MUSEUM APP DOCUMENTATION

Presented to: Dr. Manal Ismail

MUSEM APPLICATION PROJECT

MADE BY:

DINA SALAH SALEM NASR

HOSAM MAGDI LOTFI ABDEL SALAM

SEIF MOSTAFA ABDELHALIM

SHROUK ASHRAF EL SAIED EL SAIED

ZIAD HESHAM SALAH EL DIN

09/01/2022

Individual Contribution to the project:

First of all we would Like to thank Dr. Manal and Engineer Ahmed Saied and Engineer Selwan for their guidance throughout the project.

Dina Salah:

1- Specifying System Requirements including Functional and Non-Functional Requirements.
2- Developing the GUI using java programming language
3- Backend and GUI integration
4- Developing Class diagram and Sequence Diagram

Hosam Magdi:

1- Specifying Use cases based on the function requirements, determining stakeholders and actors.
2- Drawing Use Case diagram for the project
3- Developing the Backend of the project using Java programming language.
4- Backend and GUI integration

Seif Mostafa:

1- Designing System Interfaces for the overall project.
2- Developing the GUI using java programming language
3- Backend and GUI integration
4- Developing the DFD for the project

Shrouk Ashraf:

1- Providing clear explanation for the class diagram.
2- Developing the Backend of the project using Java programming language.
3- Backend and GUI integration
4- Specifying User requirements in the project.
5- Developing Early iteration for the ERD.

Ziad Hesham:

1- Obtaining problem statement, developing WBS and Gantt chart.
2- Developing the Database for the entire system and preparing the database handler class.
3- Backend and GUI integration
4- Developing the ERD for the project.

1- PROBLEM DESCRIPTION:

When visiting any museum, usually every tourist have a favourite pieces or specific monuments they want to visit, as many museum opens recently and the number of monuments and art pieces can be huge, the visit is usually wasted on trying to find what the tourist want to see, finding its whereabouts within the museum, and being amazed by the amount of monuments they didn't know it existed, also for someone interested in museum events, checking the facebook pages isn't always practical, in addition to the limited number of tickets per day, so in case a tourist could go all the way to the museum and not be able to enter due to the limited tickets number.

All of these problems could make visiting a museum a huge burden, meanwhile, exploring culture and history should be easy for everyone, and also for tourists who have a huge contribution in increasing national income.

So if a tool to be made to regulate the whole process, it could encourage more people to come and visit the museums more often, in addition to many people waiting the Grand Opening of The Grand Egyptian Museum, it's expected that the museum waiting list will be huge for the first weeks of opening, so it's needed to have a ticket booking service to solve this issue, in addition to containing a huge number of monuments, and expected that the tour guide will be heads over heels in order to cover all monuments, some kind of documentation service needs to be provided, in order to explain each monument, where it came from and the entire history of it, to lower the burden on the tour guides, and so that every visitor can have a full experience where the don't have to wait in ,line for tickets or be told to come in another day, in addition to the get to see all the monuments and actually be aware of the full story behind each monument.

As many apps are developed everyday if an App where to be made, which could contain all monuments in the museum their exact whereabouts and the history behind each monument.

Also this app could show the events made by the museum, their time, description and to have a registration service for each event.

Also for visiting tickets, this app could provide a visitor ticket purchasing service, where in this case the visitor won't have to get in line for the tickets or come back another day in case of tickets running out, also an online payment service for people who find it easier to book using their visa without having to

wait in line for tickets payment, to decrease the workload in the crowded days, and decrease the waiting time in lines which could ruin the whole experience.

Meanwhile offering an offline payment option for younger people who doesn't own visa or prefer to pay offline.

Also in the pandemic it's more convenient for minimizing the spreading of infection to have all payments and papers be online, from here providing an online payment service and online tickets can help achieve that.

In addition to having a full museum map, which will make it much easier for visitors to locate their favourite pieces and go to it without wasting time wandering around searching for the monuments.

Also having an explore page to post interesting posts and topics could increase people willingness to visit the museum and explore more about history.

In addition to having a log with all existing monuments in the museum could help the visitor identify whether the monument they want to see exists in this museum or no.

So an app was made for this user experience can be elevated a great deal which will encourage more people to visit museums, hence increasing national Income and also spreading culture and history among citizens and tourists.

2- System Requirements

a. Functional Requirements

Requirement Label	Description
REQ-I	Having a user account to hold user data for ticket booking and surveying purposes. Including(Name, Email, National ID, Gender, State, password for the account, and age)
REQ-II	Having an explore page categorizing types of monuments to ease the navigation
REQ-III	Having a detailed description for each monument, showing its era, size, owner, and the story behind it.
REQ-IV	Having a visitation ticket booking service, specifying date of visit, National ID, Name, Phone number.
REQ-V	An events list page, holding event titles, brief description and advertising image.
REQ- VI	An event description page, where event details are written, its whereabouts, date and time.
REQ- VII	An event registration page, where interested personal can register for attending the events with their information including National ID, Name, Phone number.
REQ-VIII	A payment selection page, where user can choose the option between paying Online or Offline.
REQ-IX	An online payment page where the user enter their Credit Card info to be sent to the payment portal
REQ-X	Having offline payment option for users who can’t pay online
REQ-XI	A map of the museum available for users.

b. Non-functional Requirements

XII- Reliability and capacity : the app should withstand an average of 200 users, and should have error handling to avoid crashing.

XIII- Scalability: the GUI should be made using dynamic layouts to support further scalability and addition of multiple Art pieces and painting.

XIV- Security: User data should be secure against cyber hacking attempts.

XV- Localization: this app should have localization features, thus a way to change all the language and description should be considered in the design of the system.

XVI- Usability: the app is used to be specially used as an interface for the grand museum of Egypt

Platforms: the application is initially stated to be developed for desktop users, with the capability to be converted into a mobile build with minimal changes.

Interoperability: the app should have only one instance of any manager/ driver classes.

Safety: Servers shall be kept in the museum manager office.

3- Functional Requirements Specification

a. Stakeholders:

- Tourists
- Egyptian Visitors
- Tour Guides
- Museum Manager
- Investors
- Ministry of Tourism
- Online payment (Banks)

b. Actors:

- System User: user is the initiating actor of the system, by Logging into the system the user can have multiple interaction with system use cases, either by looking up monuments or events, booking a ticket, registering for event, paying, and exploring.
- New User: initiating actor, whereby signing up they turn into System users.
- Admin: Initiating actor, in which the upload events data and any system updates.
- Online payment system: participating actor, in which payment operation is done by them.

c. Use Cases:

I- Use case Description:

Login: The Login takes the users email and password and search in the data base for the email, if found passwords are compared from user input and the database, if verified the user can continue in the app, if not a Login error occurs.
Requirement: REQ-I

Signup: a signup page so new users can create their account in the app and access app features, basic user information included (National ID, Email, Full Name, password, Age, Gender, State).
Requirement: REQ-I

Explore Categories: This use case is for categorizing the monuments.
Requirement: REQII – REQ-III.

Show categorize Name: Showing categories names in the App, where the user can select a category and explore the monuments in this category.
Requirement: REQII – REQ-III.

Book a visit: Allowing the User to book a visit ticket.
Requirement: REQ-IV

Fill in Visit info: Allow the user to specifying the date of visit, and including visiting user information, meanwhile specifying ticket price based on the type of visitor.
Requirement: REQ-IV

Make Payment: Specifies type of payment user will use (Online or Offline)
Requirement: REQ-VIII

Pay by Credit Card: Makes the user fill his credit card information so it would be sent to online payment portal.
Requirement: REQ-IX

Pay Cash: shows the user their ticket specifications, meanwhile expecting them to pay in cash.
Requirement: REQ-X.

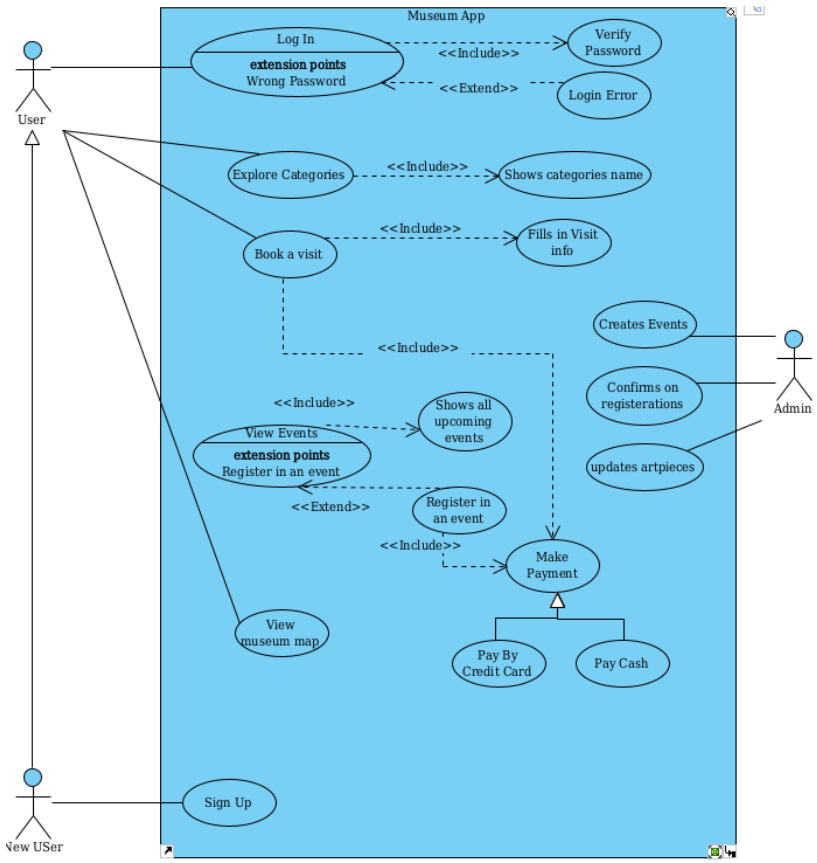
View Events: The user can view available events, meanwhile each event having an event panel including image, brief description of event, and user can
Requirement: REQ-V.

Event Details: The user is able to view event description, date and register for the event.
Requirement: REQ-VI.

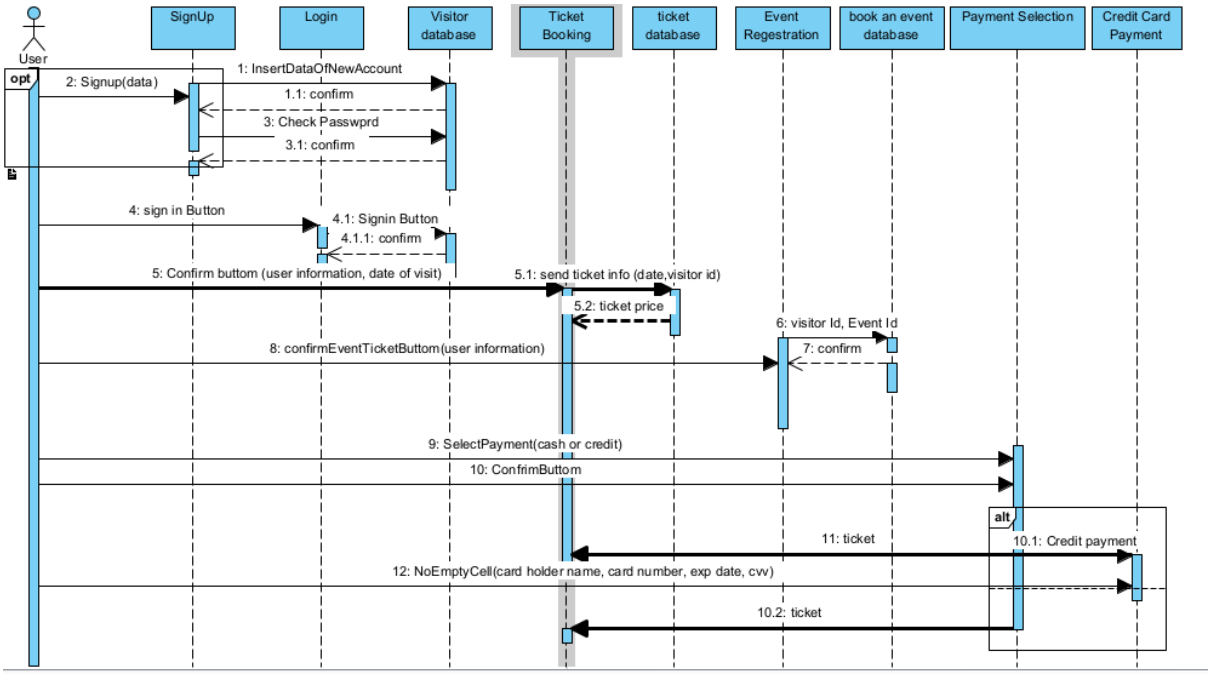
Register in an event: The user can register in the event with their information.
Requirement: REQ-VII.

View museum Map: shows a full map in the museum so the user can localize the monuments they want to see.
Requirement: REQ-XI.

II. Use case Diagram:

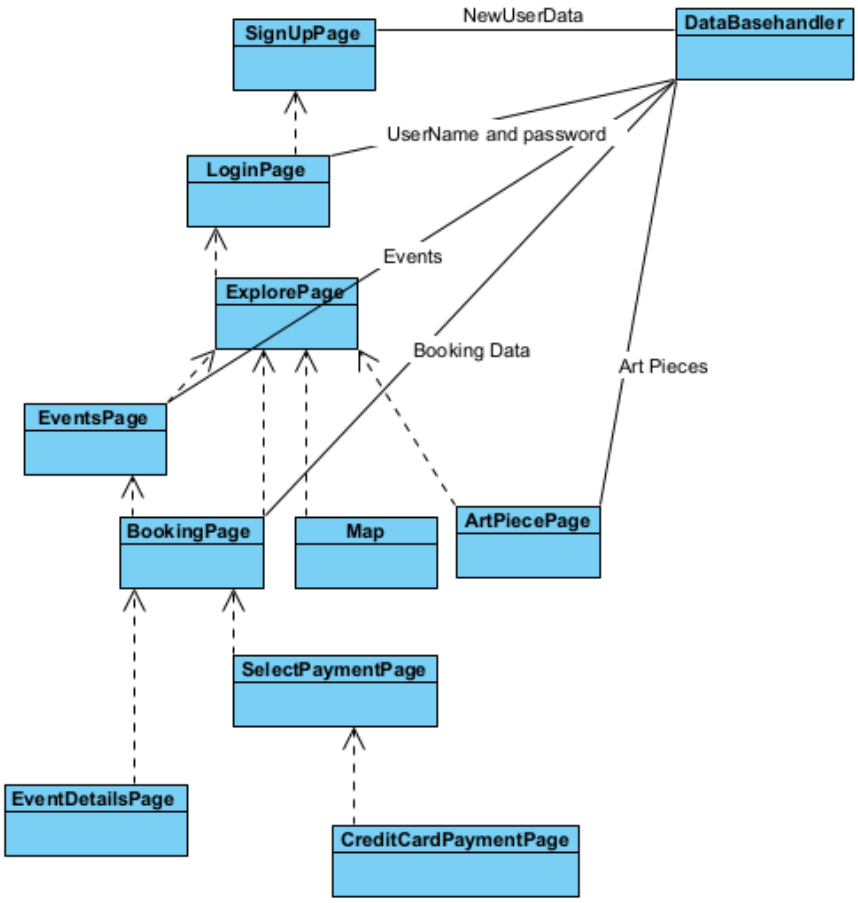


d. System Sequence Diagrams



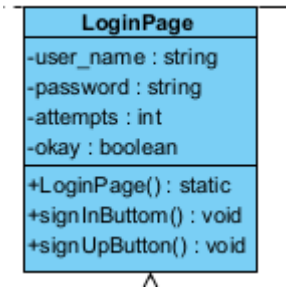
e. Class Diagram and Interface Specification

1- Overview class diagram:

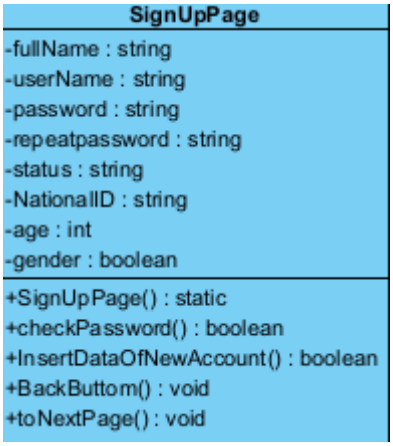


2- Partial Classes:

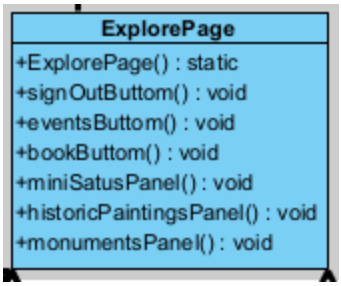
1- LoginPage: First Class of the program, associated with DataBasehandler class for information checking.



2- SignupPage: depends on LoginPage as you can't move to SignUp without Login page, associated with DataBasehandler class for data storing.

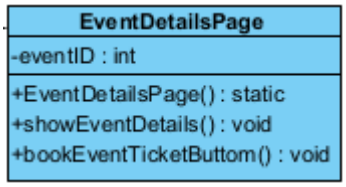


3- ExplorePage: Depends on LoginPage as a user can't access the system without Login.

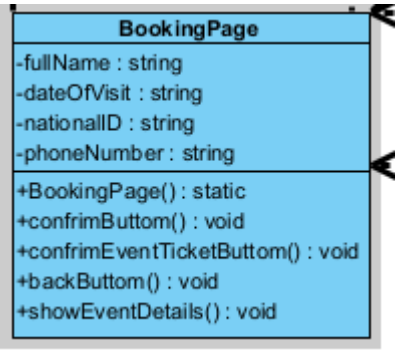


4- EventsPage: Depends on the Explore as it's accessed through it, and associated with DataBasehandler as events are imported from database.

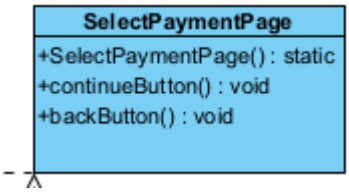
5- EventDetailsPage: Depends on EvengtsPage, as the user access event details when he clicks on specific event in Eventspage, and associated with DataBasehandler as events details are imported from database.



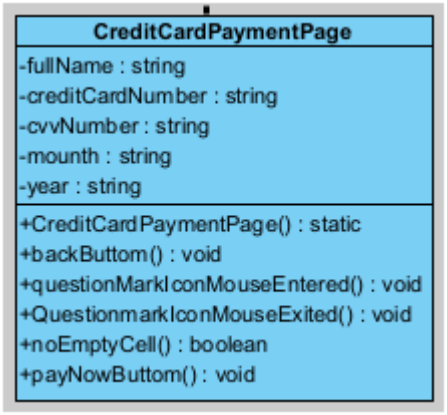
6- BookingPgae: The class depends on EventDetails as Event Booking option is in it and also normal Ticket booking so it also depends on ExplorePage in which normal Ticket booking option is available, and associated with DataBasehandler as user Id and event Id are registered in database.



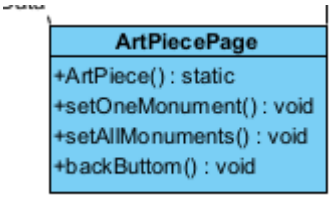
7- SelectPaymentPage: Depends on BookingPage.



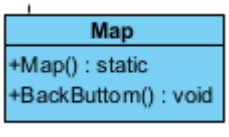
8- CreditCardPaymentPage: Depends on PaymentSelectionPage.



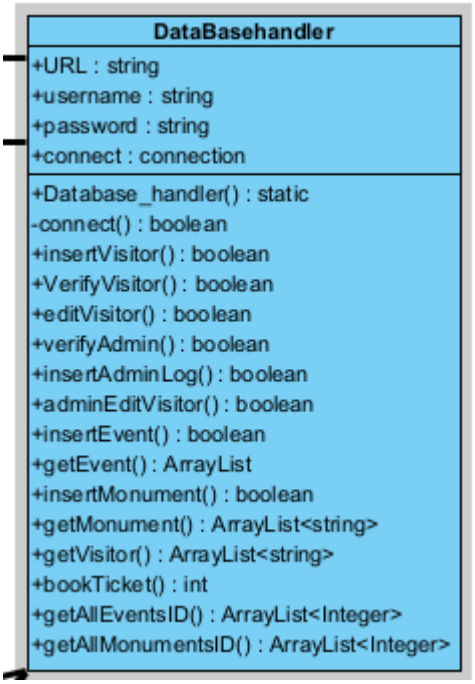
9- ArtPiecePage: Depends on ExplorePage.



10- Map: Depends on ExplorePage.



11- DataBasehandler: Associations are explained in the previous classes.



3- Class Specification:

In loginPage class:

Its attributes are:

1) user_name and password: private of type string.

2) Attempts: it is private of type string, which is the number of attempts the user enter wrong password or user_name.

3) Okay: it is private of type boolean, it verifies the user_name and password from the database. All of this attributes are private.

This class has 2 functions:

1- SignInButtom():it is a public method. It receives these Parameters: Two JTextField for username and password, one JLabel for error message and two JFrame for current page and next one. This function takes information of account from GUI, verifies them using database, displays what kind of error happens incase of wrong information added and gives user 3 attempts to handle this error. This function is used in action of clicking the sign in button in GUI. This function returns void.

2- SignUpButton(): It is a public method. This Function receives these parameters: two JFrame for the current page and the next one. It closes the current page and opens the next one. It returns void.

In SignUpPage class:

1) It's attributes are:

fullName, username, password, repeatpassword, status and nationalID: they are private of type string , and age that are private of type int and gender which is private of type boolean. All of these attributes represent the information of the user to signUp.

2) The functions are:

1- checkPassword: It is a public method. This Function receives these parameters: Two JTextField for the password and repeatpassword, one JLabel for error message. This function takes password of account from GUI and checks if there is an error and display it. It returns Boolean.

2- InsertDataOfNewAccount: It is a public method. This Function receives these parameters: Five JTextField for the name, national ID, phone number, email and age. And JPasswordField for the password and two JComboBox of type string for Gender and Status. It takes these information from GUI, adds them in the database. It returns Boolean.

In ExplorePage class:

The functions of explorePage are:

1- signOutButtom(): It is a public method. This function receives two JFrame, one for the explore page and the other for sign in page. This function moves us to signin page, closes current page and returns void.

2- eventsButtom(): It is a public method. This function receives two JFrame, one for the explore page and the other for events page. This function moves us to event page, closes current page and returns void.

3- bookButtom(): It is a public method. This function receives two JFrame, one for the explore page and the other for book page. This function moves us to book page, closes current page and returns void.

4- miniStatusPanel(): It is a public method. This function receives two JFrame, one for the explore page and the other for category page. This function moves us to category page, closes current page and returns void.

5- historicPaintingsPanel(): It is a public method. This function receives two JFrame, one for the explore page and the other for category page. This function moves us to category page, closes current page and returns void.

6- monumentsPanel(): It is a public method. This function receives two JFrame, one for the explore page and the other for category page. This function moves us to category page, closes current page and returns void.

In EventsPage class:

It has one fuction which is createAllEvents: It is a public method. This Function receives one parameter which is JPanel for the eventPanel in the GUI. It takes all event IDs from database in an arraylist. And it takes all the event datails from the database according to ID. Then add the datails to the panel in the GUI. It returns void.

In EventDetailsPage class:

It has one attribute which is eventID that is private of type int. It is used in get the event details from the database.

The functions are:

1- showEventDetails(): It is a public method. This function receives array of JLabels to display details of each event in GUI using these JLabels by getting these details from database using event ID given from pervious page in GUI. This function returns void.

2- bookEventTicketButtom(): It is a public method. This function takes two JFrame, one of the current page and the other for the next page. This function moves us to book an event ticket page, closes the current page and returns void.

In ArtPiecePage class:

Its functions are:

1- setOneMonuments(): It is a public method. This function receives monument ID and JPanel, gets information of monument from database using ID, display monument's information in GUI and returns void.

2- setAllMonuments(): It is a public method. This function receives JPanel to display all monuments in GUI. This function calls setOneMonument function inside it and gives it all IDs of monuments and returns void.

In BookingPage class:

Its attributes are: fullName, dateOfVisit, nationalID and phoneNumber all of these are private of type string. They represent all the information of the visitor.

Its functions are:

1- confirmButtom(): It is a public method. This Function receives these parameters: Four JTextField for the fullname, national ID, phone number, date of visit and two JFrame for the current page and the next one and one JLabel for error message. This function takes all information from GUI. It displays what kind of error happens in case of empty text. It adds information in the database then closes the current page and opens the next one. It returns void.

2- confirmEventTicketButtom():This Function receives these parameters: Three JTextField for the fullname, national ID and phone number and receives JLabel for the error message and two JFrame for current page and next one. This function takes all information from GUI. It displays what kind of error happens in case of empty text. It closes the current page and opens the next one. It returns void.

In SelectPaymentPage class:

Its functions are:

1- SelectPayment(): It is a public method. This function receives two JRadioButton for cash and credit card payment and returns Boolean indicate whether the user select cash or credit.

2- continueButton(): It is a public method. This function receives 3 JFrame for credit page, cash page and current page and receives a Boolean which is the return of the selectPayment method. This function moves us to the page related to the selected payment method and return void.

In creditCardPaymentPage class:

Its attributes are:

fullName, creditCardNumber, cvvNumber, mounth and year. These attributes are private of type string that has the information of the creditcard the user used.

Its functions are:

1- questionMarkIconMouseExited(): It is a public method. This function receives JLabel for msg to appear when user's mouse exit from question mark icon and returns void.

2- questionMarkIconMouseEntered(): It is a public method. This function receives JLabel for msg to appear when user hovers on question mark icon and returns void.

3- noEmptyCell(): It is a public method. This function receives three JTextField for credit card number, Cvv and name of user, two combox for month and year and JLabel for error msg. This function collect information of credit card and user and returns Boolean indicate whether there is an empty cell or not.

4- payNowButtom(): It is a public method. This function receives two JFrames for current page and next one and a Boolean which is the return of noEmptyCell Method. If noEmptyCell return true, it will take us to next page. This function returns void.

In DataBasehandler:

Its attributes are: username, URL and password they are public of type string to represent the information of database.

Its functions are:

1- connect(): it is a public method. This function used to connect with database and print wether connected or not. It returns boolean.

2- insertVisitor(): It is a public method. It receives Name, National_ID, Phone_number, Password, Email and status of type string. It also receives age of type int and gender of type boolean. It is used to insert data of visitor to database. It returns boolean.

3- verifyVisitor(): It is a public method. It receives email and password of type string and verifies whether they are in database or not. It returns boolean.

4- editVisitor(): it is a public method. It receives Name, National_ID, Phone_number , Password , Email and status of type string. It also receives age of type int and gender of type boolean. It is used to update any information for the user in database. It returns boolean.

5- verifyAdmin(): It is a public method. It receives email and password of admin of type string. It verifies whether the admin in database or not. It returns boolean.

6- insertAdminLog(): it is a public method. It receives Admin_Email of type string, and edited_visitor_ID , edited_Event_ID , edited_monument_ID of type int. it insert admin information in database. It returns boolean.

7- adminEditVisitor(): It is a public method. It receives Name, National_ID, Phone_number , Password , Email and status of type string. It also receives age of type int and gender of type boolean. And edit visitor information by the admin. It returns boolean.

8- insertEvent(): It is a public method. eventName , eventStartTime , eventEndTime , eventDate ,eventDescription and URL they are all of type string. This function inserts event information in the database. It returns boolean.

9- getEvent(): it is a public method. It receives the event ID of type it and returns all event information in arraylist. It returns arraylist.

10- insertMonument(): it is a public method. It receives String Name,int barcode , String breif,int Tag , String URL. And insert all this monument information in the database. It returns boolean.

11- getMonument(): it is a public method. It receives ID of the monument of type int and returns all monument information in arraylist. It returns arraylist.

12- getVisitor(): it is a public method. It receives the event ID of type int and return all visitor information in arraylist. It returns arraylist.

13- bookTicket(): it is a public method. It receives String Email , String Date,int price. And return the ID of the ticket. It returns int.

14- bookEvent(): it is a public method. It receives visitor email of type string and event ID of type int. it inserts in book an event table in database the visitor ID and event ID. It returns void

14- getAllEventsID(): it is a public method. It takes all events ID from database and returns it in an arraylist. It returns arraylist.

15- getAllMonumentsID(): it is a public method. It takes all monuments ID from database and returns it in an arraylist. It returns arraylist.

For all classed there are common methods which are:

1- backbutton(): It is a public method. This function receives two JFrames of pervious and current page, moves us to pervious one and returns void.