Travel Company

Sarah Hopkins

Contents

Proposal

A travel company provides travel services to a number of different cities in Europe. They require an application to produce an invoice for a customer’s travel costs. The program would require a user’s input for personal details and travel options. The program should be able to correctly process and output relevant data in an invoice. It must be able to work on a customer’s personal computer.

Methodology

This project will use Rapid Application Development.

Functionality

Input:

The program will require a user’s input for:

- First name.
- Last name.
- House number.
- Postcode.
- Travel destination.
- Number of nights staying at the travel destination.
- If travel insurance is required.

Process:

How the input data will be processed:

- A customer ID is generated from the input surname, followed by house number and postcode.
- The total accommodation cost will be calculated by multiplying the base accommodation cost and the number of nights staying at the destination.
- The total insurance cost will be calculated by multiplying the base insurance cost and the number of nights staying at the destination.
- A total price will be calculated by adding the cost of travel to a destination, the accommodation total and the insurance total.

Output:

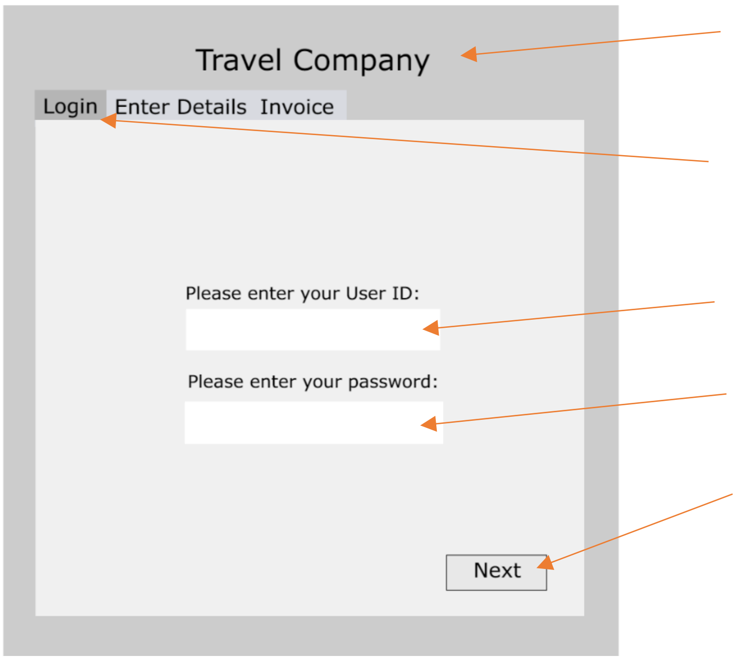
- A graphical user interface that a customer can interact with.
- An interface provides the ability for users to input data.

Output of input and processed data:

- An invoice would be visually output containing the: user ID, customer ID, destination city, and number of nights staying at destination, travel cost, total accommodation cost, total insurance cost and grand total.

UI design

Tab 1 – Login



Company name.

Login tab is a different colour to the other tabs (signifying to the user that is the tab they are on).

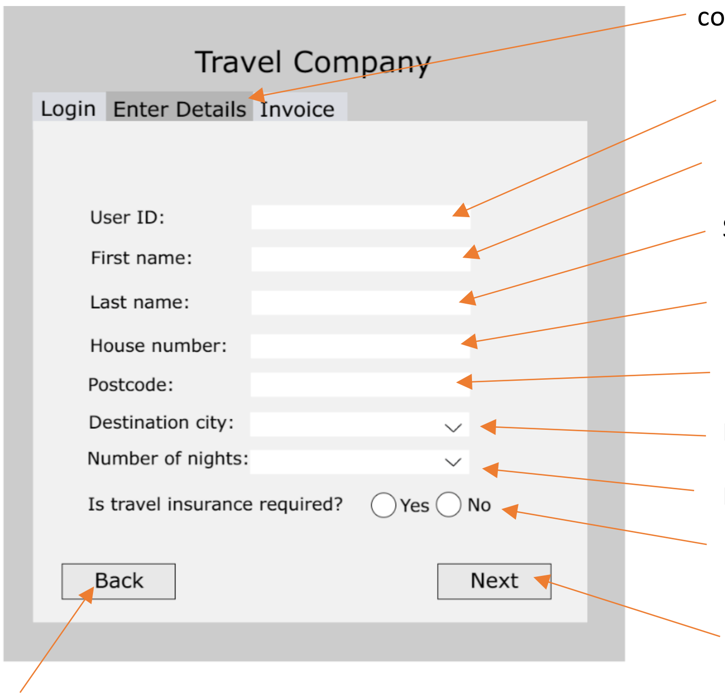
User input for variable userID.

User input for variable password.

The next button displays tab 2 (Enter Details), if the userID matches the password.

Tab 2 – Enter details

Enter details tab is a different colour to the other tabs.



Output of customer's userID (entered on tab 1)

String input for the variable firstname

String input for the variable lastname

Integer input for the variable houseNumber

String input for the variable postcode

Dropdown option for the variable destination

Dropdown option for the variable nights

A boolean grouped radio button input for the variable insurance.

The next button displays tab 2 (Invoice) if the user input passes validation.

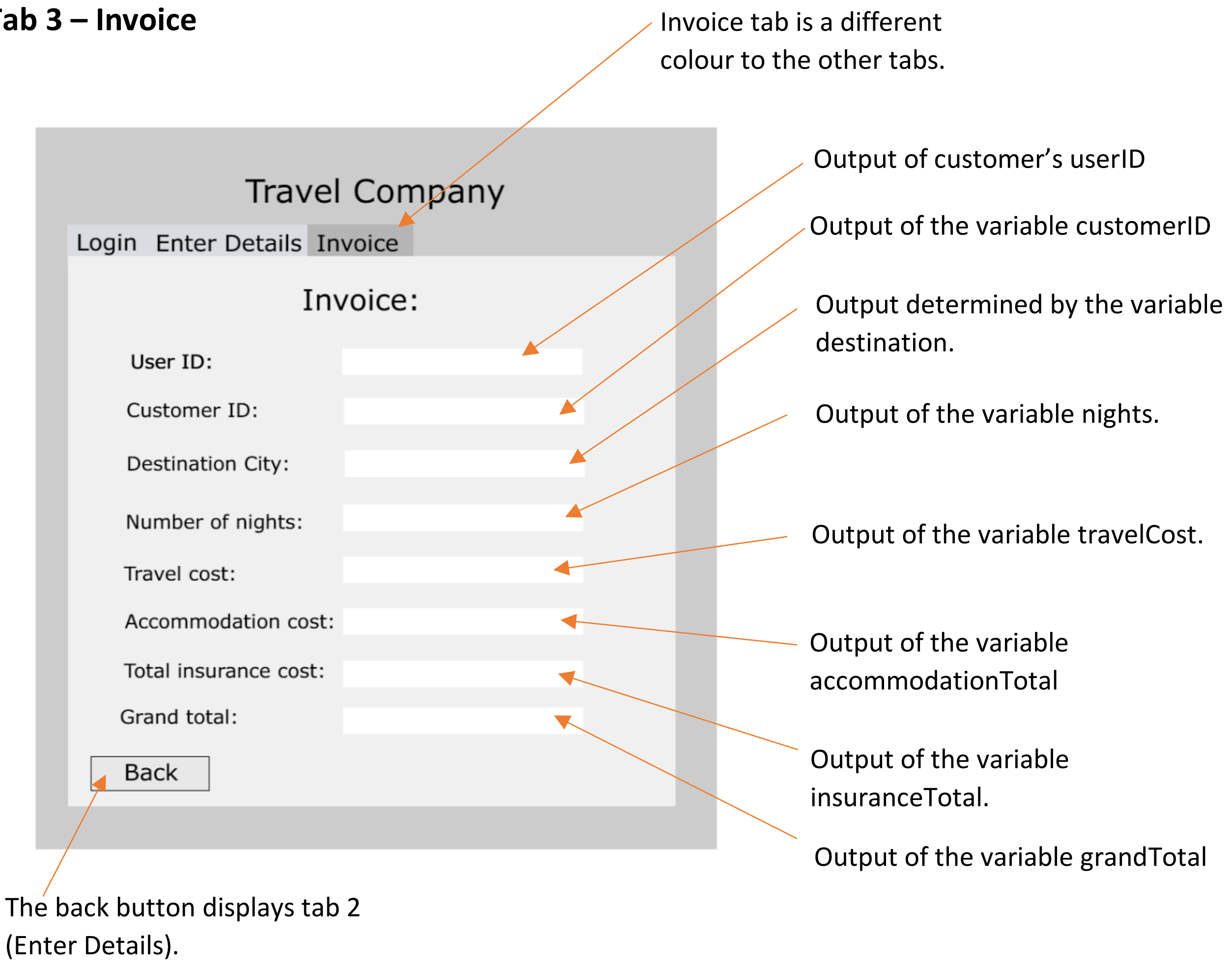
The back button displays tab 1

The input for the first name and last name input boxes would be limited to 32 letters.

House number will be limited to 4 integers.

The postcode input will be limited to 9 numbers and letters.

Tab 3 – Invoice


Invoice tab is a different colour to the other tabs.
Travel Company
Login
Enter Details
Invoice
Invoice:
User ID:
Customer ID:
Destination City:
Number of nights:
Travel cost:
Accommodation cost:
Total insurance cost:
Grand total:
Back
Output of customer's userID
Output of the variable customerID
Output determined by the variable destination.
Output of the variable nights.
Output of the variable travelCost.
Output of the variable accommodationTotal
Output of the variable insuranceTotal.
Output of the variable grandTotal
The back button displays tab 2 (Enter Details).

Data dictionary

Variable Name	Data Type	Description
userID	String	Predefined user ID.
password	String	Predefined user password (associated with the corresponding userID).
userIDInput	String	User input for their userID.
passwordInput	String	User input for their password.
firstName	String	User input for their first name.
lastName	String	User input for their last name.
houseNumber	Integer	User input for their house number.
postcode	String	User input for their postcode.
nights	Integer	Combo box options ranging from 1 to 14 . Input for the number of nights a customer requires.

destination	String	Combo box options for a user's desired travel destination. Input for the customer's chosen destination. Used to determine the travelCost value.
insurance	boolean	User choice between a yes or no option for travel insurance.
customerID	String	Concatenation of the variables: lastName, houseNumber and postcode.
firstNameTemp	boolean	The outcome of validation on the variable firstName, determines if this variable is set as true or false. This variable is then used for validation to either prevent or allow the user to progress through the program.
lastNameTemp	boolean	The outcome of validation on the variable lastName, determines if this variable is set as true or false. This variable is then used for validation to either prevent or allow the user to progress through the program.
houseNumberTemp	boolean	The outcome of validation on the variable houseNumber, determines if this variable is set as true or false. This variable is then used for validation to either prevent or allow the user to progress through the program.
postcodeTemp	boolean	The outcome of validation on the variable postcode, determines if this variable is set as true or false. This variable is then used for validation to either prevent or allow the user to progress through the program.
destinationTemp	boolean	The outcome of validation on the variable destination, determines if this variable is set as true or false. This variable is then used for validation to either prevent or allow the user to progress through the program.
travelCost	Integer	Used to store the travel costs of a specific destination (for example, if Paris is selected the variable will be set as 8000).
accommodation	Integer	Used to store the base accommodation cost.
accommodationCost	Integer	Used to store total accommodation cost. Calculated by multiplying accommodation and nights variables.
insuranceCost	Integer	Used to store the base insurance cost
insuranceTotal	Integer	Used to store total insurance cost. Calculated by multiplying nights and insuranceCost.
grandTotal	Integer	Used to store the final total of all prices. Calculated by adding the variables travelCost, insuranceTotal and accommodationTotal.

Pseudocode

Login (tab1):

```
If ((userIDInput == userID) AND (passwordInput == password)){
    Display tab2
}
```

Validating personal and travel information (tab2):

```
If ((length of personal detail > 32) OR (length of personal detail == 0)){
    Display error message
    validationVariable = false
} else{
    validationVariable = true
}
```

Personal variables will include: first name, last name, house number, postcode.

```
If (radio button is selected){
    insurance = true
} else{
    insurance = false
}
```

```
Nights = option of nights combo box
```

```
If (destination Combo box = “Please select a city”){
    validationDestination = false
} else{
    validationDestination = true
    Destination = option of destination combo box
}
```

Checking all personal and travel details are true (tab2):

```
If((validationFirstName AND validationLastName AND validationHouseNumber AND
validationPostcode AND validationDestination) == TRUE){
    Display tab3
}
```

Calculating finances for the invoice (tab2):

```
Total insurance = number of nights * insurance cost
Grand total = travel cost + insurance total + accommodation total
```

Invoice (tab3):

```
Display user ID
Display customer ID
Display destination city
Display number of nights
Display travel cost
Display accommodation total
Display insurance total
Display grand total
```

Next buttons:

```
If (tabbed pane index + 1 < tabbed pane highest index){
    Tabbed pane index = (tabbed pane index + 1)
}
```

Back buttons:

```
If (tabbed pane index > 0 ){
    Tabbed pane index = (tabbed pane index - 1)
}
```

Testing

Test Number	Test Description	Pass, Fail or Outcome.	Notes	Date Developed	Date Tested
1	Tab 1 - validation test	Failed	UserID: Successfully set input to variable. Failed to prevent the user progressing without inputting a UserID.	22/12/2020	07/01/2020
2	Tab 2 - input/ validation test	Failed	No data input: Include error messages. Full name input: Successfully sets variable from input. Create a length limit for the variable. Failed by allowing user to progress without inputting data. House Number input: Successfully sets variable. Failed number only input. Allowed user to progress without inputting data. Postcode input: Successfully sets variable. Create a length limit for the variable (9 characters). Failed by allowing user to progress	22/12/2020	07/01/2020

			without inputting data. City input: Failed - missing visually and not coded. Create a dropdown menu. Nights input: Successfully sets variable. Insurance input: Failed - not coded.		
3	Tab Selection test	Failed	User could access other tabs by clicking on them (the user should only be allowed to progress with next and back buttons).	22/12/2020	07/01/2021
4	GUI test (final)	Failed	Window needs to be fixed to a size/ have a minimum set size.	22/12/2021	07/01/2021
5	Tab selection test	Pass	Tabs could not be directly selected accessed by users. The option was disabled in the tabbed pane properties.	05/01/2021	05/01/2021
6	Tab 2 - validation test	Failed	Introduced validation requiring a length less than 32 characters and a length that is not equal to 0. This is applied to the variables firstName and lastName. (pass)	15/01/2021	17/01/2021

			Introduced validation requiring a length lesser than or equal to 5 and a length that is not equal to 0. This is applied to the variables houseNumber. (pass) Introduced validation requiring a length of less than or equal to 10 and a length that is not equal to 0. This is applied to the variable to postcode. (pass) A combo box was implemented for the variable nights. Successfully sets the variable. A check is performed to ensure all input boxes have an input. Insurance input outputs a Null Pointer error message when no option is selected. A NumberFormatException is thrown when no input is given for houseNumber. The first name and last name inputs allow integers to be input.		
7	Tab 2 - validation test	Pass	The option “No” is preselected preventing a Null	20/01/2021	20/01/2021

			Pointer error (insurance option).		
8	Tab 1 - validation test (final)	Pass	User could not progress unless input met validation requirements. User could not progress unless the UserID and password were correct.	20/01/2021	21/01/2021
9	Tab 2 - validation	Failed	Introduced error messages. If validation fails for any reason, they will appear to the right of the related variable input box. The first name and last name row will display "Max letters 32". The house number row will display "Max numbers 4". The postcode row will display "Max characters 9". House number allowed non integer input. House number allowed progression with no input. Postcode allowed special characters. (fail) First name and last name allowed special	02/02/2021	02/02/2021

			character to be input. (fail)		
10	Tab 2 - validation test (final)	Pass	A try catch statement was used to catch the NumberFormatException for houseNumber. The error message “Max numbers 4” is output if the input is not an integer and the variable houseNumberTemp is set to false (used to check all input options have an input). House number prevented progression if there was no input. Postcode prevented the use of special characters. The input is checked with regular expression (regex) to be only a number. First name and last name inputs do not allow integers to be input. They also do not allow do not allow special characters. The input is checked with regular expression (regex) to be a group of letters. Error messages are correctly displayed when an input is incorrect or not present.	02/02/2021	02/02/2021

11	Tab 3 - validation test (final)	Pass	Customer ID outputs correctly. Destination city outputs correctly. Number of nights outputs correctly. Travel cost outputs correctly. Accommodation cost outputs correctly. Total insurance cost outputs correctly. Grand total outputs correctly.	15/01/2021	02/02/2021
12	GUI test (final)	Pass	Window size is fixed and cannot be resized. All jLabels and input boxes are correctly displayed.	15/01/2021	02/02/2021

Evaluation

For functionality, the program would be fit for purpose. It fulfils the outlined requirements from the travel company. The program allows for the user to input their: userID, password, first and last names, house number, postcode destination city, number of nights staying at the destination and if insurance is required. As specified in the design, the program only allows progression through the next buttons, if all the inputs are valid. The program successfully outputs the invoice, containing all elements required by the travel company; this would include: userID, customerID, destination city, number of nights, travel cost, accommodation cost, total insurance cost and the grand total. The customerID correctly generates based on the requirement of concatenating the last name house number and postcode (in that order). It calculates the total costs (total travel costs, total accommodation costs, total insurance, and the grand total) correctly.

Graphically, the application fulfils the design specification. There are no changes made to the design in the final program.

To improve the program, more specific error messages could be used. For example, if the user uses an integer for a string input, the error message would tell the user it requires a string; if the user does not input anything into and input box, an error message would display telling the user it is a required field. Another improvement could be including a way to allow the user to save the invoice into a file.