

Model and schemas of our project

User

```
const mongoose = require("mongoose"),
    passportLocalMongoose = require("passport-local-mongoose");

const userSchema = new mongoose.Schema({
  firstName: {
    type: String,
    trim: true,
  },
  lastName: {
    type: String,
    trim: true,
  },
  username: {
    type: String,
    trim: true,
  },
  email: {
    type: String,
    trim: true,
  },
  password: String,
  joined: { type: Date, default: Date.now() },
  bookIssueInfo: [
    {
      book_info: {
        id: {
          type: mongoose.Schema.Types.ObjectId,
          ref: "Issue",
        },
      },
    },
  ],
  gender: String,
  address: String,
  userSchema.plugin(passportLocalMongoose);

module.exports = mongoose.model("User", userSchema);
```

borrow book

```
const mongoose = require('mongoose');
const Schema = mongoose.Schema;

const borrowSchema = new Schema({
  borrower: {
    type: new Schema({
      firstName: {
        type: String,
        required: true,
        minlength: 3,
        maxlength: 50,
        trim: true
      },
      lastName: {
        type: String,
        trim: true,
        minlength: 3,
        maxlength: 50,
        required: true
      },
      email: {
        type: String,
        trim: true,
        minlength: 10,
        maxlength: 100
      },
      phone: {
        type: String,
        minlength: 7,
        maxlength: 20,
        required: true
      }
    }),
    required: true
  },
  book: {
    type: new Schema({
      title: {
        type: String,
        required: true,
        trim: true,
        minlength: 5,
        maxlength: 50
      }
    })
  }
});
```

```

    },
    failedReturnFee: {
      type: Number,
      required: true,
      min: 0,
      max: 500
    }
  })),
  required: true
},
dateBorrowed: {
  type: Date,
  required: true,
  default: Date.now
},
dueDate: {
  type: Date
},
dateReturned: {
  type: Date
},
penaltyFee: {
  type: Number,
  min: 0
}
});
exports.Borrow = mongoose.model('Borrow', borrowSchema);

```

categories

```

const mongoose = require('mongoose');
const Schema = mongoose.Schema;

const categorySchema = new Schema({
  name: {
    type: String,
    required: true,
    trim: true,
    minlength: 5,
    maxlength: 50
  }
});

const mongoose = require('mongoose');
const Schema = mongoose.Schema;

```

```

const categorySchema = new Schema({
  name: {
    type: String,
    required: true,
    trim: true,
    minlength: 5,
    maxlength: 50
  }
});
exports.Category = mongoose.model('Category', categorySchema);

```

book

```

const mongoose = require("mongoose");

const bookSchema = new mongoose.Schema({
  title : String,
  ISBN : String,
  stock : Number,
  author : String,
  description : String,
  category : String,
});

module.exports = mongoose.model("Book", bookSchema);

```

issue

```

const mongoose = require("mongoose");

const issueSchema = new mongoose.Schema({
  book_info : {
    id : {
      type : mongoose.Schema.Types.ObjectId,
      ref : 'Book',
    },
    title : String,
    author : String,
    ISBN : String,
    category : String,
    stock : Number,
    issueDate : {type : Date, default : Date.now()},
    returnDate : {type : Date, default : Date.now() + 7*24*60*60*1000},
  }
});

```

```
    isRenewed : {type : Boolean, default : false},
  },

  user_id : {
    id : {
      type : mongoose.Schema.Types.ObjectId,
      ref : 'User',
    },

    username : String,
  },
});

module.exports = mongoose.model("Issue", issueSchema);
```