**HIBERNATE ASSIGNMENT 2**

**persistence.xml**

<persistence xmlns=*"http://xmlns.jcp.org/xml/ns/persistence"*

xmlns:xsi=*"http://www.w3.org/2001/XMLSchema-instance"*

xsi:schemaLocation=*"http://xmlns.jcp.org/xml/ns/persistence http://xmlns.jcp.org/xml/ns/persistence/persistence\_2\_1.xsd"*

version=*"2.1"*>

<persistence-unit name=*"PERSISTENCE"* transaction-type=*"RESOURCE\_LOCAL"*>

<provider>org.hibernate.jpa.HibernatePersistenceProvider</provider>

<class>com.books.app.model.Book</class>

<class>com.books.app.model.Subject</class>

<properties>

<property name=*"hibernate.dialect"* value=*"org.hibernate.dialect.MySQL5Dialect"* />

</properties>

</persistence-unit>

</persistence>

**spring.xml**

<beans xmlns=*"http://www.springframework.org/schema/beans"*

xmlns:xsi=*"http://www.w3.org/2001/XMLSchema-instance"* xmlns:context=*"http://www.springframework.org/schema/context"*

xmlns:tx=*"http://www.springframework.org/schema/tx"* xmlns:p=*"http://www.springframework.org/schema/p"*

xsi:schemaLocation=*"http://www.springframework.org/schema/beans*

*http://www.springframework.org/schema/beans/spring-beans-3.0.xsd*

*http://www.springframework.org/schema/context*

*http://www.springframework.org/schema/context/spring-context-3.0.xsd*

*http://www.springframework.org/schema/tx*

*http://www.springframework.org/schema/tx/spring-tx.xsd"*>

<tx:annotation-driven/>

<context:component-scan base-package=*"com.books.app.dao"* />

<context:component-scan base-package=*"com.books.app.service"* />

<bean id=*"dataSource"* class=*"org.apache.commons.dbcp.BasicDataSource"*

destroy-method=*"close"*>

<property name=*"driverClassName"* value=*"com.mysql.jdbc.Driver"* />

<property name=*"url"* value=*"jdbc:mysql://localhost:3306/bookshop"* />

<property name=*"username"* value=*"root"* />

<property name=*"password"* value=*"root"* />

</bean>

<bean id=*"entityManagerFactory"*

class=*"org.springframework.orm.jpa.LocalContainerEntityManagerFactoryBean"*>

<property name=*"persistenceUnitName"* value=*"PERSISTENCE"* />

<property name=*"dataSource"* ref=*"dataSource"* />

</bean>

<bean id=*"transactionManager"* class=*"org.springframework.orm.jpa.JpaTransactionManager"*>

<property name=*"entityManagerFactory"* ref=*"entityManagerFactory"* />

</bean>

</beans>

**pom.xml**

<project xmlns=*"http://maven.apache.org/POM/4.0.0"* xmlns:xsi=*"http://www.w3.org/2001/XMLSchema-instance"* xsi:schemaLocation=*"http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd"*>

<modelVersion>4.0.0</modelVersion>

<groupId>books.app</groupId>

<artifactId>BookShopAppSpringCore</artifactId>

<version>0.0.1-SNAPSHOT</version>

<name>BookShopAppSpringCore</name>

<properties>

<project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>

<spring.version>4.0.0.RELEASE</spring.version>

<hibernate.version>5.2.6.Final</hibernate.version>

</properties>

<dependencies> <!-- Spring -->

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-core</artifactId>

<version>${spring.version}</version>

</dependency>

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-context</artifactId>

<version>${spring.version}</version>

</dependency>

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-tx</artifactId>

<version>${spring.version}</version>

</dependency>

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-orm</artifactId>

<version>${spring.version}</version>

</dependency>

<dependency>

<groupId>org.hibernate</groupId>

<artifactId>hibernate-entitymanager</artifactId>

<version>${hibernate.version}</version>

</dependency>

<!-- dbcp -->

<dependency>

<groupId>commons-dbcp</groupId>

<artifactId>commons-dbcp</artifactId>

<version>1.2.2</version>

</dependency>

<!-- MySQL connector -->

<dependency>

<groupId>mysql</groupId>

<artifactId>mysql-connector-java</artifactId>

<version>5.1.44</version>

</dependency> <!-- Hibernate 5.2.6 Final -->

<dependency>

<groupId>org.hibernate</groupId>

<artifactId>hibernate-core</artifactId>

<version>5.2.6.Final</version>

</dependency>

</dependencies>

<build>

<sourceDirectory>src/main/java</sourceDirectory>

<plugins>

<plugin>

<artifactId>maven-compiler-plugin</artifactId>

<version>3.5.1</version>

<configuration>

<source>1.8</source>

<target>1.8</target>

</configuration>

</plugin>

</plugins>

</build>

</project>

**IBookAppDAO.java**

**package** com.books.app.dao;

**import** java.util.List;

**import** com.books.app.exception.BookAppException;

**import** com.books.app.model.Book;

**import** com.books.app.model.Subject;

**public** **interface** IBookAppDAO {

String add(Book book, String subtitle) **throws** BookAppException;

String addSubject(Subject subject) **throws** BookAppException;

**boolean** delete(**long** bookid) **throws** BookAppException;

**boolean** deleteSubject(String subtitle) **throws** BookAppException;

Book search(**long** bookid) **throws** BookAppException;

Subject searchSubject(String subtitle) **throws** BookAppException;

List<Book> view() **throws** BookAppException;

List<Subject> viewAll() **throws** BookAppException;

}

**BookAppDAOImpl.java**

**package** com.books.app.dao;

**import** java.util.List;

**import** javax.persistence.EntityManager;

**import** javax.persistence.PersistenceContext;

**import** org.springframework.stereotype.Repository;

**import** com.books.app.exception.BookAppException;

**import** com.books.app.model.Book;

**import** com.books.app.model.Subject;

@Repository

**public** **class** BookAppDAOImpl **implements** IBookAppDAO {

@PersistenceContext

**private** EntityManager em ;

@Override

**public** String add(Book book, String subtitle) **throws** BookAppException {

String bookTitle = **null**;

Subject subject = **null**;

**if** (book.getSubject() != **null**) {

List<Subject> subjects = viewAll();

**if** (subjects != **null** && !subjects.isEmpty()) {

**for** (Subject iterSubject : subjects) {

**if** (iterSubject != **null** && iterSubject.getSubtitle().equalsIgnoreCase(subtitle)) {

subject = iterSubject;

}

}

}

**if** (subject == **null**) {

subject = **new** Subject();

subject.setSubtitle(subtitle);

addSubject(subject);

}

book.setSubject(subject);

**try** {

em.persist(book);

bookTitle = book.getTitle();

} **catch** (Exception exp) {

**throw** **new** BookAppException("Book is not inserted");

}

}

**return** bookTitle;

}

@Override

**public** String addSubject(Subject subject) **throws** BookAppException {

String subjectTitle = **null**;

**if** (subject != **null**) {

**try** {

em.persist(subject);

subjectTitle = subject.getSubtitle();

} **catch** (Exception exp) {

**throw** **new** BookAppException("Subject is not inserted");

}

}

**return** subjectTitle;

}

@Override

**public** **boolean** delete(**long** bookid) **throws** BookAppException {

**try** {

Book book = em.find(Book.**class**, bookid);

em.remove(book);

**return** **true**;

} **catch** (Exception exp) {

**throw** **new** BookAppException("Book is not deleted");

}

}

@Override

**public** **boolean** deleteSubject(String subtitle) **throws** BookAppException {

**try** {

Subject subj = em.find(Subject.**class**, subtitle);

em.remove(subj);

**return** **true**;

} **catch** (Exception exp) {

**throw** **new** BookAppException("Subject is not deleted");

}

}

@Override

**public** Book search(**long** bookid) **throws** BookAppException {

Book book = **null**;

**try** {

book = em.find(Book.**class**, bookid);

} **catch** (Exception exp) {

**throw** **new** BookAppException("Book not found");

}

**return** book;

}

@Override

**public** Subject searchSubject(String subtitle) **throws** BookAppException {

Subject subj = **null**;

**try** {

subj = em.find(Subject.**class**, subtitle);

} **catch** (Exception exp) {

**throw** **new** BookAppException("Subject not found");

}

**return** subj;

}

@Override

**public** List<Subject> viewAll() **throws** BookAppException {

List<Subject> subjList = **null**;

**try** {

subjList = em.createQuery(IQueryMapper.***GET\_ALL\_SUBJECTS\_QRY***).getResultList();

} **catch** (Exception exp) {

**throw** **new** BookAppException("Subjects are not retrieved");

}

**return** subjList;

}

@Override

**public** List<Book> view() **throws** BookAppException {

List<Book> bookList = **null**;

**try** {

bookList = em.createQuery(IQueryMapper.***GET\_ALL\_BOOKS\_QRY***).getResultList();

} **catch** (Exception exp) {

**throw** **new** BookAppException("Books are not retrieved");

}

**return** bookList;

}

}

**IBookAppService.java**

**package** com.books.app.service;

**import** java.util.List;

**import** com.books.app.exception.BookAppException;

**import** com.books.app.model.Book;

**import** com.books.app.model.Subject;

**public** **interface** IBookAppService {

String add(Book book, String subtitle) **throws** BookAppException;

String addSubject(Subject subject) **throws** BookAppException;

**boolean** delete(**long** bookid) **throws** BookAppException;

**boolean** deleteSubject(String subtitle) **throws** BookAppException;

Book search(**long** bookid) **throws** BookAppException;

Subject searchSubject(String subtitle) **throws** BookAppException;

List<Book> view() **throws** BookAppException;

}

**BookAppServiceImpl.java**

**package** com.books.app.service;

**import** java.util.List;

**import** org.springframework.beans.factory.annotation.Autowired;

**import** org.springframework.stereotype.Service;

**import** org.springframework.transaction.annotation.Transactional;

**import** com.books.app.dao.IBookAppDAO;

**import** com.books.app.exception.BookAppException;

**import** com.books.app.model.Book;

**import** com.books.app.model.Subject;

@Service

**public** **class** BookAppServiceImpl **implements** IBookAppService {

@Autowired

**private** IBookAppDAO bookAppDao;

@Override

@Transactional

**public** String add(Book book, String subtitle) **throws** BookAppException {

**return** bookAppDao.add(book, subtitle);

}

@Override

@Transactional

**public** String addSubject(Subject subject) **throws** BookAppException {

**return** bookAppDao.addSubject(subject);

}

@Override

@Transactional

**public** **boolean** delete(**long** bookid) **throws** BookAppException {

**return** bookAppDao.delete(bookid);

}

@Override

@Transactional

**public** **boolean** deleteSubject(String subtitle) **throws** BookAppException {

**return** bookAppDao.deleteSubject(subtitle);

}

@Override

**public** Book search(**long** bookid) **throws** BookAppException {

**return** bookAppDao.search(bookid);

}

@Override

**public** Subject searchSubject(String subtitle) **throws** BookAppException {

**return** bookAppDao.searchSubject(subtitle);

}

@Override

**public** List<Book> view() **throws** BookAppException{

**return** bookAppDao.view();

}

}

**BooksMenuUI.java**

**package** com.books.app.ui;

**import** java.time.DateTimeException;

**import** java.time.LocalDate;

**import** java.time.format.DateTimeFormatter;

**import** java.util.Comparator;

**import** java.util.List;

**import** java.util.Scanner;

**import** java.util.Set;

**import** java.util.stream.Collectors;

**import** org.springframework.context.ApplicationContext;

**import** org.springframework.context.support.ClassPathXmlApplicationContext;

**import** com.books.app.exception.BookAppException;

**import** com.books.app.model.Book;

**import** com.books.app.model.BooksAppMenu;

**import** com.books.app.model.Subject;

**import** com.books.app.service.IBookAppService;

**import** com.books.app.util.JPAUtil;

**public** **class** BooksMenuUI {

**static** IBookAppService *bookAppService*;

**static** Scanner *scan*;

**static** DateTimeFormatter *dtFormater*;

**public** **static** **void** main(String[] args) {

**try** {

ApplicationContext ctx = **new** ClassPathXmlApplicationContext("spring.xml");

*bookAppService* = (IBookAppService)ctx.getBean(IBookAppService.**class**);

} **catch** (Exception exp) {

System.***err***.println(exp.getMessage());

System.*exit*(0);

}

*scan* = **new** Scanner(System.***in***);

*dtFormater* = DateTimeFormatter.*ofPattern*("dd-MM-yyyy");

BooksAppMenu menu = **null**;

**while** (menu != BooksAppMenu.***QUIT***) {

System.***out***.println("===========================");

System.***out***.println("Books App Menu Options");

System.***out***.println("===========================");

**for** (BooksAppMenu m : BooksAppMenu.*values*()) {

System.***out***.println(m.ordinal() + "\t" + m.name());

}

System.***out***.print("Select the action: ");

**int** action = -1;

**if** (*scan*.hasNextInt())

action = *scan*.nextInt();

**else** {

*scan*.next();

System.***out***.println("Please choose an action displayed");

**continue**;

}

**if** (action < 0 || action >= BooksAppMenu.*values*().length) {

System.***out***.println("Invalid Option");

} **else** {

menu = BooksAppMenu.*values*()[action];

**switch** (menu) {

**case** ***VIEW***:

*doView*();

**break**;

**case** ***ADD\_BOOK***:

*doAddBook*();

**break**;

**case** ***ADD\_SUBJECT***:

*doAddSubject*();

**break**;

**case** ***DELETE\_BOOK***:

*doDeleteBook*();

**break**;

**case** ***DELETE\_SUBJECT***:

*doDeleteSubject*();

**break**;

**case** ***SEARCH\_BOOK***:

*doSearchBook*();

**break**;

**case** ***SEARCH\_SUBJECT***:

*doSearchSubject*();

**break**;

**case** ***SORTBY\_SUBJECT\_TITLE***:

*doSortBySubjectTitle*();

**break**;

**case** ***SORTBY\_BOOK\_TITLE***:

*doSortByBookTitle*();

**break**;

**case** ***SORTBY\_BOOK\_PUBLISHED\_DATE***:

*doSortByBookPublishedDate*();

**break**;

**case** ***QUIT***:

System.***out***.println("Thanks for visiting!!");

**break**;

}

}

}

JPAUtil.shutdown();

*scan*.close();

}

**private** **static** **void** doSortByBookPublishedDate() {

**try** {

List<Book> list = *bookAppService*.view();

**if** (list != **null** && !list.isEmpty()) {

System.***out***.format("%-10s | %10s | %10s | %10s | %10s\n","Subject","BookTitle","Price","Volume","PublishedDate");

System.***out***.println("------------------------------------------------------------------");

List<Book> sList = list.stream().sorted(Comparator.*comparing*(Book::getPublistDt)).collect(Collectors.*toList*());

sList.forEach(e -> System.***out***.format("%-10s | %10s | %10.2f | %10d | %10s\n",e.getSubject().getSubtitle(),e.getTitle(),e.getPrice(),e.getVolume(),e.getPublistDt()));

} **else** {

System.***out***.println("No books to view !!");

}

} **catch** (BookAppException e) {

e.printStackTrace();

}

}

**private** **static** **void** doSortByBookTitle() {

**try** {

List<Book> list = *bookAppService*.view();

**if** (list != **null** && !list.isEmpty()) {

System.***out***.format("%-10s | %10s | %10s | %10s | %10s\n","Subject","BookTitle","Price","Volume","PublishedDate");

System.***out***.println("------------------------------------------------------------------");

List<Book> sList = list.stream().sorted(Comparator.*comparing*(Book::getTitle)).collect(Collectors.*toList*());

sList.forEach(e -> System.***out***.format("%-10s | %10s | %10.2f | %10d | %10s\n",e.getSubject().getSubtitle(),e.getTitle(),e.getPrice(),e.getVolume(),e.getPublistDt()));

} **else** {

System.***out***.println("No books to view !!");

}

} **catch** (BookAppException e) {

e.printStackTrace();

}

}

**private** **static** **void** doSortBySubjectTitle() {

**try** {

List<Book> list = *bookAppService*.view();

**if** (list != **null** && !list.isEmpty()) {

System.***out***.format("%-10s | %10s | %10s | %10s | %10s\n","Subject","BookTitle","Price","Volume","PublishedDate");

System.***out***.println("------------------------------------------------------------------");

List<Book> sList = list.stream().collect(Collectors.*toList*());

sList.forEach(e -> System.***out***.format("%-10s | %10s | %10.2f | %10d | %10s\n",e.getSubject().getSubtitle(),e.getTitle(),e.getPrice(),e.getVolume(),e.getPublistDt()));

} **else** {

System.***out***.println("No books to view !!");

}

} **catch** (BookAppException e) {

e.printStackTrace();

}

}

**private** **static** **void** doSearchSubject() {

**try** {

System.***out***.print("Subject Title: ");

String subtitle = *scan*.next();

Subject subj = *bookAppService*.searchSubject(subtitle);

**if** (subj != **null**) {

System.***out***.println("Subject Found : " + subj.getSubtitle());

System.***out***.println("Available books are : ");

Set<Book> books = subj.getReferences();

**if** (**null** != books) {

**for** (Book book : books) {

**if** (book != **null**) {

System.***out***.println("-------------- " + book.getTitle());

}

}

}

}

} **catch** (BookAppException exp) {

System.***out***.println(exp.getMessage());

}

}

**private** **static** **void** doSearchBook() {

**try** {

System.***out***.print("Book Id: ");

**long** bookid = *scan*.nextLong();

Book book = *bookAppService*.search(bookid);

**if** (book != **null**) {

System.***out***.println("Book Found : " + book.getTitle());

System.***out***.println("Title ----- " + book.getTitle());

System.***out***.println("Price ----- " + book.getPrice());

System.***out***.println("Volume ----- " + book.getVolume());

System.***out***.println("Published Date ----- " + book.getPublistDt());

}

} **catch** (BookAppException exp) {

System.***out***.println(exp.getMessage());

}

}

**private** **static** **void** doDeleteSubject() {

**try** {

**boolean** delsubj = **false**;

System.***out***.print("Subject Title: ");

String subtitle = *scan*.next();

delsubj = *bookAppService*.deleteSubject(subtitle);

**if** (delsubj) {

System.***out***.println("Subject deleted successfully");

} **else** {

System.***out***.println("Subject cannot be deleted");

}

} **catch** (BookAppException exp) {

System.***out***.println(exp.getMessage());

}

}

**private** **static** **void** doDeleteBook() {

**try** {

**boolean** delbook = **false**;

System.***out***.print("Book Id: ");

**long** bookid = *scan*.nextLong();

delbook = *bookAppService*.delete(bookid);

**if** (delbook) {

System.***out***.println("Book deleted successfully");

} **else** {

System.***out***.println("Book cannot be deleted");

}

} **catch** (BookAppException exp) {

System.***out***.println(exp.getMessage());

}

}

**private** **static** **void** doAddSubject() {

String newSubject = **null**;

**try** {

Subject sub = **new** Subject();

System.***out***.print("Enter Subject ID: ");

sub.setSubjectId(Long.*parseLong*(*scan*.next()));

System.***out***.print("Enter Subject Title: ");

sub.setSubtitle(*scan*.next());

System.***out***.print("Enter Duration in hrs: ");

sub.setDurationInHours(Integer.*parseInt*(*scan*.next()));

**if** (*bookAppService*.searchSubject(sub.getSubtitle()) == **null**) {

newSubject = *bookAppService*.addSubject(sub);

}

System.***out***.println("Subject " + newSubject + " is added");

} **catch** (BookAppException exp) {

System.***out***.println(exp.getMessage());

}

}

**private** **static** **void** doView() {

**try** {

List<Book> list = *bookAppService*.view();

**if** (list != **null** && !list.isEmpty()) {

System.***out***.format("%-10s | %10s | %10s | %10s | %10s\n", "Subject", "BookTitle", "Price", "Volume",

"PublishedDate");

System.***out***.println("------------------------------------------------------------------");

List<Book> sList = list.stream().collect(Collectors.*toList*());

sList.forEach(e -> System.***out***.format("%-10s | %10s | %10.2f | %10d | %10s\n",

e.getSubject().getSubtitle(), e.getTitle(), e.getPrice(), e.getVolume(), e.getPublistDt()));

} **else** {

System.***out***.println("No books to view !!");

}

} **catch** (BookAppException e) {

e.printStackTrace();

}

}

**private** **static** **void** doAddBook() {

**try** {

Subject sub = **new** Subject();

Book book = **new** Book();

System.***out***.print("Enter Book Id: ");

book.setBookId(Long.*parseLong*(*scan*.next()));

System.***out***.print("Enter Book Title: ");

book.setTitle(*scan*.next());

System.***out***.print("Enter Price: ");

book.setPrice(Double.*parseDouble*(*scan*.next()));

System.***out***.print("Enter Volume: ");

book.setVolume(*scan*.nextInt());

System.***out***.print("Enter PublishDate(dd-MM-yyyy): ");

String pubDtStr = *scan*.next();

**try** {

book.setPublistDt(LocalDate.*parse*(pubDtStr, *dtFormater*));

} **catch** (DateTimeException exp) {

**throw** **new** BookAppException("Date must be in the format day(dd)-month(MM)-year(yyyy)");

}

System.***out***.print("Enter Subject Title: ");

sub.setSubtitle(*scan*.next());

book.setSubject(sub);

String bookTitle = *bookAppService*.add(book, sub.getSubtitle());

System.***out***.println("Book " + bookTitle + " is added");

} **catch** (BookAppException exp) {

System.***out***.println(exp.getMessage());

}

}

}

**IQueryMapper.java**

**package** com.books.app.dao;

**public** **interface** IQueryMapper {

**public** **static** **final** String ***GET\_ALL\_BOOKS\_QRY*** =

"FROM Book";

**public** **static** **final** String ***GET\_ALL\_SUBJECTS\_QRY*** =

"FROM Subject";

}

**BookAppException.java**

**package** com.books.app.exception;

@SuppressWarnings("serial")

**public** **class** BookAppException **extends** Exception {

**public** BookAppException(String errMsg) {

**super**(errMsg);

}

}

**BooksAppMenu.java**

**package** com.books.app.model;

**public** **enum** BooksAppMenu {

***VIEW***,***ADD\_BOOK***,***ADD\_SUBJECT***,***DELETE\_BOOK***,***DELETE\_SUBJECT***,***SEARCH\_BOOK***,***SEARCH\_SUBJECT***,***SORTBY\_SUBJECT\_TITLE***,***SORTBY\_BOOK\_TITLE***,***SORTBY\_BOOK\_PUBLISHED\_DATE***,***QUIT***;

}

**JPAUtil.java**

**package** com.books.app.util;

**import** javax.persistence.EntityManagerFactory;

**import** javax.persistence.Persistence;

**public** **class** JPAUtil {

**private** **static** **final** String ***PERSISTENCE\_UNIT\_NAME*** = "PERSISTENCE";

**private** **static** EntityManagerFactory *factory*;

**public** **static** EntityManagerFactory getEntityManagerFactory() {

**if** (*factory* == **null**) {

*factory* = Persistence.*createEntityManagerFactory*(***PERSISTENCE\_UNIT\_NAME***);

}

**return** *factory*;

}

**public** **static** **void** shutdown() {

**if** (*factory* != **null**) {

*factory*.close();

}

}

}

**Subject.java**

**package** com.books.app.model;

**import** java.util.Set;

**import** javax.persistence.Column;

**import** javax.persistence.Entity;

**import** javax.persistence.FetchType;

**import** javax.persistence.Id;

**import** javax.persistence.OneToMany;

**import** javax.persistence.Table;

@Entity

@Table(name="subjects")

**public** **class** Subject {

@Id

@Column(name="subtitle")

**private** String subtitle;

@Column(name="subjectid")

**private** **long** subjectId;

@Column(name="durationInHours")

**private** **int** durationInHours;

@OneToMany(fetch = FetchType.***EAGER***,mappedBy="subject",orphanRemoval=**true**)

**private** Set<Book> references;

**public** **long** getSubjectId() {

**return** subjectId;

}

**public** **void** setSubjectId(**long** subjectId) {

**this**.subjectId = subjectId;

}

**public** String getSubtitle() {

**return** subtitle;

}

**public** **void** setSubtitle(String subtitle) {

**this**.subtitle = subtitle;

}

**public** **int** getDurationInHours() {

**return** durationInHours;

}

**public** **void** setDurationInHours(**int** durationInHours) {

**this**.durationInHours = durationInHours;

}

**public** Set<Book> getReferences() {

**return** references;

}

**public** **void** setReferences(Set<Book> references) {

**this**.references = references;

}

}

**Book.java**

**package** com.books.app.model;

**import** java.time.LocalDate;

**import** javax.persistence.Column;

**import** javax.persistence.Entity;

**import** javax.persistence.FetchType;

**import** javax.persistence.Id;

**import** javax.persistence.JoinColumn;

**import** javax.persistence.ManyToOne;

**import** javax.persistence.Table;

@Entity

@Table(name="books")

**public** **class** Book {

@Id

@Column(name="bookid")

**private** **long** bookId;

@Column(name="title")

**private** String title;

@Column(name="price")

**private** **double** price;

@Column(name="vol")

**private** **int** volume;

@Column(name="pdate")

**private** LocalDate publistDt;

@ManyToOne(fetch=FetchType.***EAGER***)

@JoinColumn(name="stitle" , nullable=**false**)

**private** Subject subject;

**public** Subject getSubject() {

**return** subject;

}

**public** **void** setSubject(Subject subject) {

**this**.subject = subject;

}

**public** **long** getBookId() {

**return** bookId;

}

**public** **void** setBookId(**long** bookId) {

**this**.bookId = bookId;

}

**public** String getTitle() {

**return** title;

}

**public** **void** setTitle(String title) {

**this**.title = title;

}

**public** **double** getPrice() {

**return** price;

}

**public** **void** setPrice(**double** price) {

**this**.price = price;

}

**public** **int** getVolume() {

**return** volume;

}

**public** **void** setVolume(**int** volume) {

**this**.volume = volume;

}

**public** LocalDate getPublistDt() {

**return** publistDt;

}

**public** **void** setPublistDt(LocalDate publistDt) {

**this**.publistDt = publistDt;

}

}

**RESULTS :**

===========================

Books App Menu Options

===========================

0 VIEW

1 ADD\_BOOK

2 ADD\_SUBJECT

3 DELETE\_BOOK

4 DELETE\_SUBJECT

5 SEARCH\_BOOK

6 SEARCH\_SUBJECT

7 SORTBY\_SUBJECT\_TITLE

8 SORTBY\_BOOK\_TITLE

9 SORTBY\_BOOK\_PUBLISHED\_DATE

10 QUIT

**Select the action: 7**

**Subject | BookTitle | Price | Volume | PublishedDate**

**------------------------------------------------------------------**

**History | India | 70.00 | 1 | 2018-09-02**

**History | Rights | 50.00 | 2 | 2018-09-01**

**Science | Species | 40.00 | 2 | 2018-07-01**

**Science | Genes | 50.00 | 1 | 2018-01-01**

===========================

Books App Menu Options

===========================

0 VIEW

1 ADD\_BOOK

2 ADD\_SUBJECT

3 DELETE\_BOOK

4 DELETE\_SUBJECT

5 SEARCH\_BOOK

6 SEARCH\_SUBJECT

7 SORTBY\_SUBJECT\_TITLE

8 SORTBY\_BOOK\_TITLE

9 SORTBY\_BOOK\_PUBLISHED\_DATE

10 QUIT

**Select the action: 8**

**Subject | BookTitle | Price | Volume | PublishedDate**

**------------------------------------------------------------------**

**Science | Genes | 50.00 | 1 | 2018-01-01**

**History | India | 70.00 | 1 | 2018-09-02**

**History | Rights | 50.00 | 2 | 2018-09-01**

**Science | Species | 40.00 | 2 | 2018-07-01**

===========================

Books App Menu Options

===========================

0 VIEW

1 ADD\_BOOK

2 ADD\_SUBJECT

3 DELETE\_BOOK

4 DELETE\_SUBJECT

5 SEARCH\_BOOK

6 SEARCH\_SUBJECT

7 SORTBY\_SUBJECT\_TITLE

8 SORTBY\_BOOK\_TITLE

9 SORTBY\_BOOK\_PUBLISHED\_DATE

10 QUIT

**Select the action: 9**

**Subject | BookTitle | Price | Volume | PublishedDate**

**------------------------------------------------------------------**

**Science | Genes | 50.00 | 1 | 2018-01-01**

**Science | Species | 40.00 | 2 | 2018-07-01**

**History | Rights | 50.00 | 2 | 2018-09-01**

**History | India | 70.00 | 1 | 2018-09-02**

===========================

Books App Menu Options

===========================

0 VIEW

1 ADD\_BOOK

2 ADD\_SUBJECT

3 DELETE\_BOOK

4 DELETE\_SUBJECT

5 SEARCH\_BOOK

6 SEARCH\_SUBJECT

7 SORTBY\_SUBJECT\_TITLE

8 SORTBY\_BOOK\_TITLE

9 SORTBY\_BOOK\_PUBLISHED\_DATE

10 QUIT

**Select the action: 5**

**Book Id: 1008**

**Book Found : Genes**

**Title ----- Genes**

**Price ----- 50.0**

**Volume ----- 1**

**Published Date ----- 2018-01-01**

===========================

Books App Menu Options

===========================

0 VIEW

1 ADD\_BOOK

2 ADD\_SUBJECT

3 DELETE\_BOOK

4 DELETE\_SUBJECT

5 SEARCH\_BOOK

6 SEARCH\_SUBJECT

7 SORTBY\_SUBJECT\_TITLE

8 SORTBY\_BOOK\_TITLE

9 SORTBY\_BOOK\_PUBLISHED\_DATE

10 QUIT

**Select the action: 6**

**Subject Title: Science**

**Subject Found : Science**

**Available books are :**

**-------------- Species**

**-------------- Genes**

===========================

Books App Menu Options

===========================

0 VIEW

1 ADD\_BOOK

2 ADD\_SUBJECT

3 DELETE\_BOOK

4 DELETE\_SUBJECT

5 SEARCH\_BOOK

6 SEARCH\_SUBJECT

7 SORTBY\_SUBJECT\_TITLE

8 SORTBY\_BOOK\_TITLE

9 SORTBY\_BOOK\_PUBLISHED\_DATE

10 QUIT

**Select the action: 2**

**Enter Subject ID: 6004**

**Enter Subject Title: Sports**

**Enter Duration in hrs: 14**

**Subject Sports is added**

===========================

Books App Menu Options

===========================

0 VIEW

1 ADD\_BOOK

2 ADD\_SUBJECT

3 DELETE\_BOOK

4 DELETE\_SUBJECT

5 SEARCH\_BOOK

6 SEARCH\_SUBJECT

7 SORTBY\_SUBJECT\_TITLE

8 SORTBY\_BOOK\_TITLE

9 SORTBY\_BOOK\_PUBLISHED\_DATE

10 QUIT

**Select the action: 1**

**Enter Book Id: 1009**

**Enter Book Title: Cricket**

**Enter Price: 50.00**

**Enter Volume: 2**

**Enter PublishDate(dd-MM-yyyy): 01-03-2018**

**Enter Subject Title: Sports**

**Book Cricket is added**

===========================

Books App Menu Options

===========================

0 VIEW

1 ADD\_BOOK

2 ADD\_SUBJECT

3 DELETE\_BOOK

4 DELETE\_SUBJECT

5 SEARCH\_BOOK

6 SEARCH\_SUBJECT

7 SORTBY\_SUBJECT\_TITLE

8 SORTBY\_BOOK\_TITLE

9 SORTBY\_BOOK\_PUBLISHED\_DATE

10 QUIT

**Select the action: 0**

**Subject | BookTitle | Price | Volume | PublishedDate**

**------------------------------------------------------------------**

**History | India | 70.00 | 1 | 2018-09-02**

**History | Rights | 50.00 | 2 | 2018-09-01**

**Science | Species | 40.00 | 2 | 2018-07-01**

**Science | Genes | 50.00 | 1 | 2018-01-01**

**Sports | Cricket | 50.00 | 2 | 2018-03-01**

===========================

Books App Menu Options

===========================

0 VIEW

1 ADD\_BOOK

2 ADD\_SUBJECT

3 DELETE\_BOOK

4 DELETE\_SUBJECT

5 SEARCH\_BOOK

6 SEARCH\_SUBJECT

7 SORTBY\_SUBJECT\_TITLE

8 SORTBY\_BOOK\_TITLE

9 SORTBY\_BOOK\_PUBLISHED\_DATE

10 QUIT

**Select the action: 3**

**Book Id: 1009**

**Book deleted successfully**

===========================

Books App Menu Options

===========================

0 VIEW

1 ADD\_BOOK

2 ADD\_SUBJECT

3 DELETE\_BOOK

4 DELETE\_SUBJECT

5 SEARCH\_BOOK

6 SEARCH\_SUBJECT

7 SORTBY\_SUBJECT\_TITLE

8 SORTBY\_BOOK\_TITLE

9 SORTBY\_BOOK\_PUBLISHED\_DATE

10 QUIT

**Select the action: 4**

**Subject Title: Sports**

**Subject deleted successfully**

===========================

Books App Menu Options

===========================

0 VIEW

1 ADD\_BOOK

2 ADD\_SUBJECT

3 DELETE\_BOOK

4 DELETE\_SUBJECT

5 SEARCH\_BOOK

6 SEARCH\_SUBJECT

7 SORTBY\_SUBJECT\_TITLE

8 SORTBY\_BOOK\_TITLE

9 SORTBY\_BOOK\_PUBLISHED\_DATE

10 QUIT

**Select the action: 10**

**Thanks for visiting!!**