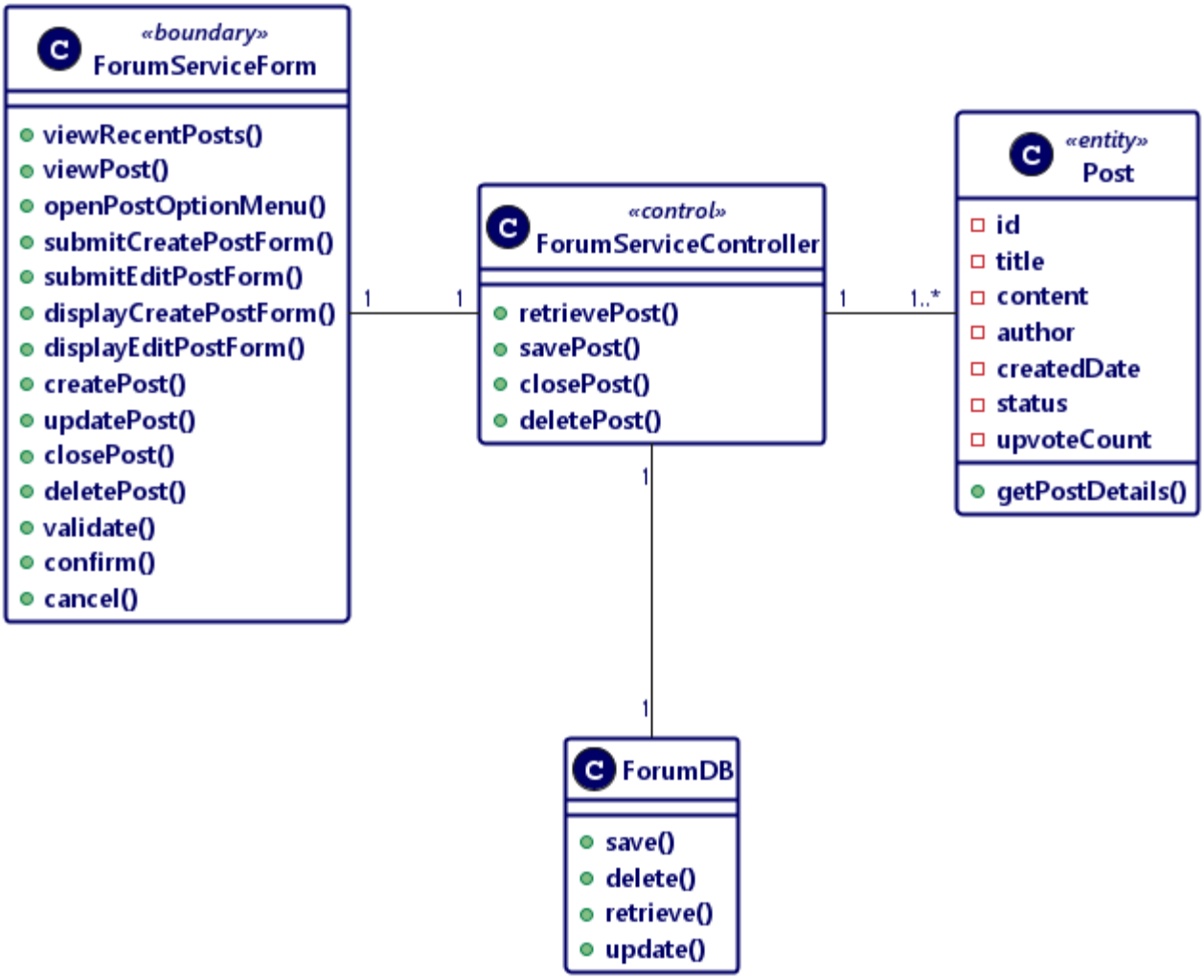

Figure 2-13: Forum Service Class Diagram (with Persistency).

2.2.5 Request Refund

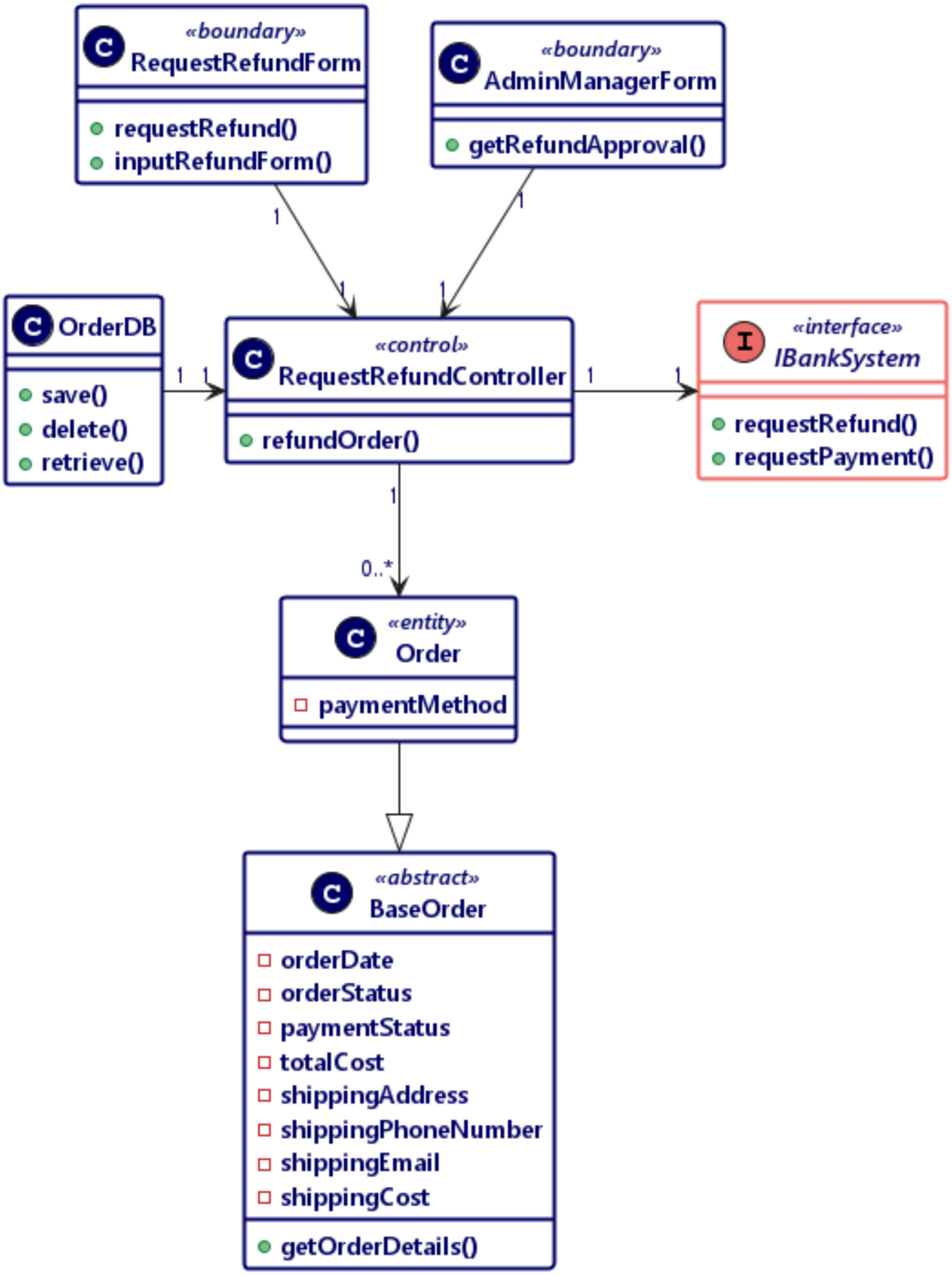

Figure 2-14: Request Refund Class Diagram (with Persistency).

2.2.6 Register for a Rental Plan

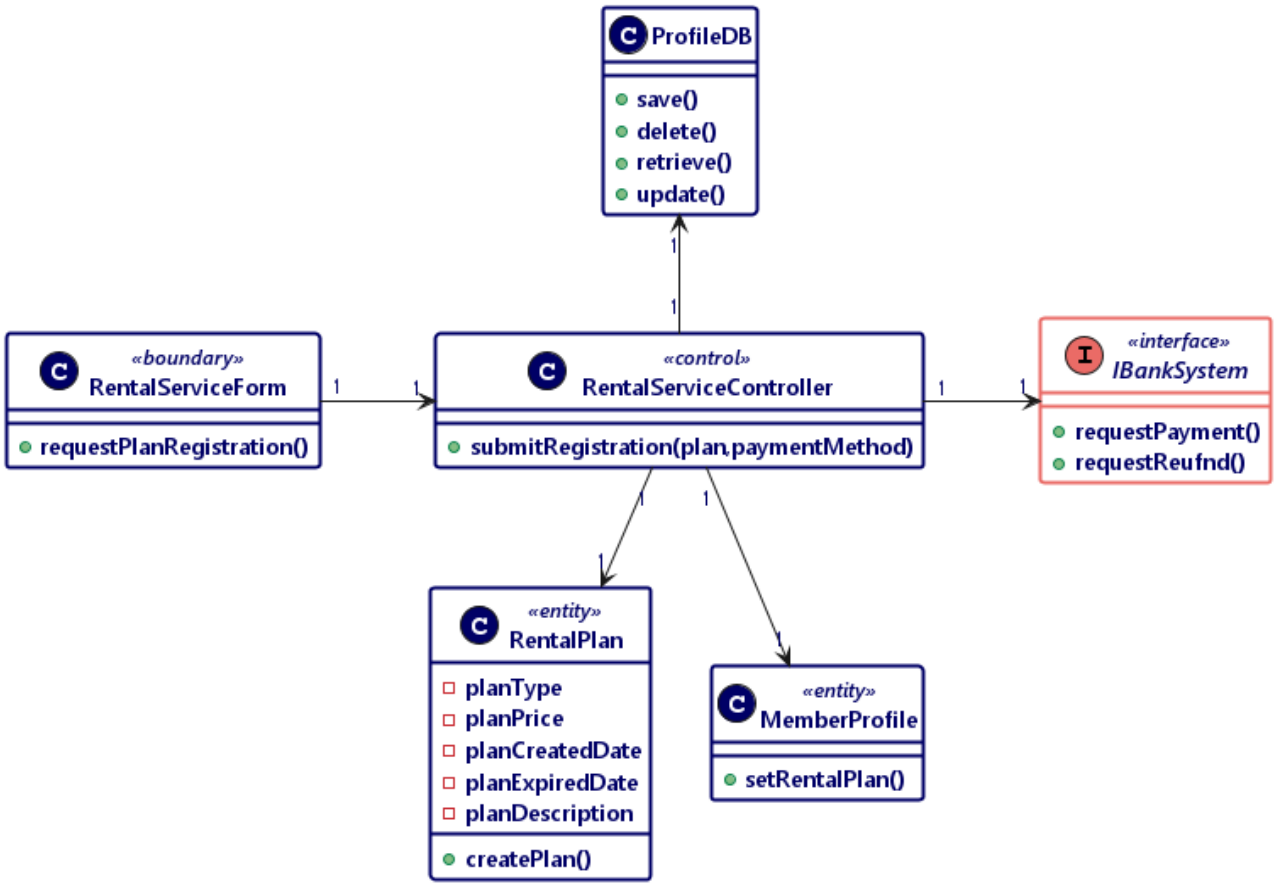

Figure 2-15: Register for a Rental Plan Class Diagram (with Persistency).

2.2.7 Review Order

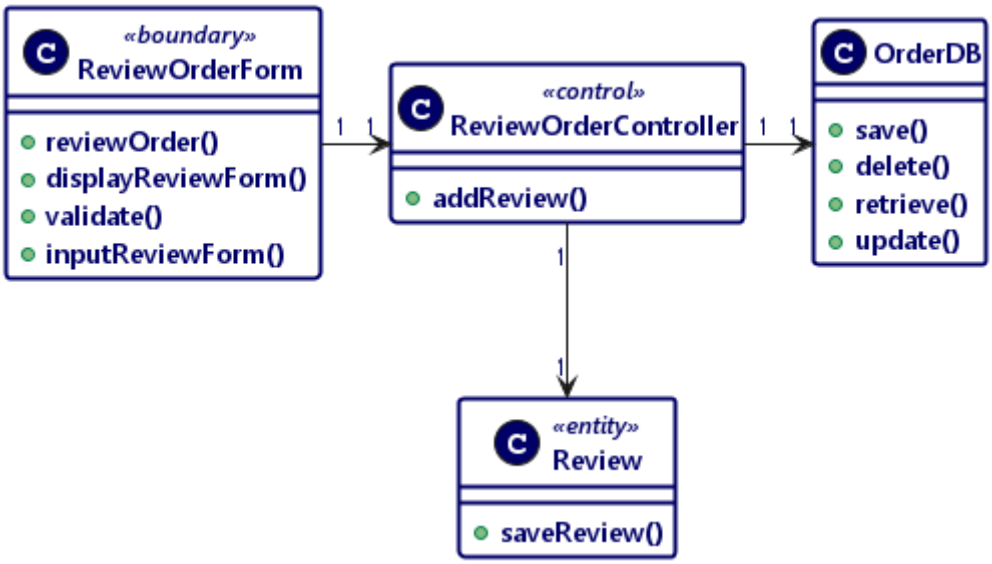

Figure 2-16: Review Order Class Diagram (with Persistency).

2.2.8 Sign Up

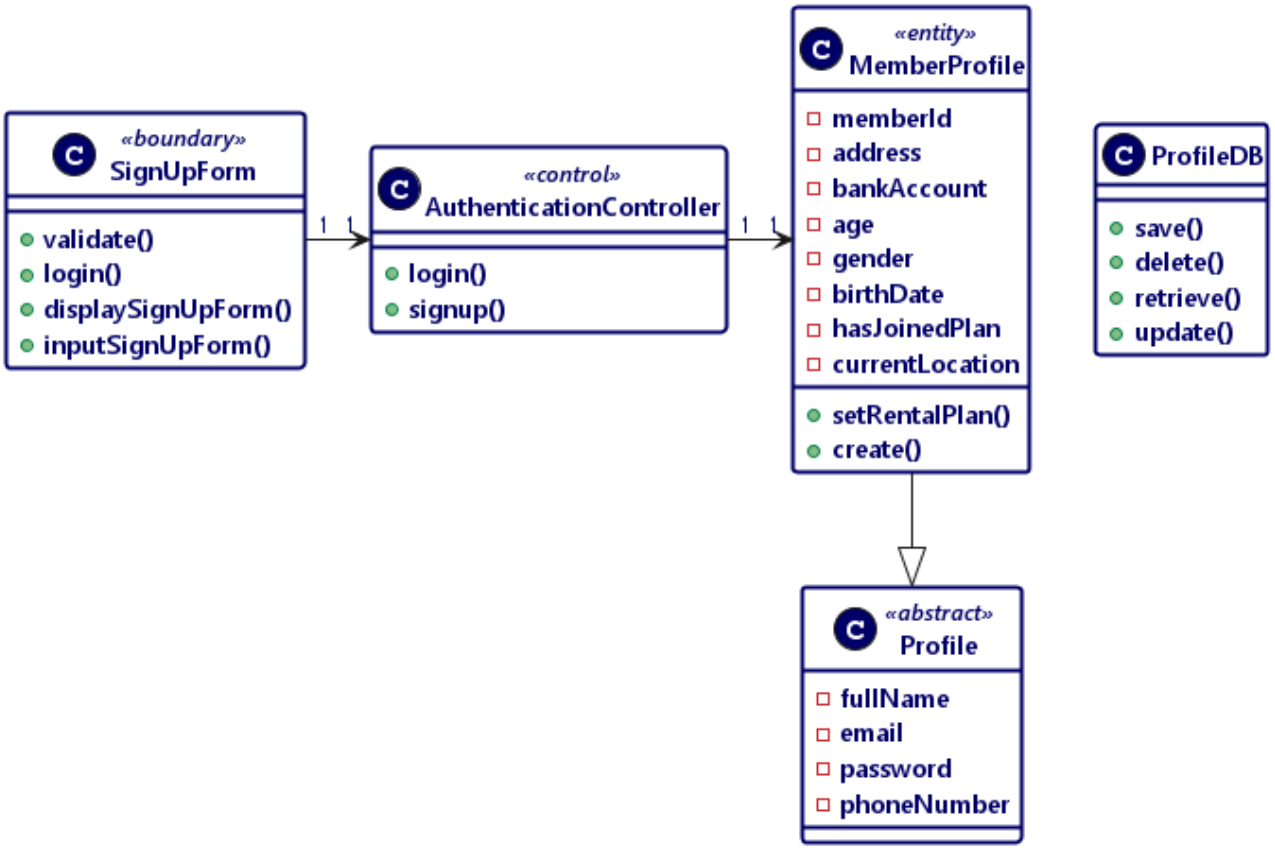

Figure 2-17: Sign Up Class Diagram (with Persistency).

2.2.9 Manage Book

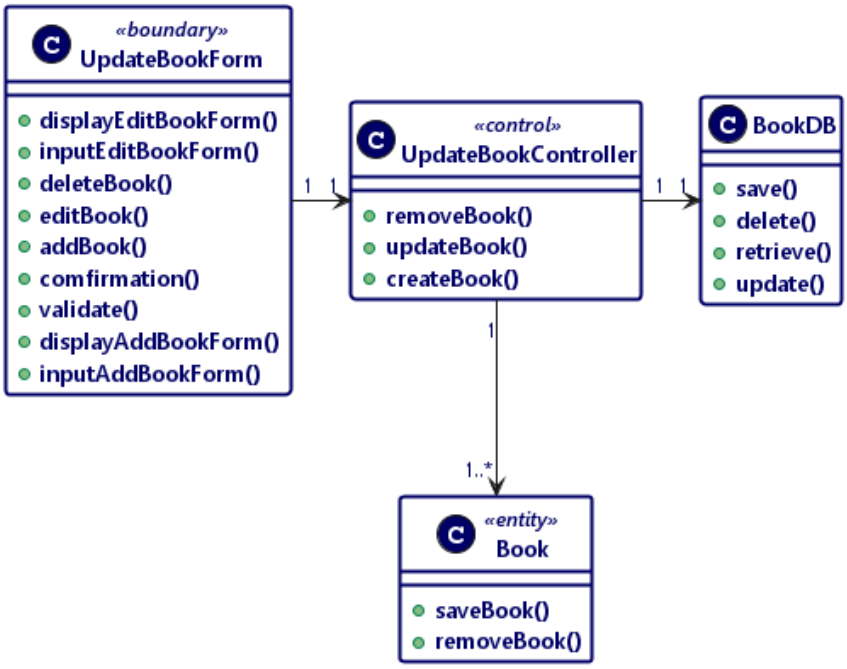

Figure 2-18: Update Book Class Diagram (with Persistency).

2.3 With Security

2.3.1 Sign In

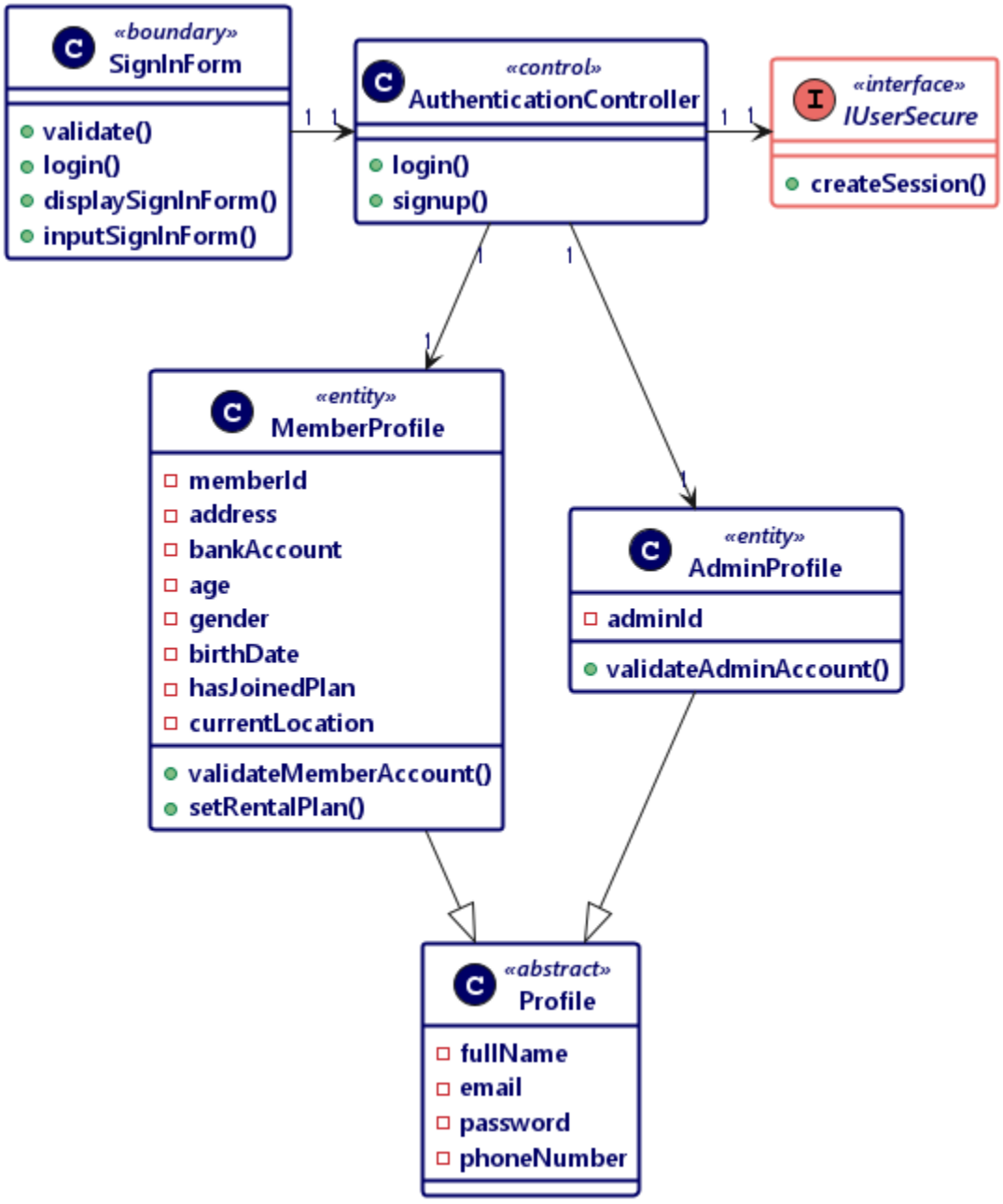

Figure 2-19: Sign In Class Diagram (with Security).