IQRA UNIVERSITY IU

WAJIHA JAWWAD ALI

TRAIN RESERVATION SYSTEM DOCUMENT

ID: 52023

PROPOSED TO: DR. ATIYA MASOOD

TRAIN RESERVATION SYSTEM

PHASE – I:

PROJECT DESCRIPTION:

The purpose of this project is to design and implement desktop application that allows users to book ticket for train. The purpose of this source is to describe the railway reservation system which provides the train timing details, reservation, billing and cancellation on various types of reservation, namely such as the possibility of giving feedbacks, as well as the opportunity to explore details of train provided. My project is based on Train Reservation System. This program has capability for the reservation of the passengers. Furthermore, it will allow the administration department to access the program and they can make certain changes.

PROJECT SCOPE:

Train reservation system is desktop application written in Java. The device calls for very low gadget resources and the gadget will work in nearly all configurations. Basically, there are some main tasks which are going to finish within the scope. This program will enable the user to see the details of available train of both business class and economy class, so they can reserve train from the available options. During reservation the program will ask the user about their destination and it will then show the train details which is destined to user’s required location. Afterwards the program will ask the user if they want to make reservation or not. If yes then it will ask the user on which name, they want to make the reservations on. After that the program will ask the number of seats they want to reserve. Once all of the required details are fed to program it will then generate the ticket which will have the following details of passenger’s name, seats number, unique train id, price as per the number of seats as well as departure time and day.

For administration, the program will ask the employee its user name and password which is unique for every employee. After entering correct ID and password the employee will be able to access the program. The employee will be able to add, delete or edit the details of any train they want. They will be able to see the details of all the trains or a specific train and also, they will be able to see the feedbacks given by the passengers.

REQUIREMENTS:

Below are mentioned the requirements according to the client which are essential for the successful software application and the software house has deemed them feasible to accomplish. After few back and forth discussion sessions with the client about what is practical in the time frame and technical limitations, following list are the results of said agreement along with their explanations.

Functional Requirements:

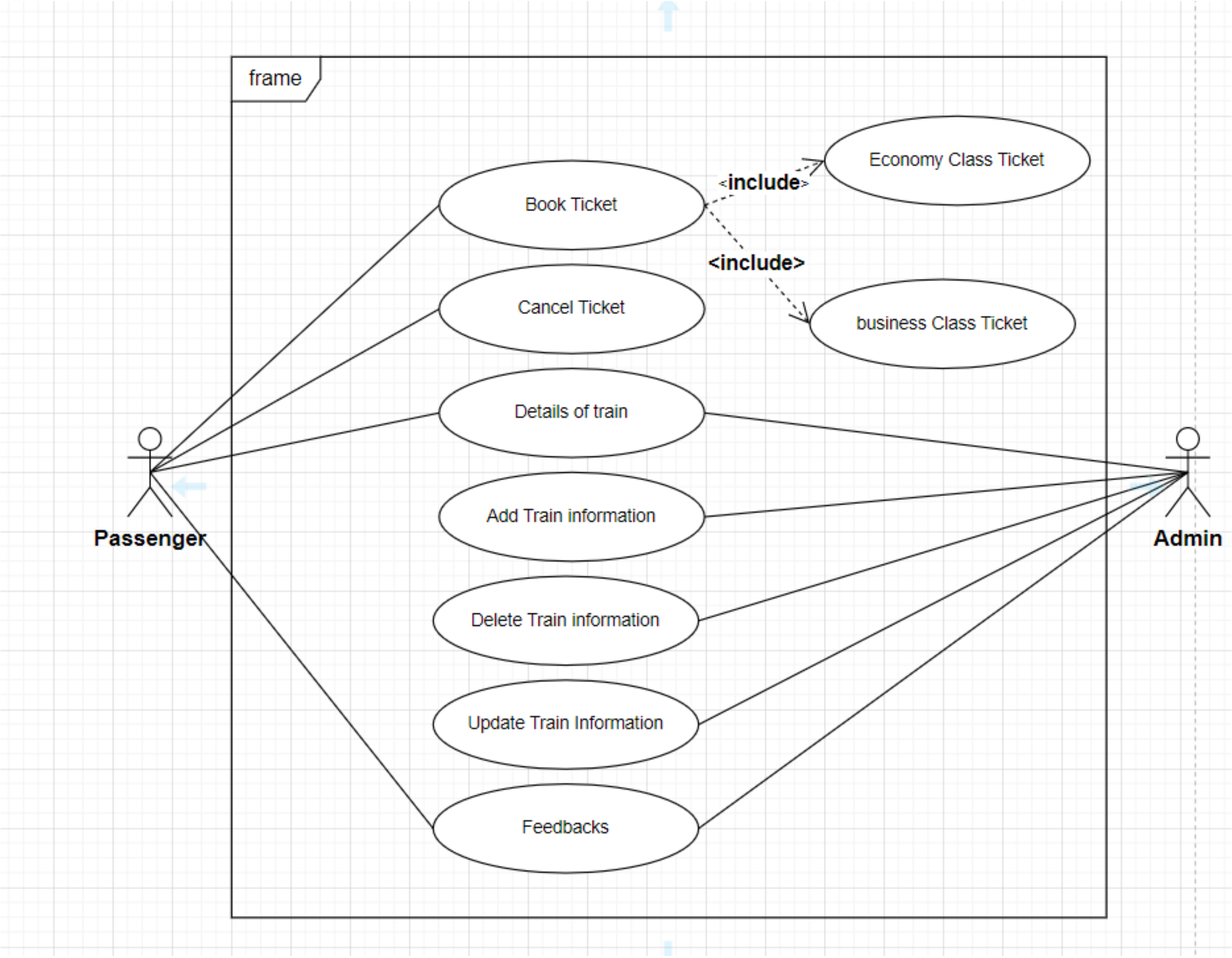
The software application must

- Have separate Customer and Admin Panels. Admin Panel will require login confirmation.
- User will be given two options (1. Passenger and 2. Admin).
- User will be able to see the details of the available trains.
- User will be able to make reservations.
- It will generate the tickets for passenger with required details (i.e. name, seat number, id, price, time and day).
- Admin will be able to access using their unique id and passcode and can make certain changes.
- Admin will be able to add, delete, and edit the information of required trains.
- Admin can see details of all the trains available at once as well as of the specific train.
- User can give their feedbacks that can be seen by admin employees.

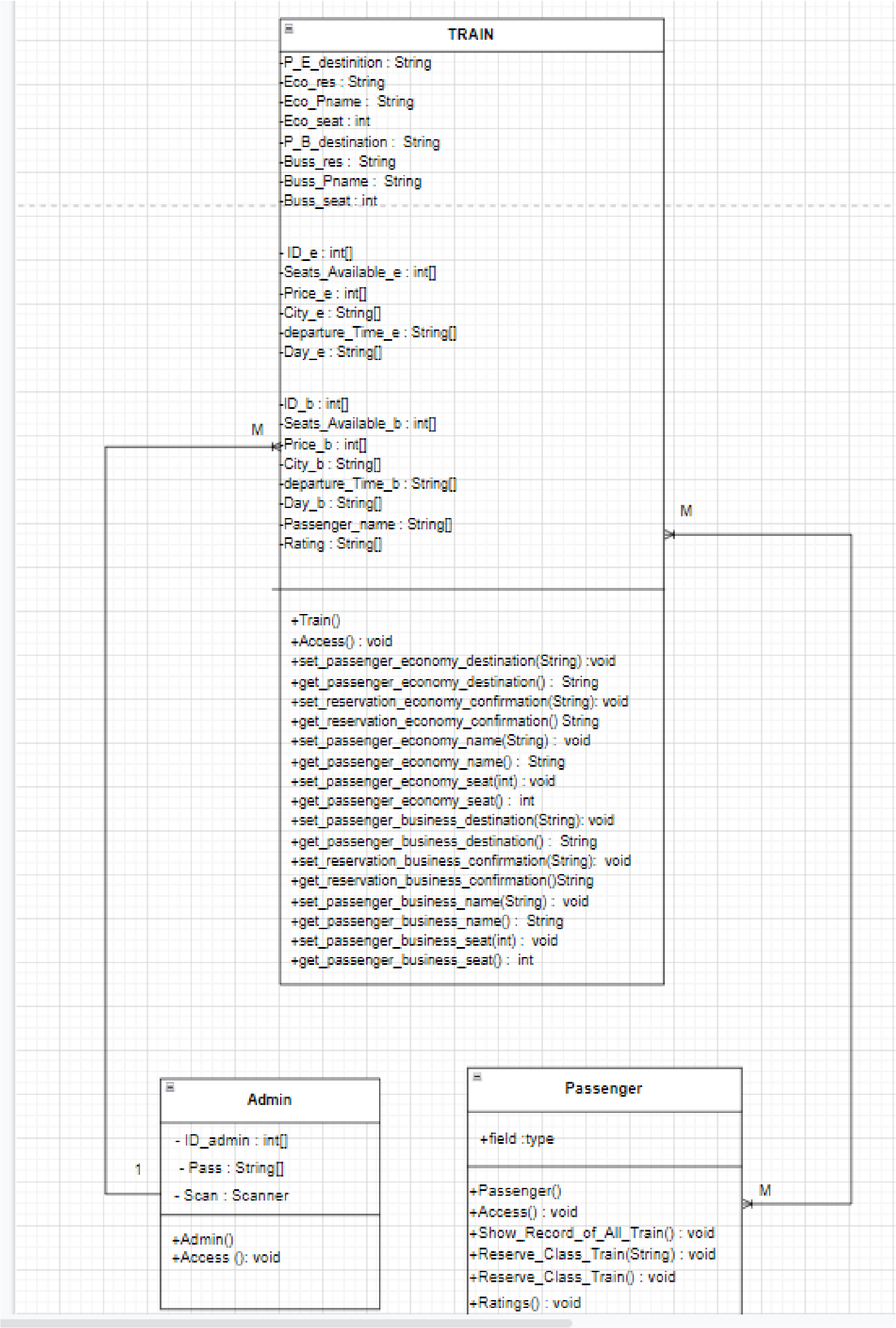
Process Model:

A software-process-model is a brief representation of how a software process works. The spiral model is a very beneficial and efficient model. It is the combination of the prototyping model and the waterfall model. The spiral model is mostly used for large, Risky, expensive, and complicated projects as best suits for railway reservation system. In this process, User requirements can be changed at later phases so it is flexible as well. In future, we can change the requirements by use Graphical User Interface and data base for this program.

USE CASE DIAGRAM:



CLASS DIAGRAM:


TRAIN
-P_E_destinition : String
-Eco_res : String
-Eco_Pname : String
-Eco_seat : int
-P_B_destination : String
-Buss_res : String
-Buss_Pname : String
-Buss_seat : int
- ID_e : int[]
-Seats_Available_e : int[]
-Price_e : int[]
-City_e : String[]
-departure_Time_e : String[]
-Day_e : String[]
-ID_b : int[]
-Seats_Available_b : int[]
-Price_b : int[]
-City_b : String[]
-departure_Time_b : String[]
-Day_b : String[]
-Passenger_name : String[]
-Rating : String[]
+Train()
+Access() : void
+set_passenger_economy_destination(String) :void
+get_passenger_economy_destination() : String
+set_reservation_economy_confirmation(String): void
+get_reservation_economy_confirmation() String
+set_passenger_economy_name(String) : void
+get_passenger_economy_name() : String
+set_passenger_economy_seat(int) : void
+get_passenger_economy_seat() : int
+set_passenger_business_destination(String): void
+get_passenger_business_destination() : String
+set_reservation_business_confirmation(String): void
+get_reservation_business_confirmation()String
+set_passenger_business_name(String) : void
+get_passenger_business_name() : String
+set_passenger_business_seat(int) : void
+get_passenger_business_seat() : int
M
M
Admin
- ID_admin : int[]
- Pass : String[]
- Scan : Scanner
+Admin()
+Access (): void
1
Passenger
+field :type
+Passenger()
+Access() : void
+Show_Record_of_All_Train() : void
+Reserve_Class_Train(String) : void
+Reserve_Class_Train() : void
+Ratings() : void
M

COCOMO MODEL:

Applying cocomo model,

Find **∑Fi:**

1. No Data base in our system, no influence = 0
2. There will be significant required communication of data, significant = 4
3. There will be some distributed function so Moderate = 2
4. No critical performance in it, significant = 4
5. the system does not run in an existing, heavily utilized operational environment, No influence = 0
6. The System does not require online entry, No influence = 0
7. The data input entry will require the input transaction to be built over a single screen, it's Average = 3
8. The master files will not be uploaded or updated online, No influence = 0
9. Input output enquires are average so Average = 3
10. internal process is Average = 3
11. code will not design for reusable no influence = 0
12. The conversion and installation is not included in the design, No influence = 0
13. The system is not designed for multiple installations in different organizations, no influence= 0
14. Usability is important Significant = 4

∑Fi = 23

For UFP:

UFP = (Fu * Wu) + (Fc * Wc)

UFP = (9 * 6) + (3 * 11)

UFP = 87

For CAF:

CAF = 0.65 + (0.01 * ΣFi)

$$CAF = 0.65 + (0.01 * 23)$$

$$CAF = 0.88$$

For FP:

$$FP = UFP * CAF$$

$$FP = 87 * 0.88$$

$$FP = 76.56$$

For LOC:

$$LOC = FP * 60$$

$$LOC = 76.56 * 60$$

$$LOC = 4593.6$$

For KLOC:

$$KLOC = LOC/1000$$

$$KLOC = 4593.6/1000$$

$$KLOC = 4.5936$$

KLOC value size lies in organic.

For Effort:

$$E = a\ (KLOC)^{b}$$

$$E= 2.4\ (4.5936)^{1.05}$$

$$E=11.89$$

For Duration:

$$D = c\ (\text{Effort})^{d}$$

$$D = 2.5\ (11.89)^{0.38}$$

$$D = 6.40 \approx 6 \text{ months}$$

For Staff:

$$\text{Staff} = \frac{E}{D}$$

$$\text{Staff} = 11.89\ /6.40$$

$$\text{Staff} = 1.85 \approx 2 \text{ person}$$