JADE Final Project

Assessment Name:	JADE Project
Weight:	30% (15% Database, 15% Interface)
Points:	100 possible points
Due:	07 Nov. 2017

1. "Flight Booking"

Create an application in JADE for a travel store where guests can view flights, passengers can check their tickets and managers can purchase tickets. The administrator will have permission for all the functions listed in the requirements.

Requirements

1. Model-View separation (2 schemas: database and interface)
2. user login is required (with the following level):
   i. Guest: no login required, can only search and view available flight information
   ii. Passenger: Guest functions as well as add/edit Passenger's information, search and apply for tickets, view and print tickets
   iii. Manager: Passenger functions as well as create/remove Passengers and accept ticket payments.
   iv. Admin: top level and will be created when the database is created and have all the rights of a Manager
3. public page for search and display of flights (no login required)
4. database structure with the following entities:
   i. Plane will have a type, size, number of seats, seat numbers seat numbers can be implemented whichever way you want
   ii. Airport will have airport code (IATA Airport Code ↗), city name, city code (three letter code)
   iii. Flight Path will have an ID*, departure airport, arrival airport* iv. Flight will have an ID, time, date, *plane*, *flight path*, flight status (such as 'delayed', 'departed', 'landed', 'scheduled', more at FlightView.com ↗)
   v. Passenger will have:
   Required: ID, title, full name, date of birth, staff/passenger
   Option fields: passport number, nationality, address, phone number, email vi.
   Ticket will have an ID*, passenger*, *flight*, seat number, baggage, payment status, price
5. a log file containing changes from every user
6. function to print a ticket once fully paid

2. Oreti Beach 5K Run

Project Brief

The city council of Invercargill has come to you for an application system for their annual "Oreti Beach 5K Run". The run is every Wednesday starting in October for eight weeks. Registration is open as soon as the application system is up and running. Certificates are given when the event is finished to registered runners who ran in more than five runs.

The organiser currently have a sensor mat and 1000 RFID anklet tags. The tags are numbered from 1 to 1000. They currently reserve the first 600's for registered runners and 601-1000 are for on-the-day signups. Runners who sign up on the day will only need to have their names recorded to be put on the weekly result sheet. Results are made available at noon the following day after the race. The organiser would like to have a user friendly interface to load result data, on-the-day signups and retrieve and change runner details.

The residents of Invercargill are very tech-savvy and would like to have a system set up to register, view their results and print their certificate. They would also like to have a 'group' account where they can manage multiple runners instead of having multiple accounts (e.g. running clubs, families). Some running enthusiasts would like a feature where they can set personal goals such as beat the 25 minutes mark or be the first 50 to finish.

Requirements from Client

1. User login with the following levels (the functionalities are cumulative):
    i. Guest: no login required, can search and view result information, and register an account
    ii. Runner/Group Account: add/edit/remove existing runner accounts, set personal goals, view and print certificates
    iii. Organiser Account: open registration, load results after each run, load on-the-day signups, create certificates, backup database
    iv. Admin Account: add/remove accounts, assign security levels to accounts
2. Front page for guests with login/logoff function
3. Easy to use application interface
4. 'Personal Goal' feature with at least three different goals
5. Backup system

Requirements for the Project

1. Working application that meets all requirement from the client
2. Model-View separation (2 schemas: database and interface)
3. Bonus feature in the Model Schema that hasn't been taught/shown in class
4. Bonus feature in the View Schema that hasn't been taught/shown in class

Submission

Report submission containing:

- a cover page with marking schedule
- Introduction
- one page describing listing members and their assigned tasks for the entire project
- entity relationship diagram
- manual of how to use the application
- interface design
- Conclusion

File submission (BlackBoard) including:

- both schema files (Model and View)
- data files (from backup)
- used images in the application

Class presentation (10 minutes max) with:

- an introduction of team members and roles
- a demonstration of running application including bonus features
- live demonstration of a random task given by the tutor
- 2 minutes Q&A session

NAMES: _____

☐ This assignment is completely my own or my team's original work.

STUDENT SIGNATURE: _____

Marking Schedule

Section	Components	Marks
Database	Classes (Flight, Passengers, Tickets, …)	5
	Properties (attributes/references)	5
	Relationships	5
	Encapsulation (full)	5
	Methods	5
	Collections	5
	Comments/Layout	5
Report and files	Complete reports and files	10
	Extra feature (using functions not covered in class)	5
	Total	50
Interface	Main page	5
	Login function (level of access)	5
	Menus function	5
	Search function	5
	Printing ticket	5
	Backup (user log file)	5
	Form layout/formatting	10
	Extra content (using controls not covered in class)	5
Presentation	Live demonstration of a random task Good handling of Q&A	5
	Total possible points	100

Comments