Advanced Programming Project Report

Team members

- Shahad malibari ID441010641
- Maarib al-sulimani ID441005504
- Mafaz basalamah ID439002072
- Manar alabdali ID438007643
- Razan al-refaey ID441005995

the leader

My VOYAGE

PROJECT IDEA

this project could be an organizational tool the helps anyone struggling with planning trips. because it will be with the user hand by hand in planning phase from the first step which is define the target city and define what are the locations the user wants to visit in this city. and it comes with a nice feature which is tracking the budget and dates in the whole trip.

The Target Group:

anyone struggling with planning trips and travels a lot.

aim of the project:

help any traveller in the planning phase and make the tracking a simple manner to the traveller to give him/her a peaceful mind and enjoy your trip to the most.

Project Functionalities

1-The ability of creating an account and save all user information in our Database.
2-Give the user a specific flow of data inserting for the cities and locations to make the application simpler and understandable.
3-The ability of viewing user information, cities information, locations information and the To-Do list.
4-Tracking the budget via a visualize report.

Project Design and Implementation

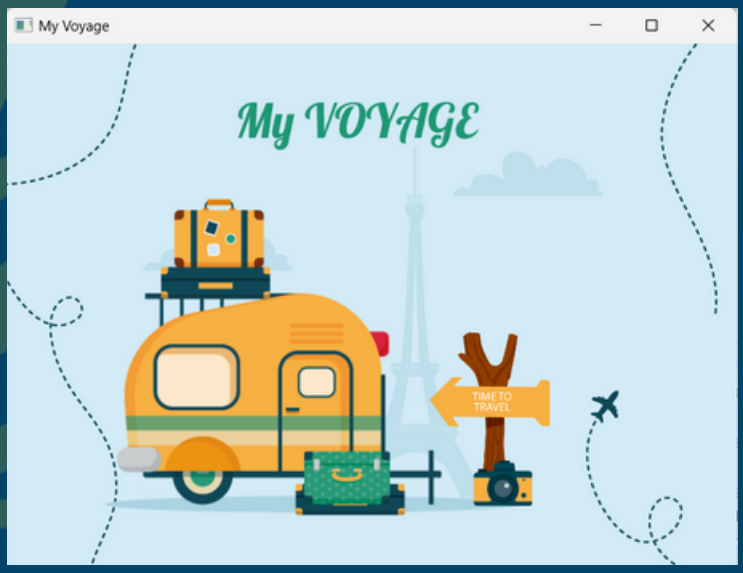
A. Graphical User Interface

What have been designed?

In start after confirming the idea, we started to analyse what we want in our application and then we start to draw hand-sketches for each interface.
then we made a design (actual UI using adobe Illustrator) and then we build the scenes in scene builder with depending on that design.

How many interfaces? And What is the purpose of each interface? and what is the flow of our application?

In total we have (13 scenes including the animation scene).
and in the next few pages there are the screenshots with explanation of each one purpose and we will view the flow of the application.

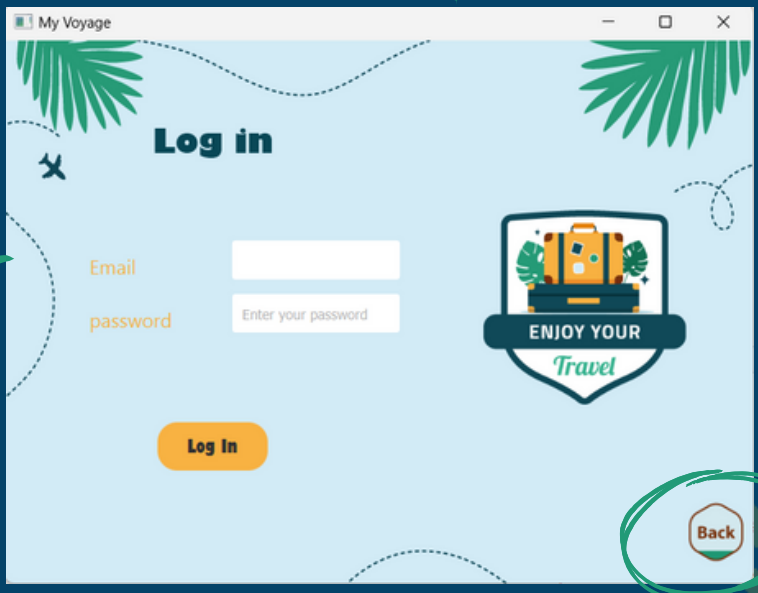


When the user run the application the animation scene will be displayed.

by clicking anywhere

the “welcome” interface will be displayed then the user should choose”between “login” or “sign up”.

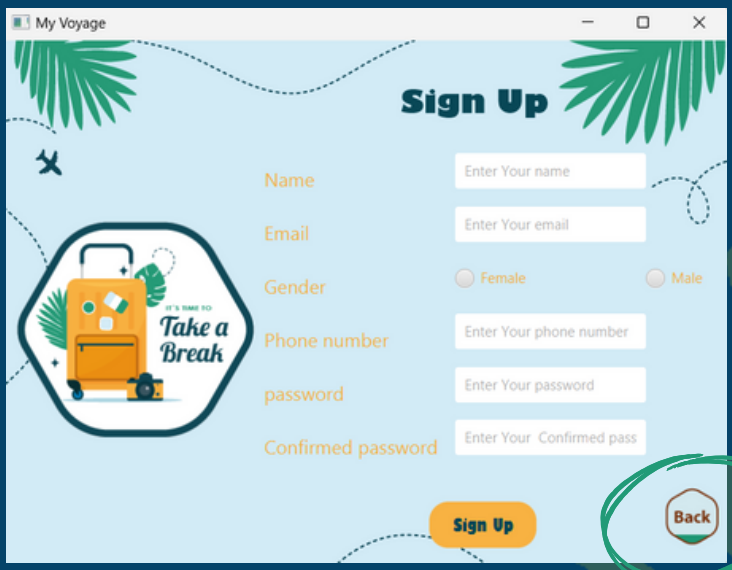
by clicking login button



the “login” interface will be displayed then the user should fill the form to continue if he/she already has an account.

return to welcome interface

by clicking sign up button



the “sign up” interface will be displayed then the user should fill the form to continue in creating an account.

return to welcome interface

after clicking sign up or login buttons

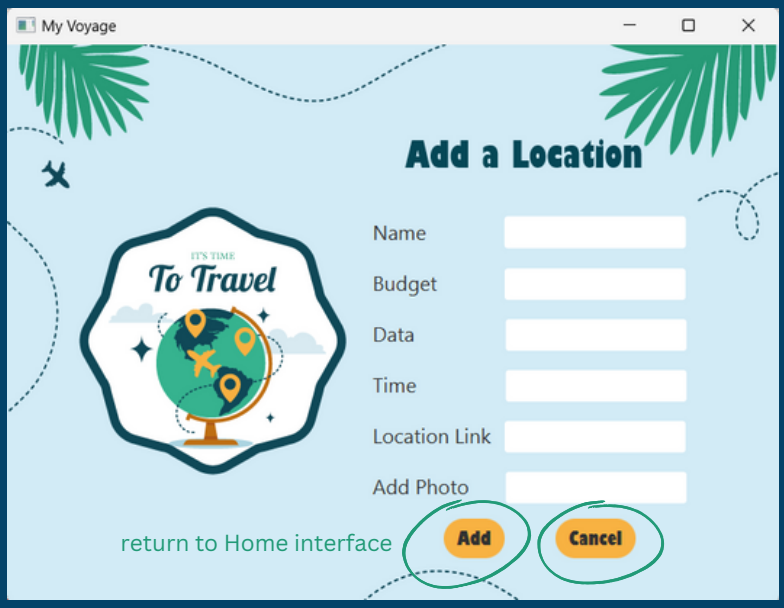
After validation process the user will be driven to the “Home” which display a navigation bar on the left for (home, profile, to do list and budget report) interfaces.
there is combobox to hold all user's cities.
The city information card in “Home” has 3 buttons (add a location, show details and done).

by clicking new trip

the form will be displayed and the user should fill the form and clicking "add" to continue and the new city information will be viewed in the city combobox in the home interface. or click on cancel to return to the home interface.

return to Home interface

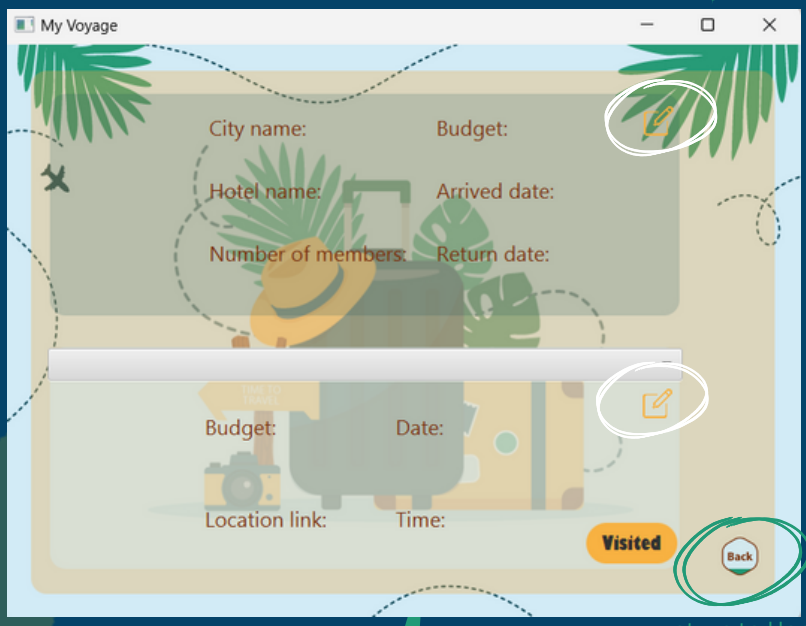
by clicking add location button



the form will be displayed and the user should fill the form and clicking add to continue and the new location will be linked to the city.
or the user can click on cancel and he/she will be returned to the home interface.

by clicking show details button

here we display the complete information about the city and list of all locations linked to this city.
In this interface the user can view all information that he/she wrote in "add city" and "add location" forms.
And we will give the user the ability to edit them if he/she click on “edit” icons.

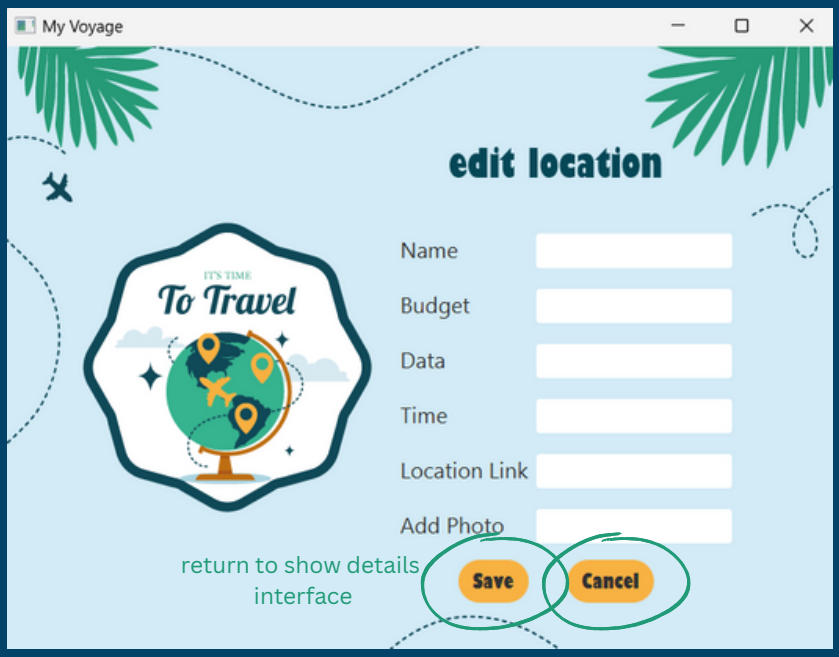


return to Home interface

by clicking edit city icon

the city information will be displayed in the form so the user can edit any information and it will be updated if there is any and then the user will be returned to "show details" interface.

by clicking edit location icon

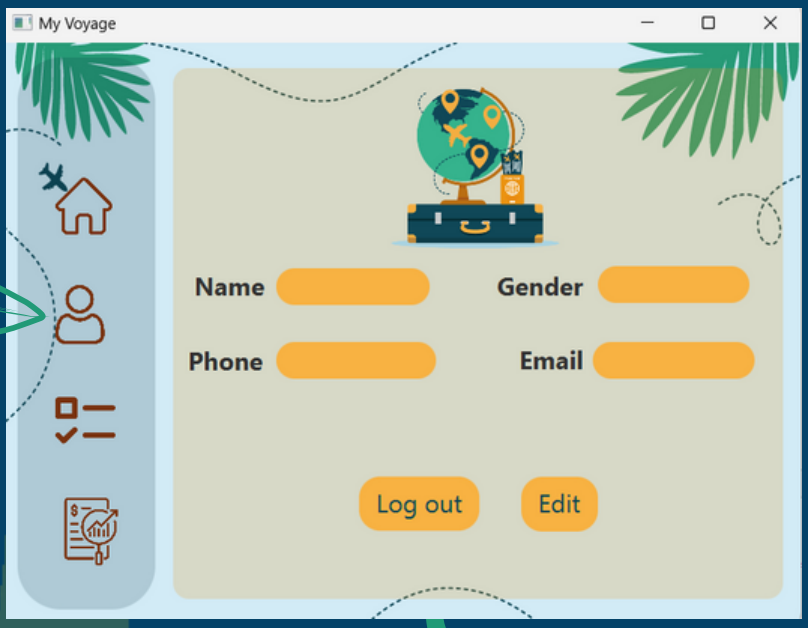


the location information will be displayed in the form so the user can edit any information and it will be updated if there is any and then the user will be returned to "show details" interface.

Pop up window: The "visited" button in location information card will promote the user to enter the money expended in that location so we can track the user budget and show it to the user in “budget report” interface.

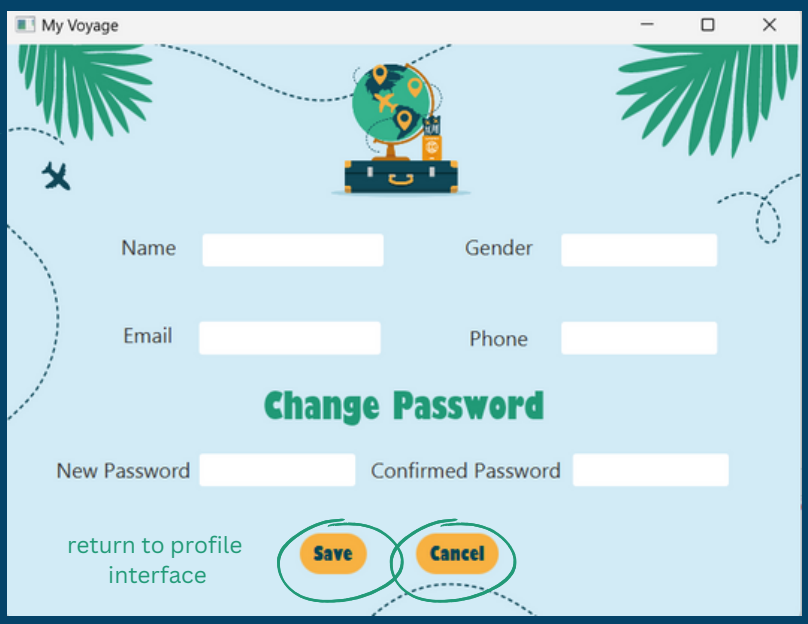
by clicking visited button in show details

by clicking second button in nav bar



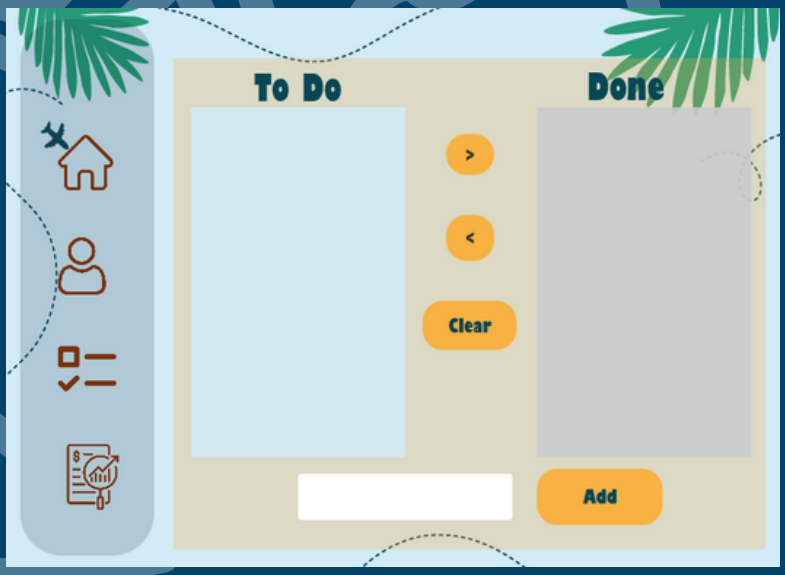
the user can view his/her account information. and from here he/she can either logout and it will be returned to “welcome” interface, or the user can edit his/her information.

by clicking edit button



the user information will be displayed in the form so the user can edit and the information will be updated if there is any then the user will be returned to "profile" interface.

by clicking third button in nav bar



the user can view his/her tasks which categorized into 2 lists: To Do tasks and Done tasks.
and the user can move the tasks between the 2 lists by select the task and click the arrows buttons.
the display will depend on the "finished" attribute
if not finished the task will displayed in ToDo otherwise it will be displayed in Done .
there is a textfield to write what to do and add it to ToDo listview.

in this interface we will display a progress bar for user spending of money depending on his/her budget when he/she filled the city form.
the progress bar changes depending on which city by selecting one from the combobox.

by clicking forth button in nav bar

B. Event-Driven Programming

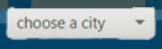
How many events?

in total there are (55) events. they are between Action, Mouse and Key events. also there are number of listeners.
in few next pages we will mention all of them with an explanation for these actions and there sources.

the source of TextInputDialog and is included in the c
ode file in the documentation) .

Type of Action	the source of the event	event handling
Mouse Event	in Animation	when the user clicks anywhere, he/she will be driven to Welcome interface.
Action event	in Welcome 	this event just drive the user to the "Login" interface
Action event	in Login 	this event return the user to the previous interface which is "Welcome"
Action event	in Login 	When you click on the button, it checks first: If the required data is present, if it does not exist, the message “Please Enter Your Email and Password” appears. Second: It verifies the data if it exists separately. If the password is not entered, the message “Please Enter Your Password” appears, and when the email is not entered, the message “Please Enter Your Email” appears. Third: Verify the correctness of the email writer. If the email is not written in the correct way, an "Incorrect Email" message appears. Fourth: It checks whether the information exists in the database so that it is allowed to enter. If the data entered is not present in the database, the message “You don’t have an account” appears. When all fields are completed, and Moving to Home Page and this event will send the user data to the Home Controller so we can pull the cities from the database and add them to the combobox.
Action event	in Welcome 	this event just drive the user to the "SignUp" interface
Action event	in SignUp 	this event return the user to the previous interface which is "Welcome"

Type of Action	the source of the event	event handling
Action event	in SignUp 	When you click on the button checks First: If the required data is present, if it is not present, the message “Please fill out the complete form” appears. Second, it checks the data entered separately 1- When writing the name, the user is required to enter only letters If you enter wrong information, a message appears, "Plases Enter Your Name" 2- Verify that the email is written in the correct way in case an incorrect email is entered The message "Incorrect Email" appears. 3- When typing the phone number, the user is required to enter 10 numbers in case the data is entered incorrectly The message "Incorrect Phone Number,must 10 digits" appears. 4- When typing the password, the user is required to enter at least 4 digits. If an incorrect password is entered, the message "Incorrect Password, At least 4 digits" appears. 5- The password is checked again. When the password does not match, the message “Password not Matches” appears. 6- A variable has been set for the gender. If any gender is selected, male or female, the value of the variable is sent to the user database When all fields are completed, the information is stored in the user's database and Moving to Home Page.
Action event	in Home 	this event sends user information to the Profile controller then drives the user to Profile interface. when the Profile controller get the user information it will display the data in UI controls.
Action event	in Home 	this event sends user information to the TodoList controller then drives the user to TodoList interface. when the TodoList controller gets the user information it will pull user tasks from the database and display the finished in Done listview and the not finished will be displayed in To Do listview.

Type of Action	the source of the event	event handling
Action event	in Home 	this event sends user information to the BudgetReport controller then drives the user to BudgetReport interface. when the BudgetReport controller get the user information it will pull user's cities from the database and display them in the cities combobox.
Action event	in Home 	this event sends user information to the Home again so we can maintain the information and does not disappear.
Mouse event	in Home 	this event sends user information to the AddCity controller then drives the user to AddCity interface. so we can link the new city with that user
Action event	in Home 	this event sends user and city information to the AddLocation controller then drives the user to AddLocation interface. so we can link the new location with that city.
Action event	in Home 	this event sends user and city information to the ShowDetails controller then drives the user to ShowDetails interface. so we can pull and display the correct informations about the city and locations.
Action event	in Home 	this event will display a pop-up window that coungrats the user because he/she is done of visiting this city.
listener	in Home 	the listener in the combobox will listen to the value of the combobox, so whenever it change we will update the displayed information in the city card in Home interface.
Action Event	in AddCity 	this event will open a fileChooser will appear, If the user selects a picture, the word "selected" will appear and we can store the path of this photo with the city instance in the database.
Key Event	in AddCity	when the user clicks ENTER the process of adding a new city will begin we will validate the entered data if there is mismatch of the required types or if the user does not fill a required text fields, error message will appear. and when it is finish we will insert the city and then send the user object to Home so we maintain the information there and move to the Home interface.

Type of Action	the source of the event	event handling
Action Event	in AddCity Add	when the user clicks the button the process of adding a new city will begin, we will validate the entered data if there is mismatch of the required types or if the user does not fill a required text fields, error message will appear. and when it is finish we will insert the city and then send the user object to Home so we maintain the information there and move to the Home interface.
Action event	in AddCity Cancel	when the user clicks the button he /she will be returned to Home interface and we will send the user object to Home controller so we maintain the information there.
Action event	in Add Location Select File	this event will open a fileChooser will appear, If the user selects a picture, the word "selected" will appear and we can store the path of this photo with the location instance in the database.
Action event	in Add Location Add	when the user clicks the button the process of adding a new location will begin , we checks if the required data is present, if it does not exist, the message “Empty textfields” appears.then, It checks If the budget is entered with numbers, If not, the message “Enter budget in numbers” appears. and after the insert the new location we will send the user object to Home so we maintain the information there and move to Home interface.
Action event	in Add Location Cancel	when the user clicks the button he /she will be returned to Home interface and we will send the user object to Home controller so we maintain the information there and move to Home interface.
Mouse Event	in Show Details 	when the user clicks on the button, the user and the received city object information will be send to EditCity controller so that interface can display the information the user wants to edit.
Mouse Event	in Show Details 	when the user clicks on the button, the user,the received city object and the chosen location from the combobox information will be send to EditLocation controller so that interface can display the information the user wants to edit.

Type of Action	the source of the event	event handling
Action Event	in Show Details 	when the user clicks on this button, a TextInputDialog will open and we ask the user of the amount of money paid at this location so we can update the budget of the location in our database so later we can do some calculations to view it in BudgetReport interface.
Action Event	in Show Details 	when the user clicks the button he /she will be returned to Home interface and we will send the user object to Home so we maintain the information there and move to Home interface.
Listener	in Show Details 	the listener in the combobox will listen to the value of the combobox, so whenever it change we will update the displayed information in the location card in ShowDetails interface.
Action Event	in Show Details 	when clicked the link of the selected location from the combobox will be opened in browser
Action Event	in EditCity 	this event will open a fileChooser will appear, If the user selects a picture, the word "selected" will appear and we can update the path of this photo with the city instance in the database.
Action Event	in EditCity 	when the user clicks the button the process of updating a city will begin, we will validate the entered data if there is mismatch of the required types, error message will appear. and when it is finish we will update the user and city objects to ShowDetails controller so we maintain the information there and move to the ShowDetails interface.
Action Event	in EditCity 	when the user clicks the button he /she will be returned to ShowDetails interface and we will send the user and city objects to ShowDetails controller so we maintain the information there.

Type of Action	the source of the event	event handling
Action Event	in Edit Location 	this event will open a fileChooser will appear, If the user selects a picture, the word "selected" will appear and we can update the path of this photo with the location instance in the database.
Action Event	in Edit Location 	when the user clicks on this button,the process of updating a location begin , we checks If the budget is entered with numbers, If not, the message “Enter budget in numbers” appears. and after the update we will send the user and city objects to ShowDetails so we maintain the information there and move to ShowDetails interface.
Action Event	in Edit Location 	when the user clicks the button he /she will be returned to ShowDetails interface and we will send the user and city objects to ShowDetails controller so we maintain the information there.
Action Event	in Profile 	this event sends user information to the TodoList controller then drives the user to TodoList interface. when the TodoList controller gets the user information it will pull user tasks from the database and display the finished in Done listview and the not finished will be displayed in To Do listview.(To maintain the information)
Action Event	in Profile 	this event sends user information to the BudgetReport controller then drives the user to BudgetReport interface. when the BudgetReport controller get the user information it will pull user's cities from the database and display them in the cities combobox. (To maintain the information)
Action Event	in Profile 	this event sends user information to the Home controller again so we can maintain the information and does not disappear.
Action Event	in Profile 	this event sends user information to the Profile controller again so we can maintain the information and does not disappear.

Type of Action	the source of the event	event handling
Action Event	in Profile 	this event will send the user information to EditProfile controller so that interface can display user information so the can be edited.
Action Event	in Profile 	this event will return the user to the Animation interface.
Action Event	in EditProfile 	when the user clicks the button the process of update user is begin, we checked the user inputs so if there is any mismatch of the types error message appear. and we check for the password and confirm password , if they are not equal there is alert window will display to the user tell that there is some error. then we update the user in the database and send the user object to Profile controller to maintain the informations then we move back to Profile interface.
Action Event	in EditProfile 	when the user clicks the button he /she will be returned to Profile interface and we will send the user object to Profile controller so we maintain the information there.
Action Event	in TodoList 	this event sends user information to the TodoList controller again so we can maintain the information and does not disappear.
Action Event	in TodoList 	this event sends user information to the BudgetReport controller then drives the user to BudgetReport interface. when the BudgetReport controller get the user information it will pull user's cities from the database and display them in the cities combobox.
Action Event	in TodoList 	this event sends user information to the Home controller so we can maintain the information and does not disappear.
Action Event	in TodoList 	this event sends user information to the Profile controller then drives the user to Profile interface. when the Profile controller get the user information it will display the data in UI controls.

Type of Action	the source of the event	event handling
Action Event	in TodoList 	when the user clicks on this button, we will get the user selection from the listview then we will add it to the Done listview and remove it from ToDo listview. then we update the state of the task in the database.
Action Event	in TodoList 	when the user clicks on this button, we will get the user selection from the listview then we will add it to the ToDo listview and remove it from Done listview. then we update the state of the task in the database.
Action Event	in TodoList 	when the user click on this button ,we will clear all items in the 2 listviews and the tasks will be deleted from the database.
Action Event	in TodoList 	when the user clicks the button we will get the entered text and set it to the description attribute of a new TodoList object then we add it to the ToDo listview and insert the object in the database.
Action Event	in Budget Report 	this event sends user information to the TodoList controller then drives the user to TodoList interface. when the TodoList controller gets the user information it will pull user tasks from the database and display the finished in Done listview and the not finished will be displayed in To Do listview.(To maintain the information)
Action Event	in Budget Report 	this event sends user information to the BudgetReport controller so we can maintain the information and does not disappear.
Action Event	in Budget Report 	this event sends user information to the Home controller so we can maintain the information and does not disappear.
Action Event	in Budget Report 	this event sends user information to the Profile controller then drives the user to Profile interface. when the Profile controller get the user information it will display the data in UI controls.

C. Java Database Programming

Why we need the database and how we utilize it?

our application is like an organizer and planner so how we can organize and help the user without saving the data? that's why we need the database,to keep and retrieve the inserted data if the data is needed.
and because we divide our application into components to deal with each one alone to make it easier and maintainable.
the classes (Object models) in our application have an equivalent tables in the database. so when we deal with an object we need to interact only with its table in all CRUD operations.

the equivalent tables of the classes we have are 4 tables (User,City,Location and ToDoList).

the object attributes are the columns in the tables:

User table:

ID "PK"
name
email
gender
phone
password

ToDoList table:

ID "PK"
description
user "FK'
finished

Location table:

ID "PK"
name
city "FK'
hotel
visit date
visit time
budget
link
photo

City table:

ID "PK"
name
user "FK'
hotel
arrival date
return date
budget
members
photo

How we connect the object models to our relational model?

we used hibernate ORM framework to map our java classes with the corresponding tables in the database.
First, we have create the schema in our database as we designed.
Second, we have create the POJO classes which are 4 (User,City, Location and TODoList) and we use annotations to specify the mapping to the corresponding tables and columns.
Third, we made the hibernate configuration of the xml file along with creating a HibernateUtil class so the POJO classes can be linked to the database.

what are the relationships between our database and our GUI?

there are many interfaces in our application interact with the database. and we will explain each one.

Sign Up interface:

in this interface we have a form consist of number of text fields and labels. these text fields are linked to the User's table columns. so after the user fills the form and click on "sign up" button, the validation process of user's inputs start and when it is over then we will check that there is no other user in our database has the same email. we will create a new User object and it will have the entered values and the object will be saved in the database in user table.

Login interface:

in this interface we have a form consist of 2 text fields and 2 labels. these text fields are linked to the User's table columns. so after the user fills the form and click on "login" button, the validation process of user's inputs start and when it is over then we will check that there is a user has the entered email in our database then we retrieve the user's information.

Home interface:

in this interface we have a combobox contains user's cities.
and to fill the combobox with the cities we need to retrieve them from the database when the method (setCity) is called from any other interface.

Add city interface:

in this interface we have a form consist of number of text fields and labels. these text fields are linked to the City's table columns.
so after the user fills the form and click on "add" button the validation process of user's inputs start and when it is over we will create a new City object and it will have the entered values and the object will be saved in the database in city table.

Add location interface:

in this interface we have a form consist of number of text fields and labels. these text fields are linked to the Location's table columns.
so after the user fills the form and click on "add" button the validation process of user's inputs start and when it is over we will create a new Location object and it will have the entered values and the object will be saved in the database in location table.

show details interface:

when the user click on "show details" button in home interface, the process in retrieving data starts. because the labels in this interface shows details about the city and its locations.
these informations are retrieved from City and Location tables in the database.

Edit city interface:

in this interface we have a form consist of number of text fields and labels. these text fields are linked to the City's table columns.
the fields will be filled with a city information that the user wants to edit . so after the user modify a value in the text field and clicks "save" button, this city object will be updated in the database.

Edit location interface:

in this interface we have a form consist of number of text fields and labels. these text fields are linked to the Location's table columns.
the fields will be filled with a location information that the user wants to edit . so after the user modify a value in the text field and clicks "save" button, this location object will be updated in the database.

Budget Report interface:

in this interface we have a combobox contains user's cities.
and to fill the combobox with the cities we need to retrieve them from the database when the method (setCity) is called from any other interface.

Edit profile interface:

in this interface we have number of text fields and labels.
these text fields are linked to the User's table columns.
the fields will be filled with the user's information. so after the user modify a value in the text field and click "save" button, this user object will be updated in the database.

ToDo list interface:

in this interface we have 2 listviews that display the user's tasks.
these tasks are retrieved from TodoList table from the database when the user "login" in the application.

and there is "clear" button if the user clicks on it, it will delete all his/her tasks from the listviews and the database .

also when the user select a task from the listview and clicks on the "arrows" buttons the todoList object will be updated with the new finished status, which is the opposite of the current.

D. Model-View Controller

How we have implemented the MVC pattern?

We have used the MVC pattern by dividing the implementation phase into 3 separate classes: POJO classes (Model), interfaces (View) and actions handler classes (Controller).
and we will mention all of them.

Model (POJO classes)	View (interfaces)	**Controller (actions handler classes)**
User class City class Location class ToDoList class	Animation.FXML	AnimationController.java
	Welcome.FXML	WelcomeController.java
	SignUp.FXML	SignUpController.java
	Login.FXML	LoginController.java
	Home.FXML	HomeController.java
	AddCity.FXML	AddCityController.java
	AddLocation.FXML	AddLocationController.java
	ShowDetailes.FXML	ShowDetailesController.java
	EditCity.FXML	EditCityController.java
	EditLocation.FXML	EditLocationController.java
	Profile.FXML	ProfileController.java
	EditProfile.FXML	EditProfileController.java
	ToDoList.FXML	ToDoListController.java
	ReportBudget.FXML	ReportBudgetController.java

what are the tools that we have used to design the application?

we used an (allistrator) to design the images and the backgrounds.

we used the (scene builder) to implement the interfaces.

E. Extra

what are the extra work that we have done ? with explanation of why and how it is related to our application?

we have implement the all the extra requirements and they are:

1- File I/O :

*store only the cities data in external text file so maybe in the future we want to do some analysis about most visited cities and what is the average budget for X members,etc.

*we use file chooser to let the user upload a photo path.

2- Effects : we add a "shadow" effect on "Done" button in "Home" interface from the scene builder.

3-animation : first interface in our application is an animation scene.

4- Multimedia : we add an audio in the background of the application so when it is run the audio will be played.

(the source of animation ,audio and file chooser are included in the code file in the documentation) .

What we learnt:

What we have learned from the project?

we learned in this project how to make use of everything we studied even if it was from other course. we learned how to use the knowledge we got in practical way like programming this project and we got some experince in how the projects in real life be done, like MVC pattern that we tried to use.

What is the advantages you gained from designing this project?

it is never the first idea!. in this project we learned that one tiny idea can be modified many times because of other factors of course like difficulty in use or some ideas can't be implemented exactly in the way that we want.
so we need to keep looking, searching and learning.this project made this skil stronger in us.

Did we feel the teamwork was very useful in programming project? are we satisfied ?

of course working in a team much better than working alone, especially like this project which contain the frontend and backend. there are some of us better than others in design and in front-end part, and in contract there are some of us better in coding and programming, that is the good of having a team in such projects.
we are satisfied about our project, of course it can implemented better but in short period and with the massive pressure we came with something nice and functional, we need to improve it in the future!.

Did we feel that we will be able to work in a large scale project later?

absolutely, our idea is strong especially these days when the eyes looking about internal tourism and how we want something simple that help us during the journeys.
we hope that someday we see this project in application market and to be a succesful.

Project difficulties:

of course we faced some problems. in programming, we got dozen of null pointer errors with no idea why it is null, some of the team members faced problems with installing scene builder and some others fails in doing xml connection even though in the previous tests it was successful.
in work distribution, there was some difficulties with some team members in managing time.
in general we know that any project will face a problems but the key for all of them is the good communication.

Project Declaration:

We confirm that the work of this project was solely undertaken by ourselves and that no help was provided from other sources as those allowed. As well as we confirm that we completely aware of the violation consequences of the academic integrity.

Shahad malibari	ID441010641
Maarib al-sulimani	ID441005504
Mafaz basalamah	ID439002072
Manar alabdali	ID438007643
Razan al-refaey	ID441005995

The distribution of the work:

Manar Al-Abdli:

Discussion on ideas.
Working on prototype information.
*design all the scenes in illustrator
*implement Animation (fxml and java controller).
*implement Welcome (fxml and java controller).
*implement SignUp (fxml and java controller).
*implement Login scene (fxml and java controller).
*implement Home scene (fxml and java controller).

Maarib Al-Sulimani:

Discussion on ideas.
Working on prototype information.
*implement Profile scene (fxml and java controller).
*implement EditProfile scene (fxml and java controller).
*implement ToDoList scene (fxml and java controller).
*transfer data from Profile controller to EditProfile controller.
*Fix an error in sending data parts in login and SignUp controllers.

Mafaz Ba Salamah:

Discussion on ideas.
Working on prototype information.
*implement AddLocation scene (fxml and java controller).
*implement EditCity scene (fxml and java controller).
*implement EditLocation scene (fxml and java controller).

Shahad Malibari:

*Discussion on ideas.
*Working on prototype information.
*implement AddCity scene (fxml and java controller).
*implement showDetails scene (fxml and java controller).
*implement budgetReport scene (fxml and java controller).
*fixing some errors during the integrating.

Razan Al-Refaey: the leader

*Discussion on ideas.
*Working on prototype information.
*Create the database.
*create the POJO classes.
*hibernate configurations.
*integrate the code.
*fixing errors during the integrating.
*edit some scenes in scene builder.
*Utility class (user defined methods deals with DB).
*adding the audio.
*adding the effect.
*I/O files in AddCity controller.
* documentation the entire project .
*the report.
*keep track of all team members.