**-----Class AddCopyView-----**

/\*\*\* Submit a POST Request with book copy info to controller\* @param BookCopy fields of object BookCopy that users enter  
\* @throws IllegalArgumentException if the parameter is of wrong type  
\*/  
public void postBookCopy(BookCopy $copy)  
throws IllegalArgumentException {

send book copy info to controller via POST request

}

/\*\*  
\* Notifies success/fail to user  
\* @param message a string representation of the message to be notified to the user.  
\*/  
public void notifyUser(String message)  
throws IOException{

notify message to user;

}

/\*\*  
\* Render Add new copy view  
\*/  
public void renderView(){

render Add new copy html

}

**-----Class BookCopyController-----**

/\*\*

\* Show the form for creating a new resource.

\*

\* @param $book\_id

\* @return \Illuminate\Http\Response

\*/

public function create(int $book\_id)

{

$book = (new Book())->find($book\_id);

if (is\_null($book)) {

return abort(404, "Page not found!");

}

return view('copy.create')->with('book', $book);

}

/\*\*

\* Store a newly created resource in storage.

\*

\* @param $book\_id

\* @param \Illuminate\Http\Request $request

\* @return \Illuminate\Http\Response

\*/

public function store(int $book\_id, Request $request)

{

// validate the data

$this->validate($request, array(

'number\_of\_copies' => 'required|numeric',

'price' => 'required|numeric',

'type\_of\_copy' => 'in:borrowable,referenced'

));

$book = (new Book())->find($book\_id);

if (is\_null($book)) {

Session::flash('fail', 'The book\_id attribute is not valid!');

return redirect()->back();

}

for($i = 1; $i <= $request->input('number\_of\_copies'); $i++){

$copy = new BookCopy();

$copy->createNewCopyForBook($book, $request);

}

$book->updateNumberOfCopies();

Session::flash('success', 'The copies was successfully added!');

return redirect()->route('book.show', $book\_id);

}

**-----Class BookCopy-----**

**/\*\***

\* Create new copy for a book and insert it to the database

\*

\* @param Book $book the book we want to create copy for

\* @param Request $request

\*/

public function createNewCopyForBook(Book $book, Request $request)

{

$copyNumber = self::generateCopyNumber($book);

$this ->setAttribute('type\_of\_copy', $request->input('type\_of\_copy'))

->setAttribute('price', $request->input('price'))

->setAttribute('copy\_number', $copyNumber)

->setAttribute('book\_id', $book->getAttribute('id'));

if($request->input('type\_of\_copy') == 'referenced'){

$this->setAttribute('copy\_status', 'referenced');

}else {

$this->setAttribute('copy\_status', 'available');

}

$this->save();

}