**FUTURE UNIVERSITY IN EGYPT**

**FACULTY OF COMPUTERS AND INFORMATION TECHNOLOGY**

|  |
| --- |
| **logo official COMPUTER** |
|
|
| **Course Name:** Software Engineering-2 | |
| **Course Code:** CS352 | |
| **Instructor:** Dr. Ramadan Moawad  **TA:** Amr Mansour  **Semester:** Fall 2020 |  |

**Topic: Library System**

**Student Names & ID:**

* **Mohamed Ashraf Mohamed 20183706**

### Class Diagram



### System Architecture

**Sign Up**

**Case1**

**MODEL**

Check if there is an unfilled data field

**VIEW** user sign up display error mark

**Send request Request**

**CONTROLLER**

Process request

**DATABASE**

**Request data**

**VIEW**

**Respond**

**MODEL**

Fail sign up

**CONTROLLER**

Return data

**Respond Return data**

**Software Architecture for Function Sign Up**

The user will not successfully sign up because of that there is an unfilled data field that should be filled

So when the user press the signup button there will be an error mark beside the field that should be filled

**Case2**

**MODEL**

Check if there is no unfilled field

(Save in data base)

Success signup

**VIEW**

Usersign up display verification page

**Send request Request**

**CONTROLLER**

Process request

**DATABASE**

**Request data**

**VIEW**

**Respond**

**MODEL**

Success sign up

**CONTROLLER**

Return data

**Respond** **Return data**

**Software Architecture for Function Sign Up**

The user will successfully be signed in

And he will fill Name –password-phone number or mail-identical card photo

The user will sign in to have an access on the web app and to access all his services then he presses sign in to do the last step before reaching the main page and this step is the verification

There will be a SMS or mail sent to the user with a verification code then he can reach the web app

**Case3**

**MODEL**

Check database if email is already saved

Fail signup

**VIEW**

Usersign up display error message

**Send request Request**

**CONTROLLER**

Process request

**DATABASE**

**Request data**

**VIEW**

**Respond**

**MODEL**

**CONTROLLER**

Return data

**Respond** **Return data**

**Software Architecture for Function Sign Up**

The user will not successfully sign up because of the entered mail is registered before and already saved in database

So when the user press the signup button there will be an error message that will be shown to him, asking the user to change the email address

**Log in**

**Case1**

**MODEL**

Check data base

**VIEW**

Userlog in display error message

**Send request Request**

**CONTROLLER**

Process request

**DATABASE**

**Request data**

**MODEL**

Fail log in

**VIEW**

**Respond**

**CONTROLLER**

Return data

**Respond** **Return data**

**Software Architecture for Function Log in**

The user or admin will not successfully login cause of wrong username or password

So when the user press the login button there will be an error message that will be shown to him

**Case2**

**VIEW**

Userlog in display login success or

Redirect to page

**MODEL**

Check data base

**Send request Request**

**CONTROLLER**

Process request

**DATABASE**

**Request data**

**MODEL**

Success log in

**VIEW**

**Respond**

**CONTROLLER**

Return data

**Respond** **Return data**

**Software Architecture for Function Log in**

The user and admin will successfully login

Username and password

When the user or the admin enter his username and password and he press login The user will login to access the main page to start search and order for the books he want and enjoying all features as a client but the admin will login to update the webapp or see the sales and the profits (manage the library on the app)

The app will redirect the user to the specified page that they supposed to be in.

**Update books (add or remove)**

**Case1**

**DATABASE**

**MODEL**

Add number of books on the current number of books

**VIEW** Display list of books to add books

**Send request Request**

**CONTROLLER**

Process request

**Request data**

**VIEW**

**Respond**

**MODEL**

**CONTROLLER**

Return data

**Respond Return data**

**Software Architecture for Function Update books (add)**

The admin should be at first successfully logged in

Info about books

The admin will update the data about books on system by adding on the current number of books

**Case2**

**DATABASE**

**MODEL**

Remove number of books on the current number of books

**VIEW** Display list of books to remove books

**Send request Request**

**CONTROLLER**

Process request

**Request data**

**VIEW**

**Respond**

**MODEL**

**CONTROLLER**

Return data

**Respond Return data**

**Software Architecture for Function Update books (remove)**

The admin should be at first successfully logged in

Info about books

The admin will update the data about books on system by removing from the current number of books

**Creation of Analysis Reports**

**Case1**

**VIEW**

The user press salaries to display salaries list

**DATABASE**

**MODEL**

Prepare arithmetic equation of salaries

**Send request Request**

**CONTROLLER**

Process request

**Request data**

**VIEW**

**Respond**

**MODEL**

**CONTROLLER**

Return data

**Respond Return data**

**Software Architecture for Function Creation of Analysis Reports (Salaries)**

The admin will successfully get his reports

By Pressing at the formula he wants to

The admin will press at salaries then he will choose the period he want to see the report in

The report or the list will appear with all data

**Case2**

**VIEW**

The user press profits to display profits list

**DATABASE**

**MODEL**

Prepare arithmetic equation of profits

**Send request Request**

**CONTROLLER**

Process request

**Request data**

**VIEW**

**Respond**

**MODEL**

**CONTROLLER**

Return data

**Respond Return data**

**Software Architecture for Function Creation of Analysis Reports (profits)**

The admin will successfully get his reports

By Pressing at the formula he wants to

The admin will press at profits then he will choose the period he want to see the report in

The report or the list will appear with all data

**Case3**

**VIEW**

The user press sales to display sales list

**DATABASE**

**MODEL**

Prepare arithmetic equation of sales

**Send request Request**

**CONTROLLER**

Process request

**Request data**

**VIEW**

**Respond**

**MODEL**

**CONTROLLER**

Return data

**Respond Return data**

**Software Architecture for Function Creation of Analysis Reports (Sales)**

The admin will successfully get his reports

By Pressing at the formula he wants to

The admin will press at sales then he will choose the period he want to see the report in

The report or the list will appear with all data

**Case4**

**VIEW**

The user press spending to display spending list

**DATABASE**

**MODEL**

Prepare arithmetic equation of spending

**Send request Request**

**CONTROLLER**

Process request

**Request data**

**VIEW**

**Respond**

**MODEL**

**CONTROLLER**

Return data

**Respond Return data**

**Software Architecture for Function Creation of Analysis Reports (Spending)**

The admin will successfully get his reports

By Pressing at the formula he wants to

The admin will press at spending then he will choose the period he want to see the report in

The report or the list will appear with all data

**Order Books**

**Case1**

**MODEL**

Checks user input (requested order and payment method) in database.

**CONTROLLER**

Receives user input (requested order and payment method) and notifies model to process it

**VIEW**

Display Order button

**Send request Request**

**DATABASE**

**R Request data**

**VIEW**

Order completed

Successfully!

**Respond**  **Return data**  **Respond**

**MODEL**

**CONTROLLER**

**Software Architecture for Function Order books**

When the user request an order, view sends the requested order &payment method to the controller to process it. Controller sends request to model to check requested order and payment method in database. When Database responds with correct payment method it responds to the view with a message of completion of taking order.

**Case2**

**VIEW**

Display Order button

**Send request Request**

**DATABASE**

Model

Checks user input (requested order and payment method) in database.

**CONTROLLER**

Receives user input (requested order and payment method) and notifies model to process it

**R Request data**

**VIEW**

Order Not Successful

Try Again

**Respond** **Return data** **Respond**

**MODEL**

**CONTROLLER**

**Software Architecture for Function Order books**

When the user request an order, view sends the requested order &payment method to the controller to process it. Controller sends request to model to check requested order and payment method in database. if Database responds with incorrect/invalid payment method it responds to the view with a message of uncompletion(**failure**) of taking order.

**Payment Option**

**Case1**

VIEW

Displays payment button & order price

**Send request** **Request data Request**

DATADBASE

MODEL

Checks order in database& payment method validity)

**CONTROLLER**

Receives user input (and payment method: visa/cash on delivery) and notifies model to process it

VIEW

Success Payment is completed

**Respond Return data Respond**

MODEL

Payment method is valid

CONTROLLER

**Software Architecture for Function Payment Option**

When the user clicks on payment button, view sends user input (payment method: visa/cash on delivery or PayPal) to the controller to process it. Controller sends request to model to check requested order and payment method validity in database. When Database responds with correct payment method it responds to the view with a message of **completion** of payment.

**Case2**

VIEW

Displays payment button & order price

**Send request** **Request data Request**

DATADBASE

MODEL

Checks order in database& payment method validity)

**CONTROLLER**

Receives user input (and payment method: visa/cash on delivery) and notifies model to process it

VIEW

Failure Payment is not completed

**Respond Return data Respond**

CONTROLLER

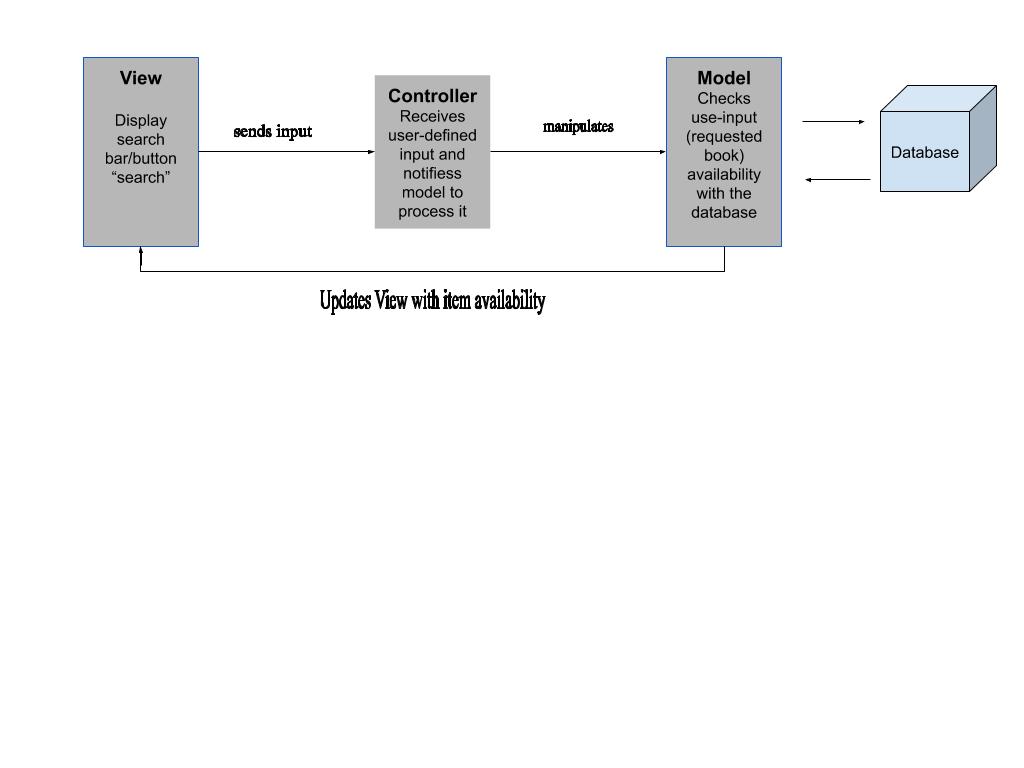
MODEL

Payment method is not valid

**Software Architecture for Function Payment Option**

When the user clicks on payment button, view sends user input (payment method: visa/cash on delivery or PayPal) to the controller to process it. Controller sends request to model to check requested order and payment method validity in database. If Database responds with invalid payment method it responds to the view with a message of in**completion/failure** of payment.

**Search Books**

****

**Software Architecture for Function Search Books**

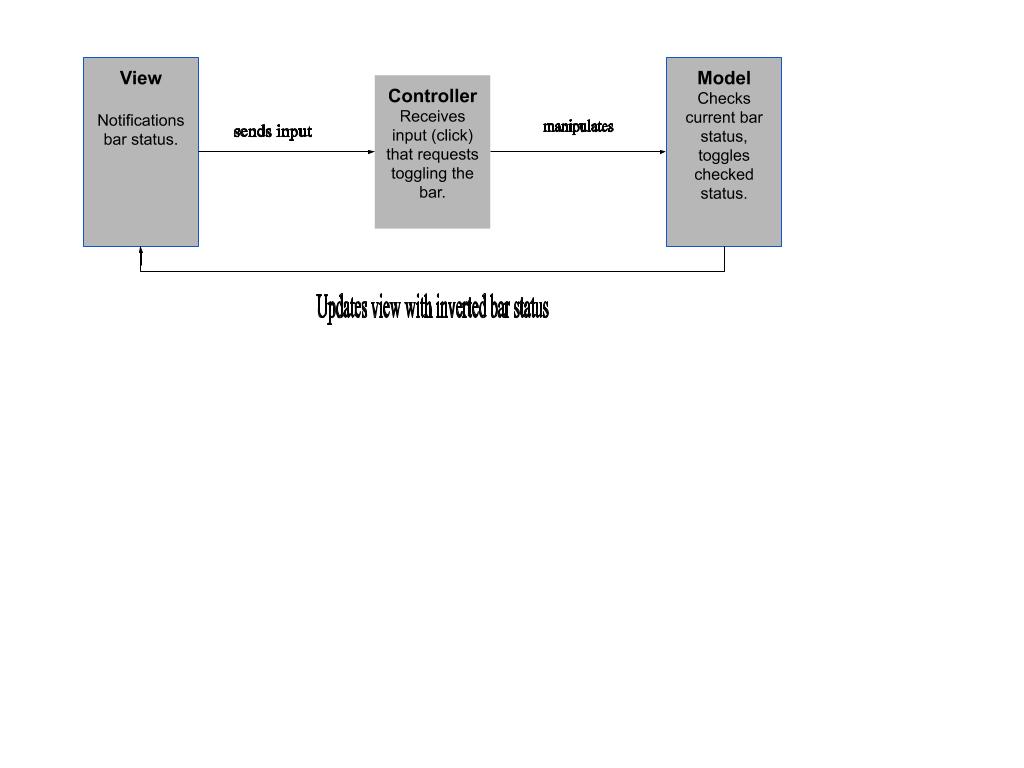
User checking notifications bar at any given time.

User clicks on the bar and triggers an event that sends such input to the Controller, Controller sends it to the Model. And the system checks with the Model the notifications bar current state.

Model toggles the bar to either expand the notifications list, or collapse it.

**Notification Function**

**Case1**

****

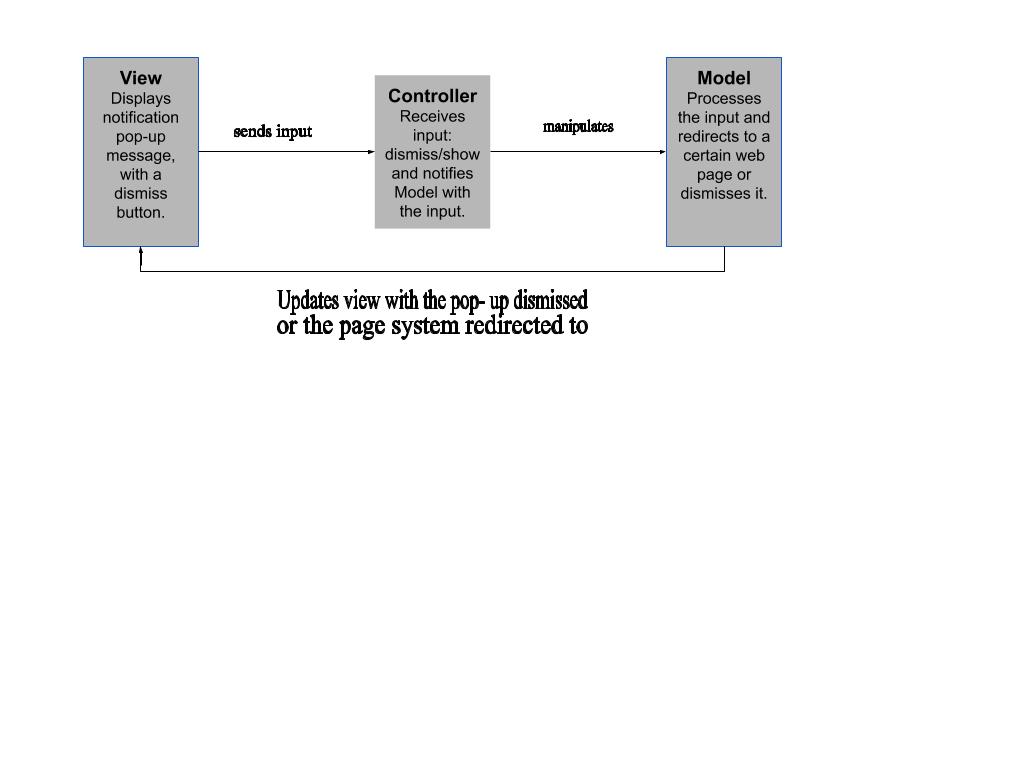
**Software Architecture for Function Notification**

User checking notifications bar at any given time.

User clicks on the bar and triggers an event that sends such input to the Controller, Controller sends it to the Model. And the system checks with the Model the notifications bar current state.

Model toggles the bar to either expand the notifications list, or collapse it.

**Case2**

****

**Software Architecture for Function Notification**

User gets an alert when a new notification is sent.

A pop-up message containing the body of the notification is viewed with:  
-a click option that directs the user to the certain related web page.  
-a button option to dismiss the pop-up.

Pop-up notifications are saved to the notification bar, in case the user needs to recheck them later.

Notification bar automatically deletes older, already checked notifications after 10 days of checking them.

### Unit Test Cases

* **removeBook(String bookName)**

|  |  |  |  |
| --- | --- | --- | --- |
| **input** | **Expected output** | **Actual output** | **Test case result** |
| bookName | True(Book was present) | True | Success |
| bookName | False(Book was not present) | True | Success |

* **searchBook(String bookName)**

|  |  |  |  |
| --- | --- | --- | --- |
| **input** | **Expected output** | **Actual output** | **Test case result** |
| bookName | Book object(Book is present) | Book object | Success |
| bookName | Book object(Book is not present) | Null | Success |

* **setSalary(float salary)**

|  |  |  |  |
| --- | --- | --- | --- |
| **input** | **Expected output** | **Actual output** | **Test case result** |
| Salary (positive) | True (Salary updated) | True | Success |
| Salary (negative) | False (Salary is not updated) | False | Success |

### System Test Cases

* **Search Product Test Case**
* **Test Case ID:** 1.1 **Test Designed By:**
* **Test Priority (Low/medium/Large):** Med **Test Designed Date:**
* **Module Name:** search book **Test Executed By:**
* **Test Title:** search book successfully **Test Execution Date:**
* **Pre-Conditions:** User login to the library system
* **Dependencies:**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Step | Test Steps | Test Data | Expected result | Actual result | Status (Pass/Fail) | Notes |
| 1 | Search for a book | Book Name | Ask if user wants to order the book | Ask if user wants to order the book | pass |  |
| 2 | Order Book | Book Name | User is navigated to enter the payment method | User is navigated to enter the payment method | pass |  |
| 3 | Payment Method | Payment method (cash or visa card) | Message: Book is ordered successfully (if method was accepted) | Message: Book is ordered successfully | pass |  |

* **Order Book Test Case**
* **Test Case ID:** 1.2 **Test Designed By:**
* **Test Priority (Low/medium/Large):** Med **Test Designed Date:**
* **Module Name:** order book  **Test Executed By:**
* **Test Title:** order book successfully **Test Execution Date:**
* **Pre-Conditions:** User login to the library system
* **Dependencies:**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Step | Test Steps | Test Data | Expected result | Actual result | Status (Pass/Fail) | Notes |
| 1 | Navigate to add book name | Book Name | navigate to books | navigated to books | pass |  |
| 2 | Order Book | Book Name | User is navigated to enter the payment method | User is navigated to enter the payment method | pass |  |
| 3 | Payment Method | Payment method (cash or visa card) | Message: Book is ordered successfully (if method was accepted) | Message: Book is ordered successfully | pass |  |

* **Delete Book Test Case**
* **Test Case ID:** 1.3 **Test Designed By:**
* **Test Priority (Low/medium/Large):** Med **Test Designed Date:**
* **Module Name:** delete book  **Test Executed By:**
* **Test Title:** delete book successfully **Test Execution Date:**
* **Pre-Conditions:** Staff login to the library system
* **Dependencies:**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Step | Test Steps | Test Data | Expected result | Actual result | Status (Pass/Fail) | Notes |
| 1 | Navigate to  book name | Book Name | Staff must navigate to books | Staff navigated to books | pass |  |
| 2 | Delete book from the system | Staff enter book name and select to delete | Message: Book removed successfully | Book removed successfully | pass |  |
| 3 | Delete book from the system | Staff enter book name and not found the book in system | Message: Book is not present in the library | Book is not present in the library | pass |  |