

# Class App

java.lang.Object [↗](#)  
App

```
public class App
extends Object ↗
```

flow of the app.

Version:

0.0.1 || 9/07/2024

Author:

Ayush Goyal

## Field Summary

### Fields

Modifier and Type	Field	Description
(package private) AdminManager	<code>adminManager</code>	
(package private) BookFileManager	<code>bookManager</code>	
(package private) <a href="#">Scanner</a> <a href="#">↗</a>	<code>sc</code>	
(package private) UserFileManager	<code>userManager</code>	

## Constructor Summary

### Constructors

Constructor	Description
<code>App()</code>	To initialise application

## Method Summary

All Methods	Static Methods	Instance Methods	Concrete Methods
Modifier and Type	Method	Description	
(package private) void	<code>adminMenu()</code>	To display main menu options to the admin.	
(package private) void	<code>adminProfile()</code>	To display admin's profile.	
(package private) int	<code>bookOptions(List&lt;Integer&gt; bks)</code>	To show books of the indexes in the list	

(package private) void	<b><code>createUser()</code></b>	To create new user
(package private) boolean	<b><code>lastValidation()</code></b>	final validation function
static void	<b><code>main(String<sup>↗</sup>[] args)</code></b>	To make app run.
(package private) void	<b><code>searchBooks()</code></b>	To give search options for books
(package private) void	<b><code>userBookList()</code></b>	To display and perform opreations over the user's list
(package private) boolean	<b><code>userLogin()</code></b>	To authenticate user
(package private) boolean	<b><code>userLoginStory()</code></b>	To verify that the user has succesfully logged in or not also to check if person logged int is user or admin
(package private) void	<b><code>userMenu()</code></b>	To display main menu option to the end user.
(package private) void	<b><code>userProfile()</code></b>	To show user's profile and perform updation over it

### Methods inherited from class `java.lang.Object`<sup>↗</sup>

`clone`<sup>↗</sup>, `equals`<sup>↗</sup>, `getClass`<sup>↗</sup>, `hashCode`<sup>↗</sup>, `notify`<sup>↗</sup>, `notifyAll`<sup>↗</sup>, `toString`<sup>↗</sup>, `wait`<sup>↗</sup>, `wait`<sup>↗</sup>, `wait`<sup>↗</sup>

## Field Details

### sc

`Scanner`<sup>↗</sup> sc

### bookManager

`BookFileManager` bookManager

### userManager

`UserFileManager` userManager

### adminManager

AdminManager adminManager

## Constructor Details

### App

```
public App()
```

To initialise application

## Method Details

### lastValidation

```
boolean lastValidation()
```

final validation function

**Returns:**

true if user is sure

### userLoginStory

```
boolean userLoginStory()
```

To verify that the user has succesfully logged in or not also to check if person logged int is user or admin

**Returns:**

true when user logges in.

### userLogin

```
boolean userLogin()
```

To authenticate user

**Returns:**

true if user in true else false

### createUser

```
void createUser()
```

To create new user

### searchBooks

```
void searchBooks()
```

To give search options for books

### bookOptions

```
int bookOptions(List<Integer> bks)
```

To show books of the indexes in the list

**Parameters:**

bks -

**Returns:**

The index of the book in the list which the user has choosen

### userBookList

```
void userBookList()  
    throws IOException
```

To display and perform opeations over the user's list

**Throws:**

[IOException](#)

### userProfile

```
void userProfile()
```

To show user's profile and perform updation over it

### adminProfile

```
void adminProfile()
```

To display admin's profile.

### userMenu

```
void userMenu()  
    throws IOException
```

To display main menu option to the end user.

**Throws:**

[IOException](#)

## adminMenu

```
void adminMenu()  
    throws IOException
```

To display main menu options to the admin.

**Throws:**

[IOException](#)

## main

```
public static void main(String[] args)  
    throws IOException
```

To make app run.

**Parameters:**

args -

**Throws:**

[IOException](#)