

## Отчёт

### Диаграмма классов и их взаимодействия

Описание Sequence Diagram Order из рис. 1.

- зарегистрированный пользователь может добавить книги в корзину или удалить их из неё, запросы обрабатывает IndexController взаимодействуя с DBController через функции AddBookToShoppingCart(userID, bookID) и RemoveBookFromShoppingCart(userID, bookID);
- пользователь может оформить заказ, IndexController начнёт отображать форму для заполнения данных по заказу DisplayOrderForm(selectedBooks), чтобы отобразить товары заказа OrderFromController запрашивает у DB через DBController все книги из корзины текущего пользователя getBooksFromShoppingCart(userID), после ответа отображает их на экране;
- если пользователь отменяет заказ, OrderFromController передаёт управление IndexController через displayIndex(), после чего пользователь увидит главный экран, товары из корзины не пропадают;
- если пользователь ввёл не все данные и попробовал оформить, поля подсветятся красным и форма останется;
- если пользователь ввёл некорректные данные, заказ не пройдёт валидацию (checkOrder(order) через OrderValidationController), не будет оформлен и ему придёт соответствующее сообщение (purchase notification message говорит о состоянии заказа);
- если же пользователь ввёл корректные данные и подтвердил заказ, а checkOrder(order) сообщила о том что всё валидно, OrderValidationController через DBController обновляет DB (UpdateDB(order)), получив ответ оповещает пользователя и отправляет чек на email (purchase notification message, send order receipt to email) и вызывает displayIndex() для отображения главного экрана и передачи управления IndexController;
- так же пользователь может просматривать все свои заказы CheckOrders(userID), тогда IndexController обрящается к OrdersController (DisplayOrders(userID)), тот в свою очередь через DBController получает все заказы текущего пользователя (getOrders(userID)) и отображает их на экране (display).

Остальные диаграммы представлены на рис. 2. и рис. 3.

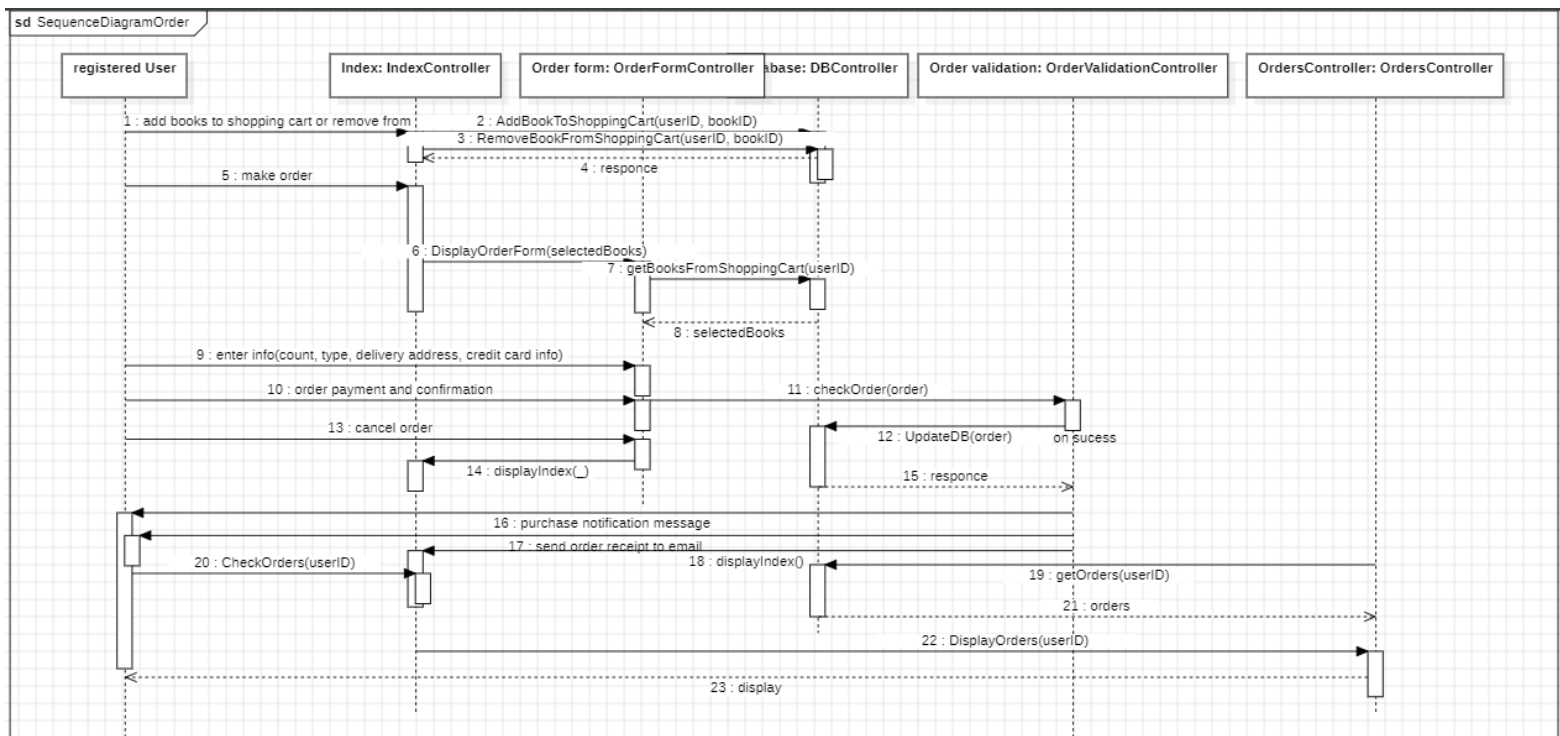


Рис. 1. Sequence Diagram Order

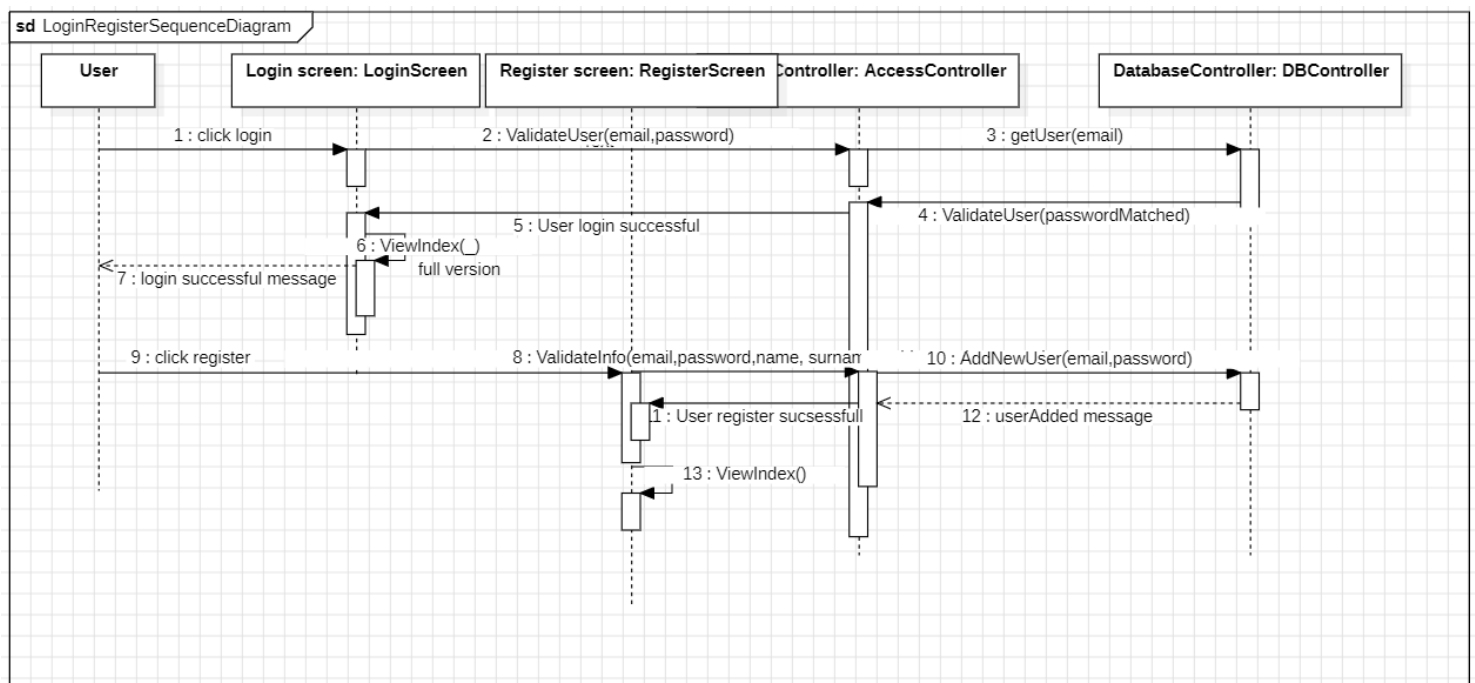


Рис. 2. Sequence Diagram Login and Register

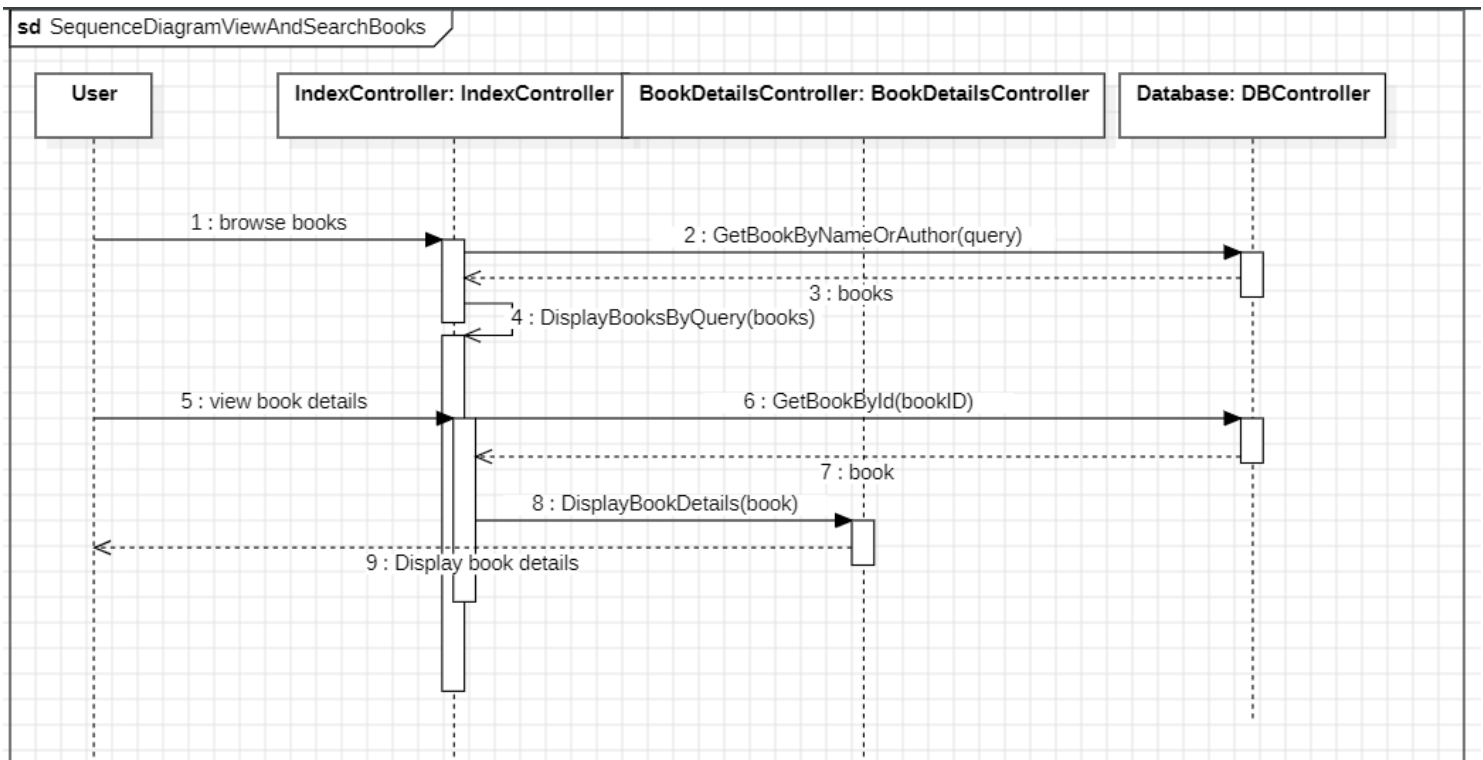


Рис. 3. Sequence Diagram View and Search books

Описание Class Diagram из рис. 4.

Класс AccessController предназначен для управления доступом пользователей к веб-магазину книг.

Краткое описание его функций:

1. `ValidateUser(email: String, password: String)`: Метод для проверки пользователя по электронной почте и паролю. Он принимает электронную почту и пароль в качестве аргументов и должен реализовывать логику аутентификации.
2. `ValidateUser(passwordMatched: boolean)`: Еще один метод для проверки подлинности пользователя, который принимает логическое значение, указывающее, совпадает ли предоставленный шифрованный пароль с сохраненным в системе паролем пользователя.
3. `ValidateInfo(email: String, password: String, name: String, surname: String, address: String, phoneNumber: String)`: Метод для проверки информации о пользователе. Он принимает электронную почту, пароль, имя, фамилию, адрес и номер телефона пользователя. Этот метод должен выполнять валидацию предоставленной информации о пользователе для регистрации.

Класс BookDetailsController предназначен для управления отображением деталей выбранной книги в веб-магазине.

Краткое описание его функций:

1. DisplayBookDetails(bookID: Int): Метод для отображения деталей книги по её идентификатору. Принимает идентификатор книги в качестве аргумента и должен реализовывать логику отображения информации о книге.
2. ViewIndex(): Метод для просмотра главной страницы сайта.

Класс DBController предназначен для управления доступом к базе данных в веб-магазине книг.

Краткое описание его функций:

1. getUser(email: String): Метод для получения информации о пользователе по его электронной почте. Принимает электронную почту пользователя в качестве аргумента.
2. AddNewUser(User: User): Метод для добавления нового пользователя в базу данных. Принимает объект пользователя в качестве аргумента.
3. GetBookById(bookID: Int): Метод для получения информации о книге по её идентификатору. Принимает идентификатор книги в качестве аргумента.
4. GetBookByNameOrAuthor(query: String): Метод для поиска книг по названию или автору. Принимает строку запроса в качестве аргумента.
5. AddBookToShoppingCart(userID: Int, bookID: Int): Метод для добавления книги в корзину покупок пользователя. Принимает идентификатор пользователя и идентификатор книги.
6. RemoveBookFromShoppingCart(userID: Int, bookID: Int): Метод для удаления книги из корзины покупок пользователя. Принимает идентификатор пользователя и идентификатор книги.
7. getBooksFromShoppingCart(userID: Int): Метод для получения списка книг из корзины покупок пользователя. Принимает идентификатор пользователя в качестве аргумента.
8. UpdateDB(order: Order): Метод для обновления базы данных после оформления заказа. Принимает объект заказа в качестве аргумента.
9. getOrders(userID: Int): Метод для получения списка заказов пользователя. Принимает идентификатор пользователя в качестве аргумента.
10. AddNewBook(book: Book): Метод для добавления новой книги в базу данных. Принимает объект книги в качестве аргумента.

Класс IndexController представляет контроллер для отображения данных на главной странице веб-магазина книг.

Краткое описание его функций:

1. DisplayBooksByQuery(books: List<Book>): Метод для отображения списка книг на основе запроса. Принимает список книг в качестве аргумента.
2. displayIndex(): Метод для отображения главной страницы.
3. CheckOrders(userID: Int): Метод для просмотра заказов пользователя. Принимает идентификатор пользователя в качестве аргумента.
4. ViewBooksDetails(bookID: Int): Метод для просмотра деталей о книге. Принимает идентификатор книги в качестве аргумента.

Класс LoginScreen представляет экран входа в систему в веб-магазине книг.

Краткое описание его функций:

1. ViewIndex(): Метод для перехода на главную страницу после успешного входа в систему.
2. display(): Метод для отображения экрана входа.

Класс RegisterScreen представляет экран регистрации в системе в веб-магазине книг.

Краткое описание его функций:

1. ViewIndex(): Метод для перехода на главную страницу после успешной регистрации в системе.
2. display(): Метод для отображения экрана регистрации.

Класс OrderFormController представляет контроллер формы заказа в веб-магазине книг.

Краткое описание его функций:

1. DisplayOrderForm(userID: Int): Метод для отображения формы заказа с учетом идентификатора пользователя. Принимает идентификатор пользователя в качестве аргумента.
2. ViewIndex(): Метод для перехода на главную страницу.

Класс OrdersController представляет контроллер для управления заказами в веб-магазине книг.

Краткое описание его функций:

1. DisplayOrders(userID: Int): Метод для отображения заказов определенного пользователя. Принимает идентификатор пользователя в качестве аргумента.
2. ViewIndex(): Метод для перехода на главную страницу.

Класс OrderValidationController представляет контроллер для проверки заказов и отправки уведомлений о покупке в веб-магазине книг.

Краткое описание его функций:

1. checkOrder(order: Order): Метод для проверки заказа. Принимает объект заказа в качестве аргумента.
2. sendOrderReceiptToEmail(receipt: Receipt): Метод для отправки чека о заказе на электронную почту. Принимает объект квитанции в качестве аргумента.
3. sendPurchaseNotificationMessage(result: PurchaseResult): Метод для отправки уведомления о состоянии покупки. Принимает объект результата покупки в качестве аргумента.

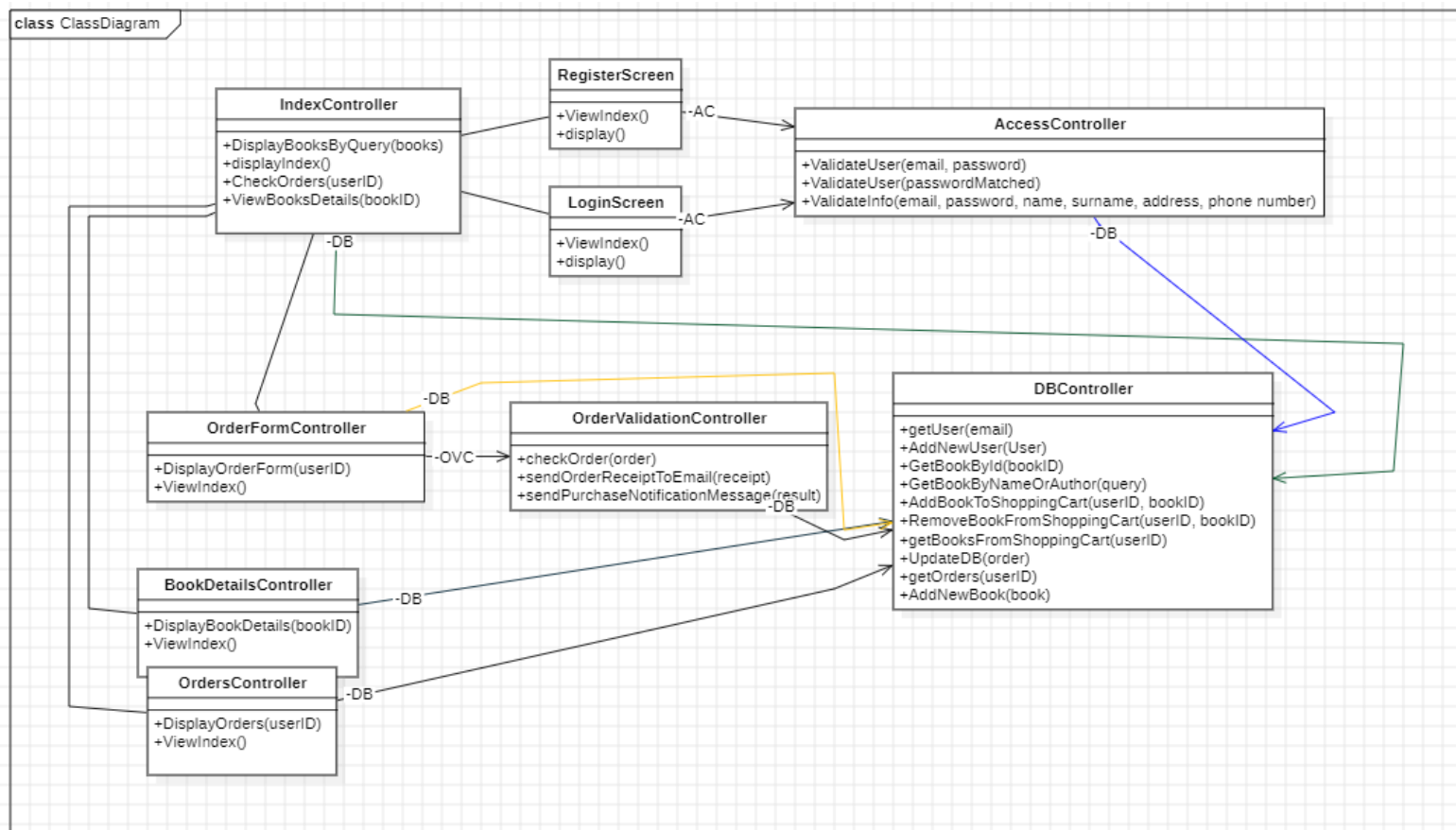


Рис. 4. Class Diagram