

FLIGHT BOOKING SYSTEM

Presented by: Shivam gupta

Registration number: 189301018

CSe-a

## INTRODUCTION OF THE PROJECT

## EVERYONE USES AIRLINES SUPPORT AND WANTS TO RISE THE VEHICLE WHICH CAN LITERALLY TAKE THEM 40,000 ft. up the sky and see the world from that height.

## THE FOLLOWING PROJECT CONTAINS WHAT WOULD A FLIGHT BOOKING SYSTEM WOULD LOOK LIKE, IF WRITTEN WITH JUST THE HELP OF CORE JAVA LANGUAGE.

## THE OUTPUT OF THE CODE MIGHT NOT BE USER FRIENDLY OR HELP IN MAKING THE BOOKING SYSTEM ANY BETTER BUT AT THE LIMITED RESOURCE AND KNOWLEDGE IT REALLY SHOWS THE PEOPLE WHAT HAPPENS BACKEND.

## THE GIVEN JAVA PROJECT IS ABOUT A NEAR TYPE FLIGHT BOOKING SYSTEM THAT WOULD BE PRESENT IN A WEBSITE.

## IT CONTAINS OPTIONS LIKE BOOKING A FLIGHT, CANCELLING A FLIGHT WITH A BOOKING ID THAT WOULD BE UNIQUE FOR A PURCAHSE. IT ALSO CONTAINS OPTION TO PRINT A BILL WITH CONVENIENT PRICES, WITH ALSO AN OPTION TO CHECK THE FLIGHT PURCHASE HISTORY

## IT ALL COULD BE DONE ONLY WHEN THE CUSTOMER IS A REGISTERED USER IN THE WEBSITE. IF HE ISN’T ONE THEN THERE IS AN OPTION TO SIGNUP IN THE CODE. FOR WHICH HE WILL BE AUTHENTICATED WITH A PASSWORD AND CHECKED IF IT IS CORRECT OR NOT.

IT TAKES HELP OF THE FILE HANDLING AND EXCEPTION TO THE GREATEST EXTENT. AND ALSO CONTAINS MANY OTHER BASIC YET CRUCIAL CONCEPTS AND FEATURES OF THE POWERFUL LANGUAGE THAT IS JAVA AT IT’S BEST.

THE FOLLOWING CODE CONTAINS SEVERAL CLASSES EACH WITH A DIFFERENT AND UNIQUE ROLE IN the PROJECT. NO PIECE OF CODE PLAYS A ROLE IN this project is useless and if removed any one of it could disrupt the entire project.

## CURRENT WORK – WHAT HAVE WE DONE TILL NOW?

## THE PROJECT STARTS BY ASKING the USER WHETHER THE HE/SHE IS AN EXISTING USER OF THE WEBSITE. THEN THE ASK THE USER TO ENTER A BINARY RESPONSE. IF HE IS NOT ONE THEN HE SIGNUP MENU OPTIONS WILL OPEN AND WILL ASK THE NECESSARY QUESTIONS.

## IF THE USER IS NOT AN EXISTING USER THEN IN THE SIGNUP MENU IT ASKS FOR AN USERNAME AND A SECURE PASSWORD. THEN THE CODE CHECKS IF THE GIVEN USERNAME ALREADY EXISTS. IF IT DOES THEN IT GIVES AN ERROR AND HE HAD TO DO THE USER ALL OVER AGAIN.

## IF THE USER IS ALREADY AN EXISTIONG USER AND HIS CREDENTIALS MATCH TO tHAT OF the DATABASE THEN HE IS SENT TO THE BOOKING TICKETS MENU.

## THERE HE IS ASKED OPTIONS LIKE BOOKING OF A FLIGHT, CANCELATIONS OF A FLIGHT, TO SHOW THE BILL, OR TO CHECK THE PURCHASE HISTORY.

## SOFTWARE REQIUREMENTS

## TECHNOLOGIES: CORE JAVA

## PLATFORM: JAVA RUNTIME ENVIRONMENT(JRE) 1.6 OR ABOVE

## OPERATING SYSTEM: WINDOWS 10/8/7/XP, LINUX

## OTHER TOOLS: TEXT EDITOR(NOTEPAD)

## EXPLAINATION

## BOOKING OF A FLIGHT– 1->HERE THE USER IS ASKED FOR BOARDING PLACE AND THE DESTINATION THEN THE FLIGHT DETAILS OF THOSE DATA SHOWS UP AND THE USER CHOOSE THE FLIGHT THROUGH THEIR TIME. 2->then after asking the personal details of the passengers the ticket is confirmed and updated in flight booking history.

## cancelation of a flight – each flight purchase a unique BOOKING ID TO TRACK THE TICKETS the DATABASE. IT ALSO HELPS IN CANCELING THE TICKET. THE USER AFTER SIGNED IN ASKED FOR THE BOOKING ID AND THEN IN THE FLIGHT PURCHASE HISTORY IT IS DELETING AND THAT BOOKING ID IS NEVER USED AGAIN TO AVOID CONFUSION WITH MANY CUSTOMERS.

## TO SHOW THE BILL – THE BILL OF THE RECENT PURCHASES IS PRINTED WITH TOTAL PRICE THE CUSTOMER NEEDS TO PAY.

## THE PUCHASE HISTORY – THE PURCHASE HISTORY OF THE SIGNED IN USER IS DISPLAYED (EXCEPT THAT OF THE ONES THAT ARE DELETED).

* FOR ALL THIS YOU HAVE TO MEMBER FIRST OF THE PROGRAM. BY THIS YOU SHOULD HAVE A TEXT FILE WITH YOUR USERNAME IN IT AND ITS PASSWORD AND IN A SEPARATE FILE which CONATINS ALL THE USERNAME. EVERY TEXT FILE IS MADE IN THE SIGN UP PROCESS (A PERSON HAVE TO SIGN UP FIRST TO BE A MEMBER).
* WHEN THE BOOKING IS DONE THEN IN THE USER\_NAME.TXT IT IS AUTOMATICALLY IS ADDED WHICH LATER WILL HELP IN THE DELETION OF THE TICKET. AND TO WATCH THE FLIGHT HISTORY. FILE HANDLING PLAYS AN IMPORTANT PART IN THIS PROJECT.
* WHEN A FILE IS DELETED it is deleted from the text file too. which in turn would not show the deleted tickets in the flight purchase history.

## concepts used in the project

**INTERFACE:**

IN THIS PROJECT ABSTRACT CLASSES ARE USED TO SAVE THE USERNAME AND PASSWORD OF THE CUSTOMER. AS IN AN ABSTARCT CLASS ALL THE VARIABLES.

**ACCESS** **MODIFIERS** :

IN JAVA THERE ARE DIFFERENT ACCESS MODIFIERS THAT DETERMINE THEIR ACCESSIBILITY TO OTHER CLASSES OR METHODS. SUCH AS: PUBLIC, STATIC. THESE HERE ARE USED FOR THE SAME. THEY DISRUPT THE REACHABILITY OF THE DATA MEMBERS WHICH IN THE MAKES THE CODE MORE REALISTIC. FOR EXAMPLE FOR THE BOOKING ID IT IS SET TO STATIC SO AS IT CAN NEVER BE REINITIALISED AGAIN OR CAN NEVER BE SAME FOR TWO PERSONS.

**INHERITANCE:**

## INHERITANCE IS ANOTHER TOOL USED HERE TO ACCESS different METHODS AND VARIABLES SO AS TO NOT REPEAT OUR CODE UNNECESSARILY. HERE INHERITENCE IS USED TO CONNECT SIGNIN AND SIGNUP TO SEE WHETHER THE CUSTOMER ALREADY EXISTS WITH THE SAME USERNAME.

**FILE HANDLING:**

FILE HANDLING PLAYS AN IMPORTANT PART IN THIS PROJECT ALMOST EVERYTHING IS DONE THROUGH FILE HANDLING. TO STORE CONTENT, TO CHECK FOR PASSWORD AND USERNAME, TO DELETE FOR SOME. AS BACKEND PROGRAMMING IS AVAILABLE TO US IT IS THE BEST WAY TO THESE TASKS WITH OUR LIMITED RESOURCES.

**INPUT OUTPUT STREAM:**

INPUT AND OUTPUT STREAM IS USED HEAVILY IN THIS PROJECT. IT IS USED TO READ THE USERNAME AND PASSWORD DURING SIGNIN AND DURING PASSWORD CHECKING. IT IS ALSO USED TO ADD FLIGHT DETAILS TO THE USERNAME FILE AND DELETE IT DURING CANCELATIONS.

## **EXCEPTION HANDLING:**

EXCEPTIONS ARE HANDLED DURING THIS PROCESS IF ANY OCCUR DUE TO CUSTOMERS WRONG INPUT OR THE FILE IS NOT FOUND IN THE PROCESS. EXCEPTION HANDLING IS VERY MUCH USEFUL AS LETS THE CODE RUNNING WITHOUT DISRUPTING THE ENTIRE CODE AND PREVENTS US FROM DOING THE WHOLE PROCESS AGAIN.

**CUSTOMS EXCEPTIONS:**

CUSTOMS EXCEPTIONS ARE MADE IN THE CODE WHENEVER THE USER ENTERS WRONG INPUT OR WHEN THE PASSWORD IS INCORRECTED DURING THE SIGN-IN. EXCEPTIONS NAMED BookingException DOES THIS JOB AND HELPS IN DOING SO.

**CONTROL STATEMENTS:**

CONTROL STATEMENTS ARE USED IN THE CODE TO GIVE THE COMPILER WHAT TO DO, WHEN TO DO IT. IF-ELSE, FOR LOOP, DO WHILE, SWITCH CASES ARE USED.

**CONSTRUCTOR:**

CLASSES LIKE SIGNIN, SIGNUP AND FLIGHT USES CONSTRUCTOR CALLING TO DEFINE THE PARAMETERS THAT ARE TO BE USED IN THE RESPECTIVE CLASSES. It constructs the values at the time of object creation. It is not necessary to write a constructor for a class. It is because java compiler creates a default constructor if your class doesn't have any.

**JAVA STRING:**

STRING CLASS IS USED EXCESSIVELY. ALMOST EVERYTHING IS USED TO STORE THE CONTENTS IN THE STRING CLASS. THE USERNAME, PASSWORD, FLIGHT BOOKING DETAILS, ETC.

**WRAPPER CLASSES:**

The **wrapper class in Java** provides the mechanism *to convert primitive into object and object into primitive*. It is used to store variables like ok, etc. as objects of wrapper classes.

**MULTI-THREADING:**

**Multithreading in java** is a process of executing multiple threads simultaneously. However, we use multithreading than multiprocessing because threads use a shared memory area. They don't allocate separate memory area so saves memory, and context-switching between the threads takes less time than process.

APPENDIX