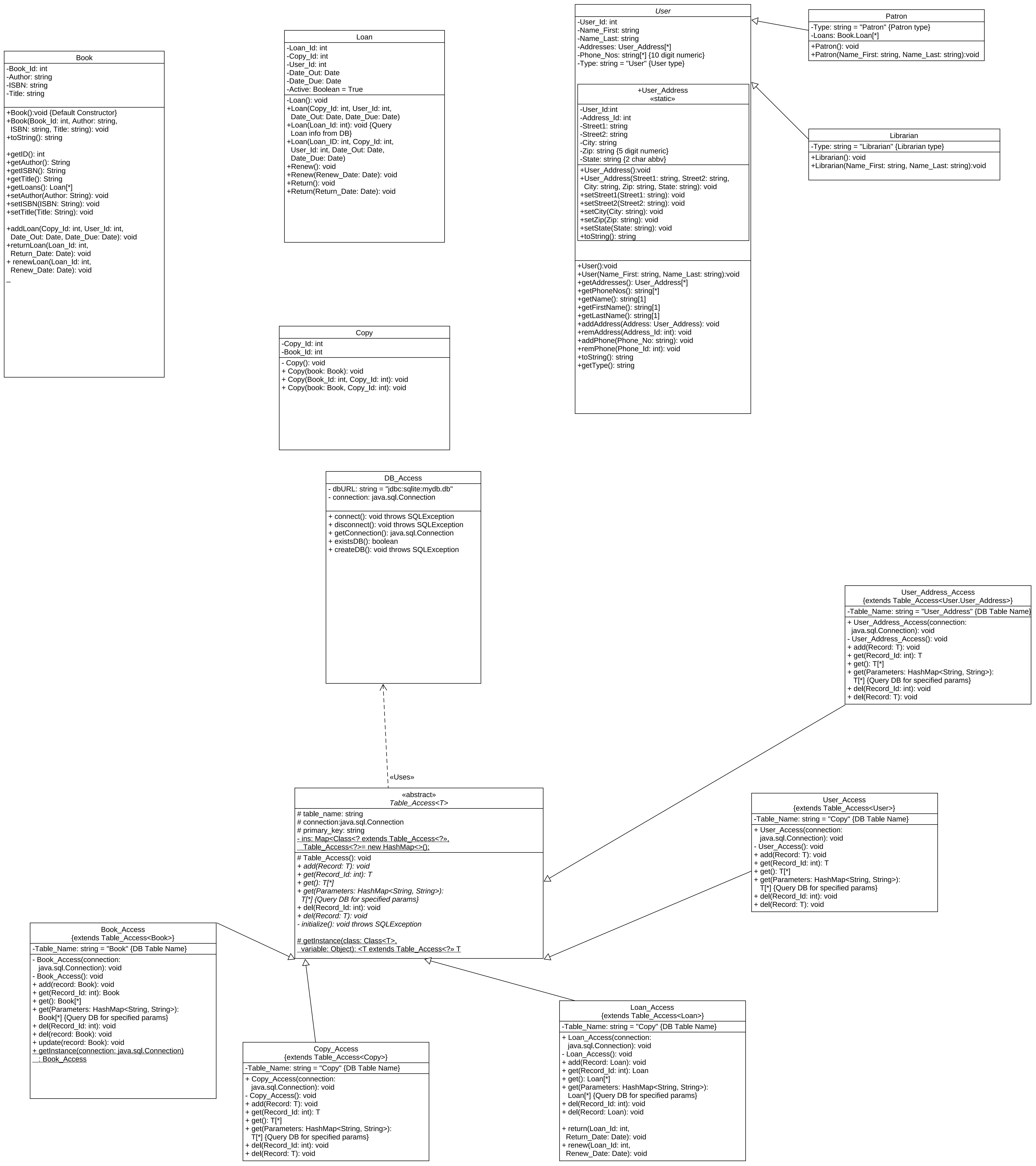
Book
-Book_Id: int
-Author: string
-ISBN: string
-Title: string
+Book():void {Default Constructor}
+Book(Book_Id: int, Author: string, ISBN: string, Title: string): void
+toString(): string
+getID(): int
+getAuthor(): String
+getISBN(): String
+getTitle(): String
+getLoans(): Loan[*]
+setAuthor(Author: String): void
+setISBN(ISBN: String): void
+setTitle(Title: String): void
+addLoan(Copy_Id: int, User_Id: int, Date_Out: Date, Date_Due: Date): void
+returnLoan(Loan_Id: int, Return_Date: Date): void
+ renewLoan(Loan_Id: int, Renew_Date: Date): void
_
Loan
-Loan_Id: int
-Copy_Id: int
-User_Id: int
-Date_Out: Date
-Date_Due: Date
-Active: Boolean = True
-Loan(): void
+Loan(Copy_Id: int, User_Id: int, Date_Out: Date, Date_Due: Date)
+Loan(Loan_Id: int): void {Query Loan info from DB}
+Loan(Loan_ID: int, Copy_Id: int, User_Id: int, Date_Out: Date, Date_Due: Date)
+Renew(): void
+Renew(Renew_Date: Date): void
+Return(): void
+Return(Return_Date: Date): void
User
-User_Id: int
-Name_First: string
-Name_Last: string
-Addresses: User_Address[*]
-Phone_Nos: string[*] {10 digit numeric}
-Type: string = "User" {User type}
+User_Address
«static»
-User_Id:int
-Address_Id: int
-Street1: string
-Street2: string
-City: string
-Zip: string {5 digit numeric}
-State: string {2 char abbv}
+User_Address():void
+User_Address(Street1: string, Street2: string, City: string, Zip: string, State: string): void
+setStreet1(Street1: string): void
+setStreet2(Street2: string): void
+setCity(City: string): void
+setZip(Zip: string): void
+setState(State: string): void
+toString(): string
+User():void
+User(Name_First: string, Name_Last: string):void
+getAddresses(): User_Address[*]
+getPhoneNos(): string[*]
+getName(): string[1]
+getFirstName(): string[1]
+getLastName(): string[1]
+addAddress(Address: User_Address): void
+remAddress(Address_Id: int): void
+addPhone(Phone_No: string): void
+remPhone(Phone_Id: int): void
+toString(): string
+getType(): string
Patron
-Type: string = "Patron" {Patron type}
-Loans: Book.Loan[*]
+Patron(): void
+Patron(Name_First: string, Name_Last: string):void
Librarian
-Type: string = "Librarian" {Librarian type}
+Librarian(): void
+Librarian(Name_First: string, Name_Last: string):void
Copy
-Copy_Id: int
-Book_Id: int
- Copy(): void
+ Copy(book: Book): void
+ Copy(Book_Id: int, Copy_Id: int): void
+ Copy(book: Book, Copy_Id: int): void
DB_Access
- dbURL: string = "jdbc:sqlite:mydb.db"
- connection: java.sql.Connection
+ connect(): void throws SQLException
+ disconnect(): void throws SQLException
+ getConnection(): java.sql.Connection
+ existsDB(): boolean
+ createDB(): void throws SQLException
«Uses»
User_Address_Access
{extends Table_Access<User.User_Address>}
-Table_Name: string = "User_Address" {DB Table Name}
+ User_Address_Access(connection: java.sql.Connection): void
- User_Address_Access(): void
+ add(Record: T): void
+ get(Record_Id: int): T
+ get(): T[*]
+ get(Parameters: HashMap<String, String>): T[*] {Query DB for specified params}
+ del(Record_Id: int): void
+ del(Record: T): void
«abstract»
Table_Access<T>
# table_name: string
# connection:java.sql.Connection
# primary_key: string
- ins: Map<Class<? extends Table_Access<?>, Table_Access<?>>= new HashMap<>();
# Table_Access(): void
+ add(Record: T): void
+ get(Record_Id: int): T
+ get(): T[*]
+ get(Parameters: HashMap<String, String>): T[*] {Query DB for specified params}
+ del(Record_Id: int): void
+ del(Record: T): void
- initialize(): void throws SQLException
# getInstance(class: Class<T>, variable: Object): <T extends Table_Access<?» T
User_Access
{extends Table_Access<User>}
-Table_Name: string = "Copy" {DB Table Name}
+ User_Access(connection: java.sql.Connection): void
- User_Access(): void
+ add(Record: T): void
+ get(Record_Id: int): T
+ get(): T[*]
+ get(Parameters: HashMap<String, String>): T[*] {Query DB for specified params}
+ del(Record_Id: int): void
+ del(Record: T): void
Book_Access
{extends Table_Access<Book>}
-Table_Name: string = "Book" {DB Table Name}
- Book_Access(connection: java.sql.Connection): void
- Book_Access(): void
+ add(record: Book): void
+ get(Record_Id: int): Book
+ get(): Book[*]
+ get(Parameters: HashMap<String, String>): Book[*] {Query DB for specified params}
+ del(Record_Id: int): void
+ del(record: Book): void
+ update(record: Book): void
+ getInstance(connection: java.sql.Connection) : Book_Access
Copy_Access
{extends Table_Access<Copy>}
-Table_Name: string = "Copy" {DB Table Name}
+ Copy_Access(connection: java.sql.Connection): void
- Copy_Access(): void
+ add(Record: T): void
+ get(Record_Id: int): T
+ get(): T[*]
+ get(Parameters: HashMap<String, String>): T[*] {Query DB for specified params}
+ del(Record_Id: int): void
+ del(Record: T): void
Loan_Access
{extends Table_Access<Loan>}
-Table_Name: string = "Copy" {DB Table Name}
+ Loan_Access(connection: java.sql.Connection): void
- Loan_Access(): void
+ add(Record: Loan): void
+ get(Record_Id: int): Loan
+ get(): Loan[*]
+ get(Parameters: HashMap<String, String>): Loan[*] {Query DB for specified params}
+ del(Record_Id: int): void
+ del(Record: Loan): void
+ return(Loan_Id: int, Return_Date: Date): void
+ renew(Loan_Id: int, Renew_Date: Date): void