ILLINOIS INSTITUTE OF TECHNOLOGY DEPARTMENT OF COMPUTER SCIENCE
CS 425 DATABASE DESIGN PROJECT:
SUNSATIONAL VACATION TRAVEL AGENCY

Group 6
Test Plan Documentation

Members
Muntaser Khan (Team Leader)
Victor Obiahu
Obinna Uzondu
Subash Luitel

Demonstration day: 5/8/2013

Requirements	Test Paths/Steps	Expected results	Generated results
Sunsational Vacations needs to store and query the demographic information of its clients. These are: a client identifier, phone number, email address, first name, last name, date of birth, zip code, and gender	The database schema was designed with that Client schema having its clientID as a primary key, and other attributes such as phone number, email, address, first name, last name, DOB, gender, zipcode, and gender which was set to non-nullable	It is expected that all columns containing candidate keys for the client table are non nullable and unique.	All columns containing candidate keys were non nullable and unique.
Sunsational Vacations also needs to store information on its agents. For each agent the following are stored: an agent identifier, phone number, email address, first name, last name, date of birth, zip code, and gender	The database schema was designed such that the Agent table had an agentID as a primary key, a phone number, email address, first name, last name, DOB, gender, zipcode and gender with non-nullable input.	It is expected that all columns of the Agent table containing candidate keys are non nullable and unique	All columns of the Agent table containing candidate keys were non nullable and unique
Additionally, Sunsational Vacations needs to store each agent's position within the company. Each agent is either a "travel agent" or an "agent manager". An agent manager can do anything that a travel agent can, and has additional abilities as well (as explained	It is required that each agent in the database table performed the function of either agent manager or travel agent. The agent manager was either assigned identifiers to show what type of agents	It is expected that every agent in the Agent database table functions either as an agent manager or a travel agent. The agent manager is allowed duality of function.	All agents in the table were either travel agents or agent managers. Their roles were dictated by their identifiers in a specified column called Agent type

later)			
Sunsational Vacations has a predefined list of destination cities (in various countries) where its clients may travel to. Additional cities may be added at any time by an agent manager. Within each destination city, Sunsational Vacations lists at least one resort (possibly many), each of whose information must be stored. This includes: the resort's name, its city and country, address, and telephone number. Also, the price of one night's stay per person in the resort hotel must be stored; each hotel may have several room types with different prices. Of course, prices will vary among the different resorts as well	The database schema is designed in a manner where the resort table had resort_id as its primary key, and other attributes such as resort_name, city, country, address, and phoneNo. Its unique key was a combination of resort_name, city, country and address. The resort table is related to the room_type table throught the resort_id which is a foreign key in table Room_type. Here the price of one night stay per person is stored based on the type of room the client chooses.	It is required that all columns in the Resort table are non-nullable and must be filled. However, in the case of the Room_type table, only the price attribute is non nullable since the other attributes are reference of attributes in tables Resort and Room. It also expected that the price varied across resorts.	The price varied across resorts. All column in the Resort table were indeed non nullable as expected.
Sunsational Vacations also has amenities at its resorts which are offered free to guests.	The table containing the amenities is designed to contain unique amenities with	It is required an agent manager from the Agent Lookup table should have the ability to delete	In the Java Server webpage there is a drop down indicating the

However, not all amenities are offered at all resorts. Thus, a list of which amenities are available at each resort must be maintained. The list of possible amenities is: gym, wireless internet, spa, kayaks, same-day laundry/dry cleaning, day care facilities, and entertainment. This list may be added to or deleted from at any time by an agent manager	amenity description as a candidate key. For each amenity available there is a resort_id which is a foreign key referencing from the Resort table and	or add amenities at any time	agentIDs and the amenities present in each resort. The delete and insert gui's are present but currently, only delete is functional and only for the agent page.
For each resort, we maintain a score of how luxurious the resort is. Thus, we keep a "Sun Rating", where one sun indicates a value resort, two suns indicate a mid-range resort, and three suns indicate a high-end resort.	It is a table that is accessible to agents in the database schema. There is a Resort_rating table containing a foreign key resort_id which is referenced from the resort table and a rate_id which serves as a candidate key. The rate_id ranges from 1 to 3 which indicates 1 as a low rating, 2 as an average rating and 3 as a high rating.	It is expected that agents can rate any resort in any location at any time	There is full functionality as agents are able to rate resorts accordingly at any time.
The database must also store client bookings, keeping all the relevant information about the trip (which resort, the booking date, the arrival and departure	This is handled by a booking table which contains a bookingID that serves as a primary key. It also contains agentID, Room_type_id	It is expected that a an agent can go through the Booking Look up table and gather the necessary information pertaining to the clientID, the arrival and	There is full implementation and functionality of the Booking table as agents are able to keep the

dates, the room type reserved, and the agent who booked the trip)	ClientID and ResortID that are foreign key being referenced from the Agent, Room_type, Client and Resort tables respectively. Necessary information like arrival and departure date are tracked in this table. All the columns are non-nullable	departure dates of the client along with the agent that made the booking.	relevant information about a trip namely: Resort_name, arrival and departure date, room type and booking agent
Should be able to insert, update, and delete all the pre-defined lists (only by an agent manager).	The agent manager has the right to insert, update and delete all the pre-defined lists.	It is expected that only the agent manager is able to perform the aforementioned actions.	Our current functionality only permits the agent manager to delete tuples and only from the agent table. As mentioned earlier, there are gui's for delete and insert but none for update.
Should be able to query any data in database	N/A	N/A	N/A
Should be able to query the data based on certain criteria	Agent and user can search based on query 4-15,but not any additional feature	N/A	N/A
Given a client name as the input, find what trips were booked by that client (give all relevant information about the trip including which agent booked each trip).	This query is handled by a combination of the Booking and Client tables. Both tables are joined on clientID to create one table with has information required to process the query.	Given the client name, it is expected that that agent is able to display information pertaining bookings for that client, as well as the specific agent that booked the trip, this resort it was booked for and what kind of	The full implementation of the functionality required by the query worked perfectly. All necessary information about a trip booked by a client are displayed as expected.

		room was selected. Also the booking date , arrival date and departure dates are expected to be displayed.	
Given the name of a resort (selected by the user), what is the Sun Rating of that resort? Find the total revenue from hotel room bookings (sum of all bookings' nightly rates for all nights plus all inclusive fees), for a given agent, across all agents, or for a given country or city.	The database is set up in a way that the user can find out the sun rating a particular resort given its name. This is accomplished by joining the Resort table with the Resort_Rating table on the Resort_ID attribute which is a foreign key on the Resort_Rating table.	It is expected that the user is able to query the database for information about the rating of a particular resort based on its name.	There is full functionality for this feature.
Which resorts offer all possible amenities?	The database is setup such that the resorts that offers all amenities is available to the user. This information is made available by joining the Resort table with the Amenity table on the resort_ID attribute.	Its is expected that the user can view the resort that offers all amenities in one clear table	There is full functionality for this feature.
List the number of resorts in a given city or country.	The database is setup such that the user can find out information about the number of resorts in a given city.	It is expected that the user can view the number of resorts for a given city	There is full functionality for this feature.
For a client, list all resorts which that individual has vacationed at, as well the number of days of their trip.	A client's first name and/or last name is entered into the text fields provided and is directed to a page displaying all the	It is expected that the user views all resorts that have been visited as well as the number of days	There is full functionality of this feature.

	resorts he/she has vacation along with the length of stay for each visit.		
List all amenities which contain a certain string (provide by a user) exactly in their description.	A client has the option to search for amenities containing key letters in a string eg searching “bar ” would return every amenity containing the desired string.	It is expected that when a certain string is passed in by a client every amenities that contains the searched string is viewed on a new page	There is full functionality of this feature. eg: searching “ym” in the text field would return the amenity “gym”
List the users who have stayed all-inclusive at any hotel within the last 3 months.	This displays the users who have stayed all inclusive at any hotel within the most recent 3 month time frame	It is expected that considering the current date that the query is made all clients who have stayed all inclusive at the hotel would be displayed on a separate page.	There is full functionality of this feature.
What is the average number of days booked per trip? For a given client?	The agent is able to view the average number of days all clients have spent at resort as well viewing the information for a particular client has stayed at a resort.	It is expected that when a client who has information in the database is queried the average number of days booked per trip is viewed.	There is full functionality of this feature.
Which agent booked the largest number of bookings?	This displays the agent that has made the largest number of bookings	It is expected that the agent who has made the largest number of bookings is displayed accurately.	There is full functionality of this feature
Which clients have booked the largest numbers of trips in a given period? Which have spent	This displays the clients that have booked the largest numbers of trips in a given	It is expected that given the client with the largest number of trips within a queried time period	There is full functionality of both features as the date period query works as

the most money in some period?	period and spent the most money within a particular date range	is dsplayed.	expected and accurately displays the client with largest number of bookings
List all agents and the number of clients who have booked with them in descending order.	This displays agents and the number of clients who have booked from greatest to least	It is expected that when the query is made all the agents who booked clients are displayed in descending order	There is full functionality of this feature.